Journal of applied logics (Print) ISSN 2631-9810
Journal of applied logics (Online) ISSN 2631-9829

Journal of Applied Logics

The IfCoLog Journal of Logics and their Applications

Volume 10 • Issue 5 • November 2023

Available online at
www.collegepublications.co.uk/journals/ifcolog/
Free open access

Journal of Applied Logics - IfCoLog
Journal of Logics and their Applications

Volume 10, Number 5

November 2023

Disclaimer
Statements of fact and opinion in the articles in Journal of Applied Logics - IfCoLog Journal of Logics and their Applications (JALs-FLAP) are those of the respective authors and contributors and not of the JALs-FLAP. Neither College Publications nor the JALs-FLAP make any representation, express or implied, in respect of the accuracy of the material in this journal and cannot accept any legal responsibility or liability for any errors or omissions that may be made. The reader should make his/her own evaluation as to the appropriateness or otherwise of any experimental technique described.

ISBN 978-1-84890-441-5
ISSN (E) 2631-9829
ISSN (P) 2631-9810

College Publications
Scientific Director: Dov Gabbay
Managing Director: Jane Spurr

http://www.collegepublications.co.uk

Editorial Board

Scope and Submissions

This journal considers submission in all areas of pure and applied logic, including:

pure logical systems
proof theory
constructive logic
categorical logic
modal and temporal logic
model theory
recursion theory
type theory
nominal theory
nonclassical logics
nonmonotonic logic
numerical and uncertainty reasoning
logic and AI
foundations of logic programming
belief change/revision
systems of knowledge and belief
logics and semantics of programming
specification and verification
agent theory
databases
dynamic logic
quantum logic
algebraic logic
logic and cognition
probabilistic logic
logic and networks
neuro-logical systems
complexity
argumentation theory
logic and computation
logic and language
logic engineering
knowledge-based systems
automated reasoning
knowledge representation
logic in hardware and VLSI
natural language
concurrent computation
planning

This journal will also consider papers on the application of logic in other subject areas: philosophy, cognitive science, physics etc. provided they have some formal content.

Submissions should be sent to Jane Spurr (jane@janespurr.net) as a pdf file, preferably compiled in LaTeX using the IFCoLog class file.

Contents

ARTICLES

Measuring Inconsistency with the Tableau Method

Jandson S. Ribeiro
Artificial Intelligence Group, University of Hagen, Germany
jandson.ribeiro@fernuni-hagen.de

Matthias Thimm
Artificial Intelligence Group, University of Hagen, Germany
matthias.thimm@fernuni-hagen.de

Abstract

We introduce a novel approach to measure inconsistency in knowledge bases that is based on the *Tableau Method* and derivations of contradictions from a knowledge base. This approach is purely syntactic and differs from previous approaches by neither taking minimal inconsistent sets nor non-classical semantics into account. We develop three concrete measures that take derivations of contradictions into account and investigate their compliance w. r. t. rationality postulates, expressivity, and computational complexity.

1 Introduction

An inconsistency measure $\mathcal{I}$ is a function mapping a knowledge base—e. g. a set of propositional sentences—to a non-negative real value, such that larger values indicate more severe inconsistency in the knowledge base [7, 9, 21]. Considering, e. g., the two knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$ defined via

$$\mathcal{K}_1 = \{a, \neg a, b\} \qquad\qquad \mathcal{K}_2 = \{a \wedge b, \neg a, \neg b\}$$

one can see that both knowledge bases are inconsistent (in the classic-logical sense), but $\mathcal{K}_2$ may be judged "more inconsistent" as it contains contradictory information about both propositional atoms a and b while $\mathcal{K}_1$ has only contradictory information about a. So an inconsistency measure $\mathcal{I}$ focusing on this aspect may give $\mathcal{I}(\mathcal{K}_1) < \mathcal{I}(\mathcal{K}_2)$. The concept of a degree of inconsistency is not easily characterisable through either formal properties or a single measure. In fact, there are many proposals for desirable properties an inconsistency measure should satisfy and many proposals for inconsistency measures that satisfy certain subsets of these properties, see [21] for a survey.

One way to classify inconsistency measures is by differentiating whether they operate on the *formula level* or on the *language level*. The former category is also called the *syntactic approach* while the latter is called the *semantic approach* [9]. Measures belonging to the syntactic approach usually make use of *minimal inconsistent subsets*, i. e., subsets of the knowledge base that are inconsistent but removing any formula renders them consistent. For example, a simple measure is $\mathcal{I}_{\mathsf{MI}}$ [11], which assigns to a knowledge base simply the number of its minimal inconsistent subsets. For the knowledge bases from before we have therefore $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_1) = 1$ and $\mathcal{I}_{\mathsf{MI}}(\mathcal{K}_2) = 2$, since $\{a, \neg a\}$ is the only minimal inconsistent subset of $\mathcal{K}_1$ and $\{a \wedge b, \neg a\}$ and $\{a \wedge b, \neg b\}$ are the minimal inconsistent subsets of $\mathcal{K}_2$. Other measures also take the relationships between minimal inconsistent subsets into account [14] or exploit other notions such as maximal consistent subsets [1], but the commonality of these approaches is that they focus on conflicts between formulae of the knowledge base. On the other hand, measures belonging to the semantic approach focus on conflicts between language components. More precisely, these measures aim at identifying those atoms of the underlying language that are conflicting and they usually employ non-classical and many-valued logics as a tool for that [20]. For example, the measure $\mathcal{I}_c$ [8] assigns to a knowledge base the number of propositional atoms participating in the inconsistency using three-valued paraconsistent semantics. Without going into details, this measure gives $\mathcal{I}_c(\mathcal{K}_1) = 1$ and $\mathcal{I}_c(\mathcal{K}_2) = 2$ as well, as one resp. propositional atoms are participating in the conflicts of $\mathcal{K}_1$ resp. $\mathcal{K}_2$.

In this paper, we propose a different perspective for measuring inconsistency based on derivations of contradictions with *logical calculi*. In fact, we argue that the current distinction between *syntactic* and *semantic* approaches is mislabelled, as our new approach is purely syntactic and does not rely on notions such as minimal inconsistent subsets or maximal consistent subsets, which are actually semantically defined concepts. We consider the *Tableau Method* [17] as a prototypical logical calculus (also called *proof system*) and consider *proofs of contradiction* as a sequence of derivation rules that shows how a logical inconsistency can be derived from the knowledge base syntactically. We use such proofs as measures of inconsistency by assuming that 1.) the existence of many such proofs and 2.) the existence of short proofs indicates a larger degree of inconsistency.

To summarise, the contributions of this paper are as follows:

1. We define three inconsistency measures based on proofs of contradictions (Sec-

tion 4). Our inconsistency measures explore the size and number of minimal tableaux to weigh the inconsistency within a knowledge base.

2. We analyse our measures in terms of rationality postulates (Section 5.1), expressivity (Section 5.2), and computational complexity (Section 5.3). Besides comparing our new measures with the existing rationality postulates, we introduce a new postulate with the objective of identifying redundant information in producing inconsistency, and we show that our measures comply with that postulate. We show that our measures are maximally expressive, in the sense that it produces infinitely many values of inconsistency. As for complexity, due to open problems in the area of *proof complexity*, $\mathsf{EXPSPACE}$ is shown to be the tightest upper bound for various decision problems related to our measures.

Sections 2 and 3 provide the formal background and Section 6 concludes.

2 Preliminaries

Let At be an arbitrary fixed finite set of propositional atoms. We assume that the special symbols $\top, \bot$ (tautology and contradiction, respectively) are always contained in At, i. e., $\top, \bot \in \mathsf{At}$.

Definition 1. *Given a set of propositional atoms At, the* propositional language *$\mathcal{L}(\mathsf{At})$ corresponds to the language generated by the following grammar:*

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi;$$

where $p \in \mathsf{At}$.

As usual, $\neg$ denotes negation, $\wedge$ is conjunction, $\vee$ is disjunction. A *knowledge base* $\mathcal{K}$ w. r. t. a language $\mathcal{L}(\mathsf{At})$ is any finite subset $\mathcal{K} \subseteq \mathcal{L}(\mathsf{At})$. Let $\mathbb{K}(\mathsf{At})$ be the set of all knowledge bases w. r. t. to the language $\mathcal{L}(\mathsf{At})$. For any formula ϕ, let $\mathsf{At}(\phi) \subseteq \mathsf{At}$ be the set of atoms appearing in ϕ. When it is clear from context, we will omit At and simply write $\mathcal{L}$ and $\mathbb{K}$.

Definition 2. *Given a set of propositional atoms At, the* length *of a formula $\phi \in \mathcal{L}(\mathsf{At})$ is given by the function $\mathtt{len} : \mathcal{L}(\mathsf{At}) \to \mathbb{Z}_{\geq 0}$ inductively defined as*

- *if $\varphi \in \mathsf{At}$ then $\mathtt{len}(\varphi) = 1$;*
- $\mathtt{len}(\neg\varphi) = \mathtt{len}(\varphi) + 1$*;*

- $\texttt{len}(\varphi \,\Box\, \psi) = \texttt{len}(\varphi) + \texttt{len}(\psi) + 1$ *for* $\Box \in \{\wedge, \vee\}$.

The size of a set A is denoted by $|A|$. An *interpretation* ω on At is a function $\omega : \mathsf{At} \to \{\mathsf{true}, \mathsf{false}\}$ with $\omega(\top) = \mathsf{true}$ and $\omega(\bot) = \mathsf{false}$. Let $\Omega(\mathsf{At})$ be the set of all interpretations on At. An interpretation ω *satisfies* an atom $a \in \mathsf{At}$, denoted as $\omega \models a$, iff $\omega(a) = \mathsf{true}$. Let $\omega \not\models \psi$ denote that ω does not satisfy a formula ψ. The relation $\models$ is inductively extended to general formulae as usual, that is,

$\omega \models \varphi \wedge \psi$ iff $\omega \models \varphi$ and $\omega \models \psi$

$\omega \models \varphi \vee \psi$ iff $\omega \models \varphi$ or $\omega \models \psi$

$\omega \models \neg\varphi$ iff $\omega \not\models \varphi$.

If $\omega \models \phi$ we also say that ω is a *model* of ϕ. Let $\mathsf{Mod}(\phi)$ denote the set of models of a formula ϕ. A formula $\phi \in \mathcal{L}(\mathsf{At})$ is *entailed* by $\psi \in \mathcal{L}(\mathsf{At})$, denoted by $\psi \models \phi$, if for all $\omega \in \Omega(\mathsf{At})$, $\omega \models \psi$ implies $\omega \models \phi$. Two formulae $\phi, \psi \in \mathcal{L}(\mathsf{At})$ are *equivalent*, denoted by $\phi \equiv \psi$, if both $\phi \models \psi$ and $\psi \models \phi$. Furthermore, two sets of formulae X_1, X_2 are *semi-extensionally equivalent* if there is a bijection $s : X_1 \to X_2$ such that for all $\alpha \in X_1$ we have $\alpha \equiv s(\alpha)$ [18]. We denote this by $X_1 \equiv^s X_2$.

3 The Tableau Method

In general, a *proof system* is a set of schematic inference rules that allows the purely syntactic transformation of formulae. Well-known proof systems are e. g. Frege's propositional calculus [5] and Gentzen-style proof systems [6]. In this section, we review the *Tableau Method* for classical propositional logics [17]. The *Tableau method* is a proof system based on refutation: given a knowledge base $\mathcal{K}$, it constructs a binary tree by applying a sequence of rules until either (i) all the branches of the tree present a contradiction or (ii) no rules can be further applied. In the first case, the knowledge base $\mathcal{K}$ is inconsistent; while in the second case, as long as there is at least one branch free of contradiction, $\mathcal{K}$ is consistent. The constructed tree is referred to as a *tableau*. In the remainder of this section, we review the *set-labelled* variant of the *Tableau Method*, where the constructed tableau is a binary tree in which each node is labelled with a set of formulae.

Definition 3. *A set-labelled tree is a tuple* $T = (N, E, \lambda)$ *where*

- (N, E) *is a tree, s.t* N *is the set of nodes,* $E \subseteq N \times N$ *the set of edges,*
- $\lambda : N \to \mathbb{K}(\mathit{At})$ *is a labelling function.*

$$(\neg\neg_e)\ \frac{\mathcal{K}\cup\{\neg\neg\varphi\}}{\mathcal{K}\cup\{\neg\neg\varphi\}\cup\{\varphi\}}$$

$$(DM_\wedge)\ \frac{\mathcal{K}\cup\{\neg(\varphi\wedge\psi)\}}{\mathcal{K}\cup\{\neg(\varphi\wedge\psi)\}\cup\{\neg\varphi\vee\neg\psi\}} \qquad (DM_\vee)\frac{\mathcal{K}\cup\{\neg(\varphi\vee\psi)\}}{\mathcal{K}\cup\{\neg(\varphi\vee\psi)\}\cup\{\neg\varphi\wedge\neg\psi\}}$$

$$(\wedge_e)\frac{\mathcal{K}\cup\{\varphi\wedge\psi\}}{\mathcal{K}\cup\{\varphi\wedge\psi\}\cup\{\varphi,\psi\}}$$

$$(\vee_e)\frac{\mathcal{K}\cup\{\varphi\vee\psi\}}{\mathcal{K}\cup\{\varphi\vee\psi\}\cup\{\varphi\}\mid\mathcal{K}\cup\{\varphi\vee\psi\}\cup\{\psi\}}$$

Figure 1: Derivation rules for the *Tableau Method*.

The labelling function λ maps each node of the tree to a set of formulae in $\mathcal{L}(\mathsf{At})$. Given a set-labelled tree $T = (N, E, \lambda)$, the children of a node n are given by $\mathsf{children}(n) = \{n' \in N \mid (n, n') \in E\}$, and the leaf nodes of T are given by $\mathsf{leaf}(T) = \{n \in N \mid \mathsf{children}(n) = \emptyset\}$. Moreover, the root of T is given by $\mathsf{root}(T)$.

We will postpone the formal definition of the set-labelled tableau (see Definition 5) until we have all the necessary ingredients. We start by giving an intuition of how the tableau method works. As mentioned above, a tableau, which is a set-labelled binary tree with some further constraints, is constructed by applying a set of non-deterministic *derivation rules*, so several tableaux can exist for a same knowledge base $\mathcal{K}$. The procedure for constructing a tableau works by first creating a tree with only the root node (called a *root tree*), which is labelled with the knowledge base $\mathcal{K}$ itself. This initial root tree is then expanded by applying one of the derivation rules depicted in Fig. 1. When applied, these rules append new nodes to one of the leaf nodes of the tree. In the derivation rules $DM_\wedge$ and $DM_\vee$, DM stands for De Morgan, as these rules correspond to the De Morgan laws. While rules $\neg\neg_e, DM_\wedge, DM_\vee$ and $\wedge_e$ append a single leaf node, rule (5) opens two branches.

Each node is labelled with a set of formulae, and therefore, there might exist more than one possible rule to be applied on such a leaf node, or even more than one choice for a same applicable rule. We define a function σ that exhibits explicitly all the possible extensions for non-branching rules, that is, rules $\neg\neg_e, DM_\wedge, DM_\vee$ and $\wedge_e$. The set of all possible extensions for the branching rule $\vee_e$ is given by the function γ below. The set of all rule names are given by $\mathcal{R}_{TB} = \{\neg\neg_e, DM_\wedge, DM_\vee, \wedge_e, \vee_e\}$.

Definition 4. *Let* $\sigma : \mathcal{R}_{TB} \times \mathbb{K}(\mathsf{At}) \to \mathbb{K}(\mathsf{At})$ *be such that*

1. $\sigma(\neg\neg_e, \mathcal{K}) = \{\mathcal{K} \cup \{\varphi\} \in \mathbb{K}(\mathsf{At}) \mid \neg\neg\varphi \in \mathcal{K}\}$

2. $\sigma(\wedge_e, \mathcal{K}) = \{\mathcal{K} \cup \{\varphi, \psi\} \in \mathbb{K}(\mathsf{At}) \mid \varphi \wedge \psi \in \mathcal{K}\}$

3. $\sigma(DM_\wedge, \mathcal{K}) = \{\mathcal{K} \cup \{\neg\varphi \vee \neg\psi\} \in \mathbb{K}(\mathsf{At}) \mid \neg(\varphi \wedge \psi) \in \mathcal{K}\}$

4. $\sigma(DM_\vee, \mathcal{K}) = \{\mathcal{K} \cup \{\neg\varphi \wedge \neg\psi\} \in \mathbb{K}(\mathsf{At}) \mid \neg(\varphi \vee \psi) \in \mathcal{K}\}$

Let $\gamma : \mathbb{K}(\mathsf{At}) \to \mathbb{K}(\mathsf{At}) \times \mathbb{K}(\mathsf{At})$ *be such that*

$\gamma(\mathcal{K}) = \{(X, Y) \in \mathbb{K}(\mathsf{At}) \times \mathbb{K}(\mathsf{At}) \mid X = \mathcal{K} \cup \{\varphi\}, Y = \mathcal{K} \cup \{\psi\},$ *for some* $\varphi \vee \psi \in \mathcal{K}\}$

Definition 5. *A* tableau *for a knowledge base* $\mathcal{K} \subseteq \mathcal{L}(\mathsf{At})$ *is a binary set-labelled tree* (N, E, λ) *such that*

- $\lambda(r) = \mathcal{K}$*, where* r *is the root node;*
- *for each node* $n \in N$*:*
  1. $\lambda(n) \neq \lambda(n')$*, for all* $n' \in \mathsf{children}(n)$*;*
  2. *if* $\mathsf{children}(n) = \{n_1\}$ *then* $\lambda(n_1) \in \sigma(\varepsilon, \lambda(n))$*, for a derivation rule* $\varepsilon \in \mathcal{R}_{TB}$ *;*
  3. *if* $\mathsf{children}(n) = \{n_1, n_2\}$ *and* $n_1 \neq n_2$ *then* $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$ *or* $(\lambda(n_2), \lambda(n_1)) \in \gamma(\lambda(n))$*.*

Conditions 1 to 3 guarantee that a tableau is generated according to the application of the rules in $\mathcal{R}_{TB}$. Condition 1 is imposed in order to avoid redundant tableaux. Specifically, the application of a rule on a node of a tableau needs to yield children nodes labelled with new formulae. This will become important since we are interested in *minimal* proofs of contradiction. The Greek letter π will be used to denote a tableau.

Example 6. *Consider the inconsistent knowledge base* $\mathcal{K} = \{a \wedge c, \neg a, b \vee d\}$*. Fig. 2 illustrates two tableaux* π_1 *and* π_2 *for* $\mathcal{K}$*. The root node of every tableau is labelled with the knowledge base itself* $\mathcal{K}$*. There are two possible rules to apply at the root node: (i) rule* $\wedge_e$ *creates a single child node with the added sub-formulae* a *and* c *(tableau* π_1*); (ii) rule* $\vee_e$ *creates two children node, one labelled with the sub-formula* b *with* $\mathcal{K}$ *and another with the sub-formula* d *with* $\mathcal{K}$ *(tableau* π_2*).*

If a formula α appears in the leaf node of a tableau π for a knowledge base $\mathcal{K}$, then we say that $\mathcal{K}$ *structurally derives* α, denoted by $\mathcal{K} \vdash \alpha$. For instance, in Example 6 the tableau π_1 has the formula c in its leaf node, therefore $\mathcal{K} \vdash c$.

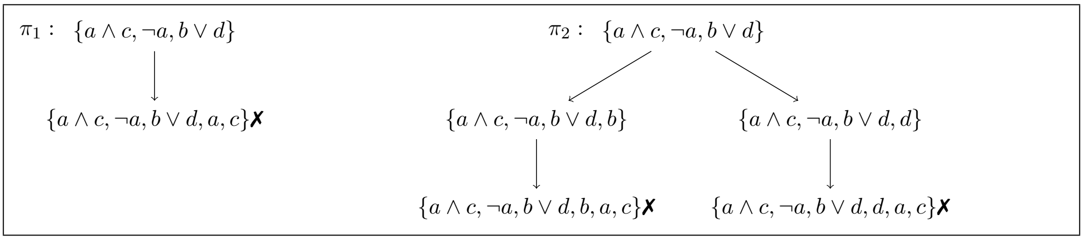


Figure 2: Example of two tableaux for the knowledge base $\mathcal{K} = \{a \wedge c, \neg a, b \vee d\}$.

If a node contains a formula and its negation then we say that such a node has a *clash*. More precisely, if there are formulae $\varphi, \neg\varphi \in \lambda(n)$ then n has a clash. Each leaf node of the tableaux π_1 and π_2 from Example 6 has a clash, as each leaf node has the formula a and its negation $\neg a$. If every leaf node of a tableau has a clash then such a tableau is said to be *closed*. The tableaux π_1 and π_2 from Example 6 are both closed. The set of all closed tableaux for a knowledge base $\mathcal{K}$ is given by $\mathcal{T}_\perp(\mathcal{K})$.

Theorem 7. *[17] A knowledge base $\mathcal{K} \in \mathbb{K}(\mathsf{At})$ is inconsistent iff $\mathcal{T}_\perp(\mathcal{K}) \neq \emptyset$.*

As we are interested in minimal proofs of contradiction, we introduce the notion of a closed tableau being shorter than other closed tableau.

Definition 8. *A closed tableau π is* shorter than a closed tableau π', *denoted as $\pi \preceq \pi'$, iff there is an injection $\tau : \mathsf{leaf}(\pi) \to \mathsf{leaf}(\pi')$ such that $\lambda(n) \subseteq \lambda(\tau(n))$. Given a knowledge base $\mathcal{K}$, a closed tableau $\pi \in \mathcal{T}_\perp(\mathcal{K})$ is* minimal *iff for all $\pi' \in \mathcal{T}_\perp(\mathcal{K})$, if $\pi' \preceq \pi$ then $\pi \preceq \pi'$. The set of minimal tableaux for a given knowledge base $\mathcal{K}$ is given by $\mathcal{T}_\perp^{min}(\mathcal{K})$.*

Intuitively, a closed tableau π is shorter than a tableau π' if each set of formulae that clashes (what are present in the leaf nodes of the tableaux) are subsets of the leaf nodes of π'. For instance, the tableau π_1 from Example 6 is shorter than the tableau π_2 from the same example. We say that a tableau is redundant if two different branches lead to the same clash of formulae labelled on their leaf nodes, as it occurs with the tableau π_2 from Example 6. The injection condition guarantees that redundant tableaux are identified and therefore are not among the minimal tableaux. For instance, the tableau π_1 from Example 6 is minimal, while π_2 is not minimal.

4 Measuring inconsistency via Tableaux

An inconsistency measure is a function $\mathcal{I} : \mathbb{K}(\mathsf{At}) \to \mathbb{R}^{\infty}_{\geq 0}$ that maps each knowledge base $\mathcal{K}$ to a non-negative real number [7, 22]. Intuitively, larger values $\mathcal{I}(\mathcal{K})$ indicate a larger degree of inconsistency in $\mathcal{K}$, while 0 is reserved to indicate the absence of inconsistency.

A closed tableau exemplifies the reasoning effort to detect the presence of an inconsistency and thus gives rise to quantitative measures of inconsistency. The following principles are our main motivation to study measures based on tableaux:

1. If there are more ways to derive inconsistency in a knowledge base $\mathcal{K}$ than there are in a knowledge base $\mathcal{K}'$, then $\mathcal{K}$ should be regarded as more inconsistent than $\mathcal{K}'$. This principle represents a form of monotonicity of inconsistency w. r. t. number of closed tableaux.

2. Smaller closed tableaux indicate a larger degree of inconsistency than larger closed tableaux. The rationale behind this principle can be motivated by the *lottery paradox* [15]: if there are many lottery tickets it is rational to assume for each ticket holder that he will not win and the less tickets there are the less rational this assumption becomes. In the first case, the inconsistency (on the fact that one ticket will win and every ticket holder thinks he will not win) is not that much apparent as in the case of just two tickets. A tableau for the first case would include many more steps to show the inconsistency than in the second case.

Both principles capture the intuition that a knowledge base is more inconsistent if the computational effort to find an inconsistency is low. This is indeed the case if there are many ways to prove inconsistency (e. g. a random method would more likely find a proof) and these proofs are short (as the depth of the search of such an algorithm does not need to be high).

We implement the above principle in the following inconsistency measures:

Definition 9. *The three inconsistency measures are* $\mathcal{I}^{\#} : \mathbb{K}(At) \to \mathbb{R}_{\geq 0}, \mathcal{I}^{\min} : \mathbb{K}(At) \to \mathbb{R}_{\geq 0}$*, and* $\mathcal{I}^{\Sigma} : \mathbb{K}(At) \to \mathbb{R}_{\geq 0}$

$$\mathcal{I}^{\#}(\mathcal{K}) = |\mathcal{T}_{\perp}^{min}(\mathcal{K})|$$

$$\mathcal{I}^{\min}(\mathcal{K}) = \begin{cases} \dfrac{1}{min\{|A| \mid A \in \mathcal{T}_{\perp}^{min}(\mathcal{K})\}} & , if\ \mathcal{T}_{\perp}(\mathcal{K}) \neq \emptyset \\ 0 & otherwise. \end{cases}$$

$$\mathcal{I}^{\Sigma}(\mathcal{K}) = \begin{cases} \sum\limits_{A \,\in\, \mathcal{T}_{\perp}^{min}(\mathcal{K})} \dfrac{1}{|A|} & \textit{if } \mathcal{T}_{\perp}^{min}(\mathcal{K}) \neq \emptyset \\ 0 & \textit{otherwise} \end{cases}$$

The inconsistency measure $\mathcal{I}^{\#}$ focuses on the first principle and simply takes the number of minimal closed tableaux as the degree of inconsistency. The measure $\mathcal{I}^{\min}$ focuses on the second principle and takes the reciprocal size of a minimal closed tableau as the degree of inconsistency. Finally, the measure $\mathcal{I}^{\Sigma}$ combines both principles by summing up the reciprocal sizes of all minimal closed tableaux.

Example 10. *Consider the knowledge bases $\mathcal{K}_1$ and $\mathcal{K}_2$ below:*

$$\mathcal{K}_1 = \{a \wedge c, \neg a, b\} \qquad\qquad \mathcal{K}_2 = \{a \wedge b, c \wedge d, \neg a, \neg d\}.$$

Note that $\mathcal{K}_1$ has only one minimal tableau:

$$\pi = \begin{array}{c} \{a \wedge c, \neg a, b\} \\ | \\ \{a \wedge c, \neg a, b, a, c\} \end{array}$$

Therefore, $\mathcal{I}^{\#}(\mathcal{K}_1) = 1$, and $\mathcal{I}^{\min}(\mathcal{K}_1) = \mathcal{I}^{\Sigma}(\mathcal{K}_1) = 1/2$. For $\mathcal{K}_2$ we have the following two minimal closed tableaux

$$\pi_1 = \begin{array}{c} \{a \wedge b, c \wedge d, \neg a, \neg d\} \\ | \\ \{a \wedge b, a, b, c \wedge d, \neg a, \neg d\} \end{array} \qquad \pi_2 = \begin{array}{c} \{a \wedge b, c \wedge d, \neg a, \neg d\} \\ | \\ \{a \wedge b, c \wedge d, c, d, \neg a, \neg d\} \end{array}$$

Therefore, $\mathcal{I}^{\#}(\mathcal{K}_2) = 2$, $\mathcal{I}^{\min}(\mathcal{K}_2) = 1/2$ and $\mathcal{I}^{\Sigma}(\mathcal{K}_2) = 2/2 = 1$.

In general, our measures take a radically different perspective on inconsistency measurement, which is also illustrated by the fact that these measures do not conform with many postulates proposed for inconsistency measures so far (see Section 5). Our aim with these measures is to investigate a new foundation of inconsistency measurement, i. e., one based on syntactic derivations instead of semantical concepts.

5 Analysis

In this section we conduct an analytical evaluation of our measures, focussing on compliance to rationality postulates, expressivity, and computational complexity.

5.1 Rationality Postulates

Many rationality postulates have been proposed for inconsistency measures, see [21] for a survey. However, many of these postulates are disputed and there is up to now no consensus on which of these postulates are desirable and which are not, see also [3] for a discussion. In fact, there is only one postulate which can be regarded as the defining property of an inconsistency measure $\mathcal{I}$ [10]:

Consistency **(CO)** $\mathcal{I}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is consistent

For all other postulates proposed in the literature, we can find (reasonable) proposals of inconsistency measures that violate these postulates, see [21] for an overview. We compile below the existing rationality postulates from the literature, and we investigate the compliance of our measures with such postulates. For the presentation of the postulates, we will first need the following auxiliary definitions:

Definition 11. *A set $M \subseteq \mathcal{K}$ is a* minimal inconsistent subset *(MI) of $\mathcal{K}$, if $M \models \perp$ and there is no $M' \subset M$ with $M' \models \perp$. Let $\mathsf{MI}(\mathcal{K})$ be the set of all MIs of $\mathcal{K}$. A formula $\alpha \in \mathcal{K}$ is called* free formula *if $\alpha \notin \bigcup \mathsf{MI}(\mathcal{K})$. Let $\mathsf{Free}(\mathcal{K})$ be the set of all free formulae of $\mathcal{K}$.*

Definition 12. *A formula $\alpha \in \mathcal{K}$ is a* safe formula *if it is consistent and $At(\alpha) \cap At(\mathcal{K} \setminus \{\alpha\}) = \emptyset$. Let $\mathsf{Safe}(\mathcal{K})$ be the set of all safe formulae of $\mathcal{K}$.*

Let $\mathcal{I}$ be any function $\mathcal{I} : \mathbb{K} \to \mathbb{R}^{\infty}_{\geq 0}$, $\mathcal{K}, \mathcal{K}' \in \mathbb{K}$, and $\alpha, \beta \in \mathcal{L}(\mathsf{At})$. The rationality postulates for inconsistency measure in the literature, see [20] for a survey on the subject, are:

Normalization **(NO)** $0 \leq \mathcal{I}(\mathcal{K}) \leq 1$

Monotony **(MO)** If $\mathcal{K} \subseteq \mathcal{K}'$ then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K}')$

Free-formula independence **(IN)** If $\alpha \in \mathsf{Free}(\mathcal{K})$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$

Dominance **(DO)** If $\alpha \not\models \perp$ and $\alpha \models \beta$ then $\mathcal{I}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}(\mathcal{K} \cup \{\beta\})$

Safe-formula independence **(SI)** If $\alpha \in \mathsf{Safe}(\mathcal{K})$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$

Super-Additivity **(SA)** If $\mathcal{K} \cap \mathcal{K}' = \emptyset$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') \geq \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$

Penalty **(PY)** If $\alpha \notin \mathsf{Free}(\mathcal{K})$ then $\mathcal{I}(\mathcal{K}) > \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$

MI-separability **(MI)** If $\mathsf{MI}(\mathcal{K} \cup \mathcal{K}') = \mathsf{MI}(\mathcal{K}) \cup \mathsf{MI}(\mathcal{K}')$ and $\mathsf{MI}(\mathcal{K}) \cap \mathsf{MI}(\mathcal{K}') = \emptyset$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') = \mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$

MI-normalization **(MN)** If $M \in \mathsf{MI}(\mathcal{K})$ then $\mathcal{I}(M) = 1$

Attenuation **(AT)** $M, M' \in \mathsf{MI}(\mathcal{K})$ and $|M| > |M'|$ implies $\mathcal{I}(M) < \mathcal{I}(M')$

Equal Conflict **(EC)** $M, M' \in \mathsf{MI}(\mathcal{K})$ and $|M| = |M'|$ implies $\mathcal{I}(M) = \mathcal{I}(M')$

Almost Consistency **(AC)** Let $M_1, M_2, \ldots$ be a sequence of minimal inconsistent sets M_i with $\lim_{i\to\infty} |M_i| = \infty$, then $\lim_{i\to\infty} \mathcal{I}(M_i) = 0$

Contradiction **(CD)** $\mathcal{I}(\mathcal{K}) = 1$ if and only if for all $\emptyset \neq \mathcal{K}' \subseteq \mathcal{K}$, $\mathcal{K}' \models \perp$

Free Formula Dilution **(FD)** If $\alpha \in \mathsf{Free}(\mathcal{K})$ then $\mathcal{I}(\mathcal{K}) \geq \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$

Irrelevance of Syntax **(SY)** If $\mathcal{K} \equiv^s \mathcal{K}'$ then $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K}')$

Exchange **(EX)** If $\mathcal{K}' \not\models \perp$ and $\mathcal{K}' \equiv \mathcal{K}''$ then $\mathcal{I}(\mathcal{K} \cup \mathcal{K}') = \mathcal{I}(\mathcal{K} \cup \mathcal{K}'')$

Adjunction Invariance **(AI)** $\mathcal{I}(\mathcal{K} \cup \{\alpha, \beta\}) = \mathcal{I}(\mathcal{K} \cup \{\alpha \wedge \beta\})$

As mentioned above, the postulate CO addresses the basic property of an inconsistency measure to differentiate between consistent and inconsistent knowledge bases. The postulate NO expresses that the degree of inconsistency is a relative notion that is normalized in the unit interval. MO states that adding information can only increase the degree of inconsistency. IN states that adding free formulae cannot change the degree of inconsistency and DO states that substituting a formula with a semantically weaker version cannot increase the degree of inconsistency. For a discussion on the rationale of the other postulates, see [20].

It is important to stress that there is no consensus about which postulates should be satisfied or which ones should not. However, there are scenarios in which some of the postulates are clearly unsuitable. This is the case of the following postulates: IN, PY, DO, SA, MN, CD, MI, AT, EC, EX, SY and AI. We explain below why each one of such postulates is not adequate under our principles of measuring inconsistency. In fact, none of our measures satisfy these postulates.

- IN: It states that the removal of a free formula does not decrease the inconsistency degree of a knowledge base. Although this intuition might seem plausible at a first glance, it is counter-intuitive under our second principle of inconsistency degree. Let $\mathcal{K} = \{(a \vee b) \wedge (a \vee \neg b), \neg b\}$, and $\mathcal{K}' = \{(a \vee b) \wedge (a \vee \neg b), \neg b, a\}$. Observe that a is free in $\mathcal{K}'$, but the presence of a in $\mathcal{K}'$ makes it much easier to prove the inconsistency of $\mathcal{K}'$ than in $\mathcal{K}$: to prove the inconsistency of $\mathcal{K}$, one needs to take the case distinction of both disjunctive formulae $a \vee b$ and $\neg a \vee b$; while for $\mathcal{K}'$ the proof of inconsistency is much easier because only the case distinction of $\neg a \vee b$ is necessary due to the presence of a. Therefore, free formulae should indeed be considered for assessing the degree of inconsistency in a knowledge base. Therefore, under our second principle IN becomes undesirable.

- PY: this postulate is the dual of IN, removing free-formulae should strictly reduce the inconsistency degree. Analogous to our reasons against IN, as adding free formulae does not necessarily contributes to augmenting the inconsistency degree, removing them should not contribute to making it less inconsistent either.

- DO: According to this postulate, stronger formulae can only make a knowledge base more inconsistent than weaker formulae. This postulate is in conflict with our second principle of inconsistency. To illustrate this, consider the knowledge base $\mathcal{K} = \{\neg a \wedge \neg b\}$, and the formulae $\alpha = c \wedge (a \vee b) \wedge (a \vee \neg b)$, and a. Observe that $\alpha \models a$. It is much easier to prove that $\mathcal{K}_1 = \mathcal{K} \cup \{a\}$ is inconsistent than to prove that $\mathcal{K}_2 = \mathcal{K} \cup \{\alpha\}$ is inconsistent, because for the former the contradiction is evident, while for the latter one needs to consider the case distinction due to the disjunction $a \vee b$. According to our principle, the knowledge base $\mathcal{K}_1$ should be more inconsistent than $\mathcal{K}_2$, opposed to DO.

- SA: This postulate imposes a strict form of monotonicity. It states that if two knowledge bases share no formulae, then their union present an inconsistency degree equal to or higher than the sum of their individual inconsistency degrees. However, this should not be taken as a rule. According to our first principle, the inconsistency degree of a knowledge base should be directly proportional to the number of minimal tableaux. It turns out that the union of knowledge bases does not accumulate their minimal tableaux. Consider, for example, the knowledge bases $\mathcal{K}_1 = \{a \wedge (a \wedge \neg a)\}$ and $\mathcal{K}_2 = \{a, \neg a\}$. Each of them presents only one minimal proof of inconsistency. Observe that, in both knowledge bases, the cause of inconsistency is the same: a and $\neg a$. For the knowledge base $\mathcal{K}_1$, we achieve this by decomposing the conjunctions, while in $\mathcal{K}_2$, this conflict is evident. Therefore, individually, $\mathcal{K}_1$ and $\mathcal{K}_2$ present inconsistency degree of 1. Thus, according to SA, $\mathcal{K}_1 \cup \mathcal{K}_2$ must have an inconsistency degree of at least 2. However, $\mathcal{K}_1 \cup \mathcal{K}_2$ presents only one minimal proof of inconsistency as well: the explicit conflict a and $\neg a$. Therefore, in all three measures we proposed, we have that $\mathcal{K}_1 \cup \mathcal{K}_2$ presents an inconsistency degree of 1 as well. It is clear that SA does not present a good behaviour for inconsistency measurement.

- MN and CD: The postulate MN states that all minimal inconsistent sets should have the same degree of inconsistency 1, while CD states that if every formula in a knowledge base $\mathcal{K}$ is inconsistent then the inconsistency degree of $\mathcal{K}$ must be 1. Both postulates are very prohibitive, as they do not allow grading neither minimal inconsistent sets nor sets containing only inconsistent formulae. If inconsistency in a minimal inconsistent set is much more apparent than in another minimal inconsistent set, then according to our two principles, it is plausible to grade the first one as more inconsistent than the second one. This argument also applies for bases with only inconsistent formulae. Such postulates, therefore, are too fragile to give a suitable notion of rationality for assessing inconsistencies.

- MI: this postulate says that if one can partition the set of minimal inconsistent subsets of a knowledge base $\mathcal{K}$ into two sets A and B then the inconsistency degree of $\mathcal{K}$ corresponds to the sum of the inconsistency degree of the knowledge base obtained from A and obtained from B. Similar to MN, this postulate disregards that the degree of inconsistency does not depend exclusively on the minimal inconsistent subsets. As our measures resort to minimal proofs, this postulate does not pose any criteria for assessing inconsistencies.

- AT and EC: these postulates state that the degree of minimal inconsistent sets should be graded according to the number of formulae in it. The size of the minimal inconsistent set, however, is not directly connected to the effort of proving that a knowledge base is inconsistent. Indeed, smaller inconsistent sets might present minimal proofs bigger than minimal proofs from greater sets (see proof of AT in Theorem 13, for an example).

- EX, SY and AI: Two bases can be logically equivalent but present different reasons of inconsistency, therefore since we are based on the effort of reasoning to measure inconsistency it is desirable that EX, SY and AI be violated.

	CO	NO	MO	IN	DO	NM	SD	SI	SA	PY	MI	MN	AT	EC	AC	CD	FD	SY	EX	AI
$\mathcal{I}^{\#}(\mathcal{K})$	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
$\mathcal{I}^{\min}(\mathcal{K})$	✓	✓	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗
$\mathcal{I}^{\Sigma}(\mathcal{K})$	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

Table 1: Compliance of $\mathcal{I}^{\min}$ with rationality postulates for inconsistency measures.

For our measures, we obtain the following.

Theorem 13. *The compliance of the measures $\mathcal{I}^{\#}$, $\mathcal{I}^{\min}$, and $\mathcal{I}^{\Sigma}$ with the rationality postulates is as presented in Table 1.*

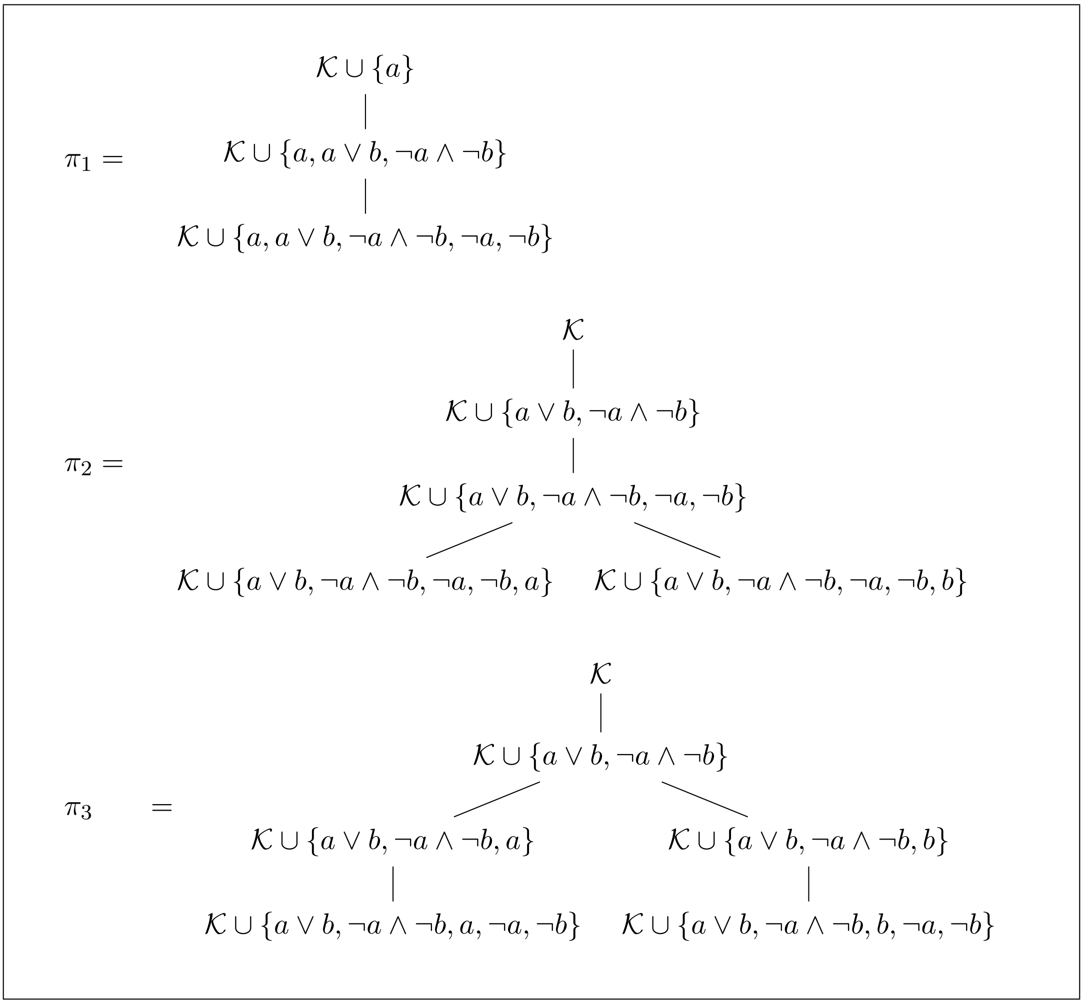


Figure 3: Some minimal tableaux form knowledge bases $\mathcal{K} = \{(a \vee b) \wedge (\neg a \wedge \neg b)\}$ and $\mathcal{K}' = \mathcal{K} \cup \{a\}$.

The measures $\mathcal{I}^{\#}$ and $\mathcal{I}^{\Sigma}$ do not comply with the MO postulate, which is satisfied by several inconsistency measures in the literature. Indeed, according to our two principles, there are cases in which it is plausible to waive MO. For instance, consider the knowledge $\mathcal{K} = \{(a \vee b) \wedge (\neg a \wedge \neg b)\}$. This knowledge base has two minimal closed tableaux: the tableaux π_2 and π_3 depicted at Fig. 3.

By adding a to $\mathcal{K}$, we obtain the knowledge base $\mathcal{K}'$, which has only one minimal closed tableau (the tableau π_1 above). Therefore, according to our first principle,

$$\pi_1 = \begin{array}{c} \{a \wedge b, \neg a\} \\ | \\ \{a \wedge b, \neg a, a, b\} \end{array} \qquad \pi_2 = \begin{array}{ccc} & \{c \vee d\} & \\ \swarrow & & \searrow \\ \{c \vee d, c\} & & \{c \vee d, d\} \end{array}$$

Figure 4: The only tableaux for knowledge base $\mathcal{K}$ and formula α from Example 15.

the inconsistency degree of $\mathcal{K}'$ must be smaller than the inconsistency degree of $\mathcal{K}$. Although this example works as an argument against MO, we argue that there are cases in which some form of monotonicity would still be desirable. For this same example, consider the formula $a \vee c$ and the knowledge base $\mathcal{K}'' = \mathcal{K} \cup \{a \vee c\}$. Observe that $a \vee c$ does not "participate" in making $\mathcal{K}''$ inconsistent, as it does not produce any new minimal proof of inconsistency. Towards this end, according to our both principles, the inconsistency degrees of $\mathcal{K}$ and $\mathcal{K}''$ should be the same. Therefore, for this specific example, some form of monotonicity should be preserved. Indeed, for all the three inconsistency measures we defined, $\mathcal{K}$ and $\mathcal{K}''$ present the same degree of inconsistency. But then, why adding $a \vee c$ should induce a monotonic behaviour, whilst adding a should not? In fact, if we inspect $a \vee c$ and a closer, we will see that a is partially "redundant" while $a \vee c$ is not "redundant". To be more precise, $\mathcal{K} \vdash a$, but $\mathcal{K} \nvdash a \vee c$. Let us properly define our notion of partial redundancy:

Definition 14. *A formula α is* partially-redundant *in $\mathcal{K}$ iff there is some formula φ such that $\mathcal{K} \vdash \varphi$ and $\alpha \vdash \varphi$.*

Example 15. *Consider the knowledge base $\mathcal{K} = \{a \wedge b, \neg a\}$ and the formula $\alpha = (c \vee d)$. Observe that the only common information derived from each consistent subset of $\mathcal{K}$ and α are tautologies. This means that α has no partially-redundant information with $\mathcal{K}$, that is, $\mathcal{K}$ is not partially-redundant. This is because no tableaux of $\mathcal{K}$ shares formulae with any tableaux of $\{\alpha\}$. In fact, $\mathcal{K}$ and α have each one single tableau, as illustrated in Fig. 4, and neither has a single formula in common.*

In the following, we investigate a further (and new) postulate that describe our new approaches and point to their specific advantages. In particular, if we restrict the addition of information to "non-redundant" information our measures do indeed behave monotonically: This monotonicity of non-redundant information is formalised as the

***Non-redundant Monotonicity* (NM)**: If ϕ is not partially-redundant in $\mathcal{K}$ then $\mathcal{I}(\mathcal{K}) \leq \mathcal{I}(\mathcal{K} \cup \{\phi\})$.

The above postulate demands that adding genuinely new information to a knowledge base cannot decrease the degree of inconsistency. Our three measures comply with this demand.

Theorem 16. *The inconsistency measures $\mathcal{I}^{\min}$, $\mathcal{I}^{\#}$ and $\mathcal{I}^{\Sigma}$ satisfy* NM.

In the analyses above, we have shown that our measures do not comply with the postulate MI. This occurs mainly because there is no correspondence between minimal inconsistent subsets and minimal tableaux, as Example 17 and Example 18 below illustrate.

Example 17. *Consider the knowledge base $\mathcal{K} = \{a \wedge c, (\neg a \vee d) \wedge (\neg c \vee d), \neg d\}$ and the following 2 minimal tableaux of this knowledge base:*

$$\pi_1 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \cup \{a, b \wedge c\} \\ | \\ \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d, \neg a\} \quad \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d, d\} \end{array}$$

$$\pi_2 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \cup \{a, b \wedge c\} \\ | \\ \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d, \neg c\} \quad \mathcal{K} \cup \{a, c, \neg a \vee d, \neg c \vee d, d\} \end{array}$$

In Example 17, the knowledge base $\mathcal{K}$ is a minimal inconsistent set and has at least two different minimal tableaux τ_1 and τ_2.

Example 18. *Let $\mathcal{K} = \{a, \neg a, b, \neg b, a \vee b\}$. Observe that this knowledge base has two minimal inconsistent subsets which are $A_1 = \{a, \neg a\}, A_2 = \{b, \neg b\}$. However, this knowledge base has only one minimal tableau which is*

$$\pi_3 = \quad \mathcal{K}$$

In Example 18, the minimal tableaux τ_3 is associated with the minimal inconsistent subsets A_1 and A_2, since the contradictions in the leaf node, which coincides with the root node, regard both A_1 and A_2: $\{a, \neg a\}$ and $\{b, \neg b\}$. Therefore, none of the three measures that we proposed are sensible to this interpretation of the number of sources of conflict. However, we can construct a measure that iteratively removes the sources of inconsistency based on the minimal tableaux, and accumulate the values, until no inconsistency is left. For example, as both A_1 and A_2 are related to τ_3, we can remove A_1 from $\mathcal{K}$ obtaining the knowledge base $\mathcal{K}' = \mathcal{K} \setminus A_1$. We then compute the minimal tableau of $\mathcal{K}'$ which contains only one node labelled with $\mathcal{K}'$. We then remove A_2 from it obtaining a consistent knowledge base. Therefore, in the end, we assign an inconsistency value of 2 to $\mathcal{K}$: since all three measure yield value 1 on both iterations.

5.2 Expressivity

Besides rationality postulates, another (complementary) dimension of evaluating an inconsistency measure is its *expressivity* [19], that is, the number of different inconsistency values a measure can attain on some certain sets of knowledge bases. This evaluation measure has been proposed in order to be able to distinguish trivial measures such as the drastic measure—which assigns 0 to consistent and 1 to inconsistent knowledge bases but still satisfies a reasonable number of rationality postulates—from more "fine-grained" assessments of inconsistency.

Before defining expressivity characteristics we need some further definitions.

$$
\begin{aligned}
\mathbb{K}^v(n) &= \{\mathcal{K} \in \mathbb{K} \mid |\mathsf{At}(\mathcal{K})| \leq n\} \\
\mathbb{K}^f(n) &= \{\mathcal{K} \in \mathbb{K} \mid |\mathcal{K}| \leq n\} \\
\mathbb{K}^l(n) &= \{\mathcal{K} \in \mathbb{K} \mid \forall \phi \in \mathcal{K} : \mathtt{len}(\phi) \leq n\} \\
\mathbb{K}^p(n) &= \{\mathcal{K} \in \mathbb{K} \mid \forall \phi \in \mathcal{K} : |\mathsf{At}(\phi)| \leq n\}
\end{aligned}
$$

Informally speaking, $\mathbb{K}^v(n)$ is the set of all knowledge bases that mention at most n different propositions, $\mathbb{K}^f(n)$ is the set of all knowledge bases that contain at most n formulae, $\mathbb{K}^l(n)$ is the set of all knowledge bases that contain only formulae with maximal length n, and $\mathbb{K}^p(n)$ is the set of all knowledge bases that contain only formulae that mention at most n different propositions each.

Definition 19. *Let $\mathcal{I}$ be an inconsistency measure and $n > 0$. Let $\alpha \in \{v, f, l, p\}$. The* α-characteristic *$\mathcal{C}^\alpha(\mathcal{I}, n)$ of $\mathcal{I}$ w. r. t. n is defined as $\mathcal{C}^\alpha(\mathcal{I}, n) = |\{\mathcal{I}(\mathcal{K}) \mid \mathcal{K} \in \mathbb{K}^\alpha(n)\}|$.*

In other words, $\mathcal{C}^\alpha(\mathcal{I}, n)$ is the number of different inconsistency values $\mathcal{I}$ assigns to knowledge bases from $\mathbb{K}^\alpha(n)$.

The following results show that our new measures are maximally expressive w. r. t. all four expressivity characteristics.

Theorem 20. *For all $n > 0$ and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\#}, \mathcal{I}^{\Sigma}\}$, $\mathcal{C}^v(\mathcal{I}, n) = \mathcal{C}^f(\mathcal{I}, n) = \mathcal{C}^p(\mathcal{I}, n) = \infty$.*

Theorem 21.

1. *For all $n > 1$, $\mathcal{C}^l(\mathcal{I}^{\#}, n) = \infty$.*
2. *For all $n > 3$, and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\Sigma}\}$, $\mathcal{C}^l(\mathcal{I}, n) = \infty$.*

All three measures are maximally expressive. All three measures present infinitely many values for knowledge bases with at least one atomic propositional symbol, or knowledge bases with at least one formula. With respect to the length of the formulae in a knowledge base, the measure $\mathcal{I}^{\#}$ presents infinitely many values for knowledge bases containing formulae with length higher than one, while for the other two measures, for length higher than 3.

5.3 Computational complexity

In the following, we will (briefly) discuss computational complexity issues of our new measures.

Following [23], we consider the following problems. Let $\mathcal{I}$ be some inconsistency measure.

Exact$_\mathcal{I}$	**Input**:	$\mathcal{K} \in \mathbb{K}$, $x \in \mathbb{R}^{\infty}_{\geq 0}$
	Output:	TRUE iff $\mathcal{I}(\mathcal{K}) = x$
Upper$_\mathcal{I}$	**Input**:	$\mathcal{K} \in \mathbb{K}$, $x \in \mathbb{R}^{\infty}_{\geq 0}$
	Output:	TRUE iff $\mathcal{I}(\mathcal{K}) \leq x$
Lower$_\mathcal{I}$	**Input**:	$\mathcal{K} \in \mathbb{K}$, $x \in \mathbb{R}^{\infty}_{\geq 0} \setminus \{0\}$
	Output:	TRUE iff $\mathcal{I}(\mathcal{K}) \geq x$
Value$_\mathcal{I}$	**Input**:	$\mathcal{K} \in \mathbb{K}$
	Output:	The value of $\mathcal{I}(\mathcal{K})$

The computational complexity of our new measures is tightly linked to the general area of *proof complexity* [4]. As there are exponential lower bounds on the size of a minimal tableau [2, 16], we cannot expect to provide membership results of any of the above computational problems to any (deterministic or non-deterministic) complexity class within the polynomial hierarchy. The most precise statement on all our measures we can make is the following.

Theorem 22. *For $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$, $\textsc{Exact}_{\mathcal{I}}$, $\textsc{Upper}_{\mathcal{I}}$, and $\textsc{Lower}_{\mathcal{I}}$ are in EXPSPACE, while $\textsc{Value}_{\mathcal{I}}$ is in FEXPSPACE (the functional variant of EXPSPACE).*

It is possible that the above bound could be improved to EXPTIME as it may not be necessary to explicitly write down every (potential) tableau (but note that whether EXPTIME≠EXPSPACE is also an open question). However, without a proof system that exhibits minimal proofs of polynomial length for all contradictions, EXPTIME is a necessary lower bound. This fact establishes our three measures to be the hardest inconsistency measures among the ones investigated in [23].

6 Summary and Conclusion

In this paper, we proposed novel approaches to measure inconsistency in knowledge bases. Our approaches are based on the notion of minimal closed tableaux, and we analysed the behaviour of these novel inconsistency measures in terms of rationality postulates, expressivity and computational complexity. The central idea of our approaches is to measure inconsistency via measuring *proof complexity*, i. e. the easier it is for a reasoner to detect inconsistency, the larger the inconsistency is to be regarded.

Using tableaux methods for constructing inconsistency measurement is novel, but [13] uses a different notion of *proof* to define an inconsistency measure. There, instead of minimal tableaux a *minimal proof* is a (not necessarily consistent) subset of the knowledge base that entails some formula and inconsistency is measured by appropriately aggregating the number of proofs of complementary literals. However, this measure makes no use of proof systems in our sense and it has also been shown in [20] that it does not satisfy CO and should therefore not be regarded as a meaningful inconsistency measure. Inconsistency measures based on conflicting variables were proposed in [12]. In their measure, the inconsistency value of a knowledge base $\mathcal{K}$ corresponds to the ratio between the conflicting variables and all the variables of $\mathcal{K}$. This focus on variables makes their measure to plateau when the addition/removal of formulae does not change the amount of conflicting variables. Consider, for example, the knowledge bases $\mathcal{K} = \{a \wedge b, \neg a \vee \neg b\}$ and $\mathcal{K}' = \mathcal{K} \cup \{a\}$. As $\mathcal{K}'$ contains more conflicting sources of inconsistencies than $\mathcal{K}$ (two minimal inconsistencies sets against one minimal inconsistent set), it would be rational to assess $\mathcal{K}'$ as more inconsistent than $\mathcal{K}$. However, the measure based on conflicting variables will assess both as equally inconsistent as they contain the same number of conflicting variables. All our three measures will assess both knowledge bases differently.

Our measures provide a new completely syntactical approach to inconsistency measurement that feature maximal expressivity in differentiating inconsistent knowledge bases (see Section 5.2). However, their computational complexity is a significant challenge for their applicability. Future work is about devising (approximate) algorithmic solutions to overcome this barrier.

Acknowledgements. This research is supported by the German Research Association (DFG), project number 465447331.

References

[1] Meriem Ammoura, Yakoub Salhi, Brahim Oukacha, and Badran Raddaoui. On an MCS-based inconsistency measure. *International Journal of Approximate Reasoning*, 80:443–459, 2017.

[2] Noriko H. Arai, Toniann Pitassi, and Alasdair Urquhart. The complexity of analytic tableaux. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing*, pages 356–363. ACM, 2001.

[3] Philippe Besnard. Revisiting postulates for inconsistency measures. In *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA'14)*, pages 383–396, 2014.

[4] Stephen Cook and Phuong Nguyen. *Logical foundations of proof complexity*, volume 11. Cambridge University Press Cambridge, 2010.

[5] G Frege. *Begriffsschrift.* Verlag von Louis Nebert, 1879.

[6] Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische zeitschrift*, 35, 1935.

[7] John Grant. Classifications for inconsistent theories. *Notre Dame Journal of Formal Logic*, 19(3):435–444, 1978.

[8] John Grant and Anthony Hunter. Measuring consistency gain and information loss in stepwise inconsistency resolution. In W. Liu, editor, *Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2011)*, number 6717 in Lecture Notes in Artificial Intelligence, pages 362–373. Springer-Verlag, 2011.

[9] Anthony Hunter and Sébastien Konieczny. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, volume 3300 of *Lecture Notes in Computer Science*, pages 189–234. Springer, 2004.

[10] Anthony Hunter and Sébastien Konieczny. Shapley inconsistency values. In *Proceedings of the 10th International Conference on Knowledge Representation (KR'06)*, pages 249–259. AAAI Press, 2006.

[11] Anthony Hunter and Sébastien Konieczny. Measuring inconsistency through minimal inconsistent sets. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2008)*, pages 358–366, Sydney, Australia, September 2008. AAAI Press, Menlo Park, California.

[12] Saïd Jabbour, Yue Ma, Badran Raddaoui, and Lakhdar Sais. Quantifying conflicts in propositional logic through prime implicates. *Int. J. Approx. Reason.*, 89:27–40, 2017.

[13] Said Jabbour and Badran Raddaoui. Measuring inconsistency through minimal proofs. In *Proceedings of the 12th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, ECSQARU'13, pages 290–301, Berlin, Heidelberg, 2013. Springer-Verlag.

[14] Said Jabbour and Lakhdar Sais. Exploiting mus structure to measure inconsistency of knowledge bases. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI'16)*, pages 991–998, 2016.

[15] Henry E. Kyburg. *Probability and the Logic of Rational Belief.* Wesleyan University Press, 1961.

[16] Neil V Murray and Erik Rosenthal. On the computational intractability of analytic tableau methods. *Logic Journal of the IGPL*, 2(2):205–228, 1994.

[17] Raymond M Smullyan. *First-order logic.* Courier Corporation, 1995.

[18] Matthias Thimm. Inconsistency measures for probabilistic logics. *Artificial Intelligence*, 197:1–24, April 2013.

[19] Matthias Thimm. On the expressivity of inconsistency measures. *Artificial Intelligence*, 234:120–151, February 2016.

[20] Matthias Thimm. Measuring inconsistency with many-valued logics. *International Journal of Approximate Reasoning*, 86:1–23, July 2017.

[21] Matthias Thimm. On the compliance of rationality postulates for inconsistency measures: A more or less complete picture. *Künstliche Intelligenz*, 31(1):31–39, March 2017.

[22] Matthias Thimm. On the evaluation of inconsistency measures. In John Grant and Maria Vanina Martinez, editors, *Measuring Inconsistency in Information*, volume 73 of *Studies in Logic*. College Publications, February 2018.

[23] Matthias Thimm and Johannes Peter Wallner. Some complexity results on inconsistency measurement. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, pages 114–123, April 2016.

Received November 2022

A Proofs for Section 5 (Analysis)

The proofs presented in this appendix include set-labelled tableaux in which the set of formulae labelled to the nodes is significantly big. For clarity, we will depict tableaux in a concise way. We will draw the root now with the whole knowledge base. However, for other nodes, instead of presenting the whole set of formulae labelled in a node, we draw only the fresh formulae added from the parent node to the child node (as illustrated in Fig. 5). The set of formulae labelled on a node n can be inferred by taking the union of the formulae from the root node to n.

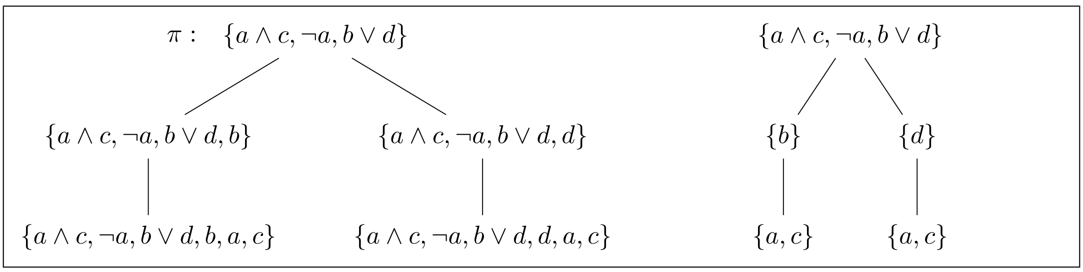


Figure 5: On the left, a closed set-labelled tableau π for the knowledge base $\mathcal{K} = \{a \wedge c, \neg a, b \vee d\}$. On the right, the concise way of representing the set-labelled tableaux π.

Lemma A.1. *If $\pi = (N, E, \lambda)$ is a tableau for a knowledge base $\mathcal{K}$ then $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$ for every $n \in N$.*

Proof. By induction on the level of n.

Base: $level(n)$ is zero, that is, n is the root. Then $\lambda(n) = \mathcal{K}$. Thus, $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.

Induction Hypothesis (IH): if $level(n') < level(n)$ then $\mathsf{At}(\lambda(n')) = \mathsf{At}(\mathcal{K})$.

Induction Step: $level(n) > 1$. Thus, n has some parent n', and either (1) $\mathsf{children}(n') = \{n\}$ or (2) $\mathsf{children}(n') = \{n, n_2\}$ with $n \neq n_2$:

(1) $\mathsf{children}(n') = \{n\}$. Then, $\lambda(n) = \lambda(n) \cup A$, where one of the following cases hold:

1. $A = \{\varphi\}$, with $\neg\neg\varphi \in \lambda(n)$. Thus, as $\mathsf{At}(\varphi) = \mathsf{At}(\neg\neg\varphi)$, we get that $\mathsf{At}(\lambda(n')) = \mathsf{At}(\lambda(n))$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.

2. $A = \{\varphi, \psi\}$ with $\varphi \wedge \psi \in \lambda(n')$. Thus, as $\mathsf{At}(\varphi \wedge \psi) = \mathsf{At}(\varphi) \cup \mathsf{At}(\psi)$, we get that $\mathsf{At}(\lambda(n')) = \mathsf{At}(\lambda(n))$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.
3. $A = \{\neg\varphi \vee \neg\psi\}$ with $\neg(\varphi \wedge \psi) \in \lambda(n')$. Thus, as $\mathsf{At}(\neg(\varphi \wedge \psi)) = \mathsf{At}(\neg\varphi \vee \neg\psi)$, we get that $\mathsf{At}(\lambda(n')) = \mathsf{At}(\lambda(n))$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.
4. $A = \{\neg\varphi \wedge \neg\psi\}$ with $\neg(\varphi \vee \psi) \in \lambda(n')$. Thus, as $\mathsf{At}(\neg(\varphi \vee \psi)) = \mathsf{At}(\neg\varphi \wedge \neg\psi)$, we get that $\mathsf{At}(\lambda(n')) = \mathsf{At}(\lambda(n))$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.

(2) $\mathsf{children}(n') = \{n, n_2\}$ with $n \neq n_2$. Thus, there is $\varphi \vee \psi \in \lambda(n')$ such that either (a) $\lambda(n) = \lambda(n') \cup \{\varphi\}$ or (b) $\lambda(n) = \lambda(n') \cup \{\psi\}$. Observe that $\mathsf{At}(\varphi) \subseteq \mathsf{At}(\varphi \vee \psi)$ and $\mathsf{At}(\psi) \subseteq \mathsf{At}(\varphi \vee \psi)$. Therefore in either cases (a or b), we get that $\mathsf{At}(n) = \mathsf{At}(n')$. Thus, it follows from HI, that $\mathsf{At}(\lambda(n)) = \mathsf{At}(\mathcal{K})$.

□

Theorem 13. *The compliance of the measures $\mathcal{I}^{\#}$, $\mathcal{I}^{\min}$, and $\mathcal{I}^{\Sigma}$ with the rationality postulates is as presented in Table 1.*

Proof. In the following, we denote by +X a proof that shows that property X is satisfied and by −X a proof that shows that property X is violated.

+CO Let $\mathcal{K}$ be a knowledge base. $\mathcal{K}$ is inconsistent if and only if there is a closed tableau π. Then, $\mathcal{I}^{\#}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is inconsistent. Analogously, $\mathcal{I}^{\min}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is inconsistent; and $\mathcal{I}^{\Sigma}(\mathcal{K}) = 0$ if and only if $\mathcal{K}$ is inconsistent.

NO The measures $\mathcal{I}^{\#}$ and $\mathcal{I}^{\Sigma}$ clearly fail NO, while $\mathcal{I}^{\min}$ satisfies it.

+ By definition $\mathcal{I}^{\min}(\mathcal{K}) = 0$, if $\mathcal{K}$ is consistent, and corresponds $\mathcal{I}^{\min}(\mathcal{K}) = 1/n$, where n is the size of the minimal closed tableaux in $\mathcal{T}_{\perp}(\mathcal{K})$. Therefore, $0 \leq \mathcal{I}^{\min}(\mathcal{K}) \leq 1$.
− Consider the following knowledge base $\mathcal{K} = \{a \wedge \neg a, b \wedge \neg b, c \wedge \neg c\}$. For this knowledge base, there are only three minimal closed tableaux, all of them of size 2. Therefore, $\mathcal{I}^{\#}(\mathcal{K}) = 3$, and $\mathcal{I}^{\Sigma}(\mathcal{K}) = \frac{3}{2} > 1$.

MO The measures $\mathcal{I}^{\#}$ and $\mathcal{I}^{\Sigma}$ clearly fail MO, while $\mathcal{I}^{\min}$ satisfies it.

+ Note that if π is a closed tableau in $\mathcal{K}$ then π is also a closed tableau in $\mathcal{K}'$ for $\mathcal{K} \subseteq \mathcal{K}'$. Therefore, the length of a minimal closed tableau can only decrease when adding information, thus $\mathcal{I}^{\min}$ can only increase.

– Let $\mathcal{K} = \{(a \vee b) \wedge (\neg a \wedge \neg b), a\}$.

There is only one minimal closed tableau for $\mathcal{K}$, which is π_1 below. On the other hand, there are two minimal closed tableau for $\mathcal{K} \setminus \{a\}$, which are π_2 and π_3 below (depicted in the concise form). We have

$$\begin{array}{lcl}
\mathcal{I}^{\#}(\mathcal{K}) = 1 & < & \mathcal{I}^{\#}(\mathcal{K} \setminus \{a\}) = 2 \\
\mathcal{I}^{\Sigma}(\mathcal{K}) = 1/3 & < & \mathcal{I}^{\Sigma}(\mathcal{K} \setminus \{a\}) = 1/5 + 1/6 = 11/30
\end{array}$$

$$\pi_1 = \begin{array}{c}
\mathcal{K} \\
| \\
\{a \vee b, \neg a \wedge \neg b\} \\
| \\
\{\neg a, \neg b\}
\end{array}
\qquad
\pi_2 = \begin{array}{c}
\mathcal{K}\setminus\{a\} \\
| \\
\{a \vee b, \neg a \wedge \neg b\} \\
| \\
\{\neg a, \neg b\} \\
/ \quad \backslash \\
\{a\} \quad \{b\}
\end{array}$$

$$\pi_3 = \begin{array}{c}
\mathcal{K}\setminus\{a\} \\
| \\
\mathcal{K} \cup \{a \vee b, \neg a \wedge \neg b\} \\
/ \qquad \backslash \\
\begin{array}{cc}
\{a\} & \{b\} \\
| & | \\
\{\neg a, \neg b\} & \{\neg a, \neg b\}
\end{array}
\end{array}$$

–IN Consider the counterexample for MO. Recall $\mathcal{K} = \{(a \vee b) \wedge (\neg a \wedge \neg b), a\}$. Observe that a is free, and $\mathcal{I}^{\#}(\mathcal{K}) \neq \mathcal{I}^{\#}(\mathcal{K} \setminus \{a\}))$, $\mathcal{I}^{\min}(\mathcal{K}) \neq \mathcal{I}^{\min}(\mathcal{K} \setminus \{a\}))$ and $\mathcal{I}^{\Sigma}(\mathcal{K}) \neq \mathcal{I}^{\Sigma}(\mathcal{K} \setminus \{a\}))$.

–DO Let $\mathcal{K} = \{\neg a, \neg b, \neg c\}$, and formulae $\alpha = a$ and $\beta = (a \vee b) \wedge (a \vee c)$. Note that $\mathbb{K} \equiv \beta$. The knowledge base $\mathcal{K} \cup \{\alpha\}$ has only one closed tableau which has size 1 ($\pi_1 = \mathcal{K} \cup \{\alpha\}$), while $\mathcal{K} \cup \{\beta\}$ has two closed tableaux (π_1 and π_2 below), both with size 4. Thus, $\mathcal{I}^{\#}(\mathcal{K} \cup \{\alpha\}) = 1$, $\mathcal{I}^{\#}(\mathcal{K} \cup \{\beta\}) = 2$, $\mathcal{I}^{\min}(\mathcal{K} \cup \{\alpha\}) = 1$,

$\mathcal{I}^{\min}(\mathcal{K} \cup \{\beta\}) = 1/4$, $\mathcal{I}^{\Sigma}(\mathcal{K} \cup \{\alpha\}) = 1$, $\mathcal{I}^{\Sigma}(\mathcal{K} \cup \{\beta\}) = 2/4 = 1/2$.

$$\pi_2 = \begin{array}{c} \mathcal{K} \cup \{\beta\} \\ | \\ \mathcal{K} \cup \{(a \vee b), (a \vee c), \beta\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \cup \{a, (a \vee b), (a \vee c), \beta\} \quad \mathcal{K} \cup \{b, (a \vee b), (a \vee c), \beta\} \end{array}$$

$$\pi_3 = \begin{array}{c} \mathcal{K} \cup \{\beta\} \\ | \\ \mathcal{K} \cup \{(a \vee b), (a \vee c), \beta\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \cup \{a, (a \vee b), (a \vee c), \beta\} \quad \mathcal{K} \cup \{c, (a \vee b), (a \vee c), \beta\} \end{array}$$

+SI Let α be a safe-formula in $\mathcal{K}$. From Proposition A.9, we have that α is non-redundant with $\mathcal{K} \setminus \{\alpha\}$. Thus α is consistent and not-redundant in $\mathcal{K} \setminus \{\alpha\}$. Thus, from Theorem A.14, $\mathcal{T}_{\perp}^{min}(K) = \bigcup_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K} \setminus \alpha)} \pi[\alpha]$. This implies that $\mathcal{I}(\mathcal{K}) = \mathcal{I}(\mathcal{K} \setminus \{\alpha\})$, for all three measures.

−SA Let $\mathcal{K} = \{a \wedge (b \wedge \neg b)\}$ and $\mathcal{K}' = \{a, \neg a\}$. Note that both $\mathcal{K}$ and $\mathcal{K}'$ have only one tableau (π_1 below, and $\mathcal{K}'$ also has only one closed tableau which is the tableau π' with only the root node labelled with $\mathcal{K}'$ itself. Moreover, $\mathcal{K} \cup \mathcal{K}'$ has only one tableau: the tableau with only the root node. Thus,

	$\mathcal{K}$	$\mathcal{K}'$	$\mathcal{K} \cup \mathcal{K}'$	$\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$
$\mathcal{I}^{\#}$	1	1	1	2
$\mathcal{I}^{\min}$	1/4	1	1	4/3
$\mathcal{I}^{\Sigma}$	1/4	1	1	4/3

$$\pi_1 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \cup \{a, b \wedge \neg b\} \\ | \\ \mathcal{K} \cup \{a, \neg b, b, b \wedge \neg a\} \end{array}$$

–PY Let $\mathcal{K} = \{a, \neg a, a \wedge b\}$. Observe that $\mathsf{MI}(\mathcal{K}) = \{\{a, \neg a\}, \{a \wedge b, \neg a\}\}$. Thus, $a \wedge b$ is not free. However, both $\mathcal{K}$ and $\mathcal{K}' = \mathcal{K} \setminus \{a \wedge b\}$ have only one minimal closed tableau each: $\mathcal{K}$ and $\mathcal{K}'$, respectively. Thus penalty is violated for all three measures.

–MI Let $\mathcal{K} = \{\neg a, a \wedge b\}$ and $\mathcal{K}' = \{a \wedge b, (\neg a \wedge b) \wedge c\}$. Note that $\mathsf{MI}(\mathcal{K}) = \{\mathcal{K}\}$, $\mathsf{MI}(\mathcal{K}') = \{\mathcal{K}'\}$, and $\mathsf{MI}(\mathcal{K} \cup \mathcal{K}') = \{\mathcal{K}, \mathcal{K}'\}$. Thus, $\mathsf{MI}(\mathcal{K}) \cap \mathsf{MI}(\mathcal{K}') = \emptyset$ and $\mathsf{MI}(\mathcal{K}) \cup \mathsf{MI}(\mathcal{K}') = \mathsf{MI}(\mathcal{K} \cup \mathcal{K}')$. The minimal closed tableau of $\mathcal{K}$ is π_1, the minimal closed tableau of $\mathcal{K}'$ is π_2 and the minimal closed tableau of $\mathcal{K} \cup \mathcal{K}'$ is π_3. All of them are shown below. Thus,

	$\mathcal{K}$	$\mathcal{K}'$	$\mathcal{K} \cup \mathcal{K}'$	$\mathcal{I}(\mathcal{K}) + \mathcal{I}(\mathcal{K}')$
$\mathcal{I}^{\#}$	1	1	1	2
$\mathcal{I}^{\min}$	1/2	1/3	1/2	1/2 + 1/3
$\mathcal{I}^{\Sigma}$	1/2	1/3	1/2	1/2 +1/3

$$\pi_1 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \cup \{a, b\} \end{array} \qquad \pi_2 = \begin{array}{c} \mathcal{K}' \\ | \\ \mathcal{K}' \cup \{c, \neg a \wedge b\} \\ | \\ \mathcal{K}' \cup \{c, \neg a, b\} \end{array} \qquad \pi_3 = \begin{array}{c} \mathcal{K} \cup \mathcal{K}' \\ | \\ \mathcal{K} \cup \mathcal{K}' \cup \{a, b\} \end{array}$$

–MN Let $\mathcal{K} = \{\neg a \wedge (\neg b \wedge \neg c), (a \vee b) \wedge (a \vee c)\}$. Note that $\mathcal{K} \in \mathsf{MI}(\mathcal{K})$. The minimal closed tableaux of $\mathcal{K}$ are π_1 and π_2 below.

Thus, $\mathcal{I}^{\#}(\mathcal{K}) = 2$, $\mathcal{I}^{\min}(\mathcal{K}) = 1/6$ and $\mathcal{I}^{\Sigma}(\mathcal{K}) = 2 \cdot 1/6 = 1/3$.

$$\pi_1 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \cup \{\neg a, \neg b \wedge \neg c\} \\ | \\ \mathcal{K} \cup \{\neg a, \neg b, \neg c, \neg b \wedge \neg c\} \\ | \\ \mathcal{K} \cup \{a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \cup \{a, a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} \qquad \mathcal{K} \cup \{b, a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} \end{array}$$

$\pi_2 =$

$$
\begin{array}{c}
\mathcal{K} \\
| \\
\mathcal{K} \cup \{\neg a, \neg b \wedge \neg c\} \\
| \\
\mathcal{K} \cup \{\neg a, \neg b, \neg c, \neg b \wedge \neg c\} \\
| \\
\mathcal{K} \cup \{a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} \\
\swarrow \qquad \searrow \\
\mathcal{K} \cup \{a, a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\} \quad \mathcal{K} \cup \{c, a \vee b, a \vee c, \neg a, \neg b, \neg c, \neg b \wedge \neg c\}
\end{array}
$$

−AT Let $\mathcal{K} = \{a \wedge (\neg a \wedge \neg b), a, \neg a\}$, $M = \{a \wedge (\neg a \wedge \neg b)\}$ and $M' = \{a, \neg a\}$. Observe that $M, M' \in \mathsf{MI}(\mathcal{K})$ and $|M| < |M'|$. The only closed tableau of M is π_1, and $\pi_2 = M'$ is the only proof of closed tableau of M'. Thus,

	M	M'
$\mathcal{I}^{\#}$	1	1
$\mathcal{I}^{\min}$	1/3	1
$\mathcal{I}^{\Sigma}$	1/3	1

$$
\pi_1 = \quad
\begin{array}{c}
M \\
| \\
M \ \cup \{a, \neg a \wedge b\} \\
| \\
M \ \cup \{a, \neg a, b, \neg a \wedge b\}
\end{array}
$$

−EC Let $\mathcal{K} = \{(a \wedge (b \wedge c)) \wedge (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$. It has only two closed tableaux, π_1 and π_2 below. Thus, $\mathcal{I}^{\#}(\mathcal{K}) = 2$, $\mathcal{I}^{\min}(\mathcal{K}) = 1/7$, and $\mathcal{I}^{\Sigma}(\mathcal{K}) = 1/7$

Below for clarity, we do not draw the whole sets in each node, but instead, only the fresh formulae just added.

$\pi_1 =$

$$\mathcal{K}$$

$$\mathcal{K} \ \cup \{(a \wedge (b \wedge c)), (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$

$$\mathcal{K} \ \cup \{a, b \wedge c, (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$

$$\mathcal{K} \ \cup \{a, b, c, (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$

$$\mathcal{K} \ \cup \{a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\}$$

$$\mathcal{K} \ \cup \{\neg a, a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\} \qquad \mathcal{K} \ \cup \{\neg b, a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\}$$

$\pi_2 =$

$$\mathcal{K}$$

$$\mathcal{K} \ \cup \{(a \wedge (b \wedge c)), (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$

$$\mathcal{K} \ \cup \{a, b \wedge c, (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$

$$\mathcal{K} \ \cup \{a, b, c, (\neg a \vee \neg b) \wedge (\neg a \vee \neg c)\}$$

$$\mathcal{K} \ \cup \{a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\}$$

$$\mathcal{K} \ \cup \{\neg a, a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\} \qquad \mathcal{K} \ \cup \{\neg c, a, b, c, (\neg a \vee \neg b), (\neg a \vee \neg c)\}$$

+-AC The inconsistency measures $\mathcal{I}^{\#}$ and $\mathcal{I}^{\Sigma}$ violates AC.

- $\mathcal{I}^{\#}$. Consider the sequence M_i, $i \in \mathbb{N}$ of minimal inconsistent sets given via

  $$M_i = \{a_1, \ldots, a_i, \neg a_1 \vee (\neg a_2 \vee (\ldots \vee \neg a_i) \ldots)\}$$

  We have $\lim_{i \to \infty} |M_i| = \infty$. Observe that each M_i has only one minimal closed tableau. Thus, $\mathcal{I}^{\#}(M_i) = 1$, which means $\lim_{i \to \infty} \mathcal{I}^{\#}(M_i) = 1$,

– $\mathcal{I}^{\Sigma}$. Consider the sequence M_i, $i \in \mathbb{N}$ of minimal inconsistent sets given via

$$\begin{aligned}
&M_1 = \{\neg a_1, a_1\}\\
&M_2 = \{\neg a_1, \neg a_2, (a_1 \vee a_2) \wedge (a_2 \vee a_1)\}\\
&M_3 = \{\neg a_1, \neg a_2, \neg a_3, (a_1 \vee a_2) \wedge (a_2 \vee a_3) \wedge (a_3 \vee a_1)\}\\
&\dots\\
&M_i = \{\neg a_1, \neg a_2, \dots, \neg a_i, (a_1 \vee a_2) \wedge (a_2 \vee a_3) \wedge \dots \wedge (a_i \vee a_1)\}\\
&\dots
\end{aligned}$$

Each M_i has exactly i minimal closed tableau. The M_1 has one with size one, M_2 has two, each with size 4. For the following ones, we can enumerate their minimal tableaux in the following way. The M_i has $i-2$ minimal tableaux, such that their sizes correspond exactly to the size of the tableaux of M_{i-1}, while the last 2 minimal tableaux have size $i+2$. In summary, (the number between commas represents the size of each tableau).

$$\begin{aligned}
&M_1 = 1\\
&M_2 = 2 \cdot 4\\
&M_3 = 4, 2 \cdot (3+2)\\
&M_4 = 4, (3+2), 2 \cdot (4+2)\\
&M_5 = 4, (3+2), (4+2), 2 \cdot (5+2)\\
&\dots\\
&M_i = 4, (3+2), (4+2), \dots (i-1+2), 2 \cdot (i+2)
\end{aligned}$$

Thus,

$$\begin{aligned}
&\mathcal{I}^{\Sigma}(M_1) = 1\\
&\mathcal{I}^{\Sigma}(M_2) = \frac{2}{4} = \frac{1}{2}\\
&\mathcal{I}^{\Sigma}(M_3) = \frac{1}{4} + \frac{2}{(3+2)}\\
&\qquad \dots
\end{aligned}$$

$$\mathcal{I}^{\Sigma}(M_i) = \frac{1}{4} + \frac{1}{(3+2)} + \frac{1}{(4+2)} + \ldots + \frac{1}{(i-1+2)} + \frac{2}{(i+2)}$$

Note that $\mathcal{I}^{\Sigma}(M_i) < \mathcal{I}^{\Sigma}(M_{i+1})$. Thus, $\lim\limits_{i\to\infty} \mathcal{I}^{\Sigma}(M_i) = \infty$.

+ $\mathcal{I}^{\min}$. Let M_i be a minimal inconsistent set, and π one of its minimal tableau. Observe that, due to the sub-formulae derivation structure of the tableau, $|\pi| \geq |M_i|$. Thus, the bigger is the set, the bigger is the tableau, which means that the smaller is its inconsistent value according to $\mathcal{I}^{\min}$. Therefore, for an infinity sequence of minimal inconsistent sets, if $\lim\limits_{i\to\infty} |M_i| = \infty$, then $\lim\limits_{i\to\infty} \mathcal{I}^{\min}(M_i) = 0$.

$-$CD Let $\mathcal{K} = \{a \wedge \neg a\}$. The knowledge base has only one minimal closed tableau and its size is 2. Thus, $\mathcal{I}^{\min}(\mathcal{K}) = \mathcal{I}^{\Sigma}(\mathcal{K}) = 1/2$. For $\mathcal{I}^{\min}$, let $\mathcal{K}' = \{\neg a \wedge (a \vee b) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (a \vee \neg c)\}$. Observe that $\mathcal{K}'$ has two minimal closed tableaux. Therefore, $\mathcal{I}^{\#}(\mathcal{K}) = 2$.

FD
- $-$ $\mathcal{I}^{\#}$ and $\mathcal{I}^{\Sigma}$. See counterexample for MO.
- + $\mathcal{I}^{\min}$. If follows from MO.

$-$SY Let $\mathcal{K} = \{a, \neg a\}$ and $\mathcal{K}' = \{(a \vee b) \wedge (a \vee \neg b) \wedge (a \vee c) \wedge (a \vee \neg c), \neg a\}$. Note that $\mathcal{K} \equiv^s \mathcal{K}'$ Observe that $\mathcal{K}$ has only one closed tableau which is $\mathcal{K}$, while $\mathcal{K}'$ has two closed tableaux: π_1 and π_2 below. Thus, $\mathcal{I}^{\#}(\mathcal{K}) = 1$, $\mathcal{I}^{\#}(\mathcal{K}') = 2$, $\mathcal{I}^{\min}(\mathcal{K}) = 1$, $\mathcal{I}^{\min}(\mathcal{K}') = 1/7$ and $\mathcal{I}^{\Sigma}(\mathcal{K}) = 1$, $\mathcal{I}^{\Sigma}(\mathcal{K}') = 1/7$. Below for clarity, we do not draw the whole sets in each node, but instead, only the fresh formulae just added.

$$\pi_1 = \quad \begin{array}{c}
\mathcal{K} \\
| \\
\mathcal{K} \cup \{(a \vee b) \wedge (a \vee \neg b),\} \\
| \\
\mathcal{K} \cup \{(a \vee b), (a \vee \neg b)\} \\
\swarrow \qquad\qquad \searrow \\
\mathcal{K} \cup \{a, (a \vee \neg b), (a \vee c) \wedge (a \vee \neg c)\} \qquad \mathcal{K} \cup \{b, (a \vee \neg b)\} \\
\qquad\qquad\qquad\qquad\qquad \swarrow \qquad\qquad \searrow \\
\qquad\qquad\qquad\qquad \mathcal{K} \cup \{a, b, (a \vee \neg b)\} \qquad \mathcal{K} \cup \{b, \neg b, (a \vee \neg b)\}
\end{array}$$

$$\pi_2 = \begin{array}{c} \mathcal{K} \\ | \\ \mathcal{K} \ \cup \{(a \vee b) \wedge (a \vee \neg b), (a \vee c) \wedge (a \vee \neg c)\} \\ | \\ \mathcal{K} \ \cup \{(a \vee b) \wedge (a \vee \neg b), (a \vee c), (a \vee \neg c)\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \ \cup \{a, (a \vee c), (a \vee \neg c)\} \qquad \begin{array}{c} \mathcal{K} \ \cup \{c, (a \vee c), (a \vee \neg c)\} \\ \swarrow \qquad \searrow \\ \mathcal{K} \ \cup \{a, c, (a \vee c), (a \vee \neg c)\} \qquad \mathcal{K} \ \cup \{\neg c, c, (a \vee c), (a \vee \neg c)\} \end{array} \end{array}$$

$-\mathsf{EX}$ See counterexample for SY

$-\mathsf{AI}$ See counter-example for SY

□

The following definition will be useful for proving the following results regarding non-redundant formulae.

Definition A.2. *The sub-structural formulae of a given formula ϕ are defined inductively as*

- $\mathsf{subs}(\varphi) = \{\varphi\}$*, if φ is a literal;*
- $\mathsf{subs}(\varphi \ \Box \ \psi) = \{\varphi \wedge \psi\} \cup \mathsf{subs}(\varphi) \cup \mathsf{subs}(\psi)$*, for $\Box \in \{\wedge, \vee\}$;*
- $\mathsf{subs}(\neg(\varphi \ \Box \ \psi)) = \{\neg(\varphi \ \Box \ \psi)\} \cup \mathsf{subs}(\neg\varphi) \cup \mathsf{subs}(\neg\psi)$*, for $\Box \in \{\wedge, \vee\}$.*

Definition A.3. *Let $\pi = (N, E, \lambda)$ be a tableau for a knowledge base $\mathcal{K}$, we define $\pi[\alpha] = (N, E, \lambda')$ such that $\lambda'(n) = \lambda(n) \cup \{\alpha\}$.*

The tableau $\pi[\alpha]$ stands for a tableau that augments each node of π with the formula φ.

Proposition A.4. *For every knowledge base $\mathcal{K}$, if α is not partially-redundant in $\mathcal{K}$ and π is a tableau for $\mathcal{K}$ then $\pi[\alpha]$ is a tableau for $\mathcal{K} \cup \{\alpha\}$.*

Proof. Let $\pi = (N, E, \lambda)$ be a tableau for $\mathcal{K}$, and α a not partially-redundant formula in $\mathcal{K}$. We will show that $\pi[\alpha] = (N, E, \lambda')$ satisfies all conditions of a tableau:

- $\lambda'(r) = \mathcal{K} \cup \{\alpha\}$, where r is the root of $\pi[\alpha]$. By definition, $\lambda'(r) = \lambda(r) \cup \{\mathbb{K}\}$ and $\lambda(r) = \mathcal{K}$. Thus, $\lambda'(r) = \mathcal{K} \cup \{\alpha\}$.

- Let $n \in N$:

  1. we will show $\lambda'(n) \neq \lambda'(n')$, for all $n' \in \mathsf{children}(n)$. Let $n' \in \mathsf{children}(n)$. As π is a tableau, $\lambda(n) \subset \lambda(n)$. By hypothesis, α is not partially-redundant in $\mathcal{K}$ which means that $\varphi \notin \lambda(w)$, for all $w \in N$. Therefore, $\lambda(n) \cup \{\alpha\} \subset \lambda(n') \cup \{\alpha\}$. By definition, $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ and $\lambda'(n') = \lambda(n') \cup \{\alpha\}$. Therefore, $\lambda'(n) \subset \lambda'(n')$ which means that $\lambda'(n) \neq \lambda'(n')$
  2. assume $\mathsf{children}(n) = \{n_1\}$. We will show that $\lambda'(n_1) \in \sigma(\varepsilon, \lambda'(n))$, for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\}$. As π is tableau for $\mathcal{K}$, we have that $\lambda(n_1) \in \sigma(\varepsilon, \lambda(n))$ for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\} = \{\wedge_e, \neg\neg_e, DM_\wedge, DM_\vee\}$:
     - "$\varepsilon = \neg\neg_e$". Thus,

       $$\lambda(n_1) = \lambda(n) \cup \{\varphi\}, \text{ for some } \neg\neg\varphi \in \lambda(n).$$

       By definition, $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$, and $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ which implies that $\neg\neg\varphi \in \lambda'(n)$ and

       $$\begin{aligned}\lambda'(n_1) &= \lambda(n) \cup \{\varphi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\varphi\}\end{aligned}$$

       By definition, $\sigma(\neg\neg_e, \lambda'(n)) = \{\lambda'(n) \cup \{\psi\} \mid \neg\neg\psi \in \lambda'(n)\}$. Thus, as $\neg\neg\varphi \in \lambda'(n)$, we get $\lambda'(n) \cup \{\varphi\} \in \sigma(\neg\neg_e, \lambda'(n))$ which means $\lambda'(n_1) \in \sigma(\neg\neg_e, \lambda'(n))$.
     - "$\varepsilon = \wedge_e$". Thus,

       $$\lambda(n_1) = \lambda(n) \cup \{\varphi, \psi\}, \text{ for some } \varphi \wedge \psi \in \lambda(n).$$

       By definition, $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$, and $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ which implies that $\varphi \wedge \psi \in \lambda'(n)$ and

       $$\begin{aligned}\lambda'(n_1) &= \lambda(n) \cup \{\varphi, \psi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\varphi, \psi\}\end{aligned}$$

       By definition, $\sigma(\wedge_e, \lambda'(n)) = \{\lambda'(n) \cup \{\varphi', \psi'\} \mid \varphi' \wedge \psi' \in \lambda'(n)\}$. Thus, as $\varphi \wedge \psi \in \lambda'(n)$, we get $\lambda'(n) \cup \{\varphi, \psi\} \in \sigma(\neg\neg_e, \lambda'(n))$ which means $\lambda'(n_1) \in \sigma(\wedge_e, \lambda'(n))$.
     - "$\varepsilon = DM_\wedge$". Thus,

       $$\lambda(n_1) = \lambda(n) \cup \{\neg\varphi \vee \neg\psi\}, \text{ for some } \neg(\varphi \wedge \psi) \in \lambda(n).$$

By definition, $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$, and $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ which implies that $\neg(\varphi \wedge \psi) \in \lambda'(n)$ and

$$\begin{aligned}\lambda'(n_1) &= \lambda(n) \cup \{\neg\varphi \vee \neg\psi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\neg\varphi \vee \neg\psi\}\end{aligned}$$

By definition, $\sigma(DM_\wedge, \lambda'(n)) = \{\lambda'(n) \cup \{\neg\varphi' \vee \neg\psi'\} \mid \neg(\varphi' \wedge \psi') \in \lambda'(n)\}$. Thus, as $\neg(\varphi \wedge \psi) \in \lambda'(n)$, we get $\lambda'(n) \cup \{\neg\varphi \vee \neg\psi\} \in \sigma(DM_\wedge, \lambda'(n))$ which means $\lambda'(n_1) \in \sigma(DM_\wedge, \lambda'(n))$.

– "$\varepsilon = DM_\vee$". Thus,

$$\lambda(n_1) = \lambda(n) \cup \{\neg\varphi \wedge \neg\psi\}, \text{ for some } \neg(\varphi \vee \psi) \in \lambda(n).$$

By definition, $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$, and $\lambda'(n) = \lambda(n) \cup \{\alpha\}$ which implies that $\neg(\varphi \vee \psi) \in \lambda'(n)$ and

$$\begin{aligned}\lambda'(n_1) &= \lambda(n) \cup \{\neg\varphi \wedge \neg\psi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\neg\varphi \wedge \neg\psi\}\end{aligned}$$

By definition, $\sigma(DM_\vee, \lambda'(n)) = \{\lambda'(n) \cup \{\neg\varphi' \wedge \neg\psi'\} \mid \neg(\varphi' \vee \psi') \in \lambda'(n)\}$. Thus, as $\neg(\varphi \vee \psi) \in \lambda'(n)$, we get $\lambda'(n) \cup \{\neg\varphi \wedge \neg\psi\} \in \sigma(DM_\vee, \lambda'(n))$ which means $\lambda'(n_1) \in \sigma(DM_\vee, \lambda'(n))$.

Thus, we conclude that $\lambda'(n_1) \in \sigma(\varepsilon, \lambda'(n))$, for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\}$.

3. assume $\mathsf{children}(n) = \{n_1, n_2\}$ with $n_1 \neq n_2$. We will show that either $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$ or $(\lambda'(n_2), \lambda'(n_1)) \in \gamma(\lambda'(n))$. Since π is tableau for $\mathcal{K}$, we have that $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$ or $(\lambda(n_2), \lambda(n_1)) \in \gamma(\lambda(n))$. Without loss of generality, let us assume that $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$. Thus, there is some $\varphi \vee \psi \in \lambda(n)$ such that

$$\begin{aligned}&\lambda(n_1) = \lambda(n) \cup \{\varphi\} \text{ and} \\ &\lambda(n_2) = \lambda(n) \cup \{\psi\}\end{aligned}$$

By definition, $\lambda'(n) = \lambda(n) \cup \{\alpha\}$, while $\lambda'(n_1) = \lambda(n_1) \cup \{\alpha\}$ and $\lambda'(n_2) = \lambda(n_2) \cup \{\alpha\}$. Thus, $\varphi \vee \psi \in \lambda'(n)$ and

$$\begin{aligned}\lambda'(n_1) &= \lambda(n) \cup \{\varphi\} \cup \{\alpha\} \qquad & \lambda'(n_2) &= \lambda(n) \cup \{\psi\} \cup \{\alpha\} \\ &= \lambda'(n) \cup \{\varphi\} & &= \lambda'(n) \cup \{\psi\}\end{aligned}$$

By definition, $\gamma(\lambda'(n)) = \{(\lambda'(n) \cup \{\varphi'\}, \lambda'(n) \cup \{\psi'\}) \mid \varphi' \vee \psi' \in \lambda'(n)\}$. Thus, as $\varphi \vee \psi \in \lambda'(n)$, we get that

$$(\lambda'(n) \cup \{\varphi\}, \lambda'(n) \cup \{\psi\}) \in \gamma(\lambda'(n)),$$

which implies $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$.
Thus, $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$ or $(\lambda'(n_2), \lambda'(n_1)) \in \gamma(\lambda'(n))$.

□

Proposition A.5. *Let $\mathcal{K}$ be a knowledge base, π a tableau for $\mathcal{K}$ and π' a tableau for $\mathcal{K} \cup \alpha$. If α is not partially-redundant in $\mathcal{K}$ then*

(a) $\lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$, and

(b) if for all node n of π', $\lambda'(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$ then for every formula $\beta \in \lambda'(n) \setminus \{\alpha\}$, $(\mathsf{subs}(\beta) \cap \mathsf{subs}(\alpha) = \emptyset$,

Proof. Let $\mathcal{K}$ be a knowledge base, π a tableau for $\mathcal{K}$ and π' a tableau for $\mathcal{K} \cup \alpha$, and α a formula not partially-redundant in $\mathcal{K}$

(a) $\lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. As α is not partially-redundant in $\mathcal{K}$, we get that $\lambda(n) \cap \mathsf{subs}(\alpha) = \emptyset$. Thus,

$$\begin{aligned}\lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha) &= (\lambda(n) \cup \{\alpha\}) \cap \mathsf{subs}(\alpha)\\ &= (\lambda(n) \cap \mathsf{subs}(\alpha)) \cup (\{\alpha\} \cap \mathsf{subs}(\alpha))\\ &= \emptyset \cup \{\alpha\} = \{\alpha\}.\end{aligned}$$

1. (b) for every formula $\beta \in \lambda'(n) \setminus \{\alpha\}$, $(\mathsf{subs}(\beta) \cap \mathsf{subs}(\alpha) = \emptyset$. The proof is by induction on the level of n

Base: level of n is 0, that is, n is the root node. Thus $\lambda'(n) \setminus \{\alpha\} = \mathcal{K}$. By hypothesis, α is not redudant in $\mathcal{K}$ which means that $\mathsf{subs}(\alpha) \cap \mathsf{subs}(\beta) = \emptyset$, for all $\beta \in \mathcal{K}$.

Induction Hypothesis: for all node n' such that $\mathit{level}(n') < \mathit{level}(n)$, $\mathsf{subs}(\alpha) \cap \mathsf{subs}(\beta) = \emptyset$, for all $\beta \in \lambda'(n') \setminus \{\alpha\}$

Induction Step: $\mathit{level}(n) > 0$. Thus, n has a parent node n', and either (i) $\mathsf{children}(n') = \{n\}$ or (ii) $\mathsf{children}(n') = \{n, n_2\}$

(i) $\mathsf{children}(n')=\{n\}$. By the definition of Tableau, $\lambda'(n) \in \sigma(\varepsilon, \lambda'(n'))$, for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\} = \{\wedge_e, \neg\neg_e, DM_\wedge, DM_\vee\}$:

– "$\varepsilon = \neg\neg_e$". Thus,

$$\lambda'(n) = \lambda'(n') \cup \{\varphi\}, \text{ for some } \neg\neg\varphi \in \lambda'(n')$$

which implies

$$\lambda'(n) = (\lambda'(n') \setminus \{\alpha\}) \cup (\{\varphi\} \setminus \{\alpha\}.$$

As n' is the parent of n, we have that $level(n') < level(n)$. We have two cases: either $\alpha = \varphi$ or $\alpha \neq \varphi$

- $\alpha = \varphi$. Thus, $\lambda'(n) = (\lambda'(n') \setminus \{\alpha\})$. Thus, from IH: $\mathsf{subs}(\alpha) \cap \mathsf{subs}(\beta) = \emptyset$, for all $\beta \in \lambda'(n') \setminus \{\alpha\}$, taht is, $\mathsf{subs}(\alpha) \cap \mathsf{subs}(\beta) = \emptyset$, for all $\beta \in \lambda'(n)$.
- $\alpha \neq \varphi$. Let $\beta \in \lambda'(n) \setminus \{\alpha\}$. Thus, $\beta \in \lambda'(n') \setminus \{\alpha\}$ or $\beta = \varphi$. For the former, it follows from IH tha $\mathsf{subs}(\beta) \cap \mathsf{subs}(\alpha) = \emptyset$. For the latter, recall that $\neg\neg\varphi \in \lambda'(n')$ and that from hypothesis $\lambda'(n') \cap \mathsf{subs}(\alpha) = \{\alpha\}$. Therefore, $\alpha \neq \neg\neg\varphi$ as $\alpha \neq \varphi$ and $\varphi \in \mathsf{subs}(\neg\neg\varphi)$. Thus $\neg\neg\varphi \in \lambda'(n') \setminus \{\alpha\}$, which implies from IH that $\mathsf{subs}(\neg\neg\varphi) \cap \{\alpha\} = \emptyset$. Thus, as $\varphi \in \mathsf{subs}(\neg\neg\varphi)$, we get that $\mathsf{subs}(\varphi) \cap \mathsf{subs}(\alpha) = \emptyset$. Thus, $\mathsf{subs}(\beta) \cap \mathsf{subs}(\alpha) = \emptyset$, as $\beta = \varphi$.

– the other cases are analagous.

(ii) $\mathsf{children}(n') = \{n, n_2\}$. Analogous to the $\wedge_e$ case.

□

Proposition A.6. *Let $\mathcal{K}$ be a knowledge base and α a formula which is not partially-redundant in $\mathcal{K}$. If π is a tableau for $\mathcal{K} \cup \{\alpha\}$, and for all $n \in \pi$, $\mathsf{subs}(\alpha) \cap \lambda(n) = \{\alpha\}$ then there is some tableau π' of $\mathcal{K}$ such that $\pi = \pi'[\alpha]$.*

Proof. Let $\mathcal{K}$ be a knowledge base, α be a formula that is not partially-redundant in $\mathcal{K}$, and $\pi = (N, E, \lambda)$ be a tableau for $\mathcal{K} \cup \{\alpha\}$ such that $\mathsf{subs}(\alpha) \cap \lambda(n) = \{\alpha\}$, for all $n \in \pi$. Let $\pi' = (N, E, \lambda')$ such that $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$. We will show that (a) π' is a tableau for $\mathcal{K}$ and (b) $\pi'[\alpha] = \pi$.

(a) We will show that π' satisfy all the conditions of a tableau. Let r be the root of π', and therefore also the root of π.

- $\lambda'(r) = \mathcal{K}$. By definition, $\lambda(\pi) = \mathcal{K} \cup \{\alpha\}$ and $\lambda'(r) = \lambda(r) \setminus \{\alpha\}$. Thus, $\lambda'(r) = \mathcal{K}$.
- let $n \in N$:

1. let $n' \in \mathsf{children}(n)$. As π is a tableau $\lambda(n) \subset \lambda(n')$. Thus, as α is labelled in both n and n', we have that $\lambda(n) \setminus \{\alpha\} \subset \lambda(n') \setminus \{\alpha\}$ which means $\lambda'(n) \subset \lambda'(n)$. Thus, $\lambda'(n) \neq \lambda'(n')$.
2. assume $\mathsf{children}(n) = \{n_1\}$. We will show that $\lambda'(n_1) \in \sigma(\varepsilon, \lambda'(n))$, for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\}$. As π is tableau for $\mathcal{K}$, we have that $\lambda(n_1) \in \sigma(\varepsilon, \lambda(n))$ for some $\varepsilon \in \mathcal{R}_{TB} \setminus \{\vee_e\} = \{\wedge_e, \neg\neg_e, DM_\wedge, DM_\vee\}$:
   - "$\varepsilon = \neg\neg_e$". Thus,

     $$\lambda(n_1) = \lambda(n) \cup \{\varphi\}, \text{ for some } \neg\neg\varphi \in \lambda(n).$$

     As π is a tableaux, $\lambda(n) \subset \lambda(n_1)$. Thus, as by hypothesis $\alpha \in \lambda(n)$, we get $\varphi \neq \alpha$. Also, observe that $\alpha \neq \neg\neg\varphi$. Otherwise, we would have that $\varphi \in \mathsf{subs}(\alpha)$, and therefore, we would get $\{\alpha, \varphi\} \subseteq \lambda(n_1) \cap \mathsf{subs}(\alpha)$, a contradiction as by hypothesis $\lambda(n_1) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. Thus, we have

     $$\alpha \neq \varphi \text{ and } \alpha \neq \neg\neg\varphi.$$

     By definition, $\lambda'(n_1) = \lambda(n_1) \setminus \{\alpha\}$ which implies

     $$\lambda'(n_1) = (\lambda(n) \cup \{\varphi\}) \setminus \{\alpha\}.$$

     Thus, as $\varphi \neq \alpha$, we get

     $$\lambda'(n_1) = (\lambda(n) \setminus \{\alpha\}) \cup \{\varphi\}$$

     By definition, $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$. Thus,

     $$\lambda'(n_1) = \lambda'(n) \cup \{\varphi\}.$$

     Moreover, as $\neg\neg\varphi \in \lambda(n)$ and $\alpha \neq \neg\neg\varphi$, we get that $\neg\neg\varphi \in \lambda'(n)$. By definition,

     $$\sigma(\neg\neg_e, \lambda'(n)) = \{\lambda'(n) \cup \{\psi\} \mid \neg\neg\psi \in \lambda'(n)\}.$$

     Thus, as $\neg\neg\varphi \in \lambda'(n)$, we get that $\lambda'(n) \cup \{\varphi\} \in \sigma(\neg\neg_e, \lambda'(n))$, which means $\lambda'(n_1) \in \sigma(\neg\neg_e, \lambda'(n))$.
   - "$\varepsilon = \wedge_e$". Thus,

     $$\lambda(n_1) = \lambda(n) \cup \{\varphi, \psi\}, \text{ for some } \varphi \wedge \psi \in \lambda(n).$$

Before we proceed, we need first to show that $\alpha \neq \varphi, \alpha \neq \psi$ and $\alpha \neq \varphi \wedge \psi$. If $\alpha = \varphi \wedge \psi$ then we would have that $\varphi, \psi \in \mathsf{subs}(\alpha)$, and therefore, we would get $\{\alpha, \varphi, \psi\} \subseteq \lambda(n) \cap \mathsf{subs}(\alpha)$, a contradiction as by hypothesis $\lambda(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. It it was that case that $\alpha = \varphi$ then we would have that $\varphi \wedge \psi, \varphi \in \lambda(n)$. This implies that $\varphi \wedge \psi \in \lambda(n) \setminus \{\alpha\}$. Note that $\mathsf{subs}(\varphi \wedge \psi) \cap \mathsf{subs}(\alpha) \neq \emptyset$. However, from Proposition A.5, we have that $\mathsf{subs}(\varphi \wedge \psi) \cap \mathsf{subs}(\alpha) = \emptyset$ a contradiction. Analogously, we get at the same contraction for $\alpha = \psi$. Therefore,

$$\alpha \neq \varphi, \alpha \neq \psi \text{ and } \alpha \neq \varphi \wedge \psi.$$

By definition, $\lambda'(n_1) = \lambda(n_1) \setminus \{\alpha\}$ which implies

$$\lambda'(n_1) = (\lambda(n) \cup \{\varphi, \psi\}) \setminus \{\alpha\}.$$

Thus, as $\alpha \neq \varphi$ and $\alpha \neq \psi$, we get

$$\lambda'(n_1) = (\lambda(n) \setminus \{\alpha\}) \cup \{\varphi, \psi\}$$

By definition, $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$. Thus,

$$\lambda'(n_1) = \lambda'(n) \cup \{\varphi, \psi\}.$$

Moreover, as $\varphi \wedge \psi \in \lambda(n)$ and $\alpha \neq \varphi \wedge \psi$, we get that $\varphi \wedge \psi \in \lambda'(n)$. By definition,

$$\sigma(\wedge_e, \lambda'(n)) = \{\lambda'(n) \cup \{\varphi', \psi'\} \mid \varphi' \wedge \psi' \in \lambda'(n)\}.$$

Thus, as $\varphi \wedge \psi \in \lambda'(n)$, we get that $\lambda'(n) \cup \{\varphi, \psi\} \in \sigma(\wedge_e, \lambda'(n))$, which means $\lambda'(n_1) \in \sigma(\wedge_e, \lambda'(n))$.

– "$\varepsilon = DM''_{\wedge}$. Thus,

$$\lambda(n_1) = \lambda(n) \cup \{\neg\varphi \vee \neg\psi\}, \text{ for some } \neg(\varphi \wedge \psi) \in \lambda(n).$$

As π is a tableaux, $\lambda(n) \subset \lambda(n_1)$. Thus, as by hypothesis $\alpha \in \lambda(n)$, we get $\alpha \neq \neg\varphi \vee \neg\psi$. Also observe that $\alpha \neq \neg(\varphi \wedge \psi)$. Otherwise, we would have that $\neg\varphi \vee \neg\psi \in \mathsf{subs}(\alpha)$, and therefore, we would get $\{\alpha, \neg\varphi \vee \neg\psi\} \subseteq \lambda(n_1) \cap \mathsf{subs}(\alpha)$, a contradiction as by hypothesis $\lambda(n_1) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. Thus, we have

$$\alpha \neq \neg(\varphi \wedge \psi) \text{ and } \alpha \neq \neg\varphi \vee \neg\psi.$$

By definition, $\lambda'(n_1) = \lambda(n_1) \setminus \{\alpha\}$ which implies

$$\lambda'(n_1) = (\lambda(n) \cup \{\neg\varphi \vee \psi\}) \setminus \{\alpha\}.$$

Thus, as $\alpha \neq \neg\varphi \vee \psi$,

$$\lambda'(n_1) = (\lambda(n) \setminus \{\alpha\}) \cup \{\neg\varphi \vee \psi\}$$

By definition, $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$. Thus,

$$\lambda'(n_1) = \lambda'(n) \cup \{\neg\varphi \vee \psi\}.$$

Moreover, as $\neg(\varphi \wedge \psi) \in \lambda(n)$ and $\alpha \neq \neg(\varphi \wedge \psi)$, we get that $\neg(\varphi \wedge \psi) \in \lambda'(n)$. By definition,

$$\sigma(DM_\wedge, \lambda'(n)) = \{\lambda'(n) \cup \{\neg\varphi' \vee \neg\psi'\} \mid \neg(\varphi' \wedge \psi') \in \lambda'(n)\}.$$

Thus, as $\neg(\varphi\wedge\psi)\in\lambda'(n)$, we get $\lambda'(n)\cup\{\neg\varphi\vee\psi\}\in\sigma(DM_\wedge, \lambda'(n))$, which means $\lambda'(n_1) \in \sigma(DM_\wedge, \lambda'(n))$.

– "$\varepsilon = DM_\vee''$. Analogous to case $\varepsilon = DM_\wedge$.

3. let $\mathsf{children}(n)=\{n_1, n_2\}$ with $n_1 \neq n_2$. We will show $(\lambda'(n_1), \lambda'(n_2))\in \gamma(\lambda'(n))$ or $(\lambda'(n_2), \lambda'(n_1)) \in \gamma(\lambda'(n))$. Since π is tableau for $\mathcal{K}$, we have that $(\lambda(n_1), \lambda(n_2)) \in \gamma(\lambda(n))$ or $(\lambda(n_2), \lambda(n_1)) \in \gamma(\lambda(n))$. Without loss of generality, let us assume that $(\lambda(n_1), \lambda(n_2))\in\gamma(\lambda(n))$. Thus, there is some $\varphi \vee \psi \in \lambda(n)$ such that

$$\lambda(n_1) = \lambda(n) \cup \{\varphi\} \text{ and}$$
$$\lambda(n_2) = \lambda(n) \cup \{\psi\}$$

Before we proceed, we need to show that $\alpha \neq \varphi, \alpha \neq \psi$ and $\alpha \neq \varphi\vee\psi$. If $\alpha = \varphi\vee\psi$ then we would have that $\varphi, \psi \in \mathsf{subs}(\alpha)$, and therefore, we would get $\{\alpha, \varphi, \psi\} \subseteq \lambda(n)\cap\mathsf{subs}(\alpha)$, a contradiction as by hypothesis $\lambda(n) \cap \mathsf{subs}(\alpha) = \{\alpha\}$. It it was that case that $\alpha = \varphi$ then we would have that $\varphi\vee\psi, \varphi \in \lambda(n)$. This implies that $\varphi\vee\psi \in \lambda(n)\setminus\{\alpha\}$. Note that $\mathsf{subs}(\varphi \vee \psi) \cap \mathsf{subs}(\alpha) \neq \emptyset$. However, from Proposition A.5, we have that $\mathsf{subs}(\varphi\vee\psi)\cap\mathsf{subs}(\alpha) = \emptyset$ a contradiction. Analogously, we get at the same contraction for $\alpha = \psi$. Therefore,

$$\alpha \neq \varphi, \alpha \neq \psi \text{ and } \alpha \neq \varphi \vee \psi$$

By definition, $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$, while $\lambda'(n_1) = \lambda(n_1) \setminus \{\alpha\}$ and $\lambda'(n_2) = \lambda(n_2) \setminus \{\alpha\}$. Thus, as $\alpha \neq \varphi \vee \psi$ and $\varphi \vee \psi \in \lambda(n)$ we get that $\varphi \vee \psi \in \lambda'(n)$. Moreover, as $\alpha \neq \varphi$ and $\alpha \neq \psi$, we get

$$
\begin{aligned}
\lambda'(n_1) &= (\lambda(n) \cup \{\varphi\}) \setminus \{\alpha\} & \qquad \lambda'(n_2) &= (\lambda(n) \cup \{\psi\}) \setminus \{\alpha\} \\
&= (\lambda(n) \setminus \{\alpha\}) \cup \{\varphi\} & &= (\lambda(n) \setminus \{\alpha\}) \cup \{\psi\} \\
&= \lambda'(n) \cup \{\varphi\} & &= \lambda'(n) \cup \{\psi\}
\end{aligned}
$$

By definition, $\gamma(\lambda'(n)) = \{(\lambda'(n) \cup \{\varphi'\}, \lambda'(n) \cup \{\psi'\}) \mid \varphi' \vee \psi' \in \lambda'(n)\}$. Thus, as $\varphi \vee \psi \in \lambda'(n)$, we get that

$$(\lambda'(n) \cup \{\varphi\}, \lambda'(n) \cup \{\psi\}) \in \gamma(\lambda'(n)),$$

which implies $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$. Thus, $(\lambda'(n_1), \lambda'(n_2)) \in \gamma(\lambda'(n))$ or $(\lambda'(n_2), \lambda'(n_1)) \in \gamma(\lambda'(n))$.

(b) We only have to show that $\lambda_{\pi'[\alpha]}(n) = \lambda(n)$, for all $n \in N$. Let $n \in N$. By definition $\lambda'(n) = \lambda(n) \setminus \{\alpha\}$, and $\lambda_{\pi'[\alpha]}(n) = \lambda'(n) \cup \{\alpha\}$. Thus, $\lambda_{\pi'[\alpha]}(n) = (\lambda(n) \setminus \{\alpha\}) \cup \{\alpha\}$. By hypothesis, $\alpha \in \lambda(n)$, as $\mathsf{subs}(\alpha) \cap \lambda(n) = \{\alpha\}$. Therefore, $\lambda_{\pi'[\alpha]}(n) = \lambda(n)$.

□

Proposition A.7. *Let $\mathcal{K}$ be a knowledge base and α a formula which is not partially-redundant in $\mathcal{K}$. If π and π' are tableaux for $\mathcal{K}$ then: $\pi \preceq \pi'$ iff $\pi[\alpha] \preceq \pi'[\alpha]$*

Proof. Let $\mathcal{K}$ be a knowledge base, α be a formula that is not partially-redundant in $\mathcal{K}$, and π and π' be tableaux for $\mathcal{K}$.

"⇒". Let $\pi \preceq \pi'$. Thus there is an injection $\tau : \mathsf{leaf}(\pi) \to \mathsf{leaf}(\pi')$ such that

$$\lambda_\pi(n) \subseteq \lambda_{\pi'}(\tau(n)) \tag{1}$$

Observe, from the definition of $\pi[\alpha]$ and $\pi'[\alpha]$, that $\mathsf{leaf}(\pi) = \mathsf{leaf}(\pi[\alpha])$, $\mathsf{leaf}(\pi') = \mathsf{leaf}(\pi'[\alpha])$. Therefore, τ is also an injection from the leaf nodes of $\pi[\alpha]$ to the leaf nodes of $\pi'[\alpha]$. We only need to show that, $\lambda_{\pi[\alpha]}(n) \subseteq \lambda_{\pi'[\alpha]}(\tau(n))$, for all $n \in \mathsf{leaf}(\pi[\alpha])$. Let $n \in \mathsf{leaf}(\pi[\alpha])$. From Eq. (1), we have that

$$\lambda(n) \cup \{\alpha\} \subseteq \lambda'(\tau(n)) \cup \{\alpha\}.$$

By definition, $\lambda_{\pi[\alpha]}(n) = \lambda_\pi(n) \cup \{\alpha\}$ and $\lambda_{\pi'[\alpha]}(\tau(n)) = \lambda_{\pi'[\alpha]}(\tau(n)) \cup \{\alpha\}$. Therefore, $\lambda_{\pi[\alpha]}(n) \subseteq \lambda_{\pi'[\alpha]}(\tau(n))$.

- "$\Leftarrow$". Let $\pi[\alpha] \preceq \pi'[\alpha]$. Thus there is an injection $\tau : \mathsf{leaf}(\pi[\alpha]) \to \mathsf{leaf}(\pi'[\alpha])$ such that
$$\lambda_{\pi[\alpha]}(n) \subseteq \lambda_{\pi'[\alpha]}(\tau(n)) \tag{2}$$
Observe, from the definition of $\pi[\alpha]$ and $\pi'[\alpha]$, that $\mathsf{leaf}(\pi) = \mathsf{leaf}(\pi[\alpha])$, $\mathsf{leaf}(\pi') = \mathsf{leaf}(\pi'[\alpha])$. Therefore, τ is also an injection from the leaf nodes of π to the leaf nodes of π'. We only need to show that $\lambda_\pi(n) \subseteq \lambda_{\pi'}(\tau(n))$, for all $n \in \mathsf{leaf}(\pi)$. Let $n \in \mathsf{leaf}(\pi)$.

  By definition, $\lambda_{\pi[\alpha]}(n) = \lambda_\pi(n) \cup \{\alpha\}$ and $\lambda_{\pi'[\alpha]}(\tau(n)) = \lambda_{\pi'}(\tau(n)) \cup \{\alpha\}$. Thus, from Eq. (2) above we get
$$\lambda_\pi(n) \cup \{\alpha\} \subseteq \lambda_{\pi'}(\tau(n)) \cup \{\alpha\} \tag{3}$$
As α is not partially-redundant in $\mathcal{K}$, we have that α is not labelled in any of the tableaux of $\mathcal{K}$. This means that $\alpha \notin \lambda_\pi(n)$ and $\alpha \notin \lambda_{\pi'}(\tau(n))$. This jointly with Eq. (3) implies
$$\lambda_\pi(n) \subseteq \lambda_{\pi'}(\tau(n)).$$

$\square$

Proposition A.8. *If α is not partially-redundant in $\mathcal{K}$ then*
$$\left(\bigcup_{\pi \in \mathcal{T}_\perp^{min}(\mathcal{K})} \pi[\alpha] \right) \subseteq \mathcal{T}_\perp^{min}(\mathcal{K} \cup \{\alpha\}).$$

Proof. Let us suppose for contradiction that there is a $\pi \in \mathcal{T}_\perp^{min}(\mathcal{K})$ such that $\pi[\alpha] \notin \mathcal{T}_\perp^{min}(\mathcal{K} \cup \{\alpha\})$. Thus, there is a $\pi' \in \mathcal{T}_\perp^{min}(\mathcal{K} \cup \{\alpha\})$ such that $\pi' \prec \pi[\alpha]$. This means that there is some leaf nodes $n' \in \pi'$ and $n \in \pi[\alpha]$ such that
$$\lambda_{\pi'}(n') \subset \lambda_{\pi[\alpha]}(n). \tag{4}$$
Observe that $\alpha \in \lambda(m)$, for all node m of every tableau of $\mathcal{K} \cup \{\alpha\}$. This means that,
$$\{\alpha\} \subseteq \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha) \text{ and } \{\alpha\} \subseteq \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$$

Thus, we have two cases: either (i) $\{\alpha\} = \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha)$ or (ii) $\{\alpha\} \subset \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha)$. We get a contradiction in either case:

- (i) $\{\alpha\} = \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha)$. Thus from Proposition A.6, there is a a tableau π_y for $\mathcal{K}$ such that $\pi' = \pi_y[\alpha]$. From Proposition A.7 we get that $\pi_y \prec \pi$ iff $\pi_y[\alpha] \prec \pi[\alpha]$. Thus, as $\pi' = \pi_y[\alpha]$, we get
$$\pi_y \prec \pi \text{ iff } \pi' \prec \pi[\alpha].$$

By hypothesis, $\pi' \prec \pi[\alpha]$ which implies that $\pi_y \prec \pi$. Therefore, $\pi \notin \mathcal{T}_{\perp}^{min}(\mathcal{K})$ which is a contradiction.

- (ii) $\{\alpha\} \subset \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha)$. It follows from Eq. (4) above that $\lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha) \subseteq \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$. Therefore,

$$\{\alpha\} \subset \lambda_{\pi'}(n') \cap \mathsf{subs}(\alpha) \subseteq \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$$

which implies that $\{\alpha\} \subset \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$. This means that $\{\alpha\} \neq \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$. However, from Proposition A.5, we have that $\{\alpha\} = \lambda_{\pi[\alpha]}(n) \cap \mathsf{subs}(\alpha)$ which is a contradiction.

□

Theorem 16. *The inconsistency measures $\mathcal{I}^{\min}$, $\mathcal{I}^{\#}$ and $\mathcal{I}^{\Sigma}$ satisfy* NM.

Proof. It follows directly from MO that $\mathcal{I}^{\min}$satisfies NM. For the other two measures, we prove compliance with NM separately:

- $\mathcal{I}^{\#}$: Let $\mathcal{K}$ be a knowledge base and α a non-redundant formula with $\mathcal{K}$. Therefore, from Proposition A.8, we get that $|\mathcal{T}_{\perp}^{min}(\mathcal{K})| \leq |\mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})|$. Thus, $\mathcal{I}^{\#}(\mathcal{K}) \leq \mathcal{I}^{\#}(\mathcal{K} \cup \{\alpha\})$.
- $\mathcal{I}^{\Sigma}$: Let $\mathcal{K}$ be a knowledge base and α a non-partially-redundant formula with $\mathcal{K}$. Observe that $|\pi| = |\pi[\alpha]|$. Therefore,

$$\sum_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \frac{1}{|\pi|} = \sum_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \frac{1}{|\pi[\alpha]|}$$

which implies

$$\mathcal{I}^{\Sigma}(\mathcal{K}) = \sum_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \frac{1}{|\pi[\alpha]|} \tag{5}$$

Let $X = \bigcup_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \pi[\alpha]$. From Proposition A.8, we get $X \subseteq \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})$. Thus, $\mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\}) = X \cup (\mathcal{K} \cup \{\alpha\} \setminus X)$. Therefore,

$$\begin{aligned}\mathcal{I}^{\Sigma}(\mathcal{K} \cup \{\alpha\}) &= \left(\sum_{\pi \in X} \frac{1}{|\pi|}\right) + \left(\sum_{\pi \in X \setminus \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})} \frac{1}{|\pi|}\right) \\ &= \left(\sum_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K})} \frac{1}{|\pi[\alpha]|}\right) + \left(\sum_{\pi \in X \setminus \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})} \frac{1}{|\pi|}\right).\end{aligned}$$

Thus, from Eq. (5), we get

$$\mathcal{I}^{\Sigma}(\mathcal{K} \cup \{\alpha\}) = \mathcal{I}^{\Sigma}(\mathcal{K}) + \left(\sum_{\pi \in X \setminus \mathcal{T}_{\perp}^{min}(\mathcal{K} \cup \{\alpha\})} \frac{1}{|\pi|} \right).$$

Thus, $\mathcal{I}^{\Sigma}(\mathcal{K} \cup \{\alpha\}) \geq \mathcal{I}^{\Sigma}(\mathcal{K})$.

□

Proposition A.9. *If a formula α is safe within $\mathcal{K}$ then α is not partially-redundant with $\mathcal{K} \setminus \{\alpha\}$.*

Proof. Let π be a tableau for $\mathcal{K} \setminus \{\alpha\}$, and π' a tableau for $\{\alpha\}$, we will show that there is no formula φ that is labelled in both π and π'. From Lemma A.1, we have that $\mathsf{At}(\varphi) \subseteq \mathsf{At}(\mathcal{K} \setminus \{\alpha\})$ and $\mathsf{At}(\psi) \subseteq \mathsf{At}(\alpha)$, for all φ that appears in π and all $\psi \in \pi'$. Thus, as α is safe with $\mathcal{K}$, we have that $\mathsf{At}(\mathcal{K} \setminus \{\alpha\}) \cap \mathsf{At}(\alpha) = \emptyset$, which means $\mathsf{At}(\varphi) \cap \mathsf{At}(\psi) = \emptyset$. Therefore, there is no common formula between π and π', that is, α is not-partially-redundant. □

Proposition A.10. *If a formula α is not partially-redundant with a knowledge base $\mathcal{K}$ and α is consistent then α is safe in $\mathcal{K} \cup \{\alpha\}$.*

Proof. Let us suppose for contradiction that for some knowledge base $\mathcal{K}$ there is a consistent formula α that is not partially-redundant with $\mathcal{K}$, but it is not safe in $\mathcal{K} \cup \{\alpha\}$. First, observe that each propositional atom in α appears in some tableau of α. By hypothesis, α is not safe in $\mathcal{K} \cup \{\alpha\}$, which means there is a formula $\varphi \in \mathcal{K}$ that shares some atomic proposition p with α, that is $p \in \mathsf{At}(\varphi) \cap \mathsf{At}(\alpha)$. But then p appears in some tableau of $\mathcal{K}$ and in some tableau of α which means that α is partially-redundant with $\mathcal{K}$. This contradicts our hypothesis. Therefore, α is safe. □

To prove compliance of our measures with the postulate SI, we will need some extra constructions. First, given a tableau π for a knowledge base $\mathcal{K}$, and a node n of π, we denote by $\mathsf{subT}(n)$ all the nodes of the subtree rooted on n. A node n has two children, say n_1 and n_2, only when such children were obtained by applying the disjunction rule $DM_{\vee}$, that is, $\lambda(n_1) \setminus \lambda(n_) = \{\varphi\}$, $\lambda(n_2) \setminus \lambda(n) = \{\psi\}$, and either $\varphi \vee \psi \in \lambda(n)$ or $\psi \vee \varphi \in \lambda(n)$. We say that such a node n is a disjunctive node. In addition, if $\mathsf{At}(\varphi \vee \psi) \cap \mathsf{At}(\alpha) \neq \emptyset$ then we say that such a disjunctive node n is α-connected. Given a tableau π for $\mathcal{K}$, let $\pi[\backslash\alpha] = (N, E, \lambda)$ be a sub-labelled tree of π, such that for each α-connected disjunctive node n of π, we remove exactly one of the sub-trees rooted on one of the two children of n. Given a $\pi[\backslash\alpha]$, we define the function $f_{\pi[\alpha]} : N \to 2^N$ where $f(n) = \{n' \in N \mid (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \alpha\}$. Imagine that we re-label each node of the tableau by removing any formula that shares some atomic proposition with α, except α itself. By doing so, some nodes might present the same new label. The function $f_{\pi[\alpha]}$ identifies such nodes whose new labels collapse. The image of $f_{\pi[\alpha]}$ is denoted by $\mathsf{Img}(f_{\pi[\alpha]})$.

We define the collapsed sub-labelled tree of $\pi[\backslash\alpha] = (N', E', \lambda')$ as the labelled tree $\tilde{\pi}[\backslash\alpha] = (\tilde{N}, \tilde{E}, \tilde{\lambda})$, where

- $\tilde{N} = \mathsf{Img}(f_{\pi[\alpha]})$;
- $\tilde{E} = \{(A, B) \in \tilde{N} \times \tilde{N} \mid A \neq B, (n, n') \in E', \text{ for some } n \in A \text{ and } n' \in B\}$;
- $\tilde{\lambda}(A) = (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$, for some $n' \in A$

Lemma A.11. *If n is a disjunctive α-connected node, and n' is a child of n then $(\lambda(n') \setminus \mathit{forms}(\alpha)) \cup \{\alpha\} = (\lambda(n) \setminus \mathit{forms}(\alpha)) \cup \{\alpha\}$.*

Proof. As n is a disjunctive node and n' is a child of it, we get that $\lambda(n') = \lambda(n) \cup \{\varphi\}$. As n is α-connected, we get that $\varphi \in \mathsf{forms}(\alpha)$. Thus,

$$\begin{aligned}(\lambda(n) \cup \{\varphi\}) \setminus \mathsf{forms}(\alpha) &= \lambda(n) \setminus \mathsf{forms}(\alpha)\\ \lambda(n') \setminus \mathsf{forms}(\alpha) &= \lambda(n) \setminus \mathsf{forms}(\alpha)\\ (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} &= (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}\end{aligned}$$

□

Proposition A.12. *If π is a tableau of a knowledge base $\mathcal{K}$, and $\alpha \in \mathcal{K}$ is safe then the collapsed sub-labelled tree $\tilde{\pi}[\backslash\alpha] = (\tilde{N}, \tilde{E}, \tilde{\lambda})$, is a tableau of $\mathcal{K}$, for every $\pi[\backslash\alpha]$.*

Proof. Let us show that each condition of the tableau is satisfied:

- $\tilde{\lambda}(r) = \mathcal{K}$. By definition, $\tilde{\lambda}(r) = (\lambda(r) \setminus \mathsf{forms}(\alpha)) \cup \alpha$ and $\lambda(r) = \mathcal{K}$. Thus, $\tilde{\lambda}(r) = (\mathcal{K} \setminus \mathsf{forms}(\alpha)) \cup \alpha$. By hypothesis, α is safe in $\mathcal{K}$, therefore, $((\mathcal{K}) \setminus \mathsf{forms}(\alpha)) = \mathcal{K} \setminus \{\alpha\}$. This implies that $\tilde{\lambda}(r) = (\mathcal{K} \setminus \alpha) \cup \alpha = \mathcal{K}$.
- let $A \in \tilde{N}$:
  1. $\tilde{\lambda}(A) \neq \tilde{\lambda}(A')$, for all children A' of A. By definition, $A \neq B$.
  2. if $\mathsf{children}(A) = \{A_1\}$, then $\lambda(A_1) \in \sigma(\varepsilon, \lambda(A))$, for some derivation rule $\epsilon \in \mathcal{R}_{TB}$. As A' is single child of A, we have that there are some $n \in A$

and $n' \in A'$ such that n' is child of n in π. Let us fix such a n and n'. As A_1 is child of A, we have that $A \neq A_1$ which implies that

$$(\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} \neq (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}.$$

Thus, from the contrapositive of Lemma A.11, we have that n is not a disjunctive α-connected node. By definition, $\tilde{\lambda}(A) = (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$ and $\tilde{\lambda}(A_1) = (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$. Therefore,

$$\tilde{\lambda}(A) \neq \tilde{\lambda}(A_1).$$

Therefore, n has only one single node which means that $\lambda(n') \in \sigma(\varepsilon, \lambda(n))$, for some derivation rule $\epsilon \in \mathcal{R}_{TB}$:

- $\lambda(n') = \lambda(n) \cup \{\varphi\}$, with $\neg\neg\varphi \in \lambda(n)$. Observe that if $\varphi \in \mathsf{forms}(\alpha)$, then we would have $(\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} = (\lambda(n') \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\}$. But we have from above that this is not the case, therefore $\varphi \notin \mathsf{forms}(\alpha)$, which means $\{\varphi\} \setminus \mathsf{forms}(\alpha) = \{\varphi\}$. This implies that,

$$\begin{aligned}\lambda(n') \setminus \mathsf{forms}(\alpha) &= (\lambda(n) \cup \{\varphi\}) \setminus \mathsf{forms}(\alpha) \\ &= (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\varphi\}\end{aligned}$$

  Thus,

$$\begin{aligned}\lambda(n') \setminus \mathsf{forms}(\alpha) \cup \{\alpha\} &= (\lambda(n) \setminus \mathsf{forms}(\alpha)) \cup \{\alpha\} \cup \{\varphi\} \\ \tilde{\lambda}(A_1) &= \tilde{\lambda}(A_1) \cup \{\varphi\}.\end{aligned}$$

  Thus, $\tilde{\lambda}(A_1) \in \sigma(DM_\wedge, \lambda(A))$.
- the other cases are analogous.

3. if $\mathsf{children}(A) = \{A_1, A_2\}$ with $A_1 \neq A_2$ then there are nodes $n \in A$, $n_1 \in A_1$ and $n_2 \in A_2$ such that both n_1 and n_2 are children of n is π. The proof is analogous to item 2 above.

□

Proposition A.13. *Let π be a tableau for a knowledge base $\mathcal{K}$. If α is safe in $\mathcal{K}$ and there is a node n such that $\mathsf{At}(\lambda(n) \setminus \{\alpha\}) \neq \emptyset$ then π is not minimal.*

Proof. The idea is simple, let us take a collapsed tableau $\tilde{\pi}$ of π. As $\mathsf{At}(\lambda(n) \setminus \{\alpha\}) \neq \emptyset$, there is some formula $\beta \in \lambda(n)$ such that $\beta \in \mathsf{forms}(\alpha)$ and $\beta \neq \alpha$. Consider the following injection $g : \mathsf{leaf}(\tilde{\pi}) \to \mathsf{leaf}(\pi)$ with $g(A) = m \in A$ such that m is a leaf. By definition, $\tilde{\lambda}(A) = \lambda(m) \setminus \mathsf{forms}(\alpha) \cup \{\alpha\} \subseteq \lambda(m)$. Therefore, $\tilde{\pi} \preceq \pi$. As both π

and $\tilde{\pi}$ are tableaux, we have that all leaf nodes reachable from n in π contains β. Let m be one of such leaf nodes reachable from n. Thus, $\tilde{\lambda}(A) = \lambda(m) \setminus \mathsf{forms}(\alpha) \cup \{\alpha\}$. As $\beta \in \mathsf{forms}(\alpha)$, $\beta \neq \alpha$ and $\beta \in \lambda(m)$, we get that $\lambda(m) \setminus \mathsf{forms}(\alpha) \cup \{\alpha\} \subset \lambda(m)$. This means $\tilde{\lambda}(A) \subset \lambda(m)$. Therefore, $\pi \not\preceq \tilde{\pi}$. Thus, $\tilde{\pi} \prec \pi$ which means π is not minimal. $\square$

Theorem A.14. *If α is safe in $\mathcal{K}$ then*

$$\mathcal{T}_{\perp}^{min}(K) = \bigcup_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K} \setminus \alpha)} \pi[\alpha]$$

Proof. As α is safe in $\mathcal{K}$,we have that α is consistent and non-partially-redundant in $\mathcal{K} \setminus \{\alpha\}$ which implies from Proposition A.8 that $\bigcup_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K} \setminus \alpha) \pi[\alpha]} \subset \mathcal{T}_{\perp}^{min}(K)$. Let $X = \bigcup_{\pi \in \mathcal{T}_{\perp}^{min}(\mathcal{K} \setminus \alpha)} \pi[\alpha]$. Thus, $\mathcal{T}_{\perp}^{min}(\mathcal{K}) = X \cup (\mathcal{T}_{\perp}^{min}(\mathcal{K}) \setminus X)$. As α is safe in $\mathcal{K}$, it follows from Lemma A.1 that for every tableau $\pi \in X$, and each node n of π: $\mathsf{At}(\lambda(n)) \cap \mathsf{At}(\alpha) = \emptyset$. Therefore, from Proposition A.13, we get that $(\mathcal{T}_{\perp}^{min}(\mathcal{K}) \setminus X) = \emptyset$. Therefore, $\mathcal{T}_{\perp}^{min}(\mathcal{K}) = X$. $\square$

Theorem 20. *For all $n > 0$ and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\#}, \mathcal{I}^{\Sigma}\}$, $\mathcal{C}^v(\mathcal{I}, n) = \mathcal{C}^f(\mathcal{I}, n) = \mathcal{C}^p(\mathcal{I}, n) = \infty$.*

Proof. We will have to split the proof for $\mathcal{I}^{\#}$ from $\mathcal{I}^{\min}$ and $\mathcal{I}^{\Sigma}$, for each item 1 and 2.

- $\mathcal{I}^{\#}$. Let us consider the following formulae

  $$\alpha_i = (\bigwedge_1^i a)$$

  And for $i \in \mathbb{N}$, consider the family of knowledge bases $\mathcal{K}_i$ defined via

  $$\mathcal{K}_i = \{(\alpha_1 \wedge \neg\alpha_1) \wedge (\alpha_2 \wedge \neg\alpha_2) \wedge \cdots \wedge (\alpha_i \wedge \neg\alpha_i)\}$$

  For example,

  $$\begin{aligned}
  \mathcal{K}_1 &= \{(a \wedge \neg a)\} \\
  \mathcal{K}_2 &= \{(a \wedge \neg a) \wedge \big((a \wedge a) \wedge \neg(a \wedge a)\big)\} \\
  \mathcal{K}_3 &= \{(a \wedge \neg a) \wedge \big((a \wedge a) \wedge \neg(a \wedge a)\big) \wedge \big((a \wedge a \wedge a) \wedge \neg(a \wedge a \wedge a)\big)\}
  \end{aligned}$$

Each $\mathcal{K}_i$ has exactly i minimal closed tableaux. To see this, observe that we can apply rule $\wedge_e$ to obtain one of the conjunctions $\alpha_j \wedge \neg\alpha_j$, for $1 \leqslant j \leqslant i$. Then we can apply rule $\wedge_e$ again to get a clash. This generates a minimal closed tableau. As we have i conjunctions $\alpha_j \wedge \neg\alpha_j$, we obtain i minmal closed tableaux.

Thus, $\mathcal{I}^{\#}(\mathcal{K}_i) = i$, for all $i > 0$. This means that each $\{\mathcal{I}^{\#}(\mathcal{K}_i) \mid i > 0\}$ is an infinite set. Also note that $|\mathcal{K}_i| = 1$, $\mathsf{At}(\mathcal{K}_i) = \{a\}$, and for all $\varphi \in \mathcal{K}_i$, $\mathsf{At}(\varphi) = \{a\}$. Therefore, for $n > 0$, $\mathcal{C}^v(\mathcal{I}^{\#}, n) = \mathcal{C}^f(\mathcal{I}^{\#}, n) = \mathcal{C}^p(\mathcal{I}^{\#}, n) = \infty$.

- $\mathcal{I}^{\min}, \mathcal{I}^{\Sigma}$. Consider the following family of knowledge bases

$$\mathcal{K}_i^+ = \{\alpha_i^+ \wedge \neg a\},$$

where

$$\begin{aligned} \alpha_1^+ &= a \\ \alpha_{i+1}^+ &= a \vee (\alpha_i^+) \end{aligned}$$

For example,

$$\begin{aligned} \mathcal{K}_1 &= \{a \wedge \neg a\} \\ \mathcal{K}_2 &= \{(a \vee a) \wedge \neg a\} \\ \mathcal{K}_3 &= \{(a \vee (a \vee a)) \wedge \neg a\} \end{aligned}$$

Each $\mathcal{K}_i$ has only one minimal closed tableau, and its size is $2i$, thus $\mathcal{I}^{\min}(\mathcal{K}_i) = \mathcal{I}^{\Sigma}(\mathcal{K}_i) = \frac{1}{2i}$. This implies that for all $i, j > 0$, if $i \neq j$ then $\mathcal{I}^{\min}(\mathcal{K}_i) \neq \mathcal{I}^{\min}(\mathcal{K}_j)$ and $\mathcal{I}^{\Sigma}(\mathcal{K}_i) \neq \mathcal{I}^{\Sigma}(\mathcal{K}_j)$. Thus, the sets $\{\mathcal{I}^{\min}(\mathcal{K}_i) \mid i > 0\}$ and $\{\mathcal{I}^{\Sigma}(\mathcal{K}_i) \mid i > 0\}$ are infinite sets. Also note that $|\mathcal{K}_i| = 1$, $\mathsf{At}(\mathcal{K}_i) = \{a\}$, and for all $\varphi \in \mathcal{K}_i$, $\mathsf{At}(\varphi) = \{a\}$. Therefore, for $n > 0$, $\mathcal{C}^v(\mathcal{I}, n) = \mathcal{C}^f(\mathcal{I}, n) = \mathcal{C}^p(\mathcal{I}, n) = \infty$, for $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\Sigma}\}$.

□

Theorem 21.

1. *For all* $n > 1$, $\mathcal{C}^l(\mathcal{I}^{\#}, n) = \infty$.
2. *For all* $n > 3$, *and* $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\Sigma}\}$, $\mathcal{C}^l(\mathcal{I}, n) = \infty$.

Proof. 1. Consider the following family of knowledge bases

$$\mathcal{K}_1 = \{a_1, \neg a_1\}$$
$$\mathcal{K}_{i+1} = \mathcal{K}_i \cup \mathcal{K}\{a_{i+1}, \neg a_{i+1}\}$$

Each $\mathcal{K}_i$ has exactly i minimal closed tableaux. Thus, $\mathcal{I}^{\#}(\mathcal{K}_i) = i$, for all $i > 0$. Observe that for all $i > 0$, and $\varphi \in \mathcal{K}_i$, $|\varphi| \leq 2$. Thus, the set $\{\mathcal{I}^{\#}(\mathcal{K}_i) \mid i > 0\}$ is infinite which implies that $\mathcal{C}^l(\mathcal{I}^{\#}, n) = \infty$, for all $n > 1$.

2. Consider the following family of knowledge bases

$$\begin{aligned}
\mathcal{K}_1^+ &= \{a_1, \neg a_1\} \\
\mathcal{K}_2^+ &= \{a_1, \neg a_1 \vee a_2, \neg a_2\} \\
\mathcal{K}_3^+ &= \{a_1, \neg a_1 \vee a_2, \neg a_2 \vee a_3, \neg a_3\} \\
&\dots \\
\mathcal{K}_{i+1}^+ &= \{a_1, \neg a_1 \vee a_2, \neg a_2 \vee a_3, \dots, \neg a_i \vee a_{i+1}, \neg a_{i+1}\}
\end{aligned}$$

For, $i > 0$, each $\mathcal{K}_i$ has exactly one minimal closed tableau π, and it is size is $|\pi_i| = 2i + 1$. Thus, $\mathcal{I}^{\min}(\mathcal{K}_i) = \mathcal{I}^{\Sigma}(\mathcal{K}_i) = \frac{1}{2i+1}$. Observe that, if $i \neq j$, then $\mathcal{I}^{\min}(\mathcal{K}_i) \neq \mathcal{I}^{\min}(\mathcal{K}_j)$ and $\mathcal{I}^{\Sigma}(\mathcal{K}_i) \neq \mathcal{I}^{\Sigma}(\mathcal{K}_j)$. Therefore, the set $\{\mathcal{I}(\mathcal{K}_i) \mid i > 0\}$ is infinite, for every $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\Sigma}\}$. Also note that for all $i > 0$, and $\varphi \in \mathcal{K}_i$, $|\varphi| \leq 4$. Thus, $\mathcal{C}^l(\mathcal{I}, n) = \infty$, for all $n > 3$, and $\mathcal{I} \in \{\mathcal{I}^{\min}, \mathcal{I}^{\Sigma}\}$.

□

Theorem 22. *For $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$, Exact$_{\mathcal{I}}$, Upper$_{\mathcal{I}}$, and Lower$_{\mathcal{I}}$ are in EXPSPACE, while Value$_{\mathcal{I}}$ is in FEXPSPACE (the functional variant of EXPSPACE).*

Proof. First, we show that Value$_{\mathcal{I}}$ is in FEXPSPACE. From this, we prove that the other problems are in EXPSPACE.

- Value$_{\mathcal{I}}$ is in FEXPSPACE, for all $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$.

  Given a knowledge base $\mathcal{K}$, we will show first how one can compute $\mathcal{I}(\mathcal{K})$, for all $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$. The idea is simple, we enumerate all tableaux, and we mark all minimal tableaux, thereafter we count and check the size of each minimal tableaux. First, note that we do not allow two nodes on the same branch of a tableau to have the same label (if the application of a rule repeats some label on the branch, we ignore this application and look for another rule application). As $\mathcal{K}$ is finite and each formula is finite, at each derivation step there is only a finite number of possible derivations and the number of possibilities reduces in the following derivation step. Therefore, the procedure eventually finishes. Each branch has at most linear size on the sum of the sizes of the formulae in $\mathcal{K}$, while a tableau can have exponential size on the sum of the sizes of the formulae in $\mathcal{K}$. And we have an exponential number of tableaux on the size of the sum of the sizes of the formulae in $\mathcal{K}$. To determine $\mathcal{I}(\mathcal{K})$, for any $\mathcal{I} \in \{\mathcal{I}^{\#}, \mathcal{I}^{\min}, \mathcal{I}^{\#}\}$, we (1) enumerate all such tableaux, (2) check which ones of them are minimal, and (3) for $\mathcal{I}^{\#}(\mathcal{K})$, we count the number of such minimal tableaux. For $\mathcal{I}^{\min}$, we visit each minimal tableaux, keeping the size of the minimal tableau visited so far. The value of $\mathcal{I}^{\min}(\mathcal{K})$ corresponds to the value obtained when we finish visiting all minimal tableaux. For $\mathcal{I}^{\Sigma}$, the process is analogous, we just need to keep a counter that is incremented every time that we find a minimal tableau with the same size as the least tableau so far computed. However, if a smaller tableau is found, then we reset this counter to one. At the end of the procedure, we obtain the correct value of $\mathcal{I}^{\Sigma}(\mathcal{K})$. This strategy takes an exponential space, since we have an exponential number of tableaux (as explained above), and each of them has at most exponential size.

- The problems Lower, Upper and Exact are easily solved by using the TM that computes Value$_{\mathcal{I}}$. In the input $\mathcal{K} \in \mathbb{K}, x \in \mathbb{R}^{\infty}_{\geq 0} \setminus \{0\}$, simulate the TM M that solves Value$_{\mathcal{K}}$, that we presented above. To compute Lower, Upper and Exact, we only need to compare x with the value returned by M.

$\square$

On the Evolution of A.I. and Machine Learning: Towards a Meta-level Measuring and Understanding Impact, Influence, and Leadership at Premier A.I. Conferences

Rafael B. Audibert
Institute of Informatics, Federal University of Rio Grande do Sul, Brazil
rbaudibert@inf.ufrgs.br

Henrique Lemos
Institute of Informatics, Federal University of Rio Grande do Sul, Brazil
hlsantos@inf.ufrgs.br

Pedro Avelar
Department of Informatics, Kings College London
*Machine Intellection Department, Institute for Infocomm Research, A*STAR, Singapore*[*]
pedro_henrique.da_costa_avelar@kcl.ac.uk

Anderson R. Tavares
Institute of Informatics, Federal University of Rio Grande do Sul, Brazil
artavares@inf.ufrgs.br

Luís C. Lamb
Institute of Informatics, Federal University of Rio Grande do Sul, Brazil
MIT Sloan School of Management, Cambridge, MA
lamb@inf.ufrgs.br, luislamb@mit.edu

[*]Work done while at Institute of Informatics, Federal University of Rio Grande do Sul, Brazil

Abstract

Artificial Intelligence is now recognized as a general-purpose technology with ample impact on human life. This work aims at understanding the evolution of AI and, in particular Machine learning, from the perspective of researchers' contributions to the field. In order to do so, we present several measures allowing the analyses of AI and machine learning researchers' impact, influence, and leadership over the last decades. This work also contributes, to a certain extent, to shed new light on the history and evolution of AI by exploring the dynamics involved in the field's evolution by looking at papers published at the flagships AI and machine learning *conferences* since the first International Joint Conference on Artificial Intelligence (IJCAI) held in 1969. AI development and evolution have led to increasing research output, reflected in the number of articles published over the last sixty years. We construct comprehensive citation-collaboration and paper-author datasets and compute corresponding centrality measures to carry out our analyses. These analyses allow a better understanding of how AI has reached its current state of affairs in research. Throughout the process, we correlate these datasets with the work of the ACM Turing Award winners and the so-called two AI winters the field has gone through. We also look at self-citation trends and new authors' behaviors. Finally, we present a novel way to infer the country of affiliation of a paper from its organization. Therefore, this work provides a deep analysis of Artificial Intelligence history from information gathered and analysed from large technical venues datasets and suggests novel insights that can contribute to understanding and measuring AI's evolution.

1 Introduction

Artificial Intelligence is now seen as a general-purpose technology that impacts the world's economy in significant ways Crafts [2021]. AI research started in academia, where seminal works in the field defined trends first in machine intelligence McCulloch and Pitts [1943], Turing [1950] and later on the early development and organization of the area ranging from symbolic to connectionist approaches Feigenbaum and Feldman [1963], Minsky [1961]. However, AI has become more than a research field explored in-depth in academia and research organizations. Applied AI research has led to uncountable products, technologies, and joint research between universities and industry, see e.g., Gomez-Uribe and Hunt [2016], Ramesh et al. [2021], Amini et al. [2020]. Recent business research Gartner [2019] has shown that AI is now being implemented widely in organizations, at least to some extent. AI research has led to groundbreaking results that caught the media's attention. For instance, in the 1990s, Deep Blue Campbell et al. [2002] became the first computing system to win a chess match against the then reigning chess world champion, Garry Kasparov, under tournament conditions.

Later, AI research would eventually lead to even higher grounds in many applications. AlphaGo Silver et al. [2016] has won a series of matches against Go world champions, Brown et al. [2020] can generate texts that suggest a future of possibly human-like competence in text generation, Cobbe et al. [2021] has shown how to solve math word problems, Jumper et al. [2021] significantly improved 3D protein structure prediction, and Park et al. [2019] can render seemingly authentic life-like images from segmentation sketches, to name a few.

Even though the area has seen a noticeable technological impact and progress, we claim that there is a need to analyse the history and evolution of AI and the dynamics involved in transforming it into a well-established field within Computer Science. Some influential researchers, such as Gary Marcus, have discussed the developments that happened in the area in recent years Marcus [2018]. Moreover, Marcus reflected upon what is to come in the next decade Marcus [2020]. The current debate has also motivated the research on new approaches to integrating the symbolic and connectionist paradigms, leading to Neurosymbolic AI d'Avila Garcez and Lamb [2003, 2006], d'Avila Garcez et al. [2009]. This approach aims to achieve more robust AI systems endowed with improved semantic understanding, cognitive reasoning, and learning abilities d'Avila Garcez and Lamb [2023], Riegel et al. [2020], Besold et al. [2022]. Further, it is even more evident now than in the dawn of AI that the field not only draws inspiration from – but also inspires advances in other areas – including cognitive psychology, neuroscience, economics, and philosophy see, e.g., Booch et al. [2020], Marcus and Davis [2021], Smolensky et al. [2022]. Kautz has recently pointed out that given AI's recent contributions, "we may be at the end of the cyclical pattern; although progress and exuberance will likely slow, there are both scientific and practical reasons to think a third winter is unlikely to occur.Kautz [2022]" making the case that we might not see another AI Winter shortly. The section on AI history briefly details the field's evolution.

A Note on Methodology and Contributions

In this paper, we look further back in Artificial Science history, explore its evolution, and contribute to understanding what led to the AI impacts we have in society today. To do so, we will investigate how the collaboration and citation networks of researchers evolved since 1969 [1] within three major AI conferences. We start our analyses with IJCAI, AAAI, and NeurIPS, *together with flagship conferences*

[1]The date was chosen because it marks the first International Joint Conference on Artificial Intelligence - IJCAI-69.

of related research areas which are impacted and influenced by AI, namely ACL, EMNLP, NAACL, CVPR, ECCV, ICCV, ICML, KDD, SIGIR, and WWW. Even though not all of these conferences had many AI-related papers published in their early years, more recently, it is clear that AI techniques are playing a more prominent role in these research communities. Therefore, we add them to our analyses to compose a "big picture" of how AI has not only grown itself but also gradually started to influence other fields. These include, e.g., computer vision, image processing, natural language processing, and information retrieval.

We proceed by exploring and enhancing an extensive dataset of papers published in Computer Science and AI venues since 1969, the v11 Arnet datasetTang et al. [2008]. We use version v11 from this data dataset, containing data originating from DBLP with further disambiguation regarding paper authorship, spanning from 1969 to 2019. There are Arnet versions v12 and v13 available with data until 2021. However, the data for the recent years are somewhat degraded in these recent datasets, thus rendering their statistical analysis inadequate and error-prone (See Section 3.1 to understand our trade-offs on using v11 instead of v13). We then use this dataset to create a new dataset of our own, modeled in several different graph representations of the same data, allowing us to explore them in a true network fashion. With their centralities already computed, these graphs are available for future research. The process to generate them involves using considerable compute power, with amounts of memory and processing not easily found outside laboratories at large-scale research universities or corporations.

Using citation and collaboration networks, our analyses then use centralities to rank both papers and authors over time. We then correlate these rankings to external factors, such as conferences' localization or the ACM's Turing Award – the most prestigious research recognition in Computer Science. These data will allow us to explore what/who, were/are the influential papers/authors in every area/venue under study here. Additionally, we will also examine the dynamics of where all this research is being produced, trying to understand the recent growth of scientific output in China concerning the other countries that led the first decades of AI research.

In these analyses (Section 4), we try to understand and show how authors do not maintain their influence in the field for an extended period. We also analyse this influence regarding ranking papers by importance, as papers can be considered relevant for a more extended period. We also show that the *average number of authors per paper is increasing* in the analysed venues and the number of self-citations too. Furthermore, we also investigate the authors who introduce most co-authors to these conferences. We also show the dynamics behind citations between conferences, showing how some meetings work better together than others.

Because of the nature of our work – converting large amounts of non-organized data into a structured data format – we also generate some side contributions besides our main work. These contributions are: *(i) a new and efficient Python library to convert XML to JSON that uses file streams instead of loading the whole data in memory; (ii) a parallel Python implementation to compute some centrality measures for graphs, using all physical threads available in a machine, and (iii) a novel structure to avoid reprocessing data already processed when its underlying structure is a graph.*

Paper Structure

We organize our paper as follows: Section 2 provides a brief history of Artificial Intelligence[2], some background information on the analysed computer science conferences, the ACM's Turing Award, and a review of graph theoretical concepts. Section 3 describes the methodology, including information about the underlying dataset and the process behind the generation of the graphs/charts used throughout this work. Section 4 presents and discusses the analyses of the aforementioned data from various perspectives. Section 5 concludes our work and presents suggestions for future work using the new datasets. The Appendix brings some tables and figures that we avoid including in the main body of the paper to facilitate the reading flow.

2 Background

2.1 A Short on Artificial Intelligence History

Artificial Intelligence history is generally defined in terms of main time periods, where the field grew stronger, interluded by two periods (the so-called AI Winters) where the area was somewhat discredited and underfunded and thought to be of limited real-world impact. The coming material is not exhaustive but provides historical background to understand how the data analysed here relates to these periods. Several works describe aspects of AI history under different perspectives on how the field evolved in time, see e.g. Kautz [2022], Russell and Norvig [2020].

[2]We present a brief, non-comprehensive history of AI, focusing only on topics related to the analysed AI publications. Our historical account is by no means comprehensive. We choose to focus only on the topics related to the AI venues analysed in *this* paper. Of course, a comprehensive history of AI would not fit into the space of a research paper. Therefore, the reader should not see or interpret our brief historical account as definitive.

2.1.1 The Dawn of AI *(1940-1974)*

Although debatable, some of the first XX-century "modern artificial intelligence" papers were published in the 1940s. One of the first artificial neural network-related papers arguably dates back to 1943, when Warren McCulloch and Walter Pitts formally defined how simple logical operations from propositional logic could be computed in a connectionist setting McCulloch and Pitts [1943]. Later, in 1950, Alan Turing published the widely cited "Computing Machinery and Intelligence" paper Turing [1950], one of the first philosophical papers reflecting upon the interplay of intelligence and machines and on the possibility of building intelligent machines. In this paper, Turing reflects if machines are able to think and also proposes the "imitation game" (now widely known as the Turing Test) in order to verify the reasoning and thinking capabilities of computing machines. Nevertheless, it was in 1956 that the term Artificial Intelligence (AI) was coined by John McCarthy during the Dartmouth Summer Research Project on Artificial Intelligence workshop. From the workshop onward, A.I. rapidly evolved into a potentially world-changing research field – at that time, especially focusing on the symbolic paradigm, influenced by logical reasoning and its computational implementations. One of the first collections of artificial intelligence articles would be published in Feigenbaum and Feldman [1963].

A well-known example of rule-based systems of the 1960s is Eliza Weizenbaum [1966], the first-ever chatbot, created in 1964, by Joseph Wiezenbaum at the Artificial Intelligence Laboratory at MIT. Today's chatbot market is considerably large, powering multi-million dollar companies revenues like Intercom[3] or Drift[4]. Eliza was created to be an automated psychiatrist, as if the human was talking to someone who understood their problems, although the system worked in this rule-based format, replying to the user with pre-fed answers. Besides the main artificial intelligence approach, we can already see how related areas are easily influenced with a chatbot clearly involving natural language processing as well.

It would also be in 1964 that Evans [1964] would show that a computer could solve what they described as "Geometry Analogy Problems", which correlates with the problems usually displayed in IQ tests where one needs to solve a question in the format "figure A is to figure B as figure C is to which of the given answer figures?" such as the one represented in Figure 1

Important research would also vouch in favor of the area, causing DARPA (the American Defense Advanced Research Projects Agency) to fund several different AI-related projects from the mid-1960s onward, especially at MIT. This era was marked

[3] https://www.intercom.com/

[4] https://www.drift.com/

Evans [1964]'s Figure 1

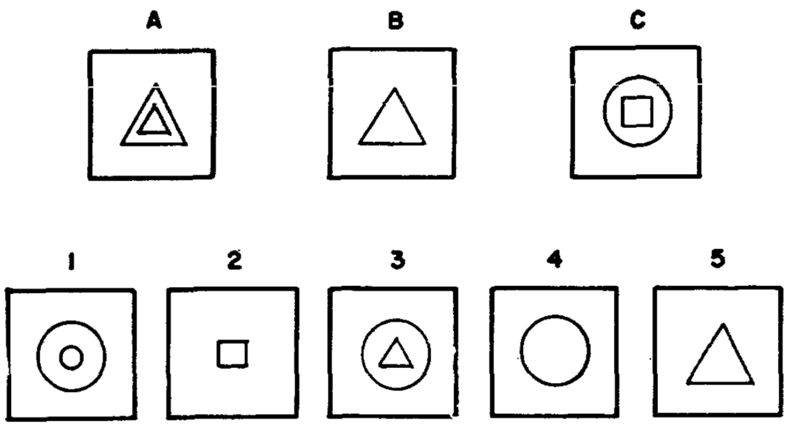


Figure 1: Geometry Analogy Problem example, which correlates with problems usually deployed in IQ tests where one needs to solve a question in the format "figure A is to figure B as figure C is to which of the given answer figures?"

by the extreme optimism in the speeches of the area practitioners. Marvin Minsky said in a 1970s Life magazine interview – one year after receiving the Turing Award (See Section 2.2) – that "from 3-8 years we will have a machine with the general intelligence of a human being". He would also, in the same interview, claim that "If were lucky, they might decide to keep us as pets.". Science Fiction fully adopted the Artificial Intelligence utopia, with the release of famous movies like the French "Alphaville" in 1965 by Jean-Luc Godard, and "2001: A Space Odyssey" by Stanley Kubrick (and screenplay from Arthur C. Clarke) in 1968.

Prior to its first Winter, however, AI had grown into a sizeable academic community. The First International Joint Conference on Artificial Intelligence (IJCAI), was then held at Stanford, in 1969. In it, out of the 63 published papers, some have been influential, such as Stanford's AI work in a "system capable of interesting perceptual-motor behavior" Feldman et al. [1969], Nilsson [1969]'s Mobius automation tool, and Green [1969]'s QA3 computer program that can write other computer programs and solve practical problems for a simple robot.

It was also before the first winter that Alain Colmerauer and Robert Kowalski would develop Prolog, now a prominent programming language, which was widely deployed in the 1970s and 1980s Kowalski [2014], Körner et al. [2022] that also influenced the field of inductive logic programming and statistical relational learning

Raedt et al. [2016]. The feature that makes Prolog stand out among other languages is the fact that it is mostly a declarative language: the program logic is expressed in terms of logic predicates (relations), represented as facts and rules. A computation is initiated by running a query over these relations Lloyd [1984]. More recently, Prolog would become a programming language also used in Watson[5], IBM's question-answering computer system.

2.1.2 The First AI Winter *(1974-1980)*

The first AI winter was defined by the hindrances found by researchers and practitioners while trying to develop deployable artificial intelligence technologies. The biggest challenge is today recognized as the lack of computing power needed by artificial intelligence algorithms, which simply did not exist at the time. Computers did not have enough memory to store the overwhelming amount of data required to build these complex rule-based systems or just did not have enough computational power to solve problems fast enough.

Minsky and Papert [1969] may have played a part in this process. Criticism of the "perceptron" (seen as a learning algorithm used in binary classifiers) possibly has influenced the Artificial Intelligence research agenda on going deeper into neural networks – highly influential now – and instead focusing on symbolic methods.[6] New results in NP-Completeness established by Cook [1971] and Karp [1972] in the early 1970s could have also played a role in raising skepticism, as results in computational complexity showed that many problems can only be solved in exponential time. This posed a risk to AI because it meant that some of the basic problems being solved by the era models would probably never be used in real-life data, where data is not represented by just a few data points. The well-known Lighthill report Lighthill [1973] also played a role, as it was interpreted as being critical of the state of AI research and its deployed results in the last 25 years leading to its publication in 1972.

2.1.3 The First AI Summer: Knowledge, Reasoning and Expert Systems *(1980-1987)*

The external perception of AI research results slowly recovered again in the early 1980s, mainly due to increasing commercial interest in *expert systems*. At that time, the symbolic school achieved higher prominence than other fields, and developments

[5] `https://www.ibm.com/watson`

[6] Pollack [1989] reviews Minsky and Papert's *Perceptrons* and clarifies several issues around this influential book.

in non-monotics and temporal logic have had a lasting influence on the field. Symbolic and logical analyses of time, in particular, led to two Turing Award recognitions: Amir Pnueli (for introducing the methods of temporal logic in Computer Science) and Edmund Clarke, Allen Emerson, and Joseph Sifakis for their work on program verification and model checking, which are based upon the theoretical foundations of temporal logics. During the 1980s, we also witnessed the creation of the US-centered National Conference on Artificial Intelligence (AAAI), now a flagship international conference. Besides that, the funds that were somewhat reduced in the first AI winter would also be back on the table, with the Japanese government funding AI research as part of their Fifth Generation Computer Project (FGCP). Prolog had a central role in the fifth generation project and this led to several developments in computational logic at the time, as reported in Shapiro [1983].[7] Some other countries would also restart their funding projects, like UK's Alvey project and DARPA's Strategic Computing Initiative.

After the *Perceptrons* book criticism, connectionism would come back to prominence in the early 1980s. Hopfield [1982] proved that what we today call a "Hopfield network" could learn in a different way than what it was being done before with perceptrons and simple artificial neural networks. Also, at the same time, Rumelhart et al. [1986] and the "PDP research group" would show the potential of "Backpropagation", a new method to easily train and "backpropagate" the gradient in (neural) machine learning models.

2.1.4 The Second AI Winter *(1987-2000)*

Criticisms over the deployment of expert systems in real-world applications, however, may have caused the second AI winter (from the late 1980s to the early 2000s), which ended up reducing AI research funding.

Hans Moravec wrote in Moravec [1988] that "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility". This, with some contributions from Rodney Brooks and Marvin Minsky, would emphasize what is now known as Moravec's Paradox: the idea that reasoning *per se* does not require much computation power, and can easily be thought/learned to/by an AI machine, but building an intelligent machine able to do what is "below conscience level for humans", i.e. motor or "soft" skills, is what

[7] Please note that a comprehensive history of symbolic AI and of the impact of logic in Artificial Intelligence is well beyond the scope of this paper. For a complete analyses of the many contributions of logic to AI and an understanding of the developments of logic-based AI methods, see Gabbay et al. [1998, 2014].

actually required enough computation power that did not yet exist at the time.

However, much happened in the 1990s as regards AI technology and its impacts. IBM success, represented by Campbell et al. [2002] Deep Blue's greatest achievement – finally beating in a match series under tournament rules the then world chess champion Garry Kasparov in 1997. Previously, in 1994, Tesauro [1994] TD-GAMMON program would illustrate the potential of reinforcement learning, by creating a self-teaching backgammon program able to play it at a master-like level. Also, although self-driving cars are usually considered recent technology, the ground for it was laid in this era, with Ernst Dickmmans's "dynamic vision" concept in Dickmanns [1988] and Thomanek and Dickmanns [1995] work where they had a manned car riding in a Paris' 3-lane highway with normal traffic at speeds of up to 130 km/h.

The late 1990s would also see an increase of research in information retrieval with the World Wide Web's boom, with research in web and AI-based information retrieval/extraction tools Freitag [2000].

2.1.5 Recent Advances in the XXI Century *(2000-present)*

The 2000s present us with wider public recognition of AI, especially if we look at the commercial impact of Machine Learning (ML), specifically Deep Learning (DL). In this context, NeurIPS (at the time, NIPS) arose, again, as perhaps the most prominent AI conference, where several influential DL papers have been published, featuring convolutional neural networks, graph neural networks, adversarial networks, and other (deep) connectionist architectures.

In the early 2000's we would see AI reaching a broader customer base in most developed countries. iRobot[8] introduced its Roomba Robot Vacuum in 2002. Apple, Google, Amazon, Microsoft, and Samsung released Siri, Google Assistant, Alexa, Cortana, and Bixby, respectively, AI-based personal assistants capable of better understanding natural language and executing a wider variety of tasks. Admittedly, they did not work that well at the beginning, circa 2010, but these technologies have improved over the last decade.

Most of the visibility in the area since 2000 is related to Deep Learning, basing itself in the Artificial Neural Network (ANN) concept, a system that tries to mimic how our brain cells work.[9] It is interesting to observe that already in 1943, in McCulloch and Pitts [1943], efforts were made to define artificial neural networks.[10]

---

[8] `https://www.irobot.com/`

[9] Of course, explaining the technical details of how artificial neural networks are deployed in machine learning, the many ANN models successfully developed over the last 40/50 years, and their technical complexities are beyond the scope of this work.

[10] Perhaps it is curious and relevant to observe that McCulloch and Pitts described how **propo-**

However, the immense computing power we have now allowed us to stack several layers of "neurons" one after the other – thus "deep" neural networks – and compute the results extremely fast. Also, given the natural parallelism of the process, the advent of Graphics Processing Units (GPUs) created the necessary hardware that led to the increasing number of deep models and their applications.

Some of the most noticeable recent achievements base themselves on Generative Adversarial Networks (GANs). They are a “framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G” Goodfellow et al. [2014]. This framework is responsible for a wave of photo-realistic procedurally-generated content at the likes of some viral websites, such as `https://this-person-does-not-exist.com/en`, `https://thiscatdoesnotexist.com/`, `http://thiscitydoesnotexist.com/`, or the recursive `https://thisaidoesnotexist.com/`.

GANs are associated with what one colloquially calls “deepfakes” – a mash of “deep learning” with “fake”. They work by superimposing one’s face with another face through a machine-learning model. Some more recent deepfakes can also alter the subject’s voice, improving the experience. These are especially bad from an ethics standpoint when one imagines that these can be used to fake audio and images of influential people Hwang [2020]. A thorough review of the area can be found in Nguyen et al. [2019].

Several researchers have recently received the ACM Turing Award for work related to AI, (probabilistic) reasoning, causality, machine learning, and deep learning. In 2010 Leslie Valiant was awarded for his foundational articles dating back to the ’80s and ’90s, with some of his most prominent works associated with AI and Machine learning having defined foundational concepts in Computational Learning Theory (specifically, PAC Learning) Valiant [1984]. Judea Pearl won it in the following year, 2011, for his "contributions [...] through the development of a calculus for probabilistic and causal reasoning" with a representation of these contributions illustrated in Pearl [1988] and Pearl [2009]. Deep learning pioneers were recognized in 2018: Geoffrey Hinton, Yoshua Bengio, and Yann LeCun are widely recognized for their work on deep (belief) networks Hinton et al. [2006], image recognition Krizhevsky et al. [2012], text recognition and convolutional neural networks Lecun et al. [1998], LeCun et al. [1989], GANs Goodfellow et al. [2014] and neural translation Bahdanau et al. [2014] among many relevant contributions. However, it is essential to recognize

sitional logic inference could be described via neural networks. At that time, AI was not an established field, and thus no division among what came to be known as the connectionist and symbolic schools of AI.

that back in the 1980s, the PDP research group played a crucial role in showing the effectiveness of neural learning through backpropagation Rumelhart et al. [1985].

As regards the impact of Deep Learning reported in the media, in particular in the growing industry of entertainment and games, Google's AlphaGo won a series of matches against the Chinese Go grandmaster Ke Jie in 2017[11], after having already won 4 out of 5 matches against Go player Lee Sedol in 2016[12]. Also in 2017, OpenAI's Dota 2 bot[13] won a 1v1 demonstration match against the Ukrainian pro player Dendi, a noticeable demonstration of power in a game with imperfect information, with almost infinite possible future states. Later, in 2019, a new version of the same bot, called OpenAI Five, won back-to-back 5v5 games against the then-world-champion Dota team, OG[14]. Also in 2019 DeepMind's AlphaStar bot reaches the most significant possible tier in Starcraft II[15].

In recent years another sign of the impressive growth in AI research is the increasing number of submitted (and published) papers in the three biggest AI-related venues. Figure 6 shows that we have over 1500+ papers at these conferences in recent years. For exact numbers, please check Table 8. By checking the figure above, it is also important to notice how Computer Vision arguably became the most visible of the related areas, with CVPR having the biggest number of papers in their proceedings, thanks to the boom in applications for image recognition and self-driving cars. We give more details of AI-related publications in Section 4.

2.1.6 Going Beyond Deep Learning: The Recent Debate on Robust and Neurosymbolic AI

The impacts of AI go beyond the results achieved by deep learning. Recently, the AI community witnessed a debate on how to build technologies that are explainable, interpretable, ethical, and trustworthy Rossi and Mattei [2019]. These led to increased attention to other fields that contribute to the challenge of constructing robust AI technologies. In particular, research that combines learning and reasoning in a principled way, for instance, neurosymbolic AI and hybrid models, have been the subject of growing research interest in academia and industry d'Avila Garcez et al. [2009], Marcus [2020], Riegel et al. [2020], Besold et al. [2022]. Further, in a recent *Communications of the ACM* article Hochreiter [2022], Sepp Hochreiter - deep learning pioneer and proponent of the Long Short-Term Memories (LSTM), one of

[11] `https://wired.com/2017/05/googles-alphago-continues-dominance-second-win-china/`
[12] `https://www.bbc.co.uk/news/technology-35797102`
[13] `https://openai.com/blog/dota-2/`
[14] `https://openai.com/five/`
[15] `https://www.ft.com/content/d659b056-fb28-11e9-a354-36acbbb0d9b6`

the most deployed deep learning models - reflects upon a broader AI that "is a sophisticated and adaptive system, which successfully performs any cognitive task by virtue of its sensory perception, previous experience, and learned skill." Hochreiter states that graph neural networks (GNNs) can play a key role in building neurosymbolic AI technologies: "GNNs are a very promising research direction as they operate on graph structures, where nodes and edges are associated with labels and characteristics. GNNs are the predominant models of neural-symbolic computing Lamb et al. [2020]." Further, Hochreiter defends that "the most promising approach to a broad AI is a neuro-symbolic AI, that is, a bilateral AI that combines methods from symbolic and sub-symbolic AI" Hochreiter [2022]. He also states that contributions to neurosymbolic AI can come from Europe, which "has strong research groups in both symbolic and sub-symbolic AI, therefore has the unprecedented opportunity to make a fundamental contribution to the next level of AIa broad AI" Hochreiter [2022]. Although much is yet to be done and shown by AI researchers and professionals, it is clear that the field has grown in stature over the last decades. Also, testimony to the impact of AI is the prominence and growth in the areas of AI ethics, policies, and regulations, as well as annual AI global impact analyses made by several leading research organizations Mishra et al. [2020], Zhang et al. [2021].

2.2 The Association for Computing Machinery Alan M. Turing Award

The annual ACM A.M. Turing Award is regarded as the foremost recognition in computer science. It is bestowed by the Association for Computing Machinery (ACM) to people with outstanding and lasting contributions to computer science.

The award was introduced in 1966, named after the British Mathematician and Computer Science pioneer Alan M. Turing. Turing influenced several different branches of science, formalizing the concept of computation that led to the concept of a universal computing machine (today's "Turing Machines") through influential publications Turing [1936]. Turing is also considered by most as a modern AI pioneer after having designed the Turing test to decide if a machine is “intelligent” or not Turing [1950]. He is also known for his work in the Second World War, helping the British to decode the Nazi German Enigma machine with his *Bombe* machines, named after the Polish *bomba kryptologiczna* decoding machine. Since 2014, however, the winners receive US$1 million, financed by Google CACM [2014] for their exceptional achievement.[16]

The prize has been awarded to 62 researchers in diverse areas of computer science research - Table 11 lists every Turing Award winner and their nationalities. 37% of

[16] `https://amturing.acm.org`

the winners were not born in the United States (some places of birth are not listed in the table) - and only 27%[17] of them credit a country other than the United States as the country where they did their main contribution. The first woman to receive the prize, Frances "Fran" Allen, received the prize for her work on the theory and practice of optimizing compiler techniques only in 2006.

For our AI evolution analyses, the relevant Turing Award Winners are those who had important contributions to this field. The Turing Award has recognized since 1966 seven researchers for their contributions to AI. The following information is provided by ACM at `https://amturing.acm.org/byyear.cfm`:

- **Marvin Minsky** (1969): *For his central role in creating, shaping, promoting, and advancing the field of Artificial Intelligence;*[18]
- **John McCarthy** (1971): *Dr. McCarthy's lecture "The Present State of Research on Artificial Intelligence" is a topic that covers the area in which he has achieved considerable recognition for his work;*[19]
- **Herbert Simon** and **Allen Newell** (1975): *In joint scientific efforts extending over twenty years, initially in collaboration with J. C. Shaw at the RAND Corporation, and subsequentially with numerous faculty and student collegues at Carnegie-Mellon University, they made basic contributions to artificial intelligence, the psychology of human cognition, and list processing;*[20][21].
- **Edward Feigenbaum** and **Raj Reddy** (1994): *For pioneering the design and construction of large-scale artificial intelligence systems, demonstrating the practical importance and potential commercial impact of artificial intelligence technology;*[22][23]
- **Leslie Valiant** (2010): *For transformative contributions to the theory of computation, including the theory of probably approximately correct (PAC) learning, the complexity of enumeration and of algebraic computation, and the theory of parallel and distributed computing;*[24]

[17] See every author page in their ACM Turing Award website: `https://amturing.acm.org/byyear.cfm`

[18] `https://amturing.acm.org/award_winners/minsky_7440781.cfm`

[19] `https://amturing.acm.org/award_winners/mccarthy_1118322.cfm`

[20] `https://amturing.acm.org/award_winners/simon_1031467.cfm`

[21] `https://amturing.acm.org/award_winners/newell_3167755.cfm`

[22] `https://amturing.acm.org/award_winners/feigenbaum_4167235.cfm`

[23] `https://amturing.acm.org/award_winners/reddy_9634208.cfm`

[24] `https://amturing.acm.org/award_winners/valiant_2612174.cfm`

- **Judea Pearl** (2011): *For fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning;*[25]
- **Geoffrey Hinton**, **Yann LeCun**, and **Yoshua Bengio** (2018): *For conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing.*[26][27][28]

2.3 Computer Science and AI Conferences

Today, there are thousands of conferences in Computer Science. In our analyses, we obviously had to narrow them down to the ones considered the flagship venues in order to properly analyse the field at its core. CSRankings is a metrics-based ranking of top computer science research institutions[29], which identifies the works from each institution for each venue. They comprise a small set of conferences considered as the top ones in the subfields of computing: In this work, we will only focus on institutions available in the CSRankings "AI" category, which are briefly described below. Some conferences are acronyms borrowed from namesake scientific associations - such as AAAI.

2.3.1 IJCAI

The International Joint Conferences on Artificial Intelligence (IJCAI) was first held in California in 1969, being the first comprehensive AI-related conference. The conference was held in odd-numbered years, but since 2016 it has happened annually. It has already been held in 15 different countries, while the 2 most COVID-19 pandemic years were virtually held in Japan and Canada. The next editions will be held in Austria (2022), South Africa (2023), and China (2024), increasing the number of countries that hosted the conference to 17 – China has already hosted it before. Similarly to AAAI, IJCAI is a comprehensive AI conference, with some publications ranging from the philosophy of AI to symbolic AI, and machine learning and applications. IJCAI has over the years published important papers from Turing Award winners, e.g. Avin et al. [2005], Feigenbaum [1977], McCarthy [1977], Valiant [1985], Verma et al. [2019].

[25] `https://amturing.acm.org/award_winners/pearl_2658896.cfm`
[26] `https://amturing.acm.org/award_winners/hinton_4791679.cfm`
[27] `https://amturing.acm.org/award_winners/lecun_6017366.cfm`
[28] `https://amturing.acm.org/award_winners/bengio_3406375.cfm`
[29] `http://csrankings.org`

2.3.2 AAAI

The Association for the Advancement of Artificial Intelligence (AAAI – pronounced "*Triple A I*") was founded in 1979 as the American Association for Artificial Intelligence. This association is responsible for promoting some prominent AI conferences since 1980, including The AAAI Conference on Artificial Intelligence (formerly the National Conference on AI). The conference used to be held once every one or two years (depending on whether IJCAI was organized in North America or not). AAAI has been held yearly since 2010. It is worthy of note that although the conference was renamed, it has actually only been held in North America (and remotely in 2021 due to the COVID-19 pandemic). The conference covers AI comprehensively as its (older) sister conference IJCAI. Similarly to IJCAI, AAAI has published several papers from influential researchers and Turing laureates, see e.g. Hinton [2000], de Mori et al. [1988], Pearl [1982], Valiant [2006].

2.3.3 NeurIPS (formerly NIPS)

The Conference and Workshop on Neural Information Processing Systems (NeurIPS) is a machine learning and computational neuroscience conference held every December since 1987. It has been held in the USA, Canada, and Spain. NeurIPS published hundreds of influential papers over the years on several learning models and architectures ranging from Long Short-Term Memories Hochreiter and Schmidhuber [1996], to Transformer architectures Vaswani et al. [2017]. The Conference Board decided to change the meeting name to NeurIPS [30] in 2018.

CSRankings defines it as a "Machine Learning & Data Mining" conference, publishing important papers, recently featuring technologies such as GPT-3 Brown et al. [2020] and PyTorch's technical papers Paszke et al. [2019], which have 31 and 21 authors, respectively. The sheer size of the venue is noticeable, with 2,334 papers accepted in 2021, outnumbering every other conference studied in this work. The "most influential papers in the recent years" snippet, is due to `https://www.paperdigest.org/`.

2.3.4 CVPR

The Conference on Computer Vision and Pattern Recognition (CVPR) is an annual conference on Computer Vision and Pattern Recognition, regarded as one of the most important conferences in its field, with 1,294 accepted papers in 2019. It will in 2023, for the first time, be organized outside the United States in Vancouver,

[30] `https://www.nature.com/articles/d41586-018-07476-w`

Canada. It was first held in 1983 and has since 1985 been sponsored by IEEE, and since 2012 by the Computer Vision Foundation, responsible for providing open access to every paper published in the conference. CVPR is a flagship Computer Vision venue and witnessed groundbreaking work in the past including Siamese Representation Learning Chen and He [2020], GANs Karras et al. [2018], and Dual Attention Networks Fu et al. [2018]. Turing award laureate Yann LeCun published in this conference on several occasions, e.g. Boureau et al. [2010], LeCun et al. [2004].

2.3.5 ECCV

ECCV stands for European Conference on Computer Vision, being CVPR's European arm – even though ECCV 2022 is going to be held in Tel Aviv - Israel. It is held biennially on even-numbered years since 1990, when it was held in Antibes, France. Even though it is considered CVPR's small sister, it had 1,360 accepted papers in 2019, also heavily focusing on Computer Vision with some publications of note such as RAFT Teed and Deng [2020], a model able to segment and predict image depth with high accuracy.

2.3.6 ICCV

Similar to ECCV, the International Conference on Computer Vision is CVPR's International arm, being held every odd-numbered year since 1987, when it was held in London, United Kingdom, been held in 14 other countries ever since. 1,077 papers made the cut in 2019, such as Shaham et al. [2019] who won the 2019's best paper award.

2.3.7 ACL

ACL is the Association for Computational Linguistics's conference held yearly since 2002, having surprisingly been held in 15 different countries in the last 20 years. The website announces ACL as "the premier international scientific and professional society for people working on computational problems involving human language, a field often referred to as either computational linguistics or natural language processing (NLP). The association was founded in 1962, originally named the Association for Machine Translation and Computational Linguistics (AMTCL), and became the ACL in 1968." Commonly referred to as an NLP-related conference, it has some highly-cited work in recent years such as Strubell et al. [2019]'s work in investigating the environmental effects of creating large language models.

2.3.8 NAACL

NAACL is the conference held by the North American Chapter of the Association for Computational Linguistics, therefore also referred to as an NLP conference. The conference is actually named NAACL-HLT (or HLT-NAACL, sometimes) – North American Chapter of the Association for Computational Linguistics: Human Language Technologies. It has been held since 2003, and it was co-located with ACL on the occasions ACL happened in North America. One of the most cited papers on the use of transformers in recent years was published there, the BERT model Devlin et al. [2018].

2.3.9 EMNLP

EMNLP stands for Empirical Methods in Natural Language Processing. The conference started in 1996 in the US based on an earlier conference series called Workshop on Very Large Corpora (WVLC) and has been held yearly since then. The recent conferences are marked by works trying to improve the BERT model Devlin et al. [2018] already explained above, such as Jiao et al. [2019], Feng et al. [2020] and Beltagy et al. [2019].

2.3.10 ICML

ICML is the International Conference on Machine Learning, a leading international academic conference focused on machine learning. The conference is held yearly since 1987, with the first one held in 1980 in Pittsburg, USA. The first conferences were all held in the United States, but the 9th conference, in 1992, was held in Aberdeen, Scotland. Since then it has been held in 10 other countries, and twice virtually because of the COVID-19 pandemic. It contains some seminal papers in Machine Learning from Pascanu et al. and Ioffe and Szegedy, and some more recent excellent research like Zhang et al. [2018] and Chen et al. [2020]. Besides Bengio's aforementioned seminal paper, Turing Award laureate Geoffrey Hinton also published important papers on ICML e.g. Nair and Hinton [2010].

2.3.11 KDD

The SIGKDD Conference on Knowledge Discovery and Data Mining is an annual conference hosted by ACM, which had its first conference in 1989's Detroit. Although it is usually held in the United States, it has already been hosted by a few other countries, namely Canada, China, France, and the United Kingdom. It is the most important conference encompassing the Knowledge Discovery and Data Mining field, with 394 accepted papers in 2021: a smaller number if we compare with the other conferences we investigate in this paper. The conference recent years have seen a significant presence of AI-related research, mostly defined by Graph Neural Networks (GNNs), with works from Qiu et al., Wu et al., Jin et al., and Liu et al., all of them accepted in 2020's SIGKDD. It is interesting to note how many of the authors of such influential papers in the 2020 conference are from China; perhaps a trend – see some insights about it in Section 4.6.

2.3.12 SIGIR

SIGIR stands for Special Interest Group on Information Retrieval, an ACM group. It has its own annual conference that started in 1978 and has happened every single year since then. It is considered to be the most important conference in the Information Retrieval (how to acquire useful and organized information from raw, unorganized, and unstructured data) area. After 43 editions, it has been hosted in 21 different countries. It used to alternate between the USA and a different country, but this rule does not hold anymore, with only one conference in the US in the last 8 years. Many papers in recent years have tackled recommender systems such as He et al., Wu et al., and Wang et al..

2.3.13 WWW

The Web Conference (WWW) is the top conference in the field. It "brings together key researchers, innovators, decision-makers, technologists, businesses, and standards bodies working to shape the Web"[31]. It is a yearly event that started at CERN in Geneva, Switzerland, in 1994. The conference heavily focuses on Semantic web and Data mining with some important results in recommender systems as well.

2.4 A Brief on Graphs and Centrality Measures

Next, we briefly introduce the concepts of graph theory used in this paper. Sections 2.4.2 to 2.4.5 describe the most widely used graph centralities from the literature. Then in Section 2.4.6 we go over some other centrality measures for completeness' sake.

A graph G is represented by a tuple $G = (V, E)$, where V is a set of nodes (vertexes) and E a set of edges $e_{u,v}$ connecting nodes u to v where $u, v \in V$. These edges can be directed or undirected – thus making us able to differentiate between

[31] `https://dl.acm.org/conference/www`

directed and undirected graphs. In the directed case of $e_{u,v}$ we call u as being the source node and v the destination node. We will always use n to represent the number of nodes in a graph, and m to represent the number of edges in it. Also, a pair of nodes (u, v) might have more than one edge connecting them: in this case, we call the graph a multigraph. Similarly, these edges might have a weight w making the graph a weighted graph. Furthermore, we can also have labeled graphs, where nodes and edges can be of different types. These are useful in knowledge representation systems, such as the graph built in Section 3.3.4. We call $p = u_1, u_2, ..., u_p$ a path between u_1 and u_p in G if $\exists\, e_{u_i,u_i+1}\, \forall\, 1 <= i <= p-1$. Basically, we have a path if we can go from node u_1 to u_p through a sequence of connected edges. We can also define the shortest path between a pair of nodes (u, v) as the path with the minimum possible quantity of intermediate nodes – note, however, that we can have more than one shortest path between any pair of nodes (u, v).

2.4.1 Centrality Measures

The interest in Graph centrality measures dates back to the 1940s, but it was more formally incorporated into graph theory in the 1970s Freeman [1978]. A fundamental motivation for the study of centrality is the belief (or relevance) that node position (that can represent a person's position) in a network impacts their access to information, status, power, prestige, and influence Grando et al. [2019], Wan et al. [2021]. Therefore, throughout this work when we want to identify the above concepts we will use graph centralities for the different networks we built. Grando et al. serves as a tutorial and survey for those interested in applying machine learning to this field.

2.4.2 Degree Centrality

We represent the degree of a node u as k_u meaning the number of other nodes connected to this node. In a directed graph we can further split this metric into two: k_u^{in} is the in-degree, representing the number of nodes $v \in V$ that have an edge $e_{v,u}$ with v as source and u as destination (i.e. number of nodes with an edge pointing to u); the opposite metric k_u^{out} is the out-degree, representing the number of nodes $v \in V$ that have an edge $E_{u,v}$. Therefore, it is possible to extend this metric to a centrality called **Degree Centrality** defined as:

$$\mathcal{C}_{Dg}(u) = \frac{k_u}{n-1}, \tag{1}$$

where n represents the number of nodes V in the graph G.

Also, in the same way we have in-degree and out-degree metrics, we can extend Equation 1 and define **In-Degree Centrality** and **Out-Degree Centrality**, respectively:

$$\mathcal{C}_{Dg_{in}}(u) = \frac{k_u^{in}}{n-1} \tag{2}$$

$$\mathcal{C}_{Dg_{out}}(u) = \frac{k_u^{out}}{n-1} \tag{3}$$

These degree metrics are used to identify how well a node is directly connected to other nodes, without considering the influence a node can pass to its neighbors.

2.4.3 Betweenness Centrality

The **Betweenness Centrality** was defined in Freeman [1977], and its measure of the importance of a node u is how many shortest paths in the graph go through u. It is defined as

$$\mathcal{C}_B(u) = \frac{\sum_{s \neq u \neq t} \frac{\partial_{s,t}(u)}{\partial_{s,t}}}{(n-1)(n-2)/2} \qquad \forall s, u, t \in V, \exists\, e_{s,t} \tag{4}$$

where $\partial_{s,t}(u)$ is the number of shortest paths between s and t that go through u, and $\partial_{s,t}$ is simply the number of shortest paths between s and t. Note that we are only ever counting paths between the pair (s,t) if there is a path between (s,t).

Betweenness is related to the notion of connectivity, where a node with a bigger betweenness actually means that it is a point of connection between several nodes. In a graph with a single connected component, a node can have the highest betweenness if it works as a bridge between two individually disconnected components. It is regarded as a measure of a nodes control over communication flow Freeman [1978].

2.4.4 Closeness Centrality

Closeness Centrality was created in Sabidussi [1966] representing the closeness of a node with every other node in the graph. It is the inverse of the farness which in turn is the sum of distances with all other nodes Saxena and Iyengar [2020]. It is defined by

$$\mathcal{C}_C(u) = \frac{n-1}{\sum_{v \neq u} d(u,v)} \qquad \forall u, v \in V \tag{5}$$

where $d(u,v)$ is the distance between the nodes u and v. This distance is simply the number of edges in the shortest path p between the pair (u,v) if the graph is unweighted, while it is the sum of every edge in the path in case the graph is unweighted. Since distance is not defined between every pair of nodes in disconnected graphs (a graph where not every node can be reached from another node) we cannot compute closeness for disconnected graphs. A node with a higher closeness indicates that the node is in the middle of a hub of other nodes. It also means that a node with big closeness values is "closer", on average, to the other nodes, hence closeness. It represents the nodes level of communication independence Freeman [1978], Cardente [2012].

2.4.5 PageRank Centrailty

Pagerank is a global centrality measure that needs the overall network to measure the importance of one node. It measures the importance of one node based on the importance of its neighbors. Saxena and Iyengar [2020]. It was developed by Brin and Page when they created Google, and it is the underlying method behind their search engine. To understand Pagerank, we need to understand that its main idea is to understand how important a web page is in the middle of all the other millions of pages on the World Wide Web. The main idea behind it is that we are considering a web page important if other important web pages link to it.

Think about it as if we had a web crawler randomly exploring the web and increasing a counter every time we enter into a specific page. Then, when you are on a page you either have the option to click on one of the links on the page or go to a random page on the web with probability $0 <= q <= 1$ – this is useful both to model real-life where we simply go to random websites and also to mimic pages without any out link. The usual value for q, also called teleportation or damping factor, is 0.15, as defined in the original paper. Therefore, with this in mind, we can define **Pagerank** as

$$\mathcal{C}_{PR}(u) = \frac{q}{n} + (1-q)\sum_{v} \frac{\mathcal{C}_{PR}(v)}{k_v^{out}} \qquad \exists e_{u,v} \in E \tag{6}$$

The equation above illustrates how this process is iterative because we depend on the Pagerank of every neighbor to be able to compute our own Pagerank. The process usually converges or can be stopped after a certain number of iterations.

2.4.6 Other centralities

There are other useful centralities present in the literature. They were not used in our work, but they would ideally be used in future work using the dataset created. Recent work has focused not only on the application of machine learning in learning centrality measures of complex graphs Grando et al. [2019], but also on analyzing the own application of Graph Neural Networks capable of multitask learning trained on the relational problem of estimating network centrality measures Avelar et al. [2019]. In summary, the reader interested on centrality measures can refer to Grando et al. [2019], Saxena and Iyengar [2020].

- **Semi-Local centrality** Chen et al. [2012] defines a metric similar to the degree centrality where we expand it to 2 levels of neighbours.

$$\mathcal{C}_{SL}(u) = \sum_{v \in N(u)} \sum_{w \in N(v)} d_2(w), \tag{7}$$

  where $\mathrm{d}_2(w)$ is the number of neighbors plus the number of neighbors for every neighbor of w – basically how many nodes you can reach in two steps.

- **Volume Centrality** Wehmuth and Ziviani [2013] is a kind of generalization from the above centrality parameterizing how far a node influence can reach and is defined by

$$\mathcal{C}_V(u) = \sum_{v \in \tilde{N}_h(u)} k_v, \tag{8}$$

  where $N_h(u)$ is the set of neighbors within a distance h of u, and $\tilde{N}_h(u) = N_h(u) \cup \{u\}$. Wehmuth and Ziviani [2013] demonstrated that $h = 2$ results in a good trade-off of identifying nodes with important relations and the cost of computing this relationship.

- **H-index** Hirsch [2005] is a well-known statistic in the research world, being exhibited as a statistic in most research-aggregator portals such as Google Scholar and DBLP. Hirsch [2005] defined that h is the highest integer value for which the author has h papers with at least h citations.

- **Coreness Centrality** Kitsak et al. [2010] represents the idea that the important nodes are at the core of a graph. It can be determined by the process of assigning each node an index (or a positive integer) value derived from the k-shell decomposition. The decomposition and assignment are as follows: Nodes with degree k= 1 are successively removed from the network until all remaining nodes have a degree strictly greater than 1. All the removed nodes at this stage are assigned to be part of the $k$shell of the network with index k_S= 1 or the 1-shell. This is repeated with the increment of $k$to assign each node to distinct k-shellsWan et al. [2021]. See Figure 2 to see an example of the definition of k-shells. Then, we can mathematically define this centrality as

$$\mathcal{C}_k(u) = \max\{k|u \in H_k \subset G\}, \tag{9}$$

where H_k is the maximal subgraph of G with all nodes having a degree of at least k in HWan et al. [2021].

Source: Tanase et al. [2015]

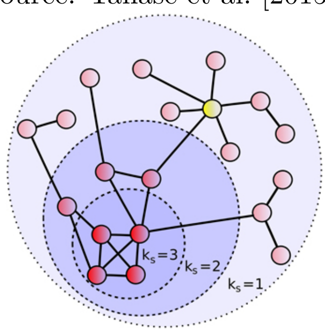


Figure 2: Example of k-shell assignment

Some more complex centralities mostly use the fact that we can define a graph by its adjacency matrix **A** and its corresponding eigenvalues and eigenvectors. Since we are not carrying out a comprehensive review or using them, we will not describe them.

2.5 Related Work

One of the main reasons motivating this work is the fact that the history of Artificial Intelligence and its dynamic evolution has not been researched in depth, at least with respect to our methodology: Xu et al. [2019] focused specifically on "explainable AI" evolution (or de-evolution, in this case); Oke [2008] does deepen their work in several different AI areas, with a review of each area, but does not go back in history further than the mid-1990s. There are also similar approaches to investigating author citation/collaboration networks such as Ding et al. [2010], Guns et al. [2011], Abbasi et al. [2012], and Wu et al. [2019], mostly focusing on the betweenness centrality. von Wartburg et al. [2005] uses closeness to analyse patent networks. Also, Krebs [2002] shows how centrality measures can be used to identify prominent actors from the 2001 Twin Tower terrorist attackers network.

Regarding the authors' country affiliation in papers, Grubbs et al. [2019] investigated coauthor country affiliation in Health research funded by the US National Institute of Health; Michalska-Smith et al. [2014] goes further by trying to correlate country of affiliation with the acceptance rate in journals and conferences; Yu et al. [2021] studied how one can infer the country of affiliation of a paper from its data in WoS[32]; Hottenrott et al. [2019] investigates the rise on multi-country affiliations in articles as well.

3 Methodology

3.1 Underlying Dataset

The most extensive public bibliography of computer science publications is probably the DBLP Database DBLP [2019], available at `https://dblp.uni-trier.de/`. Recently (in February 2022), it surpassed the 6 million publications mark (See Figure 3), containing works from almost 3 million different authors. Figure 4 shows how large is the increase in publications in the recent years, per DBLP's statistics page[33]. They provide a downloadable 664MB GZipped version of their dataset in XML format[34]. Recently (after this work had already been started and was past the dataset-choosing process), DBLP has also released its dataset in RDF format[35]. However, because their dataset contains duplicated authors and/or incorrectly merged authors, we opted to not use their dataset directly. Instead, in our work, we used Arnet's Tang et al. [2008] V11[36] paper citation network, which dates from May 2019. It contains 4,107,340 papers from DBLP, ACM, MAG (Microsoft Academic Graph), and other sources, including 36,624,464 citation relationships.

[32]`https://www.webofknowledge.com/`
[33]`https://dblp.org/statistics/index.html`
[34]`https://dblp.org/xml/release/`
[35]`https://blog.dblp.org/2022/03/02/dblp-in-rdf/`
[36]`https://lfs.aminer.cn/misc/dblp.v11.zip`

Source: `https://blog.dblp.org/2022/02/22/6-million-publications/`

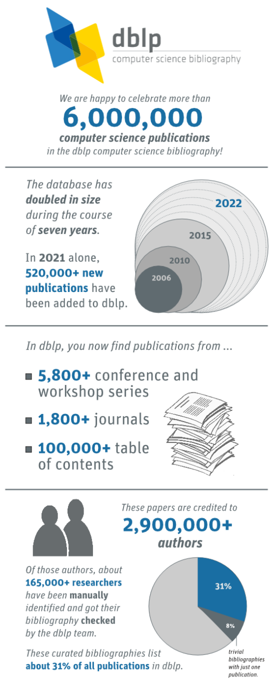


Figure 3: Excerpt of a DBLP poster acknowledging their 6 million papers mark

This dataset contains more information than DBLP's, as they better worked on author disambiguation (merging authors DBLP considered to be different ones, or separating authors DBLP considered to be the same person), providing us the ability to generate truther collaboration/citation networks.

It is important to clarify why we are using Arnet's v11 dataset instead of one of their newer datasets, namely v12 and v13 – the latter, from May 2021, contains 5,354,309 papers and 48,227,950 citation relationships, an increase of 30.3% compared to v11. First, and foremost, this work started in 2019, when versions v12 and v13 were not available yet. Also, when these newer datasets were made available, we did try to use both of them, but we faced some problems that prompted us back to the v11 dataset:

1. v12's and v13's data format is different from v11's. The format of v12 and v13 is a fully-fledged 11GB XML file, which required us to write a new Python library to convert from XML to JSON (our storage method) without loading the whole file into memory by streaming-converting it (see Section H.1). Be-

Source: `https://dblp.org/statistics/publicationsperyear.html`

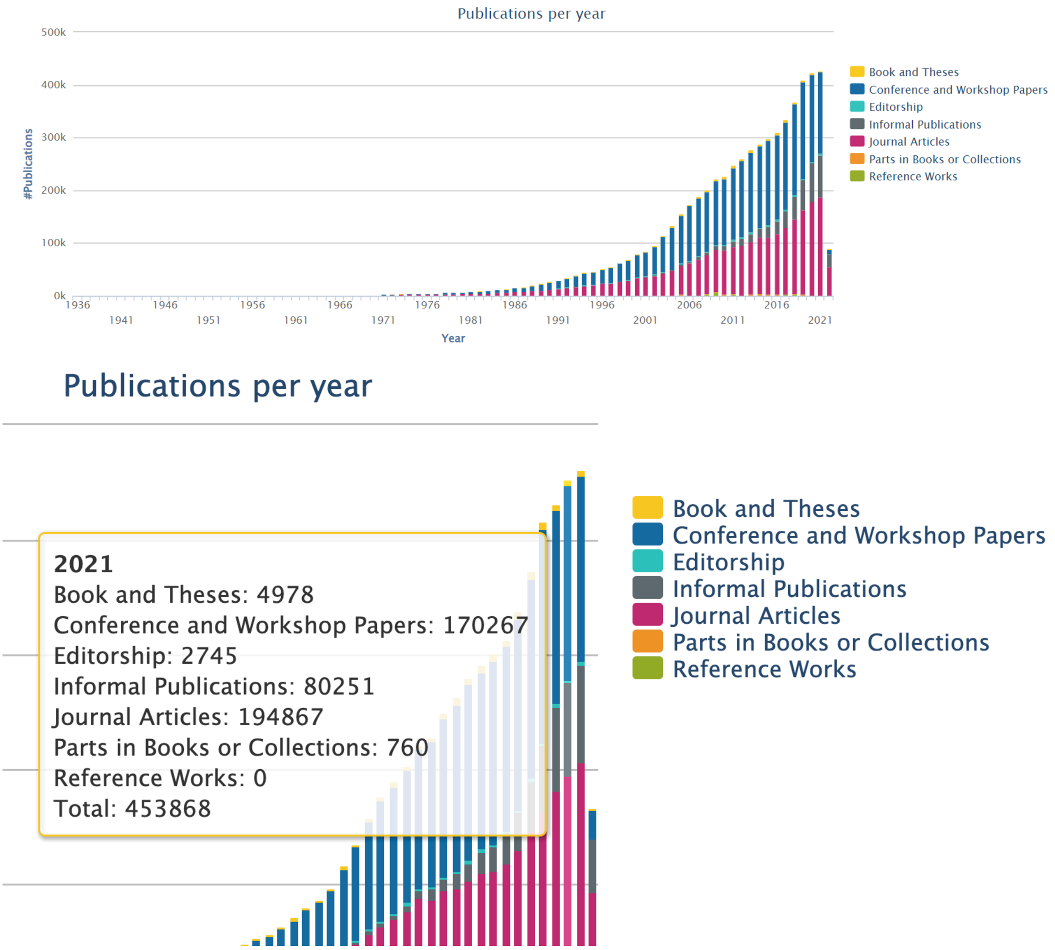


Figure 4: DBLP papers per year, with a detailed view of 2021.

sides the file being harder to read and handle, the new format also changed the IDs from an integer to a UUID-based value, causing us to rewrite the whole logic that was able to detect papers from the main AI conferences based on their past integer values.

2. There are fewer papers from the AI conferences of interest for this work. Even though we have 30% more papers in the most recent version, after carefully finding out which are the new IDs for the conferences, we could only find 58490

papers out of the 89102 (65%) present on version v11. As a smoke test, we did reduce our test only for the main AI conferences (AAAI, NeurIPS, and IJCAI): we could manually count 42082 papers in these 3 conferences – and this is a lower bound because we could not find the count of papers in some years for AAAI and IJCAI; v11 and v13 have 41414 and 20371 of them, respectively. We also tried finding the AI Conferences by name instead of IDs (at cost of some false positives) but it did not work, also finding only 20929 papers. This shows how we have twice the data in v11 compared to v13 instead of 30% more in v13 as expected.

3. Missing data in the most recent years. Even though v13 should have data until 2021, there are only a few hundred papers for the main AI conferences in 2019, 2020, and 2021, while in reality there should be 12559 of them.

All of the data compiled to build the points above can be seen in Table 1, and Figure 8. Table 8 has the raw data used to build Figure 8, where "?" data points were considered to be 0 for the sake of simplicity. An interesting statistical information one might get from Figure 5 is the fact that even though IJCAI used to happen only in odd-numbered years, even-numbered years do not have any noticeable NeurIPS and AAAI paper acceptance rates increase.

Section 4.6.1 shows some charts where it can be seen how degraded our data looks if we had used *v13* instead of *v11*.

	AI Conferences Total
Manual Count	42082
v11	41414
v13 detecting conferences by ID	20371
v13 detecting conferences by name	20929

Table 1: Comparison of paper counts with different methods

Arnet's v11 format is a variation of a JSON file with some improvements to make it easier to load the data in memory without having to load the whole file. Every line is a valid JSON object, requiring us to simply stream the file, iterating over every line, parsing the JSON file, keeping only the required information in memory, and immediately send the JSON file to be garbage collected, using no more than 8kb of memory to read the entire file.

Every JSON object in this file follows the structure defined in Table 2. We, then, for most of the work, keep only the fields tagged with an asterisk ($*$). Also,

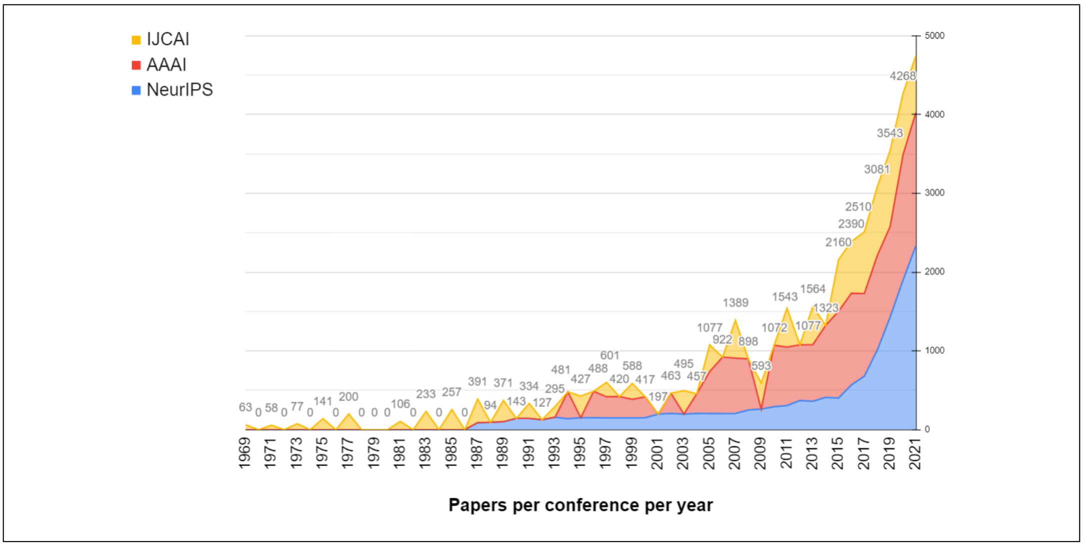


Figure 5: Manual paper count per year in AAAI, NeurIPS and IJCAI

a question mark symbol (?) indicates the field is optional and is, sometimes, not present in the data provided by Arnet. Figure 43 shows an example of such JSON entry, depicting Glorot and Bengio [2010]'s representation in the dataset.

"" indicates the field was used in this work*
"?" indicates the field is optional

Field Name	**Type**	**Description**
id*	*string*	Unique identifier for the paper
title*	*string*	Paper title
authors*	*Author[]* (See Table 5)	List of every single author
venue*	*Venue* (See Table 6)	Object with data about the venue
year*	*integer*	Year of publication
n_citation	*integer*	Citation number
page_start?	*string*	Paper start page in the Proceedings/Book/Journal
page_end?	*string*	Paper end page in the Proceedings/Book/Journal
doc_type	*string*	Place of publication
publisher?	*string*	Book/Journal publisher
volume?	*string*	Book volume
issue?	*string*	Journal issue
references*	*string[]*	List of ids this paper references
indexed_abstract*	*IndexedAbstract* (See Table 7)	Inverted index holding data about the paper abstract

Table 2: Data structure for a single entry in the Arnet JSON dataset

Figure 6 shows some raw insights about this dataset, using the conferences defined in Section 2.3. It shows that all conferences have seen an increasing trend in the number of papers in the last few years, especially CVPR and AAAI.

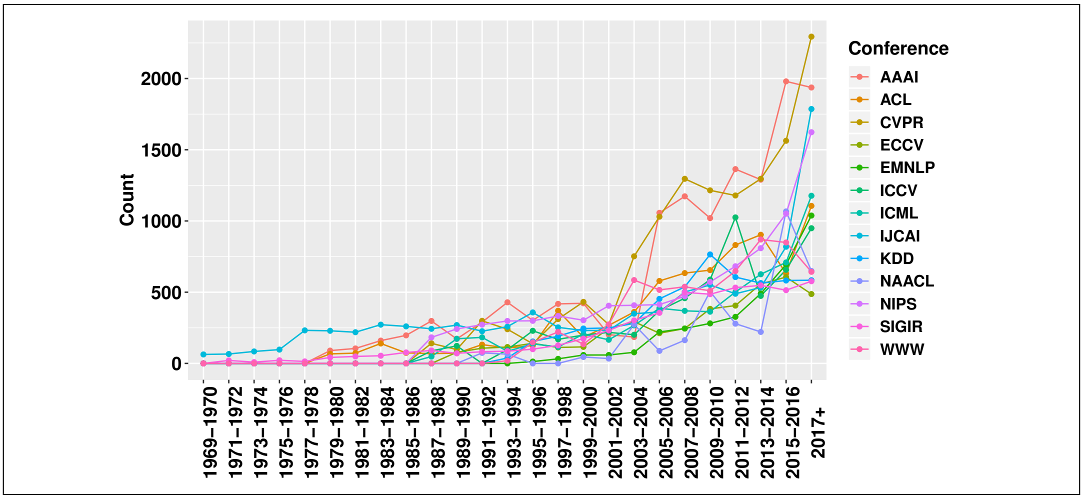


Figure 6: Number of papers per conference per year.

3.2 Artifacts

The code used to download the data, parse the dataset, and generate the graphs, analyses, and charts present in this work is available at `https://github.com/rafaeelaudibert/TCC/tree/v11` in Github. The code for this work is in branch *v11*. The *master* branch contains the code used when we were trying to parse Arnet's *v13* dataset, which did not work out as explained in the previous section. Everything data analysis was built using Python, with the help of some open-source third-party libraries (See Table 9) available in PyPi. For the most complex plots, Python was not the right tool for the job, so they were built using R and its built-in counterparts for *matplotlib*, *numpy* and *seaborn*. Unfortunately, the code for these graphs is not available anymore because it was lost during a disk formatting procedure.

3.2.1 Graph Datasets

Throughout this work, we assembled 5 new datasets, modeled in a graph structure, which are briefly described below. A thorough explanation can be found in each respective section below.

Author Citation Graph (*ACi*) Directed multigraph, where every author is a node, with edges representing citations.

Author Collaboration Graph (*ACo*) Undirected graph, where every author is a node, with edges representing co-authorship

Paper Citation Graph (*PC*) Directed graph, where every paper is a node, with edges presenting citations.

Author-Paper Citation Graph (*APC*) Directed labeled graph, where nodes can be an author or a paper, and we can have edges between papers (citation) or between authors and papers (authorship).

Countries Citation Graph (*CC*) Directed multigraph, where each node represents a country of origin, and edges represent citations.

As our work is focused on the flagship AI and adjacent fields conferences, we filtered their dataset to contain only the papers published in these conferences to build ours. The chosen conferences were based on CSRankings CSRankings [2019] top-ranked AI conferences, which include the following fields: Artificial Intelligence, Computer Vision, Machine Learning & Data Mining, Natural Language Processing, and The Web & Information Retrieval. For each of the graphs explained above, we calculated the following exact centralities: degree (in and out) (Section 2.4.2), betweenness (Section 2.4.3), closeness (Section 2.4.4), and PageRank (Section 2.4.5).

For our work, we created the cumulative graph for each year from 1969 (the first IJCAI conference) until 2019, i.e. the cumulative graph for the year 2000 contains all the papers before and including 2000. A graph for each individual year from 1969 to 2019 was also created, to help with the analysis presented in the sections below. The cumulative graphs containing all the data, including exact centralities, were made available at `https://github.com/rafaeelaudibert/conferences_insights_database`. The cumulative graphs for the entire Authors Citation dataset, not restricting it by conference, were also made available in the same repository, without computing the centralities. We can find the statistics for the size of each graph dataset in Table 3.

3.3 Types of Graphs

The graphs were built in Python using *networkx* Hagberg et al. [2008] which provides an easy interface to build various types of graphs, including multigraphs with directed edges, which we routinely use.

All graphs below are based on the data shown in Figure 7.

	Graph	*Nodes*	*Edges*
CS Conferences	ACi	104179	5654596
	ACo	104179	621644
	PC	89102	486373
	APC	193281	759386
	CC	93	4776703
Full DBLP	ACi	3655049	210362459

Table 3: Graph Statistics for the cumulative data

3.3.1 Author Citation Graph

This is a directed multi-graph, where every author is a node. An edge $e_{u,v}$ represents a paper from author u having a citing to a paper by author v. As author u can have more than one paper citing a paper by author v there might be more than one edge between the nodes, therefore we have a multi-graph. Also, authors might cite another paper from themselves, therefore we might have self-loops.[37]

Because of the way our data is organized, when we are iterating over the papers we have only the id of the papers that were referenced, but not the ID of the authors in the other papers. So, we first create a hash table with keys as the paper IDs and the value as the authors of that paper. We use this as a lookup table to identify which authors should be connected when we are iterating over the papers. See Algorithm 2 to see how this works when building the graph.

The above means that we first need to iterate over all papers and create this huge lookup table. In practice, because you cannot cite papers that have not yet been published, we split the papers into buckets by the year they were published and iterate in ascending years, which will make us keep only the "past" papers in this hash table. Algorithm 1 shows the year bucket-splitting algorithm and Algorithm 2 shows how we build this graph, with this more efficient hash table where at any year y we only have papers from years $i <= y$ in the hash table. Although at the end of the process, the table has the same size as it would have if we had built it from the beginning, this method increases local consistency improving cache results when we are iterating over the first years making this process more efficient.

Note that we might not have data for the cited paper because we are filtering the data out for only a few conferences. In this case, we simply do not include this paper.

[37]The most recent version of the code for this graph generation process can be found in `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/generate_authors_citation_graph.py`.

```
[
  {
   id: '1',
    title: 'Survey about Graphs',
    authors: [
      { id: '1', name: 'John Doe', org: 'MIT' }
    ],
    venue: { raw: 'Some Conference' },
    references: [],
    year: 1967
  },
  {
   id: '2',
    title: 'Survey about Bigger Graphs',
    authors: [
      { id: '2', name: 'Mary Jane', org: 'UFRGS' },
      { id: '3', name: 'Jane Carl', org: 'TU KL' },
    ],
    venue: { raw: 'Some Conference' },
    references: ['1'],
    year: 1970
  },
  {
   id: '3',
    title: 'Survey about Huge Graphs',
    authors: [
     { id: '2', name: 'Mary Jane', org: 'UFRGS' }
    ],
    venue: { raw: 'Some Conference' },
    references: ['1', '2'],
    year: 2003
  }
]
```

Figure 7: Sample data for graphs

Figure 8 shows an example of such a graph, given the input data from Figure 7.

3.3.2 Author Collaboration Graph

This is an undirected graph, where every author is a node. In this graph, an edge $e_{u,v}$ represents that u and v worked together in at least one paper.[38]

[38]The most recent version of the code for this graph generation can be found in `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/generate_collaboration_graph.py`.

Algorithm 1 Bucket-splitting paper per year

Require: L ▷ List of papers such as the example in Figure 7

```
papers_per_year ← empty hashtable

for year = 1969...2018 do
    papers_per_year[year] ← empty list
end for

for paper ∈ L do
    papers_per_year[paper.year] ≪ paper        ▷ ≪ means append
end for

return papers_per_year
```

Graph generated given the input data from Figure 7

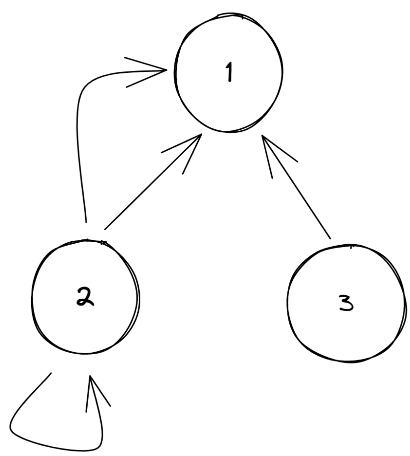


Figure 8: Example of author citation graph

This graph is easier to generate compared to the Author Citation Graph (Section 3.3.1) because data is local and we do not need to iterate twice over the data to generate a lookup table: we can simply iterate over all papers and then connect all co-authors in a clique.

Figure 9 shows an example of such a graph, given the input data from Figure 7.

Algorithm 2 Author Citation Graph

```
Require: papers_per_year                ▷ Hash table as returned by Algorithm 1
  G ← new graph with empty V and E
  old_papers ← {}

  for year = 1969...2018 do
      papers ← papers_per_year[year]
      for paper ∈ papers do
          old_papers[paper.id] ← id of every author in paper.authors
      end for

      for paper ∈ papers do
          for author ∈ paper.authors do
              G.V ← G.V ∪ {author.id}
          end for

          for citation_id ∈ paper.references do
              if citation_id ∈ old_papers.keys then
                  for cited_author ∈ old_papers[citation_id] do
                      for author ∈ paper.authors do
                          G.E ← G.E ∪ { (author.id, cited_author.id) }
                      end for
                  end for
              end if
          end for
      end for
  end for

  return G
```

3.3.3 Papers Citation Graph

This is a directed graph, where every paper is a node. A directed edge $e_{u,v}$ means that paper u cited paper v. Similar to the Authors Citation Graph we need to create a lookup table, using the same incremental procedure of loading in the lookup table data only for years $i <= y$ when iterating over year y. [39]

[39] The most recent version of the code for this graph generation can be found in `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/`

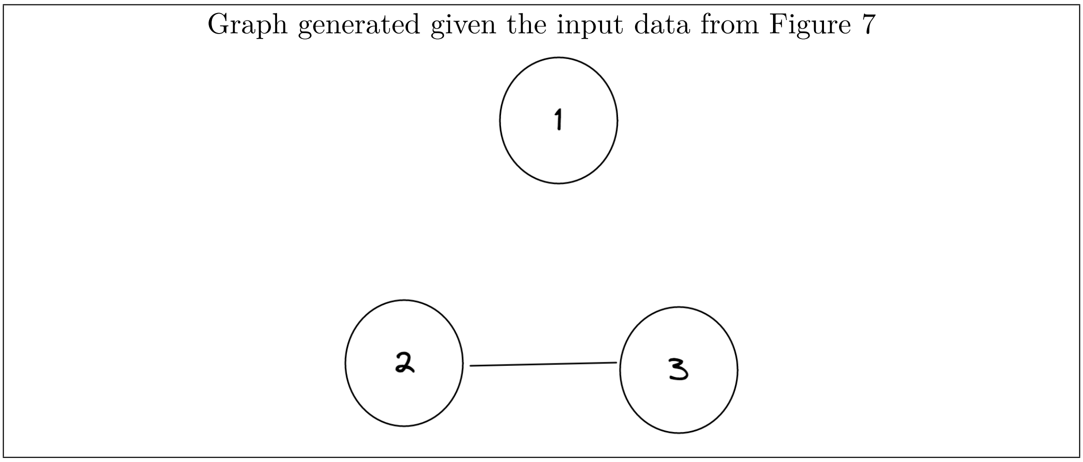


Figure 9: Author collaboration example graph

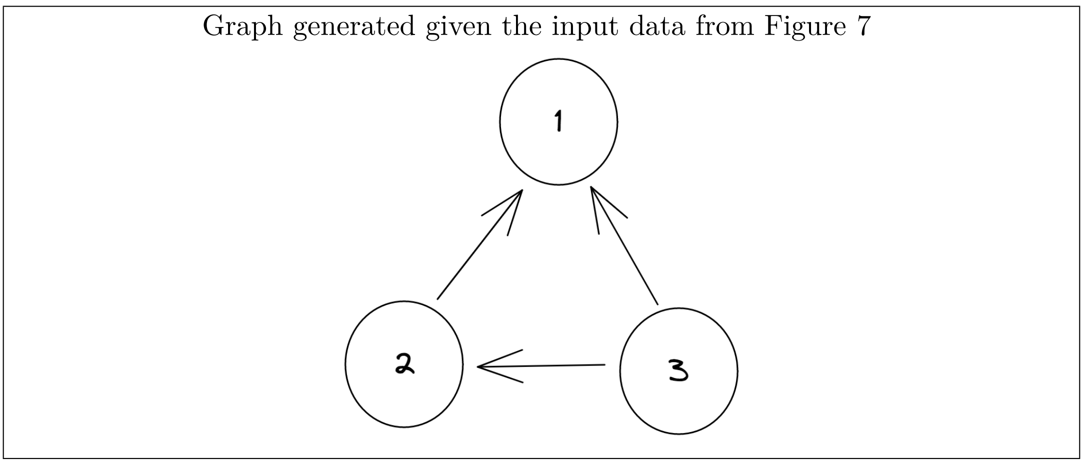


Figure 10: Paper citation example graph

Figure 10 shows an example of such a graph, given the input data from Figure 7.

3.3.4 Author-Paper Citation Graph

This is a directed labeled graph, where nodes can be either an author or a paper, and we can have edges between papers or between authors and papers, therefore this graph is more complex than the previous ones because it can represent both a paper citation network and an author citation network (through intermediate paper

---

`generate_citation_graph.py`.

nodes).[40]

This graph is built based on the *Papers Citation Graph*, with the already existent nodes being from the type *paper* V_P, and the already existent edges being from the type *citation* E_C. After, we add a node with type *author* V_A for each author, with a directed edge with type *authorship* E_A for each paper they authored.

This graph is ideal for a full picture of the data, with the possibility of inferring every possible interaction in it. Therefore, it is an ideal representation for knowledge representation tasks or even recommender systems. This is discussed in more detail in Section 5.2.

Figure 11 shows an example of such a graph, given the input data from Figure 7.

3.3.5 Country Citation Graph

This is a directed multigraph, where each node represents a country, and an edge $e_{u,v}$ represents that an author from country u cited an author from country v in a paper. Because of this two nodes might have many edges between them.[41]

After we have figured out which country an author is from (Details in Section 3.4) we can create this graph by doing the same procedure for the citation graph. Save the papers already existing by that time in a lookup table; iterate over every paper; iterate over the citations; iterate over the current paper authors and the cited paper authors; connect the country they belong to with an edge. It is possible (and quite common) to create self-loops.

Figure 12 shows an example of such graph, given the input data from Figure 7, in addition to the following mapping from organizations to countries: MIT[42] → USA; UFRGS[43] → Brazil; TU KL[44] → Germany.

3.4 Affiliation x Country mapping

It is important to note that the Arnet v11 data we collected does not always provide the country of an author in its "org" field, containing only the organization they belong to – it sometimes does not even provide the organization – which poses a problem.

[40] The most recent version of the code for this graph generation can be found in `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/generate_authors_and_papers_graph.py`.

[41] The most recent version of the code for this graph generation can be found in `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/generate_country_citation_graph.py`.

[42] `https://www.mit.edu/`

[43] `https://www.ufrgs.br`

[44] `https://www.uni-kl.de/`

Graph generated given the input data from Figure 7

Red nodes indicate they have type V_P;

Red edges indicate they have type E_C;

Blue nodes indicate they have type V_A;

Blue edges indicate they have type E_A;

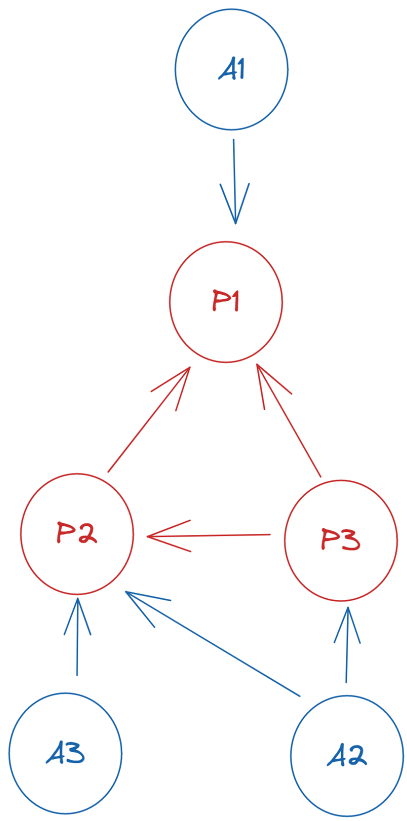


Figure 11: Author Paper Citation example graph

The "organization" field present in the data is in free-form format, which means that it does not have a clear structure from which we can extract the country of an author. Even worse, it might not even be a university name, as both companies and non-affiliated individuals can have papers in flagship venues. There is some structure in it for most of the data, though, so we have developed a pipeline where we iteratively try to detect an organization's country of origin.

In our pipeline, we first preprocess the organization by following Algorithm 4 removing cluttering and using only the text after the last comma – ideally where

Graph generated given the input data from Figure 7

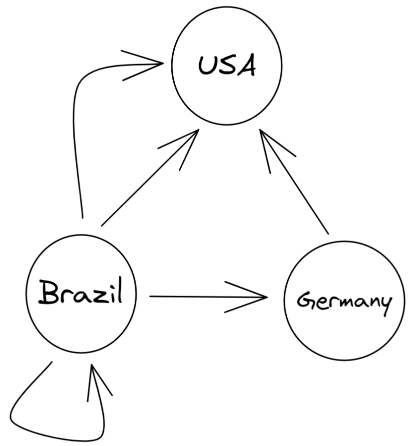


Figure 12: Countries citation example graph

the country of affiliation should be. After, Algorithm 3 is followed. We try matching the text against a lookup table that maps organizations to countries. If there's a miss, we split the text into spaces and try matching only the first word to the table, and after only the last word. If that still does not work we try matching the text without the preprocessing step.

In the end, if everything fails, we check if we matched that author previously. That is our last resort because remember that the author might change organization (and even country) throughout their academic career, so we cannot trust an author will still be in the same organization as they were the last time they published something.

The above process can be seen in the *infer_country_from* function[45].

We do have another important step not fully explained in the steps above: how we created the "lookup table" to map from organizations to countries. We manually created it over the span of 2 months, through a manual iterative labor-intensive process: manually looking at the organizations not matched using Algorithm 3 and mapping them to the countries they belong to using both our own knowledge and web searches to filter the options down.[46]. The mapping for every organization that has ever been published in AAAI, IJCAI, and NeurIPS is complete, and the process

[45] `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/generate_country_citation_graph.py`

[46] This mapping is available at `https://github.com/rafaeelaudibert/TCC/blob/v11/graph_generation/country_replacement.json`

Algorithm 3 Organization to Country Mapping

```
Require: raw_org                                    ▷ Organization name
Require: org              ▷ Organization name preprocessed by Algorithm 4
Require: author_id
Require: T                ▷ Lookup table matching organization to country
Require: PT                    ▷ Past author to organization matchings
  if org ∈ T.keys then            ▷ Check if preprocessed org is in the table
      return T[org]
  end if

  split_org ← split("org", " ") ▷ Split the text into every space, turning it into a list
  if split_org[0] ∈ T.keys then      ▷ Check if first name in org is in the table
      return T[split_org[0]]
  end if
  if split_org[-1] ∈ T.keys then      ▷ Check if last name in org is in the table
      return T[split_org[-1]]
  end if

  if raw_org ∈ T.keys then   ▷ Check if org without preprocessing is in the table
      return T[raw_org]
  end if

  if author_id ∈ PT.keys then    ▷ Check if we have already matched this author before
      return PT[author_id]
  end if

  return ∅
```

to map this for the other conferences is still ongoing. We hope this mapping can be used in the future by other works to facilitate the inference of a country from an organization. Figure 13 shows how many organizations we mapped per country – the USA does not fit in the figure for scale purposes and has a value of 2163.

Additionally, there are a few authors whose "org" field is empty. For the first years of the area (1969-1979), we did not have many papers being published, so we manually looked at every single paper with an empty organization field and generated another lookup table available at `https://github.com/rafaeelaudibert/TCC/blob/v11/graph_generation/author_country_replacement.yml`. We then

The USA does not fit in the figure; and has a value of 2163.
This does not reflect the true count of different organizations per country, because some of them could be easily identified by their country, and did not need any special treatment.

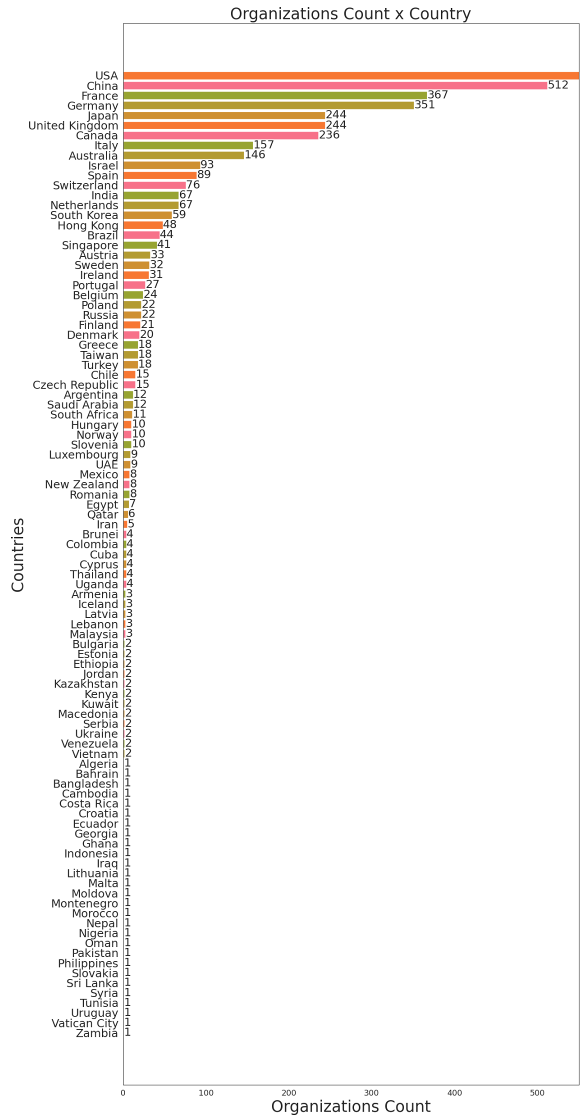


Figure 13: Quantity of **mapped** different organizations per country that appeared in our data.

check this table first before attempting the above pipeline, since it is more reliable. This table was specially built in YML instead of JSON for better readability and allows us to add comments in between the entries.

It is noticeable, though, that this problem is worse in more recent years. Arnet's data does not have organizations for most papers published from 2018 onward, so the problem is bigger in recent years. For example, in Figure 36 the "None" stacked part is bigger in recent years.

4 Data Analyses

This section presents the main analyses and insights performed on our datasets described in Section 3. We present initial statistics in Section 4.1, then analyse each graph (Sections 4.2 to 4.6). We then investigate the research impact of Turing Award winners in Section 4.7.

As already stated before, the full code for both the data generation and data analysis was made publicly available at `https://github.com/rafaeelaudibert/TCC/tree/v11`. The main code is in the branch *v11* because of the aforementioned problems with Arnet's V13.

4.1 Raw Data

Although the bulk of this work is intended to revolve around the graph datasets and their centralities built to support our claims, the raw data itself is also able to provide us with great introductory information to the following sections.

Figure 14 shows a boxplot with a rising trend in the number of authors per paper over the years. In this boxplot graph, the red dot represents the average number of authors per paper, the black line represents the median, the box per se represents the 95% percentile, while the black lines represent the 99% percentile – even showing a failure in the dataset with some papers with 0 authors in the late 1960s. The figure shows how the trend of several authors in a single paper, like Brown et al. [2020], Jumper et al. [2021], and Silver et al. [2016], is recent and rare with not more than 1% of the papers having 7 or more authors since 2004. It is noticeable how the average value jumps to almost 4 in the years past 2014.

We also intersected the authors who published in the same year in different venues. Some interesting trends arose, such as AAAI and IJCAI have the biggest overlap in authors than any combination of them with NIPS and ACL (Figure 15); CVPR has congregated more authors than NIPS and IJCAI since the beginning of the 2000s and its biggest authors overlap is always with NIPS (Figure 16); SIGIR had almost no overlap with these three conferences during the 90s and still has very

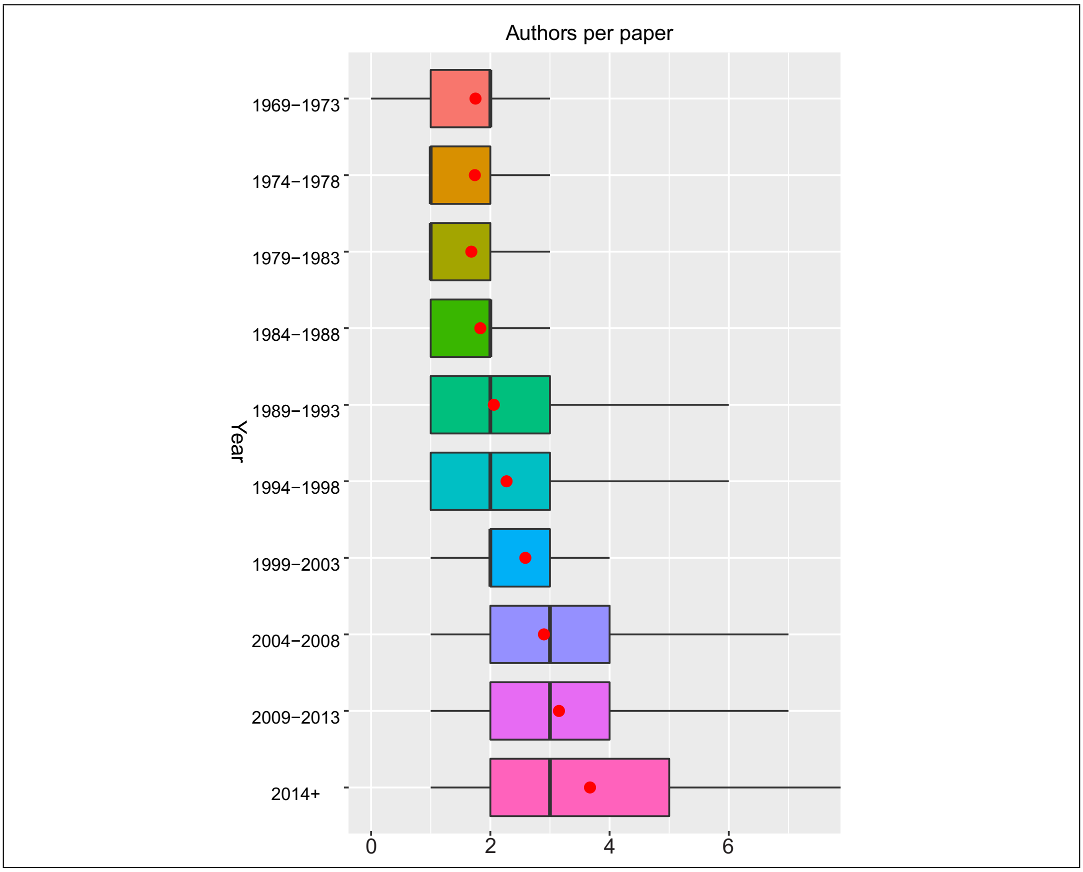


Figure 14: Boxplot of the number of different authors for each single paper per year

little overlap nowadays, despite an increase in its intersection with AAAI (Figure 17).

4.2 Author Citation Graph

4.2.1 Ranking over time

We have calculated an authors ranking regarding the aforementioned centralities from 1969 until 2019 using the accumulated citation data AC graph.

Figures 18, 19, and 20 show the evolution of PageRank, Betweenness, and In-Degree centralities, respectively, in our Author Citation Graph. In these figures, a line represents a single author and its ranking evolution over time in some predefined years (chosen to be 1969, 1977, 1985, 1993, 2001, 2009, and 2014). The only authors

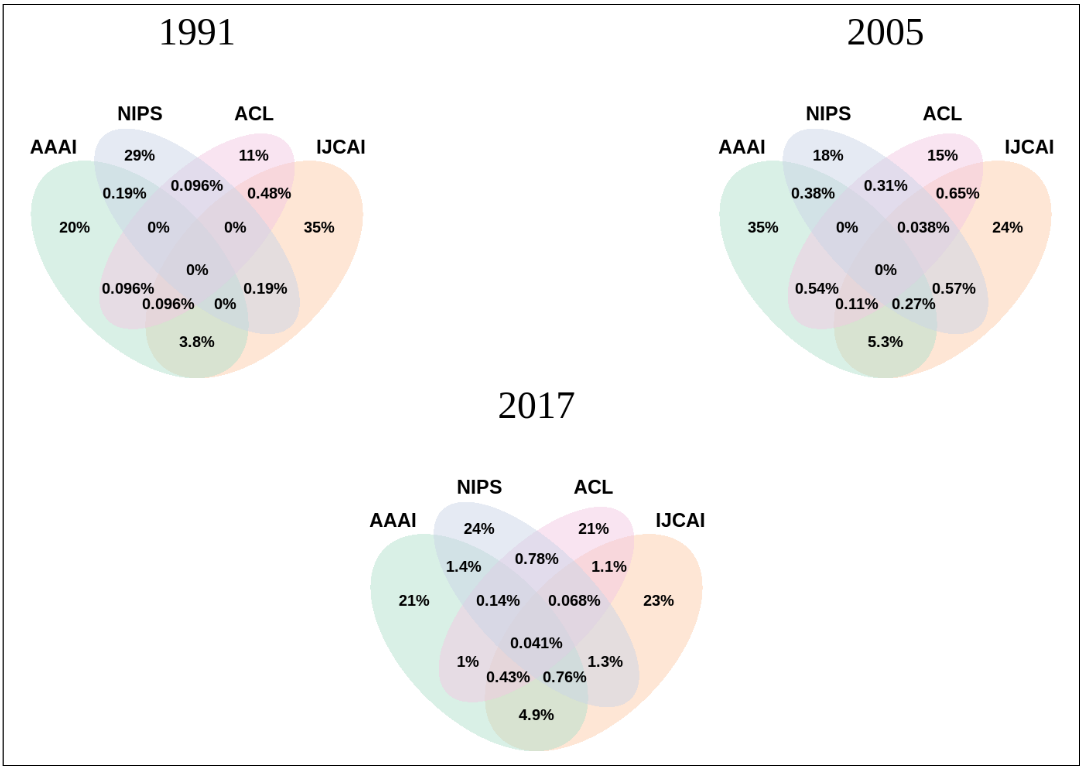


Figure 15: Percentage of overlapping authors in AAAI, NeurIPS (NIPS), ACL, and IJCAI.

shown are those who, at any point in one of these years, reached the top 10 in that specific rank. Authors who had not published yet in one of these years and, therefore, did not have any rank yet, show as *N/A*.

Although they seem chaotic, these graphs do have some interesting insights. Figure 18 is an interesting starting point because it is considerably stable, at least at the top of it. Harry Pople was the top 1 author in this ranking at least from 1977 until 2001, the longest period one will hold this position in any of our analyses. His main work is focused on Artificial Intelligence in Medicine therefore very central in-between different areas Dhar and Pople [1987]. Also in the PageRank graph, one might see that the rises tend to be meteoric with Andrew Ng going from position 974 in his debut year of 2001 to 16th 8 years later, and then 2nd after 5 more years. The same can be said for most of the dynamics present in this graph.

The aforementioned insights also hold for Figure 19 where Betweenness is analysed. This graph is a lot less stable than PageRank's, as betweenness is easier to

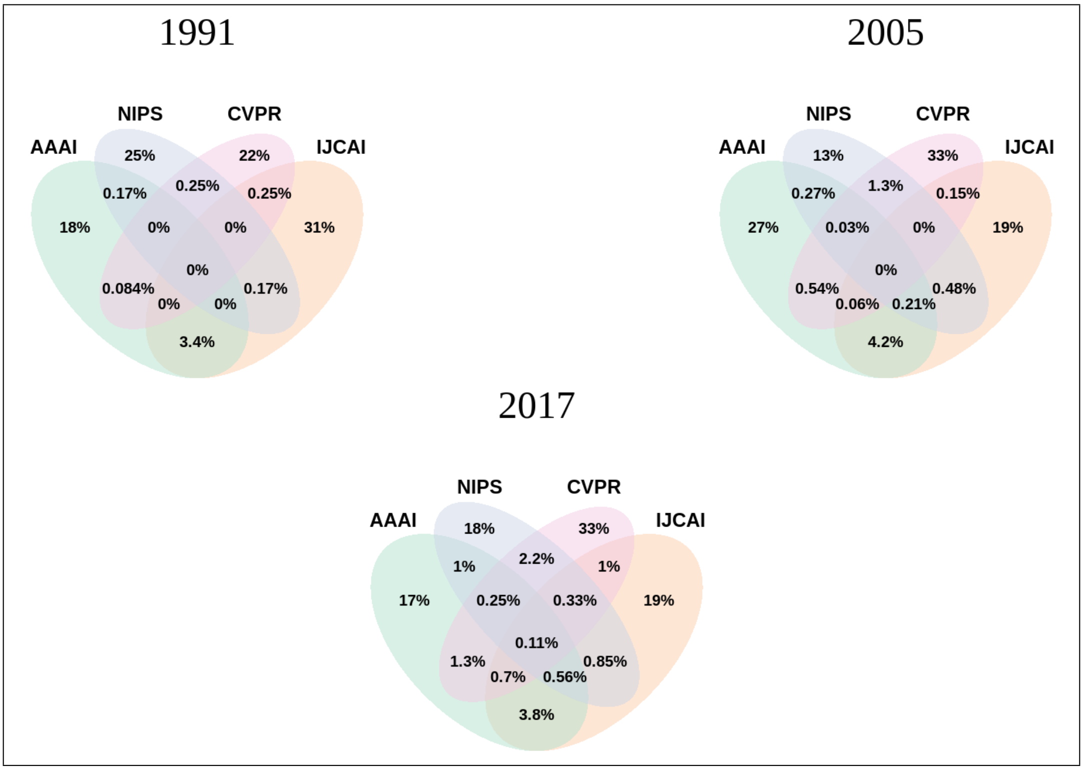


Figure 16: Percentage of overlapping authors in AAAI, NeurIPS (NIPS), CVPR, and IJCAI.

evolve when new areas in Machine Learning happen, therefore changing the flow of information in the graph, while PageRank will be more stable because important people at one time will continue to be as important as they were forever, only going down in rank if someone even more influential appears. One can see this dynamic, for example, by looking at the last position in both charts: Larry Tesler – the one but last in the PageRank chart because the last position is an outlier – is 4267th in the PageRank, while the last position in the Betweenness chart is 31159th, showing how low one might drop in the Betweenness ranking even though they once were in the top 10 most influential scientists, *in the datasets analysed here.*

The Indegree chart shows a basic and raw data point: which author is the most cited, which should reward older authors with seminal papers. The first place in this ranking belongs to Andrew Zisserman, author of papers such as Simonyan and Zisserman [2014] and Hartley and Zisserman [2003], having close to 300,000 citations over his whole life – more than 100,000 of those only for the 2 cited papers. The

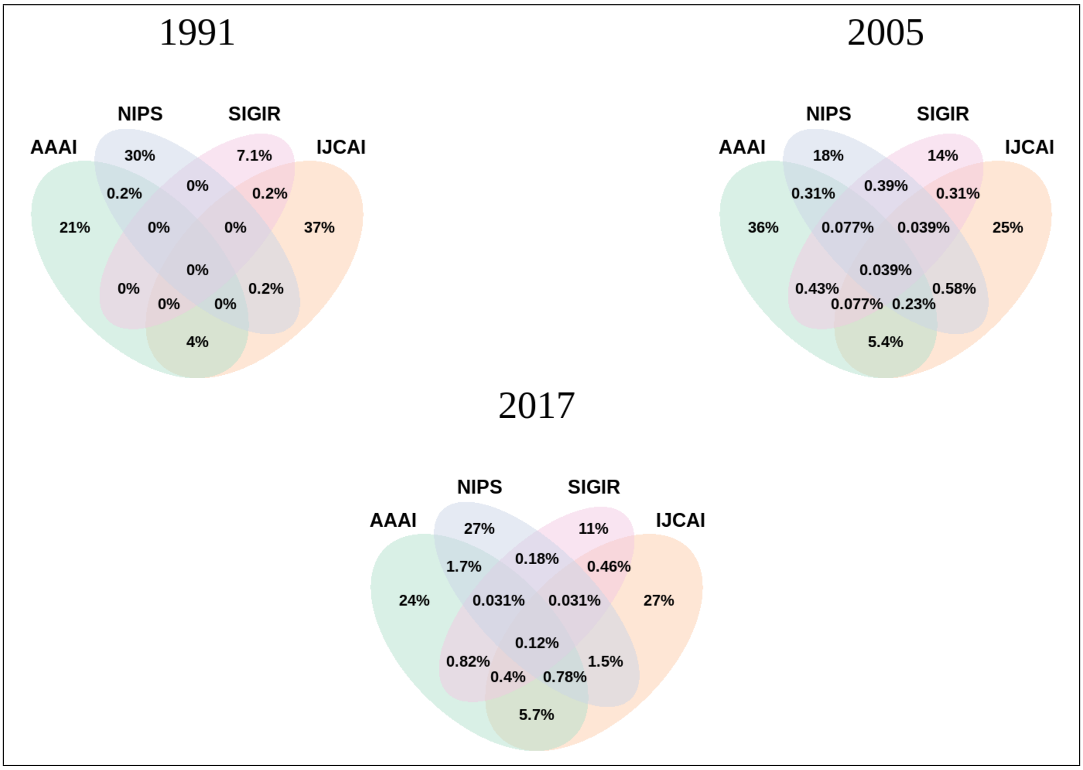


Figure 17: Percentage of overlapping authors in AAAI, NIPS, SIGIR, and IJCAI.

second position is Andrew Ng with just over a third of the number of citations that Zisserman has.

Considering all these charts together, it is interesting to see how Andrew Ng is the most influential overall author in AI when we analyse it from a citation perspective, *in the datasets analysed here*. He is the author of papers such as Blei et al. [2003], and Ng et al. [2001], having an h-index of 134, i.e. 134 papers with at least 134 citations (See H-Index on Section 2.4.6 to understand how this metric is computed), the 1403rd biggest h-index in Google Scholar[47]. He appears with the biggest betweenness value, and second-biggest indegree and PageRank ranking.

Takeo Kanade, the first position in the page rank ranking, is only 14th and 16th when looking at betweenness and indegree, respectively – although it is worthy of note that in 2001 he was first in in-degree and betweenness while third in PageRank. This is the best result, on average, that can be found in our results. Similarly,

[47] https://www.webometrics.info/en/hlargerthan100

Andrew Zisserman, the first position in the in-degree ranking, is sixth when looking at betweenness, and 8th on the PageRank ranking.

The ranking for the other computed centralities can be seen in Appendix C.

4.2.2 Self-citations

Authors might build up in their previous work, which would introduce self-edges in our graph representing self-citations. Figure 21 shows a boxplot of the evolution of self-citations count per year. Despite the average beginning stable at around two, increasingly more authors have been increasing their number of self-citations over the years.

This figure, however, does not represent the full truth because there are more papers recently. Figure 22 shows a better view of the same data, clearly showing the average number increasing. The data has its faults because if an author can publish more than one paper per year then it will help to bring the average up by not being divided twice, but this can be said for every single year, so the increasing rate of self-citations would still exist.

4.3 Author Collaboration Graph

4.3.1 Ranking over time

We have calculated an authors' ranking regarding the six aforementioned centralities from 1969 until 2019 using the accumulated collaboration data – ACo graph.

Figures 23 and 24 demonstrate how the PageRank and Betweenness rankings, respectively, evolved over time. In these figures, we chose to plot the top 10 authors each year, in an 8-year interval. Considering this gap, it is interesting to observe that only in 2009 it is possible to see all authors who appeared in the top 10 ranking during all the selected years. Also, most of the authors entered the ranking during the 80s and the 90s, regardless of the centrality. The remaining rankings (centralities) can be seen in the Appendix D.

4.3.2 Entering the Realm of AI

Every year several researchers publish their first papers in AI-related venues such as the ones we are analyzing throughout this work. Figure 25 shows the yearly share of new authors per conference. The stacked area contains spikes due to the fact that several conferences did not occur yearly. NIPS conference (currently NeurIPS) was the conference that mostly attracted new authors until the mid-90s together with IJCAI. Since then the share has become more and more split into conferences of

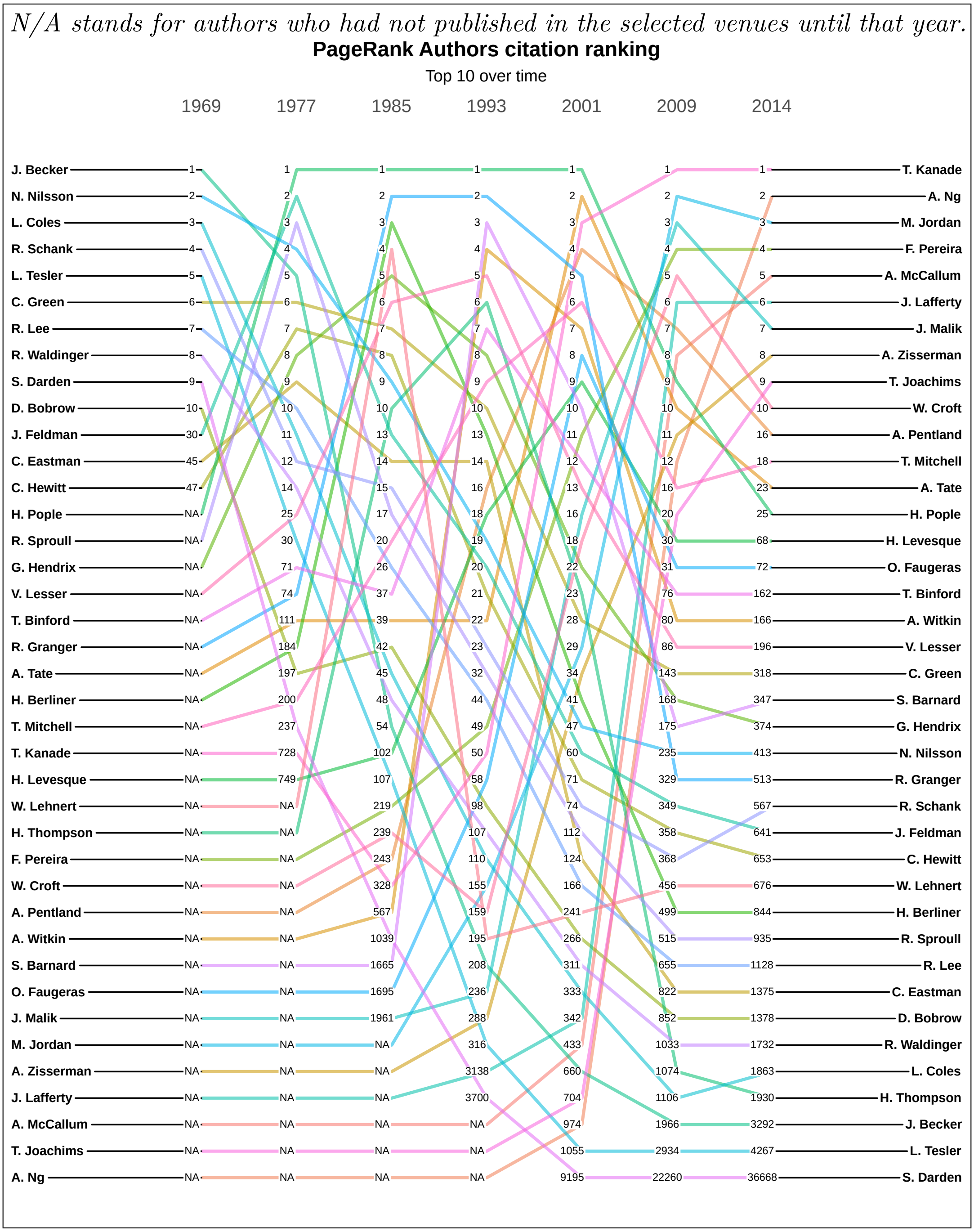


Figure 18: Author citation ranking over time according to PageRank centrality.

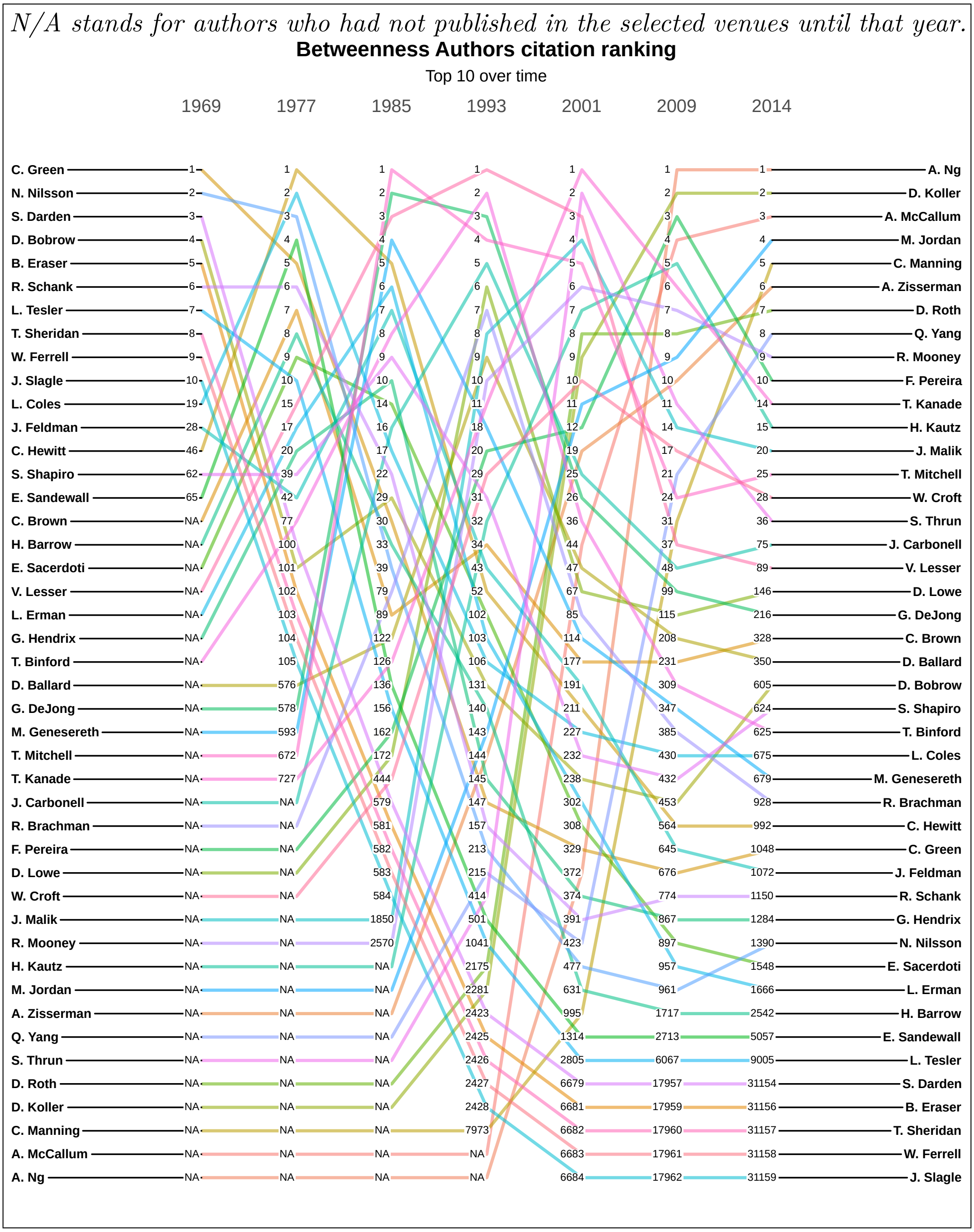


Figure 19: Author citation ranking over time according to Betweenness centrality

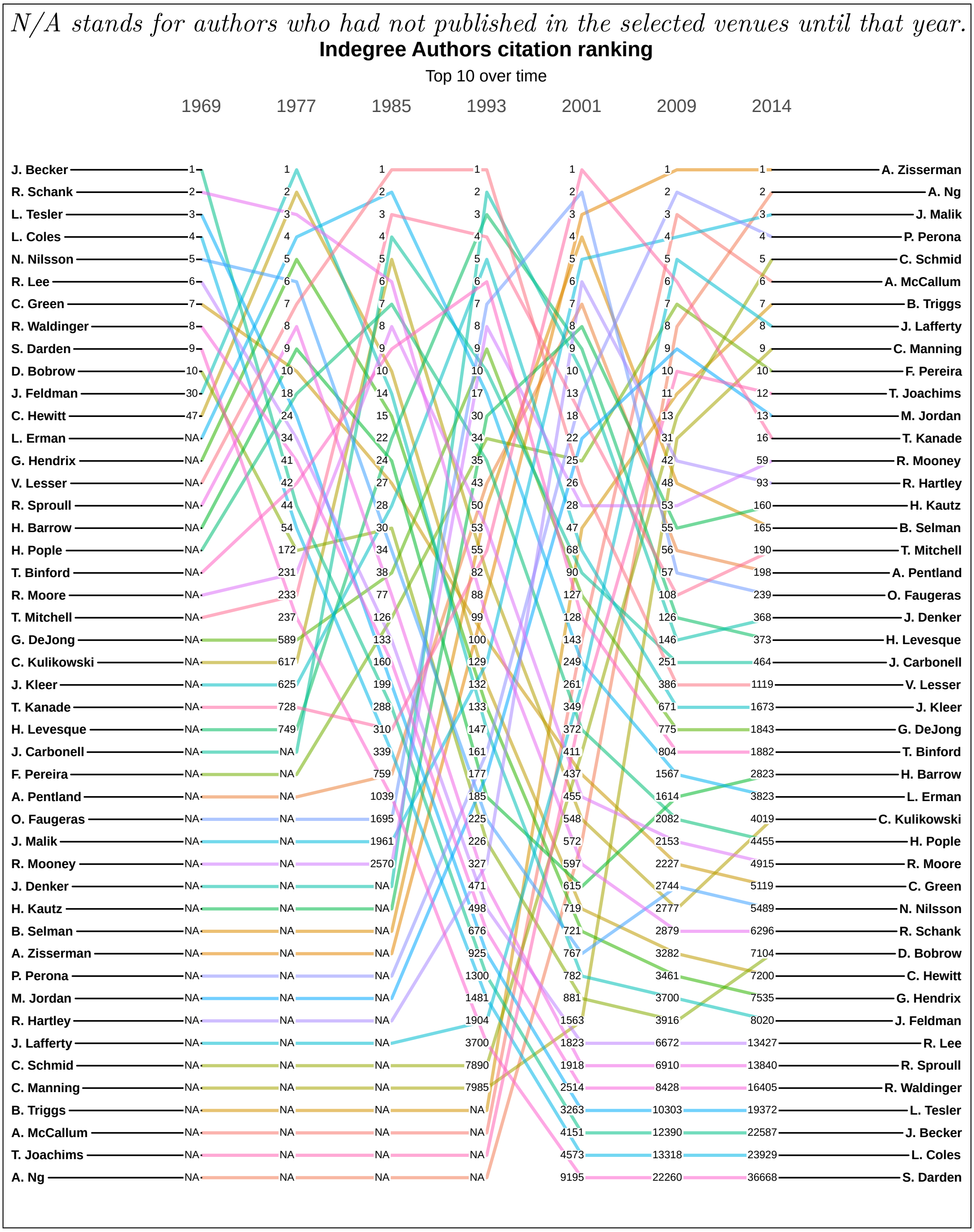


Figure 20: Author citation ranking over time according to In-degree centrality

Despite the average being stable around two, increasingly more authors have been increasing their number of self-citations over the years

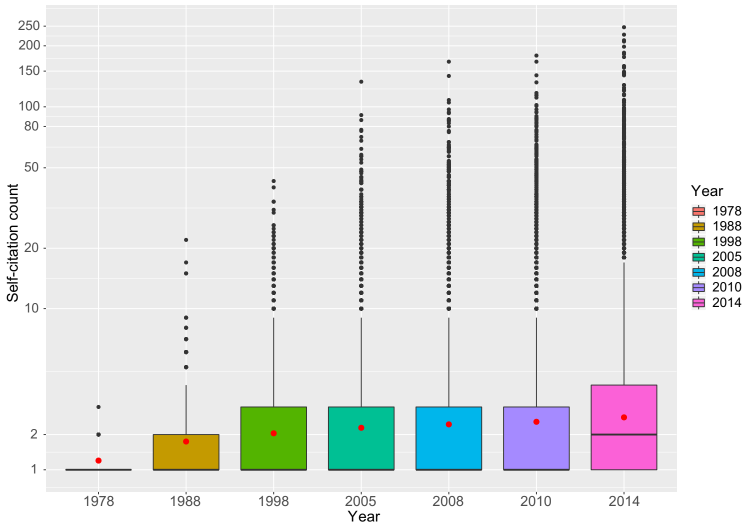
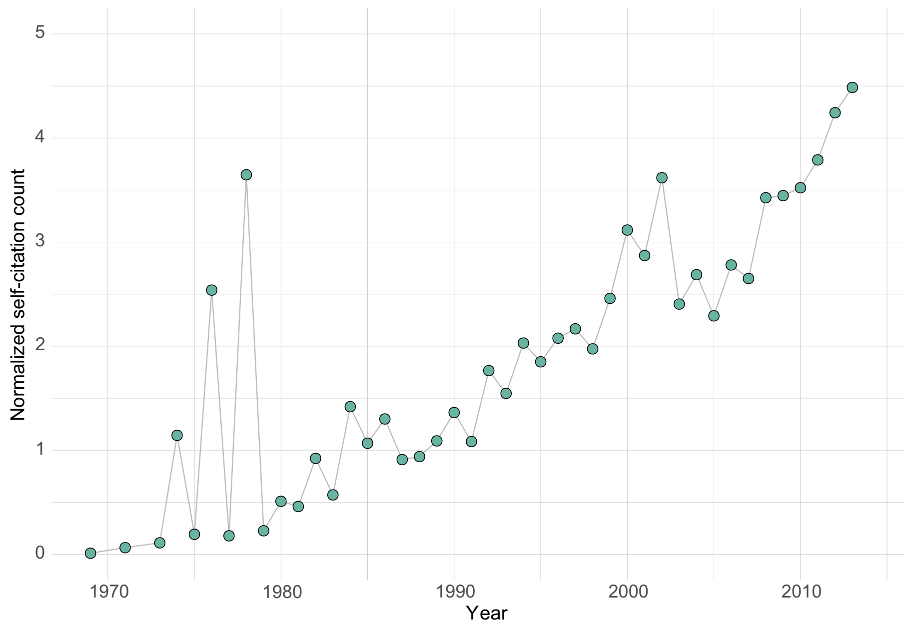


Figure 21: Boxplot of self-citation count per year

different areas. CVPR and ICCV have shown some growth in this respect in recent years, as did AAAI and WWW in the early 2000s.

Table 4 lists all authors who collaborated with more than 200 new authors since 1969. Several of them appeared in the authors' collaboration ranking (especially regarding Betweenness, Closeness, and PageRank centralities). The regular behavior, however, is better described by the average and standard deviation statistics: the average number of new authors that an author collaborated with is around 4 with a standard deviation of 12. It seems clear that these numbers are highly affected by the career age of a researcher. We estimated this age with the author's first year of publication inside our graph and we used this age to normalize the amount of collaboration with new authors, achieving a normalized average number of new authors per author of 0.3 (career time average: 11) with standard deviation of 0.94 (career time standard deviation: 9.2), which essentially means that a researcher usually brings a new author to these AI venues after 3 years of his entry into the field.

Number of self-citations divided by number of authors who published during that



year

Figure 22: Normalized self-citation count per year

4.4 Paper Citation Graph

4.4.1 Ranking over time

In every centrality measure done for this graph, whenever we plot it, we map the name of the papers to those in Table 10. This format is not ideal for readability, but it was the best method found to show this data in its full form. When it comes to citation networks, the betweenness centrality can be seen as a measure of how a node (paper) is able to connect different research areas, or how it acts to foster interdisciplinarity Leydesdorff [2007].

In this sense, Figure 26 shows how the ranking of most important papers (according to betweenness centrality) evolved. It is possible to see that the ranking itself is very volatile as no paper can remain in the top 5 for more than 2 times (inside our gap of 8 years), nevertheless the paper "Constrained K-means Clustering with Background Knowledge" Wagstaff et al. [2001] (CKCWBK01 in the figure) has been in the top 10 at least since 2009. Also, all papers in the top 5 of 2017 and 2019 were published after the year 2000, which could indicate that, despite not being seminal

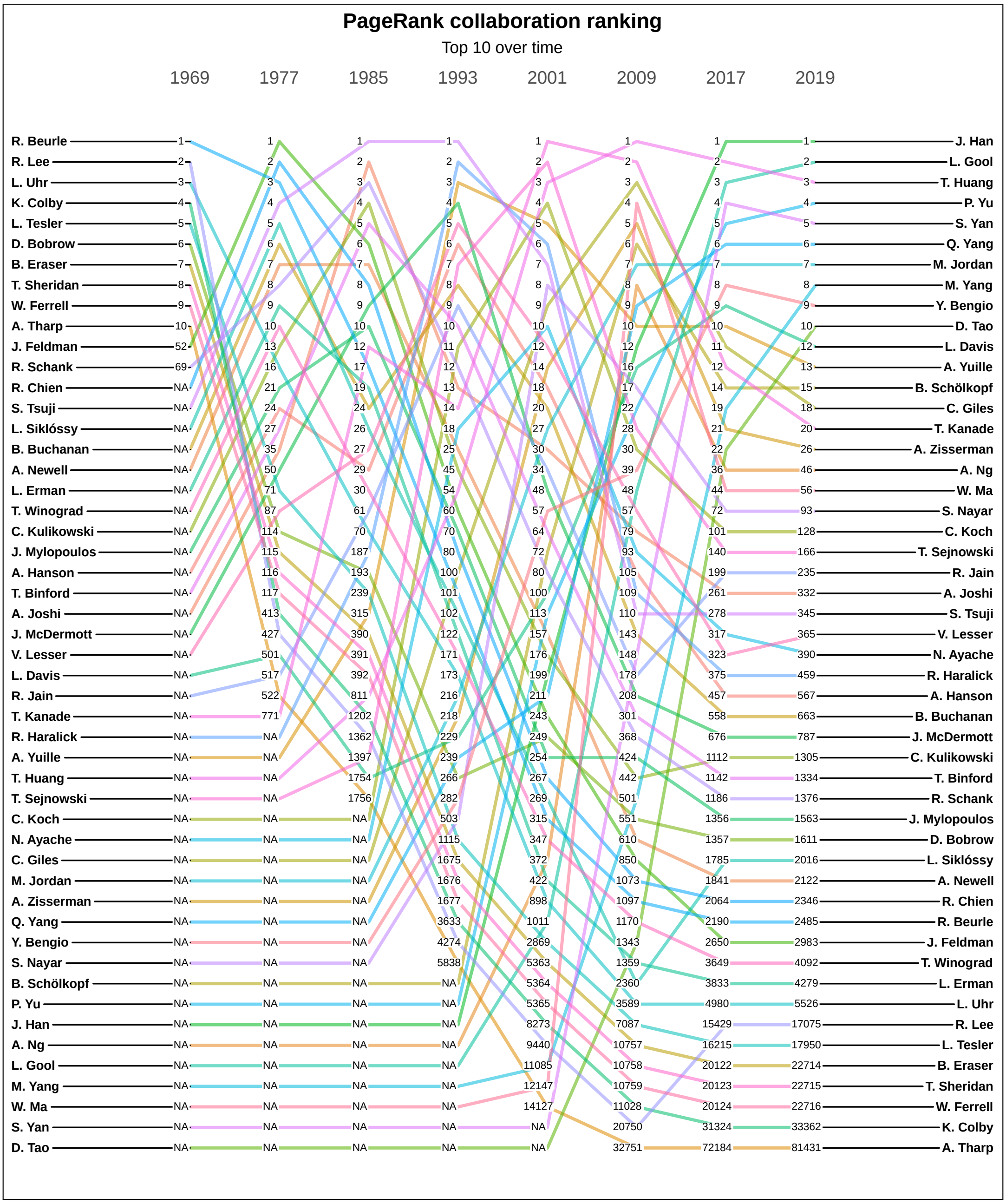


Figure 23: Authors collaboration ranking over time according to PageRank centrality. *N/A stands for authors who had not published in the selected venues until that year.*

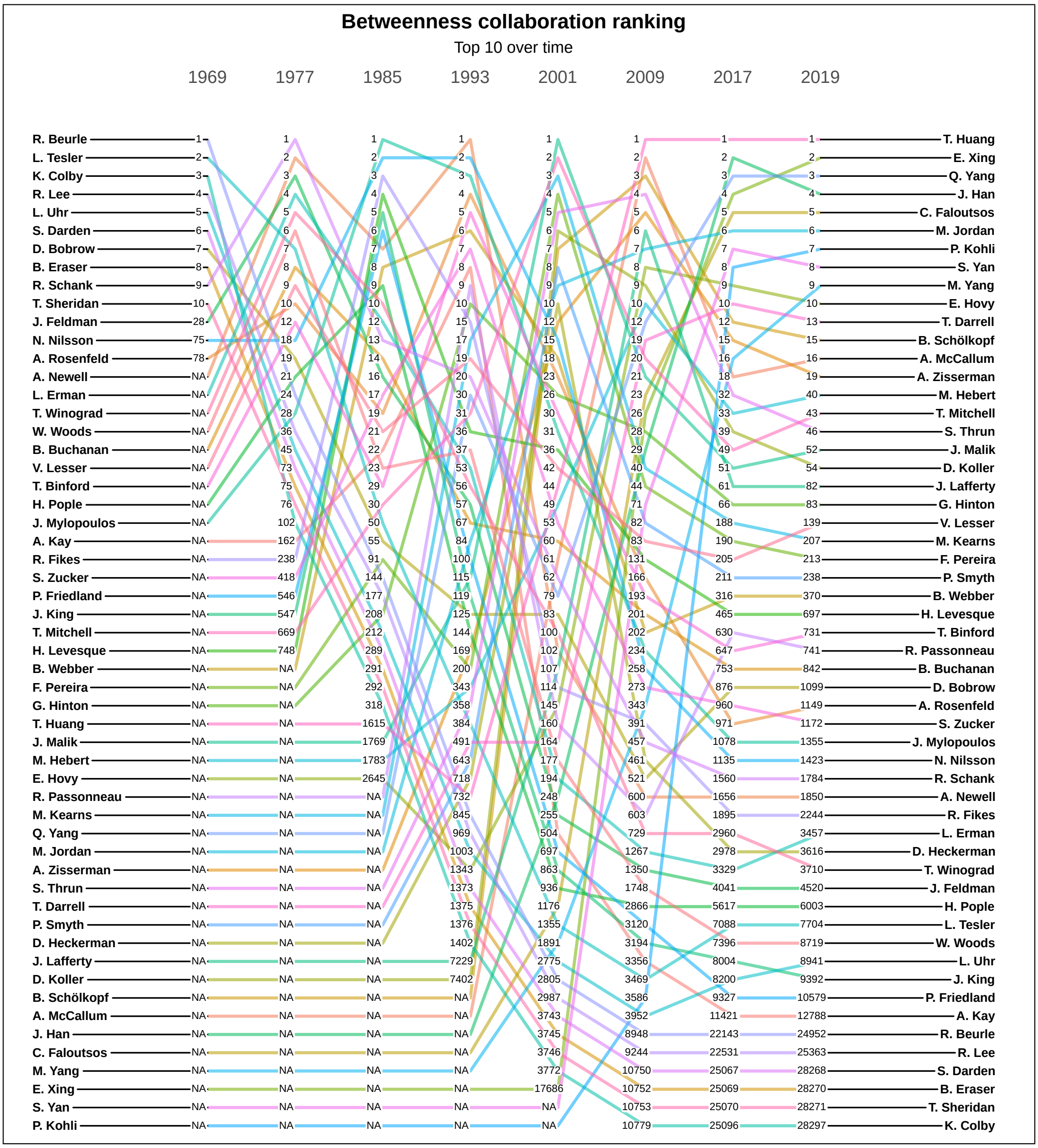


Figure 24: Authors collaboration ranking over time according to betweenness centrality.
N/A stands for authors who had not published in the selected venues until that year.

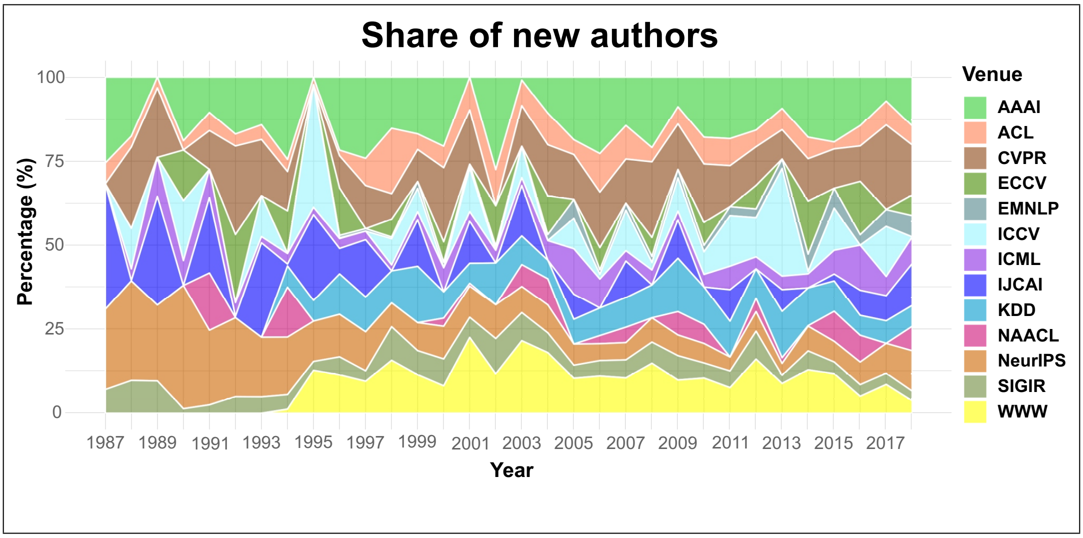


Figure 25: Share of yearly new authors per conference.

papers, these recent researches are more helpful in different areas.

Figure 27 shows the same graph data but ranked by their in-degree centrality, which simply measures how many citations a paper has received until a given year (inside our graph). The latest top 5 is composed of 3 papers related to computer vision and 2 to natural language processing. A similar pattern can be found until 1993, but back in 1985 and before most of the ranking was composed of papers that tackled reasoning, problem-solving, and symbolic learning, such as "Reasoning about knowledge and action" Moore [1977] (RAKAA77 in the figure), "A multi-level organization for problem-solving using many, diverse, cooperating sources of knowledge" Erman and Lesser [1975] (AMOFPSUMDCSOK75 in the figure) and "The art of artificial intelligence: themes and case studies of knowledge engineering" Feigenbaum [1977] (TAOAITACSOKE77 in the figure).

A very stable behavior can be seen in the PageRank ranking (Figure 28): most papers remained in the top 5 for 2 gaps (usually 8 years) and many of them for 3 gaps (16 years in the middle, 10 in the end). The paper "Towards automatic visual obstacle avoidance" Moravec [1977] (TAVOA77 in the figure) has been in the top 5 at least since 1993 and it has been leading the ranking since 2001. The second one, "Feature extraction from faces using deformable templates" Yuille et al. [1989] (FEFFUDT89 in the figure), is also a somewhat old paper related to computer vision.

Similarly, one can see the stableness and invariability to change that PageRank offers by noticing that we only had 20 different papers in the top 10 in the selected

Author	Count
Lei Zhang	488
Luc Van Gool	352
Ming-Hsuan Yang	350
Thomas S. Huang	322
Andrew Y. Ng	298
Jiawei Han	294
Dacheng Tao	282
Yang Liu	280
Philip H. S. Torr	280
Yoshua Bengio	248
Wei Wang	238
Milind Tambe	236
Yang Li	232
Ale Leonardis	224
Liang Lin	222
Qingming Huang	222
Shuicheng Yan	222
Christos Faloutsos	222
Jiri Matas	214
Michael Felsberg	212
Horst Bischof	212
Philip S. Yu	208
Richard Bowden	206

Table 4: The 23 authors who collaborated with more than 200 new authors since the year 1969

years, while we have 28 for Betweenness and Indegree, a more befitting number when compared to the figures shown in the previous sections.

The remaining rankings (Closeness, Degree, and Out-degree) can be seen in Appendix E.

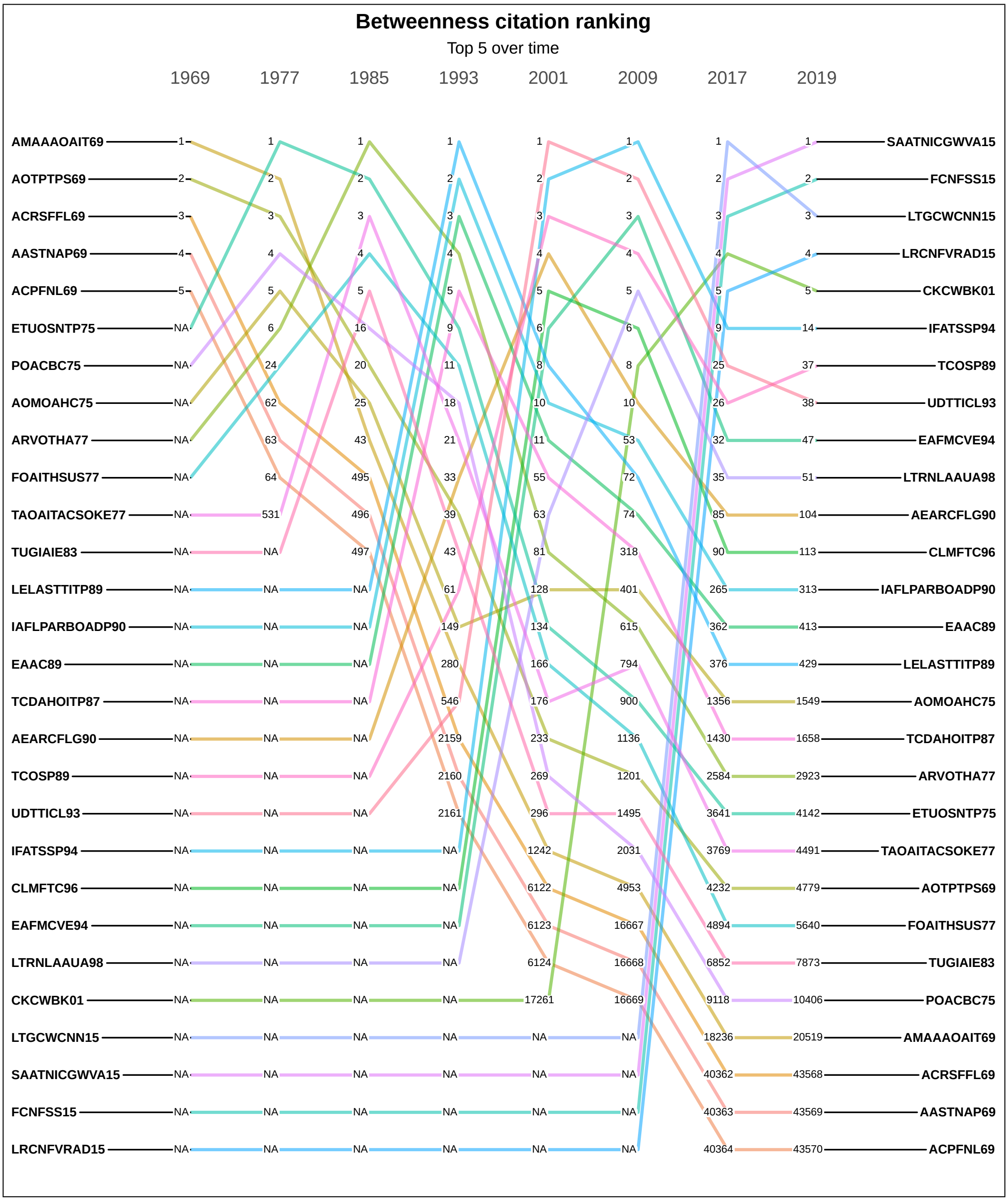


Figure 26: Papers citation ranking over time according to Betweenness centrality. *N/A stands for papers that had not been published in the selected venues until that year. Please refer Table 10 in the Appendix E to see the details of each ranked paper.*

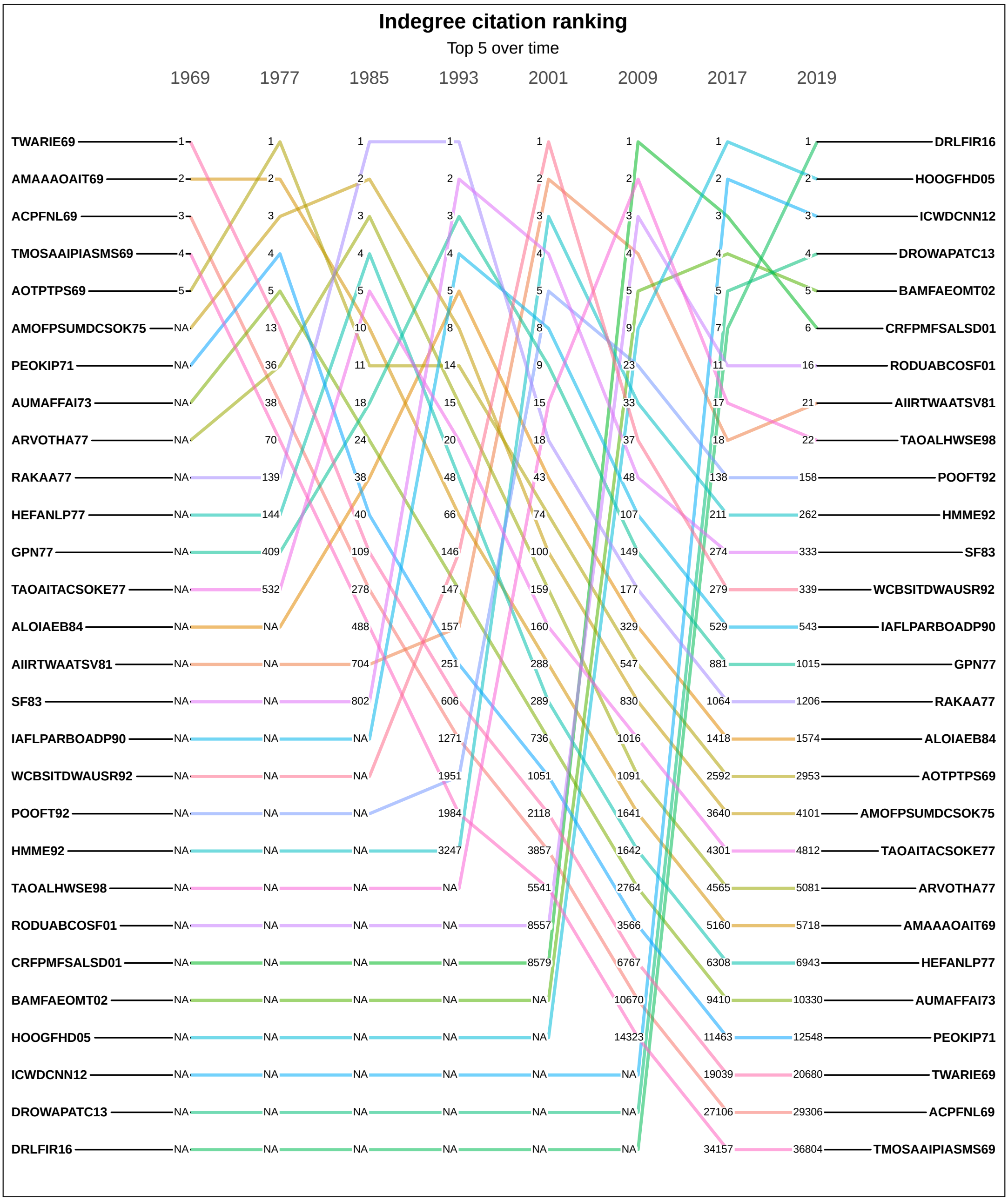


Figure 27: Papers citation ranking over time according to In-degree centrality. *N/A stands for papers that had not been published in the selected venues until that year. Please refer Table 10 in the Appendix E to see the details of each ranked paper.*

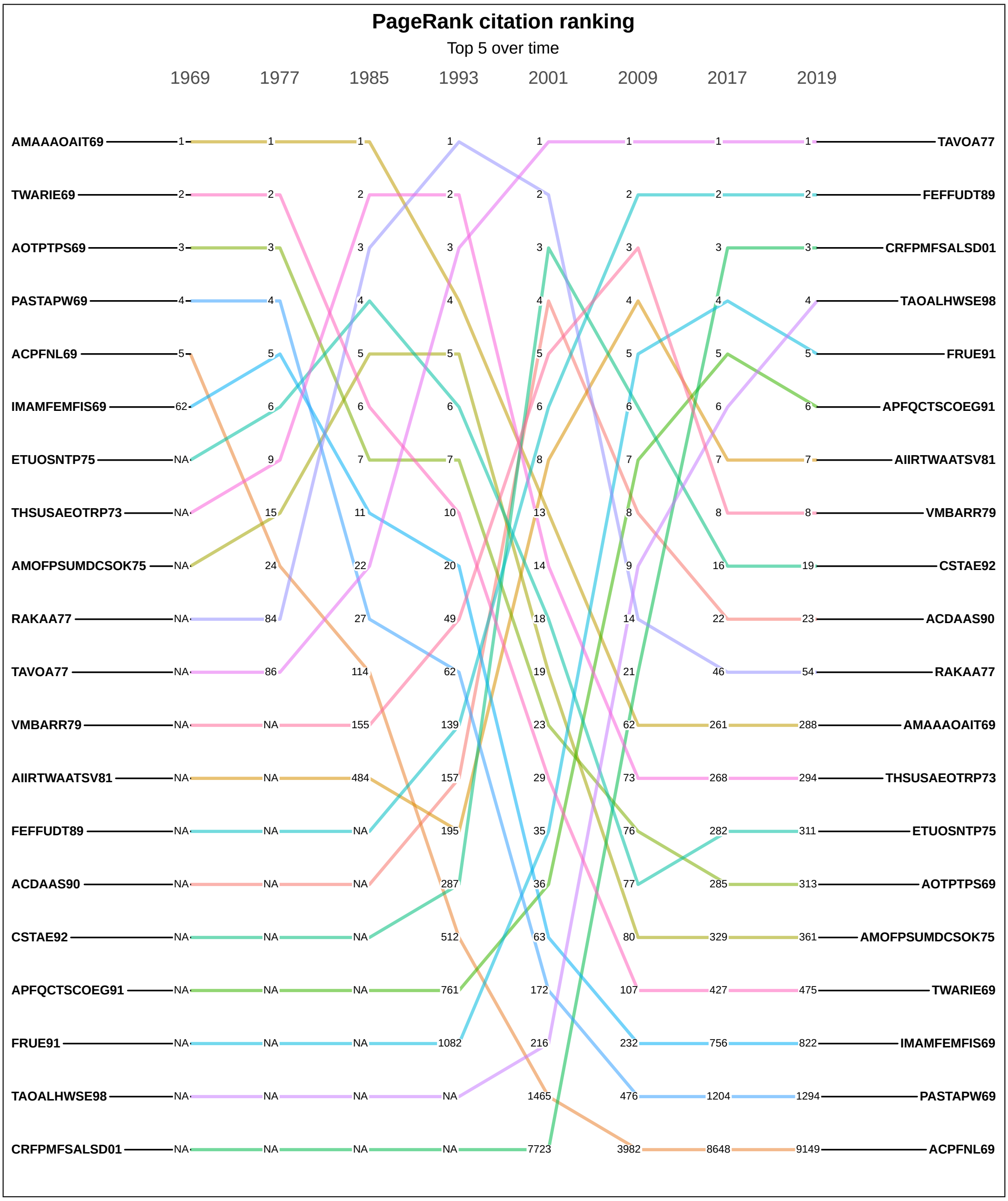


Figure 28: Papers citation ranking over time according to PageRank centrality. *N/A stands for papers that had not been published in the selected venues until that year. Please refer Table 10 in the Appendix E to see the details of each ranked paper.*

4.4.2 Share of Top 100 Ranking per Venue

Figures 29 to 31 reinforce the trend seen in the top 5 ranking in the last section: despite the centrality, computer vision-related venues are progressively gaining importance regarding their published papers, especially the CVPR. However, there has been a distinguished contribution by the ACL conference to the most important papers (according to PageRank) since 1984. These three heatmaps also show that the AAAI papers had their peak of importance during the late 1980s and the 1990s, but now they are losing their share of the ranking in the same fashion that IJCAI.

4.4.3 Share of Citations per Venue

We were also interested in how the citations of each venue have been evolving in the last few years. In this analysis, we were also able to distinguish citations to papers from arXiv, Journals, and the International Conference on Learning Representations (ICLR). For instance, back in the 1980s and early 1990s, around 50% of citations coming from NIPS papers were directed to papers from journals (see Figure 32), however, this share nowadays has been reduced to less than 25%. More than that, citations to ICLR papers and especially to arXiv papers have been increasing since the early 2010s.

A similar pattern occurs when we consider papers from AAAI and IJCAI, Figures 33 and 34 respectively. However, in their case, there is a much more divided share between all the conferences: it is possible to distinguish some influence from KDD, WWW, ACL, and EMNLP together with the increasing, yet unobtrusive, influence of arXiv and ICLR.

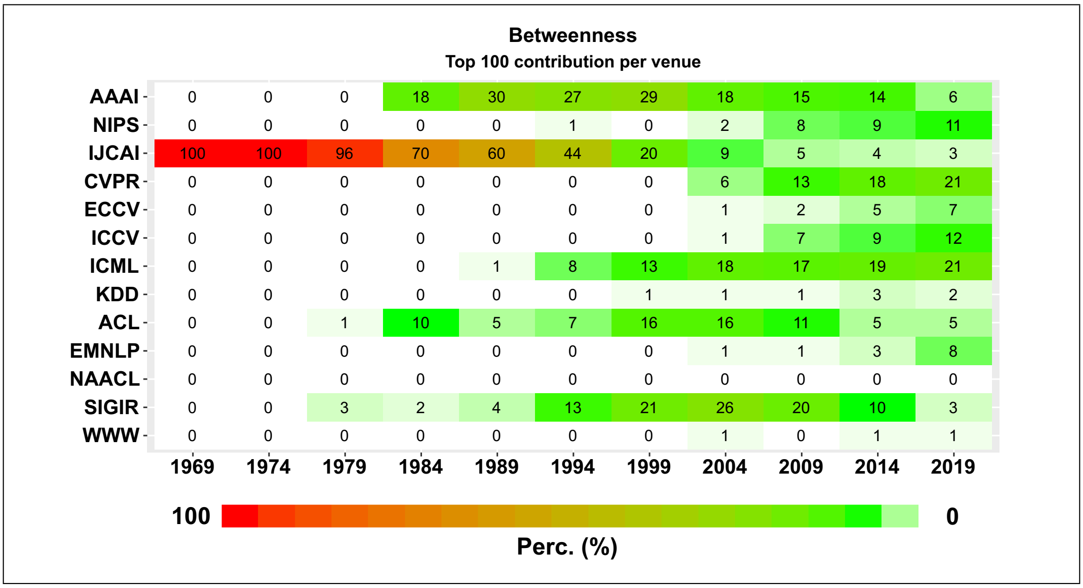


Figure 29: Venue contribution per year (accumulated) in the top 100 most important papers, according to Betweenness.

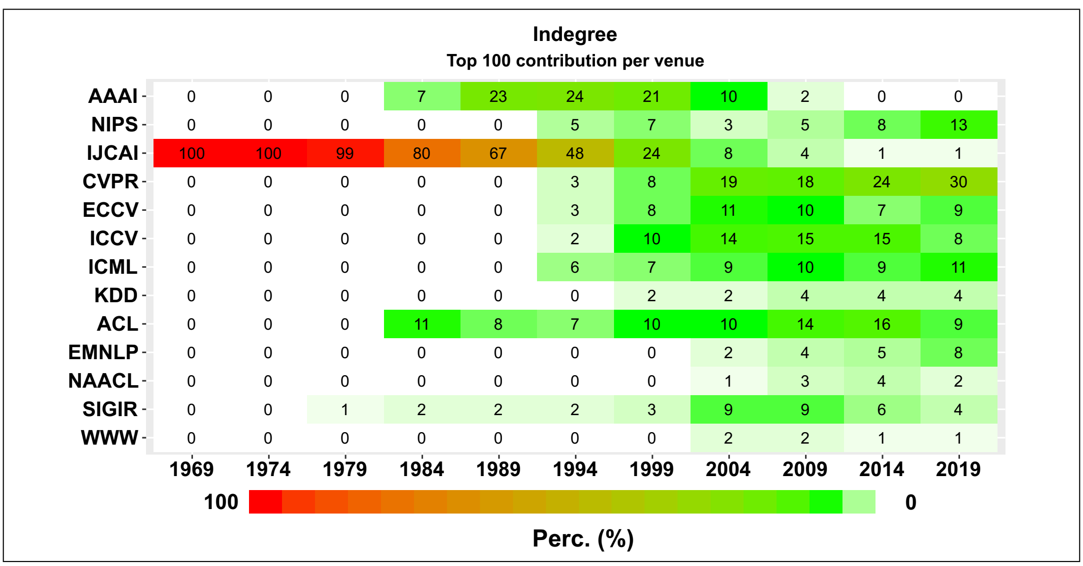


Figure 30: Venue contribution per year (accumulated) in the top 100 most important papers, according to In-Degree.

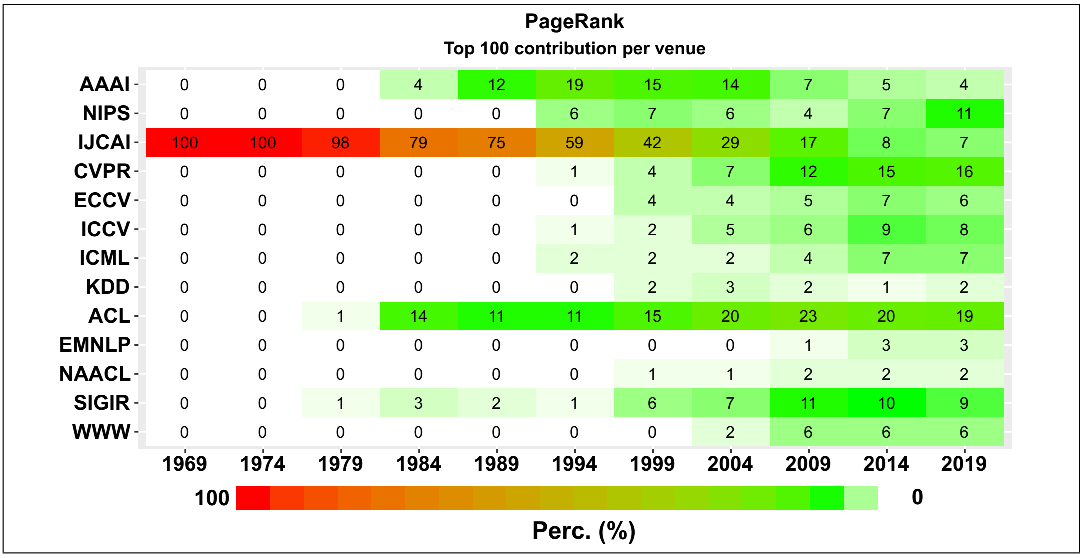


Figure 31: Venue contribution per year (accumulated) in the top 100 most important papers, according to PageRank.

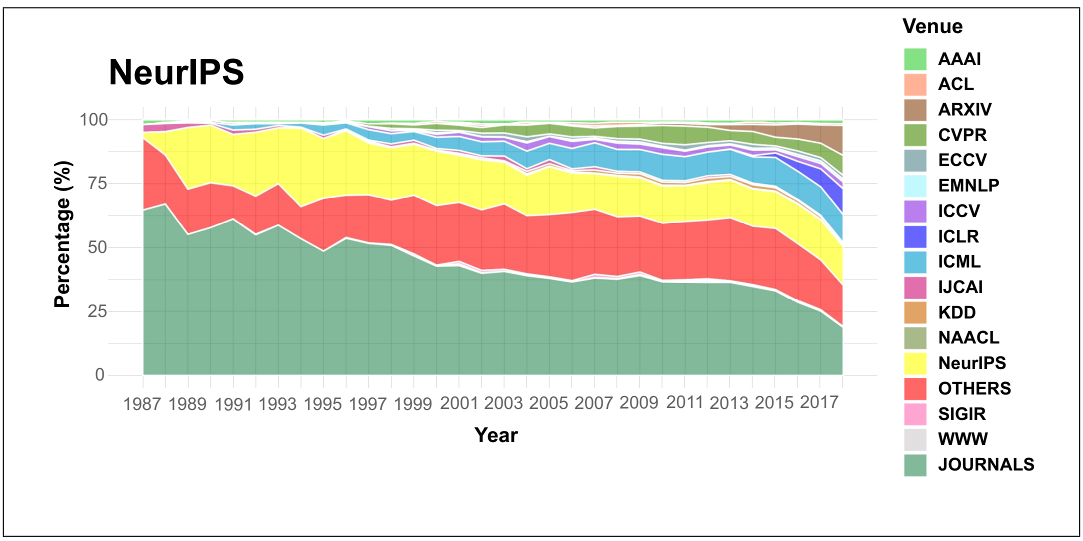


Figure 32: Where the citations coming from NeurIPS papers are pointing to share of each venue.

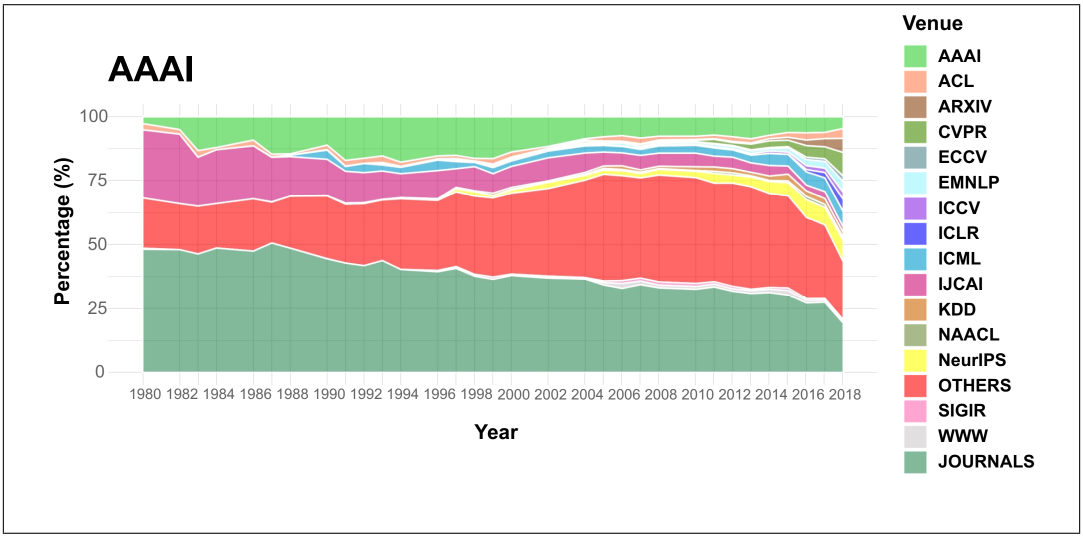


Figure 33: Where the citations coming from AAAI papers are pointing to share of each venue.

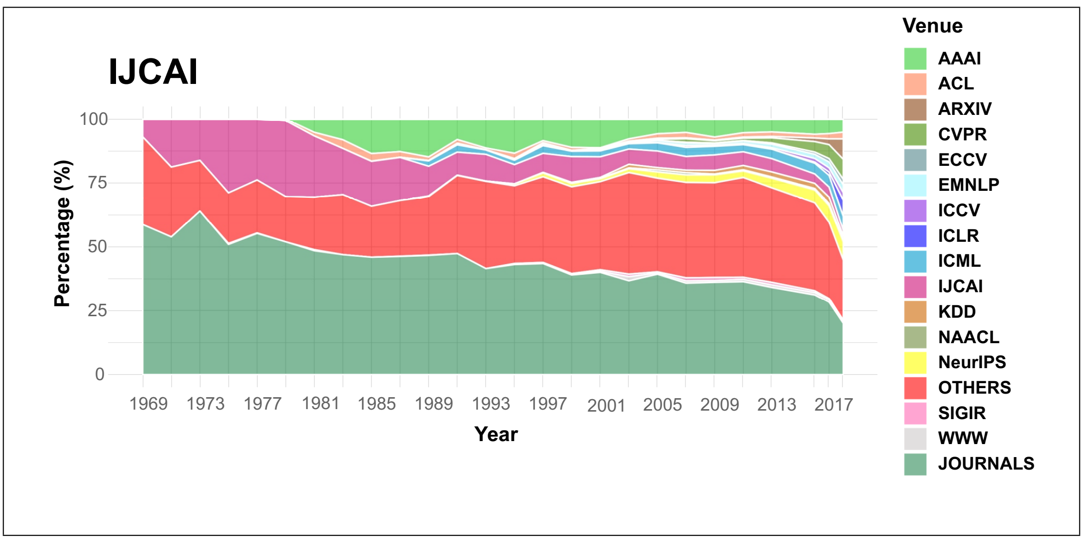


Figure 34: Where the citations coming from IJCAI papers are pointing to share of each venue.

4.5 Author-Paper Citation Graph

This graph was built to foster the research on recommender systems for papers – "given an author and his publication history, which are the most relevant papers he has not yet cited?"". In this sense, we have not developed any analysis on this graph and we expect to use it as a benchmark for future research on recommender systems: see Section 5.2.

4.6 Countries Citation Graph

This graph, computed for every country, is interesting because it has a different cardinality compared to the other graphs: while the others have tens of thousands of nodes, this graph only has 93 nodes with 4776703 edges.

Figure 35 shows the increase in the number of papers published at these conferences per year. The best-fitting line interpolates the data every 5 years. While the mid-1970s saw just a few countries participating in conferences (1974 and 1976 only had 2 countries: USA and United Kingdom, and USA and Canada, respectively) we have seen a large increase in countries participating in conferences in the last years, with 66 different countries having published in the conferences of interest in 2017. As mentioned above, in total authors from 93 different countries already published at these conferences.

Figure 36 shows a stacked percentage chart of the 15 countries with the most published papers. This data clearly shows the dominance of the United States in Artificial Intelligence research, with a slow increase in the number of papers published by authors in China.

Similarly, Figure 37 shows the same data but the pink bar at the bottom represents papers that we could not detect their country of affiliation.

Through this data representation, one can clearly see the years when IJCAI happened. Given the fact that IJCAI is commonly held outside the United States, and only in odd-numbered years, we can see a jagged-line pattern in the United States' share of papers, with a higher percentage in even-numbered years, and a lower percentage in odd-numbered years (when people from different countries have a higher chance of attending the conference, usually because of less strict visa requirements). For the same reason, after IJCAI started to be held annually (2013) the pattern disappeared. Figure 51 tries fixing this problem by creating a 2-year-wide sliding window and averaging the data before plotting it, creating a clearer view of the data.

An interesting outlier can be seen in 1979 when Japan had the highest share of published papers except for the USA. That happened exactly because IJCAI was held in Tokyo that year.

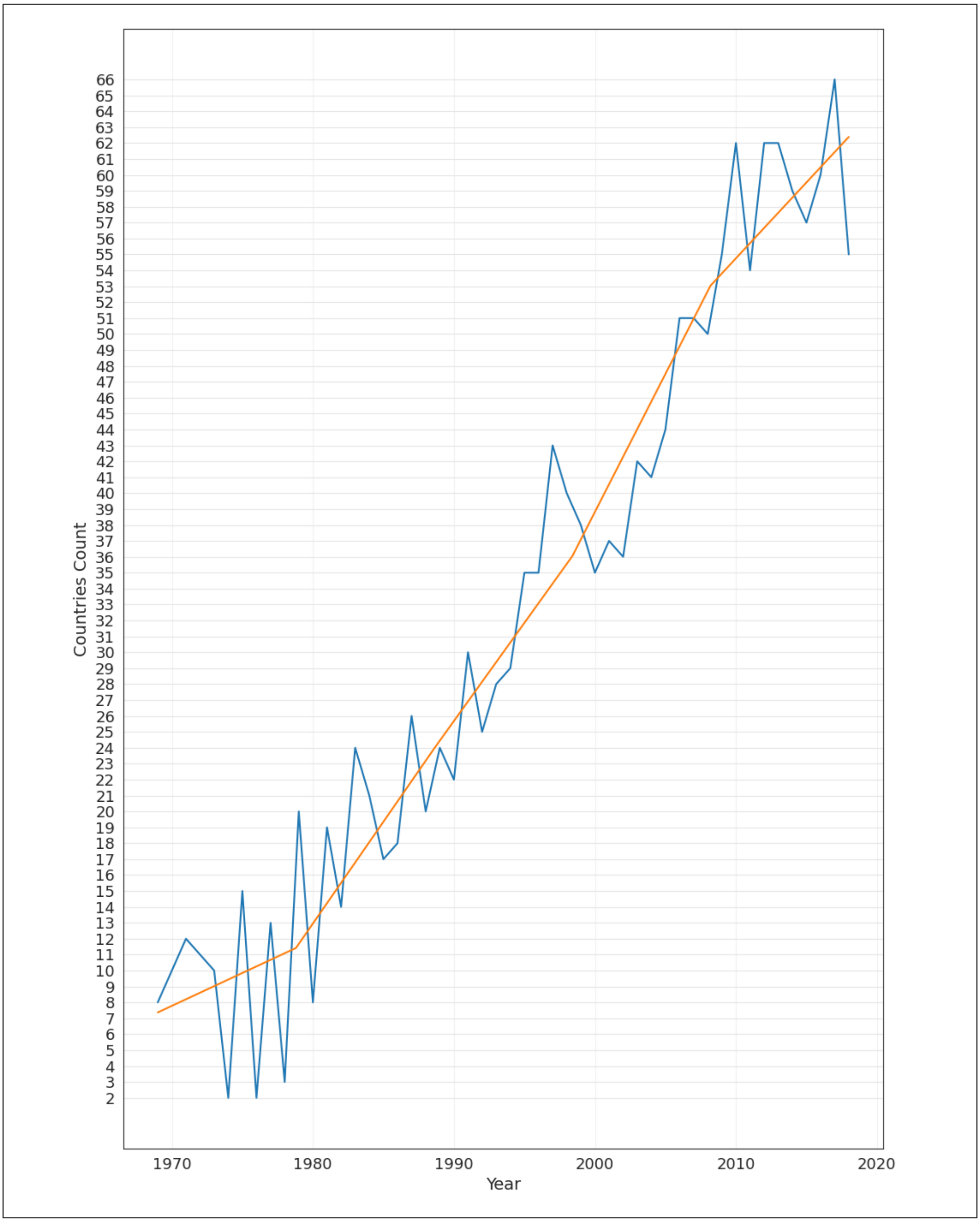


Figure 35: Number of Countries that published papers per year. The interpolating line is the best-fitting linear interpolation with 5 points

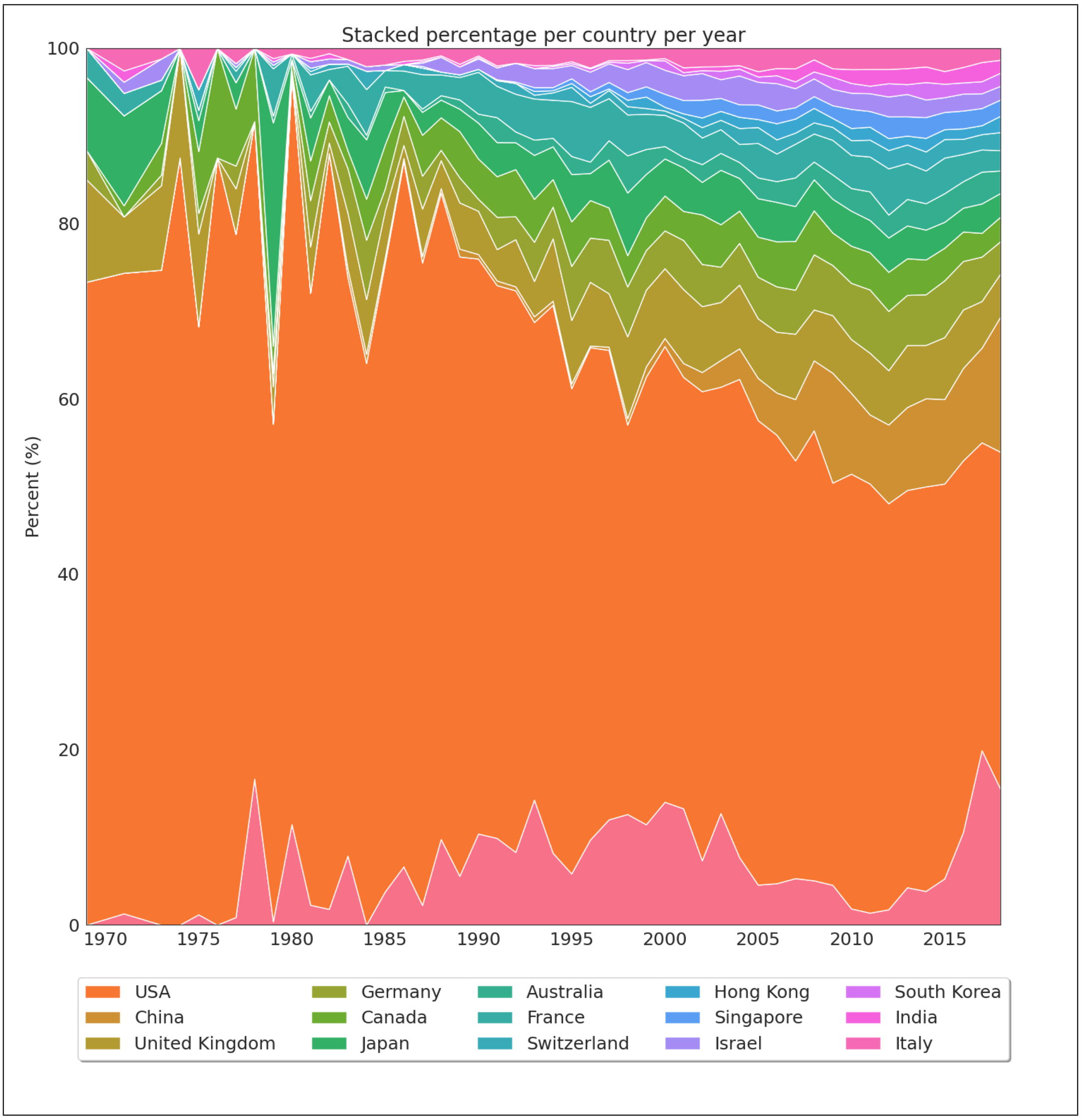


Figure 36: Stacked percentage of papers published per country per year including non-mapped ones. *The pink bar at the bottom indicates papers we could not identify which country they are from.*

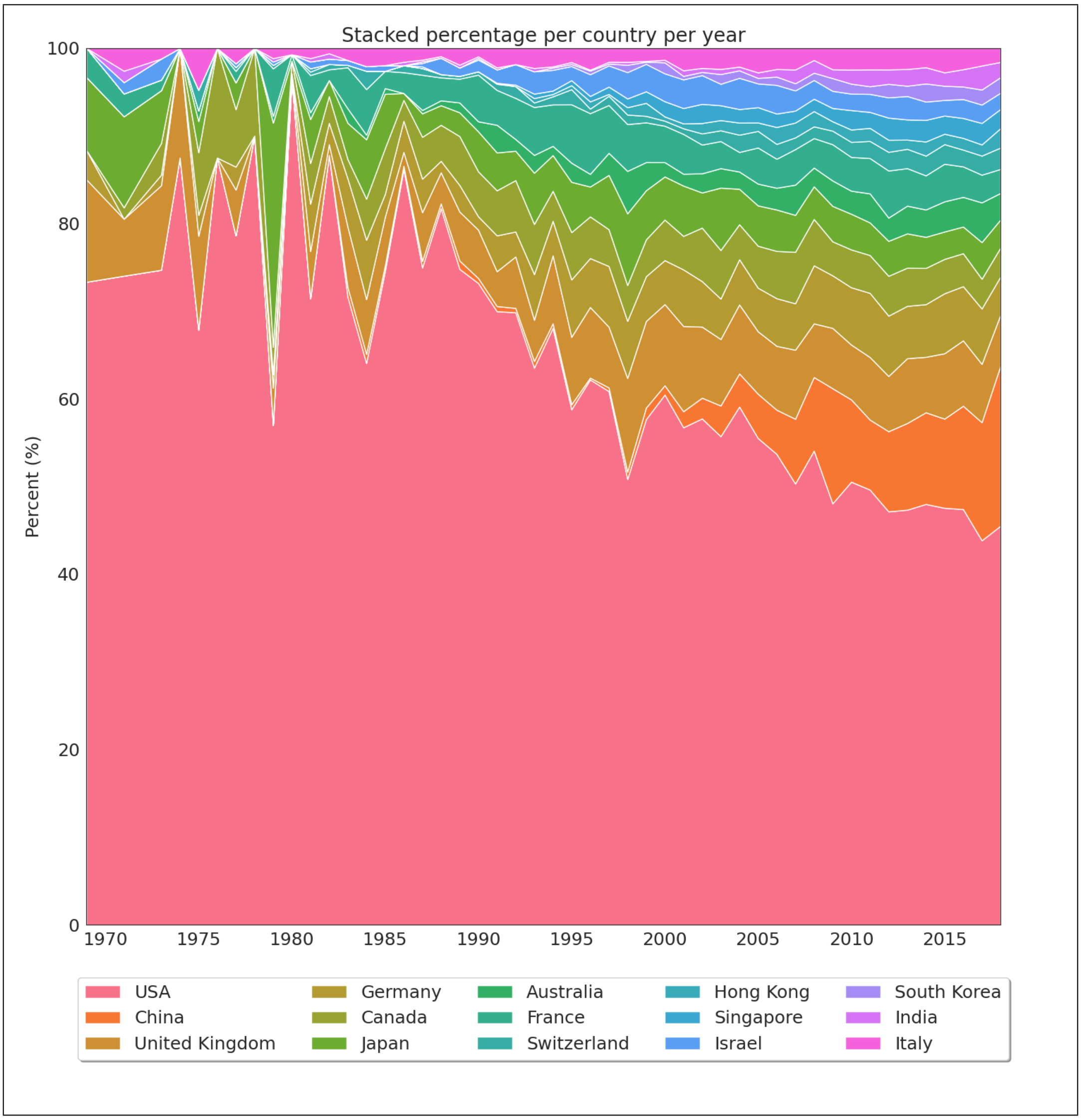


Figure 37: Stacked percentage of papers published per country per year, but not considering the ones we cannot identify.

Figure 38 shows the same data but with numbers in absolute terms instead of showing it with a stacked percentual. With it, we can see how the rate of acceptance for countries that are not the USA has grown faster than it has for the USA (the curve is steeper at the top). With it, we can also see the striking increase in papers accepted to these conferences in the last years, as already shown in numbers before.

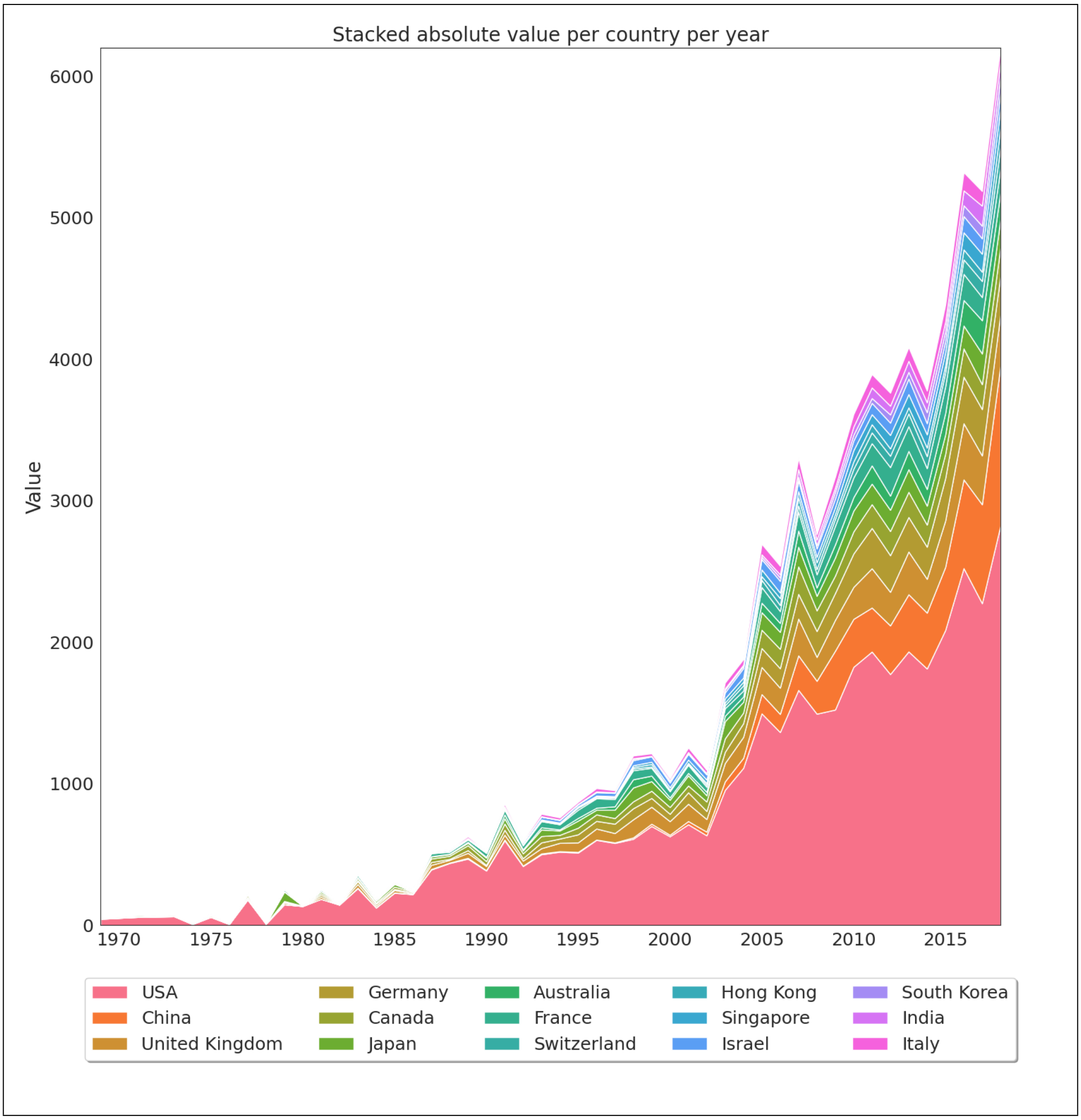


Figure 38: Quantity of papers per country per year

4.6.1 Analysis of More Recent Years

If we try using Arnet's v13 dataset to generate the above graphs, we can see in Figure 39 how much the data deteriorates. For 2019, we can only identify close to 5% of the paper country of affiliation, because for most of them, the "organization" field is empty. Note as well that even for the previous year the data is not as clear as it is in Figure 37, for instance.

We did not try applying any mapping similar to the one explained in Section 3.4 to this data because most of the non-identifiable organization fields are empty, as stated above.

4.7 Data Analysis of Turing Laureates

As previously stated, the Turing Award recognized seven researchers for their contributions to AI: Marvin Minsky (1969), John McCarthy (1971), Allen Newell and Herbert Simon (1975), Edward Feigenbaum and Raj Reddy (1994), Leslie Valiant (2010), Judea Pearl (2011) and Yoshua Bengio, Geoffrey Hinton and Yann LeCun (2018).

The Turing Award winners timeline (see Figure 40) depicts a change of focus of these highly prolific researchers over time: most recent awardees have their work divided into several venues (especially machine learning and computer vision-related ones, such as NIPS/NeurIPS and ICML and CVPR), while the older ones concentrated their efforts in AAAI or IJCAI. We also need to take into account that we are only considering conferences in this work, while most of the works published in the early days of Artificial Intelligence were published in other venues.

We also verified the Spearman correlation between the titles of papers published by Turing Award winners and the titles of papers published in the selected AI conferences (AAAI, IJCAI, and NIPS/NeurIPS) over time. To do so we compare the ranking of the TF-IDFs for the words in the Turing Award winner's paper titles in that year, related to the ranking of the TF-IDFs conference's (or group of conferences) papers titles in the same year. As the AI community, in general, has leaned towards the connectionist approach over the last years, we expected to see a decreasing trend regarding previous Turing Award winners who focused on symbolic AI and expert systems – or at least a very little correlation.

Nevertheless, the work of Marvin Minsky (1969 Turing Award laureate) is still quite in line with what is published in NIPS/NeurIPS, for instance, despite being poorly correlated with the three conferences when they are considered altogether (see Figure 41). These correlations may be however not very realistic since there are only two papers by Marvin Minsky in the entire dataset. The most positive

Figure 39: Deteriorated countries stacked chart with Arnet's V13

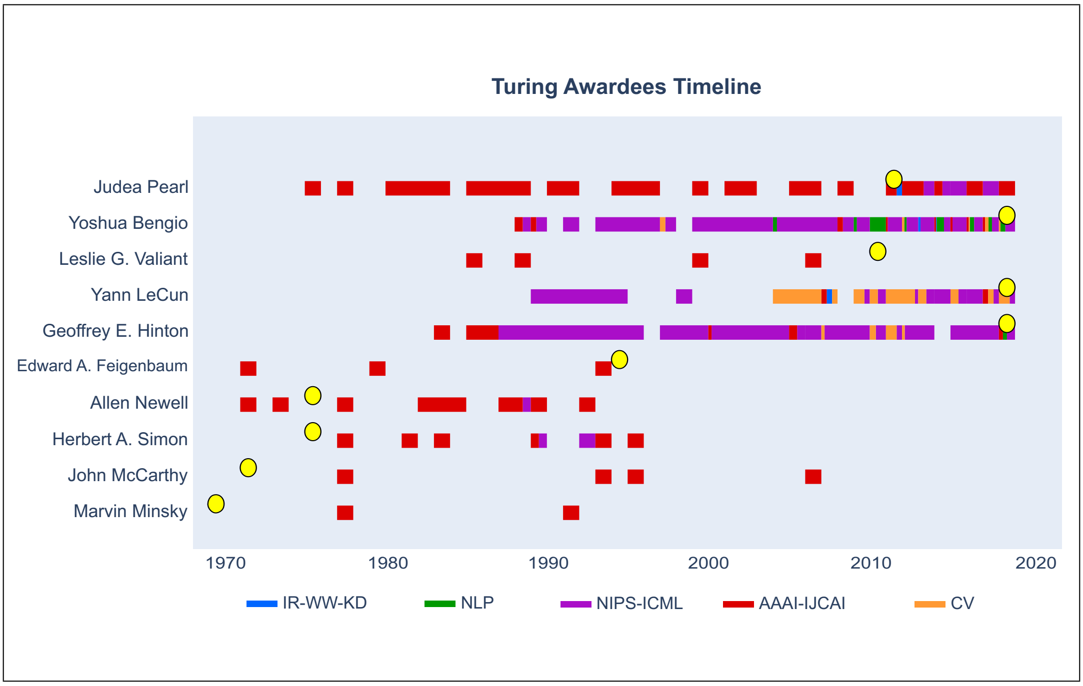


Figure 40: AI-related Turing Awardees timeline. *Each year is divided proportionally according to the number of papers published in each group of venues. Yellow ellipses are placed on the year each award was granted.*

and significant slope, however, comes from the work of the latest AI Turing Award laureates (Bengio, Hinton, and LeCun, 2018): their papers' titles have a positive and moderate correlation with all three conferences (and naturally with the average) and also show an increasing trend along the years, as depicted on Figure 42. The remaining plots can be found in Appendix G

Figure 42 clearly shows how the most recent AI Turing Awardees (Yoshua Bengio, Geoffrey Hinton, and Yann LeCun) influenced the area, with increasing rates of correlation over the years in all three main conferences from the Artificial Intelligence field. We predict that, in the next few years, if we were to plot the same data again, their correlation would likely have increased even further showing that they were able to influence AI research in general. We base our hypothesis on the fact that the 1969 Turing Award laureate Marvin Minsky, still has a positive correlation rate in some conferences such as NeurIPS, even though the same cannot be said for the AI field in general. Also, as noted in Section 2, Minsky possibly influenced the future

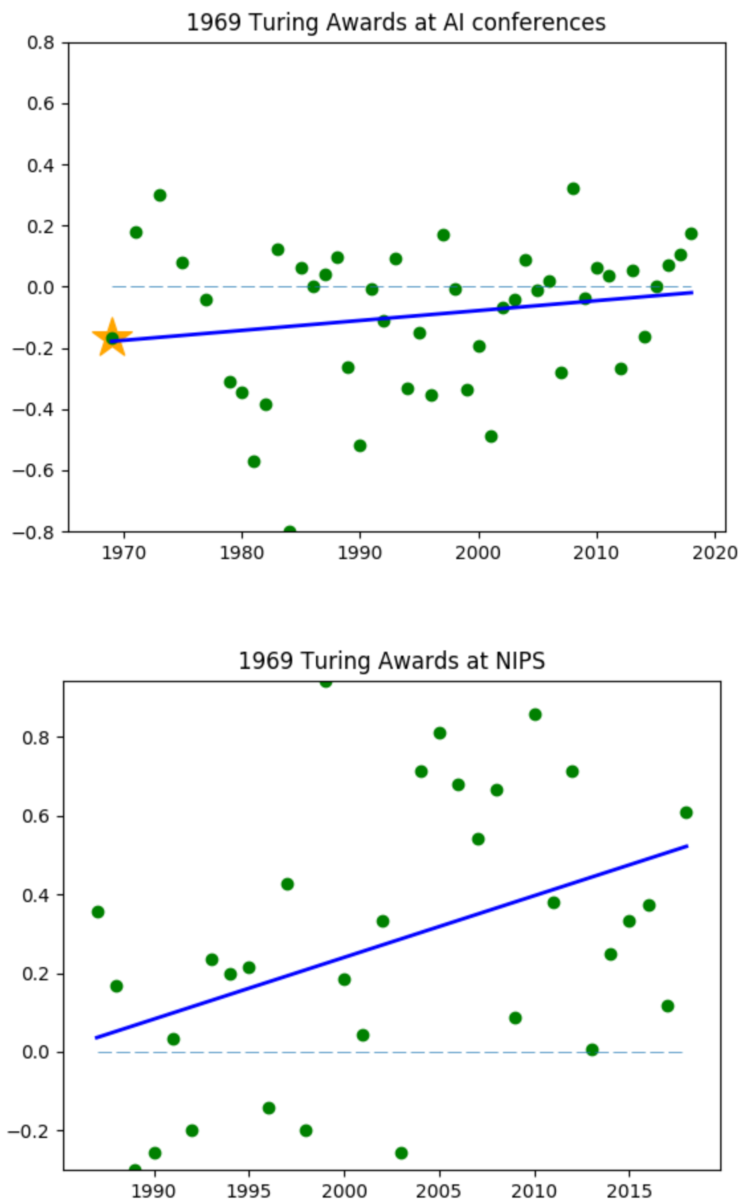


Figure 41: 1969 Turing Award Correlation with AI conferences and NIPS specifically

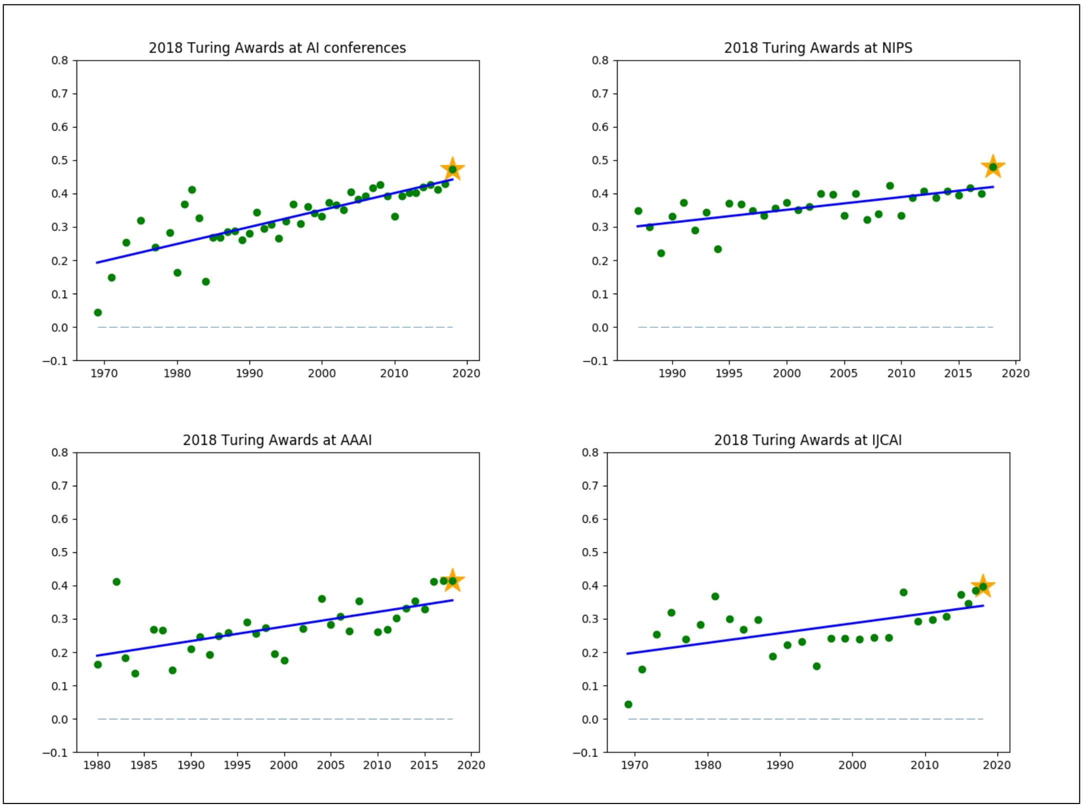


Figure 42: Correlation between titles of papers published by 2018 Turing Award winners and titles of papers published in the three AI flagship conferences.

of AI after publishing Minsky and Papert [1969].

Similarly, this data can also be seen from the opposite side, if we consider the 2018 laureates: they were closely following the trend of papers published in these conferences, therefore winning the Turing Award by researching the areas of interest. Even if possible, it is clear that their works are relevant and influenced the area in ways not influenced by others.

Also, it is worth noting that the Turing Award laureates do not appear in the ranking of authors according to the centrality measures in Section 3.3.1 and 4.3. This is probably because the Turing Awardees *have not* published a large number of papers in the venues analysed to be able to reach the top of the rankings, and also because their contributions are mostly based on some seminal, highly influential works.

5 Conclusions and Further Work

Artificial Intelligence research has accomplished much since Turing [1936], McCulloch and Pitts [1943], Turing [1950]. AI is now amply used in industry to power large high-tech corporations. Some of the processes behind this evolution are still not well understood this work aimed to make AI development, history, and evolution clear. We presented a short survey on AI history, describing the periods or seasons AI has already gone through. We have also included a quick survey on graph centrality measures, the Turing Award winners, and the flagship AI-related conferences, which is necessary to understand the overall picture of the area better.

By analyzing Arnet's v11 dataset, a dataset based on DBLP's corpus, and enhancing it to a graph-based format, we intended to ease paper/author citation/collaboration network research. This dataset generated insightful graphs and could generate even more in future works. Also, the code presented in this work makes it such that it is relatively easy to extend it to any other underlying dataset, making it possible to generate and compute the same statistics presented in this work for any other area than AI. These graphs show insights on self-citations, new authors, and author and paper importance throughout the years. We also proposed a new type of dataset intended to be used as a knowledge graph source for recommender systems, where authors, papers, citations, and collaborations are all defined in the same graph. With the Country Citation Graph, we also introduced an important dataset and pipeline capable of inferring the country of affiliation of an author based on its organization.

By investigating the Turing Award winners and comparing them against the published data, we find out that there is evidence that they actually "pull" their most published venues to their topic of research, at least for the most recent AI researchers winners. Finally, the study on countries' affiliation is, to the best of our knowledge, the first of its type, creating a new algorithm able to infer the country of affiliation of an author from their organization, as available at DBLP or Arnet.

5.1 Contributions

This work has the following specific contributions:

- Five new graph-based datasets, with fully computed centralities to ease paper/author, citation/collaboration network research.
- Analyses for these graphs, focusing both on their raw structure and the centrality rankings.

- Algorithms description, allowing anyone to replicate the graph building process in any programming language for any dataset.
- Spearman Correlation computation between Turing Award laureate papers and conference papers, showing they have a positive correlation.

Besides the theoretical contributions we also present a few important software contributions. They can be found in more detail in Appendix H, but we outline them as follows:

- Python library to convert an XML to JSON in a stream fashion, i.e. without loading the whole XML and JSON files in memory
- Parallel Python implementation for the Betweenness and Closeness Centralities
- Novel Python implementation for a Graph Parsing pipeline, avoiding duplicate work through data caching
- Python implementation of the proposed algorithm to infer a paper country of affiliation

5.2 Future Work

The dataset created in this work provides several possibilities for future work, especially when we think about the computed centralities. Our work presented several analyses of the dataset, and one might think of even more possible ways to visualize it. Additionally, it would be ideal if Coreness centrality (Section 2.4.6) was also computed for this dataset, as it displays the interesting feature of being a discrete value instead of a continuous one, thus allowing you to more easily identify the most important authors/papers according to it.

The Country Citation Graph has a lot of potential in understanding "brain-draining" by investigating the flow of authors from one country of affiliation to others – easily done with our dataset, without any extra work besides counting the number of "transitions" between countries. Similarly, we believe that comparing more advanced usages of this dataset with the Turing Awardees might bring even more interesting results. With a better dataset, one where there are abstract data available for every paper, one might be able to achieve better results when running a Spearman Correlation (Section 4.7) between the text in Turing Award winners' abstracts and the ones from the remaining venues papers.

Acknowledgements

This work is partly supported by CNPq and CAPES, Brazil, Finance Code 001.

References

Nicholas Crafts. Artificial Intelligence as a General-Purpose Technology: An Historical Perspective. *Oxford Review of Economic Policy*, 37(3):521–536, 09 2021. ISSN 0266-903X. doi:10.1093/oxrep/grab012. URL `https://doi.org/10.1093/oxrep/grab012`.

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943. ISSN 1522-9602. doi:10.1007/BF02478259. URL `https://doi.org/10.1007/BF02478259`.

A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950. ISSN 00264423. URL `http://www.jstor.org/stable/2251299`.

Edward A. Feigenbaum and Julian Feldman. *Computers and Thought*. McGraw-Hill, Inc., USA, 1963. ISBN 0070203709.

Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1): 8–30, 1961.

Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst.*, 6(4), dec 2016. ISSN 2158-656X. doi:10.1145/2843948. URL `https://doi.org/10.1145/2843948`.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. URL `https://arxiv.org/abs/2102.12092`.

Lisa Amini, Ching-Hua Chen, David D. Cox, Aude Oliva, and Antonio Torralba. Experiences and insights for collaborative industry-academic research in artificial intelligence. *AI Mag.*, 41(1):70–81, 2020. doi:10.1609/aimag.v41i1.5201. URL `https://doi.org/10.1609/aimag.v41i1.5201`.

Gartner. Gartner predics the future of ai technologies. Technical report, Gartner, 2019. URL `https://www.gartner.com/smarterwithgartner/gartner-predicts-the-future-of-ai-technologies`.

Murray Campbell, A. Joseph Hoane, and Feng-Hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1):57–83, 2002. ISSN 0004-3702. doi:https://doi.org/10.1016/S0004-3702(01)00129-1. URL https://www.sciencedirect.com/science/article/pii/S0004370201001291.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan 2016. ISSN 1476-4687. doi:10.1038/nature16961. URL https://doi.org/10.1038/nature16961.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, Aug 2021. ISSN 1476-4687. doi:10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Gaugan: Semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH*

2019 Real-Time Live!, SIGGRAPH '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450363150. doi:10.1145/3306305.3332370. URL https://doi.org/10.1145/3306305.3332370.

Gary Marcus. Deep learning: A critical appraisal, 2018. URL https://arxiv.org/abs/1801.00631.

Gary Marcus. The next decade in AI: four steps towards robust artificial intelligence. *CoRR*, abs/2002.06177, 2020. URL https://arxiv.org/abs/2002.06177.

Artur S. d'Avila Garcez and Luís C. Lamb. Reasoning about time and knowledge in neural symbolic learning systems. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003]*, pages 921–928. MIT Press, 2003. URL https://proceedings.neurips.cc/paper/2003/hash/347665597cbfaef834886adbb848011f-Abstract.html.

Artur S. d'Avila Garcez and Luís C. Lamb. A connectionist computational model for epistemic and temporal reasoning. *Neural Comput.*, 18(7):1711–1738, 2006. doi:10.1162/neco.2006.18.7.1711. URL https://doi.org/10.1162/neco.2006.18.7.1711.

Artur S. d'Avila Garcez, Luís C. Lamb, and Dov M. Gabbay. *Neural-Symbolic Cognitive Reasoning*. Cognitive Technologies. Springer, 2009. ISBN 978-3-540-73245-7. doi:10.1007/978-3-540-73246-4. URL https://doi.org/10.1007/978-3-540-73246-4.

Artur d'Avila Garcez and Luís C. Lamb. Neurosymbolic AI: The 3rd Wave. *Artificial Intelligence Review*, pages 1–20, 2023. doi:https://doi.org/10.1007/s10462-023-10448-w.

Ryan Riegel, Alexander G. Gray, Francois P. S. Luus, Naweed Khan, Ndivhuwo Makondo, Ismail Yunus Akhalwaya, Haifeng Qian, Ronald Fagin, Francisco Barahona, Udit Sharma, Shajith Ikbal, Hima Karanam, Sumit Neelam, Ankita Likhyani, and Santosh K. Srivastava. Logical neural networks. *CoRR*, abs/2006.13155, 2020. URL https://arxiv.org/abs/2006.13155.

Tarek R. Besold, Artur S. d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro M. Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luís C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. Neural-Symbolic Learning and Reasoning: A Survey and interpretation. In Pascal Hitzler and Md Kamruzzaman Sarker, editors, *Neuro-Symbolic Artificial Intelligence: The State-of-the-Art*, pages 1–51. IOS Press, 2022.

Grady Booch, Francesco Fabiano, Lior Horesh, Kiran Kate, Jon Lenchner, Nick Linck, Andrea Loreggia, Keerthiram Murugesan, Nicholas Mattei, Francesca Rossi, and Biplav Srivastava. Thinking fast and slow in AI. *CoRR*, abs/2010.06002, 2020. URL `https://arxiv.org/abs/2010.06002`.

Gary Marcus and Ernest Davis. Insights for AI from the human mind. *Commun. ACM*, 64(1):38–41, 2021. doi:10.1145/3392663. URL `https://doi.org/10.1145/3392663`.

Paul Smolensky, R. Thomas McCoy, Roland Fernandez, Matthew Goldrick, and Jianfeng Gao. Neurocompositional computing: From the central paradox of cognition to a new generation of AI systems. *CoRR*, abs/2205.01128, 2022. doi:10.48550/arXiv.2205.01128. URL `https://doi.org/10.48550/arXiv.2205.01128`.

Henry A. Kautz. The Third AI summer: AAAI Robert s. Engelmore Memorial Lecture. *AI Mag.*, 43(1):93–104, 2022. doi:10.1609/aimag.v43i1.19122. URL `https://doi.org/10.1609/aimag.v43i1.19122`.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, pages 990–998. ACM, 2008.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020. ISBN 9780134610993. URL `http://aima.cs.berkeley.edu/`.

Joseph Weizenbaum. Elizaa computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):3645, jan 1966. ISSN 0001-0782. doi:10.1145/365153.365168. URL `https://doi.org/10.1145/365153.365168`.

Thomas G. Evans. A heuristic program to solve geometric-analogy problems. In *Proceedings of the April 21-23, 1964, Spring Joint Computer Conference*, AFIPS '64 (Spring), page 327338, New York, NY, USA, 1964. Association for Computing Machinery. ISBN 9781450378901. doi:10.1145/1464122.1464156. URL `https://doi.org/10.1145/1464122.1464156`.

J. A. Feldman, G. M. Feldman, G. Falk, G. Grape, J. Pearlman, I. Sobel, and J. M. Tenebaum. The Stanford Hand-eye Project. In *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, IJCAI'69, page 521526, San Francisco, CA, USA, 1969. Morgan Kaufmann Publishers Inc.

Nils J. Nilsson. A mobius automation: An application of artificial intelligence techniques. In *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, IJCAI'69, page 509520, San Francisco, CA, USA, 1969. Morgan Kaufmann Publishers Inc.

Cordell Green. Application of theorem proving to problem solving. In *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, IJCAI'69, page 219239, San Francisco, CA, USA, 1969. Morgan Kaufmann Publishers Inc.

Robert A. Kowalski. Logic programming. In Jörg H. Siekmann, editor, *Computational Logic*, volume 9 of *Handbook of the History of Logic*, pages 523–569. Elsevier, 2014. doi:10.1016/B978-0-444-51624-4.50012-5. URL `https://doi.org/10.1016/B978-0-444-51624-4.50012-5`.

Philipp Körner, Michael Leuschel, João Barbosa, Vítor Santos Costa, Verónica Dahl, Manuel V. Hermenegildo, José F. Morales, Jan Wielemaker, Daniel Diaz, Salvador Abreu, and Giovanni Ciatto. 50 years of prolog and beyond. *CoRR*, abs/2201.10816, 2022. URL `https://arxiv.org/abs/2201.10816`.

Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2016. doi:10.2200/S00692ED1V01Y201601AIM032. URL `https://doi.org/10.2200/S00692ED1V01Y201601AIM032`.

J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, Berlin, Heidelberg, 1984. ISBN 0387132996.

Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA, 1969.

Jordan B Pollack. No harm intended: Marvin L. Minsky and Seymour A. Papert. Perceptrons: An introduction to computational geometry, Expanded Edition. Cambridge, MA: MIT Press, 1988. pp. 292. *Journal of Mathematical Psychology*, 33(3):358–365, 1989.

Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151158, New York, NY, USA, 1971. Association for Computing Machinery. ISBN 9781450374644. doi:10.1145/800157.805047. URL `https://doi.org/10.1145/800157.805047`.

Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9. URL https://doi.org/10.1007/978-1-4684-2001-2_9.

James Lighthill. Artificial intelligence: A general survey. In *Artificial Intelligence: a paper symposium, Science Research Council*, London,UK, 1973.

Ehud Y. Shapiro. The fifth generation project a trip report. *Commun. ACM*, 26 (9):637641, sep 1983. ISSN 0001-0782. doi:10.1145/358172.358179. URL https://doi.org/10.1145/358172.358179.

Dov M Gabbay, C J Hogger, and J A Robinson. *Handbook of Logic in Artificial Intelligence and Logic Programming: Volume 5: Logic Programming*. Oxford University Press, 01 1998. ISBN 9780198537922. doi:10.1093/oso/9780198537922.001.0001. URL https://doi.org/10.1093/oso/9780198537922.001.0001.

Dov M. Gabbay, Jörg H. Siekmann, and John Woods. *Handbook of the History of Logic - Computational Logic, Volume 9*. North Holland, 2014. ISBN 0444516247.

J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. doi:10.1073/pnas.79.8.2554. URL https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct 1986. ISSN 1476-4687. doi:10.1038/323533a0. URL https://doi.org/10.1038/323533a0.

Hans Moravec. *Mind Children: The Future of Robot and Human Intelligence*. Harvard University Press, USA, 1988. ISBN 0674576160.

Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Comput.*, 6(2):215219, mar 1994. ISSN 0899-7667. doi:10.1162/neco.1994.6.2.215. URL https://doi.org/10.1162/neco.1994.6.2.215.

Ernst D. Dickmanns. An integrated approach to feature based dynamic vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1988, 5-9 June, 1988, Ann Arbor, Michigan, USA*, pages 820–825. IEEE,

1988. doi:10.1109/CVPR.1988.196328. URL https://doi.org/10.1109/CVPR.1988.196328.

Frank Thomanek and Ernst D. Dickmanns. Autonomous road vehicle guidance in normal traffic. In Stan Z. Li, Dinesh P. Mital, Eam Khwang Teoh, and Han Wang, editors, *Recent Developments in Computer Vision, Second Asian Conference on Computer Vision, ACCV '95, Singapore, December 5-8, 1995, Invited Session Papers*, volume 1035 of *Lecture Notes in Computer Science*, pages 499–507. Springer, 1995. doi:10.1007/3-540-60793-5_103. URL https://doi.org/10.1007/3-540-60793-5_103.

Dayne Freitag. Machine learning for information extraction in informal domains. *Machine Learning*, 39(2):169–202, May 2000. ISSN 1573-0565. doi:10.1023/A:1007601113994. URL https://doi.org/10.1023/A:1007601113994.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Tim Hwang. Deepfakes: A grounded threat assessment, jul 2020. URL cset.georgetown.edu/research/deepfakes-a-grounded-threat-assessment/.https://doi.org/10.51593/20190030.

Thanh Thi Nguyen, Cuong M. Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection. *CoRR*, abs/1909.11573, 2019. URL http://arxiv.org/abs/1909.11573.

L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):11341142, nov 1984. ISSN 0001-0782. doi:10.1145/1968.1972. URL https://doi.org/10.1145/1968.1972.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 0934613737.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, 2006. doi:10.1162/neco.2006.18.7.1527. URL https://doi.org/10.1162/neco.2006.18.7.1527.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi:10.1109/5.726791.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi:10.1162/neco.1989.1.4.541.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL https://arxiv.org/abs/1409.0473.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Francesca Rossi and Nicholas Mattei. Building ethically bounded AI. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 9785–9789. AAAI Press, 2019. doi:10.1609/aaai.v33i01.33019785. URL https://doi.org/10.1609/aaai.v33i01.33019785.

Sepp Hochreiter. Toward a broad ai. *Commun. ACM*, 65(4):5657, mar 2022. ISSN 0001-0782. doi:10.1145/3512715. URL https://doi.org/10.1145/3512715.

Luís C. Lamb, Artur S. d'Avila Garcez, Marco Gori, Marcelo O. R. Prates, Pedro H. C. Avelar, and Moshe Y. Vardi. Graph neural networks meet neural-symbolic computing: A survey and perspective. *CoRR*, abs/2003.00330, 2020. URL https://arxiv.org/abs/2003.00330.

Saurabh Mishra, Jack Clark, and C. Raymond Perrault. Measurement in AI policy: Opportunities and challenges. *CoRR*, abs/2009.09071, 2020. URL https://arxiv.org/abs/2009.09071.

Daniel Zhang, Saurabh Mishra, Erik Brynjolfsson, John Etchemendy, Deep Ganguli, Barbara J. Grosz, Terah Lyons, James Manyika, Juan Carlos Niebles, Michael Sellitto, Yoav Shoham, Jack Clark, and C. Raymond Perrault. The AI index 2021 annual report. *CoRR*, abs/2103.06312, 2021. URL `https://arxiv.org/abs/2103.06312`.

Alan Turing. On computable numbers, with an application to the n tscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(1):230–265, 1936. doi:10.2307/2268810.

CACM. ACM's Turing Award Prize Raised to $1 million. *Commun. ACM*, 57(12): 20, nov 2014. ISSN 0001-0782. doi:10.1145/2685372. URL `https://doi.org/10.1145/2685372`.

Chen Avin, Ilya Shpitser, and Judea Pearl. Identifiability of path-specific effects. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, page 357363, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

Edward A. Feigenbaum. The art of artificial intelligence: Themes and case studies of knowledge engineering. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977*, pages 1014–1029, 1977.

John McCarthy. Epistemological problems of artificial intelligence. In Raj Reddy, editor, *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977*, pages 1038–1044. William Kaufmann, 1977. URL `http://ijcai.org/Proceedings/77-2/Papers/094.pdf`.

Leslie G. Valiant. Learning disjunction of conjunctions. In Aravind K. Joshi, editor, *Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, USA, August 1985*, pages 560–566. Morgan Kaufmann, 1985. URL `http://ijcai.org/Proceedings/85-1/Papers/107.pdf`.

Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning, 2019. URL `https://arxiv.org/abs/1903.03825`.

Geoffrey E. Hinton. Modeling high-dimensional data by combining simple experts. In Henry A. Kautz and Bruce W. Porter, editors, *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, USA*, pages 1159–1164. AAAI

Press / The MIT Press, 2000. URL http://www.aaai.org/Library/AAAI/2000/aaai00-211.php.

Renato de Mori, Yoshua Bengio, and Régis Cardin. Data-driven execution of multi-layered networks for automatic speech recognition. In Howard E. Shrobe, Tom M. Mitchell, and Reid G. Smith, editors, *Proceedings of the 7th National Conference on Artificial Intelligence, St. Paul, MN, USA, August 21-26, 1988*, pages 734–738. AAAI Press / The MIT Press, 1988. URL http://www.aaai.org/Library/AAAI/1988/aaai88-130.php.

Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In David L. Waltz, editor, *Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, PA, USA, August 18-20, 1982*, pages 133–136. AAAI Press, 1982. URL http://www.aaai.org/Library/AAAI/1982/aaai82-032.php.

Leslie G. Valiant. Knowledge infusion. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 1546–1551. AAAI Press, 2006. URL http://www.aaai.org/Library/AAAI/2006/aaai06-247.php.

Sepp Hochreiter and Jürgen Schmidhuber. LSTM can solve hard long time lag problems. In Michael Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems 9, NIPS*, pages 473–479. MIT Press, 1996. URL http://papers.nips.cc/paper/1215-lstm-can-solve-hard-long-time-lag-problems.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30 - NeurIPS*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith

Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020. URL https://arxiv.org/abs/2011.10566.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018. URL https://arxiv.org/abs/1812.04948.

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation, 2018. URL https://arxiv.org/abs/1809.02983.

Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010. doi:10.1109/CVPR.2010.5539963.

Y. LeCun, Fu Jie Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–104 Vol.2, 2004. doi:10.1109/CVPR.2004.1315150.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020. URL https://arxiv.org/abs/2003.12039.

Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image, 2019. URL https://arxiv.org/abs/1905.01164.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp, 2019. URL https://arxiv.org/abs/1906.02243.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. URL https://arxiv.org/abs/1810.04805.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding, 2019. URL https://arxiv.org/abs/1909.10351.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. Codebert: A pre-trained model for programming and natural languages, 2020. URL https://arxiv.org/abs/2002.08155.

Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics. doi:10.18653/v1/D19-1371. URL https://aclanthology.org/D19-1371.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks, 2012. URL https://arxiv.org/abs/1211.5063.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL https://arxiv.org/abs/1502.03167.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2018. URL https://arxiv.org/abs/1805.08318.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, aug 2020. doi:10.1145/3394486.3403168. URL https://doi.org/10.1145%2F3394486.3403168.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks, 2020. URL https://arxiv.org/abs/2005.11650.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks, 2020. URL https://arxiv.org/abs/2005.10203.

Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, aug 2020. doi:10.1145/3394486.3403076. URL https://doi.org/10.1145%2F3394486.3403076.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation, 2020. URL https://arxiv.org/abs/2002.02126.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, jul 2021. doi:10.1145/3404835.3462862. URL https://doi.org/10.1145%2F3404835.3462862.

Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. Global context enhanced graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, jul 2020. doi:10.1145/3397271.3401142. URL https://doi.org/10.1145%2F3397271.3401142.

Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978. ISSN 0378-8733. doi:https://doi.org/10.1016/0378-8733(78)90021-7. URL https://www.sciencedirect.com/science/article/pii/0378873378900217.

Felipe Grando, Lisandro Granville, and Luís C. Lamb. Machine learning in network centrality measures: Tutorial and outlook. *ACM Comput. Surv.*, 51(5):102:1–102:32, 2019. doi:10.1145/3237192. URL https://doi.org/10.1145/3237192.

Zelin Wan, Yash Mahajan, Beom Woo Kang, Terrence J. Moore, and Jin-Hee Cho. A survey on centrality metrics and their network resilience analysis. *IEEE Access*, 9:104773–104819, 2021. doi:10.1109/ACCESS.2021.3094196.

Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977. ISSN 00380431. URL http://www.jstor.org/stable/3033543.

Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, Dec 1966. ISSN 1860-0980. doi:10.1007/BF02289527. URL https://doi.org/10.1007/BF02289527.

Akrati Saxena and Sudarshan Iyengar. Centrality measures in complex networks: A survey, 2020. URL https://arxiv.org/pdf/2011.07190.pdf.

John Cardente. Using centrality measures to identify key members of an innovation collaboration network, 2012.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

Pedro H. C. Avelar, Henrique Lemos, Marcelo O. R. Prates, and Luís C. Lamb. Multitask learning on graph neural networks: Learning multiple graph centrality measures with a unified network. In Igor V. Tetko, Vera Kurková, Pavel Karpov, and Fabian J. Theis, editors, *ICANN 2019 - 28th International Conference on Artificial Neural Networks*, volume 11731 of *Lecture Notes in Computer Science*, pages 701–715. Springer, 2019. doi:10.1007/978-3-030-30493-5_63. URL https://doi.org/10.1007/978-3-030-30493-5_63.

Duanbing Chen, Linyuan Lü, Ming-Sheng Shang, Yi-Cheng Zhang, and Tao Zhou. Identifying influential nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, 391(4):1777–1787, 2012. ISSN 0378-4371. doi:https://doi.org/10.1016/j.physa.2011.09.017. URL https://www.sciencedirect.com/science/article/pii/S0378437111007333.

Klaus Wehmuth and Artur Ziviani. Daccer: Distributed assessment of the closeness centrality ranking in complex networks. *Computer Networks*, 57(13):2536–2548, 2013. ISSN 1389-1286. doi:https://doi.org/10.1016/j.comnet.2013.05.001. URL https://www.sciencedirect.com/science/article/pii/S1389128613001412.

J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572, 2005. doi:10.1073/pnas.0507655102. URL https://www.pnas.org/doi/abs/10.1073/pnas.0507655102.

Maksim Kitsak, Lazaros K. Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H. Eugene Stanley, and Hernán A. Makse. Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893, Nov 2010. ISSN 1745-2481. doi:10.1038/nphys1746. URL https://doi.org/10.1038/nphys1746.

Dorian Tanase, David Garcia, Antonios Garas, and Frank Schweitzer. Emotions and activity profiles of influential users in product reviews communities. *Frontiers in Physics*, 3, 11 2015. doi:10.3389/fphy.2015.00087.

Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable ai: A brief survey on history, research areas, approaches and challenges. In Jie Tang, Min-Yen Kan, Dongyan Zhao, Sujian Li, and Hongying Zan, editors, *Natural Language Processing and Chinese Computing*, pages 563–574, Cham, 2019. Springer International Publishing. ISBN 978-3-030-32236-6.

Sunday Oke. A literature review on artificial intelligence. *International Journal of Information and Management Sciences*, 19:535–570, 12 2008.

Ying Ding, Erjia Yan, Arthur R. Frazho, and James Caverlee. Pagerank for ranking authors in co-citation networks. *CoRR*, abs/1012.4872, 2010. URL `http://arxiv.org/abs/1012.4872`.

Raf Guns, Yuxian Liu, and Dilruba Mahbuba. Q-measures and betweenness centrality in a collaboration network: A case study of the field of informetrics. *Scientometrics*, 87:133–147, 04 2011. doi:10.1007/s11192-010-0332-3.

Alireza Abbasi, Liaquat Hossain, and Loet Leydesdorff. Betweenness centrality as a driver of preferential attachment in the evolution of research collaboration networks. *Journal of Informetrics*, 6(3):403–412, 2012. ISSN 1751-1577. doi:https://doi.org/10.1016/j.joi.2012.01.002. URL `https://www.sciencedirect.com/science/article/pii/S175115771200003X`.

Wenwen Wu, Yaofei Xie, Xiangxiang Liu, Yaohua Gu, Yuting Zhang, Xinlong Tu, and Xiaodong Tan. Analysis of scientific collaboration networks among authors, institutions, and countries studying adolescent myopia prevention and control: A review article. *Iranian journal of public health*, 48(4):621–631, Apr 2019. ISSN 2251-6085. URL `https://pubmed.ncbi.nlm.nih.gov/31110972`.

Iwan von Wartburg, Katja Rost, and Thorsten Teichert. Inventive progress measured by patent citation network analysis: Technological change in variable valve actuation for internal combustion engines. *Research Policy*, 34(10):1591–1607, 2005.

Valdis Krebs. Uncloaking terrorist networks, 2002. URL `https://firstmonday.org/ojs/index.php/fm/article/view/941/863`.

Joshua C. Grubbs, Roger I. Glass, and Peter H. Kilmarx. Coauthor Country Affiliations in International Collaborative Research Funded by the US National Institutes of Health, 2009 to 2017. *JAMA Network Open*, 2(11):e1915989–e1915989, 11 2019. ISSN 2574-3805. doi:10.1001/jamanetworkopen.2019.15989. URL https://doi.org/10.1001/jamanetworkopen.2019.15989.

Matthew Michalska-Smith, Cody Weinberger, Emilio Bruna, and Stefano Allesina. The scientific impact of nations: Journal placement and citation performance. *PloS one*, 9:e109195, 10 2014. doi:10.1371/journal.pone.0109195.

Jianfei Yu, Chunxiao Yin, Linlin Liu, and Tao Jia. A paper's corresponding affiliation and first affiliation are consistent at the country level in web of science, 2021. URL https://arxiv.org/abs/2101.09426.

Hanna Hottenrott, Michael Rose, and Cornelia Lawson. The rise of multiple institutional affiliations in academia, 2019. URL https://arxiv.org/abs/1912.05576.

DBLP. DBLP network dataset, 2019. URL https://dblp.uni-trier.de/.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

CSRankings. Csrankings, 2019. URL http://csrankings.org/.

Aric Hagberg, Pieter Swart, and Daniel Chult. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, 2008.

Vasant Dhar and Harry E. Pople. Rule-based versus structure-based models for explaining and generating expert behavior. *Commun. ACM*, 30(6):542555, jun 1987. ISSN 0001-0782. doi:10.1145/214762.214773. URL https://doi.org/10.1145/214762.214773.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. URL https://arxiv.org/abs/1409.1556.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. ISBN 0521540518.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):9931022, mar 2003. ISSN 1532-4435.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, page 849856, Cambridge, MA, USA, 2001. MIT Press.

Loet Leydesdorff. Betweenness centrality as an indicator of the interdisciplinarity of scientific journals. *J. Am. Soc. Inf. Sci. Technol.*, 58(9):1303–1319, July 2007. ISSN 1532-2882. doi:10.1002/asi.20614. URL `http://dx.doi.org/10.1002/asi.20614`.

Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 577–584, 2001.

Robert C. Moore. Reasoning about knowledge and action. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977*, pages 223–227, 1977.

Lee D. Erman and Victor R. Lesser. A multi-level organization for problem solving using many, diverse, cooperating sources of knowledge. In *Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, September 3-8, 1975*, pages 483–490, 1975.

Hans P. Moravec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977*, page 584, 1977.

Alan L. Yuille, David S. Cohen, and Peter W. Hallinan. Feature extraction from faces using deformable templates. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1989, 4-8 June, 1989, San Diego, CA, USA*, pages 104–109, 1989.

Rafael Baldasso Audibert. streamxml2json library, 2022a. URL `https://github.com/rafaeelaudibert/streamxml2json`.

Rafael Baldasso Audibert. streamxml2json library pypi, 2022b. URL `https://pypi.org/project/streamxml2json/`.

The PIP Developers. Pip - python package manager, 2008. URL https://pypi.org/project/pip/.

A Arnet Dataset

This section presents some extra data in the Arnet Dataset, which did not fit in the main work.

Tables 5 , 6, and 7 display the inner objects we referenced in Table 2.

Table 8 shows a manual count of papers published at AAAI, NeurIPS and IJCAI. This was built to support Figure 5 and bring the point that the v13 dataset did not have even half the data present at these conferences. In this table, a question mark (“?”) indicates that we could not infer the number of papers for that conference in that year.

To aggregate the data present in this table, we used AAAI’s website statistics, the NeurIPS API, and the IJCAI Proceedings page. The harder to get this data was IJCAI as they do not provide a paper account, only containing links to every paper published every year. Therefore we built a Javascript snippet that counted the number of links in those pages. Ultimately, we could not use this script in the years 1979 and 2001 because the available data format does not allow one to do such an analysis.

Figure 43 shows an example of data present in the Arnet dataset, and used throughout our work when we needed an example.

“” indicates the field was used in this work*
“?” indicates the field is optional

Field Name	**Type**	**Description**
id*	*string*	Unique ID for the author - unique across papers
name*	*string*	Full author name
org?*	*string*	Organization this author was in when of this paper

Table 5: Data structure for an *Author* entry in the Arnet JSON dataset

“” indicates the field was used in this work*
“?” indicates the field is optional

Field Name	**Type**	**Description**
id*	*string*	Unique ID for the venue - unique across papers
raw*	*string*	The raw name of the venue
name?	*string*	Humand readable name of the venue

Table 6: Data structure for a *Venue* entry in the Arnet JSON dataset

"" indicates the field was used in this work*

Field Name	**Type**	**Description**
IndexLength	*integer*	How many words in the abstract
InvertedIndex*	*hash<string, integer[]>*	Inverted index with the position of every word in the paper abstract

Table 7: Data structure for a *IndexedAbstract* entry in the Arnet JSON dataset

In Table 8, data for AAAI was extracted from their API; for NeurIPS it was extracted from their official statistics website; and for IJCAI it was manually (using JavaScript) counted on their website. Cells with a "?" text indicate the years we were not able to find an accurate count of papers for that conference, reinforcing the fact that this is a lower-bound estimate.

Table 8: Manual count of papers per main AI conference per year

	AAAI	**NeurIPS**	**IJCAI**
1969	0	0	63
1970	0	0	0
1971	0	0	58
1972	0	0	0
1973	0	0	77
1974	0	0	0
1975	0	0	141
1976	0	0	0
1977	0	0	200
1978	0	0	0
1979	0	0	*?*
1980	*?*	0	0
1981	0	0	106
1982	*?*	0	0
1983	*?*	0	233
1984	*?*	0	0
1985	0	0	257
1986	*?*	0	0
1987	*?*	90	301
1988	*?*	94	0

Table 8 continued from previous page

	AAAI	**NeurIPS**	**IJCAI**
1989	0	101	270
1990	0	143	0
1991	*?*	144	190
1992	*?*	127	0
1993	*?*	158	137
1994	341	140	0
1995	0	152	275
1996	336	152	0
1997	268	150	183
1998	269	151	0
1999	235	150	203
2000	265	152	0
2001	0	197	*?*
2002	256	207	0
2003	0	198	297
2004	250	207	0
2005	530	207	340
2006	718	204	0
2007	702	207	480
2008	648	250	0
2009	0	262	331
2010	780	292	0
2011	743	306	494
2012	707	370	0
2013	720	360	484
2014	912	411	0
2015	1101	403	656
2016	1163	569	658
2017	1049	679	782
2018	1201	1009	871
2019	1150	1428	965
2020	1591	1898	779
2021	1692	2334	722
Total	17627	13902	10553

```json
{
  "id": "1533861849",
  "title": "Understanding the difficulty of training deep
feedforward neural networks",
  "authors": [
    {
      "name": "Xavier Glorot",
      "id": "295353625"
    },
    {
      "name": "Yoshua Bengio",
      "id": "161269817"
    }
  ],
  "venue": {
    "raw": "international conference on artificial intelligence and
statistics",
    "id": "2622962978"
  },
  "year": 2010,
  "n_citation": 11,
  "page_start": "249",
  "page_end": "256",
  "doc_type": "Conference",
  "publisher": "",
  "volume": "",
  "issue": "",
  "references": [
    "1529808766",
    "1994197834",
    // [...]
    "2172174689"
  ],
  "indexed_abstract": {
    "IndexLength": 598,
    "InvertedIndex": {
      "Whereas": [0],
      "before": [1],
      "2006": [2],
      "it": [3, 452, 576, 593],
      // [...]
      "drastic": [596],
      "impact).": [597]
    }
  },
  "fos": [
    {"name": "Machine learning", "w": 0.45906812},
    {"name": "Gradient descent", "w": 0.454203367},
    // [...]
    {"name": "Artificial intelligence", "w": 0.0}
  ]
}
```


Figure 43: Example of a JSON entry for Glorot and Bengio [2010] in the Arnet dataset *[...] indicates some items in the array were abbreviated for sake of brevity.*

B Codebase

In this section, we present Algorithm 4 used in Section 3.4 along Algorithm 3 to be able to infer a country of origin from an organization. This basically removes the clutter present in Arnet's data.

Table 9 shows every open-source library used to develop this work. We are very thankful for every library contributor's work to the open-source community.

Library Name	**Usage**
click	Create CLI to run experiments with different parameters
fire	Create CLI to run experiments with different parameters
matplotlib	Plot the charts
networkx	Build the graph datasets
nltk	Tokenize words and detect stop words
numpy	Manipulate data arrays in a vector-fashion
scipy	Compute Spearman Correlations
seaborn	Improve matplotlib's plots look
sklearn	Generate linear models and compute TF-IDF
tqdm	Generate progress bars for long data processing pipelines

Table 9: Python libraries used in this work

Algorithm 4 Organization Name Cleaning Preprocessing

```
Require: org                                          ▷ Organization name
  org ← split(org, ",")        ▷ Split the text in every comma, turning it into a list
  org ← org[-1]                                       ▷ Last item in the array
  org ← replace(org, "#TAB#", "")                     ▷ Remove unknown tag
  org ← replace(org, "#tab#", "")                     ▷ Remove unknown tag
  org ← replace(org, /[\(\)\[\]\-_]/, "")             ▷ Regex-based replacement
      return org
```

C Author Citation

The charts presented in this section are related to centralities from Section 3.3.1.

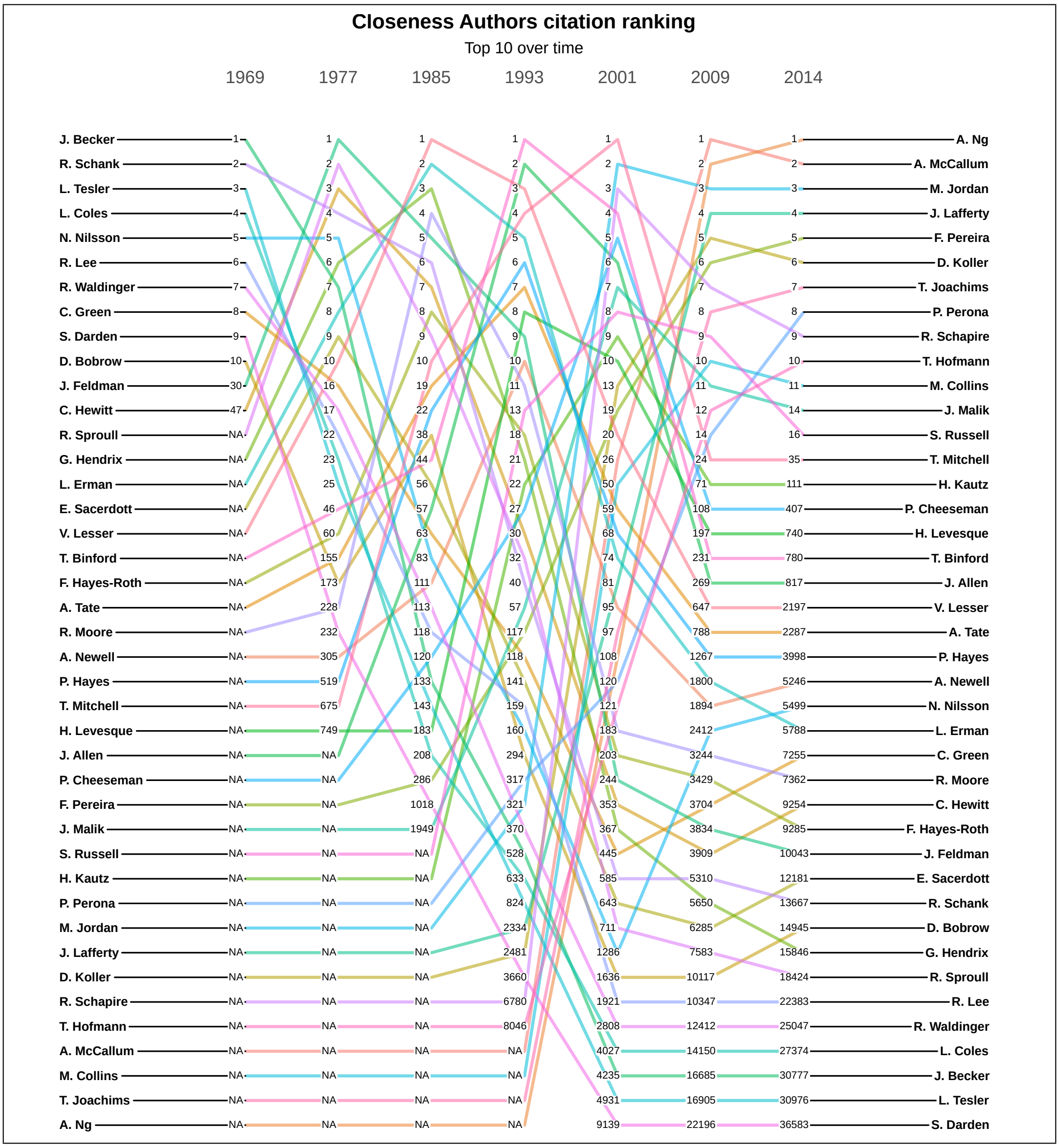


Figure 44: Authors citation ranking over time according to Closeness centrality. Figure refers to Appendix C

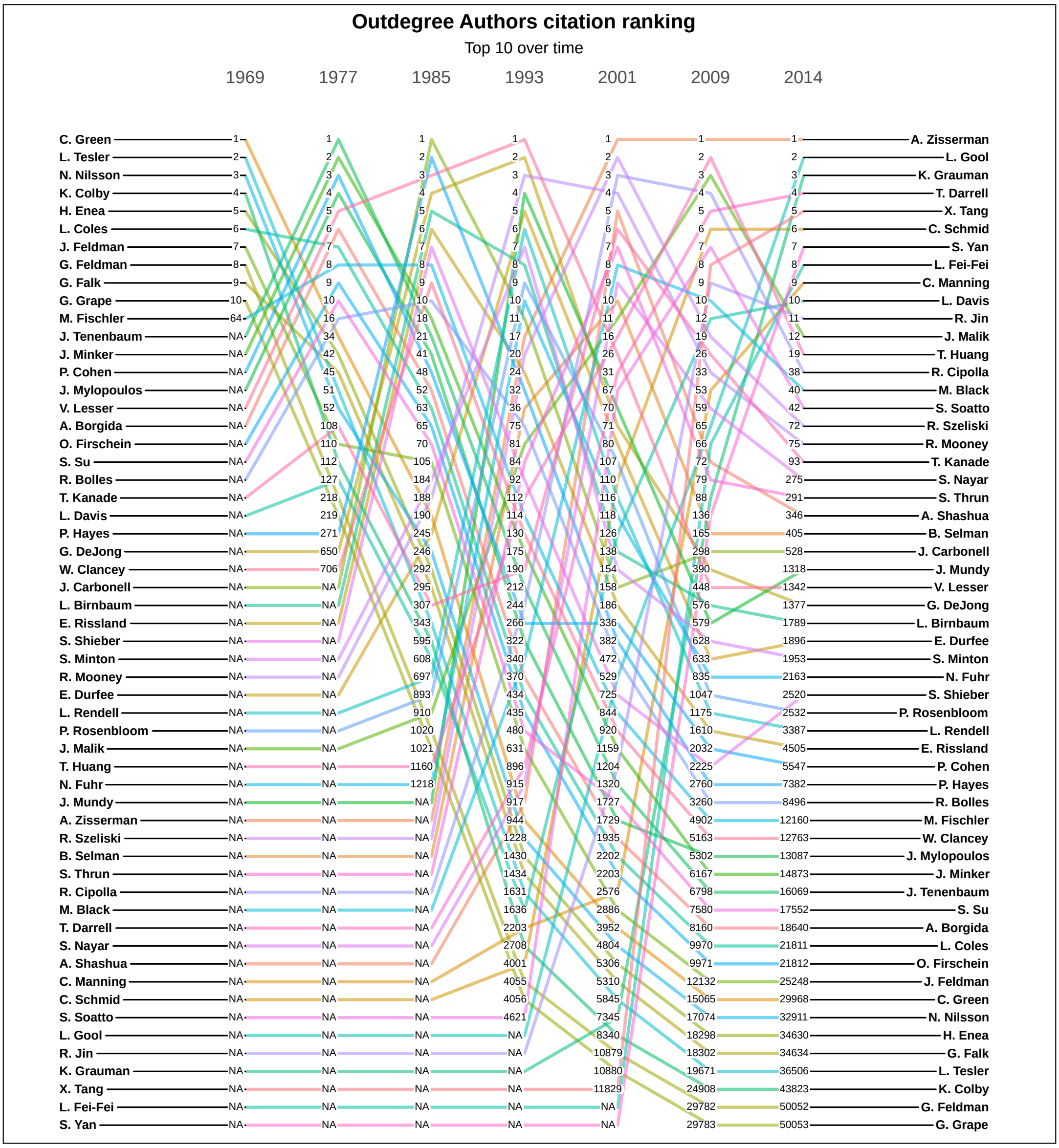


Figure 45: Authors citation ranking over time according to Out-degree centrality. Figure refers to Appendix C

D Author Collaboration

The charts presented in this section are related to centralities from Section 4.3.

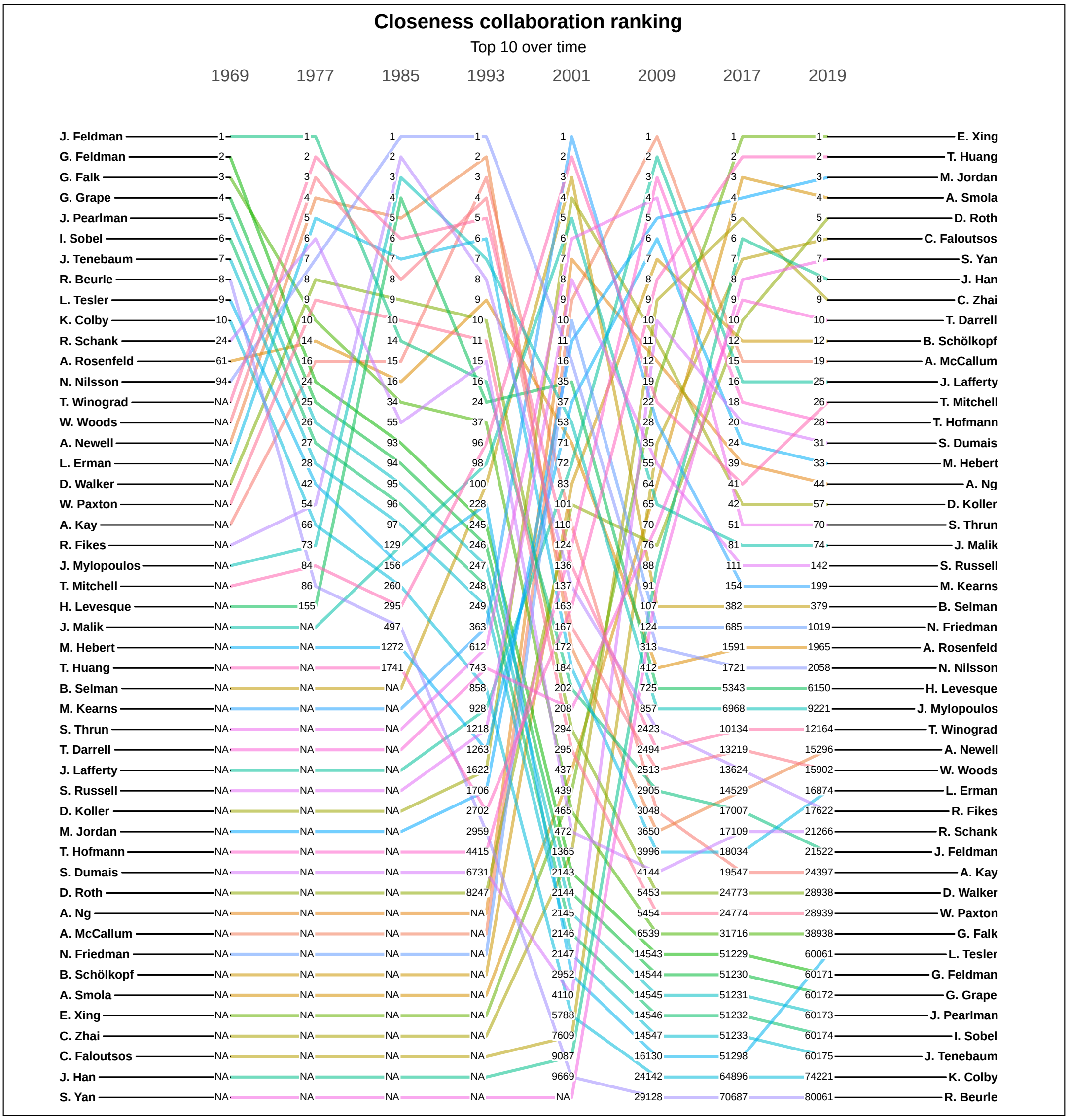


Figure 46: Authors collaboration ranking over time according to Closeness centrality. Figure refers to Appendix D

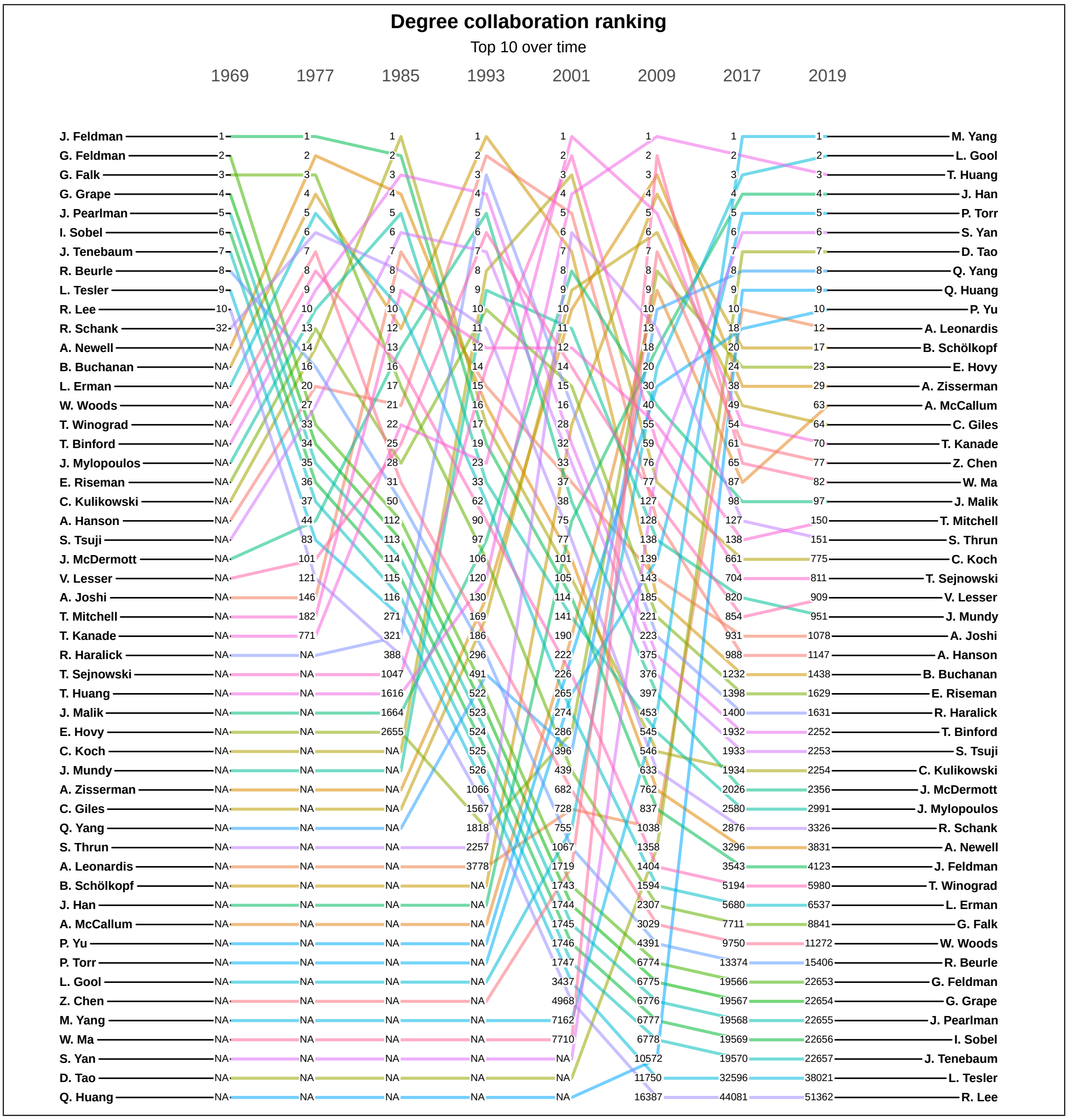


Figure 47: Authors collaboration ranking over time according to In-Degree centrality. Figure refers to Appendix D

E Paper Citation

The charts presented in this section are related to centralities from Section 4.4. Similarly, Table 10 is used to map every paper entry in these charts to a paper title and year.

Table 10: Dictionary for the papers which appeared in the Top 5 rankings

Initials	Title	Year	Venue
AAAOSL92	An asymptotic analysis of speedup learning	1992	ICML
AALR85	AI and legal reasoning	1985	IJCAI
AAOLTPAASP92	An analysis of learning to plan as a search problem	1992	ICML
AASTNAP69	An augmented state transition network analysis procedure	1969	IJCAI
ACDAAS90	Accurate corner detection: an analytical study	1990	ICCV
ACPFNL69	A conceptual parser for natural language	1969	IJCAI
ACRSFFL69	A contextual recognition system for formal languages	1969	IJCAI
ACSOICI86	A case study of incremental concept induction	1986	AAAI
AEARCFLG90	AUTOMATICALLY EXTRACTING AND REPRESENTING COLLOCATIONS FOR LANGUAGE GENERATION	1990	ACL
AIIFIR83	Artificial intelligence implications for information retrieval	1983	SIGIR
AIIRTWAATSV81	An iterative image registration technique with an application to stereo vision	1981	IJCAI
ALOIAEB84	A logic of implicit and explicit belief	1984	AAAI
AMAAAOAIT69	A mobius automation: an application of artificial intelligence techniques	1969	IJCAI
AMEMFPT96	A Maximum Entropy Model for Part-Of-Speech Tagging	1996	EMNLP

Continued on next page

Table 10 – *Continued from previous page*

Initials	Title	Year	Venue
AMOFPSUMDCSOK75	A multi-level organization for problem solving using many, diverse, cooperating sources of knowledge	1975	IJCAI
ANSFSISDAR71	A net structure for semantic information storage, deducation and retrieval	1971	IJCAI
AOMOAHC75	Acquisition of moving objects and hand-eye coordination	1975	IJCAI
AOTPTPS69	Application of theorem proving to problem solving	1969	IJCAI
APFFSC01	A probabilistic framework for space carving	2001	ICCV
APFQCTSCOEG91	A Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars	1991	NAACL
ARVOTHA77	A retrospective view of the Hearsay-II architecture	1977	IJCAI
ASNAAMOHM73	Active semantic networks as a model of human memory	1973	IJCAI
AUMAFFAI73	A universal modular ACTOR formalism for artificial intelligence	1973	IJCAI
BAMFAEOMT02	Bleu: a Method for Automatic Evaluation of Machine Translation	2002	ACL
BBOWEMDAFIIR08	Beyond bags of words: effectively modeling dependence and features in information retrieval	2008	SIGIR
BTMANEFTEOVM97	Boosting the margin: A new explanation for the effectiveness of voting methods	1997	ICML

Continued on next page

Table 10 – *Continued from previous page*

Initials	Title	Year	Venue
CCALMFIA11	Combining concepts and language models for information access	2011	SIGIR
CKCWBK01	Constrained K-means Clustering with Background Knowledge	2001	ICML
CLMFTC96	Context-sensitive learning methods for text categorization	1996	SIGIR
CPS84	Classification problem solving	1984	AAAI
CRFPMFSALSD01	Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data	2001	ICML
CSS73	Case structure systems	1973	IJCAI
CSTAE92	Camera Self-Calibration: Theory and Experiments	1992	ECCV
DCOEW93	DISTRIBUTIONAL CLUSTERING OF ENGLISH WORDS	1993	ACL
DRLFIR16	Deep Residual Learning for Image Recognition	2016	CVPR
DROWAPATC13	Distributed Representations of Words and Phrases and their Compositionality	2013	NIPS
EAAC89	Execution architectures and compilation	1989	IJCAI
EAFMCVE94	Efficient algorithms for minimizing cross validation error	1994	ICML
ETUOSNTP75	Expanding the utility of semantic networks through partitioning	1975	IJCAI
EWANBA96	Experiments with a new boosting algorithm	1996	ICML

Continued on next page

Table 10 – *Continued from previous page*

Initials	Title	Year	Venue
EWASAFTDBOAHBS69	Experiments with a search algorithm for the data base of a human belief structure	1969	IJCAI
EWSRD13	Explicit web search result diversification	2013	SIGIR
FAATIOAIOS73	Forecasting and assessing the impact of artificial intelligence on society	1973	IJCAI
FCNFSS15	Fully convolutional networks for semantic segmentation	2015	CVPR
FEFFUDT89	Feature extraction from faces using deformable templates	1989	CVPR
FOAITHSUS77	Focus of attention in the Hearsay-II speech understanding system	1977	IJCAI
FRUE91	Face recognition using eigenfaces	1991	CVPR
GPN77	Generating project networks	1977	IJCAI
HEFANLP77	Human Engineering for Applied Natural Language Processing.	1977	IJCAI
HMME92	Hierarchical Model-Based Motion Estimation	1992	ECCV
HOOGFHD05	Histograms of oriented gradients for human detection	2005	CVPR
HTUWYK75	How to use what you know	1975	IJCAI
IAA88	Interpretation as Abduction	1988	ACL
IAFLPARBOADP90	Integrated architecture for learning, planning, and reacting based on approximating dynamic programming	1990	ICML
ICWDCNN12	ImageNet Classification with Deep Convolutional Neural Networks	2012	NIPS

Continued on next page

Table 10 – *Continued from previous page*

Initials	Title	Year	Venue
IEAUWA08	Intelligent email: aiding users with AI	2008	AAAI
IFATSSP94	Irrelevant features and the subset selection problem	1994	ICML
IMAMFEMFIS69	Implicational molecules: a method for extracting meaning from input sentences	1969	IJCAI
INLG01	Instance-based natural language generation	2001	NAACL
ISARASAFD77	Information storage and retrieval: a survey and functional description	1977	SIGIR
LAAATFTOLA85	Lexical ambiguity as a touchstone for theories of language analysis	1985	IJCAI
LELASTTITP89	Lazy explanation-based learning: a solution to the intractable theory problem	1989	IJCAI
LPPKICE96	Learning procedural planning knowledge in complex environments	1996	AAAI
LRCNFVRAD15	Long-term recurrent convolutional networks for visual recognition and description	2015	CVPR
LTGCWCNN15	Learning to generate chairs with convolutional neural networks	2015	CVPR
LTRFIR10	Learning to rank for information retrieval	2010	SIGIR
LTRNLAAUA98	Learning to resolve natural language ambiguities: a unified approach	1998	AAAI
LTRWPD08	Learning to rank with partially-labeled data	2008	SIGIR

Continued on next page

Table 10 – *Continued from previous page*

Initials	Title	Year	Venue
MIFRVS91	Multidimensional indexing for recognizing visual shapes	1991	CVPR
MRLUAV98	Mobile Robot Localisation Using Active Vision	1998	ECCV
MSCBSSFVSTN3MSAR01	Multi-view scene capture by surfel sampling: from video streams to non-rigid 3D motion, shape and reflectance	2001	ICCV
MSDIAAN71	Managing semantic data in an associative net	1971	SIGIR
NATCA18	Neural Approaches to Conversational AI	2018	SIGIR
NCFPS90	NOUN CLASSIFICATION FROM PREDICATE-ARGUMENT STRUCTURES	1990	ACL
PASTAPW69	PROW: a step toward automatic program writing	1969	IJCAI
PAUAODNPID83	PROVIDING A UNIFIED ACCOUNT OF DEFINITE NOUN PHRASES IN DISCOURSE	1983	ACL
PEOKIP71	Procedural embedding of knowledge in planner	1971	IJCAI
POACBC75	Progress on a computer based consultant	1975	IJCAI
POOFT92	Performance of optical flow techniques	1992	CVPR
PSGANL83	Phrase structure grammars and natural languages	1983	IJCAI
RAKAA77	Reasoning about knowledge and action	1977	IJCAI
RODUABCOSF01	Rapid object detection using a boosted cascade of simple features	2001	CVPR

Continued on next page

Table 10 – *Continued from previous page*

Initials	**Title**	**Year**	**Venue**
SAATNICGWVA15	Show, Attend and Tell: Neural Image Caption Generation with Visual Attention	2015	ICML
SCASFPM08	Server characterisation and selection for personal metasearch	2008	SIGIR
SF83	Scale-space filtering	1983	IJCAI
SMIS14	Semantic Matching in Search	2014	SIGIR
STFSWA06	Semi-Supervised Training for Statistical Word Alignment	2006	ACL
TAOAITACSOKE77	The art of artificial intelligence: themes and case studies of knowledge engineering	1977	IJCAI
TAOALHWSE98	The anatomy of a large-scale hypertextual Web search engine	1998	WWW
TAVOA77	Towards automatic visual obstacle avoidance	1977	IJCAI
TCDAHOITP87	The classification, detection and handling of imperfect theory problems	1987	IJCAI
TCOSP89	Term clustering of syntactic phrases	1989	SIGIR
THSUSAEOTRP73	The hearsay speech understanding system: an example of the recognition process	1973	IJCAI
TMOSAAIPIASMS69	The modeling of simple analogic and inductive processes in a semantic memory system	1969	IJCAI
TSHP69	The Stanford hand-eye project	1969	IJCAI
TUGIAIE83	Tracking user goals in an information-seeking environment	1983	AAAI
TWARIE69	Talking with a robot in English	1969	IJCAI

Continued on next page

Table 10 – *Continued from previous page*

Initials	**Title**	**Year**	**Venue**
UDTTICL93	Using decision trees to improve case-based learning	1993	ICML
VMBARR79	Visual mapping by a robot rover	1979	IJCAI
WCBSITDWAUSR92	What can be seen in three dimensions with an uncalibrated stereo rig	1992	ECCV
WSNGVG88	What Size Net Gives Valid Generalization	1988	NIPS
WYCRNTKYLIK92	What your computer really needs to know, you learned in kindergarten	1992	AAAI

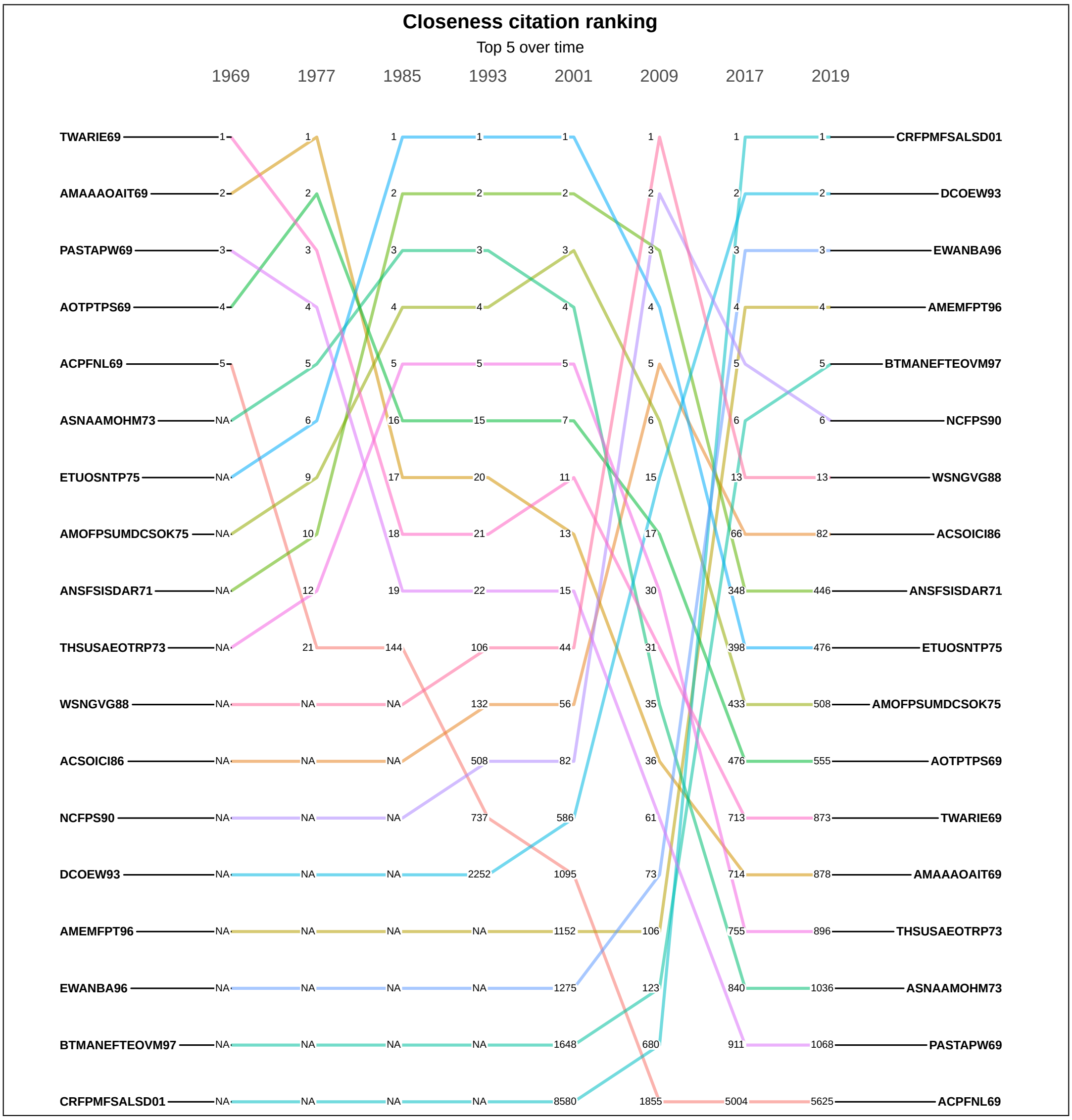


Figure 48: Papers citation ranking over time according to Closeness centrality. *N/A stands for papers that had not been published in the selected venues until that year. Please refer to Table 10 in Appendix E to see the details of each ranked paper.*

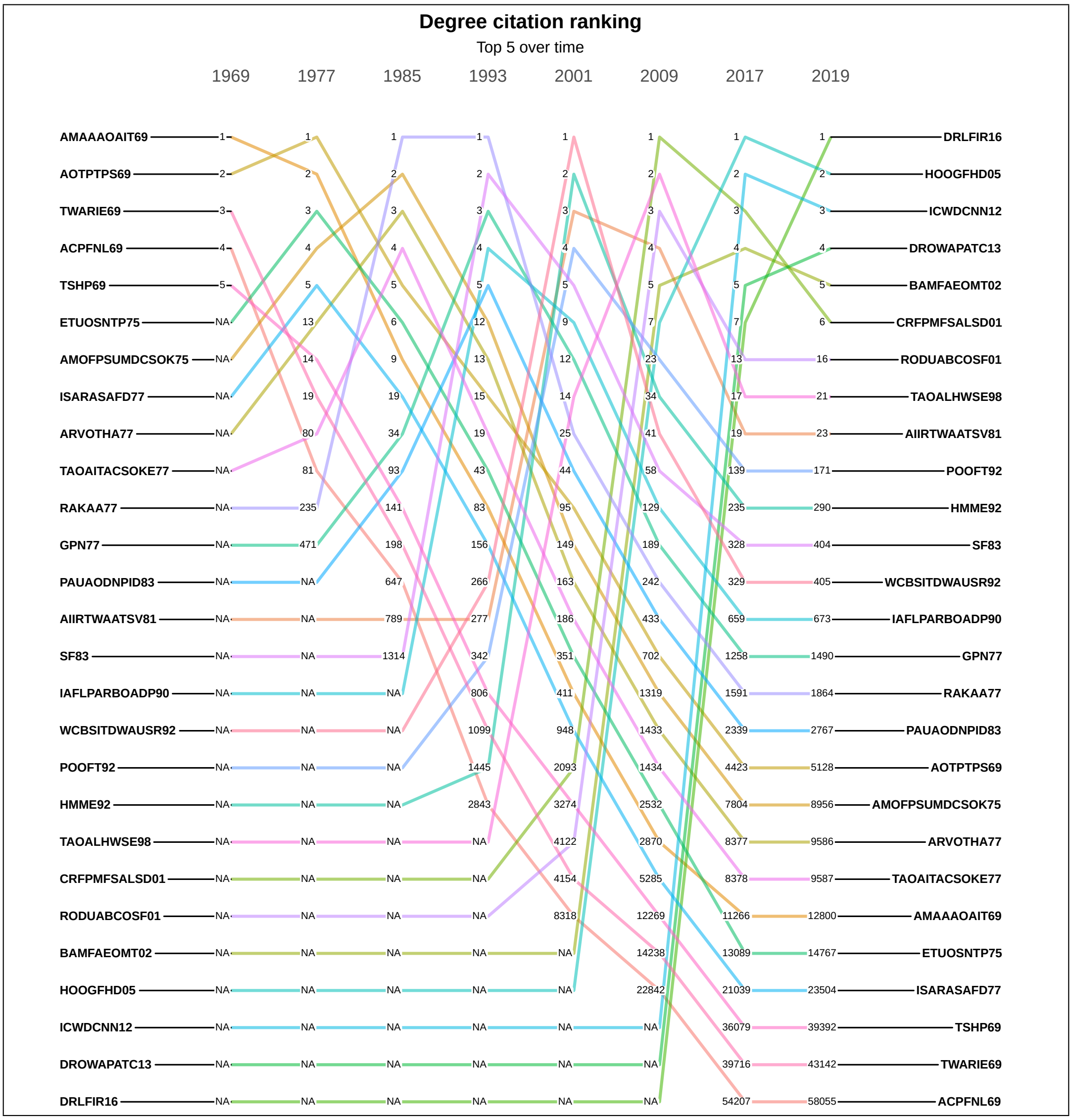


Figure 49: Papers citation ranking over time according to In-Degree centrality. *N/A stands for papers that had not been published in the selected venues until that year. Please refer to Table 10 in Appendix E to see the details of each ranked paper.*

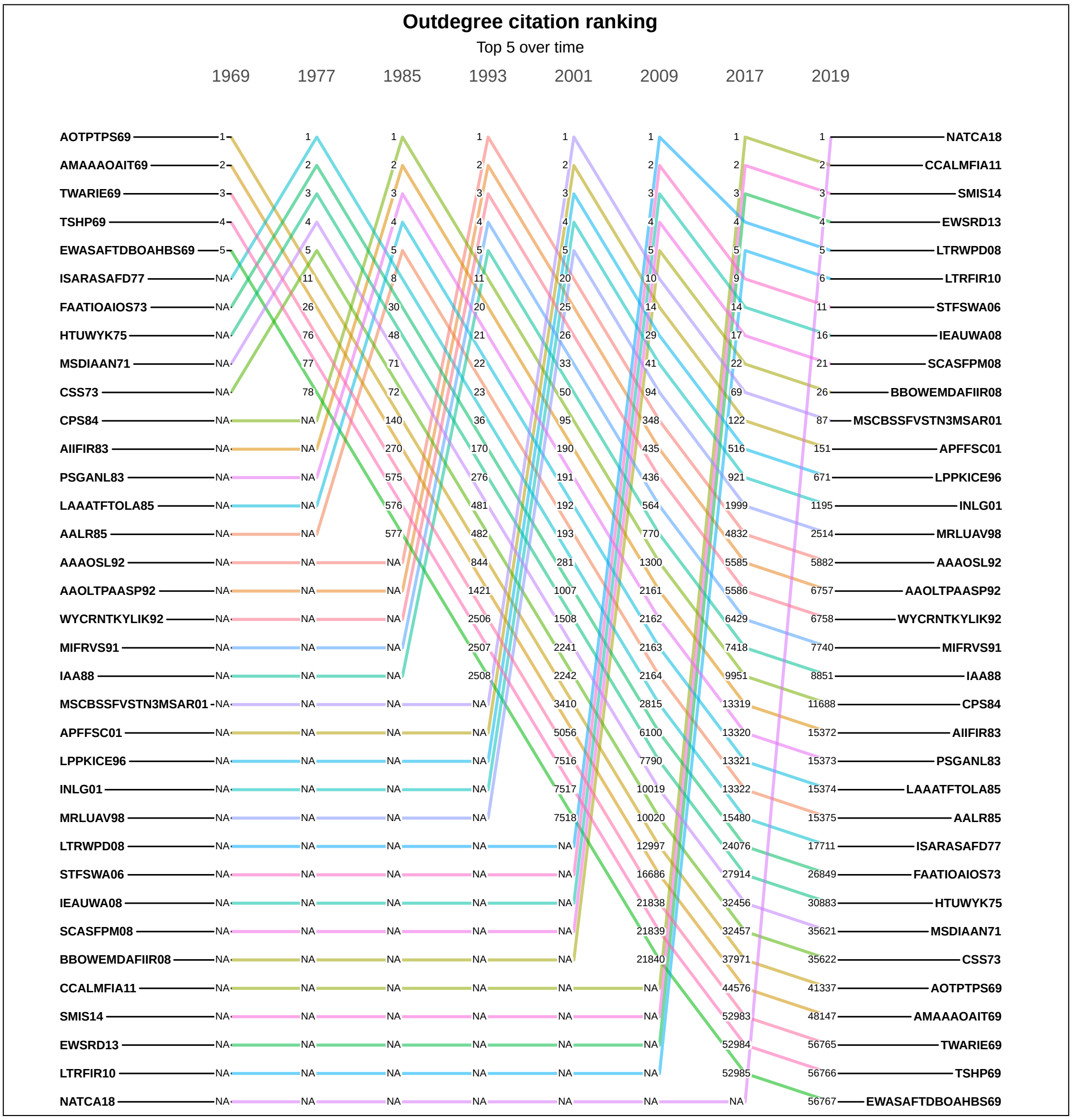


Figure 50: Papers citation ranking over time according to Out-degree centrality. *N/A stands for papers that had not been published in the selected venues until that year. Please refer to Table 10 in Appendix E to see the details of each ranked paper.*

F Country Citation Graph

Figure 51 presents a different view than the one available at Section 4.6 by generating the stacked version of the countries but with a 2-year-wide sliding average window, i.e. every datapoint is actually the average between the year and its prior window, trying to avoid the variation seen in Figure 37 because of IJCAI being held only in odd-numbered years. The steady decline of the USA share in the graph is clearly seen.

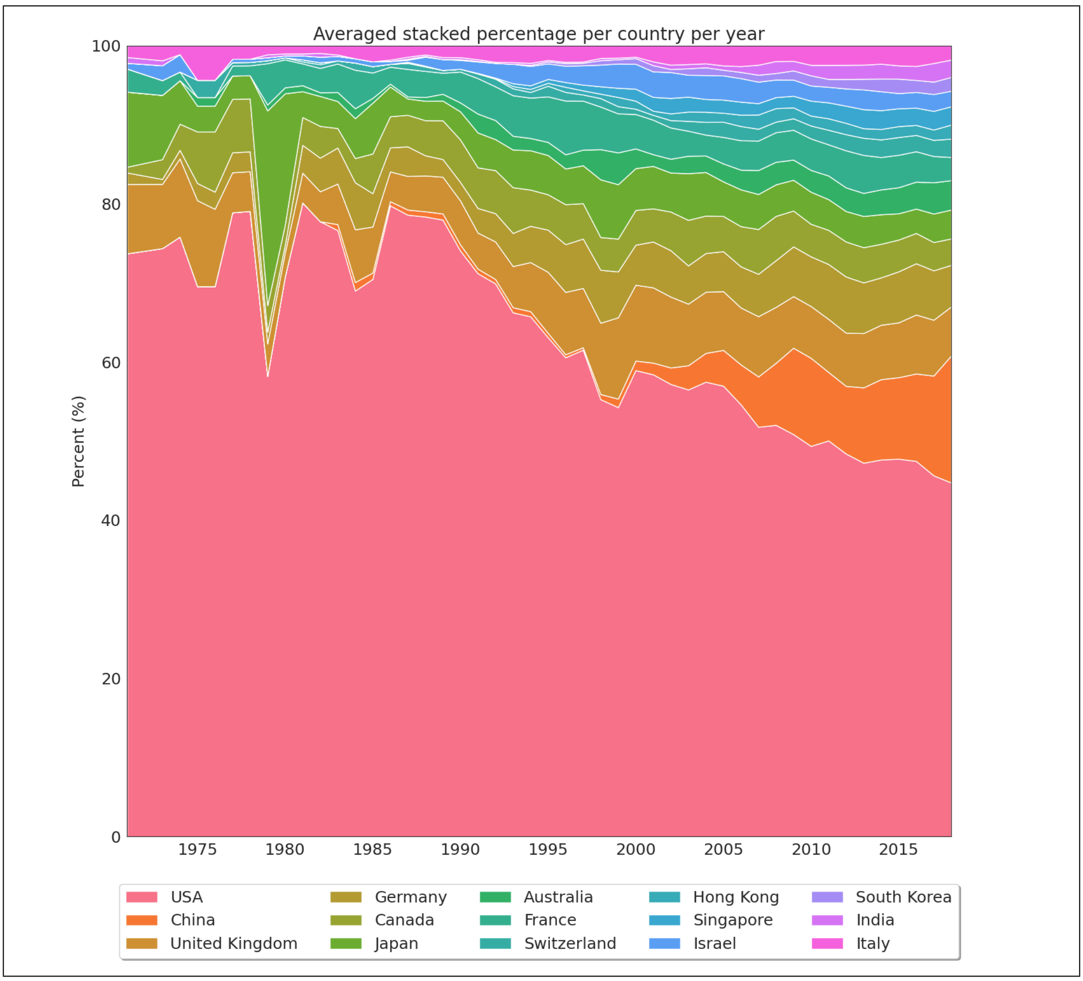


Figure 51: Stacked percentage of papers viewed with a 2-years-wide sliding average window

G Turing Award Charts

This section presents some charts related to the correlation between Turing Awardees papers abstracts and the other authors' abstracts words. We used a Spearman Correlation to compute these.

Table 11 contains every single Turnig Award winner, with the year they won the prize and also their nationality.

Table 11: Turing Award Winners per year
** indicates the winner is deceased*

Year	Winner	Nationality
1966	Perlis, Alan J. *	United States
1967	Wilkes, Maurice V. *	United Kingdom
1968	Hamming, Richard W. *	United States
1969	Minsky, Marvin *	United States
1970	Wilkinson, James Hardy ("Jim") *	United Kingdom
1971	McMarthy, John *	United States
1972	Dijkstra, Edsger Wybe *	Netherlands
1973	Bachman, Charles William *	United States
1974	Knuth, Donald ("Don") Ervin	United States
1975	Newel, Allen *	United States
	Simon, Herbert ("Herb") Alexander *	United States
1976	Rabin, Michael Oser	Israel
	Scott, Dana Stewart	United States
1977	Backus, John *	United States
1978	Floyd, Robert (Bob) W. *	United States
1979	Iverson, Kenneth E. ("Ken") *	Canada
1980	Hoare, C. Antony ("Tony") R.	United Kingdom
1981	Codd, Edgar F. ("Ted") *	United Kingdom
1982	Cook, Stephen Arthur	United States

Table 11 – *Continued from previous page*

Year	Winner	Nationality
1983	Ritchie, Dennis M. *	United States
	Thompson, Kenneth Lane	United States
1984	Wirth, Niklaus E.	Switzerland
1985	Karp, Richard ("Dick") Manning	United States
1986	Hopcroft, John E	United States
	Tarjan, Robert (Bob) Endre	United States
1987	Cocke, John *	United States
1988	Sutherland, Ivan	United States
1989	Kahan, William ("Velvel") Morton	Canada
1990	Corbato, Fernando J. ("Corby") *	United States
1991	Milner, Arthur John Robin Gorell ("Robin") *	United Kingdom
1992	Lampson, Butler W.	United States
1993	Hartmanis, Juris	United States
	Stearns, Richard ("Dick") Edwin	United States
1994	Feigenbaum, Edward A. ("Ed")	United States
	Reddy, Dabbala Rajagopal ("Raj")	India
1995	Blum, Manuel	United States
1996	Pnueli, Amir *	Israel
1997	Engelbart, Douglas *	United States
1998	Gray, James ("Jim") Nicholas *	United States
1999	Brooks, Frederick ("Fred")	United States
2000	Yao, Andrew Chi-Chih	China
2001	Dahl, Ole-Johan *	Norway
	Nygaard, Kristen	Norway
2002	Adleman, Leonard (Len) Max	United States

Table 11 – *Continued from previous page*

Year	Winner	Nationality
	Rivest, Ronald (Ron) Linn	United States
	Shamir, Adi	Israel
2003	Kay, Alan	United States
2004	Cerf, Vinton ("Vint") Gray	United States
	Kahn, Robert ("Bob") Elliot	United States
2005	Naur, Peter *	Denmark
2006	Allen, Frances ("Fran") Elizabeth *	United States
2007	Clarke, Edmund Melson *	United States
	Emerson, E. Allen	United States
	Sifakis, Joseph	France
2008	Liskov, Barbara	United States
2009	Thacker, Charles P. (Chuck) *	United States
2010	Valiant, Leslie Gabriel	United Kingdom
2011	Pearl, Judea	Israel
2012	Goldwasser, Shafi	United States
	Micali, Silvio	Italy
2013	Lamport, Leslie	United States
2014	Stonebraker, Michael	United States
2015	Diffie, Whitfield	United States
	Hellman, Martin	United States
2016	Bernes-Lee, Tim	United Kingdom
2017	Hennesy, John L.	United States
	Patterson, David	United States
2018	Bengio, Yoshua	Canada
	Hinton, Geoffrey E.	United Kingdom

Table 11 – *Continued from previous page*

Year	Winner	Nationality
	LeCun, Yann	France
2019	Catmull, Edwin E.	United States
	Hanrahan, Patrick M.	United States
2020	Aho, Alfred Vaino	Canada
	Ullman, Jeffrey David	United States
2021	Dongarra, Jack	United States

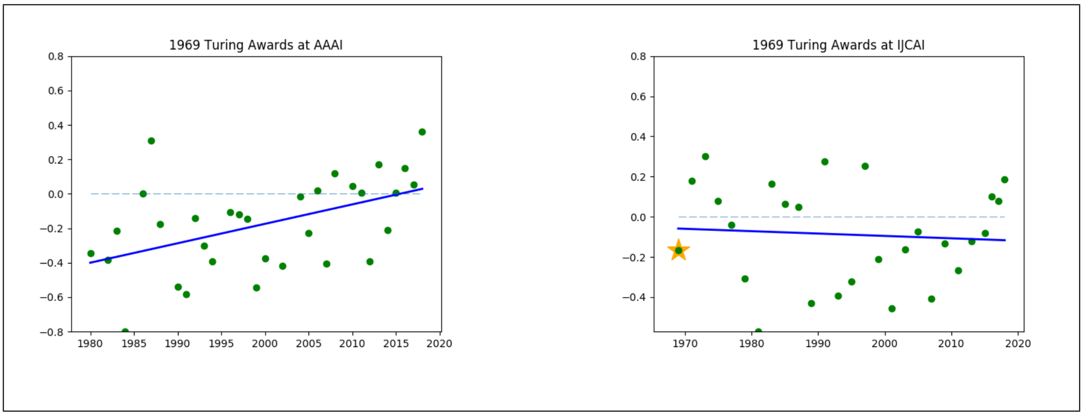


Figure 52: Correlation between 1969 Turing Award Winner papers and AAAI and IJCAI-published ones.

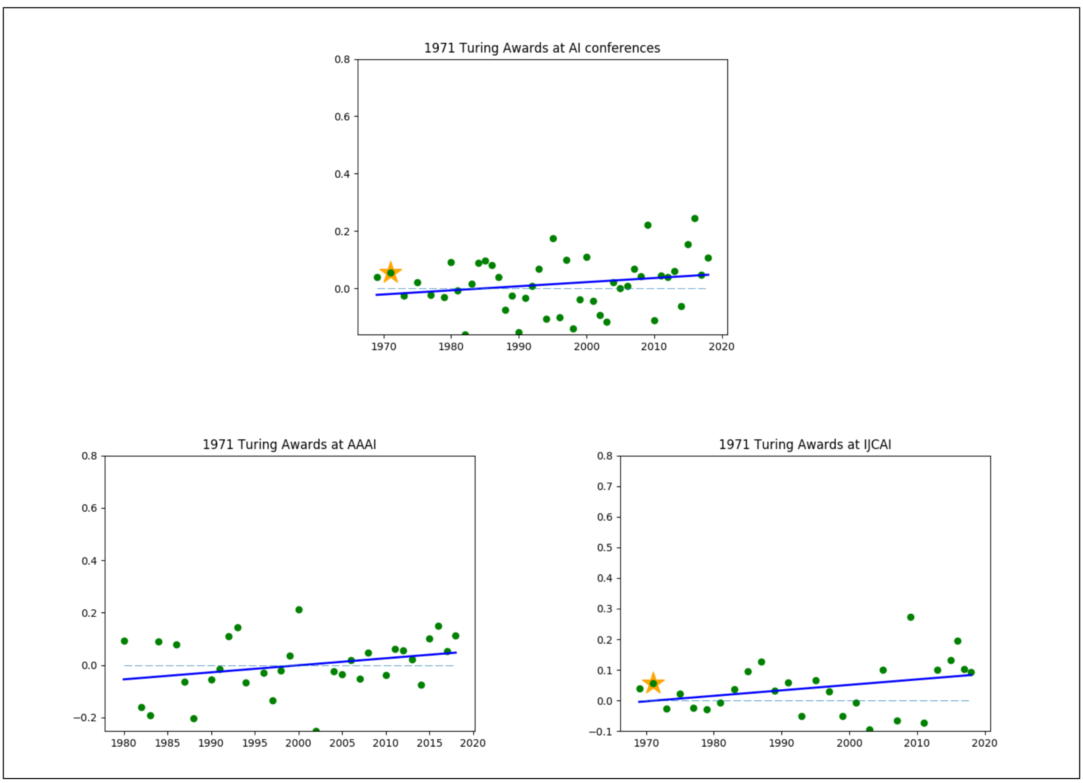


Figure 53: Correlation between titles of papers published by the 1971 Turing Award winner and titles of papers published in the three AI flagship conferences.

Figure 54: Correlation between titles of papers published by the 1975 Turing Award winners and titles of papers published in the three AI flagship conferences.

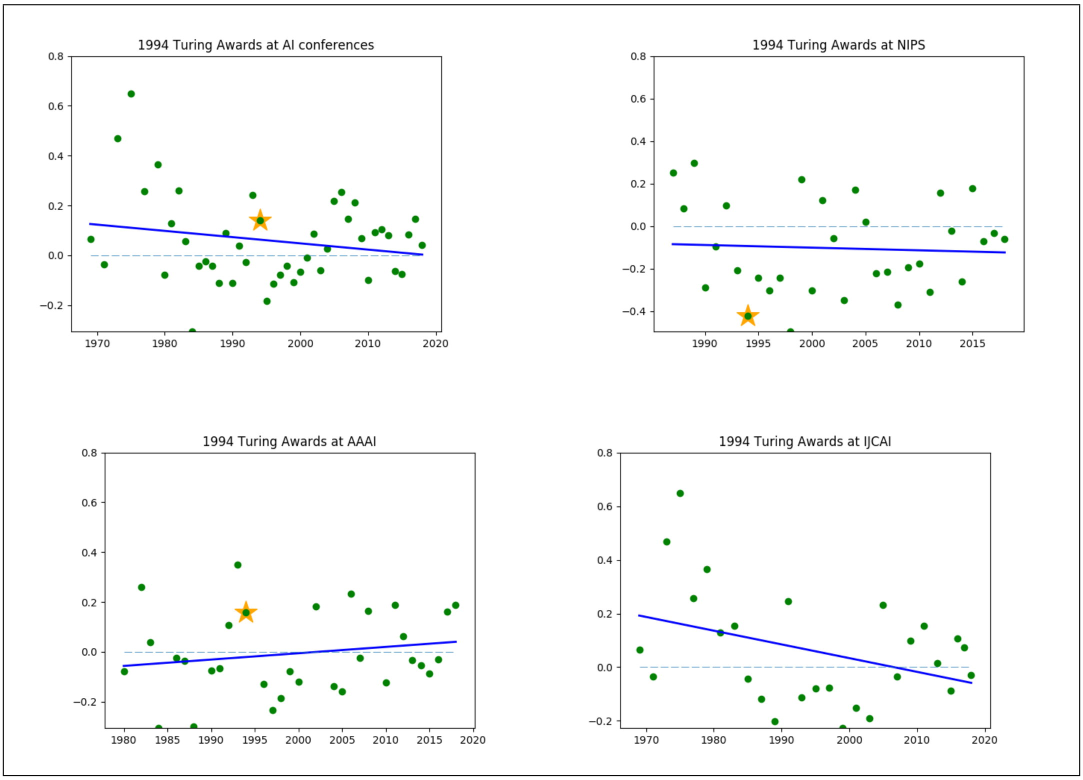


Figure 55: Correlation between titles of papers published by the 1994 Turing Award winners and titles of papers published in the three AI flagship conferences.

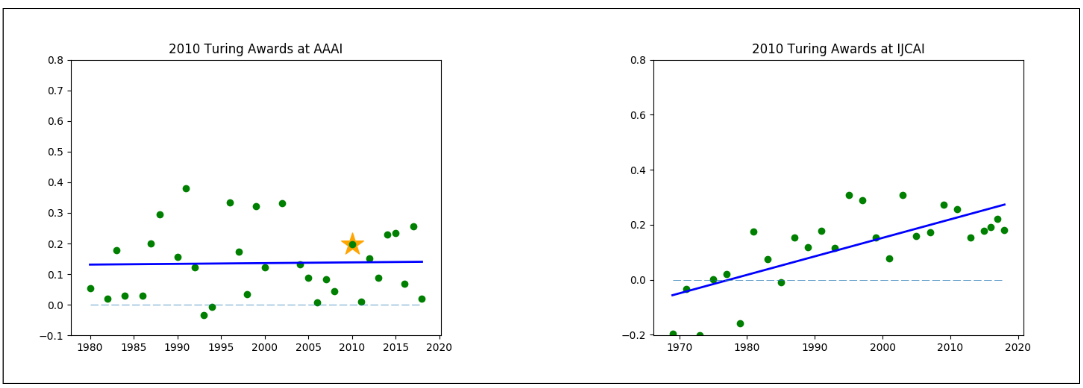


Figure 56: Correlation between titles of papers published by the 2010 Turing Award winners and titles of papers published in AAAI and IJCAI.

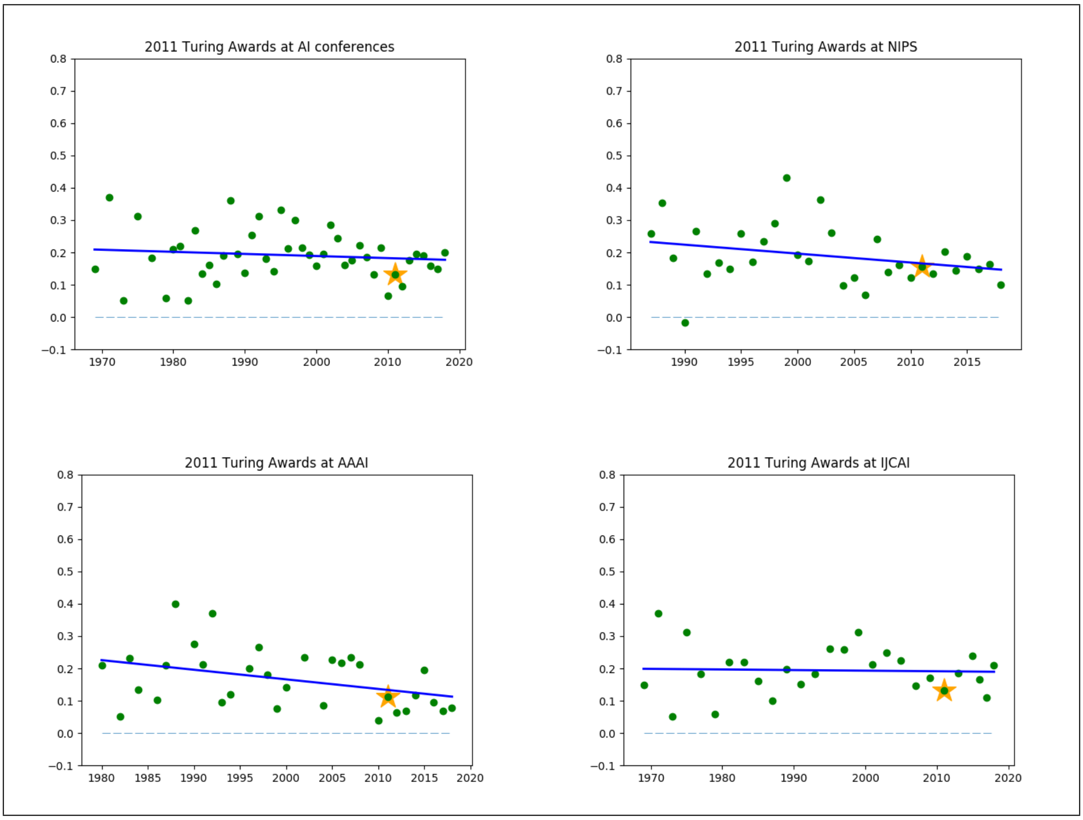


Figure 57: Correlation between titles of papers published by the 2011 Turing Award winner and titles of papers published in the three AI flagship conferences.

H Software Contributions

To carry out this work, we have built some pieces of software that might be used by others in similarly sized tasks. They are briefly described and discussed below.

H.1 *streamxml2json* Library

When we were trying to use the original DBLP dataset (See Section 3.1 for more context) we had some trouble when trying to convert the downloaded XML file to a JSON file we could more easily manipulate. The benefits of a JSON file over the XML file go from being more human-readable to the fact of it being a bit smaller (in our case, a 3.3GB XML yields a 3GB JSON file, a 10% size reduction) – therefore, easier to load in memory. It is a fact, however, that because we had the intention to parse this file in a CI environment, to be able to generate new charts (See Section 5.2) weekly we would need to be able to do this conversion from XML to JSON without loading the whole file into memory. After searching on Github and PyPi we realized that a tool to convert from XML to JSON without loading the whole file in memory did not exist.

That clarified, we decided we could build such a tool by using a few already existent libraries as building blocks: simplejson[48], jsonstreams[49] and xmltodict[50]. Streaming over any XML file and parsing only the necessary data, we can then output it to a JSON file, also through a file stream, without any substantial memory usage. Because our data was gzipped, the library supports reading directly from a *.xml.gz* file, not requiring the user to unzip it.

The library *streamxml2json* Audibert [2022a] Audibert [2022b] is available at Github in `https://github.com/rafaeelaudibert/streamxml2json` and publicly downloadable from PyPi on `https://pypi.org/project/streamxml2json/`. For the sake of completeness, the library can be downloaded if you have *pip* Developers [2008] installed in your machine by running *"pip install streamxml2json""*.

In the end, because we did not use this dataset, we do not use this library in our work, but the contribution was deemed important enough to the whole Python ecosystem in general so we are adding it to this section. We did keep in our main repository the file used to convert from XML to JSON, as a library usage example: `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/scripts/xml2json.py`.

[48] `https://github.com/simplejson/simplejson`
[49] `https://github.com/dcbaker/jsonstreams`
[50] `https://github.com/martinblech/xmltodict`

H.2 Python Parallel Centralities Implementation

Throughout our work, we used UFRGS HPC Group's (PCAD[51] supercomputers to be able to properly generate the graph we were building. We had to use their supercomputers because when we are computing graph centralities we need a lot of memory - for betweenness, we need to store the shortest path between every single node of our graph that contains more than 100,000 nodes. Computing these centralities, however, was still pretty slow because we have to do it for every single node in every single year. An easy way to increase speed in computation, especially when you are using supercomputers, is to parallelize your job across the available physical processors. In our case, we had access to a machine with 16 cores (32 threads) allowing us to compute our results a lot faster.

Therefore, using Networkx's implementations as a base, we developed a parallel Betweenness and a parallel Closeness algorithm capable of running close to 5x faster in a machine with 16 cores. The results are not 16x faster than expected because of Python's GIL which severely degrades Python's parallel performance.

The codes for these implementations can be found in Github. [52] [53].

H.3 Graph Parsing pipeline

In our work, we had to generate several different types of graphs, with several different parameters in each of them. We also wanted to be able to easily cache data we had already computed, avoiding unnecessary computation.

To solve these problems, we devised a simple structure where we could extend a base *GenerateGraph* class (available in `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/generate_graph.py`) that exposed several methods that made our job easier. Some of the exposed methods help us in the process of caching our data. Whenever we want to build a new graph, if we have no caching, we need to do these steps:

1. Filter papers from the required venues from DBLP's JSON file
2. Generate the full graph for every year
3. After the full graph is complete, compute the centralities

[51] `http://gppd-hpc.inf.ufrgs.br/`

[52] `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/parallel_betweenness.py`

[53] `https://github.com/rafaeelaudibert/conferences_insights/blob/v11/graph_generation/parallel_closeness.py`

If we always followed these steps, whenever we made a code change to the centralities computation, we would need to run everything before. We can easily solve this by calling some of the base class helper methods that know how to save a pre-parsed list of papers from selected venues, or even an already partial graph if we had only built it until a given year (imagine you noticed something wrong or an exception was raised after you had parsed half the dataset).

Also, to be able to control which type of graph we wanted to run from the command line, we built a CLI on top of this class using Google's fire[54] library. It is used to automatically generate a CLI from the parameters of a function, effectively allowing us to simply add a new parameter to a function and then pass the parameter value from the command line to properly pass the parameters to our code.

When we want a new type of graph, therefore, we simply extend this *GenerateGraph* class and add a new parameter to the main function, allowing us to easily call this new type of graph generation.

It is worth noting, however, that, ideally, *fire* should be replaced by *click*. Click[55] is a more maintained library, with better features: automatically generated fully-customizable help command; subcommands to avoid the extra work of manually creating flags when creating new types of graphs; proper filename handling; and etc.

[54] `https://github.com/google/python-fire`
[55] `https://click.palletsprojects.com/en/8.1.x/`

 Received August 2022

The Annihilating-ideal Graphs of MV-algebras

Xiaoxue Zhang
School of Mathematics and Statistics, Shandong Normal University, 250014, Jinan, P. R. China

Hongxing Liu*
School of Mathematics and Statistics, Shandong Normal University, 250014, Jinan, P. R. China
lhxshanda@163.com

Abstract

In this paper, we introduce and study the annihilating-ideal graph of an MV-algebra $(A, \oplus, *, 0)$. The algebraic structure of MV-algebras (especially Boolean algebras) are described by using the annihilating-ideal graph. The connections between the ideal theory of MV-algebras and graph theory are established, which promote the studying of the coloring of graphs. The annihilating-ideal graph $\mathbb{AG}(A)$ is a simple graph with the vertex set $V(\mathbb{AG}(A)) = \{I \in \mathbb{I}(A)\backslash\{\langle 0\rangle, A\} \mid \exists J \in \mathbb{I}^*(A)$ such that $IJ = \langle 0\rangle\}$ and the edge set $E(\mathbb{AG}(A)) = \{I - J \mid IJ = \langle 0\rangle$, where $I, J \in V(\mathbb{AG}(A))$ and $I \neq J\}$, where $\mathbb{I}(A)$ is the set of all ideals of A and $\mathbb{I}^*(A) = \mathbb{I}(A)\backslash\{\langle 0\rangle\}$. We verify that $\mathbb{AG}(A)$ is connected with $d_{max}(\mathbb{AG}(A)) \leq 3$. And we characterize some MV-algebras with $d_{max}(\mathbb{AG}(A)) = 0$ or 1, where $d_{max}(\mathbb{AG}(A))$ is the diameter of $\mathbb{AG}(A)$. If $| A | \leq 7$, we show that $\mathbb{AG}(A)$ is either a null graph, or $d_{max}(\mathbb{AG}(A)) = 1$. We restrict MV-algebras to Boolean algebras. The connections between $\mathbb{AG}(A)$ and $\Gamma(A)$ are studied, where $\Gamma(A)$ is the zero-divisor graph of A. We characterize the complete graph $\mathbb{AG}(A)$ and the star graph $\mathbb{AG}(A)$ by using $ann(A\backslash\{1\}) = \{a \in A \mid a \odot b = 0$ for all $b \in A\backslash\{1\}\}$, where $ann(A\backslash\{1\})$ is the annihilator of $A\backslash\{1\}$. Finally, we study the vertex coloring and girth of $\mathbb{AG}(A)$. We give two lower bounds and an upper bound for $\chi(\mathbb{AG}(A))$.

Keyword: The annihilating-ideal graph; MV-algebra; Boolean algebra; annihilator; ideal; k-chromatic; girth.

We are very grateful to the referees for their valuable suggestions for improving this paper. The research was supported by Natural Science Foundation of Shandong Province (Grant No. ZR2022MA043).

*Corresponding Author.

1 Introduction

The correspondences between the given objects and graph theory have been studied by introducing many different graphs on semigroups, rings, posets or MV-algebras. For a commutative ring R, the notion of the zero-divisor graph $\Gamma(R)$ was given in [7]. $\Gamma(R)$ is a simple graph whose vertices are nonzero zero-divisors of R and there is an edge $m - n$ if and only if $mn = 0$ for any $m, n \in V(\Gamma(R))$ with $m \neq n$. For more about it, we recommend [4]. The r-noncommuting graph Γ_R^r $(r \in R)$ of the finite ring R is also studied in [19], and $V(\Gamma_R^r) = R$. Then, the same definition is applied in a commutative semigroup S by DeMeyer et al. [8] which denoted by $\Gamma(S)$. For more about the zero-divisor graphs of semigroups, we recommend another paper [9]. Further, in 2020, Gan and Yang [1] introduced the zero-divisor graph $\Gamma(A)$ of an MV-algebra $(A, \oplus, *, 0)$. $\Gamma(A)$ is a simple graph with the vertex set $V(\Gamma(A)) = A\backslash\{0, 1\}$ by [1, Proposition 2] and $m, n \in V(\Gamma(A))$ are adjacent if and only if $m \odot n = 0$, where $m \neq n$. In [1], the authors deeply characterized an MV-algebra A with the diameter of $\Gamma(A)$ equal to 0, 1, 2 and 3. They studied all MV-algebras of cardinality up to 7 by using the zero-divisor graphs. In addition, they also introduced the annihilator graph $AG(A)$ for an MV-algebra A on [2]. By [2], $AG(A)$ is a simple graph with vertex set $V(AG(A)) = A\backslash\{0, 1\}$, and any two distinct vertices m, n are called adjacent if and only if $ann_A(m \odot n) \neq ann_A(m) \cup ann_A(n)$, where $ann_A(x) = \{a \in A \mid a \odot x = 0\}$ for any element x of A. Moreover, the graph $\Gamma_E(P)$ of equivalence classes of zero-divisors of a poset P is introduced by Liu [11]. The vertices of $\Gamma_E(P)$ are elements in $\overline{P}\backslash\{[0], [1]\}$, where $\overline{P} = \{[a] \mid a \in P\}$, rather than elements of P.

Further, let R be a commutative ring. In [15] and [16], the annihilating-ideal graph $\mathbb{AG}(R)$ is given whose vertices are ideals of R instead of elements. The vertex set of $\mathbb{AG}(R)$ is $\mathbb{A}^*(R) = \mathbb{A}(R)\backslash\{\langle 0\rangle\}$ and there exists an edge between I and J if and only if $IJ = \langle 0\rangle$, where $\mathbb{A}^*(R)$ is denoted to the set of all nonzero ideals with nonzero annihilators. Similarly, let S be a commutative semigroup. The the annihilating-ideal graph $\mathbb{AG}(S)$ is introduced by DeMeyer and Schneider in [14], whose vertex set consists of all nonzero annihilating-ideals of S and in which the vertex I is adjacent to the vertex J if and only if $IJ = \langle 0\rangle$ and $I \neq J$.

In this paper, since an MV-algebra can be regarded as a semigroup under the operations $\oplus$ or $\odot$, on the basis of [8, 9, 1, 14], we will study another type of graph of MV-algebras whose vertices are ideals rather than elements, which is called the annihilating-ideal graph. We define the vertex set $V(\mathbb{AG}(A))$ and the edge set $E(\mathbb{AG}(A))$ of $\mathbb{AG}(A)$ as follows:

$$V(\mathbb{AG}(A)) = \{I \in \mathbb{I}(A)\backslash\{\langle 0\rangle, A\} \mid \exists J \in \mathbb{I}^*(A) \text{ such that } IJ = \langle 0\rangle\},$$

$$E(\mathbb{AG}(A)) = \{\text{the edge } I - J \mid I, J \in V(\mathbb{AG}(A)) \text{ such that } I \neq J \text{ and } IJ = \langle 0 \rangle\},$$

where we use $\mathbb{I}(A)$ to denote the set of all ideals of A and $\mathbb{I}^*(A) = \mathbb{I}(A) \backslash \{\langle 0 \rangle\}$. For the MV-algebra A, we find that it can not guarantee that $IJ \in \mathbb{I}(A)$ for $I, J \in \mathbb{I}(A)$. However, this statement holds on Boolean algebras. The operations between ideals are more complex than those between elements. So we study the set $ann(I)$ and the principal ideals in this paper to overcome those difficulties.

The paper is structured as follows. In Section 2, we give some basic definitions and theorems about MV-algebras, semigroups and graph theory which will be used in the rest of this paper. In Section 3, for an MV-algebra A, we define the annihilating-ideal graph $\mathbb{AG}(A)$. The fact that $\mathbb{AG}(A)$ is a connected graph and $d_{max}(\mathbb{AG}(A)) \leq 3$ is verified. If $\mathbb{AG}(A)$ contains a cycle, we prove that $g(\mathbb{AG}(A)) \leq 4$. If A is an MV-algebra such that $\mid A \mid \leq 7$, then we show that $\mathbb{AG}(A)$ is either a null graph, or $d_{max}(\mathbb{AG}(A)) = 1$. In Section 4, we mainly study the annihilating-ideal graphs of MV-algebras which are star graphs. We show that $\mathbb{AG}(A)$ is a null graph if $ann(A \backslash \{1\}) = A \backslash \{1\}$. If there are two ideals I, J of A that satisfy $A = I \oplus J$ and $I \cap J = \{0\}$, where I is a 0-minimal ideal and there are no nonzero zero-divisors in J when $(J, \odot)$ is a semigroup, we prove that $\mathbb{AG}(A)$ is a star graph. In Section 5, we restrict MV-algebras to the Boolean algebras. We verify that $\langle x \rangle \in V(\mathbb{AG}(A))$ for all $x \in A \backslash \{0, 1\}$. The correspondences between $\Gamma(A)$ and $\mathbb{AG}(A)$ are studied, where $\Gamma(A)$ is the zero-divisor graph of A. We prove that $d_{max}(\mathbb{AG}(A)) = 3$ iff $\mid B \mid \geq 3$ or $\mid C \mid \geq 3$, where B, C are two MV-algebras satisfying $B \times C$ is isomorphic to A and $\mid B \mid, \mid C \mid \geq 2$. In addition, for any ideal I of A, we denote $I_A = A \backslash \{I \cup \{1\}\} \cup \{0\}$. If $\mathbb{AG}(A)$ is a star graph with I as its center and I_A is not an ideal of A, we verify that A has the unique 0-minimal ideal I. In Section 6, we study the vertex coloring and girth of $\mathbb{AG}(A)$ for an MV-algebra A. We get two lower bounds $\mid \mathbb{I}_{ann} \mid, \mid \mathbb{M}_0(A) \mid$ and an upper bound $2^{\chi(\Gamma(A))} - 1$ for $\chi(\mathbb{AG}(A))$, where $ann(A \backslash \{1\}) \subseteq A \backslash \{1\}$, $\mathbb{I}_{ann}$ is the set of all nonzero ideals of A which are contained in $ann(A \backslash \{1\})$, $\mathbb{M}_0(A)$ is the set of all 0-minimal ideals of A. Finally, we find that $g(\mathbb{AG}(A)) \in \{3, 4, \infty\}$. And we show that $g(\mathbb{AG}(A)) = 3$ if $\mid \mathbb{M}_0(A) \mid \geq 3$.

2 Preliminaries

In this section, for the convenience of readers, we give some basic definitions and theorems on MV-algebras, semigroups and graph theory which will be used in the following sections.

Definition 2.1. ([20, Definition 1.1.1]) An MV-algebra $(A, \oplus, *, 0)$ is an algebra such that $(A, \oplus, 0)$ is a commutative monoid, and for all $x, y \in A$ satisfying the followings:

(MV1) $x^{**} = x$;
(MV2) $x \oplus 0^* = 0^*$;
(MV3) $(x^* \oplus y)^* \oplus y = (y^* \oplus x)^* \oplus x$.

By Definition 2.1, let A be a set such that $0 \in A$. If $(A, \oplus, *)$ satisfies (MV1)-(MV3) for any $x, y \in A$, and the operation $\oplus$ satisfies associative-commutative, then A is an MV-algebra. Therefore, for any MV-algebra A, $(A, \oplus)$ can be regarded as a semigroup. The algebra $(A, \oplus, *, 0)$ is nontrivial if its inverse has more than one element. That is to say, an MV-algebra is nontrivial if and only if $0 \neq 1$. On each MV-algebra A, the constant 1 and the operation $\odot$ are defined as: $1 =_{def} 0^*$ and $x \odot y =_{def} (x^* \oplus y^*)^*$, which implies that $x \oplus y = (x^* \odot y^*)^*$. From [20], for each MV-algebra A and any $x \in A$, we have the following well-known properties:

• $(A, \odot, *, 1)$ is an MV-algebra. It follows that $(A, \odot, 1)$ is a commutative monoid. In addition, it is obvious that $(A, \odot)$ is also a semigroup.

• $1^* = 0$.

• $x \oplus 1 = 1$ and $x \oplus x^* = 1$. Thus, $x \odot x^* = 0$.

Let A and B be two MV-algebras. A function $H : A \longrightarrow B$ satisfying the conditions: (1) $H(0) = 0$; (2) $H(x \oplus y) = H(x) \oplus H(y)$ and (3) $H(x^*) = H^*(x)$ is called a homomorphism, where $x, y \in A$. A homomorphism H is an isomorphism if it is a surjective one-one homomorphism. We use $A \cong B$ to denote there is an isomorphism between two MV-algebras A and B by [20].

Let S_A be a subset of A with $0 \in S_A$. S_A is said a subalgebra of A if it is closed under the operations of A, and also equipped with the restriction to S_A of these operations. For all $x, y \in A$, if $x^* \oplus y = 1$, we denote $x \leq y$. It implies that " $\leq$ " is a partial order of A, which called the natural order of A. An MV-chain is an MV-algebra whose natural order is total.

For any $x, y \in A$, denote $x \ominus y = x \odot y^*$. And denote the lattice by $\mathbf{L}(A) = (A, \vee, \wedge, 0, 1)$, where the operations $\vee$ and $\wedge$ for any $x, y \in A$ are given as follows: $x \vee y = (x \odot y^*) \oplus y = (x \ominus y) \oplus y$, $x \wedge y = (x^* \vee y^*)^* = x \odot (x^* \oplus y)$. Let A be an MV-algebra. An element x of $\mathbf{L}(A)$ is complemented if there is $y \in \mathbf{L}(A)$ such that $x \vee y = 1$ and $x \wedge y = 0$. We use $\mathbf{B}(A)$ to denote the set of all complemented elements of $\mathbf{L}(A)$. Boolean elements of A are elements of $\mathbf{B}(A)$. In fact, Boolean algebras are precisely the MV-algebras satisfying the property: $x \oplus x = x$ by [20, Corollary 1.5.5]. We always use $\mathbf{B}_n$ to denote the n-element Boolean algebras.

Example 2.2. Consider the real unit interval $[0, 1]$. For any $m, n \in [0, 1]$, let $\oplus : m \oplus n = \min\{1, m + n\}$ and $* : m^* = 1 - m$. It is obvious that $([0, 1], \oplus, *, 0)$ is an MV-algebra and $m \odot n = \max\{0, m + n - 1\}$.

Obviously, for any $n \geq 2$, where n is an integer number, the subset $L_n = \{0, \frac{1}{n-1}, \frac{2}{n-1}, \cdots, \frac{n-2}{n-1}, 1\}$ is a subalgebra of $[0, 1]$. It is easy to get that L_2 is a

2-element MV-chain. In addition, from [1], we have that $L_2 \times L_2$ is isomorphic to the 4-element Boolean algebra $\mathbf{B}_4$.

Definition 2.3. ([20]) Let A be an MV-algebra. A subset I of A is called an ideal if it satisfies the following:

(i) $0 \in I$;

(ii) if $x \in I, y \in A$ and $y \le x$, then $y \in I$;

(iii) I is closed under the operation $\oplus$.

The set of all ideals of an MV-algebra A is denoted by $\mathbb{I}(A)$ and $\mathbb{I}^*(A) = \mathbb{I}(A)\backslash\{\langle 0\rangle\}$, where $\langle 0\rangle = \{0\}$ is the zero ideal of A. And we denote $IJ = \{a \odot b \mid a \in I, b \in J\}$ for any ideals I, J of A. For any subset W of A, we denote $\langle W\rangle = \{\bigcap I \mid W \subseteq I \text{ and } I \in \mathbb{I}(A)\}$, and we call that $\langle W\rangle$ is the ideal generated by W. Particularly, for any element $w \in A$, the principal ideal generated by w is denoted by $\langle w\rangle = \langle\{w\}\rangle = \{x \in A \mid x \le nw \text{ for some integer number } n \ge 0\}$, where $nw = w \oplus w \oplus \cdots \oplus w$ (n times). We denote $\langle I \cup \{w\}\rangle = \{x \in A \mid x \le nw \oplus a$ for some $n \in \mathbf{N}$ and $a \in I\}$, where I is an ideal of A. For an ideal I of the MV-algebra A, if $I \neq A$, then I is said proper. An MV-algebra A is called simple if and only if it has exactly two ideals. That is to say, $\mathbb{I}(A) = \{\langle 0\rangle, A\}$. Minimal ideals are ideals that have zero intersection with all ideals that do not contain them ([18]). An ideal I of A is called a maximal ideal if I is proper and $I \subseteq J$ if only if $J = A$. The set $Rad(A)$ is called the radical of A, which is the intersection of all maximal ideals of A.

Lemma 2.4. ([20, Lemma 1.2.1]) *Let A be an MV-algebra and $W \subseteq A$. If $W = \emptyset$, we denote $\langle W\rangle = \{0\}$. If $W \neq \emptyset$, then*

$$\langle W\rangle = \{w \in A \mid w \le w_1 \oplus w_2 \oplus \cdots \oplus w_n \text{ for some } w_1, ..., w_n \in W\}.$$

We also have the following conclusions for MV-algebras.

Lemma 2.5. ([20, Lemma 1.1.4] and [1, Lemma 1]) *Let A be an MV-algebra. Then the following conditions are equivalent for all elements x, y of A:*

(1) $x \le y$;

(2) $x^* \oplus y = 1$;

(3) $x \odot y^* = 0$;

(4) $x \vee y = y$;

(5) $y^* \le x^*$;

(6) *for any $z \in A$, $x \oplus z \le y \oplus z$ and $x \odot z \le y \odot z$.*

Theorem 2.6. ([20, Theorem 1.5.3]) *Let A be an MV-algebra. For any $x \in A$ the following are equivalent:*

(1) $x \in \mathbf{B}(A)$;
(2) $x \vee x^* = 1$;
(3) $x \wedge x^* = 0$;
(4) $x \oplus x = x$;
(5) $x \odot x = x$;
(6) $x \oplus y = x \vee y$ *for any element* $y \in A$;
(7) $x \odot y = x \wedge y$ *for any element* $y \in A$.

By Theorem 2.6, it is easy to get that $x \in \mathbf{B}(A)$ if and only if $x^* \in \mathbf{B}(A)$, which follows that $x^* \odot x^* = x^*$ and $x^* \oplus x^* = x^*$.

Lemma 2.7. ([20] and [5]) *Let A be an MV-algebra. Then for any $x, y, z \in A$, the following conditions hold in A:*
(1) $x \odot y \leq x \wedge y \leq x$;
(2) $y \leq x \vee y \leq x \oplus y$;
(3) *distributivity* : $x \oplus (y \wedge z) = (x \oplus y) \wedge (x \oplus z)$, $x \odot (y \vee z) = (x \odot y) \vee (x \odot z)$;
(4) $x \oplus y = y$ *iff* $x \wedge y^* = 0$;
(5) *if* $x \odot y = x \odot z$ *and* $x \oplus y = x \oplus z$, *then* $y = z$.

Lemma 2.8. ([20, Corollary 3.5.4]) *Let A be an MV-algebra. Then A is the simple MV-algebra and $\mid A \mid$ is finite if and only if there is an isomorphism between A and L_n for some integer number $n \geq 2$.*

Let S be a semigroup. An ideal T of S is a nonempty subset such that $ST \subseteq T$ and $TS \subseteq T$. Any nonzero ideal of S can be equal to a union of principal ideals of S by [14]. Let I be a proper ideal. I is called an annihilating-ideal of S if there is an ideal $J \neq \langle 0 \rangle$ of S such that $IJ = \langle 0 \rangle$. A subset T of a semigroup S is called subsemigroup if it is closed with respect to multiplication and $T \neq \emptyset$. For an element s of S, if there exists an element $t \in S$ such that $st = 0$ and $t \neq 0$, then s is called a zero-divisor by [10, Definition 1.3.9]. In [17], a partially ordered set X is called to satisfy the ascending chain condition (ACC) if for every ascending sequence $a_1 \leq a_2 \leq a_3 \leq \cdots$, there exists a positive integer number n satisfying $a_n = a_{n+1} = a_{n+2} = \cdots$, where $a_i \in X$ for any $i \in \{1, 2, 3, ...\}$. Similarly, if for any descending sequence $a_1 \geq a_2 \geq a_3 \geq \cdots$, there exists n such that $a_n = a_{n+1} = a_{n+2} = \cdots$, then X is said to satisfy the descending chain condition (DCC).

For any subset $T \subseteq S$, the annihilator of T is $ann(T) = \{s \in S \mid st = 0$ for all $t \in T\}$, where S is a semigroup. If an ideal I of S is minimal within the set $\mathbb{I}^*(S) = \{I \mid I$ is an nonzero ideal of $S\}$, we call that the 0-minimal ideal. In addition, recall that a family $\mathcal{F} = \{T_i : i \in \Lambda\}$ of subsets of a set X is called a partition of X if it satisfies the following conditions:

(1) $T_i \neq \emptyset$ for all $i \in \Lambda$;
(2) either $T_i = T_j$ or $T_i \cap T_j = \emptyset$ for all $i, j \in \Lambda$;
(3) $\bigcup\{T_i : i \in \Lambda\} = X$. ([12])

Call that sets S and T have the same cardinality (or cardinal number) and write $| S |=| T |$ if there is a bijection from S to T.

Next, we give some notions about graphs.

By [21], a graph G consists of $V(G)$ and $E(G)$, where $V(G)$ is the vertex set of G and $E(G)$ is a finite family of unordered pairs of vertices of G which are called edges. An edge $\{v_1, v_2\}$ is called to join the vertices v_1 and v_2. The two vertices v_1, v_2 are called the ends of this edge. Two vertices $v_1, v_2 \in V(G)$ are adjacent if they are joined by an edge, and the vertices v_1 and v_2 are incident with such an edge. Two vertices may have several edges joining them, such edges are called multiple edges. A loop is an edge with identical ends. Each loop is regarded as two edges. A walk is a way of getting from one vertex to another and consists of a sequence of edges, one following after another. If except for the beginning and end vertices which coincide, other vertices appear once, then the walk is called a path. A cycle is a walk in which no vertex appears more than once. More associated definitions are as follows:

• A graph G is connected if any two vertices of G are connected by a path.

• A simple graph is a graph with no loops and multiple edges.

• If any two distinct vertices are joined by an edge in the graph G, then G is called a complete graph. We use K_n to denote the complete graph with n-vertex.

• Let G be a graph. If $E(G)$ is empty, then G is said an empty graph.

• The graph with no vertices is called a null graph.

• We use $deg(x)$ to denote the number of edges incident with a vertex x in the graph G, which is called the degree of x.

• The distance between any two vertices v_1, v_2 in a graph is the length of the shortest path from v_1 to v_2 and denoted by $d(v_1, v_2)$. The diameter of a graph G is the maximum of the distance between any two vertices. We denote the diameter of a graph G by $d_{max}(G) = \max_{v_1, v_2 \in V(G)} d(v_1, v_2)$.

• If there is one vertex in a graph G which is adjacent to every other vertex and no other edges. Then G is a star graph, and this vertex is said the center of the graph G.

• The length of the shortest cycle in a graph G is called the girth of G, and denoted by $g(G)$. A cycle graph is a graph in which $deg(x) = 2$ for each vertex x. Particularly, the cycle graph on 1-vertex is a loop. If a graph G contains no cycles, we denote $g(G) = \infty$. For more details about applications of girth, we recommend [6].

If $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, then the graph H is said a subgraph of the graph G. The two graphs G_1 and G_2 are isomorphic if there is a bijection h from G_1

to G_2 of vertices satisfies that v_1 is adjacent to v_2 in G_1 if and only if $h(v_1), h(v_2)$ are adjacent in G_2 for all $v_1, v_2 \in V(G_1)$, and write it by $G_1 \cong G_2$. By [13], a clique of a graph G is a complete subgraph of G . That is, an n-vertex clique of a graph G is a subgraph of G which is isomorphic to K_n. A maximum clique of a graph G is a clique such that there is no clique with more vertices. We use $\omega(G)$ to denote the clique number of a graph G which is the size of the maximum clique in G.

Let G be a graph which contains no loops. If every vertex in G can be assigned one of k colors and any two adjacent vertices have different colors, then we call that G is k-colorable. The chromatic number $\chi(G)$ of a graph G is the smallest integer number k such that G is k-colorable, and write $\chi(G) = k$. In general, we say a vertex coloring is proper if there is no same color between any two adjacent vertices. For any graph G, it must be that $\omega(G) \leq \chi(G)$. For more details about graph theory, we recommend [21, 13, 3].

Gan and Yang [1] studied the zero-divisor graph $\Gamma(A)$ for an MV-algebra $(A, \oplus, *, 0)$. $\Gamma(A)$ is a simple graph whose vertex set is $V(\Gamma(A)) = \{m \in A \mid$ there is $n \in A\backslash\{0\}$ satisfying $m \odot n = 0\}$ and the edge set $E(\Gamma(A)) = \{$the edge $m - n \mid m \odot n = 0$, where $m, n \in V(\Gamma(A))$ such that $m \neq n\}$. Further, they verified that $\Gamma(A)$ is a connected graph and $d_{max}(\Gamma(A)) \leq 3$ by [1, Theorem 1] and $V(\Gamma(A)) = A\backslash\{0, 1\}$ by [1, Proposition 2]. For more details about $\Gamma(A)$, we recommend [1].

Proposition 2.9. ([1, Lemma 4]) *Let A be an MV-algebra and $\mathcal{C}_*(A) = \{a \in A \mid a = a^*\}$. Then $\mid \mathcal{C}_*(A) \mid \leq 1$.*

Lemma 2.10. ([1, Theorem 3]) *Let A be an MV-algebra and $\mid A \mid = 4$. Then $A \cong L_4$ or $A \cong \mathbf{B}_4$.*

DeMeyer and Schneider [14] introduced the annihilating-ideal graph $\mathbb{AG}(S)$ for the commutative semigroup S. $\mathbb{AG}(S)$ is a simple graph whose vertices are nonzero annihilating-ideals of S and in which any two distinct vertices I, J are adjacent if and only if $IJ = \langle 0 \rangle$, where $\langle 0 \rangle$ is the zero ideal of S. We have that $\mathbb{AG}(S)$ is also a connected graph with $d_{max}(\mathbb{AG}(S)) \leq 3$ by [14, Theorem 8]. The structure of annihilating-ideal graphs of commutative semigroups and the vertex coloring of them are deeply studied by [14]. On the basis of [14], the connectivity, diameter, girth and so on of $\mathbb{AG}(A)$ for the MV-algebra A are described in this paper.

3 The Annihilating-ideal Graphs of MV-algebras

In this section, we study another type of graph of MV-algebras which is called the annihilating-ideal graph $\mathbb{AG}(A)$ of an MV-algebra $(A, \oplus, *, 0)$. We characterize the algebra $(A, \oplus, *, 0)$ which $d_{max}(\mathbb{AG}(A))$ equal to 0 and 1. We also study the structure of $\mathbb{AG}(A)$, where the cardinality of A up to 7.

In [1], the zero-divisor graph of MV-algebras whose vertices are elements is introduced. By [9], the vertices of the zero-divisor graph of commutative semigroups are also elements. Further, the annihilating-ideal graph of commutative semigroups is given by [14], whose vertices are ideals. Note that an MV-algebra can be regarded as a semigroup under the operations $\oplus$ or $\odot$. We shall study the new type of graph of MV-algebras whose vertices are ideals.

Definition 3.1. Let A be an MV-algebra. The annihilating-ideal graph $\mathbb{AG}(A)$ of A is a simple graph with the vertex set $V(\mathbb{AG}(A)) = \{I \in \mathbb{I}^*(A) \setminus \{A\} \mid \exists J \in \mathbb{I}^*(A)$ such that $IJ = \langle 0\rangle\}$ and the edge set $E(\mathbb{AG}(A)) =\{$the edge $I - J \mid I, J \in V(\mathbb{AG}(A))$ satisfying $I \neq J$ and $IJ = \langle 0\rangle\}$.

Let $(A, \oplus, *, 0)$ be an MV-algebra and I, J be two ideals of A. Obviously, $I \cap J$ is also an ideal of A. And if one of I and J is proper, we have that $I \cap J$ is also a proper ideal of A. In addition, we claim that $IJ \subseteq I \cap J \subseteq I, J$ for ideals I, J of A. In fact, for any $x \in IJ$, there are $a \in I, b \in J$ such that $x = a \odot b \leq a, b$ by Lemma 2.7. Thus, $x \in I$ and $x \in J$.

By [15, Theorem 2.1], for a commutative ring R, $\mathbb{AG}(R)$ is a connected graph and $d_{max}(\mathbb{AG}(R)) \leq 3$. Moreover, if there is a cycle in $\mathbb{AG}(R)$, then $g(\mathbb{AG}(R)) \leq 4$. Similarly, we have the following for the annihilating-ideal graph of an MV-algebra.

Theorem 3.2. *Let A be an MV-algebra. Then the annihilating-ideal graph $\mathbb{AG}(A)$ of A is a connected graph and $d_{max}(\mathbb{AG}(A)) \leq 3$. And if there exists a cycle in $\mathbb{AG}(A)$, then $g(\mathbb{AG}(A)) \leq 4$.*

Proof. (1) Let I, J be any two distinct vertices of $\mathbb{AG}(A)$. If $IJ = \langle 0\rangle$, there is nothing to prove. Assume that $IJ \neq \langle 0\rangle$. Consider the following conditions:

• Suppose that both I^2 and J^2 are zero ideals.

Firstly, we claim that $I \cap J \in V(\mathbb{AG}(A))$. In fact, if $I \cap J = \langle 0\rangle$, it follows from $IJ \subseteq I \cap J = \langle 0\rangle$ that $IJ = \langle 0\rangle$, which is a contradiction. So $I \cap J \neq \langle 0\rangle$. We have that $I \cap J \neq I$. Otherwise, $IJ \subseteq J^2 = \langle 0\rangle$, which is also a contradiction. Similarly, it must be $I \cap J \neq J$. Thus, $I \cap J \neq \langle 0\rangle, I, J$. Since $I(I \cap J) \subseteq I^2 = \langle 0\rangle$ and $J(I \cap J) \subseteq J^2 = \langle 0\rangle$, then there exists a path $I - I \cap J - J$ between I and J.

• Suppose that one of I^2, J^2 is not equal to $\langle 0\rangle$.

Without loss of generality, suppose that $I^2 \neq \langle 0\rangle$ and $J^2 = \langle 0\rangle$. Since $I \in V(\mathbb{AG}(A))$, there must be a vertex L of $\mathbb{AG}(A)$ that satisfies $IL = \langle 0\rangle$. It must be $J \neq L$ from $IJ \neq \langle 0\rangle$. Next, we consider the proper ideal $J \cap L$.

(i) If $J \cap L = \langle 0\rangle$, we have that $JL = \langle 0\rangle$ from $JL \subseteq J \cap L = \langle 0\rangle$. So there is a path $I - L - J$ between I and J.

Then, assume that $J \cap L \neq \langle 0 \rangle$.

(ii) If $J \cap L = J$, it follows that $IJ \subseteq IL = \langle 0 \rangle$, which contradicts $IJ \neq \langle 0 \rangle$.

(iii) If $J \cap L = L$, we have $JL \subseteq J^2 = \langle 0 \rangle$, which implies that J is adjacent to L in $\mathbb{AG}(A)$. Hence, there is a path $I - L - J$ between I and J.

(iv) If $J \cap L = I$, we get that $I^2 = I(J \cap L) \subseteq IL = \langle 0 \rangle$, a contradiction to $I^2 \neq \langle 0 \rangle$.

(v) Assume that $J \cap L \neq L$. We have that $I(J \cap L) \subseteq IL = \langle 0 \rangle$ and $J(J \cap L) \subseteq J^2 = \langle 0 \rangle$. So there is a path $I - J \cap L - J$ between I and J.

• Suppose that $I^2, J^2 \neq \langle 0 \rangle$.

There are ideals $L, P \in V(\mathbb{AG}(A))$ such that $IL = JP = \langle 0 \rangle$ by $I, J \in V(\mathbb{AG}(A))$. Then we consider the proper ideal $L \cap P$.

(i) If $L \cap P = \langle 0 \rangle$, then there is a path $I - L - P - J$ between I and J from $LP \subseteq L \cap P = \langle 0 \rangle$.

(ii) If $L \cap P = I$, it follows that $IJ = J(L \cap P) \subseteq JP = \langle 0 \rangle$, a contradiction. Similarly, we have that $L \cap P \neq J$.

(iii) If $L \cap P = L$, then $JL \subseteq IP = \langle 0 \rangle$. So there is a path $I - L - J$ between I and J. Similarly, if $L \cap P = P$, there exists a path $I - P - J$ between I and J.

(iv) Assume that $L \cap P \neq \langle 0 \rangle$. Since $I(L \cap P) \subseteq IL = \langle 0 \rangle$, we get that $L \cap P \in V(\mathbb{AG}(A))$. In addition, it must be that $J(L \cap P) \subseteq JP = \langle 0 \rangle$, which implies that there exists a path $I - L \cap P - J$ between I and J.

Therefore, above cases imply that $\mathbb{AG}(A)$ is connected and $d_{max}(\mathbb{AG}(A)) \leq 3$.

(2) Suppose that $\mathbb{AG}(A)$ contains cycles and $g(\mathbb{AG}(A)) = n$. Let $I_1 - I_2 - \cdots - I_n - I_1$ be a cycle of $\mathbb{AG}(A)$, where $I_i \in V(\mathbb{AG}(A))$ and $I_i \neq I_j$ for any $i, j \in \{1, 2, ..., n\}$, $i \neq j$. If $n \leq 4$, there is nothing to prove. Assume that $n \geq 5$. It is enough to show that $I_1 \cap I_4 \neq \langle 0 \rangle$. In fact, if $I_1 \cap I_4 = \langle 0 \rangle$, we have that $I_1 I_4 \subseteq I_1 \cap I_4 = \langle 0 \rangle$. Thus, there is a cycle $I_1 - I_2 - I_3 - I_4 - I_1$ with length 4 in $\mathbb{AG}(A)$, which is a contradiction by $n \geq 5$. Then we consider the following cases:

• $I_1 \cap I_4 \neq I_1, I_4$. If $I_1 \cap I_4 = I_1$, we have that $I_1 I_3 \subseteq I_4 I_3 = \langle 0 \rangle$. It follows that there is a cycle $I_1 - I_2 - I_3 - I_1$, which is a contradiction. So $I_1 \cap I_4 \neq I_1$. Similarly, we can obtain that $I_1 \cap I_4 \neq I_4$. Otherwise, there will be a cycle $I_2 - I_3 - I_4 - I_2$, a contradiction.

• $I_1 \cap I_4 \neq I_2, I_3$. If $I_1 \cap I_4 = I_2$, then we have $I_2 I_n \subseteq I_1 I_n = \langle 0 \rangle$. It implies that there is a cycle $I_2 - I_3 - \cdots - I_{n-1} - I_n - I_2$ which with length $n-1$, which is a contradiction. Similarly, it must be $I_1 \cap I_4 \neq I_3$, for otherwise, it follows from $I_3 \subseteq I_1$ that there exists a cycle $I_3 - I_4 - \cdots - I_{n-1} - I_n - I_3$ with length $n-2$, a contradiction as well.

Hence, $I_1 \cap I_4 \neq I_1, I_2, I_3, I_4$. We have that $I_2(I_1 \cap I_4) \subseteq I_2 I_1 = \langle 0 \rangle$ and $I_3(I_1 \cap I_4) \subseteq I_3 I_4 = \langle 0 \rangle$. Then, there is a cycle $I_2 - I_1 \cap I_4 - I_3 - I_2$, again a contradiction.

By summarizing the above, $g(\mathbb{AG}(A)) \leq 4$. $\square$

For each Artinian ring R, by [15, Proposition 1.3], we have that any nonzero proper ideal corresponds to a vertex in $\mathbb{AG}(R)$. That is to say, $\mid V(\mathbb{AG}(R)) \mid = \mid \mathbb{I}(R)\backslash\{\langle 0\rangle, R\} \mid$. Moreover, for any MV-algebra A, from [1, Proposition 2] we see that $\mid V(\Gamma(A)) \mid = \mid A\backslash\{0,1\} \mid$.

Example 3.3. Let L_n be the subalgebra in Example 2.2. From [20] we see that L_n is a simple MV-chain of rank n, where $n \geq 2$. It is easy to get that $\mid V(\Gamma(L_n)) \mid = \mid L_n\backslash\{0,1\} \mid = n-2$. However, since L_n is simple, L_n has only one proper ideal $I = \langle 0\rangle$ for any $n \geq 2$. So we have that $\mid V(\mathbb{AG}(L_n)) \mid = 0$ and then $\mid E(\mathbb{AG}(L_n)) \mid = 0$. That is to say, $\mathbb{AG}(L_n)$ is a null graph for all integer number $n \geq 2$.

Let A be an MV-algebra such that $A \cong L_n$, where $n \geq 2$ and $\mid A \mid$ is finite. Then it follows from Lemma 2.8 that the annihilating-ideal graph of A is a null graph.

By [1], if A is an MV-algebra satisfying $\mid A \mid \leq 7$, then A is isomorphic to the MV-algebra L_n, the direct product $L_n \times L_m$ or the 4-element Boolean algebra $\mathbf{B}_4$. And the structure of MV-algebras with cardinality greater than 7 is difficult to describe. So let A be an MV-algebra such that $3 \leq \mid A \mid \leq 7$. Next, we will study the structure of the annihilating-ideal graph of A.

Remark 3.4. Let A be an MV-algebra with $\mid A \mid = 3, 5$ or 7. Then, we have that $V(\mathbb{AG}(A)) = \emptyset$. Suppose that $\mid A \mid = 3, 5$ or 7. From [1, Theorem 2, Theorem 6, Theorem 8] we get that $A \cong L_3, A \cong L_5$ or $A \cong L_7$. It follows that $\mathbb{AG}(A)$ is a null graph by Example 3.3.

Remark 3.5. For any MV-algebra A, from [1, Theorem 2] we know that $\mid A \mid = 3$ if and only if $\Gamma(A)$ is an empty graph. However, we claim that the converse of Remark 3.4 does not necessarily hold.

Consider the MV-algebra $M_1 = \{0, m, n, 1\}$ and the operations $*$, $\oplus$ and $\odot$ are:

$*$	0	m	n	1
	1	n	m	0

$\oplus$	0	m	n	1
0	0	m	n	1
m	m	n	1	1
n	n	1	1	1
1	1	1	1	1

$\odot$	0	m	n	1
0	0	0	0	0
m	0	0	0	m
n	0	0	m	n
1	0	m	n	1

Thus, by Lemma 2.7 we will obtain that $m \leq n$ from $m \leq m \oplus m = n$, which implies that M_1 is an MV-chain and $M_1 \cong L_4$. It is easy to get that $\mathbb{AG}(M_1)$ is a null graph by Example 3.3. However, $\mid M_1 \mid = 4 \neq 3, 5$ or 7.

In the following, we will characterize $\mathbb{AG}(A)$ which with $\mid A \mid = 4$ or 6.

By Lemma 2.10, we see that $A \cong L_4$ or $A \cong \mathbf{B}_4$ for any 4-element MV-algebra A. Particularly, let $A = \{0, a, b, 1\}$ be an MV-algebra such that $A \cong \mathbf{B}_4$. From simple operations we have that $V(\mathbb{AG}(A)) = \{\{0, a\}, \{0, b\}\}$ and $\{0, a\} \cdot \{0, b\} = \langle 0 \rangle$. It follows that $\mathbb{AG}(A) \cong K_2$. Thus, we have the following.

Theorem 3.6. *Let A be an MV-algebra and $\mid A \mid = 4$. Then either $\mathbb{AG}(A)$ is a null graph, or $\mathbb{AG}(A) \cong K_2$ with $d_{max}(\mathbb{AG}(A)) = 1$.*

Proof. Let A be an MV-algebra with $\mid A \mid = 4$. By Lemma 2.10, we have $A \cong L_4$ or $A \cong \mathbf{B}_4$. If $A \cong L_4$, then $\mathbb{AG}(A)$ is a null graph by Example 3.3. If $A \cong \mathbf{B}_4$, it implies from the above discussion that $\mathbb{AG}(A) \cong K_2$ and then $d_{max}(\mathbb{AG}(A)) = 1$. □

Proposition 3.7. *Let A be an MV-algebra such that $A \cong L_n \times L_m$, where $n, m \geq 2$. Then $\mathbb{AG}(A) \cong K_2$ with $d_{max}(\mathbb{AG}(A)) = 1$.*

Proof. Since $A \cong L_n \times L_m$, it must be that $\mathbb{AG}(A) \cong \mathbb{AG}(L_n \times L_m)$. Without loss of generality, suppose that $n \leq m$. It is obvious that $L_n \times L_m$ has only two nonzero proper ideals $I_1 = \{0\} \times L_m$, $I_2 = L_n \times \{0\}$ and $I_1 I_2 = \langle 0 \rangle$. Thus, we have that $V(\mathbb{AG}(L_n \times L_m)) = \{I_1, I_2\}$. It follows that $K_2 \cong \mathbb{AG}(L_n \times L_m) \cong \mathbb{AG}(A)$. So we have $d_{max}(\mathbb{AG}(A)) = 1$. □

Example 3.8. Let $M_2 = \{0, x, y, z, w, 1\}$ be the 6-element MV-algebra in [1, Example 3] which is defined as follows:

$*$	0	x	y	z	w	1
	1	w	z	y	x	0

$\oplus$	0	x	y	z	w	1
0	0	x	y	z	w	1
x	x	z	w	z	1	1
y	y	w	y	1	w	1
z	z	z	1	z	1	1
w	w	1	w	1	1	1
1	1	1	1	1	1	1

$\odot$	0	x	y	z	w	1
0	0	0	0	0	0	0
x	0	0	0	x	0	x
y	0	0	y	0	y	y
z	0	x	0	z	x	z
w	0	0	y	x	y	w
1	0	x	y	z	w	1

On the one hand, it is obvious that $y \leq w, x \leq z$ and $x \leq w$ from $z \oplus w = 1$ and $w \oplus w = 1$. Then it is easy to get that $V(\mathbb{AG}(M_2)) = \{\{0, y\}, \{0, x, z\}\}$ and $\{0, y\} \cdot \{0, x, z\} = \langle 0 \rangle$. For convenience, we denote $I_1 = \{0, y\}$ and $I_2 = \{0, x, z\}$. Thus, by the simple operation, the graphs $\mathbb{AG}(M_2)$ and $\Gamma(M_2)$ of M_2 are respectively:

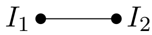


Figure 1. $\mathbb{AG}(M_2)$

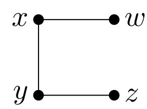


Figure 2. $\Gamma(M_2)$

We directly obtain that $d_{max}(\mathbb{AG}(M_2)) = 1$.

On the other hand, from [1, Example 3] we get that $M_2 \cong L_2 \times L_3$. Hence, from Proposition 3.7 we have that $\mathbb{AG}(M_2) \cong K_2$ and $d_{max}(\mathbb{AG}(M_2)) = 1$.

Theorem 3.9. *Let A be an MV-algebra such that $\mid A \mid= 6$. Then either $V(\mathbb{AG}(A)) = \emptyset$, or $d_{max}(\mathbb{AG}(A)) = 1$.*

Proof. Let A be an MV-algebra such that $\mid A \mid= 6$. By [1, Theorem 7], we see that $A \cong L_6$ or $A \cong L_2 \times L_3$. If $A \cong L_6$, then $\mathbb{AG}(A)$ is a null graph by Example 3.3. If $A \cong L_2 \times L_3$. It follows from Proposition 3.7 that $d_{max}(\mathbb{AG}(A)) = 1$. $\square$

4 Star Graphs

In this section, for the MV-algebra $(A, \oplus, *, 0)$, we mainly use the annihilator of $A \backslash \{1\}$ to investigate the annihilating-ideal graph $\mathbb{AG}(A)$. If there is a 0-minimal ideal I of A and an ideal J such that $A = I \oplus J$ and $I \cap J = \{0\}$, then $\mathbb{AG}(A)$ is a star graph, where J contains no nonzero zero-divisors when $(J, \odot)$ is a semigroup.

Remark 4.1. We denote $ann(A \backslash \{1\}) = \{a \in A \mid a \odot b = 0 \text{ for all } b \in A \backslash \{1\}\}$, where $ann(A \backslash \{1\})$ is called the annihilator of $A \backslash \{1\}$. It is easy to get that $ann(A \backslash \{1\}) \subseteq A$. Then we consider the following conditions:

(1) It is possible that $ann(A \backslash \{1\}) = \langle 0 \rangle$. Let consider the MV-algebra M_2 in Example 3.8. It is obvious that $ann(M_2 \backslash \{1\}) = \langle 0 \rangle$ from Example 3.8. In particular, suppose that $ann(A \backslash \{1\}) = A$. That is, $1 \in ann(A \backslash \{1\}) = \{a \in A \mid a \odot b = 0$ for all $b \in A \backslash \{1\}\}$, which implies that $A = \{0, 1\}$ by $1 \odot x = x$ for all $x \in A$.

(2) We claim that $ann(A \backslash \{1\}) = A \backslash \{1\}$ is possible. Consider the 3-element MV-algebra $M_3 = \{0, a, 1\}$ in which $a = a^*$ and $a \oplus a = 1$, which implies that $a \odot a = 0$. Then, $\mathbb{I}(M_3) = \{\langle 0 \rangle, M_3\}$ and $ann(M_3 \backslash \{1\}) = \{x \in M_3 \mid x \odot y = 0 \ (\forall y \in \{0, a\})\} = \{0, a\} = M_3 \backslash \{1\}$.

(3) For any commutative zero-divisor semigroup S, it must be that $ann(S)$ is an ideal of S, either $ann(S) = S, ann(S) = \langle 0 \rangle$ or $ann(S) \in \mathbb{A}^*(S)$ from [14]. However, for an MV-algebra A, $ann(A\backslash\{1\})$ is not necessarily an ideal of A. In fact, although it can guarantee that $y \in ann(A\backslash\{1\})$ for any $y \leq x$ and $x \in ann(A\backslash\{1\})$, $a \oplus b$ does not necessarily belong to $ann(A\backslash\{1\})$ for $a, b \in ann(A\backslash\{1\})$. For example, we consider the MV-algebra M_3 in Remark 4.1 (2). Since $a \oplus a = 1 \notin ann(M_3\backslash\{1\})$, then $ann(M_3\backslash\{1\}) = \{0, a\}$ is not an ideal of M_3 .

Next, we will use $ann(A\backslash\{1\})$ to further characterize the structure of the annihilating-ideal graph of an MV-algebra A.

Theorem 4.2. *Let A be an MV-algebra such that $ann(A\backslash\{1\}) = A\backslash\{1\}$. Then, $A \cong L_3$ and $\mathbb{AG}(A)$ is a null graph.*

Proof. Suppose that $ann(A\backslash\{1\}) = A\backslash\{1\}$. For any two nonzero elements $a, b \in A\backslash\{1\}$, we have that $a \odot a = a^* \odot a^* = b \odot b = b^* \odot b^* = a \odot b = a^* \odot b^* = 0$. It implies that $a \oplus a = a \oplus b = 1$ and $a \odot a = a \odot b = 0$, so $a = b$. Thus, $\mid A \mid = 3$. By Remark 3.4, we have that $V(\mathbb{AG}(A)) = \emptyset$. □

Remark 4.3. Consider the 4-element Boolean algebra $\mathbf{B}_4 = \{0, a, b, 1\}$ or the algebra M_2 in Example 3.8. From Theorem 3.6 and Figure 1 we see that $\mathbb{AG}(\mathbf{B}_4) \cong K_2 \cong \mathbb{AG}(M_2)$ and $d_{max}(\mathbb{AG}(\mathbf{B}_4)) = d_{max}(\mathbb{AG}(M_2)) = 1 \leq 2$. However, $ann(\mathbf{B}_4\backslash\{1\}) = ann(M_2\backslash\{1\}) = \langle 0 \rangle \notin V(\mathbb{AG}(A))$.

Proposition 4.4. *Let A be an MV-algebra. If there are a 0-minimal ideal I and an ideal J such that $I \cap J = \{0\}$ and $K \subseteq J$ for any $I \neq K \in V(\mathbb{AG}(A))$, then $\mathbb{AG}(A)$ has a vertex which is adjacent to other vertices.*

Proof. Obviously, J is proper. Otherwise, we have that $I \cap J = I$, contradicting the 0-minimality of I. Since $IJ \subseteq I \cap J = \langle 0 \rangle$, I is adjacent to J in $\mathbb{AG}(A)$. For any $K \in V(\mathbb{AG}(A))$ with $K \neq I$, since $K \subseteq J$, we get that $IK \subseteq IJ = \langle 0 \rangle$. Then, I is adjacent to K in $\mathbb{AG}(A)$ from $IK = \langle 0 \rangle$. Thus, there is a vertex I of $\mathbb{AG}(A)$ such that $IJ = \langle 0 \rangle$ for all $J \in V(\mathbb{AG}(A))$ and $I \neq J$. □

From Proposition 4.4, the vertex which is adjacent to any other vertex is exactly a 0-minimal ideal of A. Now, we will further characterize those MV-algebras whose annihilating-ideal graphs are the star graphs.

Theorem 4.5. *Let A be an MV-algebra. If there are two ideals I, J such that $A = I \oplus J$ and $I \cap J = \{0\}$, then $\mathbb{AG}(A)$ is a star graph, where I is a 0-minimal ideal of A and J contains no nonzero zero-divisors when $(J, \odot)$ is considered as a semigroup.*

Proof. Suppose that $A = I \oplus J$ and $I \cap J = \{0\}$. It is obvious that $I, J \in \mathbb{I}^*(A)\backslash\{A\}$ and $IJ \subseteq I \cap J = \{0\}$. So we have that I is adjacent to J in $\mathbb{AG}(A)$. Let $P \in V(\mathbb{AG}(A))$ such that $P \neq I$. Then, $P \subsetneqq A = I \oplus J = \{a \oplus b \mid a \in I, b \in J\}$. That is, for any element $p \in P$, there are $p_1 \in I, p_2 \in J$ such that $p = p_1 \oplus p_2$. So let $P = K \oplus H$, where K is a subset of I and $H \subseteq J$. Next, consider the following cases:

• If $P = K \oplus \langle 0\rangle$ for some $\langle 0\rangle \neq K \subseteq I$, which is a contradiction by the 0-minimality of I.

• Suppose that $P = \langle 0\rangle \oplus H$ for some $\langle 0\rangle \neq H \subseteq J$. Then, it follows from $IP \subseteq IJ = \langle 0\rangle$ that I is adjacent to P in $\mathbb{AG}(A)$. Let $Q \in V(\mathbb{AG}(A))$ such that $Q \neq P, I$. It must be that Q contains some nonzero elements of $I \oplus J$. Thus, we can claim that $QP \neq \langle 0\rangle$. In fact, if $QP = \langle 0\rangle$. Then, there is a nonzero element $a \oplus b \in (I \oplus J) \cap Q$ $(a \in I, 0 \neq b \in J)$ such that $b \odot p \leq (a \oplus b) \odot p = 0$ for all $p \in P$ from Lemma 2.5 and Lemma 2.7. Since $P \subseteq J$, it must be that J has a nonzero zero-divisor b, which is a contradiction. Thus, we conclude that P is only adjacent to I in $\mathbb{AG}(A)$.

• If $P = K \oplus H$, where K is a nonzero subset of I and $\langle 0\rangle \neq H \subseteq J$. Since $\mathbb{AG}(A)$ is connected by Theorem 3.2, there must be a vertex $L \in V(\mathbb{AG}(A))$ such that $PL = \langle 0\rangle$. That is equivalent to $(K \oplus H)L = \langle 0\rangle$. We have that $KL = HL = \langle 0\rangle$ from $KL, HL \subseteq (K \oplus H)L$. It is enough to prove that $L \cap J = \{0\}$. In fact, suppose that $L \cap J \neq \{0\}$. Then, there exists an element $x \in L \cap J$ and $x \neq 0$. It follows that x is a nonzero zero-divisor of J from $HL = \langle 0\rangle$ and $\langle 0\rangle \neq H \subseteq J$, a contradiction. Thus, since $I \cap J = \{0\}$, $I \oplus J = A$ and I is a 0-minimal ideal of A, it must be $L = I$. That is to say, any vertex that adjacent to P is equal to I.

Therefore, $\mathbb{AG}(A)$ is a star graph, and its center is the 0-minimal ideal I. □

5 The Annihilating-ideal Graphs of Boolean Algebras

In this section, we restrict MV-algebras to Boolean algebras to study the graph $\mathbb{AG}(A)$. Let A be a Boolean algebra. Connections between $\mathbb{AG}(A)$ and $\Gamma(A)$ are studied. We also verify that $\mathbb{AG}(A)$ is a star graph if and only if there is a vertex of $\mathbb{AG}(A)$ which is adjacent to every other vertex. In addition, for any ideal I of A, we denote $I_A = A\backslash\{I \cup \{1\}\} \cup \{0\}$ which is not necessarily an ideal of A. For the star graph $\mathbb{AG}(A)$ whose center is I, if I_A is not an ideal of A, we find that A has the unique 0-minimal ideal I.

Remark 5.1.

(1) By Remark 4.1 (3), $ann(A\backslash\{1\})$ is not necessarily an ideal of an MV-algebra A. However, for a Boolean algebra A, we will obtain that $ann(A\backslash\{1\}) = \langle 0\rangle$ is an

ideal of A from $x \odot x = x$ for all $x \in A$. Else, for any $x \in A$, the principal ideal generated by x is $\langle x \rangle = \{y \mid y \leq nx$ for some integer $n \geq 0\} = \{y \mid y \leq x\}$ by $x \oplus x = x$.

(2) Let A be an MV-algebra. For any two elements $m, n \in A \backslash \{0, 1\}$ and $m \neq n$, it is possible that $\langle m \rangle = \langle n \rangle$. And the principal ideal generated by $z \in A \backslash \{0, 1\}$ may be equal to A. For example, consider the algebra $M_2 = \{0, x, y, z, w, 1\}$ in Example 3.8. It is easy to get that $\langle x \rangle = \langle z \rangle = \{0, x, z\}$ and $\langle w \rangle = M_2$. However, for Boolean algebras, we have the following.

Proposition 5.2. *Let A be a Boolean algebra. Then $\langle x \rangle \in V(\mathbb{AG}(A))$ for any element $x \in A \backslash \{0, 1\}$.*

Proof. Let A be a Boolean algebra and $x \in A, x \neq 0, 1$. It follows from $0 \neq x \in \langle x \rangle = \{y \mid y \leq x\}$ that $\langle x \rangle \neq \langle 0 \rangle$. Suppose that $\langle x \rangle = A$. Then, we have that $1 \in A = \langle x \rangle$. It must be that $x = 1$, which is a contradiction. That is to say, $\langle x \rangle \in \mathbb{I}^*(A) \backslash \{A\}$ for any nonzero element $x \in A \backslash \{1\}$. Since $x \neq 0, 1$, we have $\langle x^* \rangle \in \mathbb{I}^*(A) \backslash \{A\}$ and $\langle x \rangle \cdot \langle x^* \rangle = \langle 0 \rangle$. In addition, note that $\langle x \rangle \neq \langle x^* \rangle$. Otherwise, it must be that $x = x \oplus x = 1$, a contradiction. Thus, $\langle x \rangle$ must be a vertex of $\mathbb{AG}(A)$. $\square$

For $x, y \in A$, we always say a principal ideal $\langle x \rangle$ is unique if $\langle x \rangle = \langle y \rangle$ if and only if $x = y$. In order to investigate the correspondences between $\Gamma(A)$ and $\mathbb{AG}(A)$, the following lemma is needed.

Lemma 5.3. *Let A be a Boolean algebra. Then any principal ideal $\langle x \rangle$ generated by $x \in A \backslash \{0, 1\}$ is unique and proper.*

Proof. From Proposition 5.2 we see that any principal ideal generated by $x \in A \backslash \{0, 1\}$ is proper. Suppose that $\langle a \rangle = \langle b \rangle$ for two elements $a, b \in A \backslash \{0, 1\}$ with $a \neq b$. Since $\langle a \rangle \cdot \langle a^* \rangle = \langle 0 \rangle$ and $\langle b \rangle \cdot \langle b^* \rangle = \langle 0 \rangle$, we have that $a^* \odot b = a^* \odot a = 0$ and $a^* \oplus b = a^* \oplus a = 1$, which implies that $a = b$ by Lemma 2.7. We get a contradiction. Thus, $\langle x \rangle$ is unique and proper for any $x \in A \backslash \{0, 1\}$. $\square$

Let A be a Boolean algebra. Since $V(\Gamma(A)) = A \backslash \{0, 1\}$, then we have that $\mid V(\mathbb{AG}(A)) \mid \geq \mid V(\Gamma(A)) \mid$ by Proposition 5.2 and Lemma 5.3. Therefore, we have the following.

Theorem 5.4. *Let A be a Boolean algebra. Then there exists a subgraph of $\mathbb{AG}(A)$ which is isomorphic to $\Gamma(A)$.*

Proof. Let $a \in A \backslash \{0, 1\}$. Since $\Gamma(A)$ is a connected graph, there exists $b \in A \backslash \{0, 1\}$ such that $b \neq a$ and $a \odot b = 0$. Since A is a Boolean algebra, we have that $\langle a \rangle \neq \langle b \rangle$ and $\langle a \rangle, \langle b \rangle \in V(\mathbb{AG}(A))$ by Proposition 5.2 and Lemma 5.3. We claim that $\langle a\rangle \cdot \langle b\rangle = \langle 0\rangle$. In fact, for any $w_1 \in \langle a\rangle$ and any $w_2 \in \langle b\rangle$, it must be $w_1 \leq a$ and $w_2 \leq b$. So we have that $b^* \leq w_2^*$ from Lemma 2.5 (5). By $a \odot b = (a^* \oplus b^*)^* = 0$, we get that $a \leq b^*$. Thus, $w_1 \leq a \leq b^* \leq w_2^*$. It implies $w_1 \odot w_2 = 0$. Therefore, there exists an edge $\langle a\rangle - \langle b\rangle$ in $\mathbb{AG}(A)$ by $\langle a\rangle \cdot \langle b\rangle = \langle 0\rangle$.

Conversely, suppose that $\langle a\rangle$ is adjacent to $\langle b\rangle$ in $\mathbb{AG}(A)$ for $\langle a\rangle, \langle b\rangle \in V(\mathbb{AG}(A))$. It must be that $a \odot b = 0$, where $a \neq b$ and $a, b \in A\backslash\{0,1\} = V(\Gamma(A))$. That is to say, the two distinct vertices a, b are adjacent in $\Gamma(A)$. Hence, the two vertices $\langle a\rangle, \langle b\rangle$ are adjacent in $\mathbb{AG}(A)$ iff a is adjacent to b in $\Gamma(A)$ for any $a, b \in A\backslash\{0,1\}$.

Consider the subgraph $\widehat{\mathbb{AG}(A)}$ of $\mathbb{AG}(A)$ which with the vertex set $\{\langle x\rangle \mid x \in A\backslash\{0,1\}\}$ and any two distinct vertices $\langle x\rangle, \langle y\rangle$ are adjacent iff $\langle x\rangle \cdot \langle y\rangle = \langle 0\rangle$ if and only if $x \odot y = 0$. Since vertices x, y are adjacent in $\Gamma(A)$ if and only if $x \odot y = 0$, the map $g : \widehat{\mathbb{AG}(A)} \longrightarrow \Gamma(A)$ given by $g(\langle x\rangle) = x$ is a structure-preserving bijection.

Hence, we get $\Gamma(A) \cong \widehat{\mathbb{AG}(A)}$. $\square$

We give an example to show that Theorem 5.4 does not necessarily hold on MV-algebras.

Example 5.5. Let $M_4 = \{0, p, q, s, t, u, v, w, x, y, z, 1\}$ be the 12-element MV-algebra in [2, Example 2.10] and the operations $*, \oplus$ be defined as follows:

$*$	0	p	q	s	t	u	v	w	x	y	z	1
	1	z	y	x	w	v	u	t	s	q	p	0

$\oplus$	0	p	q	s	t	u	v	w	x	y	z	1
0	0	p	q	s	t	u	v	w	x	y	z	1
p	p	q	q	t	u	u	w	x	x	z	1	1
q	q	q	q	u	u	u	x	x	x	1	1	1
s	s	t	u	s	t	u	y	z	1	y	z	1
t	t	u	u	t	u	u	z	1	1	z	1	1
u	u	u	u	u	u	u	1	1	1	1	1	1
v	v	w	x	y	z	1	v	w	x	y	z	1
w	w	x	x	z	1	1	w	x	x	z	1	1
x	x	x	x	1	1	1	x	x	x	1	1	1
y	y	z	1	y	z	1	y	z	1	y	z	1
z	z	1	1	z	1	1	z	1	1	z	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

So the operation $\odot$ is:

$\odot$	0	p	q	s	t	u	v	w	x	y	z	1
0	0	0	0	0	0	0	0	0	0	0	0	0
p	0	0	p	0	0	p	0	0	p	0	0	p
q	0	p	q	0	p	q	0	p	q	0	p	q
s	0	0	0	s	s	s	0	0	0	s	s	s
t	0	0	p	s	s	t	0	0	p	s	s	t
u	0	p	q	s	t	u	0	p	q	s	t	u
v	0	0	0	0	0	0	v	v	v	v	v	v
w	0	0	p	0	0	p	v	v	w	v	v	w
x	0	p	q	0	p	q	v	w	x	v	w	x
y	0	0	0	s	s	s	v	v	v	y	y	y
z	0	0	p	s	s	t	v	v	w	y	y	z
1	0	p	q	s	t	u	v	w	x	y	z	1

We get that $p \leq q; p, s \leq t; p, q, s, t \leq u; p, v \leq w; p, q, v, w \leq x; s, v \leq y$ and $p, s, t, v, w, y \leq z$ from $q \oplus z = t \oplus x = t \oplus z = u \oplus w = u \oplus x = u \oplus y = u \oplus z = w \oplus z = x \oplus y = x \oplus z = z \oplus z = 1$. From the simple operation we have that $V(\mathbb{AG}(M_4)) = \{P_1, P_2, P_3, P_4, P_5, P_6\}$, where $P_1 = \{0, p, q\}, P_2 = \{0, s\}, P_3 = \{0, p, q, s, t, u\}, P_4 = \{0, v\}, P_5 = \{0, p, q, v, w, x\}$ and $P_6 = \{0, s, v, y\}$. The graphs $\Gamma(M_4), \mathbb{AG}(M_4)$ of M_4 are as follows:

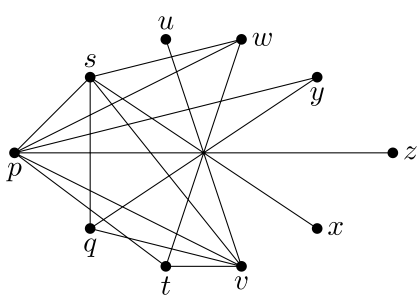


Figure 3. $\Gamma(M_4)$

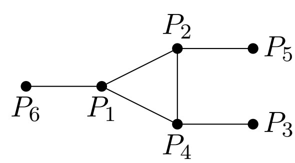


Figure 4. $\mathbb{AG}(M_4)$

The graph Figure 4 implies that $d_{max}(\mathbb{AG}(M_4)) = 3$. Obviously, from Figure 3 and Figure 4, there is no subgraph of $\mathbb{AG}(M_4)$ which is isomorphic to $\Gamma(M_4)$. In fact, it is easy to get that $\langle p \rangle = \langle q \rangle = P_1, \langle t \rangle = \langle u \rangle = P_3, \langle w \rangle = \langle x \rangle = P_5$ and $\langle z \rangle = \langle 1 \rangle$.

In addition, by Figure 4, there exists a 3-vertex subgraph $\overline{\mathbb{AG}(M_4)}$ of $\mathbb{AG}(M_4)$ which forms a clique of $\mathbb{AG}(M_4)$. The vertex set $\{P_1, P_2, P_4\}$ of $\overline{\mathbb{AG}(M_4)}$ exactly consists of all 0-minimal ideals of M_4.

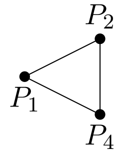


Figure 5. $\overline{\mathbb{AG}(M_4)}$

Proposition 5.6. *Let A be a Boolean algebra. If there are two MV-algebras A_1, A_2 satisfying $\mid A_1 \mid\geq 2, \mid A_2 \mid\geq 2$ and A is isomorphic to $A_1 \times A_2$, then $d_{max}(\mathbb{AG}(A)) = 3$ if and only if $\mid A_1 \mid\geq 3$ or $\mid A_2 \mid\geq 3$.*

Proof. $\Longleftarrow$) Suppose that $\mid A_1 \mid\geq 3$ or $\mid A_2 \mid\geq 3$. By [1, Proposition 4] we see that $d_{max}(\Gamma(A)) = 3$. By Theorem 5.4, there is a subgraph of $\mathbb{AG}(A)$ which is isomorphic to $\Gamma(A)$. It implies that $d_{max}(\mathbb{AG}(A)) \geq d_{max}(\Gamma(A)) = 3$. By Theorem 3.2, we get $d_{max}(\mathbb{AG}(A)) = 3$.

$\Longrightarrow$) Suppose that $d_{max}(\mathbb{AG}(A)) = 3$. Assume that $\mid A_1 \mid= 2$ and $\mid A_2 \mid= 2$. It must be $A_1 = A_2 = L_2$, then $A_1 \times A_2 = L_2 \times L_2 \cong \mathbf{B}_4$ by Example 2.2. From Remark 4.3 and $A \cong A_1 \times A_2$, we have that $d_{max}(\mathbb{AG}(A)) = d_{max}(\mathbb{AG}(\mathbf{B}_4)) = 1$, which is a contradiction. Thus, $\mid A_1 \mid\geq 3$ or $\mid A_2 \mid\geq 3$. ☐

Then we give an example to show that there do exist a Boolean algebra A such that $d_{max}(\mathbb{AG}(A)) = 3$.

Example 5.7.

(1) Let $\mathbf{B}_8 = \{0, u, v, w, x, y, z, 1\}$ be the 8-element Boolean algebra and the operations $*, \oplus$ be given as follows:

$*$	0	u	v	w	x	y	z	1
	1	z	y	x	w	v	u	0

$\oplus$	0	u	v	w	x	y	z	1
0	0	u	v	w	x	y	z	1
u	u	u	u	u	1	1	1	1
v	v	u	v	u	x	1	x	1
w	w	u	u	w	1	y	y	1
x	x	1	x	1	x	1	x	1
y	y	1	1	y	1	y	y	1
z	z	1	x	y	x	y	z	1
1	1	1	1	1	1	1	1	1

Then, the operation $\odot$ is:

$\odot$	0	u	v	w	x	y	z	1
0	0	0	0	0	0	0	0	0
u	0	u	v	w	v	w	0	u
v	0	v	v	0	v	0	0	v
w	0	w	0	w	0	w	0	w
x	0	v	v	0	x	z	z	x
y	0	w	0	w	z	y	z	y
z	0	0	0	0	z	z	z	z
1	0	u	v	w	x	y	z	1

By Lemma 2.5, it follows from $u \oplus x = 1, u \oplus y = 1$ and $x \oplus y = 1$ that $v, w \leq u; v, z \leq x$ and $w, z \leq y$. Therefore, all the principal ideals of $\mathbf{B}_8$ are: $\langle 0 \rangle = \{0\}, \langle 1 \rangle = \mathbf{B}_8, \langle u \rangle = \{0, u, v, w\}, \langle v \rangle = \{0, v\}, \langle w \rangle = \{0, w\}, \langle x \rangle = \{0, v, x, z\}, \langle y \rangle = \{0, w, y, z\}$ and $\langle z \rangle = \{0, z\}$.
Also, note that $V(\mathbb{AG}(\mathbf{B}_8)) = \{\langle u \rangle, \langle v \rangle, \langle w \rangle, \langle x \rangle, \langle y \rangle, \langle z \rangle\}$ and $\langle y \rangle \cdot \langle v \rangle = \langle w \rangle \cdot \langle v \rangle = \langle w \rangle \cdot \langle z \rangle = \langle z \rangle \cdot \langle v \rangle = \langle w \rangle \cdot \langle x \rangle = \langle z \rangle \cdot \langle u \rangle = \langle 0 \rangle$.

Therefore, from the simple operation, the graphs $\mathbb{AG}(\mathbf{B}_8)$ and $\Gamma(\mathbf{B}_8)$ of $\mathbf{B}_8$ are respectively:

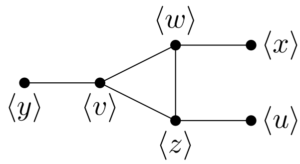


Figure 6. $\mathbb{AG}(\mathbf{B}_8)$

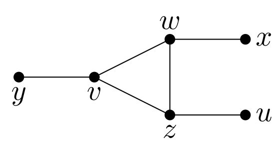


Figure 7. $\Gamma(\mathbf{B}_8)$

Obviously, Figure 6 and Figure 7 imply that $\Gamma(\mathbf{B}_8) \cong \mathbb{AG}(\mathbf{B}_8)$ and $d_{max}(\mathbb{AG}(\mathbf{B}_8)) = 3$. Moreover, there also exists a complete subgraph of $\mathbb{AG}(\mathbf{B}_8)$ which the vertex set $\{\langle v \rangle, \langle w \rangle, \langle z \rangle\}$ composed of all 0-minimal ideals of $\mathbf{B}_8$.

(2) Consider the MV-algebra $L_2 \times \mathbf{B}_4$. $L_2 \times \mathbf{B}_4$ is a Boolean algebra with $\mid L_2 \mid= 2$ and $\mid \mathbf{B}_4 \mid= 4 \geq 3$. We can directly get that $d_{max}(\mathbb{AG}(L_2 \times \mathbf{B}_4)) = 3$ from Proposition 5.6. In fact, since $\mathbf{B}_8 \cong L_2 \times \mathbf{B}_4$, it must be that $d_{max}(\mathbb{AG}(L_2 \times \mathbf{B}_4)) = d_{max}(\mathbb{AG}(\mathbf{B}_8)) = 3$ by (1).

Remark 5.8.

(1) We claim that Proposition 5.6 is not necessarily true for some MV-algebras. Let M_2 be the MV-algebra in Example 3.8. From Example 3.8, $M_2 \cong L_2 \times L_3$ and $\mid L_2 \mid= 2, \mid L_3 \mid= 3$. But, from Figure 1, $d_{max}(\mathbb{AG}(M_2)) = 1$.

(2) By [1, Corollary 1], if $\mid A \mid= n \neq 4$ for some integer number n, then $\Gamma(A) \cong \Gamma(L_n)$ if and only if $A \cong L_n$. Obviously, it must be $\mathbb{AG}(A) \cong \mathbb{AG}(B)$ if $A \cong B$ for two MV-algebras A, B. But, the converse is not necessarily true.

Consider the two MV-algebras $L_2 \times L_6$ and $L_3 \times L_4$. Then, $| L_2 \times L_6 |=| L_3 \times L_4 |$ and $\mathbb{AG}(L_2 \times L_6) \cong K_2 \cong \mathbb{AG}(L_3 \times L_4)$ by Proposition 3.7. But, it is enough to show that $L_2 \times L_6 \ncong L_3 \times L_4$. In fact, in $L_3 \times L_4$, there exist four elements such that $(\frac{1}{2}, \frac{2}{3}) \oplus (\frac{1}{2}, \frac{2}{3}) = (\frac{1}{2}, 1) \oplus (\frac{1}{2}, 1) = (1, \frac{2}{3}) \oplus (1, \frac{2}{3}) = (1,1) \oplus (1,1) = (1,1)$ by Example 2.2. However, there are only three elements of $L_2 \times L_6$ that satisfy $(1, \frac{3}{5}) \oplus (1, \frac{3}{5}) = (1, \frac{4}{5}) \oplus (1, \frac{4}{5}) = (1,1) \oplus (1,1) = (1,1)$.

Remark 5.9.

(1) Let A be a Boolean algebra and I, J be two ideals of A. We conclude that IJ is an ideal of A. In fact, for all $x \in IJ$ and $y \in A$ with $y \leq x$, there exist $m \in I$ and $n \in J$ such that $y \leq x = m \odot n \leq m, n$ by Lemma 2.7. It implies that $y \in I \cap J$. Thus, we have $y = y \odot y \in IJ$. In addition, let $x, y \in IJ$. By Lemma 2.5 and Lemma 2.7, it must be that $x \oplus y = (m_1 \odot n_1) \oplus (m_2 \odot n_2) \leq m_1 \oplus m_2, n_1 \oplus n_2$ for $x = m_1 \odot n_1$ and $y = m_2 \odot n_2$. It follows that $x \oplus y \in I \cap J$. Thus, $x \oplus y = (x \oplus y) \odot (x \oplus y) \in IJ$. Therefore, IJ is also an ideal of A. Particularly, I^k is also an ideal of A for any ideal I, where k is a positive integer number.

(2) Let A be a Boolean algebra and I be an ideal of A. It is easy to get that $I^{k+1} \subseteq I^k$ from Lemma 2.7 (1) and (2). Further, we will claim that $I^k \subseteq I^{k+1}$. That is to say, $I^{k+1} = I^k$. In fact, for any $x \in I^k$, there are $a_i \in I$ such that $x = a_1 \odot a_2 \odot \cdots \odot a_n \leq a_i$ for any $i \in \{1, 2, ..., n\}$ by Lemma 2.7, which implies that $x \in I$. So we have that $x = x \odot x = a_1 \odot a_2 \odot \cdots \odot a_n \odot x \in I^{k+1}$. Hence, $I^k = I^{k+1}$.

Next, we will characterize Boolean algebras whose annihilating-ideal graphs are star graphs.

Lemma 5.10. *Let A be a Boolean algebra and I be an ideal of A with $I \neq \langle 0 \rangle$. Then $ann(I)$ is a proper ideal of A.*

Proof. (1) $ann(I)$ is an ideal of A.

If $I = A$. Then, since $x \odot 1 = 0$ if and only if $x = 0$ for all $x \in A$, it must be that $ann(I) = ann(A) = \langle 0 \rangle$.

Suppose that I is a nonzero ideal and $I \neq A$. Obviously, $0 \in ann(I)$. Let $a \in ann(I) = \{z \in A \mid z \odot y = 0 \text{ for all } y \in I\}$ and $b \in A$ with $b \leq a$. Then, $a \leq c^*$ for any $c \in I$. So it must be $b \leq a \leq c^*$, giving that $b \odot c = 0$ and so $b \in ann(I)$. In addition, let $a, b \in ann(I)$. Then $a \leq c^*$ and $b \leq c^*$ for any $c \in I$, which implies that $a \oplus b \leq a \oplus c^* \leq c^* \oplus c^* = c^*$. That is, $a \oplus b \in ann(I)$. Therefore, $ann(I)$ is an ideal of A.

(2) $ann(I)$ is proper.

Suppose that $1 \in ann(I)$. For all $y \in I$, we have that $y = y \odot 1 = 0$, a contradiction.

Hence, $ann(I) \in \mathbb{I}(A) \backslash \{A\}$ for any nonzero ideal I. $\square$

Proposition 5.11. *Let A be a Boolean algebra such that $ann(I) \cap ann(J) \neq \langle 0 \rangle$ for any $I, J \in V(\mathbb{AG}(A))$. Then $d_{max}(\mathbb{AG}(A)) \leq 2$.*

Proof. Let $I, J \in V(\mathbb{AG}(A))$. If $IJ = \langle 0 \rangle$, there is nothing to prove. Suppose that $IJ \neq \langle 0 \rangle$. Then consider the two ideals I^2 and J^2.

It is easy to get that $I^2, J^2 \neq \langle 0 \rangle$. Obviously, $1 \notin ann(I) \cap ann(J)$ by Lemma 5.10. Since $ann(I) \cap ann(J) \neq \langle 0 \rangle$, there exists an element x satisfying $x \neq 0$ and $x \in ann(I) \cap ann(J)$. It follows that $\langle x \rangle \cdot I = \langle x \rangle \cdot J = \langle 0 \rangle$ and $\langle x \rangle \in V(\mathbb{AG}(A))$ from Proposition 5.2. Note that $I^2, J^2 \neq \langle 0 \rangle$, then $\langle x \rangle \neq I, J$. Hence, there is an edge $I - \langle x \rangle - J$ between I and J. Thus, we have that $d(I, J) = 2$ in this case.

Therefore, it must be that $d_{max}(\mathbb{AG}(A)) \leq 2$. □

Remark 5.12.

(1) The converse of Proposition 5.11 is not necessarily true. Consider the 4-element Boolean algebra $\mathbf{B}_4 = \{0, a, b, 1\}$.
From Remark 4.3 we have that $d_{max}(\mathbb{AG}(\mathbf{B}_4)) = 1 \leq 2$. However, $ann(\{0, a\}) \cap ann(\{0, b\}) = \{0, b\} \cap \{0, a\} = \{0\}$.

(2) Any nonzero ideal of the semigroup S can be regarded as a union of principal ideals of S by [14]. Now, we will show that the similar statement also holds on any MV-algebra. Let A be an MV-algebra and $I \in \mathbb{I}(A)$. We denote $W = \bigcup\limits_{x \in I} \langle x \rangle$ which is a subset of A. We will obtain that I is an ideal generated by W. That is, $I = \langle W \rangle$. Indeed, $I = W$.

In fact, if $I = \langle 0 \rangle$ or $\langle 1 \rangle$, we are done. Suppose that I is a nonzero proper ideal of A. On the one hand, for any $a \in I$, it must be $a \in \langle a \rangle \subseteq \langle W \rangle$. Thus, $I \subseteq \langle\ W \rangle$. On the other hand, for any $a \in \langle W \rangle$, we have that $a \leq w_1 \oplus w_2 \oplus \cdots \oplus w_k$ for some $w_i \in W = \bigcup\limits_{x \in I} \langle x \rangle$ by Lemma 2.4, where $i = 1, 2, ..., k$. There is $x_j \in I$ such that w_i belongs to $\langle x_j \rangle$ for each $i \in \{1, 2, ..., k\}$, where $j \in \{1, 2, ..., l\}$ and $l \leq k$. It follows that $a \leq w_1 \oplus w_2 \oplus \cdots \oplus w_k \leq n_1 x_1 \oplus n_2 x_2 \oplus \cdots \oplus n_l x_l$ for some $n_j \in \mathbf{N}$ and $j \in \{1, 2, ..., l\}$. Then $a \in I$, and so $\langle W \rangle \subseteq I$. Therefore, $I = \langle W \rangle$ for any $I \in \mathbb{I}(A)$. In particular, by Lemma 2.4, it is obvious that $\langle x \rangle \subseteq \langle W \rangle$ for any $x \in I$. Hence, for each 0-minimal ideal I of an MV-algebra A, there exists a nonzero element $x \in A \backslash \{1\}$ such that $I = \langle x \rangle$ by the 0-minimality of I. For example, let M_4 be the algebra in Example 5.5. The 0-minimal ideals P_1, P_2, P_4 of M_4 are principal ideals respectively generated by p (or q), s, v.

Theorem 5.13. *Let A be a Boolean algebra. $\mathbb{AG}(A)$ is a star graph iff there exists a vertex I of $\mathbb{AG}(A)$ satisfying $IJ = \langle 0 \rangle$ for all $J \in V(\mathbb{AG}(A))$ with $I \neq J$.*

Proof. $\Longrightarrow$) This follows directly from the definition of the star graph.

$\Longleftarrow$) Assume that there is $I \in V(\mathbb{AG}(A))$ such that $IJ = \langle 0\rangle$ for any $J \in V(\mathbb{AG}(A))$, where $J \neq I$. It is obvious that $ann(A\backslash\{1\}) = \langle 0\rangle$ by Remark 5.1 (1). Then, we will use Theorem 4.5 to prove that $\mathbb{AG}(A)$ is a star graph.

• I is a 0-minimal ideal of A.

Let $a \in I$ and $a \neq 0$. We have that $\langle a\rangle \in V(\mathbb{AG}(A))$ by Proposition 5.2. Since I is an ideal of A, we get $\langle a\rangle \subseteq I$. If $\langle a\rangle \subsetneqq I$, since $IJ = \langle 0\rangle$ for all vertices J of $\mathbb{AG}(A)$ with $I \neq J$, we obtain that $\langle a\rangle \cdot I = \langle 0\rangle$. It follows from $a \in I$ that $a \odot a = 0$. Hence, $a = a \odot a = 0$, which is a contradiction. Thus, $I = \langle a\rangle$ is a principal ideal generated by a.

Next, we will verify that $I = \langle a\rangle$ is a 0-minimal ideal of A. Otherwise, from Remark 5.12 (2), there must be a principal ideal $\langle b\rangle$ such that $\langle b\rangle \subseteq \langle a\rangle$ for some $b \in A$ and $b \neq a, 0, 1$. Then, by Proposition 5.2, we have that $\langle b\rangle \in V(\mathbb{AG}(A))$. Since $\langle a\rangle$ is adjacent to every other vertex of $\mathbb{AG}(A)$, we get that $\langle a\rangle \cdot \langle b\rangle = \langle 0\rangle$. Again since $\langle b\rangle \subseteq \langle a\rangle$, it must be that $b = b \odot b = 0$, which is a contradiction. Thus, $\langle a\rangle$ is a 0-minimal ideal of A. Further, we will claim that $\mid I \mid= 2$. First, it must be $0, a \in I$. Suppose that $\mid I \mid\geq 3$. Let $z \in I = \langle a\rangle = \{w \mid w \leq a\}$, where $z \neq 0, a$. We have that $z \leq a$, and $\langle 0\rangle \neq \langle z\rangle \subseteq \langle a\rangle$. It follows that $\langle z\rangle \subsetneqq \langle a\rangle$ by Lemma 5.3, which is contradict to the fact that $\langle a\rangle$ is a 0-minimal ideal. Thus, we get that $I = \{0, a\}$.

• $\langle a\rangle \cap ann(\langle a\rangle) = \{0\}$.

Consider the set $\langle a\rangle \cap ann(\langle a\rangle)$. Assume that there is a nonzero element $x \in \langle a\rangle \cap ann(\langle a\rangle)$. Since $\langle a\rangle = \{0, a\}$, it must be that $a = x \in ann(\langle a\rangle)$. It implies that $a = a \odot a = 0$, a contradiction to $a \neq 0$. Thus, we obtain that $\langle a\rangle \cap ann(\langle a\rangle) = \{0\}$.

• $\langle a\rangle \oplus ann(\langle a\rangle) = A$.

Let $x \in A$. If $x = 0$, it is obvious that $x = 0 = 0 \oplus 0 \in \langle a\rangle \oplus ann(\langle a\rangle)$. If $x^* = 0$, we have that $x = 1 = a \oplus a^* \in \langle a\rangle \oplus ann(\langle a\rangle)$. If $x = a$, there is nothing to prove. So suppose that $x \neq 0, 1, a$. By Proposition 5.2 and Lemma 5.3, it must be that $\langle x\rangle$ is a vertex of $\mathbb{AG}(A)$ and different from $\langle a\rangle$. Since $\langle a\rangle$ is adjacent to every other vertex of $\mathbb{AG}(A)$, we have that $\langle x\rangle \cdot \langle a\rangle = \langle 0\rangle$, which implies that $x \in ann(\langle a\rangle)$. Thus, $x = 0 \oplus x \in \langle a\rangle \oplus ann(\langle a\rangle)$. We obtain that $A \subseteq \langle a\rangle \oplus ann(\langle a\rangle)$. Obviously, $\langle a\rangle \oplus ann(\langle a\rangle) \subseteq A$. Therefore, $\langle a\rangle \oplus ann(\langle a\rangle) = A$.

Summarizing the above, we conclude that $A = I \oplus ann(I)$ and $I \cap ann(I) = \{0\}$.

• $ann(\langle a\rangle)$ contains no nonzero zero-divisors.

Otherwise, there are two elements x, y of $ann(\langle a\rangle)$ such that $x \odot y = 0$ and $x, y \neq 0$. We denote $W = \langle x\rangle \cup \langle a\rangle$. It is easy to show that $\langle W\rangle = \{w \in A \mid w \leq x, w \leq a$ or $w \leq x \oplus a\}$ is a nonzero ideal of A by $a, x \in \langle W\rangle$. Since $a \odot x = a \odot y = x \odot y = 0$, we have that $\langle y\rangle \cdot \langle W\rangle = \langle 0\rangle$. Thus, it follows that $\langle W\rangle = A$ or $\langle W\rangle \in V(\mathbb{AG}(A))$. If $\langle W\rangle \in V(\mathbb{AG}(A))$, we have $\langle W\rangle \cdot \langle a\rangle = \langle 0\rangle$. It must be $a = a \odot a = 0$, a contradiction. Then, $\langle W\rangle = A$. Since $\langle y\rangle \cdot \langle W\rangle = \langle 0\rangle$, we obtain that $y = y \odot 1 = 0$ by $1 \in \langle W\rangle$, a contradiction. Therefore, $ann(\langle a\rangle)$ contains no nonzero zero-divisors.

• By Lemma 5.10, $ann(\langle a\rangle)$ is a proper ideal of A.
Hence, we get that $\mathbb{AG}(A)$ is a star graph by Theorem 4.5. $\square$

Let A be a Boolean algebra and I be a 0-minimal ideal. From the proof of Theorem 5.13, we have that $\mid I \mid= 2$ by the uniqueness of principal ideals. However, let M_4 be the MV-algebra in Example 5.5. It is obvious that P_1 is a 0-minimal ideal of M_4, but $\mid P_1 \mid= 3$.

Let A be an MV-algebra and $I \in \mathbb{I}(A)$. I contains no zero-divisor pairs means that there are no elements $x, y \in I$ satisfying $x \odot y = 0$ and $x, y \neq 0$. Then we have the following.

Corollary 5.14. *Let A be a Boolean algebra. Then, $\mathbb{AG}(A)$ is a star graph iff $A = I \oplus ann(I)$, where $ann(I)$ contains no zero-divisor pairs and I is a 0-minimal ideal of A.*

Proof. $\Longrightarrow$) Suppose that $\mathbb{AG}(A)$ is a star graph and I is the center of $\mathbb{AG}(A)$. This follows directly from the proof of Theorem 5.13.

$\Longleftarrow$) We have that $I \cap ann(I) = \{0\}$. Otherwise, there exists an element satisfying $x \in I \cap ann(I)$ and $x \neq 0$, which implies that $x = x \odot x = 0$, a contradiction. Then, it follows from Theorem 4.5. $\square$

Let A be a Boolean algebra such that $\mathbb{AG}(A)$ be a star graph. By the proof of Theorem 5.13 we see that the center of $\mathbb{AG}(A)$ is exactly a 0-minimal ideal of A. Similarly, for an MV-algebra A, we have the following.

Proposition 5.15. *Let A be an MV-algebra. If $\mathbb{AG}(A)$ is a star graph with the center I, then I is a 0-minimal ideal of A.*

Proof. Let I be the center of $\mathbb{AG}(A)$. Assume that $J \in \mathbb{I}^*(A)\backslash\{A\}$ such that $J \subsetneq I$. Since I is the center, we have that $JK \subseteq IK = \langle 0\rangle$ for any $I \neq K \in V(\mathbb{AG}(A))$, which implies that $J \in V(\mathbb{AG}(A))$ and J is adjacent to K in $\mathbb{AG}(A)$, contradicting to the fact that $\mathbb{AG}(A)$ is a star graph. Thus, I is a 0-minimal ideal of A. $\square$

Let A be an MV-algebra. If $x \odot y = 0$ for some $x, y \in A$, there can not guarantee that $\langle x\rangle \cdot \langle y\rangle = \langle 0\rangle$. For example, consider the MV-algebra M_2 in Example 3.8. We have that $x \odot w = 0$. But, $\langle x\rangle \cdot \langle w\rangle = \{0, x, z\} \cdot M_2 \neq \langle 0\rangle$.

Let A be a Boolean algebra. By Theorem 5.4, we have that $\langle x\rangle \cdot \langle y\rangle = \langle 0\rangle \Longleftrightarrow x \odot y = 0$ for all $x, y \in A\backslash\{0, 1\}$. That is to say, there is an edge $\langle x\rangle - \langle y\rangle$ in $\mathbb{AG}(A)$ $\Longleftrightarrow$ the two distinct vertices x, y are adjacent in $\Gamma(A)$. Thus, we can obtain the following correspondences between $\mathbb{AG}(A)$ and $\Gamma(A)$.

Proposition 5.16. *Let A be a Boolean algebra. If $\mathbb{AG}(A)$ is a star graph, then $\Gamma(A)$ is a star graph.*

Proof. Suppose that $\mathbb{AG}(A)$ is a star graph with the center I. By the proof of Theorem 5.13 we have that I is a 0-minimal ideal and $I = \langle x \rangle$ for some $x \in A \backslash \{0, 1\}$. Since $\langle x \rangle$ is the center of $\mathbb{AG}(A)$ and $\langle y \rangle \in V(\mathbb{AG}(A))$ by Proposition 5.2, it must be that $\langle x \rangle \cdot \langle y \rangle = \langle 0 \rangle$, where $y \in A \backslash \{0, 1\}$ such that $y \neq x$. It follows that $x \odot y = 0$ for all $y \in V(\Gamma(A))$. In addition, for any $z \in V(\Gamma(A))$ and $z \neq x, y$, since $\mathbb{AG}(A)$ is a star graph, we obtain that $\langle z \rangle \cdot \langle y \rangle \neq \langle 0 \rangle$ by the uniqueness of principal ideals. Again from the proof of Theorem 5.4, $z \odot y \neq 0$. Thus, $\Gamma(A)$ is a star graph with the center x. $\square$

Proposition 5.17. *Let A be a Boolean algebra. If $\mathbb{AG}(A)$ is a complete graph, then $\Gamma(A)$ is a complete graph.*

Proof. Suppose that $\mathbb{AG}(A)$ is complete. Then for any two distinct elements x, y of $A \backslash \{0, 1\}$, $\langle x \rangle, \langle y \rangle \in V(\mathbb{AG}(A))$ and $\langle x \rangle \cdot \langle y \rangle = \langle 0 \rangle$ by the completelity of $\mathbb{AG}(A)$. That is to say, $x \odot y = 0$ for any $x, y \in A \backslash \{0, 1\} = V(\Gamma(A))$ $(x \neq y)$. Thus, $\Gamma(A)$ is a complete graph. $\square$

Proposition 5.16 and Proposition 5.17 are not necessarily true for MV-algebras. Consider the MV-algebra M_2 in Example 3.8. $\mathbb{AG}(M_2)$ is a star graph and a complete graph. However, $\Gamma(M_2)$ is neither a star graph nor a complete graph.

Let A be an MV-algebra and I be an ideal of A. We denote $I_A = A \backslash \{I \cup \{1\}\} \cup \{0\}$. Obviously, the set I_A may not belong to $\mathbb{I}(A)$. For example, we can consider the ideal $\langle v \rangle$ of $\mathbf{B}_8$ in Example 5.7 (1). It is easy to get that $\langle v \rangle_{\mathbf{B}_8} = \mathbf{B}_8 \backslash \{\langle v \rangle \cup \{1\}\} \cup \{0\} = \{0, u, w, x, y, z\}$, which is not an ideal of $\mathbf{B}_8$ by $u \oplus x = 1 \notin \langle v \rangle_{\mathbf{B}_8}$. By Theorem 5.13, the center I of $\mathbb{AG}(A)$ is a 0-minimal ideal of A. Next, we will study the uniqueness of I by using I_A.

Theorem 5.18. *Let A be a Boolean algebra such that $\mathbb{AG}(A)$ be a star graph whose center is I. If I_A is not an ideal of A, then A has the unique 0-minimal ideal I.*

Proof. Suppose that $I_A \notin \mathbb{I}(A)$ and I is the center of $\mathbb{AG}(A)$.

By the proof of Theorem 5.13, I is a 0-minimal ideal of A. There is an element $x \in A \backslash \{0, 1\}$ such that $I = \langle x \rangle = \{0, x\}$. Suppose that $\langle y \rangle$ is another 0-minimal ideal of A for some element $y \in A \backslash \{0, 1\}$. From Proposition 5.2, we obtain that $\langle y \rangle \in V(\mathbb{AG}(A))$. Obviously, $0 \in I_A$. Then, we consider the following conditions:

• Suppose that I_A does not satisfy Definition 2.3 (ii) let I_A be not an ideal of A. There exist $a \in I_A$ and $b \in A$ with $b \leq a$ such that $b \notin I_A = A \backslash \{I \cup \{1\}\} \cup \{0\}$, where $a \neq 0$ and $b \neq 0, 1$. That is to say, $b \in I = \{0, x\}$, so it must be that $b = x$.

Since $\langle x\rangle$ is the center of $\mathbb{AG}(A)$ and $\langle a\rangle \in V(\mathbb{AG}(A))$ by Proposition 5.2, we have that $\langle x\rangle \cdot \langle a\rangle = \langle 0\rangle$. From $x = b \leq a$, it must be that $x \in \langle a\rangle$. Thus, it follows from $\langle x\rangle \cdot \langle a\rangle = \langle 0\rangle$ that $x = x \odot x = 0$, which is a contradiction. Therefore, this assumption does not hold.

• Suppose that I_A does not satisfy Definition 2.3 (iii). There are $a, b \in I_A$ such that $a \oplus b \notin I_A$, where $a, b \neq 0, 1, x$. Since A is a Boolean algebra which has the unique and proper principal ideals, we have that $\langle a\rangle, \langle b\rangle$ are two vertices which are different from $\langle x\rangle$ by Proposition 5.2. There must be $x \in ann(\langle a\rangle)$ and $x \in ann(\langle b\rangle)$ by the fact that $\langle x\rangle$ is the center of $\mathbb{AG}(A)$. Then, both $ann(\langle a\rangle)$ and $ann(\langle b\rangle)$ are nonzero proper ideals of A by Lemma 5.10. Since $a, b \neq 0$, then $\langle a\rangle \neq ann(\langle a\rangle)$ and $\langle b\rangle \neq ann(\langle b\rangle)$. By the star shape of $\mathbb{AG}(A)$ and $\langle a\rangle \cdot ann(\langle a\rangle) = \langle b\rangle \cdot ann(\langle b\rangle) = \langle 0\rangle$, we have that $ann(\langle a\rangle) = ann(\langle b\rangle) = \langle x\rangle$. Since $a^* \in ann(\langle a\rangle) = \langle x\rangle$ and $b^* \in ann(\langle b\rangle) = \langle x\rangle$, we have that $a^*, b^* \leq x$. It follows that

$$x \oplus a = x \oplus b = 1. \tag{1}$$

Moreover, since $\langle x\rangle$ is the center, we get

$$x \odot a = x \odot b = 0. \tag{2}$$

By Lemma 2.7 (5) and Equation 5.1, Equation 5.2, it must be that $a = b$, a contradiction. Hence, we conclude that $\langle y\rangle = \langle x\rangle$.

Therefore, each 0-minimal ideal of A must be a principal ideal $I = \langle x\rangle$. That is to say, the center of the star graph $\mathbb{AG}(A)$ is exactly the unique 0-minimal ideal of A. □

Remark 5.19.

(1) Let A be an MV-algebra that contains finitely many principal ideals. By Remark 5.12 (2), A satisfies the the ascending chain condition and the descending chain condition on its ideals. Suppose that $\mathbb{AG}(A)$ is a star graph such that I is the center. If I is the unique 0-minimal ideal, we claim that I_A is not an ideal of A. In fact, by the uniqueness of I, it must be that I contains in every other nonzero ideal of A. Thus, $I_A = A\backslash\{I \cup \{1\}\} \cup \{0\}$ is not an ideal of A.

(2) Theorem 5.18 is not necessarily true for MV-algebras. Consider the 6-element MV-algebra $M_2 = \{0, x, y, z, w, 1\}$ in Example 3.8. It is easy to get that $I_{1A} = A\backslash\{I_1 \cup \{1\}\} \cup \{0\} = \{0, x, z, w\}$, which is not an ideal of M_2 by $x \oplus w = 1 \notin I_{1A}$. However, both I_1 and I_2 are 0-minimal ideals of M_2.

6 The Vertex Coloring and Girth of Annihilating-ideal Graphs

In this section, we mainly investigate the coloring and girth of $\mathbb{AG}(A)$ for the MV-algebra $(A, \oplus, *, 0)$. We find that $\chi(\mathbb{AG}(A)) = \omega(\mathbb{AG}(A)) = 0$ if $ann(A\backslash\{1\}) = A\backslash\{1\}$. We give two lower bounds and an upper bound for $\chi(\mathbb{AG}(A))$. If $\mid \mathbb{M}_0(A) \mid\geq 3$, we prove that the girth of $\mathbb{AG}(A)$ is equal to 3, where $\mathbb{M}_0(A)$ is the set of all 0-minimal ideals of A.

Proposition 6.1. *Let A be an MV-algebra. If $ann(A\backslash\{1\}) = A\backslash\{1\}$. Then we have that $\chi(\mathbb{AG}(A)) = \omega(\mathbb{AG}(A)) = 0$.*

Proof. Suppose that $ann(A\backslash\{1\}) = A\backslash\{1\}$. By Theorem 4.2, $\mathbb{AG}(A)$ is a null graph. It follows that $0 =\mid V(\mathbb{AG}(A)) \mid= \chi(\mathbb{AG}(A)) = \omega(\mathbb{AG}(A))$. □

Next, we study the bound for $\chi(\mathbb{AG}(A))$ of A. We use $\mathbb{I}_{ann}$ to denote the set of all nonzero ideals of A which are contained in $ann(A\backslash\{1\})$. That is to say, denote

$$\mathbb{I}_{ann} = \{I \text{ is a nonzero ideal of } A \mid I \subseteq ann(A\backslash\{1\})\},$$

where we always have that $1 \notin ann(A\backslash\{1\})$. In fact, if $1 \in ann(A\backslash\{1\})$, it implies that $A = \{0, 1\}$ by Remark 4.1 (1). Particularly, if $ann(A\backslash\{1\}) = A\backslash\{1\}$, we can directly get that $\chi(\mathbb{AG}(A)) = 0$ by Proposition 6.1.

To get a lower bound for $\chi(\mathbb{AG}(A))$, the following lemma is needed.

Lemma 6.2. *Let A be an MV-algebra and $ann(A\backslash\{1\}) \subseteq A\backslash\{1\}$. Then $\mathbb{I}_{ann}$ corresponds to a clique of $\mathbb{AG}(A)$.*

Proof. For any two distinct ideals $I, J \in \mathbb{I}_{ann}$, we have $I, J \subseteq ann(A\backslash\{1\})$. Since $IJ \subseteq I \cdot ann(A\backslash\{1\}) \subseteq A\backslash\{1\} \cdot ann(A\backslash\{1\}) = \langle 0\rangle$, we get that $IJ = \langle 0\rangle$. It implies that there is an edge $I - J$ in $\mathbb{AG}(A)$. That is to say, each element in $\mathbb{I}_{ann}$ is adjacent to every other element. Thus, we conclude that $\mathbb{I}_{ann}$ corresponds to a clique of $\mathbb{AG}(A)$. □

Theorem 6.3. *Let A be an MV-algebra such that $ann(A\backslash\{1\}) \subseteq A\backslash\{1\}$. Then $\mid \mathbb{I}_{ann} \mid\leq \chi(\mathbb{AG}(A))$.*

Proof. By Lemma 6.2 we see that $\mathbb{I}_{ann}$ corresponds to a clique of $\mathbb{AG}(A)$. It follows that

$$\mid \mathbb{I}_{ann} \mid\leq \omega(\mathbb{AG}(A)) \leq \chi(\mathbb{AG}(A)).$$

□

By Theorem 6.3, we get a lower bound for $\chi(\mathbb{AG}(A))$. But, we claim that this lower bound is not tight enough for all types of MV-algebras. For example, let A be a Boolean algebra, it must be that $ann(A\backslash\{1\}) = \langle 0\rangle$ from Remark 5.1 (1), which implies that $\mathbb{I}_{ann} = \emptyset$. So Theorem 6.3 gives $0 \leq \chi(\mathbb{AG}(A))$. Hence, we need to continue to study the lower bound for $\chi(\mathbb{AG}(A))$.

We use $\mathbb{M}_0(A)$ to denote the set of all 0-minimal ideals of an MV-algebra $(A, \oplus, *, 0)$. Note that $I \cap J = \{0\}$ for any two distinct elements $I, J \in \mathbb{M}_0(A)$. Otherwise, there is a nonzero element $x \in I \cap J$. Since $I \cap J$ is an ideal, it must be $\langle 0\rangle \neq \langle x\rangle \subseteq I \cap J \subseteq I, J$. We get a contradiction to the 0-minimality of I, J.

Theorem 6.4. *Let A be an MV-algebra. Then $\mid \mathbb{M}_0(A) \mid$ gives a lower bound to $\chi(\mathbb{AG}(A))$.*

Proof. Let $I, J \in \mathbb{M}_0(A)$ and $I \neq J$. By the 0-minimality of I and J, we have that $IJ \subseteq I \cap J = \{0\}$. That is to say, any two 0-minimal ideals of A are adjacent. So $\mathbb{M}_0(A)$ forms a clique of $\mathbb{AG}(A)$. Therefore, it must be

$$\mid \mathbb{M}_0(A) \mid \leq \omega(\mathbb{AG}(A)) \leq \chi(\mathbb{AG}(A)).$$

□

The following examples show that the bound in Theorem 6.4 is really tighter than that given in Theorem 6.3.

Example 6.5.

- Consider the MV-algebra M_2 in Example 3.8. It is easy to get that $ann(M_2\backslash\{1\}) = \langle 0\rangle$, and so $\mid \mathbb{I}_{ann}(M_2) \mid = 0$. Theorem 6.3 gives $0 \leq \chi(\mathbb{AG}(M_2))$. Let consider $\mathbb{M}_0(M_2) = \{\{0, y\}, \{0, x, z\}\}$. By Theorem 6.4, we get another lower bound, $\mid \mathbb{M}_0(M_2) \mid = 2 \leq \chi(\mathbb{AG}(M_2))$. Obviously, this bound is more tighter and $\chi(\mathbb{AG}(M_2)) = 2$ from Figure 1.
- Let M_4 be the MV-algebra in Example 5.5. Then, $ann(M_4\backslash\{1\}) = \langle 0\rangle$. Using Theorem 6.3, we have that $0 \leq \chi(\mathbb{AG}(M_4))$. Consider the set of all 0-minimal ideals of M_4. From Example 5.5 we have that $\mathbb{M}_0(M_4) = \{P_1, P_2, P_4\}$ which corresponds a clique $\overline{\mathbb{AG}(M_4)}$ of $\mathbb{AG}(M_4)$. It follows from Theorem 6.4 that $\mid \mathbb{M}_0(M_4) \mid = 3 \leq \chi(\mathbb{AG}(M_4))$. In fact, $\chi(\mathbb{AG}(M_4)) = 3$ from Figure 4.
- Let $\mathbf{B}_8$ be the 8-element Boolean algebra in Example 5.7 such that $ann(\mathbf{B}_8\backslash\{1\}) = \langle 0\rangle$. Similarly, $0 \leq \chi(\mathbb{AG}(\mathbf{B}_8))$. The set $\mathbb{M}_0(\mathbf{B}_8) = \{\langle v\rangle, \langle w\rangle, \langle z\rangle\}$ of $\mathbf{B}_8$ forms a 3-vertex clique of $\mathbb{AG}(\mathbf{B}_8)$. By Theorem 6.4, we have $\mid \mathbb{M}_0(\mathbf{B}_8) \mid = 3 \leq \chi(\mathbb{AG}(\mathbf{B}_8))$. Note that this bound is perfect as $\chi(\mathbb{AG}(\mathbf{B}_8)) = 3$ by Figure 6 in this case.

Finally, we use $\Gamma(A)$ to investigate the upper bound for $\chi(\mathbb{AG}(A))$.

Theorem 6.6. *Let A be an MV-algebra and $\chi(\Gamma(A))$ be finite. Then* $\chi(\mathbb{AG}(A)) \leq 2^{\chi(\Gamma(A))} - 1$.

Proof. Let $\chi(\Gamma(A)) = m$. We define a mapping $f : V(\Gamma(A)) \longrightarrow \{1, 2, ..., m\}$ which is a proper vertex coloring of $\Gamma(A)$. That is, we assign the $f(x)$th color to the vertex x of $\Gamma(A)$ and there are different colore between any two adjacent vertices. We use $A_i = f^{-1}(i)$ to denote the set of all vertices of $\Gamma(A)$ which are assigned the ith color, where $i \in \{1, 2, ..., m\}$. Now, we define a new mapping $g : V(\mathbb{AG}(A)) \longrightarrow \mathcal{P}^*(\{1, 2, ..., m\})$ by $g(I) = \{i \in \{1, 2, ..., m\} \mid I \cap A_i \neq \emptyset\}$, where $I \in V(\mathbb{AG}(A))$ and $\mathcal{P}^*(\{1, 2, ..., m\})$ is the set of all nonempty subsets of $\{1, 2, ..., m\}$.

Next, let $I, J \in V(\mathbb{AG}(A))$ such that $I \neq J$ and $IJ = \langle 0 \rangle$. Assume that $g(I) = g(J)$. Then, $\{i \in \{1, 2, ..., m\} \mid I \cap A_i \neq \emptyset\} = \{i \in \{1, 2, ..., m\} \mid J \cap A_i \neq \emptyset\}$. We claim that $I \cap A_i = J \cap A_i$ for all $i \in g(I) = g(J)$. In fact, let $x \in I \cap A_i$, and $i \in g(I) = g(J)$. For $i \in g(J) = \{i \in \{1, 2, ..., m\} \mid J \cap A_i \neq \emptyset\}$, there exists $y \in J \cap A_i$. So we have that $x \odot y = 0$ by $IJ = \langle 0 \rangle$. Since $f(x) = f(y) = i$ and f is proper, we have that $x = y \in J \cap A_i$. Thus, $I \cap A_i \subseteq J \cap A_i$ for any $i \in g(I) = g(J)$. Similarly, we also get that $J \cap A_i \subseteq I \cap A_i$. Therefore, $I \cap A_i = J \cap A_i$ for any $i \in g(I) = g(J)$.

Let $j \in \{1, 2, ..., m\} \backslash g(I)$. It follows that $I \cap A_j = J \cap A_j = \emptyset$. That is, there are no elements of I or J can be assigned by the jth color. In addition, since f is a proper vertex coloring of $\Gamma(A)$ and $\chi(\Gamma(A)) = m$, we have that $A_i \cap A_j = \emptyset$ $(i \neq j)$, $A_i \neq \emptyset$ and $\bigcup A_i = A \backslash \{0, 1\}$ for all $i, j \in \{1, 2, ..., m\}$. Thus, $A_1, A_2, ..., A_m$ are a partition of $A \backslash \{0, 1\}$. So $I = J$, which is a contradiction. Therefore, $g(I) \neq g(J)$. That is to say, g is a proper vertex coloring of $\mathbb{AG}(A)$. Thus, we conclude that $\chi(\mathbb{AG}(A)) \leq\mid \mathcal{P}^*(\{1, 2, ..., m\}) \mid= 2^m - 1 = 2^{\chi(\Gamma(A))} - 1$. □

Example 6.7.

• Let M_4 be the MV-algebra in Example 5.5. From Figure 3, we get that $\chi(\Gamma(M_4)) = 3$. Theorem 6.6 gives an upper bound for $\chi(\mathbb{AG}(M_4))$, $\chi(\mathbb{AG}(M_4)) \leq 2^{\chi(\Gamma(M_4))} - 1 = 2^3 - 1 = 7$. From Figure 4 we see that $\chi(\mathbb{AG}(M_4)) = 3$.

• Consider the 8-element Boolean algebra $\mathbf{B}_8$ in Example 5.7. By Figure 7, $\chi(\Gamma(\mathbf{B}_8)) = 3$. Then, Theorem 6.6 gives $\chi(\mathbb{AG}(\mathbf{B}_8)) \leq 2^{\chi(\Gamma(\mathbf{B}_8))} - 1 = 2^3 - 1 = 7$. From Figure 6, $\chi(\mathbb{AG}(\mathbf{B}_8)) = 3$.

• Let $M_2 = \{0, x, y, z, w, 1\}$ be the MV-algebra in Example 3.8. We have that $\chi(\Gamma(M_2)) = 2$ by Figure 2. It must be that $\chi(\mathbb{AG}(M_2)) \leq 2^{\chi(\Gamma(M_2))} - 1 = 2^2 - 1 = 3$ by Theorem 6.6. This bound is still imperfect. Since $\mathbb{AG}(M_2)$ is a 2-vertex complete graph, we have $\chi(\mathbb{AG}(M_2)) = 2 < 3$.

By Definition 3.1 we see that $\mathbb{AG}(A)$ is a simple graph. Thus, there is no loops and multiple edges in $\mathbb{AG}(A)$. It follows that $g(\mathbb{AG}(A)) \in \{3, 4, \infty\}$ by Theorem 3.2.

Theorem 6.8. *Let A be an MV-algebra such that $\mathbb{AG}(A)$ contains cycles. If* $\mid \mathbb{M}_0(A) \mid \geq 3$*, then* $g(\mathbb{AG}(A)) = 3$.

Proof. From the proof of Theorem 6.4, $\mathbb{M}_0(A)$ corresponds to a clique of $\mathbb{AG}(A)$. For any n-vertex complete graph K_n, we have that $g(K_n) = 3$, where $n \geq 3$. Therefore, we have that $g(\mathbb{AG}(A)) = 3$ by Theorem 3.2. □

Consider the MV-algebra M_2 in Example 3.8. We have that $\mid \mathbb{M}_0(M_2) \mid = 2$. From Figure 1, it is easy to get that $g(\mathbb{AG}(M_2)) = \infty$. Moreover, consider the MV-algebra M_4 in Example 5.5 and the Boolean algebra $\mathbf{B}_8$ in Example 5.7 (1). We obtain that $\mathbb{M}_0(M_4) = \{P_1, P_2, P_4\}$ and $\mathbb{M}_0(\mathbf{B}_8) = \{\langle v\rangle, \langle w\rangle, \langle z\rangle\}$. Theorem 6.8 gives that $g(\mathbb{AG}(M_4)) = g(\mathbb{AG}(\mathbf{B}_8)) = 3$. In fact, it follows from Figure 4 that there is only one cycle $P_1 - P_2 - P_4 - P_1$ in $\mathbb{AG}(M_4)$. By Figure 6, there exists a unique cycle $\langle v\rangle - \langle w\rangle - \langle z\rangle - \langle v\rangle$ in $\mathbb{AG}(\mathbf{B}_8)$.

7 Conclusion

In this paper, for an MV-algebra $(A, \oplus, *, 0)$, we introduce and study the annihilating-ideal graph $\mathbb{AG}(A)$ by using the annihilator $ann(A\backslash\{1\}) = \{a \in A \mid a \odot b = 0$ for all $b \in A\backslash\{1\}\}$ of $A\backslash\{1\}$. We show that $\mathbb{AG}(A)$ is a connected graph and $d_{max}(\mathbb{AG}(A)) \leq 3$. If there is a cycle in $\mathbb{AG}(A)$, then we show that $g(\mathbb{AG}(A)) \leq 4$. We prove that $\mathbb{AG}(A)$ is null if $ann(A\backslash\{1\}) = A\backslash\{1\}$. If $A = I \oplus J$ and $I \cap J = \{0\}$, we show that $\mathbb{AG}(A)$ is a star graph, where I is a 0-minimal ideal of A and J contains no nonzero zero-divisors as an MV-algebra. In addition, we study the annihilating-ideal graphs of Boolean algebras which are star graphs or complete graphs. We show that $\langle x\rangle \in V(\mathbb{AG}(A))$ and $\langle x\rangle$ is unique and proper for any $x \in A\backslash\{0, 1\}$. We prove that there is a subgraph of $\mathbb{AG}(A)$ which is isomorphic to $\Gamma(A)$. Also, we study the vertex coloring and girth of $\mathbb{AG}(A)$. We get two lower bounds $\mid \mathbb{I}_{ann} \mid, \mid \mathbb{M}_0(A) \mid$ and an upper bound $2^{\chi(\Gamma(A))} - 1$ for $\chi(\mathbb{AG}(A))$, where $\mathbb{I}_{ann}$ is the set of all nonzero ideals of A which contain in $ann(A\backslash\{1\})$, and $\mathbb{M}_0(A)$ is the set of all 0-minimal ideals of A.

References

[1] A. Gan, Y. Yang, The zero-divisor graphs of MV-algebras, *Soft Computing*, **24** (2020), 6059–6068.

[2] A. Gan, Y. Yang, Annihilator graphs of MV-algebras, *Journal of Algebra and its Applications*, **20** (2021), 2150188-1–2150188-18.

[3] A. Bondy, U. S. R. Murty, *Graph theory*, London, 2007.

[4] B. A. Rather, S. Pirzada, T. A. Naikoo, Y. Shang, On laplacian eigenvalues of the zero-divisor graph associated to the ring of integers modulon, *Mathematics*, **9** (2021), 482–499.

[5] C. C. Chang, Algebraic analysis of many valued logics, *Transactions of the American Mathematical society*, **88** (1958), 467–490.

[6] C. Tian, L. Sun, Partitioning planar graphs with girth at least 9 into an edgeless graph and a graph with bounded size components, *Mathematical Modelling and Control*, **1** (2021), 136–144.

[7] D. F. Anderson, P. S. Livingston, The zero-divisor graph of a commutative ring, *Journal of Algebra*, **217** (1999), 434–447.

[8] F. R. DeMeyer, T. McKenzie, K. Schneider, The zero-divisor graph of a commutative semigroup, *Semigroup Forum*, **65** (2002), 206–214.

[9] F. DeMeyer, L. DeMeyer, Zero divisor graphs of semigroups, *Journal of Algebra*, **283** (2005), 190–198.

[10] G.-M. Greuel, G. Pfister, O. Bachmann, C. Lossen, H. Schönemann, *A singular introduction to commutative algebra*, Berlin, 2002.

[11] H. Liu, The graph of equivalence classes of zero-divisors of a poset, *Glasnik Matematicki*, **49** (2014), 263–273.

[12] J. M. Howie, *Fundamentals of semigroup theory*, New York, 1995.

[13] J. A. Bondy, U. S. R. Murty, et al, *Graph theory with applications*, Macmillan London, 1976.

[14] L. DeMeyer, A. Schneider, The annihilating-ideal graph of commutative semigroups, *Journal of Algebra*, **469** (2017), 402–420.

[15] M. Behboodi, Z. Rakeei, The annihilating-ideal graph of commutative rings I, *Journal of Algebra and its Applications*, **10** (2011), 727–739.

[16] M. Behboodi, Z. Rakeei, The annihilating-ideal graph of commutative rings II, *World Scientific*, **10** (2011), 741–753.

[17] M. Atiyah, G. MacDonald, *Introduction to commutative algebra*, CRC Press, 2015.

[18] P. L. Belluce, S. Sessa , Minimal ideals and the socle in MV-algebras*, *Lectures on soft computing and fuzzy logic*, (2001), 19–31.

[19] R. K. Nath, M. Sharma, P. Dutta, Y. Shang, On r-noncommuting graph of finite rings, *Axioms*, **10** (2021), 233–247.

[20] R. L. Cignoli, I. M. D'Ottaviano, D. Mundici, *Algebraic foundations of many-valued reasoning*, Springer Science & Business Media, Dordrecht, 2013.

[21] R. J. Wilson, *Introduction to graph theory*, Pearson Education India, 1979.

Received May 2023

Comments on Yablo's Construction

Karl Schlechta
Koppeweg 24, D-97833 Frammersbach, Germany
schcsg@gmail.com

Abstract

We analyse Yablo's coding of the liar paradox by infinite acyclic graphs, [6], and show that his construction is in a certain way minimal.

This leads to a very limited representation result.

We then argue that the "right" level of description of more complicated constructions is probably the level of paths, and not of single arrows.

We also look at more complicated cells of basic contradiction ("diamonds"), and show that, at least under certain conditions, they fail to achieve Yablo's result.

1 Introduction

1.1 Overview

Yablo's idea (upon which all of the present material is based), appeared in [6].

The author's interest was kindled by [2], commented in [3], and extended in [4]. (It was shown there that conjecture 15 in [2] is wrong.)

We liberally use material from [4], in particular in Section 1, Section 2 and Section 3.

Our approach is "bottom-up". We analyse Yablo's construction, an intricate composition of contradictions in "Yablo Cells", of the graph form $x \not\twoheadrightarrow y$, $y \not\twoheadrightarrow z$, $x \not\twoheadrightarrow z$, meaning $x = \neg y \wedge \neg z$, $y = \neg z$. We distinguish y and z, calling y the knee of the cell, z the foot (and x the head). In Yablo's construction, y and z will be the head of new cells (among other arrows and constructions), etc.

Thus, x is contradictory if true (as a formula), and if x is false, all arrows starting at x will lead to new contradictions (as then y and z will be true). We take this

Retired, formerly: Aix-Marseille Université, CNRS, LIF UMR 7279, F-13000 Marseille, France

as construction principle, and show that the property is recursively true, i.e., will hold for y and z, etc., too. We can base Yablo's and similar constructions on this principle. In particular, Yablo's construction is minimal to satisfy these principles.

It is natural to look now at alternative contradictory cells as building blocks. We discuss several candidates, the only interesting one is the "diamond" of the form $x \not\twoheadrightarrow y$, $y \not\twoheadrightarrow z$, $x \not\twoheadrightarrow y'$, $y' \rightarrow z$, with the meaning $x = \neg y \wedge \neg y'$, $y = \neg z$, $y' = z$. But the recursive construction, appending new diamonds at y and y' fails, at least under the restriction of using only conjunctions for the logical form in the positive case (x true).

We also

- argue that the right level of abstraction is to consider paths, not single arrows,
- give some examples of Yablo-like graphs with more general formulas, i.e. of disjunctive normal forms, and different basic orders (i.e. not necessarily that of the natural numbers),
- illustrate postponing contradictions.

1.1.1 Related Work

The ideas in [1] and [5] are more "top down", whereas our ideas are more "naive", starting from an analysis of Yablo's construction and the basic contradictions. Thus, we needed an argument to consider paths, and not combinations of arrows as the "right" level of abstraction, whereas paths are the basic level of investigation in above articles.

The closest common points are perhaps in Section 4.5 of [1]. In addition, our Section 6 can be seen as an allusion to a game theoretic interpretation.

1.2 More Detailed Overview

Some aspects of Yablo's construction are hidden behind its elegance. Our perhaps somewhat pedantic analysis leads to the concepts of "head", "knee", and "foot", and then to "saw blades" in Section 3.

Section 2 generalizes Yablo's construction to arbitrary formulas of the type $\bigvee \bigwedge$ (disjunctive normal forms), and offers a number of easy variations of such structures by modifying the order of the graph.

Section 3 uses our idea of finer analysis of contradictory cells to build somewhat different structures - though the distinction is blurred by the necessarily recursive and infinite construction of contradictions. This is used in our limited (semi) representation results in Section 3.3 and Section 4.4.

In Section 4, we show that we can make contradictions arbitrarily complicated when seen on the level of arrows. This leads us to think that the level of arrows is too detailed, it is not a useful level of abstraction. We therefore work with paths, trivialising some intermediate nodes.

In Section 5 we look at various candidates for basic contradiction cells, one of them (Diamond) seems much more complicated, and embedding it into a recursive construction as in Yablo's work leads to not so obvious problems, unsolvable at least in the present framework.

In Section 6, we analyse the finite procrastination of contradictions using Yablo's form of contradictions.

The author of the present text did not study the literature systematically. So, if some examples are already discussed elsewhere, the author would ask to be excused for not quoting previous work.

1.2.1 General Strategy

Our general strategy (as in Yablo's original paper) will be as follows:

(1) We want to show that $x = TRUE$ is impossible, and $x = FALSE$ is impossible, too.

(2) Say x is the root of a graph, so for every possibility in this graph, $x = TRUE$ is impossible, as every set of paths from x leads to a contradiction in case $x = TRUE$.

(3) Conversely, if $x = FALSE$, then every path from x leads leads to some x' such that $x' = TRUE$ and this is impossible for the same reasons as above for $x = TRUE$.

(Thus, the construction is essentially recursive, see Section 3.3.)

(4) Moreover, we will not postpone contradictions, but e.g. in above case for $x = TRUE$ we will require the graph to branch at x (and not to be some kind of preamble, where we first have to go to some x'' where the branching happens). - This is a minor point.

See Section 6 for some exceptions.

(5) Yablo uses conjunctions of negated propositional variables, $x = \bigwedge \neg x_i$, this takes care of above ideas, conjunctions provide the bases for contradictions, and the negated x makes at least one x_i positive, this takes care of making $x = FALSE$ impossible.

(Graphically, Yablo has negative arrows from x to the x_i, we sometimes consider negative paths instead. In Section 2 we also consider mixed Or/And formulas, and in Section 5 we consider more complicated contradictions.)

1.2.2 Main Ideas of the Paper

(1) Analysis of Yablo's construction in Section 3.3, and the resulting partial representation result in Condition 4.1.

(2) Discussion of the suitable level of abstraction for the description of more complicated constructions, see Section 4.2.

(3) Discussion of more complicated basic contradiction units ("diamonds") and the result that they are unsuitable (at least under the present, limited, conditions), see Section 5.4, and in particular Section 5.4.1.

1.3 Preliminaries

We consider the problem from the graph side and the logics side.

1.3.1 The Logical Side

On the logics side, we work with propositional formulas, which may, however, be infinite.

We will see that we need infinite formulas (of infinite depth) to have nodes to which we cannot attribute truth values. See Fact 1.4.

We will work here with disjunctive normal forms, i.e. with formulas of the type $x := \bigvee\{\bigwedge x_i : i \in I\}$, where $x_i := \{x_{i,j} : j \in J_i\}$, and the $x_{i,j}$ are propositional variables or negations thereof - most of the time pure conjunctions of negations of propositional variables.

Fact 1.1.

Let $x := \bigvee\{\bigwedge\{x_{i,j} : j \in J_i\} : i \in I\}$, where the $x_{i,j}$ are propositional variables or negations thereof.

(1) Let $F := \Pi\{x_i : i \in I\}$ - the set of choice functions in the sets x_i, where $x_i := \{x_{i,j} : j \in J_i\}$.

Then $\neg x = \bigvee\{\bigwedge\{\neg x_{i,j} : x_{i,j} \in ran(f)\} : f \in F\}$.

(Argue semantically with the sets of models and general distributivity and complements of sets.)

(2) Contradictions will be between two formulas only, one a propositional variable, the other the negation of the former. □

1.3.2 The Graph Side

We work with directed, acyclic graphs. They will usually have one root, often called x_0. In diagrams, the graphs may grow upward, downward, or sideways, we will say what is meant.

Definition 1.1.

Nodes stand for propositional variables.

If a node x is not terminal, it has also a propositional formula ϕ_x attached to it, sometimes written $d(x) = \phi_x$, with the meaning $x \leftrightarrow \phi_x$, abbreviated $x = \phi_x$. The successors of x are the propositional variables occurring in ϕ_x. Thus, if $x \to x'$ and $x \to x''$ are the only successors of x in γ, ϕ_x may be $x' \vee x''$, $x' \wedge \neg x''$, but not $x' \wedge y$.

Usually, the ϕ_x are (possibly infinite) conjunctions of propositional variables or (in most cases) their negations, which we write $\bigwedge \pm x_i$ etc. We often indicate the negated variables in the graph with negated arrows, like $x \not\to y$, etc. Thus, $x \not\to x'$, $x \to x''$ usually stands for $\phi_x = \neg x' \wedge x''$.

Example 1.1.

Consider the basic construction of a contradiction (used by Yablo and here, defined "officially" in Definition 1.4).

$\Gamma := \{x \not\to y \not\to z,\ x \not\to z\}$. Γ stands for $x = \neg y \wedge \neg z$, $y = \neg z$, so $x = z \wedge \neg z$, which is impossible.

If we negate x, then $\neg x = y \vee z = \neg z \vee z$, so $\neg x$ is possible.

From the graph perspective, we have two paths in Γ from x to z, $\sigma := x \not\to y \not\to z$, and $\sigma' := x \not\to z$.

We add now $y \not\to y'$ to Γ, so $\Gamma' := \{x \not\to y \not\to z,\ x \not\to z,\ y \not\to y'\}$, thus $x = \neg y \wedge \neg z$, $y = \neg z \wedge \neg y'$, so $\neg y = z \vee y'$, and $x = (z \vee y') \wedge \neg z = (z \wedge \neg z) \vee (y' \wedge \neg z)$, and x is not contradictory any more.

Definition 1.2.

We can attribute a value to a path σ, $val(\sigma)$, expressing whether it changes a truth value from the beginning to the end. $\sigma := x \not\to y \not\to z$ does not change the value of z compared to that of x, $\sigma' := x \not\to z$ does. We say $val(\sigma) = +$, $val(\sigma') = -$, or positive (negative) path.

Formally:

Let σ, σ' be paths as usual.

(1) If $\sigma := a \to b$, then $val(\sigma) = +$, if $\sigma := a \not\to b$, then $val(\sigma) = -$.

(2) Let $\sigma \circ \sigma'$ be the concatenation of σ and σ'. Then $val(\sigma \circ \sigma') = +$ iff $val(\sigma) = val(\sigma')$, and - otherwise.

If all arrows are negative, then $val(\sigma) = +$ iff the length of σ is even.

Definition 1.3.
We call two paths σ, σ' with common start and end contradictory, and the pair a contradictory cell iff $val(\sigma) \neq val(\sigma')$. The structures considered here will be built with contradictory cells.

Remark 1.2.

(1) Note that the fact that σ and σ' are contradictory or not is independent of how we start, whether $x = TRUE$ or $x = FALSE$.

(2) We saw already in Example 1.1 that it is not sufficient for a "real" contradiction to have two contradictory paths.

We need

(2.1) (somewhere) an "AND", so both have to be valid together, an "OR" is not sufficient,

(2.2) we must not have a branching with an "OR" on the way as in Γ', an "escape" path, unless this leads again to a contradiction.

See also Definition 3.1.

Notation 1.1.

(1) When we give nodes a truth value, we will use $x+$ (and $x \bigwedge$, $x + \bigwedge$, etc. if ϕ_x has the form $\bigwedge \pm x_i$) to denote the case $x = TRUE$, and $x-$, $x \bigvee$, $x - \bigvee$, etc. for the case $x = FALSE$.

(2) We sometimes use the notation $x \oplus$ for shorthand that $x+$ is contradictory, and $x \ominus$ for shorthand that every path (mostly: every arrow) from x will lead to some y such that $y \oplus$, and $y+$ if $x-$.

Fact 1.3.
(Simplified).
Consider three paths, ρ, σ, τ, for simplicity with same origin, i.e. $\rho(0) = \sigma(0) = \tau(0)$.

(1) No contradiction loops of length 3.

(1.1) Suppose they meet at a common point, i.e. $\rho(m_\rho) = \sigma(m_\sigma) = \tau(m_\tau)$. Then it is impossible that ρ contradicts σ contradicts τ contradicts ρ (at m_ρ). ("α contradicts β" means here that for some i, j $\alpha(i) = \beta(j)$, but one has value +, the other value -.)

(Trivial, we have only 2 truth values).

(1.2) Suppose, first ρ and σ meet, then ρ (or σ) and τ meet, but once they meet, they will continue the same way (e.g., if $\rho(i) = \sigma(j)$, then for all $k > 0$ $\rho(i+k) = \sigma(j+k)$). Then it is again impossible that ρ contradicts σ contradicts τ contradicts ρ. (ρ and σ continue to be the same but with different truth values until they meet τ, so it is the same situation as above.)

(2) Above properties generalize to any loops of odd length (with more paths).

See [4] for more details.

This does not hold when the paths may branch again after meeting, as the next Example shows.

Example 1.2.

(Example 7.4.2 in [4].)

Let $\sigma_0 : x_0 \not\twoheadrightarrow x_1 \rightarrow x_2 \not\twoheadrightarrow x_3 \rightarrow x_4$, $\sigma_1 : x_0 \not\twoheadrightarrow x_1 \rightarrow x_2 \rightarrow x_4$, $\sigma_2 : x_0 \rightarrow x_2 \not\twoheadrightarrow x_3 \rightarrow x_4$, $\sigma_3 : x_0 \rightarrow x_2 \rightarrow x_4$,

then σ_0 contradicts σ_1 in the lower part, σ_2 and σ_3 in the upper part, σ_1 contradicts σ_2 and σ_3 in the upper part, σ_2 contradicts σ_3 in the lower part.

Obviously, this may be generalized to 2^ω paths.

Consider Yablo's original construction:

Example 1.3.

Let the nodes be $\{x_i : i < \omega\}$, and the arrows for x_i $\{x_i \not\twoheadrightarrow x_j : i < j\}$, expressed as a relation by $\{x_i < x_j : i < j\}$, and as a logical formula by $x_i = \bigwedge\{\neg x_j : i < j\}$.

Thus $\neg x_i = \bigvee\{x_j : i < j\}$. For any x_i, we have a contradiction by $x_i = \bigwedge\{\neg x_j : i < j\}$ and $\neg x_{i+1} = \bigvee\{x_j : i+1 < j\}$ for any x_i+, and for any x_k- for a suitable $k' > k$ $x_{k'}+$.

It is important that, although we needed to show the property $\ominus$ for x_0 only, it holds for all x_i, thus it is a recursive construction.

1.3.3 Interplay of the Graph and the Logical Side

We can either think on the logical level with formulas, or graphically with conflicting paths, as in the following Fact.

We need infinite depth and width in our constructions:

Fact 1.4.

(1) The construction needs infinite depth,

(2) the logic as used in Yablo's construction is not compact,

(3) the construction needs infinite width, i.e. the logic cannot be classical.

Proof

(1) Let x_i be a minimal element, then we can chose an arbitrary truth value x_i, and propagate this value upwards. If there are no infinite descending chains, we can do this for the whole construction.

(2) The logic as used in Yablo's construction is not compact:

Trivial. (Take $\{\bigvee\{\phi_i : i \in \omega\}\} \cup \{\neg\phi_i : i \in \omega\}$. This is obviously inconsistent, but no finite subset is.).

(3) It is impossible to construct a Yablo-like structure with classical logic:

Take an acyclic graph, and interpret it as in Yablo's construction. Wlog., we may assume the graph is connected. Suppose it shows that x_0 cannot be given a truth value. Then the set of formulas showing this does not have a model, so it is inconsistent. If the formulas were classical, it would have a finite, inconsistent subset, Φ. Define the depth of a formula as the shortest path from x_0 to this formula. There is a (finite) n such that all formulas in Φ have depth $\leq n$. Give all formulas of depth n (arbitrary) truth values, and work upwards using truth functions. As the graph is acyclic, this is possible. Finally, x_0 has a truth value.

Thus, we need the infinite $\bigwedge / \bigvee$. □

1.3.4 Further Remarks on Contradictory Structures

See Section 7.3 in [4] for more material.

Example 1.4.

We consider now some simple, contradictory cells. They should not only be contradictory for the case $x+$, but also be a potential start for the case $x-$, without using more complex cells. (Otherwise, we postpone the solution, and may forget the overly simple start.)

We will consider in Section 5 a more complicated contradiction cell ("Diamond"), and discuss here only simple cells.

For this, we order the complexity of the cases by (1) < (2.1) < (2.3) below, (2.2) is not contradictory, so it is excluded.

See Diagram 1.1, center part.

(1) The cell with 2 arrows.

It corresponds to the formula $x = y \wedge \neg y$, graphically, it has a positive and a negative arrow from x to y, so exactly one of α and β is negative.

If x is positive, we have a contradiction.

If x is negative, however, we have a problem. Then, we have $x = y \vee \neg y$. Let α be the originally positive path, β the originally negative path. Note that α is now negative, and β is positive. The α presents no problem, as y is positive, and we can append the same construction to y, and have a contradiction. However, β has to lead to a contradiction, too, and, as we will not use more complicated cells, we face the same problem again, y is negative. So we have an "escape path", assigning $\perp$ to all elements in one branch.

(Consider $x_0 \Rightarrow_\neg x_1 \Rightarrow_\neg x_2 \Rightarrow_\neg x_3 \ldots$, setting all $x_i := -$ is a consistent valuation. So combining this cell with itself does not result in a contradictory structure.)

Of course, appending at y a Yablo Cell (see below, case (2.3)) may be the beginning of a contradictory structure, but this is "cheating", we use a more complex cell.

(2) Cells with 3 arrows.

Note that the following examples are not distinguished in the graph!

Again, we want a contradiction for x positive, so we need an $\wedge$ at x, and have the possibilities (up to equivalence) $x = x' \wedge y$, $x = x' \wedge \neg y$, $x = \neg x' \wedge y$, $x = \neg x' \wedge \neg y$.

Again, if x is negative, all paths $\alpha : x \to y$, $\beta : x \to x'$ (or $\beta\gamma : x \to x' \to y$) have to lead to a contradiction.

(2.1) one negative arrow, with $x = \pm x' \wedge \pm y$

(2.1.1) $x \to x' \not\to y$, $x \to y$, corresponding to $x = x' \wedge y$, so if x is negative, this is $\neg x' \vee \neg y$.
But, at y, this possible paths ends, and we have the same situation again, with a negative start, as in Case (1).

(2.1.2) $x \to x' \to y$, $x \not\to y$
Here, we have again a positive path to y, through x', so for x negative both x' and y will be negative, and neither gives a start for a new contradiction.

(2.1.3) $x \not\to x' \to y$, $x \to y$
This case is analogous to case (2.1.1).

(Similar arguments apply to more complicated cells with an even number of negative arrows until the first branching point - see Remark 1.5 below.

(2.2) 2 negative arrows: not contradictory.

(2.3) The original type of contradiction in Yablo's construction
$x \not\to x' \not\to y$, $x \not\to y$.
This will be discussed in detail in the rest of the paper. But we see already that both paths, $x \not\to x'$ and $x \not\to y$ change sign, so x' and y will be positive, appending the same type of cell at x' and y solves the problem (locally), and offers no escape.

Definition 1.4.

(1) We call the contradiction of the type (2.3) of Example 1.4, i.e. $x \not\to x' \not\to y$, $x \not\to y$, a Yablo Cell, YC, and sometimes x its head, y its foot, and x' its knee.

(2) We sometimes abbreviate a Yablo Cells simply by ∇, without going into any further details.

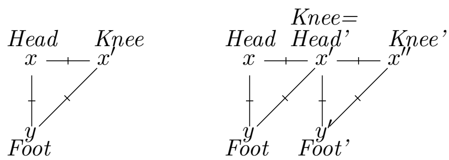
If we combine Yablo Cells, the knee for one cell may become the head for another Cell, etc.

See Diagram 1.1, upper part.

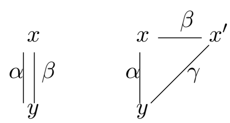
Diagram 1.1. *Simple Contradictions*

Horizontal lines indicate arrows pointing to the right,
the other lines downward pointing arrows

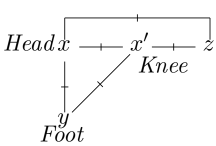
Yablo Cells



Contradictory Cells



See Remark 1.5



Remark 1.5.

(1) The distinction between x' and y, i.e. between knee and foot, is very important. In the case $x+$, we have at y a complete contradiction, at x', we have not yet constructed a contradiction. Thus, if we have at x' again an $\wedge$ (as at x), this

becomes an $\vee$, and we have to construct a contradiction for all $x' \to z$ (or $x' \not\twoheadrightarrow z$), not only for $x' \not\twoheadrightarrow y$. Otherwise, we have an escape possibility for $x+$. Obviously, the contradiction need not be immediate at z, the important property is that ALL paths through all z lead to a contradiction, and the simplest way is to have the contradiction immediately at z - as in Yablo's construction, and our saw blades.

See Diagram 1.1, lower part.

See also the discussion in Section 3.3 (page 25) and the construction of saw blades in Section 3.2.

(2) As we work for a contradiction in the case x-, too, the simplest way to achieve this is to have a negative arrow $x \not\twoheadrightarrow x'$, and at x' again an $\wedge$. This gives a chance to construct a contradiction at x'. Of course, we have to construct a contradiction at y, too, as in the case $x-$, we have an $\vee$ at x.

Of course, we may have branches originating at x', which all lead to contradictions in the case $x-$, so $x \to x'$ (resp. $\vee$ at x') is possible, too.

But, for the simple construction, we need $\not\twoheadrightarrow$ between x and x', and $\wedge$ at x'. And this leads to the construction of contradictions for all $x' \to z$ (or $x' \not\twoheadrightarrow z$) as just mentioned above.

Example 1.5.

This Example shows that infinitely many finitely branching points cannot always replace infinite branching - there is an infinite "procrastination branch" or "escape branch". This modification of the Yablo structure has one acceptable valuation for Y_1 :

Let Y_i, $i < \omega$ as usual, and introduce new X_i, $3 \leq i < \omega$.
Let $Y_i \not\twoheadrightarrow Y_{i+1}$, $Y_i \to X_{i+2}$, $X_i \not\twoheadrightarrow Y_i$, $X_i \to X_{i+1}$, with
$Y_i := \neg Y_{i+1} \wedge X_{i+2}$, $X_i := \neg Y_i \wedge X_{i+1}$.
If $Y_1 = \top$, then $\neg Y_2 \wedge X_3$, by X_3, $\neg Y_3 \wedge X_4$, so, generally,
if $Y_i = \top$, then $\{\neg Y_j : i < j\}$ and $\{X_j : i+1 < j\}$.
If $\neg Y_1$, then $Y_2 \vee \neg X_3$, so if $\neg X_3$, $Y_3 \vee \neg X_4$, etc., so, generally,
if $\neg Y_i$, then $\exists j(i < j, Y_j)$ or $\forall j\{\neg X_j : i+1 < j\}$.

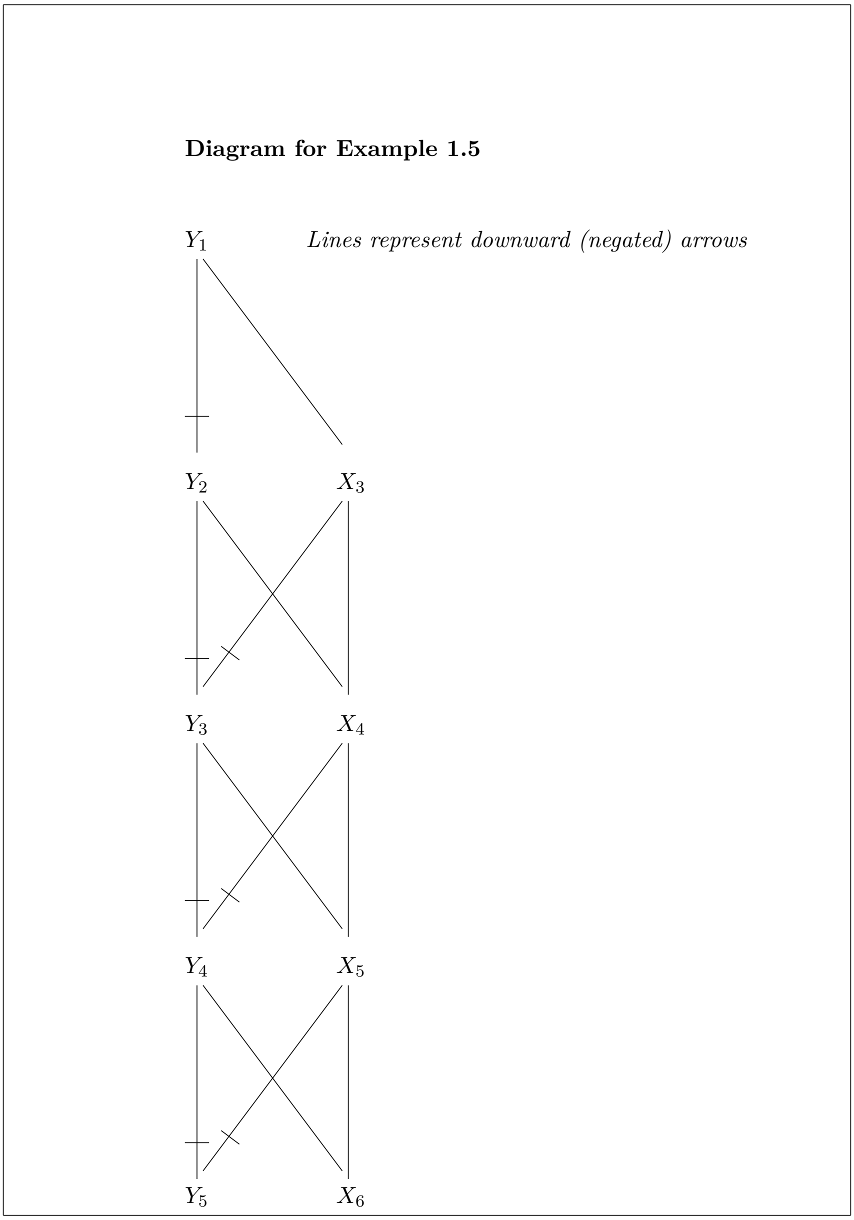
Suppose now $Y_1 = \top$, then X_j for all $2 < j$, and $\neg Y_j$ for all $1 < j$. By $\neg Y_2$ there is j, $2 < j$, and Y_j, a contradiction, or $\neg X_j$ for all $3 < j$, again a contradiction.

But $\neg Y_1$ is possible, by setting $\neg Y_i$ and $\neg X_i$ for all i.

Thus, replacing infinite branching by an infinite number of finite branching does not work for the Yablo construction, as we can always chose the "procrastinating" branch.

See Diagram 1.2.

Diagram 1.2. *Escape Path*



2 A Generalization of Yablo's Construction to Formulas in Disjunctive Normal Form - Examples

2.1 Introduction

We discuss here a general strategy and a number of examples (in Example 2.1).

They have in common that the formulas are of the type $\bigvee \bigwedge$, i.e. in disjunctive normal form. The limiting cases are pure conjunctions (as in Yablo's original approach) or pure disjunctions.

The examples are straightforward generalizations of Yablo's construction, as we have here several columns, in Yablo's construction just one column, and our choice functions g (in all columns) correspond to the choice of one element in Yablo's construction.

Consider the first example below.

More precisely, as Yablo works with $\bigwedge x_i$, there is one uniform set of x_i. We work with $\bigvee \bigwedge x_{i,j}$, so we have to distinguish the elements in the $\bigwedge$ from the sets in $\bigvee$. We define for this purpose columns, whose elements are the elements in the $\bigwedge$, and the set of columns are the sets in $\bigvee$. Negation is now slightly more complicated, not just OR of negated elements, but OR of choice functions of negated elements in the columns. We also need some enumeration of $\omega \times \omega$, or of a suitable set.

It is perhaps easiest to see the following example geometrically. We have a vertical column C_{i_0} where a certain property holds, and a horizontal line (the choice function g), where the opposite holds, and column and horizontal line meet.

More precisely, the property will not necessarily hold in all of C_{i_0}, but only from a certain height onward. As the choice functions g will chose even higher in the columns, the clash is assured.

For more details of the general strategy, see Definition 2.1, Case 5. and Example 2.1, in particular Case 6.

2.2 The Examples

We first introduce some notation and definitions.

Definition 2.1.

(1) The examples will differ in the size of the $\bigwedge$ and $\bigvee$ - where the case of both finite is trivial, no contradictions possible - and the relation in the graph, i.e. which formulas are "visible" from a given formula. (In Yablo's construction, the x_j, $j > i$, are visible from x_i.) We will write the relation by $\not\twoheadrightarrow$, $x \not\twoheadrightarrow y$, but for simplicity sometimes $x < y$, too, reminding us that variables inside

the formulas will be negated. We consider linear and ranked orders, leaving general partial orders aside. All relations will be transitive and acyclic.

For the intuition (and beyond) we order the formulas in sets of columns. Inside a column, the formulas are connected by $\bigwedge$, the columns themselves are connected by $\bigvee$. This is the logical ordering, it is different from above order relation in the graph.

(2) The basic structure.

(2.1) So, we have columns C_i, and inside the columns variables $x_{i,j}$. As we might have finitely or countably infinitely many columns, of finite or countably infinite size, we write the set of columns $C := \{C_i : i < \alpha\}$, where $\alpha < \omega + 1$, and $C_i := \{x_{i,j} : j < \beta_i\}$, where $\beta_i < \omega + 1$ again. By abuse of language, C will also denote the whole construction, $C := \{x_{i,j} : i, j < \omega\}$.

(2.2) Given C and $x_{i,j}$, $C \restriction x_{i,j} := \{x_{i',j'} : x_{i,j} \not\twoheadrightarrow x_{i',j'}\}$, the part of C visible from $x_{i,j}$.

(2.3) Likewise, $C_{i'} \restriction x_{i,j} := (C \restriction x_{i,j}) \cap C_{i'}$.

(3) Let $I(x_{i,j}) := \{i' < \alpha : C_{i'} \restriction x_{i,j} \neq \emptyset\}$. (In some i', there might be no $x_{i',j'}$ s.t. $x_{i,j} \not\twoheadrightarrow x_{i',j'}$.)

Given $i' \in I(x_{i,j})$, let $J(i', x_{i,j}) := \{j' < \beta_{i'} : x_{i,j} \not\twoheadrightarrow x_{i',j'}\}$. By $i' \in I(x_{i,j})$, $J(i', x_{i,j}) \neq \emptyset$.

(4) Back to logic. Let $x_{i,j} := \bigvee\{\bigwedge\{\neg x_{i',j'} : j' \in J(i', x_{i,j})\} : i' \in I(x_{i,j})\}$, ($\wedge$ inside columns, $\vee$ between columns). We will sometimes abbreviate $d(x_{i,j})$ by $x_{i,j}$. (Note that the $x_{i',j'}$ are exactly the elements visible from $x_{i,j}$.)

(4.1) If $x_{i,j}$ is true (written $x_{i,j}+$), then all elements in one of the $C_{i'}$, $i' \in I(x_{i,j})$, and visible from $x_{i,j}$, must all be false - but we do not know in which $C_{i'}$. We denote this $C_{i'}$ by $C_{i(x_{i,j})}$, and define $C[x_{i,j}] := C_{i(x_{i,j})} \restriction x_{i,j}$.

(4.2) Conversely, suppose $x_{i,j}$ is false, $x_{i,j}-$. Again, we consider only elements in $C \restriction x_{i,j}$, i.e. visible from $x_{i,j}$. By distributivity, $x_{i,j}- = \bigvee\{\bigwedge ran(g) : g \in \Pi\{C_{i'} \restriction x_{i,j} : i' \in I(x_{i,j})\}\}$.

(g is a choice function choosing in all columns $C_{i'}$ the "sufficiently big" elements $x_{k,m}$, i.e. above $x_{i,j}$, $ran(g)$ its range or image.) See Fact 1.1.

Note that the elements of $ran(g)$ are now positive!

The $\bigvee$ in above formula choses some such function g, but we do not know which. Let $g[x_{i,j}]$ denote the chosen one.

(4.3) Note that both $C[x_{i,j}]$ and $g[x_{i,j}]$ are undefined if there are no $x_{i',j'} > x_{i,j}$.

(5) The conflicts will be between the "vertical" (negative) columns, and "horizontal" (positive) lines of the g. It is a very graphical construction. If we start with a positive point, the negative columns correspond to the $\forall$ in Yablo's construction, the g to the $\exists$, if we start with a negative one, it is the other way round.

More precisely:

(5.1) Let $x_{i,j}+$, consider $C[x_{i,j}]$ (the chosen column) - recall all elements of $C[x_{i,j}]$ are negative.

If there is $x_{i',j'} \in C[x_{i,j}]$ which is not maximal in $C[x_{i,j}]$, then $g[x_{i',j'}]$ intersects $C[x_{i,j}]$, a contradiction, as all elements of $ran(g[x_{i',j'}])$ are positive.

Of course, such non-maximal $x_{i',j'}$ need not exist. In that case, we have to try again with the negative element $x_{i',j'}$ and Case (5.2).

(5.2) Let $x_{i,j}-$, consider $g[x_{i,j}]$ (the chosen function) - recall, all elements of $ran(g[x_{i,j}])$ are positive.

If there is $x_{i'j'} \in ran(g[x_{i,j}])$ such that $C[x_{i',j'}] \cap ran(g[x_{i,j}]) \neq \emptyset$, we have a contradiction. So $x_{i,j} \not\twoheadrightarrow x_{i',j'} \not\twoheadrightarrow x_{i'',j''} \in C[x_{i',j'}]$. (This case is impossible in Yablo's original construction.)

Otherwise, we have to try to work with the elements in $ran(g[x_{i,j}])$ and Case (5.1) above, etc.

Note that we can chose a suitable $x_{i',j'} \in ran(g[x_{i,j}])$, in particular one which is not maximal (if such exist), but we have no control over the choice of $C[x_{i',j'}]$, so we might have to exhaust all finite columns, until the choice is only from a set of infinite columns.

See Diagram 2.1, upper part, and Example 2.1, Case 6.

Example 2.1.

(1) Case 1

Consider the structure $C := \{x_{i,j} : i < \alpha, j < \omega\}$, with columns $C_i := \{x_{i,j} : j < \omega\}$.

Take a standard enumeration f of C, e.g. $f(0) := x_{0,0}$, then enumerate the $x_{i,j}$ s.t. $max\{i,j\} = 1$, then $max\{i,j\} = 2$, etc. As f is bijective, $f^{-1}(x_{i,j})$ is defined.

(More precisely, let $m := max\{i,j\}$, and e.g. we go first horizontally from left to right over the columns up to column $m-1$, then in column m upwards, i.e. $x_{0,m}, \ldots, x_{m-1,m}, x_{m,0}, \ldots, x_{m,m}$.)

Define the relation $x_{i,j} \not\twoheadrightarrow x_{i',j'}$ iff $f^{-1}(x_{i,j}) < f^{-1}(x_{i',j'})$. Obviously, $\not\twoheadrightarrow$ is transitive and free from cycles. $C \upharpoonright k := \{x_{i,j} \in C : f^{-1}(x_{i,j}) > k\}$ etc. are defined.

We now show that the structure has no truth values.

Suppose $x_{i,j}+$.

Consider $C[x_{i,j}]$, let $i' := i(x_{i,j})$ and chose $x_{i',j'} \in C[x_{i,j}]$. $x_{i',j'}$ is false, $ran(g[x_{i',j'}])$ intersects $C_{i'}$ above $x_{i',j'}$, so we have a contradiction.

In particular, $x_{0,0}+$ is impossible.

Suppose $x_{0,0}-$, then $ran(g[x_{0,0}]) \neq \emptyset$, chose $x_{i,j} \in ran(g[x_{0,0}])$, so $x_{i,j}+$, but we saw that this is impossible.

Note: for $\alpha = 1$, we have Yablo's construction.

(2) Case 2

We now show that the same construction with columns of height 2 does not work, it has an escape path.

Set $x_{i,0}+$, $x_{i,1}-$ for all i. $C_{i(x_{i,0})}$ might be C_i, so $C[x_{i,0}] = \{x_{i,1}\}$, which is possible. $ran(g[x_{i,1}])$ might be $\{x_{j,0} : j > i\}$, which is possible again.

(3) Case 3

We change the order in Case 2, to a (horizontally) ranked order:

$x_{i,j} < x_{i',j'}$ iff $i < i'$ (thus, between columns), we now have a contradiction:

Consider $x_{i,0}-$ and $g[x_{i,0}]$. $g[x_{i,0}](i)$ is undefined. Let $x_{i+1,j'} := g[x_{i,0}](i+1)$, thus $x_{i+1,j'}+$. Consider $C[x_{i+1,j'}]$. This must be some $C_{i'}$ for $i' > i+1$, so $C[x_{i+1,j'}] \cap ran(g[x_{i,0}]) \neq \emptyset$, and we have a contradiction.

For $x_{0,0}+$, consider $C[x_{0,0}]$, this must be some C_i, $i > 0$, take $x_{i,0} \in C_i$, $x_{i,0}-$, and continue as above.

(4) Case 4

Modify Case 1, to a ranked order, but this time vertically: $x_{i,j} < x_{i',j'}$ iff $j < j'$.

Take $x_{i,j}+$, consider $C[x_{i,j}]$, this may be (part of) any $C_{i'}$ (beginning at $j+1$). Take e.g. $x_{i',j+1}- \in C[x_{i,j}]$, consider $g[x_{i',j+1}]$, a choice function in $\{C_k \upharpoonright x_{i',j+1} : k, \omega\}$ this will intersect $C[x_{i,j}]$. In particular, $x_{0,0}+$ is impossible.

Suppose $x_{0,0}-$, take $x_{i,j}+ \in ran(g[x_{0,0}])$, and continue as above.

Note that the case with just one C_i is the original Yablo construction.

(5) Case 5

We modify Case 1 again: Consider the ranked order by $x_{i,j} \not\to x_{i',j'}$ iff $max\{i,j\} < max\{i',j'\}$.

This is left to the reader as an exercise.

(6) Case 6

The general argument is as follows (and applies to general partial orders, too):

(6.1) We show that $x_{i,j}+$ leads to a contradiction. (In our terminology, $x_{i,j}$ is the head.)

(6.1.1) we find $C[x_{i,j}]$ (negative elements) above $x_{i,j}$ - but we have no control over the choice of $C[x_{i,j}]$,

(6.1.2) we chose $x_{i',j'} \in C[x_{i,j}]$ - if there is a minimal such, chose this one, it must not be a maximal element in $C[x_{i,j}]$ ($x_{i',j'}$ is the knee),

(6.1.3) we find $g[x_{i',j'}]$ (positive elements) above $x_{i',j'}$ - again we have no control over the choice of this g. But, as $C[x_{i,j}] \upharpoonright x_{i',j'}$ is not empty, $ran(g[x_{i',j'}]) \cap C[x_{i,j}] \neq \emptyset$, so we have a contradiction ($x_{i'',j''} \in ran(g[x_{i',j'}]) \cap C[x_{i,j}]$ is the foot).

(6.1.4) We apply the reasoning to $x_{0,0}$.

(6.2) We show that $x_{0,0}-$ leads to a contradiction.

(6.2.1) We have $g[x_{0,0}]$ (positive elements) - but no control over the choice of g. In particular, it may be arbitrarily high up.

(6.2.2) we chose $x_{i,j} \in ran(g[x_{0,0}])$ with enough room above it for the argument about $x_{i,j}+$ in (6.1).

(6.3) Case 3 is similar, except that we work horizontally, not vertically.

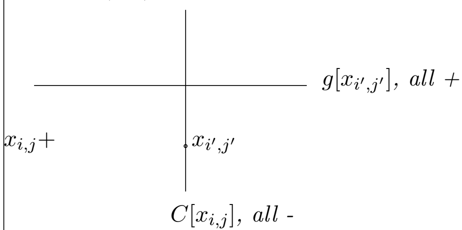
Remark 2.1.

Note that we may use here the ideas of Condition 4.1, based on the discussion in Section 3.3, to construct and check constructions for arbitrary formulas in disjunctive normal form.

Diagram 2.1. *Contradictions for Disjunctive Normal Forms*

Diagram for Definition 2.1 Case 5

Case 5, (4.1)



Case 5, (4.2)

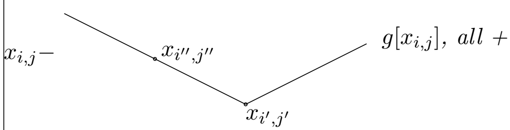
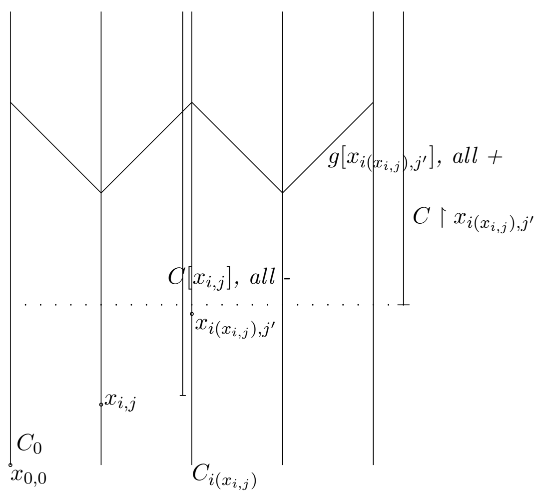


Diagram for Example 2.1 Case 1



3 Saw Blades

3.1 Introduction

Yablo works with contradictions in the form of $x_0 = \neg x_1 \wedge \neg x_2$, $x_1 = \neg x_2$, graphically $x_0 \not\twoheadrightarrow x_1 \not\twoheadrightarrow x_2$, $x_0 \not\twoheadrightarrow x_2$. They are combined in a formally simple total order, which, however, blurs conceptual differences.

We discuss here different, conceptually very clear and simple, examples of a Yablo-like construction.

In particular, we emphasize the difference between x_1 and x_2 in the Yablo contradictions. The contradiction is finished in x_2, but not in x_1, requiring the barring of escape routes in x_1. We repair the possible escape routes by constructing new contradictions for the same origin. This is equivalent to closing under transitivity in the individual "saw blades" - see below.

The main difference to Yablo's construction is that we first construct a contradiction $x_0 \not\twoheadrightarrow x_1 \not\twoheadrightarrow y_0$, $x_0 \not\twoheadrightarrow y_0$, and later $x_1 \not\twoheadrightarrow x_2$, and then $x_0 \not\twoheadrightarrow x_2$, like Yablo, so our construction is first more liberal, but we later see that we have to continue like Yablo.

So we use the same "cells" as Yablo does for the contradictions, but analyse the way they are put together.

We use the full strength of the conceptual difference between x_1 and x_2 (in above notation) only in Section 3.4, where we show that preventing x_2 from being TRUE is sufficient, whereas we need x_1 to be contradictory, see also Section 3.3. Thus, we obtain a minimal, i.e. necessary and sufficient, construction for combining Yablo cells in this way.

We analyse the constructions (Yablo's original construction and our variants) in detail, and see how they follow from the prerequisites (x_0+ is contradictory, and any arrow $x_0 \not\twoheadrightarrow z$ leads again to a contradiction in case $z+$ (and thus x_0-)).

3.2 Saw Blades

First, we show the escape route problem.

Example 3.1.

Consider Construction 3.1 without closing under transitivity, i.e. the only arrows originating in $x_{\sigma,0}$ will be $x_{\sigma,0} \not\twoheadrightarrow x_{\sigma,1}$ and $x_{\sigma,0} \not\twoheadrightarrow y_{\sigma,0}$, etc.

Let $x_{\sigma,0} = TRUE$, then $x_{\sigma,1}$ is an $\vee$, and we pursue the path $x_{\sigma,0} \not\twoheadrightarrow x_{\sigma,1} \not\twoheadrightarrow x_{\sigma,2}$, this has no contradiction so far, and we continue with $x_{\sigma,2} = TRUE$, $x_{\sigma,3}$ is $\vee$ again, we continue and have $x_{\sigma,0} \not\twoheadrightarrow x_{\sigma,1} \not\twoheadrightarrow x_{\sigma,2} \not\twoheadrightarrow x_{\sigma,3} \not\twoheadrightarrow x_{\sigma,4}$, etc., never meeting a contradiction, so we have an escape path.

Construction 3.1.

We construct a saw blade σ, SB_σ.

(1) "Saw Blades"

(1.1) Let $x_{\sigma,0} \not\to x_{\sigma,1} \not\to x_{\sigma,2} \not\to x_{\sigma,3} \not\to x_{\sigma,4}, \ldots.$
$x_{\sigma,0} \not\to y_{\sigma,0}$, $x_{\sigma,1} \not\to y_{\sigma,0}$, $x_{\sigma,1} \not\to y_{\sigma,1}$, $x_{\sigma,2} \not\to y_{\sigma,1}$, $x_{\sigma,2} \not\to y_{\sigma,2}$, $x_{\sigma,3} \not\to y_{\sigma,2}$, $x_{\sigma,3} \not\to y_{\sigma,3}$, $x_{\sigma,4} \not\to y_{\sigma,3}, \ldots.$
we call the construction a "saw blade", with "teeth" $y_{\sigma,0}$, $y_{\sigma,1}$, $y_{\sigma,2}, \ldots.$ and "back" $x_{\sigma,0}$, $x_{\sigma,1}$, $x_{\sigma,2}, \ldots.$
We call $x_{\sigma,0}$ the start of the blade.
See Diagram 3.1 (page 24), upper part.

(1.2) Add (against escape), e.g. first $x_{\sigma,0} \not\to x_{\sigma,2}$, $x_{\sigma,0} \not\to y_{\sigma,1}$, then $x_{\sigma,1} \not\to x_{\sigma,3}$, $x_{\sigma,1} \not\to y_{\sigma,2}$, now we have to add $x_{\sigma,0} \not\to x_{\sigma,3}$, $x_{\sigma,0} \not\to y_{\sigma,2}$, etc, recursively. This is equivalent to closing the saw blade under transitivity with negative arrows $\not\to$. This is easily seen.

(1.3) We set $x_{\sigma,i} = \bigwedge \neg z_{\sigma,j}$, for all $x_{\sigma,j}$ such that $x_{\sigma,i} \not\to z_{\sigma,j}$, as in the original Yablo construction.

(2) Composition of saw blades

(2.1) Add for the teeth $y_{\sigma,0}$, $y_{\sigma,1}$, $y_{\sigma,2} \ldots.$ their own saw blades, i.e. start at $y_{\sigma,0}$ a new saw blade $SB_{\sigma,0}$ with $y_{\sigma,0} = x_{\sigma,0,0}$, at $y_{\sigma,1}$ a new saw blade $SB_{\sigma,1}$ with $y_{\sigma,1} = x_{\sigma,1,0}$, etc.

(2.2) Do this recursively.
I.e., at every tooth of every saw blade start a new saw blade. See Diagram 3.2.

Note:
It is NOT necessary to close the whole structure (the individual saw blades together) under transitivity.

Fact 3.1.

All $z_{\sigma,i}$ in all saw blades so constructed are contradictory, i.e. assigning them a truth value leads to a contradiction.

Proof

The argument is almost the same as for Yablo's construction.
Fix some saw blade SB_σ in the construction.

(1) Take any $z_{\sigma,i}$ with $z_{\sigma,i}+$, i.e. $z_{\sigma,i} = TRUE$. We show that this is contradictory.

(1.1) Case 1: $z_{\sigma,i}$ is one of the $x_{\sigma,i}$ i.e. it is in the back of the blade.
Take any $x_{\sigma,i'}$ in the back such that there is an arrow $x_{\sigma,i} \not\twoheadrightarrow x_{\sigma,i'}$ ($i' := i+1$ suffices). Then $x_{\sigma,i'} = FALSE$, and we have an $\vee$ at $x_{\sigma,i'}$. Take any $z_{\sigma,j}$ such that $x_{\sigma,i'} \not\twoheadrightarrow z_{\sigma,j}$, by transitivity, $x_{\sigma,i} \not\twoheadrightarrow z_{\sigma,j}$, so $z_{\sigma,j} = FALSE$, but as $x_{\sigma,i'} = FALSE$, $z_{\sigma,j} = TRUE$, contradiction.

(1.2) Case 2: $z_{\sigma,i}$ is one of the $y_{\sigma,i}$, i.e. a tooth of the blade.
Then $y_{\sigma,i}$ is the start of the new blade starting at $y_{\sigma,i}$, and we argue as above in Case 1.

(2) Take any $z_{\sigma,i}$ with $z_{\sigma,i}-$, i.e. $z_{\sigma,i} = FALSE$, and we have an $\vee$ at $z_{\sigma,i}$, and one of the successors of $z_{\sigma,i}$, say $z_{\sigma,j}$, has to be TRUE. We just saw that this is impossible.

(For the intuition: If $z_{\sigma,i}$ is in the back of the blade, all of its successors are in the same blade. If $z_{\sigma,i}$ is one of the teeth of the blade, all of its successors are in the new blade, starting at $z_{\sigma,i}$. In both cases, $z_{\sigma,j} = TRUE$ leads to a contradiction, as we saw above.)

Remark 3.2.

Note that all $y_{\sigma,i}$ are contradictory, too, not only the $x_{\sigma,i}$. We will see in Section 3.4 that we can achieve this by simpler means, as we need to consider here the case $x_{\sigma,i}\vee$ only, the contradiction for the case $x_{\sigma,i}\wedge$ is already treated.

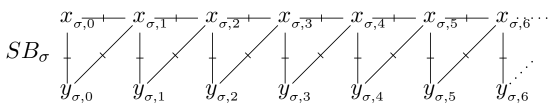
Thus, we seemingly did not fully use here the conceptual clarity of difference between x_1 and x_2 alluded to in the beginning of Section 3.2. See, however, the discussion in Section 3.3.

Diagram 3.1. *Saw Blade*

Horizontal lines stand for negative arrows from left to right,
the other lines for negative arrows from top to bottom

Diagram Single Saw Blade

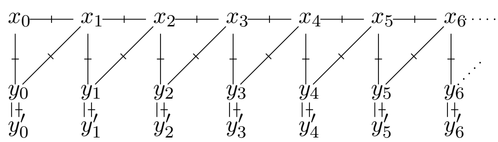
Start of the saw blade σ beginning at $x_{\sigma,0}$,
before closing under transitivity



Read $x_{\sigma,0} \not\to x_{\sigma,1} \not\to y_{\sigma,0}$, $x_{\sigma,0} \not\to y_{\sigma,0}$, etc., more precisely:
$x_{\sigma,0} = \neg x_{\sigma,1} \wedge \neg y_{\sigma,0}$, $x_{\sigma,1} = \neg y_{\sigma,0} \wedge \neg x_{\sigma,2} \wedge \neg y_{\sigma,1}$, *etc.*

Diagram Simplified Saw Blade

Start of the saw blade before closing
the blade (without "decoration") under transitivity



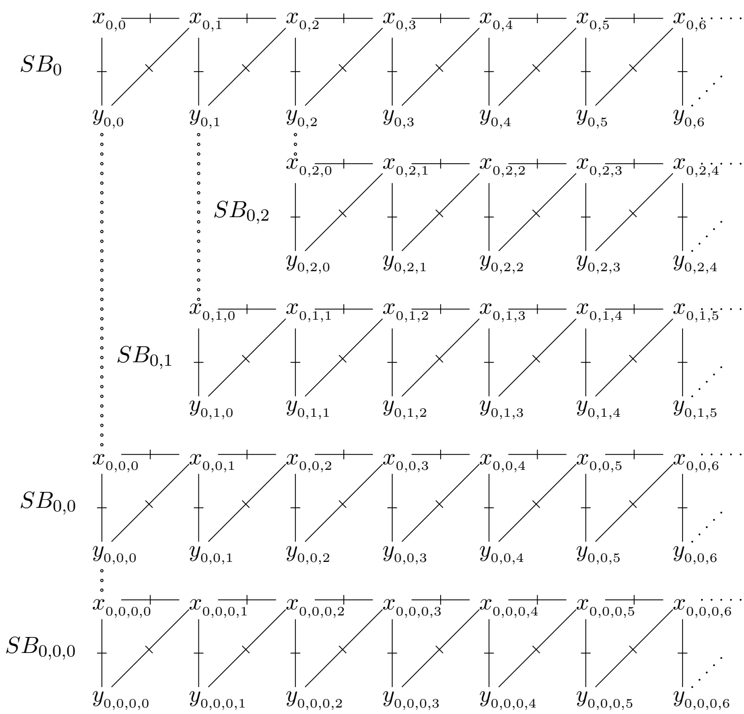
Read $y_0 = y'_0 \wedge \neg y'_0$ etc.

Diagram 3.2. *Composed Saw Blades*

Horizontal lines stand for negative arrows from left to right,
the other lines for negative arrows from top to bottom

Diagram Composition of Saw Blades

Composition of saw blades (without additional arrows)
The fat dots indicate identity, e.g. $y_{0,0} = x_{0,0,0}$



3.3 Discussion of Saw Blades

We analyse here the construction of saw blades from the prerequisites and show that Yablo's construction is a necessary substructure. The analysis applies as well to Yablo's original structure - it suffices to omit the y's.

It will be illustrated in Diagram 3.3 and Diagram 3.4. They will also show that the negative arrows are necessary - though this is not important here.

3.3.1 Preliminaries

Yablo's structure is sufficient, it shows that x_0 there does not have a consistent truth value.

We constructed in Section 3.2 a structure from the prerequisites which is more general, and contains Yablo's structure as a substructure. This shows that Yablo's structure is necessary (and minimal) under our prerequisites.

As we are mainly interested in constructing Yablo's structure from the prerequisites, we will stop analysing the rest of the saw blade structure after some initial steps, in order to simplify things.

One of the consequences of the construction is that we will have a constructive proof of the necessity of infinite depth and width.

We will work as usual with conjunctions of negations.

Recall:

Definition 3.1.

(1) A Yablo cell (YC) has the form $x \not\twoheadrightarrow x' \not\twoheadrightarrow y$, $x \not\twoheadrightarrow y$.

x is the head, x' the knee, y the foot.

Without any branching at x', $x+$ is impossible, as it is contradictory.

If $x+$, then $x' - \vee$. Thus, if we add some $x' \not\twoheadrightarrow y'$ (or $x' \rightarrow y'$), we have $x'- = y \vee y'$, so $x+$ is not contradictory any more. Any $x' \not\twoheadrightarrow y'$ opens a new possibility for $x+$ to be consistent, although x is the head of a YC.

(2) If, for all $x' \not\twoheadrightarrow y'$, we also have $x \not\twoheadrightarrow y'$, $x+$ is again impossible.

We call such a system of $x \not\twoheadrightarrow x'$, and for all $x' \not\twoheadrightarrow y_i$ also $x \not\twoheadrightarrow y_i$ a Yablo Cell System (YCS) - of course, with head x and knee x'.

Thus, if we want an contradiction at x, a YC with head x does not suffice, we need a YCS with head x.

(3) If we discuss general contradictory cells, we will write CC instead of YC, and CCS instead of YCS.

3.3.2 Prerequisites

We use the following prerequisites:

(1) x_0 is contradictory

(2) If there is an arrow $x_0 \not\twoheadrightarrow x$, then x is contradictory.

Translated to YC's and YCS's, this means:

(1) x_0 is the head of a YCS (for x_0+)

(2) For any $x_0 \not\to x$, and x_0-, thus $x + \bigwedge$, $x+$ is the head of a YCS.

3.3.3 Strategy and Details of the Construction

During the inductive construction, we will append new YC's, say $z \not\to z' \not\to z''$, $z \not\to z''$, to the construction done so far, but we will see that prerequisite (2) in Section 3.3.2 will force us to add new arrows to z', say $z' \not\to z'''$, so the YCS property at z is destroyed, and we will have to repair the YCS by adding a new arrow $z \not\to z'''$.

The final construction is the union of infinitely many steps, so all YCS's which were destroyed at some step will be repaired later again.

Construction 3.2.

(1) Start:

We begin with $x_0 \not\to x_1 \not\to y_0$, $x_0 \not\to y_0$.

See Diagram 3.3, Top.

This is a YCS with head x_0, thus x_0 is contradictory, and prerequisite (1) in Section 3.3.2 is satisfied.

(2) Prerequisite (2) is not satisfied, and we add two new YC's to x_1 and y_0 to satisfy prerequisite (2):

$x_1 \not\to x_2 \not\to y_1$, $x_1 \not\to y_1$, and

$y_0 \not\to z_0 \not\to z'_0$, $y_0 \not\to z'_0$.

See Diagram 3.3, Center, only $x_1 \not\to x_2 \not\to y_1$, $x_1 \not\to y_1$ shown.

(3) The new arrows $x_1 \not\to x_2$ and $x_1 \not\to y_1$ destroy the YCS at x_0 with knee x_1.

We could, of course, begin an new YC at x_0 with different knee, say $x_0 \not\to x'_1 \not\to y'_0$, which is a YCS, but this would take us back to step (1), step (2) would have been a dead end, and we would be in the same situation again. Thus, this does not lead anywhere.

Thus, we have to repair the situation of step (2) by adding $x_0 \not\to x_2$ and $x_0 \not\to y_1$, and we have a YCS again.

See Diagram 3.3, Bottom.

(4) Prerequisite (2) is again not satisfied, and we add two new YC's, to x_2 and y_1, say

$x_2 \not\rightarrow x_3 \not\rightarrow y_2$, $x_2 \not\rightarrow y_2$, and

$y_1 \not\rightarrow z_1 \not\rightarrow z'_1$, $y_1 \not\rightarrow z'_1$.

See Diagram 3.4, Top, only $x_2 \not\rightarrow x_3 \not\rightarrow y_2$, $x_2 \not\rightarrow y_2$ shown.

Note:

It might be possible to find or add arrows $x_2 \not\rightarrow u$ and $x_2 \not\rightarrow u'$ and $u \not\rightarrow u'$ to already constructed u, u', without introducing cycles. We continue with u, u', adding arrows $x_0 \not\rightarrow u$, $x_0 \not\rightarrow u'$, etc., but as the construction already achieved is finite, in the end, we will have to add new points and arrows. More precisely, such u, u' cannot be some x_i already constructed, this would create a cycle, it has to be some y or z. We will neglect the y's and z' anyway, as our main interest here lies in the x_i and the arrows between them.

(5) We now want to construct arrows $x_0 \not\rightarrow x_3$ and $x_0 \not\rightarrow y_2$ using a broken YCS property at x_0, but we have here a new knee, x_2 and not x_1 (which is the knee for x_0). So, we have to proceed indirectly.

(6) The new arrows $x_2 \not\rightarrow y_2$ and $x_2 \not\rightarrow x_3$ destroy the YCS at x_1 with knee x_2, and we add $x_1 \not\rightarrow x_3$ and $x_1 \not\rightarrow y_2$ to repair the YCS.

See Diagram 3.4, Center.

(7) But, now, we go from x_0 via the knee, have new arrows $x_1 \not\rightarrow x_3$ and $x_1 \not\rightarrow y_2$, and have to repair the YCS of x_0 by adding $x_0 \not\rightarrow x_3$ and $x_0 \not\rightarrow y_2$.

See Diagram 3.4, Bottom.

(8) We use here in (6) and (7) the general property of the construction that x_{i+1} is the knee of the YCS with head x_i.

Consequently, if we add an arrow $x_{i+1} \rightarrow x$, then we have to add an arrow $x_i \not\rightarrow x$ to repair the YCS with head x_i.

The argument goes downward until x_0 (as x_i is the knee of the YCS with head x_{i-1} etc.), so, if we add an arrow $x_{i+1} \not\rightarrow x$, we have to add new arrows $x_k \not\rightarrow x$ for all $0 \leq k \leq x_i$.

(9) (9.1) Thus, there are arrows $x_i \not\rightarrow x_j$ for all i, j, $i < j$, and the construction is transitive for the $x'_i s$.

(9.2) Every x_i is head of a YCS with knee x_{i+1}.

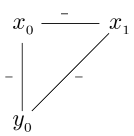
(9.3) Thus, every arrow from any x_i to any x_j goes to the head of a YCS, and not only the arrows from x_0.

This property is "accidental", and due to the fact that for any arrow $x_i \not\twoheadrightarrow x_j$, there is also an arrow $x_0 \not\twoheadrightarrow x_j$, and property (2) holds for x_0 by prerequisite.

(10) The construction has infinite depth and branching.

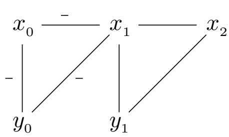
Diagram 3.3. *Inductive Construction 1*

Horizontal lines stand for negative arrows from left to right,
the other lines for negative arrows from top to bottom

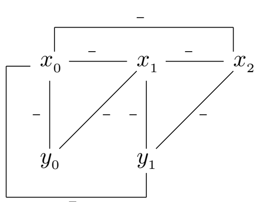


Contradiction at x_0+

(If x_0-*, all arrows from* x_0 *have to lead to a contradiction, so* $x_0 \not\twoheadrightarrow y_0$ *and* $x_0 \not\twoheadrightarrow x_1$ *have to be negative arrows.*
But if x_0+*, we need a contradiction in* $\{x_0, x_1, y_0\}$*, so* $x_1 \not\twoheadrightarrow y_0$ *has to be negative, too.)*



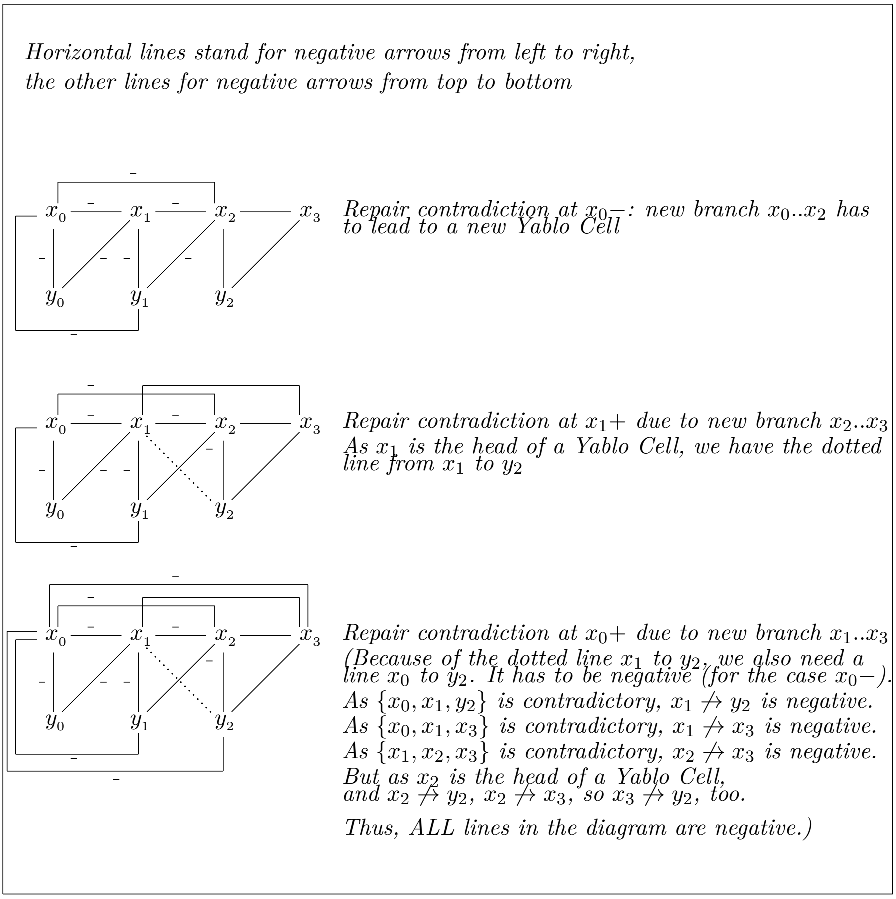
Contradiction for $x_0-, x_0..x_1$ *has to lead to new Yablo Cell*



Repair contradiction at x_0+
(due to new branches at x_1-*)*
In future, we will not draw the new arrows to the y_i'*s*

(Again, $x_0 \not\twoheadrightarrow x_2$ *and* $x_0 \not\twoheadrightarrow y_1$ *have to be negative. But, as* $\{x_0, x_1, y_1\}$ *and* $\{x_0, x_1, x_2\}$ *have to be contradictory,* $x_1 \not\twoheadrightarrow y_1$ *and* $x_1 \not\twoheadrightarrow x_2$ *have to be negative, too.)*

Diagram 3.4. *Inductive Construction 2*



3.4 Simplifications of the Saw Blade Construction

We show here that it is not necessary to make the $y_{\sigma,i}$ contradictory in a recursive construction, as in Construction 3.1. It suffices to prevent them to be true.

We discuss three, much simplified, Saw Blade constructions.

Thus, we fully use here the conceptual difference of x_1 and x_2, as alluded to at the beginning of Section 3.2.

Note, however, that the back of each saw blade "hides" a Yablo construction. The separate treatment of the teeth illustrates the conceptual difference, but it cannot escape blurring it again in the back of the blade.

First, we discuss some "false" simplifications which do not work.

3.4.1 "Simplifications" that Will Not Work

We try to simplify here the Saw Blade construction. Throughout, we consider formulas of pure conjunctions.

We start with a Yablo Cell, but try to continue otherwise.

So we have $x_0 \not\twoheadrightarrow x_1 \not\twoheadrightarrow x_2$, $x_0 \not\twoheadrightarrow x_2$. So x_0+ is impossible. We now try to treat x_0- . We see in Construction 3.3 that appending $x_2 \Rightarrow_{\pm} y_2$ may take care of the necessary contradiction at x_2, see Diagram 3.1, lower part. When we try to do the same at x_1, i.e. some $x_1 \Rightarrow_{\pm} x_3$, we solve again the necessary contradiction at x_1, but run into a problem with x_0+, as x_1 is an $\vee$. So x_3 has to be contradictory. If we continue $x_3 \Rightarrow_{\pm} x_4 \Rightarrow_{\pm} x_5$ etc., this will not work, as we may set all such x_i-, and have a model. In abstract terms, we only procrastinate the same problem without solving anything. Of course, we could append after some time new Yablo Cells, as in the saw blade construction, but this is cheating, as the "true" construction begins only later. This shows the difference between "knee" and "foot", it works with a foot, but not a knee. (Recall that all $x_i, i > 1$ are both foot and knee.)

Suppose we add not only $x_1 \Rightarrow_{\pm} x_3$, but also $x_0 \Rightarrow_{\pm} x_3$, then we solve x_0+, but x_0- is not solved.

Working with cells of the type (2.1) in Example 1.4 will lead to similar problems.

Consequently, any attempt to use a "pipeline", avoiding infinite branching, is doomed:

Instead of $x_0 \not\twoheadrightarrow x_1$, $x_0 \not\twoheadrightarrow x_2$, ... etc. we construct a "pipeline" of x'_i, with $x_0 \not\twoheadrightarrow x'_1$, $x_1 \not\twoheadrightarrow x'_2$, etc, $x'_1 \rightarrow x'_2 \rightarrow x'_3 \ldots$, and $x'_1 \rightarrow x_1$, $x'_2 \rightarrow x_2$, etc. or similarly, to have infinitely many contradictions for paths from x_0.

As this is a set of classical formulas, this cannot achieve inconsistency, see Fact 1.4.

3.4.2 Real Simplifications

Construction 3.3.

(1) Take ONE saw blade σ, and attach (after closing under transitivity) at all $y_{\sigma,i}$ a SINGLE Yablo Cell $y_{\sigma,i} \not\twoheadrightarrow u_{\sigma,i} \not\twoheadrightarrow v_{\sigma,i}$, $y_{\sigma,i} \not\twoheadrightarrow v_{\sigma,i}$. We call this the decoration, it is not involved in closure under transitivity.

(1.1) Any node z in the saw blade (back or tooth) cannot be $z+$, this leads to a contradiction:

If $z = x_i$ (in the back):

Let x_i+ : Take $x_i \not\twoheadrightarrow x_{i+1}$ (any x_j, $i < j$ would do), if $x_{i+1} \not\twoheadrightarrow r$, then by transitivity, $x_i \not\twoheadrightarrow r$, so we have a contradiction.

If $z = y_i$ (a tooth):

y_i+ is contradictory by the "decoration" appended to y_i.

(1.2) Any x_i- (x_i in the back, as a matter of fact, x_0- would suffice) is impossible:

Consider any $x_i \not\twoheadrightarrow r$, then $r+$ is impossible, as we just saw.

Note: there are no arrows from the back of the blade to the decoration.

(2) We can simplify even further. The only thing we need about the y_i is that they cannot be $+$. Instead of decorating them with a Yablo Cell, any contradiction will do, the simplest one is $y_i = y'_i \wedge \neg y'_i$. Even just one y' s.t. $y_i = y' \wedge \neg y'$ for all y_i would do. (Or a constant FALSE.)

See Diagram 3.1, lower part.

Formally, we set

$x_0 := \bigwedge\{\neg x_i : i > 0\} \wedge \bigwedge\{\neg y_i : i \geq 0\}$,

for $j > 0$:

$x_j := \bigwedge\{\neg x_i : i > j\} \wedge \bigwedge\{\neg y_i : i \geq j-1\}$,

and

$y_j := y'_j \wedge \neg y'_j$.

(3) In a further step, we see that the y_i (and thus the y'_i) need not be different from each other, one y and one y' suffice.

Thus, we set $x_j := \bigwedge\{\neg x_i : i > j\} \wedge \neg y$, $y := y' \wedge \neg y'$.

(Intuitively, the cells are arranged in a circle, with y at the center, and y' "sticking out". We might call this a "curled saw blade".

(4) When we throw away the y_j altogether, we have Yablo's construction. this works, as we have the essential part in the x_i's, and used the $y'_j s$ only as a sort of scaffolding.

Remark 3.3.

It seems difficult to conceptually simplify even further, as Section 3.3 shows basically the need for the construction of the single Saw Blades. We have to do something about the teeth, and above Construction 3.3, in particular cases (2) and (3) are simple solutions.

The construction is robust, as the following easy remarks show (see also Section 6):

(1) Suppose we have "gaps" in the closure under transitivity, so, e.g. not all $x_0 \not\twoheadrightarrow x_i$ exist, they always exist only for $i > n$. (And all other $x_k \not\twoheadrightarrow x_l$ exist.) Then x_0 is still contradictory. Proof: Suppose x_0+, then we have the contradiction $x_0 \not\twoheadrightarrow x_n \not\twoheadrightarrow x_{n+1}$ and $x_0 \not\twoheadrightarrow x_{n+1}$. Suppose x_0-, let $x_0 \not\twoheadrightarrow x_i$. As x_i is unaffected, x_i+ is impossible.

(2) Not only x_0 has gaps, but other x_i, too. Let again x_n be an upper bound for the gaps. As above, we see that x_0+, but also all x_i+ are impossible. If $x_0 \not\twoheadrightarrow x_i$, as x_i+ is impossible, x_0- is impossible.

(3) x_0 has unboundedly often gaps, the other x_i are not affected. Thus, for $i \neq 0$, x_i+ and x_i- are impossible. Thus, x_0- is impossible, as all x_i+ are, and x_0+ is, as all x_i- are.

See also Section 6.

4 The "Right" Level of Abstraction

4.1 Introduction

The basic elements in Yablo's construction are negative arrows, from which Yablo Cells are built. We show here that we can build arbitrarily complex structures equivalent to negative arrows, and as a matter of fact to any propositional logical operator. This suggests that the right level of abstraction to consider more general Yablo-like structures is not the level of single arrows, but rather of paths.

Note that [1] and [5] also work with paths in graphs.

4.2 Expressing Logical Operators by Combinations of Yablo Cells

Remark 4.1.

In the diagrams in Diagram 4.1 the inner path $x - y - z$ is barred by $x - z$, and the outer path contains 3 or 2 negations.

Note that in all diagrams Diagram 4.1, and Diagram 4.2, upper part, y will always be FALSE, and as $x = \neg y \wedge \neg z$, $\neg y$ is TRUE and will not be considered, the value of x depends only on the branch through z.

E.g., in Diagram 4.1, upper part, the path $z - x$ has uneven length, so the diagram describes negation, in Diagram 4.1, lower part, the path $z' - z - x$ has even length, so the diagram describes identity.

Definition 4.1.

We define negations and their type.

See Diagram 4.1.

A negation (diagram) is a diagram all of whose arrows are of the $\not\to$ kind.

We define negation diagrams, or, simply, negations, and their type.

- An arrow $x \not\to x'$ is a negation of type 0.

  A negation of the type $x \not\Rightarrow z$, $x \not\Rightarrow y \not\Rightarrow z$, $y \not\Rightarrow y' \not\Rightarrow z$ (see Diagram 4.1, upper left) is of type $n+1$ iff

  all negations inside are of type $\leq n$, and at least one negation inside is of type n.

Similarly, we may define the type of an identity via the type of the negations it is composed of - see Diagram 4.1, lower part.

This all shows that we may blur the basic structure almost ad libitum, making a characterisation difficult. (See here Diagram 4.2, lower part, etc. too.)

Remark 4.2.

Thus, it seems very difficult to describe Yablo type diagrams on the level of single arrows. It seems to be the wrong level of abstraction.

We now give some examples.

4.3 Some Examples

Example 4.1.

See Diagram 4.1, and Diagram 4.2.

(1) Negation (1)

$x = \neg y \wedge \neg z$, $y = \neg z \wedge \neg y'$, $y' = \neg z$, so $\neg y = z \vee y'$, and $x = (z \vee y') \wedge \neg z = (z \vee \neg z) \wedge \neg z = \neg z$.

(2) Negation (2)

$x = \neg y \wedge \neg z$, $y = \neg y'' \wedge \neg y'$, $y' = \neg z$, $y'' = z$, so $y = \neg z \wedge \neg y'$, and continue as above.

(3) Identity

$x = \neg y \wedge \neg z$, $y = \neg z \wedge \neg z'$, $z = \neg z'$, so $\neg y = z \vee z'$, and $x = (\neg z' \vee z') \wedge z' = z'$.

(4) TRUE

$x = \neg y \wedge \neg z$, $y = \neg y' \wedge \neg z$, $y' = \neg z$, $z = \neg z' \wedge \neg z''$, $z' = \neg z''$. Thus, $z = z'' \wedge \neg z''$, $\neg z = z'' \vee \neg z'' = TRUE$, $y' = TRUE$, $y = FALSE \wedge TRUE = FALSE$, and $x = TRUE \wedge TRUE$.

(5) $z' \wedge \neg u$

$x = \neg z \wedge \neg y \wedge \neg u$, $y = \neg z \wedge \neg u' \wedge \neg u$, $z = \neg z'$, $u' = \neg u$, so $y = z' \wedge u \wedge \neg u$, $\neg y = \neg z' \vee \neg u \vee u = TRUE$, and $x = z' \wedge TRUE \wedge \neg u = z' \wedge \neg u$.

.

Diagram 4.1. *Propositional Formulas 1*

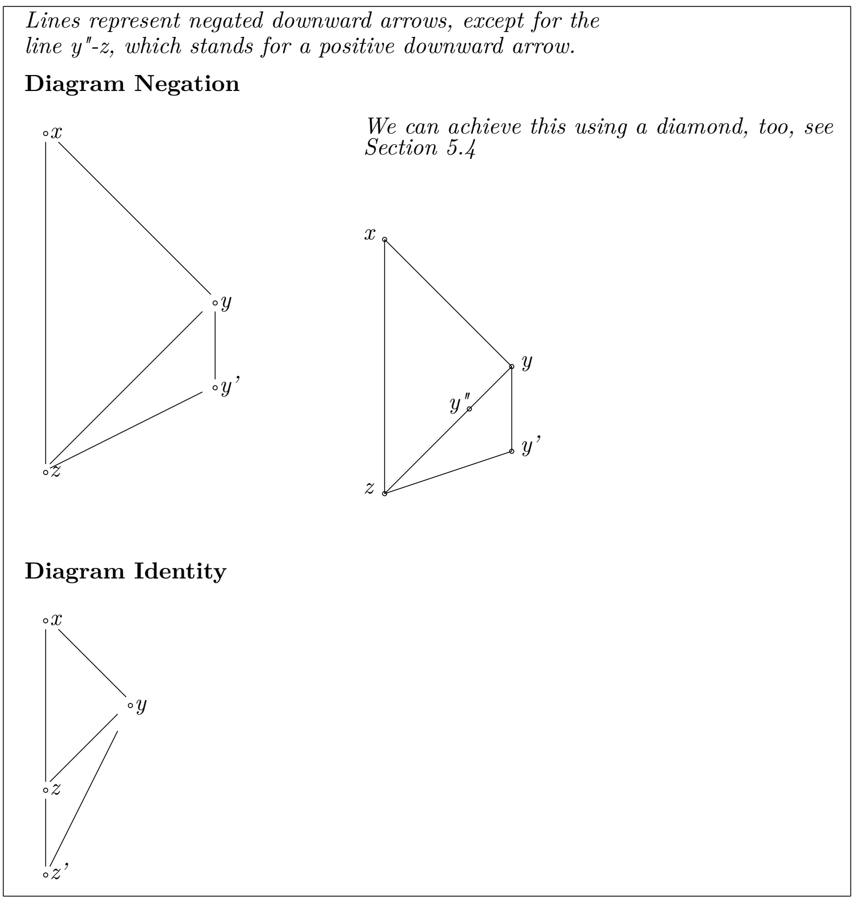
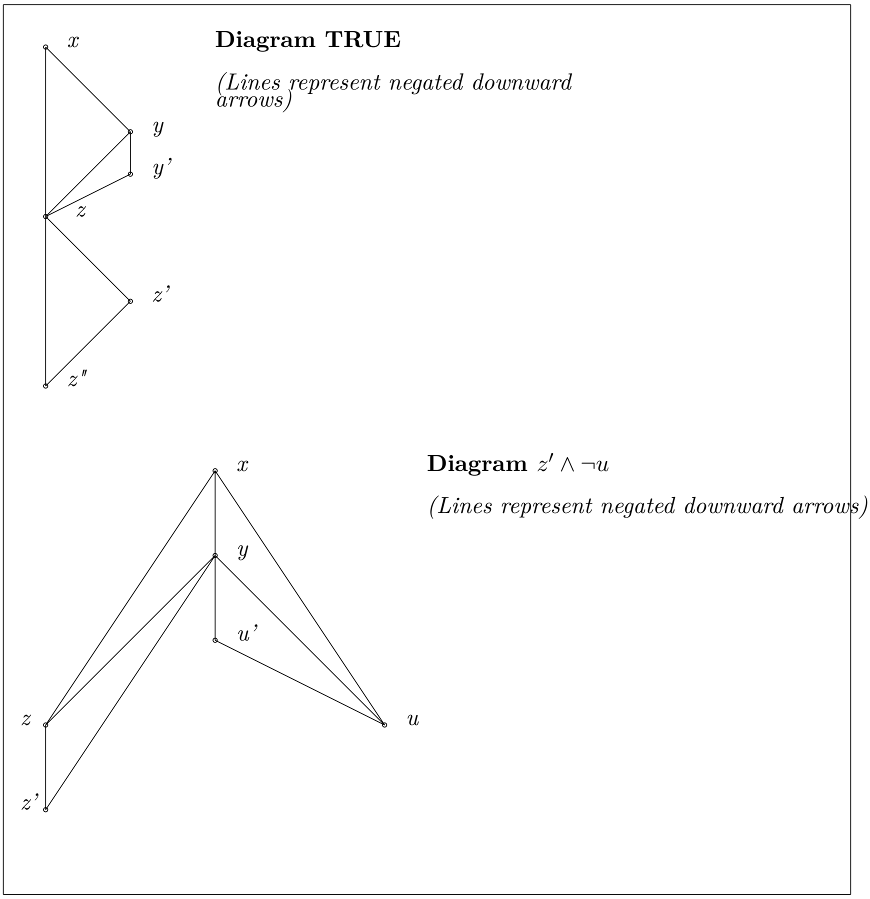


Diagram 4.2. *Propositional Formulas 2*



4.4 Paths Instead of Arrows

We concluded above that the level of arrows might be the wrong level of abstraction. We change perspective, and consider paths, i.e. sequences of arrows, sometimes neglecting branching points.

We assume all formulas attached to nodes are as in Yablo's paper, i.e. pure conjunctions of positive or negative nodes.

Condition 4.1.

Note that we speak only about special cases here: The formulas are of the type $\bigwedge \neg$, we use Yablo cells for contradictions, and every arrow (negative path below) leads directly to a new Yablo cell, see Section 3.3.2.

(We use an extension of the language beyond $\bigwedge \neg$ for simplifications ("cutting unwanted branches"), so, strictly speaking, our result is NOT an equivalence result.

In addition to $\neg x_i$ as components of $\bigwedge$, we also admit formulas of the type $(x_i \vee \neg x_i)$ or the constant TRUE.)

Suppose a graph Γ contains an infinite set of points $x_i : i \in \omega$, which can be connected by negative paths $\sigma_{i,j} : x_i \ldots x_j$ for all $i < j$, then we can interpret this graph so that x_0 (as a matter of fact, all x_i) cannot have a truth value.

Thus, the graph is directed, the arrows are not necessarily labelled (labels + or -), but labelling can be done such that for each pair of nodes x, y, where y is a successor of x, there is a negative path from x to y. (See Definition 1.2.)

Remark: The condition "negative paths" is not trivial. Suppose that $\sigma : x \ldots y$ and $\sigma' : y \ldots z$ are both negative, and $\sigma \circ \sigma'$ is the only path from x to z, then the condition is obviously false. The author does not know how to characterize graphs which satisfy the condition. Some kind of "richness" will probably have to hold.

Proof

We saw in Section 3.3 that we can generate from the prerequisites a structure isomorphic to Yablo's structure.

Conversely: Let y be any node in Γ, if $y \not\twoheadrightarrow y'$ is not part of the $\sigma_{i,j}$, then interpret $y \not\twoheadrightarrow y'$ by $(y' \vee \neg y')$. Thus, $y \not\twoheadrightarrow y'$ has no influence on the truth of y. In particular, the branching points inside the $\sigma_{i,j}$ disappear, and the $\sigma_{i,j}$ become trivial. The x_i are not influenced by anything apart from the x_j, $j > i$.

Thus, the construction of the x_i, x_j, $\sigma_{i,j}$ is equivalent to Yablo's structure.

Definition 4.2.

Our argument above was a bit sloppy. We can make it more precise.

If x is contradictory, i.e. both $x+$ and x- lead to a contradiction, we have $M(x) = \emptyset$, and $M(\neg x) = \emptyset$, with $M(.)$ the set of models.

Still, the following definition seems reasonable:

$M(x \wedge \phi) := M(x) \cap M(\phi)$, and $M(x \vee \phi) := M(x) \cup M(\phi)$.

In above proof, we used $M(x \wedge TRUE) = M(x)$.

5 Possible Contradiction Cells

5.1 Introduction

We consider alternative basic contradiction cells, similar to Yablo cells. We saw in Example 1.4 that Yablo cells are the simplest contradictions suitable to a contradictory construction (i.e. without possible truth values). We now examine more complicated contradiction cells, the only interesting one is the "diamond", but this cell fails to fit into our construction principle - at least in our logical framework, see Section 5.4.1.

The Yablo cell has the form $x \not\twoheadrightarrow y \not\twoheadrightarrow z$, $x \not\twoheadrightarrow z$, whereas the diamond has the form $x \not\twoheadrightarrow y \not\twoheadrightarrow z$, $x \not\twoheadrightarrow y' \rightarrow z$. The diamond is more complicated, as it contains two branching points, y and y', before the contradiction is complete, and we have to "synchronize" what happens at those branching points.

5.2 Prerequisites

In Yablo's construction, and our Saw Blades, the basic contradiction had the form $x \not\twoheadrightarrow y \not\twoheadrightarrow z$, $x \not\twoheadrightarrow z$ (Yablo Cell, YC). We generalize this now, as illustrated in Diagram 5.1, x_0 corresponds to x, a_2 to y, b_2 to z. The arrow $x \not\twoheadrightarrow y$ is replaced by a perhaps more complicated path via a_1, $x \not\twoheadrightarrow z$ by a perhaps more complicated path via b_2, but we still want a_2 and b_2 origins of new contradictions in the case $x_0 -$. Thus, both $\sigma : x_0 \ldots a_2$ and $\sigma' : x_0 \ldots b_2$ have to be negative paths.

The most interesting modification is of the type of Case (2.4.2) in Remark 5.1, where the contradiction originating at x_0 is via an additional point, z. We call this a "Diamond", and examine this case in more detail in Section 5.4, and Section 5.4.1.

In the Diamond case, we need a positive arrow (equivalently two negative arrows without branching in between), either from a_2 to z, or from b_2 to z.

In all cases, σ from x_0 to a_2, and σ' from x_0 to b_2 has to be negative, so x_0- results in a_2+ and $b_2 +$. Omitting a_1 and b_1, and adding a direct arrow $a_2 \not\twoheadrightarrow b_2$ results in the original Yablo Cell.

We continue to work with pure conjunctions, now of positive or negative propositional variables. (We will be forced to look at more complicated formulas in Remark 5.5.)

5.3 Various Types

Remark 5.1.

See Diagram 5.1.

We examine here the different modifications of Yablo's basic construction.

Note that e.g. $x_0 \ldots a_1$ need not be an arrow, it may be a longer path.

The contradiction need not be e.g. $a_2 \ldots b_2$, but they might take a "detour" $a_2 \ldots z$ and $b_2 \ldots z$. (The cases like $a_2 \ldots z \ldots b_2$ are already covered by $a_2..b_2$.) So z will be the "point of conflict". We will call these cases "detour to z" (and the construction "diamonds").

We do not break e.g. $x_0 \ldots a_1$ further down, this suffices for our analysis.

The situation:

(1) x_0+ must be contradictory, the contradiction formed with (perhaps part of) σ and σ',

(2) both a_2+ and b_2+ must be contradictory if x_0-.

Recall that the paths $x_0 \ldots a_1 \ldots a_2$ and $x_0 \ldots b_1 \ldots b_2$ are both negative (as a_2 and b_2 must be positive, if x_0 is negative).

We look at the following cases of building the contradiction at x_0, e.g. Case (1.1) means that we have the contradiction between $x_0 \ldots a_1 \ldots b_1$ and $x_0 \ldots b_1$, Case (1.4.1) means that we have a contradiction between $x_0 \ldots a_1 \ldots z$ and $x_0 \ldots b_1..z$.

(1) a_1 as additional branching point

- (1.1) from a_1 to b_1
- (1.2) from a_1 to b_2
- (1.3) from a_1 to b_3
- (1.4) from a_1 to z and
  - (1.4.1) from b_1 to z, b_1 as additional branching point
  - (1.4.2) from b_2 to z, b_2 as additional branching point
  - (1.4.3) from b_3 to z, b_3 as additional branching point

(2) a_2 (again b_i as additional branching points)

- (2.1) from a_2 to b_1
- (2.2) from a_2 to b_2
- (2.3) from a_2 to b_3
- (2.4) from a_2 to z and
  - (2.4.1) from b_1 to z
  - (2.4.2) from b_2 to z
  - (2.4.3) from b_3 to z

(3) a_3 (again a_3 and b_i as additional branching points)

- (3.1) from a_3 to b_1
- (3.2) from a_3 to b_2
- (3.3) from a_3 to b_3
- (3.4) from a_3 to z and
  - (3.4.1) from b_1 to z

(3.4.2) from b_2 to z

(3.4.3) from b_3 to z

We will examine the cases now.

(1) Case (1):

In all cases, we have to branch at a_1 with $a_1 + \wedge$, as we need to have a contradiction for x_0+, so we have $x_0 \to a_1$. But then, for x_0-, we branch at $a_1 - \vee$. Now, one branch will lead to a_2, the other to b_1, b_2, b_3, or z.

Cases (1.1) and (1.2):

Suppose x_0-, and a_1 is chosen at x_0. As a_1-, suppose b_1 is chosen at a_1. By prerequisite (we need a contradiction), $x_0 \ldots a_1 \ldots b_1$ contradicts $x_0 \ldots b_1$, moreover $x_0 \ldots b_1..b_2$ is negative, so $x_0 \ldots a_1 \ldots b_1 \ldots b_2$ is positive, and the chosen path misses a_2 and makes b_2 negative, contradicting (2) above in "The situation". (Thus, appending the same type of construction at b_2 again, etc. will result in an escape path x_0-, b_2-, etc.)

Case (1.3) and (1.4):

Consider x_0-, chose at x_0 a_1 and chose at a_1 b_3 (or z). Then $x \ldots a_1 \ldots b_3$ or $x \ldots a_1 \ldots z$ will not meet a_2 nor b_2.

(2) Case (2):

Cases (2.1) and (2.2) are (equivalent to) a Yablo Cell.

Cases (2.3), (2.4.2), and (2.4.3) are equivalent, and discussed below ("Diamond"), see Section 5.4. Note that, e.g. in case (2.4.2), to have a contradiction, we need $val(\rho) \neq val(\rho')$.

Cases (2.4.1) and (1.4.2) are symmetrical.

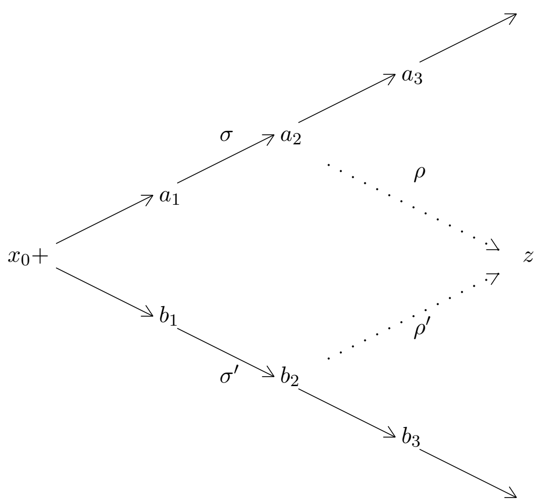
(3) Case (3):

Similar considerations as for Case (2) apply.

Diagram 5.1. *General Contradiction*

See Remark 5.1

Recall that $val(x_0..a_2) = val(x_0..b_2) = -$
but $val(\rho) \neq val(\rho')$



The solid arrows stand for positive or negative paths
For the dotted arrows see Remark 5.1, Case (2.4.2) -
they form part of the Diamond contradiction:
$x_0..b_1..b_2..z$, $x_0..a_1..a_2..z$.

5.4 Diamonds

We first give some simple examples as "warming up exercises", before we turn in Section 5.4.1 to a proof that - at least in our setting - Diamonds cannot replace Yablo Cells, see Remark 5.5.

Recall that in all examples below $val(x_0, x_1) = val(x_0, x_2) = -$.

Remark 5.2.

We may, of course, imitate a diamond $a \not\to b \not\to d$, $a \not\to c \to d$ by a triangle $a \not\to b \not\to d$, $a \not\to c \to d$, where the "detour" via c is integrated in the line $a \not\to d$. But this is "cheating", and would neglect the essential property, that each of b and c should have a new diamond attached to it.

Remark 5.3.

Case (1), see Diagram 5.2.

x_0 and x_1 should be heads of contradiction cells.

Take e.g. Case (1.1). Here, x_1 is a conjunction of positive or negative occurences of y_1, y_2, y, Thus, if x_0+, $\neg x_1$ is a disjunction of (the negatives) of above occurences. So, it might e.g. be $y_1 \vee \neg y_2 \vee y$, and we cannot be sure of going to y, and thus we are not sure to have a contradiction for the case $x_0 +$. In more detail:

If x_0+, then x_1-, and let x_1- be expressed by some logical formula $\phi = \phi_1 \vee \ldots \vee \phi_n$, with each ϕ_i a conjunction of letters. We know that, for the contradiction to work for x_0, $val(x_1, y) \neq val(x_2, y)$ has to hold. Suppose $val(x_2, y) = -$, so val $(x_1, y) = +$. Then, in each conjunction, we must have y. Take now the case x_0-, then x_1+ and consequently $val(x_1, y_1) \neq val(x_1, y_2, y_1)$ has to hold. Fix $val(y_2, y_1)$. The negation of ϕ is the disjunction of the (negated) conjunctions of all choice functions in the conjunctions for ϕ. But one of those conjunctions is just $\neg y \wedge \ldots \wedge \neg y = \neg y$, it says nothing about y_1 or y_2, so we cannot force any coherence between $val(x_1, y_1)$ and $val(x_1, y_2, y_1)$.

Note that this argument works for arbitrary infinite formulas, too.

The argument for Case (1.2) is similar.

Case (3) is more interesting, see Diagram 5.3.

We want x_0, x_1, and x_2 to be heads of contradiction cells. We discuss Case (3.1), (3.2) is similar.

Suppose $val(x_1, y) = -$, $val(x_2, y) = +$

We distinguish two possibilities:

Case A: $val(y, z) = +$.

Then $val(x_1, z) = +$, $val(x_2, z) = -$. If x_0+, then $x_1 - \bigvee$, $x_2 - \bigvee$. Suppose x_1 chooses z, and x_2 chooses y, then we have no contradiction.

Case B : $val(y, z) = -$.

Then $val(x_1, z) = -$, $val(x_2, z) = +$. Again, if x_1 chooses z, and x_2 chooses y, we have no contradiction.

Diagram 5.2. *Simple Diamonds 1*

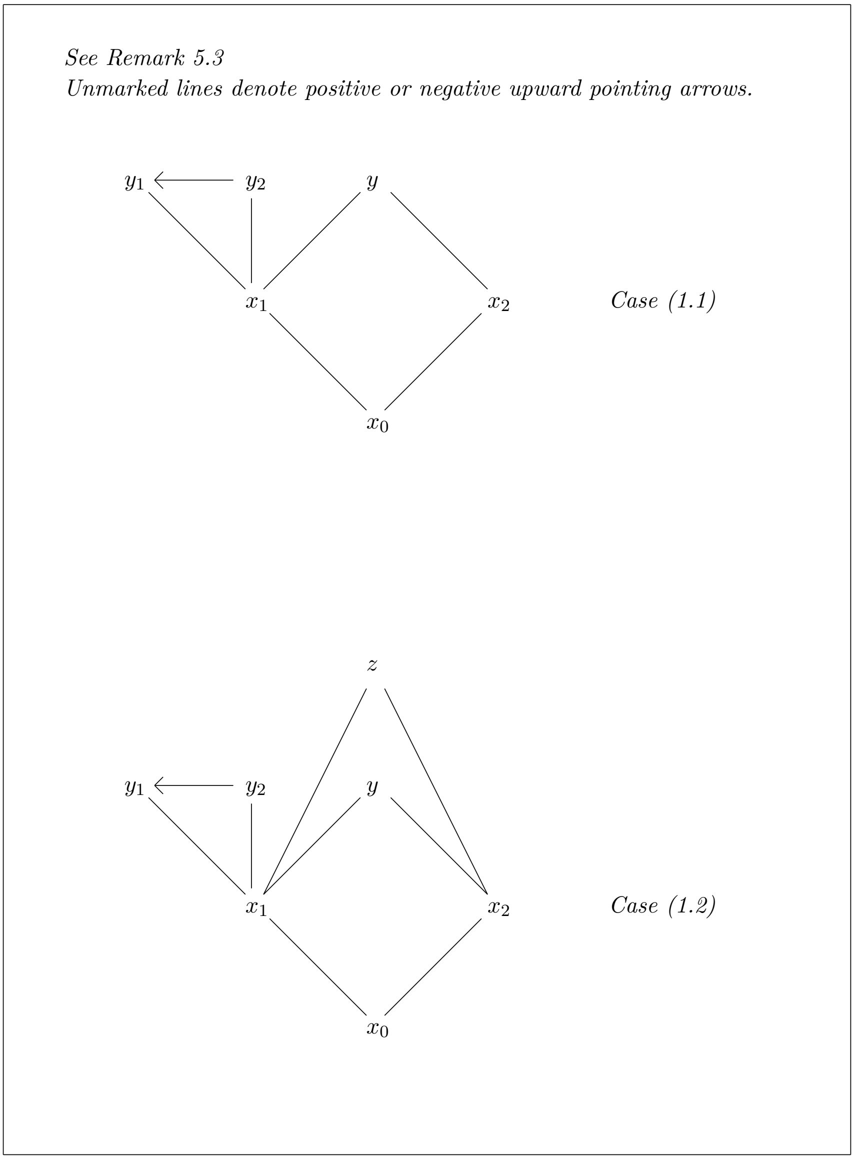
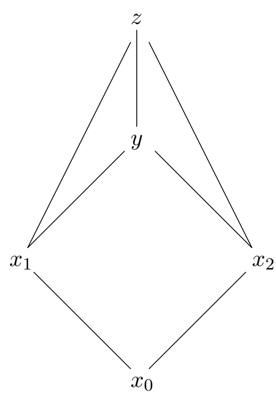
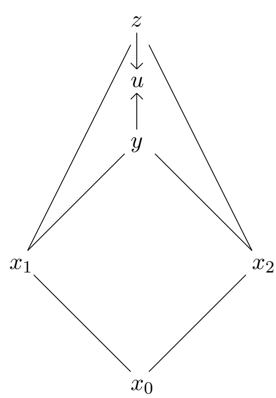


Diagram 5.3. *Simple Diamonds 2*

See Remark 5.3
Unmarked lines denote positive or negative upward pointing arrows.



Case (3.1)



Case (3.2)

5.4.1 Nested Diamonds

We present here some examples and problems of constructions with nested diamonds.

Remark 5.4.

We first try to imitate the Yablo construction using diamonds instead of triangles. See Diagram 5.4, for the moment both sides.

The basic construction is $x_0 - x_{1,1} - x_2$, $x_0 - x_{1,2} - x_2$.

To make the construction recursive, we have to add, among other things, a new diamond at $x_{1,1}$, $x_{1,1} - x_{3,1} - x_4$ and $x_{1,1} - x_{3,2} - x_4$, etc.

(1) Consider first the left hand side of the diagram.

$x_0 - x_{1,2} - x_2$ contradicts $x_0 - x_{1,1} - x_2$, and $x_0 - x_{1,2} - x_2 - x_{3,2}$ contradicts $x_0 - x_{1,1} - x_{3,2}$.

$x_0 - x_{1,1} - x_{3,1} - x_4$ and $x_0 - x_{1,2} - x_2 - x_{3,2} - x_4$ are not contradictory, just as $x_0 - x_{1,1} - x_{3,1} - x_{5,2}$ and $x_0 - x_{1,2} - x_2 - x_{3,2} - x_4 - x_{5,2}$ are not contradictory.

More generally, the paths via $x_{i,1}$ to x_{i+1} and via $x_{i,1}$ to $x_{i+2,2}$ behave the same way. It is also easy to see that $x_{i,1} - x_{i+2,2} - x_{i+3}$ and $x_{i,1} - x_{i+2,1} - x_{i+3}$ behave differently with respect to contradicting the simple vertical path, e.g. $x_0 - x_{1,2} - x_2 - x_{3,2}$ contradicts $x_0 - x_{1,1} - x_{3,2}$, but $x_0 - x_{1,1} - x_{3,1} - x_4$ and $x_0 - x_{1,2} - x_2 - x_{3,2} - x_4$ are not contradictory.

Consider now $x_0 \oplus$, then $x_{1,1} \ominus \bigvee$, and we have to make all paths to contradict the corresponding simple vertical path. This works for $x_{1,1} - x_2$ and $x_{1,1} - x_{3,2}$, but not for $x_{1,1} - x_{3,1} - x_4$ and $x_{1,1} - x_{3,1} - x_{5,2}$. As $x_{3,1} \oplus \bigwedge$, we have a "second chance" to continue via $x_0 - x_{1,1} - x_{3,1} - x_{5,1}$, but $x_{5,1}$ is $\ominus \bigvee$, so $x_{5,1} - x_6$ and $x_{5,1} - x_{7,2}$ are ok, but $x_{5,1} - x_{7,1}$ is not, and we have an escape path.

(2) The construction on the right hand side of the diagram avoids this problem, as for $x_0 \oplus$, $x_{3,1} \oplus \bigwedge$, and $x_0 - x_{1,1} - x_{3,1} - x_{6,2}$ contradicts $x_0 - x_{1,2} - x_2 - x_{3,2} - x_4 - x_5 - x_{6,2}$.

But we created a different problem for the added diamond: If $x_{1,1} \oplus$, then $x_{3,1} \ominus$, so not only $x_{1,1} - x_{3,1} - x_4$ has to contradict $x_{1,1} - x_{3,2} - x_4$, but also $x_{1,1} - x_{3,1} - x_{6,2}$ has to contradict $x_{1,1} - x_{3,2} - x_4 - x_5 - x_{6,2}$, which does not work.

In addition, we have a triangle contradiction in $x_{3,1} \rightarrow x_4 \not\rightarrow x_5 \not\rightarrow x_{6,2}$, $x_{3,1} \not\rightarrow x_{6,2}$.

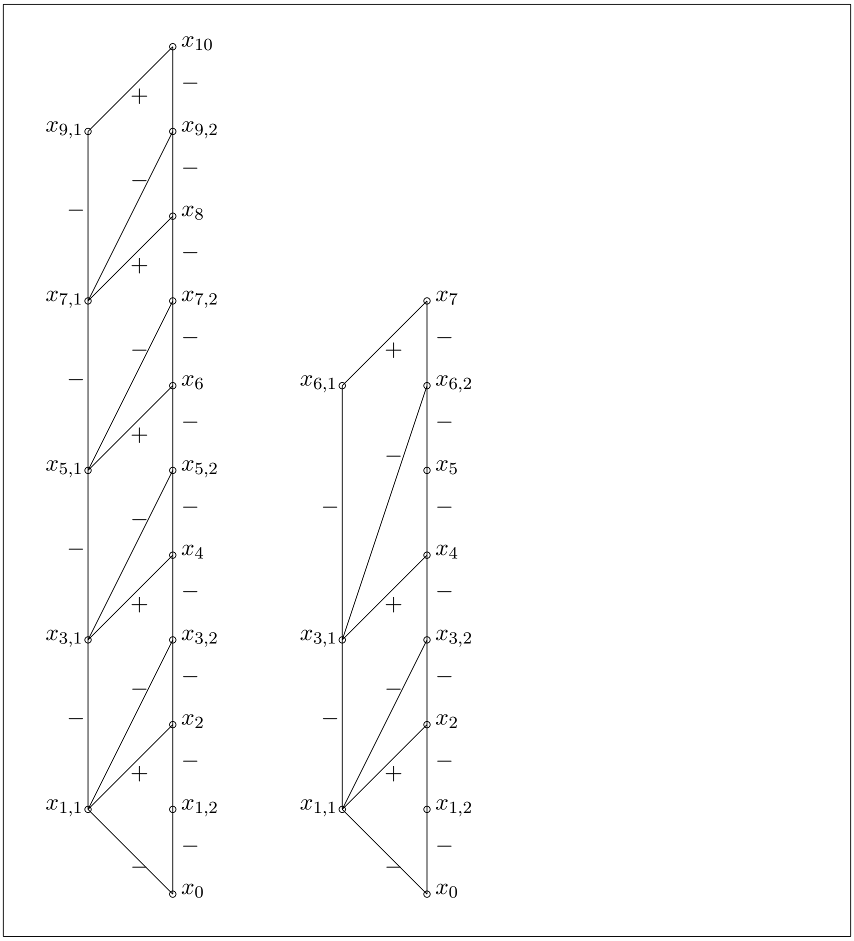
In summary: both attempts fail, but, of course, a different approach might still work.

(3) Note that the roles of $\bigwedge$ and $\bigvee$ are exchanged when we go from x_0 to $x_{1,1}$ and $x_{1,2}$.

In Yablo's construction, this presents no problem, as the structure seen from x_0 and any other x_i is the same.

This is not the case in our example.

Diagram 5.4. *Nested Diamonds 1*



Remark 5.5.

See Diagram 5.5, and in more detail Diagram 5.6.

We neglect the triangles like $x_{1,1} - x_{3,2} - x_2$, $x_{1,1} - x_2$. (See (5) below for a comment.)

The diagrams show the basic recursive construction, but additional Diamonds still have to be added on top of the existing Diamonds.

To make the Diagram 5.5 more readable, we noted some points several times (x_2, $x_{4,1}$, $x_{4,2}$), they are connected by vertical fat lines.

We have 7 diamonds, (x_0, $x_{1,1}$, x_2, $x_{1,2}$), ($x_{1,1}$, $x_{3,1}$, $x_{4,1}$, $x_{3,2}$), ($x_{1,2}$, $x_{3,3}$, $x_{4,2}$, $x_{3,4}$), ($x_{3,1}$, $x_{5,1}$, $x_{6,1}$, $x_{5,2}$), ($x_{3,2}$, $x_{5,3}$, $x_{6,2}$, $x_{5,4}$), ($x_{3,3}$, $x_{5,5}$, $x_{6,3}$, $x_{5,6}$), ($x_{3,4}$, $x_{5,7}$, $x_{6,4}$, $x_{5,8}$). They are drawn using thick lines.

In all diamonds, the bottom and the upper right lines are supposed to be negative, the upper left line is meant positive for a contradiction.

This diagram, or some modification of it seems the right way to to use diamonds instead of triangles for the contradictions recursively, as the triangles are used recursively in Yablo's construction.

We will discuss here problems with this construction.

(1) "Synchronization"

(1.1) We have a conflict between the diamonds starting at $x_{1,1}$ and $x_{1,2}$ and the diamond starting at x_0.

If $x_0 + \bigwedge$, then $x_{1,1} - \bigvee$ and $x_{1,2} - \bigvee$. As the choices at $x_{1,1}$ and $x_{1,2}$ are independent, any branch $x_0 - x_{1,1} - x_{3,1} - x_2$, $x_0 - x_{1,1} - x_{3,2} - x_2$, $x_0 - x_{1,1} - x_2$ combined with any branch $x_0 - x_{1,2} - x_{3,3} - x_2$, $x_0 - x_{1,2} - x_{3,4} - x_2$, $x_0 - x_{1,2} - x_2$ must be conflicting, thus, given $x_0 - x_{1,1} - x_2$ is negative, all branches on the left must be negative, likewise, all branches on the right must positive.

However, if $x_{1,1}$ is positive, the diamond $x_{1,1} - x_{3,1} - x_2$, $x_{1,1} - x_{3,2} - x_2$ has to be contradictory, so not both branches may be negative.

(1.2) A solution is to "synchronise" the choices at $x_{1,1}$ and $x_{1,2}$ which can be done e.g. by the formula

$x_0 = \neg x_{1,1} \wedge \neg x_{1,2} \wedge [(x_{3,1} \wedge x_{3,4}) \vee (x_{3,2} \wedge x_{3,3}) \vee (x_2 \wedge \neg x_2)]$, and

$\neg x_0 = x_{1,1} \vee x_{1,2} \vee [\neg(x_{3,1} \wedge x_{3,4}) \wedge \neg(x_{3,2} \wedge x_{3,3}) \wedge \neg(x_2 \wedge \neg x_2)] = x_{1,1} \vee x_{1,2} \vee [(\neg x_{3,1} \vee \neg x_{3,4}) \wedge (\neg x_{3,2} \vee \neg x_{3,3}) \wedge (\neg x_2 \vee x_2)]$.

(This poses a problem, as validity of $\neg x_0$ might involve validity of $\neg x_2$, so we construct an Escape path. This can be transformed into a semantical argument, so we cannot avoid considering $\neg x_2$.)

This formula is of a different type than $\bigwedge \neg$. In addition, arrows from x_0 to the $x_{3,i}$ are missing.

Even if we simplify further and consider only the paths $x_0 - x_{1,i} - x_{3,j} - x_2$ (i.e. do not consider the direct paths $x_0 - x_{1,i} - x_2$), we have the formula $x_0 = \neg x_{1,1} \wedge \neg x_{1,2} \wedge [(x_{3,1} \wedge x_{3,4}) \vee (x_{3,2} \wedge x_{3,3})]$, and $\neg x_0 = x_{1,1} \vee x_{1,2} \vee [\neg(x_{3,1} \wedge x_{3,4}) \wedge \neg(x_{3,2} \wedge x_{3,3})] = x_{1,1} \vee x_{1,2} \vee [(\neg x_{3,1} \vee \neg x_{3,4}) \wedge (\neg x_{3,2} \vee \neg x_{3,3})]$, so we can make $x_{1,1}$ and $x_{1,2}$ false, and chose ($\neg x_{3,1}$ and $\neg x_{3,3}$) or ($\neg x_{3,4}$ and $\neg x_{3,2}$), and have x_2- or x_2+, the first possibility results again in an escape path.

Note: Adding the information about $x_{1,1}$ and $x_{1,2}$ does not help: $\neg x_{1,1} = x_{3,1} \vee x_{3,2}$, $\neg x_{1,2} = x_{3,3} \vee x_{3,4}$, so $\neg x_{1,1} \wedge \neg x_{1,2} = (x_{3,1} \wedge x_{3,3}) \vee (x_{3,1} \wedge x_{3,4}) \vee (x_{3,2} \wedge x_{3,3}) \vee (x_{3,2} \wedge x_{3,4})$ still leaves the possibility ($\neg x_{3,1}$ and $\neg x_{3,3}$) by making the last disjunct true.

(1.3) Summary:

This concerns, of course, the type of constructions we analysed here: direct nesting of diamonds, and direct contradictions.

The use of diamonds, together with synchronisation problems, seems to present unsolvable problems, not only for the simple type of formulas considered here, but also to arbitrary formulas. We do not, however, have a formal proof.

There is no synchronisation problem in Yablo's construction, as there is only one choice there, and "meeting" is avoided by the universal quantifier. The latter is impossible with diamonds as we need a contradiction on both sides, thus both sides cannot be uniform.

(2) If we modify the construction, and consider for x_0+ not the direct lines $x_{1,1} - x_{3,1}$ and $x_{1,1} - x_{3,2}$, but go higher up ($x_{3,i}$ are all $+$, $\wedge$) e.g. lines $x_{4,1} - x_2$, then this will not work either, as one way to go to $x_{4,1}$ is positive, the other is negative, so the paths $x_{1,1} - x_{3,1} - x_{4,1} - x_2$ and $x_{1,1} - x_{3,2} - x_{4,1} - x_2$ cannot both be positive as required.

(3) If we try to go higher up, e.g. to $x_{5,3}$ and $x_{5,4}$ we have again an $\vee$, and need a contradiction in the diamond ($x_{3,2}$, $x_{5,3}$, $x_{6,2}$, $x_{5,4}$), so we are in the same situation as with the diamond ($x_{1,1}$, $x_{3,1}$, $x_{4,1}$, $x_{3,2}$).

(4) (If we create after $x_{3,1}$ or $x_{3,2}$ new branching points, these points will be $-\bigvee$ either for x_0+ or for $x_{1,1}+$, so one case will fail.)

(5) Back to the triangles like $x_{1,1} - x_{3,2} - x_2$, $x_{1,1} - x_2$.

The probably simplest way would be to create an intermediate point, say $x_{1,1,a}$ between $x_{1,1}$ and x_2 (here with $x_{1,1} \not\to x_{1,1,a} \not\to x_2$, on the right hand side of the diagram with $x_{1,2} \not\to x_{1,2,a} \to x_2$) to create a situation similar to the diamonds discussed here.

We did not investigate this any further, as it is outside our framework.

(6) We give a tentative, abstract approach to the problem, allowing a mixture of different types of contradictions, in Remark 5.6.

Remark 5.6.

In a general construction, we may use a mixture of different contradictory cells, Yablo triangles, diamonds (if we can solve the associate problems), etc.

Thus, it is probably not the right level of abstraction to consider full CCS's (see Definition 3.1), as they might be composed of CC's of different types.

We will, however, keep the following:

(1) the formulas will be basically of the form $\bigwedge \neg$,

(2) x_0 is the head of a contradiction, so x_0+ is impossible,

(3) every path from x_0 will lead to (the head of) a new contradiction, and this path will be negative, so we make x_0- impossible, too.

Consequently:

- if x_0+, and a negative path σ leads from x_0 to the head x_1 of a new contradiction, then $x_1 - \bigvee$, so we have to form new negative paths σ' from x_0 to contradict all x'_1 reached from x_1 to build contradictions starting at x_1. These new x'_1 will have the same properties as x_1, leading to an infinite (in depth and width) construction as detailed in Construction 3.2, Diagram 3.3, and Diagram 3.4.

  Of course, we might have to add new properties, like "synchronisation".

In Section 6 we discuss additional problems which arise if we do not construct contradictions as soon as possible, preventing local solutions.

Diagram 5.5. *Nested Diamonds 2*

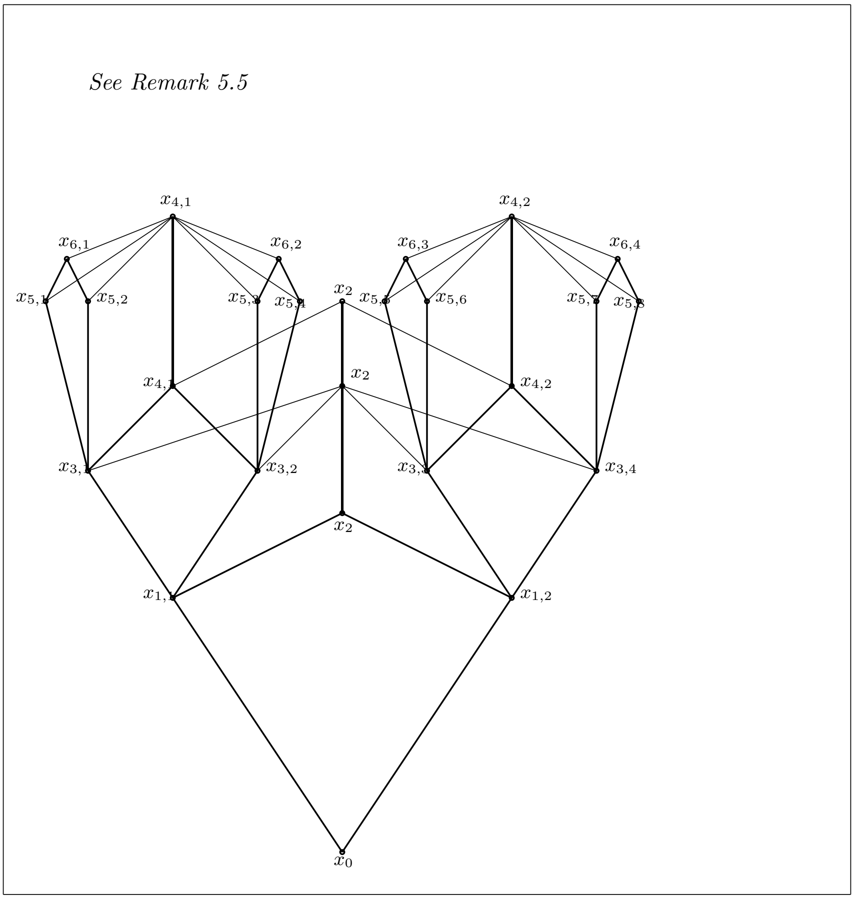
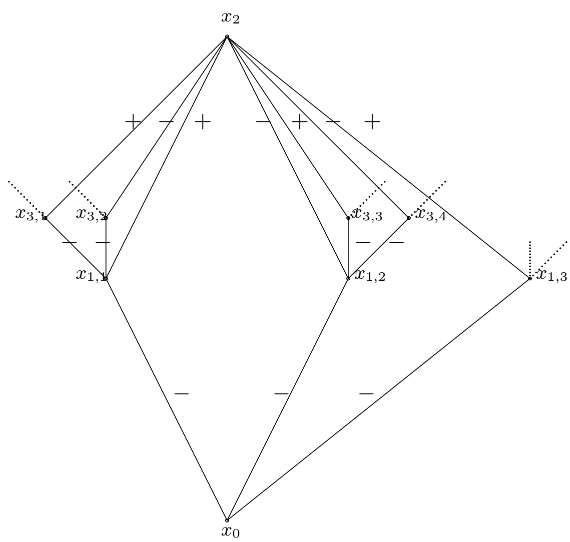


Diagram 5.6. *Nested Diamonds 2, Details*

Example for synchronisation, see Remark 5.5



Adding $x_{1,3}$ etc. does not help, as we also add the same problems again.

6 Illustration of (Finite) Procrastination

6.1 Introduction

This section is about the original Yablo construction and its modifications, postponing contradictions. Throughout, we work with Yablo's order (of the natural numbers).

In Yablo's construction, full transitivity guarantees contradictions. If full transitivity is absent, we may still have contradictions, provided we have "enough" transitivity. We discuss this here.

The most important problem seems to be to find a notation which is easy to read and write, expressing the alternation between $\forall$ (or $\bigwedge$) and $\exists$ (or $\bigvee$, arbitrary choice). We do this as follows:

(1) vertical lines beginning at some x illustrate all $y > x$, they express $\forall y.y > x$.

(2) diagonal lines to the left starting at x and going to x' express the choice of x' for x, i.e. $\exists x'.x' > x$.

First, we consider the meaning of nested quantifiers in our context. This is obvious, but writing it down helps.

We did not elaborate all details, but hope that these notes suffice to illustrate the important cases and considerations.

Note that the situation also suggests - in hindsight - a game theoretic approach, as taken e.g. in [5].

Remark 6.1.

(Trivial)

Consider the order in the graph as in Yablo's construction. Quantifiers are mostly restricted, and range only over all elements bigger than some reference point: $\forall y.y > x \ldots$, etc.

In the case of mixed quantifiers, essentially the last one decides the meaning. If the chain of quantifiers ends by $\forall$, then the property holds for an end segment of the graph, if it ends by $\exists$, then we only know that there is still some element where it holds. In the case $\forall\exists$, this means that there are cofinally many elements where is holds.

Some examples for illustration:

(1) $\exists x \forall y$: all y from $x+1$ onward have a certain property, this is a full end segment of the natural numbers.

In particular, this set has a non-empty intersection with any cofinal, i.e. infinite, sequence.

(2) $\forall x \exists y$: this might e.g. be the next prime after x, the next prime $+1$, etc.

Thus, the intersection with a cofinal sequence might be empty, but not the intersection with an end segment.

(3) $\exists\forall\exists$: this is like $\forall\exists$, starting at a certain value, again, the intersection with a cofinal sequence may be empty.

(4) $\forall x \exists y \forall z$: this is like $\exists y \forall y$, for all x there is a start, and from then onward, all z are concerned, i.e. we have a full end segment.

(5) Thus, $\forall\exists \cap \exists\forall \neq \emptyset$

(6) Thus, $\forall\exists \cap \forall\exists = \emptyset$ is possible, see above the primes and primes + 1 cases.

(7) $\forall\exists$: it helps to interpret this as $f(a)$ for all a in $\forall$, similarly $\forall\exists\forall\exists$ as $f_1(a)$, $f_2(b)$ etc. Chose first a, then $f_1(a)$, then for all $b > f_1(a)$ $f_2(b)$.

(8) In the following diagrams, the diagonal lines describe the choice functions $f(x)$ corresponding to $\exists$.

In Yablo like diagrams it sometimes helps to consider "local transitivity", which we introduce now.

Definition 6.1.

$\langle x_n, x_{n'}\rangle$ is locally transitive iff
(1) $x_n \not\twoheadrightarrow x_{n'}$ and
(2) $\forall x_{n''}$ s.t. $x_{n'} \not\twoheadrightarrow x_{n''}$, $x_n \not\twoheadrightarrow x_{n''}$ too.

Fact 6.2.

(1) If $\langle x_n, x_{n'}\rangle$ is locally transitive for some $x_{n'}$, then x_n+ is impossible.

(2) Let $m < n$, and

$\forall\ x_n$ s.t. $x_m \not\twoheadrightarrow x_n \ \exists\ x_{n'}$ s.t. $\langle x_n, x_{n'}\rangle$ is locally transitive,

then x_m- is impossible.

(3) Thus, we need (1) and (2) for x_0.

6.2 Discussion of More Complicated Cases

Example 6.1.

See Diagram 6.1.
We consider the case x_i+.

(1) Notation:

The nodes are ordered by the natural order of ω. We have here partial transitivity of $\not\twoheadrightarrow$ only, with $x \not\twoheadrightarrow x+1$ as basis.

$x \ll y$ means: $x < y$ and $x \not\twoheadrightarrow y$.

(2) The vertical lines show the nodes above the bottom node.

A full circle at a node expresses that there is an arrow $\not\twoheadrightarrow$ from the bottom node to this node, an empty node expresses that there is no such arrow. No circle expresses that the case is left open (but usually there will be an arrow) - see the text below.

The dotted slanted lines indicate choices. E.g., if x_1-, then there is some x_2, s.t. $x_1 \ll x_2$ (and thus x_2+).

(3) We want to show that x_0+ is contradictory.

(3.1) In Yablo's construction, we find a contradiction going through x_0+1, as a matter of fact, through all $x_1 > x_0$. Thus, chose any $x_1 > x_0$, then $x_1 - \bigvee$, so we have to find for all $x_1 \not\twoheadrightarrow x_2$ an arrow $x_0 \not\twoheadrightarrow x_2$, which, of course, exists.

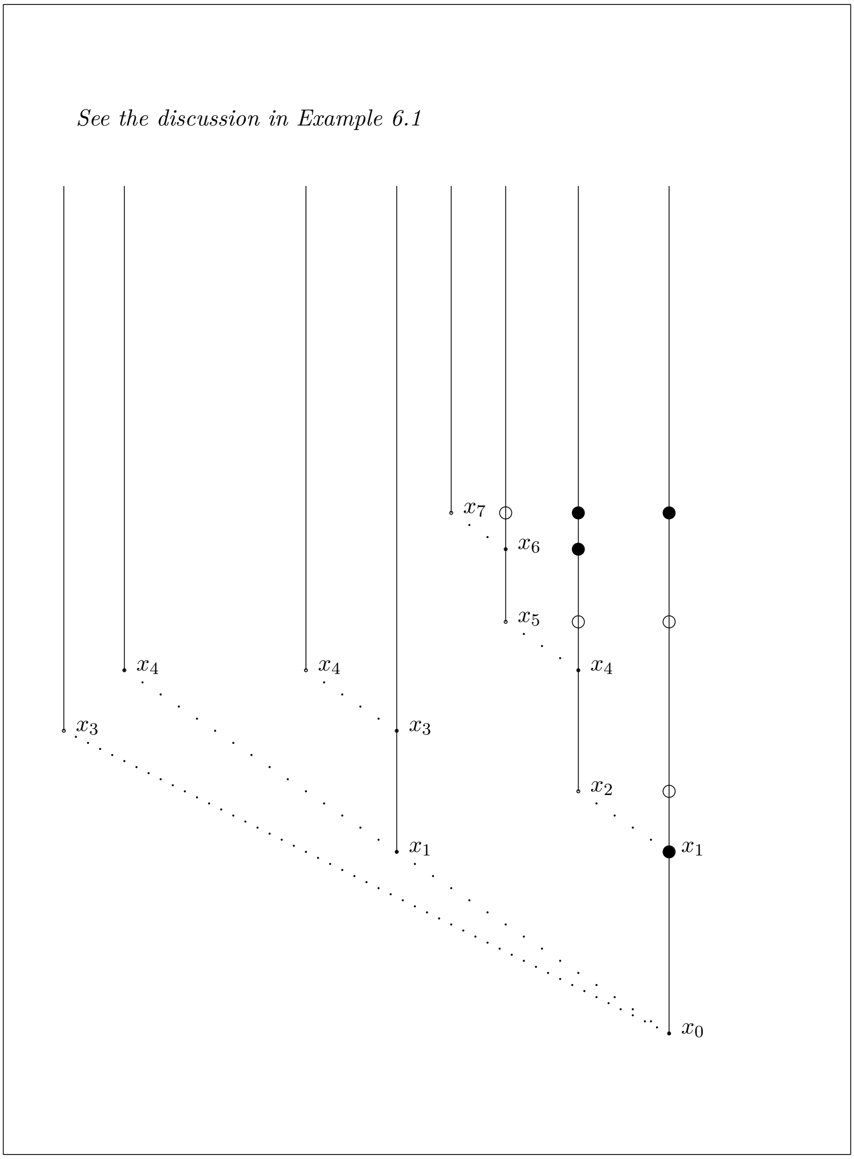
(3.2) In our diagram, we try to find a contradiction through x_1.
Suppose $x_0 \ll x_1$, so for x_0+, $x_1 - \bigvee$, and for all x_2 s.t. $x_1 \ll x_2$ we look for an arrow $x_0 \not\twoheadrightarrow x_2$, i.e. for $x_0 \ll x_2$, transitivity for $\ll$.
This does not hold, see the empty circle at x_2 in the x_0 column. So x_1 cannot be a suitable knee in the contradiction for x_0.

(3.3) We try to "mend" x_2 now, and make it contradictory itself.
Suppose $x_2 \ll x_4 \ll x_5$, then we want $x_2 \ll x_5$, but x_5 has an empty circle in the x_2 column, so this does not hold.
Suppose $x_2 \ll x_6 \ll x_7$. As noted by the full circle at x_7 in the x_2 column, $x_2 \ll x_7$.
So, for this one choice of x_7 at x_6, we have a contradiction at x_2. To have a contradiction for x_2 in all choices at x_6, we need: if $x_6 \ll y$, then $x_2 \ll y$ has to hold. This is, of course transitivity for $x_2 \ll x_6 \ll y$.
(As indicated by the full circle for x_7 in the x_0 column, we also have a contradiction between $x_0 \ll x_7$ and the path of length 4 $x_0 \ll x_1 \ll x_2 \ll x_6 \ll x_7$ - for this path, and neither for all y such that $x_6 \ll y$, nor for all y such that $x_1 \ll y$.)
Note that this is not necessary for all x_6 with $x_2 \ll x_6$, one such x_6 suffices to show that x_2+ is contradictory.

(3.4) Suppose we do not have some x_1 with $x_0 \ll x_1$ s.t. for all x_2 with $x_1 \ll x_2$ we also have $x_0 \ll x_2$, then we can look for another, better, x_1, or try to mend the old x_1.
Suppose we have $x_0 \ll x_1 \ll x_2$, but not $x_0 \ll x_2$. We make now x_2 itself contradictory, as described above in (3.3).
This repair possibility holds, of course, recursively.
If the procedure fails (for all x_1, x_6 etc.) we construct an escape path which shows that x_0 is not contradictory.

(3.5) Apart from the remark in (3.3) above, we do not discuss here more complicated contradictions with paths longer than 1, as any contradicting path, e.g. $x_0 \ll x_1' \ll x_1'' \ll x_2$ would involve a new "OR", here at x_1', so we have new branchings, and no control over meeting x_2 - unless the new branchings also meet x_2.

Diagram 6.1. *Procrastination*



References

[1] T. Beringer, T. Schindler, "A Graph-Theoretical Analysis of the Semantic Paradoxes", The Bulletin of Symbolic Logic, Vol. 23, No. 4, Dec. 2017

[2] L. Rabern, B. Rabern, M. Macauley, "Dangerous reference graphs and semantic paradoxes", in: J. Philos. Logic (2013) 42:727-765

[3] K. Schlechta, "Remarks on an article by Rabern et al.", arXiv 1808.01239

[4] K. Schlechta, "Truth and Knowledge", College Publications, Rickmansworth, UK, Series "Studies in Logic and Argumentation" 2022

[5] Michal Walicki, "There are only two paradoxes", arXiv:2309.03137v1

[6] S. Yablo, "Grounding, dependence, and paradox", Journal Philosophical Logic, Vol. 11, No. 1, pp. 117-137, 1982

Received October 2023

Evolutionary Temporal Logic for Modelling Many-Lives Argumentation Networks

D. M. Gabbay
King's College London and University of Luxembourg
dov.gabbay@kcl.ac.uk

Gadi Rozenberg
Ashkelon Academic College,
President Israel Association of Clinical Therapists

Abstract

This paper deals with the temporal aspects of the many-lives argumentation networks. The many-lives idea comes from modelling the reasoning behaviour of sex offenders, which required argumentation systems where each argument x has a natural number $M(x)$, indicating how many live attackers are needed to ensure that x is out. The temporal aspect associated with such applications (in general: how many complaints are required to take x out) is that the attackers come at different times. It seems that traditional temporal logic is unable to properly deal with such behaviour and a new type of temporal logic is required.

We call it "evolutionary temporal logic". Thus argumentation many-lives systems inspire new developments in temporal logic.

1 Introduction

This introductory section explains the ideas and results of this paper. It describes two related but independent components in formula argumentation. The idea of many-lives and the idea of evolutionary temporal argumentation. From the formal point of these two ideas are independent, from the applicational/pragrmatic point of view they are strongly related in the sense that they appear strongly intertwined in a major application area of reasoning and modelling the argumentation logic of sex offenders.

The two components are the following:

Many-lives component. In formal argumentation developed and studied Dung style [10, 11], there is the notion of attack of argument x on argument y (notation $x \twoheadrightarrow y$) and the property that if one attacker (say x) is live ("in") then the target y is dead ("out"). The notion of many-lives is an index given to any target argument y (a notation $M(y)$) which requires that at least $M(y)$ live ("in") attackers $x_1 \twoheadrightarrow y, \ldots, x_{M(y)} \twoheadrightarrow y$ on y to be able to render y dead ("out").

Evolutionary temporal logic component. Intuitively for the purpose of this Introduction, think of a time sequence of finite logical databases, $\Delta_1, \Delta_2, \ldots, \Delta_n, \ldots$ in which the constraints on the logical properties of Δ_{n+1} depends on the nature of $\{\Delta_1, \ldots, \Delta_n\}$. So there must be some algorithmic function $\mathbb{F}(\Delta_1, \ldots, \Delta_n)$ which constrains Δ_{n+1}. Examples will be given in the next subsection.

These two components strongly appear in argumentation, interacting in many ways. The simplest example for such an interaction is that when argument y with $M(y) > 1$ lives is attacked by an "in" argument x (i.e., $x \twoheadrightarrow y$) at time n, then at time $n + 1$ we have $M'(y) = M(y) - 1$.

Such interaction is very common in the area of complaints and sex offender abuse allegations. We know that one complaint is not sufficient to open a formal investigation but in many cases more and more independent complaints show up in time and there is a number M of complaints which will force action to be taken.

1.1 Motivating examples

Example 1.1 (Mr Malkinson Case)**.** *. This case is real and actually happened in 2023 (see next Example 1.1.*

Seventeen years ago, in the year 2006, Mr Malkinson was convicted of committing rape and was sentenced to prison, on the basis of circumstantial evidence and one witness. There was no DNA evidence at the time.

This year (2023) new DNA evidence emerged and on the basis of this new evidence, Mr Malkinson was declared not guilty (backwards from the year 2006) and released from prison (in the year 2023).

The law required prisoners to pay rent to the government but if they are guilty and are imprisoned, they do not have to pay rent.

Since in 2023 Mr Malkinson was declared not guilty from 2006 the prison system asked for rent for providing free lodging for him. This outraged public opinion and the law was cancelled retroactively backwards in time.

Example 1.2 (Story of the article in the Daily Mail)**.** *https://www.dailymail.co.uk/news/article-12377133/Innocent-man-wrongly-*

jailed-17-years-rape-didnt-commit-WONT-pay-staying-prison-Justice-Secretarys-intervention-following-outrage-shocking-miscarriage-justice.html

Wrongly convicted people WON'T have to pay for staying in prison after shock miscarriage of justice involving man jailed for a rape he didn't commit sparked ministerial intervention. JACK WRIGHT, PUBLISHED: 00:00 BST, 6 August 2023 | UPDATED: 09:27 BST, 6 August 2023.

The innocent man who was wrongly jailed 17 years for a rape he did not commit will not have to pay living costs covering his time in prison following a dramatic intervention by Rishi Sunak's Government. Justice Secretary Alex Chalk KC made the change covering wrongly convicted people with immediate effect on Sunday after the miscarriage of justice case of Andrew Malkinson sparked outrage.

Mr Malkinson spent 17 years in prison for a rape he did not commit, and appeal judges quashed his conviction last week after DNA linking another man to the crime was produced. The 57-year-old expressed concern that the rules meant expenses could be deducted from any compensation payment he may be awarded to cover the costs of his jail term. Downing Street indicated that the Prime Minister believed the deductions were unfair amid demands to drop the charges.

Mr Chalk has now updated the guidance dating back to 2006 to remove them from future payments made under the miscarriage of justice compensation scheme. The reform to eligible cases was broadly welcomed, but there were calls to pay back the money already deducted from wrongly convicted individuals

Let us now describe the flow of events of this story. Since the present paper is on Temporal Evolutionary aspects of argumentation networks, we will use argumentation notation and use this example to illustrate the technical details of this paper.

Remark 1.3.

1. *An argumenation network with attack and (deductive) support has the form* $(\mathbf{S}, \mathbf{R}, \mapsto)$ *where* $\mathbf{S}$ *is a set of arguments atoms* $\mathbf{R} \subseteq \mathbf{S} \times \mathbf{S}$ *is the attack relation and* $\mapsto$ *is a deductive support relation.* $\mapsto$ *is a subset of* $\mathbf{S} \times \mathbf{S}$*.* $\mapsto$ *is really a logic, say classical propositional logic provability* $\vdash$*.*

   We hasten to comment that we are giving here a very special case definition of bipolar argumentation network (i.e., with attack and support) where the support is deductive support. This is sufficient for analysing our example.

2. *To be able to explain/formalise better the Mr Malkinson example, we also add to our argumentation language the classical negation* $\neg$ *and the classical conjunction* $\wedge$*.*

So we can write $\neg x$ *and* $x \wedge y$*, when* x *and* y *are arguments. Of course we must impose the restriction that only one of* $\{x, \neg x\}$ *can appear in any network.*

3. *Thus if* x, y, z *are arguments and we write*

$$(x \wedge y) \mapsto z.$$

We mean that $(x \wedge y) \vdash z$ *in classical logic.*

When we write

$$(x \wedge y) \twoheadrightarrow z$$

we mean that $\{x, y\}$ *jointly attack* z

Note that logic is not involved in this attack, the attack is generated from the meaning/content of x, y, z*, as they appear in the application to be modelled.*

Thus for $\{x, y\}$ *to force* z *to be "out", then both* x *and* y *must be "in".*

Also note that if we have a network with arguments

$$\{\neg x, x \twoheadrightarrow y\}$$

we have by the meaning of $\neg x$ *that* x *must be "out" and therefore* y *is "in". This is not the traditional Dung language but equivalent to the network*

$$\begin{gathered} x \twoheadleftarrow\!\!\twoheadrightarrow \neg x \\ x \twoheadrightarrow y \end{gathered}$$

with the choice $\neg x$ *to resolve the loop.*

4. *Note that our purpose in this subsection 1.1 is only to provide a motivating example for the machinery of evolutionary argumentation networks. We chose a real example from the Daily Mail to formalise. For this we need to present the example as a network in time with attack and deductive support.*

   Since the example is a simple real life example we do not need to develop formally the argumentation theory of attack + (deductive) support.

Definition 1.4.

1. *Let the following denote arguments related to the Daily Mail story of Example 1.2.*

   1.1. *GM = Mr Malkinson is guilty of rape*

1.2. *LPR = The law says that prison residents have to pay rent to the prison authorities*

1.3. *PM = Mr Malkinson is a resident in prison*

1.4. *RM = Mr Malkinson has by law to pay rent for his residence in prison.*

1.5a. *By law, resident prisoner must pay rent. This is represented by* $(PM \wedge LPR) \mapsto RM$*. We need jointly PM and LPR to get RM.*

So if the law is changed and we have $\neg LPR$ *then we cannot get* RM*.*

1.5b. *Guilty prisoners need not pay rent (This is represented in argumenation by the attack arrow* $\twoheadrightarrow$ *in*

$$\begin{gathered} GM \twoheadrightarrow RM \\ (PM \wedge LPR) \mapsto RM \end{gathered}$$

1.6. *DNA = DNA was discovered proving that Mr Malkinson did not commit the rape (and is therefore not guilty of rape).*

2. *The following is the time flow of events in terms of the argumentation networks* $(\mathbf{S}, \mathbf{R})$ *available at each time interval.*

   2.1. *[2006–2022]*

   $\mathbf{S}_{2006-2022} =$ *existing arguments at the time* $= \{LPR, PM, RM, GM\}$.

   $\mathbf{R}_{2006-2022} =$ *attacks at the time:*

$$\left\{ \begin{array}{c} GM \twoheadrightarrow RM \\ (P \wedge LPR) \mapsto RM \end{array} \right\}$$

   - *The extension at the time is* $\{LPR = in, PM = in, GM = in, RM = out\}$.

   2.2. *[2003]*

   $\mathbf{S}_{2003} = \{LPR, \neg PM, RM, GM, DNA\}$.

   $\mathbf{R}_{2023} = \left\{ \begin{array}{c} DNA \twoheadrightarrow GM \twoheadrightarrow RM \\ (LPR \wedge PM) \mapsto RM \end{array} \right\}$

   - *The extension in 2023 is therefore*

$$\{DNA = in, GM = out, LPR = in, PM = out, RM = in\}.$$

Remark 1.5. *There is a problem with the description of what happens the minute the DNA proves the innocence of Mr Malkinson in 2023.*

1. *Clearly Mr Malkinson will immediately leave his residence in prison. Thus in argumentation terms PM is no longer an element of* $\mathbf{S}_{2023}$.

   Technically this means that elements of $\mathbf{S}_{t1}$ *can leave the network in* $\mathbf{S}_{t2}, t_1 < t_2$.

2. *To help us express this fact more clearly we can take out PM from the network and put in instead* $\neg PM$.

   This is a better way of doing it, as we shall see later.

3. *The second problem with the situation in 2023 is that "DNA" attacks "GM", not only in the year 2023, but the attack goes backwards in time since the conviction in 2006.*

   So in 2023, the view of history changes.

   This requires modelling in two dimensional temporal logic. The first dimension is the time of the point of view and the second dimension is the time of the event (according to the point of view). To clarify this point, we need to write the history explicitly.

   We caution the reader that the general theory of two dimensional temporal logic does not require any evolution in the view of history from time t *to time* $t + 1$. *In comparison, the view of history at time* t *in our example depends on the views and data of what happened in earlier times. So evolutionary temporal logic is a special case of two dimensional temporal logic and in fact it is so special that it needs to be formalised directly without any input/use of the theory two dimensional temporal logic.*

   In fact the theory of evolutionary temporal argumentation is even more of a special case of the theory of evolutionary temporal logic.

4. *We remark here that for the clarity of the example we assume that in 2024 the law LPR was cancelled (see Example 1.2. The law was actually changed later in 2023 but we do not want to split the year into two parts. So for clarity, we use 2024)*

Definition 1.6.

1. *Let* $\mathbb{P}$ *be a set of labels denoting points of view. It can be names of people, or moments of time, etc.*

2. *For each* $\pi \in \mathbb{P}$, *and each moment of time* t, *let* $(\mathbf{S}^{\pi}_{t}, \mathbf{R}^{\pi}_{t}, \mapsto)$ *be a deductive argumentation network (at time* t *from the point of view of* π*).*

3. *When $\mathbb{P}$ is also a set of moments of time, we can talk about evolution of the network at a fixed time t_0, through times $s_1, s_2, s_3, \ldots$. We look at $(\mathbf{S}_{t_0}^{s_j}, \mathbf{R}_{t_0}^{s_j}, \mapsto)$ for $j = 1, 2, 3, \ldots$*

Example 1.7. *Let us use Definition 1.6 to trace the evolution in the case of Mr Malkinson.*

$$\mathbf{S}_{2022}^{2022} = \{LPR, PM, RM, GM\}$$
$$\mathbf{R}_{2022}^{2022} = \left\{ \begin{array}{l} GM \twoheadrightarrow RM \\ (LPR \wedge PM) \mapsto RM \end{array} \right\}$$
$$\mathbf{S}_{2023}^{2023} = \{LPR, \neg PM, RM, GM, DNA\}$$
$$\mathbf{R}_{2023}^{2023} = \left\{ \begin{array}{l} DNA \twoheadrightarrow GM \twoheadrightarrow RM \\ (LPR \wedge PM) \mapsto RM \end{array} \right\}$$

We have a problem representing $\mathbf{R}_{2022}^{2023}$.

Intuitively the set of arguments $\mathbf{S}_{2022}^{2023}$ remains the same as $\mathbf{S}_{2022}^{2022}$.

However, we know in 2023 that the DNA argument makes Mr Malkinson not guilty (i.e. $DNA \twoheadrightarrow GM$). So GM must be out. But in the language of 2022, DNA is not present! So how can we see that GM is out?

For this reason we allowed negation $\neg$ in the language. We can write

$$\mathbf{S}_{2022}^{2023} = \{LPR, PM, RM, \neg GM\}.$$

The attacks and supports remain the same

$$\mathbf{R}_{2022}^{2023} = \left\{ \begin{array}{l} GM \twoheadrightarrow RM \\ (LPR \wedge PM) \mapsto RM \end{array} \right\}$$

Since $\neg GM$ is available, RM is not attacked and since in 2022 the 2023 point of view accepts that PM is "in" and the law "LPR" is also in. We get that M is "in" meaning that Mr Malkinson has to pay rent. This is why public opinion forced the cancellation of the law. Thus from the point of view of 2024 we have $\neg LPR$ present in $\mathbf{S}_{2022}^{2024}$.

We get that the following

$$(\mathbf{S}_{2022}^{2024}, \mathbf{R}_{2022}^{2024})$$
$$(\mathbf{S}_{2023}^{2024}, \mathbf{R}_{2023}^{2024})$$
$$\textit{and}$$
$$(\mathbf{S}_{2024}^{2024}, \mathbf{R}_{2024}^{2024})$$

cases are the same as the $(\mathbf{S}^{2023}, \mathbf{R}^{2023})$ cases except that the arguement LPR is replaced by $\neg LPR$. So in $(\mathbf{S}_{2022}^{2024}, \mathbf{R}_{2022}^{2024})$ RM is out, because $(LPR \wedge PM) \mapsto RM$ cannot be used to get RM.

Example 1.8. *This example gives a possible imaginary variation on the Mr Malkinson story, a variation which will illustrate what role the many-lives idea can play in the example.*

According to the real story, a DNA test showed conclusively in the year 2023 that Mr Malkinson was not the rapist and he was found not guilty backwards from the year 2006.

Let us consider a different possible scenario where in 2023 a new witness shows up which casts doubt about the 2006 conviction. The prosecutor is not going to hasten and find Mr Malkinson not guilty on the basis of just one new witness. The conviction has inertia and possibly many more witnesses (many-lives) are required.

Let $M = 6$ *be a reasonable number of additional new independent and solid witnesses that would force the prosecution to re-open the case and possibly declare Mr Malkinson not guilty.*

Imagine then the following variation on the sequence of temporal data:

[2022] *GM, (with* $M(GM) = 6$*)*

[2023] *GM (with* $M(GM) = 5$*),* $W1$ *and* $W1 \twoheadrightarrow GM$*.*

[2024] *GM(*$M(GM) = 4$*),* $W1, W2$ *with* $W1 \twoheadrightarrow GM; W2 \twoheadrightarrow GM$

$\vdots$

[2028] *GM is out (*$M(GM) = 0$*), and* $W1, \ldots, W6$*, with*

$$\begin{array}{l} W1 \twoheadrightarrow GM \\ \vdots \\ W6 \twoheadrightarrow GM \end{array}$$

So only in 2028 will Mr Malkinson be declared not guilty, backwards to 2006.

Remark 1.9. *This is a methodological remark about what is the kind of evolutionary temporal logic we want to use for hte case of argumentation for legal cases.*

We do not need to consider the full two dimensional temporal model. In terms of Example 1.7 we need only the following:

$$(\mathbf{S}^{2022}_{2022}, \mathbf{R}^{2022}_{2022})$$

$$(\mathbf{S}^{2023}_{2023}, \mathbf{R}^{2023}_{2023})$$

and

$$(\mathbf{S}^{2024}_{2024}, \mathbf{R}^{2024}_{2024})$$

In other words, we need to compute

$$(\mathbf{S}^{time\ k}_{time\ k}, \mathbf{R}^{time\ k}_{time\ k})$$

for $k = 1, 2, \ldots,$ *now.*

We do not need to know what time $t2$ *thought about time* $t1$*, where* $t1 < t2$*.*

This will simplify our notation considerably and we can use a single graph (argumentation network) where each element $x \in S, y \in S$ *and each attack arrow* $x \twoheadrightarrow y$ *is annotated by the moments of time up to "now" in which they exist. Example 1.1 discusses how to do it for the main Mr Malkinson example, for the years 2022, 2023, 2024.*

Remark 1.10.

1. *We are going to give a better notation for the temporal evolution of argumentation network. Before we do that, let us distill the essential features of what was going on in the Mr Malkinson real (with DNA) example.*

2. *In 2022, we had a situation of*

   1. *guilty* $\twoheadrightarrow$ *pay rent*
   2. *guilty*
   3. *law in force and resident* $\mapsto$ *pay rent*

   In 2023 the DNA evidence was introduced and we got $\neg$ *guilty because backwards in time we have*

   $$\textit{DNA of 2023} \ \twoheadrightarrow \ \textit{guilty of 2022}.$$

   So we got in 2023 that Mr Malkinson has to pay rent using the rule:

   law in force $\wedge$ *resident* $\mapsto$ *pay rent.*

   In 2024 the law in force of 2022 was cancelled. Let us represent this as an attack on the law in force by a new argument called "cancellation".

   So we have

   $$\textit{cancellation of 2024} \ \twoheadrightarrow \ \textit{law in force of 2023–2023}$$

   The above presentation shows that the reason of evolution in time are the backwards in time attacks.

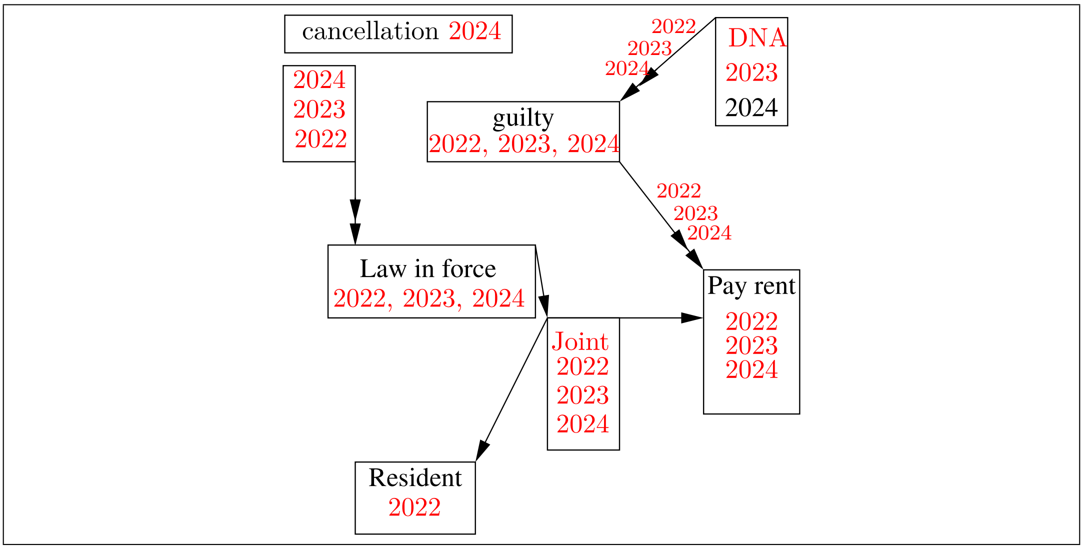


Figure 1

3. *Our model is to timestamp attackes and arguments for the time they are active. Figure 1 shows what we get.*

   Let us try to read this figure by looking at each argument and its time of existence/validity.

   cancellation: [2024]
   DNA: [2023–2024]
   Law in force: [2022–2024]
   Pay rent: [2022–2024]
   Resident: [2022]

   Note the following: The argument "Resident" is a statement of fact. It cannot be attacked. The others are legal constructs. They can be changed.

1.2 Technical introduction

Let $G = (S, R)$ be an argumentation network. This means that S is a non-empty set and $R \subseteq S \times S$ is the (attack) binary relation on S. For the purpose of this introduction, let us view $G = (S, R)$ as a directed graph R on the set S.

There are various possible operations we can perform on $G = (S, R)$. One such a general operation is a semantic function **SM** operating on $G = (S, R)$ and extracting from it a subset **CSM** $(G) \subseteq S$ using an algorithm α and a choice function on the properties of R. (See, for example, item 9 of Definition 2.1.) So $\mathbf{SM}_\alpha G$ yields a family of subsets of S and $\mathbf{CSM}_\alpha(G)$ chooses one of them.

Now we imagine that we have a temporal linear sequence of such networks. Let t be a temporal index running over $\{1, 2, 3, \ldots\}$ and for each such a t let $G_t = (S_t, R_t)$ be the network at time t of the sequence.

We can assume for the moment (but not in general) that $S_t \subseteq S_{t+1}$.

We also get a sequence $\mathbf{CSM}_\alpha G_t, t = 1, 2, \ldots$ of subsets.

From the point of view of argumentation, there are three ways of looking at the sequence $\{G_t\}, t = 1, 2, 3 \ldots$.

Timed view. This view (discussed in [4]) regards time as annotating the arguments of S, to form for each $x \in S$, an annotated argument (x, H_x), where H_x is $H_x = \{t | x \in S_t\}$. Let $S_\infty = \bigcup_t S_t$. Let $S^\sharp = \{(x, H_x) | x \in S_\infty\}$.

Define an attack relation $F^\sharp$ on $S^\sharp$ by

- $(x, H_x) R^\sharp (y, H_y)$ iff there exists a $t \in H_x \cap H_y$ such that $x R_t y$.

We thus get a timed network

$$G^\sharp = (S^\sharp, R^\sharp).$$

According to the Timed View, we are interested in studying $G^\sharp$ in a traditional Dung style way.

A major application for this view is in the area of Laws and Regulations, which keep on changing and we need to be confident that they do not contradict one another at any given moment of time.

Modal view. This is the view of [5]. We regard the time flow $\{1, 2, 3, \ldots\}$ as a modal possible world model of the form $(T, <)$, where $T = \{1, 2, 3, \ldots\}$ and $<$ is the relation of smaller than among numbers. With each "possible world" $t \in T$ we associate a classical model $(S_t, R_t) = G_t$. We use modal operators and temporal operators on this system $(T, < S_t, R_t), t \in T$ and possibly define arguments and attacks using this modal language and S_t and R_t.

Evolutionary view. This view regards the future as open and has not happened yet and we can influence it by stipulating some actions and rules. For example, if an element $x \in S_1$ has not attacked any other elements in $t = 1, 2, 3, \ldots, 10$ then we stipulate that x is a "peaceful" element and expect that it not be attacked at time 11.

Let us now continue and focus on the evolutionary view of the sequence $G_t = (S_t, R_t)$.

A major application for this view is in laws and regulations which keep on changing and we need to be sure that they do not contradict one another at any given moment.

Ordinary linear temporal logic can deal with the sequence $G_t = (S_t, R_t), t = 1, 2, 3, \ldots$ viewed as a temporally changing graph or with the sequence $\mathbf{CSM}(G_t), t = 1, 2, 3, \ldots$ viewed as a temporally changing classical model.

This is the modal view. However, if we want to deal with both sequences and deal with the effect that G_{t+1} has on G_t via the algorithm α and the choice function **CSM**, then we are dealing with a case of evolutionary temporal logic.

This is best explained by an example. Consider the network of Figure 15 (where $\twoheadrightarrow$ denotes attack). The problems with the temporal analysis of this figure are discussed at Example 3.3. For the purpose of this introduction, it is sufficient to say that for the purpose of continuity we must require that the algorithm gives two possible choices, $\{a\}, \{b\}$ for the figure at time one and the choice function for time 1 is allowed to choose one of these two options for example $\mathbf{CSM}(\text{time } 1) = \{a\}$. If this is the case then for the sake of continuity (if we wish to stipulate continuity) we expect **CSM** to yield $\{a, c\}$ for the figure at time 2. However, without the continuity principle it could choose $\{b, c\}$.

This coherence–continuity–rationality postulate becomes important when the graph is annotated. Imagine a graph $\{S, R, M\}$ where M is an annotation function giving a natural number $n = 0, 1, 2, \ldots$ for any $x \in S$. $M(x)$ means how many-lives x has (which algorithmically implies under the principle that each living attacker can take away one life) how many living attackers are needed to "neutralise it" (i.e. to reduce $M(x)$ to 0). The algorithm α for networks with many-lives then works on (S, R, M) taking into account the values of M and of R and yields several choices of a new network of the form (S, R, M_i^*) (i is the index of several choices). The choice function chooses one of them, say $\mathbf{SCM}(S, R, M) = (S, R, M^*)$.

Again, if we have a sequence of (S_t, R_t, M_t), we get a sequence of $\mathbf{SCM}(S_t, R_t, M_t) = (S_t, R_t, M_t^*)$. We need to formulate principles of continuity and choice connecting the two sequences.

A natural view of both sequences is to consider their Tensor product sequence:

$$\begin{array}{ll} G_1 \otimes \mathbf{CSM}(G_1), & \text{Time } 1 \\ \\ G_2 \otimes \mathbf{CSM}(G_2), & \text{Time } 2 \\ \vdots & \end{array}$$

Evolutionary temporal logic can talk about this sequence.

Example 1.11. *To further illustrate the need for evolutionary temporal logic is the notion of flow product [8]. Figure 2 explains it all. It is the tensor product of the*

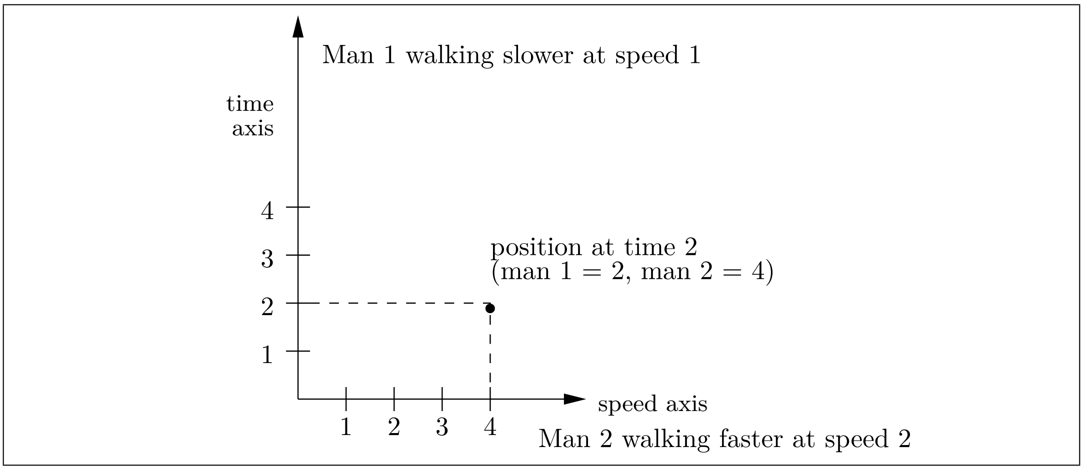


Figure 2

two axis, the horizontal for Man-1 and the Vertical for Man-2.

There are two men walking at different speeds. Man 1 at speed 1 meter per second and Man 2 at speed 2 meters per second.

Remark 1.12. *We now offer a methodological remark for the perceptive reader. The reader may ask, why introduce a new concept of evolutionary temporal logic, when all we need is the well-known two dimensional temporal logic?*

This has already been discussed in item 3 of Remark 1.5 in the context of the Malkinson Example as well as Figure 1, but it is better to revisit this discussion again.

My answer to that is that the "two" dimensions is deceptive. If you consider the case of Figure 15, it is actually a choice of two paths out of four possible histories. You need a dimension for each possible choice.

Consider Figure 3. We use the notation:

Time 1, Time 2, ... to indicate points in the Time axis. (Compare with Figure 2.) At each of such time points the following predicates get truth values:

$\mathbf{d}(x) : x$ *exists*
$in(x) : x$ *is in*
$out(x) : x$ *is out*
H_i*: temporal history* i.
Time t*:*
Path i*: composite path* i.

With the principle of continuity only H_3 *and* H_1 *are allowed.*

Time 2	$\mathbf{d}(c)$ $\mathrm{in}(a)$ $\mathrm{out}(b)$	$\mathbf{d}(c)$ $\mathrm{in}(a)$ $\mathrm{out}(b)$	$\mathbf{d}(c)$ $\mathrm{out}(a)$ $\mathrm{in}(b)$	$\mathbf{d}(c)$ $\mathrm{out}(a)$ $\mathrm{in}(b)$
Time 1	$\neg\mathbf{d}(c)$ $\mathrm{in}(a)$ $\mathrm{out}(b)$	$\neg\mathbf{d}(c)$ $\mathrm{out}(a)$ $\mathrm{in}(b)$	$\neg\mathbf{d}(c)$ $\mathrm{out}(a)$ $\mathrm{in}(b)$	$\neg\mathbf{d}(c)$ $\mathrm{in}(a)$ $\mathrm{out}(b)$
History	H_1	H_2	H_3	H_4

Figure 3

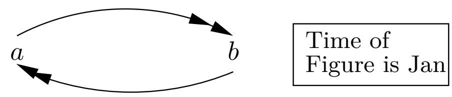


Figure 4

To consider what history is allowed, we need to list all histories! So we do not have here the traditional two dimensional temporal logic.

We now offer a better evolutionary temporal language, better than the two dimensional tensor product. We begin with a running example.

Example 1.13. *Consider another example from argumentation. The temporal story is as follows.*

1. *At January there are two arguments attacking each other. The graph for January is Figure 4. We can take the view that the figure has a graph plus a box indicating the time of its existence.*

   So, if the figure came into existence at time "January". This includes the nodes a and b and the attacks $a \twoheadrightarrow b$ and $b \twoheadrightarrow a$. So we can annotate components by the time each component exists and this way we do not need the box. We get Figure 5.

2. *In February a new attacker came into existence, call it c. The figure for February is Figure 6.*

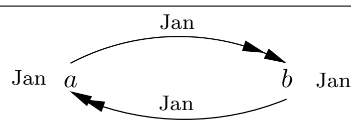


Figure 5

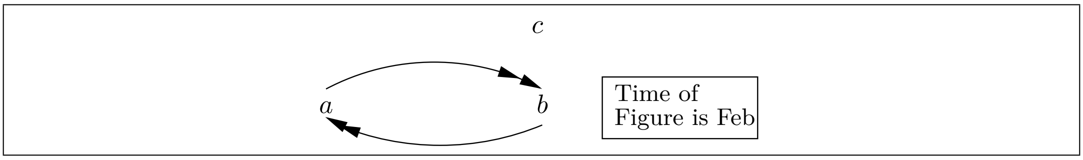


Figure 6

We can write the time of existence of components in Figure 7.

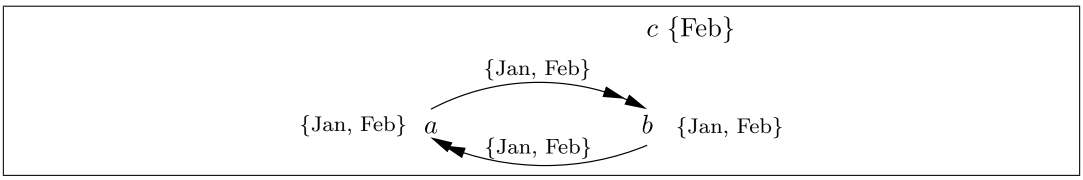


Figure 7

Figures 4 and 6 can be retrieved from Figure 7 by respectively collecting all items labelled Jan (resp. Feb) and forming the respective graphs.

3. *Let us offer you a narrative for the above graph and continue to time of March. In January a attacked b claiming that b raped a. b counter-attacked a by claiming that there was consent. Thus the January Figure 4 was created. In February a new possible witness appeared on the scene, ready to possibly testify against b (attack b). b's lawyers quickly paid c off and in March c was no longer available to testify. Furthermore by April, b settled with a as well and a simply went abroad, but did not withdraw her accusation. Thus in April a was not in the graph. However, public opinion and pressure forced a to come back in May, but did not apply any pressure on c to show up again in May because it has not attacked b. See also Example 3.13*

   We get the following May graph annotation in Figure 8.

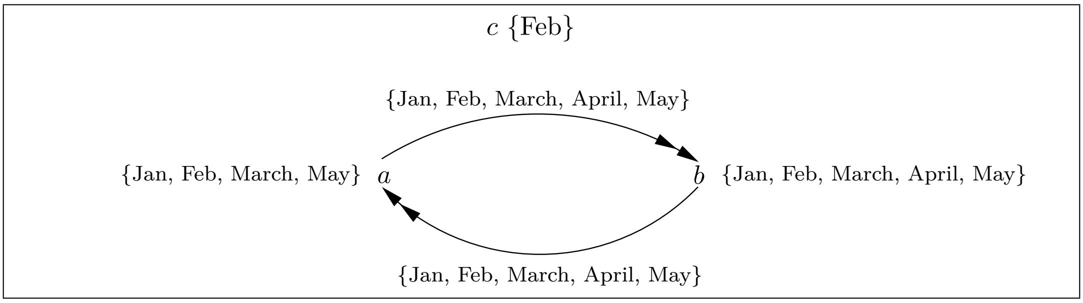


Figure 8

Definition 1.14.

1. *An evolutionary argumentation network has the form (S, R, T), where (S, R) is a directed graph, namely S is a non empty set of nodes and R is a binary relation on S, and T is a function giving to each element x is $S \cup R$ a set $T(x)$ of numbers from $\{1, 2, \ldots\}$.*

2. *Given an evolutionary argumentation network and a natural number n, we define the network existing at time n as the network $G_n = (S_n, R_n)$ where*

$$S_n = \{x \in S | n \in T(x)\}$$
$$R_n = \{(x, y) \in R | n \in T(x, y)\}$$

Remark 1.15.

1. *Note that according to item 2 of Definition 1.14, we may have in (S_n, R_n) attack arrows without any attacker or target or both.*

2. *Note that the traditional Dung machinery for defining extensions works also with networks of Definition 1.14 despite Item 1 in this Remark.*

3. *Note that item 1 of definition 1.14 may as well define a timed argumentation network, for the Timed View p. 11. However item 2 of this definition already moves towards the Evolutionary View p. 11.*

Example 1.16. *Let us give one more example showing how the sequences of $G_n = (S_n, R_n)$ and $\mathbf{CSM}(G_n)$ interact. We allow for many-lives.*

Consider the following two sequences in Figure 9. The sequences give general graphs where the arguments x also have many-lives $M(x)$

Think of a court case where b is accused of being a sex offender by a. b says a is lying. The court believes b and passes a verdict that b is innocent (option 2). Now at time 2, we get another victim which attacks b. Our questions are the following:

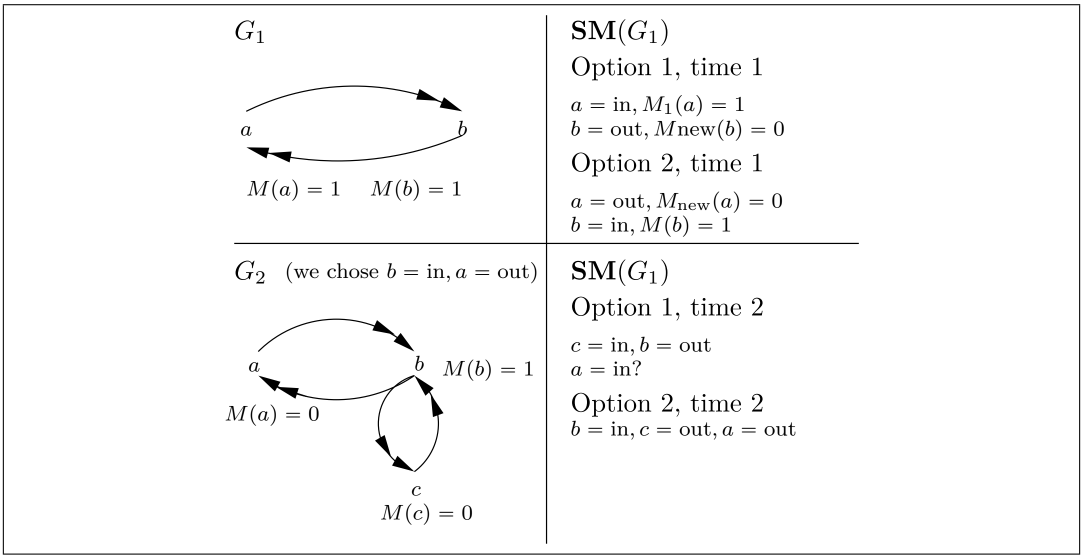


Figure 9

- *Do we believe c?*
- *Do we rely on the verdict at time 1 and say we decided b is innocent so the case is closed and we do not believe/dismiss c?*
- *What if 10 other $c_1, \ldots, c_{10}$ come at time 2 and attack b? Do we now believe that b is guilty?*
- *Do we give b more lives and believe $\{c_i\}$ only if there are at least two c's? (I.e., ($M(b) = 2$ at time 2)?*

We can formulate new policies depending on what options we choose in the past time to decide what we choose at the present time.

Remark 1.17. *We conclude this Section by explaining why linear evolutionary temporal logic is sufficient for our considerations as opposed to say open future tree temporal logic.*

The reason is very simple. Even when we have branching time, at any point on any branch the evolutionary aspect have to do with looking into the past, and the past on any branch is always linear.

Also, even when we have loops in the past , because we are dealing with argumentation, the loops will be resolved by choice at the time in the past which enable us to continue without this loop into the future.

2 Background and orientation

This section gives background from abstract argumentation and from temporal logic and identifies why traditional temporal logic cannot deal with the many-lives argumentation, evolutionary aspects and explains intuitively our proposed possible solutions.

2.1 Background and concepts from abstract argumentation

This subection presents, for the convenience of the reader, some basic concepts of what we called traditional argumentation theory. Such systems contain attacks only. We refer to such system as Argumentation with Attack only. One can also add support to the system and in this case we get systems of Argumentation with Attack and Support. We shall then explain in what way the systems required for this paper depart from the traditional ones.

There are two ways to present the semantics for argumentation with attack, the traditional set theoretical approach and the Caminada labelling approach. For the mapping connections between the two approaches, see [10, 11]. Let us briefly quote the traditional set theoretic approach:

Definition 2.1.

1. *We begin with a pair (S, R), where S is a nonempty set of points (arguments) and R is a binary relation on S (the "attack" relation, we read xRy as x attacks y). In the diagrams and figures we use the notation $a \twoheadrightarrow b$, to denote aRb.*

2. *Given (S, R), a subset E of S is said to be conflict free if for no x, y in E do we have xRy.*

3. *E protects an element $a \in S$, if for every x such that xRa, there exists a $y \in E$ such that yRx holds.*

4. *E is admissible if E is conflict free and protects all of its elements.*

5. *E is a complete extension if E is admissible and contains every element which it protects.*

6. *A subset E is a stable extension if E is a complete extension and for each $y \notin E$ there exists $x \in E$ such that xRy.*

7. *E is the grounded extension if it is the unique minimal complete extension (it exists, see Lemma 2.2).*

8. *E is a preferred extension, if E is a maximal (with respect to set inclusion) complete extension.*

9. *A Semantics is a (metalevel) property* **S** *of extensions, such as being stable, or being grounded or being preferred or being complete. Thus we can talk about* $\mathbb{S}$*-Semantics, (stable semantics, grounded semantics and preferred semantics or complete semantics) where we consider only* **S**- *extensions.*

Lemma 2.2. *For any network* (S, R) *there exists a grounded extension (which may be empty).*

Proof. This can be proved, using set theoretical methods, see [11]. ■

We can also present the complete extensions of $A = (S, R)$, using the Caminada labelling approach, see [11].

Definition 2.3. *A Caminada labelling of* S *is a function* $\lambda : S \mapsto \{in, out, und\}$ *such that the following holds.*

(C1) $\lambda(x) = in$*, if for all* y *attacking* $x, \lambda(y) = out$.

(C2) $\lambda(x) = out$*, if for some* y *attacking* $x, \lambda(y) = in$.

(C3) $\lambda(x) = und$*, if for all* y *attacking* $x, \lambda(y) \neq in$*, and for some* z *attacking* $x, \lambda(z) = und$.

Lemma 2.4.

1. *A consequence of (C1) is that if* x *is not attacked at all, then* $\lambda(x) = in$.

2. *Given an extension* E *let* λ_E *be defined by* $\lambda_E(x) = \{$ *in if* $x \in E$*, out if for some* $y \in E$ *we have* yRx*, and undecided otherwise*$\}$. *Conversely given a* λ*, define* E_λ *to be* $\{x | \lambda(x) = in\}$.

3. *Any Caminada labelling yields a complete extension and vice versa.*

4. *Any {in, out} Caminada labelling (i.e. with no "und" value) yields a stable extension and vice versa.*

5. *Set theoretic minimality or maximality conditions on extensions E correspond to the respective conditions on the "in" parts of the corresponding Caminada labellings.*

Proof. See [11]. ■

Remark 2.5 (Convenient Notation). *In anticipation of future examples and discussions and sometimes for the sake of language and expression or in anticipation of the concept of many-lives , we also use instead of the "in", "out", "extension " words used in Definition 2.1 and Definition 2.3 we use the the words below (think of a cat having nine "lives" and can "survive" 8 "deaths" and still be "alive"):*

(*) x = *"out", or* x *is "out", or* x *is "dead", or* x= *"out/dead", or* x *has "0 lives", or simply* $x = 0$.

(**) x = *"in", or* x *is "in", or* x *is "alive", or* x= *"in/alive", or* x *has "more than 0 lives", or simply* $x > 0$.

(***) "complete extension" = "Survival (picture)"

2.2 Traditional discrete linear temporal models

Our starting point is a model for the classical propositional calculus with a set of atomic propositions Q and the evolutionary connectives $\{\neg, \wedge, \vee \rightarrow\}$.

A model for this calculus is a function h giving for each $q \in Q$ a value $h(q) \in \{0, 1\}$. "0" is false ($\perp$) and "1" is true ($\top$).

The assignment function h is arbitrary, and there are no restrictions on h. In fact the set of theorems of classical propositional logic rely on this fact. If we impose restrictions on h, coming possibly from some application area, we may get a more restricted set of theorems. See Remark 2.6, where we give restriction on h coming from the area of argumentation networks.

Remark 2.6. *Note that given an argumentation network* $\mathbb{A} = (S, R)$*, which always has some extensions, we can regard each extension* E *of* A *as generating a classical propositional model* h_E *for the set of atoms* $Q_S = S$*. For* x *in* S *we define* $h_E(x) = 1$ *iff* x *is in* E *(i.e., iff* x = *in). So if* x *is out or if* x *is undecided then* $h_E(x) = 0$.

We can use the network A *as a restriction on what assignments we can give to the atoms of* $Q_S = S$.

We can turn classical propositional logic into a temporal system by adding a flow of time $(T, <)$ and making h time dependent (see [24, 25] for an extensive coverage of this area).

Let us take $T = \{1, 2, \ldots\}$ the set of natural numbers and let $<$ be the usual "smaller than" relation on the numbers. Thus the function h becomes time dependent, giving for each $t \in T$ and $q \in Q$ a truth value $h(t, q) \in \{0, 1\}$. We also write $h_t(q)$, to stress that h is dependent on the time $t \in T$.

In general, we can make any system $\mathbb{S}$ dependent on time in a methodological way. Let $\mathbb{S}$ be a system with components $\{\mathbb{C}_i\}$. We add a parameter $t \in T$ to each of the components, denoting the time dependent component by $\mathbb{C}_{i,t}$ and turning the system $\mathbb{S}$ into the time dependent system $\mathbb{S}_t = \{\mathbb{C}_{i,t}\}$.

Example 2.7. *Let us add a time parameter to an argumentation system of the form $\mathbb{A} = (S, R)$, where S is the set of arguments and $R \subseteq S \times S$ is the attack relation. We take a flow of time to be, say $(\{1, 2, 3, \ldots\},)$, and let (S_t, R_t) be time dependent networks and let $\mathbb{A}_t = (S_t, R_t)$.*

What else does temporal logic do to the time dependent system $\mathbb{S}_t$, thus defined? Let us illustrate for the case of the classical propositional calculus.

Definition 2.8. *A traditional temporal logic starts with a given flow of time of the form $(T, <)$, where T is the set of moments of time and $<$ is the transitive, irreflexive, earlier-later binary relation on T. In addition to the classical connectives, Temporal Logic adds temporal connectives to the classical language, for example the connectives $\{\mathbf{F}, \mathbf{G}, \mathbf{P}, \mathbf{H}, \mathbf{J}, \mathbf{Y}, \mathbf{T}\}$ with the following truth conditions, where $t \vDash \varphi$ means that the temporal formula φ (written using Q) and $\{\wedge, \vee, \neg, \rightarrow, \mathbf{F}, \mathbf{G}, \mathbf{P}, \mathbf{H}\}$ holds at $t \in T$ under h which is an assignment $h(t, q)$ dependent on both time t, and atomic q. Note that h is arbitrary function without restrictions.*

- $t \vDash_h q$, *if* $h(t, q) = 1$ *for* $q \in Q$
- $t \vDash_h \varphi \wedge \psi$ *iff* $t \vDash_h \varphi$ *and* $t \vDash_h \psi$.
- $t \vDash_h \varphi \vee \psi$ *iff* $t \vDash_h \varphi$ *or* $t \vDash_h \psi$.
- $t \vDash_h \neg\varphi$ *iff* $t \nvDash_h \varphi$.
- $t \vDash_h \varphi \rightarrow \psi$ *iff* $t \nvDash_h \varphi$ *or* $t \vDash_h \psi$.
- $t \vDash_h \mathbf{F}\varphi$ *iff for some* $s, t < s$ *we have that* $s \vDash_h \varphi$.
- $t \vDash_h \mathbf{P}\varphi$ *iff for some* $s < t$ *we have that* $s \vDash_h \varphi$.
- $t \vDash_h \mathbf{G}\varphi$ *iff for all* $s, t < s$ *implies* $s \vDash_h \varphi$.
- $t \vDash_h \mathbf{H}\varphi$ *iff for all* $s < t$ *we have that* $s \vDash_h \varphi$.
- $t \vDash_h \mathbf{J}\varphi$ *iff we have that* $s \vDash_h \varphi$, *where* s *is the first element of the time flow if a first element exists and otherwise* $s = t$.

- *$t \vDash_h \mathbf{Y}\varphi$ iff we have that $s \vDash_h \varphi$, where s is the immediately preceding element of t in the time flow (i.e. the Yesterday element) if such an element exists and otherwise $s = t$.*
- *$t \vDash_h \mathbf{T}\varphi$ iff we have that $s \vDash_h \varphi$, where s is the immediately following element of t in the time flow (i.e. the Tomorrow element) if such an element exists and otherwise $s = t$.*

Remark 2.9. *Traditional (as opposed to evolutionary) temporal logic is concerned with mathematical and logical properties of temporal models and languages for a variety of flows of time. In other words, the temporal connectives want to talk about variations in time of various components of the system. So for example in the case of a time dependent argumentation network of the form (S_t, R_t) temporal logic will talk about time variations in S and R, but it is not meant to, and possibly not able to, talk about extensions and how they vary in time.*

As we shall see later, for temporally dependent such networks, this is a problem because we really do want to talk about extensions and how new and old arguments in time can affect extensions. To be able to do that we need to define what we call "Evolutionary Temporal Logic for Argumentation".

In general talking about variations in time of system components $\mathbb{C}_t$ is quite valuable.

Indeed, evolutionary temporal logics have wide applications in philosophy, general logic, theoretical computer science, artificial intelligence and the formal analysis of language.

However, as we said, traditional temporal logic is not suitable for argumentation (despite papers [26, 27] which followed traditional methodology), for the following two reasons, which are certain features of traditional temporal logic:

($\sharp$1): The models $\mathbf{h}_t$, involved in temporal logic, given for each time t, come from some application area and are fixed. We are not given any details of how they are computed. So formally, our choice of the assignment h_t is arbitrary and given by us in the meta-level.

($\sharp$2): The future temporal connectives, such as $\mathbf{F}\varphi$ are reduced to the temporal behaviour of φ in the model. They are not considered as atomic, with independent values.[1]

[1] $\mathbf{F}\varphi$ is true now if φ will be true in the future. So if we know the values of φ we know the values of $\mathbf{F}\varphi$. Compare this with the connective $\mathbf{B}\varphi$ = "I believe φ". I can believe or not believe φ independent of whether φ is true or not.

($\sharp$3): There are no global restrictions on the assignments to atomic q's beyond what is forced by axioms on the connectives. For example the axiom

$$[q \wedge Gq \wedge Hq]$$

forces q to be true at all moments of time, but the axiom

$\{t|h(t,q)=1\}$ is finite

is not expressible using the connectives and thus cannot be enforced. Similarly see the restrictions mentioned in Remark 2.6.

2.3 The many-lives networks; A quick formal reminder

We give quick definitions of how to define extensions for many-lives argumentation. The exact details are not important for the investigation of the temporal aspects. It is given here just for the record. See [2]. We assume that the networks (S,R) we deal with are acyclic for the purpose of certain inductive definitions.

Definition 2.10 (Labelling annotation for a network)**.**

1. *Let (S,R,M) be an annotated network as follows: (S,R) is a finite acyclic argumentation network.*

   M is a function on S giving for each $x \in S$ a natural number in $\{1,2,3,...\}$ being the number of lives of x (in argumentation terms, to ensure that x will be labelled out, we need at least $M(x)$ nodes e such that eRx and $M(e) > 0$, i.e. e is labelled in, see next item).

2. *Let Attack(x), for $x \in S$ be the set of all y in S such that yRx holds.*

3. *Let M^* be defined for $x \in S$ using structural induction on the finite acyclic network as the function derived from M, satisfying the implicit equation (*1) and (*2) as follows:*

   *(*1) $M^*(x) = M(x)$, if there is no y in S attacking x*

   *(*2) $M^*(x) = \max\{0, (M(x)$ — the number of elements y in Attack(x) such that $M^*(y) > 0)\}$.*

4. *Using M^* we can give Caminada like in, out labelling of the nodes of (S,R,M), following our calculation in item 3 above:*

   x is out if $M^*(x) = 0$
   x is in with remaining lives $M^*(x)$ if $M^*(x) > 0$.

The number of lives of each node in this figure (S, R, M) is one

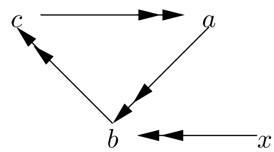


Figure 10

Example 2.11. *For practical examples of many-lives consider the following:*

1. *How many complaints of students against a lecturer can we tolerate before we open a case (hearing) against the teacher? (Probably maybe 5–8, certainly not just one.)*
2. *Driving licence example, see Example 4.1.*

Example 2.12. *We illustrate the computation of* M^* *as in Definition 2.10 and make an important point about this definition. Consider Figure 10, in this figure* $M(z)$ *for each node is 1.*

We now calculate M^**:*

- $M^*(x) = M(x) = 1$
- $M^*(b) = 0$*, because it is attacked by* x*. We do not need to care about* $M^*(a)$*, (which also attacks b because no matter what* $M^*(a)$ *is,* $M^*(b)$ *must be 0). See Remark 2.15.*
- $M^*(c) = 1$*. This is so not because c is not attacked but because it is given that* $M(c) = 1$*. Had we given* $M(c) = 0$ *we would have had* $M^*(c) = 0$ *and not* $M^*(1)$ *even though* $M^*(b) = 0$*. See clause (*2) in Definition 2.10.*
- *Since* $M^*(c) = 1$*, we get* $M^*(a) = M(a) - M^*(c) = 1 - 1 = 0$*.*

Example 2.13. *We continue Example 2.12 by modifying Figure 10 into Figure 11:*

First consider the graph of this figure as a Dung argumentation network. From that point of view we have $y =$ *in and* $x =$ *out and therefore the loop* $\{a, b, c\}$ *stands alone as a three loop with the only extension for the loop is "all undecided". The fact that there is* $x =$ *out attacking b does not help or make any difference.*

Let us now view the element of Figure 11 as having many-lives M*, each having one life, i.e. we have* $M(a) = M(b) = M(c) = M(x) = (y) = 1$*.*

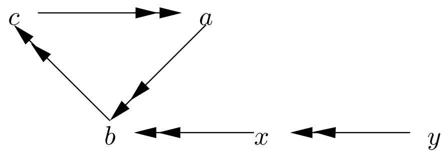


Figure 11

Let us now calculate M^* from M for this network using Definition 2.10. We get

$$\begin{aligned}&M(y) = 1,\\ &M^*(x) = 0,\\ &M^*(b), M(b) - M^*(a) - M^*(x),\\ &M^*(a) = M(a) - M^*(b),\\ &M^*(c) = M(c) - M^*(b).\end{aligned}$$

Substituting known values we get the equations:

$$\begin{aligned}M^*(b) &= 1 - M^*(a)\\ M^*(a) &= 1 - M^*(b)\\ M^*(c) &= 1 - M^*(b)\end{aligned}$$

which yields

$$\begin{aligned}M^*(c) &= M^*(a)\\ M^*(b) &= 1 - M^*(a)\end{aligned}$$

If we allow $M^*(a) = M^*(c) = 1$ we get an anomaly since $c \twoheadrightarrow a$. So we must have (M^* is a $\{0, 1\}$ function giving stable extension)

$$M^*(a) = M^*(c) = 0 \text{ and } M^*(b) = 1$$

Remark 2.14. *Example 2.13 raises several questions which require our answers:*

1. *We introduce the idea of solving Dung loops by using many-lives stable semantics. Namely:*
   (a) *Give all points in the loop single life, i.e., let $M(x) = 1$ for all x in the loop.*
   (b) *Choose a point in the loop. Fix this point (call it b).*
   (c) *Calculate M^* from M and get equations.*
   (d) *The equations in (c) might allow for several solutions. Do not allow for any solution which gives $M^*(x) = M^*(y) = 1$ when $x \twoheadrightarrow y$.*

We need to check under what conditions of the loop (geometry of the graph) we always get solutions.

Note that the calculation of M^* is not Dung like. We may have all attackers y of x all have $M^*(y) = 0$ but yet also $M^*(x) = 0$.

*This is because of item *2 of Definition 2.10.*

Remark 2.15. *In Definition 2.10 we mentioned that the function $M*$ is defined for each $x \in S$ using structural induction on the finite acyclic (S, R) network as the function derived from M. Let us explain how this is done.*

Define the Rank of $x \in S$ as follows:

- *x is of rank 1 if Attack(x) is empty.*
- *x is of rank 2 if all members of Attack(x) are of rank 1*
- *x is of rank $n + 1$ if all members of Attack(x) are of rank $< n + 1$ and at least one member of Attack(x) is of rank n.*

The structural induction is on the Rank of points x

Remark 2.16 (Case of Loops)**.** *Our starting point is the definition of M^* from M in Definition 2.10, and Remark 2.15. The assumption there is that the network (S, R, M) is acyclic, and we use structural induction on the notion of Rank to define M^*.*

If we have loops we need to define the structural induction differently to be able to define M^* from M.

We proceed as follows:

1. *By a backward chain from point y to point x we mean a sequence of points $z_1, z_2, ..., z_n$ such that for each $i(i = 1, 2, ..., n - 1)$ we have that yRz_1, z_iRz_{i+1} and z_nRx.*
2. *The length of the chain in (1) is n.*
3. *If xRx we say the length of the chain in this case is 0.*
4. *Let $S1$ be a subset of S. We say $S1$ is a loop if for every x, y in $S1$ there exists a backward chain from y to x built up all of points of $S1$.*

   $S1$ is a maximal loop if it is not properly contained in a bigger loop.
5. *We say that a maximal loop $S1$ is a top loop if for every y and x in $S1$ such that there is a backward chain from y to x we have that y is also in $S1$.*

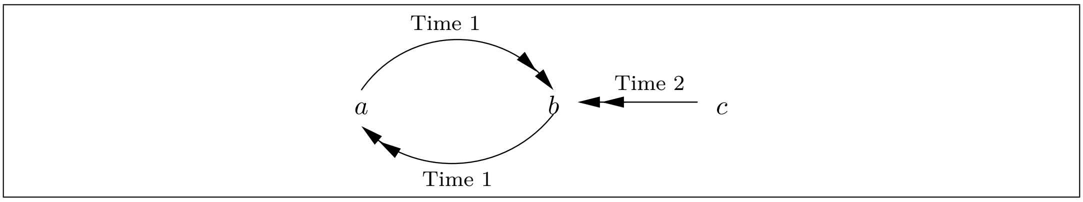


Figure 12

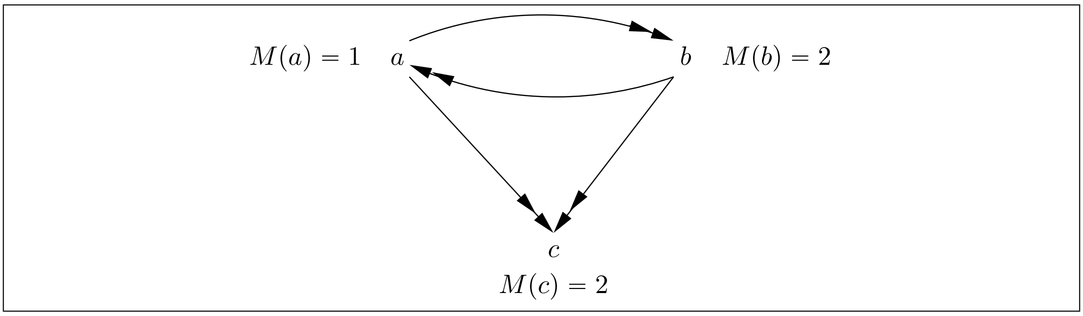


Figure 13

6. *Let $S1$ and $S2$ be two maximal loops Define a relation $\mathbb{R}$ on the set $\mathbb{S}$ of maximal loops by:*

   $S1\mathbb{R}S2$ iff for some y in $S1$ and x in $S2$ there exists a backward chain from y to x.

   Then $(\mathbb{S}, \mathbb{R})$ is finite acyclic.

7. *For a maximal loop $\mathbb{C}$ in $\mathbb{S}$ let $\mathbb{M}(\mathbb{C})$ be defined in some reasonable way as the number of lives given to the loop as a unit, taking into account the lives of the members of the loop. For example define it as $\mathbb{M}(\mathbb{C}) = \min\{M(y)|y \text{ in } \mathbb{C}\}$.*

8. *Let $\mathbb{M}^*$ be calculated out of $\mathbb{M}$ as in Definition 2.10 for the system $(\mathbb{S}, \mathbb{R}, \mathbb{M})$.*

9. *Let M^* for the system (S, R, M) be finally defined for each y in a reasonable way from the values $M(y)$ and $\mathbb{M}^*$ (the max loop containing y), for example as $M^*(y) = \max(0, M(y) - \mathbb{M}^*(\mathbb{C})$, where $\mathbb{C}$ is the the unique loop containing y.*

Example 2.17. *To illustrate the ideas of Remark 2.16, consider Figure 13:*

In this figure the top loop is $\{a, b\}$. The minimum life of the top loop is 1 and therefore the M^* for loop members if $M^*(a) = 0, M^*(b) = 1$ and propagating to c we get $M^*(c) = 2$.

This calculation is consistent with activating the simultaneous attack of all elements of the loop $\{a, b\}$ *on one another to get*

$$M_1^*(a) = 0, M_2^*(b) = 1$$

and continuing attacking c, we get $M_1^*(c) = 2$.

It is possible to give other algorithms, for example, allow members of the top loop to attack all possible targets, not just only other members of the loop. In this case both a and b will attack c and we will end up with M_2^*, *where*

$$M_2^*(a) = 0, M_2^*(b) = 1, M_2^*(c) = 1.$$

3 Evolutionary temporal argumentation

In Subsection 3.1 we introduce evolutionary temporal logic and give some example from argumentation. In the next subsection we give many more examples.

3.1 Evolutionary propositional temporal logic for argumentation

Let us start with the classical propositional calculus with atoms Q and the classical connectives $\{\neg, \wedge, \vee, \rightarrow\}$. We have already said that we can turn any assignment to the atoms into a time dependent function by taking a flow of time $(T, <)$ and for each $t \in T$ look at a function $h_t(q) = h(t, q) \in \{0, 1\}$ for each $\in T, q \in Q$.

Example 3.1. *Now consider the flow of time* $T = (1, 2, 3, \ldots)$ *and the usual* $<$ *and assume for each* t *in* T *that we have a set* $\mathbb{H}_t$ *of assignments* $\mathbb{H}_t = \{h_{t,i}, i = 1, 2, 3\}$ *to choose from. So we can get a sequence* $h_1, h_2, h_3, \ldots$ *with* h_n *in* $\mathbb{H}_n$. *We can impose conditions on the choice of sequences. Examples of such conditions can be in a meta-language talking about the sequences, (not a temporal language but any other language). For example*

- *no change,* $h_{n+1} = h_n$
- *all* $h \in \mathbb{H}_t$ *must be obtained from some algorithms (e.g. be complete extensions of a varying argumentation network)*
- *For each* t, h_t *is generated from an argumentation network* $\mathbb{A}_t$ *as in Remark 2.6. Thus* $\mathbb{H}_t$ *is the set* h_E, *of all extensions* E *of* A_t.
- $\mathbb{H}$ *can be generated probabilistically*
- *and so on.*

Let us illustrate by defining on meta-level condition as an example, and so we choose the condition of continuity.

We say that the sequence $h_1, h_2, h_3, \ldots$ *preserves continuity if for each* n, h_{n+1} *is a* minimal change *from* h_n. *We have to define what we mean by minimal change, i.e.* h_{n+1} *is chosen from* $\mathbb{H}_{n+1}$ *representing a minimal change to* h_n.

(*) *Given a set* $\mathbb{H}^*$ *of assignments* $h \in \mathbb{H}^*$ *and given* $h_1 \in \mathbb{H}$, *then* $h_2 \in \mathbb{H}^*$ *is a minimal change from* h_1 *according to a policy of change* $\mathbf{P}$ *of Hamming Distance defined in Definition 3.2.*

Note that we do not require that h_2 *be unique, only that it be minimal.*

Definition 3.2.

1. *Let* $(T, <) = (\{1, 2, 3, \ldots\}, <)$
2. *Let* $Q = \{q_1, \ldots, q_n\}$ *be a finite set of atoms.*
3. *Let* $\mathbb{H}_t$, *for each* t *in* T *be a set of assignments* h

$$h : Q \mapsto \{0, 1\}$$

4. *We now define the Hamming distance policy* $\mathbf{P}$ *as follows*
   (a) *We can regard each* $h \in \mathbb{H}_t$ *as a vector* $V_h = (h(q_1), \ldots, h(q_n))$ *and for any two* h_1, h_2 *thus define* $d(h_1, h_2) =$ *the number of coordinates* i *for which* $V_{h_1}(i)$ *is different from* $V_{h_2}(i)$.
   (b) *Let* $h_i \in \mathbb{H}_i$ *be a sequence of assignment* $i = 1, 2, 3, \ldots$. *We say that this sequence preserve continuity according to policy* $\mathbf{P}$, *iff for each* i, *and each* $h \in \mathbb{H}_{i+1}$ *we have* $d(h_i, h_{i+1}) \leqslant d(h_i, h)$.

   An evolutionary temporal model for $(T, <)$, *based on a sequence of sets of assignment* $\mathbb{H}_t, t \in T$, *is any sequence of assignments from* $\mathbb{H}_t$ *preserving continuity* $\mathbf{P}$.
5. *An evolutionary temporal model for argumentation is any model bases on sets* $\mathbb{H}_t$ *obtained from respective argumentation networks* $\mathbb{A}_t$, *as defined in Example 3.1.*

The above definition is just for illustration, it is not suitable for the notion of continuity in the case of many-lives argumentation. The next section examines what happens in argumentation and what is needed.

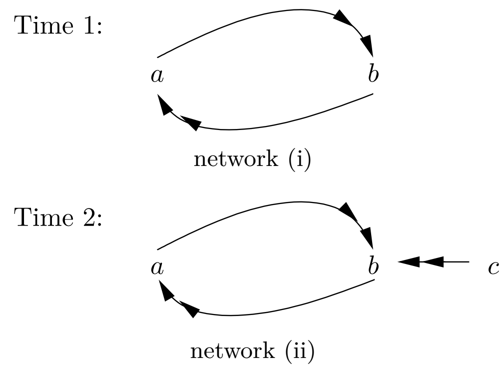


Figure 14

Note that the notion of continuity is external (meta-level) to the temporal logic semantics. Some continuity policies **P** may be expressible as axioms on the temporal connectives (e.g. no change can be written as $G(A \to GA)$) but some may not. Such lines of research belong to pure traditional temporal logic and do not concern us here.

We now give examples to illustrate evolutionary temporal logic for argumentation. Compare with (♯1) and (♯2) of Subsection 1.2.

Example 3.3. *We illustrate evolutionary temporal logic for argumentation by two example networks, that of Figure 14 and that of Figure 15.*

1. *Analysis of Figure 14:*

   The two networks in this figure (network (i) and network (ii)) show the evolution of a network from network (i):

$$\begin{aligned} S_1 &= \{a, b\}, \\ R_1 &= \{(a, b), (b, a)\} \end{aligned}$$

   into network (ii)

$$\begin{aligned} S_2 &= \{a, b, c\} \\ R_2 &= \{(a, b), (b, a)(c, b)\} \end{aligned}$$

   Evolutionary temporal logic can only talk about the change. It can only say that c showed up at Time 2 and that c attacks b.

   This is not what we are interested in argumentation. We want to look at extensions. So what we want to say is either option 1 or option 2 or option 3.

Option 1. *At Time 1 there were three possible extensions. We chose at Time 1 the extension*

$$E_2^1 = \{a = \ in,\ b = \ out\}.$$

At Time 2 we got extra information of a new c attacking b and as a result we modified the chosen extension into

$$E_2^1 = \{a = \ in, b = \ out, c = \ in\}$$

Option 2. *At Time 1 there were three possible extensions. We chose extension*

$$E_1^2 = \{a = \ out, b = \ in\}$$

At Time 2 we got an extra c attacking b and so the only extension possible at Time 2 was

$$E_2^2 = \{a = \ out, b = \ in, c = \ in\}$$

Option 3. *At Time 1 there were three possible extensions. We chose*

$$E_1^3 = \{a = \ und,\ b = \ und\}$$

At Time 2 we got an extra c attacking b and so the only possible extension was

$$E_2^3 = \{a = \ in, b = \ out, c = \ in\}$$

We note that none of these options can be expressed in traditional temporal logic, because traditional temporal logic can only give the assignment generated by the chosen extension at time 1 and time 2, and, not how the extension was calculated, nor how the associated networks changed. So we have:

Problem 1. *What kind of temporal logic do we need? How do we extend temporal logic to suit our need?*
Answer. *We need what we describe in Definition 3.2, which we call* evolutionary temporal logic.

2. *Analysis of Figure 15:*

   We make one more point. Consider Figure 15:

   In this figure there are two independent parts, and there is no change in Time 2 on the $\{a, b\}$ part of the network.

   Therefore we should expect to say that the extension chosen in Time 1 remained unchanged in Time 2 as far as $\{a, b\}$ is concerned because the network did not change on $\{a, b\}$. What we do not want to say is Option 4.

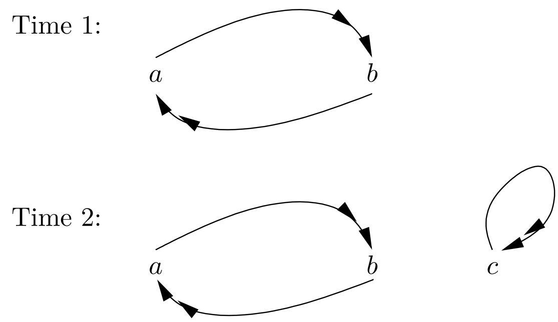


Figure 15

Option 4 (we do not want this option). *At Time 1 the extension chosen was*

$$E^1 = \{a = \ in, b = \ out\}.$$

At Time 2 we changed our mid on the $\{a, b\}$ *part and although the network did not change this part, we chose extension*

$$E^2 = \{a = \ out, b = \ in, c = und\}.$$

This presents us with a serious problem 2.

Problem 2. *Having chosen an extension* E^1 *at time 1, how do we continue modifying the same extension in future times without changing our minds like we did change in Option 4? In other words, how do we force/express continuity of our choice of extension, yielding only to necessary unavoidable change?*
Answer. *We can use the concept of continuity as item (c) of Definition 3.2. See also [30].*

Example 3.4. *Let* $a =$ *we cannot appoint Professor* X*.*
$b =$ *In the future Professor* X *can get big projects.*

We have that at Time 1 $b \twoheadrightarrow a$*. This holds independently of the question of whether b is true or not. The reason being that we do not know the future (Time 2, 3,4,...), but we need to make a decision at Time 1 (take the extension* $E^1 = \{b = \ in, a = \ out\}$*.*

Example 3.5. *We conclude with one more example showing that we may want the opposite of continuity. Consider networks (i) and (ii) of Figure 14 and assume that network (ii) comes temporally before network (i). Think that the cycle* $\{a \twoheadrightarrow b, b \twoheadrightarrow a\}$

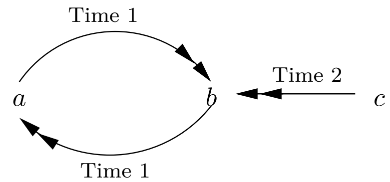


Figure 16: Combining the two parts of Figure 15, (Time 1 and Time 2) into this single figure by time stamping the arrows in it. The arrows $a \twoheadrightarrow b$ and $b \twoheadrightarrow a$ are timestamped "Time 1" and the arrow $c \twoheadrightarrow b$ is time stamped "Time 2". This is a much better notation because we have a single growing figure. See Remark 3.6.

are two arguments attacking each other and that at Time 1 we have a witness c attacking b, so the only extension of the cycle is $\{a = \ in, b = \ out\}$. At Time 2 we have network (i), i.e. c withdraws, and we have the option of $\{a = \ out, b\ in\}$. We do not want continuity in this case. Our meta-level condition is to want $b = in$. In this case we take the option $\{a = \ out, b = \ in\}$.

We cannot express this condition in temporal logic, because the condition is Fb, which cannot always be true. But we can insist on an algorithm for computing the extension at any time t (this is meta-level for obtaining an extension h_t from (S_t, R_t)) which attempts to start with $b = in$ and checks if one can find an extension containing $b = in$.

Remark 3.6. *The perceptive reader might notice that the temporal progression described and discussed in Figure 14 can be represented in a single figure, where the arguments and attacks are time-stamped. See Figure 16, also compare with the general Figure 17.*

This perception is more than an alternative representation. It implies a criticism of what we are proposing here.

Criticism. *Why propose evolutionary temporal logic for argumentation, showing a sequence of temporal nodes t and argumentation networks (S_t, R_t) attached to t, why not put them all in one big argumentation network (S, R) with time stamping as in Figure 17. In this figure each node of the form z in the figure and each attack of the form (z_1, z_2) in the figure has the further annotation of a Time Stamp $T(z)$ and $T(z_1, z_2)$ respectively indicating the temporal moments in which the item exists. The annotation is a set of moments. In case there is persistence, that is an item which exists at a moment t continue to exist (and does not disappear after time t) we can use the annotation "$t+$".*

So if $x, y \in S_t$ and $(x, y) \in R_t$ we put $(t, x), (t, y) \in S$ and $(t, x, y) \in R$.

We can retrieve (S_t, R_t) from (S, R).

Since the emphasis of "evolutionary temporal logic for argumentation" is on the argumentation part, it makes more sense to use (S, R).

Remark 3.7. *Note that formally, from the point of view of formal argumentation,* (S, R) *of Example 3.5, looks like just another annotated argumentation network. It is in the meta- level that we interpret this annotation as leading to an evolutionary argumentation network and use it in our intended application. If we have a different application in mind, (see [29]) we might interpret the annotation differently and get different results. (See Example 4.3 in our discussion in the section Comparison with the Literature, in which [29] is discussed.)*

Answer to criticism.

1. *The notion of evolutionary temporal logic for arbitrary temporal sequences of systems is more general. We can have it for modal logic, for changing preferences, etc.*
2. *However, for some properties we are interested in argumentation, the big* (S, R) *with time stamping are more transparent in the evolutionary time stamping temporal logic approach. For example if some element* x *keeps attacking over time every element* y *other than himself, then the attack behaviour over time of* x *becomes an argument which can attack* x*. We can add that as a "temporal attack behaviour" which becomes an "argument".*
3. *Why not use both methods, depending on convenience?*

3.2 Further examples

This section examines examples from the application area of complaints about sex offenders. This area actually inspired the idea of many-lives argumentation networks. The temporal aspects come from the fact that the victims of a sex offender might complain at different times and so we need to time stamp the appearance of victims and their attacks. So the correct annotation is as in Figure 17.

The notation $T(y_1, x)$ is the time that y_i complained about x. It annotates the double arrow from y_1 to x. We need to explain in our notation the interaction between the many-lives of an argument x and the question of whether x is in or out or undecided.

- If the many-lives of x is positive then x is in (also we can say that x is alive).
- If the many-lives of x is 0, then we can say that x is out, or dead.

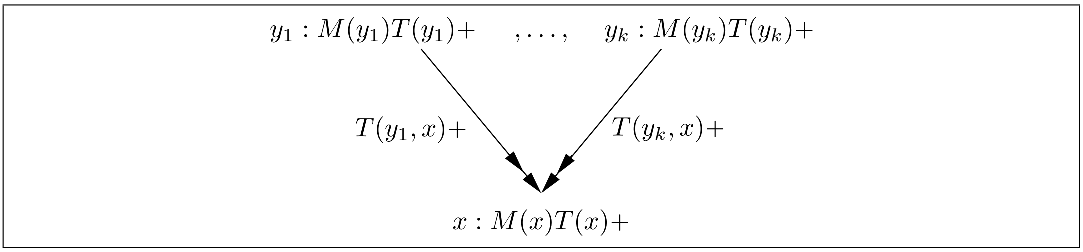


Figure 17: In this figure, x has $M(x)$ lives and exists at $T(x)$ and, afterwards, y_i, which exists at $T(y_i)$ and, afterwards, attacks x at time $T(y_i, x)$ and, continue to exist and attack afterwards, and has lives $M(y_i)$, for each $i = 1, ..., k$ and x exists from time $T(x)$ onwards. We have to assume that attacks (x, y) exist only at times where both x and y exist. However mathematically this is just a reasonable but not a necessary condition.

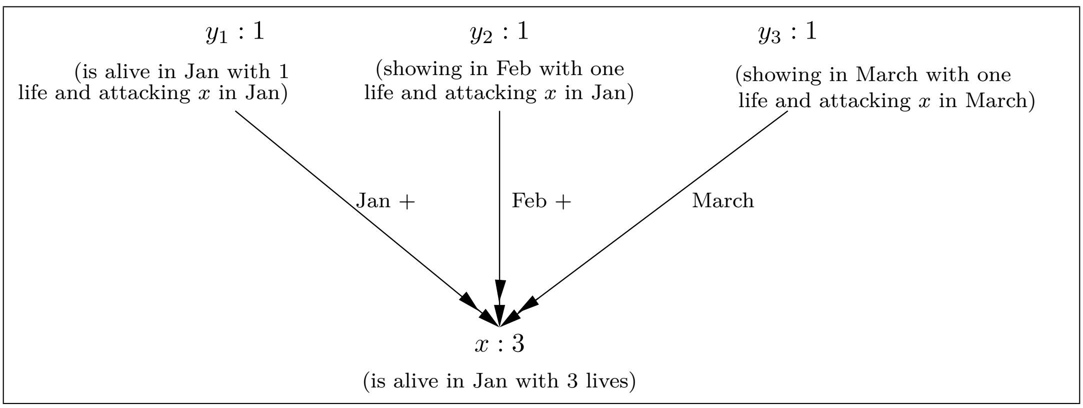


Figure 18

- If the number of lives of x is not known, or cannot be calculated because of loops, we can say that x is undecided or unknown.

The question we ask is if we look at a time, say when there were only 2 complaints, we ask how many-lives does x have at that time? The answer is that x has $M(x)-2$ lives, because of the fact that if $M(x)-2$ more attackers come forward and complain then x will be "dead".

Figure 14 is a concrete example of this annotation (see Remark 3.6 for this temporal annotation):

x has only 2 lives remaining in January. In February he has 1 life left and when in March we have the third y_3 complain then x has 0 lives, i.e. x is dead.

We note that Figure 18 is a simplification of the temporal sequence. In January

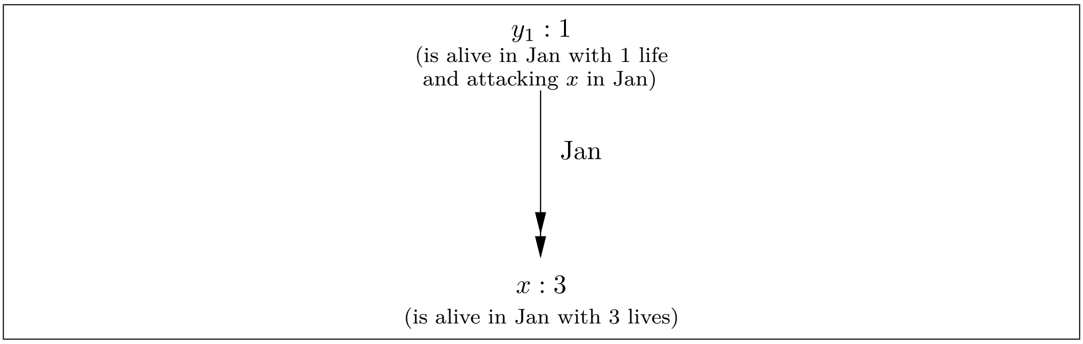


Figure 19

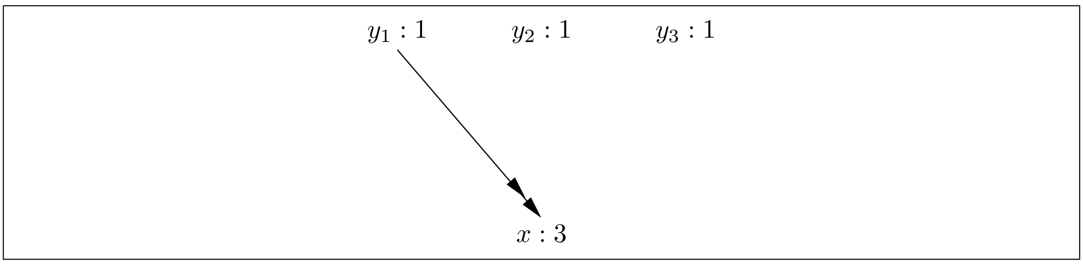


Figure 20

we had only the victim y_1 coming forward. We did not yet know about the victim y_2 who came forward in February, nor did we know of victim y_3, who came forward in March. So the January network should be the network of Figure 19 and not Figure 20.

We can use the convention that we always look at past figures from the point of view of the latest attack, in this case from the point of view of March, thus time-stamping all attacks.

So elements "show up" at the first time in which they attack others or are being attacked by others. So in Figure 18 we read that x and y_1 "showed up" in January, y_2 in February and y_3 in March.

To be consistent in using this notation/convention, we need the assumption of persistence, namely once there is an attack or an attacker or a target it does not disappear. Without this assumption we have to attach a set label to each element in $S \cup R$, stating all the time moments in which it exists. See Definition 1.14.

Obviously mathematically this is consistent but we need to consider applications where attacks can exist without an attacker and/or without a target. We further discuss this below following principle **PP4**.

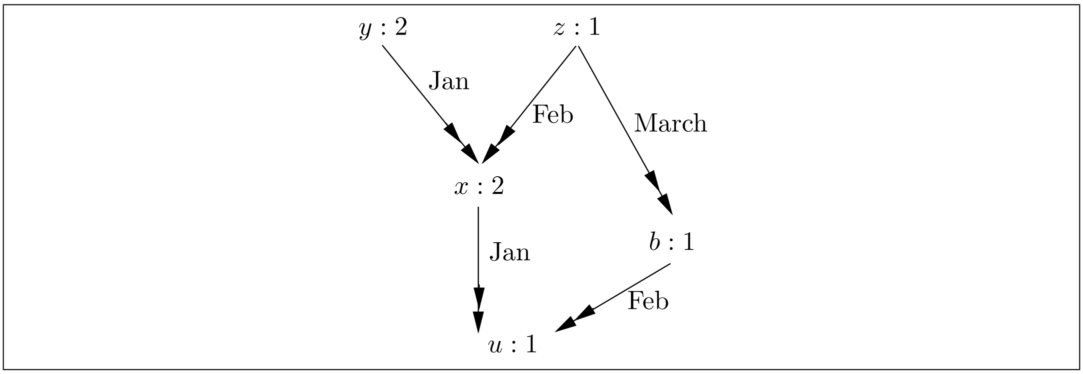


Figure 21

We now state our third principle:[2]

PP3: An attack from y to x is time stamped with a time $t = T(y, x)$ unique to y and x. Any algorithms governing the number of lives of any node in the system will take account of these time stamps.

Figure 21 describes the more general type of networks, which uses the time stamping mentioned in **PP**3.

In Figure 21, z attacks two targets at two different times. We need to calculate the situation (that is, the semantical extension, showing which element is in, or out or undecided and with how many-lives, also viewed as the survival situation, who is alive and who is dead) each month. In January the graph is very simple as illustrated in Figure 22. Note that the nodes z and b do not show up in the figure because in January they have not come forward yet.

The life of y is 2 because it is not attacked by anyone. Since y is alive it can attack x, reducing the life of x by 1. So x is still alive and the life of x is 1 even though it is attacked by y. x is alive so it can attack u and the life of u is 0. u is dead.

[2]This paper continues our previous paper [2] entitled “Introducing Abstract Argumentation with many-lives" The previous principles **PP1** and **PP2** appear there, they are:

PP1: Every element x has a number $M(x)$ of lives (including possibly the value 0 in which case the element is out, or dead). To really kill x (reduce its many-lives to 0) you need to kill it $M(x)$ times (attack it by $M(x)$ different lives/in elements). In particular non-attacked elements retain all their many-lives intact and have the capability of attacking other elements (reducing the target’s number of lives) if their value is not 0.

PP2: Although an element y may have $M(y)$ lives, when attacking any x it can kill/reduce only one of x lives.

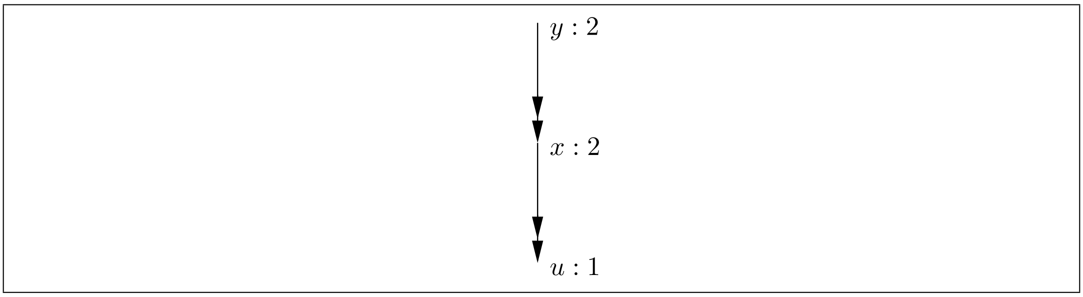


Figure 22

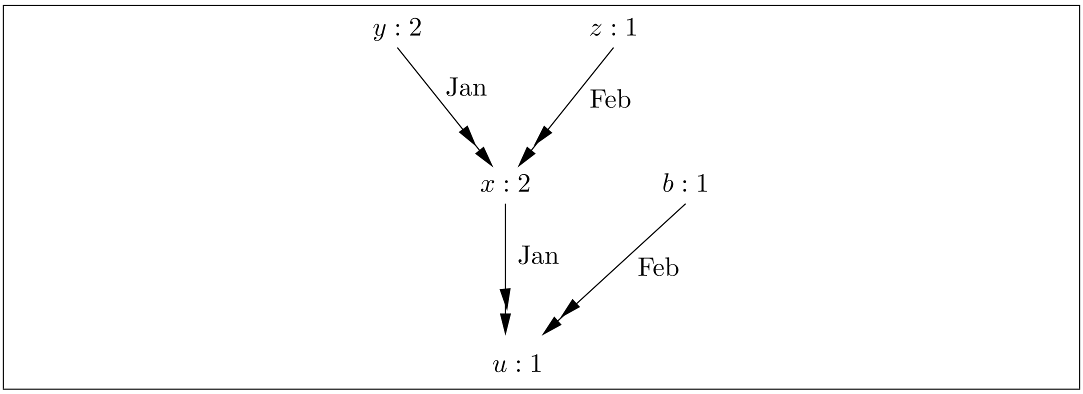


Figure 23

So the survival in January is

$$y : 2, x : 1, u : 0$$

Let us now move to February. The graph now is Figure 23 (recall our notation of Remark 2.5; where we say that alive means in and dead means out).

$y : 2$ and $z : 1$ are alive and attack $x : 2$. Thus $x : 2$ is dead and we write $x : 0$. However, $b : 1$ is alive and can attack u, and so $u : 1$ becomes $u : 0$.

We have the following survivals in February:

$$y : 2, z : 1, b : 1, x : 0, u : 0.$$

Let us now address another point of principle: we look at Figure 24.

The question we ask is what happens in January? Answer: we have Figure 25.

So the survival picture is $a : 1, b : 0$.

In February we have Figure 24. So we have a loop. We ask the question: what do we do with node b? Do we say that b was dead in January, and although we allow b to come back to life in February, b, once dead, can no longer attack?

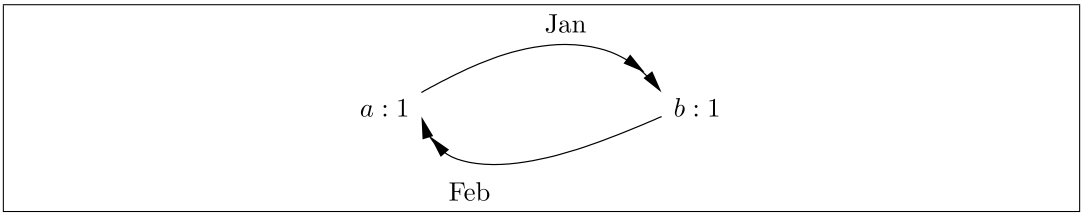


Figure 24

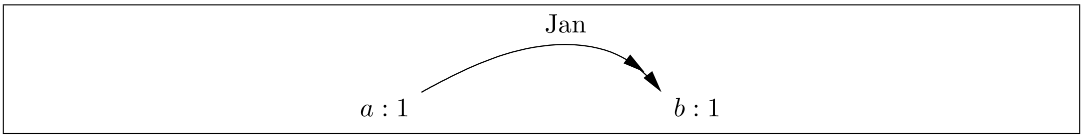


Figure 25

So b cannot attack in February.

We now introduce a new principle,

PP4: In a system with time stamps, an element y may become dead at at time t but may come back to life at a later time s, with $t < s$. In such a case, we accept that y can be alive at time s but we do not allow y to attack any more at time s.

Let us refine better our understanding of principle **PP4**. Let us look again at Figure 23, and imagine that the node $b:1$ is deleted from the figure. If $b:1$ did not exist then $u:1$ would have been alive in February, but it was dead in January. So do we consider u dead or alive? The answer is since its attacker x died in February then u would be alive in February if $b:1$ was not there.

In March we have Figure 18 and the survivors are

$$y:2, z:1, x:0, b:0, u:1.$$

If we apply this principle to Figure 24, the node b is dead in January. In February, the node cannot attack, having died in January and so is killed by node a. Without the principle **PP4**, the node b can counter-attack in February and we get the traditional network of two nodes attacking each other, which has three solutions, $\{a=1, b=0\}$, $\{a=0, b=1\}$, and $\{a=b=0\}$.[3] See the next example 3.8.

Example 3.8. *Let us illustrate our computational options and offer possible refinements to the principle* **PP4**. *Consider Figure 26.*

[3] We use the simplified notation "$x = n$" for the expression "$M(x) = n$" or equivalently the expression "$x : n$" which we use in figures. We will be explicit when needed.

$$\begin{tikzcd} b:1 \arrow[r, two heads, "\text{Feb}"] & a:1 \arrow[r, two heads, "\text{Jan}"] & y:1 \arrow[r, two heads, "\text{March}"] \arrow[d, "\text{Jan}"] & u:1 \\ & & x:1 & \end{tikzcd}$$

Figure 26

Computation January. *At this time* $b:1$ *has not come forward. Therefore* $a:1$ *is alive/in, and since* $a:1$ *attacks* $y:1$ *we have* y *is dead/out (i.e.* $M(y) = 0$*) and hence* $x:1$ *is alive/in and so is* $u:1$*. The survival in January is therefore* $\{a = y = x = u = in\}$*.*

Computation February. $b:1$ *comes forwards and attacks* $a:1$*. So* a *is dead. So* y *becomes alive since it is no longer attacked by* a*. We have two options*

1. *If we apply* **PP4** *we do not allow* y *to attack* x *at February and so* x *is alive.*

2. *Without* **PP4**, y *can attack* x *and* x *is dead. The two outcomes are therefore*

   (a) $b = in, a = out, y = in, x = in, u = in$

   (b) $b = in, a = out, y = out, x = out, u = in$

Computation March. In March we have the full Figure 26, including the March attack of $y:1$ on $u:1$, namely $y:1 \twoheadrightarrow u:1$. According to principle **PP4**, since $y:1$ was dead in January then even though it came back to life in February and in march, it cannot attack any more and so the attack of $y:1$ on $u:1$ is to be discarded and ignored. It is at this point that we might fine-tune principle **PP4** into the more sensitive new principle **PP4***. The attack of $y:1$ on $x:1$ is a January attack and this attack was discarded because in January $y:1$ was dead. Having come back to life in February, does not mean that we revive the January attack of $y:1$ on $x:1$. But the March attack of $y:1$ on $u:1$ is a new attack, newly executed in March when $y:1$ is alive. So we can argue that it should be accepted and not discarded. To give a motivating example, suppose in January $y:1$ complained that $x:1$ sexually abused $y:1$. A witness $a:1$ came forward in January saying he heard clearly $y:1$ boasting that $y:1$ invented false accusations against $x:1$. As a result of that testimony, $y:1$'s complaint was declared false and rejected and $x:1$ was declared innocent and the proceedings against $x:1$ were terminated.

$$\begin{tikzcd} b:1 \arrow[r, "\text{Feb}"] & a:1 \arrow[r, "\text{Jan}"] & y:2 \arrow[r, "\text{March}"] \arrow[d, "\text{Jan}"] & u:1 \\ & & x:2 & \end{tikzcd}$$

Figure 27

In February $b : 1$ attacked $a : 1$ saying that $a : 1$ was nowhere near $y : 1$ and could not have reported any boasting of $y : 1$. Thus the complaint of $y : 1$ against $x : 1$ is now (in February) credible. But the January proceedings against $x : 1$ are over and it stands to procedural reason that we adopt the view that "whatever is gone is gone". In February $y : 1$ credibility is reinstated and so $y : 1$ complaint against $u : 1$ is credible. There is no reason to reject it. We therefore could modify principle **PP4** into **PP4*** as follows:

PP4* In a system with timesteps an element y may become dead at time t. We thus declare dead at time t any attack emanating from y at any time $t' \leqslant t$. If at some later time s the element y comes back to life, then y coming back to life does not bring back to life any attack declared dead at any time $s' \leqslant s$.

According to **PP4***, in Figure 26, the March attack of $y : 1$ on $u : 1$ is alive and the survival picture in March is $b = \text{in}, a = \text{out}, y = \text{in}, x = \text{in}, u = \text{out}$.

Example 3.9. *Consider the situation of Figure 26 but let us give y and x the two lives. This is illustrated in Figure 27. We note that the network in the Figure is finite acyclic, allowing for the calculation which follows. If the figure contains loops or is one big loop itself , a specific algorithm for loops is required. See Remark 2.16.*

Let us calculate what happens in January. We get $a = 1$ attacks $y : 2$ and so y becomes $y : 1$. $y : 1$ is still alive and it attacks $x : 2$ and so we get $x : 1$.

The answer is

$$a = in, y = in, x = in.$$

Let us present the answer in Figure 28

Figure 28 is the same as the January part of Figure 26.

The problem is, if we look at this figure as the January part of Figure 26, we should execute the attacks indicated in the figure and get $a = 1, y = 0, x = 1$.

But if we look at this figure as the result of having already executed the attacks of Figure 27, then we do nothing and execute nothing.

This gives us ambiguity. We have two options:

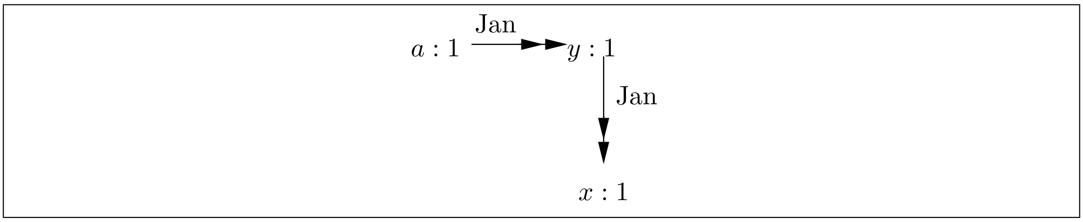


Figure 28

1. *Do nothing and say that Figure 28 comes from Figure 27 after execution.*
2. *Continue the execution until the process is stable.*
3. *We note that the question of which option to use depends on our interpretation of the network. The sex offender interpretation requires option 1. Each complaint/attack reduces one life. We do not use the complaint again as if it were another victim complaining. In comparison, if we have a baby complaining/crying because it wants its nappy changed, then it will complain again and again until the parent cannot take it any longer and does the job (i.e. parent runs out of lives).*

PP5*: Let (S, R, M) be a network as in Definition 2.1 and Remark 2.16 and Remark 2.15. Consider M^* as defined from M in the above definitions and remarks. Continue the derivation of M^{**} from M^* etc., until we reach an $M^{*}...^{*}$ such that another application of the derivation does not give anything new. Call this M function $M(*)$. Principle PP5 says use $M(*)$ and not M^*. Compare with item 3 of Example 3.9.*

So to summarise, let us calculate the survival picture of the network in Figure 27 using principle **PP5**.

Computation January.

1. We start with Figure 29

   We make one pass of calculation as described already and get the network of Figure 28.

2. We make a second pass of calculation on Figure 28 and get Figure 30.

3. If we make another pass of calculation of Figure 30 we get the same figure. So we are stable and the survival picture for January is (using **PP5**) $a = \text{in}, y = \text{out}, x = \text{in}$.

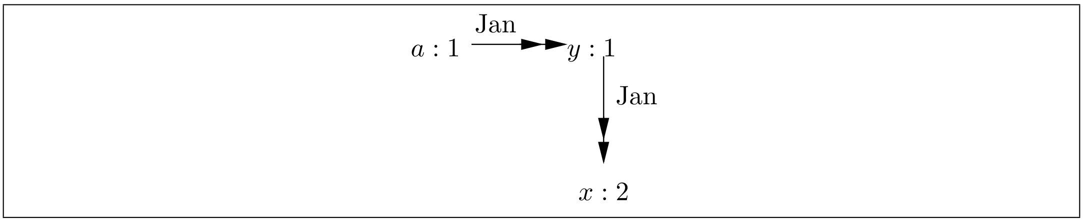


Figure 29

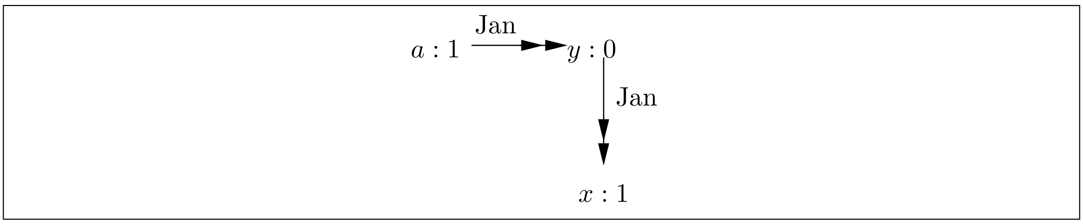


Figure 30

Computation February. The February network is Figure 31.

The attack $y : 2 \twoheadrightarrow^{\text{Jan}} x : 2$ is dead, but we left it in the figure for expositional reasons. y was dead in the January calculation (item (2)) and so it cannot attack. $x : 2$ is not attacked in Figure 31. We get the survival picture $b = \text{in}, a = \text{out}, y = \text{in}, x = \text{in}$.

Computation March. This computation works on Figure 32.

The calculation is straightforward. The survival picture is $b = 1, a = 0, y = 2, x = 2, u = 0$.

Remark 3.10. *The temporal annotation aspects have no counterpart in the traditional Dung semantics, not even in any traditional modal temporal logic version of it. The main reason for this is because we use principle* **PP**4* *on the one hand*

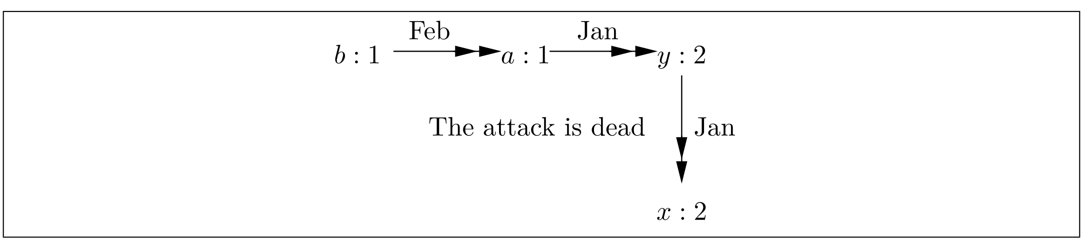


Figure 31

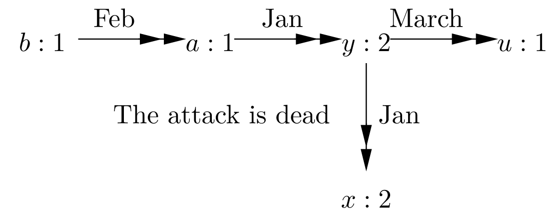


Figure 32

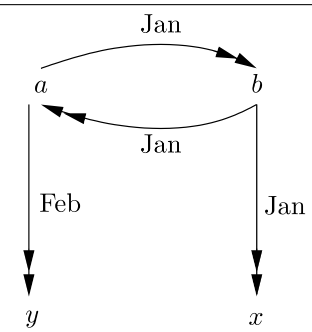


Figure 33

and there may be more than one extension at a given time on the other hand. See Example 3.11.

Example 3.11. *This example makes a methodological distinction which is useful at this point and also explains a possible problem/warning in using principle* **PP**4**.*

Consider the network of Figure 33. Assume all nodes have one life and all attacks are of strength one.

In January we have the network of Figure 34

We have three non-empty extensions in January, i.e. in the network of Figure 34

$$E_1^1 = \{a = in, b = out, x = in, y = in\}$$
$$E_2^1 = \{a = out, b = in, x = out, y = in\}$$
$$E_3^1 = \{a = b = x = \text{ undecided}, y = in\}$$

We note the following

1. *The network of January (Figure 34) is a traditional network.*

2. E_1^1 *chooses* $a =$ *in and* E_2^1 *chooses* $a =$ *out.*

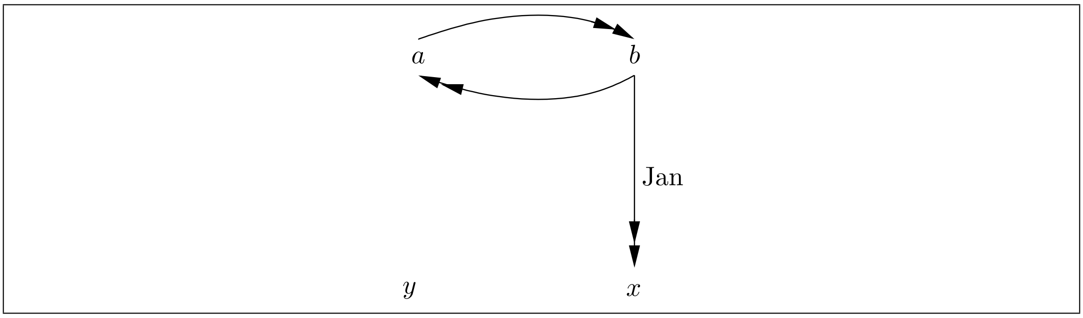


Figure 34

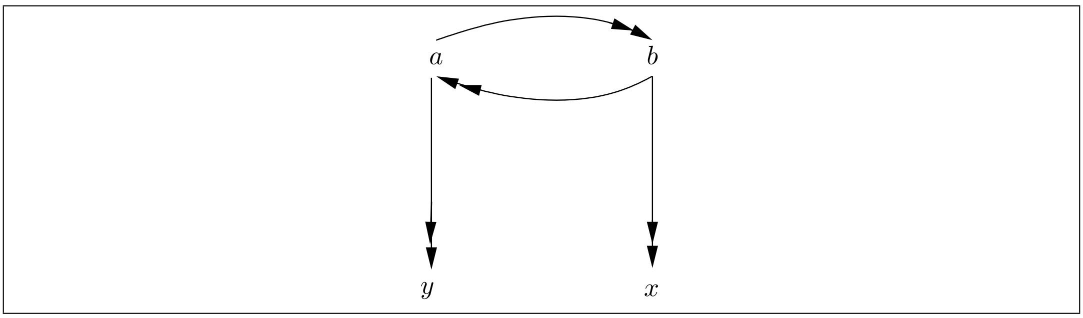


Figure 35

Let us now ask what is the network in February? It is the traditional Dung network of Figure 35.

The extensions are

$$E_1^2 = \{a = out, b = in, x = in, y = out\}$$
$$E_2^2 = \{a = out, b = in, x = out, y = in\}$$
$$E_3^2 = \{a = b = x = y = \text{ undecided}\}.$$

Our methodological point is the following: If in January we choose to resolve the top loop ($a \twoheadrightarrow b$ and $b \twoheadrightarrow a$) by letting $a = in$ can we now in February choose another extension for the top loop and take $a = out, b = in$?

There is a consequence to this change of choice because if we take E_1^1 in January and E_2^2 in February we get that b was dead ($b = out$) in January and b came alive ($b = in$) in February. But then according to principle **PP**4*, *the January attack $a \twoheadrightarrow x$ does not come back to life in February so we must have $x = in$ in February because x is not attacked in February.*

We might argue that this is not acceptable because b came back to life owing to the administrative means (choice of extension) and not because of any substance.

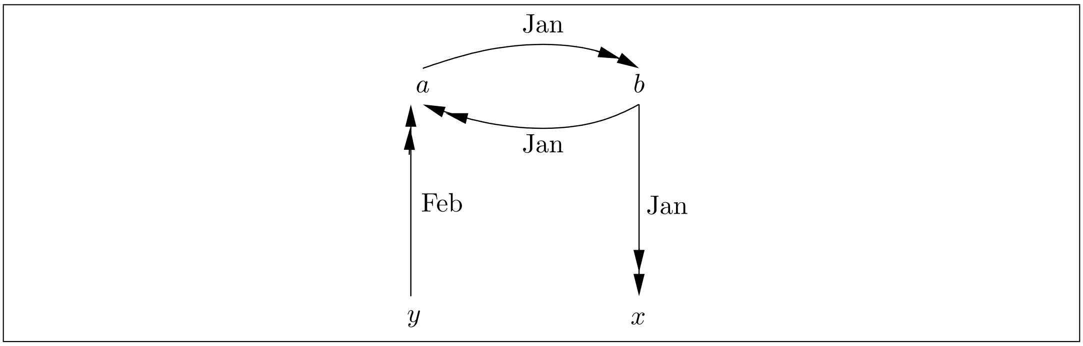


Figure 36

Let us be clear about this point.

The top loop, namely ($a \twoheadrightarrow b$ and $b \twoheadrightarrow a$) is not internally affected between January and February. Therefore we might argue/expect that if in January we chose $\{a = in, b = out\}$ then we chose the same in February and if in January we chose $\{a = out, b = in\}$ then we choose the same in February. By switching choices between January and February, we activate principle **PP**4* *generating possibly unwanted consequences. We therefore need to define/identify mathematically the circumstances under which we are making a change of choice and use this identification to modify principle* **PP**4*. *Let us call the yet to be defined principle* **CPP**4* *(***PP**4* *with continuity).*

The problem is how we formulate such a principle. If we use the declarative set theoretical definitions of extensions as in Subsection 2.1, how do we say that E_2^2 involves a change in choice and is not the correct February extension, which follows E_1^1 In other words, how do we define continuity in a set-theoretical way?

This is also a problem for traditional modal and temporal logics. Such logics do not deal with continuity in time.

To see the difficulty with the set theoretical instrument, consider the network of Figure 36

In this figure we are forced to move from E_1^1 to E_2^2 because y's attack on a forces it! However, it is not easy to tell the difference from the previous case. In fact, in February there is only one extension, so it does not matter what we did in January.

We postpone handling this question to a subsequent offshoot paper.

Remark 3.12. *There may be a way to maintain temporal continuity if the extension are chosen using an algorithm. We examine the algorithms used in previous Example 3.11. First we divide (S, R) into maximal loops (called SCC's, see [3]), then we choose points in the top SCC and propagate the attacks. We get a new (S', R') and* *repeat recursively. Each sequence of choices give an extension.*

If new points are added or deleted, we use the same algorithm but try to retain the same choices as much as possible.

Example 3.13 (Public Pressure)**.** *Case 1: On February 11, 2021, the police announced that a young man named Yarin Sharaf, who was initially suspected of raping a 13-year-old girl at the Corona Hotel, was charged with the lesser offences of consensual, sexual harassment, threats and assault. Following the announcement, there was a wide public outcry and on 25.03.21 the prosecutor's office announced that it was filing an indictment for a more serious offence, which is rape. The change was following public criticism from the victim's family and women's organizations*

Case 2: A criminal known to the police was arrested on suspicion of murdering Yuri Volkov after detectives waited outside the house where he had been staying for hours. According to the suspicion, he stabbed the deceased after the deceased and his wife warned him that he almost hit them on the road. At first the police announced that he was charged with the relatively minor offence of manslaughter, but after public pressure the charge was changed to a charge of murder.

4 Comparison with the literature

We have already compared with the literature when we presented in Section 1, subsection 1.1 the Malkinson real example, and in subsection 1.2 the three views, Timed View, Modal View and Evolutionary View . Also relevant is Remark 3.7. In this section we discuss the differences between the views in more detail.

Example 4.1. *This example is to further illustrate the difference between the Timed View and the Evolutionary View.*

Consider legislation about driving licences. In many countries, traffic offence τ can give bad points on a driving licence D (of the offender). Usually when you accumulate 3 bad points your driving licence is revoked. If you continue to drive after your licence is revoked then the offence τ becomes more serious, say τ'.

The best way to view this is to say that D has 3 lives, and τ attacks D and takes one life from D.

Figure 37 shows an evolutionary sequence:

If we want to take the timed point of view, we have to write the following arguments (doing the evolution in the syntax) and we use $D0, D1, D2, D3$ where "Dx" is "D with x lives". We get a new table 38

In this example the timed view is forced to put the evolution in the syntax!

D, x = Driving licence with x lives
τ = Offence
τ^* = Driving with no licence
$\neg D$ = Licence suspended

Time	Network
1	D with 3 lives
2	$D, \tau, \tau \twoheadrightarrow D$
	D two lives
3	$D, \tau \quad \tau \twoheadrightarrow D$
	D one life
4	$D, \tau \quad \tau \twoheadrightarrow D$
	D no lives
5	$\neg D, \tau, \tau^*$
	offence
6	$\neg D, \tau^*$
$\vdots$	$\neg D$
36	$\neg D$

Figure 37

Time 1:	$D3, \tau \twoheadrightarrow D3$
Time 2:	$D2, \tau \twoheadrightarrow D2$
Time 3:	$D1, \tau \twoheadrightarrow D1$
Time 4:	$D0, \tau \twoheadrightarrow D0$
Time 5:	$\neg D, \tau, \tau^*$
Time 6–36:	$\neg D$

Figure 38

Example 4.2. *This example shows an application where the timed view does not work.*

In the UK if one has insurance paid for on direct debit then UK law says that if when it is time to renew the policy (say by December 31st) and something goes wrong and the direct debit does not work, then one is given one month (until January 31st) to renew (from December 31st).

So in this scenario the following can happen.

We have a claim C on, say, January 15th and the claim is rejected on the grounds

Dictionary

$C[15]$:	claim for accident January 15
$P[x]$:	policy valid at time x
$\Pi(x)$:	policy is paid for at time x.

Time January 15th – January 30th

1. $\neg\Pi(x), \neg\Pi(x) \twoheadrightarrow P(x)$
2. $\neg P(x) \twoheadrightarrow C(x)$

Time January 31st

3. $\Pi(31), P(31), C(15)$
4. $\Pi(31) \twoheadrightarrow \neg P(15)$

Figure 39

that the policy is not valid, it not having been renewed! However, if payment is done by January 31st, then the policy is renewed retrospectively from December 31st and the January 15th claim is accepted.

The timed presentation of this scenario is as follows (Figure 39):

In the timed presentation we cannot avoid contradiction in the presentation at time January 31st.

We need to add to the timed presentation the attack $\Pi(31) \twoheadrightarrow \neg P(15)$ *but to explain where it comes from we need the evolutionary representation of the insurance law.*

The evolutionary reality allows for the consumer not to renew the insurance on December 31st and wait to see if a claim arises in the period January 1st to January 30th. If no claim arises the consumer can move to another new insurance company beginning a new insurance policy from January 31st. If asked why he is leaving the old company to the new one he can say he was hoping for a better deal.

Example 4.3 (Comparison with [29] Part 1)**.** *This example compares directly with two important papers [32] from COMMA 2010 and paper [29] from 2015.*

We address directly the longer paper [29]] of 2015. The authors say in their abstract, and we quote:

> *Temporal Argumentation Frameworks (TAF) represent a recent extension of Dungs abstract argumentation frameworks that consider the temporal availability of arguments.*

In a TAF, arguments are valid during specific time intervals, called availability intervals, while the attack relation of the framework remains static and permanent in time; thus, in general, when identifying the set of acceptable arguments, the outcome associated with a TAF will vary in time.

We introduce an extension of TAF, called Extended Temporal Argumentation Framework (E-TAF), adding the capability of modeling the temporal availability of attacks among arguments, thus modeling special features of arguments varying over time and the possibility that attacks are only available in a given time interval.

1. *The first and second paragraphs of the above quotation declares that TAF is a temporal extension of the Dung approach. This means (and indeed is used in their paper) that they use the concepts of conflict free subsets and admissibility to form extensions and the the "Arguments Entities" they use and to which they apply the Dung machinery are "Temporally annotated arguments units".*

   This is not the case with our paper. Our Malkinson example and discussion in Section 1.1 does not conform to the basic dung machinery but we use time in evolutionary way.

2. *The third paragraph of the above quotation adds that their system E-TAF also temporally annotates the attack arrows. We also do that in our paper but we use all annotations in an evolutionary manner.*

3. *So what paradigm example application is compatible with the authors' machinery? Our answer is the consistency checking of legal laws that apply differently at different times and we want to verify the laws do not clash.*

   For example taxation laws. The government may declare a new package of business tax increases spread forward over a period of 5 years and the author model may check whether any clashes arise. The key word is tax legislation into the future NOT LEGISTLATION INTO THE PAST.[4]

Let us now examine one of the authors examples which brings out the difference. We quote from their paper:

Begin quote 2, from [29] page 33, (I modified the notation):

The arguments are $\{A, B\}$

[4](The UK government does legislate into the past causing sometimes great resentment, and Evolutionary Temporal Argumentation is needed to model such legislation, but other EU countries do not do that and consider backwards Tax Legislation a TABOO!)

The Attack is $A \twoheadrightarrow B$

The temporal span is [0...60]

The temporal annotation for the arguments are

$E = \{(A, [0...40]), (B, [30...60])\}$*, and note that according to [29] E is the set of "Temporally annotated arguments units" for which the Dung Conflict free concept is applied.*

The attack of A on B (i.e. the double arrow $A \twoheadrightarrow B$*) is annotated by* $\{((A, B), [30...35])\}$

The authors say, and I quote

> *"Indeed, E is not a conflict-free collection of t-profiles, since the argument A attacks the arguments B in the time interval [3035]"*

Going back to our proposed interpretation of consistency of legislation, a reasonable "consistency- conflict free" is the set

$$E - Con = \{(A, [0...29], (36...40)), (B, [36...60])\}$$

of arguments from time 0 to time 60.

The period [30...35] contains a conflict between A and B. We can decide in the Meta-level that A is dominant or we may not.

From the point of view of our paper, we look at the evolution from time 0 to time 60. We see that at time 30 there is an attack from A to B. Depending on the meaning of A and B we could extend the attack into the future up to time 60. This is forward looking.

Let us give a tax interpretation to A and B:

Assume A is a new tax on Builders B of luxury apartments. If the contract B starts at the time 30 when the tax A is instituted then we can adopt the view that it will continue to be valid until time 60 when the contract B terminates, despite the fact that the tax law A was canceled at time 36. The **PP** *rationale could be that the contract B activity is still ongoing until time 60.*

Example 4.4 (Comparison with [29] Part 2)**.** *We continue our comparison with [29] by giving our own very simple algorithm which does the same job as the machinery in Sections 1–4 of [29].*

Let us take the example of Figure 4 of [29], page 30.

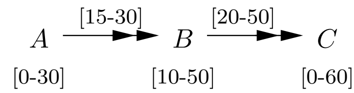


Figure 40

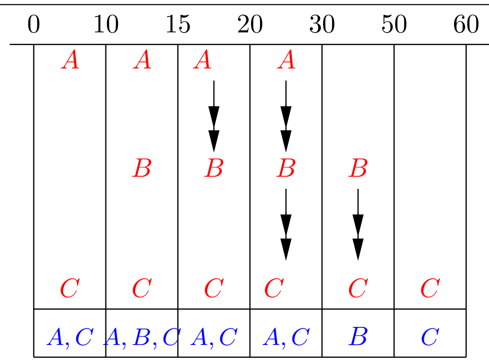


Figure 41

This example contains a time-stamped network (E-TAF) containing several independent parts. We use one of the parts to illustrate our algorithm. This illustration will make clear what [29] is doing, and how [29] is different from our paper. We concentrate on the $\{A, B, C\}$ part of Figure 4 of [29]. We represent it in our own Figure 40.

Note that all figures in [29] are finite. The temporal annotations of all components, arguments and attacks, are a finite list of intervals. We now give the algorithm.

Step 1. *Construct vertical lines, ordered according to time of all starting points and end point of each interval appearing in the Figure.*

Executing step 1 will yield the following figure 41.

Step 2. *For each minimal box of the form $[a, b]$ in the figure, we include the units (arguments and attacks) which are valid in the interval of the box. This is within Figure 41 in colour red.*

Step 3. *For each box compute all possible complete extensions. In our example there is only one. It is possible in general that there might be more. In general for each box $[a, b]$, let $E^1_{[a,b]}, E^2_{[a,b]} \ldots$ be all extensions. In Figure 41 at the bottom row of the figure we indicate the extensions for each in colour blue.*

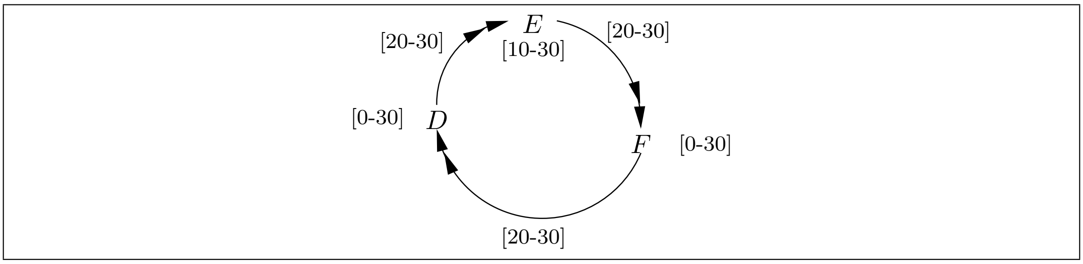


Figure 42

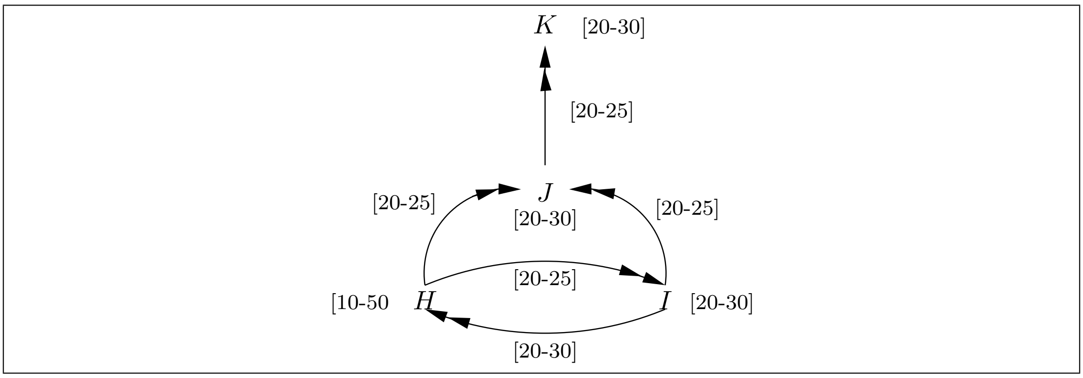


Figure 43

Step 4. *The extensions according to [29] for the annotated Figure 40 can be obtained from the blue bottom box of Figure 41.*

We get

$$\begin{array}{l} C\ [0-30],[50-60] \\ B\ [10-15],[30-50] \\ A\ [0-30] \end{array}$$

Indeed, in [29, page 31]. Example 5 (of [29]), this is exactly what is declared.

Example 4.5 (Comparison with [29], Part 3)**.** *We continue our analysis of parts of Figure 4 of [29, p. 30].*

We use our algorithm on two more subfigures of Figure 4 of [29, p. 30]. These are the loop of arguments $\{D, E, F\}$ and the loop of arguments $\{H, I, J, K\}$.

These are presented here in Figures 42 and 43.

Applying our algorithm to the network of Figure 42, we get the box Figure 44.

Applying our algorithm to the network of Figure 43, we get the box Figure 45.

The colour coding is as in the previous example 4.4.

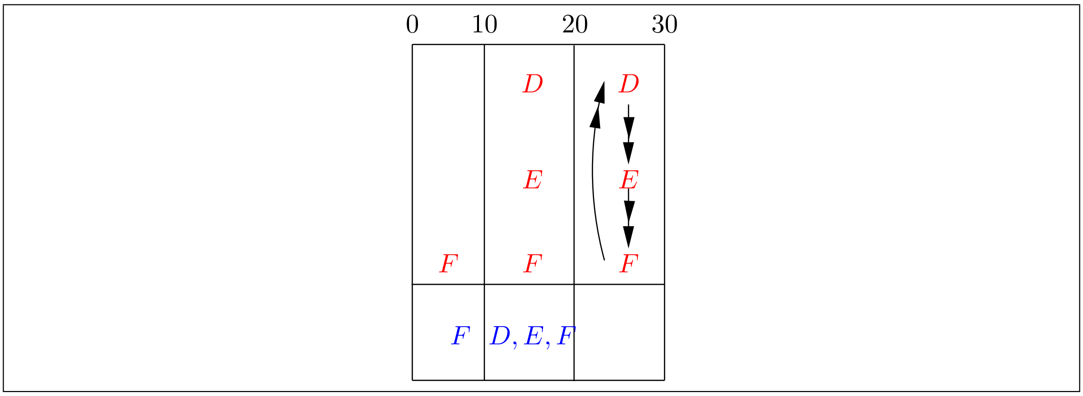


Figure 44

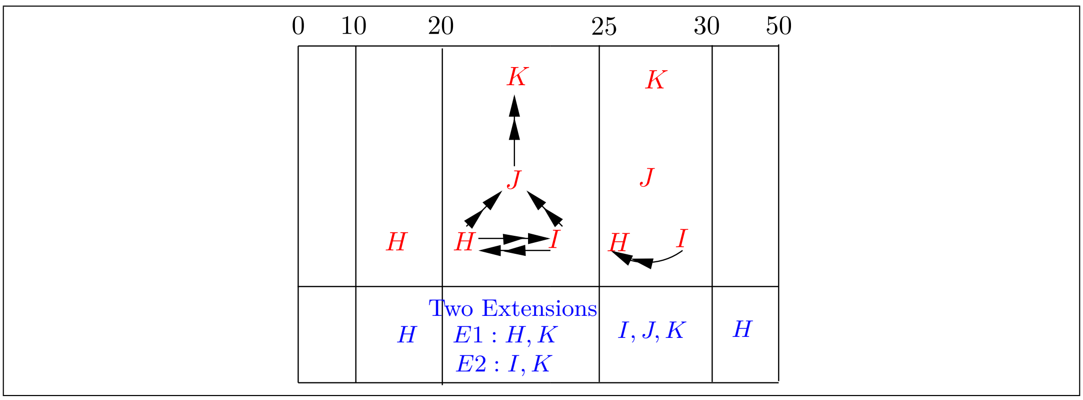


Figure 45

We make the following observations.

1. The results of the extensions we get agree fully with the extensions in [29, Example 5, bottom page 31, and top of page 32].

2. The authors of [29] use Dung machinery. For this reason, in Figure 44 the three cycle in box [20-30] has the empty extension. The way our algorithm works it allows us to choose the extensions in any box according to how we want. For example, we can choose CF2 semantics [3] for cycles in one box and Dung in another, all depending on the application area involved.

3. Our algorithm is simple conceptually and effective computationally. It can be easily generalised to an infinite number of intervals annotations. We just get an infinite number of boxes.

4. In [29, Sections 5–end] the authors investigate structured argumentation with time stamping. For the purpose of comparing with our evolutionary approach there is nothing new to compare. The temporal approach of [29] remains the same when structure is added.

5 Discussion, future research and conclusion

The many-lives approach is new (see [1, 2]), the idea of adding the many-lives function to abstract argumentation network. It does not fall under numerical argumentation. The way it is handled is inspired by the sex offenders case studies area.

There is a need for further research, investigating the place of many-lives in the general abstract argumentation landscape. There are many questions to be answered , among them the following:

1. What kind of semantics we should offer for systems with many-lives?
2. Can many-lives semantics simulate known semantics for traditional single life?

   For example can we simulate CF2 semantics by giving the elements of maximal conflict free sets more lives? (See Remark 2.14.)
3. How to handle support in the context of many-lives? Does support add lives? Is support (in the context of sex offender's many-lives) a higher level attack (on attack)?
4. How to define reinstatement? How many-lives to reinstate?
5. What is the best view of temporal change of a many-lives network?
6. What is the variation of the many-lives concepts across different application areas which use many-lives? (See Example 4.1.)
7. We are currently also looking at many-lives case studies in Nutrition. The liver for example can be attacked by a variety of foods, such as Alcohol, Sugar, Gluten and more. Such attacks combine in different ways, requiring/motivating new types of higher level attacks. In fact we require 3 dimensional argumentation networks for proper modelling.
8. We can regard many-lives as a resource (say M gives American \$). An attack destroys resources of the target but also costs resources of the attacker. We need to develop the evolutionary temporal logic of resource attack and defence.

9. The algorithm presented in the detailed analysis and comparison or our paper with the important 2015 paper [29] suggests we can write a new paper extending and simplifying their results also to the infinite case. We shall use Neibourhood Ultrafilter Semantics on the temporal line.

We leave these and other questions for follow up papers

Acknowledgements

This paper is a revised version of our paper from 2017, [6], submitted to a special issue of the *Journal of Applied Logics*. The referees did a very thorough job, but we did not revise in time. We thank the anonymous referees of that paper as well as Matthias Thimm and Leon van der Torre and the referees for most valuable comments.

References

[1] D. Gabbay, G. Rozenberg and L. Rivlin. Reasoning under the influence of universal distortion. *Ifcolog Journal of Logic and Their Applications*, 4(6), 1769–1900, July 2017.

[2] D. Gabbay, G. Rozenberg and Students of CS Ashkelon Introducing Abstract Argumentation with many-lives, In *Journal of Applied Logics*, Volume 7, Number 2, June 2020, pp 295-336.

[3] P. Baroni, M. Giacomin, G. Guida. SCC-recursiveness: a general schema for argumentation semantics. *J. Artificial Intelligence*, 168(1):162–210, 2005.

[4] Maximiliano C.D.Budán , Maria LauraCobo Diego C.Martinez Guillermo R.Simari. Bipolarity in temporal argumentation frameworks, *International Journal of Approximate Reasoning* 84 (2017) 122

[5] H. Barringer, D.M. Gabbay and J.Woods Modal and temporal argumentation networks. *Argument and Computation*, Vol. 3, Nos. 23, JuneSeptember 2012, 203227

[6] D. Gabbay, G. Rozenberg, Temporal Aspects of many-lives Argumentation Networks, Submitted to *JAL special issue for Isralog'2017* (rejected, needs revision)

[7] D. Gabbay. Meta-logical Investigations in Argumentation Networks. College Publications, 2013, 770pp.

[8] D Gabbay, I Shapirovsky and V Shehtman. Products of Modal Logics and Tensor Products of Modal Algebras, *Journal of Journal of Applied Logic* 12 (2014), pp. 570-583.

[9] H. Prakken and G. Vreeswijk. Logics for Defeasible Argumentation, in Handbook of philosophical logic vol 4, D. Gabbay and F. Guenthner, editors, Springer 2002, Pages 219-318

[10] P M Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games Artificial Intelligence Volume 77, Issue 2, September 1995, Pages 321–357

[11] M. Caminada and D. Gabbay. A logical account of formal argumentation. Studia Logica, 93(2-3): 109-145, 2009.

[12] D. Grossi and S. Modgil. On the graded acceptability of arguments. In *Proceedings of IJCAI'15*, pp. 868–874, 2015.

[13] D. M. Gabbay Equational approach to argumentation networks *Argument and Computation* 3 (2-3):87 - 142 (2012)

[14] D. Gabbay, 2017, Abstract Generic Argumentation (AGA). In preparation

[15] C. Cayrol and M.-Ch. Lagasquie-Schiex. Graduality in argumentation. *Journal of Artificial Intelligence Research*, 23:245–297, 2005. [Davey and Priestley, 1990]

[16] S. Egilmez, J. Martins, and J. Leite. Extending social abstract argumentation with votes on attacks. In *Proc. 2nd Int.Workshop on Theory and Applications of Formal Argumentation*, pages 16–31, 2013.

[17] C. Cayrol and M.-C. Lagasquie-Schiex. Gradual acceptability in argumentation systems. In *Proc 3rd CMN (International workshop on computational models of natural argument)*, Acapulco, Mexique, pages 55-58, 2003.

[18] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proc. KR'10*, pages 102–111. AAAI Press, 2010

[19] G. Brewka, S. Ellmauthaler, H. Strass, J. P. Wallner, and S. Woltran. Abstract dialectical frameworks revisited. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence* (pp. 803-809). AAAI Press. (2013, August).

[20] S. Polberg.Extension based semantics of abstract dialectical frameworks. In STAIRS 2014: *Proceedings of the 7th European Starting AI Researcher Symposium* (Vol. 264, p. 240). IOS Press. (2014, August).

[21] Claudette Cayrol, Marie-Christine Lagasquie-Schiex. Graduality in Argumentation. *J. Artif. Intell. Res.*, 2005

[22] Nielsen, SH and Parsons, S 2006, A generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. in N Maudet, S Parsons and I Rahwan (eds), *Argumentation in Multi-Agent Systems (ArgMAS)*, 2006, pp 54-73

[23] D Gabbay and M Gabbay. Disjunctive attacks in argumentation networks, *Logic journal of IGPL* Volume 24, Issue 2, Pp. 186-218. `http://jigpal.oxfordjournals.org/content/early/2015/09/10/jigpal.jzv032.full.pdf`

[24] D. Gabbay, I. Hodkinson and M. Reynolds. *Temporal Logic: Mathematical Foundations and Computational Aspects, vol 1: Mathematical Foundations* (Monograph) Oxford University Press, 1994, 671 pp.

[25]] D. Gabbay, M. Reynolds and M. Finger. *Temporal Logic: Mathematical Foundations and Computational Aspects, Vol. 2: Computational Aspects* (Monograph) Oxford University Press, 2000. 600 pages

[26] H. Barringer and Dov Gabbay. Modal and Temporal Argumentation Networks old ver-

sion, in the Amir Pnueli Memorial Volume, *Time for Verification*, Springer LNCS, Doron Peled and Zohar Manna , Editors , 2010, pp 1-25

[27] H. Barringer D. Gabbay J. Woods Modal and Temporal Argumentation Networks, expanded version, *Argument and Computation*, 2012, vol 3 issues (2-3), pp 203-227

[28] Maria Laura Cobo, Diego C. Martínez, Guillermo Ricardo Simari: On Admissibility in Timed Abstract Argumentation Frameworks. ECAI 2010: 1007-1008.

[29] Maximiliano Celmo Budán, Mauro Javier Gómez Lucero, Carlos Iván Chesñevar, Guillermo Ricardo Simari: Modeling time and valuation in structured argumentation frameworks. *Inf. Sci.* 290: 22-44(2015)

[30] Timotheus Kampik and Dov Gabbay. The Degrees of Monotony-Dilemma in Abstract Argumentation published in J. Vejnarová and N. Wilson (Eds.): *ECSQARU 2021*, LNAI 12897, pp. 89–102, 2021.

[31] D. M. Gabbay. Compromise Update and Revision A position paper. In *Dynamic Worlds*, B. Fronhoffer and R. Pareschi, eds. pp. 111148. Kluwer, 1999.

[32] N.D. Rotstein, M.O. Moguillansky, A.J. García, G.R. Simari, A dynamic argumentation framework, in: P. Baroni, F. Cerutti, M. Giacomin, G.R. Simari (Eds.), *Computational Models of Argument: Proceedings of COMMA 2010, Frontiers in Artificial Intelligence and Applications*, vol. 216, IOS Press, Italy, 2010., pp. 427–438.

Received May 2022