Representing Kinship Relations on the Semantic Web

Domenico Cantone[1], Aldo Gangemi[2,3], and Cristiano Longo[4]

[1] Università di Catania, Italy
[2] STLab, ISTC-CNR, Rome, Italy
[3] LIPN, Université Paris 13-CNRS-Sorbonne Cité, France
[4] Network Consulting Engineering (NCE) S.r.l, Valverde (CT), Italy

Abstract. Kinship plays a fundamental role in human communities as a basic principle for organizing individuals into social groups. Representing kinship relationships in a formal and precise way is then a crucial task when modelling many knowledge domains, and it may constitute a relevant benchmark for the reasoning layer of the Semantic Web. In this paper we face the problem of representing some basic aspects of kinship relationships using the fragments of OWL2 corresponding to the description logics $\mathcal{SROIQ}$ and $\mathcal{EL}^{++}$. As a first step, we provide an intuitive but formal description of some kinship terms, using an expressive description logic. From this intuitive description, we derive a set of test cases, indicating the expected inferences that should be drawn from the candidate ontologies. Finally, three different ontologies (two for $\mathcal{SROIQ}$ and one for $\mathcal{EL}^{++}$), whose *coherence* with the intuitive description of kinship terms is established, are compared by means of these test cases.

1 Introduction

This paper reports work about representing and reasoning over *kinship relationships* when using OWL2. The practical impact of this study is not only in describing a relevant ontology engineering problem, but also in being able to make sense of heterogeneous kinship data emerging from the linked open data cloud.

Kinship relationships are important knowledge crossing the biological, anthropological, social, and legal domains. Such knowledge involves non-trivial reasoning even at a purely cognitive human level. As a matter of fact, much of the historically evolved knowledge organization systems have been based on metaphors deriving from kinship knowledge: trees, transitivity, syllogistic reasoning, etc. But kinship knowledge goes well beyond purely formal relations: different conceptual dimensions intersect at the kinship level: "blood" (genetic) relations, breeding relations, nursing relations, transfer of legal rights and obligations, etc. Those dimensions play complicated roles in the cultural evolution of practices and laws, so that complicated reasoning tasks emerge, and the typical reasoning on those data is pretty simple compared to the requirements of tasks defined for kinship knowledge. The growing amount of kinship knowledge flowing from Big Data and especially in Linked Open Data is on one hand quite simple as compared to actual kinship reasoning needs, but on the other hand it is rather messy. It is therefore important to establish explicit representation and reasoning patterns over kinship (e.g., in order to enhance kinship linked data), and to exploit them in realistic applications. Kinship seems then a relevant benchmark for the reasoning layer of the Semantic Web: does OWL2 DL support kinship reasoning in ways that are nontrivial, e.g., lightweight entailment regimes of SPARQL? or do we need stronger ones, e.g., RIF?[5]

The description logic [3] varieties of OWL2[6] are globally underpinned by the $\mathcal{SROIQ}$ description logic, described in [8]. The first-order direct semantics of OWL2 enables *reasoning* on OWL2 knowledge bases, thus allowing one on the one hand to test whether a knowledge base is incoherent/inconsistent, and on the other hand to perform *inferences*, which permit to derive additional information from the facts explicitly mentioned in the knowledge base.

An example of inference consists in automatically detecting that *Alice* is the grandmother of *Charlie* from the facts that *Alice* is the mother of *Bob* and that *Bob* is the father of *Charlie*.

To be useful in practice, a description logic must admit automated algorithms for computing inferences. In other words, reasoning in the considered description logic must be *decidable*. As a consequence, the expressivity of currently used description logics is somehow limited. This is the case for the aforementioned description logic $\mathcal{SROIQ}$ which, among others, imposes some restrictions on predicates stating transitivity and irreflexivity of roles. Such kind of limitations may be frustrating when one attempts to describe a knowledge domain using a specific description logic, and the lower is the expressive power of the logic, the more pronounced is this issue (see Section 6). Moreover, due to these limitations one may be forced to provide some *counterintuitive* constraints at this stage. For example, in order to increase the inferencing capabilities of the resulting ontology, in the knowledge base $\mathcal{K}^q_{\mathcal{S}}$, reported in Section 5, we adopted a recursive definition for the concept of all persons with an Italian ancestor rather than the more intuitive definition $\mathtt{ItDescendant} \equiv \exists \mathtt{descendantOf}.\mathtt{Italian}$.

In this paper, we use a method based on a formal representation of the requirements for an ontology of the kinship knowledge domain. This starts from an *intuitive* (but precise) description $\mathcal{K}_{\mathcal{L}}$ of the knowledge domain we are describing. This description is intuitive in the sense that the correspondence between the definitions and the intuitive meaning of concept and roles is immediately clear, thanks to the shared conceptualization of (basic) kinship relations. The description is provided by means of a formal language, which is powerful enough to express suitable limitations on the specifications of conceptualizations and, therefore, not necessarily decidable. Then, given a candidate set of description logic constraints $\mathcal{K}$, its *coherence* with the intuitive description $\mathcal{K}_{\mathcal{L}}$ must be verified, in order to guarantee that no unexpected consequence can be deduced from $\mathcal{K}$. In other words, this verification phase ensures that every fact that can be

[5] `http://www.w3.org/standards/techs/rif#w3c_all`
[6] `http://www.w3.org/TR/2009/REC-owl2-overview-20091027/`

inferred from $\mathcal{K}$ must also be inferable from $\mathcal{K}_{\mathcal{L}}$ as well. On the other hand, due to the expressive limitations of decidable description logics, the inferences which can be drawn from $\mathcal{K}$ may be a strict subset of those which can be drawn from $\mathcal{K}_{\mathcal{L}}$. Thus, the inferencing capabilities of the candidate constraint set $\mathcal{K}$ have to be tested against a collection of *expected inferences* (see for example Tables 2 and 3), defined starting from the intuitive description $\mathcal{K}_{\mathcal{L}}$. Such test cases, whose execution can be easily automatized, provide an immediate view of which kinds of inferences will be drawn from a set of constraints and which will not.

The paper is organized as follows. In Section 2 work related to representing kinship using Semantic Web languages is briefly reviewed. In Section 3 some preliminary notions about the description logic framework are recalled. Section 4 provides an intuitive but formal description of kinship using a really expressive description logic. In Section 5 two different descriptions of kinship in terms of $\mathcal{SROIQ}$-constraints are provided and compared. The less expressive (but more *efficient*) description logic $\mathcal{EL}^{++}$ is used for building a kinship ontology in Section 6. Finally, in Section 7 we draw our conclusions, and provide some hints for future work.

2 Related work

In the large area of social relationships, *kinship* plays a fundamental role, since it pervades (in different ways) all human communities as a basic principle for organizing individuals into social groups. In light of this, kinship relationships have been intensively studied by anthropologists,[7] in order to understand how they influence social organization in human communities. Also population genetics is heavily involved in studying kinship relations with respect to the dynamics of genetic inheritance in humans [10]. Sociology [12], law, and jurisprudence [4] as well have substantially contributed to shape current perception and representation of kinship in human cultures.

Research focused on kinship and, more in general, on human relationships, is also active in the field of knowledge representation. The *'Friend Of A Friend'* (FOAF) vocabulary is a *de facto* standard for representing social relationships in the semantic web,[8] and it provides a generic human-to-human relation `knows`.

The RELATIONSHIP vocabulary, presented in [5], contains a basic set of kinship terms.[9] Among other human-to-human relationships, it provides vocabulary terms for representing childhood, parenthood, siblinghood, and a relation `SpouseOf`.

A more comprehensive set of kinship terms is provided by the *agrelon* vocabulary,[10] devised in the context of the *CONTENTUS* Project [6]. One of the most interesting aspects of this terminology is the distinction between *legal* and *natural* relationships. For example, natural and adoptive children can be

[7] E.g., `http://anthro.palomar.edu/kinship/kinship_5.htm`
[8] `http://www.foaf-project.org/`
[9] `http://vocab.org/relationship/.html`
[10] `http://www.contentus-projekt.de/fileadmin/download/agrelon.owl`

distinguished by means of the two distinct relations `hasBiologicalChild` and `hasAdoptiveChild`.

Some considerations about human relationships are presented in [13], where the authors point out that every relationship among persons is in some way *caused* by one or more specific events, for example childhood is caused by a birth event.

From a different perspective, in [14] the authors propose an ontology of (complex) Social Relationships, based on the foundational ontology *DOLCE* (cf. [7]), extended with the D&S (Descriptions and Situations) framework. The ontology enables the definition of *context-specific* relationships, i.e., relationships which may hold or not depending on the context we are considering. This is possible because social relations are *reified*: relation reification design patterns make relation representation very flexible, but they also prevent full-fledged description logic reasoning on kinship relations, unless appropriate binary projections of reified relations are provided.

A more reasoning-oriented work is contained in [15], which describes the *Family History Knowledge Base* (in short *FHKB*). This ontology, which has been developed as a tutorial example to highlight some features of OWL2, provides definitions and constraints for a considerable amount of kinship relations. However, the authors report some issues that prevent applications from using it. For example, the relation `isSiblingOf` is explicitly stated to be transitive and symmetric, and thus it must be necessarily reflexive.

In general, the aforementioned vocabularies do not provide a relevant amount of intensional knowledge, as they limit themselves to supply a set of names to denote different human-to-human relationships. By converse, in this paper we provide *reasoning-oriented* definitions of some kinship relationships by means of description logics. We begin with characterizing kinship relationships derived from common sense, using a very expressive description logic.

3 Description Logics

Description logics (the reader may refer to [3] for a quite complete introduction to the description logic framework) are a family of logic-based formalisms which allow one to describe a knowledge domain in terms of *individuals*, to denote domain elements, *concepts*, which denote domain subsets, and *roles*, which designate relations among domain items, and to state constraints on the domain structure.

Basic syntactic building blocks of all languages in this framework are the three denumerable infinities of *concept names*, *role names*, and *individual names*. Each description logic is mainly characterized by a set of *constructors* (see the first part of Table 1), which allow one to define complex concepts and roles starting from concept, role, and individual names, and by the types of *constraints* (see the second part of Table 1) which can form a *knowledge base* (namely, a finite set of description logic constraints). In what follows, we will use the term *concepts* to indicate concept names and complex concepts (i.e., concepts constructed from other concepts by means of concept constructors). Analogously, we will use the term *roles* for role names and complex roles.

Description logic semantics is given in terms of *interpretations*. An interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty set (*interpretation domain*) and $\cdot^{\mathcal{I}}$ is an *interpretation function* which associates domain subsets, relations on the domain, and domain items to concept names, role names, and individual names, respectively. The interpretation function extends to complex terms (concepts and roles) as indicated in the first part of Table 1, where C,D are concepts, R,S are roles, a is an individual name, and n is a nonnegative integer. An interpretation $\mathcal{I}$ *evaluates* a constraint γ by assigning a truth value to it. Evaluation of constraints by a given interpretation $\mathcal{I}$ is defined as in the second part of Table 1, with C,D concepts, R,S roles, and a,b individual names. We write $\mathcal{I} \models \gamma$ to indicate that the constraint γ is evaluated to **true** by the interpretation $\mathcal{I}$. Otherwise, we write $\mathcal{I} \not\models \gamma$. An interpretation $\mathcal{I}$ is said to be a *model* for a knowledge base $\mathcal{K}$ (and we write $\mathcal{I} \models \mathcal{K}$) if $\mathcal{I} \models \gamma$ for all $\gamma \in \mathcal{K}$. A knowledge base $\mathcal{K}$ is said to be *consistent* if $\mathcal{K}$ admits a model. Given any two knowledge bases $\mathcal{K}$ and $\mathcal{K}'$, we say that $\mathcal{K}$ *entails* $\mathcal{K}'$, and write $\mathcal{K} \Longrightarrow \mathcal{K}'$, if and only if one has $\mathcal{I} \models \mathcal{K}'$ whenever $\mathcal{I} \models \mathcal{K}$ holds, for all interpretations $\mathcal{I}$.

4 Formalizing requirements for kinship relations

In this section we introduce the basic terminology used to describe kinship relationships. We also describe a set of constraints $\mathcal{K}_{\mathcal{L}}$ intended to characterize in a precise and formal way the terms so introduced, according to their intuitive meaning. We remark that the set of constraints devised in this phase has a purely descriptive intent, and therefore one should not care about the decidability of the representation language involved. Let us put:

$$
\begin{array}{rll}
\mathcal{K}_{\mathcal{L}} = \{ & \mathtt{dom}(\mathtt{relativeOf}) \sqsubseteq \mathtt{Person}, & \textbf{[C1]} \\
& \mathtt{Sym}(\mathtt{relativeOf}), & \textbf{[C2]} \\
& \mathtt{relativeOf} \equiv \mathtt{relativeOf}^{+} \sqcap (\neg \mathtt{id}(\mathtt{Person})), & \textbf{[C3]} \\
& \mathtt{partnerOf} \sqsubseteq \mathtt{relativeOf}, & \textbf{[C4]} \\
& \mathtt{Sym}(\mathtt{partnerOf}), & \textbf{[C5]} \\
& \mathtt{childOf} \sqsubseteq \mathtt{relativeOf}, & \textbf{[C6]} \\
& \mathtt{parentOf} \equiv \mathtt{childOf}^{-}, & \textbf{[C7]} \\
& \mathtt{descendantOf} \equiv \mathtt{childOf}^{+}, & \textbf{[C8]} \\
& \mathtt{descendantOf} \sqsubseteq \mathtt{relativeOf}, & \textbf{[C9]} \\
& \mathtt{ancestorOf} \equiv \mathtt{parentOf}^{+} \ \}. & \textbf{[C10]}
\end{array}
$$

The concept `Person` is intended to denote all human beings, whether alive or not. Firstly, the generic relation `relativeOf` over persons (see **[C1]**) is defined. It may be noticed that no constraint about its range is present in $\mathcal{K}_{\mathcal{L}}$. But **[C2]** enforces the symmetricity of the `relativeOf` relation, in order to guarantee that everyone is relative of her relatives, and consequently the domain and the range of `relativeOf` will coincide. The `relativeOf` relation must be such that all the relatives of a person's relatives are relatives of the person herself. However, we have to impose the irreflexivity of the `relativeOf` relation as, for example, one

Term	**Semantics** $(\cdot)^{\mathcal{I}}$
$\top$	$\Delta^{\mathcal{I}}$
$\bot$	$\emptyset$
$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$\{a\}$	$\{a^{\mathcal{I}}\}$
$\exists R.\mathtt{Self}$	$\{u \in \Delta^{\mathcal{I}} : [u,u] \in R^{\mathcal{I}}\}$
$\forall R.C$	$\{u \in \Delta^{\mathcal{I}} : (\forall [u,v] \in R^{\mathcal{I}})(v \in C^{\mathcal{I}})\}$
$\exists R.C$	$\{u \in \Delta^{\mathcal{I}} : (\exists v \in C^{\mathcal{I}})([u,v] \in R^{\mathcal{I}})\}$
$\leq nR.C$	$\{u \in \Delta^{\mathcal{I}} : \lvert\{v \in C^{\mathcal{I}} : [u,v] \in R^{\mathcal{I}}\}\rvert \leq n\}$
$\geq nR.C$	$\{u \in \Delta^{\mathcal{I}} : \lvert\{v \in C^{\mathcal{I}} : [u,v] \in R^{\mathcal{I}}\}\rvert \geq n\}$
$\mathbb{U}$	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
$\neg R$	$(\Delta \times \Delta) \setminus R^{\mathcal{I}}$
$R \sqcup S$	$R^{\mathcal{I}} \cup S^{\mathcal{I}}$
$R \sqcap S$	$R^{\mathcal{I}} \cap S^{\mathcal{I}}$
R^-	$\{[u,v] \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} : [v,u] \in R^{\mathcal{I}}\}$
$\mathtt{id}(C)$	$\{[u,u] : u \in C^{\mathcal{I}}\}$
R^+	$(R^{\mathcal{I}})^+$
Constraint	**Semantics** $\mathcal{I} \models (\cdot)$ **iff**
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$
$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
$R \equiv S$	$R^{\mathcal{I}} = S^{\mathcal{I}}$
$\mathsf{dom}(R) \sqsubseteq C$	$(\forall [x,y] \in R^{\mathcal{I}})(x \in C^{\mathcal{I}})$
$\mathsf{range}(R) \sqsubseteq C$	$(\forall [x,y] \in R^{\mathcal{I}})(y \in C^{\mathcal{I}})$
$R_1 \circ \ldots \circ R_n \sqsubseteq P$	$R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \subseteq P^{\mathcal{I}}$
$\mathtt{Sym}(R)$	$(\forall [x,y] \in R^{\mathcal{I}})([y,x] \in R^{\mathcal{I}})$
$\mathtt{Tra}(R)$	$(\forall [x,y],[y,y'] \in R^{\mathcal{I}})([x,y'] \in R^{\mathcal{I}})$
$\mathtt{Ref}(R)$	$(\forall x \in \Delta^{\mathcal{I}})([x,x] \in R^{\mathcal{I}})$
$\mathtt{ASym}(R)$	$(\forall [x,y] \in R^{\mathcal{I}})([y,x] \notin R^{\mathcal{I}})$
$\mathtt{Irr}(R)$	$(\forall [x,y] \in R^{\mathcal{I}})(x \neq x)$
$\mathtt{Dis}(R,S)$	$R^{\mathcal{I}} \cap S^{\mathcal{I}} = \emptyset$
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$R(a,b)$	$[a^{\mathcal{I}}, b^{\mathcal{I}}] \in R^{\mathcal{I}}$
$\neg(R(a,b))$	$[a^{\mathcal{I}}, b^{\mathcal{I}}] \notin R^{\mathcal{I}}$
$a = b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

Table 1. Some common description logic constructs

can't be the mother of herself. For this reason, we do not require the transitivity of the `relativeOf` relation since this, in conjunction with symmetricity, would imply also reflexivity, but we provide instead the constraint **[C3]**. Indeed, the relation `relativeOf` is intended to be as general as possible in order to subsume all the kinship relations. It can be specialized to capture different notions of kinship in different countries and cultures.

As basic kinship relationships, we consider childhood, denoted by `childOf` (**[C6]**), and `partnerOf` (see **[C4]**), which connects all the persons that are either member of a married couple, or of an established unmarried couple, or having sex together. For kinship, in many applications this is the correct generalization. Several relations like `spouseOf`, `unmarriedPartnerOf`, `loverOf`, `havingSexWith` can all be specialization of this general property. Legally speaking, all or only some of them will be considered according to country's laws. Plainly, the `partnerOf` relation must be symmetric, as stated by the constraints **[C5]**. In addition, the irreflexivity of `partnerOf`, which ensures that no one is partner with herself, directly descends from the irreflexivity of `relativeOf`.

Next, the inverse relation `parentOf` of `childOf` in **[C7]**, stating that every person must be child of her parents and parent of her children, is introduced. Together with the `childOf` and `parentOf` relations, their transitive closures, respectively `descendantOf` (**[C8]**) and `ancestorOf` (**[C10]**), are also introduced. Notice that since `parentOf` and `childOf` are mutually inverses, so must be their transitive closures. No person can be either a descendant of any of her descendants or the ancestor of any of her ancestors. In addition, no person can be both ancestor and descendant of another person at the same time. In other words, the relations `descendantOf` and `ancestorOf` are asymmetric and pairwise disjoint. However, as these relations are transitive and mutually inverses of one another, their asymmetricity and pairwise disjointness directly descends from the irreflexivity of the super-relation `relativeOf` (see **[C9]**).

The following knowledge base is trivially entailed by $\mathcal{K}_{\mathcal{L}}$:

$$
\begin{array}{rll}
\widehat{\mathcal{K}}_{\mathcal{L}} = \{ & \mathsf{range}(\mathtt{relativeOf}) \sqsubseteq \mathtt{Person}, & \textbf{[C11]} \\
& \mathtt{Irr}(\mathtt{relativeOf}), & \textbf{[C12]} \\
& \mathtt{Irr}(\mathtt{partnerOf}), & \textbf{[C13]} \\
& \mathtt{Irr}(\mathtt{childOf}), & \textbf{[C14]} \\
& \mathtt{ASym}(\mathtt{childOf}), & \textbf{[C15]} \\
& \mathtt{parentOf} \sqsubseteq \mathtt{relativeOf}, & \textbf{[C16]} \\
& \mathtt{Irr}(\mathtt{parentOf}), & \textbf{[C17]} \\
& \mathtt{ASym}(\mathtt{parentOf}), & \textbf{[C18]} \\
& \mathtt{Irr}(\mathtt{descendantOf}), & \textbf{[C19]} \\
& \mathtt{ASym}(\mathtt{descendantOf}), & \textbf{[C20]} \\
& \mathtt{childOf} \sqsubseteq \mathtt{descendantOf}, & \textbf{[C21]} \\
& \mathtt{ancestorOf} \equiv \mathtt{descendantOf}^{-}, & \textbf{[C22]} \\
& \mathtt{ancestorOf} \sqsubseteq \mathtt{relativeOf}, & \textbf{[C23]} \\
& \mathtt{Irr}(\mathtt{ancestorOf}), & \textbf{[C24]} \\
& \mathtt{ASym}(\mathtt{ancestorOf}), & \textbf{[C25]} \\
& \mathtt{parentOf} \sqsubseteq \mathtt{ancestorOf}, & \textbf{[C26]} \\
& \mathtt{Dis}(\mathtt{ancestorOf}, \mathtt{descendantOf})\ \}\,. & \textbf{[C27]}
\end{array}
$$

The expressive definition of the primitive kinship relationships provided by $\mathcal{K}_{\mathcal{L}}$ and $\widehat{\mathcal{K}}_{\mathcal{L}}$ will be used as requirements to build some kinship ontologies in the well-known description logics $\mathcal{SROIQ}$ (cfr. $\mathcal{K}'_{\mathcal{S}}$ and $\mathcal{K}''_{\mathcal{S}}$ in Section 5) and $\mathcal{EL}^{++}$ (cfr. $\mathcal{K}_{\mathcal{E}}$ in Section 6). It will be proved that each of the proposed ontologies is entailed by $\mathcal{K}_{\mathcal{L}}$, in order to verify their *coherence* with $\mathcal{K}_{\mathcal{L}}$ itself. In addition, we derive a collection of *test cases* (see Table 2) from the constraints in $\mathcal{K}_{\mathcal{L}}$ and $\widehat{\mathcal{K}}_{\mathcal{L}}$, which will be used to compare the inferencing capabilities of the candidate ontologies. Such test cases consists of a set of assertions (namely constraints of the forms $C(a)$, $R(a,b)$, and $\neg(R(a,a))$) called *premises*, and an expected *consequence*, i.e., an assertion which should be inferred from the premises.

	Premises	**Consequence**	$\mathcal{K}_{\mathcal{S}}$	$\mathcal{K}'_{\mathcal{S}}$	$\mathcal{K}''_{\mathcal{S}}$	$\mathcal{K}_{\mathcal{E}}$
[C1]	$Alice$ `relativeOf` Bob	$Alice \in$ `Person`	Y	Y	Y	Y
[C2]	$Alice$ `relativeOf` Bob	Bob `relativeOf` $Alice$	Y	Y	Y	**N**
[C3]	$Alice$ `relativeOf` Bob Bob `relativeOf` $Charlie$ $Alice \neq Bob \neq Charlie$	$Alice$ `relativeOf` $Charlie$	**N**	**N**	**N**	**N**
[C4]	$Alice$ `partnerOf` Bob	$Alice$ `relativeOf` Bob	Y	Y	Y	Y
[C5]	$Alice$ `partnerOf` Bob	Bob `partnerOf` $Alice$	Y	Y	Y	**N**
[C6]	$Alice$ `childOf` Bob	$Alice$ `relativeOf` Bob	Y	Y	Y	Y
[C7]	$Alice$ `parentOf` Bob	Bob `childOf` $Alice$	Y	Y	Y	**N**
[C8]	$Alice$ `descendantOf` Bob Bob `descendantOf` $Charlie$ $Alice \neq Bob \neq Charlie$	$Alice$ `descendantOf` $Charlie$	**N**	Y	**N**	Y
[C9]	$Alice$ `descendantOf` Bob	$Alice$ `relativeOf` Bob	Y	Y	Y	Y
[C10]	$Alice$ `ancestorOf` Bob Bob `ancestorOf` $Charlie$ $Alice \neq Bob \neq Charlie$	$Alice$ `ancestorOf` $Charlie$	**N**	Y	**N**	Y
[C11]	$Alice$ `relativeOf` Bob	$Bob \in$ `Person`	Y	Y	Y	Y
[C12]	$Alice$ `relativeOf` x	$Alice \neq x$	**N**	**N**	Y	**N**
[C13]	$Alice$ `partnerOf` x	$Alice \neq x$	Y	Y	Y	**N**
[C14]	$Alice$ `childOf` x	$Alice \neq x$	**N**	**N**	Y	**N**
[C15]	$Alice$ `childOf` Bob	$\neg(Bob$ `childOf` $Alice)$	**N**	**N**	Y	**N**
[C16]	$Alice$ `parentOf` Bob	$Alice$ `relativeOf` Bob	Y	Y	Y	Y
[C17]	$Alice$ `parentOf` x	$Alice \neq x$	**N**	**N**	Y	**N**
[C18]	$Alice$ `parentOf` Bob	$\neg(Bob$ `parentOf` $Alice)$	**N**	**N**	Y	**N**
[C19]	$Alice$ `descendantOf` x	$Alice \neq x$	**N**	**N**	Y	**N**
[C20]	$Alice$ `descendantOf` Bob	$\neg(Bob$ `descendantOf` $Alice)$	**N**	**N**	Y	**N**
[C21]	$Alice$ `childOf` Bob	$Alice$ `descendantOf` Bob	Y	Y	Y	Y
[C22]	$Alice$ `ancestorOf`, Bob	Bob `descendantOf` $Alice$	Y	Y	Y	**N**
[C23]	$Alice$ `ancestorOf` Bob	$Alice$ `relativeOf` Bob	Y	Y	Y	Y
[C24]	$Alice$ `ancestorOf` x	$Alice \neq x$	**N**	**N**	Y	**N**
[C25]	$Alice$ `ancestorOf` Bob	$\neg(Bob$ `ancestorOf` $Alice)$	**N**	**N**	Y	**N**
[C26]	$Alice$ `parentOf`, Bob	$Alice$ `ancestorOf` Bob	Y	Y	Y	Y
[C27]	$Alice$ `ancestorOf` Bob	$\neg(Alice$ `descendantOf` $Bob)$	**N**	**N**	Y	**N**

Table 2. Test results

5 Defining kinship in $\mathcal{SROIQ}$

$\mathcal{SROIQ}$ is a very expressive description logic introduced in [8]. In this section two different $\mathcal{SROIQ}$-knowledge bases for kinship are presented and compared. They are constructed starting from the intuitive description of the knowledge domain provided by $\mathcal{K}_{\mathcal{L}}$, but having to deal with the limitations imposed by $\mathcal{SROIQ}$. Two limitations are relevant in our context: (a) Boolean operators on roles, used in **[C3]**, are not allowed; (b) irreflexivity and asymmetry cannot be stated for transitive roles, or for subroles of transitive ones. Due to these limitations, the whole set of constraints in $\mathcal{K}_{\mathcal{L}}$ is not expressible in $\mathcal{SROIQ}$, at least in an intuitive way. In fact, for example, the irreflexivity of `descendantOf`, used in conjunction with transitivity, guarantees the acyclicity of the `childOf` and `parentOf` roles. There is no optimal solution for this issue. Instead, an ontology designer has to make some design choices in terms of constraints which are to be excluded and, consequently, inferences which will not be performed by the system. Hence, we provide a basic set of $\mathcal{SROIQ}$-constraints $\mathcal{K}_{\mathcal{S}}$ and two extensions of it, $\mathcal{K}'_{\mathcal{S}}$ and $\mathcal{K}''_{\mathcal{S}}$, which guarantee different inferencing capabilities:

$$
\begin{aligned}
\mathcal{K}_{\mathcal{S}} = \{\ & \exists\mathtt{relativeOf}.\top \sqsubseteq \mathtt{Person},\ \ \mathtt{Sym}(\mathtt{relativeOf}),\\
& \mathtt{partnerOf} \sqsubseteq \mathtt{relativeOf},\ \ \mathtt{Sym}(\mathtt{partnerOf}),\ \ \mathtt{Irr}(\mathtt{partnerOf}),\\
& \mathtt{descendantOf} \sqsubseteq \mathtt{relativeOf},\ \ \mathtt{ancestorOf} \equiv \mathtt{descendantOf}^{-},\\
& \mathtt{childOf} \sqsubseteq \mathtt{descendantOf},\ \ \mathtt{parentOf} \equiv \mathtt{childOf}^{-}\ \}\\
\mathcal{K}'_{\mathcal{S}} = \mathcal{K}_{\mathcal{S}}\ & \cup \{\mathtt{Tra}(\mathtt{descendantOf})\}\\
\mathcal{K}''_{\mathcal{S}} = \mathcal{K}_{\mathcal{S}}\ & \cup \{\mathtt{Irr}(\mathtt{relativeOf}), \mathtt{ASym}(\mathtt{descendantOf})\}\,.
\end{aligned}
$$

Notice that the two constrains $\mathtt{ancestorOf} \equiv \mathtt{descendantOf}^{-}$ and $\mathtt{parentOf} \equiv \mathtt{childOf}^{-}$ violate the definition of $\mathcal{SROIQ}$ syntax reported in [8], which does not report role equivalences as allowed constrains for $\mathcal{SROIQ}$ knowledge bases. However, this issue can be easily circumvented by the technique reported in [8, Footnote 2]. For instance, the axiom $\mathtt{ancestorOf} \equiv \mathtt{descendantOf}^{-}$ can be regarded just as a *macro* which introduces a new name `ancestorOf` for $\mathtt{descendantOf}^{-}$, so that any ontology extending $\mathcal{K}_{\mathcal{S}}$ may be rewritten without this axiom (without affecting reasoning) just by replacing every occurrence of `ancestorOf` with $\mathtt{descendantOf}^{-}$.

A comparison of the three sets of constrains in terms of types of inferences they enable is provided in Table 2. It can be easily verified that $\mathcal{K}_{\mathcal{L}}$ entails $\mathcal{K}_{\mathcal{S}}$, $\mathcal{K}'_{\mathcal{S}}$, and $\mathcal{K}''_{\mathcal{S}}$, and the test results reported in Table 2 confirm that the converse entailments do not hold. The amount of inference types guaranteed by $\mathcal{K}''_{\mathcal{S}}$ overtakes that of $\mathcal{K}'_{\mathcal{S}}$, considering the inference types considered in Table 2. On the other hand, $\mathcal{K}'_{\mathcal{S}}$ enforces transitivity of the `descendantOf` and `ancestorOf` relations. This feature may be crucial in some application domains (for example, to model genealogical trees). In order to evaluate the effective impact of this feature to real-world applications, let us consider a use case in which a user needs to retrieve all those persons with an ancestor of a specified nationality, e.g., Italian. To this purpose, let us introduce the concepts `Italian` and `ItDescendant`, denoting respectively Italians and people with an Italian ancestor, and define their intuitive meaning as follows:

$$\mathcal{K}^{q}_{\mathcal{L}} = \{\ \mathtt{Italian} \sqsubseteq \mathtt{Person},\ \ \mathtt{ItDescendant} \equiv \exists\mathtt{descendantOf}.\mathtt{Italian}\ \}\,.$$

Preliminarily, we observe that both $\mathcal{K}'_{\mathcal{S}}$ and $\mathcal{K}''_{\mathcal{S}}$ can be extended with the constraints in $\mathcal{K}^q_{\mathcal{L}}$ without violating the syntactical limitations imposed by $\mathcal{SROIQ}$. Then, we provide some test cases (see the two leftmost columns of Table 3) which are consequences of $\mathcal{K}_{\mathcal{L}} \cup \mathcal{K}^q_{\mathcal{L}}$, and test the two $\mathcal{SROIQ}$ constraint sets $\mathcal{K}'_{\mathcal{S}}$ and $\mathcal{K}''_{\mathcal{S}}$ against them.

Premises	**Consequence**	$\mathcal{K}'_{\mathcal{S}} \cup \mathcal{K}^q_{\mathcal{L}}$	$\mathcal{K}''_{\mathcal{S}} \cup \mathcal{K}^q_{\mathcal{L}}$	$\mathcal{K}''_{\mathcal{S}} \cup \mathcal{K}^q_{\mathcal{S}}$	$\mathcal{K}_{\mathcal{E}} \cup \mathcal{K}^q_{\mathcal{L}}$
Alice `descendantOf` *Bob* *Bob* $\in$ `Italian`	*Alice* $\in$ `ItDescendant`	Y	Y	Y	Y
Alice `descendantOf` *Bob* *Bob* `descendantOf` *Charlie* *Charlie* $\in$ `Italian`	*Alice* $\in$ `ItDescendant`	Y	**N**	Y	Y
Alice `childOf` *Bob* *Bob* $\in$ `Italian`	*Alice* $\in$ `ItDescendant`	Y	Y	Y	Y
Bob `ancestorOf` *Alice* *Bob* $\in$ `Italian`	*Alice* $\in$ `ItDescendant`	Y	Y	Y	**N**
Bob `ancestorOf` *Alice* *Charlie* `ancestorOf` *Bob* *Charlie* $\in$ `Italian`	*Alice* $\in$ `ItDescendant`	Y	**N**	Y	**N**
Bob `parentOf` *Alice* *Bob* $\in$ `Italian`	*Alice* $\in$ `ItDescendant`	Y	Y	Y	**N**

Table 3. Test results

As expected, $\mathcal{K}'_{\mathcal{S}}$ outperforms $\mathcal{K}''_{\mathcal{S}}$ with respect to the use case under consideration, as $\mathcal{K}'_{\mathcal{S}}$ enforces transitivity of `descendantOf`. However, transitivity can be *emulated* by extending $\mathcal{K}''_{\mathcal{S}}$ with the following constraints:

$$\begin{aligned}\mathcal{K}^q_{\mathcal{S}} = \{ & \texttt{Italian} \sqsubseteq \texttt{Person}, \\ & \texttt{ItDescendant} \equiv \exists\texttt{descendantOf.Italian} \sqcup \exists\texttt{descendantOf.ItDescendant} \} .\end{aligned}$$

The constraint set obtained in this way passes all the test cases reported in Table 3. In addition, having declared the transitivity of `descendantOf` in $\mathcal{K}_{\mathcal{L}}$, $\mathcal{K}_{\mathcal{L}} \cup \mathcal{K}^q_{\mathcal{L}}$ entails $\{\exists\texttt{descendantOf.ItDescendant} \sqsubseteq \exists\texttt{descendantOf.Italian}\}$, and thus the coherence of $\mathcal{K}^q_{\mathcal{S}}$ with the intuitive description $\mathcal{K}_{\mathcal{L}} \cup \mathcal{K}^q_{\mathcal{L}}$ can be easily verified. Then, we can conclude that $\mathcal{K}''_{\mathcal{S}}$ serve our purposes better than $\mathcal{K}'_{\mathcal{S}}$ does. Of course, our investigation considered just a single use case relative to a specific application scenarios. In fact, different use cases have to be developed and studied for different application scenarios, and it cannot be excluded that $\mathcal{K}'_{\mathcal{S}}$, or another constraints set coherent with $\mathcal{K}_{\mathcal{L}}$, performs better than $\mathcal{K}''_{\mathcal{S}}$ when employed in a different context. For example, when the amount of available computational resources (time, space, CPU, etc.) is limited, using a not-so-expressive description logic, but which admits *efficient* reasoning algorithms, may be more appropriate.

6 Defining kinship in $\mathcal{EL}^{++}$

$\mathcal{EL}^{++}$ (presented in [1, 2]) is a description logic which admits polynomial-time decision procedures, in contrast with the $\mathsf{N2ExpTime}$ worst case for $\mathcal{SROIQ}$ knowledge bases (cf. [11]). This make $\mathcal{EL}^{++}$ suitable for applications when the amount of available resources is limited, but at the cost of a quite restricted expressivity. In particular, some key features for representing kinship relationships such as inversion, symmetry, and irreflexivity on roles are not available in $\mathcal{EL}^{++}$. The following is a set of $\mathcal{EL}^{++}$-constrains which aims to capture, as much as allowed by the language, the intuitive meaning of the kinship relationships reported in Section 4:

$$\begin{aligned}\mathcal{K}_{\mathcal{E}} = \{\ & \mathsf{dom}(\texttt{relativeOf}) \sqsubseteq \texttt{Person}, && \mathsf{range}(\texttt{relativeOf}) \sqsubseteq \texttt{Person}, \\ & \texttt{partnerOf} \sqsubseteq \texttt{relativeOf}, && \texttt{descendantOf} \sqsubseteq \texttt{relativeOf}, \\ & \texttt{Tra}(\texttt{descendantOf}), && \texttt{childOf} \sqsubseteq \texttt{descendantOf}, \\ & \texttt{ancestorOf} \sqsubseteq \texttt{relativeOf}, && \texttt{Tra}(\texttt{ancestorOf}), \\ & \texttt{parentOf} \sqsubseteq \texttt{ancestorOf}\ \}\,. \end{aligned}$$

This set of constraints mainly defines the hierarchy of the considered kinship relations, and enforces the transitivity of `ancestorOf` and `descendantOf`. It is easily verifiable that all the constrains in $\mathcal{K}_{\mathcal{E}}$ are entailed by $\mathcal{K}_{\mathcal{L}}$. On the other hand, the test results reported in Table 2 show that, as expected, the inferencing capability of $\mathcal{K}_{\mathcal{E}}$ is substantially lower than that of the $\mathcal{SROIQ}$-constraint sets reported in Section 5. In addition, as two relations cannot be forced to be inverse of one another, also the use case developed to test the impact of transitivity declarations (see Table 3) is just partially fulfilled.

7 Conclusions and future work

We provided a full characterization of some basic kinship relationships using an *ad-hoc* description logic with the aim of encoding the intuitive meaning of these relationships in a precise and formal way. Then, we derived some test cases that can be used to *measure* the inferencing capability of a candidate kinship ontology. Finally, we devised two $\mathcal{SROIQ}$ ontologies and an $\mathcal{EL}^{++}$ ontology, verified their coherence with the intuitive meaning of the defined kinship relationships, and compared their inferencing capabilities. In addition, the three ontologies were tested against a real-world use case.

We considered just basic kinship relationships. Other aspects of kinship have to be explored, for example *gender-specific* relations like `isFatherOf`, or the distinction between *legal* and *biological* relations. Also, further knowledge representation paradigms should be considered for kinship, in particular those based on *rules* (e.g. [9]). In our opinion, devising a standard format for test cases and test results for inferencing capabilities of ontologies may be useful, possibly extending the W3C standard *Evaluation And Reporting Language (EARL)*.[11] As mentioned in Section 4, the first step of our design methodology may involve

[11] `http://www.w3.org/standards/techs/earl#w3c_all`

an undecidable formal language. In this case, coherence checking of candidate ontologies with the provided intuitive description of the knowledge domain under consideration cannot be automatized, but must be performed by a human agent. However, proofs of this kind may be encoded in the language of some proof-verification tool, in order to be automatically verified by third-parties.

References

1. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI*, pages 364–369. Professional Book Center, 2005.
2. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope further. In Kendall Clark and Peter F. Patel-Schneider, editors, *Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, 2008.
3. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
4. Lloyd Bonfield. Seeking connections between kinship and the law in early modern england. *Continuity and Change*, 25(01), 2010.
5. Eric Vitiello Jr Ian Davis. Relationship: A vocabulary for describing relationships between people, 2004. http://purl.org/vocab/relationship.
6. Nicholas Flores-Herr, Stephan Eickeler, Jan Nandzik, Stefan Paal, I. Konya, and Harald Sack. *Contentus – Next Generation Multimedia Library,*, pages 67–88. Springer Verlag, Heidelberg, 2011.
7. Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening Ontologies with Dolce. In *EKAW 2002*, LNCS. Springer, 2002.
8. Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The Even More Irresistible SROIQ. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR2006).* AAAI Press, 2006.
9. Ian Horrocks and Peter F. Patel-Schneider. A proposal for an OWL rules language. In Stuart I. Feldman, Mike Uretsky, Marc Najork, and Craig E. Wills, editors, *WWW*, pages 723–731. ACM, 2004.
10. A.L. Hughes. *Evolution and human kinship.* Oxford University Press, 1988.
11. Yevgeny Kazakov. Sriq and sroiq are harder than shoiq. In Franz Baader, Carsten Lutz, and Boris Motik, editors, *Proceedings of the 21st International Workshop on Description Logics (DL2008), Dresden, Germany, May 13-16, 2008*, volume 353 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2008.
12. Mary Ann Lamanna and Agnes Riedmann. *Marriages & Families: Making Choices in a Diverse Society.* Thomson/Wadsworth, 2006.
13. Yutaka Matsuo, Masahiro Hamasaki, Junichiro Mori, Hideaki Takeda, and Koiti Hasida. Ontological consideration on human relationship vocabulary for foaf. In *1st Workshop on Friend of a Friend, Social Networking and Semantic Web*, 2004.
14. Peter Mika and Aldo Gangemi. Descriptions of social relations. In *1st Workshop on Friend of a Friend, Social Networking and the (Semantic) Web*, September 2004.
15. Robert Stevens and Margaret Stevens. A family history knowledge base using owl 2. In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, *Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions*, volume 432 of *CEUR Workshop Proceedings.* CEUR-WS.org, 2009.