# Automated Reasoning in Temporal *DL-Lite* (Extended Abstract)*


Sabiha Tahrat[1], German Braun[2],
Alessandro Artale[3], Marco Gario[3], and Ana Ozaki[4]

[1] Université de Paris, France sabiha.tahrat@parisdescartes.fr
[2] Universidad Nacional del Comahue, Argentina german.braun@fi.uncoma.edu.ar
[3] Free Univ. of Bolzano, Italy artale@inf.unibz.it, marco.gario@gmail.com
[4] University of Bergen, Norway Ana.Ozaki@uib.no




We investigate the practical feasibility of automated reasoning over temporal *DL-Lite* (*TDL-Lite*) knowledge bases (KBs) [1,15,4,2,3]. By '*TDL-Lite*', we consider here the $T_{FPX}$*DL-Lite*$^{\mathcal{N}}_{bool}$ logic [2], the most expressive decidable language of the *DL-Lite* family combined with Linear Temporal Logic (*LTL*). The key idea is to map a *TDL-Lite* KB—a set of TBox and ABox axioms—into an equisatisfiable *LTL* formula by applying the translation described in [2]. *TDL-Lite* admits both past and future operators interpreted over $\mathbb{Z}$ while *LTL* reasoners often can deal with only future operators interpreted over $\mathbb{N}$. Thus, we present a translation removing past operators that retains *formula satisfiability* (Gabbay [10] showed that past temporal modalities do not add expressive power, and recently Markis [16] presented an algorithm preserving *formula equivalence* where the obtained pure-future formulas have an exponential blow-up in size). Since we are interested in preserving satisfiability, we provide a *linear* in size translation that removes past operators from a *TDL-Lite* knowledge base, thus obtaining an equi-satisfiable pure-future *LTL* formula. The result is stated in the following theorem, which is more generally formulated in terms of *LTL* formulas (where $LTL_P$ denotes *LTL* extended with past operators and interpreted over $\mathbb{Z}$).

**Theorem 1.** *Let $\varphi$ be a $LTL_P$ formula, then, $\varphi$ is satisfiable iff its LTL translation $\varphi^{\mathbb{N}}$ is satisfiable. The size of $\varphi^{\mathbb{N}}$ grows linearly w.r.t $|\varphi|$.*

Since the above translation comes at the cost of increasing the number of propositional variables, we also introduce the simpler logic, called $T^{\mathbb{N}}$*DL-Lite*, that allows for temporal formulas with only future temporal operators, interpreted over $\mathbb{N}$. Using such a weaker language allows us to evaluate the impact of past operators on the runtime efficiency of reasoners when checking for satisfiability.

The complexity of reasoning over *TDL-Lite* KBs is known to be PSPACE-complete [2]. To put these results in practice, we provide a tool, named `crowd-`$\mathcal{ER}_{\mathcal{VT}}$[5], which is a non-trivial extension of `crowd` [5,6]. Our tool allows users to draw *temporal* conceptual schemas and populate them with timestamped instances, which are translated into *TDL-Lite* KBs and, ultimately, into *LTL* formulas that can be checked for satisfiability and entailment using existing off-the-shelf *LTL*



[5] crowd.fi.uncoma.edu.ar/ervt-gui/erd_editor.php

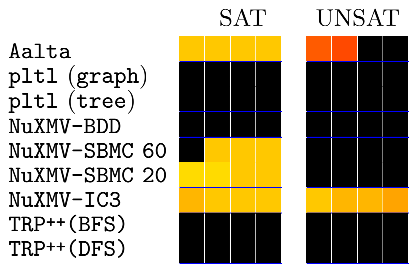


**Fig. 1:** Heat maps with TBox from Ex. 1. Each rectangle represents the runtime in CPU seconds of SAT and UNSAT KBs with different ABox sizes on different *LTL* solvers. The runtimes colours are as follows:
$<$ 0.01 sec; $>$ 0.01 sec, $\leq$ 0.25 sec; $>$ 0.25 sec, $\leq$ 0.50 sec; $>$ 0.50 sec, $\leq$ 1 sec; $>$ 1 sec, $\leq$ 2 sec;
$>$ 2 sec, $\leq$ 4 sec; $>$ 4 sec, $\leq$ 8 sec; $>$ 8 sec, $\leq$ 16 sec; $>$ 16 sec, $\leq$ 32 sec; $>$ 32 sec, $\leq$ 64 sec;
$>$ 64 sec, $\leq$ 125 sec; $>$ 125 sec, $\leq$ 250 sec; $>$ 250 sec, $\leq$ 500 sec; $>$ 500 sec, $\leq$ 1000 sec; T/O, OoM or Fail

reasoners [18,7,12,14,17]. We conduct experiments to evaluate the scalability of reasoners by randomly generating *TDL-Lite* TBoxes. We also devise a toy scenario to evaluate the performance of reasoners with ABoxes of increasing size.

*Toy Scenario Experiment.* In the chosen "toy scenario" a *TDL-Lite* TBox is paired with various ABoxes of different sizes varying from 20 to 50 assertions (distributed over different time points), which may yield either satisfiable (SAT) or unsatisfiable (UNSAT) KBs. The following example illustrates such a scenario with an ABox that is unsatisfiable w.r.t. the given TBox.

*Example 1.* Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a *TDL-Lite* TBox expressing that, at each point in time, a person has a unique $\mathsf{Name}$ which is also global (i.e., does not change over time), but the ABox $\mathcal{A}$ (0 and 1 are timestamps denoting when the assertions hold) violates the fact that $p_1$'s name is functional and global:

$$\mathcal{T} = \{\mathsf{Person} \sqsubseteq \geq 1\ \mathsf{Name}, \mathsf{Person} \sqsubseteq \neg \geq 2\ \mathsf{Name}\}$$
$$\mathcal{A} = \{\mathsf{Person}(p_1, 0), \mathsf{Name}(p_1, n_1, 0), \mathsf{Name}(p_1, n_2, 1)\}$$

Depending on the different sizes of the tested ABoxes, the number of propositional variables in the resulting *LTL* formulas translating the *TDL-Lite* KBs of the "toy scenario" ranges from 180 to 2336 variables. The results shown in Fig. 1 in the form of 'heat maps' [13] represent the runtime of the KB satisfiability checking for increasing ABox sizes (in columns) and different solvers (in lines). Solvers had better performances over SAT instances compared to UNSAT ones, except `TRP++` and `pltl`, which fail to scale already over small ABoxes. Moreover, `NuXMV-SBMC` fails regardless the size of the model in UNSAT cases. Overall, the best options for SAT and UNSAT cases were `NuXMV` with BMC and IC3, respectively. `Aalta` performs well but only when the *LTL* input formula does not exceed 1200 propositional variables.

*Randomly Generated TBoxes.* In a second experiment, we investigate the scalability of the reasoners over synthetic TBoxes (no ABoxes in this experiment) by extending the random algorithm proposed for *LTL* [8]. We benchmarked our tool against TBoxes (mostly SAT) generated randomly according to the

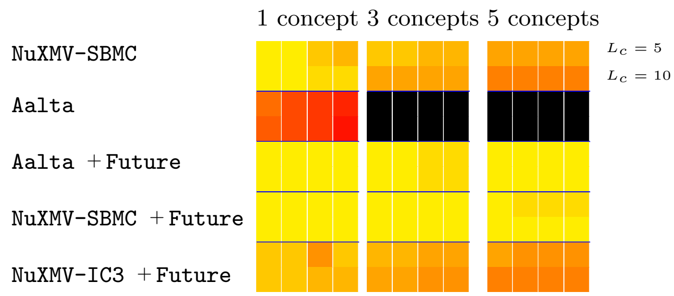


**Fig. 2:** Heat map of the runtimes on randomly generated TBoxes (colors as in Fig. 1).

following settings: (*i*) the average-behaviour analysis which covers *TDL-Lite* TBoxes in a uniform way (see the full paper [19] for more details); and (*ii*) the temporal-behaviour analysis which increases the chance of generating TBoxes with temporal operators and global roles. For the temporal-behaviour analysis (see Fig. 2), we create batches of 20 random TBoxes with the following parameters: $N = 1, 3, 5$ (number of concept names), $Q = 5$ (maximum cardinality), $L_t = 10$ (number of TBox axioms). $L_c = 5, 10$ (length of concept expressions), and by increasing the probability $P_t$ of generating temporal operators and the probability $P_g$ of generating global roles. Fig. 2 shows the runtime for different values of $N$ against *LTL* solvers that performed well in the toy scenario, namely, `NuXMV-SBMC`, `Aalta`, and `NuXMV-IC3`. For each value of $N$, the first two columns consider $P_g = 0.7$ and two values for $P_t = 0.1, 0.9$, while the last two columns consider $P_g = 0.9$ and again $P_t = 0.1, 0.9$. For each solver, the first line is the case where $L_c = 5$, while the second has $L_c = 10$. Due to the increase in the number of variables when removing the past operators, as expected solvers perform better on TBoxes expressed with only future operators (i.e., on $T^{\mathbb{N}}$*DL-Lite* TBoxes) as shown on Lines 3, 4 and 5, with the BMC option performing better than IC3. Increasing $P_t$ does not significantly impact the runtime values. This indicates that *LTL* solvers are less affected by the number of temporal operators than by the number of variables in the formula.
For more detail see the full version of the paper [19].

**Conclusions.** This work investigate the scalability and robustness of *LTL* solvers while checking *TDL-Lite* KBs for satisfiability. Two major culprits in the runtime of solvers are the size of the ABox and the presence of past operators. The increase in the number of propositional variables when removing past opertors penalizes the runtime of the solvers. Concerning ABoxes, the preliminary results show that a brute force approach makes reasoning in the presence of ABoxes almost unfeasable. As a future work, we plan to investigate reasoners able to scale in the presence of ABoxes. We will experiment with first-order temporal logic solvers to avoid the step of grounding the translation, making the number of propositional variables of the resulting *LTL* encoding not manageable. Furthermore, we plan to extend to the temporal case the existing ABox abstraction approaches which are successfully applied over OWL ontologies [9,11].

## References


1. A. Artale and E. Franconi. Temporal description logics. In *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 375–388. Elsevier, 2005.
2. A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. A cookbook for temporal conceptual data modelling with description logics. *ACM Trans. Comput. Log.*, 15(3):25:1–25:50, 2014.
3. A. Artale, A. Mazzullo, and A. Ozaki. Do you need infinite time? In *IJCAI*, 2019.
4. F. Baader, S. Ghilardi, and C. Lutz. LTL over description logic axioms. *ACM Trans. Comput. Log.*, 13(3), 2012.
5. G. A. Braun, L. A. Cecchi, and P. R. Fillottrani. Taking advantages of automated reasoning in visual ontology engineering environments. In *Proc. of the Joint Ontology Workshops 2019 Episode V: The Styrian Autumn of Ontology (JOWO'19)*, 2019.
6. G. A. Braun, C. Gimenez, L. A. Cecchi, and P. R. Fillottrani. `crowd`: A visual tool for involving stakeholders into ontology engineering tasks. *Künstl Intell (2020)*, 2020.
7. R. Cavada, A. Cimatti, M. Dorigatti, A. Griggio, A. Mariotti, A. Micheli, S. Mover, M. Roveri, and S. Tonetta. The NuXMV symbolic model checker. In *26th Int. Conf. on Computer Aided Verification, (CAV)*, volume 8559 of *Lecture Notes in Computer Science*, pages 334–342. Springer, 2014.
8. M. Daniele, F. Giunchiglia, and M. Y. Vardi. Improved automata generation for linear temporal logic. In *CAV*, 1999.
9. A. Fokoue, F. Meneguzzi, M. Sensoy, and J. Z. Pan. Querying linked ontological data through distributed summarization. In J. Hoffmann and B. Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 2012.
10. D. M. Gabbay, A. Pnueli, S. Shelah, and J. Stavi. On the temporal analysis of fairness. In *Conference Record of the 7th ACM Symposium on Principles of Programming Languages (POPL'80)*, page 163–173. ACM Press, 1980.
11. B. Glimm, Y. Kazakov, and T. Tran. Scalable reasoning by abstraction beyond DL-Lite. In M. Ortiz and S. Schlobach, editors, *Web Reasoning and Rule Systems (RR) - Proceedings of the 10th International Conference*, volume 9898 of *Lecture Notes in Computer Science*, pages 77–93. Springer, 2016.
12. U. Hustadt and B. Konev. TRP++2.0: A temporal resolution prover. In *Automated Deduction - CADE-19, 19th International Conference on Automated Deduction, Proceedings*, pages 274–278, 2003.
13. U. Hustadt, A. Ozaki, and C. Dixon. Theorem proving for metric temporal logic over the naturals. In *CADE*, pages 326–343, 2017.
14. J. Li, S. Zhu, G. Pu, L. Zhang, and M. Y. Vardi. Sat-based explicit LTL reasoning and its application to satisfiability checking. *Formal Methods Syst. Des.*, 54(2):164–190, 2019.
15. C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *Proc. of the 15th Int. Symposium on Temporal Representation and Reasoning, TIME'08*, pages 3–14. IEEE Computer Society, 2008.
16. N. Markey. Temporal logic with past is exponentially more succinct. *Bulletin of the EATCS*, 79:122–128, 2003.
17. V. Schuppan. Extracting unsatisfiable cores for LTL via temporal resolution. In C. Sánchez, K. B. Venable, and E. Zimányi, editors, *2013 20th International Symposium on Temporal Representation and Reasoning, Pensacola, FL, USA, September 26-28, 2013*, pages 54–61. IEEE Computer Society, 2013.

18. S. Schwendimann. A new one-pass tableau calculus for PLTL. In H. de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 277–291. Springer Berlin Heidelberg, 1998.
19. S. Tahrat, G. Braun, A. Artale, M. Gario, and A. Ozaki. Automated reasoning in temporal DLite. *arXiv*, To be done, 2020.