Temporal ABox Cleaning in *TDL-Lite* [⋆] [⋆⋆]

Sabiha Tahrat[1], Salima Benbernou[2], and Mourad Ouziri[3]

[1] Université de Paris, France sa.tahrat@gmail.com
[2] Université de Paris, France salima.benbernou@u-paris.fr
[3] Université de Paris, France mourad.ouziri@u-paris.fr

We propose an ABox repair approach for temporal *DL-Lite* (*TDL-Lite*) Knowledge Bases (KBs) [2,3] considering the data part of the KB as the source of inconsistency over time. As a *TDL-Lite* language, we here consider the $T^{\mathbb{N}}$*DL-Lite* fragment [11] allowing LTL future temporal operators [9] interpreted over $\mathbb{N}$ and we restrict ourselves to specifying functionality on roles and using the two future operators $\Diamond_F$ and $\Box_F$. Our goal is twofold: 1) detect data inconsistencies and 2) propose a data temporal repair. For the inconsistency detection, we propose an equi-satisfiable reduction approach from *TDL-Lite* to *DL-Lite* which allows using DL reasoners that can return the precise set of inconsistent assertions. Thereafter, we propose a method for computing the best temporal repair based on the allowed rigid predicates and the time order of assertions.
Let $\mathsf{N_C}, \mathsf{N_I}$ and $\mathsf{N_R}$ be countable sets of *concept*, *individual names* and *roles* respectively. $\mathsf{N_R}$ is the union $\mathsf{N_G} \cup \mathsf{N_L}$ where $\mathsf{N_G}$ and $\mathsf{N_L}$ are countable and disjoint sets of *global* and *local role names*, respectively. *TDL-Lite basic concepts* B, *concepts* C,*(temporal) concepts* D, and *roles* R, are formed according to the following grammar:

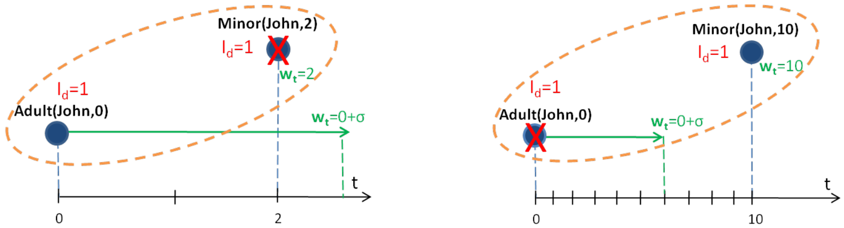
$$R ::= L \mid L^- \mid G \mid G^-, \qquad B ::= \bot \mid \top \mid A \mid \exists R,$$
$$C ::= B \mid \neg C \mid C_1 \sqcap C_2, \qquad D ::= C \mid \Diamond_F D \mid \Box_F D \mid \neg D \mid D_1 \sqcap D_2.$$

where $L \in \mathsf{N_L}$, $G \in \mathsf{N_G}$, $A \in \mathsf{N_C}$. We call *disjointness*, inclusions of the form $C \sqcap D \sqsubseteq \bot$. We also add the ability to specify *functional* roles ($funct\ R$) and *rigid predicates* which are elements from the set of *rigid concepts* $\mathsf{N_{RC}} \subseteq \mathsf{N_C}$ or of rigid roles $\mathsf{N_G}$ and for all $X \in \mathsf{N_{RC}} \cup \mathsf{N_G}$ and $i, j \in \mathbb{N}$, $X^{\mathcal{I}_i} = X^{\mathcal{I}_j}$ (denoted simply by $X^{\mathcal{I}}$).

A *TDL-Lite* KB is a set of a *TBox* and an *ABox* axioms. The TBox is a finite set of *general concept inclusions (GCI)* of the form $C \sqsubseteq D$ and the ABox is a set of *concept assertions* of the form $\bigcirc^n A(a)$ or $\bigcirc^n \neg A(a)$, or *role assertions* of the form $\bigcirc^n R(a, b)$ or $\bigcirc^n \neg R(a, b)$, where $a, b \in \mathsf{N_I}$, and $n \in \mathbb{N}$. The *DL-Lite* translation function "tr" is defined at each time point $i \in [0, m]$, where m is the maximum nesting depth of temporal operators in the TBox, as follows:

$$\begin{aligned}
&tr(C,i,m) = C_i, && tr(\neg C,i,m) = \neg tr(C,i,m)\\
&tr(R,i,m) = R_i, \quad tr(\top,i,m) = \top, && tr(\Box_F D,i,m) = \sqcap_i^m\ tr(D,i,m)\\
&tr(C \sqcap D,i,m) = tr(C,i,m) \sqcap tr(D,i,m), && tr(\Diamond_F D,i,m) = \sqcup_i^m tr(D,i,m)
\end{aligned}$$

[⋆] Extended version published as arXiv:2108.12149



(a) a case where the recent assertion is removed **(b)** a case where the rigid assertion is removed

Fig. 1: Best temporal repair in TKBs based on removing one inconsistent assertion with the highest value of the Inconsistency degree I_d (number of conflicts) from each hyperedge. If two vertices have the same I_d we remove the one with the lowest value of the temporal weight w_t.

The translation creates fresh concepts C_i and roles R_i in the *DL-Lite* KB denoting respectively the interpretation $\mathcal{I}_i$ of C and R at time point i. Now, the translation $\mathcal{T}^\dagger$ of a TBox and $\mathcal{A}^\dagger$ of an ABox is as follows:

$$\mathcal{T}^\dagger = \{ \bigwedge_{C \sqsubseteq D \in \mathcal{T}} \bigwedge_{i=0}^{m} (tr(C,i,m) \sqsubseteq tr(D,i,m)), \tag{1}$$

$$\bigwedge_{C \in \mathsf{N_{RC}}} \bigwedge_{i=0}^{m} \big(tr(C,i,m) \sqsubseteq \sqcap_i^m \; tr(C,i,m)\big), \tag{2}$$

$$\bigwedge_{R \in \mathsf{N_G}} \bigwedge_{i=0}^{m} \big(tr(R,i,m) \sqsubseteq \sqcap_i^m \; tr(R,i,m)\big), \tag{3}$$

$$\bigwedge_{Funct(R)} \bigwedge_{i=0}^{m} (Funct(R_i)) \quad \} \tag{4}$$

$$\mathcal{A}^\dagger = \bigwedge_{\bigcirc^n A(a) \in \mathcal{A}} A_n(a) \wedge \bigwedge_{\bigcirc^n R(a,b) \in \mathcal{A}} R_n(a,b). \tag{5}$$

Inconsistency Detection. We start by checking the satisfiability of the translated TBox $\mathcal{T}^\dagger$ and then we check the consistency of $\mathcal{A}^\dagger$ with $\mathcal{T}^\dagger$ using the reasoner Pellet[4]. If $\mathcal{A}^\dagger$ is inconsistent, the explanation support of Pellet points in each explanation a minimal subset $I_m = \{a_i, a_j, ...\}$ of inconsistent assertions in conflict with a constraint c_k such as a disjointness inclusion, a rigid predicate or a functional role. We report those explanations in an intuitive way by building an *inconsistency graph IG* which is similar to the *conflict-hypergraph* used to represent constraint violations in databases [7] or inconsistency-tolerant semantics for DL [5,6]. For more details, see the full version of the paper [10].

Definition 1. *An inconsistency graph for a KB* $\mathcal{K}$ *is denoted by* $IG(\mathcal{K}) = (V, E)$ *where vertices* $V = \{a_1, ..., a_n \in \bigcup_m I_m\}$ *are inconsistent assertions and* $E = \{e_1, ..., e_m\}$ *is a set of hyperedges where* $e_i = I_i ... e_m = I_m$.

[4] `http://pellet.owldl.com/`

Maximal Repair Semantics. An extreme repair solution would be to simply remove all the detected inconsistent assertions from the ABox. This would certainly not meet the principal of minimal set of changes that restore consistency. In other words, a maximal repair $\mathcal{A}'$ should be a maximal consistent subset of $\mathcal{A}$, obtained by removing a minimal set of inconsistent assertions (aka the Minimal Unsatisfiable Set MUS). In the inconsistency graph, this corresponds to remove only a single vertex from each hyperedge given that hyperedges are minimal subsets of inconsistent assertions. Moreover, the maximal repair is obtained by removing vertices in the intersection of hyperedges i.e. assertions who are involved in more than one conflict (violated constraint). In practice, the maximal repair algorithm removes from each hyperedge the vertex with the highest *Inconsistency degree* I_d. Formally, we define an inconsistency degree $I_d(a_i)$ of a vertex a_i as the number of hyperedges in which a_i belongs. When vertices have the same I_d, one is randomly removed. Hence, this may result in several possible MUSs for the same IG which are all minimal and then make the repair maximal.

Best Temporal Repair Semantics. Clearly, not just any random maximal repair is useful or interesting in the temporal setting. For instance, repairs that remove mostly recent assertions might be unwanted. Our aim is to guide the repair algorithm when removing assertions with the same I_d, by chosing to remove assertions having the lowest *temporal weight*. To do so, we assign a temporal weight w_t for each inconsistent assertion a_i associated with the timestamp t_i as follows:

$$w_t(a_i, t_i) = \begin{cases} t_i + \sigma \text{ if } a_i \in \mathsf{N_{RC}} \cup \mathsf{N_G} \\ t_i \qquad \textit{otherwise} \end{cases}$$

where σ is a predicate lifespan. If a_i is not an instance of a rigid predicate, $\sigma = 0$. The intuition behind using σ for each rigid concept $C_i \in \mathsf{N_{RC}}$ and global role $G_i \in \mathsf{N_G}$ is to set a duration after which its instances are weakened.

Figure 1 shows that the maximal repair in cases (1a) and (1b) could be the same because they share the same I_d. However, it is easy to see on a timeline that it is better to remove $\mathsf{Minor(John, 2)}$ in (1a) and $\mathsf{Adult(John, 0)}$ in (1b). The notion of temporal weight is intended to capture situations where a maximal repair is temporally better than another. Moreover, σ could be fixed using a data driven approach or a guided user approach.

Conclusions. This work is a first exploration of repairing the ABox w.r.t a TBox defined over Temporal *DL-Lite*. The temporal language considered so far is the $T^{\mathbb{N}}$*DL-Lite* with which we can express and check several useful types of temporal constraints, such as defining temporal concepts in GCIs and rigid predicates. More generally, we plan to investigate in practice repairing KBs based on multiple temporal DL-Lite logics which are First Order rewritable [1]. Also, it would be interesting to implement a repair framework and evaluate the scalability properties of our approach based on the temporal best repair semantics. Finally, in the same spirit of the proposed predicate lifespan σ, we are considering for the ABox repair adding metric operators to the *TDL-Lite* language [4, 8] that augment LTL temporal operators with time interval.

References

1. A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. Temporal OBDA with LTL and dl-lite. In *Informal Proceedings of the 27th International Workshop on Description Logics (DL 2014)*, CEUR W-S, pages 21–32, 2014.
2. A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. Temporal ontology-mediated querying: A survey. In *24th International Symposium on Temporal Representation and Reasoning (TIME17)*, Mons, Belgium, October 2017.
3. A. Artale, R. Kontchakov, A. Kovtunova, F. Wolter, and M. Zakharyaschev. First-order rewritability of temporal ontology-mediated queries. In *Proc. of the 24th Int. Joint Conference on Artificial Intelligence (IJCAI-15)*, Buenos Aires, Argentina, 25-31 July, 2015. AAAI Press.
4. F. Baader, S. Borgwardt, P. Koopmann, A. Ozaki, and V. Thost. Metric temporal description logics with interval-rigid names. *ACM Trans. Comput. Log.*, 21(4):30:1–30:46, 2020.
5. S. Benbernou and M. Ouziri. Enhancing data quality by cleaning inconsistent big RDF data. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, pages 74–79, 2017.
6. M. Bienvenu and C. Bourgaux. Querying and repairing inconsistent prioritized knowledge bases: Complexity analysis and links with abstract argumentation. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, pages 141–151, 2020.
7. J. Chomicki and J. Marcinkowski. Minimal-change integrity maintenance using tuple deletions. *Inf. Comput.*, 197(1-2):90–121, 2005.
8. V. Gutiérrez-Basulto, J. C. Jung, and A. Ozaki. On metric temporal description logics. In *ECAI*, pages 837–845, 2016.
9. A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, pages 46–57. IEEE Computer Society, 1977.
10. S. Tahrat, S. Benbernou, and M. Ouziri. Cleaning inconsistent data in temporal DL-Lite under best repair semantics. *arXiv*, arXiv:2108.12149, 2021.
11. S. Tahrat, G. A. Braun, A. Artale, M. Gario, and A. Ozaki. Automated reasoning in temporal DL-Lite (extended abstract). In *Proceedings of the 33rd International Workshop on Description Logics (DL 2020)*, volume 2663 of *CEUR Workshop Proceedings*, 2020.