NeoMaPy: A Parametric Framework for Reasoning with MAP Inference on Temporal Markov Logic Networks

Victor David
Dipartimento di Matematica e Informatica, University of Perugia
Italy
victor.david@unipg.it

Raphaël Fournier-S'niehotta
Conservatoire National des Arts et Métiers
Paris, France
fournier@cnam.fr

Nicolas Travers
Léonard de Vinci Pôle Universitaire, Research Center
Paris, France
nicolas.travers@devinci.fr

ABSTRACT

Reasoning on inconsistent and uncertain data is challenging, especially for Knowledge-Graphs (KG) to abide temporal consistency. Our goal is to enhance inference with more general time interval semantics that specify their validity, as regularly found in historical sciences. We propose a new Temporal Markov Logic Networks (TMLN) model which extends the Markov Logic Networks (MLN) model with uncertain temporal facts and rules. Total and partial temporal (in)consistency relations between sets of temporal formulae are examined. We then propose a new Temporal Parametric Semantics (TPS) which allows combining several sub-functions leading to different assessment strategies. Finally, we present the NeoMaPy tool, to compute the MAP inference on MLNs and TMLNs with several TPS. We compare our performances with state-of-the-art inference tools and exhibit faster and higher quality results.

CCS CONCEPTS

• **Computing methodologies → Semantic networks**; **Temporal reasoning**.

KEYWORDS

Temporal Markov Logic Networks, MAP Inference, Knowledge Graphs

ACM Reference Format:
Victor David, Raphaël Fournier-S'niehotta, and Nicolas Travers. 2023. NeoMaPy: A Parametric Framework for Reasoning with MAP Inference on Temporal Markov Logic Networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614757

1 INTRODUCTION

Reasoning on large data sets to obtain pieces of information is an open challenge [4, 14, 18, 22]. Most approaches model information with *Knowledge Graphs* (KGs) [13], and rely on *Ontologies* [23], *Machine Learning* [30] or *Neural Networks* [19] representations. Then, *Description Logic* [17] and *Temporal Logic* [27] may be used to verify domain rules. Historians, for example, frequently reason on sets of facts. Any fact is uncertain, and several facts may contradict one another (generating *conflicts*). Temporal information is crucial: outside of a temporal interval, a fact becomes false. Weighting facts with temporal and uncertainty information, an historian may resolve conflicts and find consistent sets of facts forming new hypotheses.

Markov Logic Networks (MLNs) [24] combine Markov networks and First Order Logic, by attaching weights to logic formulae.

It can be seen as a formalism that extends First Order Logic to allow formulae that can be violated with some penalty. MLNs allow reasoning on the set of possible worlds of a knowledge base by looking for *the most probable world*, according to a semantic computing the probability of the worlds, this process is called *Maximum A-Posteriori* (MAP) inference [20, 21, 25, 28]. Several MLNs extensions have been devised to work on different types of data [7, 26, 29], and one focused on reasoning on *Uncertain Temporal Knowledge Graphs* (UTKG) with specific temporal inference rules [6]. However, the state of the art integrating temporal information into MLN is conceptually insufficient, and computing the MAP inference also requires to build possible worlds iteratively with logic programming (ILP) on conflictual facts to find the one that satistifies the more without conflicts [8]. By parallelizing and estimating aggregations, they optimize the ILP formulation, but solving this ILP is highly dependent on the number of facts that violate constraints.

In this paper, we introduce an extension of MLNs called Temporal Markov Logic Networks (TMLN), built on (a temporal) many-sorted logic, for reasoning on both time validity and uncertainty [11]. We extend the notion of uncertainty to rules, and present an adapted reasoning to deal with both uncertain facts and rules. We define a new temporal semantics and a temporal extension to MAP inference [25]. This MAP inference produces *instantiations* (also called worlds), *i.e.,* extended sets of facts maximising the score *w.r.t.* a temporal semantics. The proposed temporal semantics is parametric: it allows combining several sub-functions for various consistency validations. Our completely different and *optimistic* approach (*i.e.,* based on the assumption that conflicts of operations on a database are rare) to compute MAP inferences, NeoMaPy, relies on building compatible worlds based on a conflict graphs, instead of iteratively building valid worlds (ILP solving).It allows computing efficiently the MAP thanks to a heuristic, and interacting with results for explaining facts choices. Finally, we present a complete implementation of NeoMaPy, built with Neo4j and the heuristic MaPy as a Python script which computes the parametric MAP inference.

Paper organisation. Section 2 exposes relevant background information on Many-Sorted First-Order logic and its reasoning. In

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom.

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3614757

Section 3, we introduce our original TMLN representation and its semantics required for MAP inference (Section 4). Then, we present our new approach to MAP inference computation in Section 5 and experiment it in Section 6, before concluding the paper.

2 BACKGROUND

In a seminal work [6], Chekol et al. formalise the *Uncertain Temporal Knowledge Graphs* (UTKG) approach, which integrates both time and uncertainty in KGs to reach a *certain world maximisation*. However, they do not take into account the possibility to have uncertain rules. We enlarge their vision by putting *time* at the heart of reasoning. We formalise the notion of *temporal uncertainty*, by combining certain and uncertain formulae, allowing for easier manipulations and better analyses.

2.1 Many-Sorted First Order Logic

We start by presenting the Many-Sorted first-Order Logic. Lowercase (resp. uppercase) Greek letters like ϕ, ψ (resp. Φ, Ψ) denote formulae (resp. sets of formulae).

Definition 1 (Many-Sorted FOL). *Let* $\mathbf{So} = \{s_1, \dots, s_n\}$ *be a set of sorts. A* Many-Sorted first-Order Logic $\texttt{MS-FOL}$*, is a set of formulae built up by induction from: a set* $\mathbf{C} = \{a_1, \dots, a_l\}$ *of constants, a set* $\mathbf{V} = \{x^s, y^s, z^s, \dots \mid s \in \mathbf{So}\}$ *of variables, a set* $\mathbf{P} = \{P_1, \dots, P_m\}$ *of predicates, a function* $\texttt{ar} : \mathbf{P} \longrightarrow \mathbb{N}$ *which tells the arity of any predicate, a function* $\texttt{sort}$ *s.t. for* $P \in \mathbf{P}$, $\texttt{sort}(P) \in \mathbf{So}^{\texttt{ar}(P)}$*, and for* $c \in \mathbf{C}$, $\texttt{sort}(c) \in \mathbf{So}$*, the usual connectives* ($\neg, \vee, \wedge, \rightarrow, \leftrightarrow$)*, Boolean constants* ($\top$ *and* $\bot$) *and quantifier symbols* ($\forall, \exists$)*. A* ground formula *is a formula without any variable.*

Example 1. *For instance let* $\mathbf{So} = \{s_1, s_2\}$*, let* $P_1 \in \mathbf{P}$ *such that* $\texttt{sort}(P_1) = s_2 \times s_1 \times s_1$*, let* $a_1, a_2, t_1, t_2 \in \mathbf{C}$ *such that* $\texttt{sort}(a_1) = \texttt{sort}(a_2) = s_2$, $\texttt{sort}(t_1) = \texttt{sort}(t_2) = s_1$ *and let* $x^{s_2} \in \mathbf{V}$*. We can then build the following* $\texttt{MS-FOL}$ *formulae:* $P_1(a_1, t_1, t_2)$, $\forall x^{s_2} P_1(x^{s_2}, t_1, t_2)$*. However,* $P_1(t_1, t_2, a_1)$ *or* $\forall x^{s_2} P_1(a_1, a_2, x^{s_2})$ *cannot be built because they do not respect the sorts.*

$\texttt{MS-FOL}$ formulae are evaluated via a notion of *structure* called *n*-sorted structures [15]. Classical first-order logic formulae are captured as 1-sorted structures.

Definition 2 (Structure). *A* n-sorted structure *is* $\mathbf{St} = (\{D_1, \dots, D_n\}, \{R_1, \dots, R_m\}, \{c_1, \dots, c_l\})$*, where* $D_1, \dots, D_n$ *are the (non-empty) domains,* $R_1, \dots, R_m$ *are relations between domains' elements, and* $c_1, \dots, c_l$ *are distinct constants in the domains.*

Our running example is presented in Example 2. Each sentence gathers biographical elements about a French philosopher from the 14th century, *Nicole Oresme*.

Example 2. *Nicole Oresme was a person and a philosopher born in the Middle Ages between 1320 and 1382. Nicole Oresme may have attended the College of Navarre around 1340-1354 and more likely around 1355-1360. Nicole Oresme possibly did not attend the College of Navarre around 1353-1370. Sometimes, a person who lived in the Middle Ages and studied at the College of Navarre came from a peasant family. Usually, a philosopher born in the Middle Ages did not come from a peasant family.*

Though without uncertainty, we may then define a suitable structure in $\texttt{MS-FOL}$.

Example 3. *An example of structure associated with the* $\texttt{MS-FOL}$ *from Example 2 is* $\mathbf{St}_{hist} = (\{Time, Concept\}, \{Person, Philosopher, LivePeriod, PeasantFamily, Studied\}, \{t_{min}, 1300, 1301, 1302, \dots, 1400, t_{max}, NO, MA, CoN\})$*, in which:*
– $Time$ is the set of time points, corresponding to the sort s_1 and $Concept$ is the set of all non-temporal objects, corresponding to the sort s_2,
– $Person, Philosopher, LivePeriod$, etc. are the predicate symbols' relations (e.g., $Person \subseteq Concept \times Time \times Time$ indicates which elements are a person).
– t_{min}, 1300, 1301, ..., 1400, t_{max} are elements of the domain $Time$ associated with the sort s_1, while NO (Nicolas Oresme), MA (Middle Ages) and CoN (College of Navarre) are elements of the domain $Concept$ associated with the sort s_2.

2.2 MS-FOL Reasoning

Now, we define $\texttt{MS-FOL}$ formulae for *interpretation*.

Definition 3 (Interpretation). *An* interpretation $\mathbf{I_{St}}$ *over a structure* $\mathbf{St}$ *assigns to elements of the* $\texttt{MS-FOL}$ *vocabulary some values in the structure* $\mathbf{St}$*. Formally,*
– $\mathbf{I_{St}}(s_i) = D_i$*, for* $i \in \{1, \dots, n\}$ *(each sort symbol is assigned to a domain),*
– $\mathbf{I_{St}}(P_i) = R_i$*, for* $i \in \{1, \dots, m\}$ *(each predicate symbol is assigned to a relation),*
– $\mathbf{I_{St}}(a_i) = c_i$*, for* $i \in \{1, \dots, l\}$ *(each constant symbol is assigned to a value).*
Then, satisfying formulae is recursively defined by:
– $\mathbf{I_{St}} \models P_i(a_1, \dots, a_k)$ *iff* $(\mathbf{I_{St}}(a_1), \dots, \mathbf{I_{St}}(a_k)) \in R_i$,
– $\mathbf{I_{St}} \models \exists x^{s_i} \phi$ *iff* $\mathbf{I}_{\mathbf{St}, x^{s_i} \leftarrow v} \models \phi$ *for some* $v \in D_i$,
– $\mathbf{I_{St}} \models \forall x^{s_i} \phi$ *iff* $\mathbf{I}_{\mathbf{St}, x^{s_i} \leftarrow v} \models \phi$ *for each* $v \in D_i$,
– $\mathbf{I_{St}} \models \phi \wedge \psi$ *iff* $\mathbf{I_{St}} \models \phi$ *and* $\mathbf{I_{St}} \models \psi$,
– $\mathbf{I_{St}} \models \phi \vee \psi$ *iff* $\mathbf{I_{St}} \models \phi$ *or* $\mathbf{I_{St}} \models \psi$,
– $\mathbf{I_{St}} \models \neg\phi$ *iff* $\mathbf{I_{St}} \not\models \phi$,
where $\mathbf{I}_{\mathbf{St}, x^{s_i} \leftarrow v}$ *is a modified version of* $\mathbf{I_{St}}$ *s.t. the variable* x^{s_i} *is replaced by a value* v *in the domain* D_i *corresponding to the sort symbol* s_i*. finally, if* Φ *is a set of formulae, then* $\mathbf{I_{St}} \models \Phi$ *iff* $\mathbf{I_{St}} \models \phi$ *for each* $\phi \in \Phi$.

Definition 3 does not target the satisfaction of implications and equivalences, while they can be defined by: $(\phi \rightarrow \psi) \equiv (\neg\phi \vee \psi)$, and $(\phi \leftrightarrow \psi) \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$. For instance, the set of interpretations of the formula $P(a) \vee P(b)$ is equal to $\{\{P(a)\}, \{P(b)\}, \{P(a), P(b)\}\}$ and for $P(a) \wedge P(b)$ is $\{\{P(a), P(b)\}\}$.

With structures and interpretations on TMLNs, we now define the consequence relations and logical consequences over $\texttt{MS-FOL}$.

Definition 4 (Consequence Relation). *Let* ϕ *and* ψ *be two* $\texttt{MS-FOL}$ *formulae. We say that* ψ is a consequence of ϕ*, denoted by* $\phi \vdash \psi$*, if for any structure* $\mathbf{St}$*, and any interpretation* $\mathbf{I_{St}}$ *over* $\mathbf{St}$, $\mathbf{I_{St}} \models \phi$ *implies* $\mathbf{I_{St}} \models \psi$.

Definition 5 (Logical Consequences - $\texttt{Cn}$). *Let* $\phi \in \texttt{MS-FOL}$*. The function* $\texttt{Cn}(\phi)$ *is the set of all logical consequences of* ϕ*, i.e.,* $\texttt{Cn}(\phi) = \{\psi \in \texttt{MS-FOL} \mid \phi \vdash \psi\}$.

$\texttt{Cn}$ returns an infinite set of formulae, but for clarity we consider only one formula per equivalent class and only the predicates and constants appearing in the original formulae. So that $\texttt{Cn}(P(a) \vee$

$P(b)) = \{P(a) \vee P(b)\}$ and $\mathtt{Cn}(P(a) \wedge P(b)) = \{P(a), P(b), P(a) \wedge P(b), P(a) \vee P(b)\}$.

Since we are working with Temporal Formulae (TF), we extend inferences according to predicates' temporal interval such that for each formula, each predicate, we can also infer all possible temporal subsets. For example, $\mathtt{Cn}(P(a,t_1,t_2) \wedge P(b,t_2,t_2)) = \{P(a,t_1,t_1), P(a,t_1,t_2), P(a,t_2,t_2), P(b,t_2,t_2), P(a,t_1,t_1) \wedge P(b,t_2,t_2), \ldots, P(a,t_2,t_2) \vee P(b,t_2,t_2)\}$. In the rest of the article, if not specified we consider the temporally extended version of Cn.

3 TEMPORAL AND UNCERTAIN KNOWLEDGE REPRESENTATION

Markov Logic Networks (MLNs) combine Markov Networks and first-Order Logic (FOL) by attaching weights to first-order formulae and treating them as feature templates for Markov Networks [24]. We extend this framework to temporal information by resorting to Many-Sorted first-Order Logic (MS-FOL).

3.1 Temporal Markov Logic Networks

We start with the Temporal Many-Sorted first-Order Logic TF-FOL consisting of *Temporal Formulae*, *i.e.,* combined formulae and temporal predicates from a temporal domain.

Definition 6 (Temporal Many-Sorted FOL). *A* TF-FOL *evaluated by a structure* **St** *is a constrained* MS-FOL *where* $|\mathbf{So}| \geq 2$*, for any interpretation* $\mathbf{I_{St}}(s_1) = Time$*, any predicate* $P_i \in$ TF-FOL *has* $\mathtt{ar}(P_i) \geq 3$ *with the sort of the last two parameters belonging to* s_1 *and* t_{min} *and* t_{max} *are time constants indicating the minimum and maximum time points for any pre-order between the time constants.*

Using this constrained MS-FOL accompanied with a temporal domain ($Time$) and temporal predicates (the last two parameters indicate the validity temporal bounds), we may represent temporal facts and rules. Finally, Temporal Markov Logic Networks (TMLN) extend TF-FOL (resp. MLN) by associating a degree of certainty to each formula (resp. by adding a temporal validity to the predicates).

Definition 7 (TMLN). *A Temporal Markov Logic Network* $\mathbf{M} = (\mathbf{F}, \mathbf{R})$*, based on a* TF-FOL*, is a set of weighted temporal facts and rules where* **F** *and* **R** *are sets of pairs s.t.:*
– $\mathbf{F} = \{(\phi_1, w_1), \ldots, (\phi_n, w_n)\}$ *with* $\forall i \in \{1, \ldots, n\}$, $\phi_i \in$ TF-FOL *such that it is a ground formula and* $w_i \in [0, \infty[$,
– $\mathbf{R} = \{(\phi'_1, w'_1), \ldots, (\phi'_k, w'_k)\}$ *with* $\forall i \in \{1, \ldots, k\}$, $\phi'_i \in$ TF-FOL *such that it is not a ground formula and in the form (premises, conclusion), i.e.,* $(\psi_1 \wedge \ldots \wedge \psi_l) \rightarrow \psi_{l+1}$ *where* $\forall j \in \{1, \ldots, l+1\}$, $\psi_j \in$ TF-FOL*, and* $w_i \in [0, \infty[$.
The universe of all TMLNs is denoted by TMLN.

We want to extend MLN to reason on uncertain and temporal knowledge graphs which contain only facts and rules. A fact is a ground formula and a rule contains variables. However, a not ground formula is not always a rule, for instance $\forall x P(x)$ is not a rule. We choose to use one specific syntax $(\psi_1 \wedge \ldots \wedge \psi_l) \rightarrow \psi_{l+1}$ to define a rule but any equivalent syntax works (e.g. $\neg\psi_1 \vee \ldots \vee \neg\psi_l \vee \psi_{l+1}$).

In our UTKGs we have uncertain knowledge which are described with a weight $w \in [0, 1[$, and we have certain information or hard constraints, represented with a very large numbers, *e.g.,* $w = 10^{10}$.

In the following, we simplify the example by directly using the structure defined in Example 3 (*c.f.* Section 2.1).

Example 2 (Continued). The TMLN representation of our running example can be found in Table 1. We identify 6 independent facts and 2 rules, each one with temporal validity and certainty weights (arbitrary extracted from Example 2).

In classical MLN approaches, before reasoning on the model, one must first produce a ground MLN by instantiating the variables in the formulae according to a set of constants. However, starting from a UTKG, we have a finite amount of initial information (facts) to reason on with rules and the parameters' order in a predicate is important. For instance, according to Example 2, in the rule R_1, we cannot instantiate *Studied(x, CoN, t, t')* by *Studied(NO, CoN, 1340, 1382)* (this information with this interval is unknown) or it would be a non-sense to instantiate the predicate as *Studied(MA, CoN, 1340, 1354)* (the first parameter must be a Person).

For these reasons, we choose to obtain the ground MLN by replacing the rules' variables by constants according to the facts present in our TMLN. We call this step instantiation.

3.2 TMLN Instantiation

Let **M** be a TMLN, we denote by $\mathtt{MI}(\mathbf{M})$ the *Maximal TMLN Instantiation* of **M**. $\mathtt{MI}(\mathbf{M})$ contains the set of **M**'s facts and all ground rules that can be constructed by instantiating all its predicates containing variables by other deductible ground predicates (Reasoning with Def. 4 and 5). A ground rule's weight is the minimum of the weights of the formulae in **M** used to construct the instantiated rule.

Formally, to define the set of instantiations, we have to define two useful notions. Firstly, we denote by $\mathtt{TF}(\mathbf{M}) = \bigcup_{(\phi,w)\in\mathbf{M}} \phi$ the set of temporal formulae (without weight) of $\mathbf{M} \in$ TMLN. Secondly, we define the function $\mathtt{W} : \text{TF-FOL} \times \text{TMLN} \rightarrow [0, \infty[$, returning the maximal weight of a temporal formula deductible from a TMLN: $\mathtt{W}(\phi, \mathbf{M}) = \mathtt{max}(\mathtt{min_w}(\mathbf{M}_1), \ldots, \mathtt{min_w}(\mathbf{M}_m))$ s.t. $\{\mathbf{M}_1, \ldots, \mathbf{M}_m\} = \{\mathbf{M}_i \subseteq \mathbf{M} \mid \mathtt{TF}(\mathbf{M}_i) \vdash \phi$ and $\nexists \mathbf{M}'_i \subset \mathbf{M}_i$ s.t. $\mathtt{TF}(\mathbf{M}'_i) \vdash \phi\}$ and $\mathtt{min_w}(\mathbf{M}_i = \{(\psi_1, w_1), \ldots, (\psi_l, w_l)\}) = \mathtt{min}(w_1, \ldots, w_l)$.

Definition 8 (TMLN Instantiation). *Given* $\mathbf{M} = (\mathbf{F}, \mathbf{R}) \in$ TMLN*, the set of* instantiations MI *of* **M** *is defined as follows:*
$\mathtt{MI}(\mathbf{M}) = \mathbf{F} \cup \{((\phi'_1 \wedge \ldots \wedge \phi'_k \rightarrow \phi_{l+1})_{V \leftarrow C}, w') \mid \exists(\phi_1 \wedge \ldots \wedge \phi_l \rightarrow \phi_{l+1}, w) \in \mathbf{R}$ *s.t.* $\phi'_1 \wedge \ldots \wedge \phi'_k \vdash \phi_1 \wedge \ldots \wedge \phi_l, V = \{v_1, \ldots, v_n\}$ *is the set of variables in* $\phi_1 \wedge \ldots \wedge \phi_k \rightarrow \phi_{k+1}, C = \langle c_1, \ldots, c_n \rangle$ *is a vector of constants replacing each occurrence of the variables,* $V'_i \subseteq V$ *is the set of variables in* $\phi_i, C'_i \subseteq C$ *is the vector of constants replaced in* ϕ_i *and the instantiated rule satisfies the 2 following conditions:*
1. $\forall \phi'_i \in \{\phi'_1, \ldots, \phi'_k\}, \phi'_{iV'_i \leftarrow C'_i} \in \mathtt{Cn}(\mathtt{TF}(\mathbf{M}))$
2. $w' = \mathtt{min}(w, \mathtt{W}(\phi_{1V'_1 \leftarrow C'_1}, \mathbf{M}), \ldots, \mathtt{W}(\phi_{kV'_k \leftarrow C'_k}, \mathbf{M})\}$
where $\phi_{V \leftarrow C}$ *is the formula* ϕ *s.t. all the occurrences of the variable* $v_i \in V$ *are replaced by the constant* $c_i \in C$.

Currently, we only deal with universal (*i.e.,* $\forall$) rules and no existential one (*i.e.,* $\exists$), to simplify the maximal TMLN instantiation. Indeed, with existential rules, we would have to deal with a set of sets of instantiations. Given that we would not know which set of instantiations would be true. We keep this question for future works.

From Example 2, the instantiation of R_1 (resp. R_2) consists of GR_{11} (from F_1, F_3 and F_4) and GR_{12} (from F_1, F_3 and F_5) (resp.

F_1	$(Person(NO, 1320, 1382)$	$, \mathbf{10^{10}})$
F_2	$(Philosopher(NO, 1320, 1382)$	$, \mathbf{10^{10}})$
F_3	$(LivePeriod(NO, MA, 1320, 1382)$	$, \mathbf{10^{10}})$
F_4	$(Studied(NO, CoN, 1340, 1354)$	$, \mathbf{0.4})$
F_5	$(Studied(NO, CoN, 1355, 1360)$	$, \mathbf{0.7})$
F_6	$(\neg Studied(NO, CoN, 1353, 1370)$	$, \mathbf{0.5})$
R_1	$(\forall x^{s_2}, t_1^{s_1}, t_1'^{s_1}, t_2^{s_1}, t_2'^{s_1}, t_3^{s_1}, t_3'^{s_1}, (Person(x^{s_2}, t_1^{s_1}, t_1'^{s_1}) \land LivePeriod(x^{s_2}, MA, t_2^{s_1}, t_2'^{s_1}) \land Studied(x^{s_2}, CoN, t_3^{s_1}, t_3'^{s_1})) \rightarrow PeasantFamily(x^{s_2}, t_{min}, t_{max})$	$, \mathbf{0.5})$
R_2	$(\forall x^{s_2}, t_1^{s_1}, t_1'^{s_1}, t_2^{s_1}, t_2'^{s_1}, (Philosopher(x^{s_2}, t_1^{s_1}, t_1'^{s_1}) \land LivePeriod(x^{s_2}, MA, t_2^{s_1}, t_2'^{s_1})) \rightarrow \neg PeasantFamily(x^{s_2}, t_{min}, t_{max})$	$, \mathbf{0.8})$

Table 1: Example of a TMLN for Nicole Oresme.

$GR_{11} = ((Person(NO, 1320, 1382) \land LivePeriod(NO, MA, 1320, 1382) \land Studied(NO, CoN, 1340, 1354)) \rightarrow PeasantFamily(NO, t_{min}, t_{max})$	$\mathbf{0.4})$
$GR_{12} = ((Person(NO, 1320, 1382) \land LivePeriod(NO, MA, 1320, 1382) \land Studied(NO, CoN, 1355, 1360)) \rightarrow PeasantFamily(NO, t_{min}, t_{max}),$	$\mathbf{0.5})$
$GR_2 = ((Philosopher(NO, 1320, 1382) \land LivePeriod(NO, MA, 1320, 1382)) \rightarrow \neg PeasantFamily(NO, t_{min}, t_{max}),$	$\mathbf{0.8})$

Table 2: Ground Rules Instantiating R_1 and R_2 (from Table 1) for Nicole Oresme.

GR_2 from F_2 and F_3). Hence GR_{11} has a weight of 0.4, GR_{12} of 0.5 and GR_2 of 0.8, see Table 2. A TMLN instantiation $I \subseteq \mathtt{MI}(\mathbf{M})$ is a TMLN only composed of ground formulae. I is also called *a state of the TMLN* $\mathbf{M}$. The universe of all TMLN instantiations is denoted by $\mathtt{TMLN}^*$. An instantiation can be inconsistent. In our example, GR_{11}, F_1, F_3, F_4 imply $PeasantFamily(NO, t_{min}, t_{max})$ while GR_2, F_2, F_3 imply $\neg PeasantFamily(NO, t_{min}, t_{max})$.Then, to obtain the most consistent set of instantiations and find *the most probable state of the world* [6], we compute the *Maximum A-Posteriori* (MAP) inference.

4 TEMPORAL AND UNCERTAIN KNOWLEDGE REASONING

We integrate semantics to TMLNs, then we examine the notions of temporal (in)consistency.

4.1 Temporal MAP Inference

Usually, a Markov logic network $\mathcal{M}$ defines a log-linear probability distribution over possible worlds (*i.e.,* instantiations) ω as follows $p_{\mathcal{M}}(\omega) = \frac{1}{Z} \exp\left(\sum_{(\phi,w)\in\mathcal{M}} wn_\phi(\omega)\right)$, where $n_\phi(\omega)$ is the number of true groundings of ϕ in the possible world ω and Z is a normalisation constant to ensure that $p_{\mathcal{M}}$ can be interpreted as a probability distribution. One common inference task in MLNs is the MAP inference, given a set of ground formulae (the facts) the goal is to compute the most probable instantiations. For each ω, $\sum_{(\phi,w)\in\mathcal{M}} wn_\phi(\omega)$ evaluates the value of the instantiation ω and we call this computation: a semantics. For the rest of this article, we will omit any mention of the normalisation that takes place after semantics, which is only a technical detail.

Semantics computes the strength of a TMLN state. We denote the universe of all semantics by $\mathtt{Sem}$, such that for any $\mathcal{S} \in \mathtt{Sem}$, $\mathcal{S} : \mathtt{TMLN}^* \rightarrow [0, +\infty[$.

Temporal Maximum A-Posteriori (MAP) Inference in TMLN returns the most probable, temporally consistent, and expanded state *w.r.t.* a given semantics. Given $\mathbf{M} \in \mathtt{TMLN}$ and $\mathcal{S} \in \mathtt{Sem}$, a method solving a MAP problem is denoted by: $\mathtt{map} : \mathtt{TMLN} \times \mathtt{Sem} \rightarrow \mathcal{P}(\mathtt{TMLN}^*)$ where $\mathcal{P}(X)$ denote the powerset of X, such that:

$$\mathtt{map}(\mathbf{M}, \mathcal{S}) = \{I \mid I \in \underset{I \subseteq \mathtt{MI}(\mathbf{M})}{\operatorname{argmax}} \mathcal{S}(I) \text{ and } \nexists I' \in \underset{I' \subseteq \mathtt{MI}(\mathbf{M})}{\operatorname{argmax}} \mathcal{S}(I') \text{ s.t. } I \subset I'\}.$$

4.2 Temporal Consistency and Inconsistency

We study here new temporal consistency interactions required to define our *Temporal MAP inference. Temporal Consistency* relations need to be refined according to predicates temporal validity. For a predicate and its negation, no clear definition exists to express the temporal consistency based on their time intervals. We propose a temporal consistency with a general case (*partial*) and a special case (*total*).

To establish the different temporal consistency relations, we introduce a function $\mathtt{TI}$ to create pre-orders between the temporal constants in the domain $Time$ of a $\mathtt{TF\text{-}FOL}$ and which extracts the time points interval from two constants.

Definition 9 (Temporal (in)consistency). *Let a set of formulae* $\Phi \subseteq \mathtt{TF\text{-}FOL}$. *In all the following notions of temporal consistency, we will exceptionally use the classical (non-temporal) logical consequence of* $\mathtt{Cn}$ *in order to work on the original maximal predicate interval (and not all its subsets).*

Temporal consistency:

– Φ *has a partial temporal consistency denoted by* $\mathtt{pCon}(\Phi)$ *iff:* $\forall \phi, \psi \in \mathtt{Cn}(\Phi)$ *s.t.* $\phi = P(x_1, \ldots, x_k, t_1, t_1')$ *and* $\psi = \neg P(x_1, \ldots, x_k, t_2, t_2')$, $(\mathtt{TI}(t_1, t_1') \setminus \mathtt{TI}(t_2, t_2') \neq \emptyset) \land (\mathtt{TI}(t_2, t_2') \setminus \mathtt{TI}(t_1, t_1') \neq \emptyset)$.
Otherwise $\neg\mathtt{pCon}(\Phi)$ *is true.*

– Φ *has a total temporal consistency denoted by* $\mathtt{tCon}(\Phi)$ *iff:* $\forall \phi, \psi \in \mathtt{Cn}(\Phi)$ *s.t.* $\phi = P(x_1, \ldots, x_k, t_1, t_1')$ *and* $\psi = \neg P(x_1, \ldots, x_k, t_2, t_2')$, $(\mathtt{TI}(t_1, t_1') \cap \mathtt{TI}(t_2, t_2') = \emptyset)$.
Otherwise $\neg\mathtt{tCon}(\Phi)$ *is true.*

Temporal inconsistency:

– Φ *has a partial temporal inconsistency denoted by* $\mathtt{pInc}(\Phi)$ *iff:* $\exists \phi, \psi \in \mathtt{Cn}(\Phi)$ *s.t.* $\phi = P(x_1, \ldots, x_k, t_1, t_1')$, $\psi = \neg P(x_1, \ldots, x_k, t_2, t_2')$ *and* $\mathtt{TI}(t_1, t_1') \cap \mathtt{TI}(t_2, t_2') \neq \emptyset$.
Otherwise $\neg\mathtt{pInc}(\Phi)$ *is true.*

– Φ *has a total temporal inconsistency denoted by* $\mathtt{tInc}(\Phi)$ *iff:* $\exists \phi, \psi \in \mathtt{Cn}(\Phi)$ *s.t.* $\phi = P(x_1, \ldots, x_k, t_1, t_1')$, $\psi = \neg P(x_1, \ldots, x_k, t_2, t_2')$ *and* $(\mathtt{TI}(t_1, t_1') = \mathtt{TI}(t_2, t_2'))$.
Otherwise $\neg\mathtt{tInc}(\Phi)$ *is true.*

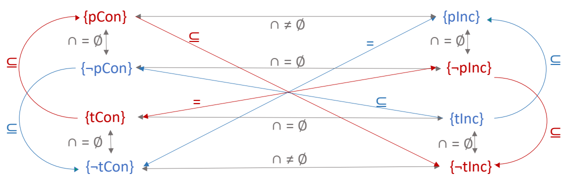

Figure 1: Links between temporal consistency and inconsistency relations.

We now examine the interaction properties between pCon, tCon, pInc and tInc, such as complementarity, subsumption and inclusion.

Definition 10 (Complementarity & Subsumption). *$\forall \Phi \subseteq$ TF-FOL, $\forall$ relation r_1, r_2 if:*
– $r_1(\Phi) \leftrightarrow \neg r_2(\Phi)$ then r_1 and r_2 are complementary.
– $r_1(\Phi) \rightarrow r_2(\Phi)$ then r_1 subsume r_2.

The next two propositions show that firstly tCon and pInc are complementary; secondly different subsumption relations exist between the temporal consistencies.

Proposition 1. *(Complementarity: temporal consistencies) For any $\Phi \subseteq$ TF-FOL:*
$\neg$tCon$(\Phi) \leftrightarrow$ pInc(Φ) *and* tCon$(\Phi) \leftrightarrow \neg$pInc$(\Phi)$.

Proposition 2. *(Subsumption: temporal consisencies) For any $\Phi \subseteq$ TF-FOL:*
pCon$(\Phi) \rightarrow \neg$tInc(Φ), tInc$(\Phi) \rightarrow \neg$pCon(Φ),
$\neg$pCon$(\Phi) \rightarrow$ pInc(Φ), *and* $\neg$pInc$(\Phi) \rightarrow$ pCon(Φ).

In the following, for temporal consistency and inconsistency relations, we denote by $\{r\} = \{\Phi \subseteq \text{TF-FOL} \mid r(\Phi)\}$ their set of formulae sets respecting their condition, where $r \in \{$pCon, tCon, pInc, tInc, $\neg$pCon, $\neg$tCon, $\neg$pInc, $\neg$tInc$\}$.

Definition 11 (Inclusion). *Let two relations of temporal consistency $r_1, r_2 \in \{$pCon, tCon, pInc, tInc, $\neg$pCon, $\neg$tCon, $\neg$pInc, $\neg$tInc$\}$, r_1 is considered included in r_2 if: $\{r_1\} \subseteq \{r_2\}$ iff $\forall \Phi \subseteq$ TF-FOL, $r_1(\Phi) \rightarrow r_2(\Phi)$.*

Proposition 3. *(Inclusion: temporal consistencies)*
$\{$tCon$\} = \{\neg$pInc$\} \subseteq \{$pCon$\} \subseteq \{\neg$tInc$\}$
$\{$tInc$\} \subseteq \{\neg$pCon$\} \subseteq \{$pInc$\} = \{\neg$tCon$\}$

Some inclusions of temporal consistency relations may be defined between the sets of formulae sets that respect them (see Figure 1).

4.3 Temporal Parametric Semantics

To avoid defining several different semantics, we decompose the construction of semantics and identify three steps. Thus, we propose the definition of *Temporal Parametric Semantics*, relying on the combination of three functions: i) a *validation* function Δ of instantiations integrating various consistency relations, ii) a *selecting* function σ able to modify the weight of the formulae of an instantiation and iii) an *aggregate* function Θ returning the final strength.

Definition 12 (Temporal Parametric Semantics). *A* temporal parametric semantics *is a tuple* TPS $= \langle \Delta, \sigma, \Theta \rangle \in$ Sem, *s.t.:*
– Δ : TMLN$^* \rightarrow \{0, 1\}$,
– σ : TMLN$^* \rightarrow \bigcup_{k=0}^{+\infty} [0, 1]^k$,
– $\Theta : \bigcup_{k=0}^{+\infty} [0, 1]^k \rightarrow [0, +\infty[$,
For any $\mathbf{M} \in$ TMLN, $I \subseteq$ MI$(\mathbf{M})$*, the strength of a temporal parametric semantics* TPS $= \langle \Delta, \sigma, \Theta \rangle$ *is computed by:*
TPS$(I) = \Delta(I) \cdot \Theta\Big(\sigma(I)\Big)$.

Once temporal consistency relations are defined, we may enhance semantics for MAP inference with temporal validation functions. One TMLN instantiation can be valid or not according to different criteria (*i.e.,* accept an instantiation).

Definition 13 (Temporal Consistency Constraint Function). *Let* $\mathbf{M} \in$ TMLN*, an instantiation* $I \subseteq$ MI$(\mathbf{M})$ *and* $x \in \{$pCon, tCon, pInc, tInc$\}$*. We define* Δ_x : TMLN$^* \rightarrow \{0, 1\}$*, a temporal consistency validation function according to* x*, s.t.:*

$$\Delta_{\text{pCon}}(I) = \begin{cases} 1 \text{ if pCon}(\text{TF}(I)) \\ 0 \text{ if } \neg\text{pCon}(\text{TF}(I)) \end{cases} \quad \Delta_{\text{tCon}}(I) = \begin{cases} 1 \text{ if tCon}(\text{TF}(I)) \\ 0 \text{ if } \neg\text{tCon}(\text{TF}(I)) \end{cases}$$

$$\Delta_{\text{pInc}}(I) = \begin{cases} 0 \text{ if pInc}(\text{TF}(I)) \\ 1 \text{ if } \neg\text{pInc}(\text{TF}(I)) \end{cases} \quad \Delta_{\text{tInc}}(I) = \begin{cases} 0 \text{ if tInc}(\text{TF}(I)) \\ 1 \text{ if } \neg\text{tInc}(\text{TF}(I)) \end{cases}$$

Corollary 1. *For any* $I \subseteq$ TMLN**,* $\Delta_{\text{tCon}}(I) = \Delta_{\text{pInc}}(I)$*.*

Corollary 2. *Let* $x \in \{$pCon, tCon, pInc, tInc$\}$*, each* Δ_x *is well-behaved.*

Then, we can order the value of the Δ_x for any instantiation.

Proposition 4. *Let* $\mathbf{M} \in$ TMLN *and* Δ_x *a temporal consistency validation function such that* $x \in \{$pCon, tCon, pInc, tInc$\}$*. For any instantiation* $I \subseteq$ MI$(\mathbf{M})$*:* $\Delta_{\text{tCon}}(I) = \Delta_{\text{pInc}}(I) \leq \Delta_{\text{pCon}}(I) \leq \Delta_{\text{tInc}}(I)$

Theorem 1 shows that the strength of the temporal MAP inferences with σ and Θ on any TMLN is ranked according to the temporal consistency validation functions Δ_x.

Theorem 1. *Let* $\mathbf{M} \in$ TMLN*, for any* σ *and* Θ*, as:*
– TPS$_{\text{tCon}} = \langle \Delta_{\text{tCon}}, \sigma, \Theta \rangle$, TPS$_{\text{pInc}} = \langle \Delta_{\text{pInc}}, \sigma, \Theta \rangle$,
– TPS$_{\text{pCon}} = \langle \Delta_{\text{pCon}}, \sigma, \Theta \rangle$, TPS$_{\text{tInc}} = \langle \Delta_{\text{tInc}}, \sigma, \Theta \rangle$.
Hence:
$\forall I_{\text{tCon}} \in$ map$(\mathbf{M}$, TPS$_{\text{tCon}})$, $\forall I_{\text{pInc}} \in$ map$(\mathbf{M}$, TPS$_{\text{pInc}})$,
$\forall I_{\text{pCon}} \in$ map$(\mathbf{M}$, TPS$_{\text{pCon}})$, $\forall I_{\text{tInc}} \in$ map$(\mathbf{M}$, TPS$_{\text{tInc}})$,
TPS$_{\text{tCon}}(I_{\text{tCon}})$ = TPS$_{\text{pInc}}(I_{\text{pInc}}) \leq$ TPS$_{\text{pCon}}(I_{\text{pCon}}) \leq$ TPS$_{\text{tInc}}(I_{\text{tInc}})$.

We study different instances of the aggregate function, using different sums. This type of parameters will determine the strength of an instantiation, in various ways.

Definition 14 (Aggregate Functions). *Let* $\{w_1, \ldots, w_n\}$ *such that* $n \in [0, +\infty[$ *and* $\forall i \in [0, n]$, $w_i \in [0, +\infty[$.
– $\Theta_{sum}(w_1, \ldots, w_n) = \sum_{i=1}^{n} w_i$*, if* $n = 0$ *then* $\Theta_{sum}() = 0$.
– $\Theta_{sum,\alpha}(w_1, \ldots, w_n) = \left(\sum_{i=1}^{n} (w_i)^\alpha\right)^{\frac{1}{\alpha}}$ *s.t.* $\alpha \geq 1$*, if* $n = 0$ *then* $\Theta_{sum,\alpha}() = 0$.

Those aggregate functions target different kinds of semantics. For instance, $\Theta_{sum,\alpha}$ emphasises strong weights for inference, while Θ_{sum} considers weights without appriori.

We propose below a selective function σ_{id} which returns all the weights and another function selecting weights with a threshold ($\sigma_{thresh,\alpha}$).

TPS	MAP Inferences	Example of Conclusion
$\langle \Delta_{\mathtt{tCon}}, \sigma_{id}, \Theta_{sum} \rangle$	$\{\{F_6, GR_{11}, GR_{12}, GR_2\}\}$	$(\neg PF(NO, t_{min}, t_{max}), 0.8)$
$\langle \Delta_{\mathtt{pCon}}, \sigma_{id}, \Theta_{sum} \rangle$	$\{\{F_4, F_6, GR_{12}, GR_2\}\}$	$(\neg PF(NO, t_{min}, t_{max}), 0.8)$
$\langle \Delta_{\mathtt{tInc}}, \sigma_{id}, \Theta_{sum} \rangle$	$\{\{F_4, F_5, F_6, GR_{11}, GR_{12}\}\}$	$(PF(NO, t_{min}, t_{max}), 0.5)$
$\langle \Delta_{\mathtt{tCon}}, \sigma_{id}, \Theta_{sum,2} \rangle$	$\{\{F_6, GR_{11}, GR_{12}, GR_2\}\}$	$(\neg PF(NO, t_{min}, t_{max}), 0.8)$
$\langle \Delta_{\mathtt{pCon}}, \sigma_{id}, \Theta_{sum,2} \rangle$	$\{\{F_4, F_6, GR_{12}, GR_2\}\}$	$(\neg PF(NO, t_{min}, t_{max}), 0.8)$
$\langle \Delta_{\mathtt{tInc}}, \sigma_{id}, \Theta_{sum,2} \rangle$	$\{\{F_4, F_5, F_6, GR_2\}$, $\{F_5, F_6, GR_{11}, GR_2\}\}$	$(\neg PF(NO, t_{min}, t_{max}), 0.8)$

Table 3: *TPS example.* To be short in each instantiation of MAP inferences we omit F_1, F_2, F_3 which are in all instantiations and we abbreviate PeasantFamily by PF.

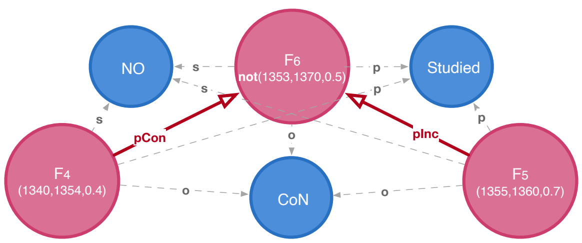

Figure 2: A TMLN property graph representation.

Definition 15 (Selective Functions). *Let* $\mathbf{M} \in \mathtt{TMLN}$, $\{(\phi_1, w_1),\dots,(\phi_n, w_n)\} \subseteq \mathtt{MI}(\mathbf{M})$*:*

– $\sigma_{id}(\{(\phi_1, w_1),\dots,(\phi_n, w_n)\}) = (w_1,\dots, w_n)$

– $\sigma_{thresh,\alpha}(\{(\phi_1, w_1),\dots,(\phi_n, w_n)\}) =$
$(\max(w_1 - \alpha, 0),\dots, \max(w_n - \alpha, 0))$ *s.t.* $\alpha \in [0, +\infty[$

In Chekol et al. [6], the MAP inference uses a semantics working on Herbrand models (with temporal formulae, included in TF-FOL) and is built from uncertain ($w < 1$), certain ($w = 10^{10}$) temporal facts and TMLN certain rules.

This semantics also determines the temporal inconsistency by a classical consistency (if there is no formula φ such that $\Phi \vdash \varphi$ and $\Phi \vdash \neg\varphi$) and by summing all the weights of facts in the instantiation. Therefore, for TMLNs without any uncertain rule, the MAP inference will return the same instantiations as ours, using the temporal parametric semantics $\langle \Delta_{\mathtt{tInc}}, \sigma_{id}, \Theta_{sum} \rangle$. Our Temporal MAP inference generalises their work.

4.4 Example of Reasoning on TMLN

Example 2 (Continued).

In our running theoretical example, let us focus on the temporal consistency parameters and the different aggregations. Then, we are showing in experiments the interest of the selecting function.

Table 3 illustrates different TPS (left column), the results of their MAP inferences (middle column) *i.e.,* the most probable coherent worlds (according to the semantics), and (right column) an example of consistent inference deduced in the MAPs about whether *Nicolas Oresme* comes from a peasant family or not (with its associated probability).

In this example, information that worth processing (impacted by different strategies, *i.e.,* TPS) are F_4, F_5, F_6, GR_{11}, GR_{12} and GR_2. In fact, the issue is to know if they are consistent or not, and if not which ones to choose. To simplify parameters explanation and TPS strategies let's divide the study into two subgroups of information (while all information is kept together during the MAP inference).

First, to illustrate temporal parameterisations between F_4, F_5, F_6, we observe that F_4 and F_5 have an opposite polarity to F_6. Depending on the choice of temporal consistency, some partially contradictory information is considered valid, which provides different consistent sets.

Second, let us analyse aggregations to determine which set of information is more likely to be chosen between GR_{11}, GR_{12} and GR_2. By taking Θ_{sum}, then $\Theta_{sum}(GR_{11}, GR_{12}) > \Theta_{sum}(GR_2)$ ($0.4 + 0.5 > 0.8$). While more probable information should have more impact than sets of less probable ones. Thus, using $\Theta_{sum,2}$ we get $\Theta_{sum,2}(GR_{11}, GR_{12}) < \Theta_{sum,2}(GR_2)$ ($(0.4^2 + 0.5^2)^{\frac{1}{2}} = 0.64 < 0.8$).

Even if $PeasantFamily(NO, t_{min}, t_{max})$ has a weight of 0.5 and its negation 0.8, and we have more inference on the negation, we cannot conclude that $\neg PeasantFamily(NO, t_{min}, t_{max})$ is more likely. For instance, $\langle \Delta_{\mathtt{tInc}}, \sigma_{id}, \Theta_{sum} \rangle$, corresponding to Chekol's semantics, would advocate that *Nicole Oresme* was a Peasant while the one with $\Delta_{\mathtt{pCon}}$ would infer the opposite. The parametric choice of our approach allows historians to decide which inference better corresponds to their own reasoning.

5 TEMPORAL MAP INFERENCE WITH NEOMAPY

In literature, MAP inference computation relies on an ILP solving process which checks ground facts that violate rules and build solutions according to possible solutions with *TeCoRe* [8]. This *model finding* process is optimised by both aggregating ILP sharing common predicates and parallel solving. The complexity of those approaches highly depends on the number of aggregated violating TF.

We propose `NeoMaPy` a totally different approach based on conflicts produced by constraints. It relies on building compatible worlds instead of building iteratively valid worlds. We extract a *conflict graph* between facts, based on rules as in [5] but with weighted nodes. Thus, the particularity of the MAP inference is to find conflict-free graphs by maximising node weights and not by minimising conflict pruning [16].

5.1 Graph of Conflicts

For conflict extraction, we represent a TMLN instantiation I by a Labeled Property Graph [1, 10], where constants and predicates becomes *Concept* nodes. Ground formulae combining those concept nodes are represented as *TF nodes*, with temporal predicates and weights as properties. Rules are expressed as queries on the graph of interactions between *TF* nodes based on their properties, constants and predicates, producing conflict edges between *TF* nodes, labelled with a conflict type. Figure 2 illustrates an instantiation of a TMLN where concept (blue) and TF (red) nodes are represented. The node "*NO*" is the subject (`:s`), "*Studied*" is the predicate node (`:p`) and "*CoN*" is the object (`:o`), for the TF F_4, F_5, F_6. The three TF, in red, correspond to ground information from Ex. 2, with corresponding time frames and polarities. Applying semantics corresponds to a pattern matching query on the graph, searching for conflicts between `TF` nodes. Each constraint (*i.e.,* temporal consistency) is a pattern starting with a common *Concept* node (s, o, p).

For our example, we start with the node "*NO*", for which we find several TFs with opposite polarities while sharing the same concepts (object and predicate). Since their time frames ([1340, 1354] and [1655, 1360] against [1353, 1370]) are not disjoint, we can infer `pCon` and `pInc` conflicts. The type of conflict becomes a property on the extracted conflict link.

After applying all constraints (pattern matching), all conflicts are identified. For a given parametric semantics, we provide for each `TF` node the list of conflictual nodes.

5.2 MAP inference computation

Once the set of conflictual nodes has been obtained, the MAP inference is computed in two steps. We first pre-process our data into a set of connected components (*i.e.,* if there is no path between two nodes, they are not connected). Then, we infer the MAP with our `MaPy` algorithm (Alg. 1).

`MaPy` works as follows: for each node in a dictionary of conflictual nodes (*i.e.,* a connected component), we create step-by-step the list of possible solutions. A solution has three elements: *1)* a list of "solution" nodes, *2)* a list of conflict nodes with the solution and *3)* the solution's weight. The solution is initialised by the first node (line 1). For each node (line 2), we search an existing solution for a compatible node (line 4) for which the node is merged to the solution (line 6). Otherwise, we extract the maximum compatible sub-solution with this node and merge them together as a new solution (line 8).

For optimisation, we firstly remove any solution that can be included in previous one (line 9). Secondly, we look for the maximum potential solution starting from the best current one and naively add the maximum of remaining nodes leading to a maximum potential solution (MPS, line 10). Thirdly, we augment each solution with all non-conflictual nodes with the base solution (worst case) and, when it is lower than MPS, we delete wrong solutions (line 11). Finally, we keep the top-*k* best solutions (*k*=50 in experiments) to avoid useless ones (lines 12-13). After processing all the nodes, each solution is checked for potential missed compatible nodes (line 14). Then, the best solution is kept among the top-*k* (line 15).

Algorithm 1 MaPy(Dico, k)

Input: Dico = {IdNd: [W_Nd, [ConfNd, ...]], ...}, $k \in \mathbb{N}$
Output: Best_Sol = [{Nd, ...}, {ConfNd, ...}, W_Sol]

```
1:  List_Sol = [{Dico[0]}, {Dico[0][1]}, Dico[0][0]];
2:  for nd ∈ Dico do
3:      for sol ∈ List_Sol do
4:          (new_sol,compat) = Compat_Merge(nd,sol);
5:          if compat then
6:              sol.update(nd);
7:          else
8:              List_Sol.add(new_sol);
9:      List_Sol.delete_Include();
10:     Max_Potential_Sol = search_MPS(List_Sol);
11:     List_Sol.delete_Wrong_Solutions(Max_Potential_Sol);
12:     if len(List_Sol) > k then
13:         List_Sol = Top_MAP(List_Sol, k);
14: List_Sol.add_Missing_Nodes(Dico);
15: Best_Sol = Top_MAP(List_Sol, 1);
```

6 EXPERIMENTS

Our approach has been implemented in two distinct parts: 1) building and extracting the graph of conflicts in `Neo4j`[1], 2) processing the MAP inference in `MaPy` (in Python). The source code is available on GitHub[2] and for more information on the tool, a demonstration has been published [12].

6.1 The MAP Inference Extractor

Conflicts extraction from *TF* nodes has been implemented over `Neo4j`. The graph is composed of *Concept* and *TF* nodes (see Section 5.1) with (s, o, p) relationships. Rules are applied to instantiate ground rules as *Cypher* queries.

The rule R_2 (Table 1) is illustrated below. It searches for a pattern composed of a subject "*s*" (`:s`) connected to the"*LivePeriod*" predicate (`:p` & *livp*) from a first TF (*tf1*) and the "*Philosopher*" predicate (`:p` & *phil*) from the second TF (*tf2*). For each pattern match on the graph, it instantiates a new TF (*new_tf*) with the corresponding time frame (T_{min}, T_{max}) with a negative polarity (as stated in rule R_2). This *new_tf* is the connected to its subject *pers*, predicate *PeasantFamily* and premises (*tf1* and *tf2*).

```
MATCH (phil:Concept{ID:"Philosopher"})<-[:p]-(tf1:TF)-[:s]->(pers),
      (livp:Concept{ID:"LivePeriod"}) <-[:p]-(tf2:TF)-[:s]->(pers),
      (mage:Concept{ID:"MA"})         <-[:o]-(tf2)
MERGE (new_tf:TF{start:Tmin,end:Tmax,polarity:false})-[:s]->(pers)
MERGE (mage)<-[:o]-(new_tf)-[:p]->(p4:Concept{ID:"PeasantFamily"})
MERGE (tf1) -[:rule{ID:"R2"}]-> (new_tf) <-[:rule{ID:"R2"}]-(tf2);
```

Then, to provide for each TF its list of conflicts required by Alg. 1 (*Dico*), each constraint from Δ is checked on the graph as a *Cypher* query and materialised as conflicts between incompatible TFs. The *Cypher* query below illustrates the generation of conflicts for the constraint `pCon`. If TF_1 and TF_2 share same concepts (s, o, p) with opposite polarities and a time frames intersection, it produces a "*pCon*" conflict between tf1 and tf2. For optimisation purposes, concept IDs (s, o, p) are repeated in *TF* nodes (*e.g., tf1.p=tf2.p*). Every conflict and inference rule relationship is typed.

```
MATCH (tf1:TF) -[:s]-> (:Concept) <-[:s]- (tf2:TF)
WHERE tf1.p=tf2.p and tf1.o=tf2.o and tf1.polarity <>
  tf2.polarity AND (tf1.start < tf2.start and tf2.start < tf1.end
             AND tf1.end < tf2.end)
MERGE (tf1)-[c:conflict{type:"pCon"}]-(tf2);
```

The final graph may be used to explain the MAP inference. Moreover, for inference a parametric semantics corresponds to a Cypher query that extracts corresponding conflicts (temporal consistency), rules (if a premise is ignored, so does the inferred TF), thresholds (filter on weights), etc.

6.2 Performances

Experiments have been computed on an Intel Xeon-E 2136 - 6c/12t - 3.3 GHz/4.5 GHz with 64GB RAM. *Neo4j* v5.5 has been containerised in *Docker* with 4 cores and 60GB RAM.

As stated, Chekol et al. [8] compute the MAP inference with a parallelised ILP solving process with *TeCoRe* [8] over *n-RockIt*. We compare our approach with them using their dataset [6]. It is composed of different datasets from 5k to 200k predicates (TF)

[1] https://neo4j.com
[2] https://github.com/cedric-cnam/NeoMaPy_Daphne

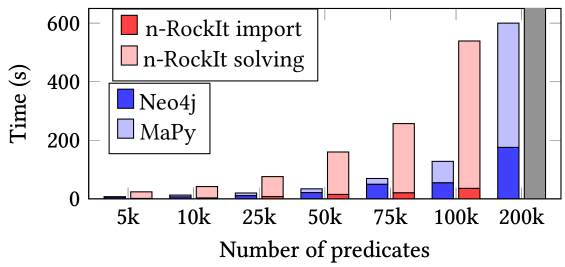

Figure 3: MAP computation time wrt. number of TF (0% false). A gray bar means that computation timed out.

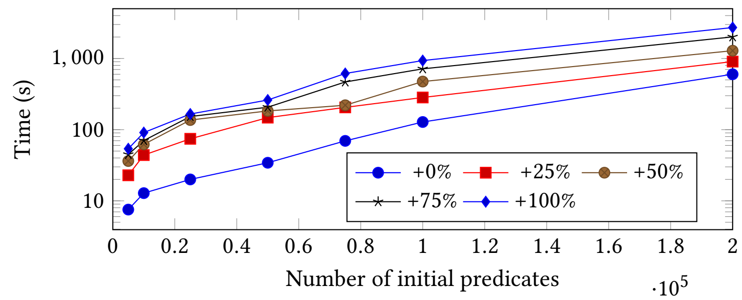

Figure 4: MAP Inference computation wrt. injected conflicts.

		Avg	Rate of injected false					Number of TFs						
			10%	25%	50%	75%	100%	5k	10k	25k	50k	75k	100k	200k
n-rockit	Nb false	63.2%	68.2%	68.2%	56.4%	60.3%	59.0%	65.9%	69.2%	61.6%	57.0%	43.2%	64.8%	N/A
$\sigma_{th,0}$	Nb false	62.8%	67.3%	60.2%	56.9%	55.8%	54.7%	64.1%	68.2%	61.5%	54.8%	53.5%	54.4%	57.2%
top-50	MAP gain	**+2.31%**	**+2.68%**	+4.08%	**+0.54%**	**+5.46%**	**+1.12%**	**+3.07%**	**+3.95%**	**+0.21%**	**+3.58%**	**+0.87%**	**+2.02%**	N/A
$\sigma_{th,0.05}$	Nb false	70.1%	73.7%	71.5%	66.8%	65.9%	67.4%	75.8%	75.4%	70.3%	65.2%	64.6%	65.1%	65.6%
top-50	MAP gain	+1.47%	+1.93%	**+4.39%**	-0.52%	+3.97%	-0.3%	+1.54%	+2.98%	-0.32%	+2.89%	+0.32%	+1.93%	N/A
$\sigma_{th,0.1}$	Nb false	**78.5%**	**81.0%**	**79.4%**	**75.9%**	**75.5%**	**76.5%**	**83.0%**	**81.8%**	**78.7%**	**74.9%**	**74.3%**	**74.8%**	**75.1%**
top-50	MAP gain	-0.22%	+1.36%	**+3.27%**	-2.77%	+0.33%	-4.6%	-0.2%	+1.04%	-2.21%	+0.7%	-0.94%	+1.73%	N/A
$\sigma_{th,0.05}$	Nb false	71.0%	74.6%	72.6%	67.8%	67.1%	68.4%	76.5%	76.3%	71.3%	66.6%	65.7%	66.4%	66.0%
top-5	MAP gain	+0.46%	+1.09%	+3.26%	-1.71%	+2.47%	-1.59%	+0.36%	+1.57%	-1.04%	+2%	-0.36%	+1.14%	N/A

Table 4: MAP Inference quality (TMLN weight & injected false TF) for MaPy (top-k) TPS $\langle \Delta_{\texttt{tInc}}, \sigma_{thresh,x}, \Theta_{sum} \rangle$.

based on wikidata, after adding false predicates (from +25% to +100% more). For a fair comparison, we only focus on a TMLN without polarities and a MAP inference without parametric semantics, while our implementation would allow it.

6.2.1 Efficiency. Figure 3 shows time efficiency of *n-RockIt* vs `NeoMaPy`. We differentiate the preprocessing (import/neo4j) from the MAP inference computation (heuristic/MaPy). We can see that *n-RockIt* spends most of the time to search for the MAP inference while `NeoMaPy` spends more time creating the conflict graph for small datasets while more time in computing the MAP Inference for bigger graphs (200k). It is worth noting that processing graphs bigger than 100k predicates on *n-RockIt* led to a timeout (after 1 hour delay - graphs > 100k). Thanks to our optimistic strategy and heuristic of conflict resolutions, our solution outperforms *n-RockIt*.

Figure 4 shows the computation time of `NeoMaPy` on graphs from 5k to 200k predicates after adding false predicates (from +0% to +100%). Those false predicates produce each time at least one conflict with existing nodes. For example, the graph with 200k predicates with +100% is a graph with 400k predicates and 528,227 conflicts. It is worth noting that our approach keeps a polynomial computation time (shown in log scale) depending on the total number of predicates (including injected false) and conflicts.

6.2.2 MAP Inference Quality. We now study the quality measurement of the MAP Inference (Table 4). For each dataset we computed the solutions' weight (chosen TMLN) and the number of injected false detected. We compare our solutions only when *n-Rockit* did not end with a timeout. `NeoMaPy` runs were computed with various TPS ($\sigma_{thresh,x}$ with x from 0 to 0.1) and top-k for the *MaPy* heuristic (5 & 50). We compare the solutions' weight with those provided by *n-Rockit* and the TMLNs' weight gain with `NeoMaPy`. We can see that we obtain better solutions with a basic strategy ($\sigma_{thresh,0}$, top-*50*) with +2.31% on average. By filtering weights on the TMLN, we reduce the global weight of the TMLN while we keep a positive gain for $\sigma_{thresh,0.05}$.

According to detected false TFs, we can see that on average the basic strategy does not perform well (62.8%). On the other hand, by applying a threshold on weights we remove conflict nodes with few impacts and keep the heuristic on more important nodes. Thus, the ratio of detected false TF grows rapidly up (up to 78.5% with $\sigma_{thresh,0.1}$). The top-5 has also an impact since the heuristic focuses on TFs that contribute more rapidly to the MAP. Consequently, the TPS $\sigma_{thresh,0.05}$ with a top-*5* provides a good compromise between TMLN's weight (+0.46%) and detected false (71%).

Eventually, we show the evolution of false detection and the MAP inference gain (compared to n-Rockit) wrt. the number of injected false and the number of initial predicates. MaPy detects more false TFs from 10% to 25% of injection as well as small graphs (25k) while converging quickly (75% with $\sigma_{thresh,0.1}$). According to the MAP inference gain, neither injection nor graph size have a significant impact on the gain for TMLN's weights.

Interestingly, our parametric MAP Inference enables obtaining MAP inferences of better quality.

Notice that our TMLN worlds rely on grounded facts and not on all possible constants instantiation. However, the comparison is sound since our dataset does not contain any ungrounded formula (with variables), the latters are validated with given constraints.

7 CONCLUSION

Reasoning on KGs has recently been approached with MLNs, to find the most probable state of the world. However, they were limited to a strict temporal inconsistency. In this article, we propose:
– to extend MLN by TMLN with `MS-FOL` which is capable of combining temporal facts and rules;

– to extend MAP inference semantics to temporal data;
– a new temporal parametric semantics (TPS) which offers flexibility and explainability, to tailor semantics reasoning to one's needs;
– a new NeoMaPy approach that outperforms the best existing one (n-RockIt), on both efficiency and quality of the MAP Inference using various TPS.

For future work, we first wish to extend our reasoning model with a learning system to automatically obtain weights on TMLN information, and then test them in practice for goal recognition or other historical research problems. Secondly, we want to extend this work to existential rules, to capture more real data. Thirdly, some relationships between formulae in TMLNs are not yet exploited, and could be represented with argumentation graphs (*e.g.,* the notion of support [9] or similarity between pieces of information [2, 3]), to enhance the weight of inferred knowledge. Eventually, we should investigate more properties of the parametrisation functions to analyse their specific behaviours and adequate strategies.

8 APPENDIX

Proposition 1. Let $\Phi \subseteq$ TF-FOL such that $\texttt{tCon}(\Phi)$, i.e. from Definition 9: $\forall \phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$, $(\texttt{TI}(t_1, t_1') \cap \texttt{TI}(t_2, t_2') = \emptyset)$.

Then its negation $\neg\texttt{tCon}(\Phi)$ is equivalent to: $\neg(\neg\exists\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$ and $(\texttt{TI}(t_1, t_1') \cap \texttt{TI}(t_2, t_2') \neq \emptyset))$.

Hence $\neg\texttt{tCon}(\Phi)$ is equivalent to $\texttt{pInc}(\Phi)$: $\exists\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$ and $(\texttt{TI}(t_1, t_1') \cap \texttt{TI}(t_2, t_2') \neq \emptyset)$.

Moreover its negation, $\neg\texttt{pInc}(\Phi)$ is equivalent to: $\neg(\neg\forall\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$, $(\texttt{TI}(t_1, t_1') \cap \texttt{TI}(t_2, t_2') = \emptyset))$. Therefore $\neg\texttt{pInc}(\Phi)$ is equivalent to $\texttt{tCon}(\Phi)$. □

Proposition 2. Let $\Phi \subseteq$ TF-FOL, from Definition 9:

(1) $\texttt{pCon}(\Phi) \rightarrow \neg\texttt{tInc}(\Phi)$:
- $\neg\texttt{tInc}(\Phi)$ iff, $\neg\exists\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$, $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$ and $(\texttt{TI}(t_1, t_1') = \texttt{TI}(t_2, t_2'))$
- $\texttt{pCon}(\Phi)$ iff $\forall\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$, $(\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') \neq \emptyset) \wedge (\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') \neq \emptyset)$ which is equivalent to: $\neg\exists\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$ and $(\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') = \emptyset) \vee (\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') = \emptyset)$
- If there not exists $\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') = \emptyset$ or $\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') = \emptyset$ then there not exists $\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') = \emptyset$ and $\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') = \emptyset$ (i.e. $\texttt{TI}(t_1, t_1') = \texttt{TI}(t_2, t_2')$), therefore $\texttt{pCon}(\Phi) \rightarrow \neg\texttt{tInc}(\Phi)$.

(2) $\texttt{tInc}(\Phi) \rightarrow \neg\texttt{pCon}(\Phi)$:
- $\neg\texttt{pCon}(\Phi)$ iff $\neg(\neg\exists\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$ and $(\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') = \emptyset) \vee (\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') = \emptyset))$ i.e., $\exists\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$ and $(\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') = \emptyset) \vee (\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') = \emptyset)$
- $\texttt{tInc}(\Phi)$ iff $\exists\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$, $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$ and $(\texttt{TI}(t_1, t_1') = \texttt{TI}(t_2, t_2'))$
- If there exists $\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') = \emptyset$ and $\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') = \emptyset$ $(\texttt{TI}(t_1, t_1') = \texttt{TI}(t_2, t_2'))$, then there exists $\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') = \emptyset$ or $\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') = \emptyset$, therefore $\texttt{tInc}(\Phi) \rightarrow \neg\texttt{pCon}(\Phi)$.

(3) $\neg\texttt{pCon}(\Phi) \rightarrow \texttt{pInc}(\Phi)$:
- $\texttt{pInc}(\Phi)$ iff $\exists\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$, $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$ and $(\texttt{TI}(t_1, t_1') \cap \texttt{TI}(t_2, t_2') \neq \emptyset)$
- If there exists $\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') = \emptyset$ or $\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') = \emptyset$ then there exists $\texttt{TI}(t_1, t_1') \cap \texttt{TI}(t_2, t_2') \neq \emptyset$, therefore $\neg\texttt{pCon}(\Phi) \rightarrow \texttt{pInc}(\Phi)$.

(4) $\neg\texttt{pInc}(\Phi) \rightarrow \texttt{pCon}(\Phi)$:
- $\neg\texttt{pInc}(\Phi)$ is equivalent to: $\neg(\neg\forall\phi, \psi \in \texttt{Cn}(\Phi)$ s.t. $\phi = P(x_1, \cdots, x_k, t_1, t_1')$ and $\psi = \neg P(x_1, \cdots, x_k, t_2, t_2')$, $(\texttt{TI}(t_1, t_1') \cap \texttt{TI}(t_2, t_2') = \emptyset)$)
- Therefore, for any $\texttt{TI}(t_1, t_1'), \texttt{TI}(t_2, t_2')$, if $\texttt{TI}(t_1, t_1') \cap \texttt{TI}(t_2, t_2') = \emptyset$ then $\texttt{TI}(t_1, t_1') \setminus \texttt{TI}(t_2, t_2') \neq \emptyset$ and $\texttt{TI}(t_2, t_2') \setminus \texttt{TI}(t_1, t_1') \neq \emptyset$, therefore $\neg\texttt{pInc}(\Phi) \rightarrow \texttt{pCon}(\Phi)$.

□

Proposition 3. From Proposition 1 and 2, we know that for any $\Phi \subseteq$ TF-FOL:
$\texttt{tCon}(\Phi) \leftrightarrow \neg\texttt{pInc}(\Phi) \rightarrow \texttt{pCon}(\Phi) \rightarrow \neg\texttt{tInc}(\Phi)$
$\texttt{tInc} \rightarrow \neg\texttt{pCon}(\Phi) \rightarrow \texttt{pInc}(\Phi) \leftrightarrow \neg\texttt{tCon}(\Phi)$

Therefore from Definition 11:
$\{\texttt{tCon}\} = \{\neg\texttt{pInc}\} \subseteq \{\texttt{pCon}\} \subseteq \{\neg\texttt{tInc}\}$
$\{\texttt{tInc}\} \subseteq \{\neg\texttt{pCon}\} \subseteq \{\texttt{pInc}\} = \{\neg\texttt{tCon}\}$ □

Proposition 4. From Definition 11, let two relations of temporal consistency $r_1, r_2 \in \{\texttt{pCon}, \texttt{tCon}, \texttt{pInc}, \texttt{tInc}, \neg\texttt{pCon}, \neg\texttt{tCon}, \neg\texttt{pInc}, \neg\texttt{tInc}\}$, r_1 is considered included in r_2 if: $\{r_1\} \subseteq \{r_2\}$ iff $\forall\Phi \subseteq \texttt{TS} - \texttt{FOL}, r_1(\Phi) \rightarrow r_2(\Phi)$.

From Proposition 3: $\{\texttt{tCon}\} = \{\neg\texttt{pInc}\} \subseteq \{\texttt{pCon}\} \subseteq \{\neg\texttt{tInc}\}$ and $\{\texttt{tInc}\} \subseteq \{\neg\texttt{pCon}\} \subseteq \{\texttt{pInc}\} = \{\neg\texttt{tCon}\}$.

Hence, for any set of formulae $\Phi \subseteq$ TF-FOL: $(\texttt{tCon}(\Phi) \leftrightarrow \neg\texttt{pInc}(\Phi)) \rightarrow \neg\texttt{pInc}(\Phi) \rightarrow \texttt{pCon}(\Phi) \rightarrow \neg\texttt{tInc}(\Phi)$.

From Definition 13, using the case equal to 1 and given that $(\texttt{tCon}(\Phi) \leftrightarrow \neg\texttt{pInc}(\Phi)) \rightarrow \neg\texttt{pInc}(\Phi) \rightarrow \texttt{pCon}(\Phi) \rightarrow \neg\texttt{tInc}(\Phi)$, for any instantiation $I \subseteq \texttt{MI}(\mathbf{M})$: $\Delta_{\texttt{tCon}}(I) = \Delta_{\texttt{pInc}}(I) \leq \Delta_{\texttt{pCon}}(I) \leq \Delta_{\texttt{tInc}}(I)$. □

Theorem 1. Let $x \in \{\texttt{pCon}, \texttt{tCon}, \texttt{pInc}, \texttt{tInc}\}$, $\mathbf{M} \in$ TMLN and $\forall I \subseteq \texttt{MI}(\mathbf{M})$, we know from Proposition 4 that:
$\Delta_{\texttt{tCon}}(I) = \Delta_{\texttt{pInc}}(I) \leq \Delta_{\texttt{pCon}}(I) \leq \Delta_{\texttt{tInc}}(I)$.

From Definition 12, a temporal parametric semantics is defined as follows: $\texttt{TPS}(I) = \Delta(I) \cdot \Theta\Big(\sigma(I)\Big)$.

Consequently, $\Delta(I)$ is a coefficient of the combination of σ and Θ. Thus, for any I, σ and Θ, we may order the results according to the $\Delta(I)$ coefficient.

Finally, given that the MAP Inference returns the instantiation with the maximum strength, all strengths of the MAP inferences are equal. Therefore, if we denote by:
- $\texttt{TPS}_{\texttt{tCon}} = \langle\Delta_{\texttt{tCon}}, \sigma, \Theta\rangle$, $\texttt{TPS}_{\texttt{pInc}} = \langle\Delta_{\texttt{pInc}}, \sigma, \Theta\rangle$,
- $\texttt{TPS}_{\texttt{pCon}} = \langle\Delta_{\texttt{pCon}}, \sigma, \Theta\rangle$, $\texttt{TPS}_{\texttt{tInc}} = \langle\Delta_{\texttt{tInc}}, \sigma, \Theta\rangle$.

Hence: $\forall I_{\texttt{tCon}} \in \texttt{map}(\mathbf{M}, \texttt{TPS}_{\texttt{tCon}})$, $\forall I_{\texttt{pInc}} \in \texttt{map}(\mathbf{M}, \texttt{TPS}_{\texttt{pInc}})$, $\forall I_{\texttt{pCon}} \in \texttt{map}(\mathbf{M}, \texttt{TPS}_{\texttt{pCon}})$, $\forall I_{\texttt{tInc}} \in \texttt{map}(\mathbf{M}, \texttt{TPS}_{\texttt{tInc}})$, $\texttt{TPS}_{\texttt{tCon}}(I_{\texttt{tCon}}) = \texttt{TPS}_{\texttt{pInc}}(I_{\texttt{pInc}}) \leq \texttt{TPS}_{\texttt{pCon}}(I_{\texttt{pCon}}) \leq \texttt{TPS}_{\texttt{tInc}}(I_{\texttt{tInc}})$.
□

REFERENCES

[1] Jacky Akoka, Isabelle Comyn-Wattiau, Cédric du Mouza, and Nicolas Prat. 2021. Mapping Multidimensional Schemas to Property Graph Models. In *Advances in Conceptual Modeling*, Iris Reinhartz-Berger and Shazia Sadiq (Eds.). Springer International Publishing, Cham, 3–14.

[2] Leila Amgoud and Victor David. 2018. Measuring Similarity between Logical Arguments. In *Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning KR*. 98–107.

[3] Leila Amgoud and Victor David. 2021. A General Setting for Gradual Semantics Dealing with Similarity. In *35th AAAI Conference en Artificial Intelligence (AAAI 2021)*, AAAI Press, Virtual Conference, United States. https://hal.archives-ouvertes.fr/hal-03141729

[4] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence* (Hyderabad, India) *(IJCAI'07)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2670–2676.

[5] L. Bertossi. 2011. *Database Repairs and Consistent Query Answering*. Morgan & Claypool Publishers. https://books.google.fr/books?id=leVcAQAAQBAJ

[6] Melisachew Chekol, Giuseppe Pirrò, Joerg Schoenfisch, and Heiner Stuckenschmidt. 2017. Marrying uncertainty and time in knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[7] Melisachew Wudage Chekol, Jakob Huber, Christian Meilicke, and Heiner Stuckenschmidt. 2016. Markov Logic Networks with Numerical Constraints. In *Proceedings of the Twenty-Second European Conference on Artificial Intelligence* (The Hague, The Netherlands) *(ECAI'16)*. IOS Press, NLD, 1017–1025.

[8] Melisachew Wudage Chekol, Giuseppe Pirro, Joerg Schoenfisch, and Heiner Stuckenschmidt. 2017. TeCoRe: temporal conflict resolution in Knowledge Graphs. *Proceedings of the VLDB Endowment* 10 (2017), Iss–12.

[9] Andrea Cohen, Sebastian Gottifredi, Alejandro J García, and Guillermo R Simari. 2014. A survey of different approaches to support in argumentation systems. *The Knowledge Engineering Review* 29, 5 (2014), 513–550.

[10] Gwendal Daniel, Gerson Sunyé, and Jordi Cabot. 2016. UMLtoGraphDB: Mapping Conceptual Schemas to Graph Databases. In *Conceptual Modeling*, Isabelle Comyn-Wattiau, Katsumi Tanaka, Il-Yeol Song, Shuichiro Yamamoto, and Motoshi Saeki (Eds.). Springer International Publishing, Cham, 430–444.

[11] Victor David, Raphaël Fournier-S'niehotta, and Nicolas Travers. 2022. Parameterisation of Reasoning on Temporal Markov Logic Networks. https://doi.org/10.48550/ARXIV.2211.16414

[12] Victor David, Raphaël Fournier-S'niehotta, and Nicolas Travers. 2023. NeoMaPy: A Framework for Computing MAP Inference on Temporal Knowledge Graph. 32nd International Joint Conference on Artificial Intelligence, IJCAI.

[13] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, New York, USA) *(KDD'14)*. ACM, New York, NY, USA, 601–610. https://doi.org/10.1145/2623330.2623623

[14] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open Information Extraction from the Web. *Commun. ACM* 51, 12 (dec 2008), 68–74. https://doi.org/10.1145/1409360.1409378

[15] Jean H. Gallier. 1985. *Logic for Computer Science: Foundations of Automatic Theorem Proving*. Harper & Row Publishers, Inc., USA.

[16] Keith W. Hipel, Liping Fang, and D. Marc Kilgour. 2020. The Graph Model for Conflict Resolution: Reflections on Three Decades of Development. *Group Decision and Negotiation* 29, 1 (2020), 11–60. https://doi.org/10.1007/s10726-019-09648-z

[17] Loan T.T Ho, Somjit Arch-int, and Ngamnij Arch-int. 2017. Introducing Fuzzy Temporal Description Logic. In *Proceedings of the 3rd International Conference on Industrial and Business Engineering* (Sapporo, Japan) *(ICIBE 2017)*. Association for Computing Machinery, New York, NY, USA, 77–80. https://doi.org/10.1145/3133811.3133830

[18] Mohamad Yaser Jaradeh, Kuldeep Singh, Markus Stocker, and Sören Auer. 2021. Plumber: A Modular Framework to Create Information Extraction Pipelines. In *Companion Proceedings of the Web Conference 2021*. ACM, New York, NY, USA, 678–679.

[19] Erin Macdonald and Denilson Barbosa. 2020. Neural Relation Extraction on Wikipedia Tables for Augmenting Knowledge Graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM'20*. ACM, New York, NY, USA, 2133–2136. https://doi.org/10.1145/3340531.3412164

[20] Feng Niu, Christopher Ré, AnHai Doan, and Jude Shavlik. 2011. Tuffy: Scaling up Statistical Inference in Markov Logic Networks Using an RDBMS. *Proc. VLDB Endow.* 4, 6 (mar 2011), 373–384. https://doi.org/10.14778/1978665.1978669

[21] Jan Noessner, Mathias Niepert, and Heiner Stuckenschmidt. 2013. RockIt: Exploiting parallelism and symmetry for MAP inference in statistical relational models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 27. 739–745.

[22] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge Graph Identification. In *The Semantic Web – ISWC 2013*, Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 542–557.

[23] Sabbir M. Rashid, Amar K. Viswanathan, Ian Gross, Elisa Kendall, and Deborah L. McGuinness. 2017. Leveraging Semantics for Large-Scale Knowledge Graph Evaluation. In *Proceedings of the 2017 ACM on Web Science Conference* (Troy, New York, USA) *(WebSci '17)*. ACM, New York, NY, USA, 437–442. https://doi.org/10.1145/3091478.3162385

[24] Matthew Richardson and Pedro Domingos. 2006. Markov Logic Networks. *Machine Learning* 62, 1 (01 Feb 2006), 107–136. https://doi.org/10.1007/s10994-006-5833-1

[25] Sebastian Riedel. 2008. Improving the Accuracy and Efficiency of MAP Inference for Markov Logic. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence* (Helsinki, Finland) *(UAI'08)*. AUAI Press, Arlington, Virginia, USA, 468–475.

[26] Romain Rincé, Romain Kervarc, and Philippe Leray. 2018. Complex Event Processing Under Uncertainty Using Markov Chains, Constraints, and Sampling. In *Rules and Reasoning*, Christoph Benzmüller, Francesco Ricca, Xavier Parent, and Dumitru Roman (Eds.). Springer, Cham, 147–163.

[27] Alena Rodionova, Ezio Bartocci, Dejan Nickovic, and Radu Grosu. 2016. Temporal Logic as Filtering. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control* (Vienna, Austria) *(HSCC '16)*. Association for Computing Machinery, New York, NY, USA, 11–20. https://doi.org/10.1145/2883817.2883839

[28] Somdeb Sarkhel, Deepak Venugopal, Parag Singla, and Vibhav Gogate. 2014. Lifted MAP inference for Markov logic networks. In *Artificial Intelligence and Statistics*. PMLR, 859–867.

[29] Lauro Snidaro, Ingrid Visentini, and Karna Bryan. 2015. Fusing uncertain knowledge and evidence for maritime situational awareness via Markov Logic Networks. *Information Fusion* 21 (2015), 159–172.

[30] Nasser Zalmout, Chenwei Zhang, Xian Li, Yan Liang, and Xin Luna Dong. 2021. All You Need to Know to Build a Product Knowledge Graph. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, New York, NY, USA, 4090–4091. https://doi.org/10.1145/3447548.3470825