Temporal Probabilistic Logic Programs: State and Revision

Zoran Majkić
International Society for Research in Science and Technology,
PO Box 2464 Tallahassee, FL 32316 - 2464 USA
majk.1234@yahoo.com, http://zoranmajkic.webs.com/

Abstract

There are numerous applications where we have to deal with temporal uncertainty associated with events. The Temporal Probabilistic (TP) Logic Programs should provide support for valid-time indeterminacy of events, by proposing the concept of an indeterminate instant, that is, an interval of time-points (event’s time-window) with an associated, lower and upper, probability distribution. In particular, we propose the new semantics, for the TP Logic Programs of Dekhtyar and Subrahmanian. Our semantics, based on the possible world semantics is a generalization of the possible world semantics for (non temporal) Probabilistic Logic Programming, and we define the new syntax for PT-programs, with time variable explicitly represented in all atoms, and show how the standard role of Herbrand interpretations used as possible worlds for probability distributions is coherently extended to Temporal Probabilistic Logic Programming.

1 Introduction

The reasoning with probabilistic information based on PSAT (Probabilistic Satisfiability) is the problem of determining whether a set of assignments of probabilities to a collection of boolean formulas of atomic events is consistent has a long history [1, 2, 3], and is proven that is NP-complete. But, the probabilities derived from any sources may have tolerances associated with them, and Fenstad [4] has shown that when enough information is not available about the interaction between events, the probability of compound events cannot be determined precisely: one can only give bounds, lower and upper probability bound. Consequently, the probability intervals used for uncertain information are the simplest extension of the traditional probability modes [5, 6, 7] and are used also as belief measure for uncertainty in fuzzy logics. Such metric for non temporal logic programming (p-programs) is used in a number of papers [8, 9, 11, 12].

We assume that every event occurs at a point in time with a probability interval of reals \([a, b] \subseteq [0, 1]\). An instant (time point or chronon) \(t\) is specified w.r.t a given time granularity of a linear calendar structure \(\mathcal{T}\); for example "day/month/year". Often, however, we do not know the exact time point; instead, we only know that the instant is located sometime during a time interval. We call such an instant an indeterminate instant [13]. Dyreson and Snodgrass have drawn attention to the fact that, in many temporal database applications, there is often uncertainty about the start time of events, the end time of events, and the duration of events. The indeterminacy refers to the time when an event occurred, not whether the event occurred or not. An indeterminate instant is described by lower and upper time bound, and a probability distribution (mass) function (PDF) [14] which, for every time point in this interval, returns with lower and upper probability value assigned to a chronons. Generally, for the interval-based lattice the first introduction of interval-based Temporal Probabilistic Logic Programs (TP-programs) is presented in [15], and is extended to TP-databases [14], so that the semantic of interval-based Probabilistic Logic Programs based on possible worlds and the fixpoint semantics for such programs [8] is considered valid for more than 13 years. But recently the author, had the possibility to approach the general problems with such TP databases [14], to consider the semantics of TP-Logic Programs and to realize that it is not correctly defined.

The initial suspect for the validity of the fixpoint semantics w.r.t the model theory of the p-programs (Probabilistic programs), defined in the seminal paper [8] and successively repeated in all other papers, was based on the two observations: on an unnatural semantics for the probability interval-based bilattice used for computation of the fixpoint, and on the intuition that would be possible to convert the p-programs with
interval-based annotated atoms into the probabilistic constraint programs, and for them there is no guarantee that the solution will contain only simple probabilistic intervals for atoms.

The first consideration was analyzed and presented in [17]; briefly:

In the bilattice $L_B$ of (closed) probability intervals we associate to each fact of knowledge database the belief measure $[x, y] \in L_B$. Such belief is consistent if $x \leq y$, that is, when the lower boundary is less than upper boundary.

The belief (or truth) ordering in $L_B$ is defined as follows: $[x, y] \leq_B [x_1, y_1]$ iff $x \leq x_1$ and $y \leq y_1$. It means that the belief $[x_1, y_1]$ is higher than the belief $[x, y]$, that is, for any probability $a$ which satisfies the first belief (i.e., $a \in [x_1, y_1]$) there is a probability $b$ that satisfies the second belief such that $a \leq b$. The element $0_B = [0, 0]$ and $1_B = [1, 1]$ are the bottom and the top element of this ordering.

There exists also the precision (or knowledge) ordering in $L_B$ and is defined as follows:

$x, y \leq_K [x_1, y_1]$ iff $x \leq x_1$ and $y \geq y_1$.

It means that the belief $[x_1, y_1]$ is more precise than the belief $[x, y]$. The join and meet operations for this ordering, $\vee_{kn}$, $\wedge_{kn}$, respectively, are defined as follows:

$[x, y] \wedge_{kn} [x_1, y_1] = \left[ \min \{x, x_1\}, \max \{y, y_1\} \right]$,

$[x, y] \vee_{kn} [x_1, y_1] = \left[ \max \{x, x_1\}, \min \{y, y_1\} \right]$.

The probabilistic interpretation of conjunction and disjunction correspond to the ignorance strategy, that is $[x, y] \wedge_{ig} [x_1, y_1] = \left[ \max \{0, x + x_1 - 1\}, \min \{y, y_1\} \right]$, $[x, y] \vee_{ig} [x_1, y_1] = \left[ \max \{x, x_1\}, \min \{1, y + y_1\} \right]$, which are not meet/join lattice operators for any of these two orderings (usually the fixpoint computation uses the meet/join lattice operators). In [8] is used the knowledge (precision) ordering for a computation of the least fixpoint semantics for interval-based logic programs, but the disjunction $\vee_{kn}$ can produce as result inconsistent values $[x, y]$ such that $x > y$; moreover, while to the bottom solution is reasonably assigned the whole interval $0_K = [0, 1]$ (bottom value of the lattice), to the 'best solution' is assigned the top value of the lattice $1_K = [0, 1]$ (denominated empty interval $\emptyset$ in [8] which is inconsistent (i.e., the best solutions result inconsistent)!

Also the second observation was investigated by the author and the result is presented in [18] with the reduction of TP-databases into Constraint Logic Programs: consequently, we are able to apply interval PSAT in order to find the models of such interval-based probabilistic programs, but, such models cannot, in general case, be described by single intervals associated with atoms of a program.

In fact, the author discovered that there are cases when the least fixpoint of a p-program is not model of P: it happens always when there is a rule with an atom in the body with the probability interval more thin then the interval for this atom assigned by the least fixpoint (so that this rule could not be satisfied during the least fixpoint computation) and with the atom in the head of this rule with the probability interval more thin that the interval for this atom computed by the least fixpoint. In order to make serious revision for TP-programs and their semantics, we needed some additional mathematical tools also, based on the concepts of the predicate compression and Higher-order Herbrand model types [19] (used also for 'abstracted' databases in [18]).

Remark: Independently form this author’s investigation about the validity of the given semantics for interval-based p-programs, and presumably in the same time (such coincidence is astounding), also the two coauthors which previously worked on this issue for more than 5 years [11, 15], discovered the incorrectness of their previous definition of fixed point semantics for interval-based p-programs. So that in the first paper [20] they presented the contra examples, and proved that the fixpoint semantics of p-programs is unsound (their Proposition 1 shows that the fixpoint semantics derives the interpretations which are not models of a p-program) and incomplete (their Proposition 3 shows that there are models of a p-program which are not interval-based, so that the fixpoint operator is unable to find any solution). The correct semantics for probabilistic programs (p-programs), based on interval PSAT, is presented recently [21] and shows that the entailment problem for p-programs is co-NP-complete. Only for the particular subset of p-programs (denominated simple strict programs in [21]) this complexity is polynomial and can be computed by the original fixpoint semantics defined in [8].

But in this brief history for the relevant work, the TP-programs, which are more complex than p-programs, and which have taken the principal attention of the author during his collaboration for the definition of algebra for TP-databases with aggregations, where not taken into right consideration. This is the main aim of this paper.

The author’s opinion is that the main drawbacks of the work in [15] can be summarized as follows: its fixpoint semantics is incorrect w.r.t. its model theory in the analog way described above in the case of more simple p-programs; the second is based on the fact that its semantics for probability, based on possible worlds, is apparently taken without any plausible connection with the standard semantics for p-programs based on Herbrand interpretations, principally because they did
not explored the possible reductions of TP-programs into p-programs (similarly as in the case when in [8] was not considered the possible reduction into Constraint Logic Programs, and the price was the incorrect fixpoint semantics). Based on these observations, the more important contributions in this work w.r.t. the work in [15] can be summarized as follows:

1. The definition of the new syntax and the model theory for Temporal Probabilistic programs, denominated PT-programs (Probabilistic Temporal Programs), where is considered the full temporal property of events by including the attribute for time-points inside of all atoms (basic events): such atoms will be denominated t-atoms in what follows. By this intuitive and simple operation we obtain t-Herbrand models and indirectly the reduction of PT-programs into p-programs with t-atoms, so that the possible world semantics for the PT-programs with this new syntax is based on standard Herbrand models.

2. Such new PT-programs has the same possible world semantics for p-programs [21] which can be solved, in the general case, by interval PSAT as discussed in precedence.

3. We show how these PT-programs can be transformed in the previous version of TP-programs described in [15], by means of predicate compression for the temporal attribute: thus, the possible worlds of old TP-programs is is the set of Higher-order Herbrand interpretations which are result of this predicate compression. The TP-programs obtained by this transformation (which is knowledge invariant) do not suffer the semantics drawback as in [15], and can be considered as the minimal revision of the work presented in [15].

The plan of this work is the following: After brief introduction in invariant flattening-compression knowledge transformation, in Section 3 we introduce the new syntax for temporal probabilistic logic programs (PT-programs) with t-atoms and more expressive interval-based probabilistic annotation w.r.t. the definition in [15]. In Section 4 we develop the model theoretic semantics for PT-programs, by reduction to ordinary probabilistic p-programs, and we define the complexity for consistency and entailment problem for PT-programs. In Section 5 we make comparison of the new PT-program’s semantics and the model theoretic semantics for TP-programs given in [15]: we show their coincidence, and explain that the possible worlds in [15] are the higher-order Herbrand interpretation obtained by compression of temporal variable in PT-programs.

Finally, in Section 6 we apply the PT-programming for the evolution in time, which modify only p-annotations, of ordinary (non temporal) p-programs. We also discuss the future work and the challenges for effective query-answering in PT-programming for Temporal Probabilistic Databases.

2 Invariant Knowledge transformation: Flattening and compression duality

The higher-order Herbrand interpretations of logic programs (for example Databases) [19], produce models where the true values for ground atoms are not truth constants but functions. In this section we will give the general definitions for such higher-order Herbrand interpretation types for logic programs and their models. More detailed information can be found in [22].

We denote by $A \Rightarrow B$, or $B^A$, the set of all functions from $A$ to $B$, and by $2 = \{0, 1\}$ the set of logic values (0 for the false, 1 for the true logic value).

**Definition 1** (Higher-order Herbrand interpretation types [19]): Let $H^{com}$ be a Herbrand base, then, the higher-order Herbrand interpretations are defined by $I_{com} : H^{com} \rightarrow T$, where $T$ denotes the functional space $W_1 \Rightarrow (\ldots(W_n \Rightarrow 2)\ldots)$, denoted also as $\ldots((2^{W_n})^{W_{n-1}})\ldots W_1$, and $W_i$, $i \in [1, n]$, $n \geq 1$, the sets of parameters.

In the case $n = 1$, $T = (W_1 \Rightarrow 2)$, we will denote this interpretation by $I_{com} : H^{com} \rightarrow 2^{W_1}$.

The interpretations $I_{com} : H^{com} \rightarrow 2^W$ are higher-order types of Herbrand interpretations: the set of truth values for them are functions instead of constants. We pass from a flat truth structure for atoms in a Herbrand interpretations to non flat functional space truth structure for atoms in the compressed Herbrand base $H^{com}$.

Now we will introduce the top-down transformation, called flattening, where the context (uncertain or approximated information), defined as the set $W$, is fused into the Herbrand base by enlarging original predicates of old theory with new attributes taken from the context. In this way the hidden information of the context becomes a visible information and a visible part of the logic language. In what follows, for any given k-ary predicate symbol $r$, for a given tuple of constants $d = d_1, \ldots, d_k >$ in a given Herbrand universe, $r(d)$ denotes a ground atom of the resulting Herbrand base $H^{com}$.

**Definition 2** (Flattening - Global decompression [23])

Each higher-order Herbrand interpretation $I_{com} : H^{com} \rightarrow T$, where $T$ denotes the functional space $W_1 \Rightarrow (\ldots(W_n \Rightarrow 2)\ldots)$, and $W = W_1 \times \ldots \times W_n$ cartesian product, can be flattened into the Herbrand interpretation $I_F : H_F \rightarrow 2$, where $H_F = \{r_F(d, w) | r(d) \in H^{com} \text{ and } w \in W\}$.
is the Herbrand base of new predicates $r_F$, obtained as extension of original predicates $r$ by parameters (attributes for domains in $W_i$, $1 \leq i \leq n$), such that for any $r_F(d, w) \in H_F$, $w = (w_1, ..., w_n) \in W$, holds that $I_F(r_F(d, w)) = I_{com}(r(d))(w_1)...(w_n)$.

We define as parameterizable Herbrand base any Herbrand base such that all its atoms have the common set of attributes $y = \{y_1, y_2, ..., y_k\}$. In this, most simple case of compression, we can obtain a compressed Herbrand base, denoted by $H_{com}$, in the way that these common attributes become hidden attributes.

**Definition 3** (Global compression [23]).

Let $I_F : H_F \rightarrow 2$ be the 2-valued Herbrand interpretation for a parameterizable Herbrand base $H_F$. Then the interpretation for its compressed Herbrand base $H_{com} = \{r(d) \mid r_F(d, w) \in H_F\}$ is defined by $I_{com} : H_{com} \rightarrow 2^W$, such that $I_{com} = [I_F \circ is]$, where $W = \text{Dom}_{y_1} \times ... \times \text{Dom}_{y_n}$ is the set of all parameter tuples. This bijective correspondence of $I_F$ and $I_{com}$ is given by the following commutative diagram

\[
\begin{array}{ccc}
2^W \times W & \xrightarrow{\text{eval}} & 2 \\
I_{com} = [I_F \circ is] & \xrightarrow{\text{id}W I_F \circ is} & I_F \\
H_{com} \times W & \xrightarrow{is} & H_F
\end{array}
\]

where $is : H_{com} \times W \simeq H_F$ is a bijection, such that for any $r(d) \in H_{com}, w \in W$, $is(r(d), w) = r_F(d, w)$. $\lfloor . \rfloor$ is the currying ($\lambda$ abstraction) for functions, $2^W$ is the set of functions from $W$ to $2 = \{0, 1\}$, and eval is the evaluation of the function in $2^W$ for the values in $W$.

No one of subsets $S \subseteq H_{com}$ can be a model for a compressed database; that is, models for compressed database are not ordinary Herbrand models but a kind of higher-order type of Herbrand models.

### 3 New Probabilistic Temporal programs: Syntax of PT-programs

We will use the same terminology as in [15]. The main difference is that our event atoms, differently form event atoms [15] have also the temporal attribute $y$ with a domain represented by $S_y$, the set of all valid time points $t \in S_y$ of a calendar of a type $T$. For example, let $p(d)$ be an ordinary atom with an $n$-ary predicate symbol $p$ and a tuple of $n$ constants or variables $d$. Then $A = r_F(d, y)$, where $r_F$ is $n+1$-ary predicate symbol obtained from $p$ by enriching it with a new temporal attribute, is an event’s t-atom (temporal-atom).

When the tuple $d$ is composed by only constants and $y$ is a time point in $S_y$ then $A$ is said to be ground t-atom. If $A_1, ..., A_k$ are the (simple) t-atoms, then $A_1 \land ... \land A_k$ and $A_1 \lor ... \lor A_k$ are called compound t-atoms.

Let $L$ be a language generated for compound events by a given set of constants (Herbrand universe) and temporal predicate symbols. We assume that all variable symbols from $L$ are partitioned into three classes: the object variables (contains the regular first order logic variable symbols: variables in a tuple $d$ of the example above), the probabilistic variables (range over the interval of reals in $[0, 1]$) and temporal variables (range over the set of time points $S_y$ of a given calendar: in the examples we will use integer numbers for time points): the temporal variable $y$ in $t$-atoms will be called principal variable $y$ and all other temporal variables will be called independent.

Let $Var$ be a set of variables and $S$ be a set of constants. The terms are defined as follows:

1. all variables $x \in Var$, and constants $d \in S$ are terms;
2. if $f : S^n \rightarrow S$ is a functional symbol of arity $n$ and $\lambda_1, ..., \lambda_n$ are terms, then $f(\lambda_1, ..., \lambda_n)$ is a term.

We define two types of terms: the temporal terms, when $S = S_T$; and probabilistic terms, when $S = [0, 1]$.

**Temporal Constraint:** A temporal constraint $C = c(y_1, ..., y_k)$ with principal variable $y$ and other variables $y_1, ..., y_k$ is defined inductively:

1. let $\lambda$ be a temporal term with the set of variables $y_1, ..., y_k$, then $(y \ op \ \lambda)$, where $op \in \{\leq, <, =, \neq, >, \geq\}$, is a temporal constraint. The $y : \lambda_1 \sim \lambda_1$ is a short denotation for $y \geq \lambda_1 \land \lambda_1 \leq \lambda_1$.
2. if $C_1$ and $C_2$ are temporal constraints with the same principal variable $y$, then $C_1 \land C_2, C_1 \lor C_2$, and $\neg C_1$ are temporal constraints.

A temporal constraint is called normal if it does not contain variables different from the principal variable. Let $C = c(y)$ be a normal temporal constraint, the the solution set of time points of $C$ is equal to $\text{sol}(C) = \{t \mid t \in S_y$ and $c(t)$ is true $\}$, with the cardinality $|\text{sol}(C)|$.

**Probabilistic weight function:** for any given temporal constraint $C = c(y_1, ..., y_k)$, we define the function $\omega_C : S_{y}^{k+1} \rightarrow [0, 1]$, such that for any $\{t, t_1, ..., t_k\} \in S_{y}^{k+1}$, if $\omega_C(t, t_1, ..., t_k) \neq 0$ then $t \in \text{sol}(C)$.

Alternatively, in the case when $|\text{sol}(C)| = m$ is a finite number, we will specify the weight function in the form of the time-ordered set of values $\{v_1, ..., v_m\}$. For example, if $\text{sol}(c(y)) = \{t_1, ..., t_3\}$, a weight function $\omega_C$ can be represented as $\{0.4, 1, 0.5\}$, and it will mean that $\omega_C(t_1) = 0.4, \omega_C(t_2) = 1, \omega_C(t_3) = 0.5$. We will denote by $\sharp$ the constant weight function (equal to 1).
for the constraints with \(|sol(C)| = 1\).
The intuition underlying the above definition is that a
probabilistic weight function \(\omega_C\), of a given temporal
constraint \(C\), assigns a probability \(p \geq 0\) to each time
point in the solution set of this temporal constraint (for
all other time points it must be equal to zero).

**Temporal probabilistic annotation:** a tp-
annotation is a triple \(\langle C, \omega_{C_L}, \omega_{C_U}\rangle\) where \(C\) is tem-
poral constraint, \(\omega_{C_L}\) and \(\omega_{C_U}\) are probabilistic weight
functions for lower and upper bound respectively.

**Remark:** this is more general definition then in [15],
but gives us possibility to model lower and upper prob-
ability boundaries independently.

**Definition 4** Let \(F = A_1 \ldots A_k\) be a compound event
t-atom, where \(* \in \{\land, \lor\}\), and \(\mu = \langle C, \omega_{C_L}, \omega_{C_U}\rangle\) be a
Tp-annotation, then \(F: \mu\) is a tp-annotated basic for-
mula.

Let \(A: \mu, F_1: \mu_1, \ldots F_m: \mu_m\) be tp-annotated basic for-
mulae and \(A\) a t-atom. Then \(A: \mu \leftarrow F_1: \mu_1 \land \cdots \land F_m: \mu_m\)
is a tp-clause.

A Probabilistic Temporal Program (PT-program)
is a finite set of tp-clauses. If \(P\) is a PT-program, we
let ground\((P)\) denote the set of all ground instances of
rules of \(P\). By \(H_P\) we denote the Herbrand base of
a program \(P\) for a given set of constants for object vari-
ables.

**Remark:** as we can see this syntax is simile to the syn-
tax for TP-programs presented in [15]. The main dif-
fERENCE is that all atoms in our definition of PT-programs
are t-atoms: as the consequence we will have that the
temporal constraint in tp-basic formulae is an ‘internal’
annotation for the t-atoms (the temporal attribute of
any t-atom corresponds to the dependent variable of
the temporal constraint), while the probabilistic anno-
tation remains an external (standard) annotation for
t-atoms.

To underlay these simple modification we will use the
same example (Example 2 presented in [15]), but with
a new syntax for PT-programs:

**Example 1:** For a company which deals with pro-
jected arrivals of the packages shipped by the company.
First two rules provide the information on the proba-
bility distribution of the arrival time of an arbitrary
package sent to any place. The third rule gives some
extra information about the arrival time of packages
sent to Paris via express-mail. Three facts about ship-
ments complete this program.

\[
\text{arrived}_P(\text{Item}, \text{Place}, y): \langle y = 3 \sim 5, \{25, 15, 1\},
\{4, 24, 16\}\rangle
\]

\[
\text{sent}_P(\text{Item}, \text{Place}, y): \langle y = 1, \{0.9\}, \sharp\rangle,
\text{arrived}_P(\text{Item}, \text{Place}, y): \langle y = 3 \sim 4, \{3, 2\},
\{54, 36\}\rangle
\]

\[
\text{arrived}_P(\text{Item}, \text{Place}, y): \langle y = 3 \sim 4, \{3, 2\},
\{54, 36\}\rangle
\]

4 Model Theory for PT-programs

In this section we will show that each PT-program
\(P\) has the standard probabilistic model theory based
on the Herbrand base \(H_P\) of \(P\), with the set of possible
worlds equal to the set \(I_F \in 2^{H_F}\) of Herbrand inter-
pretations of \(P\), \(I_F : H_F \rightarrow 2\), where \(2 = \{0, 1\}\) is
the set of logic values (0 for the false, 1 for the true logic
value). Each model theory assumes that in real world
each t-atom in \(H_F\) is either true or false. In our case,
in any possible world \(I_F \in 2^{H_F}\), for any t-predicate
symbol \(r_F\) and the tuple of constants \(d\) of its object
variables, the set of time points \(\{t_i | r_F(d, t_i) \in H_F\)
and \(I_F(r_F(d, t_i)) = 1\) corresponds to the temporal
uncertainty of this event: in each time point of this
set the event’s uncertainty is bounded in a form of a
probability interval.

A variable assignment \(\sigma\) maps each object variable to
an object constant and each temporal variable to the
set \(S_t\) of time points of the calendar. The truth of the
events \(\phi \in L\) in \(I_F\) under \(\sigma\), denoted by \(I_F \models_\sigma \phi\),
is inductively defined as follows:

1. \(I_F \models_\sigma \text{r}_F(a_1, \ldots, a_k, y)\) if \(I_F(\text{r}_F(\sigma(a_1), \ldots, \sigma(y))) = 1\)
   for every t-atom \(r_F(a_1, \ldots, a_k, y)\);
2. \(I_F \models_\sigma \phi \land \psi\) if \(I_F \models_\sigma \phi\) and \(I_F \models_\sigma \psi\);
3. \(I_F \models_\sigma \phi \lor \psi\) if \(I_F \models_\sigma \phi\) or \(I_F \models_\sigma \psi\).

An event \(\phi\) is true in a possible world \(I_F\), or \(I_F\) is a
model of \(\phi\), denoted \(I_F \models \phi\), if \(I_F \models_\sigma \phi\) for all variable
assignments \(\sigma\).

In order to be able to apply the results of the standard
possible world semantics for PT-programs, we have to
show that each PT-program corresponds to standard
probabilistic program (p-program).

**Proposition 1** Each PT-program is a pure Probabilis-
tic Logic Program.

**Proof:** It can be shown by simply unfolding of the
temporal constraints in tp-annotated basic formulae,
that is, by partial grounding of the temporal attributes
of t-atoms in a given PT-program. That is, given a
tp-clause \(A: \mu_0 \leftarrow \bigwedge_{1 \leq k \leq m} F_k: \mu_k\), with the t-atom
\(A = r_F(v, y)\) with a tuple of object variables \(v/o
constants in \(v\) and \(\mu_k = \langle C_k, \omega_{C_{L_k}}, \omega_{C_{U_k}}\rangle, 0 \leq k \leq m\),
we can unfold this tp-clause in the following finite set
(because the calendar is finite) of p-clauses:
\[ \{ F(v, t) : [\omega_{a,b}(t), \omega_{a,b}(t)] \} = \bigwedge_{1 \leq i \leq m} (\bigwedge_{t \in \text{sol}(C_\mu)} F_\mu(t)), \]
where \( F_\mu(t) = F_k(t) : [\omega_{a,b}(t), \omega_{a,b}(t)] \), and \( F_k(t) \) is obtained from the \( F_k \) by substitution of the temporal variable in t-atoms of \( F_k \) by the constant (time point) \( t \).

Thus, we obtain a p-program where all annotations of basic p-formulae are constant probabilistic intervals.

□

Example 2: Let us consider the PT-programs of the Example 1. The first tp-clause of these programs will be unfolded into the set of the following three p-rules:
\[ \begin{align*}
\text{arrived}_F(\text{Item, Place}, 3) : [25, 4] & \leftarrow \text{sent}_F(\text{Item, Place}, 1) : [9, 1], \\
\text{arrived}_F(\text{Item, Place}, 4) : [15, 24] & \leftarrow \text{sent}_F(\text{Item, Place}, 1) : [9, 1], \\
\text{arrived}_F(\text{Item, Place}, 5) : [1, 16] & \leftarrow \text{sent}_F(\text{Item, Place}, 1) : [9, 1].
\end{align*} \]

Notice that such simple transformation for the definition of the TP-programs in [15] is impossible because their version does not include the temporal attribute in event’s atoms.

□

Thus, given Herbrand base \( H_F \) of a PT-program \( P \) (equal to the Herbrand base of the p-program obtained by the unfolding described above), a world probability density function \( KI \) is defined as \( KI : 2^{HF} \rightarrow [0, 1] \), such that for all \( I_F \in 2^{HF}, KI(I_F) \geq 0 \) and \( \sum_{I_F \in 2^{HF}} KI(I_F) = 1 \) (Kolmogorov axioms).

A probabilistic interpretation (p-interpretation) \( I : HF \rightarrow [0, 1] \) of a PT-program \( P \) is defined as follows:
\[ I(A) = \sum_{I_F \in 2^{HF}} I_F(A) \cdot KI(I_F), \]
for any ground t-atom \( A \in HF \).

That is, p-interpretation assigns probabilities to individual ground t-atoms of \( HF \) by adding up the probabilities of all worlds \( I_F \) in which a given t-atom is true (i.e., \( I_F(A) = 1 \)).

Given a p-interpretation \( I \), it can be extended to all compound events in \( L \) by the mapping \( Pr : L \rightarrow [0, 1] \), such that the probability of an event \( \phi \) in the probabilistic interpretation \( Pr \) under a variable assignment \( \sigma \), denoted \( Pr_\sigma(\phi) \), is the sum of all \( KI(I_F) \) such that \( I_F \in 2^{HF} \) and \( I_F \models_\sigma \phi \) (we write \( Pr(\phi) \) when \( \phi \) is ground), that is
\[ Pr_\sigma(\phi) = \sum_{I_F \in 2^{HF}, I_F \models_\sigma \phi} KI(I_F). \]

P-interpretations specify the model-theoretic semantics of p-programs, as follows:

1. \( Pr_\sigma(F) \in [a, b] \) iff \( Pr_\sigma(F) \leq b \), that is, \( a \leq Pr_\sigma(F) \leq b \);
2. \( Pr_\sigma(F_1 : [a_1, b_1] \wedge ... \wedge F_n : [a_n, b_n]) \) iff
\[(\forall 1 \leq i \leq n) (Pr_\sigma(F_i : [a, b]));
3. \( Pr_\sigma(F : [a, b]) \leftarrow F_1 : [a_1, b_1] \wedge ... \wedge F_n : [a_n, b_n] \) iff \( Pr_\sigma(F) = [a, b] \) or \( Pr_\sigma(F) \neq [a, b] \).

As we can see, from the point 1 above, the satisfaction of p-programs is based on the Interval PSAT for the system of inequalities: any assignment by \( I \) (that is, \( Pr_\sigma \)) of point probabilities to the atoms, that satisfies these constraints is a model of \( P \).

Now we are ready to specify the model-theoretic semantics for PT-programs, as follows:

Definition 5 (Satisfaction) Let \( \sigma \) be an assignment only for object variables, then
\[ I \models_\sigma P \text{ iff } (\forall t \in \text{sol}(C)) (Pr_\sigma(F) : [\omega_{a,b}(t), \omega_{a,b}(t)]), \]
where \( \mu = (C, \omega_{a,b}, \omega_{a,b}) \) and \( F(t) \) is obtained from \( F \) by substitution of the temporal variable in t-atoms of \( F \) by the constant (time point) \( t \);
\[ I \models_\sigma F_1 : \mu_1 \wedge ... \wedge F_n : \mu_n \text{ iff } (\forall 1 \leq i \leq n) (I \models_\sigma F_i : \mu_i); \]
\[ I \models_\sigma A : \mu \leftarrow F_1 : \mu_1 \wedge ... \wedge F_n : \mu_n \text{ iff } I \models_\sigma A : \mu \text{ or } I \not\models_\sigma F_1 : \mu_1 \wedge ... \wedge F_n : \mu_n. \]

A tp-clause \( Cl \) is true in a probabilistic interpretation \( I \) (that is, in its extension \( Pr_\sigma \)), or \( I \) is a model of \( Cl \), denoted \( I \models Cl \), iff \( Pr_\sigma \models Cl \) for all object variable assignments \( \sigma \).

\( I \) is a model of a PT-program \( P \) if it is a model for all tp-clauses in \( P \). Let \( Mod(P) \) denote the set of all models of a PT-program \( P \); \( P \) is called consistent iff \( Mod(P) \neq \emptyset \), otherwise \( P \) is called inconsistent.

A PT-program \( P \) is satisfiable iff a model of \( P \) exists.

A tp-annotated basic formula \( F : \mu \) is a logical consequence of a PT-program \( P \), or \( P \) entails \( F : \mu \), denoted \( P \models F : \mu \), iff each model of \( P \) is also model of \( F : \mu \).

Proposition 2 The consistency problem for PT-programs is NP-complete, while the entailment problem for PT-programs is co-NP-complete.

It derives from the reduction of PT-programs into ordinary p-programs, and form the complexity of interval PSAT for linear inequalities (see Th.4.11 in [23] and Th.3 in [21]).

By this results we have shown that the fixpoint semantics for TP-programs, as defined in [15] is incorrect, that is unsound and incomplete, similarly to the simpler case of the fixpoint semantics of p-programs defined in [8] and propagated in the dozen of the papers published after this seminal paper.

Instead, in what follows we will try to save the part of the work in [15] which is correct and can be alternatively used for the temporal probabilistic programming.
5 Comparison of TP and PT Model theories

As we can see by easy verify, the definition of the satisfaction relation, for PT-programs given in Definition 5 and for TP-programs in [15], syntactically is equivalent (consider that the point 1 of the Definition 5 can be reduced to $I \models r : \mu \quad \text{iff} \quad (\forall t \in \text{sol}(C))(Pr_r(F(t)) \in [\omega_C(t), \omega_C(t)])$, which is syntactically equivalent to the point 1 of the definition in [15]. But the $Pr$ used in Definition 5 is based on the probabilistic interpretation $I : H_F \rightarrow [0,1]$ where the set of possible worlds is the set of Herbrand interpretations (as in standard world-based probability model theory), while in [15] on the thread function $th : B_L \rightarrow 2^{S_T}$, with the set of possible worlds equal to the set of all threats (without the clear explanation what is the connection with the standard probabilistic model theory).

In what follows we will show that also their model theory is correct, w.r.t. the different syntax for TP-programs, and we will show how the set of threats used for possible worlds is canonically derived from the standard model theory of PT-programs (reducible to pure p-programs) defined in Section 4.

Let $P$ be a PT-program with t-atoms, the Herbrand base $H_F$ and the Herbrand model $I_F : H_F \rightarrow 2$. Then by the global compression, described in Definition 5 for the temporal attribute of all t-atoms in $P$, we obtain the TP-program $P_{comp}$ with compressed atoms which contain only object variables, and with the higher-order Herbrand model $I_{comp} : H_{comp} \rightarrow 2^W$, such that $I_{comp} = [I_F \circ is]$, $W = S_T$, with the Herbrand base $H_{comp} = \{r(d) : \exists w, r_F(d, w) \in H_F\}$, that is, $H_{comp} = B_L$ and $I_{comp} = th$. The diagram in Definition 5 is as follows:

Thus the set of threats $TH$ for the obtained TP-program $P_{comp}$ corresponds to the set of higher-order Herbrand interpretations, i.e., $TH = (2^{S_T})^{B_L}$, bijective with the set of possible worlds of the PT-program $P$, that is, $(2^{S_T})^{B_L} \simeq 2^{H_F}$, so that the probability density function $KI$ for $P$ and $P_{comp}$ is the same, and the satisfaction relation for $P_{comp}$ is identical to the satisfaction relation for $P$.

Consequently the model-theoretic semantics for TP-programs, defined in [15], is equivalent to the model-theoretic semantics for PT-programs defined in this paper. That is, given a ground atom (with only object variables) $A = r(d) \in B_L$, the probability of this atom $p_M(r(d), t)$ in a given point of time $t$ is equal to the probability of the ground atom $r_F(d, t) \in H_F$, as we can verify $p_M(r(d), t) = \sum_{th \in TH, th(r(d)) \gg t} KI(th)$ (from Def. in [15]) $= \sum_{I_F \in 2^{H_F}} I_F(r_F(d, t) \cdot KI(I_F)$ $= \sum_{I_F \in 2^{H_F}} I_F(r_F(d, t) \cdot KI(I_F)$ $= I(r_F(d, t))$.

So, from this point of view, both syntactical versions for temporal probabilistic logic programming can be used for applications: to have or not visible the time variable of events directly in the atoms of logic programs is left to the user’s choice. With the semantic revision of the old syntax version in [15], and explanation of their possible-world semantics based on the higher-order Herbrand models, TP-programs and PT-programs will have the same solution for their models, based on Interval PSAT.

6 Probabilistic logic in time

In this section we will investigate the generalization of a (non temporal) probabilistic logic in time. We will consider a program $P$ with the same set of rules, but with consecutive (in time) fitting of the probability intervals in its p-annotated basic formulae; the granularity of calendar can be, for example, day, week, month, etc...

We will consider the following evolution of the same p-clause of a p-program $P$, in the i-th instance of time $t_i$: $A : [a_i, b_i] \leftarrow F_1 : [a_{1,i}, b_{1,i}] \land ... \land F_k : [a_{k,i}, b_{k,i}]$ The question is: can we capture these evolutions in time of the same p-program $P$ in a unique PT-program, by replacing the p-clauses with the equivalent tp-clauses, where only tp-annotations are modified. In what follows we will show that it is possible, while it is not possible by the syntax version of TP-programs in [15].

**Definition 6 (Probability Distribution Evolution)** Let $P$ be a probabilistic logic program (p-program) with a Herbrand base $B_L$, which in a given instance of time $t$ has the world probability distribution $PI(t) : 2^{B_L} \rightarrow [0,1]$, which is a model of $P$ (satisfies all p-clauses in $P$).
The complete set of these probability distributions, for all time points in a given interval of time \( \Delta_r \subseteq S_r \), will define the mapping \( PI : \Delta_r \rightarrow ([0, 1]^{2^{B_L}}) \), which represents the evolution of the world probability distribution for the given \( p \)-program \( P \) in this interval of time.

Let \( H_F = \{ r_F(d, t) \mid r(d) \in B_L \text{ and } t \in S_r \} \) be the extended Herbrand base for all \( t \)-atoms, derived from original atoms of the Herbrand base \( B_L \) of a \( p \)-program \( P \). Let us define the subset \( D_\Delta \) of Herbrand interpretations for this new Herbrand base \( H_F \) as follows:

\[ D_\Delta = \{ I_F \mid I_F : H_F \rightarrow 2 \}, \text{ such that all true } t \text{-atoms in } I_F \text{ have the same (distinct) time point } t \in \Delta_r. \]

It is easy to verify that each \( I_F \in D_\Delta \), for which the distinct time point is \( t \), corresponds to some interpretation \( v : B_L \rightarrow 2 \) of the \( p \)-program in the time point \( t \). That is, holds the following bijection \( \text{isf} : D_\Delta \simeq \Delta_r \times 2^{B_L} \), such that for any \( I_F \in D_\Delta \), \( \text{isf}(I_F) = (t, J : B_L \rightarrow 2) \), where for any \( r(d) \in B_L \), \( J(r(d)) = 1 \) if \( I_F(r_F(d, t)) = 1 \); 0 otherwise.

Thus, the following diagram commutes:

\[
\begin{array}{ccc}
[0, 1]^{2^{B_L}} \times 2^{B_L} & \xrightarrow{\text{eval}} & [0, 1] \\
\downarrow{PI} & \quad & \downarrow{[PI]^{-1} \circ \text{isf}} \\
\Delta_r \times 2^{B_L} & \xrightarrow{\text{isf}} & D_\Delta \\
\end{array}
\]

Let us define the conservative extension \( DI : 2^{H_F} \rightarrow [0, 1] \) of the mapping \( [PI]^{-1} \circ \text{isf} : D_\Delta \rightarrow [0, 1] \), such that for any \( I_F \in D_\Delta \) it is equal to the mapping \( [PI]^{-1} \circ \text{isf} \), and for all other \( I_F \in 2^{H_F} \) with \( I_F \notin D_\Delta \) it is equal to zero.

**Definition 7 (Evolution PT-program)** Let \( P \) be a \( p \)-program with a Herbrand base \( B_L \) with the evolution probability distribution \( PI : \Delta_r \rightarrow ([0, 1]^{2^{B_L}}) \) in the time interval \( \Delta_r = [t_1, t_N] \), where \( N \) is the cardinality of \( \Delta_r \).

We define the PT-program \( P_\Delta \), denominated Evolution PT-program in the time interval, with a Herbrand base composed by \( t \)-atoms \( H_F = \{ r_F(d, t) \mid r(d) \in B_L \} \), as follows:

- each \( p \)-annotated basic formula \( F : [a, b] \) of the \( p \)-program \( P \), where \( p \)-annotation in the \( i \)-th instance of time \( t_i, 1 \leq i \leq N \) has the value \( [a_i, b_i] \), is substituted by the correspondent \( tp \)-annotated basic formula \( F_T : \{ y : t_1 \sim t_N, \{a_1, ..., a_N\}, \{b_1, ..., b_N\} \} \), where \( F_T \) is the formula \( F \) where all atoms (with only object variables) are replaced by equivalent \( t \)-atoms with the same object variables and the temporal variable \( y \).

Now we will demonstrate that the whole evolution of the \( p \)-program in time can be equivalently represented in the unique PT-program: that is, all versions of \( p \)-programs are contained in this unique PT-program.

**Proposition 3** Let \( P_\Delta \) be the evolution PT-program of the given \( p \)-program \( P \) with a Herbrand base \( B_L \), the evolution \( PI : \Delta_r \rightarrow ([0, 1]^{2^{B_L}}) \) and its canonical extension \( DI \). Then the mapping \( KI = DI/|S_r| : 2^{H_F} \rightarrow [0, 1] \) is the model of this PT-program \( P_\Delta \).

**Proof:** As first, we will show that \( KI \) is the world probability density function for the PT-program \( P_\Delta \): in fact, for all \( I_F \in 2^{H_F} \), \( KI(I_F) = DI(I_F)/|S_r| \geq 0 \), and

\[
\sum_{I_F \in 2^{H_F}} KI(I_F) = \sum_{I_F \in 2^{H_F}} DI(I_F)/|S_r| = \sum_{t \in S_r} \sum_{v \in 2^{B_L}} [PI]^{-1}(t, v)/|S_r| = \sum_{t \in S_r} 1/|S_r| = 1,
\]

from the fact that \( PI(t) \) is the world probability distribution for \( p \)-program \( P \) in a time instance \( t \) and, consequently, satisfies the Kolmogorov axioms.

As consequence also \( KI \) satisfies the Kolmogorov axioms for the evolution PT-program \( P_\Delta \), so it is its world probability density function.

Now we have only to show that it is also a model for \( P_\Delta \).

Let us show how it works for \( tp \)-annotated basic formula

(1) \( F_T : \{ y : t_1 \sim t_N, \{a_1, ..., a_N\}, \{b_1, ..., b_N\} \} \).

We will reason w.r.t. each single instant of time: let the \( p \)-annotated basic formula \( F : [a_1, b_1] \) of the \( p \)-program \( P \) in the instant time \( t_i \) be satisfied by the world probability distribution \( PI(t) : 2^{B_L} \rightarrow [0, 1] \).

Then we have to show that \( KI \) must satisfy the formula (1) for the instant time \( t_i \). In fact we have that

\[
KI = (1/|S_r|)[PI]^{-1} \circ \text{isf} : 2^{B_L} \rightarrow [0, 1],
\]

is equivalent to

\[
(1/|S_r|)\text{eval} \circ (PI \times id) : \Delta_r \times 2^{B_L} \rightarrow [0, 1],
\]

which is equivalent to disjunctive sum of mappings

\[
\sum_{t \in \Delta_r} PI(t) : 2^{B_L} \rightarrow [0, 1],
\]

that is, for the instant of time \( t_i \) it corresponds to the mapping \( PI(t) : 2^{B_L} \rightarrow [0, 1] \) which is a model of a \( p \)-programs in the time instance \( t_i \) and satisfies the probabilistic constraint (the probability interval \( [a_i, b_i] \), in the instance \( t_i \) of the formula (1). By structural induction we can extend the proof to all formulæ and \( tp \)-clauses in the Evolution PT-program \( P_\Delta \).
7 Future work and conclusion

In this paper we defined a new syntax version for temporal probabilistic logic programs (PT-programs) which uses explicitly the time variable in its t-atoms, and more expressive tp-annotations for interval probabilities, and we have shown that it can be reduced to pure probabilistic programs, with the standard world-based probabilistic model theory based on Herbrand interpretations.

We have shown that each PT-program has the model theoretic semantics equivalent to the model theoretic semantics of TP-programs defined in [15], and explain the reasons why the fixed point semantics in [15] is generally non valid. Moreover, we explain also the meaning of "threads", use for possible worlds in [15], in terms of higher-order Herbrand models of PT-programs.

In the significant example for a kind of version-system for ordinary p-programs, we have shown that the probabilistic program evolution in time, by modifying p-annotations for its basic formulae, can be embedded into the unique PT-program with the same set of clauses, by defining tp-annotations of its basic formulae in order to support this probability-interval modifications of the original p-program. Such feature can not be supported by the original syntax for TP-programs presented in [15]. We used the reduction of PT-programs into ordinary p-programs to define the complexity for the consistency (NP-complete) and entailment problem (co-NP-complete) in the general case of PT-programs (it must hold also for PT-programs defined in [15], because they have the same model theoretic semantics).

By incorporation of the time variable into all atoms of PT-programs, we obtain that the facts of PT-programs define the TP-tuples of Temporal Probabilistic Databases, so that the whole PT-program can be seen as a kind of virtual TP-database, and a kind of Probabilistic-DATALOG. The relationship with many-valuedness and intensionality is presented in [24].

The future work will be dedicated to explore such PT-programs for TP-databases, especially a query-answering in such virtual TP-databases. The query-answering in such TP-databases (i.e., PT-programs) is closely related to the entailment in PT-programs: a ground query $F: \mu$ is just the entailment of this formula from a given PT-program; a query with variables will return with a set of ground formulae, each one entailed from the PT-program. Thus, the high complexity of query answering (co-NP-complete) can be a problem for the big TP-databases. If we consider a PT-program with only one binary atom and 10 constants, the number of possible worlds (that is, the number of variables for Interval SAT) is equal to $2^{100} \approx 10^{30}$ !

Thus we need more investigation, in order to reduce this complexity. One of them is to reduce the complexity of PT-programs, for example, the simple strict programs (see for more details in [21]) have the polynomial complexity, but are to much strong simplification for real problems: we need to investigate some minor syntax restriction for PT-programs but with in practice acceptable query-answering complexity.

The other possibility is to consider only a strict subset of models of PT-programs for the plausible query-answering, as usually applied in query-answering in databases with subset of preferred inconsistency repairings [23], or in non monotonic logic programming. The extremal point of reductions, to the unique preferred model, can be applied if we chose, for example, to use the model with maximum entropy (MA) already stated in the work by Nilsson [20]. The maximum entropy model is the unique probabilistic interpretation $KI: 2^{HF} \rightarrow [0,1]$, which is a model of a PT-program $P$ with a Herbrand base $H_F$, and has the greatest entropy among all the models of $P$, where the entropy of $P_r$, denoted $H(Pr)$, is defined by:

$$H(KI) = - \sum_{I_F \in 2^{HF}} KI(I_F) \cdot \log KI(I_F).$$

Principle of maximum entropy may be taken to compute degrees of belief of formulae [27], and it is shown in [28] for the consistent probabilistic inference. This method applied to probabilistic logic programming [29], based on conditional probabilistic clauses, has shown that reduces the original entropy maximizations to relatively small optimization problems, which can easily be solved by existing MA-technology.

References

[1] G.Boole, “The laws of thought,” Macmillan, London, 1884.

[2] T.Halperin, “Best possible inequalities for the probability of a logical function of events,” American Mathematical Monthly, Vol 72, pp. 343–359, 1965.

[3] G.Georgakopoulos, D.Kavvadias, and C.H.Papadimitriou, “Probabilistic satisfiability,” Journal of Complexity, Vol.4, pp. 1–11, 1988.

[4] J.E.Fenstad, “The structure of probabilities defined on first-order languages,” Studies in inductive logic and probabilities, Vol.2, University of California Press., pp. 251–262, 1980.

[5] P.Walley, “Statistical reasoning with imprecise probabilities,” Chapman and Hall, 1991.
[6] L.M. de Campos, J.F.Huete, and S.Moral, “Probability intervals: A tool for uncertain reasoning,” Int. Journal of Uncertainty, Fuzziness and Knowledge-based Systems, Vol. 2(2), pp. 167–196, 1994.

[7] K. Weichselberger, “The theory of interval-probability as a unifying concept for uncertainty,” Proc. 1st Int. Symp. on Imprecise Probabilities and Their Applications, 1999.

[8] R.T.Ng and V.S.Subrahmanian, “Probabilistic logic programming,” Information and Computation, 101(2), pp. 150–201, 1992.

[9] V.S. Lakshmanan and F.Sadri, “Probabilistic deductive databases,” In Proc. Intl. Logic Programming Symposium, Ithaca, NY:MIT Press, pp. 254–268, 1994.

[10] M. Dekhtyar, A.Dekhtyar, and V.S.Subrahmanian, “A flexible probabilistic database system,” ACM Trans. Database Systems, pp. 419–469, 1997.

[11] M.Dekhtyar, A.Dekhtyar, and V.S.Subrahmanian, “Temporal probabilistic logic programs,” ICLP 99, Las Cruces, USA, pp. 109–123, 1999.

[12] Z.Majkić, O.Udrea, and V.S.Subrahmanian, “Temporal probabilistic logic programs,” In Proc. ECSQARU, LNCS 1638, pp. 279–292, 1999.