Constraint Logic Programming and Logic Modality for Event’s Valid-time Approximation

Zoran Majkić / Copyright IICAI

University of Maryland, College Park, USA
zoran@cs.umd.edu
http://www.cs.umd.edu/~zoran/

Abstract. The Temporal Probabilistic (TP) Database management systems 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, users should be able to control, via query language constructs, the amount of temporal/probability approximation present in derived information. In this paper we present the new, equivalent to the denotational specification, the logic specification of TP-databases based on Constraint Logic Programs: TP-database can be seen as a deductive database with incomplete information, and, consequently, with a number of possible Herbrand models. We develop an epistemic extension of modal Temporal logic with the Kripke model semantics determined by Herbrand models of a TP-database also. Such modal language for events can be used to build the SQL-like query language extensions.

1 Introduction

There are numerous applications where we have to deal with temporal uncertainty associated with events. We assume that every event occurs at a point in time. The ability to automatically store and manipulate time, probabilities, and events is important. An instant (time point or chronon) \( t \) is specified w.r.t a given time granularity of a linear calendar structure \( T \); 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 [1]. The indeterminate instance of some event can be considered as a kind of incomplete (temporal) information. The main contribution in this work is to define a logic for algebraically defined TP-database structures [2] in order to be able to reason about its knowledge: it is Constraint Logic Programming (CLP), so that we are able:

1. To reexamine the valid-time indeterminacy from the classical incomplete-information point of view, based on Herbrand models for Constraint Logic Programs (each Herbrand model is a kind of completion for such incomplete information).
2. To use CLP Herbrand models to build a Kripke frame for an epistemic multi-modal framework, where uncertain event’s information is given by possible-world semantics, and the formal definition of an event is given by simple modal formula: such modal formulae are basic elements for event-based query languages.

Let us briefly introduce the distinction between events and their data component and
consider a predicate \( r^T(x, t) \) for some class of events, with proper (manifest) attribute variables \( x = \{x_1, ..., x_n\} \), and time-occurrence attribute \( t \). The indeterminacy refers to the time \( t \) when an event (a tuple \( r^T(c, t) \), with \( c = \{c_i|1 \leq i \leq n\} \) constants of a database domain) occurred, not whether the event occurred or not: thus we need to distinguish the known part of event-information (all its proper (manifest) attribute values, that is truth of formula \( \exists t r^T(c, t) \), denoted by ordinary tuple \( r(c) \) of an abstracted predicate \( r = \lambda x. (r^T) \) where \( \lambda \) is curring lambda operator), from uncertain time/probability information of \( r^T(c, t) \). Such known part can be considered as ordinary tuple \( r(c) \) of a database and can be manipulated by ordinary relational algebra. But when we consider such fact as a component of some event with uncertain time/probability information, denoted by \( E r(c) \), where the operator \( E \) represents the temporal/probability modality, we have to use the extended algebra operators (extended SQL query language) to derive truth over events. That is, the extended temporal query-language (and its underlying algebra) must be downward compatible with ordinary relational query language (i.e., relational algebra).

In [1] such compatibility for temporal SQL query language is obtained, and we need to consider such issue also for implicit definition of TP-databases.

The plan of this work is following: After a brief introduction to TP-databases and technical preliminaries, in Section 2 we present the Constraint Logic Programming for TP-translation of the static TP-database properties. Such approach is extended in Section 3 in order to obtain full logic translation from the mathematical structures of temporal/probabilistic properties of events into a Constraint Logic Program with incomplete temporal/probabilistic information. Each Herbrand model for these logic programs defines a particular possible probabilistic distribution of events in their time-window.

Finally, in Section 4 we develop the Kripke semantics for an epistemic temporal modal logic, determined by the set of Herbrand models of the Constraint logic program for a TP-database. We show how such modal language can be used in order to define a logic SQL-like query language which correctly implements the ‘implicit’ TP-algebra in [2, 3].

1.1 Introduction to TP-databases

For more exhaustive presentation of TP-databases we refer to the work in [2], so we will restrict only to necessary concepts for the work presented in this paper, but we will use the same terminology. We denote by \( S_T \) the set of all valid time points of a calendar of a type \( T \).

**Temporal Constraint**: A temporal constraint \( C \) over calendar \( T \) is defined inductively:

1. Any atomic temporal constraint, \((t \text{ op } c_i)\), where \( c_i \) is the time-value and \( \text{op} \in \{\leq, <, =, \neq, >, \geq\} \), is a temporal constraint.
2. If \( C_1 \) and \( C_2 \) are temporal constraints, then \( C_1 \land C_2, C_1 \lor C_2, \) and \( \neg C_1 \) are temporal constraints.

The extension (the set of time points) of a temporal constraint \( C_i \) is denoted by \( \text{sol}(C_i) \).

**TP-case statement and TP-tuple**: A TP-case statement

\[ \gamma = \{ \gamma_i = \langle C_i, D_i, L_i, U_i, \delta_i \rangle \mid 1 \leq i \leq n, n \geq 1 \} \]

is an expression where \( C_i \) and \( D_i \) are temporal constraints, \( L_i \) and \( U_i \) are lower and upper probability boundaries, and \( \delta_i : \text{sol}(D_i) \to [0,1] \) is a probability distribution, so that for any \( t \in \text{Sol}(D_i) \),
\( \delta_i(D_i, t) \) is a probability value at \( t \); and the following conditions are satisfied for all \( 1 \leq i \leq n \):

1. \( 0 \leq L_i \leq U_i \leq 1 \). \( \text{sol}(C_i) \subseteq \text{sol}(D_i) \).
2. \( |\text{sol}(C_i)| \geq 1 \), that is, at least one time point satisfy \( C_i \) and \( D_i \).
3. \( \forall j \leq n \). \( (i \neq j \implies \text{sol}(C_i) \cap \text{sol}(C_j) = \{ \}) \).

Let \( A = \{A_1, ..., A_k\} \) be an ordinary relational schema, correspondent to the predicate \( r(x_1, ..., x_k) \), and \( \text{d} = \{d_1, ..., d_k\} \) its ordinary tuple from a database domain, then \( tp = (\text{d}, \gamma) \) is a TP-tuple over a schema \( A \).

Intuitively, \( \gamma \) gives the probability \( p \) for each \( t \in \sigma_\tau \), that the event with data \( d \) occurs at time \( t \); as default, we consider that for each \( t \in \sigma_\tau = \bigcup_{\gamma \in \gamma} \text{sol}(C_i) \) the probability is zero; that is \( \bigcup_{\gamma \in \gamma} \text{sol}(C_i) \) represents the temporal uncertainty for this event. Other important assumption is that each event in a TP-Database is uniquely determined by its data tuple \( \text{d} \). A TP-relation over a relational schema \( A \) is a multiset of TP-tuples over a schema \( A \).

These TP-tuples and TP-relations are implicitly defined by means of temporal constraints \( C_i \) in \( \gamma \) and are not ordinary relations, so cannot be defined by ordinary predicates. Their equivalent, explicit, definition by ordinary tuples and relations is given by so called annotated relations: an annotated tuple (at) provides probabilistic information \((\{L_i, U_i\})\) for one data tuple \( \text{d} \) at one point in time \( t \).

**Annotated relations:** Let \( tp = (\text{d}, \gamma) \) be a TP-tuple over a relational schema \( A \), where \( \gamma = \{\gamma_i = (C_i, D_i, L_i, U_i, \delta_i) \mid 1 \leq i \leq n, n \geq 1\} \).

Then the annotated relation for this TP-tuple \( tp \) is defined as the set \( \{(\text{d}, t, L_i \cdot x, U_i \cdot x) \mid t \in \bigcup_{\gamma \in \gamma} \text{sol}(C_i), x = \delta_i(D_i, t)\} \).

Intuitively, in the above definition, \( x \) represents the percentage of \( \text{sol}(D_i) \)’s probability which is associated with time point \( t \) according to \( \delta \). If the original relation for a schema \( A \) is denoted by the predicate \( r(x_1, ..., x_k) \), the correspondent predicate for annotated relation will be denoted by \( r^T(x_1, ..., x_k, t, L_i, U_i) \), where are added the temporal attribute and lower and upper probability attributes.

### 1.2 Introduction to multi-modal logic

We denote by \( \mathcal{M} = (\mathcal{W}, \{R_i \mid i \geq 1\}, S, V) \) the multi-modal Kripke model with:

1. the set of possible worlds \( \mathcal{W} \),
2. accessibility relations \( R_i \subseteq \mathcal{W} \times \mathcal{W} \),
3. non empty set of individuals \( S \),
4. \( V \) is a function defined in the following two cases:
   4.1 \( V : \mathcal{W} \times F \rightarrow \bigcup_{n \in \omega} S^n \), with \( F \) a set of functional symbols of the language, such that for any world \( w \in \mathcal{W} \), \( f \in F \), \( V(w, f) : S^n \rightarrow S \) is a function (interpretation of \( f \) in \( w \)).
   4.2 \( V : \mathcal{W} \times P \rightarrow \bigcup_{n \in \omega} 2^n \), with \( P \) a set of predicate symbols of the language,

where \( \mathbf{d} \) denotes a tuple of constants in \( S \).

We define the set of terms of this modal logic as follows:
1. all variables \( x \in \text{Var} \), and constants \( d \in S \) are terms;
2. if \( f \in F \) is a functional symbol of arity \( n \) and \( t_1, \ldots, t_n \) are terms, then \( f(t_1, \ldots, t_n) \) is a term.

The extension of an expression \( \varphi \), w.r.t. a model \( M \), a world \( w \in W \), and assignment \( g \) is denoted by \( \| \varphi \|^M_{w,g} \). Thus, if \( r \in F \cup P \) then 
\[
\| r \|^M_{w,g} = V(w, r),
\]
that is, for a set of terms \( t_1, \ldots, t_n \) where \( n \) is arity of \( r \),
\[
\| r(t_1, \ldots, t_n) \|^M_{w,g} = V(w, r)((\| t_1 \|^M_{w,g}, \ldots, \| t_n \|^M_{w,g})),
\]
where, if \( t \) is a variable then \( \| t \|^M_{w,g} = g(t) \).

For any formula \( \varphi \) we define \( M \models_{w,g} \varphi \) iff \( \| \varphi \|^M_{w,g} = 1 \).

The semantics for the universal modal operator \( \square \text{a} \) is defined by: \( M \models_{w,g} \square_i \varphi \) iff
\( M \models_{w',g} \varphi \) for all \( w' \) such that \( (w, w') \in R_i \).

The existential modal operator \( \Diamond \) is equal to \( \neg \square \neg \).

A formula \( \varphi \) is said to be true in a model \( M \) if \( M \models_{w,g} \varphi \) for each assignment function \( g \) and possible world \( w \). A formula is said to be valid if it is true in each model.

## 2 Logic programming for stable TP-flattening

As explained in precedent, each TP-tuple \( tp = (d, \gamma) \) over a schema \( A \), represented by a predicate \( r(x) \), is a complex structure which defines by time/probability approximation some event. Such specification is incomplete, that is, there can be more then one logical model for which such structure is satisfied. In fact, by its flattening we obtain a set of possible annotated tuples \( r^T(d, t, L_t, U_t) \), of its time/probability predicate extension \( r^T \). The full temporal/probability incompleteness of such one predicate can be formulated with the existentially quantified formula \( \exists (t, L_t, U_t) r^T(d, t, L_t, U_t) \), that is, by the following clauses with variables in \( \{ t, L_t, U_t \} \):

\[
\begin{align*}
& r^T(d, t, L_t, U_t) \leftarrow (t_S \leq t) \land (t \leq t_F) \land (L_t \leq U_t) \\
& f \leftarrow r^T(d, t, L_t, U_t) \land r^T(d, t, L'_t, U'_t) \land ((L_t, U_t) \neq (L'_t, U'_t))
\end{align*}
\]

where \( d \) is a data constant of this event, \( t_S \) and \( t_F \) are start and end time points of a calendar \( S_r \).

The second clause is a constraint which specifies that for a given time point can exist only one probability interval \([L_t, U_t] \).

Thus, the logic program above can have a lot of Herbrand models for \( r^T \), the well known property for theory with incomplete information. The temporal/probability approximation instead, restricts the number of Herbrand models.

**Proposition 1** (Logic Program for TP-flattening): Let \( tp = (d, \gamma) \), where \( \gamma = \{ \gamma_i = (C_i, D_i, L_i, U_i, \delta_i) \mid 1 \leq i \leq n, n \geq 1 \} \), be a TP-tuple over a schema \( A \), represented by a predicate \( r(x) \). Then its semantics can be specified by the following logic program

\[
\begin{align*}
& r^T(d, t, L_t, U_t) \leftarrow (C_1(t) \land (L_t = L_1 \cdot \delta_1(D_1, t)) \land (U_t = U_1 \cdot \delta_1(D_1, t))) \lor \\
& \quad \lor (C_n(t) \land (L_t = L_n \cdot \delta_n(D_n, t)) \land (U_t = U_n \cdot \delta_n(D_n, t)))
\end{align*}
\]

where \( C_i(t), 1 \leq i \leq n \), are the atoms of derived built-in predicates from temporal constraints \( C_i \).

**Proof:** By this translation we obtain a positive logic program, thus, with a unique Herbrand model \( I : H_B \rightarrow 2 \), where \( 2 = \{0, 1\} \) denotes the set of false and true logic.
values, and $H_B$ the Herbrand base of this database. It is easy to verify that for any TPrelation $r^T$ of a TP-database $DB_{TP}$ its annotated relation is equal to the set of ground atoms: $\{r^T(d, t, L_t, U_t) \mid I(r^T(d, t, L_t, U_t)) = 1\}$.

There are two important points to discuss for this logic program translation:
1. It defines only invariant data-based structure of events. In fact, this feature derives from the fact that such logic programs have a unique Herbrand model, in contrast to logic theory with incomplete information, which must have a number of models, each one for some consistent completion of missing information.
2. By this translation we do not take in consideration the real meaning of event-based phenomena:

an event $d$, specified by the TP-tuple $tp = (d, \gamma)$, at time point $t$ (for which we obtain an annotated tuple $(d, t, L_t, U_t)$) may have any possible probability value $p$ such that $L_t \leq p \leq U_t$.

### 3 Constraint logic programming for TP-databases

By last considerations in the precedent paragraph, in order to capture the semantics of event’s temporal/probabilistic uncertainty, we realized that the original logic programs introduced for TP-flattening need some ’dynamic’ expansion. We also take in consideration that such extension will result in kind of constraint logic programs which will necessarily have more than one minimal Herbrand model.

In order to respond to these requirements, and to be able to (logically) model the probabilistic behavior of events, we will introduce a new predicate for events, $r^T_p(e, t, L_t, U_t, p)$, enriched by probability attribute $p$. The meaning of this new ’dynamic’ predicate for events is that, for any given event $d$, specified by the TP-tuple $tp = (d, \gamma)$, the meaning of the atom $r^T_p(d, t, L_t, U_t, p_i)$ is ’event $d$ happens at time point $t_i$ with a probability $p_i$ bounded in $[L_t, U_t]$’. Such probability values must be distributed in time w.r.t the probability distribution function $\delta$, that is (the constraints in Prop. 2):

$$p_i = \delta(D, t_i) \cdot \sum_{k=1}^{D_n} \frac{pk}{\sum_{t' \in \text{sol}(D)} \delta(D, t')}. $$

Moreover, the relation obtained by this event’s predicate $r^T_p(d, t, L_t, U_t, p_i)$ in some of possible Herbrand models, must be reduced to a function: for a given event $d$ and a time point $t$ can exist exactly one probability value (the last constraint in Prop. 2).

Now, with these considerations we are able to define the semantics for event-based TP-databases by the following constraint logic programming (CLP): to render more clear idea, we will use the existential quantifier in heads of rules for a probability $p$, which can be eliminated by introducing a Skolem function $h(t)$ in the place of $p$.

**Definition 1. (CLP for TP-databases).** Let $tp = (d, \gamma)$, where $\gamma = \{\gamma_i = \langle C_i, D_i, L_i, U_i, \delta_i \rangle \mid 1 \leq i \leq n, \ n \geq 1\}$, be a TP-tuple over a schema $A$, represented by a predicate $r(x)$. Then its event-based semantics can be specified by the following CLP:

$$\exists p \ r^T_p(d, t, L_t, U_t, p) \iff (C_1(t) \land (L_1 = L_1 \cdot \delta_1(D_1, t)) \land (U_1 = U_1 \cdot \delta_1(D_1, t))) \lor \ldots \lor (C_n(t) \land (L_1 = L_n \cdot \delta_n(D_n, t)) \land (U_1 = U_n \cdot \delta_n(D_n, t)))$$
Each Herbrand model \(DB_P\) we denote by \(r\) number of time points in sol built-in predicates, obtained from temporal constraints \(C_t \leftarrow f\). It is easy to verify that, in any given Herbrand model \(P\) model: integrity constraints only reduce the number of such models. Let us see now heads), so that each Skolem completion of it gives a particular minimal Herbrand Clearly, a CLP introduces constrained atoms. TP引入 an event \(d\), the truth of the set of atoms \(r^T_P(d, t, L_t, U_t, p)\) defines a possible temporal/probability approximation of this event: the integrity constraints (rules with the head \(f\) false value) substantially require that such approximation satisfy the probability distribution of functions \(\delta_t\). The last integrity constraint requires that an event for each point of time has a unique probability value in the specified interval.

**Remark:** From this logic translation we see clearly that TP-databases are constrained databases. In fact the (non integrity) rules in LCP, corresponding to some TP-tuple, are just constrained atoms.

**Proposition 2** *(Semantics for TP-databases).*

A program \(P_{TP}(DB_{TP})\) for a TP-database \(DB_{TP}\) can have (infinitely) many finite Herbrand models: in all of these Herbrand models, the extension \(\{d, t, L_t, U_t\}\), that is annotated relation, of any event’s predicate \(r^T(d, t, L_t, U_t, p)\) is invariant; these Herbrand models differ only in event’s probability approximations \(p\).

Clearly, a CLP introduces incomplete information (by existential quantification of rule heads), so that each Skolem completion of it gives a particular minimal Herbrand model: integrity constraints only reduce the number of such models. Let us see now how each Herbrand interpretation of a (constraint) logic program \(P_{TP}(DB_{TP})\) determines a particular probability distribution for events.

**Definition 2.** Each Herbrand model \(I : H_B \rightarrow 2\), where \(H_B\) is a Herbrand base of a logic program \(P_{TP}(DB_{TP})\) with a set of predicate symbols \(Pr\), determines a particular probability distribution mapping \(D_r : Dom \times S_r \rightarrow [0, 1]\), with the following set of data-tuples \(Dom = \bigcup_{T \in Pr, \text{arity}(T) - 1} S^n\), where \(x\) is the set of manifest (proper) attributes of the event and \(S\) is a set of constants (database domain), as follows: for any predicate symbol \(r^T_P\) in a logic program \(P_{TP}(DB_{TP})\), that is, \(r^T_P \in Pr\), and tuples \(d \in S^{arity(T)-1}, t \in S_r\)

\[D_r(d, t) = p, \text{ if } \exists r^T_P(d, t, L_t, U_t, p)) \in H_B \text{ such that } I(r^T_P(d, t, L_t, U_t, p)) = 1; 0, \text{ otherwise.}\]

It is easy to verify that, in any given Herbrand model \(I\) (and its probability distribution \(D_r\), for any true ground atom \(r^T_P(d, t, L_t, U_t, p)\) holds that \(L_t \leq p = D_r(d, t) \leq U_t\).
That is, the program $P_{TP}(DB_{TP})$ models the meaning for the temporal/probability behavior of events: each Herbrand model of $P_{TP}(DB_{TP})$ is a possible temporal/probability approximation for specified events in $DB_{TP}$.

Shortly, if the mathematical structure of ”TP-tuples”, $tp = (d, \gamma)$, can be seen as a denotational specification of events, the constraint logic programs $P_{TP}(DB_{TP})$ can be seen as declarative-logical specification of events: so, we move from mathematical (algebraic) toward logical semantics of databases for events, based on classical two-valued Herbrand models. Such logic point of view can be useful for considering reasoning capabilities of event-based databases, for data-integration semantics when we integrate two or more event-based databases into one global virtual event-based database, for the logic definition of query language (as in relational database: duality between SQL predicate-logic language and relational algebra), etc..

Example 1: Let us consider a simple example of a $DB_{TP}$, defined by a single TP-tuple over a schema $A$ (defined by the predicate $r(x)$), $tp = (d, \gamma)$, where $\gamma = \{< C_1, D_1, 0.4, 0.8, u >\}$, $u$ is an uniform distribution, and $sol(D_1) = \{t_i | 1 \leq i \leq 8\} \supseteq sol(C_1) = \{t_j | 3 \leq j \leq 6\}$.

Then, the following set of ground atoms is one Herbrand model of $P_{TP}(DB_{TP})$:

$$\{r_T(d, t_j, 0.05, 0.1, p) | 3 \leq j \leq 6\}$$

with $p = 0.06$.

Moreover, for any real number $p$, such that $0.05 \leq p \leq 0.1$, we obtain some possible Herbrand model of this simple $P_{TP}(DB_{TP})$, with the probability distribution for the event $d$, $D_T(d, t) = p$, if $t_3 \leq t \leq t_6$; 0 otherwise.

It is easy to verify that the number of possible Herbrand models depends of temporal/probabilistic approximation quality for a given event $d$: when the temporal/probabilistic information is complete, we obtain just a unique Herbrand model, while, as in cases of ordinary databases, with incomplete information we generally obtain a big number of possible Herbrand models.

Example 2: The best approximation - Complete information. Let consider the case in example 1 when $\gamma = \{< C_1, D_1, 1, 1, u >\}$, $u$ is an uniform distribution, and $sol(C_1) = sol(D_1) = \{t_i\}$. Then, the following set of ground atoms is the unique Herbrand model of $P_{TP}(DB_{TP})$:

$$\{r_T(d, t_1, 1, 1, 1)\}$$

with the probability distribution for the event $d$, $D_T(d, t) = 1$, if $t = t_1$; 0 , otherwise.

Example 3: Fully incomplete time-information.

Let consider the case in example 1 when $\gamma = \{< C_1, D_1, 0, 1, u >\}$, $u$ is an uniform distribution, and $sol(C_1) = sol(D_1) = S_r$.

Then, the following set of ground atoms, for any $0 < p \leq 1.0/|S_r|$, is a Herbrand model of $P_{TP}(DB_{TP})$:

$$\{r_T(d, t_j, 0, 1.0/|S_r|, p) | t_j \in S_r\}$$

with the constant event’s distribution $D_T(d, t) = p$.

4 Temporal/probability approximation as a logic modality

It is well-known that the incomplete information in databases has as a consequence a big number of possible Herbrand models: the query answering to a query $q(x)$, re-
stricted to known facts only, has to be true in all such models. That is, we qualify such queries by a modal formulae $Kq(x)$ of an autoepistemic S5 modal logic with the universal modal quantifier $K$ ("I know"). Now we want to investigate which kind of epistemic logic for events we obtain by considering the temporal/probabilistic incompleteness of TP-databases.

As we explained in the introduction, there is an epistemic difference between the ontological truth of the data facts of an event $d$, that is the atom $r(d)$, and the truth of the event itself (which incorporate the complex temporal/probability approximated information): the truth of it cannot be determined by the ontological truth of $r(d)$ only. Thus when we are reasoning about events, we are interested for the epistemic truth of $r(d)$ which corresponds to Time/probability distribution of this event. Roughly, if we take only temporal indeterminacy, we can tell that $r(d)$ is true only in the temporal window when it can happen and false in all other time points of a calendar: in the example 2, we will have that $r(d)$ is epistemically true only in the time point $t_1$ and epistemically false in all other time instants.

See the Appendix for a more general consideration of abstractions for parameterizable databases.

### 4.1 Modal epistemic logic language for events

Let us consider a predicate $r^p(x, t, L_t, U_t, p)$ for some class of events, with proper (manifest) attribute variables $x = \{x_1, \ldots, x_n\}$, and time/probability attributes $t, L_t, U_t$ and $p$. The indeterminacy refers to the time $t$ and the probability $p \in [L_t, U_t]$ when an event $d$ occurred, not whether the event occurred or not. The set of all clauses which define the event $d$, in the Definition 1, we can denote shortly by a clause

(a) $r^p_d(t, L_t, U_t, p) \iff \Phi(t, L_t, U_t, p)$, where

$$\Phi(t, L_t, U_t, p) \iff \bigwedge_{1 \leq j \leq n} \left( C_j(t) \land \left( L_t = L_j \cdot \delta_j(D_j, t) \right) \land \left( U_t = U_j \cdot \delta_j(D_j, t) \right) \land \left( L_j \cdot \delta_j(D_j, t) \leq p \right) \land \left( p \leq U_j \cdot \delta_j(D_j, t) \right) \right).$$

Let $E_d$ be an event-encapsulation operator which encapsulate the semantics of the temporal/probability modality of an event $d$. Then $E_d r(d)$ is a ground atom which represents a data component of this event $d$, can be considered logically equivalent to the logic formula obtained as conjunction of all true ground atoms $r^p_d(t, L_t, U_t, p)$ of this event, that is to the closed formula $\forall\Phi(t, v_L, v_U, v_p) r^p_d(d, t, v_L, v_U, v_p)$. The limited universal quantifier $\forall\Phi$ (applied to the subset $\{\Phi(t, v_L, v_U, v_p)\}$ of values of tuples $(t, v_L, v_U, v_p)$ for which the formula $\Phi(t, v_L, v_U, v_p)$ is true) corresponds to the universal quantifier $\forall$ only for the full-incomplete temporal/probability information of an event.

It is easy to verify that for each Herbrand model of $\mathcal{P}(DB_{TP})$, the $t$ is a key-attribute in a relation $\{\Phi(t, v_L, v_U, v_p)\}$, so the set of time points in this relation can be seen as all possible worlds where the ground atom $r(d)$ is epistemically true. That is a way to pass from Constraint logic programming toward more compact Epistemic modal logic for events: The limited universal quantifier $\forall\Phi$ over temporal/probability predicate $r^p_d(x, t, v_L, v_U, v_p)$ will be replaced by universal modal quantifier $E_d$ over the basic predicate $r(x)$.

So, the clause (a) of Constraint logic program, in this modal framework will be replaced by the epistemic clause.
(b) $E_d r(d) \leftarrow t$.
That is, we transform the mathematical structure of a TP-tuple $tp = (d, \gamma)$ into the epistemic logical formula $E_d r(d)$ which logically defines an event $d$.

The meaning of $E_d r(d)$ is "the event, which data information is given by $r(d)$, is necessarily true". Intuitively, the modality (semantics) of this operator $E_d$ specifies (and hides) a particular temporal/probability dependent necessity that this event $d$ really happened.

We will demonstrate that $E_d$ is really an universal alethic modal operator: "it is necessary that". Thus, an event-based TP-database can be defined as a Multi-modal logic, where for each event $d$ we have its particular alethic modal operator $E_d$.

In order to obtain the Kripke semantics for these modal operators, we introduce the set of possible worlds $W = S_\tau$ and for any given Herbrand model $I : H_B \rightarrow 2$ of $P_T(P(DB_{TP}))$, we denote by $\mathcal{M}$ the model of this multi-modal logic, and we define the epistemic truth value of $r(d)$, at a world $t \in W$, by

$$\mathcal{M} |_t r(d) \iff \exists (L_t, U_t, p)(I(r_T^p(d, t, L_t, U_t, p)) = 1)$$

that is, for a given assignment function $g$ such that $g(x) = d$, holds

$$\mathcal{M} |_g r(x) \iff \exists (L_t, U_t, p)(I(r_T^p(g(x), t, L_t, U_t, p)) = 1).$$

**Definition 3.** (Epistemic Kripke Model).
Let $I : H_B \rightarrow 2$ be a Herbrand model of a constraint logic program $P_T(P(DB_{TP}))$, with a set $Pr$ of predicate symbols, for a TP-database $DB_{TP}$. We define a Kripke model $\mathcal{M}_I = (W, \mathcal{R}_d \mid d \text{ is an event}, S, V)$, for a new set of derived predicates, without Temporal/probability attributes, $P = \{r \mid r = \lambda x, r_T^p, r_T^p \in Pr\}$, where:

1. $W = S_\tau$ (words are time points)
2. for any two $t, t' \in W$
   $$(t, t') \in \mathcal{R}_d \iff \exists (L_t, U_t, p)(I(r_T^p(d, t, L_t, U_t, p)) = 1)$$
   and $\exists (L_{t'}, U_{t'}, p')(I(r_T^p(d, t', L_{t'}, U_{t'}, p')) = 1) = 1$
3. $S$ is a non empty set of constants of a database $DB_{TP}$ (each event $d \in \bigcup_{n<\omega} S^n$)
4. $V : W \times P \rightarrow 2^{S^n}$, such that for any $t \in W$, $r \in P$, and $d$ holds:
   - $V(t, r)(d) = 1$, if $\exists (L_t, U_t, p)(I(r_T^p(d, t, L_t, U_t, p)) = 1)$
   - $0$, otherwise.

where $S^n$ denotes the set of all $n$-tuples of constants, and $2^{S^n}$ the set of all functions from the set $S^n$ to the set $2 = \{0, 1\}$.

In this model, the accessibility relation $\mathcal{R}_d$ of an event $d$ is generated only for time points in which this event has the probability to happen. The accessibility relations $\mathcal{R}_d$, for all events $d$ in a database $DB_{TP}$, are reflexive, transitive and symmetric, that is, this Kripke multi-modal logic is a S5 system. The following proposition holds:

**Proposition 3** (Event’s temporal modality). Let $\mathcal{M} = \{\mathcal{M}_I \mid I$ is a Herbrand model of $P_T(P(DB_{TP}))\}$ be the set of Kripke models obtained from the set of Herbrand models of a TP-database $DB_{TP}$.

Then, for any TP-tuple $tp = (d, \gamma)$ in this TP-database $DB_{TP}$, which is mathematical definition of the temporal/probabilistic approximation of an event, the logically defined event-formula $E_d r(d)$ is valid formula in $\mathcal{M}$.

**Proof:** Notice that in this case we obtain the same Kripke model for each Herbrand model $I$ of $P_T(P(DB_{TP}))$, i.e., $\mathcal{M} = \{\mathcal{M}_I\}$ is a singleton. $E_d$ is universal modal
operator for accessibility relation $R_d$, and for all time points $t$ which belongs to this binary relation holds $M_t \models r(d)$ for all $t'$ such that $(t, t') \in R_d$, thus, $E_d r(d)$ is true formula in $M_t$, i.e., true in all models $M$, so, it is valid in $M$.

In other words, for this set of models in $M$, the epistemic (modal) logic program, composed by the set of modal formulae $E_d r(d)$ for each event $d$ in TP-database, is equivalent to the constraint logic program $P_{TP}(DB_{TP})$ for this TP-database. In this way we are able to formulate a language based on events: for instance, to make the assertion "the event $d_1$ logically implies the event $d_2$", we can use the logical formula $E_{d_1} r_1(d_1) \rightarrow E_{d_2} r_2(d_2)$, etc.

Notice that the representation of an event $d$ by this modal formula $E_d r(d)$ is valid also for events without temporal/probabilistic uncertainty as in example 2: in that case for this event the accessibility relation is a singleton $R_d = \{(t_1, t_1)\}$, so that $E_d r(d)$ tells us that $r(d)$ is true just in single world (point of time) $t_1$. In this way $E_d r(d)$ can be flattened in the simple true ground atom $r_T(d, t_1)$, where $r_T$ is a predicate with added time-attribute, as used in practice for temporally-determined events.

**Example 4**: Fully incomplete time-information.

In this case the "event"-formula $E_d r(d)$ is a shorthand for the following formula $(U = 1.0/|S_r|)$:

$$\bigwedge \exists 0 < p \leq U \{ r_T(d, t_j, 0, p) | t_j \in S_r \}$$

In fact this long formula is true in all Herbrand models of $P_{TP}(DB_{TP})$ in example 3; analogously, the formula $E_d r(d)$ is true in all models $M$.

### 4.2 Temporal extension for a modal query language for events

In this subsection we will investigate how we can enrich the modal epistemic language, defined in precedence, with explicit consideration of temporal properties of events. The semantics of query answering based on the TP-algebra (in [2]) over an event-based TP-database $DB_{TP}$, is a certain (or known) query-answering with respect to Herbrand models of a constrain logic program $P_{TP}(DB_{TP})$, that is, each retrieved TP-tuples $tp_k = (d_k, \gamma_k)$ (that is, the event $d_k$) corresponds to the valid formula $E_{d_k} r(d_k)$ (true in all Kripke models in $M$ for the modal epistemic logic program of a event's database).

The temporal properties of events, as for example "the event $d_1$ occurred before the event $d_2$", or "two events, $d_1$ and $d_2$, (partially) overlap" need also the two modal operators $[F], [P]$ of temporal modal logics, meaning at all future times and at all past times respectively.

A frame for this extended language needs also a two new accessibility relations $R_F$ and $R_P$ respectively, defined as follows [4]: for any two time points $t, t' \in W = S_r$, such that $t \leq t'$, holds $(t, t') \in R_F$ and $(t', t) \in R_P$.

Now we will extend the Epistemic Kripke model, defined in Def.3, by these two new accessibility relations, $R_F, R_P$, and the following set of functional symbols, $F = \{f_r, m_r, M_r | r \in P\}$, for calculation of the probability value, initial and final time point of an event, respectively.
Definition 4. (Epistemic-temporal Kripke model).
Let $I : H_B \rightarrow 2$ be a Herbrand model of a constraint logic program $P_{TP}(DB_{TP})$, with a set $Pr$ of predicate symbols, for a TP-database $DB_{TP}$. We define a Kripke model $\mathcal{M}_I = (\mathcal{W}, \{R_d \mid d \text{ is an event } \} \cup \{R_P, R_P\}, S, V)$, for a new set of derived predicates, without Temporal/probability attributes, $P = \{r \mid r = \lambda x.r^P_d, r^P_d \in Pr\}$, and a set of functional symbols, $F = \{f_r, m_r, M_r \mid r \in P\}$, where:
Points 1, 2, 3, and 4 are as in Definition 3, plus
5. $V : \mathcal{W} \times F \rightarrow \bigcup_{t \in \mathbb{R}} S^t$, interprets each symbol in $F$ by the following functions (we consider that $S_{t_r}, [0, 1] \subseteq S$):

5.1 The function $v_r = V(t, f_r) : \text{Sarity}(r) \rightarrow S$, such that for any tuple $d \in \text{Sarity}(r)$, holds that $V(t, f_r)(d) = D_r(d, t)$, that is, $v_r(d)$ is a value of probability of an event $d$ in a world (point of time) $t$.

5.2 The functions, $min_r = V(t, m_r), max_r = V(t, M_r)$, such that for any tuple $d \in \text{Sarity}(r)$, holds that $V(t, m_r)(d) = \min\{r' \mid D_r(d, r') > 0, r' \in S_t\}$, and $V(t, M_r)(d) = \max\{r' \mid D_r(d, r') > 0, r' \in S_t\}$; that is, $min_r(d)$ and $max_r(d)$ are the first and the last time points, respectively, of an event $d$.

We denote by $\mathcal{M} = \{\mathcal{M}_I \mid I$ is a Herbrand model of $P_{TP}(DB_{TP})\}$ the set of all Kripke models for a TP-database determined by the set of Herbrand models of the constraint logic program $P_{TP}(DB_{TP})$ for a TP-database $DB_{TP}$.

Notice that in this case, when we want to use a modal logic language, capable to consider also temporal/probabilistic properties of events, for any two Herbrand models $I$ and $J$, such that $I \not= J$, we obtain two distinct Kripke models, $\mathcal{M}_I \not= \mathcal{M}_J$; in fact, there is a bijection between the set of Herbrand models of a constraint logic program $P_{TP}(DB_{TP})$ for a TP-database $DB_{TP}$, and this set of Kripke models $\mathcal{M}$. But as in Proposition 3, for any TP-tuple $tp = (d, \gamma)$ in this TP-database $DB_{TP}$, the logically defined event-formula $E_{dr}(d)$ is true in every Kripke model $\mathcal{M}_I \in \mathcal{M}$, so it is valid formula in $\mathcal{M}$ and defines logically the semantics of the event’s structure $tp = (d, \gamma)$.

We can call this modal logic as Epistemic temporal logic for events. Obviously, we can add to such Modal logic language also other temporal logic operators, as for example Next modal operator, to extend the expressive power of this event-based modal logic.

With this extension of a modal logic language we are able to specify the following temporal/probability relationships between events:

Example 5: BEFORE-relationships.
The fact that ”the event $d$ happens BEFORE the event $d'$” is true if the following modal formula is valid: $E_dr(d) \land E_dr'(d') \land max_r(d) \leq min_{r'}(d')$, where $r, r' \in P$ are predicates for these two events respectively. $\square$

Example 6: OVERLAP-relationships.
Let show how the temporal modal operators can be used in order to specify the fact that ”the event $d$ and the event $d'$ (partially) OVERLAP”, by the validity of the following modal formula: $E_dr(d) \land E_dr'(d') \land (v_r(d) \land v_r'(d') > 0 \lor \neg\neg[F](v_r(d) \land v_r'(d') = 0) \lor \neg\neg[F](v_r(d) \land v_r'(d') = 0))$.

Obviously, we can define such property by using the functions $min_r$ and $max_r$ for
these two events.

This modal language can be used as a computation level language for a SQL-like query language, where the two specifications above can be defined as two ground queries \( r(d) \text{ BEFORE } r'(d') \) and \( r(d) \text{ OVERLAP } r'(d') \).

5 Related work and conclusion

Despite the wealth of research in incomplete information databases, there are few efforts that address temporal incompleteness. Much of the previous research in incomplete information databases has concentrated on issues related to null values [5–9]. Another primary research thrust has studied the applicability of fuzzy set theory to relational databases [10, 11]. Dutta uses a fuzzy set approach to handle generalized temporal events [12], that is, events that have multiple occurrences.

Another proposal intertwines support for value and temporal incompleteness by combining the different kinds of incomplete information [13], a wide spectrum of attribute values are simultaneously modelled. Koubarakis [14] proposes the use of constraints for representing event occurrences. Another important work is that of Brusoni et al. who developed a system called LaTeR.

The field of Probabilistic Database Models (PDM) covers a wide spectrum of different uses of probabilistic information. Probabilistic weights have been attached to attribute values to model situations where an attribute value could be one of several more or less likely values [16]. More recently, in their ProbView, the PDM is extended to eliminate several assumptions, including independence of elements.

In [1] is presented a syntactic extension to SQL, needed to support valid-time indeterminacy, and its operational semantic extension to ordinary SQL. In other approach [2], based on the extension of probabilistic databases with time dimension (Probabilistic Temporal Databases, or shortly TP-databases), is defined a relational algebra extension that integrates both time and probabilities: such algebra supports also partial probabilistic distribution for events. In this work was shown how the PDM can be modified to support temporal indeterminacy, even if there might be several million elements in a set of possible chronons: it is obtained by passing from explicit (annotated) classical relations to implicit representation of events (called also TP-relations) and executing the “implicit” algebra operations directly over such representation.

While in [1], the events are modelled as classical tuples in relational database, with added time attribute for events, in [2] such explicit relational structure is enriched by two added probability attributes for relations (lower and upper bound). The implicit representation of events, that is “TP-relations” defined in [2], is instead, a more complex logical structure which does not correspond to classical relational databases and, from our point of view, needs a confront with other higher database types, as, for instance, deductive databases.

Consequently, we have presented a constraint logic programming for TP-databases which is a logical translation of complex mathematical temporal/probability structures of events. Such logic programs have a number of possible Herbrand models, derived from the fact that TP-database contain incomplete temporal/probability information (approximation) for events. Such incompleteness is orthogonal to any other kind of
a database incomplete information, so that also other kind of incomplete information may be supported by this constraint logic programming. The ‘implicit’ TP-algebra for such TP-databases in this case corresponds to the transformation of logic programs. The advantage of this logical point of view for a TP-database is that it can be easily integrated with ordinary deductive database. By adding to the program $P_{TP}(DB_{TP})$ the set of clauses for ordinary database relations (which are not event-based relations) we obtain a unique deductive logic theory where we integrate the ordinary database relations with also temporal/probability event-based relations. Another advantage is that we can also define a logic theory for integration of different TP-mixed-databases, based on views, as in ordinary GLAV (Global and Local As View) Data Integration Systems. Based on the set of Herbrand models for such logic programs, we define the event-based modal logic language and the set of Kripke models which define the semantics for it. Such modal language may be seen as an epistemic extension of Temporal logics, and can be used as basis for definition of the high-level SQL-like query language for TP-databases also.

The future work will be dedicated to explore such logic SQL-like query languages for TP-databases, which correctly implements the underlying denotational semantics of the ‘implicit’ TP-algebra.

References

The Herbrand model predicate is \( r \) with a radius \( d \leq i \) as \( y \). In this paper, we will hide the uncertain information of a TP-database, so that we define this set become hidden attributes: their values will be called parameters. For example, in this abstract such database, denoted by \( DB^\text{abs} \), in the way that these common attributes become hidden attributes: their values will be called parameters. For example, in this paper we will hide the uncertain information of a TP-database, so that we define this set as \( y = \{ \mathbf{t}, L, U, p \} \).

The abstraction of a relational schema (predicate) \( r_Y(x, y) \in DB \), where \( x = \{ x_i | 1 \leq i \leq n \} \) is the set of visible attributes, generates the new predicate \( r(x) = \lambda x_r(x, y) \) in the relational schema of abstracted database, i.e., \( r(x) \in DB^\text{abs}(y) \).

Notice that \( r(x) \) is not a simple projection of the original predicate over attributes in \( x \), that is, \( r(x) \neq \pi_x r_Y(x, y) \), because it contains also all hidden parameters.

**Example A:** Let us consider for example a logic theory with a single predicate "sphere with a radius \( d \) and the center in \((x_0, y_0, z_0)\)"; that is, \( r(y(x, y, d, x_0, y_0, z_0)) \) where \( x, y, z \) are variables and \( d, x_0, y_0, z_0 \) are constants, and consider that we have only 2-dimensional device for a visualization (the third dimension \( z \) is hidden), so that we obtain the abstracted predicate \( r(x, y, d, x_0, y_0) = \lambda z, z_0, r_Y(x, y, z, d, x_0, y_0, z_0) \). Such predicate is "circles with a radius less than \( d \) and the center in \((x_0, y_0)\)"; each value for a hidden parameter \(-d \leq z - z_0 \leq d \) produces a particular circle, and their totality (all of them together) corresponds to the original sphere concept \( r_Y \).

Notice that the true of a (single) abstracted atom \( r(x, y, d, x_0, y_0) \) is not enable to define the concept of sphere, so that we need some more complex formula, based on this abstracted atom, to define it. In what follows we will demonstrate that such formula must be a modal logic formula: "it is necessary that \( r(x, y, d, x_0, y_0) \) is a circle for all hidden parameter values \(-d \leq z - z_0 \leq d \)."

Let us consider now the Herbrand interpretations in these two cases: for the "normal" predicate \( r_Y \) and the predicate \( r \) obtained by abstraction (we hide the attribute \( z \)).

The Herbrand model \( I : H_B \rightarrow 2 \) for the sphere with the fixed center \((x_0, y_0, z_0)\) and (constant) radius \( d \) is the following subset of the Herbrand base for this predicate (The domain for its attributes is the set of real numbers \( \mathcal{R} \)),

\[
M_B = \{ r_Y(x_k, y_k, z_k, d, x_0, y_0, z_0) | x_k, y_k, z_k \in \mathcal{R}, (x_k - x_0)^2 + (y_k - y_0)^2 + (z_k - z_0)^2 = d^2 \},
\]

that is

\[
I(r_Y(x_k, y_k, z_k, d, x_0, y_0, z_0)) = 1 \iff (x_k - x_0)^2 + (y_k - y_0)^2 + (z_k - z_0)^2 = d^2.
\]

The same model for the predicate \( r \) which abstracts the sphere predicate \( r_Y \) in this

case is the higher-order Herbrand interpretation $I_{abs}: H_B^{abs} \rightarrow 2^W$, where $H_B^{abs}$ is the Herbrand base for the abstracted predicate $r$, and $W$ is a set of reals (for values of the hidden coordinate $z$ of the sphere), such that, the logical value for a ground atom $r(x_k, y_k, d, x_0, y_0, z_0)$ is the following function (that is higher-order logic value) $I_{abs}(r(x_k, y_k, d, x_0, y_0, z_0)) = f : W \rightarrow 2$, with the property that for a constant $z \in W$, 

$$f(z) = 1 \text{ iff } (x_k - x_0)^2 + (y_k - y_0)^2 + (z_k - z_0)^2 = d^2,$$

so that $I(r_Y(x_k, y_k, z_k, d, x_0, y_0, z_0)) = I_{abs}(r(x_k, y_k, d, x_0, y_0, z_0))(z_k)$, as can be verified from the commutative diagram in the next definition.

In what follows we will expose the general model transformation from Herbrand interpretations of an ordinary database $DB$ into non-Herbrand models of its abstracted version $DB_{abs}(y)$. Let us introduce now the interpretations for abstracted databases.

**Definition 5.** (Abstracted interpretation). Let $I : H_B \rightarrow 2$ be the 2-valued Herbrand interpretation for a parameterizable database $DB$ with the Herbrand base $H_B$ and the model $M_H = \{ r_Y(d, w) \mid r_Y(d, w) \in H_B \text{ and } I(r_Y(d, w)) = 1 \}$. Then the interpretation for its abstracted database $DB_{abs}(y)$ is defined by

$I_{abs} : H_B^{abs} \rightarrow 2^W$, such that $I_{abs} = [I \circ is]$, where $W = \text{Dom}_y \times \ldots \times \text{Dom}_y$, is the set of all parameter tuples, $H_B^{abs}$ is a Herbrand base for the abstracted database $DB_{abs}(y)$, i.e., $H_B^{abs} = \{ r(d) \mid r_Y(d, w) \in H_B \}$, $is : H_B^{abs} \times W \simeq H_B$ is a bijection, $[\lambda]$ is the currying ($\lambda$ abstraction) for functions, and $2^W$ is the set of functions from $W$ to $2 = \{0, 1\}$.

The following commutative diagram corresponds to the Abstracted Interpretation:

$$
\begin{array}{ccc}
2^W \times W & \xrightarrow{eval} & 2 \\
\downarrow I_{abs} = [I \circ is] & & \\
H_B^{abs} \times W & \xrightarrow{id_W} & H_B \\
\downarrow is & & \downarrow I \circ is \\
& H_B \\
\end{array}
$$

where $eval$ takes any function $f \in 2^W$ and a value $z \in W$, and returns with the value $f(z) \in 2$.

The abstracted interpretations are higher type of Herbrand interpretations: the set of truth values for them are functions instead of constants. We pass from a linear truth structure for atoms in a Hebrand interpretations of original database $DB$, to non linear functional space truth structure for atoms in the abstracted Herbrand base $H_B^{abs}$. The hidden parameters make curve the truth space for these atoms, as what happens when the real astronomic space curves with a presence of hidden gravitational mass.

No one of subsets $S \subseteq H_B^{abs}$ can be model for $DB_{abs}(y)$; that is, the models for abstracted database are not ordinary Herbrand models but some kind of higher type of Herbrand models. In what follows we will demonstrate that such higher Herbrand model types are Kripke models.

Let us define, for any ground atom $r(d)$ in $H_B^{abs}$ the set

$Q(d) = \{ r_Y(d, w) \mid r_Y(d, w) \in H_B \text{ and } I(r_Y(d, w)) = 1 \} \subseteq M_H$. 

Then the following bijection (isomorphism) hold:
\[ Q(d) \simeq \{ (r(d), w) \mid r_Y(d, w) \in H_B \text{ and } I(r_Y(d, w)) = 1 \} \]
\[ \simeq \{ r(d) \mid \{ w \mid r_Y(d, w) \in H_B \text{ and } I(r_Y(d, w)) = 1 \} \} \]
\[ \simeq \{ w \mid \{ r(d) \mid r_Y(d, w) \in H_B \text{ and } I(r_Y(d, w)) = 1 \} \} \]
\[ \simeq \{ w \in W, r(d) \in H_B \text{ and } I_{abs}(r(d))(w) = 1 \} \]

Thus, we can consider the set \( S_W(d) \) as a set of points \( w \) where the abstracted atom \( r(d) \) is true, that is the fact that \( I(r_Y(d, w)) = 1 \) is equivalent to the fact that \( r(d) \) is true in \( w \). So we pass from Herbrand models of original database \( DB \) to, equivalent to them, Kripke models of abstracted database \( DB_{abs}(y) \).

**Definition 6.** (Abstracted database models). Let \( I : H_B \to 2 \) be the 2-valued Herbrand interpretation for a parameterizable database \( DB \) with the Herbrand base \( H_B \) and the model \( M_H = \{ r_Y(d, w) \mid r_Y(d, w) \in H_B \text{ and } I(r_Y(d, w)) = 1 \} \).

Then \( M_I = (W, \{ R_d \mid \exists r_Y(d, w) \in M_H \}, S, V) \) is the Kripke model for abstracted database \( DB_{abs}(y) \), such that:

1. Accessibility relations: for any \( r(d) \) such that exists \( r_Y(d, w) \in M_H \)
\[ R_d = Q_W(d) \times Q_W(d) \]

2. \( S \) is a non empty set of constants of a database \( DB \).

3. \( V : W \times P \to \bigcup_{n<\omega} 2^{S^n} \), such that for any \( w \in W, r \in P \) where \( P \) is a set of all abstracted predicates in \( DB_{abs}(y) \), and \( d \in S^n \) holds:
\[ V(w, r)(d) = 1, \text{ if } I_{abs}(r(d))(w) = 1; \quad 0 \text{ otherwise.} \]

where \( S^n \) denotes the set of all \( n \)-tuples of constants, and \( 2^{S^n} \) the set of all functions from the set \( S^n \) to the set \( 2 \).

Notice that the binary accessibility relations are cartesian products, thus are reflexive, symmetric and transitive relations, so that we obtain the S5 multi-modal Kripke models.

Let denote by \( E_d \) the universal modal operator for any accessibility relation \( R_d \); for this S5 system it is alethic modal operator "it is necessary that". So that, the modal formula \( E_d r(d) \), for this abstract database \( DB_{abs}(y) \), means "\( r(d) \) is necessarily true".

**Example B:** Let us consider the constant \( d = (x_1, y_1, d, x_0, y_0) \) for the example A. Then the sentence "it is necessary that \( r(d) \)"., that is, "it is necessary that \( (x_1, y_1) \) belongs to the sphere \( r_Y\)", is the modal formula \( E_d r(x_1, y_1, d, x_0, y_0) \), with \( Q_W(d) = \{(z_1, z_0), (z_2, z_0)\} \), and \( z_i = z_0 \pm \sqrt{d^2 - (x_i - x_0)^2 - (y_i - y_0)^2} \), \( i = 1, 2 \).

\[ \square \]

**Proposition 4** Let \( M_I = (W, \{ R_d \mid \exists r_Y(d, w) \in M_H \}, S, V) \) be the Kripke model for abstracted database \( DB_{abs}(y) \) obtained from the Herbrand model \( M_H \) of the original database \( DB \). Then, for any \( d \) such that exists \( r_Y(d, w) \in M_H \), the modal formula \( E_d r(d) \) is true in the abstract database model \( M_I \).

**Proof:** Consider that holds \( M_I \models_w r(d) \) iff \( I(r_Y(d, w)) = 1 \) (from the point 3 of Def. 6), and the definition of \( R_d \).

\[ \square \]

**Remark:** In this paper, for TP-databases we reduced \( W \) to the set of time points, \( S_T \), because other parameters \( L_t, U_t \) and \( p \) are functionally dependent on it for a given event \( d \); for any \( t \) there exists at maximum one true ground atom \( r^T_{P_t}(d, t, L_t, U_t, p) \) in a database \( DB_{TP} \).