Representing text eventualities in a Logic Programming framework

Irene Pimenta Rodrigues and José Gabriel Lopes
CRIA/UNINOVA, Portugal
email: `ipr,gpl@fct.unl.pt`

Abstract

In this paper we present a language with a temporal ontology adequate for representing the facts retrieved from a text. The semantics of this representation language completes some of the information conveyed by text sentences in a suitable way for the pragmatic interpretation of text temporal entities. We also present the translation of the knowledge base language into a Logic Programming framework, the contradiction removal system [14].

1 Introduction

It is our goal to build a knowledge base containing the facts automatically extracted from descriptive texts. This poses several problems reported in [17, 18, 15, 16] and requires a methodological approach [21, 5, 9, 17, 15, 11]. In this paper we present a language for representing the facts conveyed by descriptive texts in a way that is suitable for the interpretation of each text sentence in the context made available by the interpretation of the previous text sentences. The proposed language has as basic temporal entities events, states, and time intervals and predicates are used to explicitly represent contingency relations [12] (such as: cause, consequence, etc) between eventualities (events or states). With this language we are able to handle state persistence and event and state prediction in the processes for temporal anaphora resolution which are well documented forms of temporal reasoning but in the works of [21, 9, 10, 2] they do not take it into account in the processes of temporal anaphora resolution.

The pragmatic interpretation of a text sentence in the context provided by the interpretation of the previous sentences, namely for temporal anaphora resolution, requires a set of referents earlier introduced which can be used as anaphoric referents [7, 13, 9, 10, 17, 15, 6].

Regarding the pragmatic interpretation process we follow Hobbs et al [5] proposal and we take the pragmatic interpretation as abduction. This means that the interpretation of a text is the minimal explanation of why the text would be true. So, the sentence tense and aspect interpretation is the explanation of why the main sentence eventuality would be true in the context of the previously acquired temporal facts. As a sentence eventuality is always a *state* or an *event* the explanation of the eventuality is always a set of assumptions that will allow the *prediction* or *explanation* of the eventuality from the context.

In order to have the interpretation context for each text sentence there must be a structure that will provide the visible entities (the more salient ones). In our approach [16, 15] we use the *text temporal structure*, i.e. we build discourse segments that reflect the temporal properties of their sentences. Each text segment has features for kind, time, eventuality, and subsegments. According to the kind of segment there subsegments obey to some temporal properties, and the feature time and event is computed from the features of its subsegments. Any sentence is represented by a segment. Its pragmatic interpretation will involve its introduction in the text segment. As the segmentation is based in the temporal properties of the segments there are some temporal relations that must be inferred in order to be able to introduce a new segment in the text segment.

The text temporal structure structure (the text segment) supplies the set of *visible entities* for the introduction of a new segment, and for the pragmatic interpretation of the next text sentence.

The theory for the pragmatic interpretation of sentences' tense is build using the visible facts from the current text temporal structure, and a set of rules that convey: general knowledge about eventualities, and linguistic knowledge about tense interpretation. The rules that convey linguistic knowledge should allow the inference of the more probable temporal relation between the main eventuality of the text sentence currently being interpreted and a visible eventuality in the text temporal structure.

In [17, 16] the sentence interpretation process is driven by the attempt to justify the sentence's main eventuality. So the tense interpretation theory, given an intermediate sentence semantic representation, gives rise to a set of abducted predicates in order to explain the sentence from knowledge acquired with the previous text sentences. However, the sentence tense is interpreted in the sentence context, i.e. only the information contained in visible segments of the text temporal structure is taken into account. So, some of the solutions for the tense interpretation can be contradictory with the information conveyed by previous text sentences that are not in visible segments. To eliminate these solutions we have to check if the the knowledge base with the information conveyed by the previous sentences updated with the new sentence representation is consistent. To perform this operation of testing for consistency in the knowledge base we shall implement the knowledge base in a logic programming framework, contradiction removal system [14]. This implementation also allows us to check if an expression is a logic consequence of the knowledge base. This operation is important during the the pragmatic interpretation of sentences' tense, namely for testing the preconditions for the abduction of a contingency relations between two eventualities.

In this paper we first present the knowledge base temporal ontology, then we present the language and in order to give a model-theoretic semantics to the knowledge base language we define what an interpretation for the language is. After we present and motivate the axioms of the knowledge base language as conditions for an interpretation to be valid because the language has not enough power to represent those axioms. Then we present the semantics of the language, the preferred models of the knowledge base, these models are adequate for the temporal reasoning during the pragmatic interpretation of temporal entities and they have nice properties that allow us to implement the knowledge base in a logic programming framework.

In the next section we present the translation of the knowledge base into a logic programming framework, contradiction removal system whose semantics is equivalent to the models that we associate with the expressions of the proposed language, the preferred models.

2 Representing text temporal facts

The temporal ontology of the knowledge base with the text facts should be as close to natural language temporal concepts as possible, in order to facilitate the temporal reasoning necessary for the pragmatic interpretation of the temporal entities.

States and events introduced by natural language expressions are represented by discourse entities to simplify the process of anaphora resolution. States and events are always associated with a temporal preposition (the event or state type) and to a time interval. In order to eliminate the proliferation of assertions about states and events, the time interval associated with these entities is always a maximal interval, i.e. the maximal interval where the event was occurring or the maximal interval where the state was holding.

In the knowledge base language we consider the following temporal entities: events, states and time intervals:

- *Time intervals* . The temporal relations between time intervals are $\{<, >, \subset, \supset, \propto\}$[1].
- *Events.* An event E_1 is associated with its occurrence time T_1 by the predicate: *occurs(*E_1*,*T_1*).*
  An event E_1 is associated with a temporal proposition R (the event type) by the predicate: *evt(*E_1*,*R*).*
  For ex.: an event of type "John shot at goose1" that occurred at a time before *now* is represented by:
  $\exists e_1, t_1$: occurs(e_1, t_1), evt(e_1,shoot(John,goose1)), $t_1 < now$.
- *States.* A state S_1 is related to its time T_1 by the predicate: *holds(*S_1*,*T_1*).*
  A state S_1 is associated with a temporal proposition (the *state* type) by the predicate: *evt(*S_1*,*R*).*
  To represent a state of type R holding at a time interval T, we use the predicate: *holds_at(*T, R*).*
  For instance the fact that "John was cooking goose1" is represented by the expression:
  $\exists s_1, t_1$: holds(s_1, t_1), evt(s_1, cook(John,goose1)), $t_1 < now$.
  Where t_1 must be a maximal interval for proposition "cook(John,goose1)".

2.1 The language for representing the text facts

We first present the language syntax and subsequently define an interpretation for expressions of the language.

The alphabet of the knowledge base language is:

• $T_c = \{t_1, t_2, \ldots\}$, $E_c = \{e_1, e_2, \ldots\}$, $S_c = \{s_1, s_2, \ldots\}$ e $U_c = \{u_1, u_2, \ldots\}$ are sets of discourse referents of *time intervals, events, states* and other *individuals*, respectively.

• The set of temporal predicate signatures:

[1] $t_i \propto t_j$ means that the intersection of the two time intervals is nonempty

$P_t = \{p^1_{t_1}, p^1_{t_2}, \ldots, \ldots, p^n_{t_1}, p^n_{t_2} \ldots \}$

P_t is the set of "event and state types", i.e. a set of propositions that need to be associated with a time interval, in order to be interpreted.

• The set of atemporal predicate signatures:

$P_a = \{p^1_{a_1}, p^1_{a_2}, \ldots, \ldots, p^n_{a_1}, p^n_{a_2} \ldots \}$

E.g. this set includes the following predicate symbols whose semantics is presented later:

– $\{occurs^2, holds^2\}$.

– $R_t = \{ <^2 \; 8 >^2, \subset^2, \supset^2, \propto^2 \}$

– $\{cause^2, prep^2, consq^2\}$

– $\{=^2, \neq^2\}$

– $\{ gen\_cause^2, gen\_prep^2, gen\_consq^2 \}$

– $n\_punctual^2$

– $abnormal^2$

– $incompatible^2$

• $V_T = \{v_{t_1}, v_{t_2}, \ldots\}$, $V_E = \{v_{e_1}, v_{e_2}, \ldots\}$, $V_S = \{v_{s_1}, v_{t_2}, \ldots\}$, $V_U = \{v_{u_1}, v_{u_2}, \ldots\}$, are sets of variables for *time intervals, events, states* and other *individuals*, respectively.

• $holds\_at^2$

• evt^2

• the logic connectives: {⇒, ','[2]}.

With this alphabet, a well formed expression of the language is a set of expressions, *expr*, defined by the grammar:

• expr: Term

• expr: Term "⇒" Body

• Body: Term

• Body: Term "," Body

• Term: $p^n(l_1, \ldots l_n)$ with $p^n \in P_a \cup \{evt/2, holds\_at/2\}$ and $l_1, \ldots l_n \in T_C \cup T_E \cup T_S \cup T_U \cup V_T \cup V_E \cup V_S \cup V_U$

2.2 An Interpretation for the knowledge base language

An interpretation for the language of the knowledge base with facts from the text is a tuple $\langle \mathcal{E}, \mathcal{S}, \mathcal{I}, \mathcal{U}, F_P \rangle$ where:

- $\mathcal{E}$ is a set of events.
- $\mathcal{S}$ is a set of states.
- $\mathcal{U}$ is a set of time individuals.
- $\mathcal{I}$ is a set of time intervals.
- F_P a function such that:
  - if $p^n \in P_t$ then $F_P(p) \subset \{\langle e, u_1, \ldots u_n \rangle \,|\, e \in \mathcal{E} \cup \mathcal{S}, u_1, \ldots u_n \in \mathcal{U}\}$.
  - if $p^n \in P_a$ then $F_P(p) \subset \{\langle l_1, \ldots l_n \rangle \,|\, l_1, \ldots l_n \in \mathcal{I} \cup \mathcal{U} \cup \mathcal{E} \cup \mathcal{S}\}$.

[2] ',' represents conjunction.

Let I be a interpretation, with f a pre-interpretation and g a variable assignment. An expression E of the knowledge base language is true in I if:

E	Conditions for E to be true in I
$evt(a, p(a_1, \ldots, a_n))$	$\langle f(a), f(a_1), \ldots, f(a_n))\rangle \in F_P(p)$
holds_at(p($a_1, \ldots, a_n$),b)	$\exists s \in S, \exists t \in T$: $\langle s, f(a_1), \ldots, f(a_n))\rangle \in F_P(p)$, $\langle s, t\rangle \in F_P(holds)$, $\langle t, f(b)\rangle \in F_P(\supset)$
$a = b$	$f(a) = f(b)$
$a \neq b$	$f(a) \neq f(b)$
$p(a_1, \ldots, a_n)$, with p^n a predicate different from: $\{evt,\ holds\_at,\ =,\ \neq\}$	$\langle$f(a_1), ..., f(a_n)$\rangle \in F_P(p^n)$
$\forall x Exp$	if $x \in \tau_v$, for all $c \in \tau$ Exp is true in I with $g(x) = c$. $(\tau_v, \tau) \in \{(T_v, T), (E_v, E), (S_v, S), (U_v, U)\}$

Conjunction and implication have the usual meaning.

2.3 Axioms of a theory of action and change for representing text facts

In order to ensure that the time intervals associated to events and states are maximal intervals we require that valid interpretations satisfy the axioms:

- axiom 1: $\forall E \in \mathcal{E}, \exists! I \in \mathcal{I} : \langle E, I\rangle \in F_P(occurs)$
- axiom 2: $\forall S \in \mathcal{S}, \exists! I \in \mathcal{I} : \langle S, I\rangle \in F_P(holds)$

These two axioms require that every event or state will have one and only one time interval associated with it. These conditions are important for not having a proliferation of states, events and times.

Next we present more axioms in order to model adequately: the events and states, the structure of events and states due to world knowledge about event and state types, and times.

2.3.1 Events and States

In natural language, an eventuality should be represented as an event or a state depending on the verb aspect and on the temporal modifiers. For instance, the sentence (1) gives rise to an eventuality that should be represented as a state.

The plane was landing at Faro airport. (1)

The sentence (2) should be represented as an event.

$$\text{The plane landed at Faro airport.} \tag{2}$$

By knowing that the event of landing happened, we known that a state where the plane was landing also exists. So for every event[3] there is a state with the same proposition associated. This axiom will enable us to solve some temporal anaphora that refer to the preparatory phase of an event already introduced in text. For instance in the text (3) the second sentence's subordinate clause refers to the preparatory phase of the event introduced in the first sentence.

Sunday, a plane landed at Faro airport (1). When the plane was landing the pilot sneezed (2). (3)

So, every valid interpretation for the knowledge base language must satisfy axiom 3:

- axiom 3 : $\forall e \in \mathcal{E}\ \exists s \in \mathcal{S}\ \exists p \in P_t, \exists u_1, \ldots, u_n \in \mathcal{U}, \exists t_e, t_s \in \mathcal{I}$:

$$\left.\begin{array}{r} \langle e, t_e\rangle \in F_P(occurs) \\ \langle e, u_1, \ldots u_n\rangle \in F_P(p) \\ \langle e\rangle \in F_P(n\_punctual) \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \langle s, t_s\rangle \in F_P(holds) \\ \langle s, u_1, \ldots u_n\rangle \in F_P(p), \\ \langle t_s, t_e\rangle \in F_P(\subset) \end{array}\right.$$

By knowing that a state of landing existed we should assume that the landing was completed successfully, i.e. that there exists an event of landing, unless we know that the state of landing was abnormal.

This axiom is important to deal with the imperfective paradox, i.e. sometimes a state does not end successfully. In the text (4) the state "crossing the street" is ended abruptly without success.

John was crossing the street (1).A truck ran him over (2). (4)

The assumption that every state ends successfully unless we know that it is abnormal is a useful assumption to complete the information conveyed by a text and is important to solve temporal anaphora.

Mary arrived home at 12 o'clock (1). John was writing a letter (2).
She cooked a meal (2) and they ate it (3). She went out at 3 pm (4) (5)
and mailed John's letter (5).

In the text (5) the "letter" in sentence 5 should be the letter that John was writing (sentence 2). So, we must assume that John has successfully finished his writing.

An interpretation for the knowledge base language must also satisfy axiom 4:

- axiom 4 : $\forall s \in \mathcal{S}\ \exists e \in \mathcal{E}\ \exists p \in P_t, \exists u_1, \ldots, u_n \in \mathcal{U}, \exists t_e, t_s \in \mathcal{I}$:

$$\left.\begin{array}{r} \langle s, t_s\rangle \in F_P(holds) \\ \langle s, u_1, \ldots u_n\rangle \in F_P(p) \\ \langle s\rangle \notin F_P(abnormal) \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \langle e, t_e\rangle \in F_P(occurs) \\ \langle e, u_1, \ldots u_n\rangle \in F_P(p), \\ \langle t_s, t_e\rangle \in F_P(\subset) \end{array}\right.$$

[3]Except for punctual events, i.e. events that are instantaneous.

2.3.2 The events structure

In the interpretation of a sentences' tense and aspect it is important to consider the possible relations between eventualities (e.g. cause, sub-action and consequence). Texts (6) to (8) have the same syntactic form (sentence tenses are simple past) but they may have different temporal interpretations:

Mary fell. John pushed her. (6)

Mary fell. John helped her. (7)

John cooked a salmon. He cut it in small pieces. (8)

Text (6) can have the following interpretations:

1. "John pushing" happens before "Mary's fall" only if we assume that the "pushing" caused the "fall" [21, 9].
   To represent that that eventuality e_j is caused by eventuality e_i we use the notation: cause(e_i, e_j).
2. "John pushing" happens after "Mary's fall" [13, 7, 21, 9].

Text (7) can be interpreted as:

1. "John helping" happens after "Mary's fall".

Text (8) can be interpreted as:

1. "John cutting" happens during "John cooking" only if we assume that "John cooked the salmon by cutting it in small pieces", i.e "John cutting" is a sub-action or is in the preparatory phase of "John cooking" [12, 21, 9].
   To represent that that eventuality e_j is a sub-action (or is in the preparatory phase) of the eventuality e_i we use the notation: prep(e_i, e_j).
2. "John cutting" happens after "John cooking".

Texts (9) and (10) also have the same syntactic form (first sentence in the simple past and second sentence progressive) but different temporal interpretations are possible.

John shot at a goose. The goose was dead. (9)

John saw a goose. The goose was dead. (10)

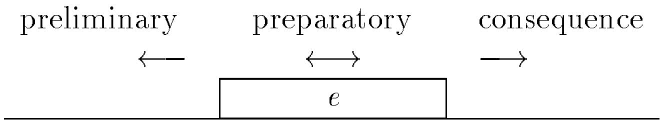
Text (9) can be interpreted as:

1. "the goose died" after "John shot at it" only if "John shoot" has "the death of the goose" as a consequence [4, 9].
   To represent that eventuality e_i has eventuality e_j as consequence we use the notation: consq(e_i, e_j).
2. "the goose was already dead" before "John shot at it" [13, 7, 4].

Text (10) can be interpreted as:

1. “the goose was already dead” before “John saw it” [13, 7, 4].

So, to take into account the possible relations between the eventualities, we assume that generally an event, e, has the following structure:



Due to the qualification problem, in general, knowledge about events and state types is defeasible. We represent the event/state structure by explicit predicates such as:

- cause(e_i,e_j) to represent that event e_i has the event e_j in its preliminary phase.
- prep(e_i,e_j) to represent that event e_i has the event e_j in its preparatory phase.
- consq(e_i,e_j) to represent that event e_i has state e_j as a consequence.

By explicitly representing the relations between eventualities, a system can make commitments with some structure of the events that is relevant during the pragmatic interpretation of temporal entities.

To represent general knowledge about the event structure we use the predicates:

- gen_cause(e_i,e_j) to represent that generally an event e_i has the event e_j in its preliminary phase.
- gen_prep(e_i,e_j) to represent that generally an event e_i has the event e_j in its preparatory phase.
- gen_consq(e_i,e_j) to represent that generally an event e_i has state e_j as a consequence.

We also have rules about event and state types, such as the one below, for representing knowledge about the consequences of a “shooting at” event[4]. i.e. generally a event of type “shooting at some animal with a gun” can have a state of type “the animal is dead” as consequence if “the gun was loaded at the time of the shooting”.

$$\begin{aligned} gen\_consq(v_{e_1}, v_{e_2}) \Leftarrow\ & evt(v_{e_1}, shoot\_at(v_{x_1}, v_{x_2}, v_{x_3})) \wedge \\ & evt(v_{e_2}, dead(v_{x_4})) \wedge \\ & v_{x_2} = vx_4 \ \wedge occurs(v_{e_1}, v_{t_1}) \wedge \\ & holds(v_{e_2}, v_{t_2}) \ \wedge v_{t_1} < v_{t_2} \wedge \\ & holds\_at(v_{t_1}, loaded(v_{x_3})) \end{aligned}$$

The predicates that represent general knowledge about the events structure are necessary conditions for supporting the relations between the eventualities, i.e. we can relate an event e_i and a state e_j through consq(e_i, e_j) if and only if gen_consq(e_i, e_j) is true for those eventualities.

So, a valid interpretation for the knowledge base language must also verify the axioms:

[4]Note that the predicate gen_consq has a role similar to the predicate *initiates* of the *event calculus* [8, 19]

- axiom 5: $\forall e_i, e_j \in \mathcal{E}\, \langle e_i, e_j \rangle \in F_P(cause) \Rightarrow \langle e_i, e_j \rangle \in F_P(gen\_cause)$
- axiom 6: $\forall e_i, e_j \in \mathcal{E}\, \langle e_i, e_j \rangle \in F_P(prep) \Rightarrow \langle e_i, e_j \rangle \in F_P(gen\_prep)$
- axiom 7: $\forall e_i \in \mathcal{E}, \forall e_j \in \mathcal{S}\, \langle e_i, e_j \rangle \in F_P(consq) \Rightarrow \langle e_i, e_j \rangle \in F_P(gen\_consq)$

We also consider as "world knowledge" about eventualities the incompatibility between two eventualities, i.e. states that can not be true at overlapping intervals. For instance, "alive" and "dead" are two eventualities that can not overlap.

We use the predicate:

- incompatible(e_i, e_j) to represent that the eventuality e_i is incompatible with eventuality e_j. ex: incompatible(v_{e_i}, v_{e_j}) $\Leftarrow$ evt(v_{e_i}, alive(v_{x_1})), evt(v_{e_j}, dead(v_{x_2})),$v_{x_1} = v_{x_2}$

To take into account this information in the semantics for the knowledge base language the axiom 8 should be satisfied by a valid interpretation for the language:

- axiom 8: $\forall e_i, e_j \in \mathcal{S},\ \exists t_i, t_j \in \mathcal{I}$:
$$\begin{array}{lcl} \langle e_i, e_j \rangle \in F_P(incompatible) & \Rightarrow & \langle t_i, t_j \rangle \notin F_P(\propto) \\ \langle e_i, t_i \rangle \in F_P(holds),\ \langle e_j, t_j \rangle \in F_P(holds) & & \end{array}$$

with $t_1 \not\propto t_2 \Rightarrow t_i < t_j \vee t_i > t_j$.

We must consider yet another axiom, axiom 9, to ensure that in every valid interpretation for the language of the knowledge base the time intervals can be interpreted as intervals over real numbers.

- axiom 9: There exists a function T: $\mathcal{I} \rightarrow \mathcal{I}nt(\mathbb{R})$[5], such that :
$\forall t_1, t_2 \in \mathcal{I},\ \exists i_{1_i}, i_{1_f}, i_{2_i}, i_{2_f} \in \mathbb{R}$:
$T(t_1) = [i_{1_i}, i_{1_f}], T(t_2) = [i_{2_i}, i_{2_f}]$,
$(\langle t_1, t_2 \rangle \in F_R(<) \Leftrightarrow i_{1_f} < i_{2_i})$.
$(\langle t_1, t_2 \rangle \in F_R(>) \Leftrightarrow i_{2_f} < i_{1_i})$.
$(\langle t_1, t_2 \rangle \in F_R(\subset) \Leftrightarrow i_{2_i} \leq i_{1_i}, i_{1_f} \leq i_{2_f})$,
$(\langle t_1, t_2 \rangle \in F_R(\supset) \Leftrightarrow i_{1_i} \leq i_{2_i}, i_{2_f} \leq i_{1_f})$,
$(\langle t_1, t_2 \rangle \in F_R(\propto) \Leftrightarrow i_{1_i} \leq i_{2_f}, i_{2_i} \leq i_{1_f})$.

So, every valid interpretation must satisfy the 9 axioms presented and motivated in this section. Note that because the language of the knowledge base does not have enough expressive power to represent the axioms instead of writing these axioms as rules of the knowledge base we have presented the 9 axioms as conditions that the interpretation must satisfy .

[5]Reals numbers intervals.

2.4 The organization of the knowledge base containing the facts extracted from the text

This knowledge base has the semantic representation of the text sentences already processed, i.e. after the pragmatic interpretation of each sentence.

The knowledge with the text facts is organized into three theories:

• ST, the scenario theory, this theory has the facts conveyed by the text sentences already processed. Note that this theory is a set of expressions without variables (all entities are represented by constants) and without implications because these expressions are always the result of the translation or interpretation of natural language expressions[6]

John cooked a salmon (1). He cut it in small pieces (2). (11)

The scenario theory for the text (11) is the expression (12). Note that sentence 2 has already pronominal anaphora and temporal entities interpreted, i.e. $u_3 = u_1$, $u_4 = u_2$, $t_1 \supset t_2$, prep(e_1, e_2).

$$\begin{array}{ll} \text{sentence 1:} & \text{occurs}(e_1, t_1),\ \text{evt}(e_1,\ \text{cook}(u_1, u_2)), \\ & \text{John}(u_1),\ \text{salmon}(u_2), \\ \text{sentence 2:} & \text{occurs}(e_2, t_2),\ \text{evt}(e_2,\ \text{cut}(u_3, u_4)), \\ & u_3 = u_1,\ u_4 = u_2,\ t_1 \supset t_2,\ \text{prep}(e_1, e_2) \end{array} \tag{12}$$

• DT, the domain dependent theory, this theory has rules and facts to model world knowledge that is domain dependent.

Example:

To enable the interpretation of the text (11) this theory should have the rule (13).

$$\begin{array}{lll} \text{gen\_prep}(v_{e_1}, v_{e_2}) & \Leftarrow & \text{evt}(v_{e_1},\text{cook}(v_{u_1}, v_{u_2}) \\ & & \text{evt}(v_{e_2},\ \text{cut}(v_{u_3}, v_{u_4})) \\ & & \text{occurs}(v_{e_1}, v_{t_1}),\text{occurs}(v_{e_2}, v_{t_2}), \\ & & v_{u_3} = v_{u_1},\ v_{u_4} = v_{u_2},\ v_{t_1} \supset v_{t_2}. \end{array} \tag{13}$$

• IT, the domain independent theory, this theory has rules and facts to model world knowledge that is domain independent.

The only rules that this theory should have are the 9 axioms presented in the previous section. However, since the knowledge base language does not have enough power to represent them we shall consider that this theory is empty.

2.5 Semantics for the knowledge base language

Definition 2.1 *If an expression Exp is true in an valid interpretation M then M is a model of Exp and we write $M \models Exp$.*

In order to be able to implement the knowledge base it is convenient to define what are the minimal models of an expression, Exp. The minimal models are also adequate models

[6]We only consider text that have sentences that describe facts, i.e. we do not consider sentences that convey general rules or default rules.

to perform the reasoning for the pragmatic interpretation of temporal entities because they reduce the set of discourse entities to those that are introduced in text sentences or that are implied by text sentences.

Definition 2.2 *The minimal models of an expression are those that are minimal under the order defined by set inclusion ($\subset$).*

Note that given a pre-interpretation a knowledge base can have more then a minimal model, namely due to the temporal relations between the time intervals.

For instance, the expression (14) will have three minimal models due to axiom 9.

$$t_1 < t_2,\ t_2 \subset t_3 \tag{14}$$

The model M_1 in expression (15) is one of the three models.

$$\begin{array}{l} M_1 = \langle \{\}, \{\}, \{t_1, t_2, t_3\}, \{\}, F_{P_1} \rangle \text{, with:} \\ F_{P_1}(<) = \{\langle t_1, t_2 \rangle, \langle t_1, t_3 \rangle\} \\ F_{P_1}(>) = \{\langle t_2, t_1 \rangle, \langle t_3, t_1 \rangle\} \\ F_{P_1}(\subset) = \{\langle t_2, t_3 \rangle\} \\ F_{P_1}(\supset) = \{\langle t_3, t_2 \rangle\} \\ F_{P_1}(\propto) = \{\langle t_2, t_3 \rangle, \langle t_3, t_2 \rangle\} \\ F_{P_1}(P) = \{\}\ for\ P \notin \{<, >, \propto, \subset, \supset\} \end{array} \tag{15}$$

The minimal models are not adequate to be taken as models of the knowledge base with the facts extracted from a text because in those models can exist entities that are not supported by text facts. For instance, the expression (16) has a minimal model with a state that is not supported by text facts because in order to satisfy axiom 7 there must be a state where the gun is loaded.

$$\begin{array}{ll} \mathsf{ST}: & occurs(e_1, t_1), evt(e_1, shoot\_at(x_1, x_2, x_3)), gun(x_2), \\ & holds(e_2, t_2), evt(e_2, dead(x_4)), consq(e_1, e_2). \\ \mathsf{DT}: & gen\_consq(ve_1, v_{e_2}) \Leftarrow \; evt(v_{e_1}, shoot\_at(v_{x_1}, v_{x_2}, v_{x_3})) \wedge \\ & \qquad evt(v_{e_2}, dead(v_{x_4})) \wedge \\ & \qquad v_{x_2} = vx_4 \ \wedge occurs(v_{e_1}, v_{t_1}) \wedge \\ & \qquad holds(v_{e_2}, v_{t_2}) \ \wedge v_{t_1} < v_{t_2} \wedge \\ & \qquad holds\_at(v_{t_1}, loaded(v_{x_3})) \end{array} \tag{16}$$

So, we define the preferred models to be the knowledge base models.

Definition 2.3 *The preferred models of an expression are minimal models where every state and event is supported by a text fact, i.e. there exists a expression on ST introducing it or there is a rule in DT whose body is true in the model and whose head introduces it.*

With this definition the preferred models are supported models. The Well founded semantics for logic programs also associates a supported model to a logic program. In the next section we are going to represent the knowledge base in a logic programming system that assigns the Well founded semantics to the logic program, this seems an appropriate framework because both semantics assign supported models to the languages.

3 Implementing the knowledge base with the text facts

In order to implement the knowledge base we use current work in logic programming, namely, the system for contradiction removal of Pereira et al [14].

We have to translate our knowledge base theory into a logic program such that the models of the logic program are equivalent to the preferred models of the knowledge base.

Taking into account that the preferred models of the knowledge base normally are more then one we choose a logic programming system that is able to associate more then one model to each program, the system for contradiction removal [14].

3.1 Contradiction removal system

The contradiction removal system is a triple $\langle P, Rev, IC \rangle$, where:

- P is a logic program extended with explicit negation [3, 1].

Definition 3.1 *A program P extended with explicit negation is a set of rules:*

$L_0 \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$

with $n \geq m \geq 0$ and each L_i is an objective literal, i.e. an atom A or its explicit negation $\neg A$; not *is negation by failure.*

- Rev is a set of literals that can be revised.

Definition 3.2 *The set of revisables of a program P [14] is a subset of the default literals of the form:* not L*, where L is an objective literal such that there is no rule in P with L in the head of the rule, i.e.* not L *is always true by CWA (closed world assumption).*

- IC is a set of integrity constraints

Definition 3.3 *A set of integrity constraints [14] is a set of rules:*

$\leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_n$

with $n \geq m \geq 0$ and each L_i is an objective literal,

Definition 3.4 *[1] A program P is contradictory if and only the program does not have an Well Founded model for eXplicit negation, WFX.*

Definition 3.5 *[14] $\langle P, Rev, IC \rangle$ is contradictory if and only if:*

- *either P is contradictory,*
- *or exists an integrity constraint in IC with the form:*

$\leftarrow L_1, \ldots, L_n$

such that $\{L_1, \ldots, L_n\} \in$ WFX(P)

Definition 3.6 *[14] S is a revision of $\langle P, Rev, IC \rangle$ if and only if:*

- *$S \in Rev$*
- *and $\langle P \cup S, Rev, IC \rangle$ is not contradictory.*

Definition 3.7 *[14] S is a minimal revision of $\langle P, Rev, IC\rangle$ if and only if:*
- *S is a revision of $\langle P, Rev, IC\rangle$*
- *and there is no revision S' of $\langle P, Rev, IC\rangle$ such that $S' \subset S$.*

3.2 Translation of the knowledge base theories

We implement the knowledge base with the text facts in order to be able to know:

• If the knowledge base is consistent, i.e if there exists any preferred model of the knowledge base theories. This is an operation that is very important during the pragmatic interpretation of text entities because the pragmatic interpretation is performed in the interpretation context of each entity which can give rise to interpretations that contradict the information conveyed by previous sentences that are not in the interpretation context of the entity.

• To check if an expression is a logic consequence of the knowledge base, i.e. if any of the knowledge base preferred models is a model of the expression. This operation is important during the pragmatic interpretation of temporal entities in order to take into account the information conveyed by previous text sentences.

The translation of the knowledge base theories into a logic program is immediate because the knowledge base language is a subset of logic programs syntax.

The main difficulty is the translation of the nine axioms that a valid interpretation of the knowledge base must satisfies. In order to ensure that the semantics of the revised logic program satisfy the those axioms we have to represent those axioms by integrity constraints of the contradiction removal system.

Another problem that we have in the translation of the knowledge base theories into a contradiction removal system is related to the choice of the literals that can be revised.

3.2.1 The set of revisable literals

We consider as revisable the literals that result from expressions that can have a arbitrary interpretation. Those literals are:

• "$E_1 = E_2$", the equality of two discourse referents can be made by an pre-interpretation even when there is no expression that requires that they are equal.

• "$T_1\ R\ T_2$", with $R \in \{<, >, \subset, \supset, \propto\}$, a temporal relation between two temporal entities can be satisfied in a preferred model of an expression even when that relation is not in the expression.

As the literals that can be revised can not appear at the head of a rule we declare the literals of the form: $\{A_1 =' A_2,\ A_1 <' A_2,\ A_1 >' A_2,\ A_1 \subset' A_2,\ A_1 \supset' A_2, A_1 \propto' A_2\}$ as revisable literals and we add the following rules to the program P: $\{\ A_1 = A_2 \leftarrow A_1 =' A_2;$

$A_1 < A_2 \leftarrow A_1 <' A_2;\ A_1 > A_2 \leftarrow A_1 >' A_2;\ A_1 \subset A_2 \leftarrow A_1 \subset' A_2;\ A_1 \supset A_2 \leftarrow A_1 \supset' A_2;\ A_1 \propto A_2 \leftarrow A_1 \propto A_2\}$

3.2.2 The integrity constraints

The integrity constraints that we have to consider are the translation of the axioms that an interpretation must satisfy in order to be valid.

The following axioms are translated by the following integrity constraints:

- Axiom 1 $\leftarrow occurs(E_1, T_1), occurs(E_2, T_2), E_1 = E_2, not\ T_1 = T_2$
- Axiom 2 $\leftarrow holds(E_1, T_1), holds(E_2, T_2), E_1 = E_2, not\ T_1 = T_2$
- Axiom 3 $\leftarrow occurs(E_1, T_1), holds(E_1, s(E_1)), not\ s(E_1) \subset T_1$
  However the logic program must have the rule:
  $holds(E_1, s(E_1)) \leftarrow occurs(E_1, T_1), n\_punctual(E_1)$
- Axiom 4 $\leftarrow holds(E_1, T_1), occurs(E_1, e(E_1)), not\ e(E_1) \supset T_1$
  However the logic program must have the rule:
  $occurs(E_1, e(E_1)) \leftarrow holds(E_1, T_1), not\ abnormal(E_1)$
- Axiom 5 $\leftarrow cause(E_1, E_2), not\ gen\_{c}ause(E_1, E_2)$
- Axiom 6 $\leftarrow prep(E_1, E_2), not\ gen\_prep(E_1, E_2)$
- Axiom 7 $\leftarrow consq(E_1, E_2), not\ gen\_consq(E_1, E_2)$
- Axiom 8 $\leftarrow \neg T_1 \propto T_2, not\ T_1 < T_2, not\ T_1 > T_2$
  However the logic program must have the rule:
  $\neg T_1 \propto T_2 \leftarrow holds(S_1, T_1), holds(S_2, T_2), incompatible(S_1, S_2)$
- Axiom 9 will be ensured by collecting all the temporal relations in the well founded model and then using a path consistency algorithm to check if they are consistent.

Note that it will be easy to collect all temporal relations from the WFX of the program P if we do not have any rule with a nonempty body in P with a temporal relation on the head nor the negation as failure of a temporal relation in the body of a rule of P. This way all the temporal relations in the WFX of the revised P are either on P in rules without a body or in S (a minimal revision of P).

3.2.3 Relation between the preferred models of the knowledge base and the contradiction removal system

Definition 3.8 *A knowledge base has a preferred model if there exists a minimal revision S of $\langle P, IC, Rev\rangle$ and the temporal relations in ST and in S are path consistent*[7]

Definition 3.9 *An expression G is a logic consequence of the knowledge base if $Trad(G)$ is the translation of G into a logic program and there is an minimal revision S of: $\langle P, IC \cup \{\leftarrow notTrad(G)\}, Rev\rangle$ and the temporal relations in ST and in S are path consistent.*

These two definitions are two theorems (they show that the translation is complete) that have been proved to be true but whose proof we do not present here due to lack of space.

[7]Path consistent [20] is an algorithm that allows us to check if a set of constraints is consistent when those constraints are build with a set of relations over intervals that can be converted into relations between their end points.

4 Conclusions

In this paper we have presented a language that is adequate to represent facts extracted from text sentences. This language is suitable to perform the temporal reasoning necessary to the pragmatic interpretation of text entities. The language and its semantics is similar to Kamp's Discourse representation Theory [6]. They differ in the semantics given to the states and events and the proposed language can be considered as an extension to the discourse representation theory. The main advantages of the proposed language are:

- It allows the representation of discourse entities, namely states and events as discourses referents. This is important for the pragmatic interpretation of temporal entities because it allows the coreference of events and states.
- It has a semantics for which it is possible to use current work in logic programming, namely, in nonmonotonic reasoning [14]. This is possible, namely, because we assign a semantics to the language that is *supported* and the semantics of the used logic programming framework, contradiction removal system [14], also is *supported* (Well Founded semantics for programs with eXplicit negation); this permits us to make a translation of the proposed language into the logic programming framework that is complete.

With the proposed language it is possible to represent defeasible knowledge about entities (including events and states) because we are translating the language into a system [14] for which it is easy to express that kind of knowledge.

And, we have demos (theorem provers) for the contradiction removal system that allow us to implement the operations on the knowledge base with the facts described by a text.

- It has a semantics that is adequate for the temporal reasoning necessary during the pragmatic interpretation of text entities and allows the system to check if the interpretation of an entity is consistent with the previously interpreted text.

This knowledge base makes possible to perform the pragmatic interpretation of temporal entities as abduction in the interpretation context of each entity because it provides a way for checking the consistency of the facts already extracted from the text.

Just for the reader to have an idea of the complexity of the consistency checking of the facts extracted from a text we present times (in seconds) for the processing of the text (17).

John cooked a salmon (1). He cut it in small peaces (2). He ate it (3). His finger was aching (4). He went to hospital (5). He had inserted a bone in his finger (5). (17)

The table below presents the times for the interpretations of each text sentence. The second column has the time for the pragmatic interpretation of each sentence, note that the pragmatic interpretation is done in the sentence interpretation context, only a small set of text entities is visible for the interpretation of each sentence so the time for the interpretation of the first sentence is similar to the one for the interpretation of the others sentences. This is one of the main advantages of doing the interpretation as abduction in the interpretation context of each sentence. The column 3 has the times for checking the consistency of the knowledge base with all the text facts. Note that the times for this operation growths as the text growths in facts. The column 4 has the number of interpretations that are calculated for a sentence by the process of pragmatic interpretation of the temporal entities, some of them may be inconsistent with the facts of the previous sentences. The last column has the number of valid interpretations (those that are consistent with the previous sentences) for each sentence.

sentence	pragmatic interpretation	check consistency	number of int calculated	number of int valid
2	.76	1.2	2	2
3	1.8	5.7	4	2
4	.48	17.8	4	2
5	.75	44.6	6	4
6	1.8	93.1	12	6

Table with the times for text sentences processing (in seconds).

References

[1] José Júlio Alferes. *Semantics of Logic Programs with Explicit Negation.* PhD thesis, Universidade Nova de Lisboa, October 1993.

[2] Kurt Eberle. On representing the temporal structure of a natural language text. In *Proceedings of the COLING'92*, pages 288–294, 1992.

[3] Michael Gelfond and Vladimir Lifschitz. Logic programs with classical negation. In David H. D. Warren and Peter Szeredi, editors, *Proceedings of the Seventh International Conference on Logic Programming*, pages 579–597, Jerusalem, 1990. The MIT Press.

[4] Erhard Hinrichs. Tense,quantifiers, and contexts. *Computational Linguistics*, 14(2):3–14, June 1988.

[5] Jerry Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. Interpretation as abduction. Technical Report 499, SRI International, December 1990.

[6] Hans Kamp and Uwe Reyle. *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Dordrecht: D. Reidel., 1993.

[7] Hans Kamp and Christian Rohrer. Tense in texts. In C. Bauerle, R. Schwarze and A. von Stechow, editors, *Use and Interpretation of Language.* de Gruyter, 1983.

[8] Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[9] Alex Lascarides and Nicholas Asher. Discourse relations and defeasible knowledge. In *Proceedings of the 29th Annual Meeting of ACL*, pages 55–62, 1991.

[10] Alex Lascarides, Nicholas Asher, and Jon Oberlander. Inferring discourse relations in context. In *Proceedings of the 30th Annual Meeting of ACL*, pages 55–62, 1992.

[11] Alex Lascarides and Jon Oberlander. Temporal connectives in a discourse context. In *Proceedings of the 6th European chapter of ACL, EACL'93*, 1993.

[12] Mark Moens and Mark Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28, June 1988.

[13] Barbara H. Partee. Nominal and temporal anaphora. *Linguistics and Phylosophy*, 7:243–286, 1984.

[14] L.M. Pereira, C. Damásio, and J.J. Alferes. Diagnosis and debugging as contradiction removal. In Luis Moniz Pereira and Anil Nerode, editors, *Proceedings of the Second International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 316–330. The MIT Press, 1993.

[15] Irene Pimenta Rodrigues and José Gabriel Lopes. Building the text temporal structure. In *Progress in Artificial Intelligence: 6th Portuguese Conference on AI*. Springer-Verlag, 1993.

[16] Irene Pimenta Rodrigues and José Gabriel Lopes. Text processing in a logic programming framework: The interpretation of tense and aspect. In *Proceedings of the fourth International Workshop on Natural Language Understanding and Logic Programming, NLULP4*, 1993.

[17] Irene Pimenta Rodrigues and José Gabriel Pereira Lopes. Discourse temporal structure. In *Proceedings of the COLING'92*, 1992.

[18] Irene Pimenta Rodrigues and José Gabriel Pereira Lopes. A framework for text interpretation. In B. du Bouley and V. Sgurev, editors, *Artificial Intelligence V — methodology, systems and applications*, pages 181–190. North Holland, 1992.

[19] Murray Shanahan. Prediction is deduction but explanation is abduction. In *Proceedings of IJCAI 89*, pages 1055–1060. Morgan Kaufmann Publishers, Inc., 1989.

[20] M. Vilain and H. Kautz. Constraint propagation algorithms for temporal reasoning. In *Proceedings of the National Conference on Artificial Inteligence, AAAI*. Morgan Kaufmann Publishers Inc., 1986.

[21] Boonie Lynn Webber. Tense as discourse anaphor. *Computational Linguistics*, 14(2):61–73, June 1988.