Towards a Comparison Criteria for $\mathcal{C}DeLP$

María Laura Cobo* **Guillermo Ricardo Simari**
Artificial Intelligence Research and Development Laboratory (LIDIA)
Universidad Nacional del Sur
Bahía Blanca, B8000CPB, Argentina
[mlc,grs]@cs.uns.edu.ar

Abstract

The development of systems with the ability to reason about change notion and actions has been of great importance for the artificial intelligence community. The definition and implementation of systems capable of managing defeasible, incomplete, unreliable, or uncertain information has been also an area of much interest. With a few exceptions research on these two ways of reasoning was independently pursued. Nevertheless, they are complementary and closely related, since many applications that deal with defeasible information also depends on the occurrence of events and time.

$DeLP$ is an argumentative system appropriate for commonsense reasoning. The defeasible argumentation basis of $DeLP$ allows to build applications that deal with incomplete and contradictory information in dynamic domains. Thus, the resulting approach is suitable for representing agent's knowledge and for providing an argumentation based reasoning mechanism for that agent (see for example [6, 1]). It is interesting to extend this system adding mechanisms to manage events and time as $\mathcal{C}DeLP$ [7]. Here we analyze how to develop a comparison criteria for arguments built up from causal information and considers *commonsense rules of inertia*.

Key words: Argumentative Systems, Knowledge Representation, Defeasible Reasoning, Commonsense Reasoning, Temporal Reasoning, Reasoning about change notion and actions.

1 Introduction

The development of systems with the ability to reason about causal information (information inferred from events occurrence), has been of great importance for the artificial intelligence community. Research in this area has been interested on this issue as a way to find a solution

*Partially supported by Agencia de Promoción Científica y Tecnológica (PICT 13096, PICT 15043, PAV 2003 Nro. 076) and the Universidad Nacional del Sur

for a wider variety of problems where the occurrence of actions and the moment they occur makes a difference [9].

Argumentation Systems [5, 17], were developed in order to deal with incomplete or unreliable information. In real scenarios this situation is quite common, specially when we deal with dynamic systems, *i.e.*, systems where the knowledge available to reason with changes frequently (new information become available or information that used to be available became unavailable or invalid). Usually, since it is very difficult to represent all the information related to the objects under consideration, the information that appears as supporting our reasoning is incomplete. As a matter of fact, there are formalisms such as the *Situation Calculus* [20] where this problem is quite relevant. When new information about an entity becomes available, *i.e.*, knowledge changes, we must revise all the representation.

In order to improve on this problem we will follow an *argumentative* approach. $DeLP$ [8] is an argumentation system, which combines results from Logic Programming and Defeasible reasoning providing tools for knowledge representation and commonsense reasoning. We are interested in the development of an argumentative system based on $DeLP$ that can deal with causal information, a first approach leads us to basics definition of $\mathcal{C}DeLP$ [7]. This extension introduces the concept of causal information, *i.e.* information *events* and/or *time* dependant.

This work takes $\mathcal{C}DeLP$ basic definition and analyzes some difficulties in arguments comparison criteria, since $DeLP$ criteria is not proper enough to solve some new situations, particularly observable from inertia chains and actions occurrences.

In some literature actions and events are consider as different things, it is necessary then to clarify that in the context of the present work this terms are considered as synonyms.

The paper is structured as follows, on Section 2 a refresh of basic $\mathcal{C}DeLP$s definitions is made. Then a criteria for arguments comparison, taking in consideration only the strong information of each argument, is presented. Later on, on Section 4 we establish a way to compare looking only at inertial chains over arguments. On Section 5 we show some examples where difficulties to compare arguments from the amount and kind of actions, each argument uses, are emphasized. The last section presents the conclusions.

2 $\mathcal{C}DeLP$ Definition

In a previous work [7] we presented a new argumentation language based on a clausal language *Event Calculus* [10, 22, 14] called $\mathcal{C}DeLP$. The basics of this language depend on the main predicates of a simple version of *Event Calculus*, (*SEC*), which are:

$happens(E,T)$:	E takes place on T.
$holdsAt(F,T)$:	F holds at T.
$initiates(E,F,T)$:	F starts to hold after E, and is not freed on $T+1$.
$terminates(E,F,T)$:	F ceases to hold after E at T.
$releases(E,F,T)$:	F is not subject to inertia after E at T
$initiallyP(F)$:	F holds form time zero.

where E represents events, T time moments and F fluents. The calculus complete axiomatization depends on the chosen time ontology. For example if we consider a discrete ontology,

we can use the ontology presented by Mueller [15], or the more complete ontology from Miller and Shanahan research [14].

Once we have the syntax of the literals of the language, we can define it:

DEFINITION **1** ($\mathcal{C}DeLP$)

A $\mathcal{C}DeLP$ program is defined in terms of three disjoint sets: a set Π of *facts* and *strict rules*, a set Δ of *defeasible rules*, and a set Υ of *inertial defeasible rules*, where

- A *fact* is a literal, *i.e.*, a ground atom, o a negated ground atom.
- A *strict rule* is a rule denoted as "$Head \leftarrow Body$", where $Head$ is a literal and $Body$, is finite set of literals. A strict rule can also be written as: $L_0 \leftarrow L_1, \ldots, L_n (n > 0)$, where each $L_i, i \geq 0$ is a literal.
- A *defeasible rule* is a rule noted as $L_0 \prec L_1, \ldots, L_n$. Again L_i is a literal and $i \geq 0$
- A *inertial defeasible rule* is a *defeasible rule* that denotes some fluent inertia.

Consequently, this modification causes a change on the definition of argument:

DEFINITION **2** *($\mathcal{C}DeLP$ Causal Argument Structure)*

Let L be a literal and $\mathcal{P} = (\Pi, \Delta, \Upsilon)$ a $\mathcal{C}DeLP$ program. We say that $\langle \mathcal{A}, \mathcal{B}, L \rangle$ is a causal argument structure for L, if $\mathcal{A}$ is a set of defeasible rules from Δ and $\mathcal{B}$ is a set of defeasible rules form Υ, such that $\mathcal{A} \cup \mathcal{B}$ verifies $DeLP$ argument structure definition [8].

$$\Delta_1 = \left\{ \begin{array}{l} holdsAt(flies(X), T) \prec holdsAt(bird(X), T), \\ holdsAt(flies(X), T) \prec \sim holdsAt(injured(X), T), holdsAt(bird(X), T) \\ \sim holdsAt(flies(X), T) \prec holdsAt(injured(X), T) \\ \sim holdsAt(injured(X), T+5) \prec holdsAt(injured(X), T), \end{array} \right\}$$

$$\Upsilon_1 = \left\{ \begin{array}{l} holdsAt(injured(X), T+1) \prec holdsAt(injured(X), T), \\ \sim holdsAt(injured(X), T+1) \prec \sim holdsAt(injured(X), T), \end{array} \right\}$$

$\langle \mathcal{D}, \mathcal{I} \sim holdsAt(flies(tina), 5) \rangle$ is the proper definition of argument $\mathfrak{A}_1$ for fluent $\sim holdsAt(flies(tina), 5)$, where:

$$\mathcal{D} = \left\{ \sim holdsAt(flies(tina), 5) \prec holdsAt(injured(tina), 5), \right\}$$

$$\mathcal{I} = \left\{ \begin{array}{l} holdsAt(injured(tina), 5) \prec holdsAt(injured(tina), 4), \\ holdsAt(injured(tina), 4) \prec holdsAt(injured(tina), 3), \\ holdsAt(injured(tina), 3) \prec holdsAt(injured(tina), 2), \\ holdsAt(injured(tina), 2) \prec holdsAt(injured(tina), 1), \\ holdsAt(injured(tina), 1) \prec holdsAt(injured(tina), 0) \end{array} \right\}$$

In previous work [7] the comparison criteria among argument was pendant. In the following sections we will discuss several aspects the criteria should take on consideration. At the same time some previous definition to achieve this goal will be presented.

3 Comparing arguments from the base

In order to decide when an argument is better than other, we must design a comparison criteria that allow the proper choice of the wining argument. As a first step, this criteria should take on consideration the argument base, *i.e.* the subset of Π that origins the defeasible reasoning that takes place after that. This subset may be formed by fluents ($holdsAt(f,t)$ kind of formula) or sentences that defines events occurrence ($happens(a,t)$ kind of formula). This last kind of formula clearly represents actions that actually take place or are granted to happen through strong information use only.

Several aspects should be taken in consideration about the type of information that appear at the tree's base, for example we must evaluate if the same fluent, on different moments, appears in the base an root of the tree; or the amount of events the argument considers, as well as possible repetitions of the same event. At this point is necessary to point out that all the events considered are facts or can be proven only form information in the Π set of the program, this means that are always 'strong' information. This status of actions makes an argument with more actions more plausible.

The fact that the same fluent is in the base and the root may imply the use of an inertial rule in a direct way or that its state is a precondition to further gain of other fluents. As a first approximation lets consider the simple example that follows:

EXAMPLE **1** Let P_1 be a $CDeLP$ program defined as: $P_1 = (\Pi_1, \Delta_1, \Upsilon_1)$, where:

$$\Pi_1 = \left\{ \begin{array}{l} holdsAt(ligh\_off, 0) \\ happens(switch, 1) \\ happens(switch, 3) \\ happens(switch, 5) \\ holdsAt(ligh\_on, T) \leftarrow \sim holdsAt(light\_off, T) \\ holdsAt(ligh\_off, T) \leftarrow \sim holdsAt(light\_on, T) \end{array} \right\}$$

$$\Delta_1 = \left\{ \begin{array}{l} holdsAt(light\_on, T+1) \prec holdsAt(light\_off, T), happens(switch, T+1) \\ holdsAt(light\_off, T+1) \prec holdsAt(light\_on, T), happens(switch, T+1) \end{array} \right\}$$

$$\Upsilon_1 = \left\{ \begin{array}{l} holdsAt(light\_off, T+1) \multimap holdsAt(light\_off(X), T), \\ holdsAt(light\_on, T+1) \multimap holdsAt(light\_on(X), T), \end{array} \right\}$$

Form P_1 we can built the arguments shown in figure 1. In this example, we clearly see that

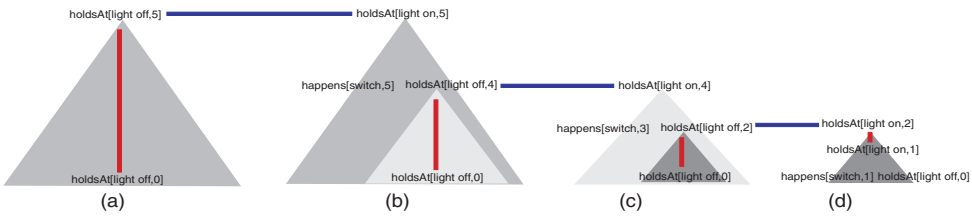

Figure 1: reasoning line

the first argument only appeals to the use of an inertial rule to warrant $holdsAt(light\_on, 5)$

(this fact makes this argument very weak in a discussion since it will be very easy to defeat); the other argument also uses an inertial rule, but the chain is shorter and it uses event's occurrence, so clearly is a better supporting argument than the first one.

If we take a look to the arguments against $holdsAt(light\_on, 5)$ we can see we count with two as well. In this case both arguments uses inertial rules and the happening of actions. We can also see that the event is the same, but since the happening of events is part of environment check (part of Π) the argument with more events is clearly more specific and more adjusted to reality.

Looking at the previous example we can draw definitions, in order to formalize the ideas shown there. One of them decide when two arguments are incomparable (from base), informally this situation takes place if we can not decide which one is better for the basic information they use to build up their conclusion. Formally this notion is captured on definition 3

DEFINITION **3** *(incomparable from base)*

Suppose we have arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ the set of fluents and $\mathcal{E}_1$ and $\mathcal{E}_2$ the set of events (both $\in \Pi$) such that are the basics of each argument. We state that the arguments are **incomparable from base** if:

$$\text{If } \mathcal{F}_1 \cap \mathcal{F}_2 = \emptyset \text{ and,} \qquad \text{If } \mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$$

As we explain before the amount of events that take place are important in order to determine the strength of an argument, particularly when the event that actually happens is the same over and over again, as in the example 1 where the action that happens is $switch$ obviously on different moments. We can consider events repetition as a sign of strength and define a criteria that considers this:

DEFINITION **4** *(repeated events)*

Suppose we are comparing two arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$, such that are not *incomparable*. And some action a such that $happens(a, T) \in \mathcal{E}_1 \cap \mathcal{E}_2$ if $happens(a, T_2) \in \mathcal{E}_1$ but $\notin \mathcal{E}_1 \cap \mathcal{E}_2$ with $T_2 \neq T$ Then $\mathfrak{A}_1$ is stronger than $\mathfrak{A}_2$

or we can extend the criteria not analyzing the kind of event but only the amount like in the following definition:

DEFINITION **5** *(preferring from the base)*

Suppose we are comparing two arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$, such that are not *incomparable*.

$\mathfrak{A}_1$ is going to be preferred over $\mathfrak{A}_2$ if the following condition holds:

1. $\mathfrak{A}_1$ uses more observed events than $\mathfrak{A}_2$. Formally: $|\mathcal{E}_1 - (\mathcal{E}_1 \cap \mathcal{E}_2)| = n$, $|\mathcal{E}_1 - (\mathcal{E}_1 \cap \mathcal{E}_2)| = m$ with $n > m$.

2. $\mathfrak{A}_1$ uses more observed fluents than $\mathfrak{A}_2$. Formally: $|\mathcal{F}_1 - (\mathcal{F}_1 \cap \mathcal{F}_2)| = n$, $|\mathcal{F}_1 - (\mathcal{F}_1 \cap \mathcal{E}_2)| = m$ with $n > m$.

Observed fluent or event are those that can be inferred only form information available in the Π set of the corresponding program.

4 Comparing arguments that uses inertia

If conflict appears through the use of inertial chains that are comparable, then the shorter one should be preferred since it has less points of attacks, making the argument stronger in certain sense.

In order to make an analysis over inertial chains, we must take in consideration the subset of Υ used in both arguments (the inertial rules used on the construction of the arguments). If the intersection of those sets is non empty then, both uses the same inertial rule. Being the case, we should prefer the argument with the shorter inertial chain for that fluent.

DEFINITION **6** *(inertial chains length)*

For some argument structure $\mathfrak{A} = \langle \delta, \upsilon, L \rangle$ and some literal l granted through an inertial chain. Then an inertial rule, $i \in \upsilon$, is used several times. i has the form $holdsAt(l,t) \multimap body$, where $body$ is a proper body for the rule. The length of the chain is evaluated like this:

Let T_{max} be the maximal time point of the inertial chain and let T_{min} be the minimal value of parameter t in the argument structure such that:

$$\forall m \; T_{min} < m < T_{max}[holdsAt(l,t) \multimap body \in \upsilon]$$

and there is no other time points $x > T_{max}, y < T_{min}$ that verifies the previous condition. Then the *length of the inertial chain* is $T_{max} - T_{min}$

DEFINITION **7** *(analyzing inertial chains)*

Suppose we are comparing two arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$, such that are not *incomparable*. Let $\mathfrak{A}_1 = \langle \delta_1, \upsilon_1, L \rangle$ and $\mathfrak{A}_2 = \langle \delta_2, \upsilon_2, L \rangle$. If $\upsilon_1 \cap \upsilon_2 \neq \emptyset$ (both arguments uses inertia over the same fluent) then, we should consider the following algorithm:

```
no argument is marked for each rule in the intersection do
while only one or none is marked do
  take the literal on the head of the rule
  get the length of inertial rule from both arguments
  compare lengths
  mark the argument with shorter inertial chain length
If both are marked then return blocking situation
  else return as preferred argument the marked one
```

Notice that can be the case that we can decide that neither are preferable since more than one inertial chains are applied and the shorter one is not in the same argument.

If we consider example 1 we can see that among the fact that the argument that ends to be preferred form the base (has more events), that same argument is preferred under definition above. Looking at figure 1 argument (a) has an inertial chain of length 5 for fluent $light\_off$, while the one for argument (b) is 4, 2 for argument (c) and finally 1 for argument (d).

Let us illustrate a blocking situation through a very well known example in frame problem literature, the *Yale Shooting Problem*. This problem presents a scenario where the necessity of the inertial rules on the specification of causal information is clear. In this sense we

are going to use Steve Hanks and Drew McDermott [9] problem to show a pair of arguments that are defeated by a mutual blocking status.

EXAMPLE **2** Let consider the following program $\mathcal{P}_2 = (\Pi_2, \Delta_2, \Upsilon_2)$, where:

$$\Pi_2 = \left\{ \begin{array}{l} initiallyP(alive(fred)) \\ initiallyP(moving(fred)) \\ happens(load(gun), 0) \\ happens(shoot(gun), 2) \\ \\ holdsAt(loaded(gun), T) \leftarrow happens(load(gun), T) \\ holdsAt(moving(X), T) \leftarrow happens(startMovement(X), T) \\ \sim holdsAt(moving(X), T) \leftarrow happens(endMovement(X), T) \\ \\ holdsAt(alive(X), T) \leftarrow holdsAt(alive(X), T_1), \sim happens(shoot(Y), T_1), T = T_1 + 1 \end{array} \right\}$$

$$\Delta_2 = \left\{ \begin{array}{rcl} \neg holdsAt(alive(X), T) & \prec & holdsAt(loaded(Y), T), happens(shoot(Y), T) \\ holdsAt(alive(X), T) & \prec & holdsAt(moving(X), T), happens(shoot(Y), T) \end{array} \right\}$$

$$\Upsilon_2 = \left\{ \begin{array}{l} holdsAt(loaded(Y), T+1) \multimap holdsAt(loaded(Y), T) \\ \sim holdsAt(loaded(Y), T+1) \multimap \sim holdsAt(loaded(Y), T) \\ holdsAt(moving(X), T+1) \multimap holdsAt(moving(X), T) \\ \sim holdsAt(moving(X), T+1) \multimap \sim holdsAt(moving(X), T) \end{array} \right\}$$

If we take a look to arguments in figure 2 we see that no choice among them is possible.

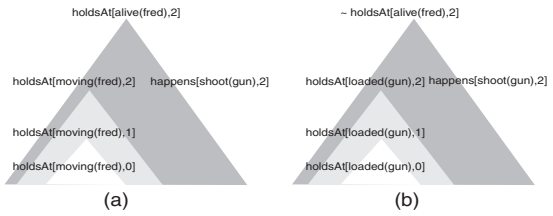

Figure 2: complementary arguments that leads to blocking defeat

Since both arguments are structurally identical, both uses a chain of inertial rules but over different fluents, independent fluents *i.e.* not comparable from the information available in the program.
If we eliminate $intiallyP(moving(fred))$ from set Π, and then we add $happens(startMovement(fred), 1)$ the arguments shown in figure 2 will look like this: In this case, the argument (a) has de shorter inertial chain than the argument shown in argument (b). If we decide to choose this measure as a way to avoid blocking then argument (a) should be preferred; but we can avoid the fact that this is the only fact that favors one of the arguments because looking at the base both sets are not comparable.

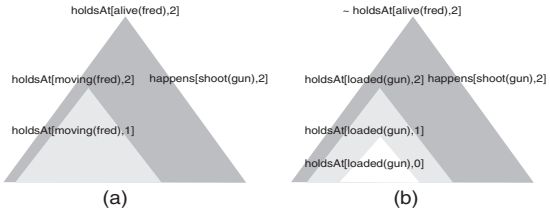

Figure 3: complementary arguments

5 Analyzing 'defeasible' events

There may events that are not registered in Π in a direct way, *i.e.* their real happening may depend on fluents that are granted in a defeasible way (*'defeasible' events*). If this is the case, we must analyze the supporting information for that event to take place, for granting the hole argument. If this is the case, the analysis of the base is not enough an we must take a look to the inner nodes of the tree in order to detect this kind of situation.

In general fluents are properties that holds on certain time point, *i.e.* properties that are either true or false at any time (*'strong' fluents*). With the introduction of defeasibility we lead to a situation where some fluents are *granted* or not. This means that we can build a defeasible proof for the fluent, since the defeasible nature of the proof a fluent granted in this way has less strength (from a knowledge perspective) than 'strong' fluents. So the following definition is needed:

DEFINITION **8** *Defeasible fluents*

Let $\mathcal{F}$ be a fluent. $\mathcal{F}$ is a defeasible fluent if its status is granted through an argument structure that can not be defeated.-

Let see an example where event occurrence depend on defeasible fluents:

EXAMPLE **3** Imagine an scenario where certain alarm clock must ring at some determine moment, only if it has enough battery. Here is the program, $\mathcal{P}_3 = (\Pi_3, \Delta_3, \Upsilon_3)$, where:

$$\Pi_3 = \left\{ \begin{array}{l} initiallyP(asleep(fred)) \\ initiallyP(enough\_battery) \\ happens(clock\_rings, T) \leftarrow holdsAt(enough\_battery, T) \\ \\ \sim holdsAt(asleep(fred), T) \leftarrow happens(clock\_rings, T) \end{array} \right\}$$

$$\Delta_3 = \left\{ \begin{array}{l} \sim holdsAt(enough\_battery, T) \prec \quad holdsAt(enough\_battery, T_1), T = T_1 + 10) \\ \end{array} \right\}$$

$$\Upsilon_3 = \left\{ \begin{array}{l} holdsAt(asleep(X), T+1) \multimap holdsAt(asleep(X), T) \\ \sim holdsAt(asleep(X), T+1) \multimap \sim holdsAt(asleep(X), T) \\ holdsAt(enough\_battery, T+1) \multimap holdsAt(enough\_battery, T) \end{array} \right\}$$

Taking a look to arguments, we can see that $clock\_rings$ depends on fluent $enough\_battery$ to take place. This fluent can or can not be granted in a defeasible way. Can be granted

through an inertial rule an can not by a defeasible rule that captures the notion of information's ageing (more details can be found at [7]). Clearly in this case the argument that uses the defeasible rule beats the one that uses de inertial rule. Situation like the one in figure 3

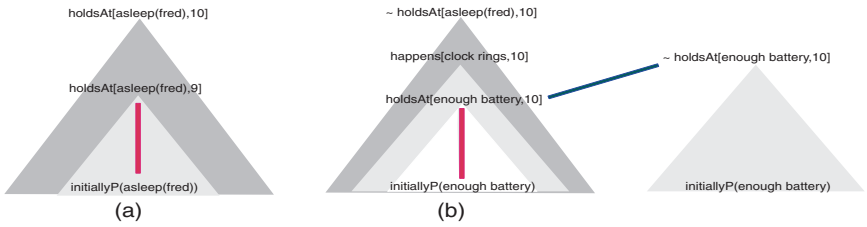

Figure 4: events that depend on defeasible fluents

are solved through the following definition:

DEFINITION **9** *(Inertia, Actions and Ageing)* Suppose we are comparing two arguments $\mathfrak{A}_1$ and $\mathfrak{A}_2$, and some fluent f such that $holdsAt(f,t)$ is granted on both arguments if:

1. there is a rule $i \in \mathfrak{A}_1$ such that forms an inertial chain that granted f at other time point T; and there is a rule $r \in \mathfrak{A}_2$ such that is an *information ageing rule* or an *event rule* such that grants $\sim f$ at the same T, or

2. there is a rule $a \in \mathfrak{A}_1$ such that is an *information ageing rule* such that grants f; and there is a rule $e \in \mathfrak{A}_2$ such that is an *event rule* such that grants f at the same T

then $\mathfrak{A}_2$ is preferred over $\mathfrak{A}_1$

It must be taken in consideration that an *information ageing rule* is a sentence $a \in \Delta$ of the form $head \prec body$ where $body$ does not include literals like $happens(action, time)$ and must have $head$ at a previous moment. On the other hand an *event rule* is also a sentence $\in \Delta$ also of the form $head \prec body$ but in this case $body$ must include a literals like $happens(action, time)$. As example of information ageing rule we can take $\sim holdsAt(enough\_battery, T) \prec holdsAt(enough\_battery, T_1), T = T_1 + 10)$ while $holdsAt(enough\_battery, T) \prec happens(replace\_battery, T)$ is an event rule.

Clearly we can consider the possibility of changing the battery, modifying the program on example 3. Leading to the same program with the addition of the boxed sentence.

$$\Pi_3 = \left\{ \begin{array}{l} initiallyP(asleep(fred)) \\ initiallyP(enough\_battery) \\ happens(clock\_rings, T) \leftarrow holdsAt(enough\_battery, T) \\ \boxed{happens(replace\_battery, T) \leftarrow \sim holdsAt(enough\_battery, T)} \\ \sim holdsAt(asleep(fred), T) \leftarrow happens(clock\_rings, T) \end{array} \right\}$$

$$\Delta_3 = \left\{ \begin{array}{l} \neg holdsAt(enough\_battery, T) \prec holdsAt(enough\_battery, T_1), T = T_1 + 10) \\ \boxed{holdsAt(enough\_battery, T) \prec happens(replace\_battery, T)} \end{array} \right\}$$

$$\Upsilon_3 = \left\{ \begin{array}{l} holdsAt(asleep(X), T+1) \multimap holdsAt(asleep(X), T) \\ \sim holdsAt(asleep(X), T+1) \multimap \sim holdsAt(asleep(X), T) \\ holdsAt(enough\_battery, T+1) \multimap holdsAt(enough\_battery, T) \end{array} \right\}$$

In this case we can see that the wining argument is build due to the occurrence of new event introduced. Figure 3 illustrate this new argument.

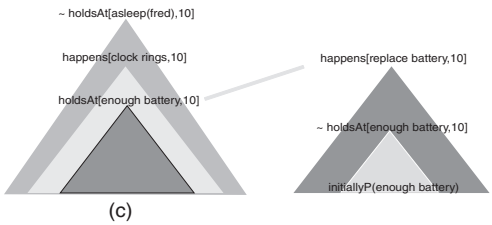

Figure 5: additional argument

Other situation that follows from the previous one, is what happens if the action under consideration can only takes place once, like $born$. This is so because a certain individual can $born$ only once. In other to deal with this kind of events we need to introduce axioms on set Π of the program that sets the uniqueness on action occurrence and use *Temporal disagreement* notion as well (see definition 10).

DEFINITION **10** *(Temporal Disagreement)*

Two arguments $\mathcal{A}_1^{t_1} = \langle \mathcal{A}_1, (l_1, t_1)) \rangle$, $\mathcal{A}_2^{t_2} = \langle \mathcal{A}_1, (l_2, t_2)) \rangle$ are in *temporal disagreement* of there are time points t_1, t_2 such that $\mathcal{A}_1^{t_1}$, $\mathcal{A}_2^{t_2}$ are in disagreement. Meaning that: $\Pi \cup \{(l_1, t_1), (l_2, t_2)\} \vdash \bot$

Let see an example where this kind of disagreement actually happens.

EXAMPLE **4** Imagine an scenario where certain alarm clock must ring at some determine moment, only if it has enough battery. Here is the program, $\mathcal{P}_4 = (\Pi_4, \Delta_4, \emptyset)$, where:

$$\Pi_4 = \left\{ \begin{array}{rl} \sim happens(born(baby), T) \leftarrow & happens(born(baby), T_1), T > T_1 \\ \sim happens(fall\_umbilical\_cord(baby), T) \leftarrow & happens(fall\_umbilical\_cord(baby), T_1), \\ & T > T_1 \\ & \\ happens(born(mary's\_baby), T) \leftarrow & holdsAt(at\_term, T) \\ happens(born(mary's\_baby), T) \leftarrow & holdsAt(before\_term, T) \\ & \\ happens(fall\_umbilical\_cord(mary's\_baby), T) \leftarrow & happens(born(baby, T_1), \\ & holdsAt(before\_term, T_1), T = T_1 + 10 \\ happens(fall\_umbilical\_cord(mary's\_baby), T) \leftarrow & happens(born(baby, T_1), \\ & holdsAt(at\_term_1), T = T_1 + 7 \\ & \\ happens(become\_pregnant(mary), 1) & holdsAt(first\_baby(mary), T) \\ holdsAt(calm\_person(mary), T) & \end{array} \right\}$$

$$\Delta_4 = \left\{ \begin{array}{rl} holdsAt(at\_term, T) \prec & happens(become\_pregnant(mary), T_1), \\ & holdsAt(calm\_person(mary), T_1), T = T_1 + 40) \\ holdsAt(before\_term, T) \prec & happens(become\_pregnant(mary), T_1), \\ & holdsAt(first\_baby(mary), T_1), T = T_1 + 38) \end{array} \right\}$$

We can observe that these arguments are very similar, their base has the same number of sentences and they are of the same kind also. At first sight it seems to be no conflict among them, but both appearing events can only happens once. If we choose to keep with the first happening of the event (over the time line) then for event $fall\_umbilical\_cord(baby)$ we must choose argument (b), but for event $born(baby)$ the sub-argument of (a) must win.

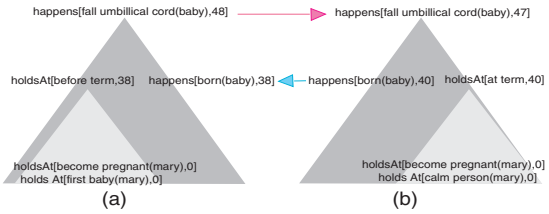

Figure 6: unique events on conflict

Although these considerations are extremely important for the preference criteria, the addition of these facts to a proper definition is in current analysis.

6 Conclusions and Future work

Argumentative systems such as $DeLP$ have been significant for the evolution of commonsense reasoning area. We have already presented $\mathcal{C}DeLP$ as an alternative argumentative system that captures today's definition of this kind of reasoning (*i.e* reason with events, time, cause-effect principle).

$\mathcal{C}DeLP$, differentiates two kind of defeasible rules, the ones already available at plain $DeLP$ and the ones that captures inertia over fluents truth value. These modifications make urgent a change in preference criteria among arguments. In that sense we present several definitions to solve some of the critical points.

A complete definition of the comparison criteria is needed. The analysis must take in consideration how the use of inertial rules are used in the construction of arguments and counterarguments. Clearly, an argument that uses inertial rules is weaker than an argument that uses only non-inertial defeasible rules. Definitions and examples presented in this work offers a solution to these items. Nevertheless, some other questions remain. We must define how to determine preference when 'defeasible' events are used on arguments. In which order we should apply the definitions presented in this work and the remaining ones in order to get better results?

References

[1] CAPOBIANCO, M., CHESÑEVAR, C. I., AND SIMARI, G. R. argumentation and the dynamics of warranted beliefs in changing environments. *Autonomouns Agents and Multi-Agent Systems 11*, 2 (2005), 127–151.

[2] CARTHY, J. M. First order theories of individual concepts and propositions. In *Machine Intelligence 9*, B. Meltzer and e. D. Michie, Eds. Edinburgh Univerity Press, Edinburgh, 1979, pp. 120–147.

[3] CARTHY, J. M. Applications of Circunscription to Formalizing commonsense Knowledge. *Artificial Intelligence 28* (1986), 89–116.

[4] CARTHY, J. M., AND HAYES, P. J. Some philosophical problems from the standpoint of artificial intelliegence. In *Machine Intelligence 4*, B. Meltzer and e. D. Michie, Eds. Edinburgh Univerity Press, Edinburgh, 1969, pp. 463–502.

[5] CHESÑEVAR, C. I., MAGUITMAN, A. G., AND LOUI, R. Logical models of argument. *ACM Computing Surveys 4*, 32 (2000), 337–383.

[6] CHESÑEVAR, C. I., SIMARI, G. R., GODO, L., AND ALSINET, T. Expansion operators for modelling agent reasoning in possiblistic defeasible programming. In *EUMAS* (2005), pp. 474–475.

[7] COBO, M. L., AND SIMARI, G. R. Combining a causal language with argumentation: A first approach. In *Proceedings of the XII Congreso Argentino de Cs. de la Computación, CACIC06* (San Luis, 2006), pp. 1231–1242.

[8] GARCIA, A. J., AND SIMARI, G. R. Defeasible logic programming: an argumentative approach. *TPL 4* (2004), 95–138.

[9] HANKS, S., AND DERMOTT, D. M. Nonmonotonic logic and temporal projection. *Artificial Intelligence 33* (1987), 379–412.

[10] KOWALSKI, R., AND SERGOT, M. A logic-based calulus of events. *New Generation Computing 4*, 1 (1986), 67–895.

[11] LLOYD, J. W. *Foundations of Logic Programming*, third edition ed. Springer-Verlag, New York, 1995.

[12] MCCARTHY, J. A Form of Non-Monotonic Reasoning. *Artificial Intelligence 13* (1980), 27–39.

[13] MCDERMOTT, D., AND DOYLE, J. Nonmonotonic logic 1. *Artificial Intelligence 13* (1980), 41–72.

[14] MILLER, R., AND SHANNAHAN, M. Some alternative formulations of the event calculus. *Computational Logic: Logic Programming and Beyond 14* (2004), 703–730.

[15] MUELLER, E. T. Event calculus reasoning through satisfiability. *Journal of Logic and Computation* (2002), 452–490.

[16] MUELLER, E. T. *Commonsense Reasoning*. Morgan Kaufman an imprint of ELsevier, 2006.

[17] PRAKKEN, H., AND VREESWIJK, G. Logics for defeasible argumentation. In *Handbook of Philosophical Logic*, D. G. (ed.), Ed. Kluwer Academic Publishers, 1998.

[18] PRIOR, A. *Past, Present and Future*. Clarendon Press, 1967.

[19] REITER, R. A Logic for Default-Reasoning. *Artificial Intelligence 13* (1980), 81–132.

[20] REITER, R. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.

[21] RESCHER, N., AND URQUHART, A. *Temporal Logic*. Springer-Verlag, 1971.

[22] SHANAHAN, M. Representing continuous change in the event calculus. In *Proceedings ECAI 90* (1990), pp. 598–603.

[23] VAN EMDEM, M., AND KOWALSKI, R. The Semantics of Predicate Logic as a Programming Language. *Journal of the Association for Computing Machinery 23*, 4 (1976), 733–742.