An approach for Temporal Argumentation Using Labeled Defeasible Logic Programming (ℓ-DeLP)

Maximiliano C. Budán[1,2,3] Mauro Gómez Lucero[1,2]
Guillermo R. Simari[2]

[1] Argentine National Council of Scientific and Technical Research (CONICET), ARGENTINA
[2] AI Research and Development Laboratory (LIDIA) – Universidad Nacional del Sur, ARGENTINA
[3] Universidad Nacional de Santiago del Estero, ARGENTINA
E-mail: {mcdb,mjg,grs} @cs.uns.edu.ar

Abstract

In the last decade, several argument-based formalisms have emerged, with application in many areas, such as legal reasoning, autonomous agents and multi-agent systems; many are based on Dung's seminal work characterizing *Abstract Argumentation Frameworks (AF)*. Recent research in the area has led to *Temporal Argumentation Frameworks (TAF)*, that extend AF by considering the temporal availability of arguments. On the other hand, different more concrete argumentation systems exists, such as *Defeasible Logic Programming (DeLP)*, specifying a knowledge representation language, and how arguments are built.

In this work we combine time representation capabilities of TAF with the representation language and argument structure of DeLP, defining a rule-based argumentation framework that considers time at the object language level. In order to do this, we use an extension of DeLP, called *Labeled DeLP (ℓ-DeLP)* to establish, for each program clause, the set of time intervals in which it is available, and to determine from this information the temporal availability of arguments. Acceptability semantics for TAF can then be applied to determine argument acceptability on time.

Keyword: Argumentation, Temporal Availability, Defeasible Logic Programming, Labeled Defeasible Logic Programming.

1 Introduction

Argumentation is the process of defending a given affirmation by giving reasons for its acceptance. Both the original claim and its support are subject to consideration, since reasons supporting conflicting claims can be proposed.

An argument is a structure composed of a claim (or conclusion) along with the reasons supporting it, called premisses, and a description of the way in which such premisses support the claim. The latter component is commonly referred to as inference.

The process of argumentation can be understood as a game between two players: the proponent which proposes an initial argument and tries to defend it, and then opponent whose role in the game is to interfere the original argument. An argument offered to undermine another argument is called counterargument. The proponent's arguments are called arguments *pro* and the opponent's *op*. Thus, the process begins with the introduction of a pro argument by the proponent, then the opponent offers their arguments op. In this moment, the proponent becomes opponent of his opponent and offers counterarguments to their counterarguments. The process continues in this way until all the relevant arguments where considered. As a result of this argumentation process, the initially proposed argument will be considered as accepted if and only if, is successfully defended from all its counterarguments in this process.

Initially proposed by Dung [7], the abstract model of argumentation considers a set of arguments, as atomic (structureless) units, and a binary relation defined on the set of arguments representing attack, allowing to concentrate on the acceptability analysis described before. Formally, an abstract argumentation framework is a pair AF = $\langle AR, Attacks\rangle$, where AR is the set of arguments considered and $Attacks$ represents the relationship of attack on AR. Different argument acceptability semantics were proposed for this abstract framework.

There exist many extensions that emerged from this abstract formalism, among them is the *Timed Abstract Framework (TAF – [5, 6])*, which allows to specify time availability for arguments, and determine the time intervals in which an argument is acceptable. TAF extends the AF of Dung by adding a function Av that defines the time intervals in which the arguments are available, resulting in a 3-tuple $\langle AR, Attacks$,Av$\rangle$, and different acceptability semantics considering time were defined

for TAF.

Besides abstract argumentation approaches, different more concrete argumentation systems exists, specifying a knowledge representation language, and how arguments are built. One of those systems is *Defeasible Logic Programming (DeLP* - [8]), a formalism that combines results of *Logic Programming* and *Defeasible Argumentation*. DeLP allows representing information in the form of weak rules in a declarative way, from which arguments supporting conclusions are constructed, and provides a defeasible argumentation inference mechanism for determining *warranted* conclusions. 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 to agents.

The aim of this paper is to introduce a rule-based argumentation framework considering time at the object language level. We used a extension of DeLP called *Labeled DeLP(ℓ-DeLP– [9])* to establish, for each program clause, the set of time intervals in which it is available, and to determine from this information the temporal availability of arguments. Finally acceptability semantics for TAF can be applied to determine argument acceptability on time.

2 Abstract Argumentation

Argumentation has evolved into a powerful paradigm to formalize commonsense reasoning. Phan Minh Dung [7] introduced the notion of *Argumentation Framework (AF)* as an abstraction of a defeasible argumentation system. In the AF an argument is considered as an abstract entity with unspecified internal structure, and whose role is determined only by its attack relations with other arguments.

Definition 1 (Argumentation Framework [7]) *An argumentation framework (AF) is a pair $\langle AR, Attacks\rangle$, where AR is a set of arguments, and $Attacks$ is a binary relation on AR i.e., $Attacks \subseteq AR \times AR$.*

Given an AF, an argument A is considered *acceptable* if it can be defended of all its attackers (arguments) with other arguments in AR. This intuition is formalized in the following definitions, originally presented in [7].

Definition 2 (Acceptability) *Let $AF = \langle AR, Attacks\rangle$ be an argumentation framework.*

- *A set $S \subseteq AR$ is called* conflict-free *if there are no arguments $A, B \in S$ such that $(A, B) \in Attacks$.*
- *An argument $A \in AR$ is* acceptable *with respect to a set $S \subseteq AR$ iff for each $B \in AR$, if B attacks A then B is attacked by S.*
- *A conflict-free set $S \subseteq AR$ is* admissible *iff each argument in S is acceptable with respect to S.*
- *An admissible set $E \subseteq AR$ is a* complete extension *of AF iff E contains each argument that is acceptable with respect to E.*
- *A set $E \subseteq AR$ is the* grounded extension *of AF iff E is minimal with respect to set inclusion, such that is admissible and complete.*

Dung [7] also presented a fixed-point characterization of the grounded semantics based on the characteristic function F defined below.

Definition 3 *Let $\langle AR, Attacks\rangle$ be an AF. The associated characteristic function is defined as follows: $F : 2^{AR} \to 2^{AR}$, $F(S) =_{def} \{A \in AR \mid A$ is acceptable w.r.t. $S\}$.*

The following proposition suggests how to compute the grounded extension associated with a *finitary* AF (*i.e.*, such that each argument is attacked by at most a finite number of arguments) by iteratively applying the characteristic function starting from $\emptyset$.

Proposition 1 *[7] Let $\langle AR, Attacks\rangle$ be a finitary AF. Let $i \in \mathbb{N} \cup \{0\}$ such that $F^i(\emptyset) = F^{i+1}(\emptyset)$. Then $F^i(\emptyset)$ is the least fixed point of F, and corresponds to the grounded extension associated with the AF.*

Example 1 *Consider the AF $\langle AR, Attacks\rangle$ (graphically represented in Fig. 1), where $AR = \{A, B, C, D, E, F, G\}$ and $Attacks = \{(B, A), (C, B), (E, A), (G, E), (F, G), (G, D)\}$.*

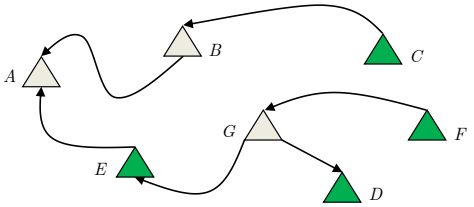

Figure 1: Argumentation Framework

The set $S = \{C, D, E, F\}$ is admissible, since it defends all the arguments it contains. S is also complete since it contains all the arguments in AR defended by S. Finally, it can be verified that S is the minimal set satisfying the previous conditions, and therefore it corresponds to the grounded extension of AR. Next we show how to obtain the grounded extension by applying the fixed point characterization from Prop. 1.
$F^0(\emptyset) = \emptyset$

$F^1(\emptyset) = F(\emptyset) = \{C, F\}$
$F^2(\emptyset) = F(\{C, F\}) = \{C, F, D, E\}$
$F^3(\emptyset) = F(\{C, F, D, E\}) = F^2(\emptyset)$

3 Modeling Temporal Argumentation with TAF

Many formalizations are based on Dung´s proposed framework, extending its representation capabilities. Among them is the *Timed Abstract Framework (TAF) [5, 6]*, a novel argumentation formalism where arguments are valid only during specific intervals of time (called availability intervals). Attacks between arguments are considered *only* when both the attacker and the attacked arguments are available. Thus, when identifying the set of acceptable arguments the outcome associated with a TAF may vary in time.

In order to represent time we assume that a correspondence was defined between the time line and the positive reals set (the positive X-axis). A *time interval*, representing a period of time without interruptions, will be then represented as a real interval $[a-b]$ (we use '$-$' instead of ',' as a separator for legibility reasons).

Definition 4 (Time Interval) *A time interval, or just interval, is a real interval.*

As is usual for real intervals, to indicate that one of the endpoints (extremes) of the interval is to be excluded, the corresponding square bracket will be replaced with a parenthesis (*e.g.*, $(a, b]$ to exclude the endpoint a).

Now, to model discontinuous periods of time we introduce the notion of *time intervals set*. Although a time intervals set suggests a representation as a set of sets (set of intervals), we chose a flattened representation as a set of reals (the set of all real numbers contained in any of the individual time intervals). In this way, we can directly apply traditional set operations and relations on time intervals sets.

Definition 5 (Time Intervals Set) *A time intervals set, or just intervals set, is a subset $S \subseteq \Re$.*

When convenient we will use the set of sets notation for time intervals sets. Concretely, a time interval set S ($\subseteq \Re$) will be denoted as the set of all disjoint and $\subseteq$-maximal individual intervals included in the set. For instance, we will use $\{(1-3],\ [4.5-8)\}$ to denote the time interval set $(1-3] \cup [4.5-8)$

Now we formally introduce the notion of *Timed Argumentation Framework*, which extends the AF of Dung by incorporating an additional component, the availability function, which will be used to capture those time intervals where arguments are available.

Definition 6 (Timed Argumentation Framework) *A timed argumentation framework (or simply TAF) is a 3-tuple $\langle AR, Attacks, Av\rangle$ where AR is a set of arguments, $Attacks$ is a binary relation defined over AR and Av is an availability function for timed arguments, defined as $Av : AR \longrightarrow \wp(\Re)$, such that $Av(A)$ is the set of availability intervals of an argument A.*

Example 2 *Let $\Phi = \langle AR, Attacks, Av\rangle$ be a TAF where:*
$AR = \{A, B, C, D, E, F, G\}$
$Attacks = \{(B, A), (C, B), (E, A), (G, E), (F, G), (G, D)\}$
$Av = \{(A, \{[10-50], [80-120]\}); (B, \{[55-100]\}); (C, \{[40-90]\}); (D, \{[10-30]\}); (E, \{[20-75]\}); (F, \{[5-30]\}); (G, \{[10-40]\})\}$ *(See Fig. 2)*

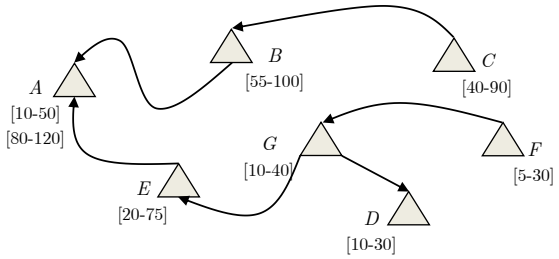

Figure 2: TAF corresponding to example 2

The following definitions formalize argument acceptability in TAF, and are extensions of the acceptability notions presented in section 2 for AF. Firstly we present the notion of t-profile, binding an argument to a set of time intervals, which constitutes a fundamental component for the formalization of time-based acceptability.

Definition 7 (T-Profile) *Let $\Phi = \langle AR, Attacks, Av\rangle$ be a TAF. A* timed argument profile *in Φ, or just* t-profile*, is a pair $\rho = (A, \tau)$ where $A \in AR$ and τ is a set of time intervals. The t-profile $(A, Av(A))$ is called the* basic *t-profile of A.*

Since the availability of arguments varies in time, the acceptability of a given argument A will also vary in time. The following definitions reformulate Dung's original formalization for abstract argumentation by considering t-profiles instead of arguments.

Definition 8 (Defense of A from B w.r.t. S) *Let S be a set of t-profiles. Let A and B be arguments. The defense t-profile of A from B w.r.t. S is $\rho_A = (A, \tau_A^B)$, where: $\tau_A^B =_{def} Av(A) - Av(B) \bigcup_{\{(C,\tau_C)\in S \mid C\ Attacks\ B\}} (Av(A) \cap Av(B) \cap \tau_C)$*

Intuitively, A is defended from the attack of B when B is not available ($Av(A) - Av(B)$), but also in those intervals where, although the attacker B is available, it is in turn attacked by an argument C in the base set S. The following definition captures the defense profile of A, but considering all its attacking arguments.

Definition 9 (Acceptable t-profile of A w.r.t. S) *Let S be a set of t-profiles. The acceptable t-profile for A w.r.t. a set S is $\rho_A = (A, \tau_A)$, where $\tau_A =_{def} \cap_{\{B \ Attacks \ A\}} \tau_A^B$ and (A, τ_A^B) is the defense t-profile of A from B w.r.t. S.*

Since an argument must be defended of all its attacks to be considered acceptable, we have to intersect the set of time intervals in which it is defended of each of its attackers.

Definition 10 (Acceptability) *Let $AF = \langle AR, Attacks, Av \rangle$ be a temporal argumentation framework.*

- *A set S of t-profiles is called* t-conflict-free *if there are no t-profiles $(A, \tau_A), (B, \tau_B) \in S$ such that $(A, B) \in Attacks$ and $\tau_A \cap \tau_B \neq \emptyset$.*
- *A t-conflict-free set S of t-profiles is a t-admissible set iff $\forall \ (A, \tau_A) \in S$ it holds that (A, τ_A) is the acceptable t-profile of A w.r.t. S.*
- *A t-admissible set S is a* t-complete extension *of TAF iff S contains all the t-profiles that are acceptable with respect to S.*
- *A set S is the* t-grounded extension *of TAF iff S is minimal with respect to set inclusion such that is t-admissible and t-complete.*

In particular, the fixed point characterization for grounded semantics proposed by Dung can be directly applied to TAF by considering the following modified version of the characteristic function.

Definition 11 *Let $\langle AR, Attacks, Av \rangle$ be a TAF. Let S be a set of t-profiles. The associated characteristic function is defined as follows: $F(S) =_{def} \{(A, \tau) \mid A \in AR$ and (A, τ) is the acceptable t-profile of A w.r.t. $S\}$.*

Example 3 *Suppose we want to establish the acceptability of A in the TAF Φ presented in example 2. As shown in example 1 of AF, by considering only the relation $Attacks$ we could say that the argument A is not acceptable. Let us obtain the t-grounded extension of Φ by applying the fixed point characterization.*

$F^0(\emptyset) = \emptyset$

$F^1(\emptyset) = \{(A, \{[10-20), (100-120]\}); (C, \{[40-90]\});$
$(F, \{[5-30]\}); (B, \{(90-100]\}); (E, \{(40-75]\}; \}$
$(G, \{(30-40]\})\}$

$F^2(\emptyset) = \{(A, \{[10-40], [80-90], (100-120]\});$
$(C, \{[40-90]\}); (F, \{[5-30]\}); (B, \{(90-100]\});$
$(E, \{[20-30], (40-75]\}); (G, \{(30-40]\})\}$

$F^3(\emptyset) = \{(A, \{[10-20), (30-40), [80-90), (100-120]\});$
$(C, \{[40-90]\}); (F, \{[5-30]\}); (B, \{(90-100]\});$
$(E, \{[20-30], (40-75]\}); (G, \{(30-40]\})\}$

$F^4(\emptyset) = F^3(\emptyset)$

Consequently, $F^3(\emptyset)$ is the t-grounded extension of Φ. Next we describe how the temporal availability of A was obtained in $F^3(\emptyset)$ by applying the definitions 8 and 9 from $F^2(\emptyset)$. By applying definition 8:

$\tau_A^B = (Av(A) - Av(B)) \bigcup_{\{(C, \tau_C)\}} (Av(A) \cap Av(B) \cap \tau_C)$
$= (\{[10-50], [80-120]\} - \{[55-100]\}) \cup$
$(\{[10-50], [80-120]\} \cap \{[55-100]\} \cap \{[40-90]\})$

$= \{[10-50], (100-120]\} \cup [80-90]$
$= \{[10-50], [80-90], (100-120]\}$

$\tau_A^E = (Av(A) - Av(E)) \bigcup_{\{(G, \tau_G)\}} (Av(A) \cap Av(B) \cap \tau_G)$
$= \{[10-20), (30-40], [80-120]\}$

By applying definition 9:

$\tau_A = \cap_{\{X \ Attacks \ A\}} \tau_A^X = \tau_A^B \cap \ \tau_A^E$

$= \{[10-50], [80-90], (100-120]\} \cap$
$\{[10-20), (30-40], [80-120]\}$

$= \{[10-20), (30-40], [80-90], (100-120]\}$

4 Defeasible Logic Programming

Besides abstract argumentation approaches, like those presented in sections 2 and 3, different more concrete argumentation systems exists, specifying a knowledge representation language, and how arguments are built. One of those systems is *Defeasible Logic Programming (DeLP* - [8]), a formalism that combines results of *Logic Programming* and *Defeasible Argumentation*. DeLP allows representing information in the form of weak rules in a declarative way, from which arguments supporting conclusions are constructed, and provides a defeasible argumentation inference mechanism for determining *warranted* conclusions. 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 to agents.

Below we present the definitions of *program* and *argument* in DeLP.

Definition 12 (DeLP Program) *A DeLP program $\mathcal{P}$ is a pair (Π, Δ) where (1) Δ is a set of defeasible rules*

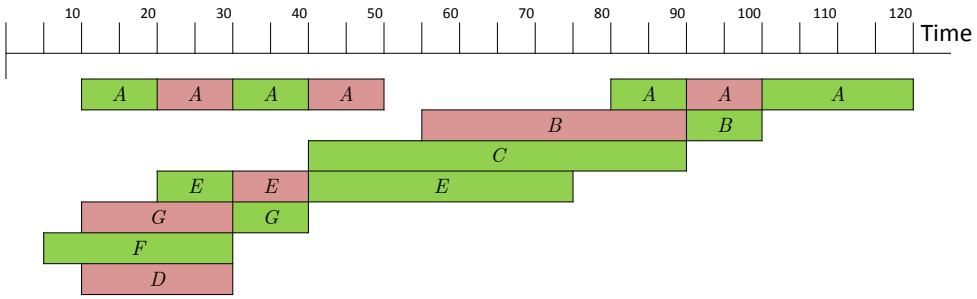

Figure 3: Representation of the arguments associated with Ex. 3 in a time line

of the form $L \prec P_1, \ldots, P_n$*, with* $n > 0$*, where* L *and each* P_i *are literals, and (2)* Π *is a set of strict rules of the form* $L \leftarrow P_1, \ldots, P_n$*, with* $n \geq 0$*, where* L *and each* P_i *are literals. A literal* L *is a ground atom* A *or a negated ground atom* $\sim A$*, where '*$\sim$*' represents the strong negation.*

Pragmatically, strict rules can be used to represent strict (non defeasible) information, whereas defeasible rules are used to represent tentative or weak information. In particular, a strict rule $L \leftarrow P_1, \ldots, P_n$ with $n = 0$ is called *fact*, and will be denoted just as L. It is important to remark that the set Π must be consistent as it represents strict (undisputed) information. In contrast, the set Δ will generally be inconsistent, since it represents tentative information.

Definition 13 (Defeasible Derivation) *Let* $\mathcal{P}$ *be a DeLP program and* L *a ground literal. A defeasible derivation of* L *from* $\mathcal{P}$ *consists of a finite sequence* $L_1, \ldots, L_n = L$ *of ground literals, such that for each* i*,* $1 \leq i \leq n$*,* L_i *is a fact or there exists a rule* R_i *in* $\mathcal{P}$ *(strict or defeasible) with head* L_i *and body* $B_1, \ldots, B_m$*, such that each literal on the body of the rule is an element* L_j *of the sequence appearing before* L_i *(*$j \leq i$*). We will use* $\mathcal{P} \mid\sim L$ *to denote that there exists a defeasible derivation of* L *from* $\mathcal{P}$*.*

We say that a given set of DeLP clauses is contradictory if and only if there exists a defeasible derivation for a pair of complementary literals (w.r.t. strong negation) from this set.

Definition 14 (Argument) *Let* L *be a literal and* $\mathcal{P} = (\Pi, \Delta)$ *be a DeLP program. An* argument *for* L *is a pair* $\langle A, L\rangle$*, where* A *is a set of defeasible rules of* Δ*, such that:*

1. *there is a defeasible derivation for* L *from* $\Pi \cup A$*.*
2. $\Pi \cup A$ *is not contradictory, and*
3. A *is a minimal, i.e., there exist no proper subset* A'*,* $A' \subset A$ *satisfying conditions (1) and (2).*

We say that an argument $\langle B, Q\rangle$ *is a* sub-argument *of* $\langle A, L\rangle$ *iff,* $B \subseteq A$*.*

DeLP provides an argumentation based mechanism to determine *warranted* conclusions. This procedure involves constructing arguments from programs, identifying conflicts or *attacks* among arguments, evaluating pairs of arguments in conflict to determine if the attack is successful, becoming a *defeat*, and finally analyzing defeat interaction among all relevant arguments to determine warrant.

Below we briefly present the formalization of the previously mentioned notions, as introduced in [8].

Definition 15 (Disagreement) *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a DeLP program. Two literals* L *and* L' *are in* disagreement *if and only if the set* $\Pi \cup \{L, L'\}$ *is contradictory.*

The simplest example of literals in disagreement are two complementary literals as "p" and "$\sim p$", since $\Pi \cup \{p, \sim p\}$ is a contradiction, whatever the set Π. However, two non-complementary literals, like "p" and "q", can disagree *e.g.*, if $\Pi = \{h \leftarrow p, \sim h \leftarrow q\}$.

Definition 16 (Attack) *Let* $\mathcal{P} = (\Pi, \Delta)$ *be a DeLP program. Let* $\langle A_1, L_1\rangle$ *and* $\langle A_2, L_2\rangle$ *be two arguments in* $\mathcal{P}$*. We say that* $\langle A_1, L_1\rangle$ counter-argues*,* rebuts*, or* attacks $\langle A_2, L_2\rangle$ *at the literal* L *if and only if there is a sub-argument* $\langle A, L\rangle$ *of* $\langle A_2, L_2\rangle$ *such that* h *and* L_1 *are in* disagreement*. The argument* $\langle A, L\rangle$ *is called* disagreement sub-argument*, and the literal* L *will be the counter-argument point.*

In order to decide if a partial attack really succeeds, constituting a defeat, a comparison criterion must be used, establishing the relative strength of the arguments involved in the attack. In this work we will use the criterion adopted by default in DeLP, called *specificity*, which favors arguments based on more information or supporting their conclusions more directly. The formal definition is presented below.

Definition 17 (Specificity) *Let $\mathcal{P} = (\Pi, \Delta)$ be a DeLP program, and let Π_G be the set of all strict rules from Π (without including facts.) Let F be the set of all literals that have a defeasible derivation from $\mathcal{P}$ (F will be considered as a set of facts.) Let $\langle A_1, L_1\rangle$ and $\langle A_2, L_2\rangle$ be two arguments from $\mathcal{P}$. $\langle A_1, L_1\rangle$ is strictly more specific than $\langle A_2, L_2\rangle$ (denoted $\langle A_1, L_1\rangle \succ \langle A_2, L_2\rangle$) if the following conditions hold:*

1. *For all $H \subseteq F$: if $\Pi_G \cup H \cup A_1 \sim L_1$ and $\Pi_G \cup H \nvdash L_1$, then $\Pi_G \cup H \cup A_2 \sim L_2$ and*
2. *there exists $H' \subseteq F$ such that $\Pi_G \cup H' \cup A_2 \sim L_2$ and $\Pi_G \cup H' \nvdash L_2$, and $\Pi_G \cup H' \nvdash L_1$*

Definition 18 (Defeat) *Let $\mathcal{P} = (\Pi, \Delta)$ be a DeLP program. Let $\langle A_1, L_1\rangle$ and $\langle A_2, L_2\rangle$ be two arguments in $\mathcal{P}$. We say that $\langle A_2, L_2\rangle$* defeats *$\langle A_1, L_1\rangle$ if and only if there exists a sub-argument $\langle A, L\rangle$ of $\langle A_1, L_1\rangle$ such that $\langle A_2, L_2\rangle$ counter-argues $\langle A_1, L_1\rangle$ at literal h and it holds that:*

1. *$\langle A_2, L_2\rangle$ is strictly more specific that $\langle A, L\rangle$ (proper defeater), or*
2. *$\langle A_2, L_2\rangle$ is unrelated to $\langle A, L\rangle$ (blocking defeater)*

In DeLP a literal L will be warranted if there exists a non-defeated argument structure $\langle A, L\rangle$; to establish whether $\langle A, h\rangle$ is non-defeated, the set of defeaters for A will be considered. Since each defeater D for A is itself an argument structure, defeaters for D will in turn be considered, and so on. This analysis suggests a tree structure with nodes representing arguments and edges representing defeats, called dialectical tree, in which DeLP bases the analysis of warrant.

5 DeLP with time through Labels

In real application domains of argumentation requiring the explicit treatment of time, temporal information is not in general directly associated with arguments, but instead it is attached to the basic pieces of knowledge (in general logical rules) from which arguments are built. Since we will use the DeLP language to instantiate abstract arguments in TAF, we need a way to associate temporal availability information to DeLP clauses.

In this direction we will use an extension of DeLP, called *ℓ-DeLP (labelled DeLP – [9])* which incorporates the possibility to add meta-information (through labels) to DeLP clauses for specific purposes. The meta-information could be: probability, certainty, reliability of the source, and even time intervals.

The formalism ℓ-DeLP considers the following elements:

1. A domain of labels Γ, used to represent the meta-information.
2. An association of a label of the domain to each program clause.
3. A function *ALS* (*Argument Label Synthesis*), $ALS : 2^{\Gamma} \longrightarrow \Gamma$, specifying how to obtain the label associated with an argument from the labels of the clauses composing it.

In order apply ℓ-DeLP to capture the availability of arguments we will use labels to represent time intervals sets, and we will define *ALS* function as the intersection of the labels associated with individual clauses. Formally:

1. The labels domain Γ is the set $2^{\mathbb{R}}$.
2. An association of a time interval set (label) to each program clause.
3. The function *ALS* is defined as follows: $ALS(\tau_1, \tau_2, ..., \tau_n) =_{def} \tau_1 \cap \tau_2 \cap ... \cap \tau_n$.

Example 4 *Next we present a ℓ-DeLP program. This program will give rise to the TAF of the example 2. (See Figure 2).*

$$\mathcal{P} = \left\{ \begin{array}{ll} a \prec s, k : \{[10-60];[80-150]\} & \sim n \leftarrow m, t : \{[40-65]\} \\ s \prec j, l : \{[10-60];[80-130]\} & \sim f \leftarrow j, p : \{[5-30]\} \\ c \leftarrow j, k : \{[0-120]\} & r \leftarrow p, t : \{[10-30]\} \\ k \prec m, l : \{[10-50];[70-140]\} & j : \{[0-150]\} \\ \sim k \prec j, n : \{[30-75]\} & l : \{[0-150]\} \\ n \prec c, l : \{[55-100]\} & m : \{[0-150]\} \\ \sim s \prec j, r : \{[0-90]\} & t : \{[0-150]\} \\ r \leftarrow l, p : \{[20-130]\} & p : \{[0-150]\} \\ \sim r \leftarrow j, f : \{[10-60]\} & c : \{[0-150]\} \\ f \leftarrow p, m : \{[0-40]\} & \end{array} \right\}$$

Figure 4 depicts an argument with respect to program $\mathcal{P}$ (corresponding to the argument A in example 2), and describes how the argument label is computed from the labels of its composing clauses through ALS *application.*

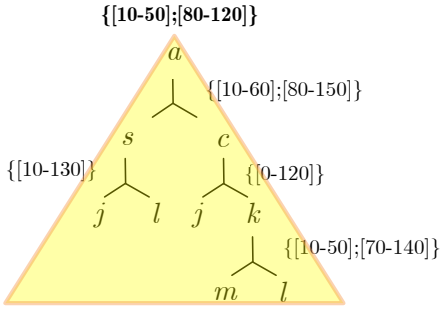

Figure 4: Argument A and associated label

Finally, in order to define the acceptability of arguments in our ℓ-DeLP instantiation, we will just construct, from the labeled program, a TAF involving all the constructible arguments with their availability captured through labels (see Figure 5).

In this case the TAF constructed corresponds to the one in example 2, resulting in the same acceptability analysis.

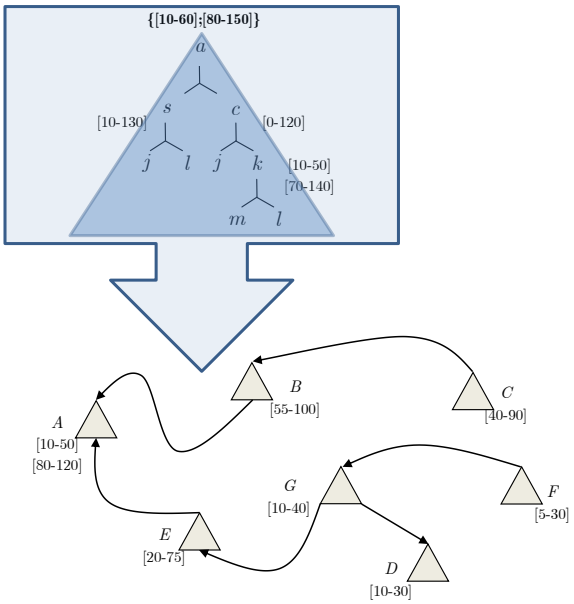

Figure 5: TAF obtained from a ℓ-DeLP program $\mathcal{P}$

6 Conclusions. Related and Future Work

Argumentation has contributed with a human-like mechanism to the formalization of commonsense reasoning. Among the main argumentation based approaches is the *abstract framework (AF)* of Dung, which has proven to be fruitful for developing several extensions with application in different contexts (e.g. [3, 2, 4, 1], among many others).

On the one hand, our work focuses on one of those extensions, called TAF, incorporating to AF the capability to represent temporal availability associated with (abstract) arguments, and considering argument acceptability varying on time. On the other hand, to provide structure to temporal arguments, we consider a concrete, fully specified (non-abstract) argumentation formalism called DeLP.

Then, in this work we combined TAF and DeLP, introducing a rule-based argumentation framework considering time at the object language level. We used an extension of DeLP, called ℓ-DeLP, to represent, for each program clause, the set of time intervals in which it is available, and to determine from this information the temporal availability of arguments. Finally we shown how acceptability semantics for TAF can be applied to determine argument acceptability on time.

As future work we will develop an implementation of ℓ-DeLP by using the existing DeLP system [1] as a basis. The resulting implementation will be exercised in different domains requiring to model availability of the information varying over time. We are also interested in analyzing the salient features of our formalization in the context of other argumentation frameworks, such as the ASPIC+ framework [10], where rationality postulates for argumentation are explicitly considered.

[1] See http://lidia.cs.uns.edu.ar/delp

References

[1] Leila Amgoud and Caroline Devred. Argumentation frameworks as constraint satisfaction problems. In Salem Benferhat and John Grant, editors, *SUM*, volume 6929 of *Lecture Notes in Computer Science*, pages 110–122. Springer, 2011.

[2] Gerhard Brewka, Paul E. Dunne, and Stefan Woltran. Relating the semantics of abstract dialectical frameworks and standard afs. In Toby Walsh, editor, *IJCAI*, pages 780–785. IJCAI/AAAI, 2011.

[3] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *KR*. AAAI Press, 2010.

[4] Martin Caminada and Gabriella Pigozzi. On judgment aggregation in abstract argumentation. *Autonomous Agents and Multi-Agent Systems*, 22(1):64–102, 2011.

[5] Maria Laura Cobo, Diego C. Martínez, and Guillermo R. Simari. On admissibility in timed abstract argumentation frameworks. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *ECAI*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 1007–1008. IOS Press, 2010.

[6] Maria Laura Cobo, Diego C. Martínez, and Guillermo R. Simari. Acceptability in timed frameworks with intermittent arguments. In Lazaros S. Iliadis, Ilias Maglogiannis, and Harris Papadopoulos, editors, *EANN/AIAI (2)*, volume 364 of *IFIP Publications*, pages 202–211. Springer, 2011.

[7] Phan M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[8] Alejandro J. García and Guillermo R. Simari. Defeasible logic programming: An argumentative approach. *Theory Practice of Logic Programming*, 4(1):95–138, 2004.

[9] M. Gómez Lucero, C. Chesñevar, G. Simari, and A. García. Extensión de la argumentación rebatible para considerar etiquetas. *VIII Workshop de*

Investigadores en Ciencias de la Computación, pages 189–193, 2006.

[10] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.