Modeling Argumentation with Labeled Deduction: Formalization and Theoretical Considerations [⋆]

Carlos Iván Chesñevar, `cic@cs.uns.edu.ar`
Guillermo Ricardo Simari, `grs@cs.uns.edu.ar`

Departamento de Ciencias de la Computación – Universidad Nacional del Sur
Av.Alem 1253 – B8000CPB Bahía Blanca – República Argentina
Tel/Fax: (+54) (291) 459 5135/5136 – Email: {cic,grs}@cs.uns.edu.ar

Abstract. In the last years there has been an increasing demand of a variety of logical systems, prompted mostly by applications of logic in AI, logic programming and other related areas. Labeled Deductive Systems (LDS) were developed as a flexible methodology to formalize such a kind of complex logical systems.
During the last decade defeasible argumentation has proven to be a confluence point for many approaches to formalizing commonsense reasoning. Different formalisms have been developed, many of them sharing common features.
This paper summarizes the most relevant features of LDS_{ar}, a logical framework for defeasible argumentation based on LDS. We present a syntactic characterization of the framework, and discuss some emerging properties. We also show how different existing argumentation frameworks are subsumed in LDS_{ar}.

Key words: defeasible argumentation; knowledge representation; logic programming; Labeled deduction.

1 Introduction and motivations

Defeasible argumentation [SL92,CML00,PV99] has proven to be a successful approach to finding a suitable formalization for reasoning with incomplete and potentially inconsistent information. Recent research (notably [BDKT97]) has shown that defeasible argumentation constitutes a confluence point for characterizing traditional approaches to model non-monotonic reasoning, such as extended logic programs [GL90] and default logic [Rei80], among many others.

During the last decade a number of alternative formalisms for argumentation has been developed, resulting in technically different models which share some common underlying features (such as the notion of argument, attack between arguments, defeat, dialectical analysis, etc.). This constituted a motivation for the definition of a new, unified ontology in which such notions could be abstracted away by specifying a suitable underlying logical language and appropriate inference rules to capture the argumentative inference process.

Labeled Deductive Systems (LDS) [Gab96] provided the adequate tool to achieve this goal. In LDS, logical formulas are enriched by *labels* which carry metalevel information not encoded in the object language itself. Thus, different components in a complex logical system (such as logic-based argumentation systems) can be represented in terms of labels.

This paper presents the main features of LDS_{ar}, a logical framework for defeasible argumentation based on LDS. The proposed formalization allows to conclude several interesting

[⋆] This paper extends previous research work presented in [CS00]. The paper summarizes some of the main results of the first author's PhD Thesis [Che01] written under the direction of Guillermo Simari. An electronic version of the Thesis is available at `http:\\cs.uns.edu.ar\ ~cic` theoretical results concerning argumentation theory. We will also show how LDS_{ar} allows to classify and interrelate other existing argumentation frameworks in terms of knowledge encoding capabilities and expressive power. The paper is structured as follows: first, in section 2 we will briefly sketch the LDS_{ar} fundamentals, introducing the object language $\mathcal{L}_{\mathsf{Arg}}$ as well as the notion of argumentative theory. Then in section 3 we will present two consequence relations which will allow to model argument construction and dialectical analysis, respectively. Section 4 discusses the main theoretical contributions which could be proven within LDS_{ar}. We will also discuss how different argumentation frameworks can be interrelated in terms of the proposed formalization. Finally, section 5 concludes. For space reasons, we will assume that the reader is familiarized with basic notions of defeasible reasoning and argumentation theory (see [CML00,PV99] for more details).

2 LDS_{ar}: fundamentals

In this section we will introduce a *knowledge representation language* $\mathcal{L}_{\mathsf{KR}}$ for performing defeasible inference, together with a *labeling language* $\mathcal{L}_{\mathsf{Labels}}$. These languages will be used to define the object language $\mathcal{L}_{\mathsf{Arg}}$. Following Gabbay's terminology [Gab96], the basic information units in $\mathcal{L}_{\mathsf{Arg}}$ will be called *declarative units*, having the form *Label:wff*. In our approach we will restrict wffs in labeled formulas to ground literals. As we will see in section 3, a ground literal can be understood as a *conclusion* of an *argument*, which will be defined by the label.

A label in a formula $L{:}\alpha$ will provide three elements which are convenient to take into account when formalizing defeasible argumentation, namely:

1. For every declarative unit $L{:}\alpha$ the label L will distinguish whether that declarative unit corresponds to *defeasible* or *non-defeasible* information.
2. The label L will also provide an *unique name* to identify a wff in the knowledge base Γ.
3. When performing the inference of a declarative unit $L{:}\alpha$ from a set Γ of declarative units, the label L will provide a *trace* of the wffs needed to infer $L{:}\alpha$ from Γ.

Wffs in our knowledge representation language $\mathcal{L}_{\mathsf{KR}}$ will be a subset of a classic propositional language $\mathcal{L}$, restricted to *rules* and *facts*. The set of all rules and facts in $\mathcal{L}_{\mathsf{KR}}$ will be denoted $\mathsf{Rules}(\mathcal{L}_{\mathsf{KR}})$ and $\mathsf{Facts}(\mathcal{L}_{\mathsf{KR}})$, resp. We define $\mathsf{ProgClauses}(\mathcal{L}_{\mathsf{KR}}) = \mathsf{Rules}(\mathcal{L}_{\mathsf{KR}}) \cup \mathsf{Facts}(\mathcal{L}_{\mathsf{KR}})$. A modality (label) will be attached to wffs in $\mathcal{L}_{\mathsf{KR}}$, indicating whether they are *defeasible* or *non-defeasible*. Formally:

Definition 2.1 (Language $\mathcal{L}_{\mathsf{KR}}$. Wffs in $\mathcal{L}_{\mathsf{KR}}$). *The language $\mathcal{L}_{\mathsf{KR}}$ will be composed of*

1. A countable set of propositional atoms*, possibly subindicated. We will denote propositional atoms with lowercase letters. Example: a, b, c, d, e, ... , a_1, a_2, a_3 are propositional atoms.*
2. Logical connectives $\wedge$, $\neg$ and $\leftarrow$.

Wffs in $\mathcal{L}_{\mathsf{KR}}$ will be defined as follows:

1. If α is an atom in $\mathcal{L}_{\mathsf{KR}}$, then α and $\sim\alpha$ are wffs called literals *in $\mathcal{L}_{\mathsf{KR}}$.*
2. If $\alpha_1, \ldots \alpha_k$, β are literals in $\mathcal{L}_{\mathsf{KR}}$, then $\beta \leftarrow \alpha_1, \ldots \alpha_k$ is a wff in $\mathcal{L}_{\mathsf{KR}}$.

For the sake of simplicity, when referring to the language $\mathcal{L}_{\mathsf{KR}}$ the following conventions will be used: Greek lowercase letters α, β, γ will refer to any wff in $\mathcal{L}_{\mathsf{KR}}$. Lowercase letters (such as h, q, etc.) will be used for referring to ground literals in $\mathcal{L}_{\mathsf{KR}}$. Greek uppercase letters Υ, Φ, Γ will refer to a *set* of wffs in $\mathcal{L}_{\mathsf{KR}}$. The conjunction $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_k$ will be simply written as $\alpha_1, \alpha_2, \ldots, \alpha_k$.

Definition 2.2 (Labeling constants). *A set Labels* $= \{\mathsf{n}_1, \mathsf{n}_2, \ldots, \mathsf{d}_1, \mathsf{d}_2, \ldots\}$ *of* labeling constants *will include constant names with the form* n_i *and* d_i*, standing for* non-defeasible *and* defeasible *information, resp. A* set *of labeling constants will be denoted as* $\mathsf{L}_1, \mathsf{L}_2, \ldots, \mathsf{L}_k$.

Definition 2.3 (Labeling language $\mathcal{L}_{\mathsf{Labels}}$). *A label L in our labeling language* $\mathcal{L}_{\mathsf{Labels}}$ *can be either an* argument label *or a* dialectical label*, defined as follows:*

1. *An* argument label *will be a tuple* $\langle \mathsf{L}_i, \Phi \rangle$ *where* $\mathsf{L}_i \subseteq$ *Labels,* $\Phi \subseteq \wp(Wffs(\mathcal{L}_{\mathsf{KR}}))$.
2. *If* $\langle \mathsf{L}_i, \Phi \rangle$ *is an argument label, then* $\mathbf{T}^U_j(\langle \mathsf{L}_i, \Phi \rangle)$*, with* $j \in \mathsf{Nat}$ *and* $\mathbf{T}^D_k(\langle \mathsf{L}_i, \Phi \rangle)$*, with* $k \in \mathsf{Nat}$ *are* dialectical labels *in* $\mathcal{L}_{\mathsf{Labels}}$*. For the sake of simplicity, we will write* $\mathbf{T}^D_k$ *to denote a generic dialectical label* $\mathbf{T}^D_k(\langle \mathsf{L}_i, \Phi \rangle)$*, for a given argument label* $\langle \mathsf{L}_i, \Phi \rangle$*. We will also write* $\mathbf{T}_k$ *to denote either the functor* $\mathbf{T}^D_k$ *or the functor* $\mathbf{T}^U_k$.
3. *If* $\mathbf{T}_1, \ldots, \mathbf{T}_k$ *are dialectical labels, then* $\mathbf{T}^U_n(\mathbf{T}_1, \ldots, \mathbf{T}_k)$*, with* $j \in \mathsf{Nat}$*,* $n \notin \{1 \ldots k\}$*, and* $\mathbf{T}^D_m(\mathbf{T}_1, \ldots, \mathbf{T}_k)$*, with* $k \in \mathsf{Nat}$ $m \notin \{1 \ldots k\}$ *will also be dialectical labels in* $\mathcal{L}_{\mathsf{Labels}}$.

Definition 2.4 (Defeasible Labeled Language $\mathcal{L}_{\mathsf{Arg}}$). *If* $\mathcal{L}_{\mathsf{Labels}}$ *is a labeling language, and* $\mathcal{L}_{\mathsf{KR}}$ *is a knowledge representation language, then the* defeasible labeled language*, denoted* $\mathcal{L}_{\mathsf{Arg}}$*, is defined as* $\mathcal{L}_{\mathsf{Arg}} = (\mathcal{L}_{\mathsf{Labels}}, \mathcal{L}_{\mathsf{KR}})$.

Definition 2.5 (Argumentative formula). *Given a declarative unit f=Label:α, where Label=*$[\Phi, \mathsf{L}]$ *is an argument label, then f will be called an* argumentative formula*. The set* Φ *in* $[\Phi, \mathsf{L}]$*:α will be called* support set *in α. The set* L *will provide a trace of the wffs used for deriving α. Alternatively, we will use the notation f=*$\mathcal{A}$*:α, where* $\mathcal{A} = \Phi$.

Since $\mathcal{L}_{\mathsf{KR}}$ is a Horn-like logic language, we will assume an underlying inference mechanism $\vdash_{\text{SLD}}$ equivalent to SLD resolution [Llo87], properly extended to handle a negated literal $\sim p$ as a new constant name *no_p*. Given $P \subseteq \mathsf{ProgClauses}(\mathcal{L}_{\mathsf{KR}})$, we will write $P \vdash_{\text{SLD}} \alpha$ to denote that α follows from P via $\vdash_{\text{SLD}}$.

Definition 2.6 (Contradictory set of wffs in $\mathcal{L}_{\mathsf{KR}}$). *Given a set P of wffs in* $\mathcal{L}_{\mathsf{KR}}$*, P will be called* contradictory *iff literals p y* $\overline{p}$ *can be derived via* $\vdash_{\text{SLD}}$ *from P. This situation will be denoted as* $P \vdash_{\text{SLD}} \bot$.

Definition 2.7 (Argumentative theory Γ). *A finite set* $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_k\}$ *with* $\Gamma \subseteq \mathsf{Wffs}(\mathcal{L}_{\mathsf{Arg}})$ *will be called an* argumentative theory*. We will assume that the set of non-defeasible information in any argumentative theory* Γ *is non-contradictory.*

In any argumentative theory we will distinguish:

- *Non-defeasible* information, characterized by formulas of the form $[\emptyset, \{\mathsf{n}_\mathsf{i}\}]$:$\alpha$, where $\alpha \in \mathsf{ProgClauses}(\mathcal{L}_{\mathsf{KR}})$.
- *Defeasible* information, characterized by formulas of the form $[\{\alpha\}, \{\mathsf{d}_\mathsf{i}\}]$:$\alpha$ where $\alpha \in \mathsf{Rules}(\mathcal{L}_{\mathsf{KR}})$.

Some distinguished sets will be considered. $\mathsf{Strict}(\Gamma)$ is the set of all non-defeasible formulas in Γ; $\mathsf{Defeasible}(\Gamma) = \Gamma - \mathsf{Strict}(\Gamma)$; $\Pi(\Gamma)$ is the set of all $\mathcal{L}_{\mathsf{KR}}$ formulas in Γ whose support set is empty; $\Delta(\Gamma)$ is the set of all $\mathcal{L}_{\mathsf{KR}}$ formulas in Γ whose support set is non-empty.

Example 2.8 (Adapted from [CSG00]). Consider an agent involved in controlling an engine with three switches $sw1$, $sw2$ and $sw3$. These switches regulate different features of the engine, such as pumping system, speed, etc. Suppose we have defeasible information about how this engine works.

$[\emptyset, \{\mathsf{n}_1\}]{:}\sim\!\mathit{fuel\_ok} \leftarrow \mathit{pump\_clogged}$
$[\emptyset, \{\mathsf{n}_2\}]{:}\mathit{sw1} \leftarrow$
$[\emptyset, \{\mathsf{n}_3\}]{:}\mathit{sw2} \leftarrow$
$[\emptyset, \{\mathsf{n}_4\}]{:}\mathit{sw3} \leftarrow$
$[\emptyset, \{\mathsf{n}_5\}]{:}\mathit{heat} \leftarrow$
$[\{\mathit{pump\_fuel\_ok} \leftarrow \mathit{sw1}\}, \{\mathsf{d}_1\}]{:}\mathit{pump\_fuel\_ok} \leftarrow \mathit{sw1}$
$[\{\mathit{fuel\_ok} \leftarrow \mathit{pump\_fuel\_ok}\}, \{\mathsf{d}_2\}]{:}\mathit{fuel\_ok} \leftarrow \mathit{pump\_fuel\_ok}$
$[\{\mathit{pump\_oil\_ok} \leftarrow \mathit{sw2}\}, \{\mathsf{d}_3\}]{:}\mathit{pump\_oil\_ok} \leftarrow \mathit{sw2}$
$[\{\mathit{oil\_ok} \leftarrow \mathit{pump\_oil\_ok}\}, \{\mathsf{d}_4\}]{:}\mathit{oil\_ok} \leftarrow \mathit{pump\_oil\_ok}$
$[\{\mathit{engine\_ok} \leftarrow \mathit{fuel\_ok}, \mathit{oil\_ok}\}, \{\mathsf{d}_5\}]{:}\mathit{engine\_ok} \leftarrow \mathit{fuel\_ok}, \mathit{oil\_ok}$
$[\{\sim\!\mathit{engine\_ok} \leftarrow \mathit{fuel\_ok}, \mathit{oil\_ok}, \mathit{heat}\}, \{\mathsf{d}_6\}]{:}\sim\!\mathit{engine\_ok} \leftarrow \mathit{fuel\_ok}, \mathit{oil\_ok}, \mathit{heat}$
$[\{\sim\!\mathit{oil\_ok} \leftarrow \mathit{heat}\}, \{\mathsf{d}_6\}]{:}\sim\!\mathit{oil\_ok} \leftarrow \mathit{heat}$
$[\{\mathit{pump\_clogged} \leftarrow \mathit{pump\_fuel\_ok}, \mathit{low\_speed}\}, \{\mathsf{d}_7\}]{:}\mathit{pump\_clogged} \leftarrow \mathit{pump\_fuel\_ok}, \mathit{low\_speed}$
$[\{\mathit{low\_speed} \leftarrow \mathit{sw2}\}, \{\mathsf{d}_8\}]{:}\mathit{low\_speed} \leftarrow \mathit{sw2}$
$[\{\sim\!\mathit{low\_speed} \leftarrow \mathit{sw2}, \mathit{sw3}\}, \{\mathsf{d}_9\}]{:}\sim\!\mathit{low\_speed} \leftarrow \mathit{sw2}, \mathit{sw3}$
$[\{\mathit{fuel\_ok} \leftarrow \mathit{sw3}\}, \{\mathsf{d}_{10}\}]{:}\mathit{fuel\_ok} \leftarrow \mathit{sw3}$

Fig. 1. Argumentative theory Γ_{engine} (example 2.8)

- If the pump is clogged, then the engine gets no fuel.
- When $sw1$ is on, normally fuel is pumped properly.
- When fuel is pumped properly, fuel usually works ok.
- When $sw2$ is on, usually oil is pumped.
- When oil is pumped, usually it works ok.
- When there is oil and fuel, usually the engine works ok.
- When there is fuel, oil, and heat, then the engine is usually not ok.
- When there is heat, normally there are oil problems.
- When fuel is pumped and speed is low, there are reasons to believe that the pump is clogged.
- When $sw2$ is on, usually speed is low.
- When $sw3$ is on, usually fuel is ok.

Suppose we know $sw1$, $sw2$ and $sw3$ are on, and there is heat. This situation can be modeled by the argumentative theory Γ_{engine} shown in figure 1.

3 Argument construction. Dialectical analysis

Our first goal is to define a logical system $(\Gamma, \mathrel{|\!\!\sim}_{Arg})$, where Γ is a knowledge base and $\mathrel{|\!\!\sim}_{Arg}$ is a consequence relation for constructing *generalized arguments*. The object language will be $\mathcal{L}_{\mathsf{Arg}}$. Imposing an additional requirement (minimality) will result in the final notion of *argument* as originally defined in [SL92]. Since arguments can be in conflict, we will then define an extended logical system $(\Gamma, \mathrel{|\!\!\sim}_{\mathcal{T}})$, in which a dialectical analysis among arguments can be carried out.

3.1 Formalizing argument construction

1. **Introducing non-defeasible information** (Intro-NR): Non-defeasible argumentative formulas can be introduced in a proof.

$$\frac{}{[\emptyset, \{\mathsf{n_i}\}]{:}\alpha}$$

for any $[\emptyset, \{\mathsf{n_i}\}]{:}\alpha \; \in \mathsf{Strict}(\Gamma)$.

2. **Introducing defeasible information** (Intro-RE): Defeasible argumentative formulas can be introduced in a proof whenever its support set is non-contradictory wrt $\Pi(\Gamma)$.

$$\frac{\Pi(\Gamma) \cup \Phi \not\vdash_{\text{SLD}} \bot}{\Gamma,\ [\Phi, \{\mathsf{d_i}\}]{:}\alpha}$$

for any $[\Phi, \{\mathsf{d_i}\}]{:}\alpha \; \in \mathsf{Defeasible}(\Gamma)$.

3. **Introducing conjunction** (Intro-$\wedge$): If $[\Phi_1, \mathsf{L_1}]{:}\alpha_1,\ [\Phi_2, \mathsf{L_2}]{:}\alpha_2,\ \ldots,\ [\Phi_k, \mathsf{L_k}]{:}\alpha_k$, are wffs such that $\Phi_1 \cup\ \Phi_2 \ldots \cup\ \Phi_k \not\vdash_{\text{SLD}} \bot$, then the conjunction $\alpha_1, \alpha_2, \ldots, \alpha_k$ can be derived.

$$\frac{\Gamma, \quad [\Phi_1, \mathsf{L_1}]{:}\alpha_1 \quad\quad [\Phi_2, \mathsf{L_2}]{:}\alpha_2 \quad\quad \ldots\ [\Phi_k, \mathsf{L_k}]{:}\alpha_k \quad\quad \Pi(\Gamma) \cup \bigcup_{i=1\ldots k} \Phi_i \not\vdash_{\text{SLD}} \bot}{\Gamma, \quad [\bigcup_{i=1\ldots k} \Phi_i, \bigcup_{\mathsf{i=1\ldots k}} \mathsf{L_i}]{:}\alpha_1, \alpha_2, \ldots, \alpha_k}$$

4. **Eliminating implication** (Elim-$\leftarrow$):

$$\frac{\Gamma,\ [\Phi_1, \mathsf{L_1}]{:}\beta{\leftarrow}\alpha_1, \ldots, \alpha_k \quad\quad [\Phi_2, \mathsf{L_2}]{:}\alpha_1, \ldots, \alpha_k \quad\quad \Pi(\Gamma) \cup \Phi_1 \cup\ \Phi_2 \not\vdash_{\text{SLD}} \bot}{\Gamma,\ [\Phi_1 \cup \Phi_2, \mathsf{L_1} \cup \mathsf{L_2}]{:}\beta}$$

Definition 3.1 (Generalized argument. Argument. Subargument). *Let Γ be an argumentative theory, and let $h \in \mathsf{Lit}(\mathcal{L}_{\mathsf{KR}})$ such that $\Gamma \mid\!\sim_{Arg} \mathcal{A}{:}h$ Then $\mathcal{A}$ will be called a* generalized argument *for h. If it is not the case that $\Gamma \mid\!\sim_{Arg} \mathcal{B}{:}h$, with $\mathcal{B} \subset \mathcal{A}$, then $\mathcal{A}{:}h$ is called a* minimal argument *or just* argument.[1] *An argument $\mathcal{A}{:}h$ is a* subargument *of another argument $\mathcal{B}{:}q$ if $\mathcal{A} \subset \mathcal{B}$.*

Example 3.2. From Γ_{engine} the arguments $\mathcal{A}{:}engine\_ok$, $\mathcal{B}{:}{\sim}fuel\_ok$, $\mathcal{C}{:}{\sim}low\_speed$, $\mathcal{D}{:}fuel\_ok$ and $\mathcal{E}{:}{\sim}engine\_ok$ can be derived via $\mid\!\sim_{Arg}$, with

$$\begin{aligned}
\mathcal{A} = \{\ & (pump\_fuel\_ok \leftarrow sw1),\ (pump\_oil\_ok \leftarrow sw2),\\
& (fuel\_ok \leftarrow pump\_fuel\_ok),\ (oil\_ok \leftarrow pump\_oil\_ok),\\
& (engine\_ok \leftarrow fuel\_ok, oil\_ok)\ \}\\
\mathcal{B} = \{\ & (pump\_fuel\_ok \leftarrow sw1),\ (low\_speed \leftarrow sw2),\\
& (pump\_clogged \leftarrow pump\_fuel\_ok, low\_speed)\ \}\\
\mathcal{C} = \{\ & ({\sim}low\_speed \leftarrow sw2, sw3)\ \}\\
\mathcal{D} = \{\ & ({\sim}low\_speed \leftarrow sw2, sw3)\ \}\\
\mathcal{E} = \{\ & (pump\_fuel\_ok \leftarrow sw1),\ (pump\_oil\_ok \leftarrow sw2),\\
& (fuel\_ok \leftarrow pump\_fuel\_ok),\ (oil\_ok \leftarrow pump\_oil\_ok),\\
& ({\sim}engine\_ok \leftarrow fuel\_ok, oil\_ok, heat)\ \}
\end{aligned}$$

The following proposition can be proven to hold in our framework.[2]

Proposition 3.3 (Basic properties of arguments). *Let Γ be an argumentative theory, and let $\mathcal{A}{:}h$ be an argument in Γ. Then the following properties hold:*

1. ***Derivability:*** *$\Pi(\Gamma)\ \cup\ \mathcal{A} \vdash_{\text{SLD}} h$.*
2. ***Non-contradiction:*** *$\Pi(\Gamma)\ \cup\ \mathcal{A} \not\vdash_{\text{SLD}} p, {\sim}p$, for any $p \in \mathcal{L}_{\mathsf{KR}}$.*
3. ***Minimality:*** *$\nexists \mathcal{B} \subset \mathcal{A}$ such that $\Pi(\Gamma)\ \cup\ \mathcal{B} \not\vdash_{\text{SLD}} h$.*

[1] When needed we will denote this situation by writing $\mathsf{Minimal}_{\Gamma}(\mathcal{A}{:}h)$, or just $\mathsf{Minimal}(\mathcal{A}{:}h)$.

[2] Proof not included for space reasons. The interested reader is referred to [Che01].

3.2 Conflict among arguments

Given an argument $\mathcal{A}$:h based on an argumentative theory Γ, there may exist other conflicting arguments based on Γ that *defeat* it. Conflict among arguments is captured by the notion of contradiction (def. 2.6). Defeat among arguments involves a partial order which establishes a preference criterion on them (*e.g.* specificity [SL92]).

Definition 3.4 (Counterargument). *Let Γ be an argumentative theory, and let $\mathcal{A}$:h and $\mathcal{B}$:q be arguments in Γ. Then $\mathcal{A}$:h* counter-argues *$\mathcal{B}$:q if there exists a subargument $\mathcal{B}'$:s of $\mathcal{B}$:q such that $\Pi(\Gamma) \cup \{h, s\}$ is contradictory. The argument $\mathcal{B}'$:s will be called* disagreement subargument.

Definition 3.5 (Preference order $\preceq$). *Let Γ be an argumentative theory, and let $\mathsf{Args}(\Gamma)$ be the set of arguments that can be obtained from Γ. A* preference order *$\preceq \subseteq \mathsf{Args}(\Gamma) \times \mathsf{Args}(\Gamma)$ is any partial order on $\mathsf{Args}(\Gamma)$.*

Definition 3.6 (Binary relations $\succ$ and $\asymp$). *Let $\mathcal{A}$:h and $\mathcal{B}$:q be arguments. Then we will write*

1. *$\mathcal{A}$:$h \succ \mathcal{B}$:q if $\mathcal{B}$:$q \preceq \mathcal{A}$:h, and $\mathcal{A}$:$h \not\preceq \mathcal{B}$:$q$. In this case we will say that $\mathcal{A}$:h is* strictly preferred *over $\mathcal{B}$:q.*
2. *$\mathcal{A}$:$h \asymp \mathcal{B}$:q if*
   (a) $\mathcal{A}$:$h \preceq \mathcal{B}$:q and $\mathcal{B}$:$q \preceq \mathcal{A}$:h.
   (b) $\mathcal{A}$:h and $\mathcal{B}$:q cannot be compared by $\preceq$ (i.e. $\mathcal{A}$:$h \not\preceq \mathcal{B}$:$q$ and $\mathcal{B}$:$q \not\preceq \mathcal{A}$:$h$).

Definition 3.7 (Defeater). *Let Γ be an argumentative theory, such that $\Gamma \mid\!\sim_{Arg} \mathcal{A}$:$h$ and $\Gamma \mid\!\sim_{Arg} \mathcal{B}$:$q$. We will say that $\mathcal{A}$:h* defeats *$\mathcal{B}$:q (or equivalently $\mathcal{A}$:h is a* defeater *for $\mathcal{B}$:q) if*

1. *$\mathcal{A}$:h counterargues $\mathcal{B}$:q, with disagreement subargument $\mathcal{B}'$:q'.*
2. *(a) It holds that $\mathcal{A}$:$h \succ \mathcal{B}'$:q'*
   (b) It holds that $\mathcal{A}$:$h \asymp \mathcal{B}'$:q'

In case 2a, we will say that $\mathcal{A}$:h is a proper *defeater for $\mathcal{B}$:q. In case 2b, we will say that $\mathcal{A}$:h is a* blocking *defeater for $\mathcal{B}$:q.*

Example 3.8. Consider the argumentative theory from example 2.8. Note that $\mathcal{B}$:$\sim$*fuel_ok*, and $\mathcal{E}$:$\sim$*engine_ok*, are counter-arguments for $\mathcal{A}$:*engine_ok*, whereas $\mathcal{C}$:$\sim$*low_speed* and $\mathcal{D}$:*fuel_ok* are counter-arguments for $\mathcal{B}$:$\sim$*fuel_ok*. In each of these cases, counter-arguments are also defeaters according to the specificity preference criterion [SL92].

3.3 Dialectical analysis

A dialectical label can be thought of as a *dialogue tree* between proponent and opponent. Branches of the tree are called *argumentation lines*. Special marks will be associated to dialectical labels in order to determine whether its elements correspond to *defeated* or *undefeated* arguments.

Definition 3.9 (Marking a dialectical label). *Given a dialectical formula $\mathbf{T}(\mathcal{A}, \dots)$:$h$, we will distinguish three kinds of 'marks' associated with $\mathbf{T}$:*

1. ** denoting "no mark"*

2. U denoting that $\mathcal{A}$ in **T** *is an undefeated argument for literal h.*
3. D denoting that $\mathcal{A}$ in **T** *is a defeated argument for literal h.*

Definition 3.10 (Argumentation line λ in T). *An* argumentation line λ *in a dialectical label* **T** *is a sequence $[\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k]$ where every $\mathcal{A}_i$, $i = 1 \ldots k$ is a formula in* **T**. *Formally:*

– *If $\mathbf{T}(\mathcal{A})$ is a dialectical label, then the sequence $\lambda = [\mathcal{A}]$ is an* argumentation line *in $\mathbf{T}(\mathcal{A})$.*
– *If $\mathbf{T}(\mathcal{A}, \mathbf{T}_1, \mathbf{T}_2, \ldots \mathbf{T}_k)$ is a dialectical label, and $\lambda = [\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_j]$ is an argumentation line associated with an immediate sublabel $\mathbf{T}_i$, for some $i = 1 \ldots k$, then the sequence that results from appending $[\mathcal{A}]$ with λ is an* argumentation line *in $\mathbf{T}(\mathcal{A}, \mathbf{T}_1, \mathbf{T}_2, \ldots \mathbf{T}_k)$.*

Given a dialectical label **T**, *we will denote as $Lines(\mathbf{T})$ the set of all argumentative lines associated with* **T**.

Definition 3.11 (Supporting and interfering argumentation line). *Given an argumentation line $\lambda = [\mathcal{A}_0, \mathcal{A}_1, \ldots \mathcal{A}_n]$, we will distinguish two sequences associated with λ:*

1. Supporting argumentation line λ_S, *formed by even-indexed members in λ, i.e. $\lambda_S = [\mathcal{A}_0, \mathcal{A}_2, \mathcal{A}_4 \ldots \mathcal{A}_{2k}]$.*
2. Interfering argumentation line λ_I, *formed by odd-indexed members of λ, i.e. $\lambda_I = [\mathcal{A}_1, \mathcal{A}_3, \mathcal{A}_5, \ldots \mathcal{A}_{2k+1}]$.*

Given a dialectical label **T**, *we will denote as $SLines(\mathbf{T})$ and $ILines(\mathbf{T})$ the set of all supporting and interfering argumentation lines, respectively, associated with* **T**.

All formulas in $\mathcal{L}_{\mathsf{Labels}}$ present in a supporting (resp. interfering) argumentation line should satisfy certain constraints, captured by the notions of *commitment set* and *circularity.*

Definition 3.12 (Commitment set). *Let $\lambda = [\mathcal{A}_0, \mathcal{A}_1, \ldots \mathcal{A}_n]$ be a supporting (resp. interfering) argumentation line. The* commitment set *associated with λ is defined as $ComSet(\lambda) = \bigcup_{i=1 \ldots n} A_i$.*

Definition 3.13 (Circularity). *Let $\lambda = [\mathcal{A}_0, \mathcal{A}_1, \ldots \mathcal{A}_n]$ be argumentation line. We will say that λ is* circular *if $\mathcal{A}_j \subseteq \mathcal{A}_i$, $0 \quad i < j \quad n$.*

Definition 3.14 (Acceptable argumentation line in T). *Let $\mathbf{T}(\mathcal{A}, \ldots)$:$h$ be a dialectical formula. Let $\mathbf{T}_i(\mathcal{B}_i, \ldots)$ be an immediate sublabel in $\mathbf{T}(\mathcal{A}, \ldots)$ (if any), associated with a dialectical formula $\mathbf{T}_i(\mathcal{B}_i, \ldots)$:$q_i$. Let $\mathbf{T}_1(\mathcal{B}_1, \ldots)$ be an immediate sublabel of $\mathbf{T}_i(\mathcal{B}_i, \ldots)$ (if any), associated with a dialectical formula $\mathbf{T}_j(\mathcal{C}_j, \ldots)$:$w_j$. Given the argumentation line $\lambda = [\mathcal{A}, \mathcal{B}_i, \mathcal{C}_j \ldots]$, we will say that it is* acceptable *if it satisfy the following conditions:*

Progressive defeat *The argumentative formula $\mathcal{B}_i$:q_i is a defeater for $\mathcal{A}$:h, and the formula $\mathcal{C}_j$:w_j is a defeater for $\mathcal{B}_i$:q_i. In particular, if $\mathcal{B}_i$:q_i is a blocking defeater for $\mathcal{A}$:h, then $\mathcal{C}_j$:w_j should be a proper defeater for $\mathcal{B}_i$:q_i.*
Non-contradiction *The commitment sets of supporting and interfering argumentation lines associated with $\lambda$are non-contradictory wrt Γ, i.e. $\Pi(\Gamma) \cup ComSet(\lambda_S) \not\vdash_{\text{SLD}} \perp$ and $\Pi(\Gamma) \cup ComSet(\lambda_I) \not\vdash_{\text{SLD}} \perp$.*
No circularity *λ is not circular.*

Definition 3.15 (Condition VSTree). *Let $\mathbf{T}'(\mathcal{B}, \ldots)$:$q$ and $\mathbf{T}(\mathcal{A}, \ldots)$:$h$ be two dialectical formulas. Let $Lines(\mathbf{T}') = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ be the set of argumentation lines in* **T***'. Then* **T***' will be a valid immediate sublabel associated with* **T** *, denoted $\mathsf{VSTree}(\mathcal{A}, \mathbf{T}'(\mathcal{B}, \ldots))$, if every argumentation line $[\mathcal{A} \mid \lambda_i]$, for $i = 1 \ldots k$, is an acceptable argumentation line.*

Definition 3.16 (Acceptable dialectical label). *Let* $\mathbf{T}(\mathcal{A},\dots)$*:*$h$ *be a dialectical formula, such that* $\mathbf{T}(\mathcal{A},\dots)$ *is its associated dialectical label. That label will be* acceptable *iff every argumentation line in* $\mathbf{T}(\mathcal{A},\dots)$ *is acceptable.*

3.4 Building dialectical labels

Inference rules in natural deduction style are provided, in order to characterize an inference relationship $\mid\!\sim_{\mathcal{T}}$ which allows to build dialectical labels. The natural deduction rules for $\mid\!\sim_{\mathcal{T}}$ are the following:

1. **Introducing a dialectical tree** (Intro-1D): A minimal argument $\mathcal{A}$:h constitutes an atomic dialectical formula. Formally:

$$\frac{\mathcal{A}{:}h \qquad \mathsf{Minimal}(\mathcal{A}{:}h)}{\mathbf{T}^{*}(\mathcal{A}){:}h}$$

2. **N-level dialectical tree** (Intro-ND): Intro-ND Given a dialectical tree with a single node, a new dialectical formula $\mathbf{T}$ can be built by introducing $\mathbf{T}^{*}_{1},\dots,\mathbf{T}^{*}_{k}$ as immediate sublabels such that they are valid wrt $\mathcal{A}$:h (according to def. 3.15).

$$\frac{\mathbf{T}^{*}(\mathcal{A}){:}h \qquad \mathbf{T}^{*}_{1}(\mathcal{B}_1,\dots){:}q_1 \quad \mathbf{T}^{*}_{k}(\mathcal{B}_k,\dots){:}q_k \qquad \mathsf{VSTree}(\mathcal{A},\,\mathbf{T}^{*}_{i})}{\mathbf{T}^{*}(\mathcal{A},\mathbf{T}^{*}_{1},\dots,\mathbf{T}^{*}_{k}){:}h}$$

1. **Marking an atomic dialectical formula:** (Mark-Atom) An atomic dialectical formula is *warranted* if there are no valid sublabels associated with it. Formally:

$$\frac{\mathbf{T}^{*}(\mathcal{A}){:}h}{\mathbf{T}^{U}(\mathcal{A}){:}h}$$

2. **Marking a dialectical formula as defeated**: (Mark-1D) A dialectical label can be marked as defeated if there exists at least one immediate sublabel as defeated if there exists *at least* one immediate sublabel marked as undefeated.

$$\frac{\mathbf{T}^{*}(\mathcal{A},\mathbf{T}^{*}_{1},\dots,\mathbf{T}^{*}_{i},\dots,\mathbf{T}_{k}){:}h \qquad \mathbf{T}^{U}_{i}(\mathcal{B}_i\dots){:}q_i \;:\mathsf{VSTree}(\mathcal{A},\mathbf{T}^{U}_{i})}{\mathbf{T}^{D}(\mathcal{A},\mathbf{T}^{*}_{1},\dots,\mathbf{T}^{*}_{i-1},\mathbf{T}^{U}_{i},\mathbf{T}^{*}_{i+1},\dots,\mathbf{T}^{*}_{k}){:}h}$$

for some $\mathbf{T}^{*}_{i}$, $i = 1\dots k$

3. **Marking a dialectical tree as undefeated**: (Mark-ND) A dialectical label can be marked as undefeated if *every* immediate dialectical sublabel can be marked as defeated.

$$\frac{\mathbf{T}^{*}(\mathcal{A},\mathbf{T}^{*}_{1},\dots,\mathbf{T}^{*}_{i},\dots,\mathbf{T}^{*}_{k}){:}h \qquad \mathbf{T}^{D}_{i}(\mathcal{B}_i,\dots){:}q_i \;:\mathsf{VSTree}(\mathcal{A},\mathbf{T}^{D}_{i})}{\mathbf{T}^{U}(\mathcal{A},\mathbf{T}^{D}_{1},\dots,\mathbf{T}^{D}_{i},\dots,\mathbf{T}_{k}){:}h}$$

$\forall\ \mathbf{T}^{*}_{i}$, $i = 1\dots k$

The notion of *warrant* corresponds to the notion of "ultimately undefeated". A warranted belief is that one which is accepted at some time of the dialectical process, and remains in that state.

Definition 3.17 (Warrant – preliminary version). *Let* $Cn^{k}_{*}(\Gamma)$ *be the set of all dialectical formulas that can be obtained from* Γ *via* $\mid\!\sim_{\mathcal{T}}$ *by* i *applications of inference rules (*$i \quad k$*). A literal* h *is said to be* warranted *iff* $\mathbf{T}^{U}(\mathcal{A},\dots)$*:*$h \in Cn^{k}_{*}(\Gamma)$*, and there is no* $k' > k$*, such that* $\mathbf{T}^{D}(\mathcal{A},\dots)$*:*$h \in (Cn^{k'}_{*}(\Gamma) \setminus Cn^{k}_{*}(\Gamma))$*.*

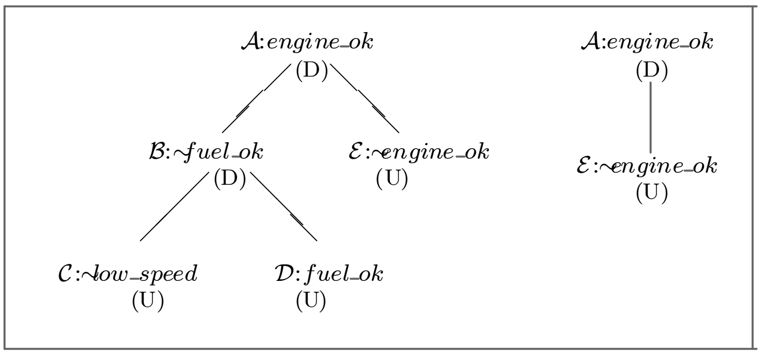
$\mathcal{A}$:*engine_ok*
(D)
$\mathcal{B}$:∼*fuel_ok*
(D)
$\mathcal{E}$:∼*engine_ok*
(U)
$\mathcal{C}$:∼*low_speed*
(U)
$\mathcal{D}$:*fuel_ok*
(U)
$\mathcal{A}$:*engine_ok*
(D)
$\mathcal{E}$:∼*engine_ok*
(U)

Fig. 2. Dialectical tree $\mathcal{T}_{\mathcal{A}:engine\_ok}$ and associated pruned tree $Pruned(\mathcal{T}_{\mathcal{A}:engine\_ok})$

This approach resembles Pollock's original ideas of (ultimately) justified belief [Pol95]. Note that it forces us to compute the closure under $\mid\!\sim_{\mathcal{T}}$ in order to determine whether a literal is warranted or not. Fortunately this is not the case, since warrant can be captured in terms of a *precedence relation* "$\sqsubset$ " between dialectical labels. Informally, we will write $\mathbf{T} \sqsubset \mathbf{T}$' whenever $\mathbf{T}$ reflects a state in a dialogue which is previous to $\mathbf{T}$' (in other words, $\mathbf{T}$' stands for a dialogue which evolves from $\mathbf{T}$ by incorporating new arguments). A *final label* is a label that cannot be further extended.

Definition 3.18 (Warrant – final version). *Let Γ be an argumentative theory, such that $\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_i^U(\mathcal{A}, \ldots){:}h$ and $\mathbf{T}_i^U$ is a final label (i.e., it is not the case that $\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_j^D(\mathcal{A}, \ldots){:}h$ and $\mathbf{T}_i^U \sqsubset \mathbf{T}_j^D$). Then $\mathbf{T}_i^U(\mathcal{A}, \ldots){:}h$ is a* warrant. *We will also say that h is a warranted literal, or alternatively that $\mathcal{A}{:}h$ is a* warrant *in Γ, abbreviating this as $\Gamma \mid\!\sim_{\mathcal{T}} \mathcal{A}{:}h^U$ (or just $\Gamma \mid\!\sim_{\mathcal{T}} h^U$).*

Example 3.19. Consider the argumentative theory from example 2.8 and the arguments and defeat relations from examples 3.2 and 3.8. From the argumentative theory Γ_{engine} the following formulas can be inferred via $\mid\!\sim_{\mathcal{T}}$:

$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_1^*(\mathcal{A}){:}engine\_ok \quad \text{via Intro-1D} \tag{1}$$
$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_2^*(\mathcal{B}){:}{\sim}fuel\_ok \quad \text{via Intro-1D} \tag{2}$$
$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_3^*(\mathcal{C}){:}{\sim}low\_speed \quad \text{via Intro-1D} \tag{3}$$
$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_4^*(\mathcal{D}){:}fuel\_ok \quad \text{via Intro-1D} \tag{4}$$
$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_5^*(\mathcal{E}){:}{\sim}engine\_ok \quad \text{via Intro-1D} \tag{5}$$
$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_2^*(\mathcal{B}, \mathbf{T}_3^*(\mathcal{C}), \mathbf{T}_4^*(\mathcal{D})){:}{\sim}fuel\_ok \quad \text{via Intro-ND, (3) and (4)} \tag{6}$$
$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_1^*(\mathcal{A}, \mathbf{T}_2^*(\mathcal{B}, \mathbf{T}_3^*(\mathcal{C}), \mathbf{T}_4^*(\mathcal{D})), \mathbf{T}_5^*(\mathcal{E})){:}engine\_ok \quad \text{via Intro-ND and (6)} \tag{7}$$
$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_5^U(\mathcal{E}){:}{\sim}engine\_ok \quad \text{via Mark-Atom} \tag{8}$$
$$\Gamma \mid\!\sim_{\mathcal{T}} \mathbf{T}_1^D(\mathcal{A}, \mathbf{T}_2^*(\mathcal{B}, \mathbf{T}_3^*(\mathcal{C}), \mathbf{T}_4^*(\mathcal{D})), \mathbf{T}_5^U(\mathcal{E})){:}engine\_ok \quad \text{via Mark-1D and (8)} \tag{9}$$

Note that the formula obtained in step (7) has a final label associated with it, since it cannot be 'expanded' from previous formulas. Hence, following definition 3.18, we can conclude that *engine_ok* is not warranted.

4 Some interesting theoretical results

Next we will discuss some interesting theoretical results that were obtained from the preceding formalization.[3] First, we will present an equivalence theorem which links different ways of computing warrant. We will then discuss some details of a model-theoretic approach for LDS_{ar}. We will also introduce the role of transformation rules in LDS_{ar}. Finally, we will detail how different argumentation frameworks can be subsumed within our approach.

4.1 Equivalence results

As a result from previous research work, the notion of *dialectical tree* for an argument $\mathcal{A}{:}q$ (denoted $\mathcal{T}_{\mathcal{A}:q}$) was introduced. This notion corresponds to a procedural, top-down approach to compute warrant. That tree could be *pruned* according to an $\alpha - \beta$ pruning criterion, resulting in a new, smaller tree $Pruned(\mathcal{T}_{\mathcal{A}:q})$. On the contrary, the LDS approach provides a bottom-up construction procedure, resulting in complex labels that are built from more simple ones.

Example 4.1. Consider the dialectical label rooted in $\mathcal{A}{:}engine\_ok$ associated with the final dialectical label in example 3.19. This label can be represented as a *dialectical tree* $\mathcal{T}_{\mathcal{A}:engine\_ok}$, and it can be represented as shown in figure 2 (left). The root node of $\mathcal{T}_{\mathcal{A}:engine\_ok}$ is labeled as D-node. Note that it is not necessary to compute the whole tree in order to label the root node as defeated. In fact, considering $Pruned(\mathcal{T}_{\mathcal{A}:engine\_ok})$ as shown in figure 2 (right), an equivalent answer would have been obtained.

An equivalence theorem proves that warrant can be computed by either of these approaches. Pruning aspects in the top-down approach correspond to selection of inference rules in the bottom-up approach.

Theorem 4.2 (Equivalence of top-down and bottom-up computation of warrant). *Given an argumentative theory Γ, the following three cases are equivalent:*

1. *The root of $\mathcal{T}_{\mathcal{A}:q}$ is marked as U-node.*
2. *The root of $Pruned(\mathcal{T}_{\mathcal{A}:q})$, is marked as U-node.*
3. *It is the case that $\Gamma \mid\!\sim_{\mathcal{T}} \mathcal{A}{:}h^U$.*

4.2 Semantics: soundness and completeness

A semantics for LDS_{ar} based on special models (called *Π-models*) was developed. *Π-models* satisfy the non-defeasible knowledge, $\Pi(\Gamma)$. We will say that a model $\mathcal{M}$ is a *Π-model* for $\mathcal{A}{:}h$ if $\mathcal{M}$ is a model for $\mathcal{A}$. The class $\mathfrak{M}_{\mathcal{A}}$ of all Π-models for $\mathcal{A}{:}h$ is called *Π-frame* for $\mathcal{A}{:}h$.

Semantical counterparts of the relationships of counterargument and defeat between argumentative formulas were defined. Preference order between formulas was mapped into a set-inclusion relation between classes of *activation models*. The defeat relation between Π-frames resulted in the notion of *frame tree*. A frame tree rooted in *Π-frame* for $\mathcal{A}{:}h$. captures the semantical equivalent of dialectical tree (according to def. 3.16). A frame tree allows to determine whether a given frame is *finally preferred* (*i.e.*, it is ultimately undefeated wrt other conflicting frames).

The syntactic characterization of warrant (def. 3.18) is equivalent to its semantical counterpart. Therefore the consequence relation $\mid\!\sim_{\mathcal{T}}$ in LDS_{ar} satisfies both soundness and completeness wrt the proposed semantics, as shown in the following theorem:

[3] For space reasons the proof of the theorems and propositions in this section are not included in this paper. For details the reader is referred to [Che01].

Theorem 4.3 (Soundness and completeness of warrant). *Let $\mathfrak{M}_0$ be a frame for an argument $\mathcal{A}$:h. Then $\Gamma \mid\!\sim_T \mathcal{A}:h^U$ iff $\mathfrak{M}_0$ is a finally preferred frame wrt Γ.*

4.3 Transformation properties for LDS_{ar}

The complexity of the analysis required to determine whether an given argumentative formula is warranted is directly linked to the size of the associated argumentative theory Γ. Since Γ can be arbitrarily large, it is desirable (if possible) to simplify the information it contains whenever possible. Such a simplification should be formalized in terms of semantics-preserving *transformation* (or *rewriting*) rules.

In [BD98], a number of transformation rules were introduced which allow to "simplify" a normal logic program (*nlp*) P to get its well-founded semantics (WFS). The application of these rules led to a new, simplified *NLP* P' from which its WFS can be easily read off. In the case of LDS_{ar}, similar rules were developed for simplifying both defeasible and non-defeasible rules. Two distinguished variants of LDS_{ar} deserved particular attention, namely SA_{not} and SA_{neg} (LDS_{ar} restricted to default and strict negation, resp.). The relation between these variants of *SDE* and normal logic programming was explored. Different criteria under which both strict and defeasible rules could be rewritten into a simpler but semantically equivalent form were defined.

4.4 Defining a taxonomy of argumentative systems

Another interesting issue concerns the definition of *variants* for LDS_{ar}. Since LDS_{ar} is a logical framework, its knowledge-encoding capabilities are determined by the underlying logical language, whereas the inference power is characterized by its natural deduction rules. Adopting a different KR language or modifying the existing inference rules will lead to different variants of LDS_{ar}. Thus, for instance, adopting a full first-order language will lead to a logical system with a behavior similar to the SL framework [SL92]. On the other hand, restricting the KR language to Horn clauses will result in a formulation closer to normal logic programming (NLP) under well-founded semantics.[4] Figure 3 summarizes some of these variants and shows how they can be related to some existing argumentation frameworks, such as Simari-Loui's [SL92], MTDR [SCG94], *DeLP* [Gar00] and NLP (normal logic programming), conceptualized in an argumentative setting as suggested in [BDKT97].

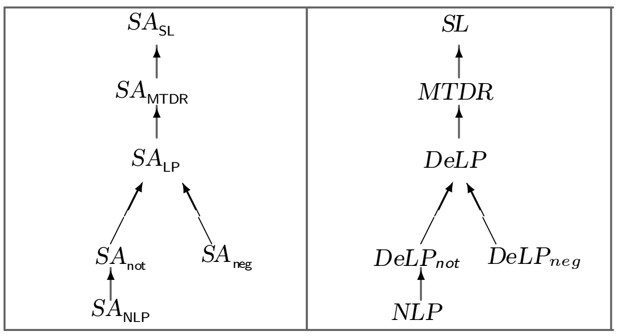

Fig. 3. A taxonomy relating the expressive power of LDS_{ar} and different argumentation systems

[4] A full discussion of different argumentative frameworks encompassed by LDS_{ar} can be found in [Che01].

5 Conclusions

Labelled Deductive Systems offer a powerful tool for formalizing different aspects of defeasible argumentation. On the one hand, the notion of label allows to capture the concept of argument as a set of wffs supporting a given proposition. On the other hand, the concept of dialectical tree can be also captured by a complex label, defined in terms of more simple ones. The LDS_{ar} framework has been defined upon these two notions.

During the last decade, a 'clash of intuitions' has appeared within the argumentation community, where different, alternative approaches have been intended. As we have briefly sketched in section 4.4, having a logical system such as LDS_{ar} makes it easier to analyze, compare and relate different features associated with existing argumentative frameworks, providing at the same time a test-bed for studying other related issues (such as argumentation protocols, resource-bounded reasoning, etc.). Research in this direction is currently being pursued.

References

[BD98] S. Brass and J. Dix. Characterizations of the Disjunctive Well-founded Semantics: Confluent Calculi and Iterated GCWA. *Journal of Automated Reasoning*, 20(1):143–165, 1998.

[BDKT97] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1-2):63–101, 1997.

[Che01] Carlos Iván Chesñevar. *Formalización de los Procesos de Argumentación Rebatible como Sistemas Deductivos Etiquetados.* PhD thesis, Departamento de Ciencias de la Computación - U.N.S. - Argentina, January 2001.

[CML00] Carlos Iván Chesñevar, Ana Maguitman, and Ronald Loui. Logical Models of Argument. *ACM Computing Surveys*, 32(4):337–383, December 2000.

[CS00] Carlos I. Chesñevar and Guillermo R. Simari. Formalizing Defeasible Argumentation using Labelled Deductive Systems. *Journal of Computer Science & Technology*, 1(4):18–33, 2000.

[CSG00] Carlos I. Chesñevar, Guillermo R. Simari, and Alejandro García. Pruning Search Space in Defeasible Argumentation. In *Proc. of the Workshop on Advances and Trends in Artificial Intelligence.* XX International Conference of the SCCC – Santiago, Chile, November 2000.

[Gab96] Dov Gabbay. *Labelling Deductive Systems (vol.1).* Oxford University Press (Volume 33 of Oxford Logic Guides), 1996.

[Gar00] Alejandro J. García. *Programación en Lógica Rebatible: Lenguaje, Semántica Operacional y Paralelismo.* PhD thesis, Dep. de Cs. de la Computación, Universidad Nacional del Sur, December 2000.

[GL90] Michael Gelfond and Vladimir Lifschitz. Logic programs with classical negation. In *Proceedings of the 7th International Conference on Logic Programming. Jerusalem*, June 1990.

[Llo87] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, New York, second edition edition, 1987.

[Pol95] John L. Pollock. *Cognitive Carpentry: A Blueprint for How to Build a Person.* Massachusetts Institute of Technology, 1995.

[PV99] Henry Prakken and Gerard Vreeswijk. Logics for Defeasible Argumentation. In Dov Gabbay, editor, *Handbook of Philosophical Logic.* Kluwer Academic Publisher, 1999.

[Rei80] Raymond Reiter. A Logic for Default Reasoning. *Artificial Intelligence*, 13(1,2):81–132, April 1980.

[SCG94] Guillermo R. Simari, Carlos I. Chesñevar, and Alejandro J. García. The role of dialectics in defeasible argumentation. In *Anales de la XIV Conferencia Internacional de la Sociedad Chilena para Ciencias de la Computación.* Universidad de Concepción, Concepción (Chile), November 1994.

[SL92] Guillermo R. Simari and Ronald P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53:125–157, 1992.