A Distributed Argumentation Framework using Defeasible Logic Programming

Matthias Thimm   Gabriele Kern-Isberner

Technische Universität Dortmund

May 29, 2008
Outline

1 Defeasible Logic Programming

2 The Distributed Argumentation Framework

3 Remarks and conclusion
Outline

1. Defeasible Logic Programming
2. The Distributed Argumentation Framework
3. Remarks and conclusion
Overview

- DeLP (Defeasible Logic Programming) consists of facts, strict and defeasible rules

  \[
  \text{Bird(tweety).} \quad \text{(fact)} \\
  \text{Bird}(X) \leftarrow \text{Penguin}(X). \quad \text{(strict rule)} \\
  \text{Flies}(X) \leftarrow \text{Bird}(X). \quad \text{(defeasible rule)}
  \]

- A defeasible logic program (de.l.p.) \( \mathcal{P} \) is a tuple \( \mathcal{P} = (\Pi, \Delta) \) with a set \( \Pi \) of facts and strict rules and a set \( \Delta \) of defeasible rules.
Let $\mathcal{P} = (\Pi, \Delta)$ be a de.l.p.

**Definition (Argument, subargument)**

$\langle A, h \rangle$ with $A \subseteq \Delta$ is an *argument* iff
- $A \cup \Pi \not\models h$
- $A \cup \Pi \not\models \bot$
- $A$ is minimal

$\langle A_1, h_1 \rangle$ is a *subargument* of $\langle A_2, h_2 \rangle$ iff $A_1 \subseteq A_2$. 
Arguments and counterarguments

Let $\mathcal{P} = (\Pi, \Delta)$ be a de.l.p.

**Definition (Argument, subargument)**

$\langle A, h \rangle$ with $A \subseteq \Delta$ is an argument iff

- $A \cup \Pi \not\models h$
- $A \cup \Pi \not\models \bot$
- $A$ is minimal

$\langle A_1, h_1 \rangle$ is a subargument of $\langle A_2, h_2 \rangle$ iff $A_1 \subseteq A_2$.

**Definition (Counterargument)**

$\langle A_1, h_1 \rangle$ is a counterargument of $\langle A_2, h_2 \rangle$ at a literal $h$ iff

\[
\exists \langle A, h \rangle : A \subseteq A_2 : \Pi \cup \{h, h_1\} \not\models \bot \quad (h \text{ and } h_1 \text{ disagree})
\]
Let $\mathcal{P}$ be a \textit{de.l.p.}

**Definition (Acceptable argumentation line)**

$\Lambda = [\langle A_1, h_1 \rangle, \ldots, \langle A_n, h_n \rangle]$ is an \textit{acceptable argumentation line} iff

1. $\Lambda$ is finite
Let $\mathcal{P}$ be a \textit{de.l.p}.

**Definition (Acceptable argumentation line)**

$\Lambda = [\langle A_1, h_1 \rangle, \ldots, \langle A_n, h_n \rangle]$ is an \textit{acceptable argumentation line} iff

1. $\Lambda$ is finite,
2. every argument is an attack on its predecessor;
Let $\mathcal{P}$ be a de.l.p.

**Definition (Acceptable argumentation line)**

$\Lambda = [\langle A_1, h_1 \rangle, \ldots, \langle A_n, h_n \rangle]$ is an *acceptable argumentation line* iff

1. $\Lambda$ is finite,
2. every argument is an attack on its predecessor; there are no two consecutive blocking attacks (given a preference relation under arguments)
Acceptable argumentation lines

Let $\mathcal{P}$ be a *de.l.p.*

**Definition (Acceptable argumentation line)**

$\Lambda = [\langle A_1, h_1 \rangle, \ldots, \langle A_n, h_n \rangle]$ is an *acceptable argumentation line* iff

1. $\Lambda$ is finite,
2. every argument is an attack on its predecessor; there are no two consecutive blocking attacks (given a preference relation under arguments),
3. the set of supporting arguments is consistent with respect to $\Pi$,
4. the set of interfering arguments is consistent with respect to $\Pi$
Acceptable argumentation lines

Let $P$ be a *de.l.p.*

**Definition (Acceptable argumentation line)**

$\Lambda = [\langle A_1, h_1 \rangle, \ldots, \langle A_n, h_n \rangle]$ is an acceptable argumentation line iff

1. $\Lambda$ is finite,
2. every argument is an attack on its predecessor; there are no two consecutive blocking attacks (given a preference relation under arguments),
3. the set of supporting arguments is consistent with respect to $\Pi$,
4. the set of interfering arguments is consistent with respect to $\Pi$,
5. no argument $\langle A_k, h_k \rangle$ is a subargument of a preceding argument.
The warrant procedure

Representation of the dialectical process in a *dialectical tree*:

![Diagram]

- $\langle A, a \rangle$
- $\langle B_1, \neg b \rangle$
- $\langle B_2, \neg b \rangle$
- $\langle B_3, \neg b \rangle$
- $\langle C_1, \neg f \rangle$
- $\langle C_2, \neg f \rangle$
- $\langle D_1, \neg h \rangle$
The warrant procedure

Representation of the dialectical process in a dialectical tree:
Definition (Warrant)

A literal \( h \) is \textit{warranted}, iff there exists an argument \( \langle A, h \rangle \) for \( h \), such that the root of the marked dialectical tree \( T^*_A,h \) is marked “undefeated”. Then \( \langle A, h \rangle \) is a \textit{warrant} for \( h \).
Outline

1. Defeasible Logic Programming

2. The Distributed Argumentation Framework

3. Remarks and conclusion
Overview 1/2

The Distributed Argumentation Framework
Definition (Global belief base)

A *global belief base* $\Pi$ is a non-contradictory set of facts and strict rules.

→ The global belief base consists of common beliefs.
Definition (Global belief base)

A *global belief base* $\Pi$ is a non-contradictory set of facts and strict rules.

→ The global belief base consists of common beliefs.

Definition (Local belief base)

Let $\Delta$ be a set of defeasible rules and $\Pi$ a global belief base. If $\Delta \cup \Pi$ is consistent (treating defeasible rules as strict rules), $\Delta$ is called *local beliefbase* relative to $\Pi$.

→ A local belief base reflects an agent’s own beliefs besides the common beliefs.
The moderator is the central component in the architecture. He receives queries from outside, coordinates the argumentation process and returns answers.
The moderator is the central component in the architecture. He receives queries from outside, coordinates the argumentation process and returns answers.

**Definition (Moderator)**

A *moderator* is a tuple \((\mu, \chi, \eta)\) with

- a decision function \(\mu\) (evaluates a set of dialectical trees regarding a given query),
The moderator is the central component in the architecture. He receives queries from outside, coordinates the argumentation process and returns answers.

**Definition (Moderator)**

A *moderator* is a tuple $(\mu, \chi, \eta)$ with

- a decision function $\mu$ (evaluates a set of dialectical trees regarding a given query),
- an analysis function $\chi$ (evaluates the marking of the root argument of a given dialectical tree)
The moderator is the central component in the architecture. He receives queries from outside, coordinates the argumentation process and returns answers.

**Definition (Moderator)**

A *moderator* is a tuple $(\mu, \chi, \eta)$ with

- a decision function $\mu$ (evaluates a set of dialectical trees regarding a given query),
- an analysis function $\chi$ (evaluates the marking of the root argument of a given dialectical tree) and
- an acceptance function $\eta$ (checks whether a given argumentation line acceptable).
The Distributed Argumentation Framework

The Moderator 2/2

The diagram shows a flow of information and processes within a distributed system. It includes:

- **External Communication Module**: Receives queries and sends answers.
- **Coordination Module**: Handles coordination between the external and internal communication modules.
- **Analysis Module**: Processes arguments and interacts with the global beliefbase.
- **Internal Communication Module**: Sends arguments to the external communication module.
- **Global Beliefbase**: Stores and updates the global state of the system.

The flow of information is as follows:

- Queries from external and internal communication are processed.
- Responses are generated and sent.
- Arguments are analyzed and updated in the global beliefbase.

This framework supports efficient and coordinated argumentation in distributed environments.
An agent generates initial arguments for a given literal and counterarguments for a given argument.
An agent generates initial arguments for a given literal and counterarguments for a given argument.

Definition (Agent)

An agent is a tuple \((\Delta, \phi, \psi, \eta)\) with
- a local belief base \(\Delta\),
- a root argument function \(\phi\) (returns all arguments from \(\Delta\) for a given literal),
Agents 1/2

An agent generates initial arguments for a given literal and counterarguments for a given argument.

**Definition (Agent)**

An agent is a tuple \((\Delta, \varphi, \psi, \eta)\) with

- a local belief base \(\Delta\),
- a root argument function \(\varphi\) (returns all arguments from \(\Delta\) for a given literal),
- a counterargument function \(\psi\) (returns all counterargument for a given argument).
Agents 1/2

An agent generates initial arguments for a given literal and counterarguments for a given argument.

Definition (Agent)
An agent is a tuple \((\Delta, \varphi, \psi, \eta)\) with
- a local belief base \(\Delta\),
- a root argument function \(\varphi\) (returns all arguments from \(\Delta\) for a given literal),
- a counterargument function \(\psi\) (returns all counterargument for a given argument)
- and an acceptance function \(\eta\) (checks whether a given argumentation line acceptable).
Agents 2/2

The Distributed Argumentation Framework

Thimm, Kern-Isberner (TU Dortmund)  Distributed DeLP  May 29, 2008
Definition (Argumentation-based multi agent system (ArgMAS))

An ArgMAS is a tuple \((M, \Pi, \{A_1, \ldots, A_n\})\) with a moderator \(M\), a global belief base \(\Pi\) and agents \(A_1, \ldots, A_n\).
The argumentation process

Definition (Argumentation-based multi agent system (ArgMAS))

An ArgMAS is a tuple \((M, \Pi, \{A_1, \ldots, A_n\})\) with a moderator \(M\), a global belief base \(\Pi\) and agents \(A_1, \ldots, A_n\).

Definition (Argumentation product)

Let \(h\) be a query (a literal) and \(T\) an ArgMAS. An argumentation product \(\varphi\) of \(T\) and \(h\) is a dialectical tree with:

1. The root argument of \(\varphi\) is an element of \(\varphi_j(h)\) for a \(j \in \{1, \ldots, n\}\).
The argumentation process

Definition (Argumentation-based multi agent system (ArgMAS))

An ArgMAS is a tuple \((M, \Pi, \{A_1, \ldots, A_n\})\) with a moderator \(M\), a global belief base \(\Pi\) and agents \(A_1, \ldots, A_n\).

Definition (Argumentation product)

Let \(h\) be a query (a literal) and \(T\) an ArgMAS. An argumentation product \(\nu\) of \(T\) anf \(h\) is a dialectical tree with:

1. The root argument of \(\nu\) is an element of \(\varphi_j(h)\) for a \(j \in \{1, \ldots, n\}\)
2. For every path \(\Lambda = [\langle A_1, h_1 \rangle, \ldots, \langle A_n, h_n \rangle]\) in \(\nu\) it holds for the set \(K\) of all children of \(\langle A_n, h_n \rangle\)

\[
K = \{ \langle B, h' \rangle | \langle B, h' \rangle \in \psi_1(\Lambda) \cup \cdots \cup \psi_n(\Lambda) \land \eta(\Lambda + \langle B, h' \rangle) = 1 \}.
\]
An application scenario

- Assume two agents, acting as accuser and defender in a legal dispute.
- Then the moderator can be identified with the judge.
- A reasonable query for this multi agent system would be the question of guilt of the accused.
- As a first step to answer this query, the judge asks the accuser and the defender to propose initial arguments for and against the statement “The accused is guilty”.
- Both, the defender and the accuser, can react to the arguments of their counterpart with counterarguments.
- Eventually, the judge analyses the resulting argumentation lines and returns “guilty” or “not guilty” to the questioner, i.e. the people.
Outline

1. Defeasible Logic Programming
2. The Distributed Argumentation Framework
3. Remarks and conclusion
Remarks and conclusion

- We presented a distributed and centralized approach for defeasible argumentation, which is useful in dispute scenarios.
- The proposed system was applied to a real world legal dispute and turned out well.
Remarks and conclusion

- We presented a distributed and centralized approach for defeasible argumentation, which is useful in dispute scenarios.
- The proposed system was applied to a real world legal dispute and turned out well.
- It can be shown that every de.l.p. can be translated into the proposed framework while preserving answer behaviour.
- There are instances of the distributed framework which can not be modeled in general DeLP.
- We are currently investigating these relationships more closely.
We presented a distributed and centralized approach for defeasible argumentation, which is useful in dispute scenarios.

The proposed system was applied to a real world legal dispute and turned out well.

It can be shown that every de.l.p. can be translated into the proposed framework while preserving answer behaviour.

There are instances of the distributed framework which can not be modeled in general DeLP.

We are currently investigating these relationships more closely.

Thank you for your attention