The DIAMOND System for Argumentation
Preliminary Report

Stefan Ellmauthaler    Hannes Strass

Computer Science Institute
Leipzig University
Germany

ASPOCP 2013
August 25th 2013

UNIVERSITÄT LEIPZIG
Outline

1 Motivation

2 Background
   - Abstract Argumentation Frameworks
   - Abstract Dialectical Frameworks

3 DIAMOND

4 Conclusion & Future Work
Abstract Argumentation Frameworks (AFs)

- syntactically: directed graphs

\[ a \rightarrow b \]
\[ c \rightarrow d \]

- conceptually: nodes are arguments, edges denote attacks between arguments

- semantically: *extensions* are sets of “acceptable” arguments

- immensely popular in the argumentation community

- drawback: can only express attack
Motivation: ADFs

Recent improvements

Abstract Dialectical Frameworks (ADFs)

- generalise AFs, arguments are now called statements
- can also (although less directly) be visualised as graphs
- edges express that there is some relationship between the two statements
- relationship need not be “attack”, precise nature specified by acceptance condition for each statement
- acceptance condition specifies status of node given status of direct predecessors
Motivation: Argumentation in practice
Recent Projects and Ideas

- lawsuit analysis with argumentation\(^1\)
- argumentation for discussion analysis on social media (e.g. facebook)\(^2\)
- argumentation for blog-discussions\(^3\)


Abstract Argumentation
is for determining acceptance of abstract arguments

- argumentation framework
  \[ F = (A, R) \]
- \( A \) . . set of arguments
- \( R \subseteq A \times A \) . . attack relation
- argument \( a \in A \) is \textit{defended} by a set \( S \subseteq A \) if all attackers of \( a \) are attacked by \( S \):
  \[
  (\forall b \in A)(bRa \implies (\exists c \in S)cRb)
  \]

---

AF Semantics
are defined via extensions

for AF $F = (A, R)$, a set $S \subseteq A$ of arguments is

- **conflict-free** iff for all $a, b \in S$, $(a, b) \notin R$
- a conflict-free set $S$ is
  - **admissible** iff it defends all arguments it contains
  - **preferred** iff it is $\subseteq$-maximal admissible
  - **complete** iff it contains exactly the arguments it defends
  - **grounded** iff it is $\subseteq$-minimal complete
  - **stable** iff it attacks all arguments not in $S$
AF Semantics

an example framework

- AF $F = (A, R)$ with $A = \{a, b, c, d\}$ and $R = \{(a, b), (c, d), (d, c)\}$:

  ![Diagram](attachment:image.png)

- grounded extension: $G = \{a\}$
- stable extensions: $E_1 = \{a, c\}$ and $E_2 = \{a, d\}$
- preferred extensions: $E_1, E_2$
- complete extensions: $G, E_1, E_2$
**Definition: Abstract Dialectical Framework**

An abstract dialectical framework (ADF) is a triple \( D = (S, L, C) \),

- \( S \) ... set of statements (correspond to AF arguments)
- \( L \subseteq S \times S \) ... links
- \( C = \{C_s\}_{s \in S} \) ... acceptance conditions

- links denote some kind of dependency relation
- acceptance condition: Boolean function \( C_s : 2^{par(s)} \rightarrow \{t, f\} \)
- here: \( C_s \) often specified by propositional formula \( \varphi_s \)

---

Abstract Dialectical Frameworks

Example

\[ \varphi_a = t \]  \hspace{1cm}  \varphi_b = b \\
\[ \varphi_c = a \land b \]  \hspace{1cm}  \varphi_d = \neg b \\

\begin{tikzpicture}
    \node (a) at (0,0) {$a$};
    \node (b) at (2,0) {$b$};
    \node (c) at (1,-1) {$c$};
    \node (d) at (2,-1) {$d$};

    \path[->,thick]
    (a) edge (b)
    (a) edge (c)
    (b) edge (d)
    (c) edge (d);
\end{tikzpicture}
### Truth values, interpretations

- **truth values**: true \( t \), false \( f \), unknown \( u \)
- **interpretation**: \( v : S \rightarrow \{ t, f, u \} \)
- **interpretations** can be represented as consistent sets of literals

### Information ordering

- \( u <_i t \) and \( u <_i f \) \hspace{1cm} \text{(as usual } x \leq_i y \text{ iff } x <_i y \text{ or } x = y \)
- **consensus** \( \sqcap \) is greatest lower bound \( \text{w.r.t.} \leq_i \):
  - \( t \sqcap t = t \) and \( f \sqcap f = f \), otherwise \( x \sqcap y = u \)
- **information ordering** generalised to interpretations:
  - \( v_1 \leq_i v_2 \) iff \( v_1(s) \leq_i v_2(s) \) for all \( s \in S \)
Abstract Dialectical Frameworks

Semantics

**Characteristic Operator**

- for a valuation $\nu$, we define $[\nu]_2 = \{ \nu \leq_i w \mid w \text{ two-valued} \}$
- for ADF $D$, we define an operator $\Gamma_D$ on interpretations
- for interpretation $\nu : S \rightarrow \{ t, f, u \}$, the operator yields a new interpretation (the consensus over $[\nu]_2$)

\[
\Gamma_D(\nu) : S \rightarrow \{ t, f, u \} \quad s \mapsto \bigcap \{ w(\varphi_s) \mid w \in [\nu]_2 \}
\]

**Semantics**

- two-valued $\nu$ is a model of $D$ iff $\nu(s) = \nu(\varphi_s)$ for all $s \in S$
- $\nu$ is the grounded model of $D$ iff it is the least fixpoint of $\Gamma_D$
Abstract Dialectical Frameworks

Semantics: Example

\[ \varphi_a = t \quad \varphi_b = b \]

\[ \varphi_c = a \land b \quad \varphi_d = \neg b \]

- models:
  \[ v_1 = \{a \mapsto t, b \mapsto t, c \mapsto t, d \mapsto f\} \stackrel{\hat{\cdot}}{=} \{a, b, c, \neg d\} \]
  \[ v_2 = \{a \mapsto t, b \mapsto f, c \mapsto f, d \mapsto t\} \stackrel{\hat{\cdot}}{=} \{a, \neg b, \neg c, d\} \]

- grounded model:  \[ v_3 = \{a \mapsto t, b \mapsto u, c \mapsto u, d \mapsto u\} \stackrel{\hat{\cdot}}{=} \{a\} \]
Definition: Admissible

Interpretation $\nu$ is *admissible* for ADF $D$ iff $\nu \leq_i \Gamma_D(\nu)$.

- intuitively: does not contain too much information
- example ADF has 16 admissible interpretations:
Abstract Dialectical Frameworks
Complete Semantics

Definition: Complete

Interpretation $\nu$ is complete for ADF $D$ iff $\nu = \Gamma_D(\nu)$.

- Complete interpretations are stationary w.r.t. revision operator

S. Ellmauthaler, H. Strass
The DIAMOND System for Argumentation
ASPOCP 2013, Istanbul
Set of encodings utilizing the Potsdam Answer Set Solving Collection\textsuperscript{6}

- Python script for command-line-interface
- ECL\textsuperscript{i}PS\textsuperscript{e}-Prolog program for fast input-format conversions

Propositional Formula Representation

- Statements are defined via the unary predicate `statement(a)`.
- Their acceptance condition is a propositional formula, represented by `ac(s, φ)`, where `φ` is a propositional formula in prefix notation.
- Links between statements are implicitly defined by variable occurrence in `φ`.
- Allowed constants in `φ` are `c(v)`, `c(f)`.
- Allowed operators are `neg`, `and`, `or`, `imp`, and `iff`.

Example

```
ac(a,c(v)). ac(b,b).
ac(c, and(a,b)). ac(d, neg(b)).
```
Predicates

- unary predicate \( s(a) \). to represent statements
- binary predicate \( l(b,a) \). to represent links
- unary and ternary predicates \( ci \) and \( co \) to model the Boolean acceptance function

Example

\[
\begin{align*}
  & s(a). \ s(b). \ s(c). \ s(d). \\
  & l(a,c). \ l(b,b). \ l(b,c). \ l(b,d). \\
  & ci(a). \\
  & co(b). \ ci(b,1,b). \\
  & co(c). \ co(c,1,a). \ co(c,2,b). \ ci(c,3,a). \ ci(c,3,b). \\
  & ci(d). \ co(d,1,b). 
\end{align*}
\]
Semantics are based upon fixpoint computation of $\Gamma_D$

- Binary predicates $\text{in}(X,I)$ and $\text{out}(X,I)$ represent an interpretation at $\text{step}(I)$
- Operator applied if $\text{step}(I)$ exists
“Interesting” links at step I

\[
\begin{align*}
\text{ciui}(S,J,I) &\quad \text{:- lin}(X,S,I), \not\text{ci}(S,J,X), \text{ci}(S,J). \\
\text{ciui}(S,J,I) &\quad \text{:- lout}(X,S,I), \text{ci}(S,J,X). \\
\text{cii}(S,J,I) &\quad \text{:- not ciui}(S,J,I), \text{ci}(S,J), \text{step}(I).
\end{align*}
\]

Validity check

\[
\begin{align*}
\text{pmodel}(S,I) &\quad \text{:- cii}(S,J,I). \\
\text{pmodel}(S,I) &\quad \text{:- verum}(S), \text{step}(I). \\
\text{pmodel}(S,I) &\quad \text{:- not lin}(S,I), \text{ci}(S), \text{step}(I). \\
\text{valid}(S,I) &\quad \text{:- pmodel}(S,I), \not\text{imodel}(S,I).
\end{align*}
\]

Fixpoint check

\[
\begin{align*}
\text{nofp}(I) &\quad \text{:- in}(X,I), \not\text{valid}(X,I), \text{step}(I). \\
\text{nofp}(I) &\quad \text{:- valid}(X,I), \not\text{in}(X,I), \text{step}(I). \\
\text{nofp}(I) &\quad \text{:- out}(X,I), \not\text{unsat}(X,I), \text{step}(I). \\
\text{nofp}(I) &\quad \text{:- unsat}(X,I), \not\text{out}(X,I), \text{step}(I). \\
\text{fp}(I) &\quad \text{:- not nofp}(I), \text{step}(I).
\end{align*}
\]
Admissible/Complete

```
step(0).
in(S,0):s(S).
out(S,0):s(S).
:- in(S,0), out(S,0).
:- in(S), not valid(S,0).
:- out(S), not unsat(S,0).

:- nofp(0).
```
DIAMOND utilizes ASP for implementing abstract argumentation (like ASPARTIX\textsuperscript{7} does for AFs)

All currently known representations for ADFs are understood by DIAMOND

More complex semantics (e.g. preferred) can be implemented with Meta-ASP\textsuperscript{8}

ADFsys\textsuperscript{9} is a similar system, but it

- uses different semantics, and
- it already needs disjunctive programs for the computation of $\Gamma_D$

\textsuperscript{7}Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. “Answer-set programming encodings for argumentation frameworks”. In: Argument and Computation 1.2 (2010), pages 147–177.


Future Work

- Implementation of further semantics for ADFs
- Add more usability to DIAMOND
- Use ADFs and DIAMOND for discussion analysis in the web
- ...
Thank you!
AF Semantics: Alternative Formulation
via operators

**Definition**
For AF $F = (A, R)$, define two operators
- $U_F(S) = \{ a \in A \mid S \text{ does not attack } a \}$ (Pollock)
- $V_F(S) = \{ a \in A \mid S \text{ defends } a \}$ (Dung)

- $U_F$ is $\subseteq$-antimonotone ($S_1 \subseteq S_2$ implies $U_F(S_2) \subseteq U_F(S_1)$)
- $V_F$ is $\subseteq$-monotone ($S_1 \subseteq S_2$ implies $V_F(S_1) \subseteq V_F(S_2)$)

**Lemma (Dung)**
For any AF $F$, we have $U_F^2 = V_F$. 

S. Ellmauthaler, H. Strass
The DIAMOND System for Argumentation
ASPOCP 2013, Istanbul
AF Semantics: Alternative Formulation

are defined via extensions

for AF $F = (A, R)$, a set $S \subseteq A$ of arguments is

- **conflict-free** iff $S \subseteq U_F(S)$
- **admissible** iff $S \subseteq U_F(S)$ and $S \subseteq V_F(S)$
- **preferred** iff it is $\subseteq$-maximal admissible
- **complete** iff $S \subseteq U_F(S)$ and $S = V_F(S)$
- **grounded** iff it is the $\subseteq$-least fixpoint of $V_F$
- **stable** iff $S = U_F(S)$
Stable Models
for ADFs

Definition: Associated Extension

For an interpretation \( \nu \), the set \( E_{\nu} = \{ s \in S \mid \nu(s) = t \} \) defines the unique extension associated with \( \nu \).

Definition: Stable Model

Let \( D = (S, L, C) \) be an ADF with \( C = \{ \varphi_s \}_{s \in S} \). A two-valued model \( \nu \) of \( D \) is a stable model of \( D \) iff \( E_{\nu} \) equals the grounded extension of the reduced ADF \( D^\nu = (E_{\nu}, L^\nu, C^\nu) \), where

- \( L^\nu = L \cap (E_{\nu} \times E_{\nu}) \) and
- for \( s \in E_{\nu} \) we set \( \varphi^\nu_s = \varphi_s[b/f : \nu(b) = f] \).
Abstract Dialectical Frameworks

Semantics: Example

\[ \varphi_a = t \quad \varphi_b = b \]

\[ \varphi_c = a \land b \quad \varphi_d = \neg b \]

- \( v_1 \widehat{=} \{a, b, c, \neg d\} \): reduct \( D^{v_1} \) with grounded extension \( \{a\} \), \( \{a\} \neq E_{v_1} \), thus \( v_1 \) not stable (statements \( b \) and \( c \) unjustified)

- \( v_2 \widehat{=} \{a, \neg b, \neg c, d\} \): reduct \( D^{v_2} \) with grounded extension \( \{a, d\} = E_{v_2} \), thus \( v_2 \) stable
Thank you!