A Dynamic Logic Framework for Abstract Argumentation

Andreas Herzig
University of Toulouse, IRIT-CNRS, France

joint work with Sylvie Doutre and Laurent Perrussel

Cardiff Argumentation Forum
Cardiff, July 6, 2016

Why is dynamic logic relevant for argumentation frameworks and their modification?

- Dung argumentation frameworks usually encoded in propositional logic
  - characterise argumentation semantics by means of propositional formulas:

$$\mathsf{Fml}(\mathsf{Stable}) = \bigwedge_{a \in \mathcal{A}} \left(\mathsf{In}_a \leftrightarrow \neg \bigvee_{b \in \mathcal{A}} (\mathsf{In}_b \wedge \mathsf{Att}_{b,a}) \right)$$

- sometimes also encoded in QBF
  - useful to prove complexity results
- dynamic logic will give us more for the same price:
  - construct extensions = execute a program
  - modify an argumentation framework = execute a program
  - import complexity results

Outline

Assignments and QBF

Which logical language for knowledge representation?

- boolean formulas: talk about a single valuation (alias a state)

$$\begin{array}{lll} s \models p & \text{if} & p \in s \\ s \models \neg\varphi & \text{if} & s \not\models \varphi \\ & \ldots & \end{array}$$

- Quantified Boolean Formulas (QBF): talk about valuations and their modification

$$\begin{array}{lllll} s \models \exists p.\varphi & \text{if} & s \cup \{p\} \models \varphi & \text{or} & s \setminus \{p\} \models \varphi \\ s \models \forall p.\varphi & \text{if} & s \cup \{p\} \models \varphi & \text{and} & s \setminus \{p\} \models \varphi \end{array}$$

- Dynamic Logic of Propositional Assignments (DL-PA): also about valuations and their modification, but more fine-grained than QBF

$$\begin{array}{lll} s \models \langle +p \rangle \varphi & \text{if} & s \cup \{p\} \models \varphi \\ s \models \langle -p \rangle \varphi & \text{if} & s \setminus \{p\} \models \varphi \end{array}$$

$\Rightarrow$ assignments of propositional variables to truth values

Assignments and propositional quantification have same expressivity

- from DL-PA to QBF:

$$\begin{aligned} \langle +p \rangle \varphi &= \exists p.(p \land \varphi) \\ \langle -p \rangle \varphi &= \exists p.(\neg p \land \varphi) \end{aligned}$$

- from QBF to DL-PA:

$$\begin{aligned} \exists p.\varphi &= \langle +p \rangle \varphi \lor \langle -p \rangle \varphi \\ \forall p.\varphi &= \langle +p \rangle \varphi \land \langle -p \rangle \varphi \end{aligned}$$

- ...but DL-PA moreover has complex assignment programs

Assignment programs as relations on valuations

- atomic

$$s \xrightarrow{+p} s \cup \{p\}$$

$$s \xrightarrow{-p} s \setminus \{p\}$$

- sequential composition

$$s_1 \xrightarrow{\pi_1;\pi_2} s_3 \text{ iff there is } s_2 \text{ such that } s_1 \xrightarrow{\pi_1} s_2 \xrightarrow{\pi_2} s_3$$

- nondeterministic composition

$$s \xrightarrow{\pi_1 \sqcup \pi_2} s' \text{ iff } s \xrightarrow{\pi_1} s' \text{ or } s \xrightarrow{\pi_2} s'$$

- finite iteration ('Kleene star')

$$s \xrightarrow{\pi^*} s' \text{ iff there is } n \text{ such that } s \xrightarrow{\pi^n} s'$$

- test

$$s \xrightarrow{\varphi?} s' \text{ iff } s = s' \text{ and } s \models \varphi$$

- converse, intersection,...

Capturing standard programming constructions in dynamic logic

$$\begin{aligned}
\textbf{skip} &= \top? \\
\textbf{fail} &= \bot? \\
\textbf{if } \varphi \textbf{ then } \pi_1 \textbf{ else } \pi_2 &= (\varphi?; \pi_1) \sqcup (\neg\varphi?; \pi_2) \\
\textbf{while } \varphi \textbf{ do } \pi &= (\varphi?; \pi)^*; \neg\varphi?
\end{aligned}$$

Language of DL-PA

- grammar of programs π and formulas φ:

$$\pi \quad ::= \quad +p \mid -p \mid \pi;\pi \mid \pi \sqcup \pi \mid \pi^* \mid \pi^{-1} \mid \varphi?$$
$$\varphi \quad ::= \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\pi\rangle\varphi \mid [\pi]\varphi$$

where p ranges over set of propositional variables $\mathbb{P}$

- reading:

$$\langle\pi\rangle\varphi = \text{“}\varphi \text{ is true after some execution of } \pi\text{”}$$
$$[\pi]\varphi = \text{“}\varphi \text{ is true after every execution of } \pi\text{”} = \neg\langle\pi\rangle\neg\varphi$$

- therefore, more compactly:

$$\exists p.\varphi = \langle +p \sqcup -p\rangle\varphi$$
$$\forall p.\varphi = [+p \sqcup -p]\varphi$$

Semantics of DL-PA: (1) formulas

- *valuation* = subset of $\mathbb{P}$
- model of a formula φ = set of valuations $\mathtt{Mod}(\varphi) \subseteq 2^{\mathbb{P}}$

$$
\begin{aligned}
\mathtt{Mod}(p) &= \{s \ : \ p \in s\} \\
\mathtt{Mod}(\top) &= 2^{\mathbb{P}} \\
\mathtt{Mod}(\bot) &= \emptyset \\
\mathtt{Mod}(\neg\varphi) &= \ldots \\
\mathtt{Mod}(\varphi \vee \psi) &= \ldots \\
\mathtt{Mod}(\langle\pi\rangle\varphi) &= \Big\{s \ : \ \text{there is } s' \text{ such that } s \xrightarrow{\pi} s' \ \ \& \ \ s' \in \mathtt{Mod}(\varphi)\Big\} \\
\mathtt{Mod}([\pi]\varphi) &= \Big\{s \ : \ \text{for every } s' : s \xrightarrow{\pi} s' \Longrightarrow s' \in \mathtt{Mod}(\varphi)\Big\}
\end{aligned}
$$

- write $(s, s') \in \mathtt{Mod}(\pi)$ instead of $s \xrightarrow{\pi} s'$

Semantics of DL-PA: (1) formulas

- *valuation* = subset of $\mathbb{P}$
- model of a formula φ = set of valuations $\texttt{Mod}(\varphi) \subseteq 2^{\mathbb{P}}$

$$\begin{aligned}
\texttt{Mod}(p) &= \{s \ : \ p \in s\} \\
\texttt{Mod}(\top) &= 2^{\mathbb{P}} \\
\texttt{Mod}(\bot) &= \emptyset \\
\texttt{Mod}(\neg\varphi) &= \ldots \\
\texttt{Mod}(\varphi \vee \psi) &= \ldots \\
\texttt{Mod}(\langle\pi\rangle\varphi) &= \Big\{s \ : \ \text{there is } s' \text{ such that } s \xrightarrow{\pi} s' \ \ \& \ \ s' \in \texttt{Mod}(\varphi)\Big\} \\
\texttt{Mod}([\pi]\varphi) &= \Big\{s \ : \ \text{for every } s' : s \xrightarrow{\pi} s' \Longrightarrow s' \in \texttt{Mod}(\varphi)\Big\}
\end{aligned}$$

- write $(s, s') \in \texttt{Mod}(\pi)$ instead of $s \xrightarrow{\pi} s'$

Semantics of DL-PA: (2) programs

- model of a program π = relation on the set of valuations $2^{\mathbb{P}}$

$$\begin{aligned}
\mathtt{Mod}(+p) &= \{(s, s') \ : \ s' = s \cup \{p\}\} \\
\mathtt{Mod}(-p) &= \{(s, s') \ : \ s' = s \setminus \{p\}\} \\
\mathtt{Mod}(\pi; \pi') &= \mathtt{Mod}(\pi) \circ \mathtt{Mod}(\pi') \\
\mathtt{Mod}(\pi \sqcup \pi') &= \mathtt{Mod}(\pi) \cup \mathtt{Mod}(\pi') \\
\mathtt{Mod}(\pi^*) &= \big(\mathtt{Mod}(\pi)\big)^* \ = \ \bigcup_{k \in \mathbb{N}_0} \big(\mathtt{Mod}(\pi)\big)^k \\
\mathtt{Mod}(\pi^{-1}) &= \big(\mathtt{Mod}(\pi)\big)^{-1} \\
\mathtt{Mod}(\varphi?) &= \{(s, s) \ : \ s \in \mathtt{Mod}(\varphi)\}
\end{aligned}$$

Properties of DL-PA

- compares favourably to PDL:
  - PSPACE complete both for model checking and satisfiability checking [Balbiani, Herzig & Troquard 2014]
    - PDL: SAT is EXPTIME complete
  - consequence relation is compact
    - PDL: fails
- interesting generalisation of QBF:
  - same expressivity, same complexity
  - conjecture: more succinct

Outline

Dung argumentation frameworks [Dung, 1995]

- graph $(\mathcal{A}, \mathcal{R})$
  - $\mathcal{A} = \{a_1, \ldots, a_n\}$ (finite set of abstract arguments)
  - $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ (attack relation)
- accepted arguments $E \subseteq \mathcal{A}$ ('extensions')
  - which are 'good'?
  - many candidate semantics

Argumentation frameworks in propositional logic

1. introduce *attack variables*:

$$\mathtt{ATT} = \{\mathsf{Att}_{a,b} \; : \; (a,b) \in \mathcal{A} \times \mathcal{A}\}$$

   ⇒ describe attack relation by a propositional formula:

$$\mathsf{Fml}(\mathcal{R}) = \left(\bigwedge_{(a,b)\in\mathcal{R}} \mathsf{Att}_{a,b} \right) \land \left(\bigwedge_{(a,b)\in(\mathcal{A}\times\mathcal{A})\setminus\mathcal{R}} \neg \mathsf{Att}_{a,b} \right)$$

2. introduce *acceptance variables*:

$$\mathtt{IN} = \{\mathsf{In}_{a_1}, \ldots, \mathsf{In}_{a_n}\}$$

   ⇒ describe extensions $E \subseteq \mathcal{A}$ by propositional formula:

$$\mathsf{Fml}(E) = \left(\bigwedge_{a\in E} \mathsf{In}_a \right) \land \left(\bigwedge_{a\in \mathtt{IN}\setminus E} \neg \mathsf{In}_a \right)$$

3. define semantics . . .

Argumentation frameworks in propositional logic: defining semantics

- stable:

$$\mathsf{Fml}(\mathsf{Stable}) = \bigwedge_{a \in \mathcal{A}} \left(\mathsf{In}_a \leftrightarrow \neg \bigvee_{b \in \mathcal{A}} (\mathsf{In}_b \land \mathsf{Att}_{b,a}) \right)$$

- admissible:

$$\mathsf{Fml}(\mathsf{Adm}) = \bigwedge_{a \in \mathcal{A}} \left(\mathsf{In}_a \to \bigwedge_{b \in \mathcal{A}} \left(\mathsf{Att}_{b,a} \to \left(\neg \mathsf{In}_b \land \bigvee_{c \in \mathcal{A}} (\mathsf{In}_c \land \mathsf{Att}_{c,b}) \right) \right) \right)$$

- complete:

$$\mathsf{Fml}(\mathsf{Compl}) = \ldots$$

- ...

[Besnard & Doutre, NMR 2004; Baroni & Giacomin, AIJ 2007]
[Baroni & Giacomin, 2009; Besnard, Doutre & H, IPMU 2014]

Argumentation frameworks in propositional logic: two examples

$a \longrightarrow b$ $(\mathcal{A}, \mathcal{R}_1)$

$a \rightleftarrows b$ $(\mathcal{A}, \mathcal{R}_2)$

- description of attack relation:

$$\mathrm{Fml}(\mathcal{R}_1) = \neg \mathrm{Att}_{a,a} \wedge \neg \mathrm{Att}_{b,b} \wedge \mathrm{Att}_{a,b} \wedge \neg \mathrm{Att}_{b,a}$$
$$\mathrm{Fml}(\mathcal{R}_2) = \neg \mathrm{Att}_{a,a} \wedge \neg \mathrm{Att}_{b,b} \wedge \mathrm{Att}_{a,b} \wedge \mathrm{Att}_{b,a}$$

- $(\mathcal{A}, \mathcal{R}_2)$ has two stable extensions: $E_a = \{a\}$ and $E_b = \{b\}$
- in logic: $\mathrm{Fml}(\mathcal{R}_2) \wedge \mathrm{Fml}(\mathrm{Stable})$ has two models

$$s_a = \{\mathrm{Att}_{a,b}, \mathrm{Att}_{b,a}, \mathrm{In}_a\}$$
$$s_b = \{\mathrm{Att}_{a,b}, \mathrm{Att}_{b,a}, \mathrm{In}_b\}$$

Argumentation frameworks in propositional logic: general pattern

Dung	propositional logic
arg. framework $(\mathcal{A}, \mathcal{R})$	$\mathsf{Fml}(\mathcal{R}) = \left(\bigwedge_{(a,b)\in\mathcal{R}} \mathsf{Att}_{a,b}\right) \wedge \left(\bigwedge_{(a,b)\notin\mathcal{R}} \neg\mathsf{Att}_{a,b}\right)$
candidate extension $E \subseteq \mathcal{A}$	$\mathsf{Fml}(E) = \left(\bigwedge_{a\in E} \mathsf{In}_a\right) \wedge \left(\bigwedge_{a\notin E} \neg\mathsf{In}_a\right)$
semantics σ	$\mathsf{Fml}(\sigma) = \ldots$
σ-extensions of $(\mathcal{A}, \mathcal{R})$	models of $\mathsf{Fml}(\mathcal{R}) \wedge \mathsf{Fml}(\sigma)$
E is a σ-extension of $(\mathcal{A}, \mathcal{R})$	$\models \big(\mathsf{Fml}(\mathcal{R}) \wedge \mathsf{Fml}(E)\big) \to \mathsf{Fml}(\sigma)$

Proposition (Besnard & Doutre, NMR 2004)

E stable extension of $(\mathcal{A}, \mathcal{R})$ *iff* $\models \big(\mathsf{Fml}(\mathcal{R}) \wedge \mathsf{Fml}(E)\big) \to \mathsf{Fml}(\mathsf{Stable})$

E admissible set of $(\mathcal{A}, \mathcal{R})$ *iff* $\models \big(\mathsf{Fml}(\mathcal{R}) \wedge \mathsf{Fml}(E)\big) \to \mathsf{Fml}(\mathsf{Adm})$

E complete extension of $(\mathcal{A}, \mathcal{R})$ *iff* $\models \big(\mathsf{Fml}(\mathcal{R}) \wedge \mathsf{Fml}(E)\big) \to \mathsf{Fml}(\mathsf{Compl})$

Outline

Building extensions in DL-PA

$$\boxed{\mathtt{makeExt}^{\sigma} = \mathtt{vary}(\mathtt{IN}); \mathsf{Fml}(\sigma)?}$$

where $\mathtt{vary}(\mathtt{IN}) = (+\mathsf{In}_{a_1} \sqcup -\mathsf{In}_{a_1}); \cdots; (+\mathsf{In}_{a_n} \sqcup -\mathsf{In}_{a_n})$

- $\mathtt{vary}(\mathtt{IN})$ does not modify attack variables
  $\Rightarrow$ keeps given argumentation framework fixed
- $\mathtt{vary}(\mathtt{IN})$ nondeterministically modifies acceptance variables
  $\Rightarrow$ visits all candidate extensions
- $\mathsf{Fml}(\sigma)?$ tests whether the valuation is a σ-extension
  $\Rightarrow$ output of program will be a σ-extension

Proposition

Let σ be any semantics that can be described by a propositional formula. Then

$$\mathtt{Mod}(\mathtt{makeExt}^{\sigma}) = \{(s_1, s_2) \ : s_2 \in \mathtt{Mod}(\mathsf{Fml}(\sigma)) \textit{ and } s_1 \cap \mathtt{ATT} = s_2 \cap \mathtt{ATT}\}$$

Building extensions in DL-PA

- $\texttt{makeExt}^{\sigma}$ follows a simple 'generate-and-test' schema
- more sophisticated algorithms: [Nofal et al., AIJ 2014;...]
- building blocks:

$$\texttt{AttByAcc}(a) = \bigvee_{b \in \mathcal{A}} \Big(\mathsf{Dec}_b \land \mathsf{In}_b \land \mathsf{Att}_{b,a} \Big)$$

$$\texttt{DefendedByAcc}(a) = \bigwedge_{b \in \mathcal{A}} \Big(\mathsf{Att}_{b,a} \to \bigvee_{c \in \mathcal{A}} \Big(\mathsf{Dec}_c \land \mathsf{In}_c \land \mathsf{Att}_{c,b} \Big) \Big)$$

Building extensions in DL-PA: a better algorithm

$$\mathop{;}_{a \in \mathcal{A}} -\mathsf{Dec}_a \;;$$

$$\mathop{;}_{a \in \mathcal{A}} \Big(\textbf{if} \bigwedge_{b \in \mathcal{A}} \neg \mathsf{Att}_{b,a} \textbf{ then } +\mathsf{In}_a ; +\mathsf{Dec}_a \textbf{ else skip}\Big) \;;$$

$$\textbf{while} \bigvee_a \neg \mathsf{Dec}_a \textbf{ do}$$

$$\quad \textbf{while} \bigvee_a \Big(\big(\mathtt{AttByAcc}(a) \vee \mathtt{DefendedByAcc}(a)\big)\Big) \textbf{ do}$$

$$\quad\quad \mathop{;}_{a \in \mathcal{A}} \Big(\textbf{if } \mathtt{AttByAcc}(a) \textbf{ then } -\mathsf{In}_a ; +\mathsf{Dec}_a \textbf{ else skip}\Big) \;;$$

$$\quad\quad \mathop{;}_{a \in \mathcal{A}} \Big(\textbf{if } \mathtt{DefendedByAcc}(a) \textbf{ then } +\mathsf{In}_a ; +\mathsf{Dec}_a \textbf{ else skip}\Big)$$

$$\quad \textbf{if} \bigwedge_a \mathsf{Dec}_a \textbf{ then skip else } \bigsqcup_{a \in \mathcal{A}} \Big(\neg \mathsf{Dec}_a? ; (+\mathsf{In}_a \sqcup -\mathsf{In}_a) ; +\mathsf{Dec}_a\Big)$$

$$\mathtt{Fml}(\sigma)?$$

Building extensions in DL-PA: verification

- prove π^σ correct:

$$\mathtt{Mod}(\pi^\sigma) = \mathtt{Mod}(\mathtt{makeExt}^\sigma)$$

  $\Rightarrow$ can be done in the logic!

- so:
  a skeptically σ-accepted in $(\mathcal{A}, \mathcal{R})$ iff $\models_{\text{DL-PA}} \mathtt{Fml}(\mathcal{R}) \to [\pi^\sigma]\text{In}_a$
  a credulously σ-accepted in $(\mathcal{A}, \mathcal{R})$ iff $\models_{\text{DL-PA}} \mathtt{Fml}(\mathcal{R}) \to \langle\pi^\sigma\rangle\text{In}_a$

Reasoning about argument influence in DL-PA (cf. [Murphy et al., this workshop])

- hypotheses:
  - background framework $(\mathcal{A}, \mathcal{R})$
  - persuader and persuadee agree on $\mathcal{R}$
  - only a subset of $\mathcal{A}$ has been put on the table (by persuader)
  - effect of putting forward some argument a?
- in DL-PA:
  - introduce new propositional variables:

$$\text{Pub}_a = \text{“}a \text{ is public”}$$

  - definition of extension takes only public arguments into account

$$\text{Fml(Stable)} = \bigwedge_{a \in \mathcal{A}} \left(\text{Pub}_a \to \left(\text{In}_a \leftrightarrow \neg \bigvee_{b \in \mathcal{A}} (\text{Pub}_b \wedge \text{In}_b \wedge \text{Att}_{b,a}) \right) \right)$$

  - persuader puts forward a = assignment '$+\text{Pub}_a$'
  - persuader reasons:

$$\overset{?}{\models}_{\text{DL-PA}} \texttt{Fml}(\mathcal{R}) \to \langle +\text{Pub}_a \rangle [\texttt{makeExt}^{\sigma}] \text{In}_b$$

Outline

Belief change operations

$B \circ A$ = modification of belief base B accomodating input A

- many operations $\circ$ in the literature; most prominent:
  - Winslett's possible models approach PMA [Winslett, AAAI 1988]
  - Winslett's standard semantics WSS [Winslett 1995]
  - Forbus's update operation [Forbus, IJCAI 1989]
  - Dalal's revision operation [Dalal, AAAI 1988]
- concrete operations: different from parametrised operations *à la* AGM or KM (that are built from orderings or distances)
- semantical
  1. state = subset of $\mathbb{P}$
  2. model of formula = set of states
  3. result of update/revision = set of states

     $B \circ A$ subset of $2^{\mathbb{P}}$

Forbus's update operation [Forbus, IJCAI 1989]

- Hamming distance between states

  $$\mathtt{h}(\{p,q\},\{q,r\}) = \mathrm{card}(\{p,r\}) = 2$$

- update B by A = "for each B-state, find the *closest* A-states w.r.t. $\mathtt{h}(.,.)$; then collect the resulting states"
  1. $s \diamond^{\text{forbus}} A = \left\{s' \in \mathtt{Mod}(A) \ : \ \text{there is no } s'' \text{ s.th. } \mathtt{h}(s,s'') < \mathtt{h}(s,s')\right\}$
  2. $S \diamond^{\text{forbus}} A = \bigcup_{s \in S} s \diamond^{\text{forbus}} A$

Example

$\neg p \wedge \neg q \diamond^{\text{forbus}} p \vee q = \mathtt{Mod}(p \oplus q)$ (exclusive $\vee$)

$p \oplus q \diamond^{\text{forbus}} p = \mathtt{Mod}(p)$

Dalal's revision operation [Dalal, AAAI 1988]

- revise B by A = "go to the A-states that are closest to B w.r.t. $\mathrm{h}(.,.)$"
- ...

The embeddings in a nutshell

- polynomial translations into DL-PA
  - object language operat*ors* (vs. metalanguage operat*ions*)
  - regression ⇒ representation of $B \circ A$ in propositional logic
- update by atomic formula is 'built in':
  - $+p$ = "update by p!"
    $-p$ = "update by $\neg p$!"
- update by complex formula A = complex assignment π_A
  - depends on belief change operation:

$$\pi^{\text{wss}}_{\neg p \vee \neg q} = -p \sqcup -q \sqcup (-p; -q)$$
$$\pi^{\text{pma}}_{\neg p \vee \neg q} = \ldots$$

  - to be proved for each change operation $\circ^{op}$:

$$B \circ^{op} A \;=\; \mathtt{Mod}\Big(\Big\langle (\pi^{op}_A)^{-1} \Big\rangle B\Big)$$

- details in the next slides

Some useful programs and formulas

- nondeterministically assign truth values to $p_1, \ldots, p_n$:

$$\mathtt{vary}(\{p_1, \ldots, p_n\}) = (+p_1 \sqcup -p_1) \; ; \; \cdots \; ; \; (+p_n \sqcup -p_n)$$

- nondeterministically flip one of $p_1, \ldots, p_n$:

$$\begin{aligned} \mathtt{flip1}\big(\{p_1, \ldots, p_n\}\big) = {} & (p_1?; -p_1) \sqcup (\neg p_1?; +p_1) \sqcup \\ & \cdots \sqcup \\ & (p_n?; -p_n) \sqcup (\neg p_n?; +p_n) \end{aligned}$$

- Hamming distance to closest A-state at least m:

$$\mathtt{H}(A, \geq m) = \begin{cases} \top & \text{if } m = 0 \\ \neg\big\langle \mathtt{flip1}^{\leq m-1}\big(\mathbb{P}_A\big)\big\rangle A & \text{if } m \geq 1 \end{cases}$$

Expressing Forbus's operation in DL-PA

Theorem ([H, KR 2014])

Let $\pi^{\text{forbus}}(A)$ *be the* DL-PA *program*

$$\left(\bigcup_{0\leq m\leq \text{card}(\mathbb{P}_A)} \texttt{H}(A, \geq m)?; \texttt{flip1}^m\big(\mathbb{P}_A\big)\right); A?$$

Then

$$B \diamond^{\text{forbus}} A = \texttt{Mod}(\langle(\pi^{\text{forbus}}(A))^{-1}\rangle B)$$

- program length cubic in length of A

Expressing Dalal's operation in DL-PA

...

(cf. [Herzig, KR 2014])

Other operations

- other update/revision operations can be captured as well
  - Winslett's standard semantics WSS [H., KR 2014]
  - Winslett's possible models approach PMA [H., KR 2014]
    - requires copying of variables

Outline

Argumentation framework modification

$$(\mathcal{A},\mathcal{R}) \overset{\text{modif}}{\Longrightarrow} (\mathcal{A}',\mathcal{R}')$$

- a lot of work recently:
  - [Cayrol et al., JAIR 2010; Bisquert et al., SUM 2012, 2013] [Bisquert, Phd 2014]
  - [Baumann, ECAI 2012; Baumann & Brewka, IJCAI 2015]
  - [Booth et al., TAFA 2013]
  - [Coste-Marquis et al., KR 2014; IJCAI 2015; Mailly, Phd 2015]
  - [Diller et al., IJCAI 2015]
  - [Niskanen et al., AAAI 2016; IJCAI 2016]
- minimal change involved ⇒ use AGM belief revision
  - . . . or KM belief update
    (typically: revise a single model only ⇒ revision=update)

Argumentation framework modification

$$(\mathcal{A}, \mathcal{R}) \overset{\text{modif}}{\Longrightarrow} (\mathcal{A}', \mathcal{R}')$$

1. add/delete elements of $\mathcal{R}$
2. add/delete elements of $\mathcal{A}$
3. enforce some goal property G
   - enforce status of some arguments ('in' or 'out')
     - skeptical version: $\mathcal{A}^+$ subset of *every* extension of $(\mathcal{A}, \mathcal{R}')$ $\mathcal{A}^-$ disjoint from every extension of $(\mathcal{A}, \mathcal{R}')$
     - credulous version: ...
   - enforce an extension E
     - non-strict version: E subset of *some* extension of $(\mathcal{A}, \mathcal{R}')$

Two simple modifications in DL-PA

- modify the attack relation $\mathcal{R}$
  - easy: by atomic assignments $+\text{Att}_{a,b}$ and $-\text{Att}_{a,b}$
- modify the set of arguments $\mathcal{A}$
  - not all possible arguments currently considered
  - new propositional variables Cons_a = "a is currently considered"
  - add/remove an argument = perform assignment on Cons_a
  - see [Doutre, H & Perrussel, KR 2014]

Enforcement: example

$$a \rightleftarrows b$$

- has two stable extensions: $E_a = \{a\}$ and $E_b = \{b\}$
- modify such that no stable extension contains a
  - minimal modification of attack relation such that a is in none of its extensions
  - several frameworks may result ($\neq$ standard revision/update)
  - several definitions of minimality; here: Forbus update

Enforcement: definition

- attack relation of a valuation s:

$$\mathcal{R}(s) = \{(a,b) \ : \ \mathsf{Att}_{a,b} \in s\}$$

- skeptical enforcement with Forbus update:

$$s \diamond^{\sigma}_{\mathtt{skep}} G = \Big\{s' \ : \ \text{every } \sigma\text{-extension of } \mathcal{R}(s') \text{ satisfies } G \text{ and there is no } s'' \text{ such that } \mathtt{h}(s \cap \mathtt{ATT}, s'' \cap \mathtt{ATT}) < \mathtt{h}(s \cap \mathtt{ATT}, s' \cap \mathtt{ATT}) \text{ and every } \sigma\text{-extension of } \mathcal{R}(s'') \text{ satisfies } G \Big\}$$

$$(\mathcal{A}, \mathcal{R}) \diamond^{\sigma}_{\mathtt{skep}} G = \bigcup_{s \in \mathtt{Mod}(\mathsf{Fml}(\mathcal{R}))} s \diamond^{\sigma}_{\mathtt{skep}} G$$

- credulous enforcement with Forbus update:

$$s \diamond^{\sigma}_{\mathtt{cred}} G = \Big\{s' \ : \ \text{some } \sigma\text{-extension of } \mathcal{R}(s') \text{ satisfies } G \text{ and} \ldots \Big\}$$

$$(\mathcal{A}, \mathcal{R}) \diamond^{\sigma}_{\mathtt{cred}} G = \ \ldots$$

Enforcement in DL-PA

- Hamming distance wrt attack variables only:

$$\mathtt{H}\Big(\langle \mathtt{makeExt}^{\sigma} \rangle G, \mathtt{ATT}, \geq m\Big) = \ldots$$

- assignment programs minimally modify attack variables such that some/all extensions satisfy the goal:

$$\mathtt{credEnf}^{\sigma}(G) = \Big(\bigcup_{m \leq \mathsf{card}(\mathtt{ATT})} \mathtt{H}\big(\langle \mathtt{makeExt}^{\sigma} \rangle G, \mathtt{ATT}, \geq m\big)? \;;\; \big(\mathtt{flip1}(\mathtt{ATT})\big)^{m} \Big) \;; \langle \mathtt{makeExt}^{\sigma} \rangle G?$$

$$\mathtt{skepEnf}^{\sigma}(G) = \Big(\bigcup_{m \leq \mathsf{card}(\mathtt{ATT})} \mathtt{H}\big([\mathtt{makeExt}^{\sigma}] G, \mathtt{ATT}, \geq m\big)? \;;\; \big(\mathtt{flip1}(\mathtt{ATT})\big)^{m} \Big) \;; [\mathtt{makeExt}^{\sigma}] G?$$

- update by a counterfactual!

Enforcement in DL-PA: results

Theorem

DL-PA *encoding is correct:*

$$(\mathcal{A}, \mathcal{R}) \diamond^{\sigma}_{\mathtt{skep}} G = \mathtt{Mod}(\langle(\mathtt{credEnf}^{\sigma}(G))^{-1}\rangle \mathsf{Fml}(\mathcal{R}))$$
$$(\mathcal{A}, \mathcal{R}) \diamond^{\sigma}_{\mathtt{cred}} G = \mathtt{Mod}(\langle(\mathtt{skepEnf}^{\sigma}(G))^{-1}\rangle \mathsf{Fml}(\mathcal{R}))$$

Theorem

satisfies success postulate:

$$\models [\mathtt{credEnf}^{\sigma}(G)]\, \langle \mathtt{makeExt}^{\sigma}\rangle\, G$$
$$\models [\mathtt{skepEnf}^{\sigma}(G)]\, [\mathtt{makeExt}^{\sigma}]\, G$$

Theorem

satisfies vacuity postulate:

$$\models (\mathsf{Fml}(\mathcal{R}) \land \langle \mathtt{makeExt}^{\sigma}\rangle G \land C) \to [\mathtt{credEnf}^{\sigma}(G)]\, C$$
$$\models (\mathsf{Fml}(\mathcal{R}) \land [\mathtt{makeExt}^{\sigma}] G \land C) \to [\mathtt{skepEnf}^{\sigma}(G)]\, C$$

Extension enforcement in DL-PA: pushing the envelope

- replace $\diamond^{\text{forbus}}$ by other concrete update semantics (e.g. PMA)
- replace $\diamond^{\text{forbus}}$ by concrete revision operations
  - Dalal's Hamming distance-based revision
- replace $\diamond^{\text{forbus}}$ by *prioritised* version [Mailly et al., JELIA 2014]
  - up to now: "minimise ATT only" politics

$$(\mathcal{A}, \mathtt{ATT}) \diamond^{\text{forbus}}_{\mathtt{ATT}} \big(\langle \mathtt{makeExt}^{\sigma} \rangle G\big)$$

  - replace by "first minimise IN, then ATT":
    1. minimally change IN variables to make $\langle \mathtt{vary}(\mathtt{ATT}) \rangle G$ true
    2. minimally change the ATT variables in order to make Goal true
  - in DL-PA: two Forbus updates in sequence:

$$\Big((\mathcal{A}, \mathtt{ATT}) \diamond^{\text{forbus}}_{IN} \big(\underline{\langle \mathtt{vary}(\mathtt{ATT}) \rangle G}\big)\Big) \diamond^{\text{forbus}}_{ATT} G$$

  - multiple extensions: rather take Dalal revision?

Outline

Conclusion

- dynamic logic account of Dung argumentation frameworks
  - build extensions = execute DL-PA program
  - program can be more or less deterministic
  - program can be verified in DL-PA
- dynamic logic account of Dung argumentation framework modification
  - enforcement = update by a counterfactual
  - enforce on all extensions: use $[\pi^\sigma]$
  - enforce on some extension: use $\langle\pi^\sigma\rangle$
- structured argumentation?