A simple logic for reasoning about actions, update, revision, argumentation, planning, and all that

Andreas Herzig
University of Toulouse, IRIT-CNRS, France

JELIA 2014, Funchal, September 26, 2014

Putting talks together

Dynamic Logic of Propositional Assignments DL-PA
[H. et al., IJCAI 2011, Balbiani, H.&Troquard, LICS 2013]

- a sort of Hilbert program for knowledge representation: capture metalinguistic definitions in a logical language
  - update and revision operations [H., KR 2014]
  - merging operations [H. et al., FOIKS 2014]
  - abstract argumentation frameworks and their modification [Doutre, H.&Perrussel, KR 2014]
  - planning tasks and their modification [H. et al., ECAI 2014]
  - active integrity constraints ⇒ talk this afternoon

Outline

Assignments and QBF

Which logical language for knowledge representation?

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

$$s \models p \quad \text{if} \quad p \in s$$
$$s \models \neg\varphi \quad \text{if} \quad s \not\models \varphi$$
$$\ldots$$

- quantified boolean formulas (QBF): talk about valuations and their modification

$$s \models \exists p.\varphi \quad \text{if} \quad s\cup\{p\} \models \varphi \quad \text{or} \quad s\setminus\{p\} \models \varphi$$
$$s \models \forall p.\varphi \quad \text{if} \quad s\cup\{p\} \models \varphi \quad \text{and} \quad s\setminus\{p\} \models \varphi$$

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

$$s \models \langle p{\leftarrow}\top\rangle\varphi \quad \text{if} \quad s\cup\{p\} \models \varphi$$
$$s \models \langle p{\leftarrow}\bot\rangle\varphi \quad \text{if} \quad s\setminus\{p\} \models \varphi$$

$\Rightarrow$ assignments of propositional variables to truth values

Assignments and QBF are equi-expressive

- express assignments in QBF:

$$\begin{aligned} \langle p{\leftarrow}\top\rangle\varphi &= \exists p.(p \wedge \varphi) \\ \langle p{\leftarrow}\bot\rangle\varphi &= \exists p.(\neg p \wedge \varphi) \end{aligned}$$

- express propositional quantifiers in DL-PA:

$$\begin{aligned} \exists p.\varphi &= \langle p{\leftarrow}\top\rangle\varphi \vee \langle p{\leftarrow}\bot\rangle\varphi \\ \forall p.\varphi &= \langle p{\leftarrow}\top\rangle\varphi \wedge \langle p{\leftarrow}\bot\rangle\varphi \end{aligned}$$

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

Assignment programs as relations on valuations

- atomic assignment programs

$$s \xrightarrow{p\leftarrow\top} s\cup\{p\}$$

$$s \xrightarrow{p\leftarrow\bot} s\setminus\{p\}$$

- 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

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

- 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 languages

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

Language of DL-PA

- existential and universal modal operators:

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

  - therefore more compactly:

$$\exists p.\varphi = \langle p{\leftarrow}\top \cup p{\leftarrow}\bot\rangle\varphi$$
$$\forall p.\varphi = [p{\leftarrow}\top \cup p{\leftarrow}\bot]\varphi$$

- language of DL-PA: programs π and formulas φ

$$\begin{array}{lcl} \varphi & ::= & p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\pi\rangle\varphi \mid [\pi]\varphi \\ \pi & ::= & p{\leftarrow}\top \mid p{\leftarrow}\bot \mid \varphi? \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \mid \pi^{-1} \end{array}$$

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

Semantics of DL-PA: (1) formulas

- interpretation of a formula φ = set of valuations $\|\varphi\| \subseteq 2^{\mathbb{P}}$

$$\|p\| = \{s \ : \ p \in s\}$$
$$\|\top\| = 2^{\mathbb{P}}$$
$$\|\bot\| = \emptyset$$
$$\|\neg\varphi\| = \ \ldots$$
$$\|\varphi \vee \psi\| = \ \ldots$$
$$\|\langle\pi\rangle\varphi\| = \left\{s \ : \ \text{there is } s' \text{ such that } s \xrightarrow{\pi} s' \ \ \& \ \ s' \in \|\varphi\|\right\}$$
$$\|[\pi]\varphi\| = \left\{s \ : \ \text{for every } s' : s \xrightarrow{\pi} s' \Longrightarrow s' \in \|\varphi\|\right\}$$

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

Semantics of DL-PA: (1) formulas

- interpretation of a formula φ = set of valuations $\|\varphi\| \subseteq 2^{\mathbb{P}}$

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

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

Semantics of DL-PA: (2) programs

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

$$
\begin{aligned}
\|p{\leftarrow}\top\| &= \{(s, s') \ : \ s' = s \cup \{p\}\} \\
\|p{\leftarrow}\bot\| &= \{(s, s') \ : \ s' = s \setminus \{p\}\} \\
\|\varphi?\| &= \{(s, s) \ : \ s \in \|\varphi\|\} \\
\|\pi; \pi'\| &= \|\pi\| \circ \|\pi'\| \\
\|\pi \cup \pi'\| &= \|\pi\| \cup \|\pi'\| \\
\|\pi^*\| &= (\|\pi\|)^* = \bigcup_{k \in \mathbb{N}_0} (\|\pi\|)^k \\
\|\pi^{-1}\| &= (\|\pi\|)^{-1}
\end{aligned}
$$

DL-PA: eliminating the dynamic operators

1. eliminate the program operators
   1. eliminate the Kleene star:
   $$\langle \pi^* \rangle \varphi \leftrightarrow \langle \pi^{2^{\leq \mathrm{card}(\mathbb{P}_\pi)}} \rangle \varphi$$
   2. eliminate converse operators:
   $$(\pi_1;\pi_2)^{-1} \equiv \pi_2^{-1};\pi_1^{-1} \qquad p{\leftarrow}\top^{-1} \equiv p?;(\mathsf{skip} \cup p{\leftarrow}\bot)$$
   $$\ldots \qquad p{\leftarrow}\bot^{-1} \equiv \ldots$$
   3. eliminate all other program operators:
   $$\langle \pi_1 \cup \pi_2 \rangle \varphi \leftrightarrow \langle \pi_1 \rangle \vee \langle \pi_2 \rangle \varphi \qquad \langle \psi? \rangle \varphi \leftrightarrow \psi \wedge \varphi \qquad \ldots$$
2. eliminate atomic programs $\langle p{\leftarrow}\top \rangle$ and $\langle p{\leftarrow}\bot \rangle$:
   - distribute over $\wedge$, $\vee$, $\neg$
   - can be eliminated when facing atomic formulas:

$$\langle p{\leftarrow}\top \rangle q \leftrightarrow \begin{cases} \top & \text{if } q = p \\ q & \text{otherwise} \end{cases} \qquad \langle p{\leftarrow}\bot \rangle q \leftrightarrow \begin{cases} \bot & \text{if } q = p \\ q & \text{otherwise} \end{cases}$$

Proposition ('regression')

For every DL-PA *formula there is an equivalent boolean formula*

(that might be exponentially longer)

DL-PA: eliminating the dynamic operators

Example

$$
\begin{aligned}
\langle p{\leftarrow}\bot^{-1}\rangle(p{\wedge}q) &\leftrightarrow \langle \neg p?\,;\,(\mathsf{skip}\cup p{\leftarrow}\top)\rangle(p{\wedge}q)\\
&\leftrightarrow \langle \neg p?\rangle\langle(\mathsf{skip}\cup p{\leftarrow}\top)\rangle(p{\wedge}q)\\
&\leftrightarrow \neg p\wedge\langle(\mathsf{skip}\ \cup\ p{\leftarrow}\top)\rangle(p{\wedge}q)\\
&\leftrightarrow \neg p\wedge\Big(\langle\mathsf{skip}\rangle(p\wedge q)\vee\langle p{\leftarrow}\top\rangle(p{\wedge}q)\Big)\\
&\leftrightarrow \neg p\wedge\Big(\langle\mathsf{skip}\rangle(p\wedge q)\vee(\langle p{\leftarrow}\top\rangle p\wedge\langle p{\leftarrow}\top\rangle q)\Big)\\
&\leftrightarrow \neg p\wedge\Big((p\wedge q)\vee(\top\wedge q)\Big)\\
&\leftrightarrow \neg p\wedge\Big((p\wedge q)\vee q\Big)\\
&\leftrightarrow \neg p\wedge q
\end{aligned}
$$

Properties of DL-PA

- compares favourably to PDL:
  - PSPACE complete both for model checking and satisfiability checking [Balbiani et al., ongoing]
    - flawed proof in [Balbiani et al., LICS 2013]
    - PDL: SAT is EXPTIME complete
  - compactness
    - fails for PDL
- claim: in applications with finite domains, DL-PA does the same job as PDL
- rest of talk: how to capture some popular KR formalisms

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, KM (built from background ordering or distance)
- semantical
  1. state = subset of $\mathbb{P}$
  2. interpretation of formula = set of states
  3. result of update/revision = set of states

     $B \circ A$ subset of $2^{\mathbb{P}}$
- syntactical construction of the new base?
  - disjunction of the formulas describing the models of $B \circ A$
  - via forgetting: only for WSS [H.&Rifi, AIJ 1999]
  - via prime implicants [Bittencourt et al., AMAI 2010]

Forbus's update operation [Forbus, IJCAI 1989]

- Hamming distance between states

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

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

Example

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

$p \oplus q \diamond^{\text{forbus}} p = \|p\|$

Dalal's revision operation [Dalal, AAAI 1988]

- revise B by A = "go to the A-states that are closest w.r.t. Hamming distance to the B-states"

$$B *^{\text{dalal}} A = \Big\{ s_A \in \|A\| \ : \ \text{there is } s_B \in \|B\| \text{ s.t. there are no } s'_A, s'_B \text{ with } \mathrm{h}(s'_A, s'_B) < \mathrm{h}(s_A, s_B) \Big\}$$

(see [H. et al., ECAI 2014] for the case where B is unsatisfiable)

- update vs. revision:
  - $B *^{\text{dalal}} A = B \diamond^{\text{forbus}} A$ if B is complete
  - $B *^{\text{dalal}} A = \|B \wedge A\|$ if $\|B \wedge A\| \neq \emptyset$

Example

$\neg p \wedge \neg q *^{\text{dalal}} p \vee q = \|p \oplus q\|$
$p \oplus q *^{\text{dalal}} p = \|p \wedge \neg q\|$

The embeddings in a nutshell

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

$$\pi^{\mathrm{wss}}_{\neg p \vee \neg q} = p{\leftarrow}\bot \cup q{\leftarrow}\bot \cup (p{\leftarrow}\bot; q{\leftarrow}\bot)$$
$$\pi^{\mathrm{pma}}_{\neg p \vee \neg q} = \ldots$$

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

$$B \circ^{op} A \;=\; \left\| \left\langle (\pi^{op}_A)^{-1} \right\rangle B \right\|$$

- details in the next slides

Some useful programs and formulas

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

$$\mathtt{vary}\big(\{p_1, \ldots, p_n\}\big) = (p_1 \leftarrow \top \cup p_1 \leftarrow \bot) \; ; \; \cdots \; ; \; (p_n \leftarrow \top \cup p_n \leftarrow \bot)$$

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

$$\mathtt{flip1}\big(\{p_1, \ldots, p_n\}\big) = p_1 \leftarrow \neg p_1 \cup \cdots \cup p_n \leftarrow \neg p_n$$

- 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*

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

Then $B \diamond^{\text{forbus}} A = \|\langle(\pi^{\text{forbus}}(A))^{-1}\rangle B\|$.

- program length cubic in length of A

Expressing Dalal's operation in DL-PA

Theorem ([H, KR 2014])

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

$$\mathtt{vary}\big(\mathbb{P}_B\big)\ ;\ B?\ ;\ \left(\bigcup_{0\leq m\leq \mathsf{card}(\mathbb{P}_A)} \Big([\mathtt{vary}\big(\mathbb{P}_B\big)\ ;\ B?]\mathtt{H}(A, \geq m)\Big)?\ ;\ \mathtt{flip1}^m\big(\mathbb{P}_A\big)\right);A?$$

Then for satisfiable B*:* $B *^{\text{dalal}} A = \|\langle(\pi^{\text{dalal}}(A, B))^{-1}\rangle\top\|$.

- program length cubic in length of A + length of B

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
  - Satoh's revision (t.b.d.)

Outline

Classical planning

- classical planning task:

$$\begin{aligned} \langle \mathbb{P}, & \quad \text{finite set of propositional variables} \\ s_0, & \quad \text{initial state} \\ S_g, & \quad \text{set of goal states} \\ \mathtt{A}\rangle & \quad \text{finite set of STRIPS actions} \end{aligned}$$

- interpretation of an action $a \in \mathtt{A}$ = relation on the set of states

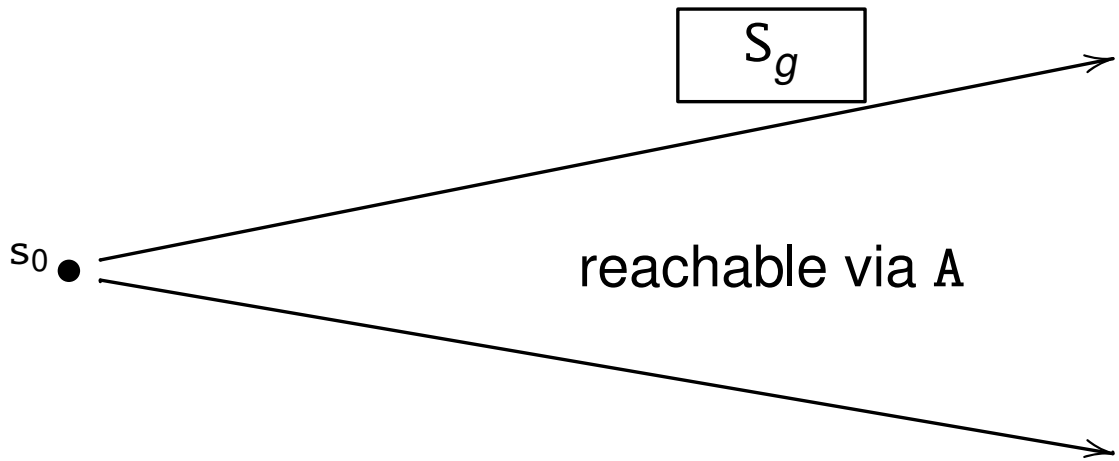
$$\|a\| = \left\{(s, s') \ : \ s \in \|\mathrm{pre}_a\| \text{ and } s' = (s \setminus \mathrm{del}_a) \cup \mathrm{add}_a\right\}$$

(deterministic: each $\|a\|$ is a function)

- s is *reachable* from s_0 via $\mathtt{A}$ iff ...
- planning task is *solvable* iff some state in S_g is reachable from s_0 via $\mathtt{A}$

Planning task modification

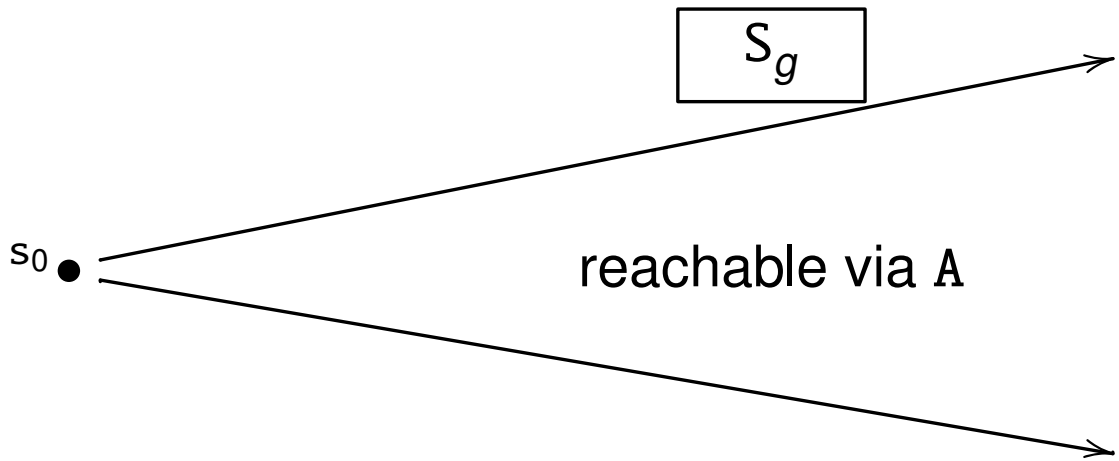
- suppose $(\mathbb{P}, \mathtt{A}, s_0, S_g)$ has no solution



- modify task
  [Smith, ICAPS 2004; Göbelbecker et al., ICAPS 2010]:
  1. increase or decrease the set of objects of the domain
  2. augment the set of actions $\mathtt{A}$
  3. change the initial state s_0 (‘find good excuses’)
  4. change the goal description S_g (‘over-subscription planning’)
- here: 2, 3 and 4

Planning task modification

- suppose $(\mathbb{P}, \mathtt{A}, s_0, S_g)$ has no solution



- modify task
  [Smith, ICAPS 2004; Göbelbecker et al., ICAPS 2010]:
  1. increase or decrease the set of objects of the domain
  2. augment the set of actions A
  3. change the initial state s_0 ('find good excuses')
  4. change the goal description S_g ('over-subscription planning')
- here: 2, 3 and 4

Changing the initial state

- candidate initial states:

  $S_0' = \{s_0' \ : \ \text{there is a goal state that is reachable from } s_0' \text{ via } \mathtt{A}\}$

- candidate initial states closest to s_0:

$$s_0 \Diamond^{\text{forbus}} \mathtt{Fml}(S_0')$$

- alternative: $s_0 \Diamond^{\text{pma}} \mathtt{Fml}(S_0')$ [Göbelbecker et al., ICAPS 2010]
- for both:
  - "update s_0 such that S_g becomes reachable"
  - problem: counterfactual statement $\Rightarrow$ non-boolean

Changing the goal

- candidate goal states:

$$S'_g = \{s'_g \ : \ s'_g \text{ is reachable from initial state via } \mathtt{A}\}$$

- candidate goal states closest to S_g:

$$S_g *^{\text{dalal}} \mathtt{Fml}(S'_g)$$

  - "*revise* S_g such that goal becomes reachable from s_0"

    N.B.: update would be too permissive
  - problem: counterfactual statement $\Rightarrow$ non-boolean

Augmenting the set of actions

- given: planning task $(\mathbb{P}, \mathtt{A}, \mathtt{s}_0, \mathtt{S}_g)$
- set of background actions $\mathtt{A}_0$
- only $\mathtt{A}$ is initially *usable*

$$\mathtt{s}_0 \models \left(\bigwedge_{a \in \mathtt{A}} \mathtt{u}_a \right) \wedge \left(\bigwedge_{a \in \mathtt{A}_0 \setminus \mathtt{A}} \neg \mathtt{u}_a \right)$$

  - add $\mathtt{u}_a$ to the precondition of all actions
  - change the $\mathtt{u}_a$ minimally such that $\mathtt{S}_g$ gets reachable

Classical planning tasks in DL-PA

- a with add list $\{p_1, \ldots, p_m\}$ and delete list $\{q_1, \ldots, q_n\}$:

$$\|a\| = \|\mathrm{pre}_a?; p_1 \leftarrow \top; \cdots; p_m \leftarrow \top; q_1 \leftarrow \bot; \cdots; q_n \leftarrow \bot\|$$

$\Rightarrow$ view every a_i in $\mathtt{A} = \{a_1, \ldots, a_n\}$ as a DL-PA program

- define DL-PA program $\mathtt{iterate}_{\mathtt{A}} = (a_1 \cup \cdots \cup a_n)^*$

$$(\mathbb{P}, \mathtt{A}, \mathtt{s}_0, \mathtt{S}_g) \text{ solvable} \quad \text{iff} \quad \mathtt{Fml}(\mathtt{s}_0) \rightarrow \langle \mathtt{iterate}_{\mathtt{A}} \rangle \mathtt{Fml}(\mathtt{S}_g) \text{ DL-PA valid}$$

Changing the initial state in DL-PA

- set of candidate initial states:

$$S_0' = \left\| \langle \mathtt{iterate}_A \rangle \mathtt{Fml}(S_g) \right\|$$

Theorem

The set of initial states closest to s_0 *from which* S_g *is reachable is*

$$s_0 \diamond^{\text{forbus}} \mathtt{Fml}(S_0') = \left\| \left\langle \left(\pi^{\text{forbus}}(\mathtt{Fml}(S_0')) \right)^{-1} \right\rangle \mathtt{Fml}(s_0) \right\|$$
$$= \left\| \left\langle \left(\pi^{\text{forbus}}(\langle \mathtt{iterate}_A \rangle \mathtt{Fml}(S_g)) \right)^{-1} \right\rangle \mathtt{Fml}(s_0) \right\|$$

Changing the goal in DL-PA

- set of candidate goal states:

$$S'_g = \left\| \langle \mathtt{iterate}_A^{-1} \rangle \mathtt{Fml}(s_0) \right\|$$

Theorem

The set of goal states closest to S_g that are reachable from s_0 is

$$S_g *^{\text{dalal}} \mathtt{Fml}(S'_g) = \left\| \left\langle \left(\pi^{\text{dalal}}(\mathtt{Fml}(S'_g), \mathtt{Fml}(S_g)) \right)^{-1} \right\rangle \top \right\|$$
$$= \left\| \left\langle \left(\pi^{\text{dalal}}(\langle \mathtt{iterate}_A^{-1} \rangle \mathtt{Fml}(s_0), \mathtt{Fml}(S_g)) \right)^{-1} \right\rangle \top \right\|$$

Outline

Argumentation frameworks in propositional logic

- abstract argumentation framework [Dung, 1995]: $\mathcal{AF} = (\mathcal{A}, R)$
  - $\mathcal{A}$ = finite set of abstract arguments
  - $R \subseteq \mathcal{A} \times \mathcal{A}$ = binary relation on $\mathcal{A}$
    - $(a, b) \in R$ = a attacks b
- in a logical language:
  - a set of *attack variables*

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

  - theory of $\mathcal{AF} = (\mathcal{A}, R)$

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

Argumentation frameworks in propositional logic, ctd.

- set of *acceptance variables* $\mathtt{IN}^{\mathcal{A}} = \{\mathsf{In}_a \ : \ a \in \mathcal{A}\}$
  - In_a = "argument a is in the extension"
- stable extensions:

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

- admissible sets:

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

- complete extension:

$$\mathsf{Complete}_{\mathcal{A}} = \ldots$$

Argumentation frameworks in propositional logic, ctd.

- $\text{Semantics}_{\mathcal{A}}$ = propositional formula describing the semantics σ (stable/admissible/complete)
- description of an extension $E \subseteq \mathcal{A}$:

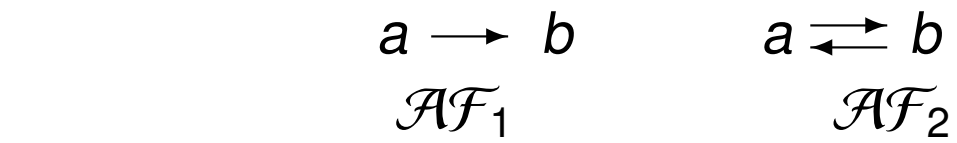
$$\text{Th}_{E,\mathcal{A}} = \bigwedge_{a \in E} \text{In}_a \wedge \bigwedge_{a \mathcal{A} \setminus E} \neg \text{In}_a$$

- compute the extensions of $\mathcal{AF}$ under σ = find models of $\text{Th}_{\mathcal{AF}} \wedge \text{Semantics}_{\mathcal{A}}$

Proposition (Besnard&Doutre, NMR 2004)

1. *E is a stable extension of $\mathcal{AF}$ iff* $\models \left(\text{Th}_{\mathcal{AF}} \wedge \text{Th}_{E,\mathcal{A}}\right) \rightarrow \text{Stable}_{\mathcal{A}}$
2. *E is an admissible set of $\mathcal{AF}$ iff* $\models \left(\text{Th}_{\mathcal{AF}} \wedge \text{Th}_{E,\mathcal{A}}\right) \rightarrow \text{Adm}_{\mathcal{A}}$
3. *E is a complete extension of $\mathcal{AF}$ iff* $\models \left(\text{Th}_{\mathcal{AF}} \wedge \text{Th}_{E,\mathcal{A}}\right) \rightarrow \text{Complete}_{\mathcal{A}}$

Argumentation frameworks: example



- description in logic:

$$\mathsf{Th}_{\mathcal{AF}_1} = \mathsf{Att}_{a,b} \wedge \neg\mathsf{Att}_{b,a} \wedge \neg\mathsf{Att}_{a,a} \wedge \neg\mathsf{Att}_{b,b}$$
$$\mathsf{Th}_{\mathcal{AF}_2} = \mathsf{Att}_{a,b} \wedge \mathsf{Att}_{b,a} \wedge \neg\mathsf{Att}_{a,a} \wedge \neg\mathsf{Att}_{b,b}$$

- $\mathcal{AF}_2$ has two stable extensions: $E_a = \{a\}$ and $E_b = \{b\}$
- in logic: $\mathsf{Th}_{\mathcal{AF}_2} \wedge \mathsf{Stable}_{\mathcal{A}_2}$ has two models

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

Building Extensions in DL-PA

$$\mathtt{makeExt}^{\sigma}_{\mathcal{A}} = \mathtt{vary}(\mathtt{IN}^{\mathcal{A}}); \mathsf{Semantics}_{\mathcal{A}}?$$

Proposition

Let σ be either the stable, complete or admissible semantics.

$$\|\mathtt{makeExt}^{\sigma}_{\mathcal{A}}\| = \left\{(s_1, s_2) \ : \ s_2 \in \mathsf{Semantics}_{\mathcal{A}} \textit{ and } s_1 \cap \mathtt{ATT}^{\mathcal{A}} = s_2 \cap \mathtt{ATT}^{\mathcal{A}}\right\}$$

N.B.: 'generate-and-test' schema; other algorithms from the literature can be implemented, too, e.g. for $\mathcal{A} = \{a_1, \ldots, a_n\}$:

$$\left(\neg \bigvee_{a_i \in \mathcal{A}} \mathsf{Att}_{a_i,a_1}?\right); \mathsf{In}_{a_1} \leftarrow \top; \cdots; \left(\neg \bigvee_{a_i \in \mathcal{A}} \mathsf{Att}_{a_i,a_n}?\right); \mathsf{In}_{a_n} \leftarrow \top; \ldots$$

Argumentation framework modification

given argumentation framework $\mathcal{AF} = (\mathcal{A}, R)$

[Cayrol, Dupin&Lagasquie, JAIR 2010]:

1. modify the set of arguments $\mathcal{A}$ by adding or removing an argument
   - see [Doutre, H.&Perrussel, KR 2014]
2. modify the attack relation R by adding or removing an edge between two arguments
   - easy: by atomic assignments $\mathrm{Att}_{a,b} \leftarrow \top$ and $\mathrm{Att}_{a,b} \leftarrow \bot$
3. example:
   - $\mathcal{AF}_2$: $a \rightleftarrows b$
     two stable extensions: $E_a = \{a\}$ and $E_b = \{b\}$
   - modify $\mathcal{AF}$ such that no stable extension contains a
     $\Rightarrow$ minimal modification of R_2 such that a is in none of its extensions
   - several $\mathcal{AF}'$ may result (in standard revision/update: result is a single belief set)

Argumentation framework update

- skeptical Forbus update:

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

$$\mathcal{AF} \diamond^{\sigma}_{\mathtt{skep}} A = \bigcup_{s \in \|\mathsf{Th}_{\mathcal{AF}}\|} s \diamond^{\sigma}_{\mathtt{skep}} A$$

- credulous Forbus update:

$$s \diamond^{\sigma}_{\mathtt{cred}} A = \Big\{ s' \ : \ \text{some } \sigma\text{-extension of } \mathcal{AF}(s') \text{ satisfies } A \text{ and}$$
$$\text{there is no } s'' \text{ such that } \mathtt{h}(\cdot \mathtt{ATT}^{\mathcal{A}})(s, s'') < \mathtt{h}(\cdot \mathtt{ATT}^{\mathcal{A}})(s, s')$$
$$\text{and some } \sigma\text{-extension of } \mathcal{AF}(s'') \text{ satisfies } A \Big\}$$

$$\mathcal{AF} \diamond^{\sigma}_{\mathtt{cred}} A = \bigcup_{s \in \|\mathsf{Th}_{\mathcal{AF}}\|} s \diamond^{\sigma}_{\mathtt{cred}} A$$

Argumentation framework update in DL-PA

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

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

Theorem

$$\mathcal{AF} \diamond^{\sigma}_{\mathtt{skep}} A = \left\| \left\langle (\mathtt{credEnf}^{\sigma}(A))^{-1} \right\rangle \mathsf{Th}_{\mathcal{AF}} \right\|$$

$$\mathcal{AF} \diamond^{\sigma}_{\mathtt{cred}} A = \left\| \left\langle (\mathtt{skepEnf}^{\sigma}(A))^{-1} \right\rangle \mathsf{Th}_{\mathcal{AF}} \right\|$$

Outline

Conclusion

applicability of DL-PA:

- prominent update and revision operations
  - DL-PA allows syntactical construction of the resulting belief base
- planning and planning task modification
  - builds on embedding of update/revision operations into DL-PA
  - modification of initial state = update by a counterfactual
  - modification of goal = revision by a counterfactual
- Dung argumentation frameworks
  - enforce a property on all/some extensions = update by a counterfactual

Conclusion, ctd.

- more applications:
  - Reiter's basic action theories [van Ditmarsch, H.&de Lima, JLC 2012]
    - $p_1, \ldots, p_n$ = variables in whose successor state axioms a occurs
    - $a = p_1 \leftarrow \gamma_{p_1}(a); \ldots; p_n \leftarrow \gamma_{p_n}(a)$
  - belief merging [H., Pozos&Schwarzentruber, FOIKS 2014]
  - update of ASP programs [Fariñas, H.&Su, LPNMR 2013]
  - coalition logic of propositional control [H et al, IJCAI 2011]
  - active integrity constraints $\Rightarrow$ this afternoon
- ongoing: epistemic extension
  - action preconditions become *epistemic actions*
    - public/semi-public/private/... $\Rightarrow$ DEL s
  - undecidable in general (due to *) [Miller&Moss, 2004]
  - single agent decidable [Bolander et al. 2012]
  - star-free fragment enough to embed Scherl&Levesque's epistemic extension of the SitCalc [van Ditmarsch, H.&de Lima, JLC 2012]
  - other decidable fragments?