Dynamic Logic of Propositional Assignments and its applications to update, revision and planning

Andreas Herzig
University of Toulouse, IRIT-CNRS, France

Workshop on Planning, Logic and Social Intelligence, Copenhagen, April 4, 2014
Aim of talk

- propaganda for Dynamic Logic of Propositional Assignments
  [Herzig et al., IJCAI 2011; Balbiani et al., LICS 2013]
  - simpler than PDL
  - *the* basic logic to for dynamic systems
  - aim: canonical problems between NP and PSPACE
    - nicer than QBF

- specifically:
  - Forbus’s update operation [Forbus 1989]
  - Dalal’s revision operation [Dalal 1988]
  - planning tasks and their modification
    [Smith, ICAPS 2004; Göbelbecker et al., ICAPS 2010]
Outline

1. DL-PA: dynamic logic of propositional assignments
2. Update and revision via DL-PA programs
3. Forbus update
4. Dalal revision
5. Planning tasks and their modification
Dynamic Logic of Propositional Assignments DL-PA with converse

- instantiates good old Propositional Dynamic Logic PDL
  - atomic programs: assignments of propositional variables
    \[ p \leftarrow \top = \text{“make } p \text{ true”} \]
    \[ p \leftarrow \bot = \text{“make } p \text{ false”} \]
  - complex programs and formulas: as usual in PDL

\[ \pi^m \overset{\text{def}}{=} \begin{cases} \text{skip} & \text{if } m = 0 \\ \pi; \pi^{m-1} & \text{otherwise} \end{cases} \]

\[ \pi^{\leq m} \overset{\text{def}}{=} \begin{cases} \text{skip} & \text{if } m = 0 \\ (\text{skip} \cup \pi); \pi^{m-1} & \text{otherwise} \end{cases} \]

\[ p \leftarrow \varphi \overset{\text{def}}{=} (\varphi?; p \leftarrow \top) \cup (\neg \varphi?; p \leftarrow \bot) \]
Dynamic Logic of Propositional Assignments DL-PA with converse

- language, for $p \in \text{PVar}$:
  \[
  \pi ::= p \gets T \mid p \gets \bot \mid \varphi? \mid \pi; \pi \mid \pi \cup \pi \mid \pi^* \mid \pi^{-1}
  \]
  \[
  \varphi ::= p \mid T \mid \bot \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle \pi \rangle \varphi
  \]

- propositional assignments are an old idea:
  - [Tiomkin & Makowsky, TCS, 1985] (global and local)
  - [Wilm, TCS 1991]
  - [van Eijck, Studia Logica 2000] (no Kleene star)
  - [Guelev et al., Information Security 2004] (deterministic programs)

- but the above language was not considered before
  - [Balbiani et al., LICS 2013]
Semantics of DL-PA: formulas

- **valuation** = subset of PVar
  - set of all valuations is $2^{PVar} = \{v, v', \ldots\}$
  - $p \in v$: $p$ true in $v$
  - $p \notin v$: $p$ false in $v$

- interpretation of formula $\varphi$ = set of valuations $||\varphi|| \subseteq 2^{PVar}$

\[
||p|| = \{v : p \in v\}
\]
\[
||\top|| = 2^{PVar}
\]
\[
||\bot|| = \emptyset
\]
\[
||\neg \varphi|| = 2^{PVar} \setminus ||\varphi||
\]
\[
||\varphi \lor \psi|| = ||\varphi|| \cup ||\psi||
\]
\[
||\langle \pi \rangle \varphi|| = \{v : \text{there is } v', (v, v') \in ||\pi|| \text{ and } v' \in ||\varphi||\}\]
Semantics of DL-PA: programs

interpretation of program \( \pi = \text{relation } ||\pi|| \subseteq 2^{PVar} \times 2^{PVar} \)

\[
\begin{align*}
||p & \leftarrow \top|| = \{ (v, v') : v' = v \cup \{ p \} \} \\
||p & \leftarrow \bot|| = \{ (v, v') : v' = v \setminus \{ p \} \} \\
||\varphi ?|| &= \{ (v, v) : v \in ||\varphi|| \} \\
||\pi ; \pi'|| &= ||\pi|| \circ ||\pi'|| \\
||\pi \cup \pi'|| &= ||\pi|| \cup ||\pi'|| \\
||\pi^*|| &= (||\pi||)^* = \bigcup_{k \in \mathbb{N}_0} (||\pi||)^k \\
||\pi^{-1}|| &= (||\pi||)^{-1}
\end{align*}
\]
DL-PA: eliminating the program operators

- eliminate converse operator:
  \[ \| p \leftarrow \top^{-1} \| = \| p ?; (\text{skip} \cup p \leftarrow \bot) \| \]
  \[ \| p \leftarrow \bot^{-1} \| = \ldots \]

- eliminate Kleene star:
  \[ \| \pi^* \| = \| \pi^{\leq \text{card}(\text{PVar}(\pi))} \| \]

- eliminate the other program operators: as in star-free PDL

Proposition
([Herzig et al., IJCAI 2011; Balbiani et al., LICS 2013])

*For every DL-PA formula \( \varphi \) there is an equivalent formula \( \varphi' \) such that no program operators occur in \( \varphi' \).*
eliminate atomic programs:
- atomic programs $\langle p \leftarrow \top \rangle$ and $\langle p \leftarrow \bot \rangle$ distribute over $\land$, $\lor$, $\neg$
- can be eliminated when facing atomic formulas:

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

Proposition
([Herzig et al., IJCAI 2011; Balbiani et al., LICS 2013])

*For every DL-PA formula there is an equivalent boolean formula.*
DL-PA: eliminating the dynamic operators

Example

\[
\langle p \leftarrow \bot^{-1} \rangle (p \land q) \iff \langle \neg p \ ; \ (\text{skip} \cup p \leftarrow \top) \rangle (p \land q)
\]
\[
\iff \langle \neg p \rangle \langle (\text{skip} \cup p \leftarrow \top) \rangle (p \land q)
\]
\[
\iff \neg p \land \langle (\text{skip} \cup p \leftarrow \top) \rangle (p \land q)
\]
\[
\iff \neg p \land (\langle \text{skip} \rangle (p \land q) \lor \langle p \leftarrow \top \rangle (p \land q))
\]
\[
\iff \neg p \land (\langle \text{skip} \rangle (p \land q) \lor (\langle p \leftarrow \top \rangle p \land \langle p \leftarrow \top \rangle q))
\]
\[
\iff \neg p \land ((p \land q) \lor (\top \land q))
\]
\[
\iff \neg p \land ((p \land q) \lor q)
\]
\[
\iff \neg p \land q
\]
Properties and applications of DL-PA

- **properties**
  - no nondeterministic composition: NP complete
  - star-free: PSPACE complete [Herzig et al., IJCAI 2011]
  - full language: PSPACE complete [Balbiani et al., ongoing]
    - flawed EXPTIME hardness proof in [Balbiani et al., LICS 2013]

- **applications**
  - ...
  - ...
  - ...
  - here:
    - various update and revision operations [Herzig, KR 2014]
    - plan existence, planning task modification
Outline

1. DL-PA: dynamic logic of propositional assignments
2. Update and revision via DL-PA programs
3. Forbus update
4. Dalal revision
5. Planning tasks and their modification
Update and revision operations

- $B \circ A = \text{modification of belief base } B \text{ accommodating input } A$
  - $B, A$ boolean formulas
  - object language operators (vs. metalanguage operations)

- two kinds of change:
  - update = “world changes” [Katsuno&Mendelzon 1992]
  - revision = “knowledge about world changes” [Alchourrón et al., JSL 1985]

- two different accounts:
  1. parametrised operations
     - built from arbitrary orderings or distances
     - characterised by postulates
  2. concrete operations
     - based on particular distances between valuations
     - mainly studied from semantic perspective
     - $\Rightarrow B \circ A = \text{a set of valuations of classical propositional logic}$
     - $\Rightarrow \text{syntactical representation?}$
     - $\Rightarrow \text{“disjunction of formulas describing the models of } B \circ A”$
Embedding into DL-PA

- idea:
  - update by atomic formula ≡ atomic assignment
    - update by $p$ ≡ $p \leftarrow \top$
    - update by $\neg p$ ≡ $p \leftarrow \bot$
  - update by complex formula $A \approx$ complex assignment $\pi(A)$
  - depends on belief change operation: $\pi^{\text{wss}}(A) \neq \pi^{\text{forbus}}(A)$, etc.
    - $\pi^{\text{wss}}(\neg p \lor \neg q) = p \leftarrow \bot \cup q \leftarrow \bot \cup (p \leftarrow \bot; q \leftarrow \bot)$
    - $\pi^{\text{forbus}}(\neg p \lor \neg q) = \ldots$

- aim: find polynomial embeddings into DL-PA
  1. syntactical construction of the new base via reduction to boolean formulas
  2. general framework for belief change

  to be proved for each change operation $\circ^\omega$:

  $B \circ^\omega A = \langle (\pi^\omega(A))^{-1} \rangle B$

- here: based on Hamming distance [Forbus 1989; Dalal 1988]
- other operations: see [Herzig, KR 2014]
Some useful DL-PA programs

- nondeterministically assign truth values to \( p_1, \ldots, p_n \):
  \[
  \text{vary}(\{p_1, \ldots, p_n\}) = (p_1 \leftarrow T \cup p_1 \leftarrow \bot) ; \cdots ; (p_n \leftarrow T \cup p_n \leftarrow \bot)
  \]

- nondeterministically flip one of \( p_1, \ldots, p_n \):
  \[
  \text{flip1}(\{p_1, \ldots, p_n\}) = p_1 \leftarrow \neg p_1 \cup \cdots \cup p_n \leftarrow \neg p_n
  \]
Outline

1. DL-PA: dynamic logic of propositional assignments
2. Update and revision via DL-PA programs
3. Forbus update
4. Dalal revision
5. Planning tasks and their modification
Forbus’s update operation [Forbus89]

- Hamming distance between valuations $v$ and $v'$
  \[ h(\{p, q\}, \{q, r, s\}) = \text{card}(\{p, r, s\}) = 3 \]

- Valuation-wise update:
  1. for each model $v \in ||B||$:
     - select $A$-valuations closest w.r.t. the Hamming distance
  2. collect the resulting valuations

\[
  v \diamond_{\text{forbus}} A = \{v' : v' \in ||A|| \text{ and there is no } v'' \text{ such that } h(v, v'') < h(v, v') \}
\]

**Example**

\[
\begin{align*}
  \neg p \land \neg q \diamond_{\text{forbus}} p \lor q &= ||p \oplus q|| \\
  \neg p \land \neg q \land \neg r \diamond_{\text{forbus}} (p \land q) \lor r &= ||\neg p \land \neg q \land r||
\end{align*}
\]
Expressing Forbus’s operation in DL-PA

**Proposition**

Let $A$, $B$ be propositional formulas. Let

$$H(A, \geq m) = \begin{cases} \top & \text{if } m = 0 \\ \neg \langle \text{flip}^{m-1}(\text{PVar}(A)) \rangle A & \text{if } m \geq 1 \end{cases}$$

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

$$\bigcup_{0 \leq m \leq \text{card(PVar}(A))} H(A, \geq m); \text{flip}^m(\text{PVar}(A)); A?$$

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

- program length cubic in length of $A$
Outline

1. DL-PA: dynamic logic of propositional assignments
2. Update and revision via DL-PA programs
3. Forbus update
4. Dalal revision
5. Planning tasks and their modification
Dalal’s revision operation

- based on minimisation of the Hamming distance between valuations
- revise $B$ by $A = \text{“go to the } A\text{-valuations that are closest w.r.t. Hamming distance to the } B\text{-valuations”}\n  - if $B$ is satisfiable:
    \[
    B \star^{\text{dalal}} A = \{v_A \in ||A|| : \text{there is } v_B \in ||B|| \text{ s.t. there are no } v'_A, v'_B \text{ with } h(v'_A, v'_B) < h(v_A, v_B)\}\n    
  - if $B$ is unsatisfiable:
    \[
    B \star^{\text{dalal}} A = ||A||
    \]
- revision operation: satisfies AGM preservation postulate
  \[
  \text{if } ||B \land A|| \neq \emptyset \text{ then } B \star A = ||B \land A||
  \]

**Example**

\[
\neg p \lor \neg q \star^{\text{dalal}} p = ||p \land \neg q||
\]
- different from update: $\neg p \lor \neg q \diamond^{\text{forbus}} p = ||p||$
Proposition

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

\[
\text{vary}(\text{PVar}(B)) \ ; \ B ? \ ; \\
\bigcup_{0 \leq m \leq \text{card}(\text{PVar}(A))} \left( [\text{vary}(\text{PVar}(B)) \ ; \ B?] \mathcal{H}(A, \geq m)? \ ; \text{flip}^m(\text{PVar}(A)) \right) \ ; \ A ?
\]

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

- program depends on the input $A$ and on the base $B$
- program length cubic in length of $A + \text{length of } B$
Outline

1. DL-PA: dynamic logic of propositional assignments
2. Update and revision via DL-PA programs
3. Forbus update
4. Dalal revision
5. Planning tasks and their modification
Classical planning tasks

- action (planning operator): $a = (\text{pre}_a, \text{add}_a, \text{del}_a)$ with:
  - $\text{pre}_a \in \text{Fml}_{\text{bool}}$ (precondition)
  - $\text{add}_a, \text{del}_a \subseteq \text{PVar}$ (finite add list + delete list)
- semantics:

\[
\|a\| = \{(v, v') : v \in \|\text{pre}_a\| \text{ and } v' = (v \setminus \text{del}_a) \cup \text{add}_a\}
\]

\[
\|a\| = \|\text{pre}_a\?; q_1 \leftarrow \bot; \cdots; q_n \leftarrow \bot; p_1 \leftarrow T; \cdots; p_m \leftarrow T\|
\]

for $\text{add}_a = \{p_1, \ldots, p_m\}$ and $\text{del}_a = \{q_1, \ldots, q_n\}$
Classical planning tasks

valuation \( v \) is *reachable* from valuation \( s_0 \) via a set of actions \( A \) iff there is a sequence of actions \((a_1, \ldots, a_n)\) and a sequence of valuations \((v_0, \ldots, v_n)\) such that

- \( v_0 = s_0 \),
- \( v_n = v \), and
- \((v_{k-1}, v_k) \in \|a_k\|\) for every \( k \) such that \( 1 \leq k \leq n \).

**classical planning task:** \((\text{PVar}, A, s_0, S_g)\) where:

- \( A \) is a set of actions
- \( s_0 \subseteq \text{PVar} \) (initial state)
- \( S_g \subseteq 2^{\text{PVar}} \) (goal)

**solvable** iff there is \( v \in S_g \) reachable from \( s_0 \) via \( A \)

**hypothesis:** \( \text{PVar} \) and \( A = \{a_1, \ldots, a_n\} \) finite

- \( \text{iterate}_A = (a_1 \cup \cdots \cup a_n)^* \)

\[(\text{PVar}, A, s_0, S_g) \text{ solvable } \iff \text{Fml}(s_0) \rightarrow \langle \text{iterate}_A \rangle \text{Fml}(S_g) \text{ DL-PA valid}\]
suppose \((P\text{Var}, A, s_0, S_g)\) has no solution

task modification [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 \(s_0\)
4. change the goal description \(S_g\) (‘over-subscription planning’)

here: 3 and 4
Changing the initial state

- set of candidate initial states:

\[ S'_0 = \{ s'_0 : \text{there is } s_g \in S_g \text{ such that } s_g \text{ is reachable from } s'_0 \text{ via } A \} \]

\[ = \| \langle \text{iterate}_A \rangle \text{Fml}(S_g) \| \]

- set of initial states closest to \( s_0 \) from which \( S_g \) is reachable:

\[ s_0 \diamond^{\text{forbus}} \text{Fml}(S'_0) \]

Alternatively: \( s_0 \diamond^{\text{pma}} \text{Fml}(S'_0) \) [Göbelbecker et al., ICAPS 2010]
Changing the goal

- given: planning task \((PVar, A, s_0, S_g)\)
- set of candidate goal states:

\[
S'_g = \{ s'_g : s'_g \text{ is reachable from } s_0 \text{ via } A \}
= \|\langle \text{iterate}_A^{-1}\rangle Fml(s_0)\| \\

- set of goal states closest to \(S_g\) that are reachable from \(s_0\):

\[
S_g^*_{dalal} Fml(S'_g)
\]

\[
S_g^*_{dalal} \|\langle \text{iterate}_A^{-1}\rangle Fml(s_0)\| = \|\langle (\pi_{dalal}(\langle \text{iterate}_A^{-1}\rangle Fml(s_0), Fml(S_g)))^{-1}\rangle Fml(S_g)\|
\]
embedding of prominent belief change operations into DL-PA
- Forbus’s operation operation
- Winslett’s update operations WSS and PMA
- Dalal’s revision operation
- Satoh’s revision operation

embedding of planning tasks and their modifications
  involves update/revision by counterfactuals

allows for the syntactical construction of the result

range of applicability of DL-PA:
- coalition logic of propositional control [Herzig et al, IJCAI 2011]
- normative systems [Herzig et al, CLIMA 2011]
- deontic action logic [Herzig et al, DEON 2012]
- update of ASP programs [Fariñas et al., LPNMR 2013]
- belief merging [Herzig et al., FOIKS 2014]
- modification of abstract argumentation frameworks à la Dung [Doutre et al., KR 2014]