Planification et logique : une longue histoire

Andreas Herzig
University of Toulouse, IRIT-CNRS, France

JFPDA @ PFIA 2015, Rennes, 1 juillet 2015

What has planning to do with logic?

- where should I publish my paper on action and planning?

  case
  when (complex concepts$\vee$complex models) **then** submit(KR)**;**
  when (implemented$\wedge$fast) **then** submit(ICAPS)
  esac

- two diverging communities
  - logicians and most KR people focus on concepts and models
  - planning community focus on efficient reasoning

What has planning to do with logic? (ctd.)

- since 2012: KR and ICAPS no longer colocated

	ICAPS		KR
2004		Canada	
2006		UK	
2008		Australia	
2010		Canada	
2012	Brazil		Italy
2014	USA		Austria
2016	?		South Africa

- since ~2012:
  - the planning community goes multiagent
  - needs more complex concepts and models

The logic engineering perspective

logic = semantics + reasoning

1. semantics
   - which language?
     - which concepts?
     - which logical form? (arguments,. . .)
     - has truth values? (facts do, actions don't)
   - which models?
2. reasoning
   - non-mechanisable (Hilbert-style axiomatisations, natural deduction . . .)

     ⇒ complete? decidable?
   - mechanisable methods: sequent systems, resolution, Davis&Putnam,. . . , semantic tableaux; model checking

     ⇒ complete? decidable?

     ⇒ worst/average case complexity? implementations?

The four central concepts in planning

1. initial state(s)
   - made up of fluents
   - simplest: state = each fluent either true or false
     = valuation of classical propositional logic
   - alternatively: proba/fuzzy/epistemic/...logic
2. goal
   - simplest: set of states (alternatively: proba/...)
   - more challenging:
     - temporal logics
     - logics of goals and intentions (BDI logics)
       $\Rightarrow$ beliefs, goals, committed goals (intentions), plans, actions

The four central concepts in planning

3 action (alias planning operator)

- simplest model: action = $\langle \mathtt{precond}, \mathtt{add}, \mathtt{del} \rangle$
- more challenging:
  - conditional effects, sensing
  - nondeterministic effects, probabilistic effects
  - domain laws

  $\Rightarrow$ many KR problems: frame problem, ramification problem, qualification problem

4 plan

- simplest: sequence of actions
- more challenging:
  - conditional plans (if-then-else), knowledge-based programs
  - high-level intentions and plans + refinement (BDI model)
  - strategies (coalition against its opponents)

Outline

1. A short history of planning and logic
2. A simple logic of actions and plans
3. Planning tasks in DL-PA
4. Updating and revising by DL-PA programs
5. Planning task modification in DL-PA
6. Conclusion

Theory vs. Practice: 1970-1990

- practice: first steps
  - General Problem Solver
  - classical planning: STRIPS [Fikes&Nilsson 1971]
- theory: many tentatives
  - logics plagued by the frame problem:
    - Algorithmic Logic [Salwicki 1970]
    - Dynamic Logic [Pratt 1976, Segerberg 1977]
    - Linear Temporal Logic [Pnueli 1977, Gabbay 1980]
  - complicated action formalisms:
    - SitCalc [McCarthy 1963]
    - EventCalc [Kowalski&Sergot 1986]
    - FluentCalc [Thielscher 1997]
  - and an UFO: Linear Logic [Girard 1987]

Theory vs. Practice: 1990-2000

- theory: some paradigms emerge
  - Reiter's SitCalc solution to the frame problem [Reiter 1991]
    - successor state axioms model conditional effects
    - requires 2nd-order logic!
  - complicated belief-desire-intention (BDI) logics [Cohen&Levesque 1990; Rao&Georgeff 1990]
    - desires $\Rightarrow$ can be inconsistent
    - intentions commit agent to act $\Rightarrow$ must be consistent
    - Cohen&Levesque require 2nd-order logic!
- practice: successful planners
  - based on boolean SAT solvers
  - based on SMT solvers
  - based on heuristic search
  - ...

Theory vs. Practice: 2000-2010

- theory: mature formalisms
  - game theory-inspired logics for strategic reasoning: Coalition Logic [Pauly 2000], Alternating-time Temporal Logic ATL [Alur et al. 1997], ATL*, Strategy Logic [Mogavero et al. 2010]
    - "coalition of agents $\{i_1, \ldots, i_n\}$ has a strategy to achieve φ"
  - Dynamic Epistemic Logics (DELs): Public Announcement Logic [Plaza 1989], Group Announcement Logic [Ågotnes et al. 2010],...
    - "after the truthful public announcement that φ is true, ψ will hold"
    - "coalition of agents $\{i_1, \ldots, i_n\}$ can achieve common knowledge of φ"
    - SAT problem often in PSpace
  - Separation Logic [Reynolds, O'Hearn et al. 2002]
    - resource logic (successors of linear logic)
    - 'built-in' solution to the frame problem
- practice: consolidation
  - PDDL [McDermott 1998/2000]; benchmarks & competitions
  - implemented BDI agents
    - plan libraries only

$\Rightarrow$ remained single-agent & diverged from logic

Theory vs. Practice: 2010-2020

- theory and practice converge
  - Dagstuhl workshops on multiagent planning in 2008, 2014
  - ‘ICAPS goes multiagent’
    - ICAPS 2005 and 2008 Multiagent Planning Workshop
    - since ICAPS 2013: workshop series ‘Distributed and Multi-Agent Planning’ (DMAP)
      [Petrick, Geffner, Domshlak, Brafman, Kambhampati, Nebel,. . .]
  - ‘DEL goes planning’
    [Bolander, van der Hoek, Wooldridge, Aucher, Schwarzentruber,. . .]
    - difficult: plan existence undecidable in general [Aucher&Bolander 2013], in ExpSpace in some cases [Bolander et al. 2015]
  - simpler BDI logics get simpler
    [Shoham 2009, Icard et al. 2010, van Zee et al. 2015]
    - ‘database perspective’

This talk

- propaganda for a simple logic of actions and plans allowing for planning with conditional and nondeterministic effects
  - similar to but more natural than QBF
  - based on propositional assignments
  - decidable
  - SAT/validity/model checking problem: all PSpace complete

  ⇒ *in* the logic! (cf. Hilbert's program)
- account of visibility-based epistemic reasoning

  ⇒ v. Faustine's talk
- account of planning problem modification
  [Smith, ICAPS 2004; Göbelbecker et al., ICAPS 2010, Herzig et al., ECAI 2014]

  ⇒ v.i.

Outline

Extending the language of QBF

- 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

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

- beyond: talk about programs modifying valuations
  $\Rightarrow$ assignments of propositional variables to truth values

$$\begin{array}{lll} s \models \langle p \leftarrow \top \rangle \varphi & \text{if} & s \cup \{p\} \models \varphi \\ s \models \langle p \leftarrow \bot \rangle \varphi & \text{if} & s \setminus \{p\} \models \varphi \end{array}$$

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

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

- nondeterministic composition

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

- finite iteration ('Kleene star')

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

- test

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

- converse, intersection,. . .

Capturing standard programming languages

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

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 $||\varphi|| \subseteq 2^{\mathbb{P}}$

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

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

Semantics of DL-PA: (1) formulas

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

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

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

Semantics of DL-PA: (2) programs

- interpretation of a program π = 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:

$$(\pi_1 ; \pi_2)^{-1} \equiv \pi_2^{-1} ; \pi_1^{-1} \qquad p{\leftarrow}\top^{-1} \equiv p? ; (\mathrm{skip} \cup p{\leftarrow}\bot)$$
$$\ldots \qquad p{\leftarrow}\bot^{-1} \equiv \ldots$$

   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]
    - in [Balbiani et al., LICS 2013] PDL: SAT is EXPTIME complete
  - consequence relation is compact and has interpolation
    - fails for PDL
- rest of talk:
  - how to capture planning and plan task modifications

Outline

Classical planning

- classical planning task:

$\langle\mathbb{P},$	finite set of propositional variables
$\mathtt{s}_0,$	initial state
$S_g,$	set of goal states
$\mathtt{A}\rangle$	finite set of STRIPS actions

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

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

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

- s is *reachable* from $\mathtt{s}_0$ via $\mathtt{A}$ iff ...
- planning task is *solvable* iff some state in S_g is reachable from $\mathtt{s}_0$ via $\mathtt{A}$

Classical planning tasks in DL-PA

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

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

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

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

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

Beyond classical planning

- nondeterministic effects:

  if pre_a
  then $\pi_1 \cup \pi_2$

- conditional effects:

  if $\text{pre}_a \wedge C_1$
  then π_1
  else if $\text{pre}_a \wedge C_2$
  then π_2

  (precise definition requires copies of variables)

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 $2^{\mathbb{P}}$

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"

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

- 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 ⇒ 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*

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

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*

$$\begin{aligned}&\mathtt{vary}\big(\mathbb{P}_B\big) \,;\, B? \,;\\ &\left(\bigcup_{0 \le m \le \text{card}(\mathbb{P}_A)} \big([\mathtt{vary}\big(\mathbb{P}_B\big) \,;\, B?]\mathrm{H}(A, \ge m)\big)? \,;\, \mathtt{flip1}^m\big(\mathbb{P}_A\big)\right); A?\end{aligned}$$

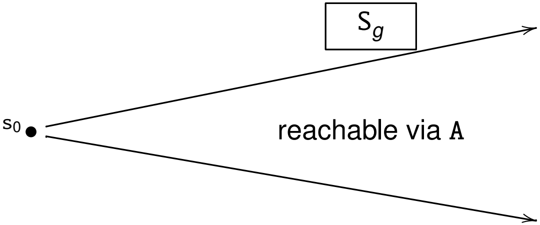
Then for satisfiable B*:* $B *^{\text{dalal}} A = \|\langle(\pi^{\text{dalal}}(A, B))^{-1}\rangle\top\|$.

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

Outline

Planning task modification

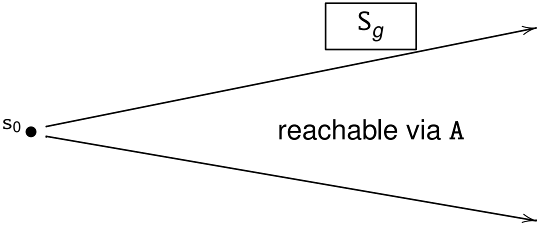
- suppose $(\mathbb{P}, \mathtt{A}, s_0, S_g)$ has no solution



- 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 $\mathtt{A}$
  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

Changing the initial state

- candidate initial states:

  $S'_0 = \{s'_0 \; : \;$ there is a goal state that is reachable from s'_0 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 ⇒ 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

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:

$$\mathtt{S}'_g = \left\| \langle \mathtt{iterate}_{\mathtt{A}}^{-1} \rangle \mathtt{Fml}(\mathtt{s}_0) \right\|$$

Theorem

The set of goal states closest to $\mathtt{S}_g$ that are reachable from $\mathtt{s}_0$ is

$$\begin{aligned}\mathtt{S}_g *^{\mathsf{dalal}} \mathtt{Fml}(\mathtt{S}'_g) &= \left\| \left\langle \left(\pi^{\mathsf{dalal}}(\mathtt{Fml}(\mathtt{S}'_g), \mathtt{Fml}(\mathtt{S}_g)) \right)^{-1} \right\rangle \top \right\| \\ &= \left\| \left\langle \left(\pi^{\mathsf{dalal}}(\langle \mathtt{iterate}_{\mathtt{A}}{}^{-1} \rangle \mathtt{Fml}(\mathtt{s}_0), \mathtt{Fml}(\mathtt{S}_g)) \right)^{-1} \right\rangle \top \right\|\end{aligned}$$

Outline

1. A short history of planning and logic
2. A simple logic of actions and plans
3. Planning tasks in DL-PA
4. Updating and revising by DL-PA programs
5. Planning task modification in DL-PA
6. Conclusion

Conclusion

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 [H.&Feuillade, JELIA 2014]
  - Dung argumentation frameworks
    - enforce a property on all/some extensions = update by a counterfactual
- 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

Conclusion, ctd.

- ongoing: epistemic extension
  - action preconditions become *epistemic actions*
    - public/semi-public/private/... ⇒ 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?
  - grounded versions: cf. Faustine's talk at IAF'2015
- t.b.d.: strategic version
  - based on propositional control (cf. boolean games)