The dynamic logic of assignments

Andreas Herzig
University of Toulouse and CNRS, IRIT, France

joint work with Philippe Balbiani (Toulouse) and Nicolas Troquard (Trento)

Workshop ANR DynRes, Paris, June 2012

Overview

- PDL: abstract actions only

  Propositional Dynamic Logic "abstracts away from the nature of the domain of computation and studies the pure interaction between programs and propositions" [Harel et al. 2000]

- update logics: concrete programs [van Benthem], [Baltag and Moss], [van Ditmarsch et al.], . . .
  - $\varphi!$ = "φ is publicly announced"
    - relativise model M to $||\varphi||_M$: $W^\varphi = ||\varphi||_M$, $R^\varphi = R|_{||\varphi||_M}$
    - alternatively: $R^\varphi = R \cap (W \times ||\varphi||_M)$
  - $p{\leftarrow}\varphi$ = "p is publicly assigned the truth value of φ"
    - $V^{p\leftarrow\varphi}(q) = \begin{cases} ||\varphi||_M & \text{if } q = p \\ V(p) & \text{if } q \neq p \end{cases}$

Overview

- PDL: abstract actions only

  Propositional Dynamic Logic "abstracts away from the nature of the domain of computation and studies the pure interaction between programs and propositions" [Harel et al. 2000]

- update logics: concrete programs [van Benthem], [Baltag and Moss], [van Ditmarsch et al.], ...
  - $\varphi!$ = "φ is publicly announced"
    - relativise model M to $||\varphi||_M$: $W^\varphi = ||\varphi||_M$, $R^\varphi = R|_{||\varphi||_M}$
    - alternatively: $R^\varphi = R \cap (W \times ||\varphi||_M)$
  - $p{\leftarrow}\varphi$ = "p is publicly assigned the truth value of φ"
    - $V^{p\leftarrow\varphi}(q) = \begin{cases} ||\varphi||_M & \text{if } q = p \\ V(p) & \text{if } q \neq p \end{cases}$

Outline

1. The logic of public announcements and public assignments
2. Dynamic Logic of Propositional Assignments
3. Reasoning about agents' capabilities: encoding coalition logic of propositional control

PAL-PA: language

- $\mathtt{Prp} = \{p, q, \ldots\}$ = set of propositional variables
- events:
  - $\varphi!$ = "φ is publicly announced"
  - $p \leftarrow \varphi$ = "p is publicly assigned the truth value of φ"
    - N.B.: don't confuse with assignments of object variables $x \leftarrow t$ of first-order Dynamic Logic
  - lists of public assignments
    - ϵ = empty list
    - executed in parallel
    - in case of conflict: leftmost assignments wins
      $\alpha = (p \leftarrow \bot, p \leftarrow \top)$ makes p false
  - complex events: ...
- formulas: ...

PAL-PA: language, ctd.

- BNF for *assignments* α, *programs* π and *formulas* φ:

$$\begin{array}{lcl} \alpha & ::= & \epsilon \mid (p{\leftarrow}\varphi, \alpha) \\ \pi & ::= & \alpha \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \mid \varphi? \mid \varphi! \\ \varphi & ::= & p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \vee \varphi \mid [\pi]\varphi \mid K\varphi \end{array}$$

- else just as PDL:
  - $\texttt{skip} \stackrel{\text{def}}{=} \top?$
  - $\texttt{if } \varphi \texttt{ then } \pi_1 \texttt{ else } \pi_2 \stackrel{\text{def}}{=} \ldots$
  - $\texttt{while } \varphi \texttt{ do } \pi \stackrel{\text{def}}{=} \ldots$

- for ease of presentation: single agent
  - but everything extends to multiagent case

Models

- S5 models: $M = \langle W, \sim, V \rangle$ such that
  - W nonempty set
  - $\sim \subseteq W \times W$ equivalence relation
  - $V : \mathtt{Prp} \longrightarrow 2^W$ valuation

- interpretation of a formula = set of pointed models
  - $||\varphi|| = \{(M_1, w_1), (M_2, w_2), \ldots\}$

- interpretation of a modality =
  relation on the set of pointed models
  - $||\pi|| = \left\{\langle (M_1, w_1), (M'_1, w'_1) \rangle,\ \langle (M_2, w_2), (M'_2, w'_2) \rangle, \ldots \right\}$

Interpretation of formulas

business as usual:

$$
\begin{aligned}
||\top|| &= \{(M,w) : (M,w) \text{ is a pointed S5 model}\} \\
||\bot|| &= \emptyset \\
||p|| &= \{(M,w) : w \in V(p)\} \\
||\neg\varphi|| &= \ldots \\
||\varphi\vee\psi|| &= \ldots \\
||\Box\varphi|| &= \{(M,w) : \text{for every } (M',w') \text{ s.th. } (M,w)||\Box||(M',w'), \\
&\qquad (M',w') \in ||\varphi||\}
\end{aligned}
$$

where $\Box$ is any modal operator

Interpretation of epistemic operators

change actual world w according to epistemic relation $\sim$:

$$(M, w) \,\|K\|\, (M', w') \quad \text{iff} \quad \begin{cases} W' &= W, \\ \sim' &= \sim, \\ V' &= V, \\ w' &\sim w \end{cases}$$

Interpretation of announcements

relativisation:

$$(M, w)\ ||\varphi!||\ (M', w') \quad \text{iff} \quad \begin{cases} W' & = W \cap ||\varphi||_{\mathbf{M}}, \\ \sim' & = \sim \cap\, (W' \times W'), \\ V'(p) & = V(p) \cap W', \\ w' & = w \end{cases}$$

where $||\varphi||_{\mathbf{M}}$ is the extension of φ in M:

$$\begin{aligned} ||\varphi||_{\mathbf{M}} &= \{w \ : \ (M, w) \in ||\varphi||\} \\ &= ||\varphi|| \cap \{(M, w) \ : \ w \text{ world of } M\} \end{aligned}$$

Interpretation of assignments

update valuation V by list of assignments α:

$$(M, w) \, \|\alpha\| \, (M', w') \quad \text{iff} \quad \begin{cases} W' & = W, \\ \sim' & = \sim, \\ V'(p) & = \|\alpha(p)\|_M, \\ w & = w' \end{cases}$$

where list applies with priority to leftmost assignments:

$$\epsilon(p) = p$$

$$(q \leftarrow \varphi, \alpha)(p) = \begin{cases} \varphi & \text{if } q = p \\ \alpha(p) & \text{if } q \neq p \end{cases}$$

Interpretation of complex programs

business as usual:

$$
\begin{array}{lcl}
||\pi_1 ; \pi_2|| & = & ||\pi_1|| \circ ||\pi_2|| \\
||\pi_1 \cup \pi_2|| & = & ||\pi_1|| \cup ||\pi_2|| \\
||\pi^*|| & = & (||\pi||)^* \\
||\varphi?|| & = & \{ \langle (M, w), (M, w) \rangle \ : \ w \in ||\varphi||_M \}
\end{array}
$$

Satisfiability and validity

- business as usual:

$$\begin{array}{lll} \varphi \text{ satisfiable} & \text{iff} & ||\varphi|| \neq ||\bot|| \\ \varphi \text{ is valid} & \text{iff} & ||\varphi|| = ||\top|| \end{array}$$

Complexity of satisfiability for fragments of PAL-PA

1. the whole language: undecidable [Miller&Moss 2003]
   - announcements $\varphi!$ and Kleene star π^* are enough
2. no PDL operators, no assignments: decidable [Plaza 1989]
   - monoagent case: NP complete [Lutz 2007]
   - multiagent case: PSPACE complete [Lutz 2007]
   - common knowledge: EXPTIME complete [Lutz 2007]
3. no complex programs: decidable [van Ditmarsch et al. 2007]
   - complexity as above [van Ditmarsch et al., JANCL 2012]
4. non-epistemic fragment: decidable (v.i.)
   - no complex programs: NP complete (apply reduction axioms)
   - no π^*: PSPACE complete [Herzig et al. IJCAI 2011]
   - whole fragment: PSPACE complete (v.i.)

Outline

1. The logic of public announcements and public assignments
2. Dynamic Logic of Propositional Assignments
3. Reasoning about agents' capabilities: encoding coalition logic of propositional control

Dynamic Logic of Propositional Assignments DL-PA

DL-PA = non-epistemic fragment of PAL-PA:

$$\pi \quad ::= \quad p{\leftarrow}\top \mid p{\leftarrow}\bot \mid \pi;\pi \mid \pi \cup \pi \mid \pi^* \mid \varphi?$$

- pointed model = a single valuation [v. Eijck 2000]
- $p{\leftarrow}\varphi$ has same interpretation as $(\varphi?; p{\leftarrow}\top) \cup (\neg\varphi?; p{\leftarrow}\bot)$
- plus abstract actions à la PDL: undecidable [Tiomkin and Makowsky 1985]

DL-PA: decision procedure

key step: eliminate the Kleene star

1. choose some π^* such that π is star-free
2. transform π into

$$(\varphi_1?;\alpha_1) \cup \cdots \cup (\varphi_n?;\alpha_n)$$

   where every α_k is a sequence of assignments
3. make all the assignment sequences α_k assign exactly the same variables:

$$(\varphi_1?;\alpha_1) \cup \cdots \cup (\varphi_n?;\alpha_n) \quad \text{and} \quad \mathtt{Prp}_{\alpha_1} = \ldots = \mathtt{Prp}_{\alpha_n}$$

4. replace π^* by

$$((\varphi_1?;\alpha_1) \cup \cdots \cup (\varphi_n?;\alpha_n))^{\leq n}$$

(uses that $\mathtt{Prp}_{\alpha_k} = \mathtt{Prp}_{\alpha_l}$ implies $\alpha_k;\alpha_l = \alpha_l$)

DL-PA: complexity

Theorem

DL-PA model checking is PSPACE-complete.

- hardness: encode QBF
- membership: deterministic algorithm working in polynomial space

Theorem

DL-PA satisfiability checking is PSPACE-complete.

- hardness: encode QBF
- membership:
  1. satisfiability is in NPSPACE:
     - guess valuation V
     - model check in PSPACE whether $V \in ||\varphi||$ (v.s.)
  2. NPSPACE $=$ PSPACE [Savitch]

Outline

Propositional control in one slide

- *Coalition Logic of Propositional Control* CL-PC [v.d. Hoek et al. AIJ 2005, JAIR 2010]
  - stem from the language of ATL model checker MOCHA
  - model = valuation + 'agents control propositional variables'
    - agents can only assign truth values to variables they control
  - language:
    $\langle J \rangle \varphi$ = "coalition J can achieve φ (*if other agents do nothing*)"
  - express capability operator of Coalition Logic CL:
    $\langle J \rangle [\bar{J}] \varphi$ = "J can achieve φ (*whatever the other agents do*)"

- in DL-PA:
  - model = valuation (non epistemic)
  - language:
    $\langle p \leftarrow \top \rangle \varphi$ = "after making p true, φ will be true"
    $\langle p \leftarrow \bot \rangle \varphi$ = "after making p false, φ will be true"
  - 'get more for the same price':
    - polynomial translation of CL-PC
    - same complexity as CL-PC
    - extensible: norms, counts-as relation, knowledge, ...

Propositional control in one slide

- *Coalition Logic of Propositional Control* CL-PC
  [v.d. Hoek et al. AIJ 2005, JAIR 2010]
  - stem from the language of ATL model checker MOCHA
  - model $=$ valuation + 'agents control propositional variables'
    - agents can only assign truth values to variables they control
  - language:
    $\langle J \rangle \varphi$ $=$ "coalition J can achieve φ (*if other agents do nothing*)"
  - express capability operator of Coalition Logic CL:
    $\langle J \rangle [\bar{J}] \varphi$ $=$ "J can achieve φ (*whatever the other agents do*)"

- in DL-PA:
  - model $=$ valuation (non epistemic)
  - language:

$$\langle p \leftarrow \top \rangle \varphi \quad = \quad \text{"after making } p \text{ true, } \varphi \text{ will be true"}$$
$$\langle p \leftarrow \bot \rangle \varphi \quad = \quad \text{"after making } p \text{ false, } \varphi \text{ will be true"}$$

  - 'get more for the same price':
    - polynomial translation of CL-PC
    - same complexity as CL-PC
    - extensible: norms, counts-as relation, knowledge, . . .

Ability to perform an assignment

- finite set of agents $\mathbb{A} = \{i, j, \ldots\}$
- countable set of propositional variables $\mathtt{Prp}$ is such that

$$\mathtt{Prp} = \mathtt{Prp}^0 \cup \{\mathrm{A}_i(p{\leftarrow}\top), \mathrm{A}_i(p{\leftarrow}\bot) \ : \ i \text{ agent}, p \text{ variable}\}$$

  - $\mathtt{Prp}^0 =$ *basic* atomic facts
  - $\mathrm{A}_i(p{\leftarrow}\top) =$ "*i* is able to make *p* true"
  - $\mathrm{A}_i(p{\leftarrow}\bot) =$ "*i* is able to make *p* false"
- basic assignments $\alpha^0 =$ assignment of variable in $\mathtt{Prp}^0$
- also possible:
  - higher-order assignments
    - $\mathrm{A}_j(p{\leftarrow}\top){\leftarrow}\bot =$ hinder *j* to set *p* to true
    - ...
  - higher-order abilities
    - $\mathrm{A}_i(\mathrm{A}_j(p{\leftarrow}\top){\leftarrow}\bot) =$ *i* can hinder *j* to set *p* to true
    - ...

Ability to perform an assignment

- finite set of agents $\mathbb{A} = \{i, j, \ldots\}$
- countable set of propositional variables $\mathtt{Prp}$ is such that

$$\mathtt{Prp} = \mathtt{Prp}^0 \cup \{\mathrm{A}_i(p{\leftarrow}\top), \mathrm{A}_i(p{\leftarrow}\bot) \ : \ i \text{ agent}, p \text{ variable}\}$$

  - $\mathtt{Prp}^0$ = *basic* atomic facts
  - $\mathrm{A}_i(p{\leftarrow}\top)$ = "*i* is able to make *p* true"
  - $\mathrm{A}_i(p{\leftarrow}\bot)$ = "*i* is able to make *p* false"
- basic assignments α^0 = assignment of variable in $\mathtt{Prp}^0$
- also possible:
  - higher-order assignments
    - $\mathrm{A}_j(p{\leftarrow}\top){\leftarrow}\bot$ = hinder *j* to set *p* to true
    - ...
  - higher-order abilities
    - $\mathrm{A}_i(\mathrm{A}_j(p{\leftarrow}\top){\leftarrow}\bot)$ = *i* can hinder *j* to set *p* to true
    - ...

Ability to perform an assignment

- finite set of agents $\mathbb{A} = \{i, j, \ldots\}$
- countable set of propositional variables $\mathtt{Prp}$ is such that

$$\mathtt{Prp} = \mathtt{Prp}^0 \cup \{\mathrm{A}_i(p{\leftarrow}\top), \mathrm{A}_i(p{\leftarrow}\bot) \; : \; i \text{ agent}, p \text{ variable}\}$$

  - $\mathtt{Prp}^0$ = *basic* atomic facts
  - $\mathrm{A}_i(p{\leftarrow}\top)$ = "i is able to make p true"
  - $\mathrm{A}_i(p{\leftarrow}\bot)$ = "i is able to make p false"
- basic assignments α^0 = assignment of variable in $\mathtt{Prp}^0$
- also possible:
  - higher-order assignments
    - $\mathrm{A}_j(p{\leftarrow}\top){\leftarrow}\bot$ = hinder j to set p to true
    - ...
  - higher-order abilities
    - $\mathrm{A}_i(\mathrm{A}_j(p{\leftarrow}\top){\leftarrow}\bot)$ = i can hinder j to set p to true
    - ...

Basic capability to achieve a state of affairs

$\Diamond_J^{\mathrm{A}^0}\varphi \quad = \quad$ "coalition J can achieve φ by J's basic assignments (if other agents do nothing)"

- interpretation of capability operator:

$V \| \Diamond_J^{\mathrm{A}^0} \| V'$ iff there are *basic* assignments $\alpha_1^0, \ldots, \alpha_n^0$ s.th.
(a) $V \| \alpha_1^0; \ldots; \alpha_n^0 \| V'$
(b) for every α_k^0 there is $i \in J$ with $V \in \|\mathrm{A}_i(\alpha_k^0)\|$

(same as Coalition Logic of Propositional Control CL-PC)

Basic capability to achieve a state of affairs

$\Diamond_J^{\mathrm{A}^0}\varphi \quad = \quad$ "coalition J can achieve φ by J's basic assignments (if other agents do nothing)"

- interpretation of capability operator:

$V \| \Diamond_J^{\mathrm{A}^0} \| V'$ iff there are *basic* assignments $\alpha_1^0, \ldots, \alpha_n^0$ s.th.
(a) $V \| \alpha_1^0; \ldots; \alpha_n^0 \| V'$
(b) for every α_k^0 there is $i \in J$ with $V \in \|\mathrm{A}_i(\alpha_k^0)\|$

(same as Coalition Logic of Propositional Control CL-PC)

Basic capability to achieve a state of affairs

$\Diamond_J^{\mathrm{A}^0}\varphi$ = "coalition J can achieve φ by J's basic assignments (if other agents do nothing)"

- interpretation of capability operator:

$V \| \Diamond_J^{\mathrm{A}^0} \| V'$ iff there are *basic* assignments $\alpha_1^0, \ldots, \alpha_n^0$ s.th.
(a) $V \| \alpha_1^0; \ldots; \alpha_n^0 \| V'$
(b) for every α_k^0 there is $i \in J$ with $V \in \|\mathrm{A}_i(\alpha_k^0)\|$

(same as Coalition Logic of Propositional Control CL-PC)

Basic capability: embedding CL-PC

Theorem

Formula φ is satisfiable in CL-PC models iff

$$\varphi \wedge Sym_\varphi \wedge Exh_\varphi \wedge Excl_\varphi$$

is DL-PA satisfiable, where:

$$\begin{array}{lcl}
Sym_\varphi & = & \bigwedge_{i \in \mathbb{A}_\varphi, p \in \mathrm{Prp}_\varphi} \mathrm{A}_i(p{\leftarrow}\top) \leftrightarrow \mathrm{A}_i(p{\leftarrow}\bot) \\
Exh_\varphi & = & \begin{cases} \bigwedge_{p \in \mathrm{Prp}_\varphi} \bigvee_{i \in \mathbb{A}} \mathrm{A}_i(p{\leftarrow}\bot) & \textit{when } \mathbb{A}_\varphi = \mathbb{A} \\ \bigwedge_{p \in \mathrm{Prp}_\varphi} \bigvee_{i \in \mathbb{A}_\varphi \cup \{j_0\}} \mathrm{A}_i(p{\leftarrow}\bot) & \textit{when } \mathbb{A}_\varphi \neq \mathbb{A}, \textit{for some } j_0 \in \mathbb{A} \setminus \mathbb{A}_\varphi \end{cases} \\
Excl_\varphi & = & \bigwedge_{i,j \in \mathbb{A}_\varphi, i \neq j, p \in \mathrm{Prp}_\varphi} \neg\left(\mathrm{A}_i(p{\leftarrow}\bot) \wedge \mathrm{A}_j(p{\leftarrow}\bot)\right)
\end{array}$$

$\Rightarrow$ CL-PC can be polynomially embedded into DL-PA plus $\Diamond_J^{\mathrm{A}^0}$

Basic capability: eliminating $\Diamond_J^{\mathrm{A}^0}$

Theorem

Let $\mathtt{Prp}_\varphi = \{p_1, \ldots, p_n\}$ *the propositional variables occurring in* φ*. Then:*

$$\Diamond_J^{\mathrm{A}^0}\varphi \leftrightarrow$$

$$\langle \mathtt{skip} \cup (\bigvee_{i\in J} \mathrm{A}_i(p_1{\leftarrow}\top)?; p_1{\leftarrow}\top) \cup (\bigvee_{i\in J} \mathrm{A}_i(p_1{\leftarrow}\bot)?; p_1{\leftarrow}\bot)\rangle$$

$$\vdots$$

$$\langle \mathtt{skip} \cup (\bigvee_{i\in J} \mathrm{A}_i(p_n{\leftarrow}\top)?; p_n{\leftarrow}\top) \cup (\bigvee_{i\in J} \mathrm{A}_i(p_n{\leftarrow}\bot)?; p_n{\leftarrow}\bot)\rangle \varphi$$

$\Rightarrow \Diamond_J^{\mathrm{A}^0}$ can be polynomially reduced to DL-PA formulas

Conclusions

- DL-PA = PDL with concrete programs
  - PSPACE complete
- DL-PA, PAL-PA = 'Swiss knife' for MAS
  - concrete programs provide for an appropriate modelling in all concrete applications
    - embeds van der Hoek and Wooldridge's CL-PC
    - distinguish physical and *legal* ability [Herzig et al., CLIMA 2011]
    - Reiter's solution to the Frame Problem in reasoning about actions [Reiter 1990] can be polynomially encoded in DL-PA [van Ditmarsch et al., JLC 2012]
    - do multi-agent simulation in logic (Schelling's segregation game) [Gaudou et al., MABS 2011]