Uniform Strategies in the Dynamic Epistemic Logic of Propositional Control

Andreas Herzig
University of Toulouse, IRIT-CNRS, France
www.irit.fr/~Andreas.Herzig

joint work with

Tiago de Lima (Lens), Emiliano Lorini (Toulouse), Nicolas Troquard (Trento)

Workshop “Dynamics in Logic II”
Lille, March 1, 2012

An invitation to join www.sintelnet.eu

- "European Network for Social Intelligence" (SINTELNET)
  - FET open coordination action
  - coordination: David Pearce, Dirk Walther
- idea: revisit basic concepts of philosophy, humanities and social sciences in the light of new forms of information technology-enabled social environments
- actions:
  - working groups:
    1. action and agency (chair: Marek Sergot)
    2. interactive communication (chair: Andrew Jones)
    3. group attitudes (chairs: Andreas Herzig, Emiliano Lorini)
    4. socio-technical epistemology (chairs: Cristiano Castelfranchi, Luca Tummolini)
    5. social coordination (chair: Pablo Noriega)
  - interdisciplinary workshops
  - short term academic visits
  - production of guidelines and policy documents
- just started $\Rightarrow$ join!

Uniform Strategies in the Dynamic Epistemic Logic of Propositional Control

Andreas Herzig
University of Toulouse, IRIT-CNRS, France
www.irit.fr/~Andreas.Herzig

joint work with

Tiago de Lima (Lens), Emiliano Lorini (Toulouse), Nicolas Troquard (Trento)

Workshop “Dynamics in Logic II”
Lille, March 1, 2012

Motivation

2 very different families of logics of action:

1. $\langle\pi\rangle\varphi$ = "there is a possible execution of π after which φ"
   - aim: prove correctness of programs
     - algorithmic logic [Salwicki 1970]
   - typically: dynamic logics [Pratt 1976; Parikh; Segerberg;. . .]
   - focus: both means (program π) and result (proposition φ)
2. $\mathtt{Stit}_i\varphi$ = "agent *i* sees to it that φ (whatever $-i$ does)"
   - focus: result of action
   - aim: clarify "being agentive for a proposition"
   - typically: stit logics [von Kutschera, Belnap, Perloff, Horty, Wölfl,. . .]
   - embed Alternating-time Temporal Logic ATL [Broersen, Herzig&Troquard 2007]
   - reasoning about uniform strategies: better than ATEL [Herzig&Troquard 2006; Broersen et al. 2009; Herzig&Lorini2011]

⇒ relation? blend?

Motivation

2 very different families of logics of action:

1. $\langle\pi\rangle\varphi$ = "there is a possible execution of π after which φ"
   - aim: prove correctness of programs
     - algorithmic logic [Salwicki 1970]
   - typically: dynamic logics [Pratt 1976; Parikh; Segerberg;. . .]
   - focus: both means (program π) and result (proposition φ)
2. $\mathtt{Stit}_i\varphi$ = "agent *i* sees to it that φ (whatever $-i$ does)"
   - focus: result of action
   - aim: clarify "being agentive for a proposition"
   - typically: stit logics
     [von Kutschera, Belnap, Perloff, Horty, Wölfl,. . .]
   - embed Alternating-time Temporal Logic ATL
     [Broersen, Herzig&Troquard 2007]
   - reasoning about uniform strategies: better than ATEL
     [Herzig&Troquard 2006; Broersen et al. 2009; Herzig&Lorini2011]

⇒ relation? blend?

Dynamic logic: advantages and shortcomings

advantages:

- means-end reasoning
- program operators
- standard possible worlds semantics

shortcomings:

1. no agents
   - action = event brought about by an agent
2. about opportunity rather than about action itself
   - $\langle\pi\rangle\varphi$ = “there is a possible execution of π such that…”
     $\Rightarrow$ no reasoning about what I am actually doing
3. not suited for reasoning about actions in AI [McCarthy, Reiter]
   - no solution to the frame problem

1st idea: add agents to dynamic logic programs

- language:

$$\alpha ::= i{:}\pi_0 \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \varphi?$$

  where π_0 is an atomic program and i is an agent

- semantics: an agent's action repertoire
  [van der Hoek et al., AIJ 2005; Herzig et al., IJCAI 2011]

2nd idea: add a ‘do’ modality to dynamic logic

- language: another dynamic operator
  [Cohen&Levesque 1990; Herzig&Lorini, JoLLI 2010; . . .]

  $\langle\alpha\rangle\varphi$ = “∃ possible execution of α s.th. φ true afterwards”
  $\langle\!\langle\alpha\rangle\!\rangle\varphi$ = “α is going to be executed and φ is true afterwards”

- semantics: integrate linear time (histories)

3rd idea: solve the frame problem in dynamic logic

- language: atomic programs = propositional assignments
  [van Ditmarsch, Herzig&de Lima, JLC 2011]

  $$p{:=}\varphi$$

  $\Rightarrow$ frame axioms 'built in':

  $$\models q \rightarrow \langle p{:=}\varphi \rangle q \quad \text{for } p \neq q$$

- semantics: assignments update models (cf. DEL)

Outline

Propositional Assignments

- propositional DL: “abstracts away from the nature of the domain of computation and studies the pure interaction between programs and propositions” [Harel et al. 2000]
  - abstract atomic programs
  - interpreted by accessibility relations
- first-order DL: assignments $x{:=}t$ of object variables to terms
  - example: $x{:=}x{+}1$
- here: assignments of propositional variables to truth values (“commands” [v. Eijck 2000])

$$\begin{array}{lcl} +p & = & \text{“make } p \text{ true”} \\ -p & = & \text{“make } p \text{ false”} \end{array}$$

Adding assignments to dynamic logic

[Tiomkin&Makowsky 1985; Wilm 1991; v. Eijck 2000]

- abstract programs plus assignments: two options
  - $\pm p$ modifies valuations of possible worlds globally
    $\Rightarrow$ meaning?
  - $\pm p$ modifies valuations of possible worlds locally
    $\Rightarrow$ undecidable [Tiomkin&Makowsky 1985]
- here: atomic programs = assignments [v.Eijck 2000]
  - no abstract programs
  - a single possible world is enough
    - model = valuation of classical propositional logic
    - small (interesting for model checking)

Language of DL-PC: assignments

- $\mathbb{P} = \{p, q, \ldots\}$ = set of propositional variables
- assignments:
  - $+p$ = "p becomes true"
  - $-p$ = "p becomes false"
- $+P = \{+p \ : \ p \in P\}$
  - set of positive assignments of the variables in $P \subseteq \mathbb{P}$
- $-P = \ldots$
  - $\ldots$
- $\pm P = +P \cup -P$
  - $\pm p$ = arbitrary assignment from $+\mathbb{P} \cup -\mathbb{P}$

Language of DL-PC: actions and joint actions

- $\mathbb{A} = \{i, j, \ldots\}$ = set of agents ('individuals')
- $\mathcal{JA} = \mathbb{A} \times \pm\mathbb{P}$ = set of all joint actions
    - $i{:}{+p}$ = "i makes p true"
    - $i{:}{-p}$ = ...
- group J's part in joint action α:

$$\alpha_J = \alpha \cap (J \times \pm\mathbb{P})$$
$$= \{i{:}{\pm p} \in \alpha \; : \; i \in J\}$$

Language of DL-PC: action operators, agency operators

- executability of an action (opportunity):
  $\langle\alpha\rangle\varphi \quad = \quad$ "each action in α may happen
  and φ is true after the joint performance of α"
- execution of an action:
  $\langle\langle\alpha\rangle\rangle\varphi \quad = \quad$ "each action in α is going to occur
  and φ is true after the joint performance of α"
- being agentive for a proposition:
  $\mathtt{Stit}_J\varphi \quad = \quad$ "group J sees to it that φ"

Language of DL-PC: formulas

φ	$::=$	$p \mid$	
		$\top \mid$	
		$\neg\varphi \mid$	
		$\varphi \wedge \varphi \mid$	
		$\langle\alpha\rangle\varphi \mid$	opportunity of action
		$\langle\langle\alpha\rangle\rangle\varphi \mid$	action
		$\mathtt{Stit}_J\varphi \mid$	agency
		$\mathbf{X}\varphi$	temporal 'next'

Outline

Valuations and their updates

- $\mathsf{Val} \subseteq \mathbb{P}$ (valuation)
- update of Val by a joint action α
  - what if update by $\{i{:}{+}p, j{:}{-}p\}$?
    $\Rightarrow$ don't change p's truth value

$$\begin{aligned}\mathsf{Val}^{\alpha} \quad = \quad & (\mathsf{Val} \setminus \{p \ : \ \exists i{:}{-}p \text{ in } \alpha \text{ and } \nexists j{:}{+}p \text{ in } \alpha\}) \cup \\ & \{p \ : \ \exists i{:}{+}p \text{ in } \alpha \text{ and } \nexists j{:}{-}p \text{ in } \alpha\}\end{aligned}$$

Action repertoires and their updates

- Rep $\subseteq \mathbb{A} \times \pm\mathbb{P}$ (action repertoire)
  - Rep_i $=$ agent *i*'s repertoire of actions
- joint action $\alpha \in \mathcal{JA}$ respects Rep iff $\alpha \subseteq$ Rep
- update of Rep by joint action α:

$$\text{Rep}^{\alpha} \quad = \quad \text{Rep}$$

(but one may think of actions modifying repertoires)

Successor functions and their updates

- Succ : $\mathcal{JA}^* \longrightarrow \mathcal{JA}$ (successor function)
  - $\mathcal{JA}^*$ = the set of all finite sequences of joint actions
  - Succ(σ) = joint action that will be performed after the sequence of joint actions σ has occurred
  - Succ(nil) = joint action that is going to be performed now

    (nil = empty sequence)

- update of Succ by joint action α:

$$\mathrm{Succ}^{\alpha}(\sigma) \quad = \quad \mathrm{Succ}(\alpha \cdot \sigma)$$

($\alpha \cdot \sigma$ = composition of joint action α with sequence σ)

Models and their updates

- $M = (\mathrm{Val}, \mathrm{Rep}, \mathrm{Succ})$ where
  - $\mathrm{Val} \subseteq \mathbb{P}$ ('valuation')
  - $\mathrm{Rep} \subseteq \mathcal{JA}$ ('repertoire')
  - $\mathrm{Succ} : \mathcal{JA}^* \longrightarrow \mathcal{JA}$ such that $\mathrm{Succ}(\sigma)$ respects Rep, for all σ ('successor function')
- update of M by joint action α:

$$M^\alpha \quad = \quad (\mathrm{Val}^\alpha, \mathrm{Rep}^\alpha, \mathrm{Succ}^\alpha)$$

Varying the successor function

- interpretation of $\texttt{Stit}_J$: quantify over the actions of $-J$ ("whatever the agents outside J do")
- $\text{Succ} \sim_J \text{Succ}'$ iff for all σ, $(\text{Succ}(\sigma))_J = (\text{Succ}'(\sigma))_J$
  - "Succ and Succ′ agree on J's strategy"
- $M \sim_J M'$ iff $\text{Val} = \text{Val}', \text{Rep} = \text{Rep}'$, and $\text{Succ} \sim_J \text{Succ}'$

Truth conditions

for $M = (\mathrm{Val}, \mathrm{Rep}, \mathrm{Succ})$ a DL-PC model:

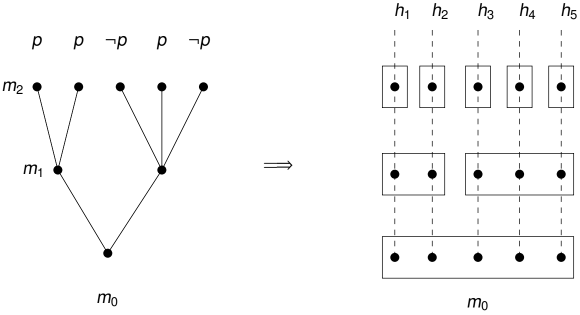
$M \models \langle\alpha\rangle\varphi$ iff $\alpha \subseteq \mathrm{Rep}$ and $M^\alpha \models \varphi$

$M \models \langle\!\langle\alpha\rangle\!\rangle\varphi$ iff $\alpha \subseteq \mathrm{Succ}(\mathrm{nil})$ and $M^\alpha \models \varphi$

$M \models \mathtt{Stit}_J\varphi$ iff $M' \models \varphi$ for every M' such that $M \sim_J M'$
(keep J's part in next joint action; vary $-J$'s part)

Outline

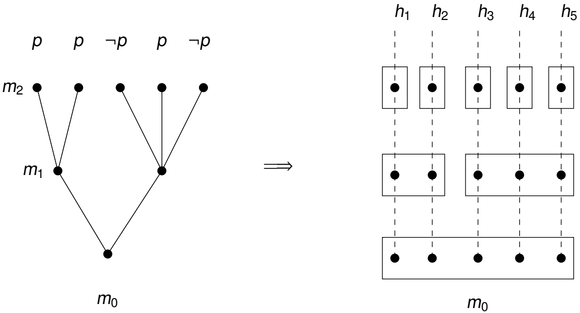
Models of the Chellas stit: branching time (*BT*)



discrete *BT* structure $(Mom, <)$:

- set of moments *Mom*
- relation of temporal precedence $<$
  - history = maximally $<$-ordered set of moments
  - *Hist* = set of all histories
  - $Hist_m$ = set of histories passing through moment m
  - discrete: $succ(m, h)$

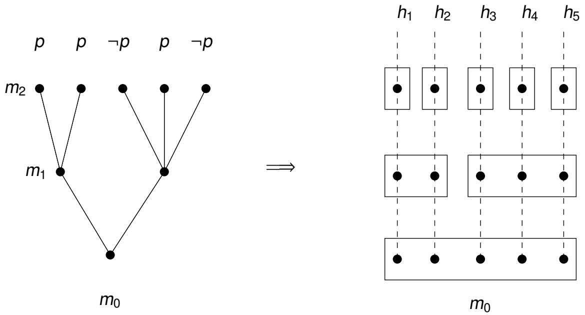
Models of the Chellas stit: agents' choices (AC)



- $Choice : \mathbb{A} \times Mom \longrightarrow Hist \times Hist$ such that each $Choice_i^m$ is an equivalence relation on $Hist_m$

  ($Choice_i^m$ = set of 'choice cells' for agent i at moment m)
- no choice between undivided histories: ...
- independence of agents: ...
- choice function can be extended to groups: $Choice_J^m = \bigcap_{i \in J} Choice_i^m$

$BT+AC$ models



$BT+AC$ model $\mathcal{M} = (Mom, <, Choice, v)$, where:

- $\langle Mom, < \rangle$ is a discrete branching time structure
- *Choice* is a choice function
- $v : (Mom \times Hist) \longrightarrow 2^{\mathbb{P}}$ valuation function

The Chellas stit: truth conditions

formulas evaluated at a moment/history pairs m/h:

$\mathcal{M}, m/h \models p$ iff $p \in v(m/h)$

$\mathcal{M}, m/h \models \neg\varphi$ iff $\ldots$

$\mathcal{M}, m/h \models \varphi \wedge \psi$ iff $\ldots$

$\mathcal{M}, m/h \models \mathtt{X}\varphi$ iff $\mathcal{M}, succ(m, h)/h \models \varphi$

$\mathcal{M}, m/h \models \mathtt{Stit}_J\varphi$ iff $\mathcal{M}, m/h' \models \varphi$ for all h' s.th. $(h, h') \in Choice_J^m$

$\mathtt{Stit}_J\varphi$ = "the alternative that is presently and actually chosen by J guarantees that φ is true"
= "J sees to it that φ"

Outline

Turning a DL-PC model into a $BT{+}AC$ model

for every DL-PC model $M = (\mathrm{Val}, \mathrm{Rep}, \mathrm{Succ})$:

- $Mom_M = (2^{\mathrm{Rep}})^*$ (finite sequences of joint actions)
- $\sigma <_M \sigma'$ iff $\sigma' = \sigma \cdot \sigma''$ for some $\sigma' \neq \mathrm{nil}$ (prefix relation)
  - history = infinite sequence of joint actions
  - $Hist_\sigma$ = histories passing through moment σ
    $= \{h : \sigma \text{ is a prefix of } h\}$
- $Choice_i^\sigma = \{(h, h') : \text{there are } \alpha, \alpha' \text{ such that } \sigma \cdot \alpha \in h, \sigma \cdot \alpha' \in h', \text{ and } \alpha_i = \alpha'_i\}$
- recursive definition of valuation v_M:

$$\begin{aligned} v_M(\mathrm{nil}, h) &= Val \\ v_M(\sigma \cdot \alpha, h) &= (v(\sigma, h))^\alpha \end{aligned}$$

$(Mom_M, <_M, Choice_M, v_M)$ is a discrete $BT{+}AC$ model

The relation with the Chellas stit

- for DL-PC formulas φ without $\langle\langle\alpha\rangle\rangle$, $\langle\alpha\rangle$:
  - $M \models \varphi$ iff $(Mom, <, Choice, v), nil/h_M \models \varphi$

    where $h_M = (\text{nil}, \text{Succ}(\text{nil}), \text{Succ}(\text{Succ}(\text{nil})), \ldots)$
  - if φ is valid in discrete $BT+AC$ models then φ is DL-PC valid
- converse does not hold:
  - $p \to \texttt{Stit}_i p$ valid in DL-PC, but not in $BT+AC$ models
  - $\texttt{Stit}_i(p \vee q) \to (\texttt{Stit}_i \vee \texttt{Stit}_i q)$ valid in DL-PC, but not in $BT+AC$ models
- open question: are there schematic validities distinguishing DL-PC from $BT+AC$ models?

Outline

Decision procedure (1)

simplify $\langle\!\langle . \rangle\!\rangle$:

$$\begin{aligned}
\langle\!\langle \alpha \rangle\!\rangle \varphi &\leftrightarrow \langle \alpha \rangle \varphi \wedge \langle\langle \alpha \rangle\rangle \top \\
\langle\langle \emptyset \rangle\rangle \top &\leftrightarrow \top \\
\langle\langle \alpha \cup \beta \rangle\rangle \top &\leftrightarrow \langle\langle \alpha \rangle\rangle \top \wedge \langle\langle \beta \rangle\rangle \top
\end{aligned}$$

Decision procedure (2)

simplify $\langle . \rangle$:

$$
\begin{aligned}
\langle\alpha\rangle p \quad &\leftrightarrow \quad \begin{cases} \langle\alpha\rangle\top & \text{if } \exists i : i{:}{+}p \in \alpha \text{ and } \nexists j : j{:}{-}p \in \alpha \\ \bot & \text{if } \exists i : i{:}{-}p \in \alpha \text{ and } \nexists j : j{:}{+}p \in \alpha \\ \langle\alpha\rangle\top \wedge p & \text{if } \exists i, j : i{:}{+}p, j{:}{-}p \in \alpha \\ & \text{or } \forall i, j : i{:}{+}p, j{:}{-}p \notin \alpha \end{cases} \\
\langle\alpha\rangle\neg\varphi \quad &\leftrightarrow \quad \langle\alpha\rangle\top \wedge \neg\langle\alpha\rangle\varphi \\
\langle\alpha\rangle(\varphi \wedge \psi) \quad &\leftrightarrow \quad \langle\alpha\rangle\varphi \wedge \langle\alpha\rangle\psi \\
\langle\alpha\rangle\langle\beta\rangle\top \quad &\leftrightarrow \quad \langle\alpha\rangle\top \wedge \langle\beta\rangle\top
\end{aligned}
$$

$\Rightarrow$ result: Boolean combination of modal atoms:

- propositional variables
- $\langle i{:}{\pm}p\rangle\top$
- $\overline{M}\langle\langle i{:}{\pm}p\rangle\rangle\top$, where $\overline{M}$ is a sequence of $\langle\alpha\rangle$ and $\mathtt{X}$

Decision procedure (3)

reduction axioms for $\mathtt{Stit}_J$ (cf. dynamic epistemic logics):

- $\mathtt{Stit}_J(\varphi_1 \wedge \varphi_2) \leftrightarrow \mathtt{Stit}_J\varphi_1 \wedge \mathtt{Stit}_J\varphi_2$
- $\mathtt{Stit}_J(p \vee \varphi) \quad \leftrightarrow p \vee \mathtt{Stit}_J\varphi$
  $\mathtt{Stit}_J(\neg p \vee \varphi) \leftrightarrow \neg p \vee \mathtt{Stit}_J\varphi$
- $\mathtt{Stit}_J(\langle\alpha\rangle\top \vee \varphi) \quad \leftrightarrow \langle\alpha\rangle\top \vee \mathtt{Stit}_J\varphi$
  $\mathtt{Stit}_J(\neg\langle\alpha\rangle\top \vee \varphi) \leftrightarrow \neg\langle\alpha\rangle\top \vee \mathtt{Stit}_J\varphi$
- $\mathtt{Stit}_J(\overline{M}\langle\!\langle i, \pm p\rangle\!\rangle\top \vee \varphi) \;\; \leftrightarrow \overline{M}\langle\!\langle i, \pm p\rangle\!\rangle\top \vee \mathtt{Stit}_J\varphi$ if $i \in J$
  $\mathtt{Stit}_J(\neg\overline{M}\langle\!\langle i, \pm p\rangle\!\rangle\top \vee \varphi) \leftrightarrow \neg\overline{M}\langle\alpha_n\rangle\langle\!\langle i, \pm p\rangle\!\rangle\top \vee \mathtt{Stit}_J\varphi$ if $i \in J$
- Let P and Q be two finite sets of modal atoms that are all of the form $\overline{M}\langle\!\langle i, \pm p\rangle\!\rangle\top$ with $i \notin J$. Then

$$\mathtt{Stit}_J\Big(\big(\bigvee P\big) \vee \neg\big(\bigwedge Q\big)\Big) \leftrightarrow \begin{cases} \top & \text{if } P \cap Q \neq \emptyset \\ \neg \bigwedge_{\overline{M}\langle\!\langle i,\pm p\rangle\!\rangle\top \in Q} \langle i, \pm p\rangle\top & \text{if } P \cap Q = \emptyset \end{cases}$$

Decision procedure (4)

given a DL-PC formula φ:

1. take some innermost $\mathtt{Stit}_J\psi$
2. transform ψ into a Boolean combination of modal atoms
3. eliminate $\mathtt{Stit}_J$
4. iterate until no more agency operators $\mathtt{Stit}_J$
   $\Rightarrow$ result: φ' = Boolean combination of modal atoms
5. call a SAT solver for $\varphi' \wedge (\bigwedge \Gamma_{\varphi'})$, where modal atoms are viewed as propositional variables and where

$$\Gamma_{\varphi'} = \{\overline{M}\langle\langle i{:}\pm p\rangle\rangle\top \to \langle i{:}\pm p\rangle\top \ : \ \overline{M}\langle\langle i{:}\pm p\rangle\rangle\top \text{ is a modal atom of } \varphi'\}$$

Outline

Adding knowledge to DL-PC: models

- $M = (W, \{\approx_i\}_{i \in \mathbb{A}}, \{\mathrm{Val}_w\}_{w \in W}, \{\mathrm{Rep}_w\}_{w \in W}, \{\mathrm{Succ}_{w \in W}\}_w)$ where
  - W set of possible worlds
  - $\approx_i \subseteq W \times W$, equivalence relation
  - $\mathrm{Val}_w \subseteq \mathbb{P}$
  - $\mathrm{Rep}_w \subseteq \mathcal{JA}$
  - $\mathrm{Succ}_w : \mathcal{JA}^* \longrightarrow \mathcal{JA}$ s.th. $\mathrm{Succ}_w(\sigma) \subseteq \mathrm{Rep}$ for all $\sigma \in \mathcal{JA}$

- constraints:
  - $\mathrm{Succ}_w(\sigma) \subseteq \mathrm{Rep}_w$, for all σ
  - if $w \approx_i w'$ then $(\mathrm{Rep}_w)_i = (\mathrm{Rep}_{w'})_i$
  - if $w \approx_i w'$ then $(\mathrm{Succ}_w(\sigma))_i = (\mathrm{Succ}_{w'}(\sigma))_i$, for all σ
    $\Rightarrow$ will be valid:
    - $\langle \alpha \rangle \top \rightarrow \mathrm{K}_i \langle \alpha_i \rangle \top$
    - $\langle\langle \alpha \rangle\rangle \top \rightarrow \mathrm{K}_i \langle\langle \alpha_i \rangle\rangle \top$

Adding knowledge to DL-PC: truth conditions

$M, w \models \langle\!\langle \alpha \rangle\!\rangle \varphi$	iff	$\alpha \subseteq \mathrm{Succ}_w(\mathrm{nil})$ and $M^{\langle\!\langle \alpha \rangle\!\rangle \top}, w \models \varphi$
$M, w \models \langle \alpha \rangle \varphi$	iff	$\alpha \subseteq \mathrm{Rep}_w$ and $M^{\langle \alpha \rangle \top}, w \models \varphi$
$M, w \models \mathtt{Stit}_J \varphi$	iff	$M', w \models \varphi$ for every M' such that $M \sim_J M'$
$M, w \models \mathtt{K}_i \varphi$	iff	$M, w' \models \varphi$ for every w' s.th. $w \approx_i w'$

update = announcement of executability/execution of α:

- $W^{\langle \alpha \rangle \top} = \{w \in W \ : \ \alpha \subseteq \mathrm{Rep}_w\}$
- $W^{\langle\!\langle \alpha \rangle\!\rangle \top} = \{w \in W \ : \ \alpha \subseteq \mathrm{Succ}_w(\mathrm{nil})\}$

⇒ Dynamic Epistemic Logic of Propositional Control (DEL-PC)

Outline

Uniform strategies

- STIT plus knowledge better suited than ATEL
  [Herzig&Troquard 2006, Broersen et al. 2009, Herzig&Lorini 2010]
- $\Box\varphi \overset{\text{def}}{=} \mathtt{Stit}_{\emptyset}\varphi$ (historic necessity)
  $\Diamond\varphi \overset{\text{def}}{=} \neg\Box\neg\varphi$ (historic possibility)
- *i* knows that he can ensure φ:

$$\mathtt{K}_i\Diamond\mathtt{Stit}_i\varphi$$

- *i* knows how to ensure φ:

$$\Diamond\mathtt{K}_i\mathtt{Stit}_i\varphi$$

Uniform strategies: example

- hypotheses:
  - $\langle i{:}-p\rangle\top$ (i can make p false)
  - $\mathtt{K}_i p$ (i knows that p)
  - $\neg\mathtt{K}_i q \wedge \neg\mathtt{K}_i\neg q$ (i uncertain about q)
- valid in DL-PC:
  - i knows that he can ensure that $p \leftrightarrow q$
    - $\models$ *Hypotheses* $\rightarrow \mathtt{K}_i\Diamond\mathtt{Stit}_i\mathtt{X}(p \leftrightarrow q)$
  - ...but i does not know how to ensure that $p \leftrightarrow q$
    - $\models$ *Hypotheses* $\rightarrow \neg\Diamond\mathtt{K}_i\mathtt{Stit}_i\mathtt{X}(p \leftrightarrow q)$

Conclusion

- DL-PC = PDL with assignments as the only atomic programs
  - complete axiomatisation
  - decidable (≠ group stit [Herzig&Schwarzentruber])
  - with program operators:
    - Kleene star can be eliminated
    - SAT complexity: ExpTime complete
- DEL-PC = DL-PC plus epistemic operator
  - agents know what they are going to play
  - allows to reason about uniform strategies
  - t.b.d.: decidability & complexity of epistemic extension
- good basis for a logic of agent interaction
  - more elaborate account of constitutive rules: brute facts, institutional facts, roles [Herzig et al., CLIMA 2011]
  - social simulation [Gaudou et al. MABS 2011]