Dynamic Coalition Logic

Granting and Revoking Dictatorial Powers

Rustam Galimullin[1] and Thomas Ågotnes[1,2]
{rustam.galimullin, thomas.agotnes}@uib.no
1: University of Bergen, Norway
2: Southwest University, China

Coalition logic

Coalition logic

- Coalition logic (CL) [Pauly, 2002] is used to reason about abilities of groups of agents in the presence of opponents

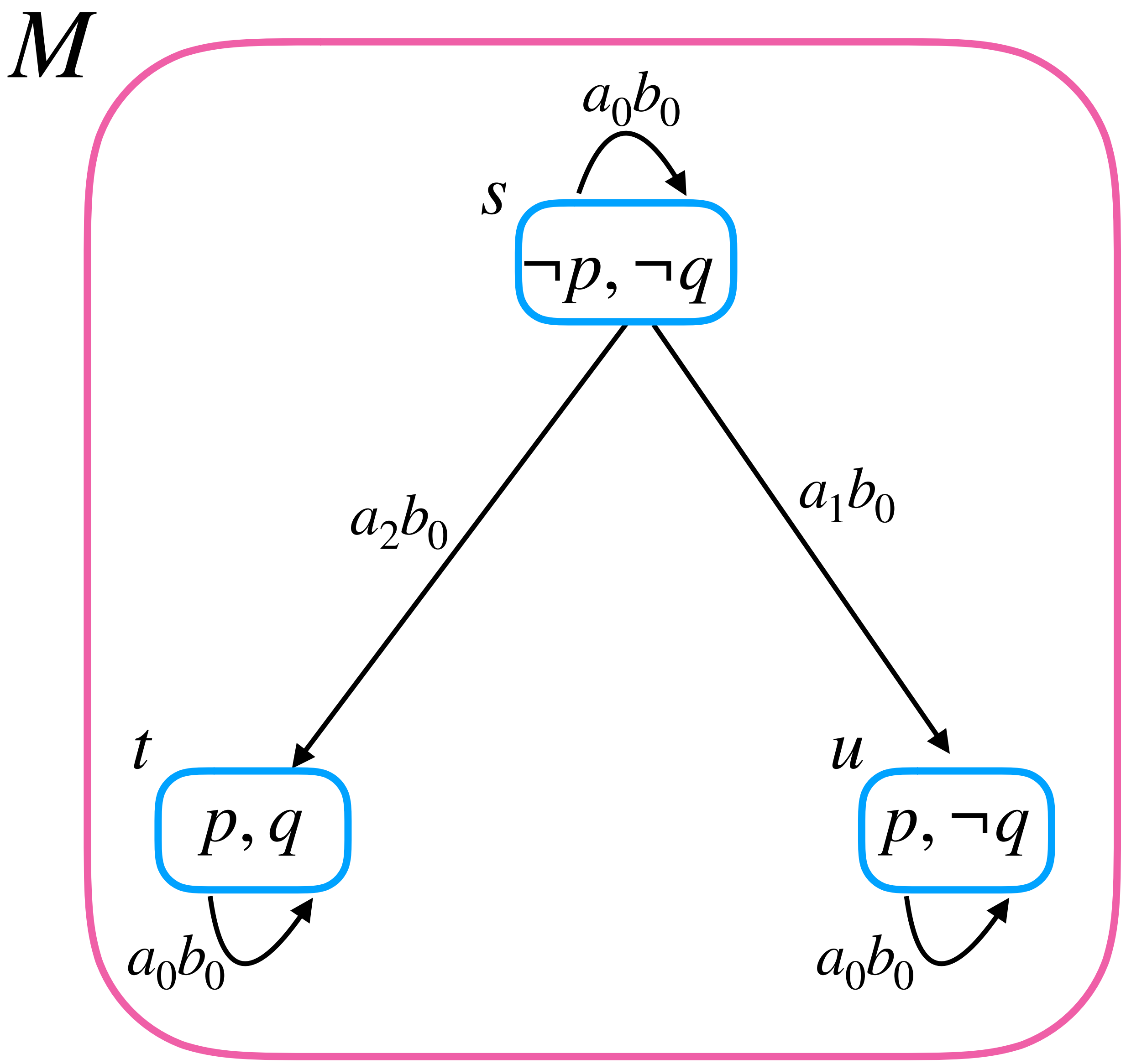
Coalition logic

- Coalition logic (CL) [Pauly, 2002] is used to reason about abilities of groups of agents in the presence of opponents
- Language of CL: $\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\!\langle C \rangle\!\rangle \varphi$

Coalition logic

- Coalition logic (CL) [Pauly, 2002] is used to reason about abilities of groups of agents in the presence of opponents
- Language of CL: $\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle\varphi$
- $\langle\langle C \rangle\rangle\varphi$ is read as ‘coalition C can bring about φ by a joint action no matter what agents outside of the coalition do.’

Coffee scenario



p: agent a has a cup of coffee
q: agent b has a cup of coffee

$M_s \models \neg p \wedge \neg q$
$M_s \models \langle\langle a \rangle\rangle (p \wedge \neg q)$
$M_s \not\models [[b]]q$

Models

A concurrent game model (CGM) is a tuple $M = (A, S, Act, act, out, L)$, where

- A is a non-empty finite set of agents;
- S is a non-empty set of states;
- Act is a non-empty set of actions;
- act assigns to each agent and each state a non-empty set of actions;
- out assigns to each state and each combination of actions available to agents a unique outcome state;
- L is the valuation function.

We will denote by α_C a set of actions such that for each $i \in C$ there is exactly one action of i in α_C.

Semantics

The semantics of $\langle\langle C \rangle\rangle \varphi$ is

$M_s \models \langle\langle C \rangle\rangle \varphi$ **iff** $\exists \alpha_C, \forall \alpha_{\overline{C}} : M_t \models \varphi,$ **where** $t = out(s, \alpha_C \cup \alpha_{\overline{C}})$

The semantics of $[\![C]\!] \varphi$ is

$M_s \models [\![C]\!] \varphi$ **iff** $\forall \alpha_C, \exists \alpha_{\overline{C}} : M_t \models \varphi,$ **where** $t = out(s, \alpha_C \cup \alpha_{\overline{C}})$

Dynamic coalition logic

Dynamic coalition logic

- We can consider a CGM as a policy that specifies abilities of coalitions and individual agents

Dynamic coalition logic

- We can consider a CGM as a policy that specifies abilities of coalitions and individual agents
- CL, however, cannot capture updates of such a policy

Dynamic coalition logic

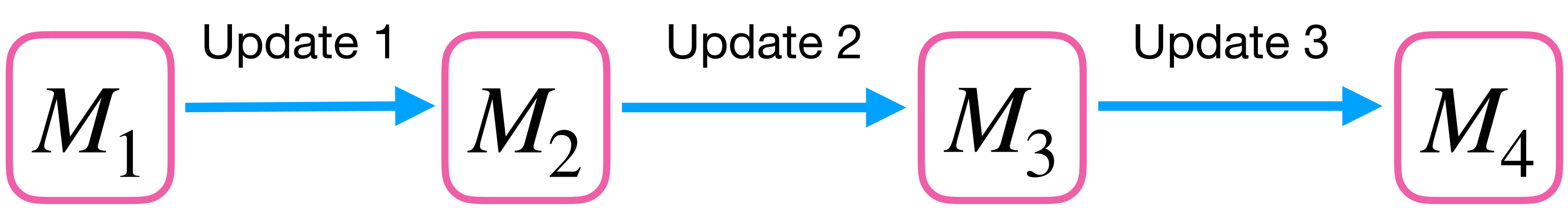
- We can consider a CGM as a policy that specifies abilities of coalitions and individual agents

- CL, however, cannot capture updates of such a policy

- Thus, we propose a study of dynamic coalition logic

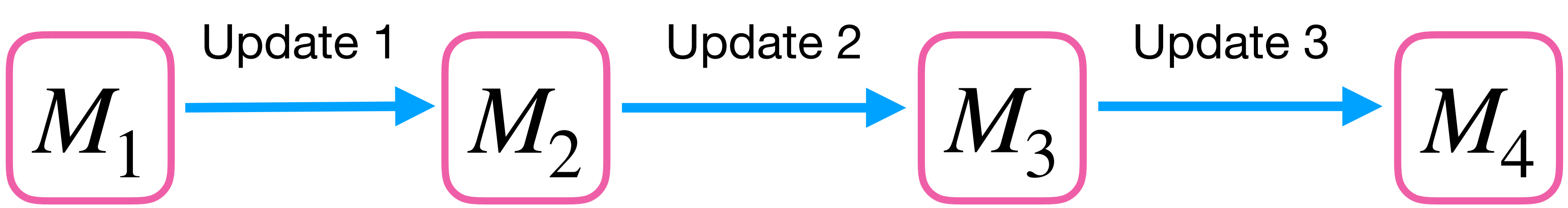
Dynamic coalition logic

- We can consider a CGM as a policy that specifies abilities of coalitions and individual agents
- CL, however, cannot capture updates of such a policy
- Thus, we propose a study of dynamic coalition logic



Dynamic coalition logic

- We can consider a CGM as a policy that specifies abilities of coalitions and individual agents
- CL, however, cannot capture updates of such a policy
- Thus, we propose a study of dynamic coalition logic



Inspired by the dynamics of changes of smart contracts on a blockchain [Herlihy and Moir, 2016]

Dictatorial powers

Dictatorial powers

- Our first step is considering granting and revoking dictatorial powers to single agents

Dictatorial powers

- Our first step is considering granting and revoking dictatorial powers to single agents
- We will update CGMs by adding or removing arrows

Dictatorial powers

- Our first step is considering granting and revoking dictatorial powers to single agents
- We will update CGMs by adding or removing arrows
- We borrow syntax from arrow update logic [Kooi and Renne, 2011]

Dictatorial powers

- Our first step is considering granting and revoking dictatorial powers to single agents
- We will update CGMs by adding or removing arrows
- We borrow syntax from arrow update logic [Kooi and Renne, 2011]
- $(\varphi, a, \psi)^+$ stands for '*grant agent* a *the power to force* ψ*-states from any* φ*-state*'

Dictatorial powers

- Our first step is considering granting and revoking dictatorial powers to single agents
- We will update CGMs by adding or removing arrows
- We borrow syntax from arrow update logic [Kooi and Renne, 2011]
- $(\varphi, a, \psi)^+$ stands for '*grant agent a the power to force ψ-states from any φ-state*'
- $(\varphi, a, \psi)^-$ stands for '*preserve a's power to force ψ-states from any φ-state*'

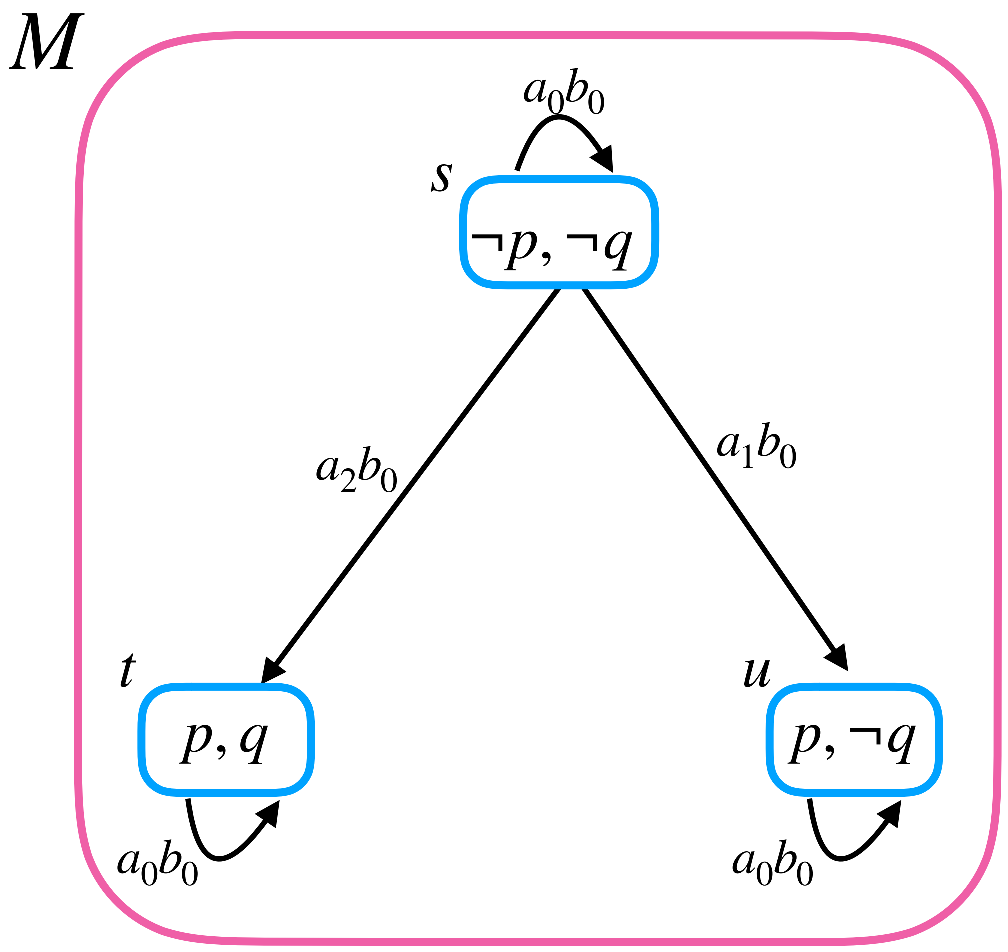
Granting dictatorial powers

The language of positive dictatorial dynamic coalition logic ($DDCL^+$) is

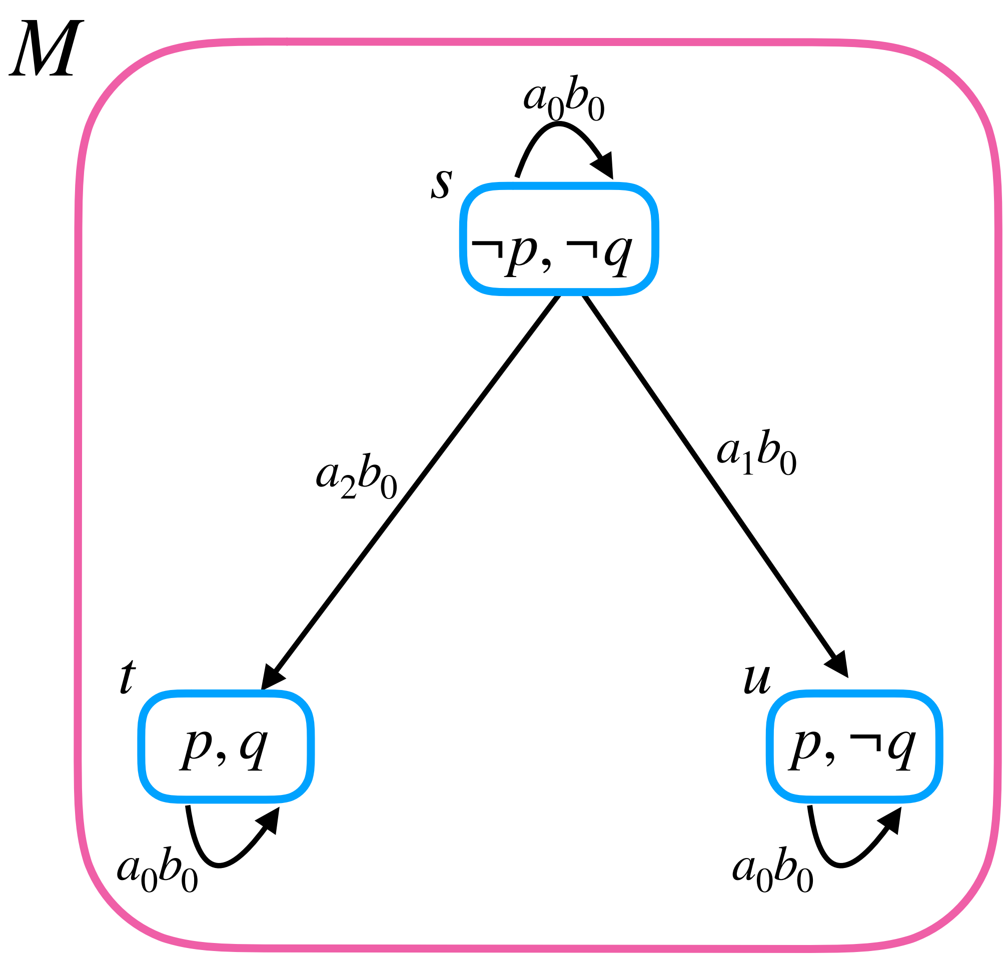
$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle\varphi \mid [+U]\varphi$$
$$+U ::= (\varphi, a, \varphi)^+ \mid (\varphi, a, \varphi)^+, +U$$

where $+U$ is called a positive update

Coffee scenario

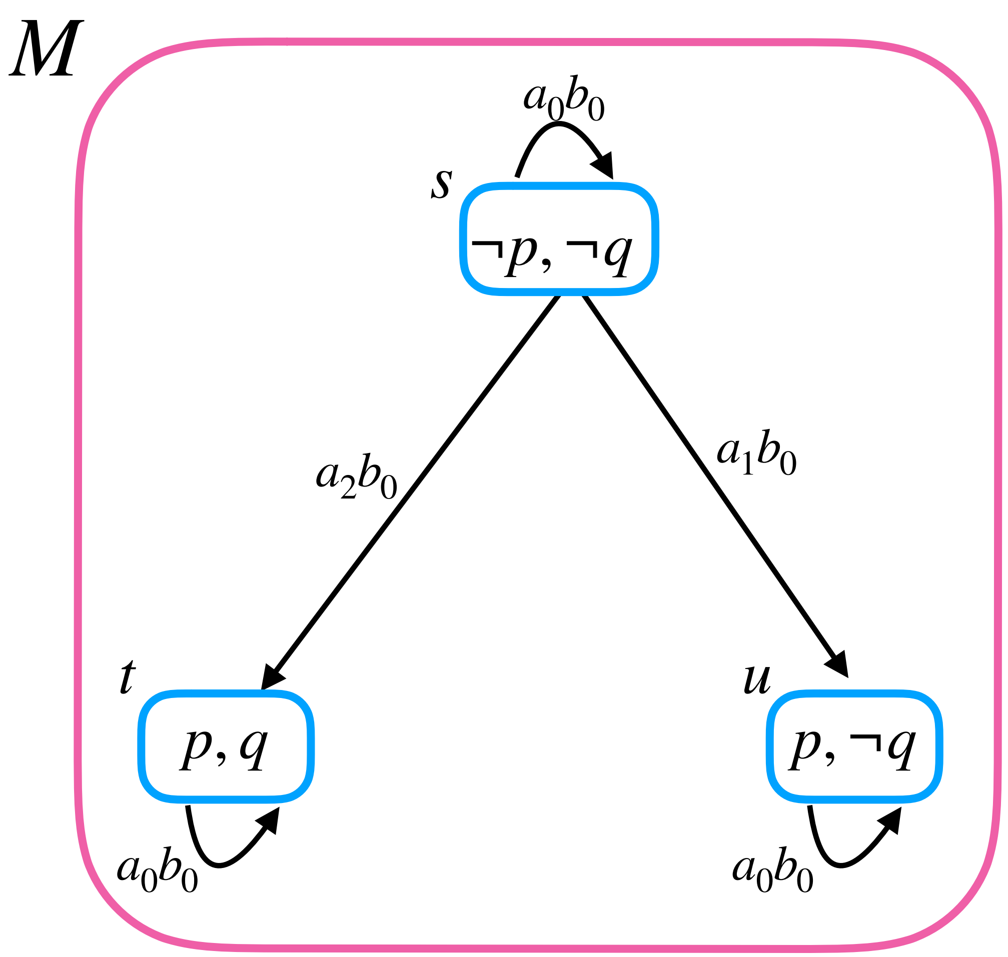


Coffee scenario



New policy in the office: b can get a cup of coffee whenever she does not have one (and a cannot preclude her from doing this)

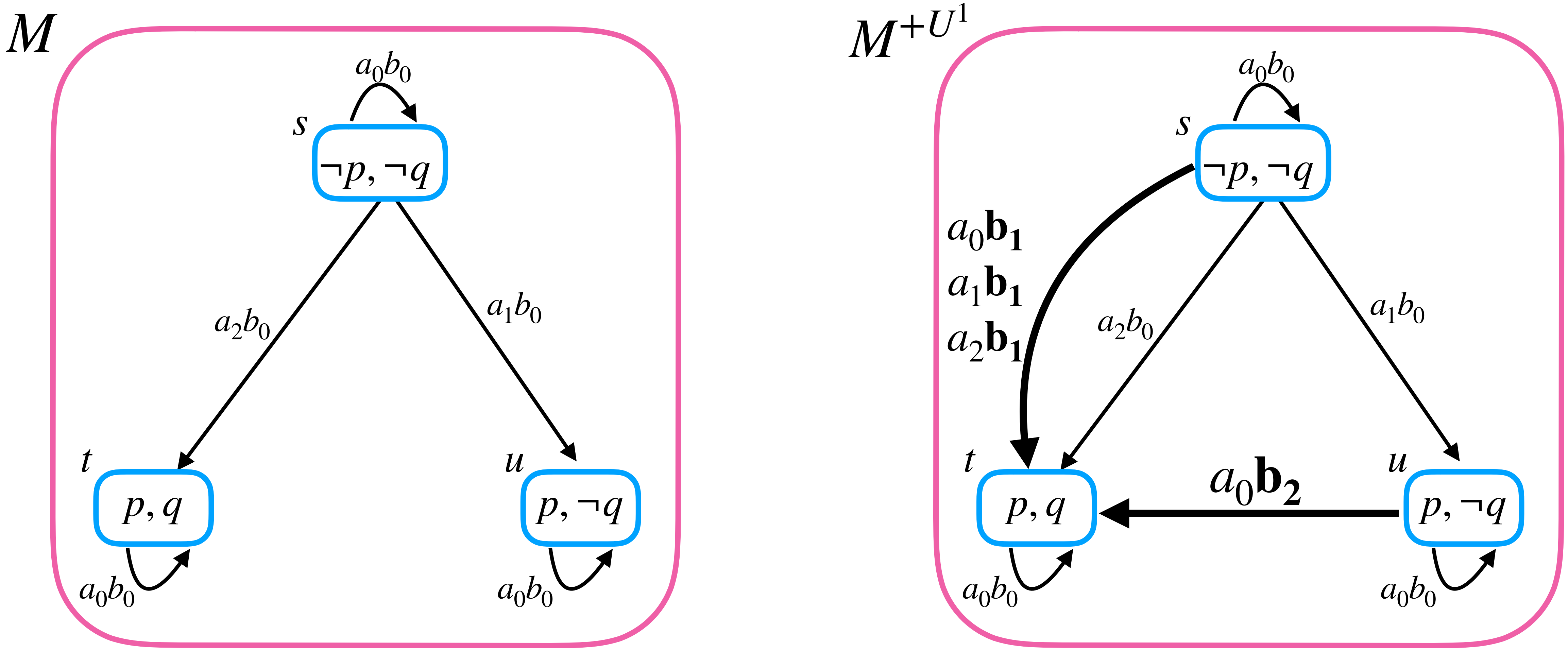
Coffee scenario



New policy in the office: b can get a cup of coffee whenever she does not have one (and a cannot preclude her from doing this)

$$+U^1 = \{(\neg q, b, q)^+\}$$

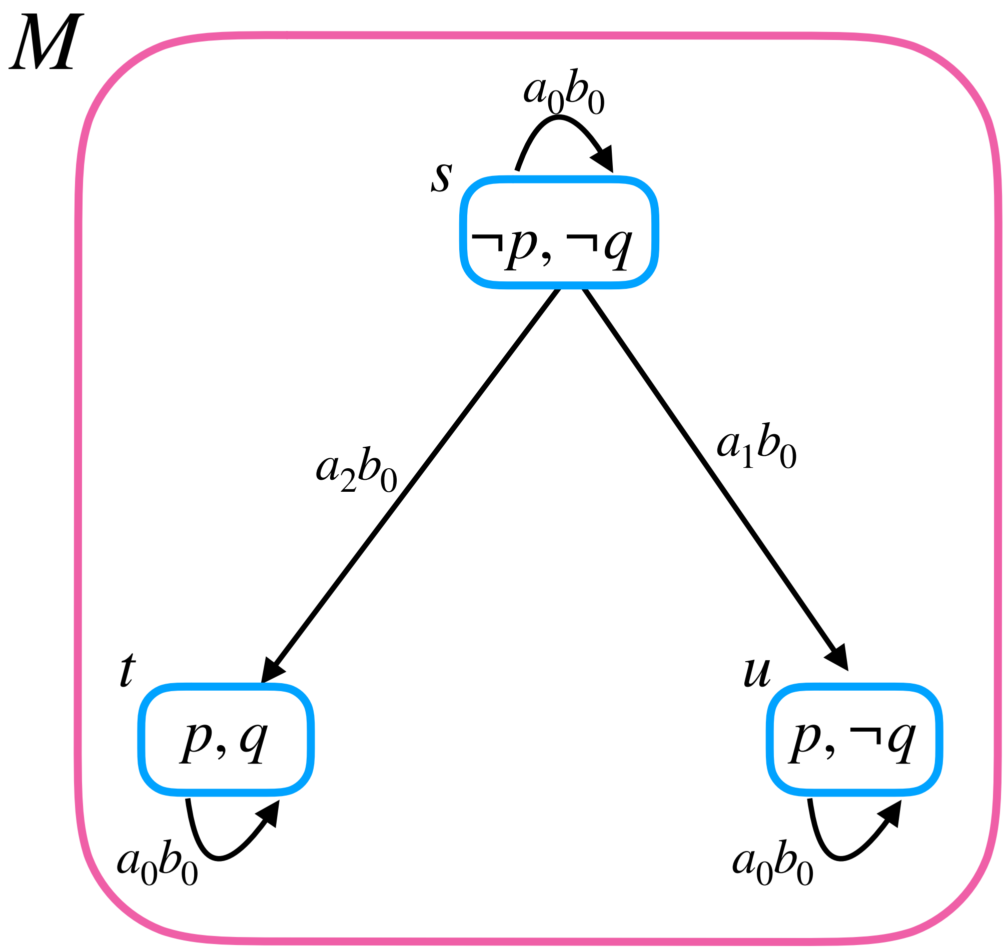
Coffee scenario



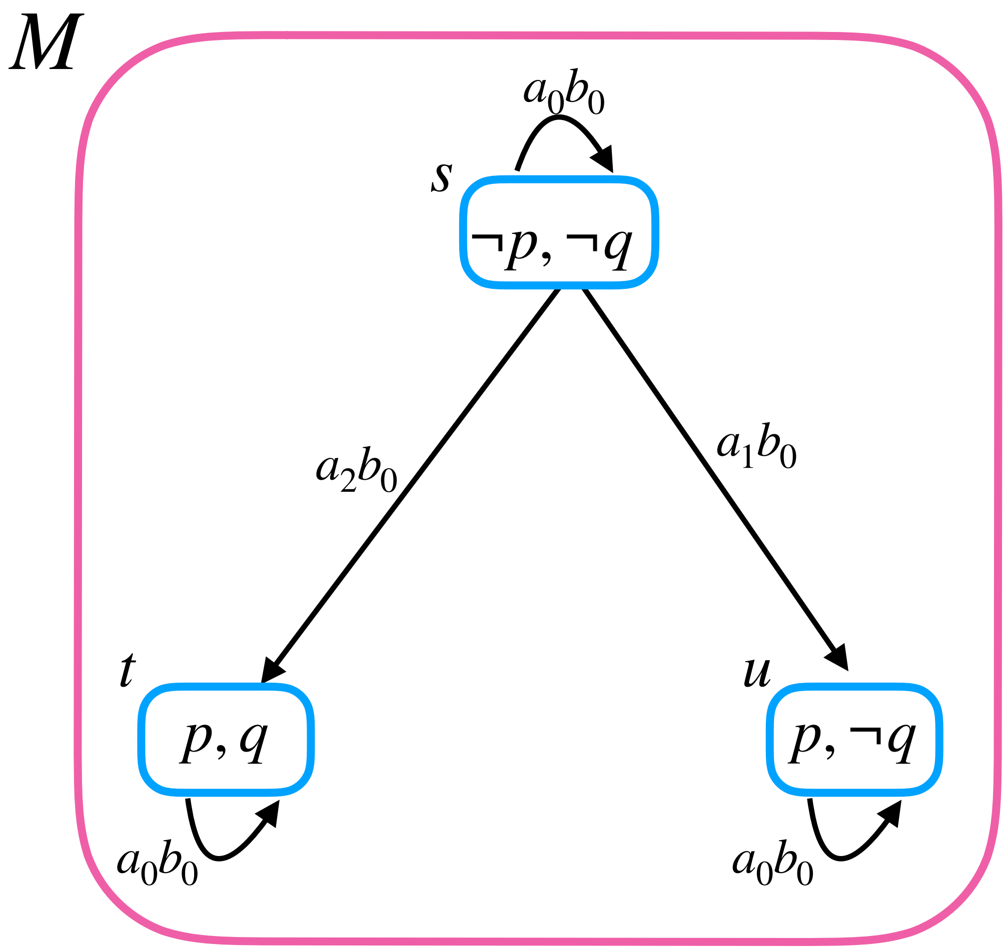
New policy in the office: b can get a cup of coffee whenever she does not have one (and a cannot preclude her from doing this)

$$+U^1 = \{(\neg q, b, q)^+\}$$

Coffee scenario

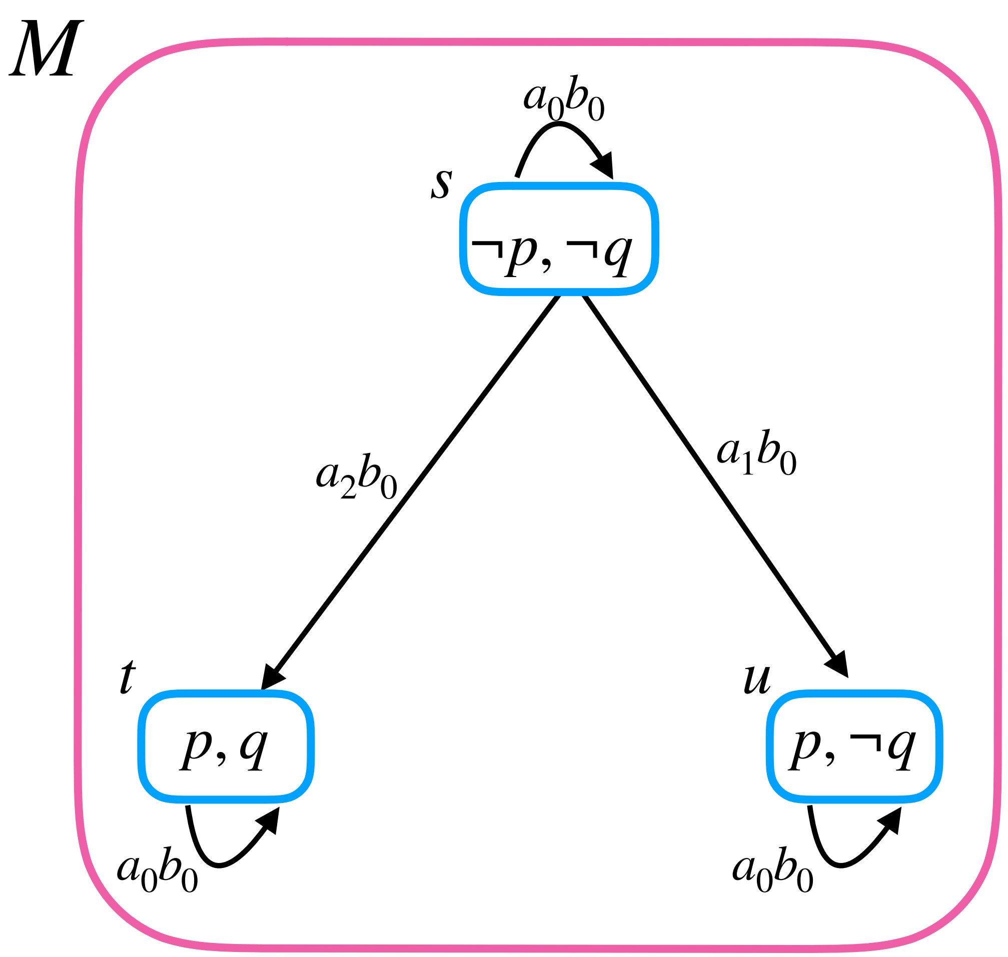


Coffee scenario



Not every update can be implemented

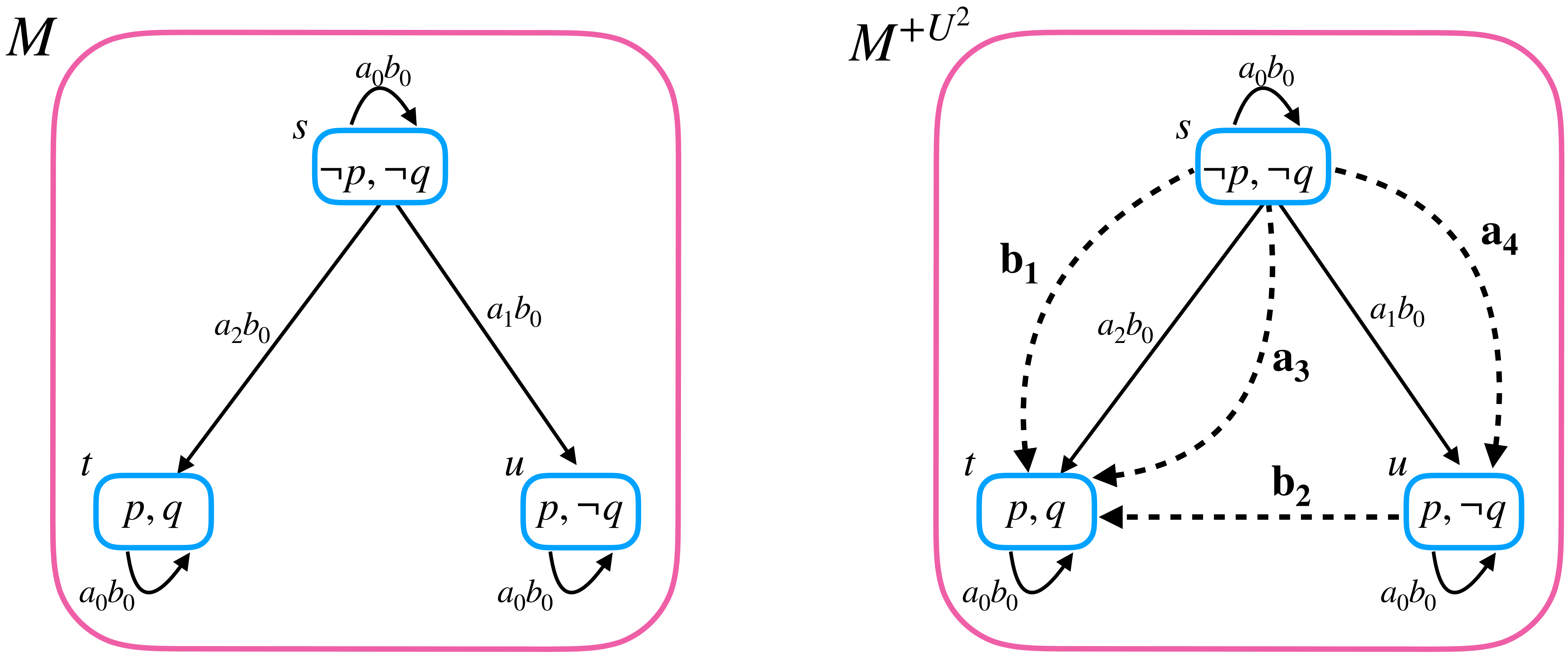
Coffee scenario



Not every update can be implemented

$$+U^2 = \{(\neg q, b, q)^+, (\neg p, a, p)^+\}$$

Coffee scenario



Not every update can be implemented

$$+U^2 = \{(\neg q, b, q)^+, (\neg p, a, p)^+\}$$

Semantics of DDCL^{+}

Semantics of DDCL+

We call update $+U$ executable in model M iff for all $(\varphi, i, \psi)^+$, $(\chi, j, \tau)^+ \in +U$: $\|\varphi\|_M \cap \|\chi\|_M = \varnothing$ whenever $i \neq j$

Semantics of DDCL⁺

We call update $+U$ executable in model M iff for all $(\varphi, i, \psi)^+$, $(\chi, j, \tau)^+ \in +U$: $\|\varphi\|_M \cap \|\chi\|_M = \emptyset$ whenever $i \neq j$

$M_s \models [+U]\varphi$ **iff** U^+ **is executable in** M **implies** $M_s^{+U} \models \varphi$

Semantics of DDCL+

We call update $+U$ executable in model M iff for all $(\varphi, i, \psi)^+$, $(\chi, j, \tau)^+ \in +U$: $\|\varphi\|_M \cap \|\chi\|_M = \emptyset$ whenever $i \neq j$

$M_s \models [+U]\varphi$ **iff** U^+ **is executable in** M **implies** $M_s^{+U} \models \varphi$

$M_s \models \langle +U \rangle\varphi$ **iff** U^+ **is executable in** M **and** $M_s^{+U} \models \varphi$

Semantics of DDCL⁺

We call update $+U$ executable in model M iff for all $(\varphi, i, \psi)^+$, $(\chi, j, \tau)^+ \in +U$: $\|\varphi\|_M \cap \|\chi\|_M = \varnothing$ whenever $i \neq j$

$M_s \models [+U]\varphi$ **iff** U^+ **is executable in** M **implies** $M_s^{+U} \models \varphi$

$M_s \models \langle +U \rangle\varphi$ **iff** U^+ **is executable in** M **and** $M_s^{+U} \models \varphi$

Proposition:

Monotonicity: $\langle +U \rangle\varphi \wedge \langle +U \rangle\psi \leftrightarrow \langle +U \rangle(\varphi \wedge \psi)$ is valid

Union: $\langle +U^1 \rangle\varphi \wedge \langle +U^2 \rangle\varphi \rightarrow \langle +U^1 \cup +U^2 \rangle\varphi$ is not valid

Commutativity: $\langle +U^1 \rangle\langle +U^2 \rangle\varphi \rightarrow \langle +U^2 \rangle\langle +U^1 \rangle\varphi$ is not valid

Revoking dictatorial powers

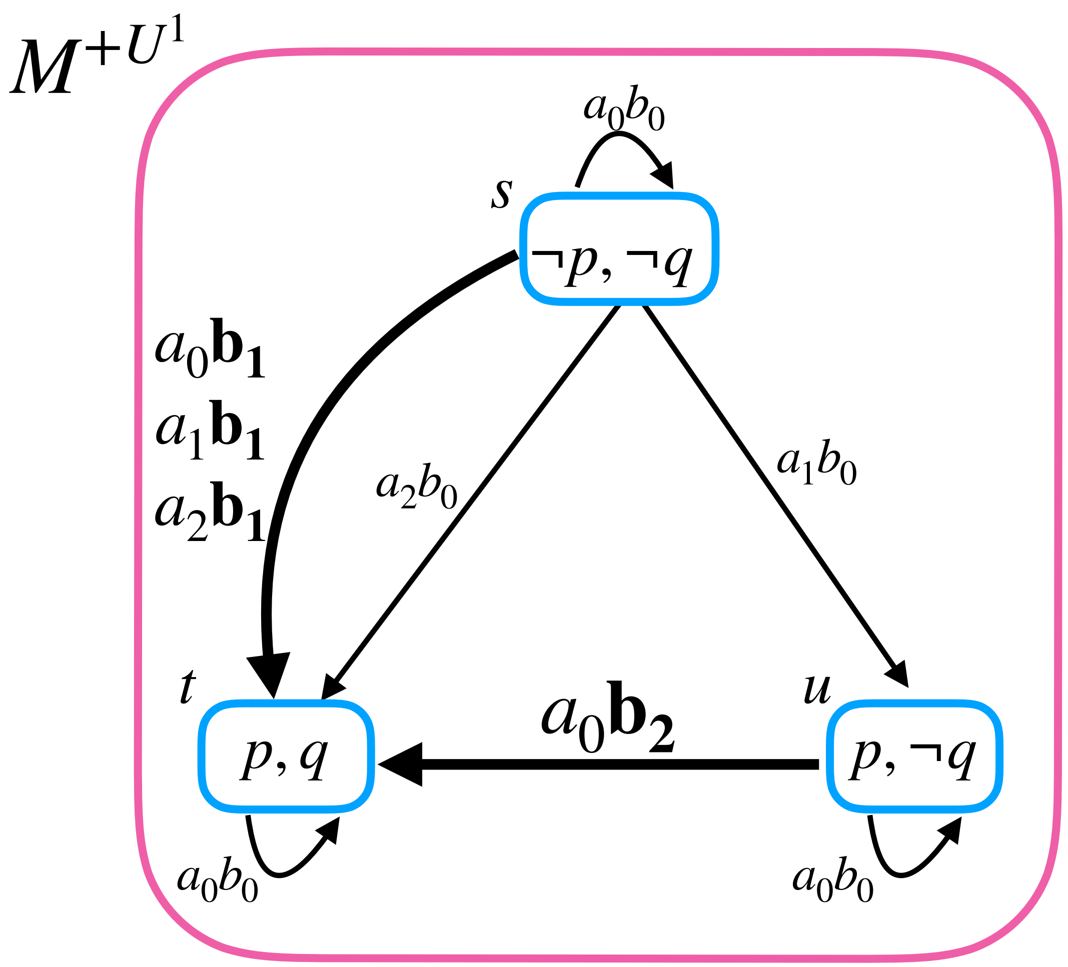
The language of negative dictatorial dynamic coalition logic (DDCL$^-$) is

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \langle\langle C \rangle\rangle \varphi \mid [-U]\varphi$$
$$-U ::= (\varphi, a, \varphi)^- \mid (\varphi, a, \varphi)^-, -U$$

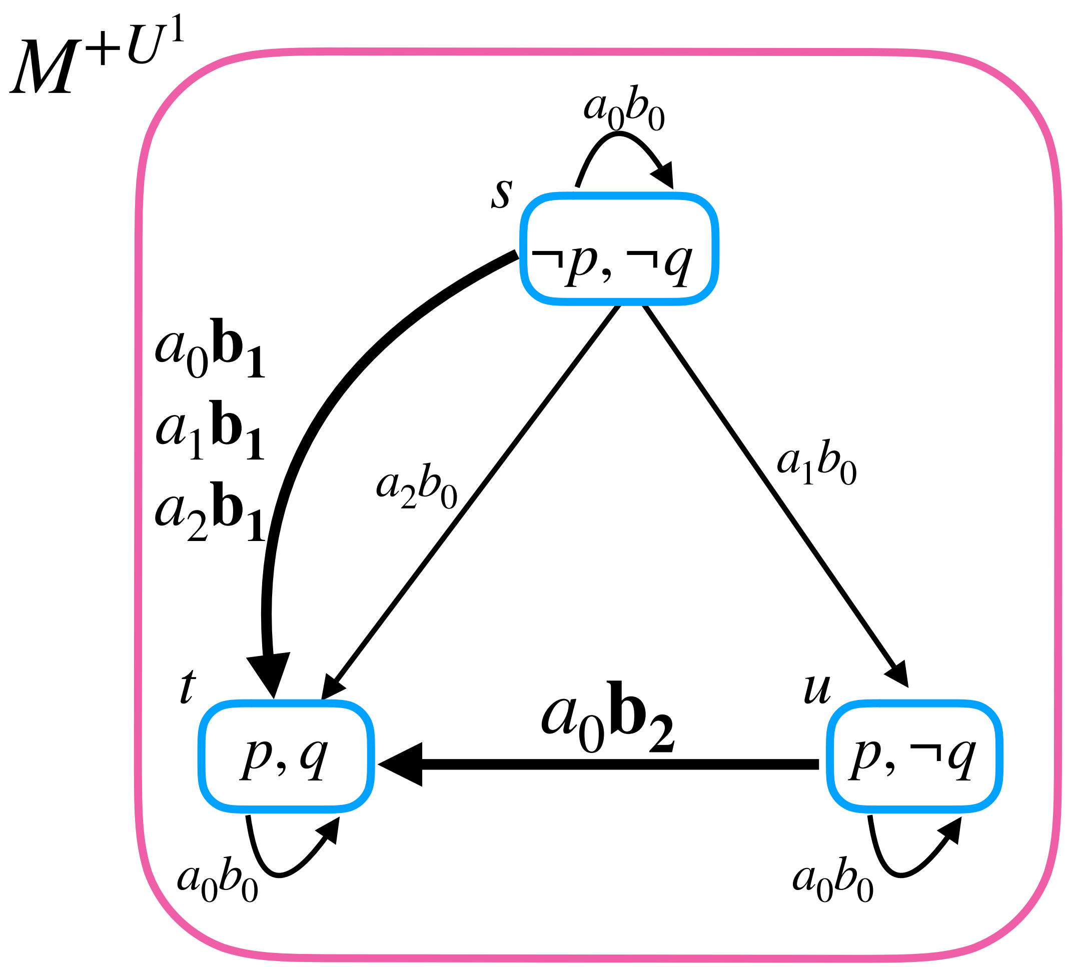
where $-U$ is called a negative update

$(\varphi, a, \psi)^-$ stands for 'preserve a's power to force ψ-states from any φ-state' (and remove all other dictatorial powers)

Coffee scenario

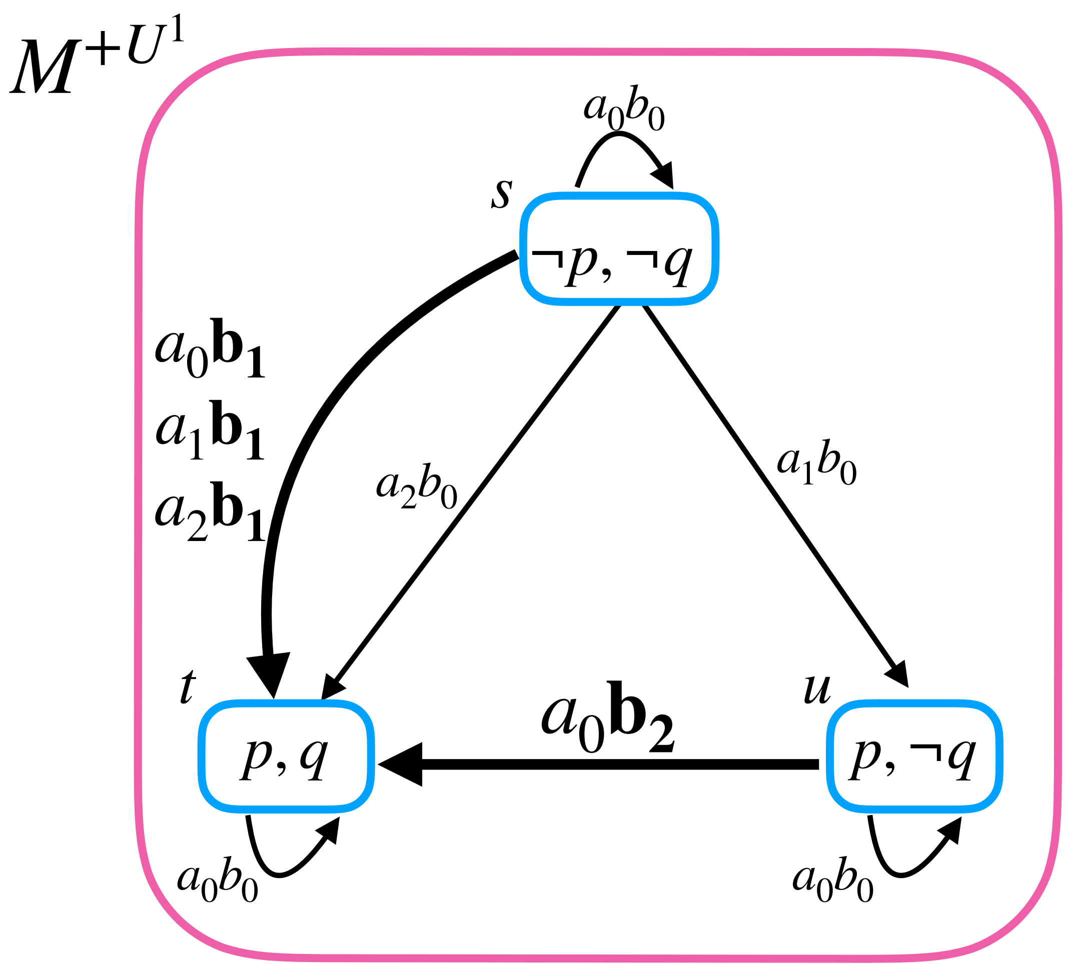


Coffee scenario



Policy correction: b can get a cup of coffee whenever both agents do not have one

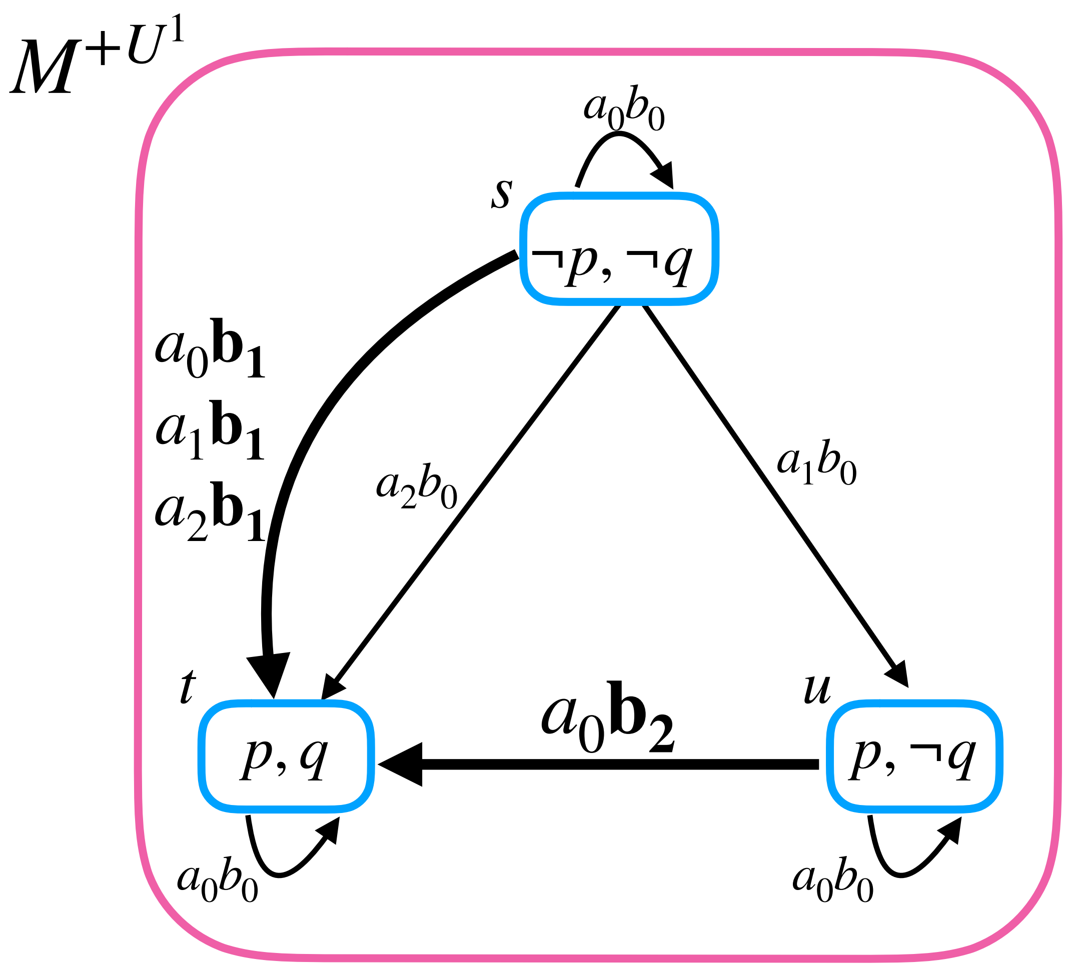
Coffee scenario

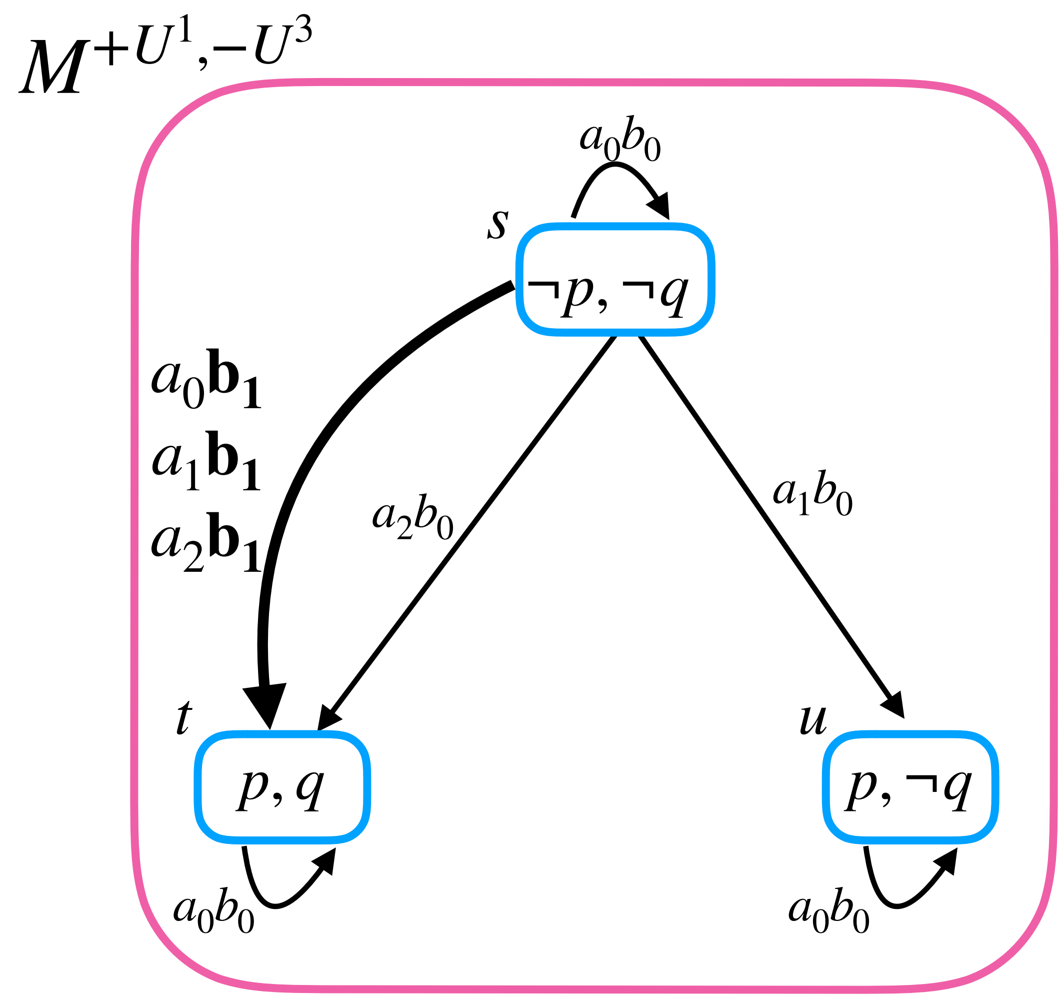


Policy correction: b can get a cup of coffee whenever both agents do not have one

$$-U^3 = \{(p, b, p)^-, (p, a, p)^-, (\neg p \land \neg q, b, p \land q)^-\}$$

Coffee scenario

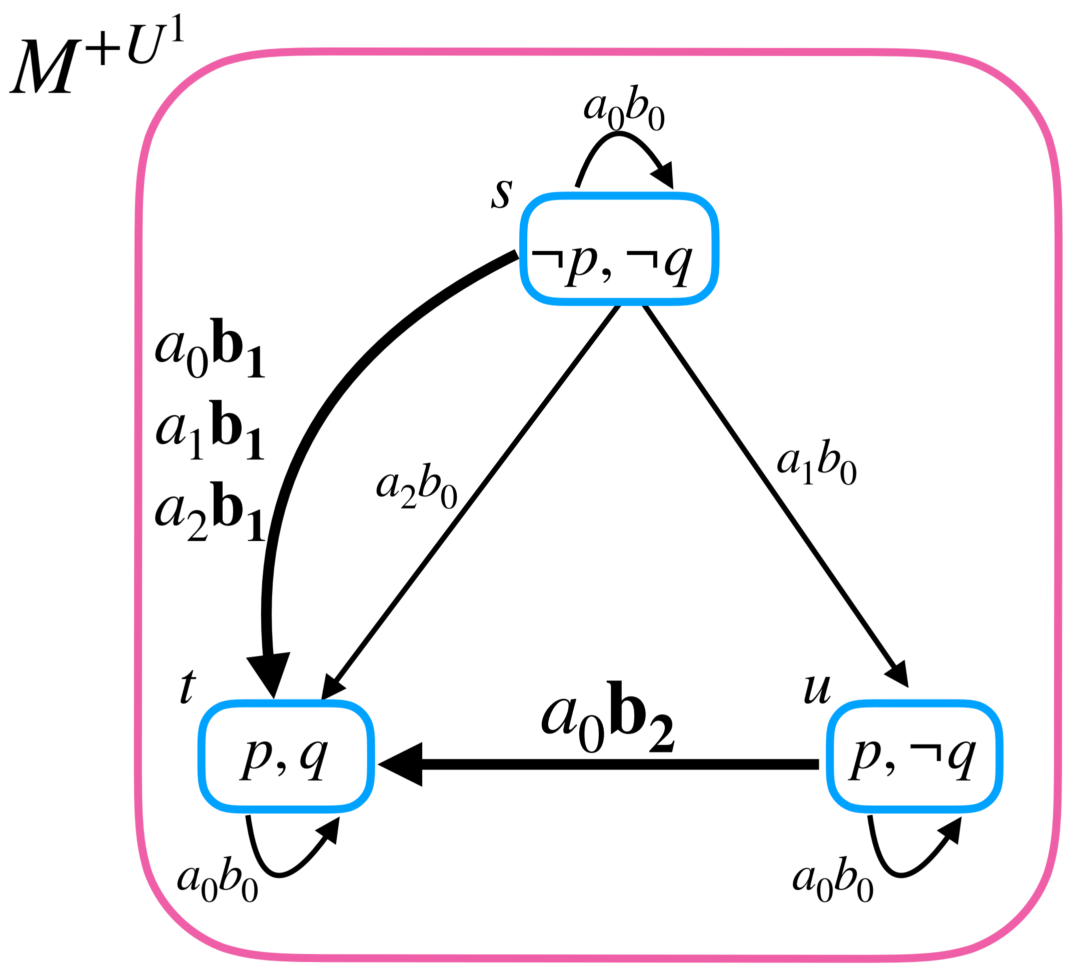




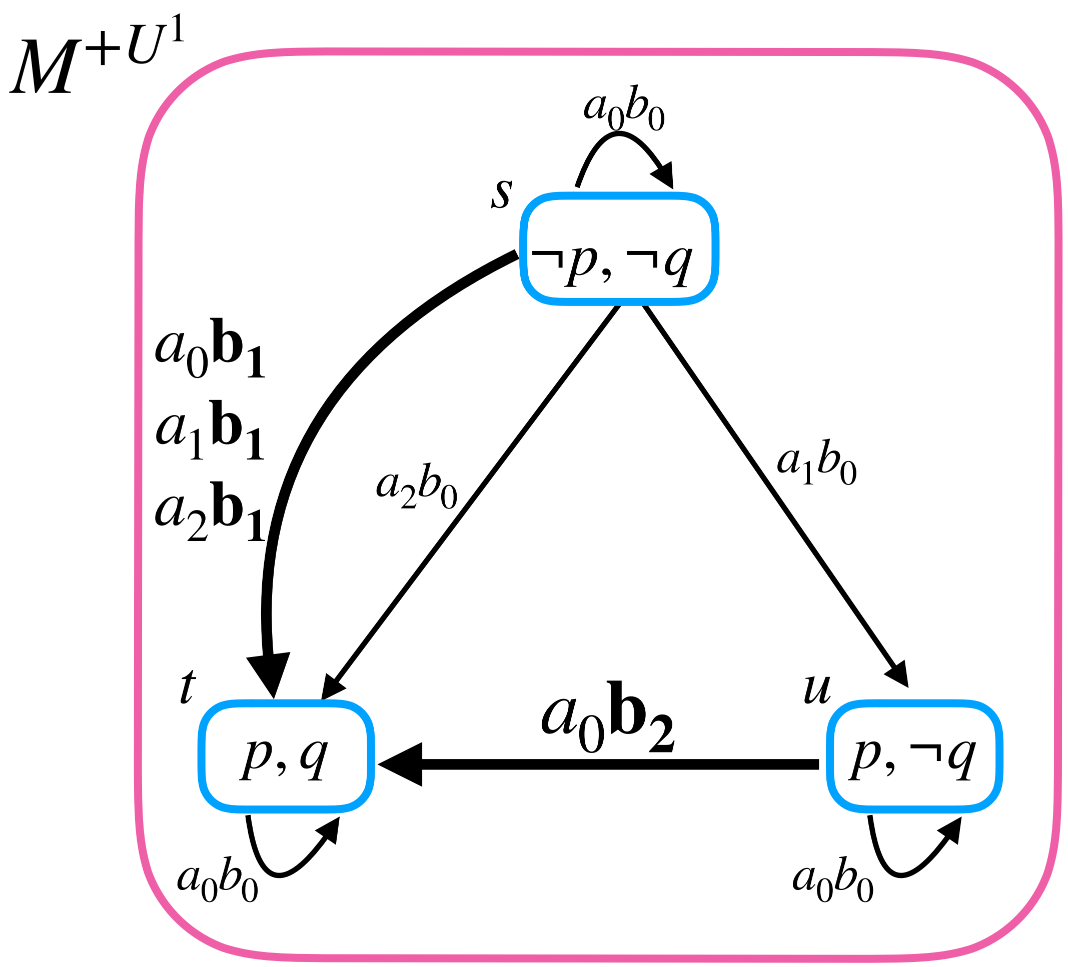
Policy correction: b can get a cup of coffee whenever both agents do not have one

$$-U^3 = \{(p, b, p)^-, (p, a, p)^-, (\neg p \wedge \neg q, b, p \wedge q)^-\}$$

Coffee scenario

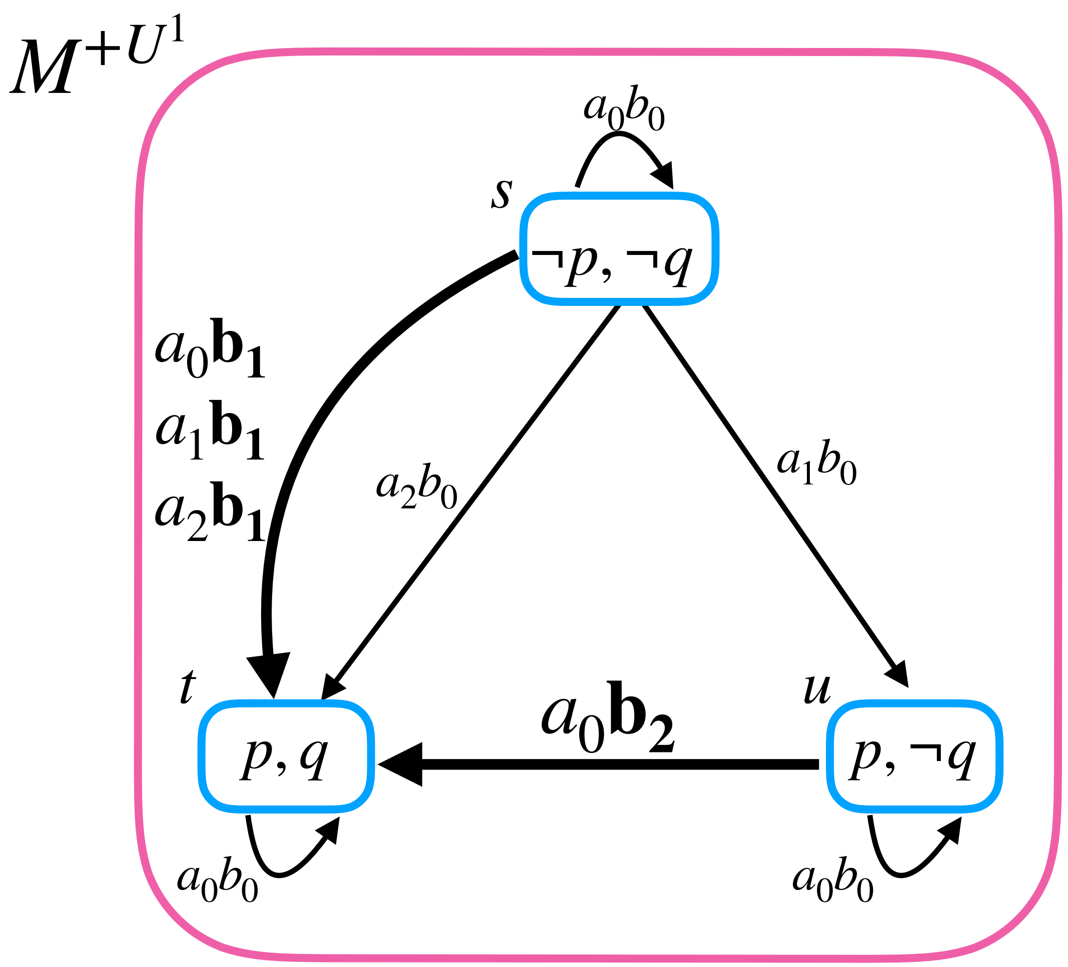


Coffee scenario



Not every update can be implemented

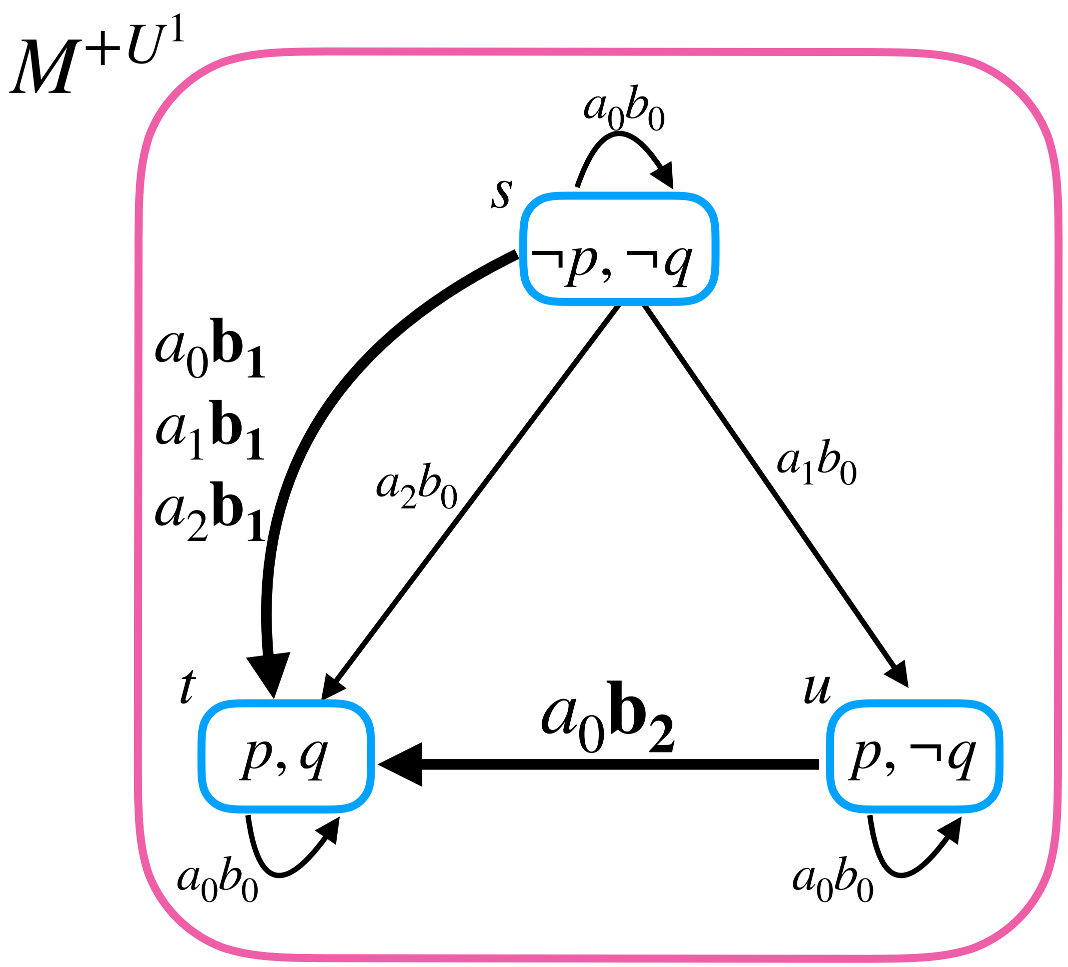
Coffee scenario



Not every update can be implemented

$$-U^4 = \{(\bot, b, \bot)^-\}$$

Coffee scenario

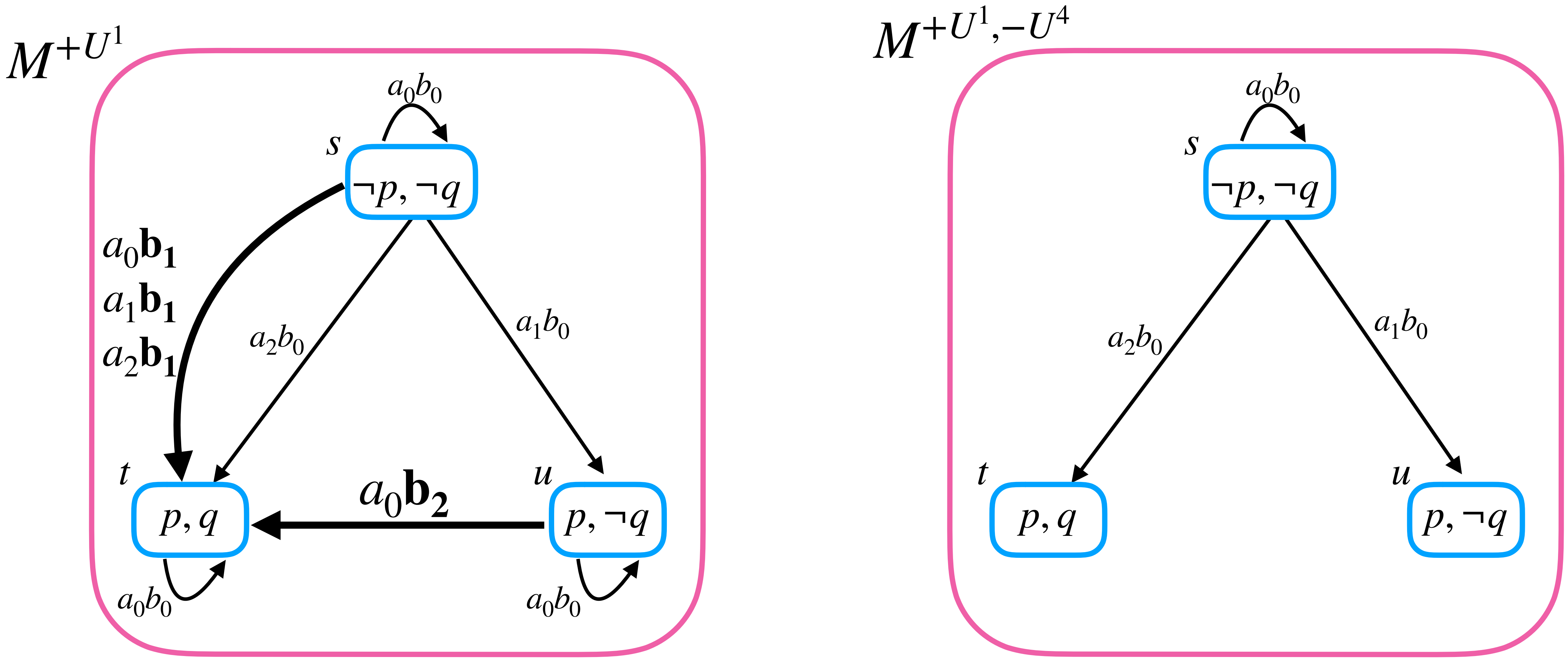


Not every update can be implemented

$$-U^4 = \{(\perp, b, \perp)^-\}$$

Remove all dictatorial (forcing) actions of all agents

Coffee scenario

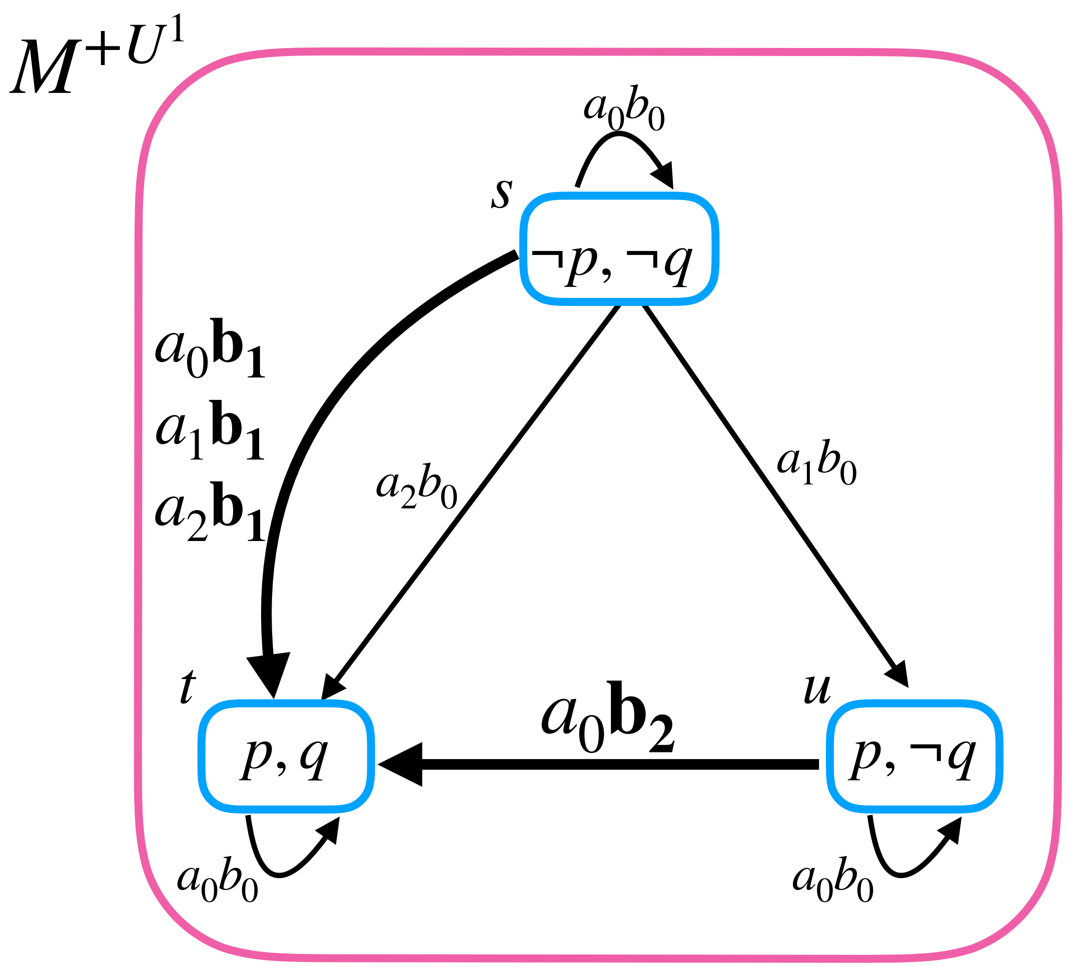


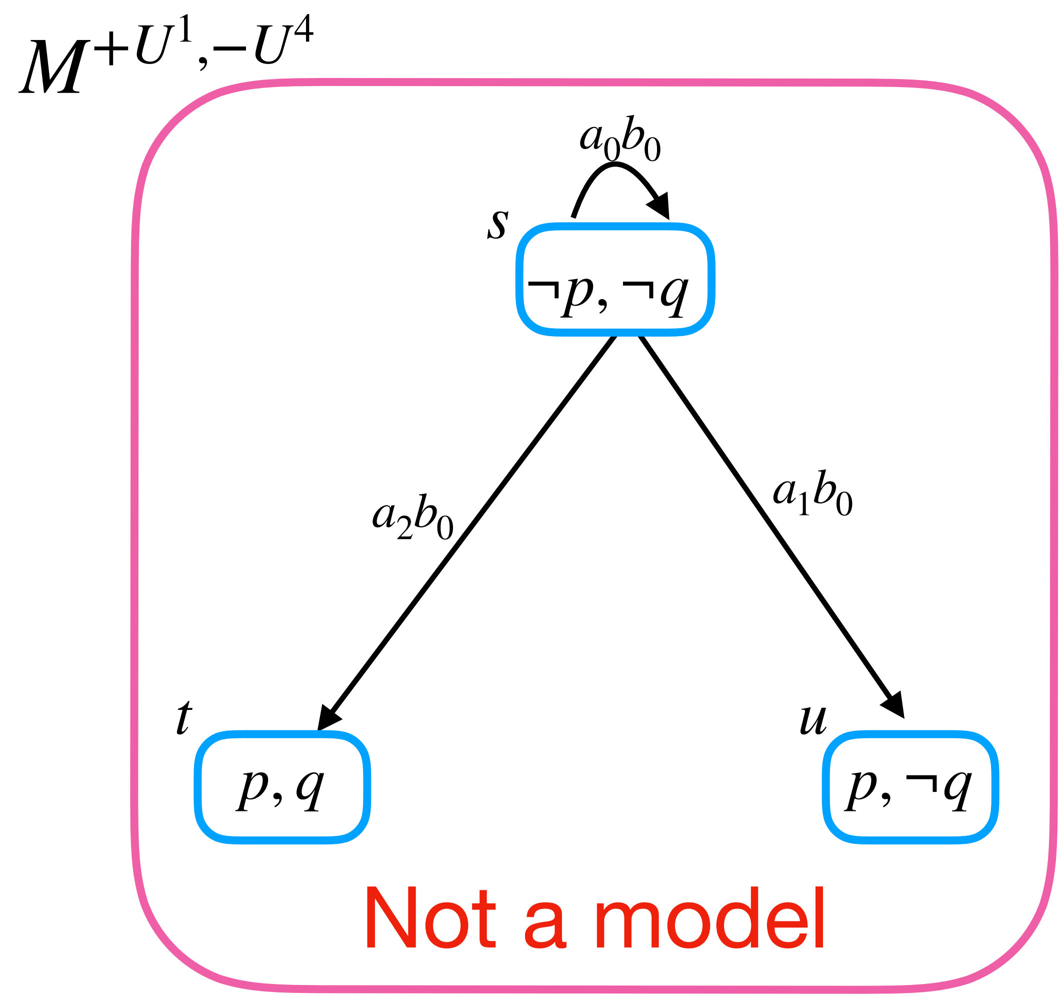
Not every update can be implemented

$$-U^4 = \{(\perp, b, \perp)^-\}$$

Remove all dictatorial (forcing) actions of all agents

Coffee scenario





Not every update can be implemented

$$-U^4 = \{(\bot, b, \bot)^-\}$$

Remove all dictatorial (forcing) actions of all agents

Semantics of DDCL$^-$

Semantics of DDCL⁻

We call update $-U$ executable in model M iff for all $i \in A$ and $s \in S$ either we preserve at least one forcing action, or there are other, non-forcing actions, or both

Semantics of DDCL$^-$

We call update $-U$ executable in model M iff for all $i \in A$ and $s \in S$ either we preserve at least one forcing action, or there are other, non-forcing actions, or both

$M_s \models [-U]\varphi$ **iff** U^- **is executable in** M **implies** $M_s^{-U} \models \varphi$

Semantics of DDCL$^-$

We call update $-U$ executable in model M iff for all $i \in A$ and $s \in S$ either we preserve at least one forcing action, or there are other, non-forcing actions, or both

$$M_s \vDash [-U]\varphi \text{ **iff** } U^- \text{ **is executable in** } M \text{ **implies** } M_s^{-U} \vDash \varphi$$

$$M_s \vDash \langle -U \rangle\varphi \text{ **iff** } U^- \text{ **is executable in** } M \text{ **and** } M_s^{-U} \vDash \varphi$$

Semantics of DDCL⁻

We call update $-U$ executable in model M iff for all $i \in A$ and $s \in S$ either we preserve at least one forcing action, or there are other, non-forcing actions, or both

$$M_s \vDash [-U]\varphi \text{ **iff** } U^- \text{ **is executable in** } M \text{ **implies** } M_s^{-U} \vDash \varphi$$

$$M_s \vDash \langle -U \rangle\varphi \text{ **iff** } U^- \text{ **is executable in** } M \text{ **and** } M_s^{-U} \vDash \varphi$$

Proposition:

Monotonicity: $\langle -U \rangle\varphi \wedge \langle -U \rangle\psi \leftrightarrow \langle -U \rangle(\varphi \wedge \psi)$ is valid

Union: $\langle -U^1 \rangle\varphi \wedge \langle -U^2 \rangle\varphi \rightarrow \langle -U^1 \cup -U^2 \rangle\varphi$ is not valid

Commutativity: $\langle -U^1 \rangle\langle -U^2 \rangle\varphi \rightarrow \langle -U^2 \rangle\langle -U^1 \rangle\varphi$ is not valid

Expressivity

Expressivity

- Formulas φ and ψ are equivalent if for all M_s, $M_s \models \varphi$ iff $M_s \models \psi$

- Language $\mathscr{L}_1$ is at least as expressive as $\mathscr{L}_2$ ($\mathscr{L}_2 \leqslant \mathscr{L}_1$) if for all $\varphi \in \mathscr{L}_2$ there is an equivalent $\psi \in \mathscr{L}_1$. If $\mathscr{L}_2 \leqslant \mathscr{L}_1$ and $\mathscr{L}_1 \nleqslant \mathscr{L}_2$ we say that $\mathscr{L}_1$ is more expressive than $\mathscr{L}_2$. If $\mathscr{L}_2 \nleqslant \mathscr{L}_1$ and $\mathscr{L}_1 \nleqslant \mathscr{L}_2$, then $\mathscr{L}_1$ and $\mathscr{L}_2$ are incomparable

Expressivity

- Formulas φ and ψ are equivalent if for all M_s, $M_s \models \varphi$ iff $M_s \models \psi$

- Language $\mathscr{L}_1$ is at least as expressive as $\mathscr{L}_2$ ($\mathscr{L}_2 \leqslant \mathscr{L}_1$) if for all $\varphi \in \mathscr{L}_2$ there is an equivalent $\psi \in \mathscr{L}_1$. If $\mathscr{L}_2 \leqslant \mathscr{L}_1$ and $\mathscr{L}_1 \nleqslant \mathscr{L}_2$ we say that $\mathscr{L}_1$ is more expressive than $\mathscr{L}_2$. If $\mathscr{L}_2 \nleqslant \mathscr{L}_1$ and $\mathscr{L}_1 \nleqslant \mathscr{L}_2$, then $\mathscr{L}_1$ and $\mathscr{L}_2$ are incomparable

Theorem:

- Both DDCL$^+$ and DDCL$^-$ are more expressive than CL
- DDCL$^+$ and DDCL$^-$ are incomparable

Expressivity

- Formulas φ and ψ are equivalent if for all M_s, $M_s \vDash \varphi$ iff $M_s \vDash \psi$

- Language $\mathscr{L}_1$ is at least as expressive as $\mathscr{L}_2$ ($\mathscr{L}_2 \leqslant \mathscr{L}_1$) if for all $\varphi \in \mathscr{L}_2$ there is an equivalent $\psi \in \mathscr{L}_1$. If $\mathscr{L}_2 \leqslant \mathscr{L}_1$ and $\mathscr{L}_1 \nleqslant \mathscr{L}_2$ we say that $\mathscr{L}_1$ is more expressive than $\mathscr{L}_2$. If $\mathscr{L}_2 \nleqslant \mathscr{L}_1$ and $\mathscr{L}_1 \nleqslant \mathscr{L}_2$, then $\mathscr{L}_1$ and $\mathscr{L}_2$ are incomparable

Theorem:

- Both DDCL$^+$ and DDCL$^-$ are more expressive than CL
- DDCL$^+$ and DDCL$^-$ are incomparable

Corollary: formulas with updates cannot be equivalently rewritten into formulas of CL

Recap and open questions

Recap and open questions

We proposed a study of dynamic coalition logic

We proposed a logical treatment of granting and revoking dictatorial powers based on arrow updates and the notion of executability

We studied the relative expressivity of considered logics

Recap and open questions

We proposed a study of dynamic coalition logic

We proposed a logical treatment of granting and revoking dictatorial powers based on arrow updates and the notion of executability

We studied the relative expressivity of considered logics

?Granting powers to coalitions, not just individual agents?

?Proof systems for DDCL's?

?Alternative approach to updates, e.g. similar to action models of BMS?