Temporal logics for multi-agent systems

Nicolas Markey

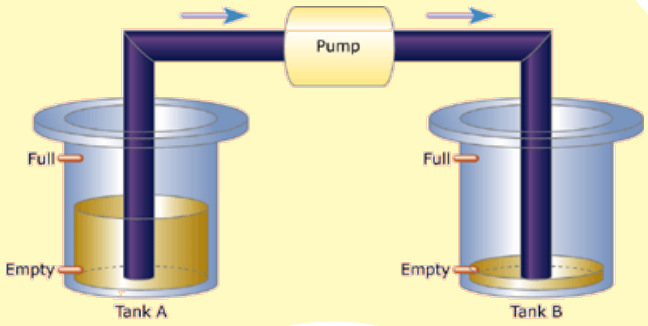
CNRS – IRISA (Rennes, France)

(based on joint works with Patricia Bouyer, Patrick Gardy, and François Laroussinie)

MFCS'17 – August 23, 2017

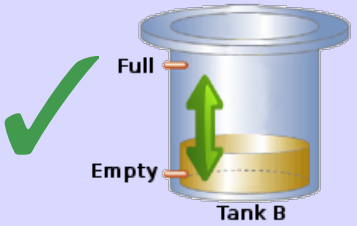
Model checking and synthesis

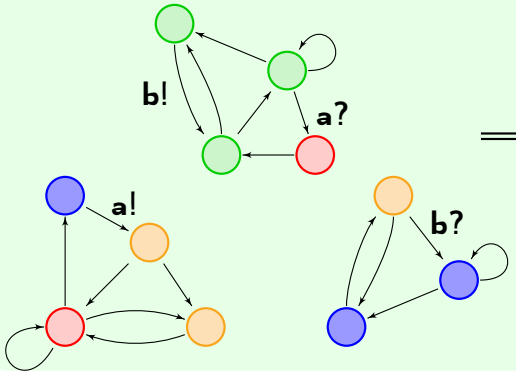
system:



[http://www.embedded.com]

property:



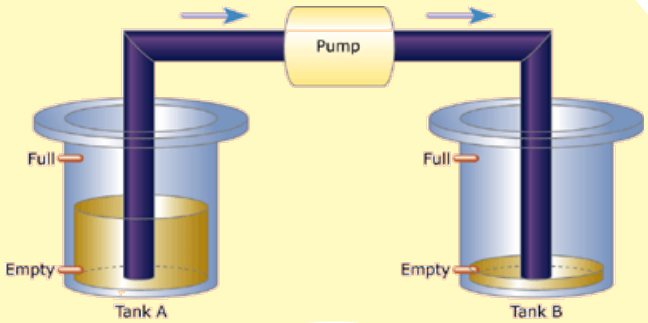


$\Longrightarrow$ model-checking algorithm $\Longleftarrow$ $\mathbf{A}\,\mathbf{G}(\neg\,\texttt{B.overfull} \wedge \neg\,\texttt{B.dried\_up})$

yes/no

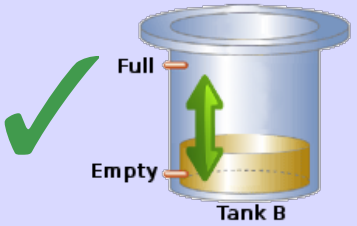
Model checking and synthesis

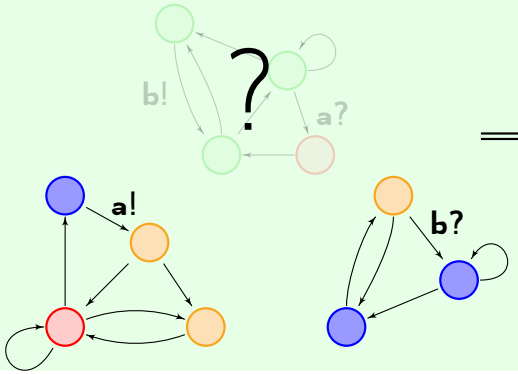
system:



[http://www.embedded.com]

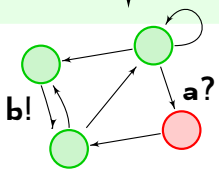
property:





synthesis algorithm

$\mathbf{A}\,\mathbf{G}(\neg\,\texttt{B.overfull} \wedge \neg\,\texttt{B.dried\_up})$



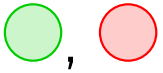
Computation-Tree Logic (CTL)

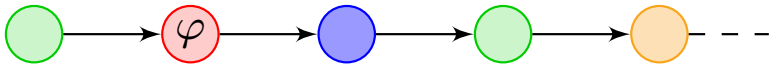
- **atomic propositions:**  ,  , ...

Computation-Tree Logic (CTL)

- **atomic propositions:**  ,  , ...
- **boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...

Computation-Tree Logic (CTL)

- **atomic propositions:**  , ...
- **boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities:**

$\mathbf{X}\,\varphi$ 

"next φ"

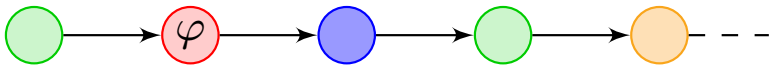
$\varphi\ \mathbf{U}\ \psi$ 

"φ until ψ"

Computation-Tree Logic (CTL)

- **atomic propositions:** ◯, ◯, ...
- **boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities:**

$\mathbf{X}\,\varphi$



"next φ"

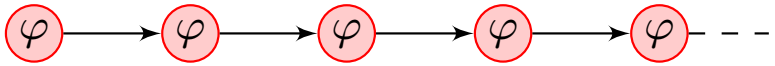
$\varphi\ \mathbf{U}\ \psi$

"φ until ψ"

$\texttt{true}\ \mathbf{U}\ \varphi \equiv \mathbf{F}\,\varphi$

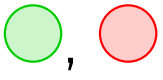
"eventually φ"

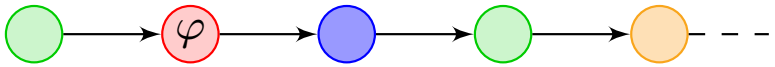
$\neg\ \mathbf{F}\ \neg\varphi \equiv \mathbf{G}\,\varphi$



"always φ"

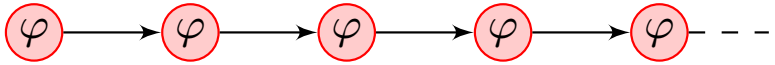
Computation-Tree Logic (CTL)

- **atomic propositions:**  , ...
- **boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities:**

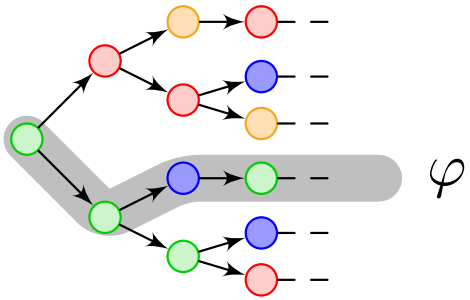
$\mathbf{X}\,\varphi$  "next φ"

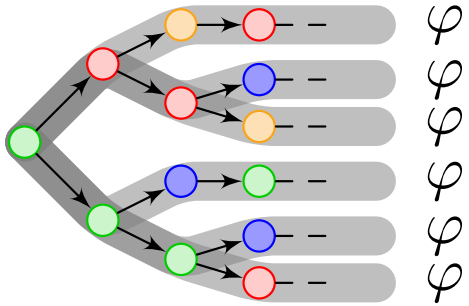
$\varphi\ \mathbf{U}\ \psi$  "φ until ψ"

$\mathtt{true}\ \mathbf{U}\ \varphi \equiv \mathbf{F}\,\varphi$  "eventually φ"

$\neg\,\mathbf{F}\,\neg\varphi \equiv \mathbf{G}\,\varphi$  "always φ"

- **path quantifiers:**

$\mathbf{E}\varphi$ 

$\mathbf{A}\varphi$ 

Computation-Tree Logic (CTL)

- **atomic propositions:** ◯, ◯, ...
- **boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities**

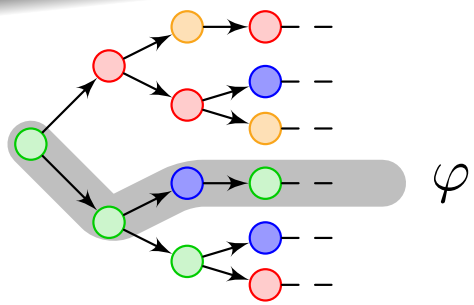
true U

¬ F

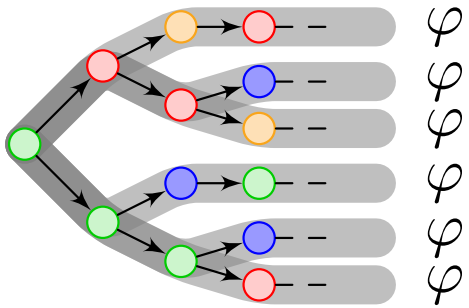
Examples

- **safety**: **AG** `safe`
  only `safe` states will be reached
- **fairness**: **AG AF** `fair`
  `fair` states will be visited infinitely many times
- **request-response**: **AG**(`request` ⇒ **EF** `grant`)
  any `request` can eventually be granted

Eφ



Aφ



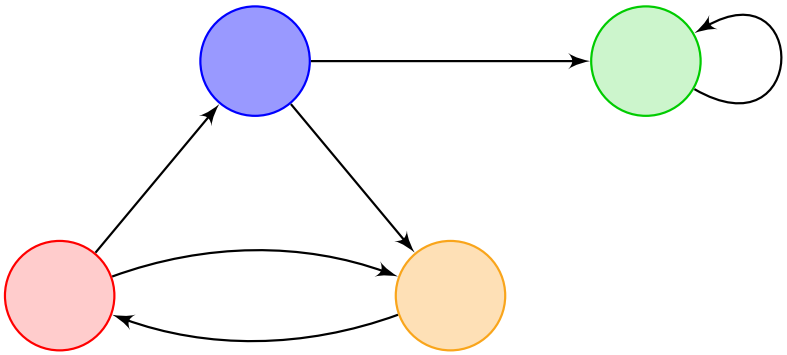
CTL model checking

In CTL, each temporal modality is in the immediate scope of a path quantifier.

CTL model checking

In CTL, each temporal modality is in the immediate scope of a path quantifier.

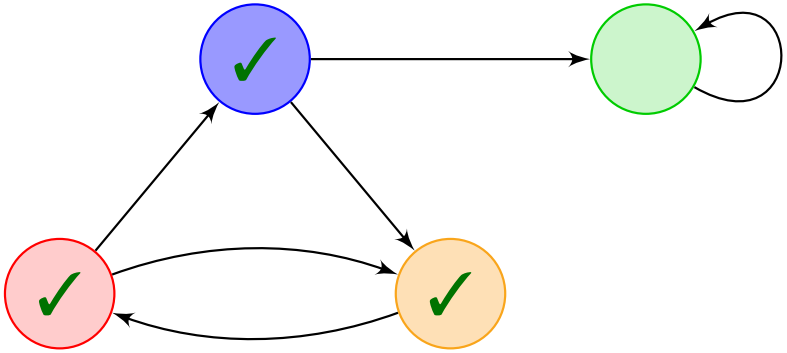
E F   is reachable



CTL model checking

In CTL, each temporal modality is in the immediate scope of a path quantifier.

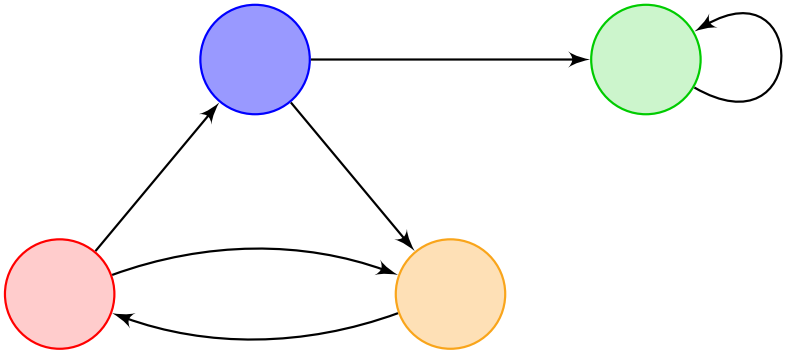
E F   is reachable



CTL model checking

In CTL, each temporal modality is in the immediate scope of a path quantifier.

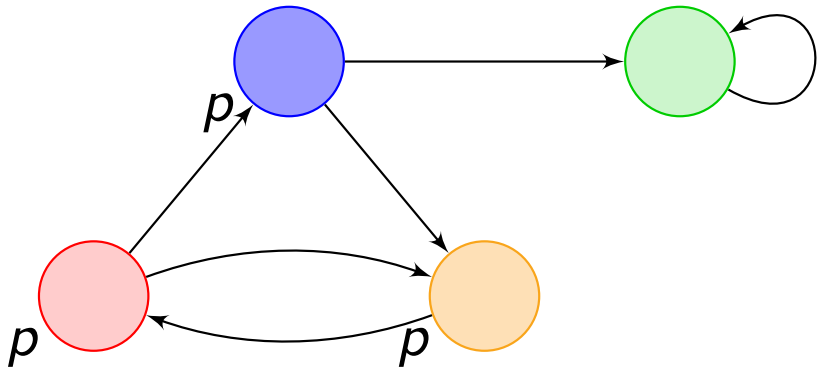
$\mathbf{E}\,\mathbf{G}(\neg$ ● $\wedge\ \mathbf{E}\,\mathbf{F}$ ●$)$ there is a path along which ● is always reachable, but never reached



CTL model checking

In CTL, each temporal modality is in the immediate scope of a path quantifier.

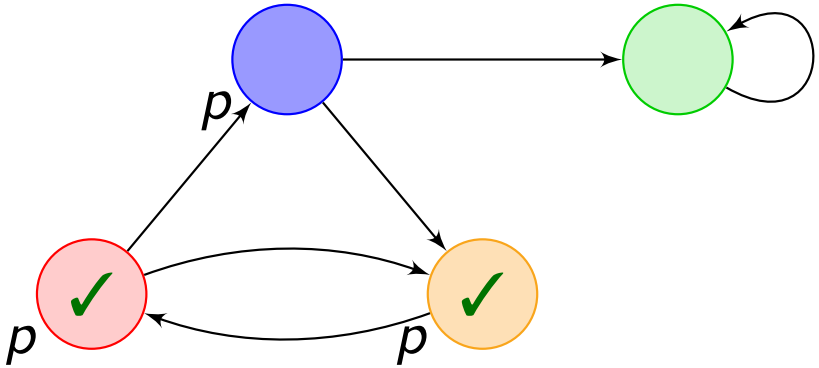
$\mathbf{E}\,\mathbf{G}(\neg$ ● $\wedge \underbrace{\mathbf{E}\,\mathbf{F}\,●}_{p})$ there is a path along which ● is always reachable, but never reached



CTL model checking

In CTL, each temporal modality is in the immediate scope of a path quantifier.

$\mathbf{E}\,\mathbf{G}(\neg$  $\wedge \underbrace{\mathbf{E}\,\mathbf{F}}_{p}$  $)$ there is a path along which  is always reachable, but never reached



CTL model checking

In CTL, each temporal modality is in the immediate scope of a path quantifier.

Theorem ([CE81,QS82])

CTL model checking is PTIME*-complete.*

[CE81] Clarke, Emerson. Design and Synthesis of Synchronization Skeletons... LOP, 1981.
[QS82] Queille, Sifakis. Specification and verification of concurrent systems in CESAR. SOP, 1982.

CTL model checking

In CTL, each temporal modality is in the immediate scope of a path quantifier.

Theorem ([CE81,QS82])

CTL model checking is PTIME*-complete.*

Remark ([KVW94])

CTL model checking on product structures is PSPACE-complete.

[CE81] Clarke, Emerson. Design and Synthesis of Synchronization Skeletons... LOP, 1981.
[QS82] Queille, Sifakis. Specification and verification of concurrent systems in CESAR. SOP, 1982.
[KVW94] Kupferman, Vardi, Wolper. An automata-theoretic approach to branching-time... CAV, 1994.

Outline of the presentation

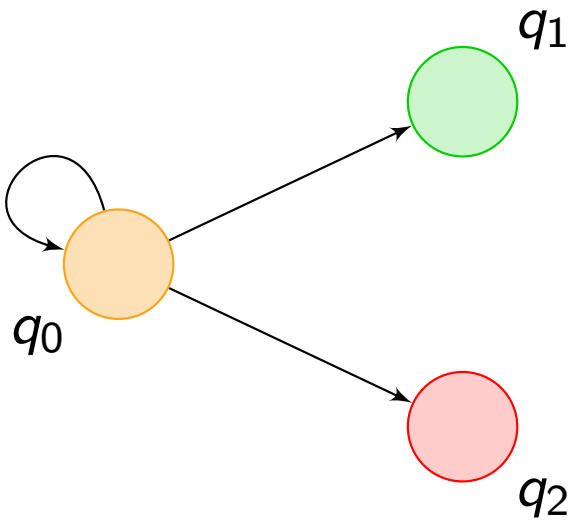
Outline of the presentation

Games on graphs

Concurrent games

A concurrent game is made of

- a transition system;

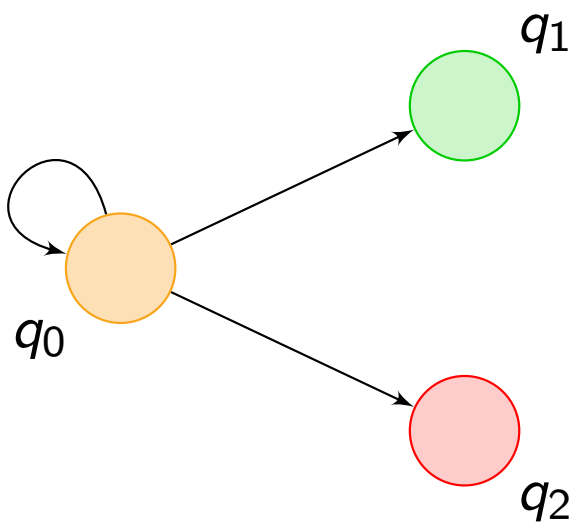


Games on graphs

Concurrent games

A concurrent game is made of

- a transition system;
- a set of agents (or players);

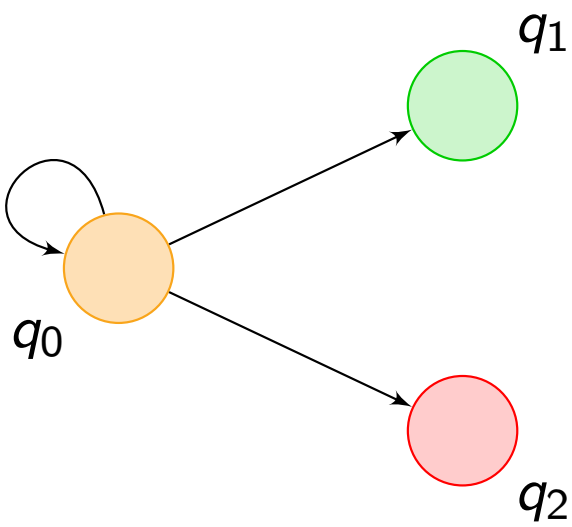


Games on graphs

Concurrent games

A concurrent game is made of

- a transition system;
- a set of agents (or players);
- a table indicating the transition to be taken given the actions of the players.



		player 1		
				
player 2		q_0	q_2	q_1
		q_1	q_0	q_2
		q_2	q_1	q_0

Games on graphs

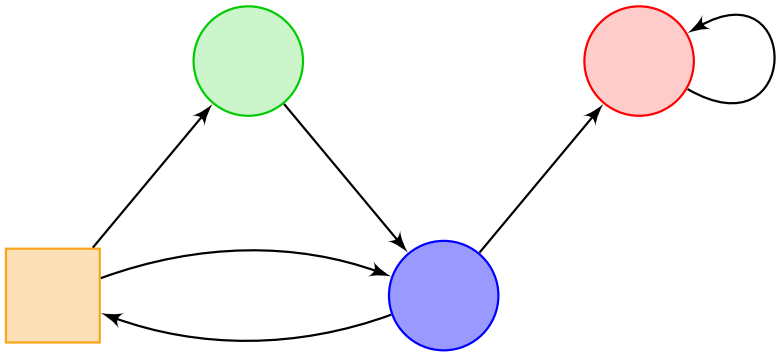
Concurrent games

A concurrent game is made of

- a transition system;
- a set of agents (or players);
- a table indicating the transition to be taken given the actions of the players.

Turn-based games

A turn-based game is a game where only one agent plays at a time.



Reasoning about open systems

Strategies

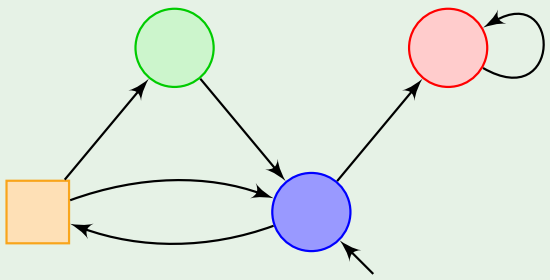
A (pure) strategy for a given player is a function telling which action to play depending on what has happened previously.

Reasoning about open systems

Strategies

A (pure) strategy for a given player is a function telling which action to play depending on what has happened previously.

Example



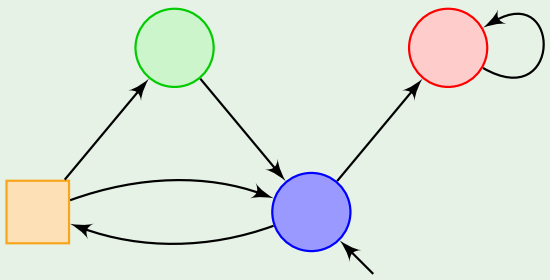
Reasoning about open systems

Strategies

A (pure) strategy for a given player is a function telling which action to play depending on what has happened previously.

Example

Strategy for player



Reasoning about open systems

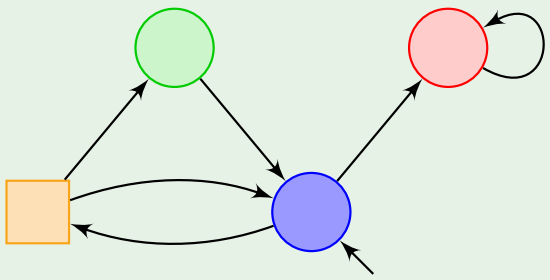
Strategies

A (pure) strategy for a given player is a function telling which action to play depending on what has happened previously.

Example

Strategy for player

alternately go to and (starting with).



Reasoning about open systems

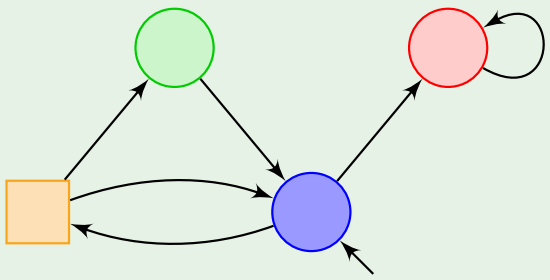
Strategies

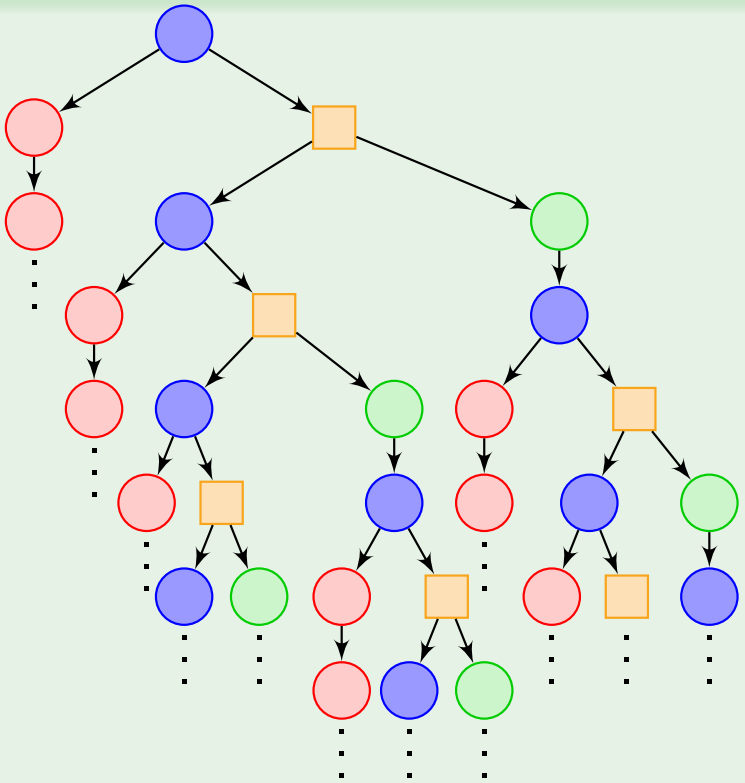
A (pure) strategy for a given player is a function telling which action to play depending on what has happened previously.

Example

Strategy for player

alternately go to and (starting with).





Reasoning about open systems

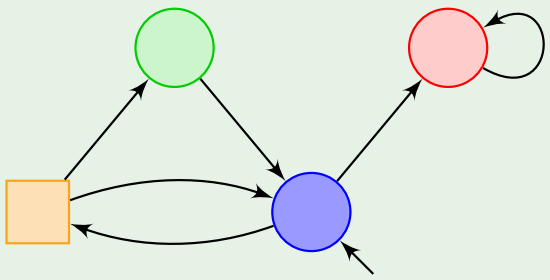
Strategies

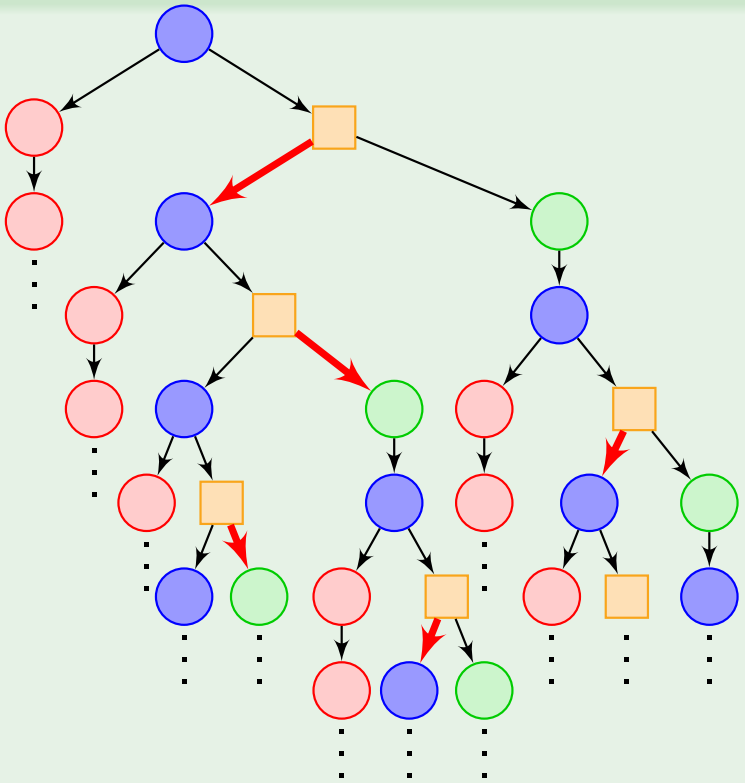
A (pure) strategy for a given player is a function telling which action to play depending on what has happened previously.

Example

Strategy for player

alternately go to and (starting with).





Reasoning about open systems

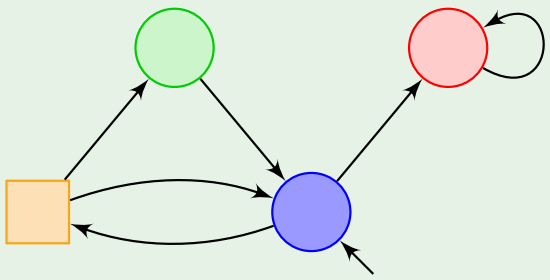
Strategies

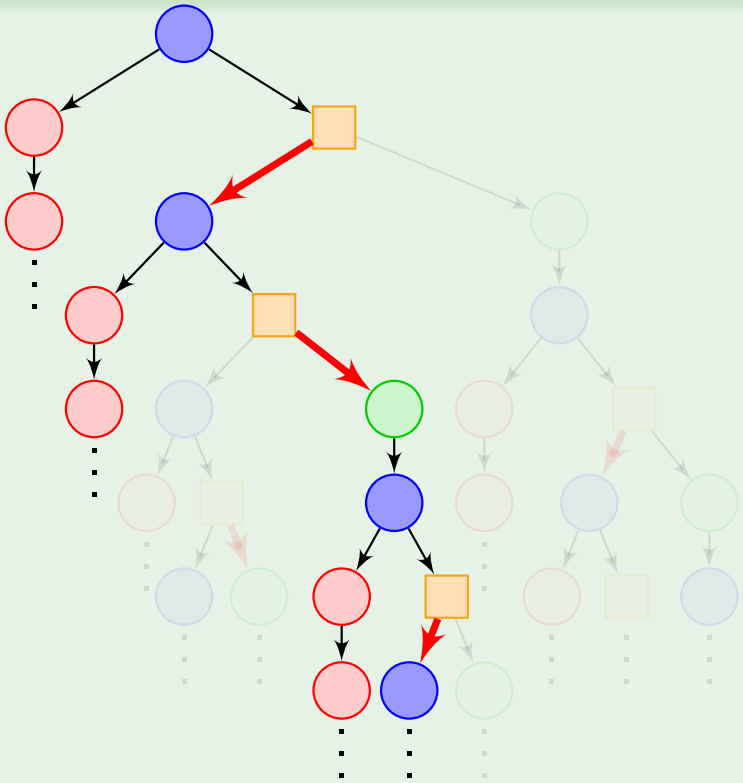
A (pure) strategy for a given player is a function telling which action to play depending on what has happened previously.

Example

Strategy for player

alternately go to and (starting with).





Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle \varphi$ expresses that A has a strategy to enforce φ.

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.

Semantics of $\langle\langle A \rangle\rangle\, \varphi$

Existential quantification (over strategies) implicitly includes a universal quantification (over outcomes):

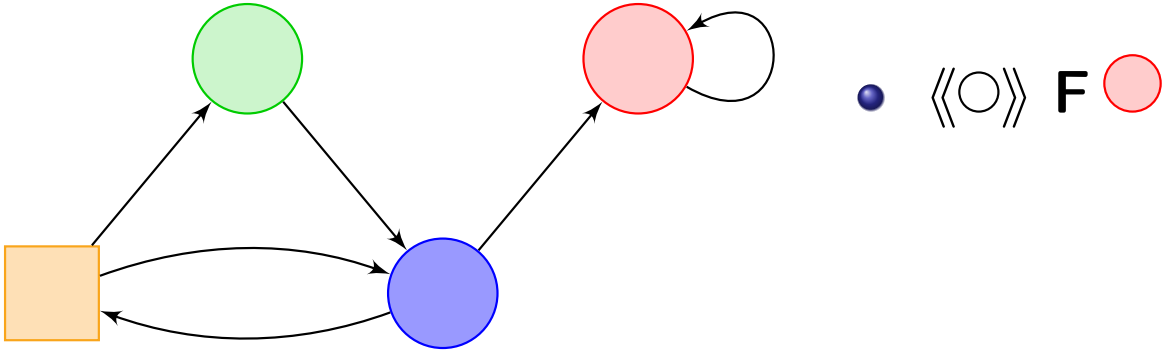
$$\mathcal{G}, \bigcirc \models \langle\langle A \rangle\rangle\, \varphi \quad \iff \quad \exists \sigma_A.\ \forall \pi \in \mathsf{Out}(\bigcirc, \sigma_A).\ \pi \models \varphi.$$

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.

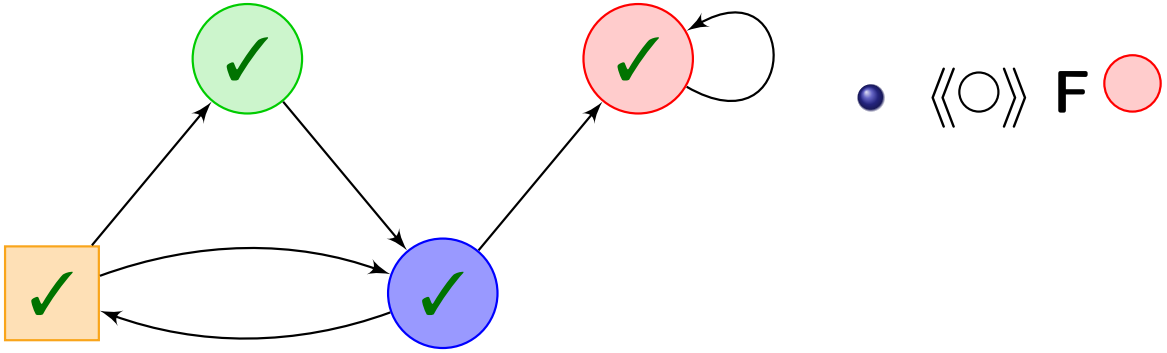


[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.

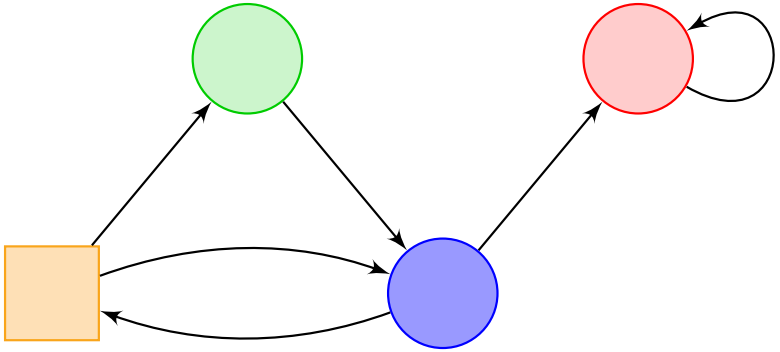


[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.



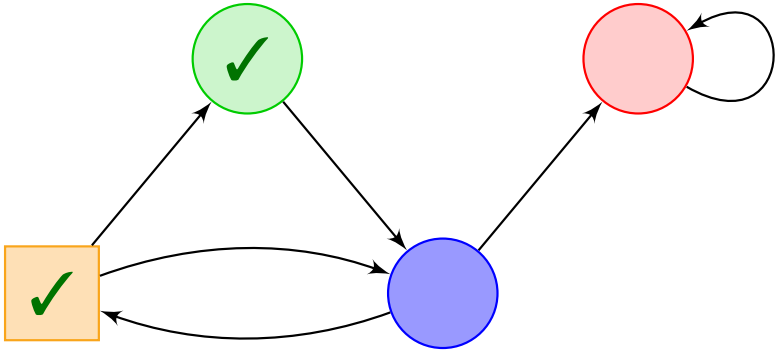
- $\langle\langle \bigcirc \rangle\rangle$ **F**
- $\langle\langle \Box \rangle\rangle$ **F**

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.



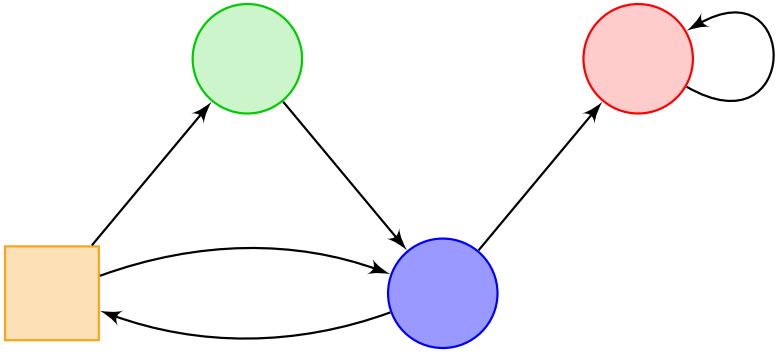
- $\langle\langle \bigcirc \rangle\rangle$ **F**
- $\langle\langle \Box \rangle\rangle$ **F**

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.



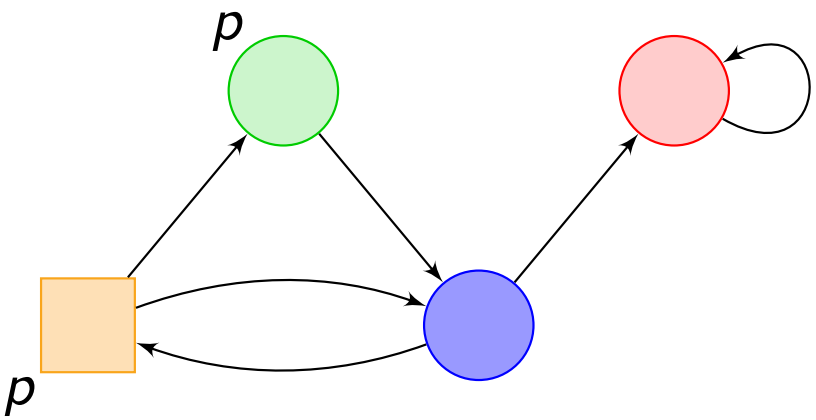
- $\langle\langle \bigcirc \rangle\rangle\ \mathbf{F}$
- $\langle\langle \Box \rangle\rangle\ \mathbf{F}$
- $\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \Box \rangle\rangle\ \mathbf{F}\ )$

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.



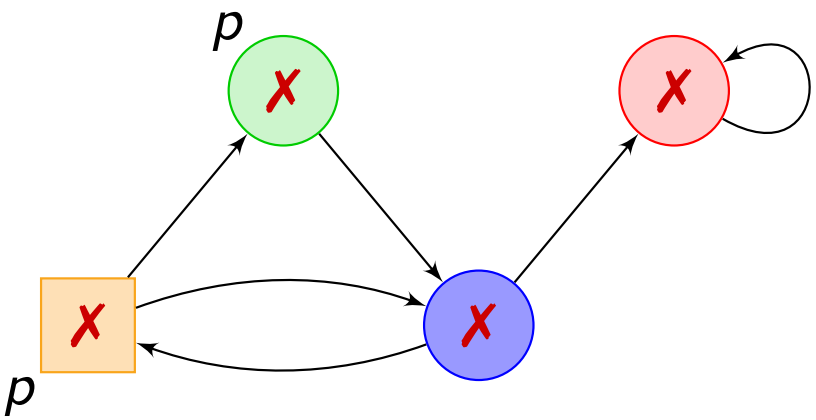
- $\langle\langle \bigcirc \rangle\rangle\, \mathbf{F}$
- $\langle\langle \Box \rangle\rangle\, \mathbf{F}$
- $\langle\langle \bigcirc \rangle\rangle\, \mathbf{G}(\underbrace{\langle\langle \Box \rangle\rangle\, \mathbf{F}}_{p}) \equiv \langle\langle \bigcirc \rangle\rangle\, \mathbf{G}\, p$

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.



- $\langle\langle \bigcirc \rangle\rangle\, \mathbf{F}$
- $\langle\langle \Box \rangle\rangle\, \mathbf{F}$
- $\langle\langle \bigcirc \rangle\rangle\, \mathbf{G}(\underbrace{\langle\langle \Box \rangle\rangle\, \mathbf{F}}_{p}) \equiv \langle\langle \bigcirc \rangle\rangle\, \mathbf{G}\, p$

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.

Theorem ([AHK02])

Model checking ATL is PTIME*-complete.*

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.

Temporal logics for games: ATL [AHK02]

ATL extends CTL with strategy quantifiers

$\langle\langle A \rangle\rangle\, \varphi$ expresses that A has a strategy to enforce φ.

Theorem ([AHK02])

Model checking ATL is PTIME*-complete.*

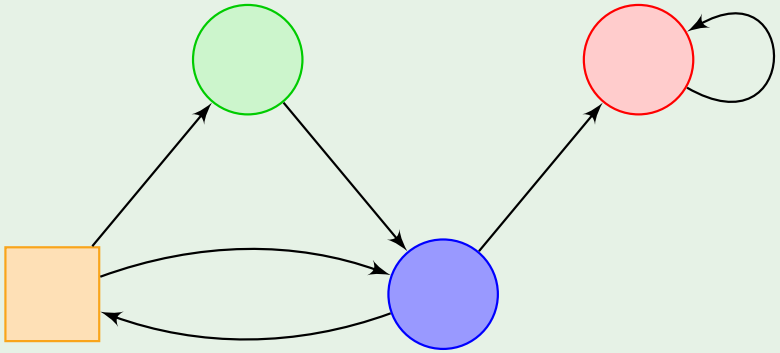
Remark ([LMO08])

In PTIME only if the transition table is given explicitly (size $|\mathsf{Moves}|^{|\mathsf{Agt}|}$)

[AHK02] Alur, Henzinger, Kupferman. Alternating-time Temporal Logic. J. ACM, 2002.
[LMO08] Laroussinie, Markey, Oreiby. On the Expressiveness and Complexity of ATL. LMCS, 2008

ATL with strategy contexts [DLM10]

Example

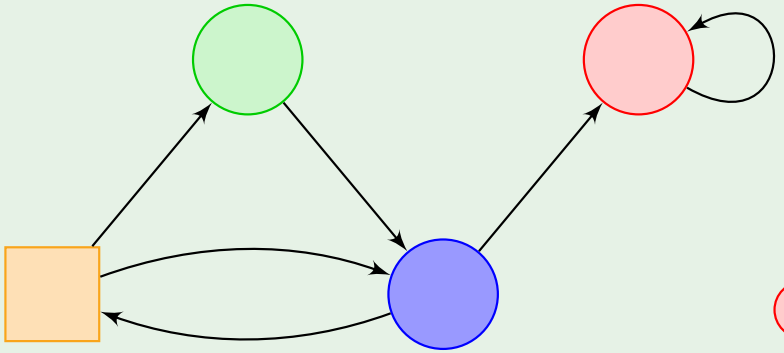


$\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \Box \rangle\rangle\ \mathbf{F}\ \bigcirc\ )$

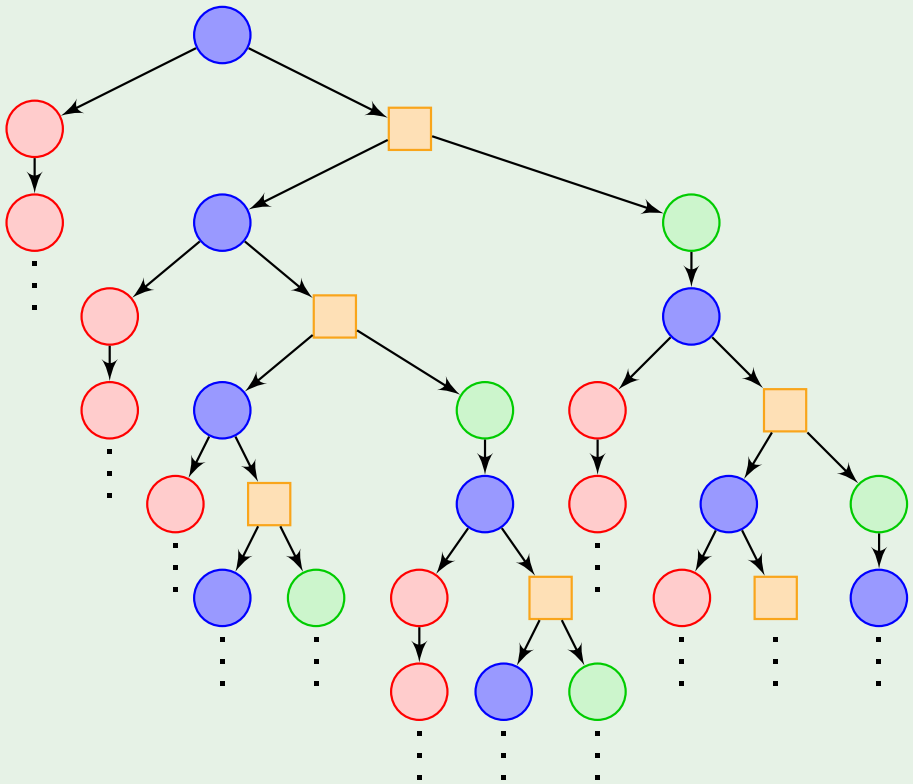
[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

ATL with strategy contexts [DLM10]

Example



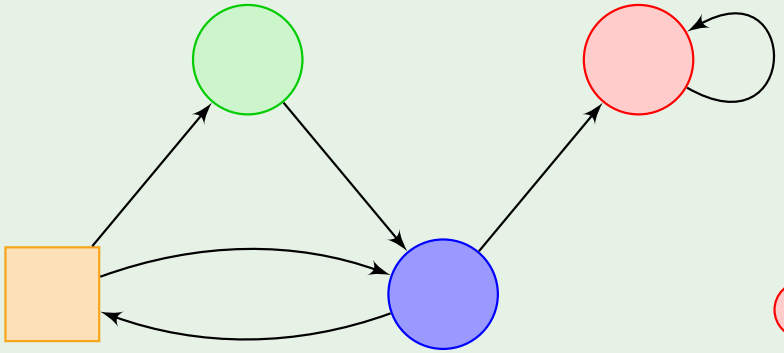
$\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \Box \rangle\rangle\ \mathbf{F}\ \bigcirc\ )$



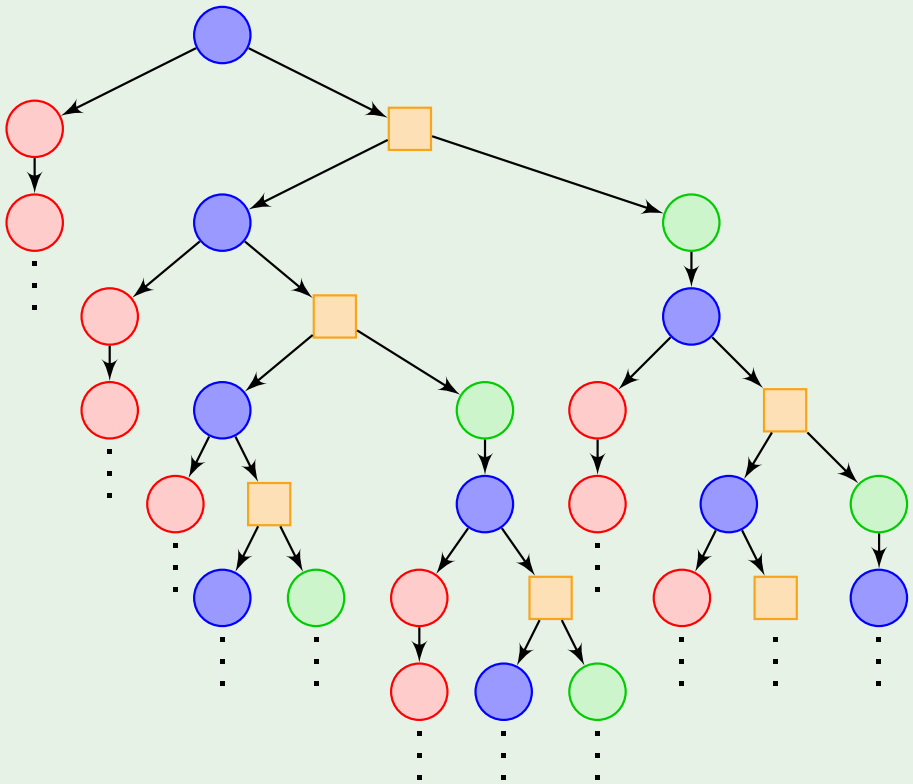
[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

ATL with strategy contexts [DLM10]

Example



$\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \square \rangle\rangle\ \mathbf{F}\ \bigcirc\ )$

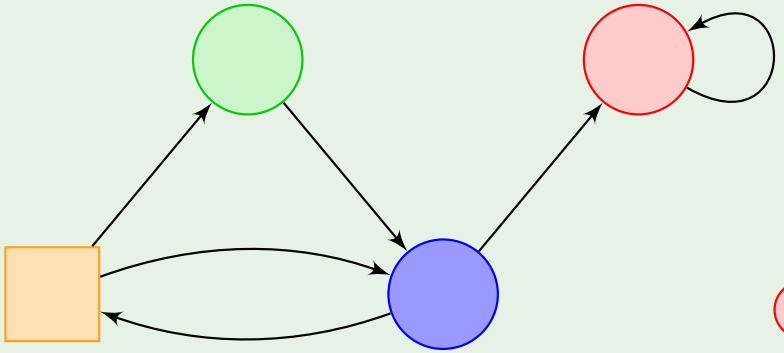


- Player ○ in ● always plays to □.

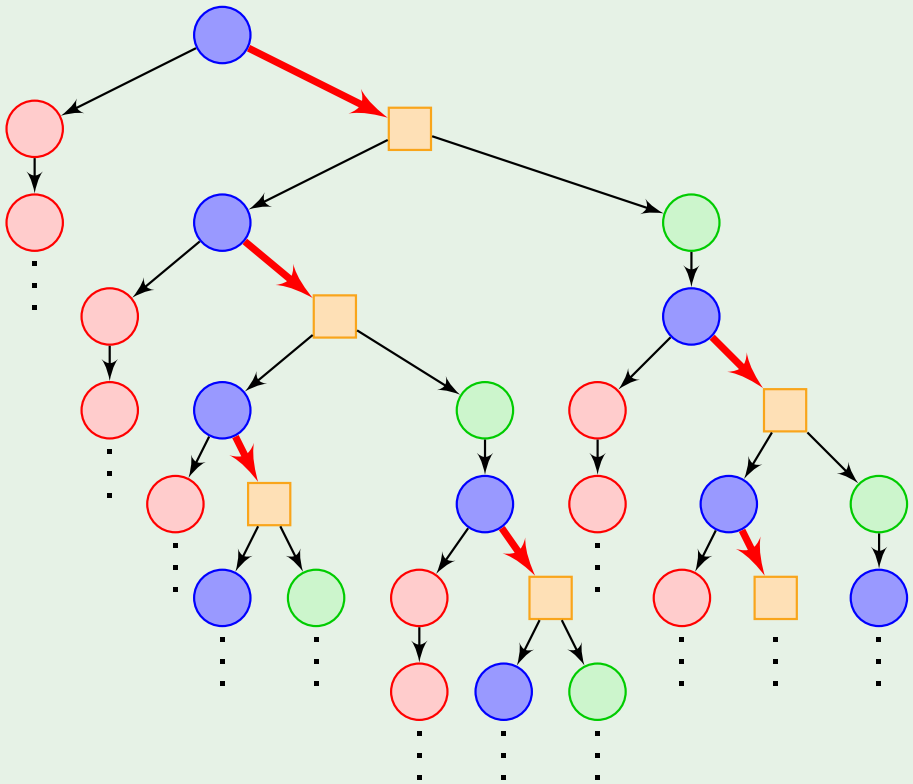
[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

ATL with strategy contexts [DLM10]

Example



$\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \square \rangle\rangle\ \mathbf{F}\ \bigcirc\ )$

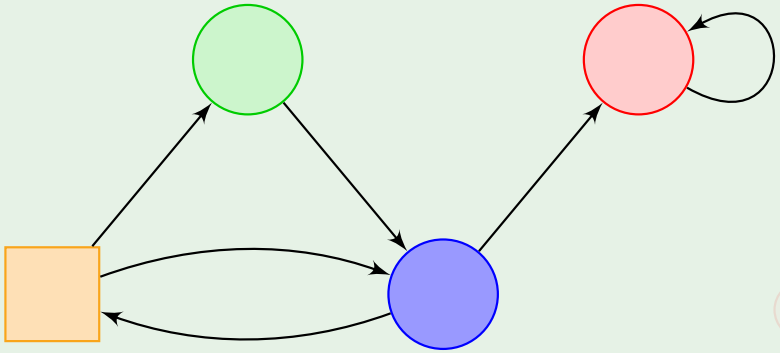


- Player ○ in ● always plays to □.

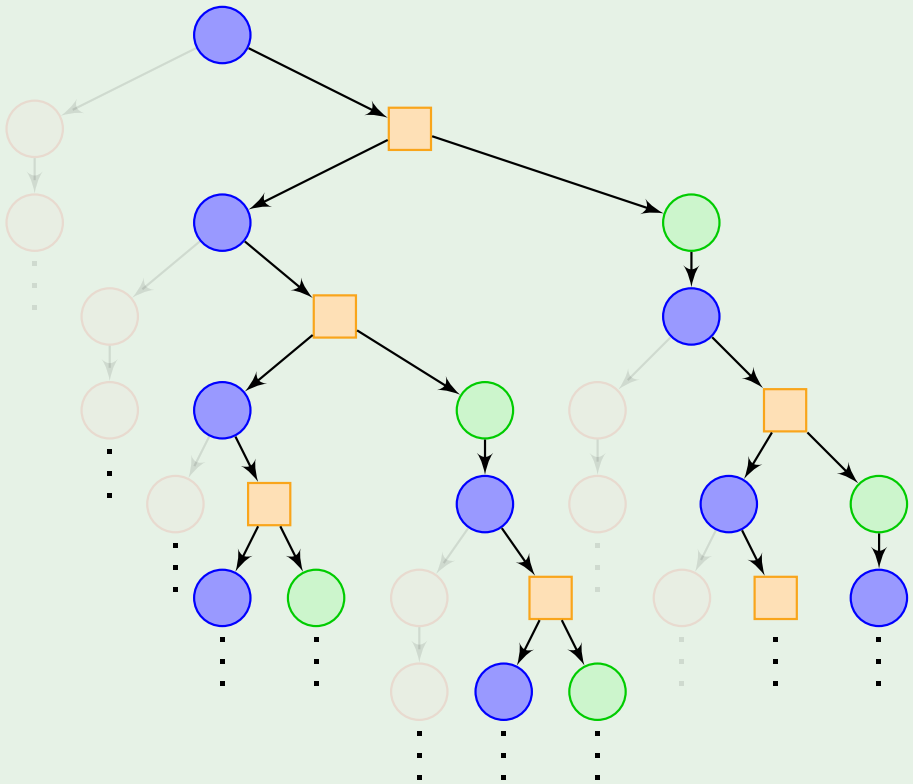
[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

ATL with strategy contexts [DLM10]

Example



$\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \square \rangle\rangle\ \mathbf{F}\ \bigcirc\ )$

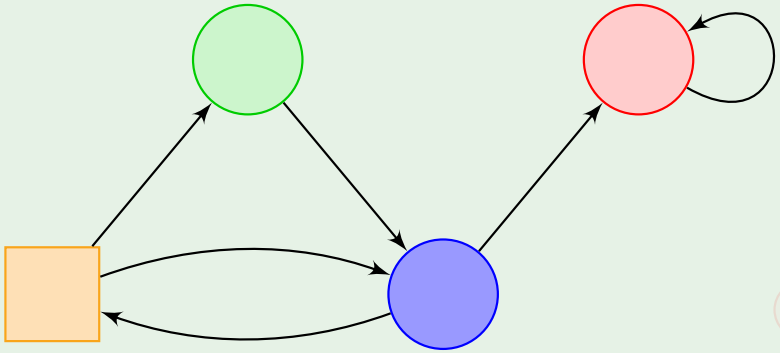


- Player ○ in ● always plays to □.

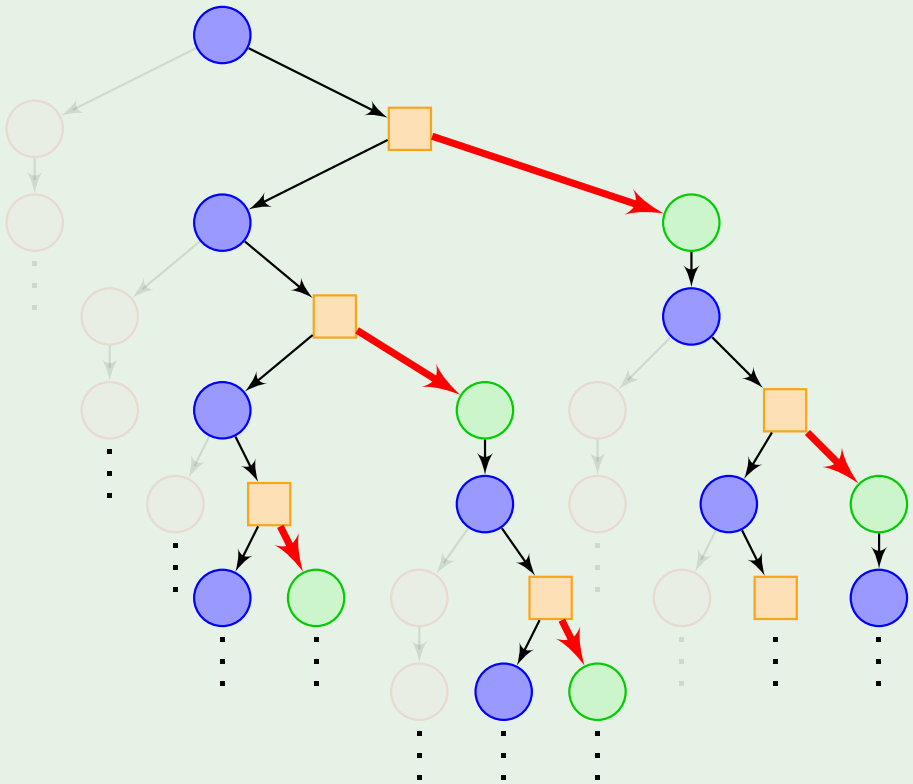
[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

ATL with strategy contexts [DLM10]

Example



$\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \square \rangle\rangle\ \mathbf{F}\ \bigcirc\ )$



- Player ○ in ● always plays to □;
- Player □ in □ then plays to ○.

[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

ATL with strategy contexts [DLM10]

Definition

ATL_{sc} has a new strategy quantifier:

- $\langle\cdot A\cdot\rangle\,\varphi$ is similar to $\langle\langle A\rangle\rangle\,\varphi$ but assigns the corresponding strategy to A for evaluating φ;

[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

ATL with strategy contexts [DLM10]

Definition

ATL_{sc} has a new strategy quantifier:

- $\langle\cdot A\cdot\rangle\,\varphi$ is similar to $\langle\langle A\rangle\rangle\,\varphi$ but assigns the corresponding strategy to A for evaluating φ;

Definition

Semantics of ATL strategy quantifier:

$\mathcal{G}, \bigcirc \models \langle\langle A\rangle\rangle\,\varphi \quad \Leftrightarrow \quad \exists\sigma_A.\ \forall\pi \in \mathrm{Out}(\bigcirc, \sigma_A).\ \pi \models \varphi$

[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

ATL with strategy contexts [DLM10]

Definition

ATL_{sc} has a new strategy quantifier:

- $\langle\cdot A\cdot\rangle\,\varphi$ is similar to $\langle\langle A\rangle\rangle\,\varphi$ but assigns the corresponding strategy to A for evaluating φ;

Definition

Semantics of ATL strategy quantifier:

$\mathcal{G}, \bigcirc \models \langle\langle A\rangle\rangle\,\varphi \quad \Leftrightarrow \quad \exists\sigma_A.\ \forall\pi \in \mathrm{Out}(\bigcirc, \sigma_A).\ \pi \models \varphi$

Semantics of ATL_{sc} strategy quantifier:

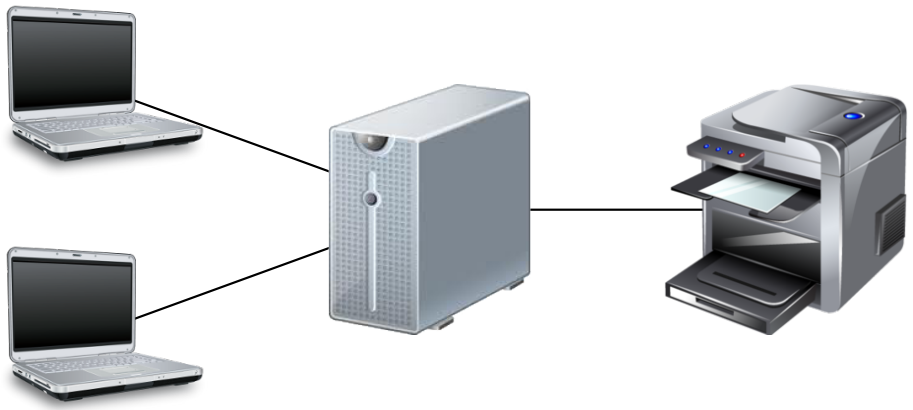
$\mathcal{G}, \bigcirc \models_{\sigma_B} \langle\cdot A\cdot\rangle\,\varphi \quad \Leftrightarrow \quad \exists\sigma_A.\ \forall\pi \in \mathrm{Out}(\bigcirc, \sigma_A \cup \sigma_B).\ \pi \models_{\sigma_A\cup\sigma_B} \varphi$

[DLM10] Da Costa, Laroussinie, Markey. ATL with strategy contexts: expressiveness and ... FSTTCS, 2010.

Expressiveness of ATL_{sc}

- Client-server interactions for accessing a shared resource:

$$\langle\cdot\mathrm{Server}\cdot\rangle\ \mathbf{G}\left[\begin{array}{l} \bigwedge_{c\in\mathrm{Clients}} \langle\cdot c\cdot\rangle\ \mathbf{F}\ \mathrm{access}_c \\ \wedge \\ \neg \bigwedge_{c\neq c'} \mathrm{access}_c \wedge \mathrm{access}_{c'} \end{array}\right]$$



Expressiveness of ATL_{sc}

- Client-server interactions for accessing a shared resource:

$$\langle\cdot \mathrm{Server} \cdot\rangle\ \mathbf{G} \left[\begin{array}{l} \bigwedge_{c \in \mathrm{Clients}} \langle\cdot c \cdot\rangle\ \mathbf{F}\ \mathrm{access}_c \\ \wedge \\ \neg \bigwedge_{c \neq c'} \mathrm{access}_c \wedge \mathrm{access}_{c'} \end{array} \right]$$

- Existence of Nash equilibria:

$$\langle\cdot A_1, \ldots, A_n \cdot\rangle \bigwedge_i \left(\left[\langle\cdot A_i \cdot\rangle\, \varphi_{A_i} \right] \Rightarrow \varphi_{A_i} \right)$$

Expressiveness of ATL_{sc}

- Client-server interactions for accessing a shared resource:

$$\langle\cdot \mathrm{Server} \cdot\rangle\ \mathbf{G} \left[\begin{array}{l} \bigwedge_{c \in \mathrm{Clients}} \langle\cdot c \cdot\rangle\ \mathbf{F}\ \mathrm{access}_c \\ \wedge \\ \neg \bigwedge_{c \neq c'} \mathrm{access}_c \wedge \mathrm{access}_{c'} \end{array} \right]$$

- Existence of Nash equilibria:

$$\langle\cdot A_1, \ldots, A_n \cdot\rangle \bigwedge_i \left(\left[\langle\cdot A_i \cdot\rangle\, \varphi_{A_i} \right] \Rightarrow \varphi_{A_i} \right)$$

- Existence of dominating strategy:

$$\langle\cdot A \cdot\rangle\ [\cdot B \cdot]\, (\neg \varphi \Rightarrow [\cdot A \cdot]\, \neg \varphi)$$

Outline of the presentation

Quantified CTL [ES84,Kup95,Fre01,DLM12]

QCTL extends CTL with propositional quantifiers

$\exists p.\ \varphi$ means that there exists a labelling of the model with p under which φ holds.

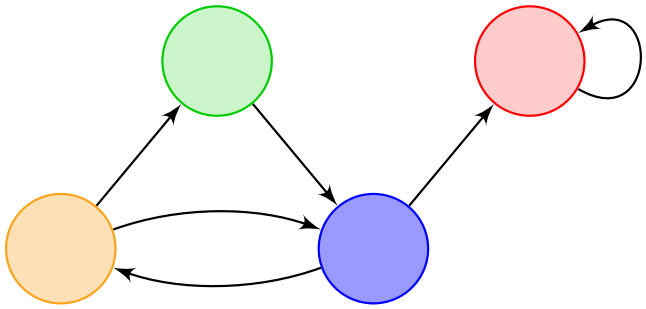
[ES84] Emerson and Sistla. Deciding Full Branching Time Logic. Information & Control, 1984.
[Kup95] Kupferman. Augmenting Branching Temporal Logics with Existential... CAV, 1995.
[Fre01] French. Decidability of Quantifed Propositional Branching Time Logics. AJCAI, 2001.
[DLM12] Da Costa, Laroussinie, Markey. Quantified CTL: ... CONCUR, 2012.

Quantified CTL [ES84,Kup95,Fre01,DLM12]

QCTL extends CTL with propositional quantifiers

$\exists p.\ \varphi$ means that there exists a labelling of the model with p under which φ holds.

$$\mathbf{E}\,\mathbf{F}\,\bigcirc \;\wedge\; \neg\Big(\exists p.\ \big[\mathbf{E}\,\mathbf{F}(\bigcirc \wedge p) \wedge \mathbf{E}\,\mathbf{F}(\bigcirc \wedge \neg p)\big]\Big)$$



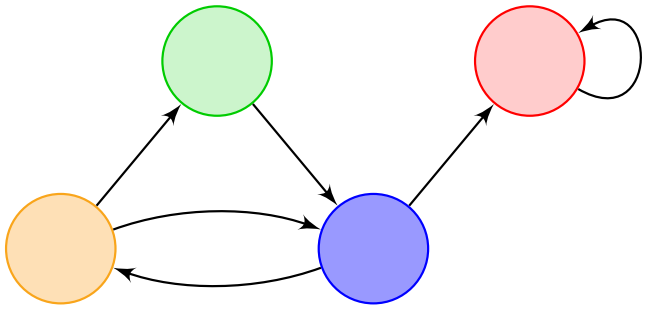
[ES84] Emerson and Sistla. Deciding Full Branching Time Logic. Information & Control, 1984.
[Kup95] Kupferman. Augmenting Branching Temporal Logics with Existential... CAV, 1995.
[Fre01] French. Decidability of Quantifed Propositional Branching Time Logics. AJCAI, 2001.
[DLM12] Da Costa, Laroussinie, Markey. Quantified CTL: ... CONCUR, 2012.

Quantified CTL [ES84,Kup95,Fre01,DLM12]

QCTL extends CTL with propositional quantifiers

$\exists p.\ \varphi$ means that there exists a labelling of the model with p under which φ holds.

$$\mathbf{E}\,\mathbf{F}\,\bigcirc \;\wedge\; \neg\Big(\exists p.\ \big[\mathbf{E}\,\mathbf{F}(\bigcirc \wedge p) \wedge \mathbf{E}\,\mathbf{F}(\bigcirc \wedge \neg p)\big]\Big) \equiv \mathsf{uniq}(\bigcirc)$$



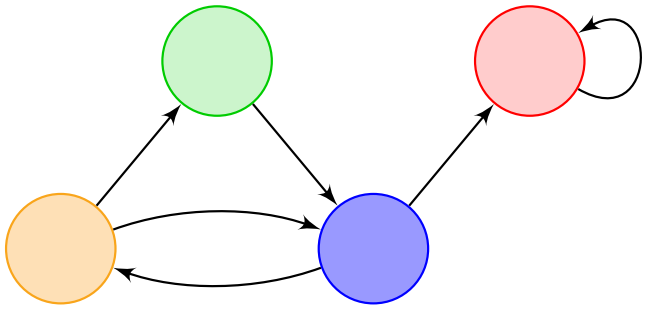
[ES84] Emerson and Sistla. Deciding Full Branching Time Logic. Information & Control, 1984.
[Kup95] Kupferman. Augmenting Branching Temporal Logics with Existential... CAV, 1995.
[Fre01] French. Decidability of Quantifed Propositional Branching Time Logics. AJCAI, 2001.
[DLM12] Da Costa, Laroussinie, Markey. Quantified CTL: ... CONCUR, 2012.

Quantified CTL [ES84,Kup95,Fre01,DLM12]

QCTL extends CTL with propositional quantifiers

$\exists p.\ \varphi$ means that there exists a labelling of the model with p under which φ holds.

$$\mathbf{E}\,\mathbf{F}\,\bigcirc \;\wedge\; \neg\Big(\exists p.\ \big[\mathbf{E}\,\mathbf{F}(\bigcirc \wedge p) \wedge \mathbf{E}\,\mathbf{F}(\bigcirc \wedge \neg p)\big]\Big) \equiv \mathrm{uniq}(\bigcirc)$$



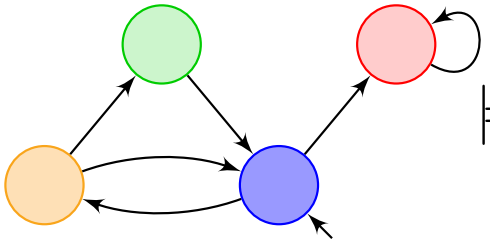
$\rightsquigarrow$ true if we label the Kripke structure;

$\rightsquigarrow$ false if we label the computation tree;

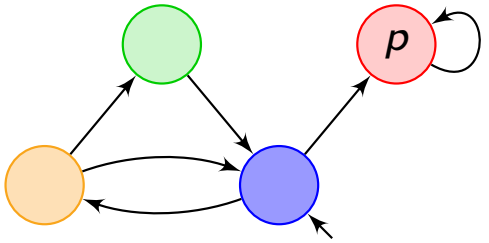
[ES84] Emerson and Sistla. Deciding Full Branching Time Logic. Information & Control, 1984.
[Kup95] Kupferman. Augmenting Branching Temporal Logics with Existential... CAV, 1995.
[Fre01] French. Decidability of Quantifed Propositional Branching Time Logics. AJCAI, 2001.
[DLM12] Da Costa, Laroussinie, Markey. Quantified CTL: ... CONCUR, 2012.

Semantics of QCTL

- structure semantics:



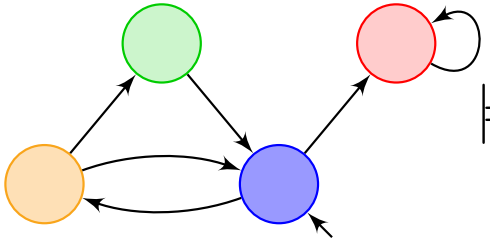
$\models_s \exists p.\varphi \quad \Leftrightarrow$



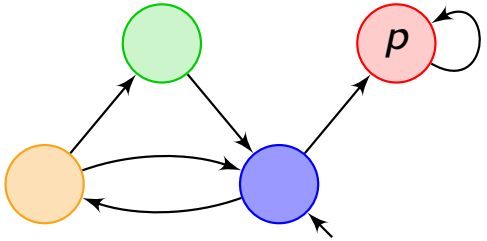
$\models \varphi$

Semantics of QCTL

- structure semantics:

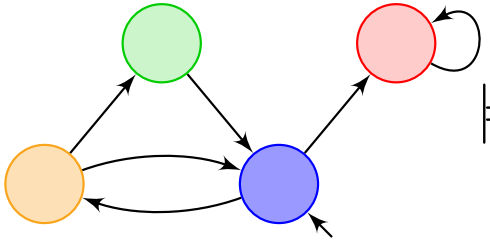


$\models_s \exists p.\varphi \quad \Leftrightarrow$

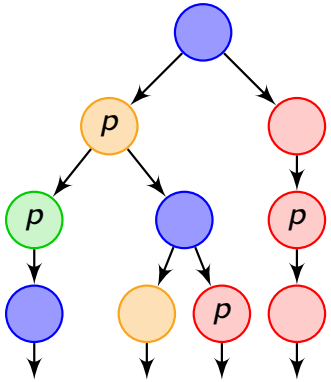


$\models \varphi$

- tree semantics:



$\models_t \exists p.\varphi \quad \Leftrightarrow$



$\models \varphi$

Expressiveness of QCTL

- QCTL can "count":

$$\mathbf{E}\,\mathbf{X}_1\,\varphi \equiv \mathbf{E}\,\mathbf{X}\,\varphi \wedge \forall p.\ [\mathbf{E}\,\mathbf{X}(p \wedge \varphi) \Rightarrow \mathbf{A}\,\mathbf{X}(\varphi \Rightarrow p)]$$
$$\mathbf{E}\,\mathbf{X}_2\,\varphi \equiv \exists q.\ [\mathbf{E}\,\mathbf{X}_1(\varphi \wedge q) \wedge \mathbf{E}\,\mathbf{X}_1(\varphi \wedge \neg q)]$$

Expressiveness of QCTL

- QCTL can "count":

$$\mathbf{E}\,\mathbf{X}_1\,\varphi \equiv \mathbf{E}\,\mathbf{X}\,\varphi \wedge \forall p.\ [\mathbf{E}\,\mathbf{X}(p \wedge \varphi) \Rightarrow \mathbf{A}\,\mathbf{X}(\varphi \Rightarrow p)]$$
$$\mathbf{E}\,\mathbf{X}_2\,\varphi \equiv \exists q.\ [\mathbf{E}\,\mathbf{X}_1(\varphi \wedge q) \wedge \mathbf{E}\,\mathbf{X}_1(\varphi \wedge \neg q)]$$

- QCTL can express (least or greatest) fixpoints:

$$\mu T.\varphi(T) \equiv \exists t.\ [\mathbf{A}\,\mathbf{G}(t \iff \varphi(t)) \wedge (\forall t.'(\mathbf{A}\,\mathbf{G}(t' \iff \varphi(t')) \Rightarrow \mathbf{A}\,\mathbf{G}(t \Rightarrow t')))]$$

Expressiveness of QCTL

- QCTL can "count":

$$\mathbf{E}\,\mathbf{X}_1\,\varphi \equiv \mathbf{E}\,\mathbf{X}\,\varphi \wedge \forall p.\ [\mathbf{E}\,\mathbf{X}(p \wedge \varphi) \Rightarrow \mathbf{A}\,\mathbf{X}(\varphi \Rightarrow p)]$$
$$\mathbf{E}\,\mathbf{X}_2\,\varphi \equiv \exists q.\ [\mathbf{E}\,\mathbf{X}_1(\varphi \wedge q) \wedge \mathbf{E}\,\mathbf{X}_1(\varphi \wedge \neg q)]$$

- QCTL can express (least or greatest) fixpoints:

$$\mu T.\varphi(T) \equiv \exists t.\ [\mathbf{A}\,\mathbf{G}(t \iff \varphi(t)) \wedge (\forall t.'(\mathbf{A}\,\mathbf{G}(t' \iff \varphi(t')) \Rightarrow \mathbf{A}\,\mathbf{G}(t \Rightarrow t')))]$$

Theorem

QCTL and MSO are equally expressive.

Decision problems for QCTL [LM14]

Theorem

Under the structure semantics:

- *model checking is* PSPACE*-complete;*
- *satisfiability is undecidable.*

[LM14] Laroussinie, Markey. Quantified CTL: expressiveness and complexity. LMCS, 2014.

Decision problems for QCTL [LM14]

Theorem

Under the structure semantics:

- *model checking is* PSPACE*-complete;*
- *satisfiability is undecidable.*

Under the tree semantics: for formulas with k nested quantifiers,

- *model checking is* k-EXPTIME*-complete;*
- *satisfiability is* $(k+1)$-EXPTIME*-complete.*

[LM14] Laroussinie, Markey. Quantified CTL: expressiveness and complexity. LMCS, 2014.

QCTL with structure semantics

Theorem

Model checking QCTL for the structure semantics is PSPACE-*complete.*

QCTL with structure semantics

Theorem

Model checking QCTL for the structure semantics is PSPACE-*complete.*

Proof

Membership:

Iteratively
- (nondeterministically) pick a labelling,
- check the subformula.

Hardness:

QBF is a special case (without even using temporal modalities).

QCTL with structure semantics

Theorem

Model checking QCTL for the structure semantics is PSPACE-*complete.*

Proof

Membership:

Iteratively
- (nondeterministically) pick a labelling,
- check the subformula.

Hardness:

QBF is a special case (without even using temporal modalities).

Theorem

QCTL satisfiability for the structure semantics is undecidable.

QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is $(k+1)$-*EXPTIME*-complete.*

QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is $(k+1)$-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

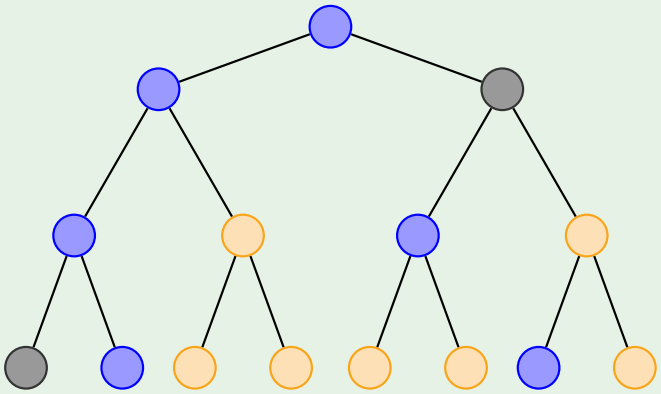
QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

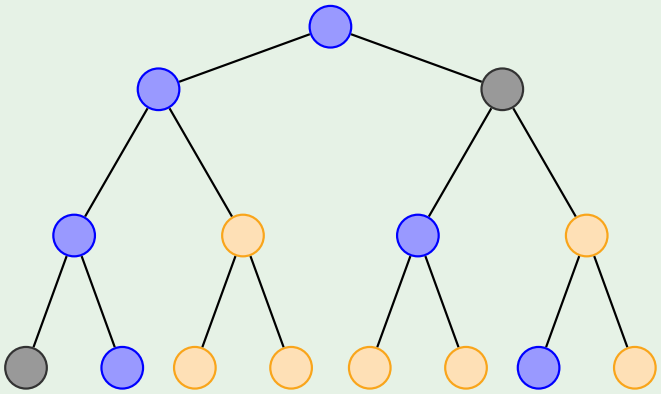
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \bullet) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

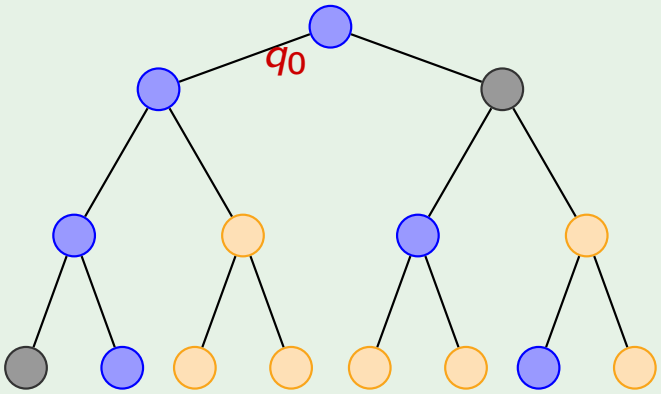
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \bullet) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

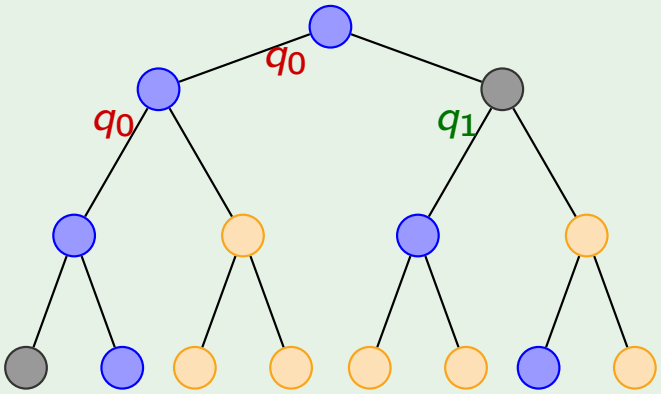
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \bullet) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

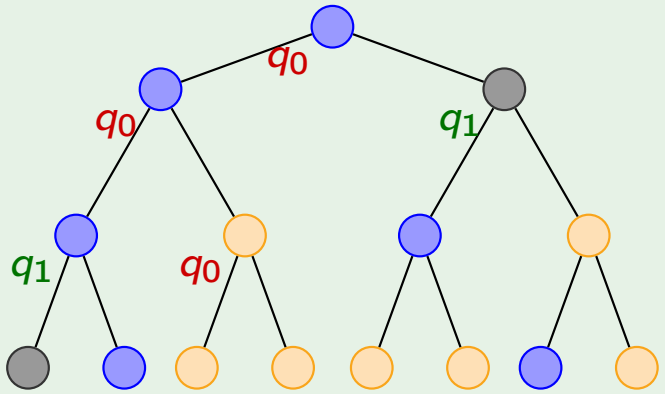
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \bullet) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

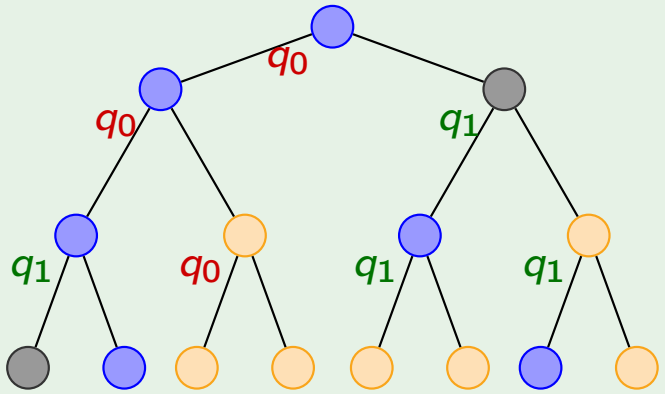
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \bullet) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

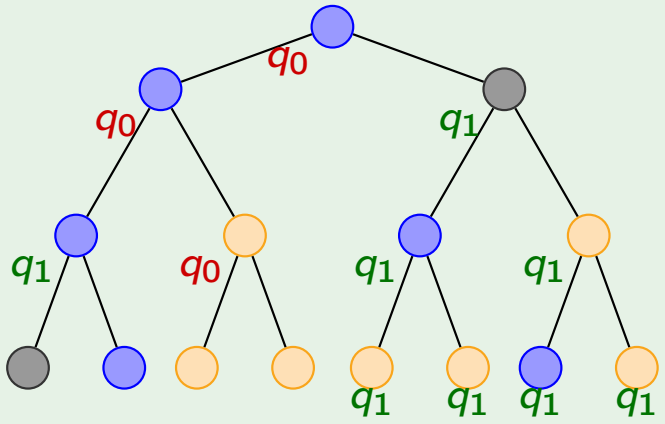
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \circ) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

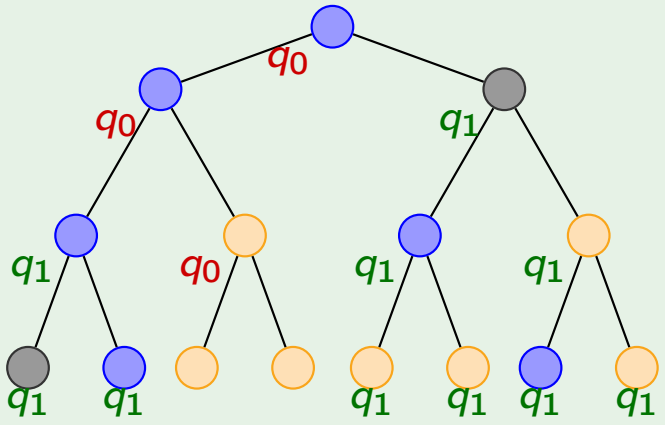
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \circ) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

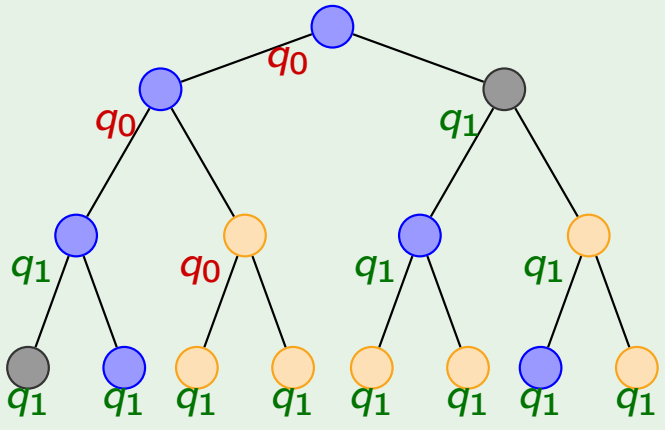
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \bullet) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



QCTL with tree semantics

Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

Using (alternating) parity tree automata:

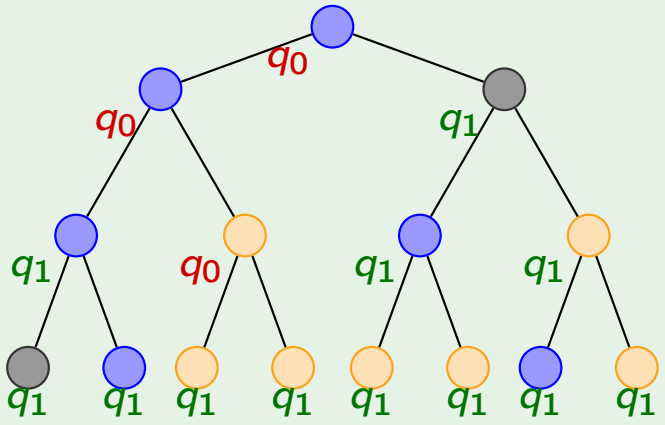
$\delta(q_0, \bullet) = (q_0, q_1) \vee (q_1, q_0)$

$\delta(q_0, \circ) = (q_1, q_1)$

$\delta(q_0, \bullet) = (q_2, q_2)$

$\delta(q_1, \circledast) = (q_1, q_1)$

$\delta(q_2, \circledast) = (q_2, q_2)$



This automaton corresponds to **E** ● **U** ○

QCTL with tree semantics

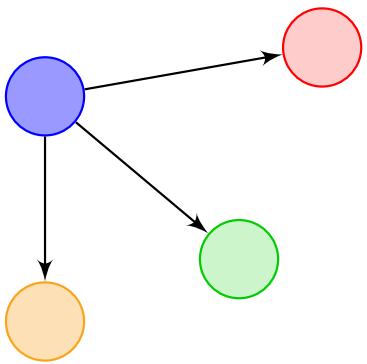
Theorem

- *Model checking QCTL with k quantifiers in the tree semantics is k-*EXPTIME*-complete.*
- *Satisfiability of QCTL with k quantifiers in the tree semantics is (k+1)-*EXPTIME*-complete.*

Proof

- polynomial-size automata for CTL;
- quantification is handled by projection, which first requires removing alternation (exponential blowup);
- an automaton equivalent to a QCTL formula can be built inductively;
- emptiness of an alternating parity tree automaton can be decided in exponential time.

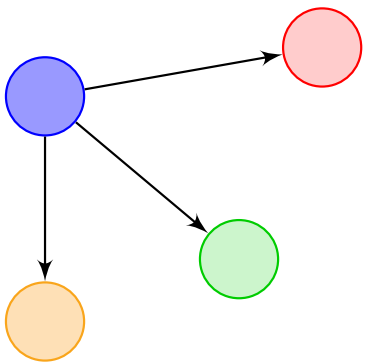
Translating ATL_{sc} into QCTL [DLM12]



- player A has moves m_1^A, ..., m_n^A;
- from the transition table, we can compute the set Next(, A, m_i^A) of states that can be reached from when player A plays m_i^A.

[DLM12] Da Costa, Laroussinie, Markey. Quantified CTL: ... CONCUR, 2012.

Translating ATL_{sc} into QCTL [DLM12]



- player A has moves m_1^A, ..., m_n^A;
- from the transition table, we can compute the set $\mathrm{Next}(\bullet, A, m_i^A)$ of states that can be reached from $\bullet$ when player A plays m_i^A.

$\langle \cdot A \cdot \rangle\, \varphi$ can be encoded as follows:

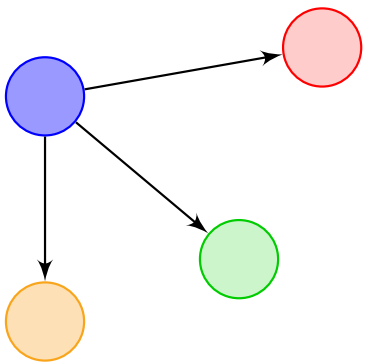
$\exists m_1^A.\ \exists m_2^A \ldots \exists m_n^A.$

- this corresponds to a strategy: $\mathbf{A}\,\mathbf{G}(m_i^A \Leftrightarrow \bigwedge \neg\, m_j^A)$;
- the outcomes all satisfy φ:

$$\mathbf{A}\big[\mathbf{G}(q \wedge m_i^C \Rightarrow \mathbf{X}\,\mathrm{Next}(q, C, m_i^C)) \Rightarrow \varphi\big].$$

[DLM12] Da Costa, Laroussinie, Markey. Quantified CTL: ... CONCUR, 2012.

Translating ATL_{sc} into QCTL [DLM12]



- player A has moves m_1^A, ..., m_n^A;
- from the transition table, we can compute the set Next(●, A, m_i^A) of states that can be reached from ● when player A plays m_i^A.

Corollary

*ATL_{sc} model checking is decidable (*TOWER*-complete).*

[DLM12] Da Costa, Laroussinie, Markey. Quantified CTL: ... CONCUR, 2012.

Outline of the presentation

Strategy Logic [CHP07,MMV10]

Strategy logic

Explicit quantification over strategies + strategy assignement:

$$\exists \sigma_A.\ \forall \sigma_B.\mathsf{assign}(A \to \sigma_A, B \to \sigma_B).\ \varphi$$

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.
[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

Strategy Logic [CHP07,MMV10]

Strategy logic

Explicit quantification over strategies + strategy assignement:

$$\underline{\exists \sigma_A.\ \forall \sigma_B}.\underline{\mathsf{assign}(A \to \sigma_A, B \to \sigma_B)}.\ \underline{\varphi}$$

Quantification over strategies

Assignment of strategies to players

Path property

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.

[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

Strategy Logic [CHP07,MMV10]

Strategy logic

Explicit quantification over strategies + strategy assignement:

$$\underline{\exists \sigma_A.\ \forall \sigma_B}.\underline{\mathsf{assign}(A \to \sigma_A, B \to \sigma_B)}.\ \underline{\varphi}$$

Quantification over strategies

Assignment of strategies to players

Path property

Theorem

SL is at least as expressive as ATL_{sc}.

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.

[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

Strategy Logic [CHP07,MMV10]

Strategy logic

Explicit quantification over strategies + strategy assignement:

$$\underline{\exists \sigma_A.\ \forall \sigma_B}.\underline{\mathsf{assign}(A \to \sigma_A, B \to \sigma_B)}.\ \underline{\varphi}$$

Quantification over strategies

Assignment of strategies to players

Path property

Theorem

SL is at least as expressive as ATL_{sc}.

Theorem

*SL model-checking is decidable (*TOWER*-complete).*

[CHP07] Chatterjee, Henzinger, Piterman. Strategy Logic. CONCUR, 2007.
[MMV10] Mogavero, Murano, Vardi. Reasoning about strategies. FSTTCS, 2010.

Outline of the presentation

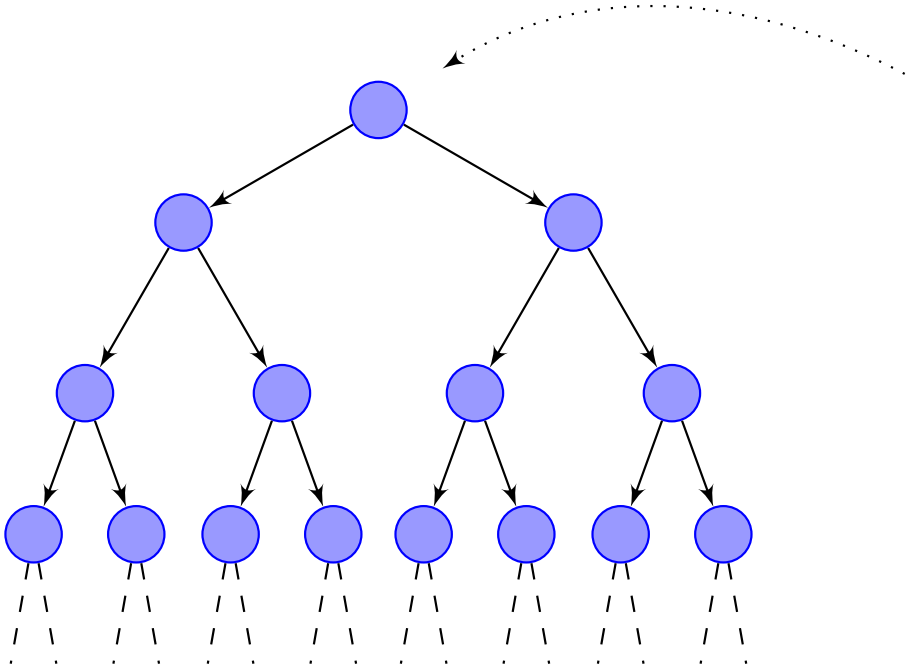
What is the exact semantics? [BGM16]

$$\exists \sigma.\ \mathbf{AG}(\mathsf{assign}(A \to \sigma).\ \varphi)$$

[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

What is the exact semantics? [BGM16]

$$\exists\sigma.\ \mathbf{AG}(\text{assign}(A \to \sigma).\ \varphi)$$

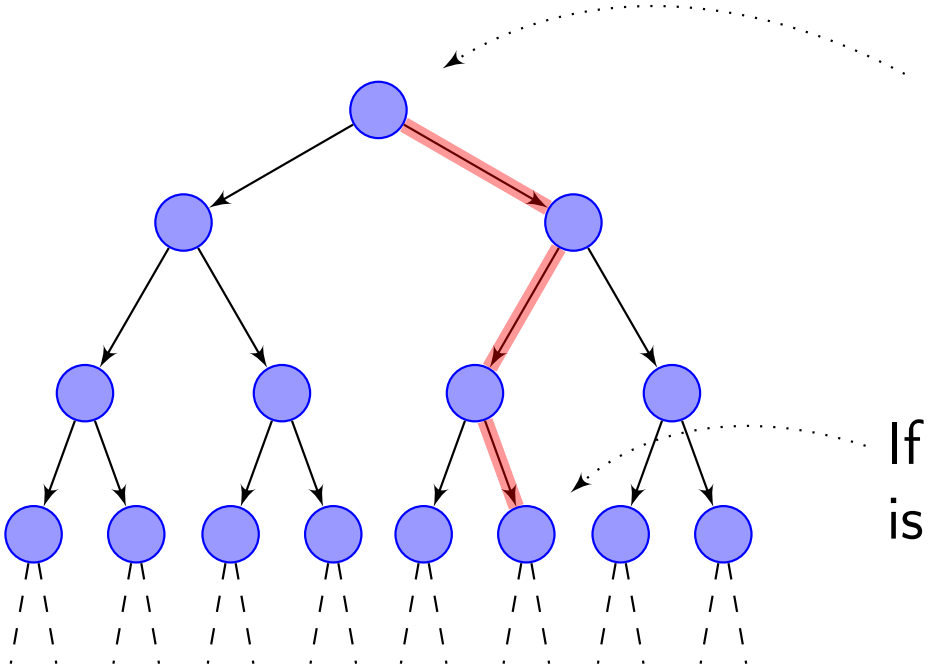


Strategy σ is selected here

[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

What is the exact semantics? [BGM16]

$$\exists\sigma.\ \mathbf{AG}(\text{assign}(A \to \sigma).\ \varphi)$$



Strategy σ is selected here

If strategy σ is assigned here,
is the prefix part of the history?

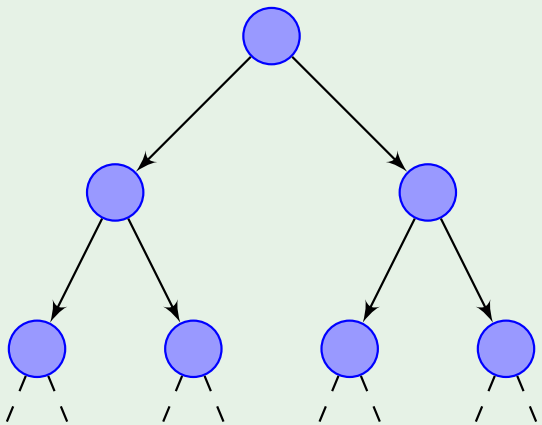
[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

What is the exact semantics? [BGM16]

$$\exists \sigma.\ \mathbf{AG}(\text{assign}(A \to \sigma).\ \varphi)$$

Example

If σ prescribes to alternate between m_0 and m_1, starting with m_0:



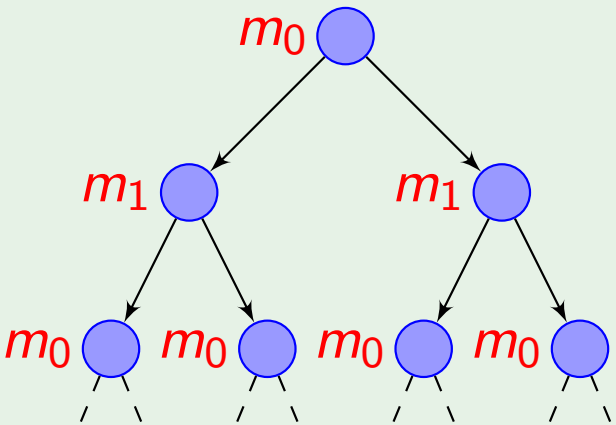
[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

What is the exact semantics? [BGM16]

$$\exists\sigma.\ \mathbf{AG}(\text{assign}(A \to \sigma).\ \varphi)$$

Example

If σ prescribes to alternate between m_0 and m_1, starting with m_0:



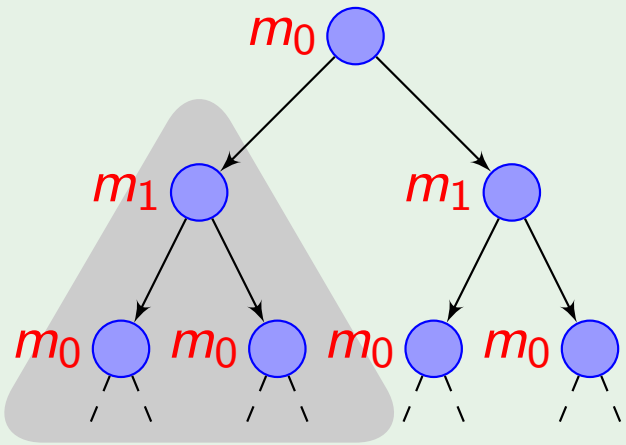
[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

What is the exact semantics? [BGM16]

$$\exists\sigma.\ \mathbf{AG}(\text{assign}(A \to \sigma).\ \varphi)$$

Example

If σ prescribes to alternate between m_0 and m_1, starting with m_0:



History starts when strategy is selected.

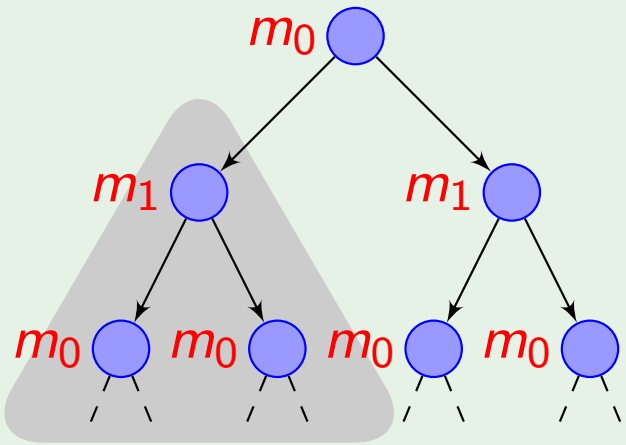
[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

What is the exact semantics? [BGM16]

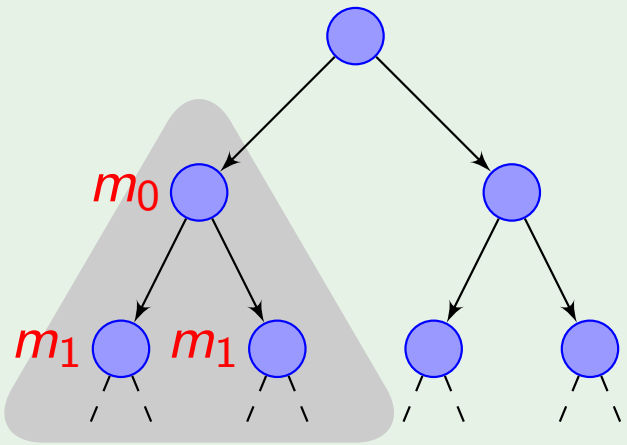
$$\exists\sigma.\ \mathbf{AG}(\text{assign}(A \to \sigma).\ \varphi)$$

Example

If σ prescribes to alternate between m_0 and m_1, starting with m_0:



History starts when strategy is selected.



History starts when strategy is assigned.

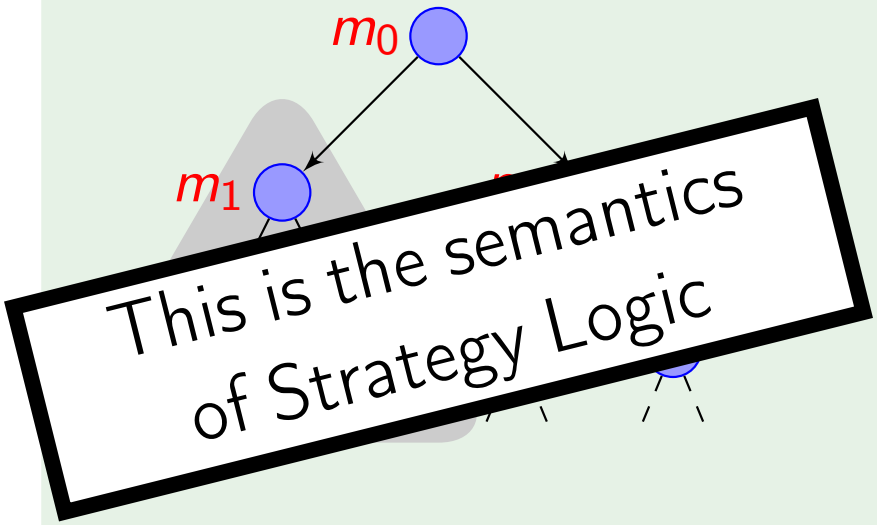
[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

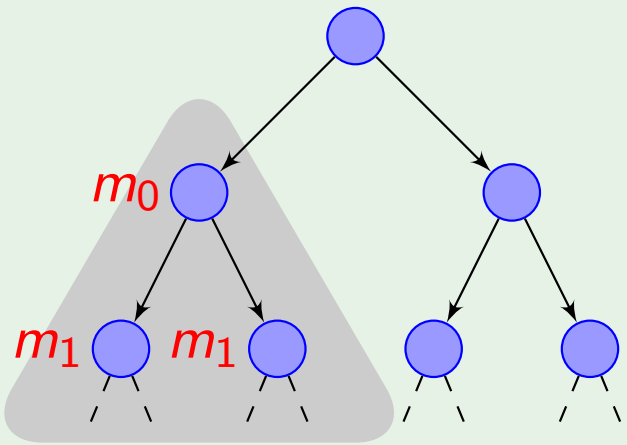
What is the exact semantics? [BGM16]

$$\exists\sigma.\ \mathbf{AG}(\text{assign}(A \to \sigma).\ \varphi)$$

Example

If σ prescribes to alternate between m_0 and m_1, starting with m_0:





History starts when strategy is selected.

History starts when strategy is assigned.

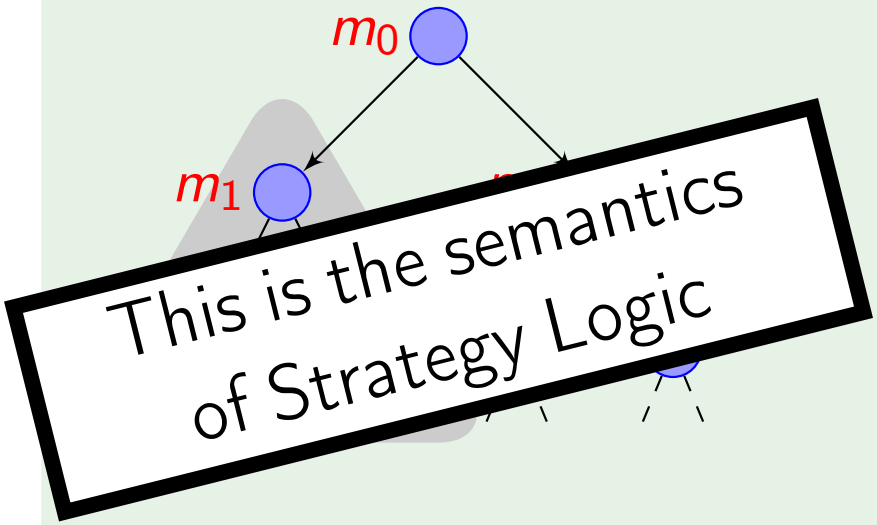
[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

What is the exact semantics? [BGM16]

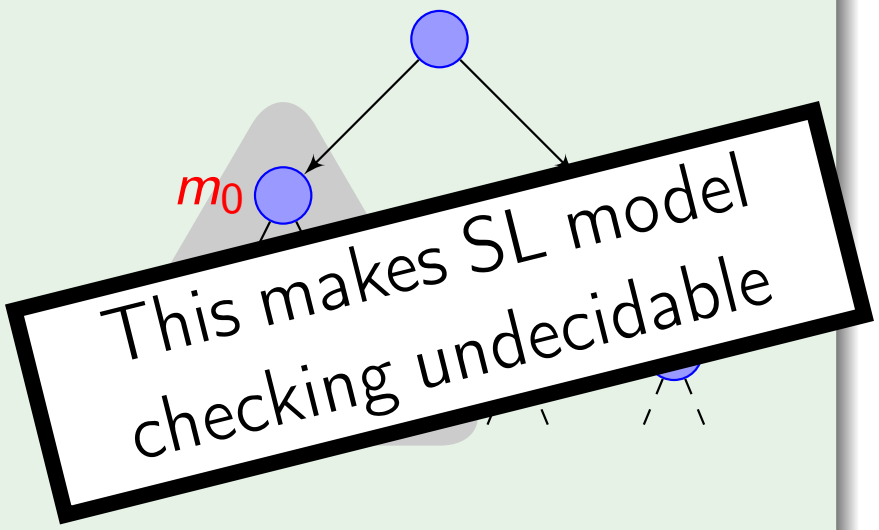
$$\exists\sigma.\ \mathbf{AG}(\text{assign}(A \to \sigma).\ \varphi)$$

Example

If σ prescribes to alternate between m_0 and m_1, starting with m_0:



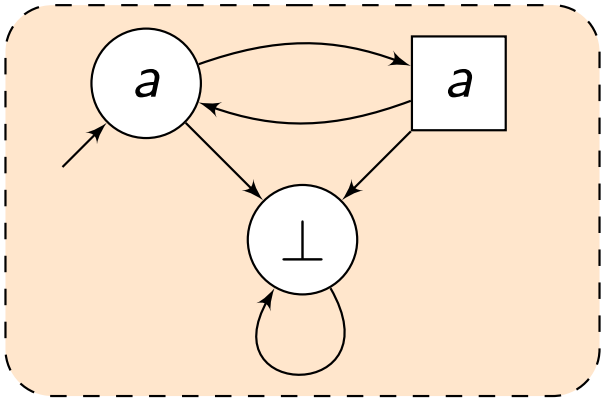
History starts when strategy is selected.



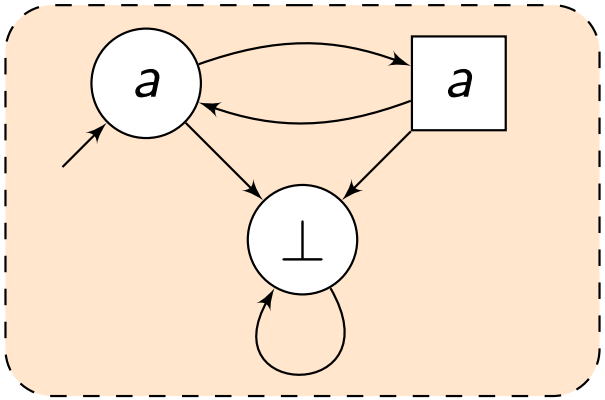
History starts when strategy is assigned.

[BGM16] Bouyer, Gardy, Markey. On the semantics of SL. IPL, 2016.

Encoding of a 2-counter machine

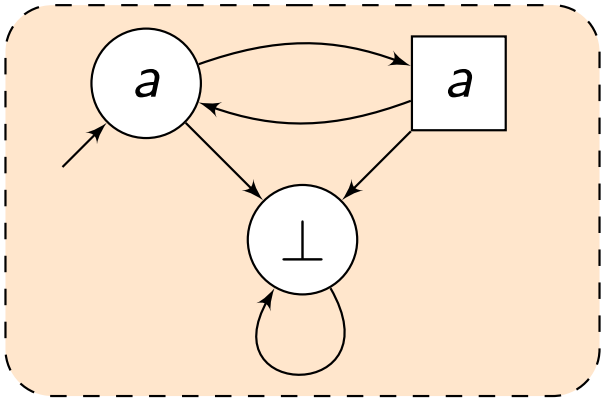


Encoding of a 2-counter machine



Strategy $\sigma_{\bigcirc}$ encodes counter value k if it plays k times to $\boxed{a}$ before playing to $\perp$ (similarly for $\sigma_{\square}$).

Encoding of a 2-counter machine



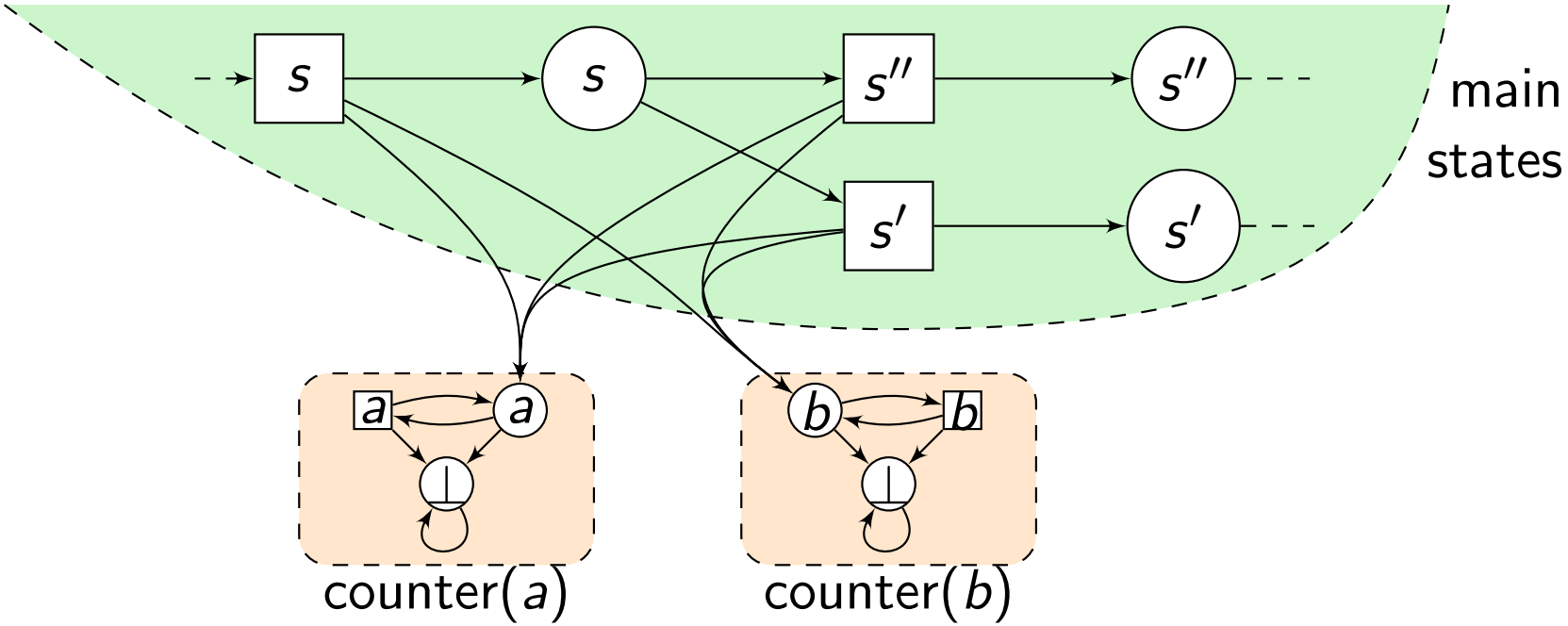
Strategy $\sigma_{\bigcirc}$ encodes counter value k if it plays k times to $\boxed{a}$ before playing to $\bot$ (similarly for $\sigma_{\square}$).

Lemma

Two strategies $\sigma_{\bigcirc}$ and $\sigma_{\square}$ encode the same counter value iff, when entering the game above and applying these strategies, it holds

$$\mathbf{G}(\boxed{a} \Rightarrow \mathbf{X}\,a) \wedge \mathbf{F}\left[(a \wedge \mathbf{X}\,\bot) \wedge \neg\langle\cdot\bigcirc\cdot\rangle\,\neg\,\mathbf{X}\,\mathbf{X}\,\bot\right] \qquad (\varphi_{=})$$

Encoding of a 2-counter machine



- A strategy for ○ encodes a run of the 2-counter machine;
- At each step, Player □ can check correct values of counters (this is where we use our special semantics).

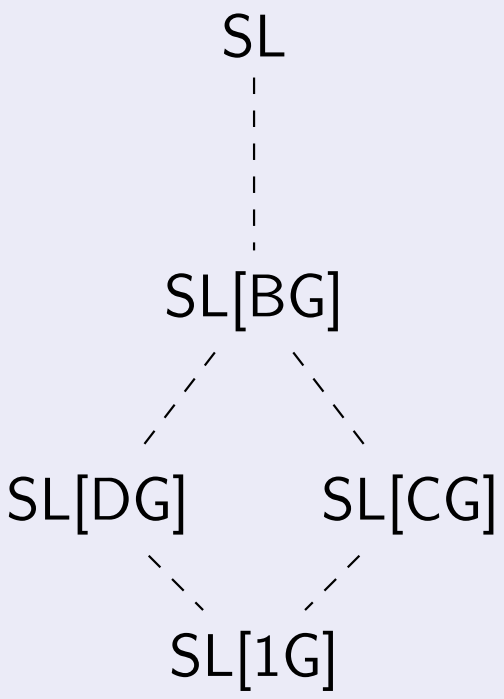
Strategy dependences (ongoing work) [MMPV11,GBM17]

[MMPV11] Mogavero, Murano, Perelli, Vardi. Reasonning about strategies: ... J. ACM, 2011.
[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Strategy dependences (ongoing work) [MMPV11,GBM17]

Note

From this point on, we only consider formulas in SL[BG]:



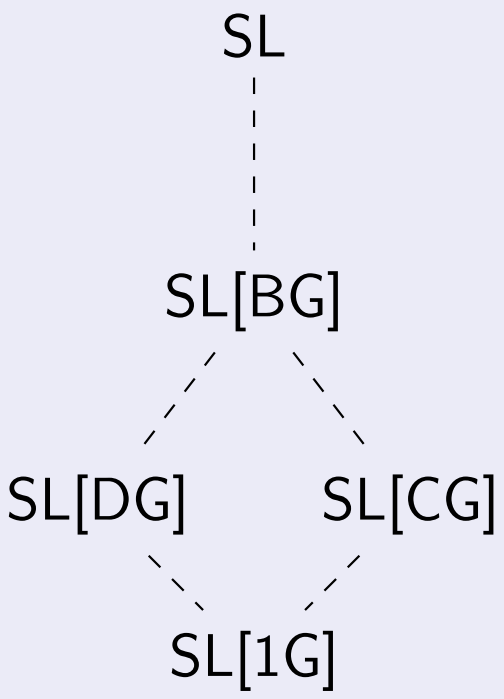
$$\underbrace{(Q_i\sigma_i).}_{\text{quantifier block}}\ \underbrace{\mathsf{assign}(A_j \mapsto \sigma_i).\ \varphi_{\mathsf{LTL}}}_{\text{goal}}$$

[MMPV11] Mogavero, Murano, Perelli, Vardi. Reasonning about strategies: ... J. ACM, 2011.
[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Strategy dependences (ongoing work) [MMPV11,GBM17]

Note

From this point on, we only consider formulas in SL[BG]:



disjunction/conjunction of goals

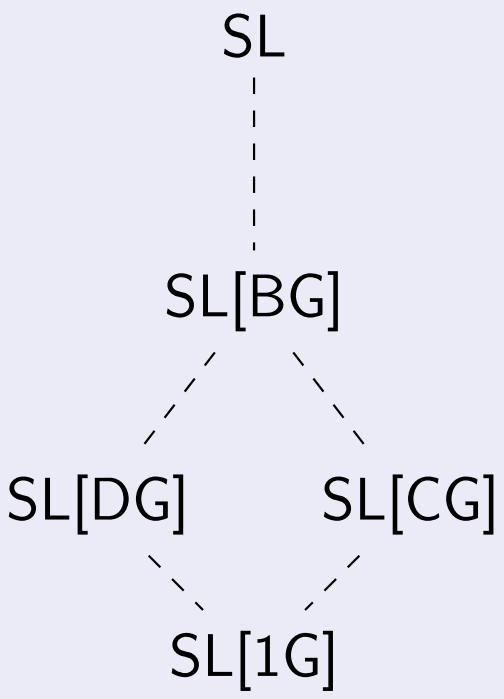
$$\underbrace{(Q_i\sigma_i).}_{\text{quantifier block}}\ \underbrace{\text{assign}(A_j \mapsto \sigma_i).\ \varphi_{\text{LTL}}}_{\text{goal}}$$

[MMPV11] Mogavero, Murano, Perelli, Vardi. Reasonning about strategies: ... J. ACM, 2011.
[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Strategy dependences (ongoing work) [MMPV11,GBM17]

Note

From this point on, we only consider formulas in SL[BG]:



boolean combination of goals

disjunction/conjunction of goals

$$\underbrace{(Q_i\sigma_i).}_{\text{quantifier block}}\ \underbrace{\mathsf{assign}(A_j \mapsto \sigma_i).\ \varphi_{\mathsf{LTL}}}_{\text{goal}}$$

[MMPV11] Mogavero, Murano, Perelli, Vardi. Reasonning about strategies: ... J. ACM, 2011.

[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

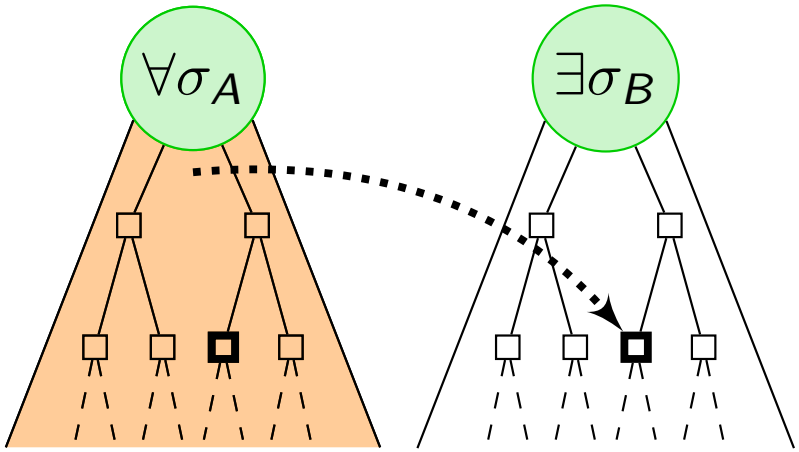
Strategy dependences (ongoing work) [MMPV11,GBM17]

$$\forall \sigma_A \exists \sigma_B.\ \mathsf{assign}(A \mapsto \sigma_A; B \mapsto \sigma_B).\ \varphi$$

[MMPV11] Mogavero, Murano, Perelli, Vardi. Reasonning about strategies: ... J. ACM, 2011.
[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Strategy dependences (ongoing work) [MMPV11,GBM17]

$$\forall \sigma_A \exists \sigma_B.\ \mathsf{assign}(A \mapsto \sigma_A; B \mapsto \sigma_B).\ \varphi$$

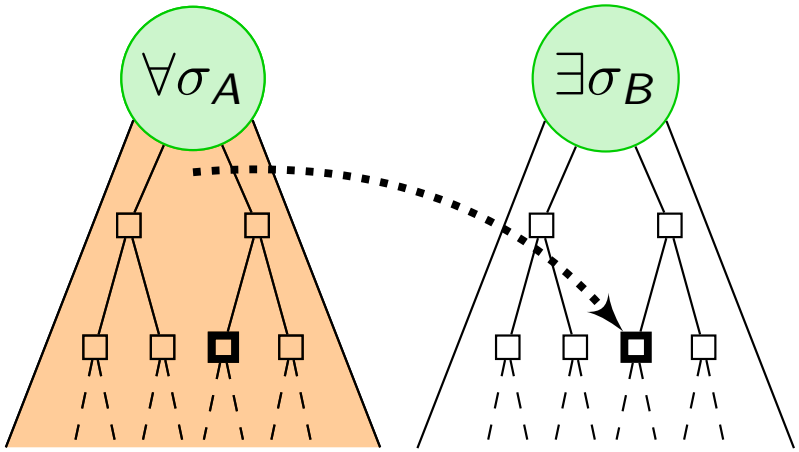


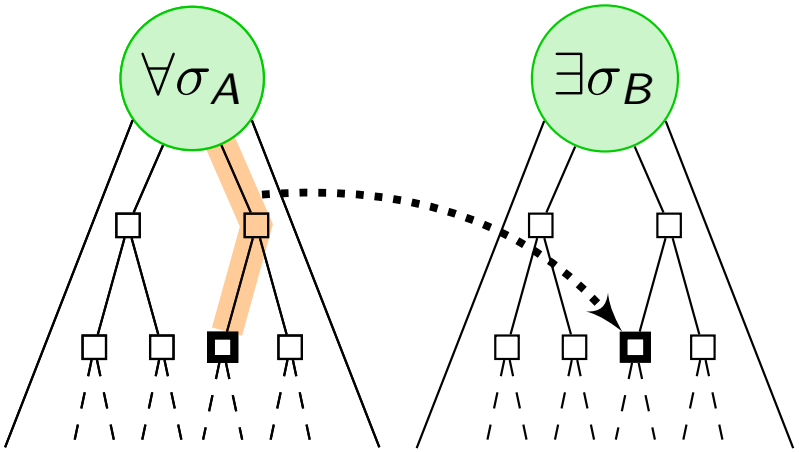
Strategy σ_B at node n may depend on the whole strategy σ_A.

[MMPV11] Mogavero, Murano, Perelli, Vardi. Reasonning about strategies: ... J. ACM, 2011.
[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Strategy dependences (ongoing work) [MMPV11,GBM17]

$$\forall \sigma_A \exists \sigma_B.\ \mathsf{assign}(A \mapsto \sigma_A; B \mapsto \sigma_B).\ \varphi$$





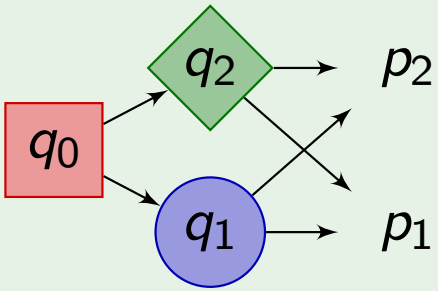
Strategy σ_B at node n may depend on the whole strategy σ_A.

Strategy σ_B at node n only depends on σ_A along the branch to n.

[MMPV11] Mogavero, Murano, Perelli, Vardi. Reasonning about strategies: ... J. ACM, 2011.
[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Dependence maps (Skolemization)

Example

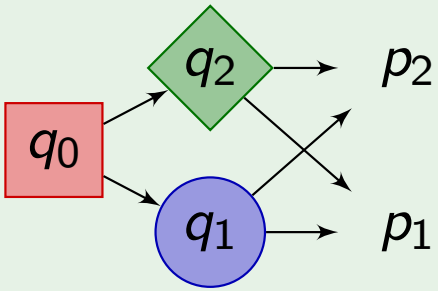


$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_2 \end{cases}$$

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_2 \end{cases}$$

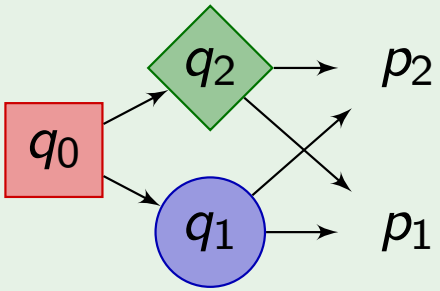
Dependence maps

Dependence maps select $\exists$-strategies from $\forall$-strategies:

$$\begin{array}{rcl} \theta\colon (\mathcal{V}^{\forall} \to \text{Strats}) & \to & (\mathcal{V}^{\exists} \to \text{Strats}) \\ w & \mapsto & \theta(w) \end{array}$$

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_2 \end{cases}$$

Dependence maps

Dependence maps select $\exists$-strategies from $\forall$-strategies:

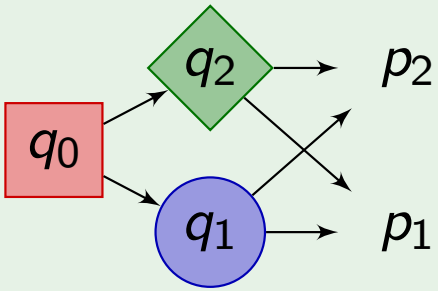
$$\begin{array}{rcl} \theta\colon (\mathcal{V}^{\forall} \to \text{Strats}) & \to & (\mathcal{V}^{\exists} \to \text{Strats}) \\ w & \mapsto & \theta(w) \end{array}$$

Then

$$q_0 \models (Q_i \sigma_i).\ \Phi \qquad \Leftrightarrow \qquad \exists \theta.\ \forall w.\ q_0 \models_{\theta(w) \cup w} \Phi.$$

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\, p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\, p_2 \end{cases}$$

Dependence maps

Dependence maps select $\exists$-strategies from $\forall$-strategies:

$$\begin{array}{rcl} \theta\colon (\mathcal{V}^{\forall} \to \text{Strats}) & \to & (\mathcal{V}^{\exists} \to \text{Strats}) \\ w & \mapsto & \theta(w) \end{array}$$

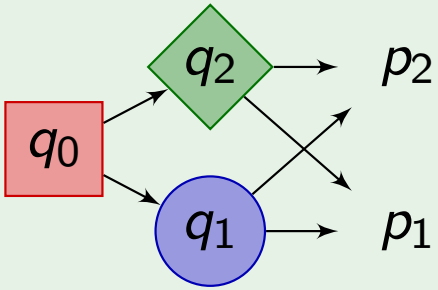
Then

$$q_0 \models (Q_i \sigma_i).\ \Phi \qquad \Leftrightarrow \qquad \exists \theta.\ \forall w.\ q_0 \models_{\theta(w) \cup w} \Phi.$$

$q_0 \not\models \varphi$ is (a priori) not equivalent to $q_0 \models \neg \varphi$.

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_2 \end{cases}$$

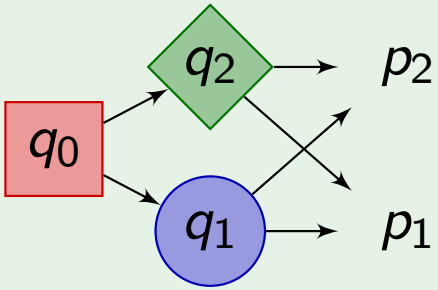
Dependence maps

- classical SL semantics ($\models^C$):
  - only depend on previously-quantified variables:

$$\text{if } w(y) = w'(y), \text{ then } \theta(w)(z) = \theta(w')(z)$$

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_2 \end{cases}$$

Dependence maps

- classical SL semantics ($\models^C$):
  - only depend on previously-quantified variables:

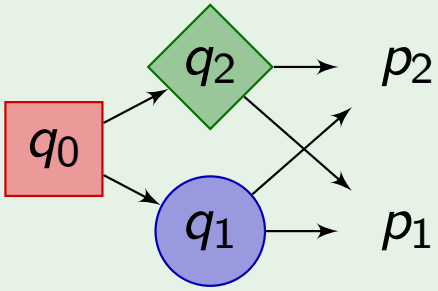
$$\text{if } w(y) = w'(y), \text{ then } \theta(w)(z) = \theta(w')(z)$$

- elementary semantics ($\models^E$):
  - only depend on previously-quantified variables along current prefix:

$$\text{for all } n, \text{ if } w(y)(m) = w'(y)(m) \text{ for all } m \preceq n, \text{ then } \theta(w)(z)(n) = \theta(w')(z)(n)$$

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_2 \end{cases}$$

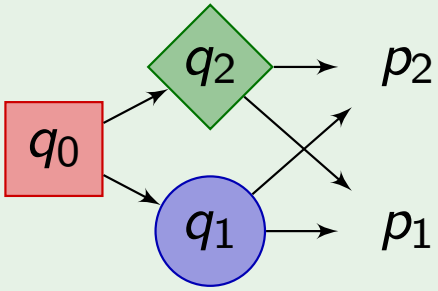
Theorem

For any $\varphi \in SL[BG]$:

$$q_0 \models^C \varphi \ \Leftrightarrow\ q_0 \models \varphi \qquad\qquad q_0 \models^E \varphi \ \Rightarrow\ q_0 \models^C \varphi$$

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_2 \end{cases}$$

- $q_0 \models^C \varphi$: z plays to p_2 iff y plays to p_1.

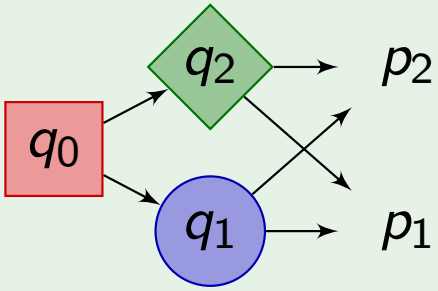
Theorem

For any $\varphi \in SL[BG]$:

$$q_0 \models^C \varphi \ \Leftrightarrow\ q_0 \models \varphi \qquad\qquad q_0 \models^E \varphi \ \Rightarrow\ q_0 \models^C \varphi$$

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\ p_2 \end{cases}$$

- $q_0 \models^C \varphi$: z plays to p_2 iff y plays to p_1.
- $q_0 \not\models^E \varphi$: y only plays in q_1, so z will not depend on y.

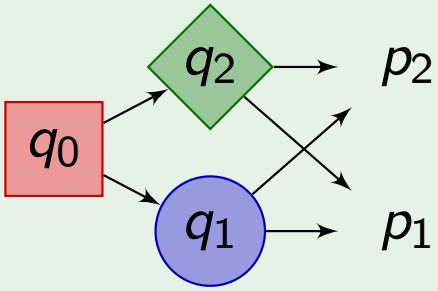
Theorem

For any $\varphi \in SL[BG]$:

$$q_0 \models^C \varphi \ \Leftrightarrow\ q_0 \models \varphi \qquad\qquad q_0 \models^E \varphi \ \Rightarrow\ q_0 \models^C \varphi$$

Dependence maps (Skolemization)

Example



$$\varphi = \forall y.\exists z.\forall x_A.\forall x_B.$$

$$\bigvee \begin{cases} \text{assign}(\square \mapsto x_A; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\, p_1 \\ \text{assign}(\square \mapsto x_B; \bigcirc \mapsto y; \diamond \mapsto z).\ \mathbf{F}\, p_2 \end{cases}$$

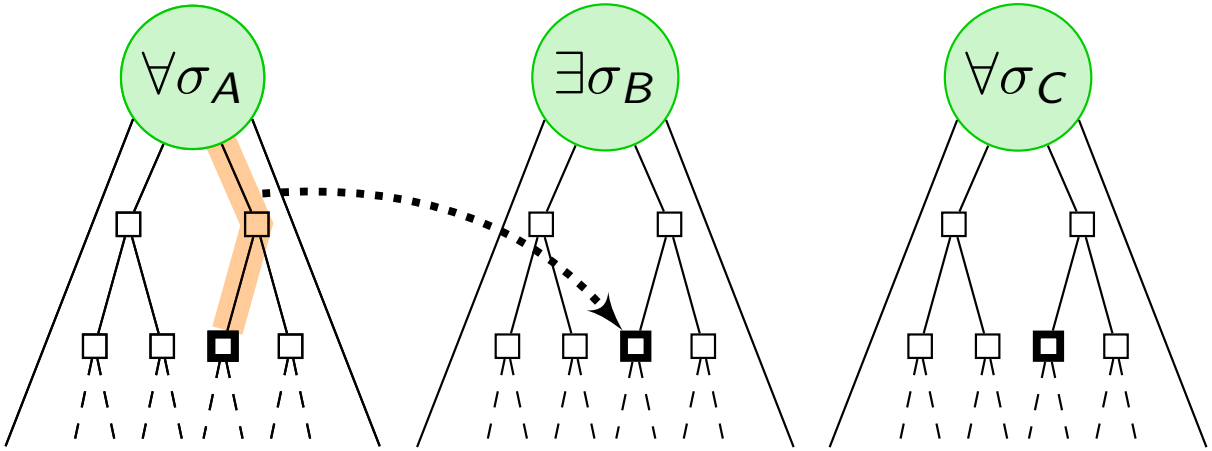
- $q_0 \models^C \varphi$: z plays to p_2 iff y plays to p_1.
- $q_0 \not\models^E \varphi$: y only plays in q_1, so z will not depend on y.
- $q_0 \not\models^E \neg\varphi$: because otherwise $q_0 \models^C \neg\varphi$...

Theorem

For any $\varphi \in SL[BG]$:

$$q_0 \models^C \varphi \;\Leftrightarrow\; q_0 \models \varphi \qquad\qquad q_0 \models^E \varphi \;\Rightarrow\; q_0 \models^C \varphi$$

Beyond elementary dependences [GBM17]

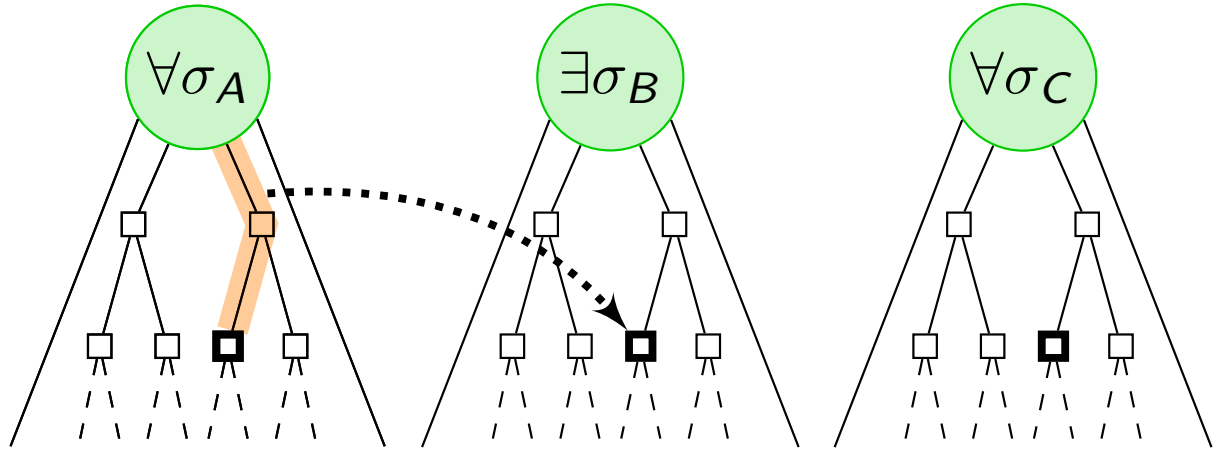


Elementary semantics

Strategy σ_B at node n only depends on σ_A along the branch to n.

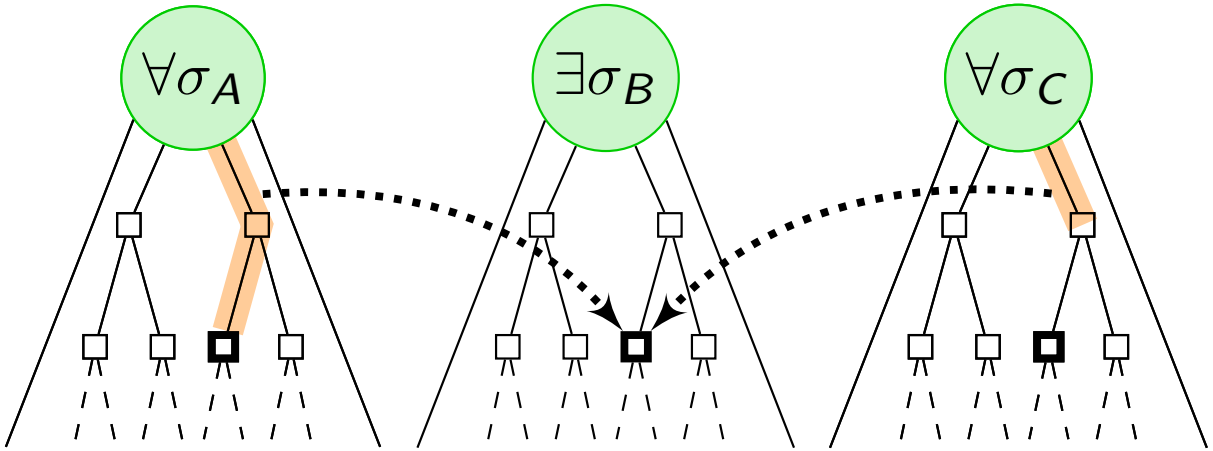
[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Beyond elementary dependences [GBM17]



Elementary semantics

Strategy σ_B at node n only depends on σ_A along the branch to n.

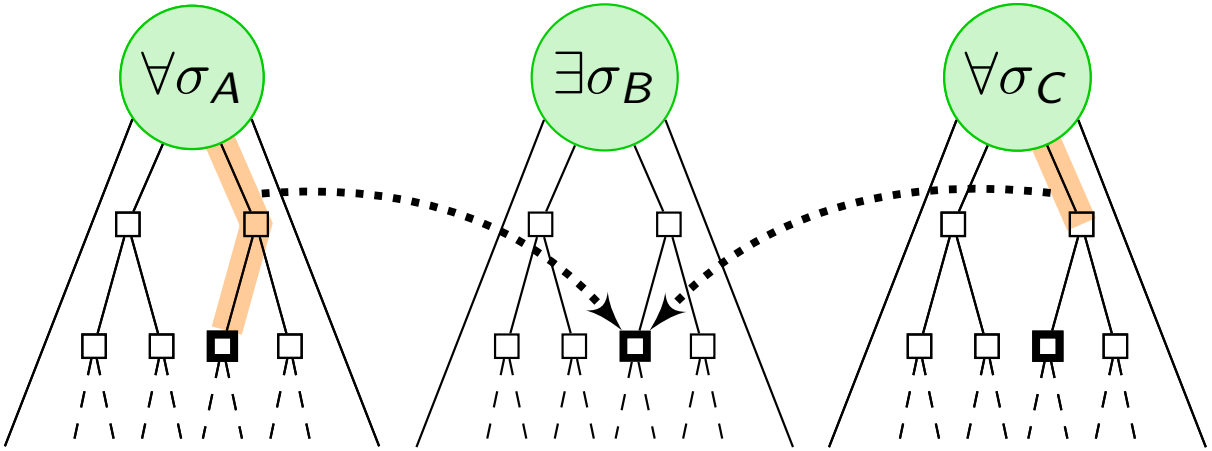


Timeline semantics

Strategy σ_B also depends on σ_C along strict prefixes of n.

[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Beyond elementary dependences [GBM17]



Timeline semantics

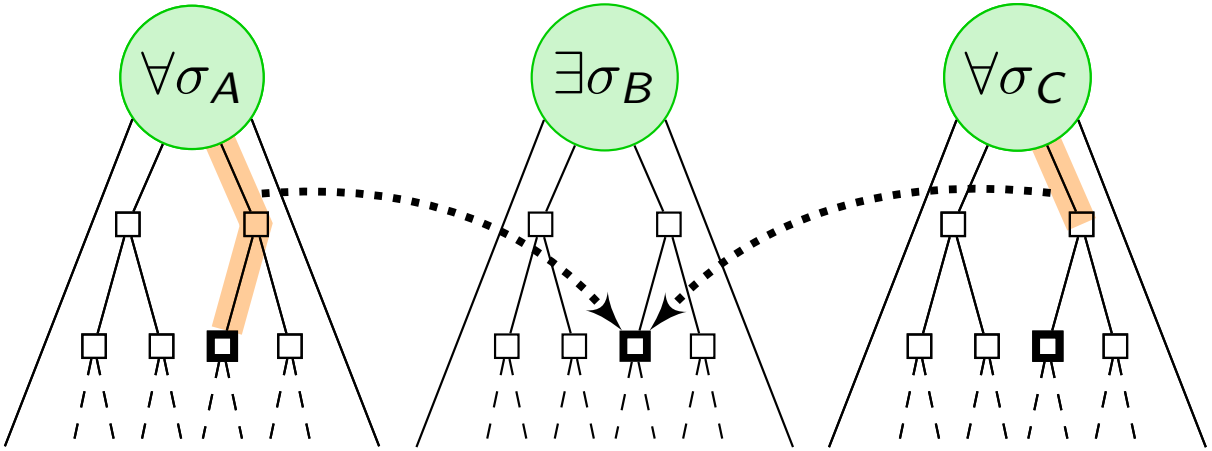
Strategy σ_B also depends on σ_C along strict prefixes of n.

Theorem

For any $\varphi \in SL[BG]$, $\quad q_0 \models^E \varphi \;\Rightarrow\; q_0 \models^T \varphi$

[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Beyond elementary dependences [GBM17]



Timeline semantics

Strategy σ_B also depends on σ_C along strict prefixes of n.

Theorem

For any $\varphi \in SL[BG]$, $\quad q_0 \models^E \varphi \;\Rightarrow\; q_0 \models^T \varphi$

Theorem

For a large fragment of SL[BG] (containing SL[CG] and SL[DG]),

$$q_0 \models^T \varphi \;\Leftrightarrow\; q_0 \not\models^T \neg\varphi.$$

*Model checking this fragment is 2*EXPTIME*-complete (for $\models^T$).*

[GBM17] Gardy, Bouyer, Markey. Dependences in Strategy Logic. Submitted (arXiv 1708.05849).

Outline of the presentation

Conclusions and future works

Conclusions

- Temporal logics have been successful for model checking "closed" systems;
- For multi-agent systems, ATL is not expressive enough;
- But ATL_{sc} and Strategy Logic are too expressive:
  - model checking is very hard;
  - does the semantics correspond to what we expect/need?

Conclusions and future works

Conclusions

- Temporal logics have been successful for model checking "closed" systems;
- For multi-agent systems, ATL is not expressive enough;
- But ATL_{sc} and Strategy Logic are too expressive:
  - model checking is very hard;
  - does the semantics correspond to what we expect/need?

Conclusions

- Many open questions:
  - better understand the semantics
  - identify expressive fragments with efficient algorithms
  - extend with stochastic strategies, quantities, partial observation...