Quantified-CTL model checking

Arnaud Da Costa[1], François Laroussinie[2], Nicolas Markey[1]

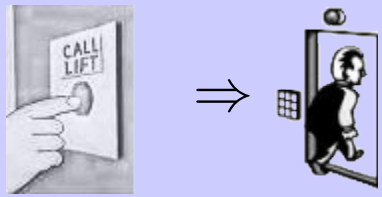
[1] LSV, CNRS & ENS Cachan, France
[2] LIAFA, CNRS & Univ. Paris7-Diderot, France

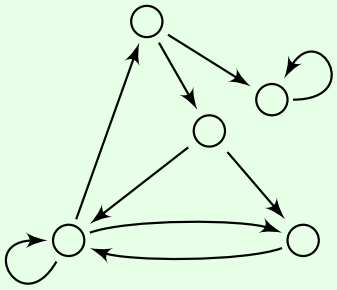
September 10, 2011

Model checking

system:

property:





model-checking algorithm

$\mathbf{G}(\texttt{request} \Rightarrow \mathbf{F}\,\texttt{grant})$

yes/no

Computation-Tree Logic (CTL)

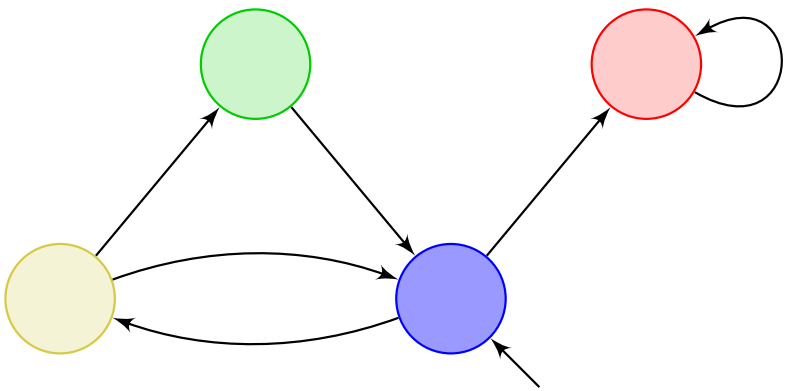
Definition

$$\text{CTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{E}\,\mathbf{X}\,\varphi \mid \mathbf{E}\,\mathbf{G}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi$$

Computation-Tree Logic (CTL)

Definition

$$\text{CTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{E}\,\mathbf{X}\,\varphi \mid \mathbf{E}\,\mathbf{G}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi$$

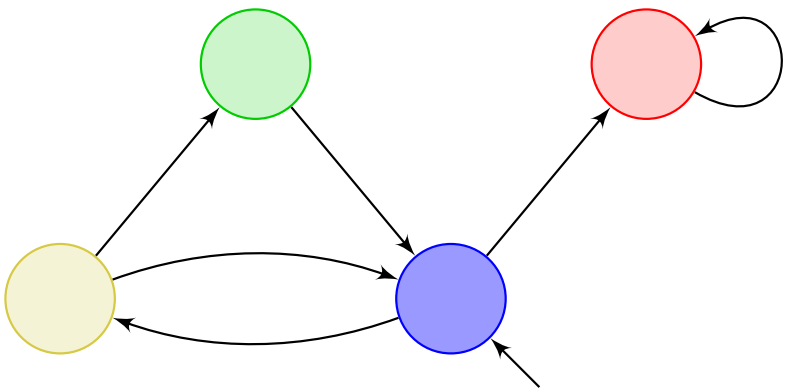


✓ $\mathbf{E}(\texttt{true}\ \mathbf{U}\ \bigcirc) \equiv \mathbf{E}\,\mathbf{F}\,\bigcirc$

Computation-Tree Logic (CTL)

Definition

$$\text{CTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{E}\,\mathbf{X}\,\varphi \mid \mathbf{E}\,\mathbf{G}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi$$

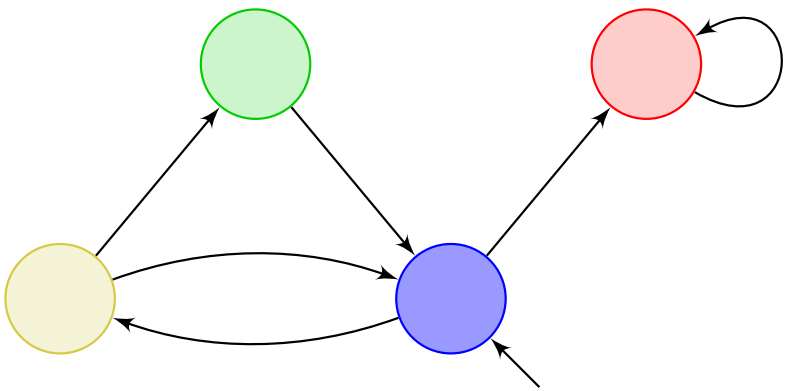


- ✓ $\mathbf{E}(\texttt{true}\ \mathbf{U}\ \bigcirc) \equiv \mathbf{E}\,\mathbf{F}\,\bigcirc$
- ✓ $\mathbf{E}\,\mathbf{G}\,\neg\,\bigcirc \equiv \neg(\mathbf{A}\,\mathbf{F}\,\neg\,\bigcirc)$

Computation-Tree Logic (CTL)

Definition

$$\text{CTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{E}\,\mathbf{X}\,\varphi \mid \mathbf{E}\,\mathbf{G}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi$$

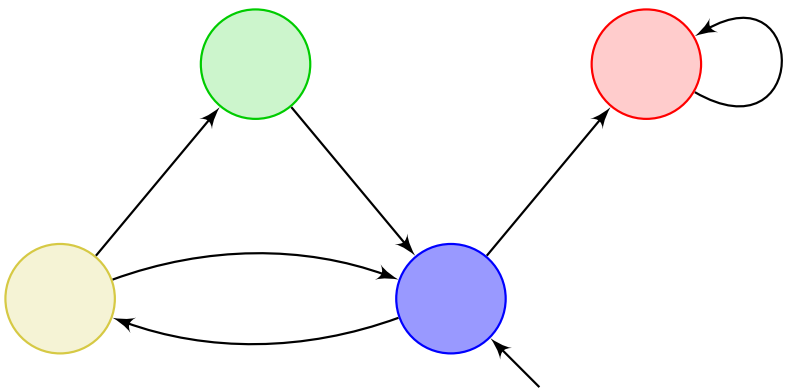


- ✓ $\mathbf{E}(\texttt{true}\ \mathbf{U}\ \bigcirc) \equiv \mathbf{E}\,\mathbf{F}\,\bigcirc$
- ✓ $\mathbf{E}\,\mathbf{G}\,\neg\,\bigcirc \equiv \neg(\mathbf{A}\,\mathbf{F}\,\neg\,\bigcirc)$
- × $\mathbf{E}(\neg\,\bigcirc\ \mathbf{U}\ \bigcirc)$

Computation-Tree Logic (CTL)

Definition

$$\text{CTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{EG}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi$$

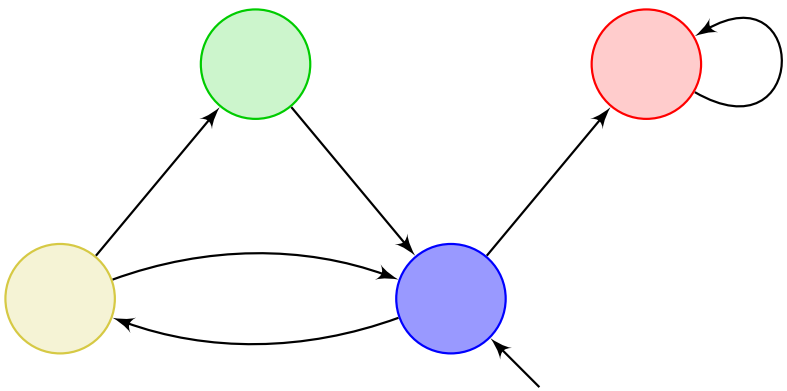


- ✓ $\mathbf{E}(\texttt{true}\ \mathbf{U}\ \bigcirc) \equiv \mathbf{EF}\,\bigcirc$
- ✓ $\mathbf{EG}\,\neg\bigcirc \equiv \neg(\mathbf{AF}\,\neg\bigcirc)$
- × $\mathbf{E}(\neg\bigcirc\ \mathbf{U}\ \bigcirc)$
- ✓ $\mathbf{EG}(\neg\bigcirc \wedge \mathbf{EF}\,\bigcirc)$

Computation-Tree Logic (CTL)

Definition

$$\text{CTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{EG}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi$$



✓ $\mathbf{E}(\texttt{true}\ \mathbf{U}\ \bigcirc) \equiv \mathbf{EF}\,\bigcirc$

✓ $\mathbf{EG}\,\neg\bigcirc \equiv \neg(\mathbf{AF}\,\neg\bigcirc)$

× $\mathbf{E}(\neg\bigcirc\ \mathbf{U}\ \bigcirc)$

✓ $\mathbf{EG}(\neg\bigcirc \wedge \mathbf{EF}\bigcirc)$

Theorem

CTL model checking is PTIME*-complete.*

Quantified-CTL

Definition

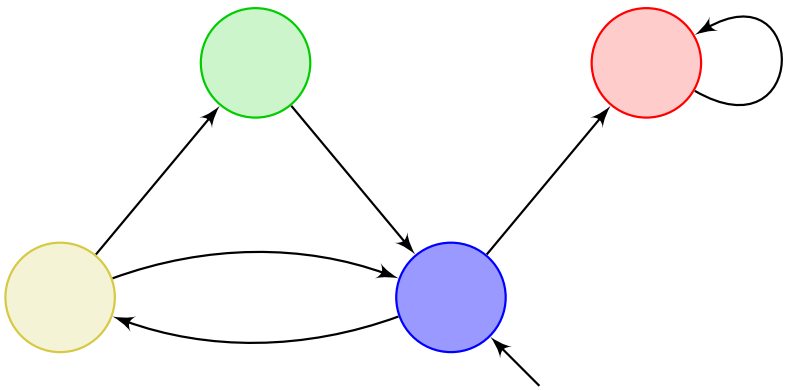
$$\text{QCTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg \varphi \mid \mathbf{E}\,\mathbf{X}\,\varphi \mid \mathbf{E}\,\mathbf{G}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi \mid \exists \bigcirc.\ \varphi$$

Quantified-CTL

Definition

QCTL $\ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg \varphi \mid \mathbf{E}\,\mathbf{X}\,\varphi \mid \mathbf{E}\,\mathbf{G}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi \mid \exists \bigcirc.\ \varphi$

$\exists \bigcirc.\ \mathbf{E}\,\mathbf{G}\,\bigcirc$

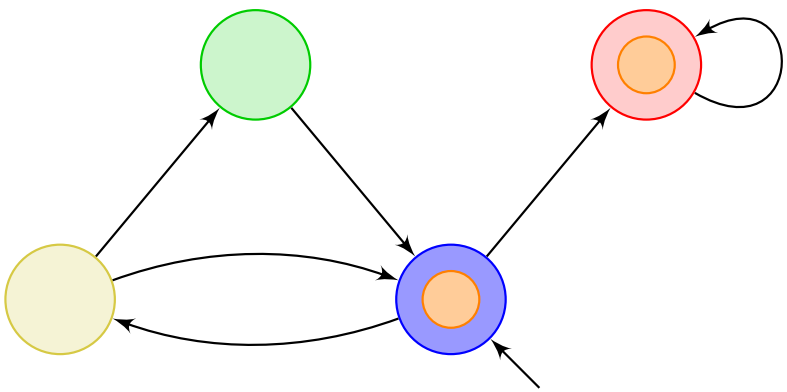


Quantified-CTL

Definition

QCTL $\ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg \varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{EG}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi \mid \exists \bigcirc.\ \varphi$

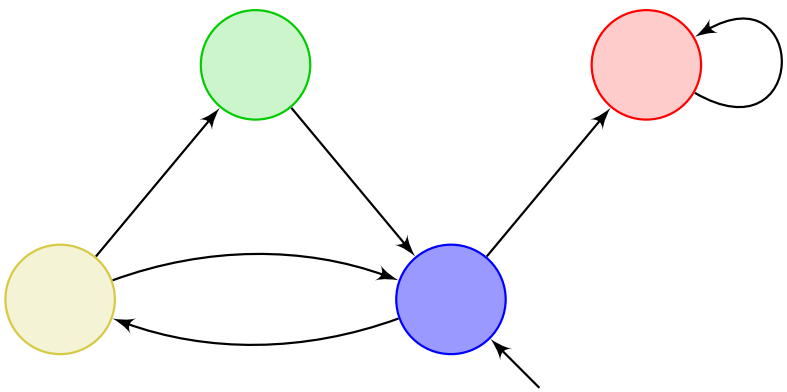
✓ $\exists \bigcirc.\ \mathbf{EG}\ \bigcirc$



Quantified-CTL

Definition

$$\text{QCTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{EG}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi \mid \exists \bigcirc.\ \varphi$$



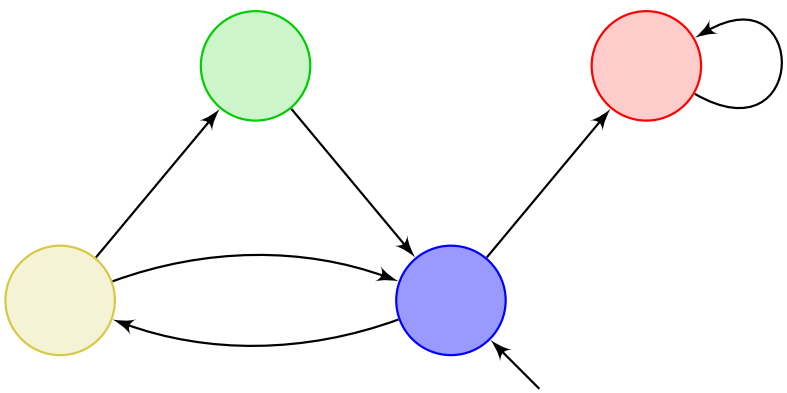
✓ $\exists \bigcirc.\ \mathbf{EG}\,\bigcirc$

$\mathbf{EF}(\forall \bigcirc.\ (\bigcirc \Rightarrow \mathbf{EX}\,\bigcirc))$

Quantified-CTL

Definition

$$\text{QCTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{EG}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi \mid \exists \bigcirc.\ \varphi$$



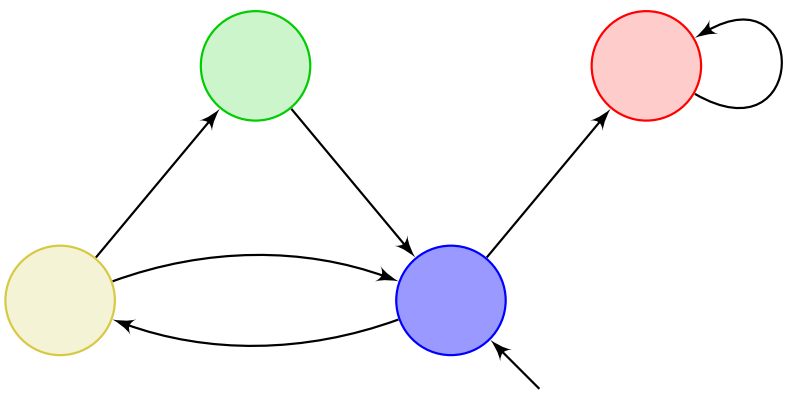
✓ $\exists \bigcirc.\ \mathbf{EG}\,\bigcirc$

✓ $\mathbf{EF}(\forall \bigcirc.\ (\bigcirc \Rightarrow \mathbf{EX}\,\bigcirc))$

Quantified-CTL

Definition

$$\text{QCTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{EG}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi \mid \exists \bigcirc.\ \varphi$$

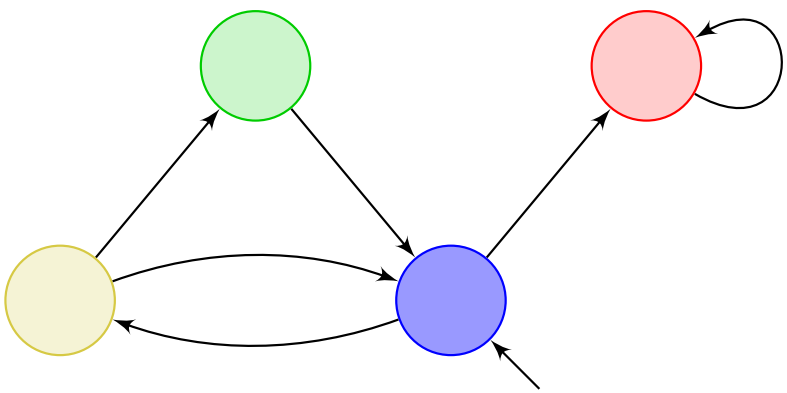


- ✓ $\exists \bigcirc.\ \mathbf{EG}\,\bigcirc$
- ✓ $\mathbf{EF}(\forall \bigcirc.\ (\bigcirc \Rightarrow \mathbf{EX}\,\bigcirc))$
- ✓ $\mathbf{EF}(\forall \bigcirc.\ \mathbf{EX}\,\bigcirc \Rightarrow \mathbf{AX}\,\bigcirc)$

Quantified-CTL

Definition

$$\text{QCTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{EG}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi \mid \exists \bigcirc.\ \varphi$$



✓ $\exists \bigcirc.\ \mathbf{EG}\,\bigcirc$

✓ $\mathbf{EF}(\forall \bigcirc.\ (\bigcirc \Rightarrow \mathbf{EX}\,\bigcirc))$

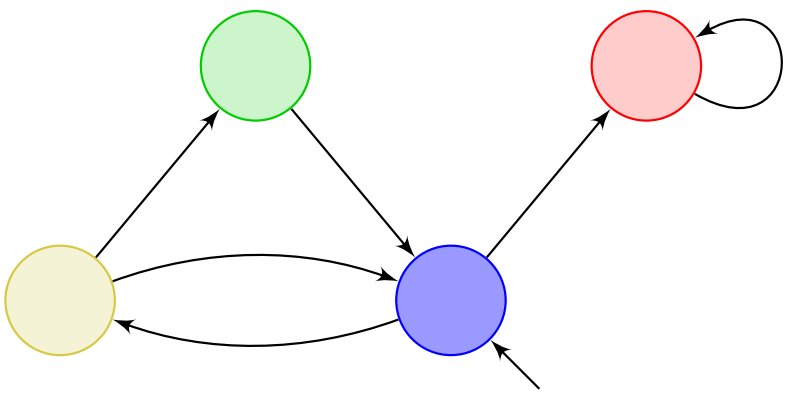
✓ $\mathbf{EF}(\forall \bigcirc.\ \mathbf{EX}\,\bigcirc \Rightarrow \mathbf{AX}\,\bigcirc)$

$\exists \bigcirc.\ (\bigcirc \wedge \mathbf{AXAG}\,\neg\bigcirc)$

Quantified-CTL

Definition

$$\text{QCTL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{EG}\,\varphi \mid \mathbf{E}\varphi\ \mathbf{U}\ \varphi \mid \exists \bigcirc.\ \varphi$$



✓ $\exists \bigcirc.\ \mathbf{EG}\,\bigcirc$

✓ $\mathbf{EF}(\forall \bigcirc.\ (\bigcirc \Rightarrow \mathbf{EX}\,\bigcirc))$

✓ $\mathbf{EF}(\forall \bigcirc.\ \mathbf{EX}\,\bigcirc \Rightarrow \mathbf{AX}\,\bigcirc)$

$\exists \bigcirc.\ (\bigcirc \wedge \mathbf{AXAG}\,\neg\bigcirc)$

× in the *structure* semantics

✓ in the *tree* semantics

●-equivalent structures

Definition

Given a Kripke structure $\mathcal{S} = \langle W, R, \ell \rangle$ with $\ell \colon W \to 2^{\mathsf{AP}}$ and an atomic proposition ●, we define $\mathcal{S}_{\bullet}$ as the Kripke structure $\langle W, R, \ell' \rangle$ with $\ell'(w) = \ell(w) \setminus \{\bullet\}$.

Definition

Two Kripke structures $\mathcal{S} = \langle W, R, \ell \rangle$ and $\mathcal{S}' = \langle W', R', \ell' \rangle$ are ●-equivalent if $\mathcal{S}_{\bullet} = \mathcal{S}'_{\bullet}$.

Example

●-equivalent structures

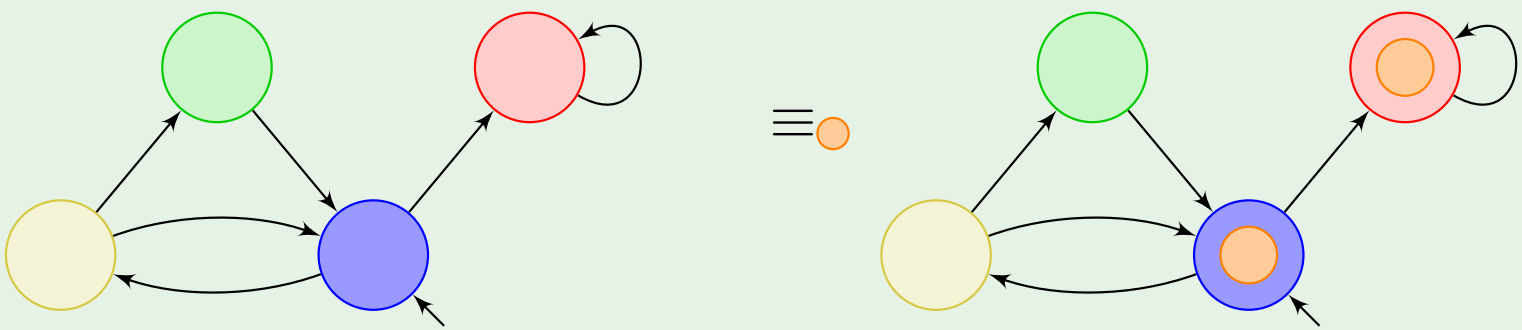
Definition

Given a Kripke structure $\mathcal{S} = \langle W, R, \ell \rangle$ with $\ell \colon W \to 2^{\mathsf{AP}}$ and an atomic proposition ●, we define $\mathcal{S}_{●}$ as the Kripke structure $\langle W, R, \ell' \rangle$ with $\ell'(w) = \ell(w) \setminus \{●\}$.

Definition

Two Kripke structures $\mathcal{S} = \langle W, R, \ell \rangle$ and $\mathcal{S}' = \langle W', R', \ell' \rangle$ are ●-equivalent if $\mathcal{S}_{●} = \mathcal{S}'_{●}$.

Example



Semantics of QCTL

Definition (structure semantics)

The structure semantics of QCTL extends that of CTL with

$$\mathcal{S}, q \models_s \exists \bigcirc. \varphi \quad \Leftrightarrow \quad \mathcal{S}', q \models_s \varphi$$

for some $\mathcal{S}' \equiv_{\bigcirc} \mathcal{S}$.

Definition (tree semantics)

The tree semantics of QCTL is obtained from the structure semantics by

$$\mathcal{S}, q \models_t \varphi \quad \Leftrightarrow \quad \mathrm{tree}(\mathcal{S}, q), \epsilon \models_s \varphi.$$

Semantics of QCTL

Definition (structure semantics)

The structure semantics of QCTL extends that of CTL with

$$\mathcal{S}, q \models_s \exists \bigcirc. \varphi \quad \Leftrightarrow \quad \mathcal{S}', q \models_s \varphi$$

for some $\mathcal{S}' \equiv_{\bigcirc} \mathcal{S}$.

Definition (tree semantics)

The tree semantics of QCTL is obtained from the structure semantics by

$$\mathcal{S}, q \models_t \varphi \quad \Leftrightarrow \quad \mathsf{tree}(\mathcal{S}, q), \epsilon \models_s \varphi.$$

Related work

Linear-time

- studied by Sistla, Vardi, Wolper (1987);
- model-checking is k-EXPSPACE-c. with quantifier alternation k;

Related work

Linear-time

- studied by Sistla, Vardi, Wolper (1987);
- model-checking is k-EXPSPACE-c. with quantifier alternation k;

Branching-time

- Kupferman (1995):
  - model-checking is NP-c. (resp. EXPTIME-c.) with quantifier depth 1 for structure (resp. tree) semantics;
- Kupferman, Madhusudan, Thiagarajan, Vardi (2000):
  - model-checking is 2-EXPTIME-complete with quantifier depth 2 (tree semantics);
- French (2003):
  - expressiveness: QCTL and QCTL* are equally expressive for the tree semantics.

Alternating-time Temporal Logic (ATL)

Definition

ATL extends CTL with *strategy quantifiers*:

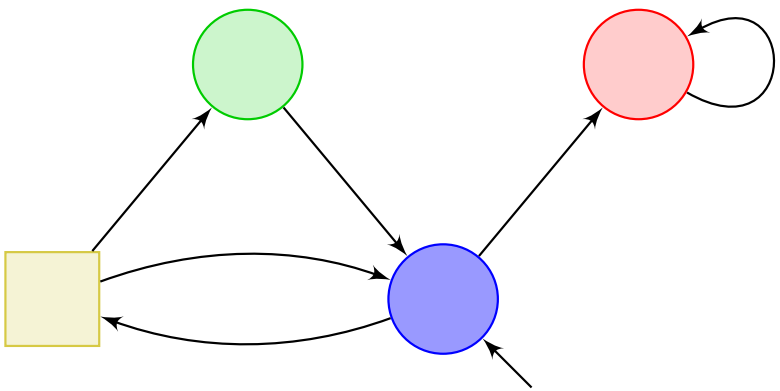
$$\text{ATL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\langle A \rangle\rangle \mathbf{X}\,\varphi \mid \langle\langle A \rangle\rangle \varphi\ \mathbf{U}\ \varphi \mid \langle\langle A \rangle\rangle \neg(\varphi\ \mathbf{U}\ \varphi)$$

Alternating-time Temporal Logic (ATL)

Definition

ATL extends CTL with *strategy quantifiers*:

$$\text{ATL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\langle A \rangle\rangle \mathbf{X}\,\varphi \mid \langle\langle A \rangle\rangle \varphi\ \mathbf{U}\ \varphi \mid \langle\langle A \rangle\rangle \neg(\varphi\ \mathbf{U}\ \varphi)$$



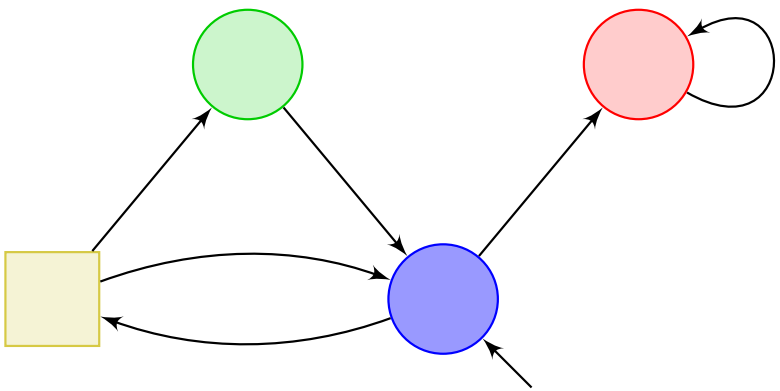
✓ $\langle\langle \bigcirc \rangle\rangle\ \mathbf{F}\ \bigcirc$

Alternating-time Temporal Logic (ATL)

Definition

ATL extends CTL with *strategy quantifiers*:

$$\text{ATL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\langle A \rangle\rangle \mathbf{X}\,\varphi \mid \langle\langle A \rangle\rangle \varphi\ \mathbf{U}\ \varphi \mid \langle\langle A \rangle\rangle \neg(\varphi\ \mathbf{U}\ \varphi)$$



✓ $\langle\langle \bigcirc \rangle\rangle\ \mathbf{F}\ \bigcirc$

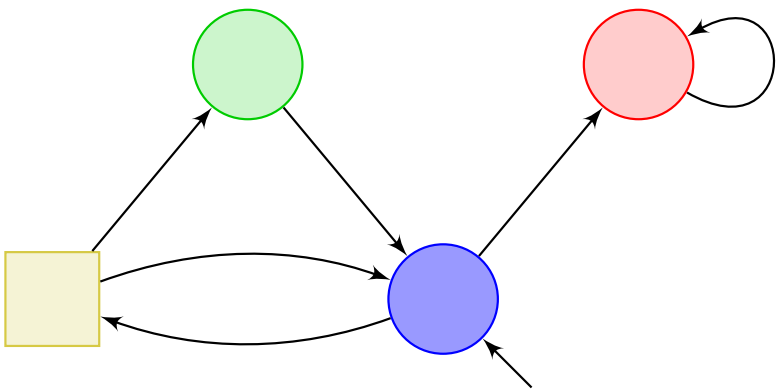
× $\langle\langle \Box \rangle\rangle\ \mathbf{F}\ \bigcirc$

Alternating-time Temporal Logic (ATL)

Definition

ATL extends CTL with *strategy quantifiers*:

$$\text{ATL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\langle A \rangle\rangle \mathbf{X}\,\varphi \mid \langle\langle A \rangle\rangle \varphi\ \mathbf{U}\ \varphi \mid \langle\langle A \rangle\rangle \neg(\varphi\ \mathbf{U}\ \varphi)$$



✓ $\langle\langle \bigcirc \rangle\rangle\ \mathbf{F}\ \bigcirc$

× $\langle\langle \square \rangle\rangle\ \mathbf{F}\ \bigcirc$

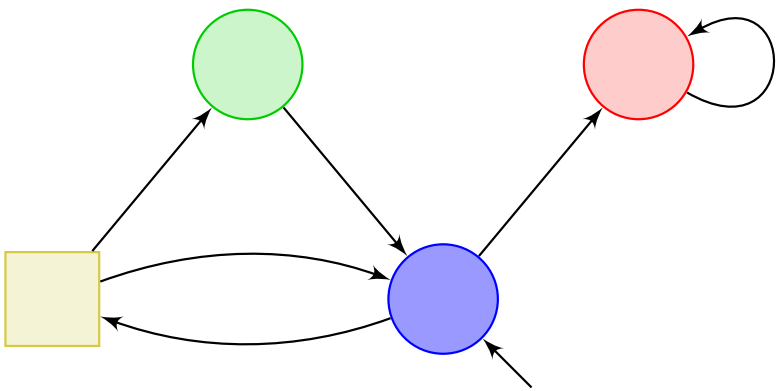
$\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \square \rangle\rangle\ \mathbf{F}\ \bigcirc)$

Alternating-time Temporal Logic (ATL)

Definition

ATL extends CTL with *strategy quantifiers*:

$$\text{ATL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\langle A \rangle\rangle \mathbf{X}\,\varphi \mid \langle\langle A \rangle\rangle \varphi\ \mathbf{U}\ \varphi \mid \langle\langle A \rangle\rangle \neg(\varphi\ \mathbf{U}\ \varphi)$$



✓ $\langle\langle \bigcirc \rangle\rangle\ \mathbf{F}\ \bigcirc$

× $\langle\langle \Box \rangle\rangle\ \mathbf{F}\ \bigcirc$

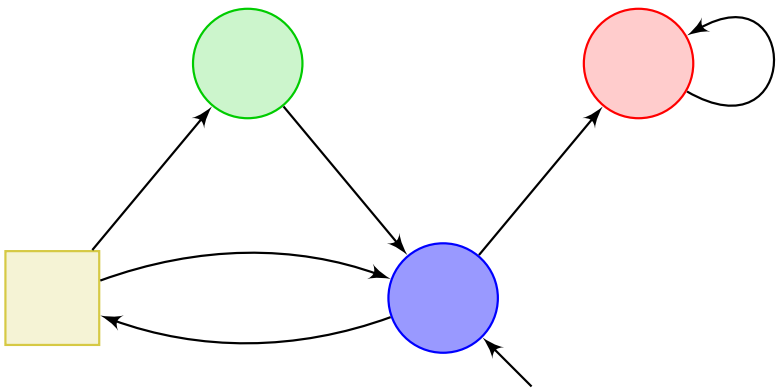
× $\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \Box \rangle\rangle\ \mathbf{F}\ \bigcirc)$

Alternating-time Temporal Logic (ATL)

Definition

ATL extends CTL with *strategy quantifiers*:

$$\text{ATL} \ni \varphi ::= \bigcirc \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\langle A \rangle\rangle \mathbf{X}\,\varphi \mid \langle\langle A \rangle\rangle \varphi\ \mathbf{U}\ \varphi \mid \langle\langle A \rangle\rangle \neg(\varphi\ \mathbf{U}\ \varphi)$$



✓ $\langle\langle \bigcirc \rangle\rangle\ \mathbf{F}\ \bigcirc$

× $\langle\langle \Box \rangle\rangle\ \mathbf{F}\ \bigcirc$

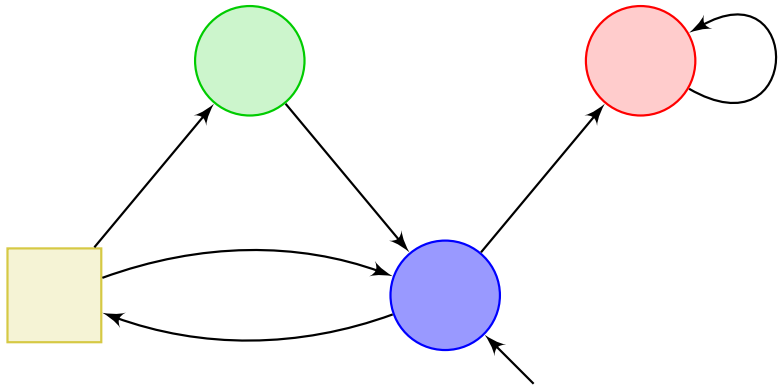
× $\langle\langle \bigcirc \rangle\rangle\ \mathbf{G}(\ \langle\langle \Box \rangle\rangle\ \mathbf{F}\ \bigcirc\ )$

Theorem

ATL model checking is PTIME*-complete.*

ATL with strategy contexts

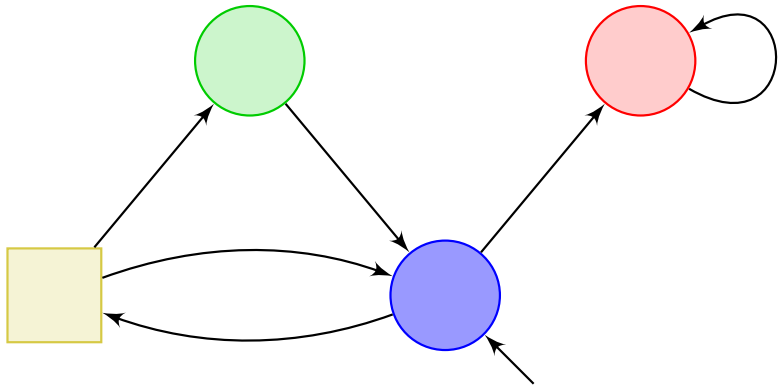
ATL_{sc} has the same syntax as ATL, but different semantics:



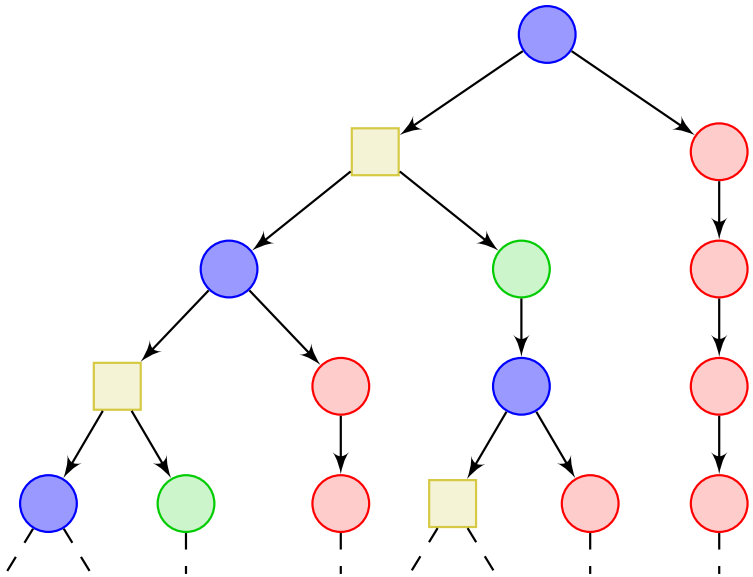
$\langle\langle\bigcirc\rangle\rangle$ **G**($\langle\langle\square\rangle\rangle$ **F** ●)

ATL with strategy contexts

ATL_{sc} has the same syntax as ATL, but different semantics:



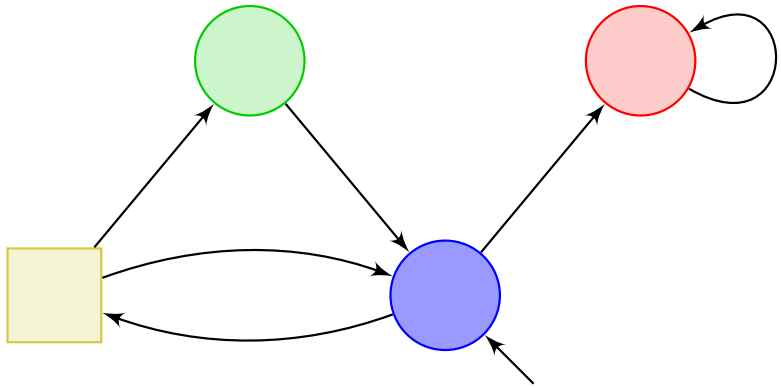
$\langle\langle\bigcirc\rangle\rangle$ **G**($\langle\langle\square\rangle\rangle$ **F** )



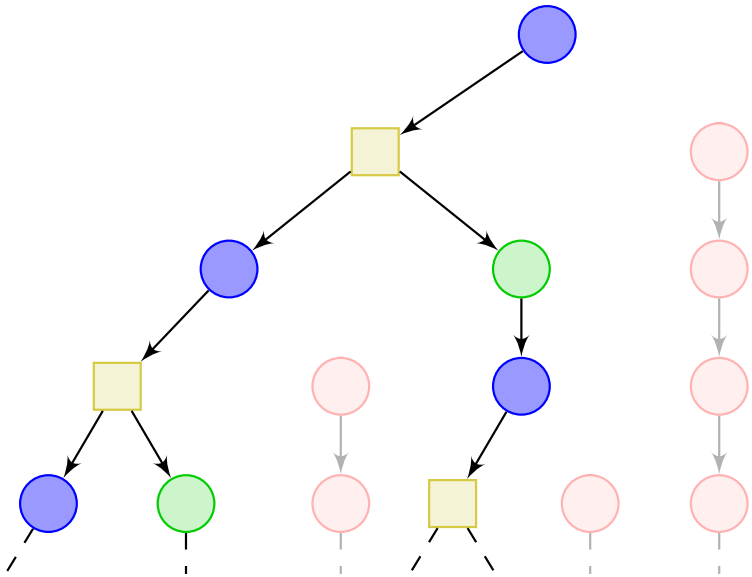
Evaluate the formula on the execution tree:

ATL with strategy contexts

ATL_{sc} has the same syntax as ATL, but different semantics:



$\langle\langle \bigcirc \rangle\rangle$ **G**($\langle\langle \square \rangle\rangle$ **F** )

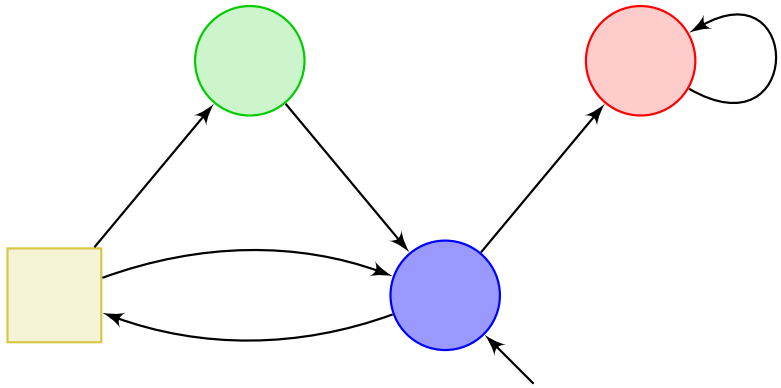


Evaluate the formula on the execution tree:

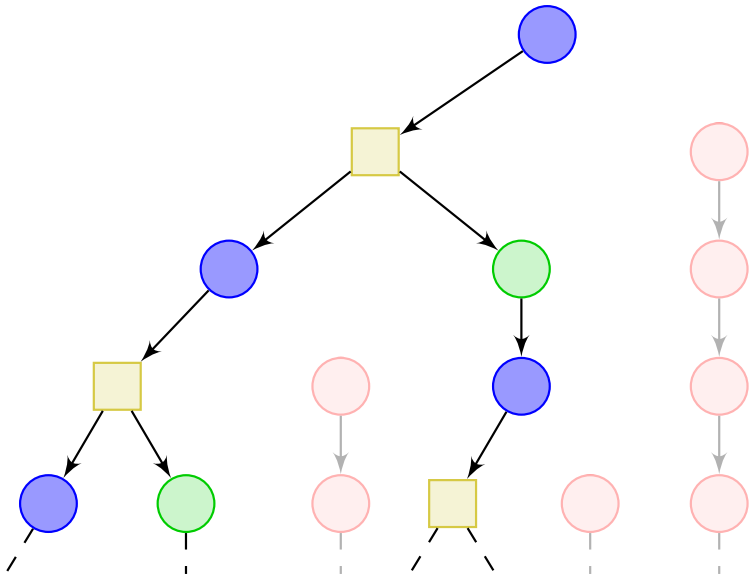
- apply a strategy of Player $\bigcirc$;

ATL with strategy contexts

ATL_{sc} has the same syntax as ATL, but different semantics:



$\langle\langle \bigcirc \rangle\rangle$ **G**($\langle\langle \square \rangle\rangle$ **F** )

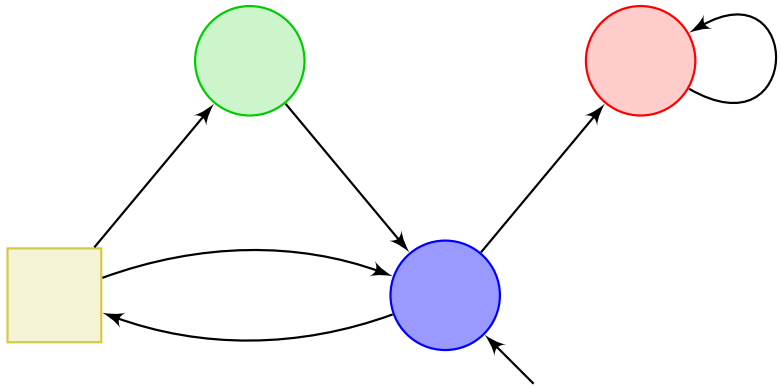


Evaluate the formula on the execution tree:

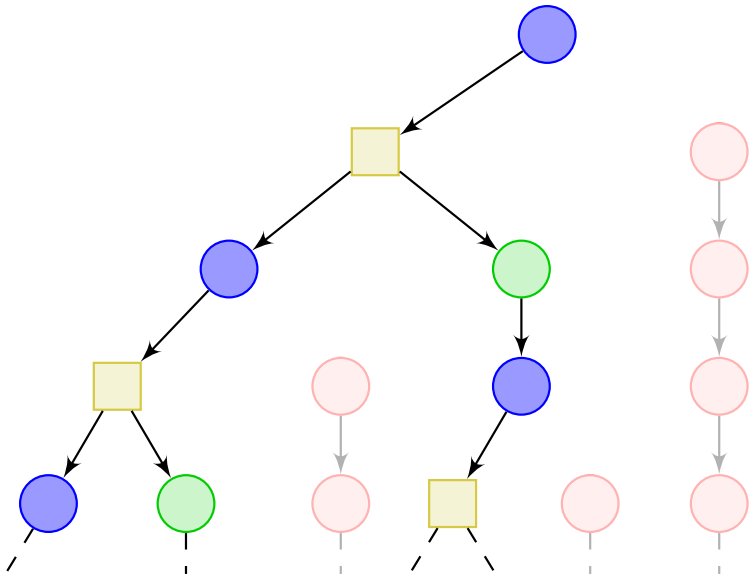
- apply a strategy of Player $\bigcirc$;
- in the remaining tree, check that Player $\square$ can always enforce a visit to  .

ATL with strategy contexts

ATL_{sc} has the same syntax as ATL, but different semantics:



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



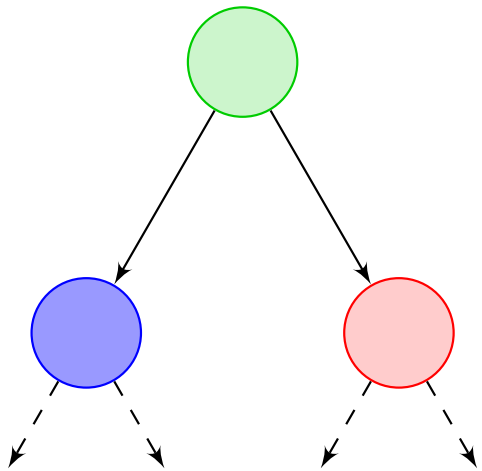
Evaluate the formula on the execution tree:

- apply a strategy of Player $\bigcirc$;
- in the remaining tree, check that Player $\Box$ can always enforce a visit to  .

From ATL_{sc} to QCTL

ATL_{sc} and QCTL are very tightly connected:

strategy = labelling of configurations with the action to be played.

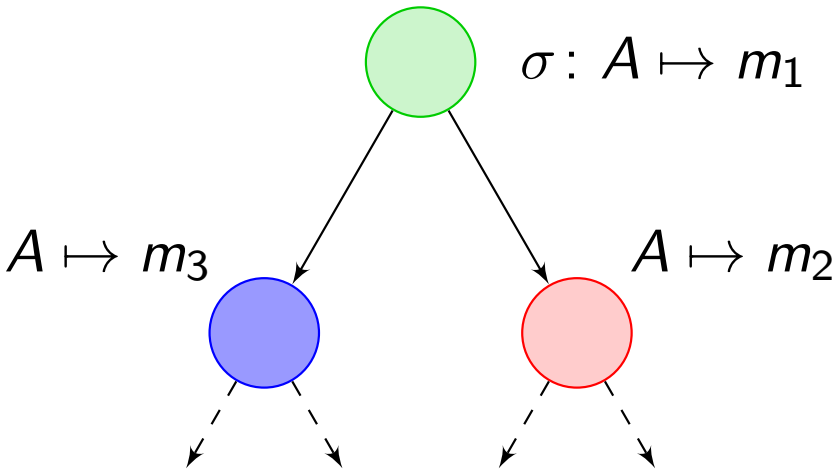


$\langle\langle A \rangle\rangle \varphi$

From ATL_{sc} to QCTL

ATL_{sc} and QCTL are very tightly connected:

strategy = labelling of configurations with the action to be played.

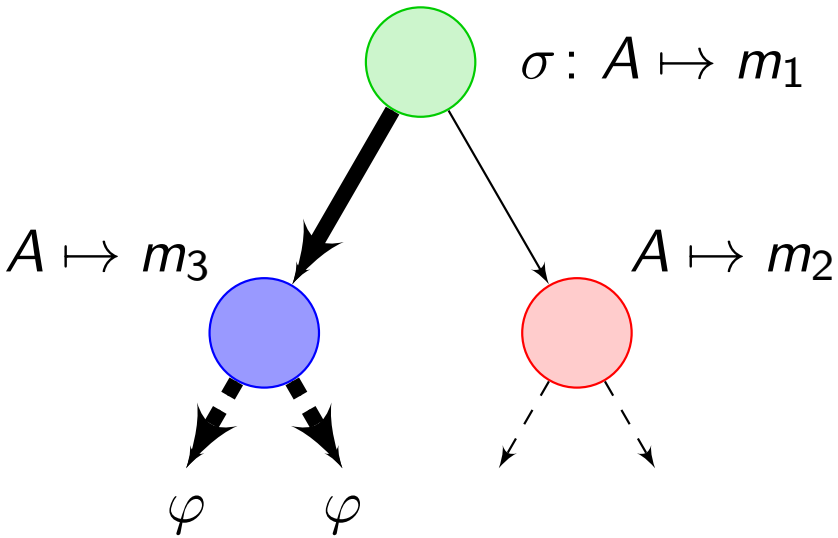


$\langle\langle A \rangle\rangle \varphi$

From ATL_{sc} to QCTL

ATL_{sc} and QCTL are very tightly connected:

strategy = labelling of configurations with the action to be played.

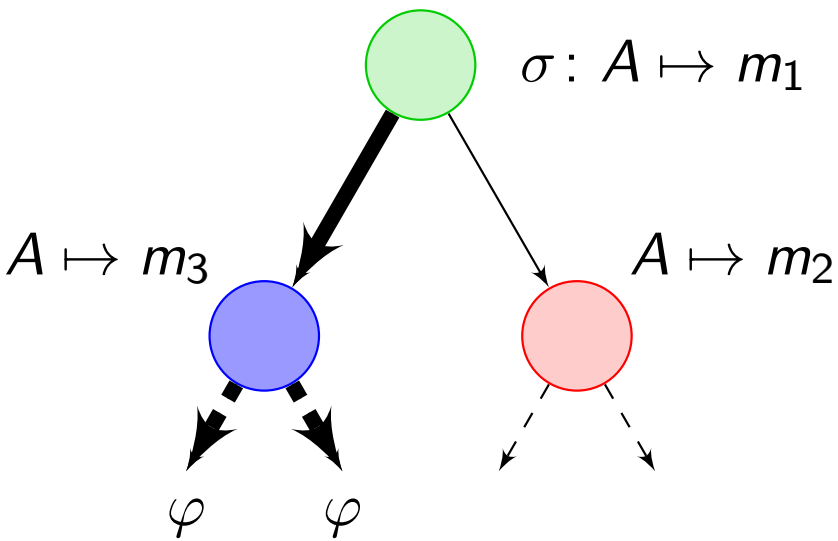


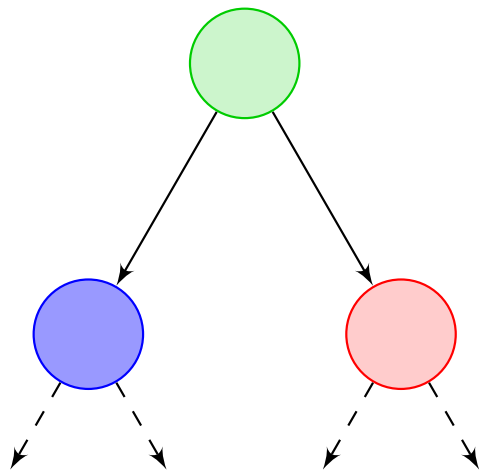
$\langle\langle A \rangle\rangle \varphi$

From ATL_{sc} to QCTL

ATL_{sc} and QCTL are very tightly connected:

strategy = labelling of configurations with the action to be played.



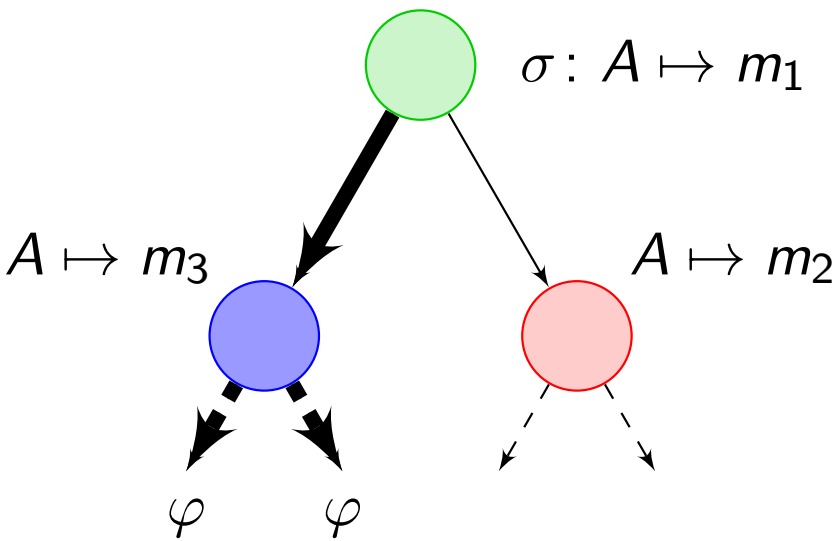


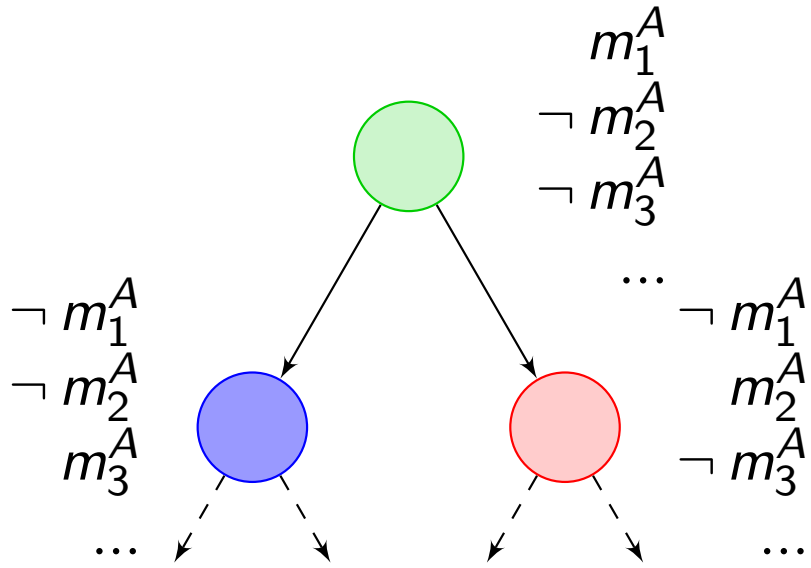
$\langle\langle A \rangle\rangle \varphi$

From ATL_{sc} to QCTL

ATL_{sc} and QCTL are very tightly connected:

strategy = labelling of configurations with the action to be played.



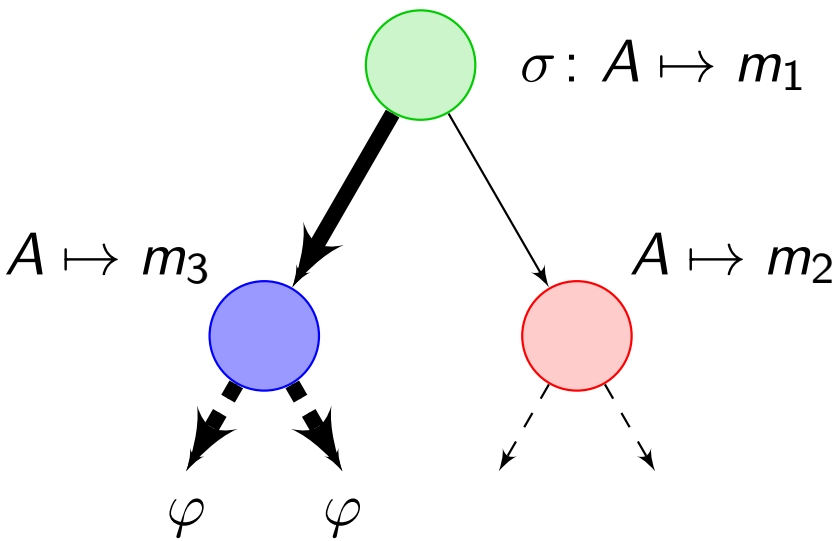


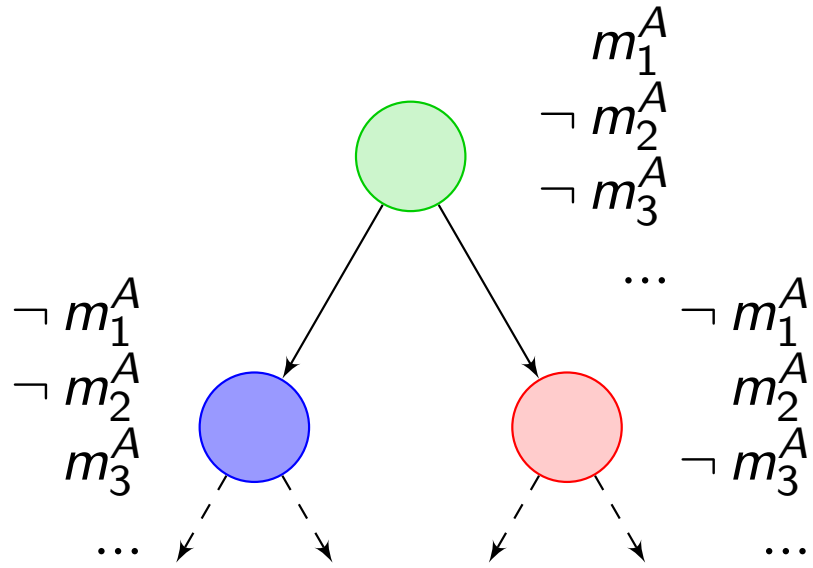
$\langle\!\langle A \rangle\!\rangle\, \varphi$

From ATL_{sc} to QCTL

ATL_{sc} and QCTL are very tightly connected:

strategy = labelling of configurations with the action to be played.





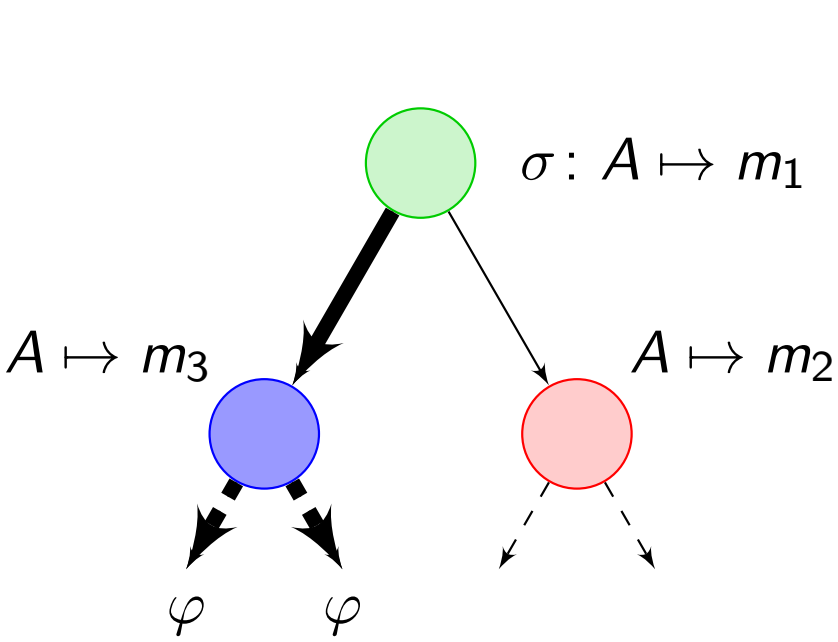
$\langle\langle A \rangle\rangle \varphi$

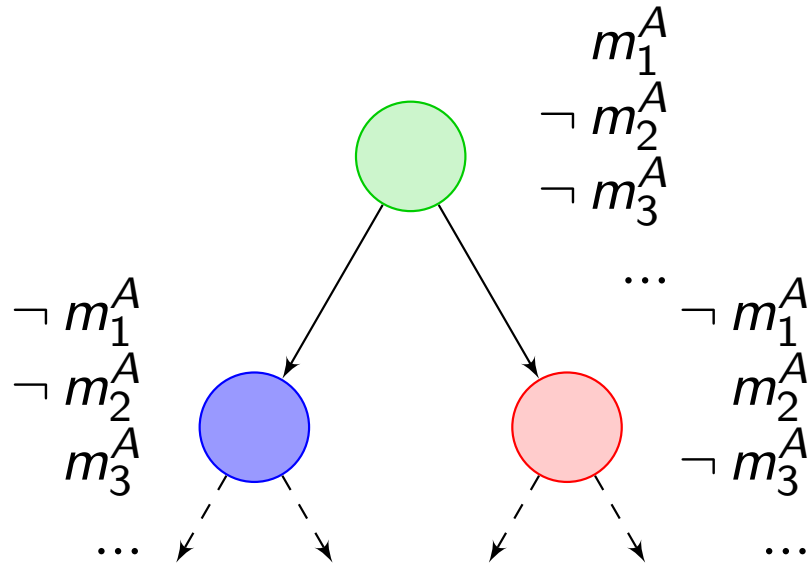
$\exists m_i^A.\ \mathbf{A}\,\mathbf{G}(\text{one}(m_1^A, ..., m_k^A)) \wedge$
$\mathbf{A}(\text{outcome} \Rightarrow \hat{\varphi})$

From ATL_{sc} to QCTL

ATL_{sc} and QCTL are very tightly connected:

strategy = labelling of configurations with the action to be played.





- tree semantics corresponds to ATL_{sc};
- structure semantics represents memoryless-strategy quantification.

Outline of the talk

Outline of the talk

Expressiveness of QCTL and QCTL*

Theorem

QCTL and QCTL* are equally expressive.

Proof

QCTL can express μ-calculus, hence CTL*, hence QCTL*.

$$q \models \mu T.\varphi(T) \quad \Leftrightarrow \quad q \models \exists T.\Big[T \wedge \mathbf{A}\,\mathbf{G}(T \Leftrightarrow \varphi(T) \wedge \forall U.(\mathbf{A}\,\mathbf{G}(U \Leftrightarrow \varphi(U)) \Rightarrow \mathbf{A}\,\mathbf{G}(T \Rightarrow U))\Big]$$

Expressiveness of QCTL and QCTL*

Theorem

QCTL and QCTL* are equally expressive.

Proof

QCTL can express μ-calculus, hence CTL*, hence QCTL*.

$$q \models \mu T.\varphi(T) \quad \Leftrightarrow \quad q \models \exists T.\Big[T \wedge \mathbf{A}\,\mathbf{G}(T \Leftrightarrow \varphi(T) \wedge \forall U.(\mathbf{A}\,\mathbf{G}(U \Leftrightarrow \varphi(U)) \Rightarrow \mathbf{A}\,\mathbf{G}(T \Rightarrow U))\Big]$$

Expressiveness of QCTL

Theorem

QCTL and QCTL* are as expressive as MSO.

Remark

The following formula expresses Hamiltonicity (under structure semantics):

$$\mathbf{E}\,\mathbf{G}(\exists z.\, \forall z'.\, [\mathrm{state}(z) \wedge \mathrm{state}(z') \wedge z \wedge \neg z'] \Rightarrow \mathbf{X}(\neg z\ \mathbf{U}\ z'))$$

Using similar ideas, it can express Eulerianity, which cannot be expressed in MSO.

But these are not QCTL* formulas: in QCTL*, propositional quantifiers must be followed by path quantifiers.

Expressiveness of QCTL

Theorem

QCTL and QCTL* are as expressive as MSO.

Remark

The following formula expresses Hamiltonicity (under structure semantics):

$$\mathbf{E}\,\mathbf{G}(\exists z.\, \forall z'.\, [\mathrm{state}(z) \wedge \mathrm{state}(z') \wedge z \wedge \neg z'] \Rightarrow \mathbf{X}(\neg z\ \mathbf{U}\ z'))$$

Using similar ideas, it can express Eulerianity, which cannot be expressed in MSO.

But these are not QCTL* formulas: in QCTL*, propositional quantifiers must be followed by path quantifiers.

Expressiveness of QCTL

Theorem

Any QCTL formula can be turned in prenex normal form.

Expressiveness of QCTL

Theorem

Any QCTL formula can be turned in prenex normal form.

Example

$$\mathbf{E}\,\mathbf{X}(\forall x.\varphi) \equiv \exists z.\forall x.\mathbf{E}\,\mathbf{X}(z \wedge \varphi)$$

Expressiveness of QCTL

Theorem

Any QCTL formula can be turned in prenex normal form.

Example

$$\mathbf{E}\,\mathbf{X}(\forall x.\varphi) \equiv \exists z.\forall x.\mathbf{E}\,\mathbf{X}(z \wedge \varphi)$$

Proof

Transform path quantification into propositional quantification.

- DAG-size linear in the DAG-size of the original formula.
- quantifier alternation linear in the quantifier depth of the original formula.

Outline of the talk

Model checking under the structure semantics

Theorem

QCTL and QCTL* model checking is PSPACE*-complete.*

Proof

- PSPACE-hardness from QBF;
- in PSPACE by enumerating the possible labellings.

Model checking under the structure semantics

Theorem

QCTL and QCTL* model checking is PSPACE*-complete.*

Proof

- PSPACE-hardness from QBF;
- in PSPACE by enumerating the possible labellings.

Theorem

- *EQ^kCTL model checking is $\Sigma^{\mathrm{P}}_{k+1}$-complete.*
  EQ^kCTL is the fragment in prenex normal form, with at most k quantifier alternations, starting with existential quantification.
- *Q^kCTL model checking is $\Delta^{\mathrm{P}}_{k+1}$-complete.*
  Q^kCTL is the fragment with at most k quantifiers.

Model checking under the tree semantics

Theorem

- *$EQ^k CTL$ and $Q^k CTL$ model checking are* k-EXPTIME-*complete.*
- *$EQ^k CTL^*$ and $Q^k CTL^*$ model checking are* (k+1)-EXPTIME-*complete.*

Proof

Algorithm for $\mathbf{EQ^k CTL}$:

- build (exponential-size) tree automaton for CTL formula;
- use projection for existential quantification (universal quantification by complementing, with exponential blowup).
- build product with the Kripke structure, and check emptiness.

Hardness: encoding of $k-1$-exponential-space alternating Turing machine (Sistla, Vardi, Wolper (1987)).

Conclusions and future work

Conclusions

- QCTL is a nice and natural extension of CTL:
  - same expressiveness as MSO;
  - high complexity.
- understanding QCTL helps us understanding ATL_{sc} (and Strategy Logic).

Conclusions and future work

Conclusions

- QCTL is a nice and natural extension of CTL:
  - same expressiveness as MSO;
  - high complexity.
- understanding QCTL helps us understanding ATL_{sc} (and Strategy Logic).

Future works

- other semantics for modelling finite-memory strategies?
- can QCTL help us find the right bisimulation notion that corresponds to ATL_{sc}?