Action-Based Temporal Logics and Model Checking (part I)

Radu Mateescu

Inria and LIG / Convecs

http://convecs.inria.fr

Action-based temporal logics

- Introduction
- Modal logics
- Branching-time logics

Why temporal logics?

- Need to specify *high-level properties* of concurrent systems, e.g.:
  - ***Mutual exclusion*** (at most one process in critical section)
  - ***Progress*** (each process eventually enters its critical section)
- **Temporal logics** (TLs):

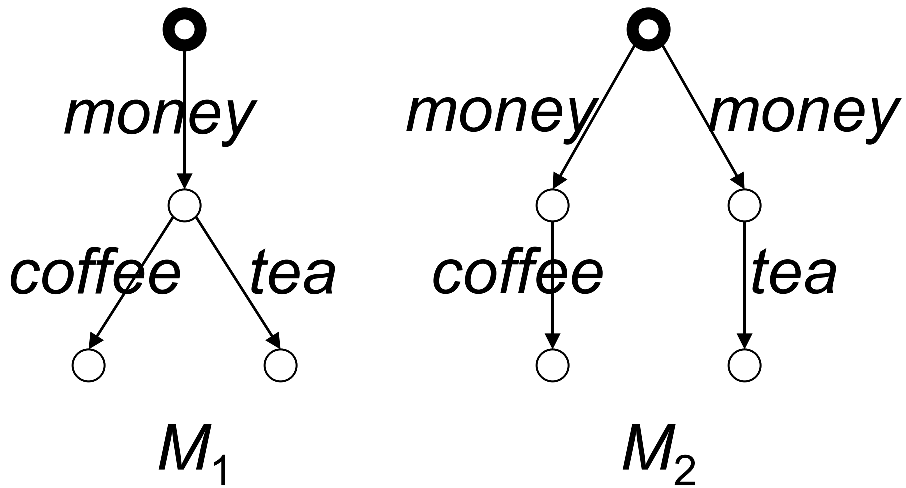
  formalisms describing the ordering of states (or actions) during the execution of a concurrent system
- TL specification = list of logical formulas, each one expressing a specific property of the system
- Benefits of TL [Manna-Pnueli-90]:
  - ***Abstraction***: properties expressed in TL are independent from the description/implementation of the system
  - ***Modularity***: one can add/change/remove a property without impacting the other properties of the specification

(Rough) classification of TLs

	State-based	Action-based
Linear-time (properties about execution sequences)	LTL (SPIN tool) linear mu-calculus	TLA (TLA+ tool) action-based LTL (LTSA tool)
Branching-time (properties about execution trees)	CTL (nuSMV tool) CTL*	ACTL (JACK tool) ACTL* modal mu-calculus (CWB, Concurrency Factory, CADP)

Example

(coffee machine)



$L\ (M_1) = L\ (M_2) =$
$\{$ *money*.*coffee*, *money*.*tea* $\}$

- A linear-time TL cannot distinguish the two LTSs M_1 and M_2, which have the same set of execution sequences, but are not behaviourally equivalent (modulo *strong bisimulation* – see Chapter on equivalences)
- A branching-time TL can capture nondeterminism and thus can distinguish M_1 and M_2

Interpretation of (branching-time) TLs on LTSs

- LTS model $M = \langle S, A, T, s_0 \rangle$, where:
  - S: set of states
  - A: set of actions (events)
  - $T \in S \times A \times S$: transition relation
  - $s_0 \in S$: initial state
- Interpretation of a formula φ on M:

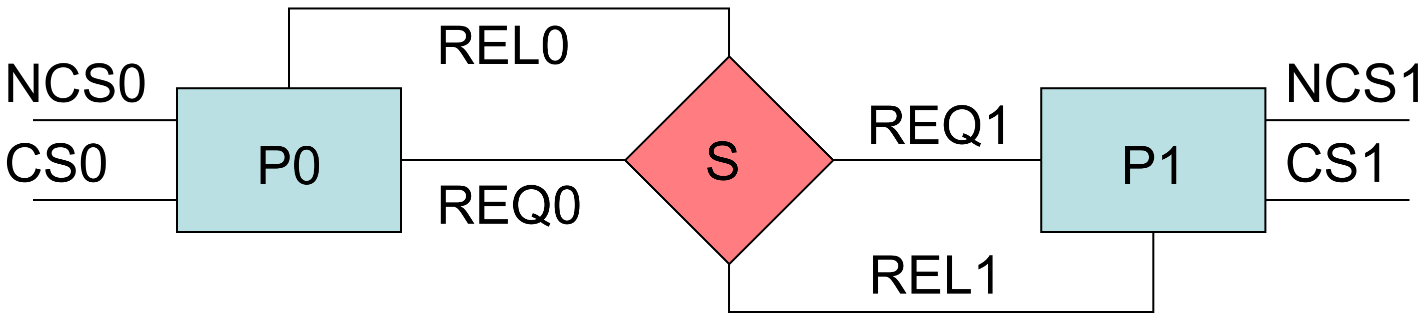
  $$[[\ \varphi\ ]] = \{\ s \in S \mid s \models \varphi\ \}$$

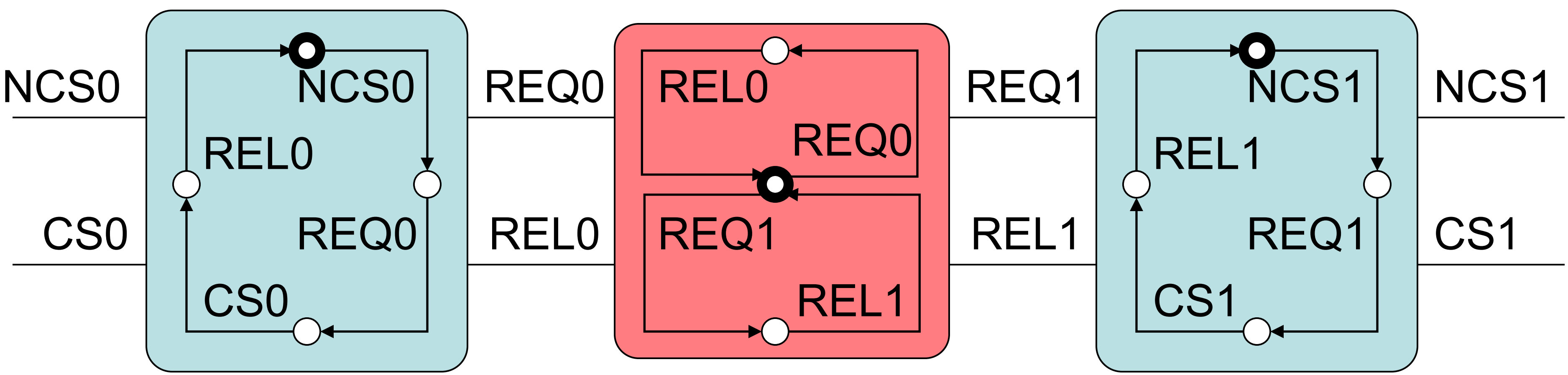
  $[[\ \varphi\ ]]$ is defined inductively on the structure of φ
- An LTS M satisfies a TL formula φ ($M \models \varphi$)

  iff its initial state satisfies φ :

  $$M \models \varphi \quad \Leftrightarrow \quad s_0 \models \varphi \quad \Leftrightarrow \quad s_0 \in [[\ \varphi\ ]]$$

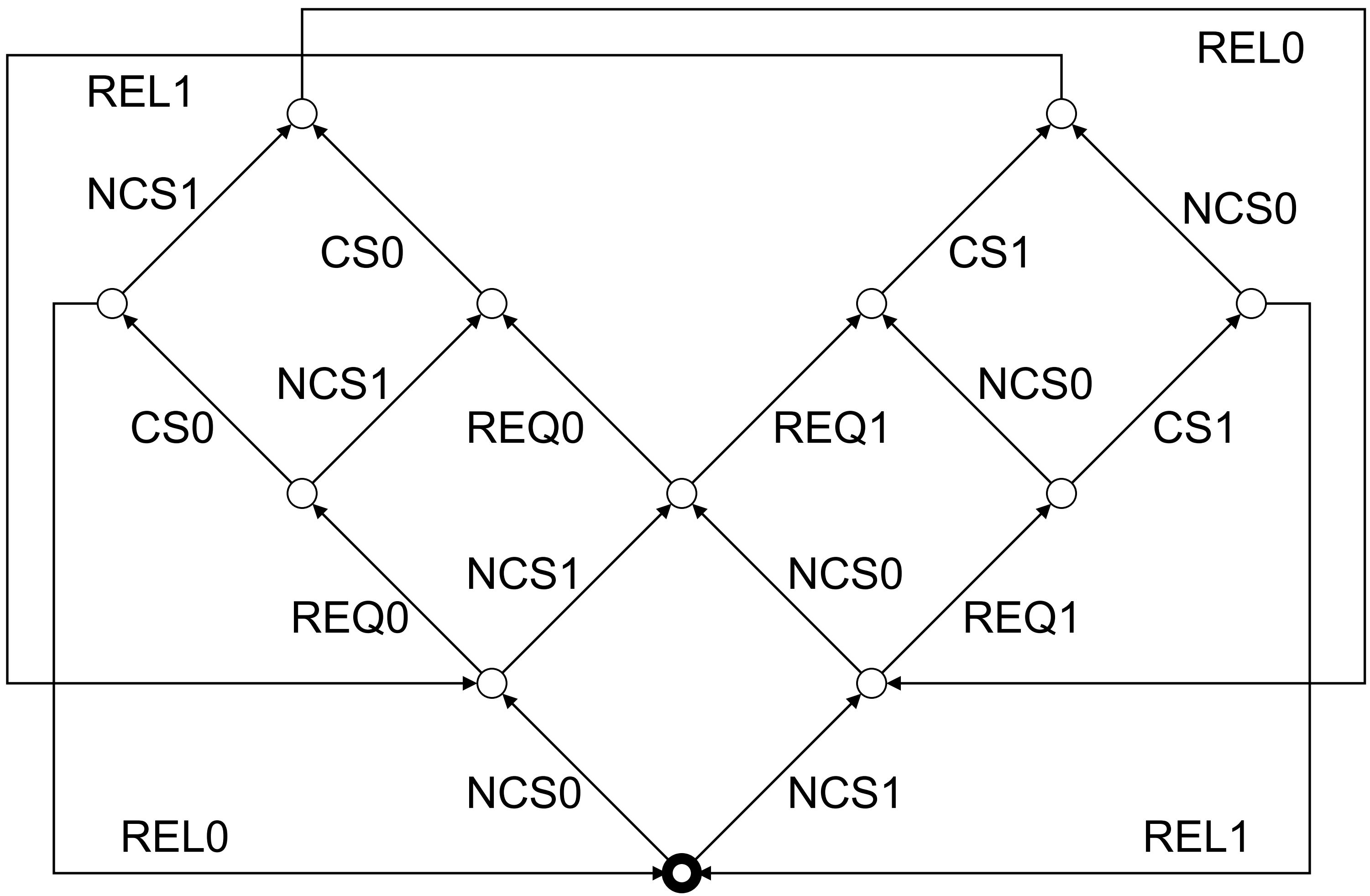
Running example: mutual exclusion with a semaphore





Description using communicating automata

LTS model


REL0
REL1
NCS1
CS0
NCS0
CS1
NCS1
REQ0
REQ1
NCS0
CS1
CS0
NCS1
REQ0
NCS0
REQ1
NCS0
NCS1
REL0
REL1

Modal logics

- These are the simplest logics allowing one to reason about the sequencing and branching of transitions in an LTS
- Basic modal operators:
  - ***Possibility***

    from a state, there exists (at least) an outgoing transition labeled by a given action and leading to a given state
  - ***Necessity***

    from a state, all the outgoing transitions labeled by a given action lead to given states
- **Hennessy-Milner Logic** (HML) [Hennessy-Milner-85]

Action predicates

(syntax)

$\alpha ::= a$	atomic proposition ($a \in A$)
$\mid$ true	constant "true"
$\mid$ false	constant "false"
$\mid \alpha_1 \vee \alpha_2$	disjunction
$\mid \alpha_1 \wedge \alpha_2$	conjunction
$\mid \neg\alpha_1$	negation
$\mid \alpha_1 \Rightarrow \alpha_2$	implication ($\neg\alpha_1 \vee \alpha_2$)
$\mid \alpha_1 \Leftrightarrow \alpha_2$	equivalence ($\alpha_1 \Rightarrow \alpha_2 \wedge \alpha_1 \Rightarrow \alpha_2$)

Action predicates

(semantics)

Let $M = (S, A, T, s_0)$. Interpretation $[[\ \alpha\ ]] \subseteq A$:

- $[[\ a\ ]] = \{\ a\ \}$
- $[[\ \text{true}\ ]] = A$
- $[[\ \text{false}\ ]] = \varnothing$
- $[[\ \alpha_1 \vee \alpha_2\ ]] = [[\ \alpha_1\ ]] \cup [[\ \alpha_2\ ]]$
- $[[\ \alpha_1 \wedge \alpha_2\ ]] = [[\ \alpha_1\ ]] \cap [[\ \alpha_2\ ]]$
- $[[\ \neg\alpha_1\ ]] = A \setminus [[\ \alpha_1\ ]]$
- $[[\ \alpha_1 \Rightarrow \alpha_2\ ]] = (A \setminus [[\ \alpha_1\ ]]) \cup [[\ \alpha_2\ ]]$
- $[[\ \alpha_1 \Leftrightarrow \alpha_2\ ]] = ((A \setminus [[\ \alpha_1\ ]]) \cup [[\ \alpha_2\ ]])$
  $\cap\ ((A \setminus [[\ \alpha_2\ ]]) \cup [[\ \alpha_1\ ]])$

Examples

A = { NCS_0, NCS_1, CS_0, CS_1, REQ_0, REQ_1, REL_0, REL_1 }

- [[true]] = { NCS_0, NCS_1, CS_0, CS_1, REQ_0, REQ_1, REL_0, REL_1 }
- [[false]] = $\varnothing$
- [[NCS_0]] = { NCS_0 }
- [[$\neg NCS_0$]] = { NCS_1, CS_0, CS_1, REQ_0, REQ_1, REL_0, REL_1 }
- [[$NCS_0 \wedge \neg NCS_1$]] = { NCS_0 } = [[NCS_0]]
- [[$NCS_0 \vee NCS_1$]] = { NCS_0, NCS_1 }
- [[$(NCS_0 \vee NCS_1) \wedge (NCS_0 \vee REQ_0)$]] = { NCS_0 }
- [[$NCS_0 \wedge NCS_1$]] = $\varnothing$ = [[false]]
- [[$NCS_0 \vee \neg NCS_0$]] =
  { NCS_0, NCS_1, CS_0, CS_1, REQ_0, REQ_1, REL_0, REL_1 } = [[true]]

HML logic

(syntax)

φ ::=	true	constant "true"
|	false	constant "false"
|	$\varphi_1 \vee \varphi_2$	disjunction
|	$\varphi_1 \wedge \varphi_2$	conjunction
|	$\neg\varphi_1$	negation
|	$\langle \alpha \rangle \varphi_1$	possibility
|	$[\alpha] \varphi_1$	necessity

- Duality: $[\alpha] \varphi = \neg\langle \alpha \rangle \neg\varphi$

HML logic

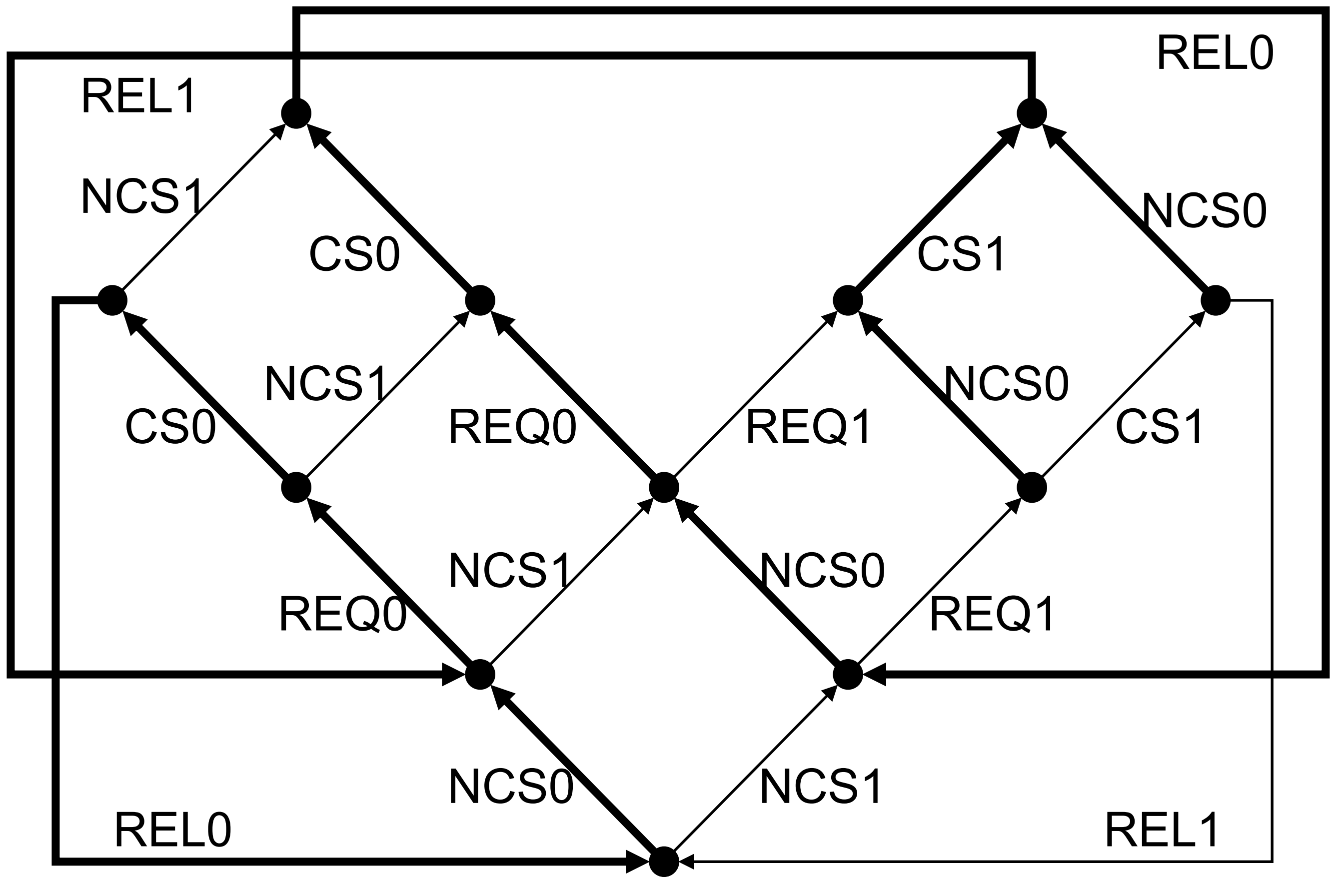
(semantics)

Let $M = (S, A, T, s_0)$. Interpretation $[[\ \varphi\ ]] \subseteq S$:

- $[[\ \text{true}\ ]] = S$
- $[[\ \text{false}\ ]] = \emptyset$
- $[[\ \varphi_1 \vee \varphi_2\ ]] = [[\ \varphi_1\ ]] \cup [[\ \varphi_2\ ]]$
- $[[\ \varphi_1 \wedge \varphi_2\ ]] = [[\ \varphi_1\ ]] \cap [[\ \varphi_2\ ]]$
- $[[\ \neg\varphi_1\ ]] = S \setminus [[\ \varphi_1\ ]]$
- $[[\ \langle\ \alpha\ \rangle\ \varphi_1\ ]] = \{\ s \in S \mid \exists\ (s, a, s') \in T\ .$
  $a \in [[\ \alpha\ ]] \wedge s' \in [[\ \varphi_1\ ]]\ \}$
- $[[\ [\ \alpha\ ]\ \varphi_1\ ]] = \{\ s \in S \mid \forall\ (s, a, s') \in T\ .$
  $a \in [[\ \alpha\ ]] \Rightarrow s' \in [[\ \varphi_1\ ]]\ \}$

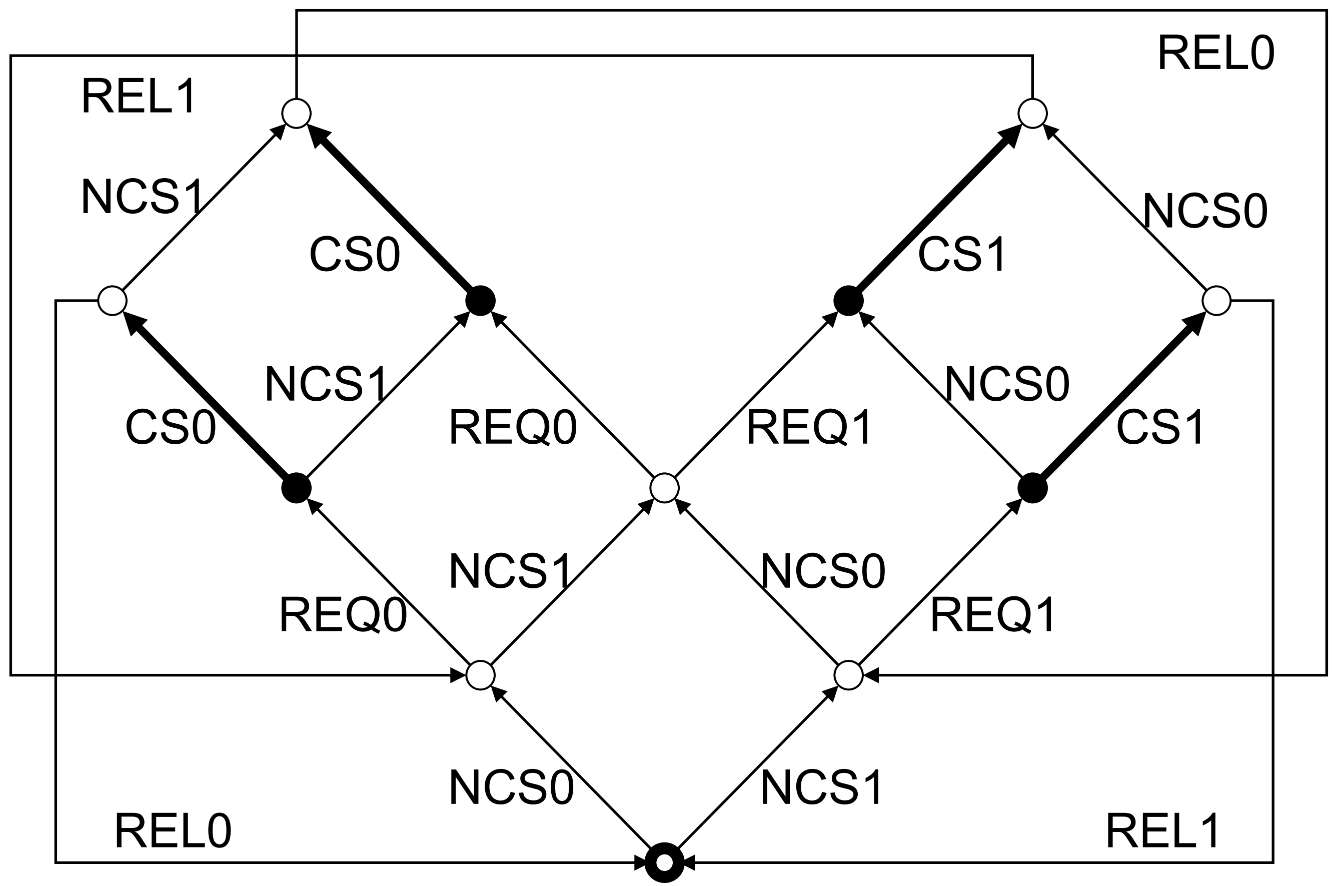
Example (1/4)

Deadlock freedom: $\langle$ **true** $\rangle$ **true**



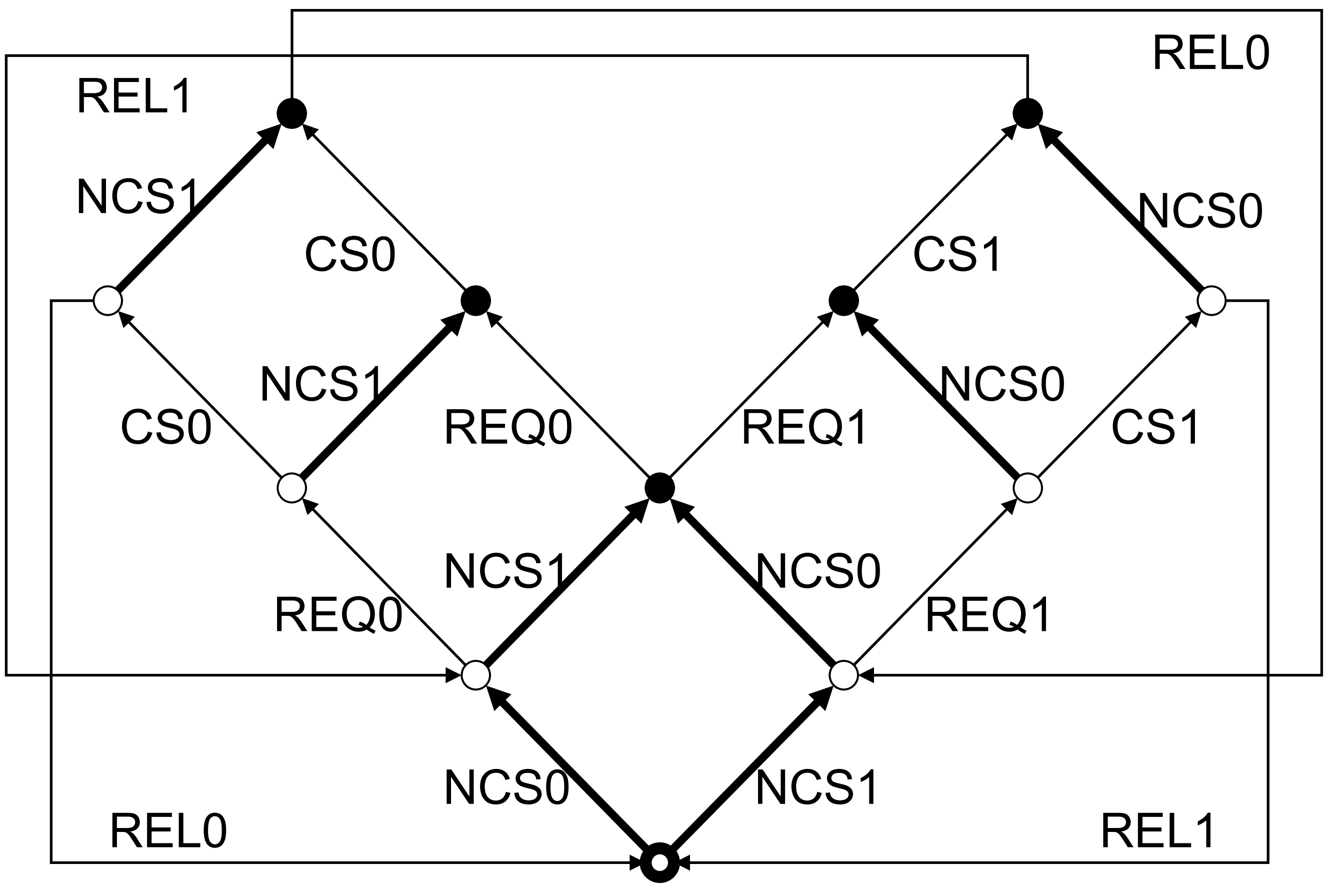
Example (2/4)

Possible execution of a set of actions: $\langle\, CS_0 \vee CS_1 \,\rangle$ **true**



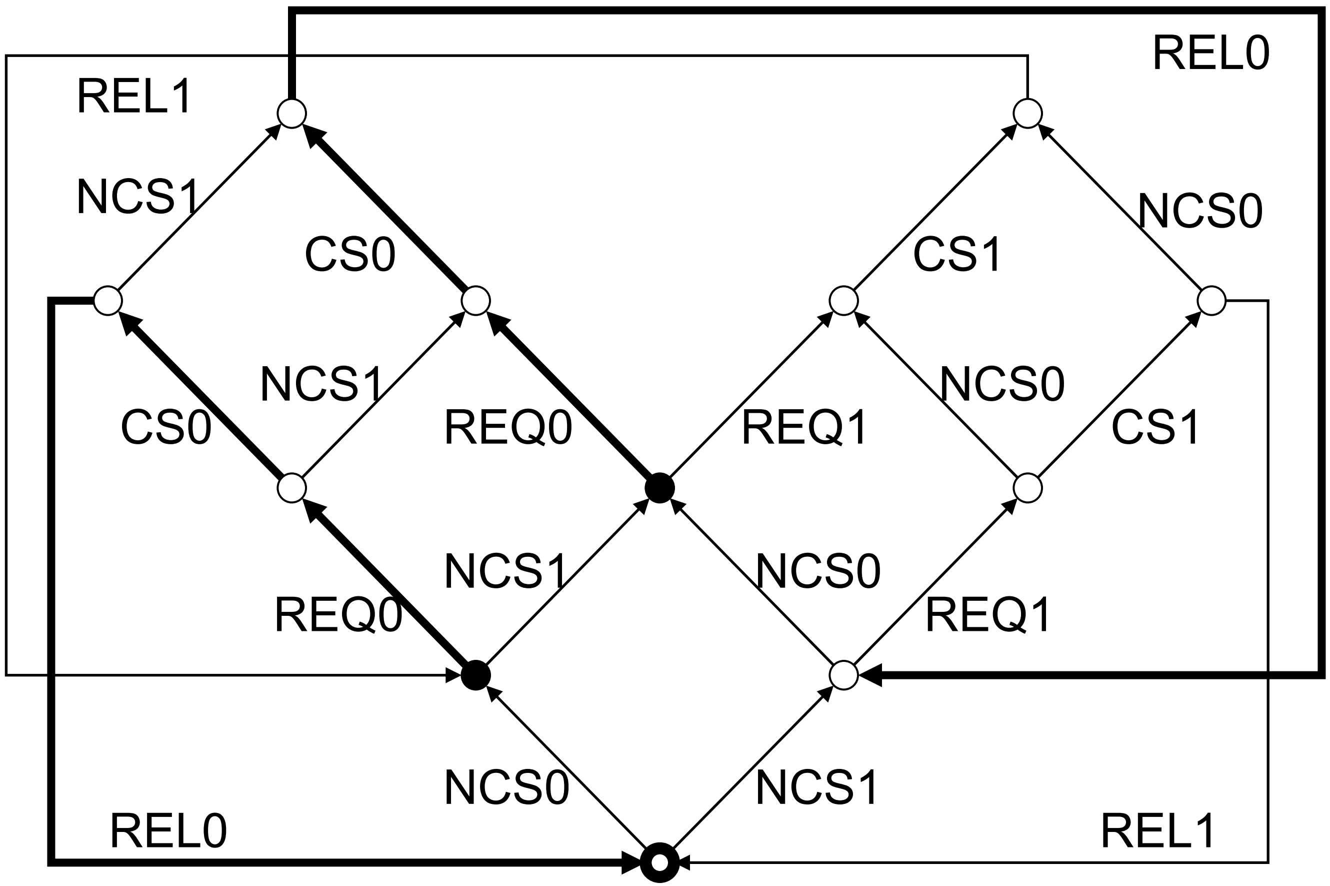
Example (3/4)

Forbidden execution of a set of actions: **[$NCS_0 \vee NCS_1$] false**



Example (4/4)

Execution of an action sequence: $\langle\, \mathbf{REQ_0} \,\rangle \langle\, \mathbf{CS_0} \,\rangle \langle\, \mathbf{REL_0} \,\rangle$ **true**



Some identities

- Contradiction and tautology:
  - $\langle \alpha \rangle$ false = $\langle$ false $\rangle \varphi$ = false
  - $[\alpha]$ true = [false] φ = true
- Distributivity of modalities over $\vee$ and $\wedge$:
  - $\langle \alpha \rangle \varphi_1 \vee \langle \alpha \rangle \varphi_2 = \langle \alpha \rangle (\varphi_1 \vee \varphi_2)$
  - $\langle \alpha_1 \rangle \varphi \vee \langle \alpha_2 \rangle \varphi = \langle \alpha_1 \vee \alpha_2 \rangle \varphi$
  - $[\alpha] \varphi_1 \wedge [\alpha] \varphi_2 = [\alpha] (\varphi_1 \wedge \varphi_2)$
  - $[\alpha_1] \varphi \wedge [\alpha_2] \varphi = [\alpha_1 \vee \alpha_2] \varphi$
- Monotonicity of modalities over φ:
  - $(\varphi_1 \Rightarrow \varphi_2) \Rightarrow (\langle \alpha \rangle \varphi_1 \Rightarrow \langle \alpha \rangle \varphi_2) \wedge ([\alpha] \varphi_1 \Rightarrow [\alpha] \varphi_2)$
  - $(\alpha_1 \Rightarrow \alpha_2) \Rightarrow (\langle \alpha_1 \rangle \varphi \Rightarrow \langle \alpha_2 \rangle \varphi) \wedge ([\alpha_2] \varphi \Rightarrow [\alpha_1] \varphi)$

Exercises

- Show the following identity in HML:

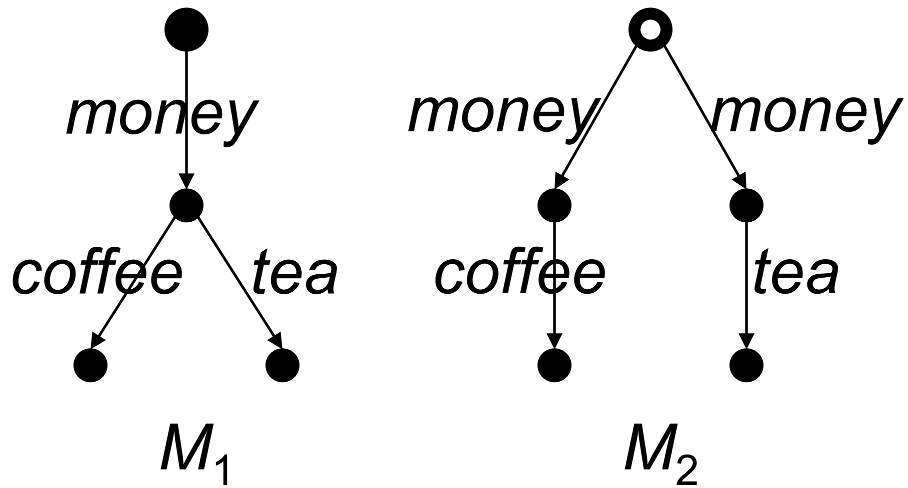
$$\langle \alpha \rangle \text{true} \wedge [\, \alpha \,]\, \varphi \;=\; \langle \alpha \rangle \varphi \wedge [\, \alpha \,]\, \varphi$$

- Show the following implications (monotonicity of modalities over α):

  if $\alpha_1 \Rightarrow \alpha_2$ then $\quad \langle \alpha_1 \rangle \varphi \Rightarrow \langle \alpha_2 \rangle \varphi$

  if $\alpha_1 \Rightarrow \alpha_2$ then $\quad [\, \alpha_2 \,]\, \varphi \Rightarrow [\, \alpha_1 \,]\, \varphi$

Characterization of branching



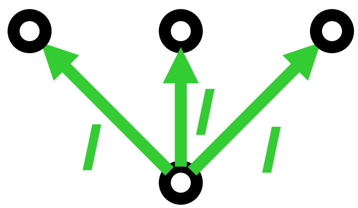
- Modal formula distinguishing between M_1 and M_2:

$$\varphi = [\, \mathit{money} \,]\ (\langle \mathit{coffee} \rangle \text{true} \wedge \langle \mathit{tea} \rangle \text{true})$$

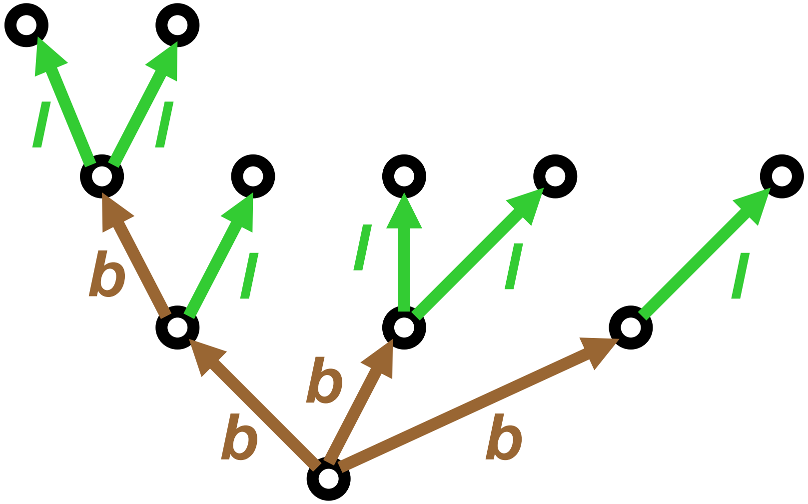
$$M_1 \models \varphi \qquad \text{and} \qquad M_2 \not\models \varphi$$

Exercise

- Characterize in HML the tree-like LTSs below
  - Action predicates:
    l (*leaf*)
    b (*branch*)



bush



shrub

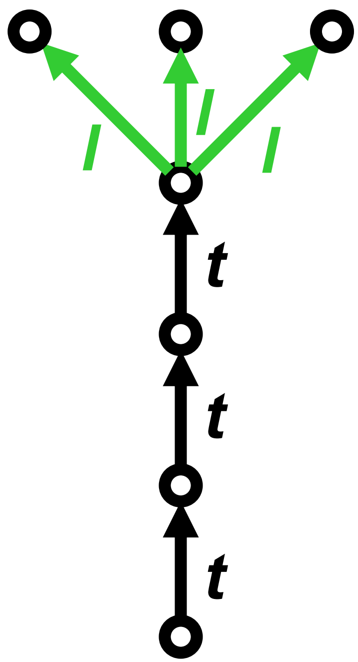
Exercise

- Characterize in HML the tree-like LTS below
  - Action predicates:

    l (*leaf*)

    b (*branch*)

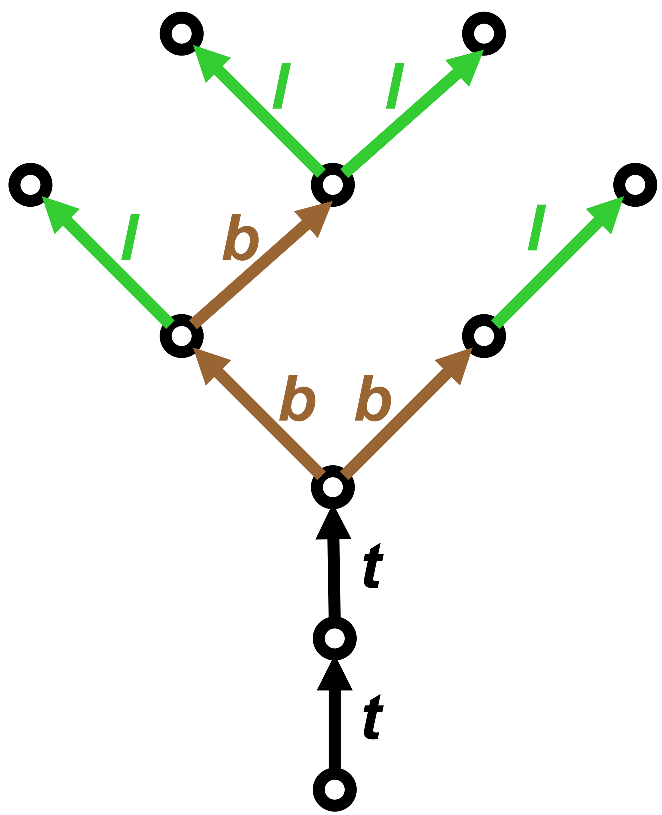
    t (***trunk***)



palm tree

Exercise

- Characterize in HML the tree-like LTS below



tree

Modal logics

(summary)

- Are able to express simple branching-time properties involving states $s \in S$ and actions $a \in A$ of an LTS
- But:
  - Take into account only a ***finite*** number of steps around a state (limited by the nesting of modalities)
  - Cannot express properties about transition sequences or subtrees of arbitrary length
- Example: the property

  "from a state *s*, there exists a sequence leading to a state *s'* where the action *a* is executable"

  cannot be expressed in modal logic, because it would need a formula of unknown length

$$\langle \text{ true } \rangle \langle \text{ true } \rangle \ldots \langle \text{ true } \rangle \langle \, a \, \rangle \text{ true}$$

Branching-time logics

- These logics allow one to reason about the (infinite) execution trees contained in an LTS
- Basic temporal operators:
  - ***Potentiality***

    from a state, there exists an outgoing, finite transition sequence leading to a given state
  - ***Inevitability***

    from a state, all outgoing transition sequences lead, after a finite number of steps, to given states
- **Action-based Computation Tree Logic** (ACTL) [DeNicola-Vaandrager-90]

ACTL logic

(syntax)

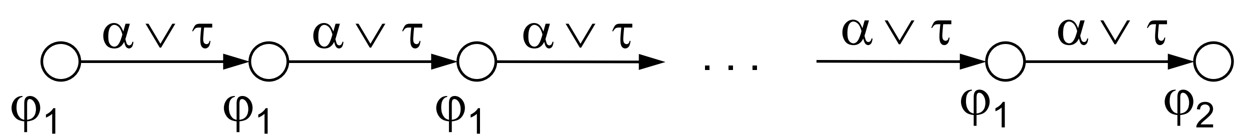
$\varphi ::= \text{true} \;\mid\; \text{false}$	boolean constants
$\mid\; \varphi_1 \vee \varphi_2 \;\mid\; \neg\varphi_1$	boolean connectors
$\mid\; E\,[\,\varphi_{1\alpha 1}\, U\, \varphi_2\,]$	potentiality 1
$\mid\; E\,[\,\varphi_{1\alpha 1}\, U_{\alpha 2}\, \varphi_2\,]$	potentiality 2
$\mid\; A\,[\,\varphi_{1\alpha 1}\, U\, \varphi_2\,]$	inevitability 1
$\mid\; A\,[\,\varphi_{1\alpha 1}\, U_{\alpha 2}\, \varphi_2\,]$	inevitability 2

ACTL logic

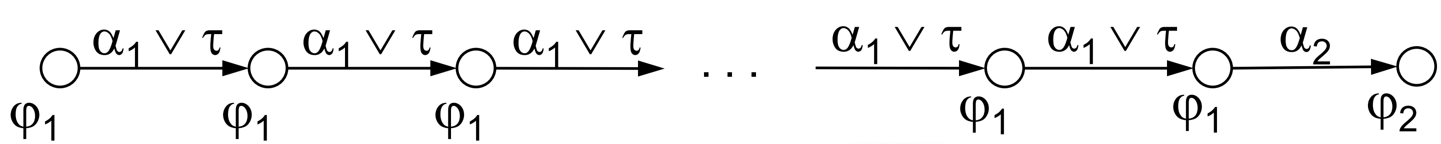
(semantics – potentiality operators)

Let $M = (S, A, T, s_0)$. Interpretation $[[\ \varphi\ ]] \subseteq S$:

- $[[\ E\ [\ \varphi_{1\alpha}\ U\ \varphi_2\ ]\ ]] = \{\ s \in S \mid \exists s(=s_0) \rightarrow^{a0} s_1 \rightarrow^{a1} s_2 \rightarrow \ldots\ .$
  $\exists k \geq 0.\ \forall 0 \leq i < k.\ (s_i \in [[\ \varphi_1\ ]] \wedge a_i \in [[\ \alpha \vee \tau\ ]]) \wedge$
  $s_k \in [[\ \varphi_2\ ]]\ \}$



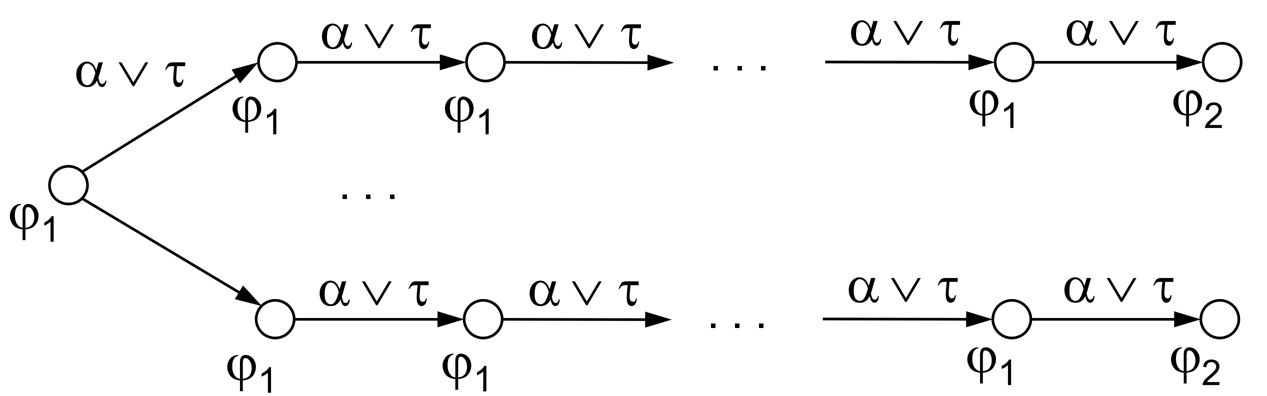
- $[[\ E\ [\ \varphi_{1\alpha 1}\ U_{\alpha 2}\ \varphi_2\ ]\ ]] = \{\ s \in S \mid \forall s(=s_0) \rightarrow^{a0} s_1 \rightarrow^{a1} s_2 \rightarrow \ldots\ .$
  $\exists k \geq 0.\ \forall 0 \leq i < k.\ (s_i \in [[\ \varphi_1\ ]] \wedge a_i \in [[\ \alpha_1 \vee \tau\ ]] \wedge$
  $s_k \in [[\ \varphi_1\ ]] \wedge a_k \in [[\ \alpha_2\ ]] \wedge s_{k+1} \in [[\ \varphi_2\ ]]\ \}$



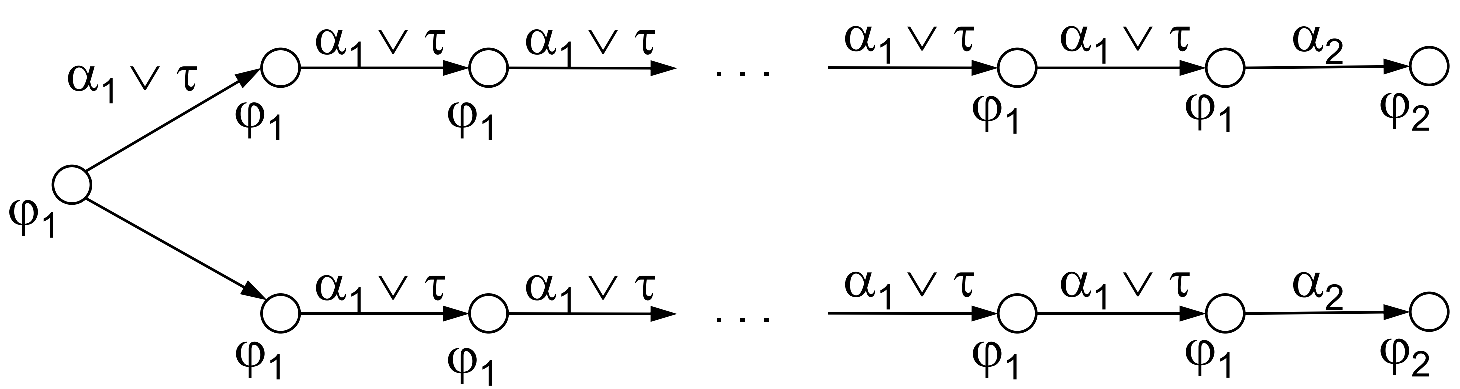
ACTL logic

(semantics – inevitability operators)

- $[[\ A\ [\ \varphi_{1\alpha}\ U\ \varphi_2\ ]\ ]]$:



- $[[\ A\ [\ \varphi_{1\alpha1}\ U_{\alpha2}\ \varphi_2\ ]\ ]]$:



ACTL logic

(derived operators)

- $EF_\alpha\ \varphi \quad = E\ [\ true_\alpha\ U\ \varphi\ ]$ basic potentiality
- $AF_\alpha\ \varphi \quad = A\ [\ true_\alpha\ U\ \varphi\ ]$ basic inevitability

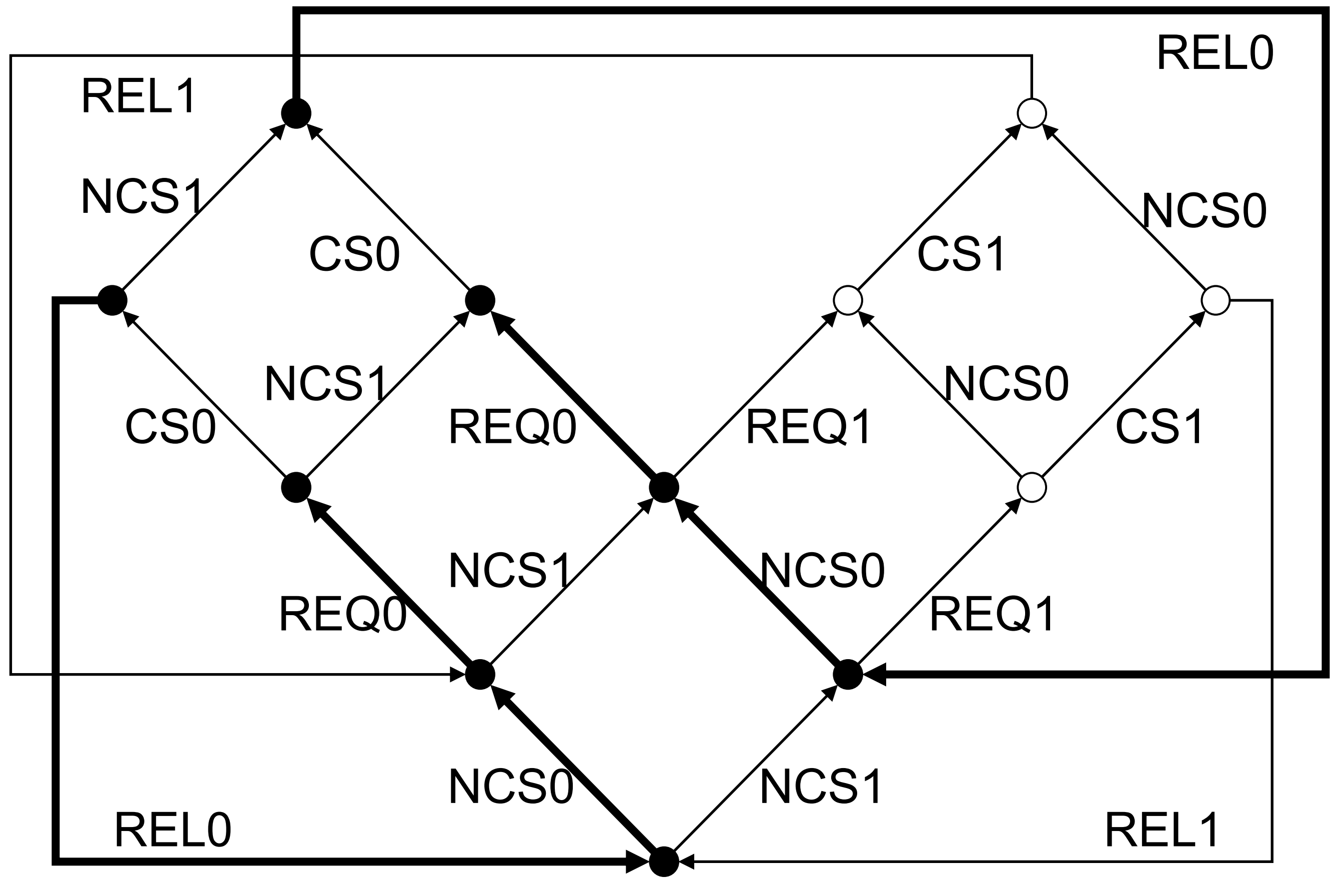
- $AG_\alpha\ \varphi \quad = \neg\ EF_\alpha\ \neg\varphi$ invariance
- $EG_\alpha\ \varphi \quad = \neg\ AF_\alpha\ \neg\varphi$ trajectory

dualities

- $\langle\ \alpha\ \rangle\ \varphi = E\ [\ tt_{ff}\ U_\alpha\ \varphi\ ]$ (weak) possibility
- $[\ \alpha\ ]\ \varphi = \neg\ \langle\ \alpha\ \rangle\ \neg\ \varphi$ (weak) necessity

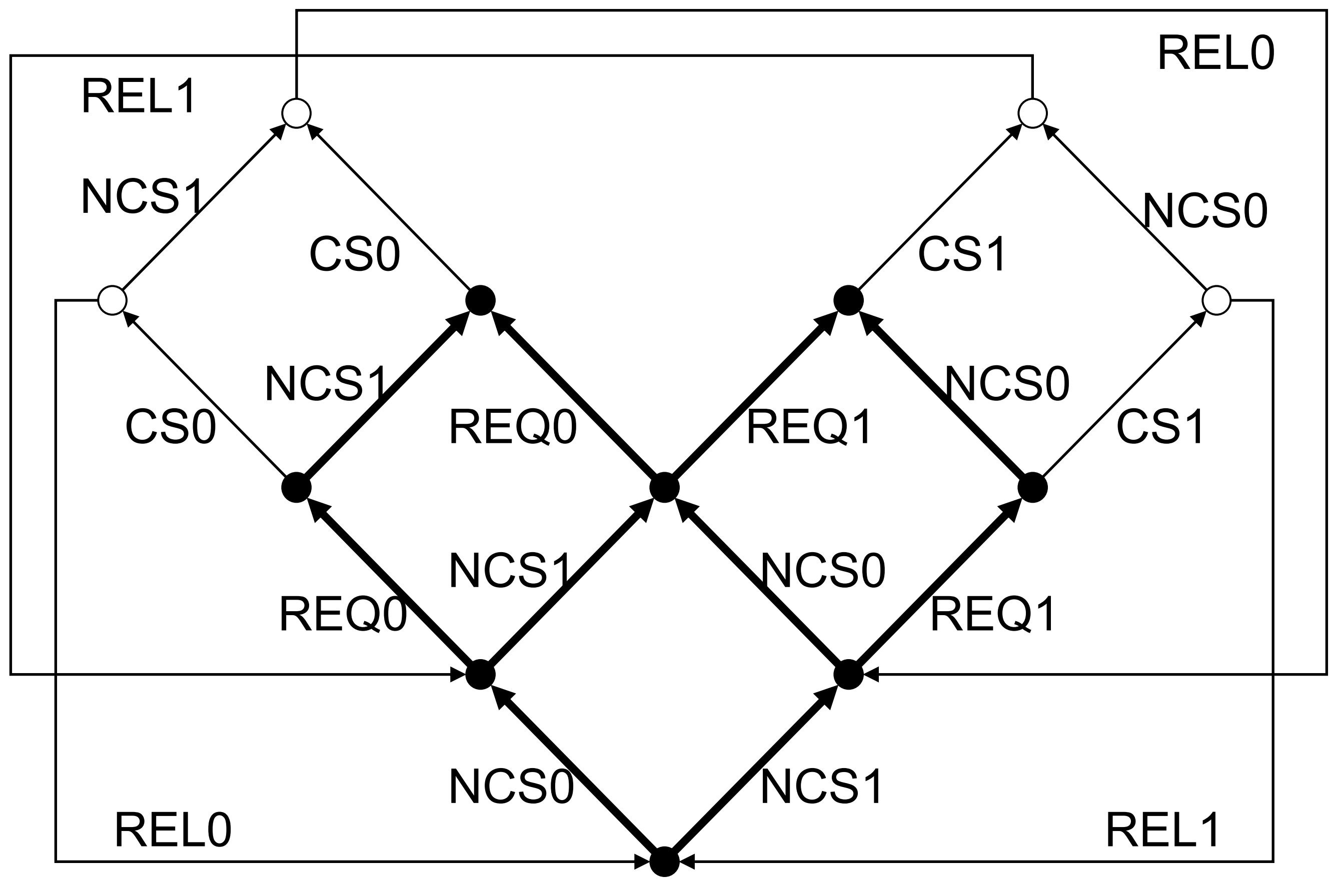
Example (1/4)

Potential reachability: $\mathbf{EF}_{\neg\, \mathbf{REL1}} \langle \mathbf{CS_0} \rangle\, \mathbf{true}$



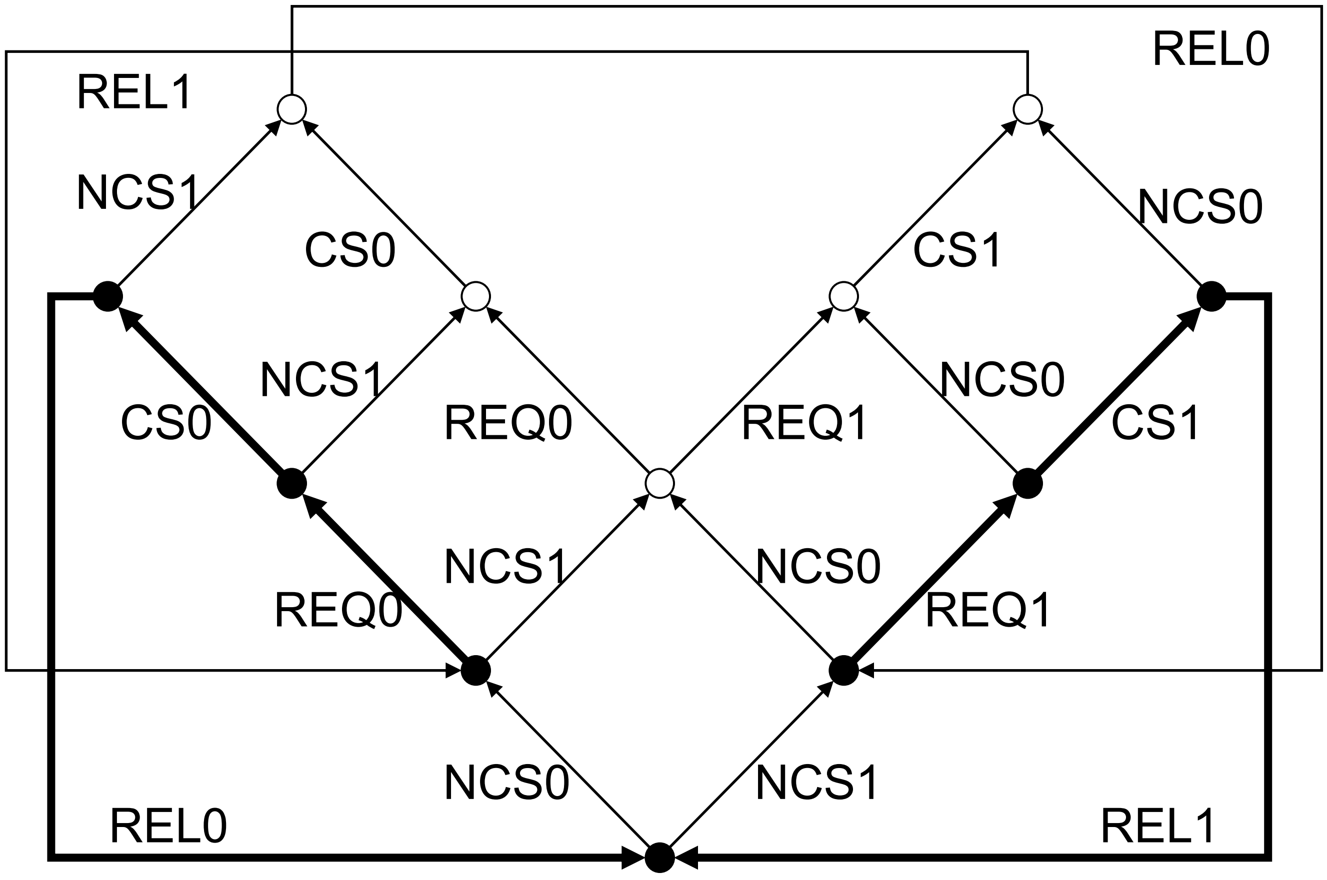
Example (2/4)

Inevitable reachability: $\mathbf{AF}_{\neg\,(\mathbf{REL0} \vee \mathbf{REL1})} \langle\, \mathbf{CS_0} \vee \mathbf{CS_1} \,\rangle\, \mathbf{true}$



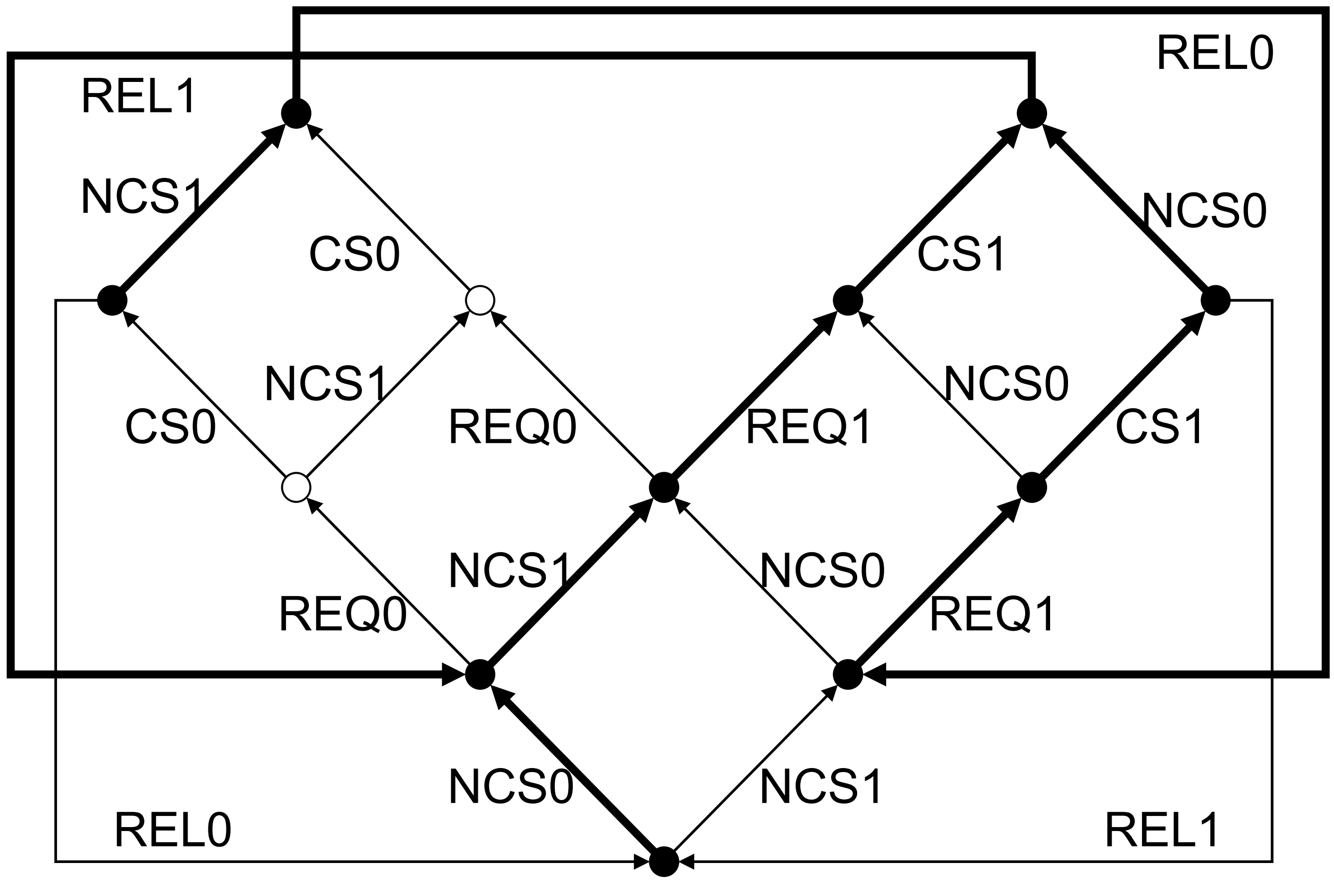
Example (3/4)

Invariance: $\mathbf{AG}_{\neg\,(\mathbf{NCS0} \vee \mathbf{NCS1})} \langle\, \mathbf{NCS_0} \vee \mathbf{NCS_1} \,\rangle\, \mathbf{true}$



Example (4/4)

Trajectory: $\mathbf{EG}_{\neg\, CS0}\ [\ CS_0\ ]\ \mathbf{false}$

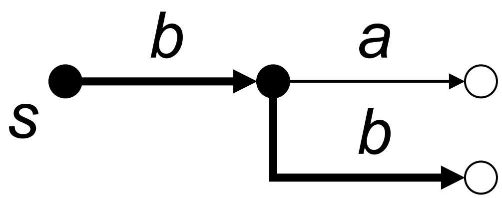


Remark about inevitability

- ***Inevitable reachability***: all sequences going out of a state lead to states where an action *a* is executable

  $\mathrm{AF}_{\mathrm{tt}} \langle a \rangle \mathrm{true}$

- ***Inevitable execution***: all sequences going out of a state contain the action *a*
- Inevitable execution $\Rightarrow$ inevitable reachability
  but the converse does not hold:



$s \models \mathrm{AF}_{\mathrm{tt}} \langle a \rangle \mathrm{true}$

- Inevitable execution must be expressed using the basic inevitability operators of ACTL:

$s \not\models \mathbf{A}\,[\, \mathbf{tt}_{\mathbf{tt}}\ \mathbf{U}_{a}\ \mathbf{true}\,]$

Safety properties

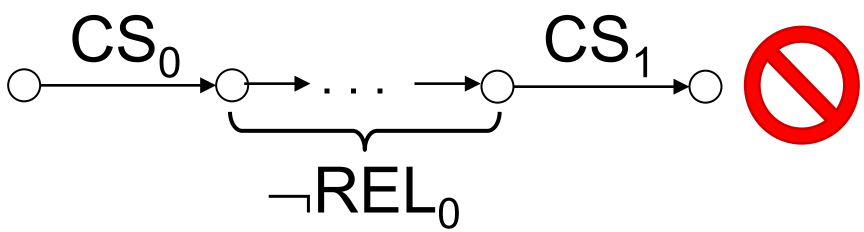
- Informally, safety properties specify that
  "something bad never happens"
  during the execution of the system
- One way of expressing safety properties:

  forbid undesirable execution sequences

  - Mutual exclusion:

    $\neg \langle \mathrm{CS}_0 \rangle \, \mathrm{EF}_{\neg \mathrm{REL0}} \langle \mathrm{CS}_1 \rangle$ true

    $= [\mathrm{CS}_0] \, \mathrm{AG}_{\neg \mathrm{REL0}} [\mathrm{CS}_1]$ false



- In ACTL, forbidding a sequence is expressed by combining the $[\, \alpha \,] \, \varphi$ and $\mathbf{AG}_{\alpha} \, \varphi$ operators

Liveness properties

- Informally liveness properties specify that
  "something good eventually happens"
  during the execution of the system
- One way of expressing liveness properties:

  require desirable execution sequences / trees

  - Potential release of the critical section:

    $\langle NCS_0 \rangle EF_{tt} \langle REQ_0 \rangle EF_{tt} \langle REL_0 \rangle$ true
  - Inevitable access to the critical section:

    A [tt_{tt} U_{CS0} true]
- In ACTL, the existence of a sequence is expressed by combining the $\langle \alpha \rangle \varphi$ and $\mathbf{EF}_{\alpha}\ \varphi$ operators

Branching-time logics

(summary)

- The temporal operators of ACTL: strictly more powerful than the HML modalities $\langle\, \alpha \,\rangle\, \varphi$ and $[\, \alpha \,]\, \varphi$
- They allow to express branching-time properties on an unbounded depth in an LTS
- But:
  - They do not allow to express the unbounded repetition of a subsequence
- Example: the property

  "from a state *s*, there exists a sequence *a*.*b*.*a*.*b* ... *a*.*b* leading to a state *s*' where an action *c* is executable"

  cannot be expressed in ACTL