

Tecniche di Specifica e di Verifica

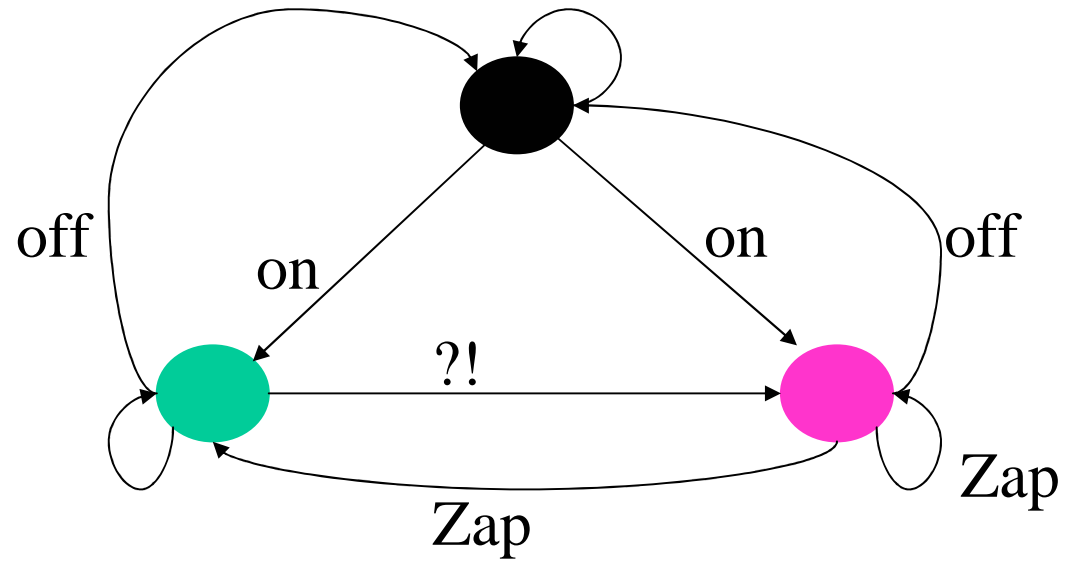
Branching Time Temporal Logics

Outline

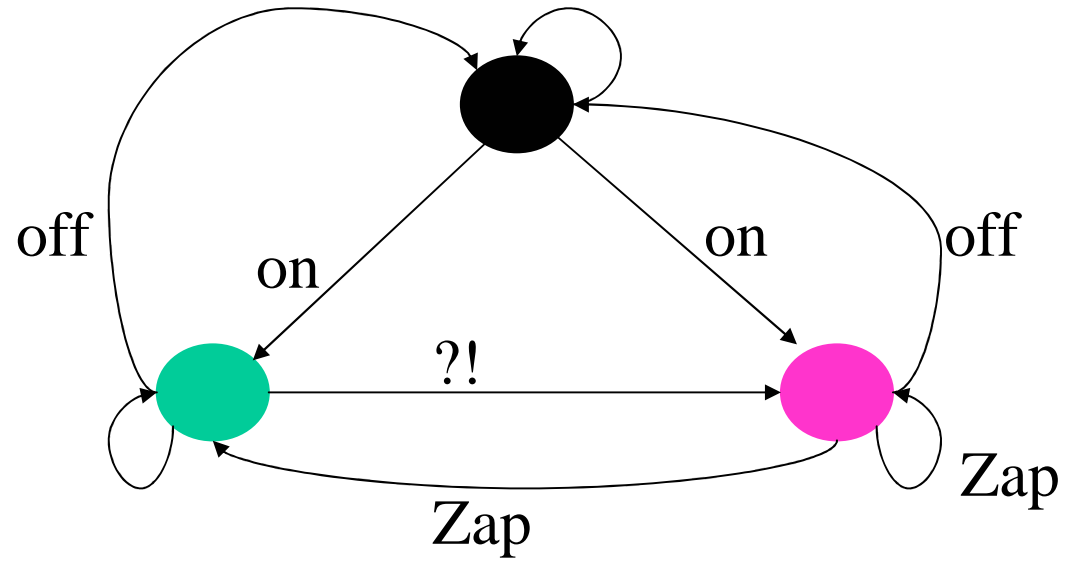
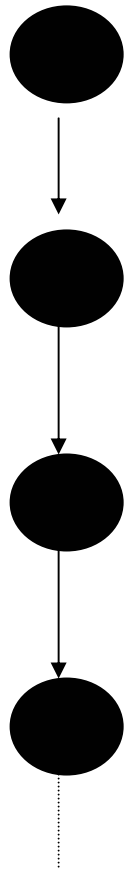
- ***CTL*** (**C**omputation **T**ree **L**ogic)
 - **Branching Time**
 - Unwindings --- computation trees
 - Syntax and semantics of CTL.

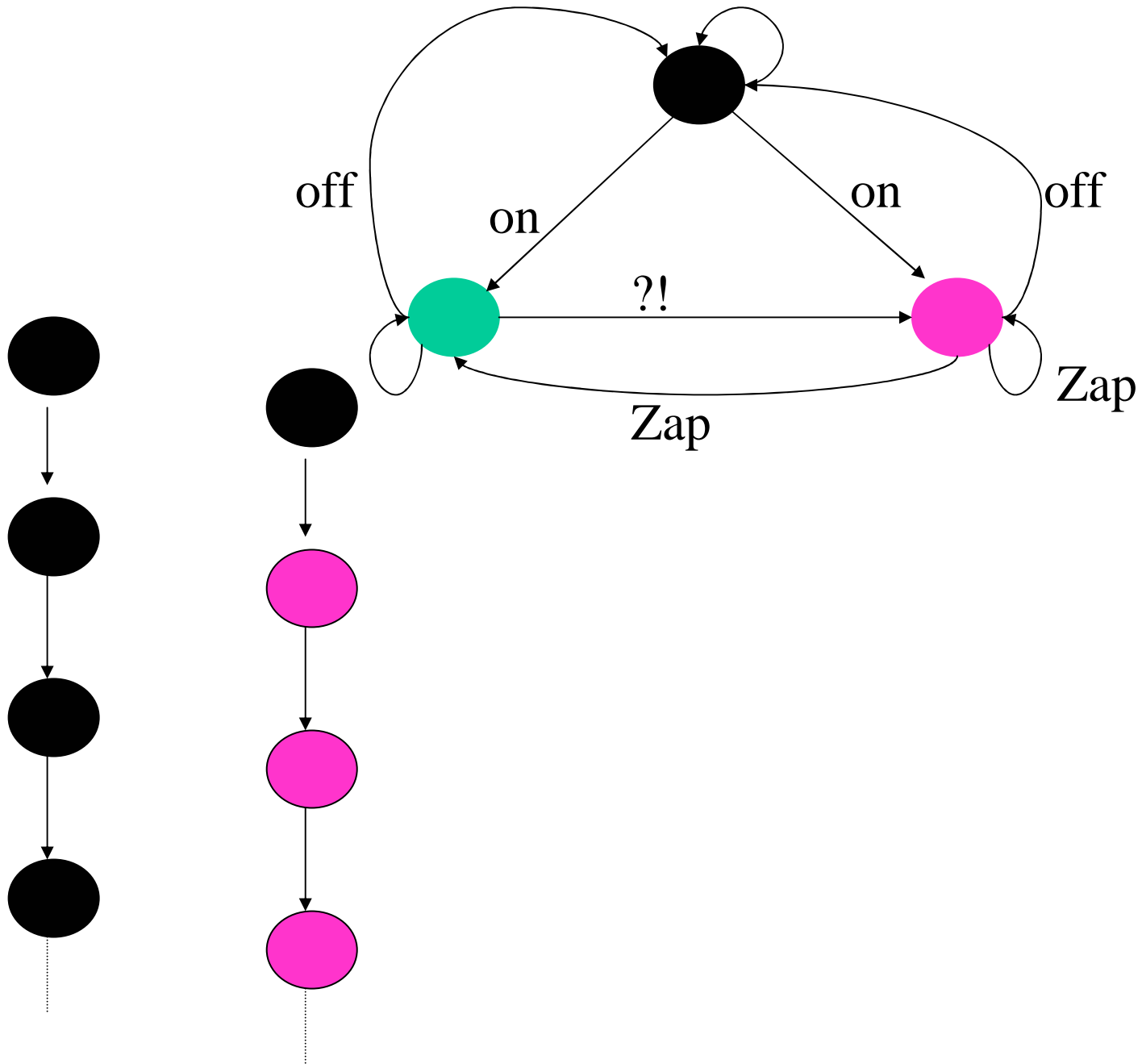
Branching Time Structures

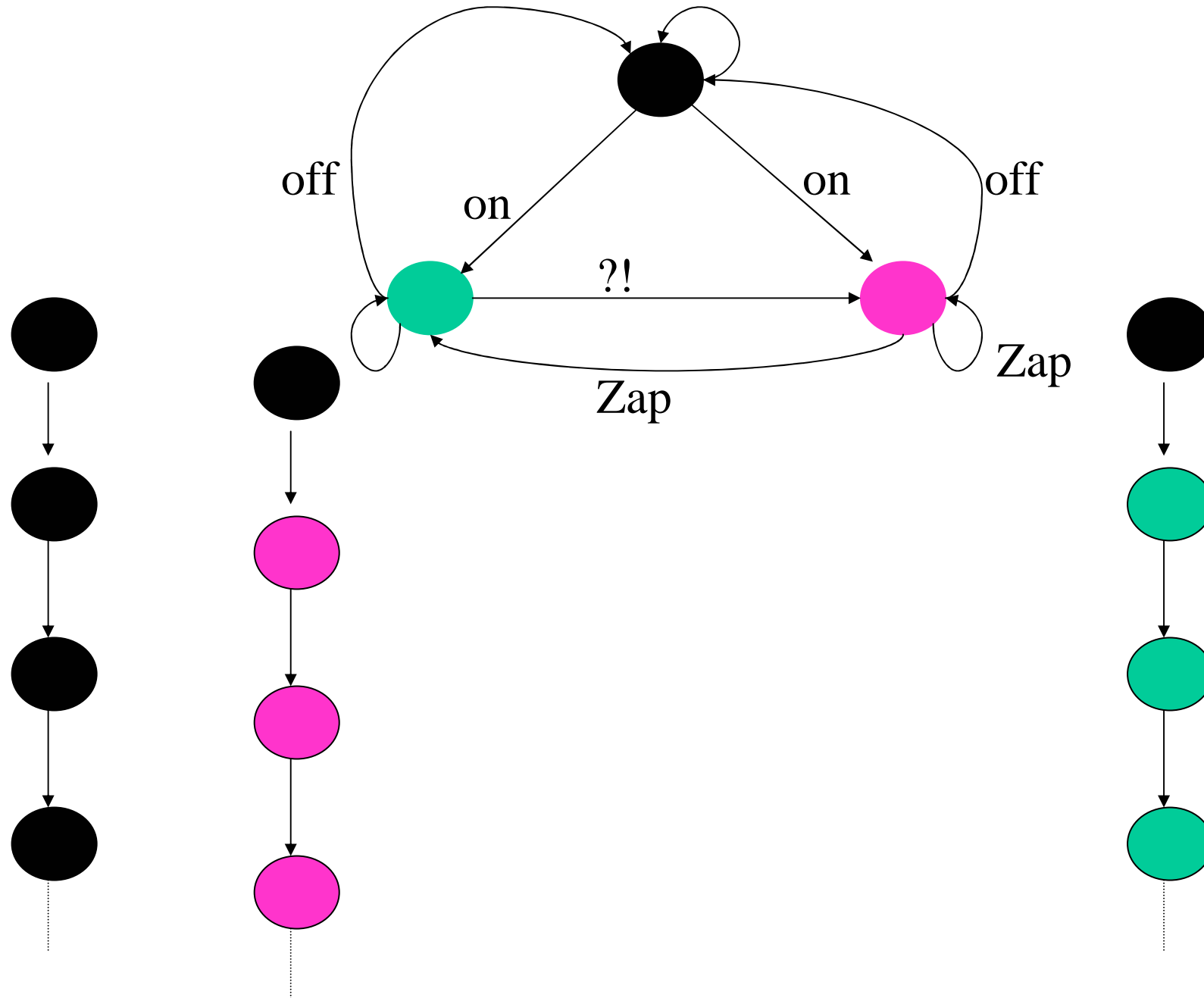
- **Linear Time:**
 - A *computation* at its first state satisfies a property.
 - Property ---- **LTL** formula
- **Branching Time**
 - The *computation tree* at its root satisfies a property.
 - Property: **CTL** (**CTL***, **μ -calculus**) formula.
 - **Computation Tree**
 - *All computations* starting from a state *glued together* (to form a tree structure).
- In branching time, *the decisions* taken during a run are taken into account.

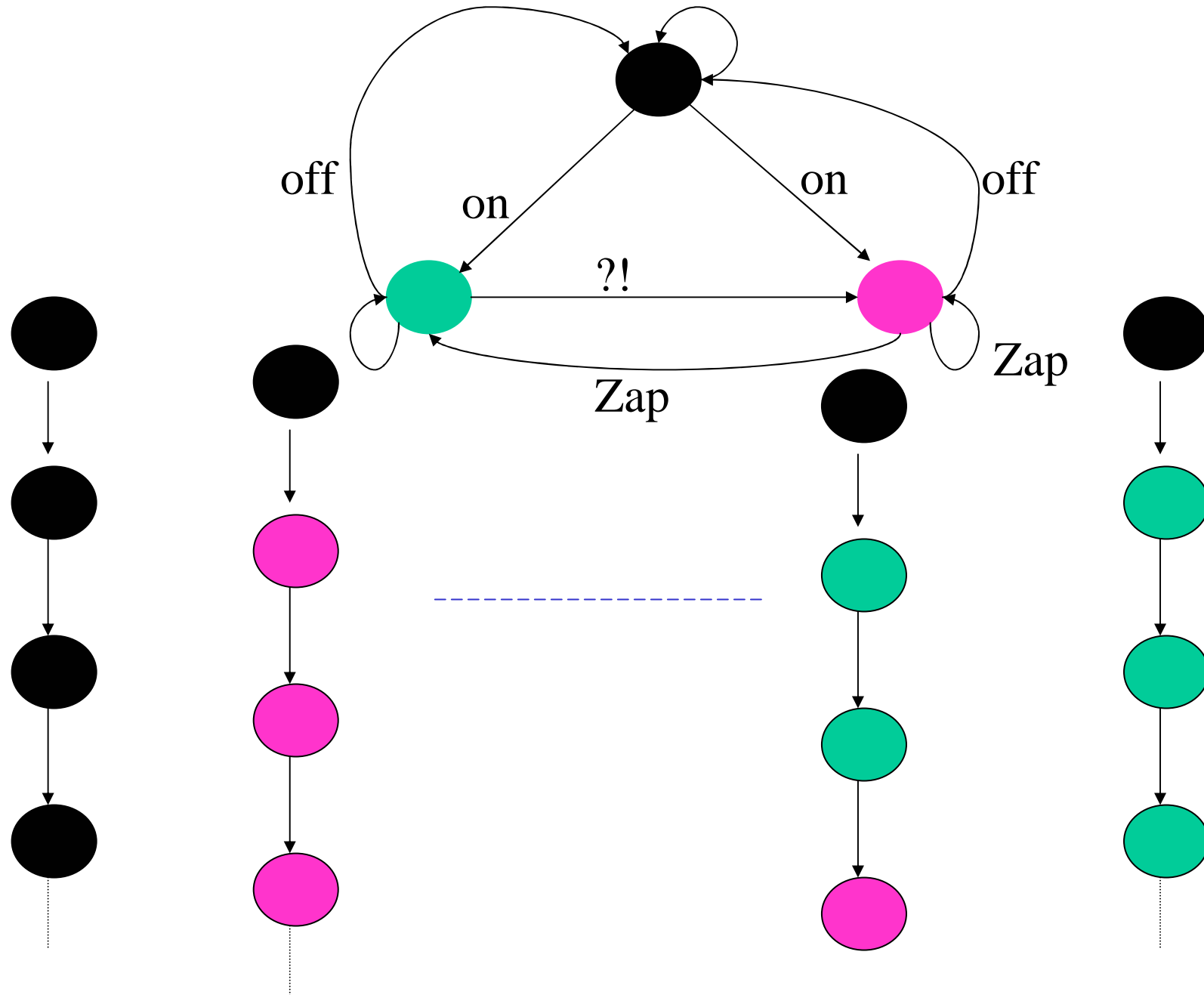


The TV Example

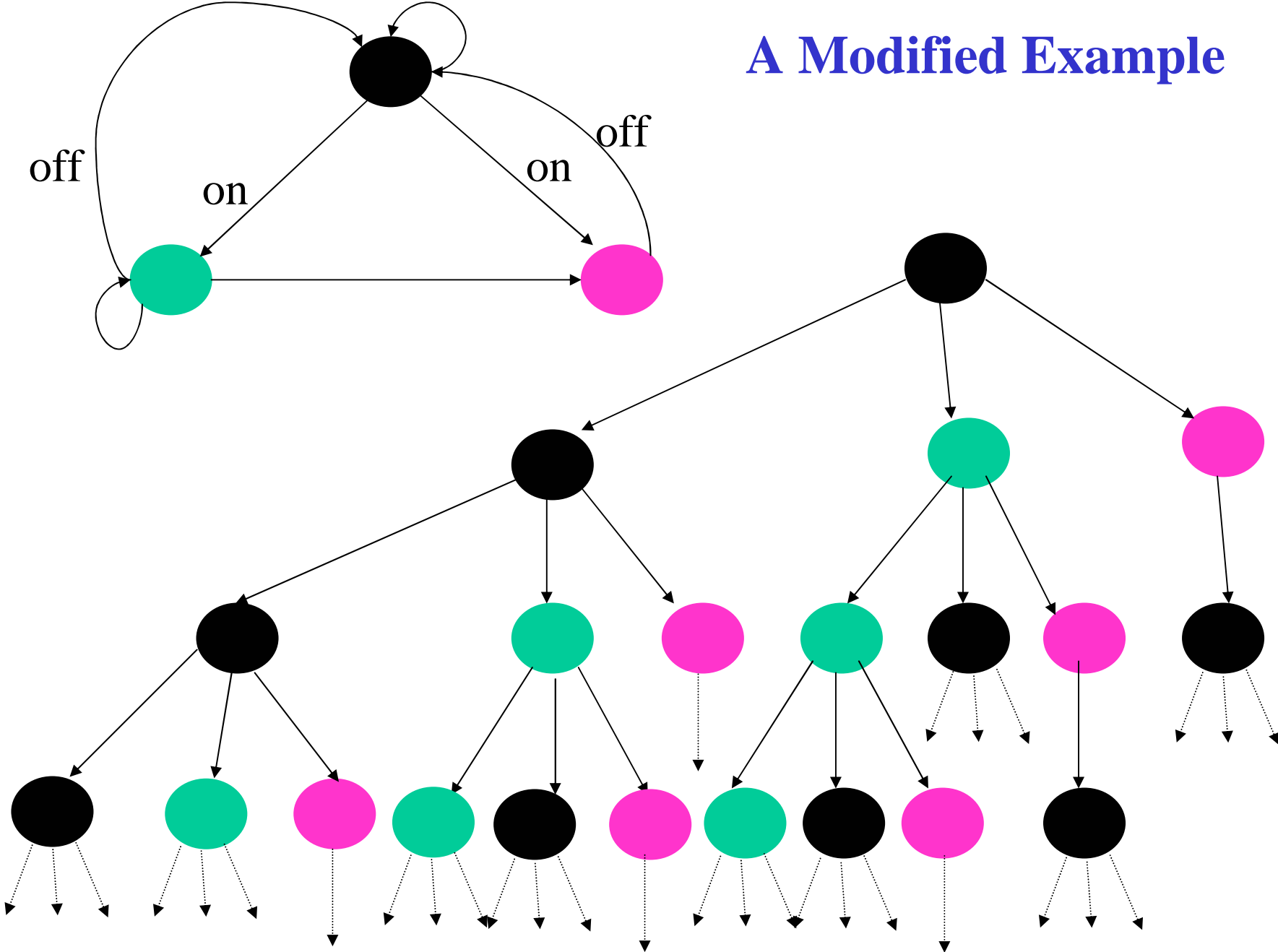








A Modified Example



Branching Time Temporal Logic

- $\mathbf{K} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, \mathbf{AP}, \mathbf{L})$
- $\mathbf{K}, \mathbf{s} \models \psi$ -- the computation tree rooted at \mathbf{s} satisfies ψ .
- $\mathbf{K} \models \psi$ iff $\mathbf{K}, \mathbf{s}_0 \models \psi$ for every $\mathbf{s}_0 \in \mathbf{S}_0$.
- **Branching Time Temporal Logics:**
 - **CTL**
 - **CTL***
 - (The modal) **μ -calculus**

Unwinding

- $\mathbf{K} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, \mathbf{AP}, \mathbf{L}) \quad \mathbf{s}_0 \in \mathbf{S}$
- $\mathbf{TR}(\mathbf{K}, \mathbf{s}_0)$ --- The computation tree rooted at \mathbf{s}_0 .

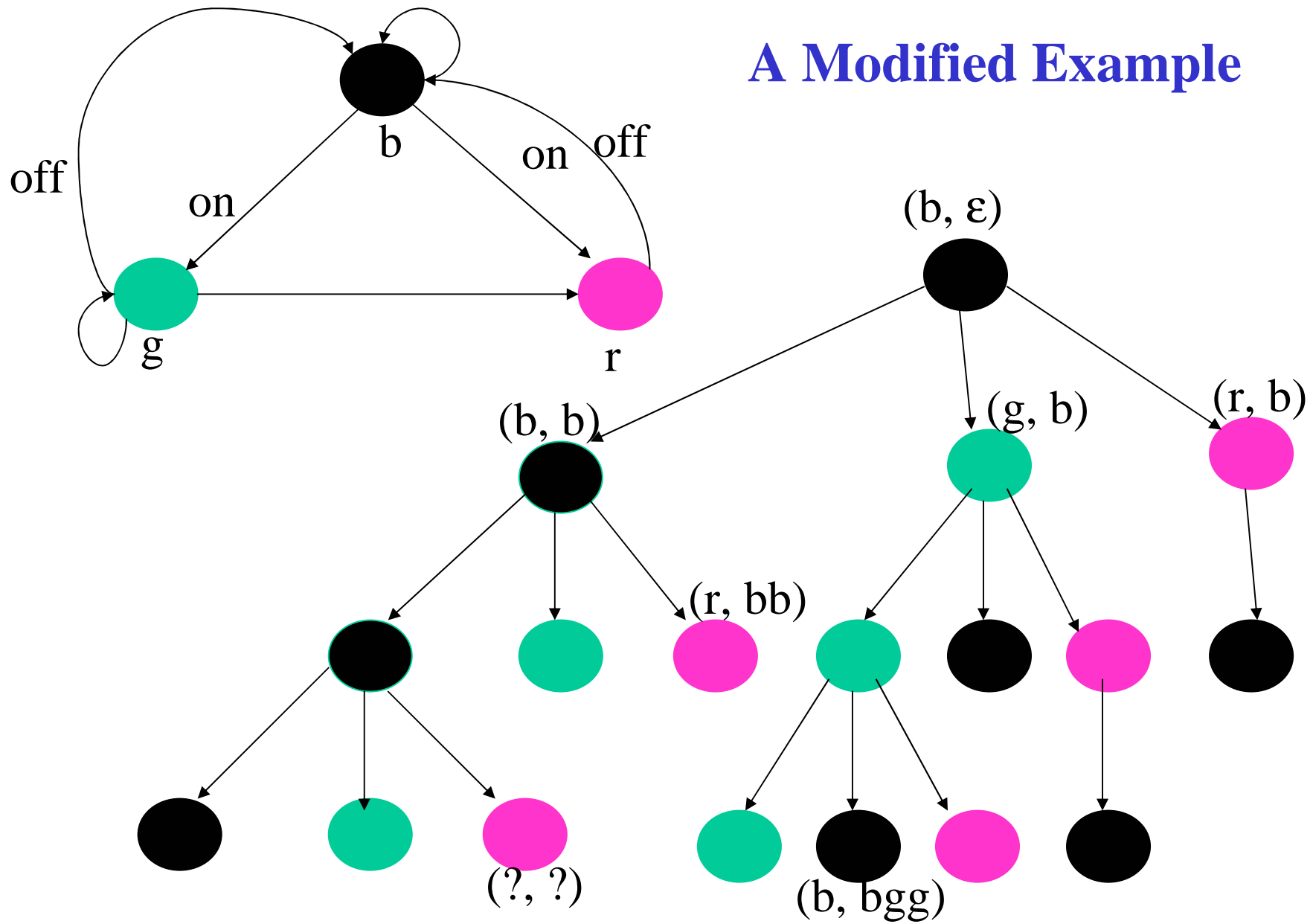
- $\mathbf{TR}(\mathbf{K}, \mathbf{s}_0) = (\mathcal{S}_{\mathbf{s}_0}, (\mathbf{s}_0, \epsilon), \mathcal{R}_{\mathbf{s}_0}, \mathbf{AP}, \mathcal{L}_{\mathbf{s}_0})$
 - $(\mathbf{s}_0, \epsilon) \in \mathcal{S}_{\mathbf{s}_0}$;
 - If $(\mathbf{s}_1, \sigma) \in \mathcal{S}_{\mathbf{s}_0}$ and $\mathbf{R}(\mathbf{s}_1, \mathbf{s}_2)$ then
 - $(\mathbf{s}_2, \sigma.\mathbf{s}_1) \in \mathcal{S}_{\mathbf{s}_0}$ and
 - $\mathcal{R}_{\mathbf{s}_0}((\mathbf{s}_1, \sigma), (\mathbf{s}_2, \sigma.\mathbf{s}_1))$;
 - $\mathcal{L}((\mathbf{s}_1, \sigma)) = \mathbf{L}(\mathbf{s}_1)$.

Therefore, for all $(\mathbf{s}, \sigma) \in \mathcal{S}_{\mathbf{s}_0}$, $\mathbf{s} \in \mathbf{S}$ and $\sigma = \mathbf{s}_0 \mathbf{s}_1 \dots \mathbf{s}_n$ is a path in \mathbf{K} from \mathbf{s}_0 to \mathbf{s}_n and $\mathbf{R}(\mathbf{s}_n, \mathbf{s})$ (hence, $\sigma.\mathbf{s}$ is a path in \mathbf{K} from \mathbf{s}_0 to \mathbf{s})

Unwinding

- $\text{TR}(\mathbf{K}, \mathbf{s})$ is almost a Kripke structure.
 - \mathcal{S}_s will typically be infinite.
 - But \mathcal{R}_s is *tree-like*.
 - The “*graph*” of $\text{TR}(\mathbf{K}, \mathbf{s})$ is a tree rooted at $(\mathbf{s}, \varepsilon)$.
- $\text{TR}(\mathbf{K}, \mathbf{s})$ is the *computation tree rooted* of \mathbf{K} at \mathbf{s} .

A Modified Example



Linear time Vs Branching time

- There are *properties* that can be *expressed in LTL* but which *can not be expressed in CTL*.
- There are *properties* that can be *expressed in CTL* but *not in LTL*.
- The *LTL model checking problem* can be *converted* into a *restricted* kind of a *CTL* model checking problem*.

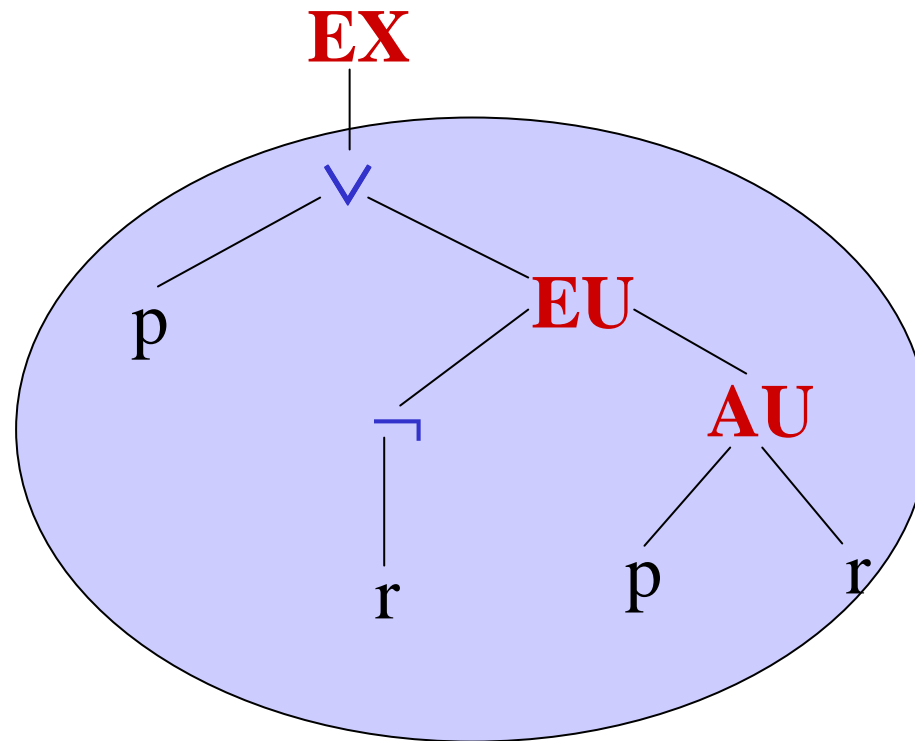
CTL

- **Syntax**

- **AP** – a finite set of *atomic propositions*.
- $p \in \mathbf{AP}$ is a formula.
- If ψ and ψ' are formulas then so are $\neg\psi$ and $\psi \vee \psi'$.
- If ψ is a formula then so is **EX** ψ
- If ψ_1 and ψ_2 are formulas then so are **EU**(ψ_1, ψ_2) and **AU**(ψ_1, ψ_2).

Formulas

- **EX**(**p** \vee **EU**(\neg **r**, **AU**(**p**, **r**)))



Semantics

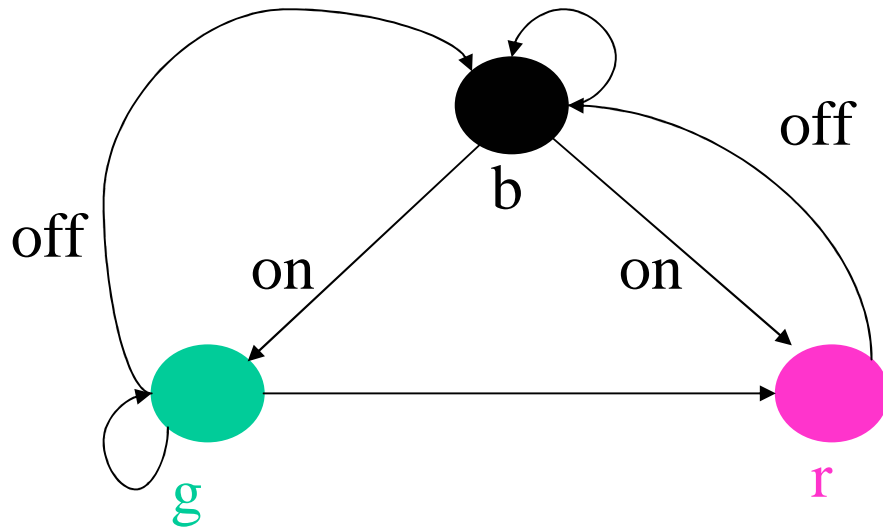
- $\mathbf{K} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, \mathbf{AP}, \mathbf{L})$
 - $\mathbf{L} : \mathbf{S} \rightarrow 2^{\mathbf{AP}}$
- ψ a **CTL** formula and $\mathbf{s} \in \mathbf{S}$
- $\mathbf{K}, \mathbf{s} \models \psi$
- ψ (holds) *is satisfied* at \mathbf{s} .
- **FACT:**
 $\mathbf{K}, \mathbf{s} \models \psi$ iff $\mathbf{TR}(\mathbf{K}, \mathbf{s}), (\mathbf{s}, \epsilon) \models \psi$.

Semantics

- **CTL ::= p | $\neg\psi$ | $\psi_1 \vee \psi_2$ | **EX**(ψ) |
| **EU**(ψ_1, ψ_2) | **AU**(ψ_1, ψ_2)**
- **K = (S, S₀, R, AP, L) ; L: S \longrightarrow 2^{AP} ; s \in S**
- **K, s \models p iff p \in L(s).**
- **K, s $\models \neg\psi$ iff not K, s $\models \psi$**
- **K, s $\models \psi_1 \vee \psi_2$ iff
K, s $\models \psi_1$ or K, s $\models \psi_2$.**

Semantics

- **CTL ::= p | $\neg\psi$ | $\psi_1 \vee \psi_2$ | **EX**(ψ) |
| **EU**(ψ_1, ψ_2) | **AU**(ψ_1, ψ_2)**
- **K = (S, S₀, R, AP, L)** ; **L: S \longrightarrow 2^{AP}** ; **s \in S**
- **K, s \models EX(ψ)** iff there exists **s'** such that:
 - **s \longrightarrow s'** (i.e. **R(s, s')**) and **K, s' $\models \psi$****s** has a successor state **s'** at which **ψ** holds.



$AP = \{n, h, uh\}$

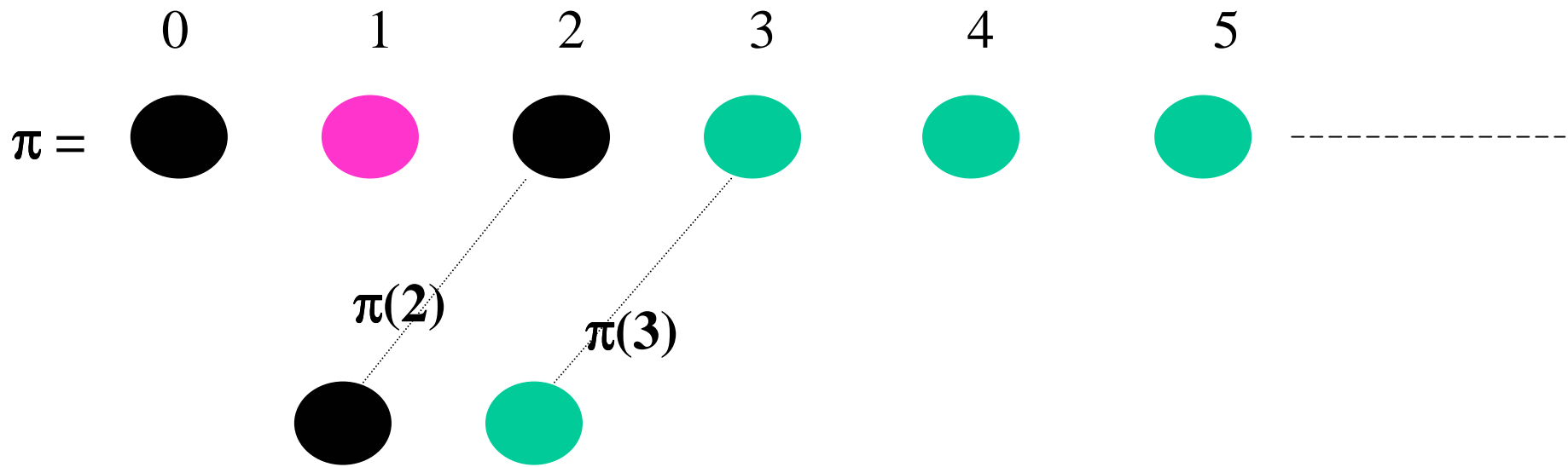
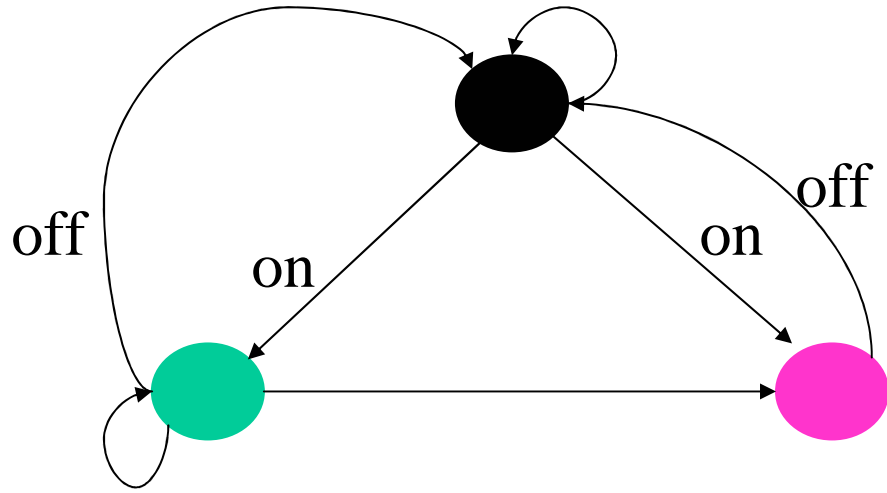
$K, b \models EX(uh) ? \quad K, b \models EX(\neg uh) ?$

$K, g \models EX(uh) ?$

$K, r \models EX(h) ?$

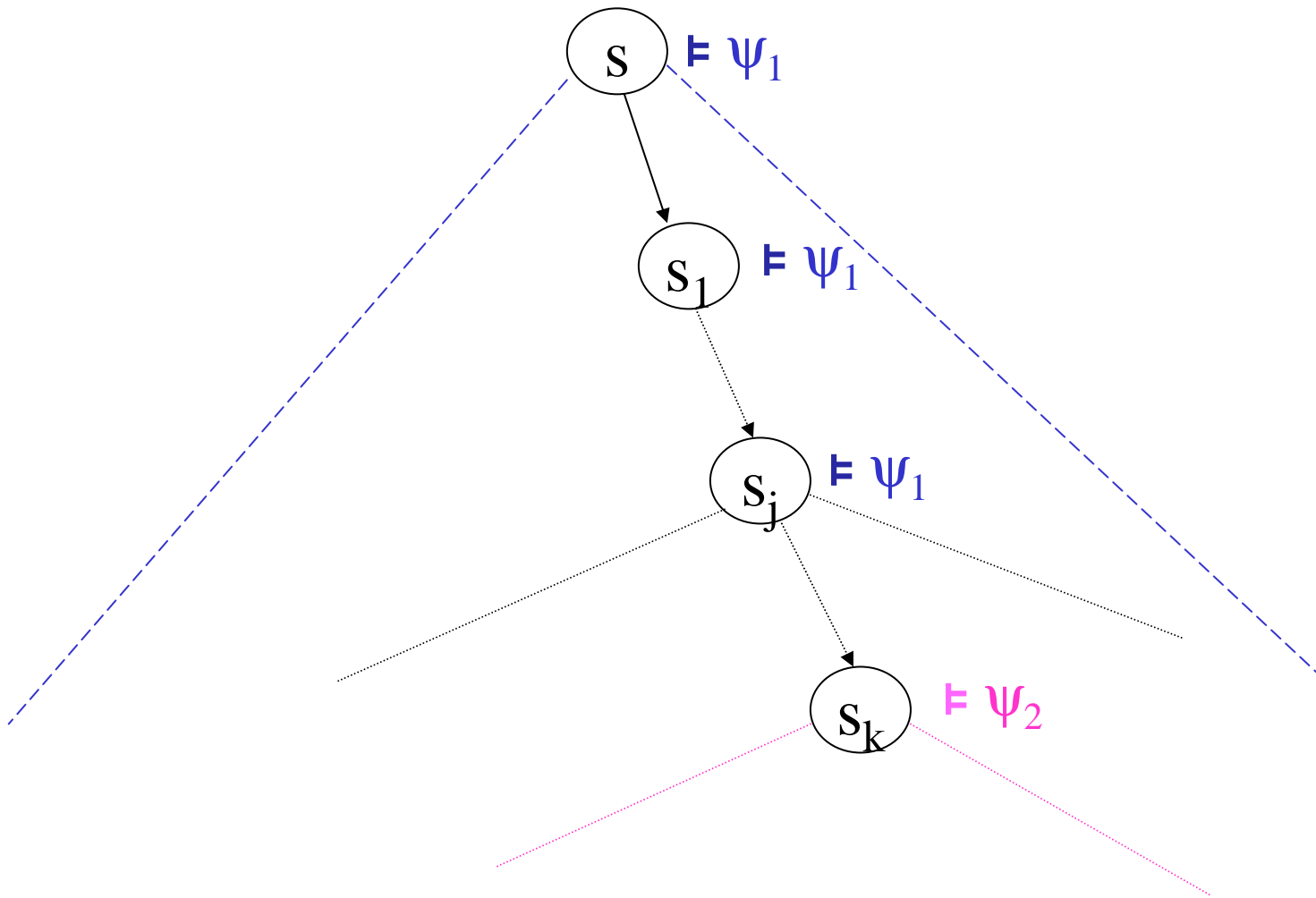
Semantics

- $\mathbf{K} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, \mathbf{AP}, \mathbf{L})$; $\mathbf{L}: \mathbf{S} \longrightarrow 2^{\mathbf{AP}}$; $\mathbf{s} \in \mathbf{S}$
- *A path from \mathbf{s}* is a (infinite) sequence of states
 $\pi = \mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_i, \mathbf{s}_{i+1}, \dots$ s.t:
 - $\mathbf{s} = \mathbf{s}_0$
 - $\mathbf{s}_i \longrightarrow \mathbf{s}_{i+1}$ (i.e. $\mathbf{R}(\mathbf{s}_i, \mathbf{s}_{i+1})$) for every \mathbf{i} .
- $\pi(\mathbf{i}) = \mathbf{s}_i$ the \mathbf{i} -th element of π .



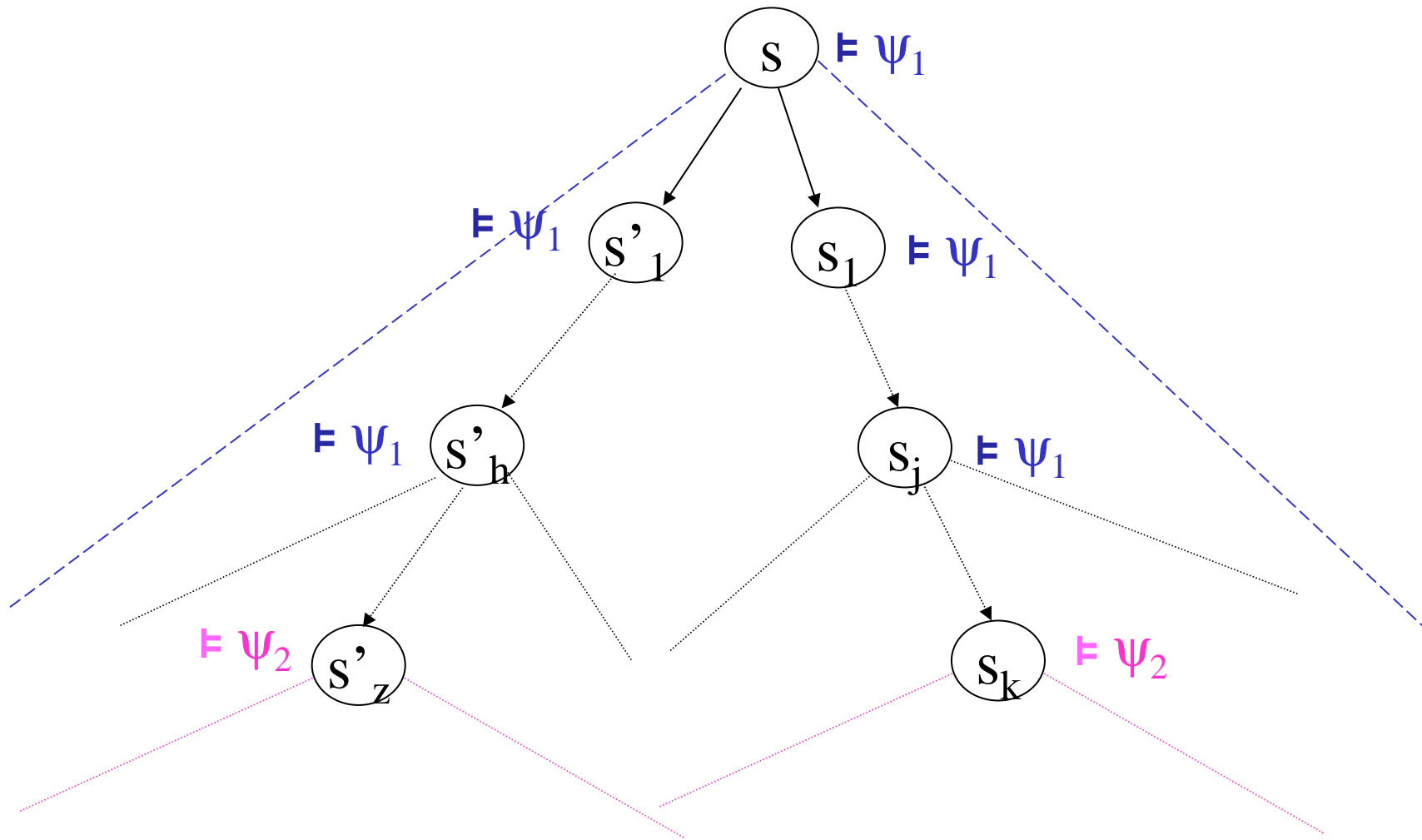
Semantics

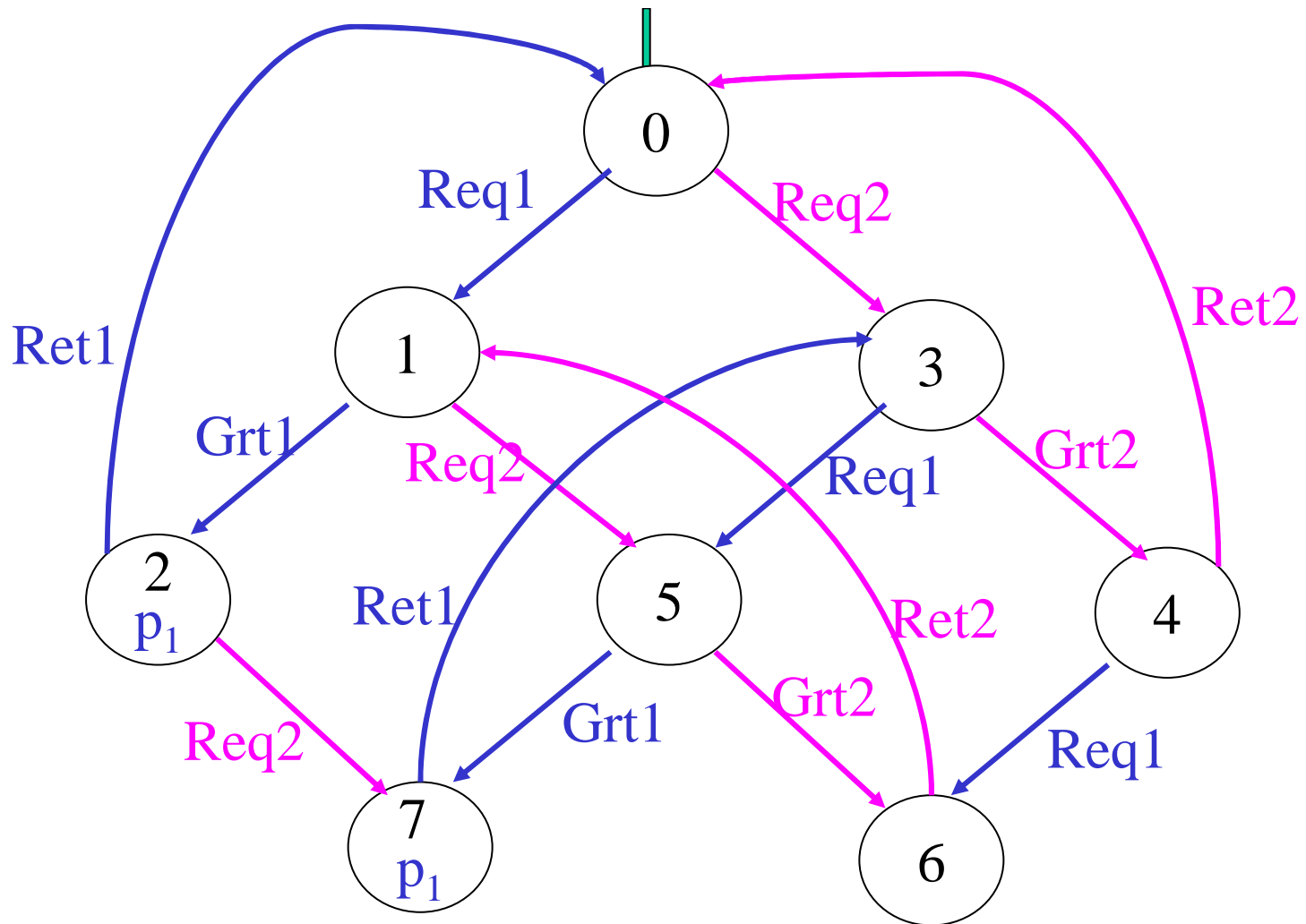
- **CTL ::= p | $\neg\psi$ | $\psi_1 \vee \psi_2$ | EX(ψ) |**
| EU(ψ_1, ψ_2) | AU(ψ_1, ψ_2)
- **$K = (S, S_0, R, AP, L)$; $L: S \longrightarrow 2^{AP}$; $s \in S$**
- **$K, s \models \text{EU}(\psi_1, \psi_2)$ iff *there exists a path***
 $\pi = s_0, s_1, \dots$ from s (i.e. $s_0=s$) and $k \geq 0$ such
that:
 - **$K, \pi(k) \models \psi_2$**
 - **$K, \pi(j) \models \psi_1$, for all $0 \leq j < k$.**



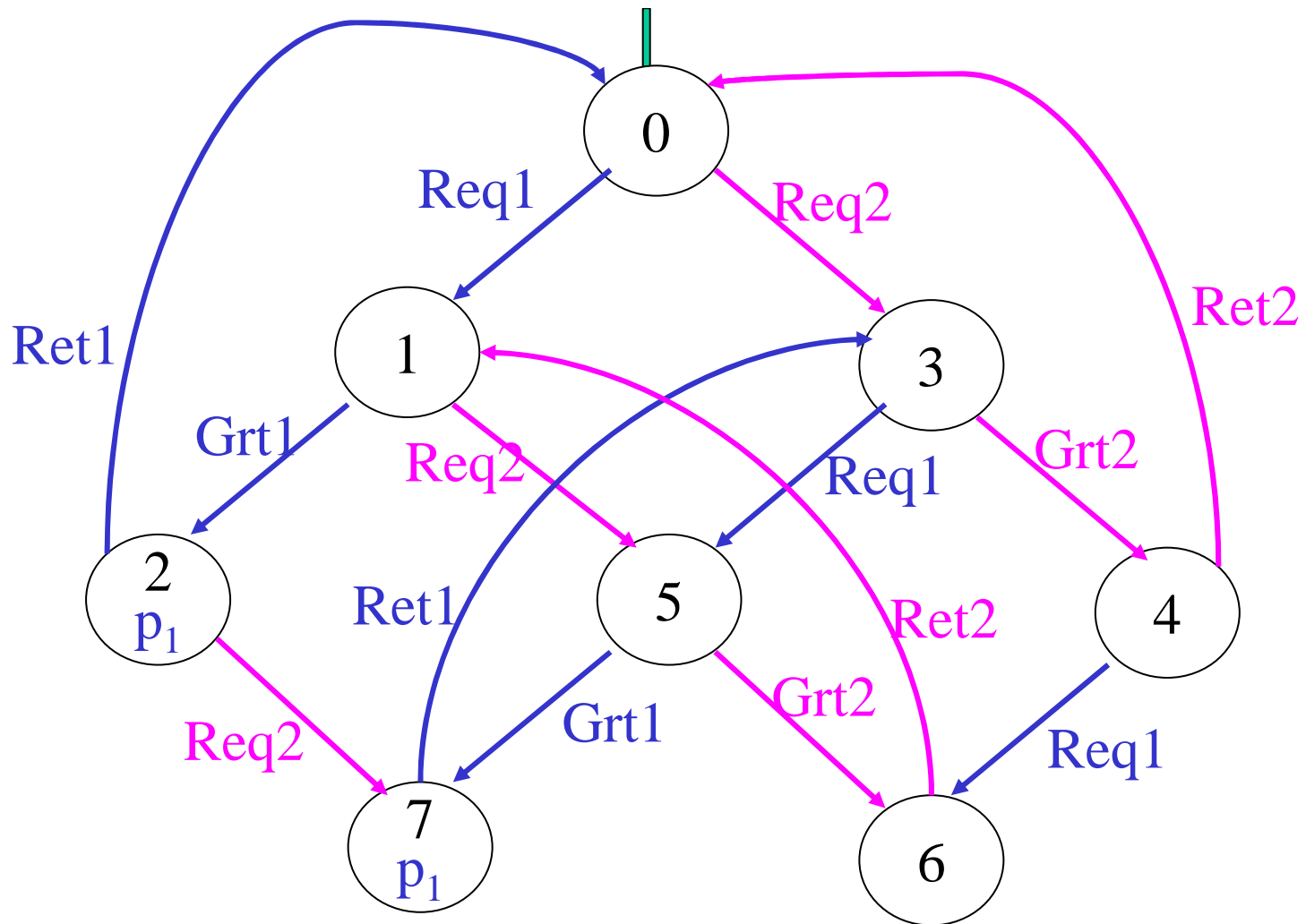
Semantics

- **CTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{AU}(\psi_1, \psi_2)$
- $\mathbf{K} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, \mathbf{AP}, \mathbf{L})$; $\mathbf{L}: \mathbf{S} \longrightarrow 2^{\mathbf{AP}}$; $s \in \mathbf{S}$
- $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$ iff *for every path*
 $\pi = s_0, s_1, \dots$ from s there exists $k \geq 0$ such that:
 - $\mathbf{K}, \pi(k) \models \psi_2$
 - $\mathbf{K}, \pi(j) \models \psi_1$, for all $0 \leq j < k$.

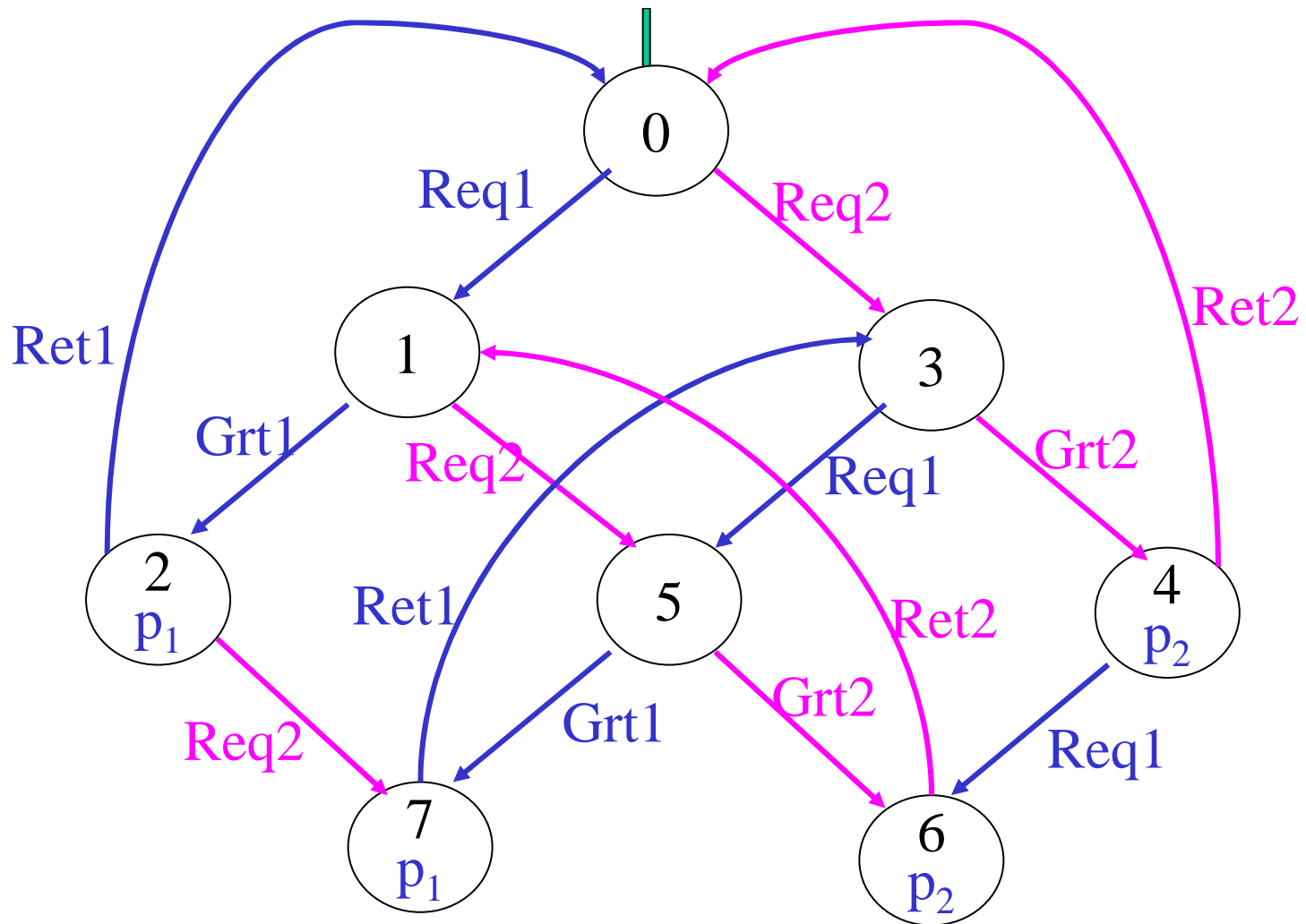




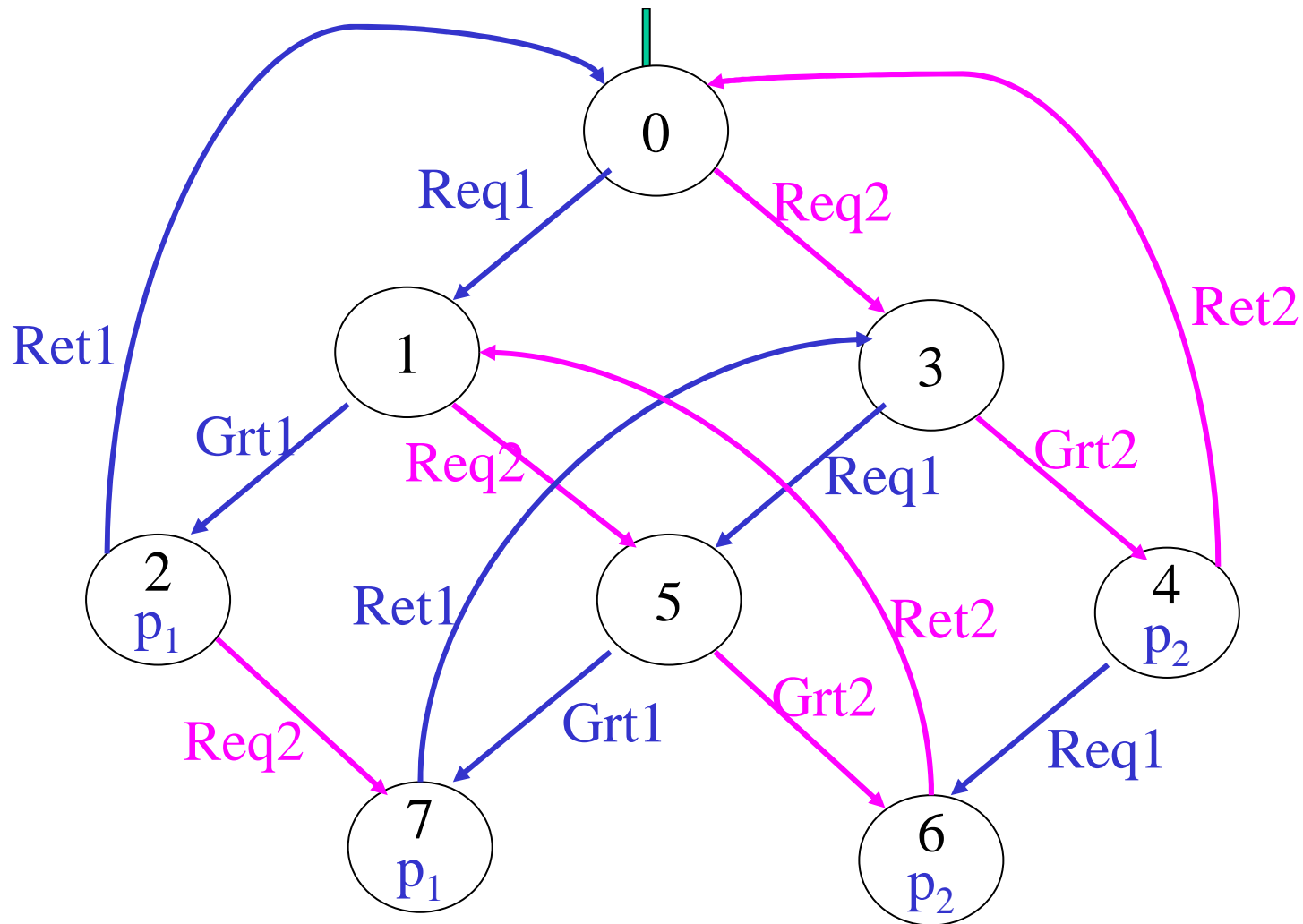
$M, 0 \models EU(\top, p_1)$?



$M, 0 \models AU(\tau, p_1) ?$

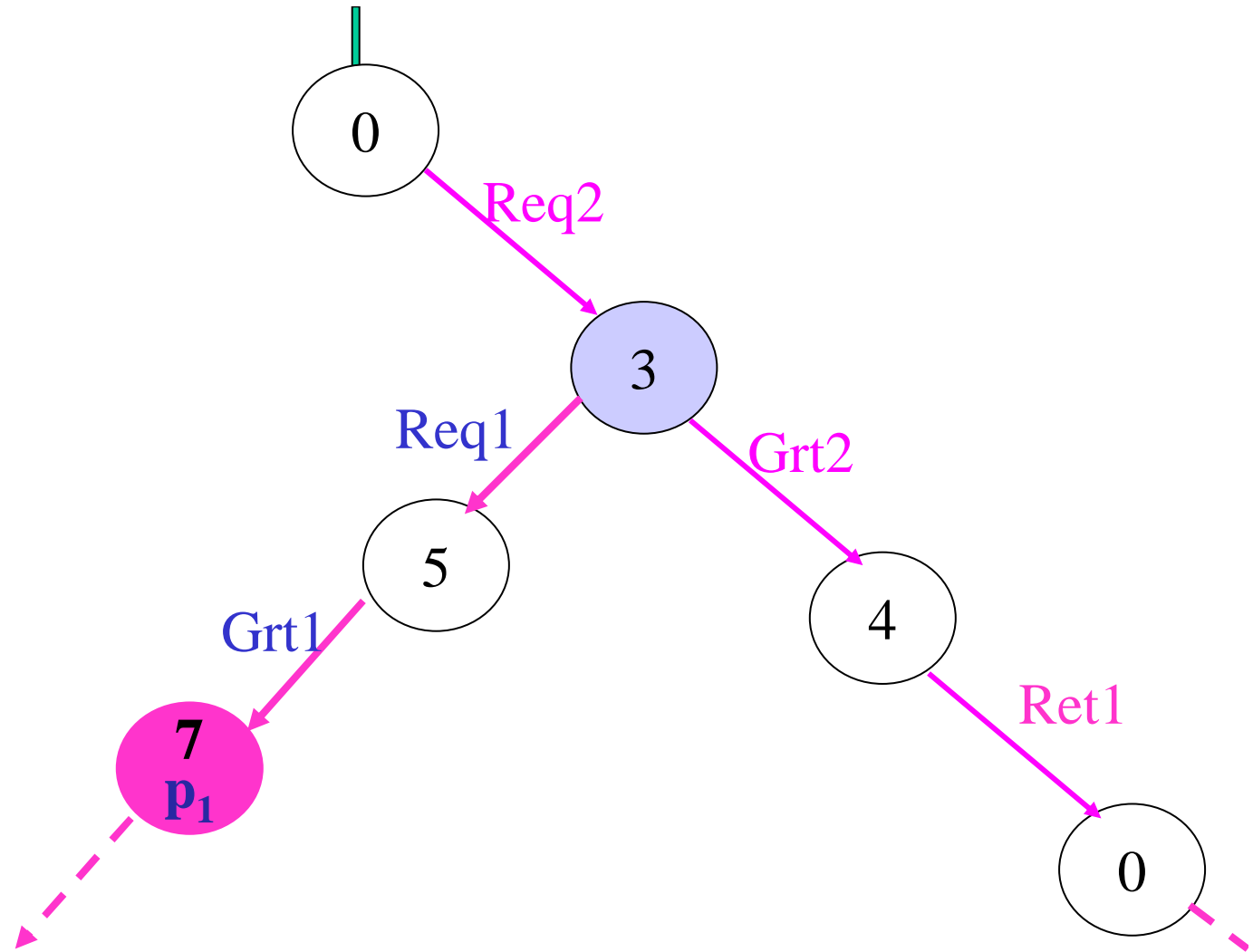


$M, 0 \models AU(\top, p_1 \vee p_2) ?$



$M, 0 \models \text{AU}(\top, \text{EU}(\top, p_1))$?

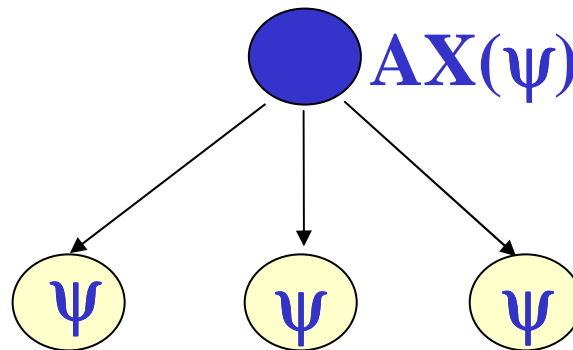
From s_0 , *all* the computations will reach a point, where it is *possible* for 1 to print *eventually*.



$M, 0 \models \text{AU}(\top, \text{EU}(\top, p_1))$?

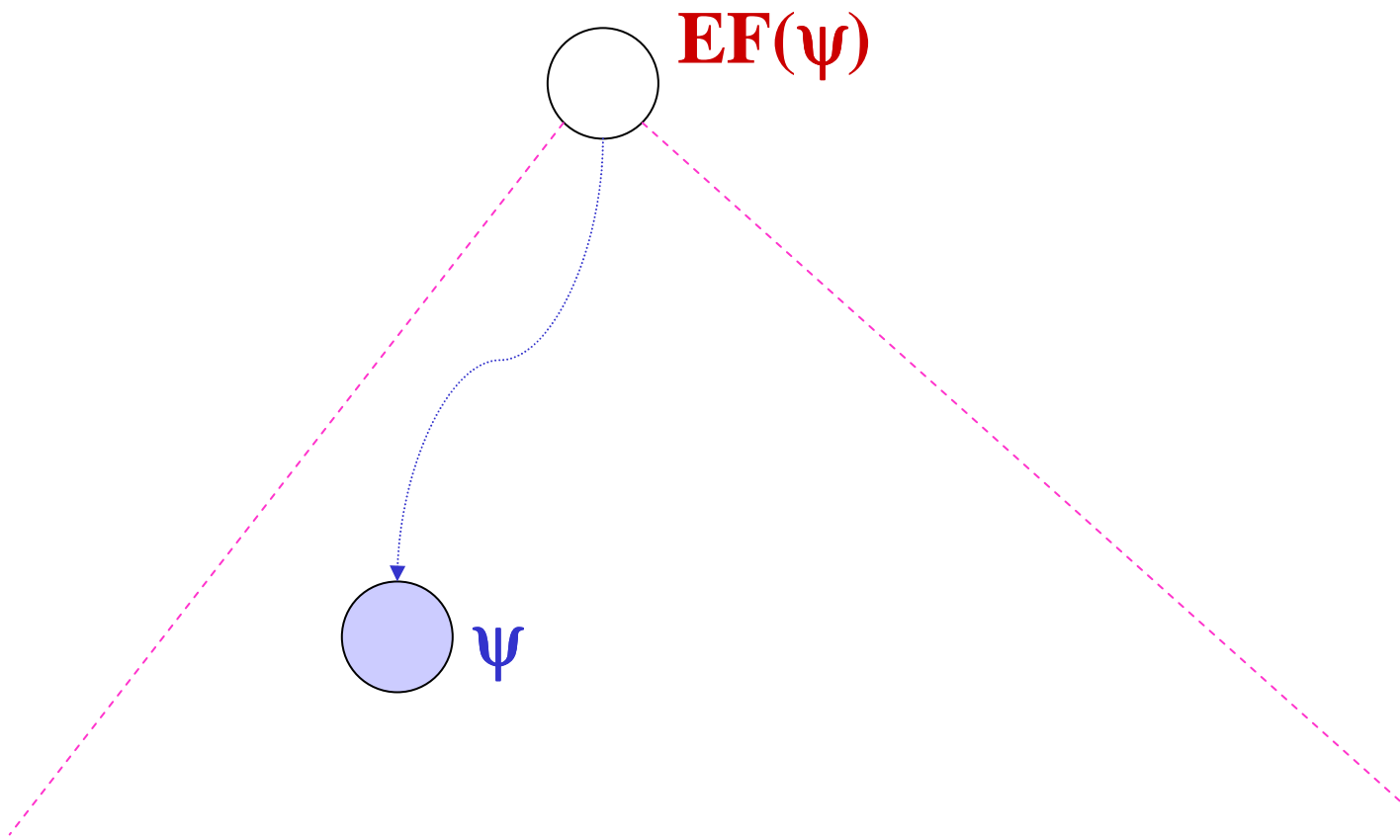
Derived Operators

- **$AX(\psi) = \neg EX(\neg\psi)$**
 - It is not the case there exists a next state at which ψ does not hold, equivalent to
 - *For every next state ψ holds.*



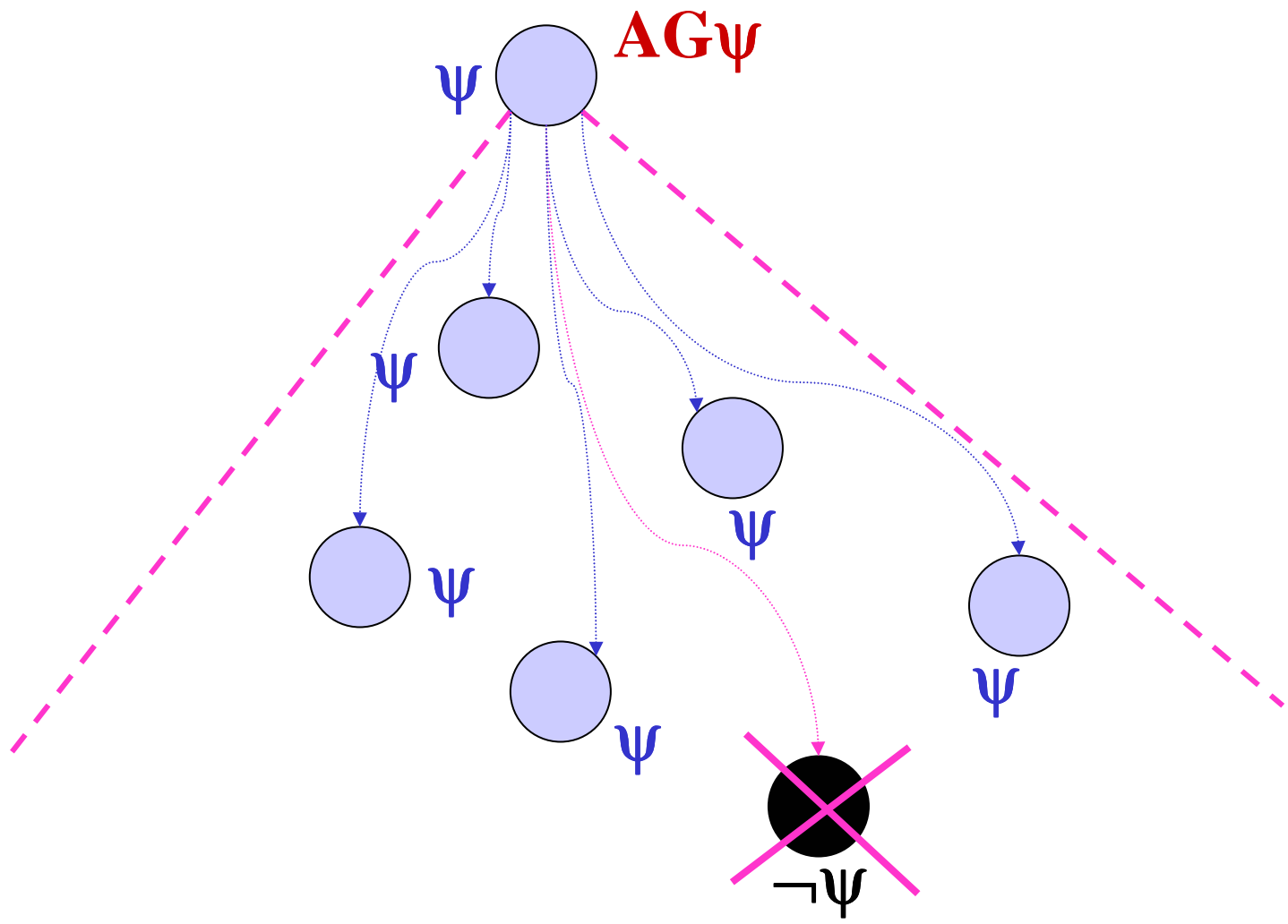
Derived Operators

- $\mathbf{K}, \mathbf{s} \models \mathbf{EF}(\psi)$
- $\mathbf{EF}(\psi) = \mathbf{EU}(\top, \psi)$
 - There exists a path π (from \mathbf{s}) and $\mathbf{k} \geq \mathbf{0}$ such that:
 - $\mathbf{K}, \pi(\mathbf{k}) \models \psi.$



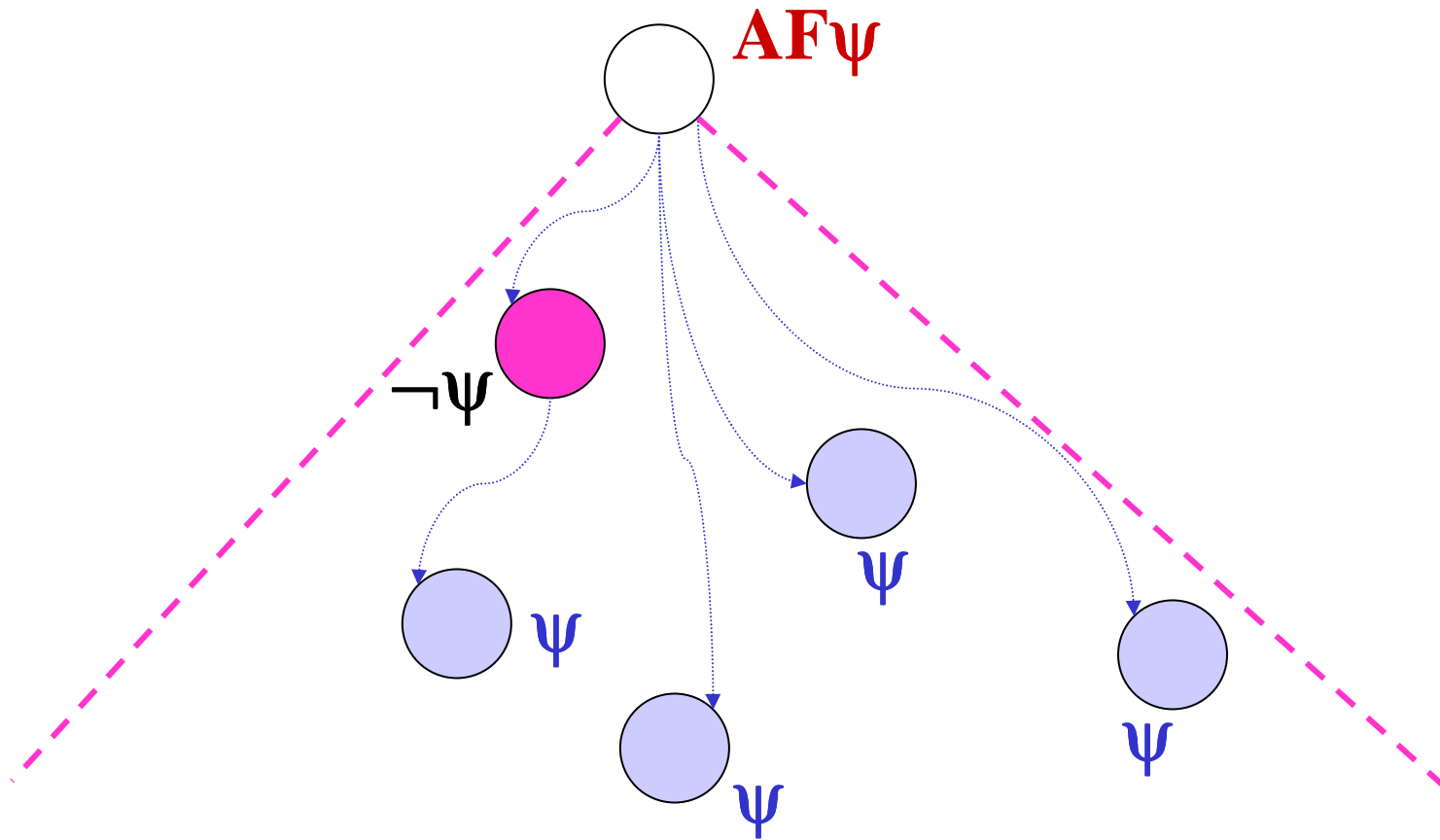
Derived Operators

- $\mathbf{K}, s \models \mathbf{AG}(\psi)$
- $\mathbf{AG}(\psi) = \neg \mathbf{EF}(\neg \psi)$
 - It is *not* the case *there exists a path* π (from s) and $\mathbf{k} \geq \mathbf{0}$ such that:
 - $\mathbf{K}, \pi(\mathbf{k}) \models \neg \psi$
 - *For every path* π (from s) and *every* $\mathbf{k} \geq \mathbf{0}$:
 - $\mathbf{K}, \pi(\mathbf{k}) \models \psi$



Derived Operators

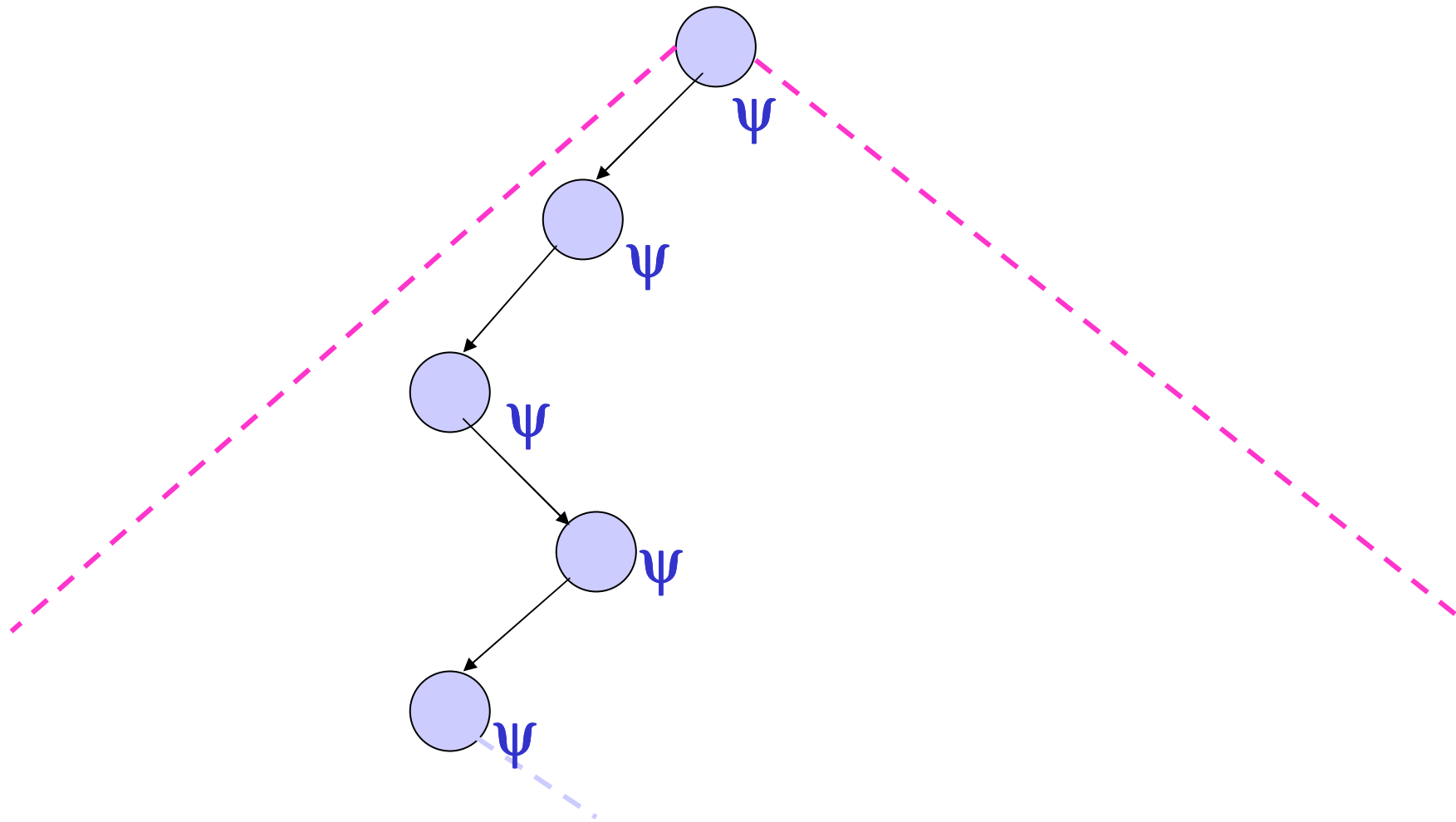
- $\mathbf{K}, \mathbf{s} \models \mathbf{AF}(\psi)$
- $\mathbf{AF}(\psi) = \mathbf{AU}(\top, \psi)$
 - *For every path π from \mathbf{s} , there exists $\mathbf{k} \geq \mathbf{0}$ such that:*
 - $\mathbf{K}, \pi(\mathbf{k}) \models \psi.$



Derived Operators

- $\mathbf{K}, \mathbf{s} \models \mathbf{EG}(\psi)$
- $\mathbf{EG}(\psi) = \neg \mathbf{AF}(\neg \psi)$
 - It is **not** the case that *for every path* π from \mathbf{s} there is a $\mathbf{k} \geq \mathbf{0}$ such that $\mathbf{K}, \pi(\mathbf{k}) \models \neg \psi$.
 - *There exists a path* π from \mathbf{s} such that, for every $\mathbf{k} \geq \mathbf{0}$:
 - $\mathbf{K}, \pi(\mathbf{k}) \models \psi$.

EG ψ



A more convenient CTL

- **NCTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\quad \quad \quad \mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{EG}(\psi)$
- **CTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\quad \quad \quad \mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{AU}(\psi_1, \psi_2)$
- **NCTL** is more convenient for model checking!
- Clearly **NCTL** can be defined in terms of **CTL**.

A more convenient CTL

- **NCTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{EG}(\psi)$
- **CTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{AU}(\psi_1, \psi_2)$
- **CTL** can be defined in terms of **NCTL**!
- The semantics of **NCTL** is given in the obvious way.

A more convenient CTL

- **NCTL** ::= **p** | $\neg\psi$ | $\psi_1 \vee \psi_2$ | **EX**(ψ) |
| **EU**(ψ_1, ψ_2) | **EG**(ψ)
- **K**, **s** \models **EG**(ψ) iff *there exists a path* π
from **s** such that for every **k** \geq **0**:
 - **K**, $\pi(\mathbf{k}) \models \psi$

A more convenient CTL

- **NCTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\quad \quad \quad \mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{EG}(\psi)$
- **CTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\quad \quad \quad \mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{AU}(\psi_1, \psi_2)$
- $\mathbf{AU}(\psi_1, \psi_2) = \neg\mathbf{EU}(\neg\psi_2, (\neg\psi_1 \wedge \neg\psi_2)) \wedge \neg\mathbf{EG}(\neg\psi_2)$

i.e., along any path: ψ_2 must hold eventually and $(\neg\psi_1 \wedge \neg\psi_2)$ can only happen after ψ_2 (recall the *before operator* of LTL)

A more convenient CTL

$$\mathbf{AU}(\psi_1, \psi_2) = \neg \mathbf{EU}(\neg \psi_2, (\neg \psi_1 \wedge \neg \psi_2)) \wedge \neg \mathbf{EG}(\neg \psi_2)$$

ψ_1 cannot become false, while ψ_2 stays false!

ψ_2 does not stay false forever! (i.e. ψ_2 will eventually become true).

A more convenient CTL

$$\mathbf{AU}(\psi_1, \psi_2) = \neg \mathbf{EU}(\neg \psi_2, (\neg \psi_1 \wedge \neg \psi_2)) \wedge \neg \mathbf{EG}(\neg \psi_2)$$

\Rightarrow Assume $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$

– Let π be a path from s . Then there exists $k \geq 0$ with:

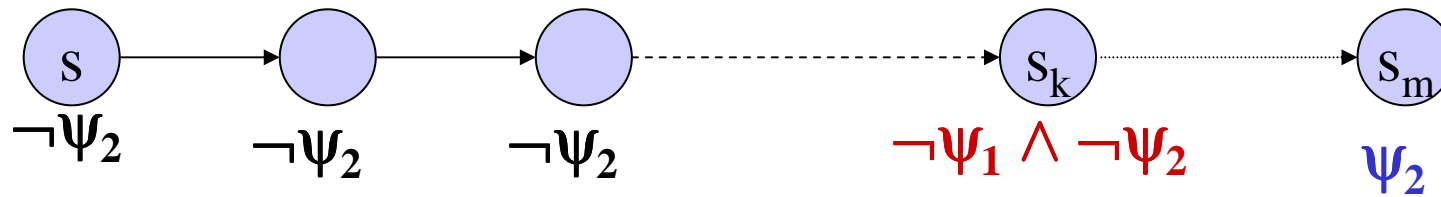
$$\blacksquare \mathbf{K}, s \models \psi_2$$

– Hence, **not** $\mathbf{K}, s \models \mathbf{EG}(\neg \psi_2)$

– Equivalent to $\mathbf{K}, s \models \neg \mathbf{EG}(\neg \psi_2)$

A more convenient CTL

- $\mathbf{AU}(\psi_1, \psi_2) = \mathbf{NewAU}(\psi_1, \psi_2) =$
 $\neg\mathbf{EU}(\neg\psi_2, (\neg\psi_1 \wedge \neg\psi_2)) \wedge \neg\mathbf{EG}(\neg\psi_2)$
- Clearly $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$ implies $\mathbf{K}, s \models \neg\mathbf{EG}(\neg\psi_2)$
- Let $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$
 - Suppose now $\mathbf{K}, s \models \mathbf{EU}(\neg\psi_2, \neg\psi_1 \wedge \neg\psi_2)$
 - Let π be any path from s witnessing the above:
 - Let now k be the least integer such that:
 - $\mathbf{K}, \pi(k) \models \neg\psi_1 \wedge \neg\psi_2$
 - $\mathbf{K}, \pi(j) \models \neg\psi_2$ for $0 \leq j < k$.



- Suppose $\mathbf{K}, \pi(\mathbf{m}) \models \psi_2$, required by $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$
- Take \mathbf{m} to be the least such number.
- Then $\mathbf{m} > \mathbf{k}$, since $\mathbf{K}, s \models \mathbf{EU}(\neg\psi_2, \neg\psi_1 \wedge \neg\psi_2)$
- But $0 \leq \mathbf{k} < \mathbf{m}$ and $\mathbf{K}, \pi(\mathbf{k}) \models \neg\psi_1$
- Hence **not** $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$. *Contradiction!*
- Thus $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$ also implies:
 - $\mathbf{K}, s \models \neg\mathbf{EU}(\neg\psi_2, \neg\psi_1 \wedge \neg\psi_2)$
- So $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$ implies $\mathbf{K}, s \models \mathbf{NewAU}(\psi_1, \psi_2)$

From CTL to NCTL

- In a similar way we can argue that:

if $\mathbf{K}, s \models \text{newAU}(\psi_1, \psi_2)$
then $\mathbf{K}, s \models \text{AU}(\psi_1, \psi_2)$.

- Hence *CTL* can be expressed in terms of *NCTL*.

A more convenient CTL

- **NCTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{EG}(\psi)$
 - **CTL** ::= $p \mid \neg\psi \mid \psi_1 \vee \psi_2 \mid \mathbf{EX}(\psi) \mid$
 $\mid \mathbf{EU}(\psi_1, \psi_2) \mid \mathbf{AU}(\psi_1, \psi_2)$
 - $\mathbf{AU}(\psi_1, \psi_2) = \mathbf{NewAU}(\psi_1, \psi_2) =$
 $\frac{\neg(\mathbf{EU}(\neg\psi_2, (\neg\psi_1 \wedge \neg\psi_2)))}{\text{red underline}} \quad \wedge \quad \frac{\mathbf{AF}(\psi_2)}{\text{red underline}}$
 - $\mathbf{NewAU}_1 = \neg \mathbf{EU}(\neg\psi_2, (\neg\psi_1 \wedge \neg\psi_2))$
 - $\mathbf{NewAU}_2 = \mathbf{AF}\psi_2$
- $\neg\mathbf{EG}\neg\psi_2 = \mathbf{AF}\psi_2$

From CTL to NCTL

- Let $\mathbf{K} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, \mathbf{AP}, \mathbf{L})$ and $s \in \mathbf{S}$.
- We need to argue:
 - $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$ iff
 $\mathbf{K}, s \models \mathbf{NewAU}_1 \wedge \mathbf{NewAU}_2$
- We already argued that:
 - If $\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$ then
 $\mathbf{K}, s \models \mathbf{NewAU}_1 \wedge \mathbf{NewAU}_2$

From CTL to NCTL

$$\mathbf{AU}(\psi_1, \psi_2) = \neg\mathbf{EU}(\neg\psi_2, (\neg\psi_1 \wedge \neg\psi_2)) \wedge \neg\mathbf{EG}(\neg\psi_2)$$

← We need to argue that:

– If $\mathbf{K}, s \models \mathbf{NewAU}_1 \wedge \mathbf{NewAU}_2$ then

$$\mathbf{K}, s \models \mathbf{AU}(\psi_1, \psi_2)$$

- So assume $\mathbf{K}, s \models \mathbf{NewAU}_1 \wedge \mathbf{NewAU}_2$.
- $\mathbf{NewAU}_1 = \neg\mathbf{EU}(\neg\psi_2, (\neg\psi_1 \wedge \neg\psi_2))$.
- $\mathbf{NewAU}_2 = \neg\mathbf{EG}\neg\psi_2 = \mathbf{AF}\psi_2$

From CTL to NCTL

- Let π be some path from s .
- We need to show that there exists $k \geq 0$ such that:
 - $\mathbf{K}, \pi(k) \models \psi_2$
 - $\mathbf{K}, \pi(j) \models \psi_1$ if $0 \leq j < k$.
- But $\mathbf{K}, s \models \mathbf{AF} \psi_2$ implies that along any path (and also along π) there exists $k \geq 0$ such that:
 - $\mathbf{K}, \pi(k) \models \psi_2$
- Assume k is the *least* such number along π .

From CTL to NCTL

Now consider an arbitrary \mathbf{m} with $\mathbf{0} \leq \mathbf{m} < \mathbf{k}$.

CLAIM: $\mathbf{K}, \sigma(\mathbf{m}) \models \psi_1$

- If the **CLAIM** is true then we are done.
- Suppose instead that $\mathbf{K}, \sigma(\mathbf{m}) \models \neg\psi_1$.
 - Then $\mathbf{K}, \sigma(\mathbf{m}) \models \neg\psi_1 \wedge \neg\psi_2$ ($\mathbf{m} < \mathbf{k}$) *WHY???*
 - and $\mathbf{K}, \sigma(\mathbf{j}) \models \neg\psi_2$ if $\mathbf{0} \leq \mathbf{j} < \mathbf{m}$, since $\mathbf{j} < \mathbf{m} < \mathbf{k}$
 - Hence $\mathbf{K}, \sigma(\mathbf{0}) \models \mathbf{EU}(\neg\psi_2, \neg\psi_1 \wedge \neg\psi_2)$
 - Therefore, **not** $\mathbf{K}, \mathbf{s} \models \mathbf{NewAU}_1$ which is a *contradiction!*

CTL Model Checking

- $\mathbf{K} \models \psi$ *iff*
 $\mathbf{K}, s_0 \models \psi$ for every $s_0 \in S_0$.
- The *CTL model checking problem*.
 - $\mathbf{K} = (\mathbf{S}, S_0, \mathbf{R}, \mathbf{AP}, \mathbf{L})$ (system model)
 - ψ a *CTL* formula (spec. of the property)
- Given \mathbf{K} and ψ *determine whether or not* $\mathbf{K} \models \psi$

CTL Model Checking

- The actual model checking problem:
 - Given $\mathbf{K} = (\mathbf{S}, \mathbf{S}_0, \mathbf{R}, \mathbf{AP}, \mathbf{L})$
 - Given $\mathbf{s} \in \mathbf{S}$
 - Given ψ , **an NCTL formula.**
 - Determine whether:

$$\mathbf{K}, \mathbf{s} \models \psi$$

The Sub-formulas of ψ

- **SF(ψ)** is the *least set of formulas* satisfying:
 - $\psi \in \mathbf{SF}(\psi)$
 - If $\neg\alpha \in \mathbf{SF}(\psi)$ then $\alpha \in \mathbf{SF}(\psi)$.
 - If $\alpha \vee \beta \in \mathbf{SF}(\psi)$ then $\alpha, \beta \in \mathbf{SF}(\psi)$
 - If $\mathbf{EX}\alpha \in \mathbf{SF}(\psi)$ then $\alpha \in \mathbf{SF}(\psi)$.
 - If $\mathbf{EU}(\alpha, \beta) \in \mathbf{SF}(\psi)$ then $\alpha, \beta \in \mathbf{SF}(\psi)$
 - If $\mathbf{EG}\alpha \in \mathbf{SF}(\psi)$ then $\alpha \in \mathbf{SF}(\psi)$.
- **SF(ψ)** ---- The *set of sub-formulas* of ψ .

The Labeling Procedure.

- $\mathbf{K} = (S, S_0, R, AP, L)$
 - $s \in S$
 - ψ a *NCTL* formula (built out of **AP**).

- **Strategy:**

- Construct **Labels: $S \longrightarrow 2^{SF(\psi)}$**
- $2^{SF(\psi)}$, the set of subsets of **$SF(\psi)$** .
- Each state of **K** is assigned a subset of a **$SF(\psi)$** by the **Labels** function.

- **$K, s \models \psi$ iff $\psi \in \mathbf{Labels}(s)$.**

The Labels function

- **Stage 1:**
 - For every $t \in S$:
 - **Labels**(t) = **L**(t) ($K = (S, S_0, R, AP, L)$)

•

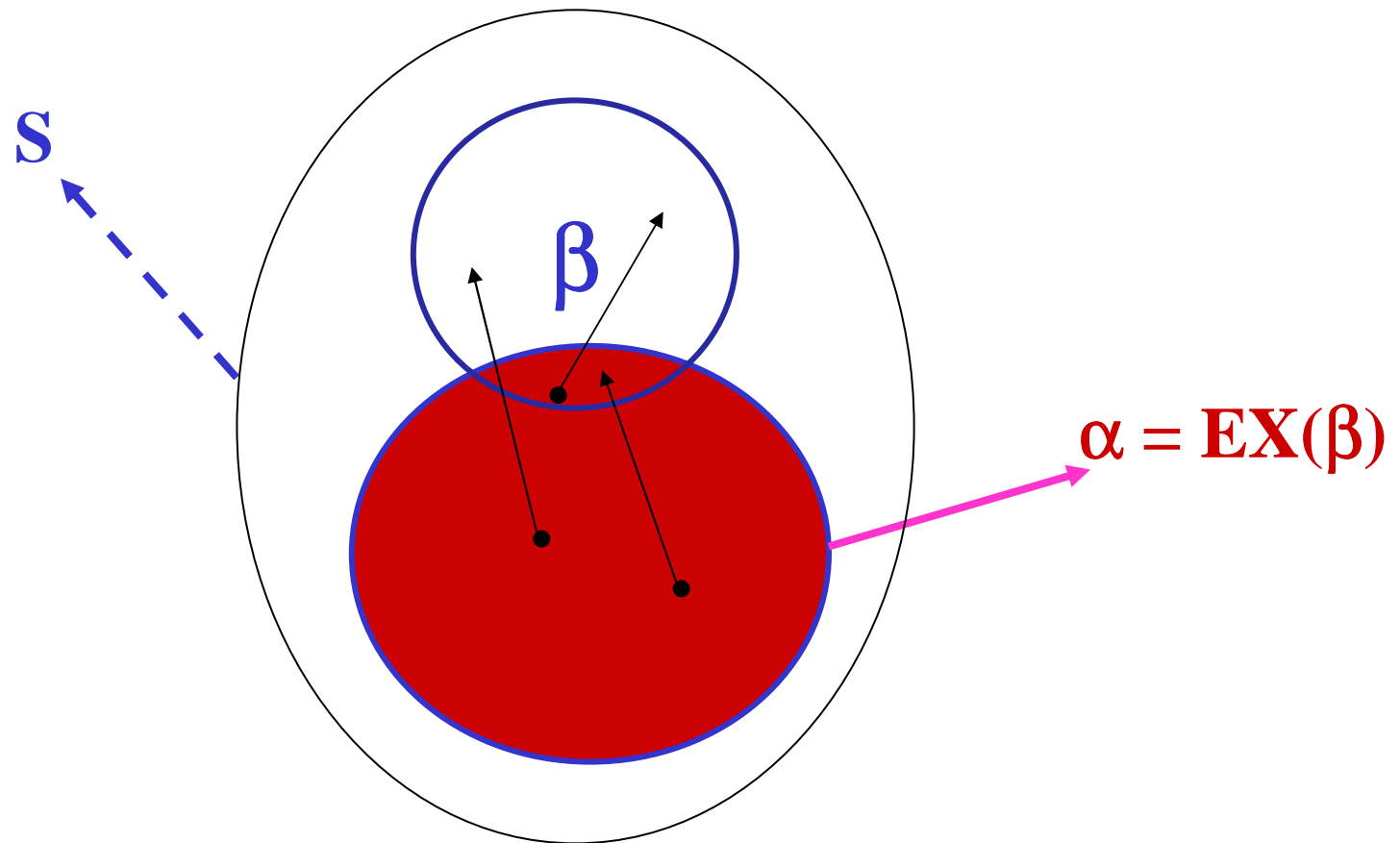
Assume we have done up to stage i .

- **Stage $i + 1$:**
 - For every $t \in S$:
 - If $\alpha = \neg\beta$ then
 $\alpha \in \text{Labels}(t)$ *iff* $\beta \notin \text{Labels}(t)$.

The Labels function

- **Stage $i + 1$:**
 - For every $t \in S$:
 - If $\alpha = \beta_1 \vee \beta_2$ then
 $\alpha \in \text{Labels}(t)$ *iff* $\beta_1 \in \text{Labels}(t)$ **or** $\beta_2 \in \text{Labels}(t)$
 - If $\alpha = \mathbf{EX}\beta$ then
 $\alpha \in \text{Labels}(t)$ *iff* there exists $s \in S$ such that
 $\beta \in \text{Labels}(s)$ and $\mathbf{R}(t, s)$ [i.e. $t \rightarrow s$]

The Labels Function



Computing the labeling for $EX(\beta)$

Complexity: $O(|M|)$

Algorithm Check_EX(β)

$\mathbf{T} := \{s \mid \beta \in \mathbf{Labels}(s)\};$

while $\mathbf{T} \neq \emptyset$ do

 choose $s \in \mathbf{T};$

$\mathbf{T} := \mathbf{T} \setminus \{s\};$

 for each $t \in \mathbf{S}$ such that $t \rightarrow s$ do

$\mathbf{Labels}(t) := \mathbf{Labels}(t) \cup \{\mathbf{EX} \beta\};$

The Labels Function

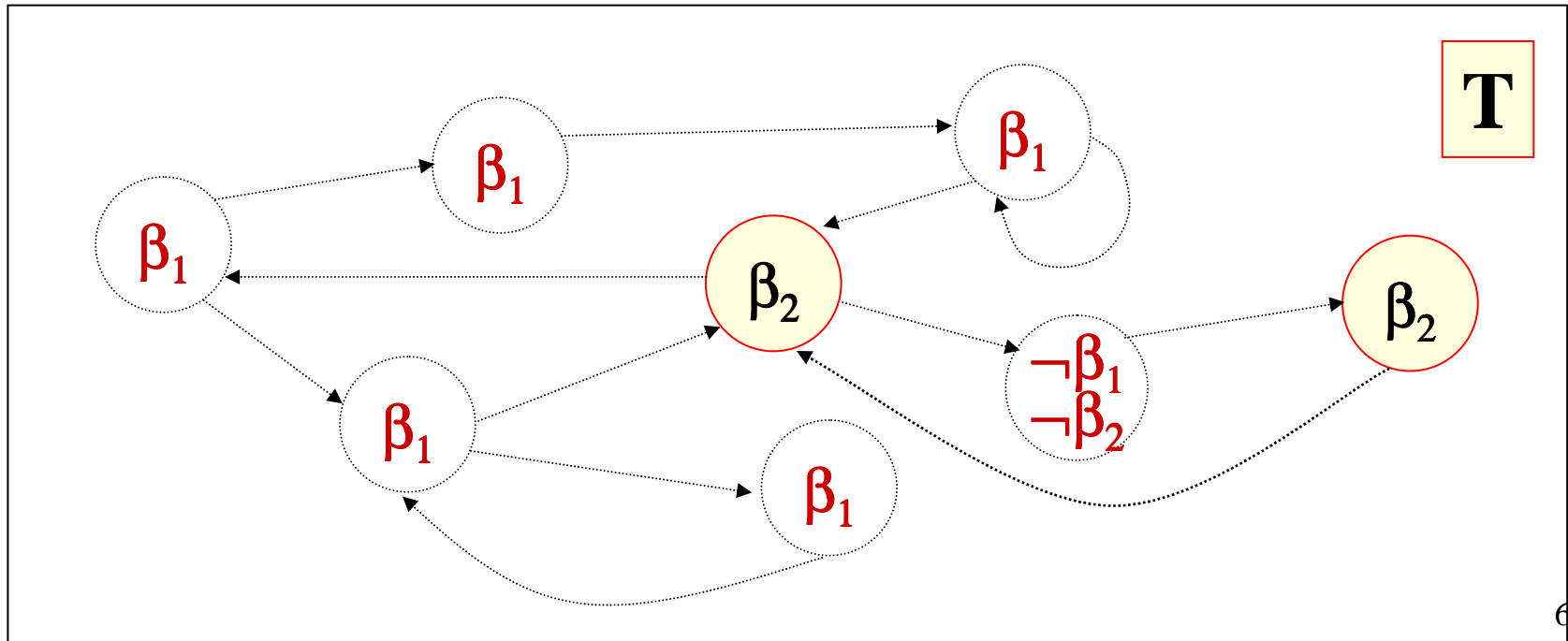
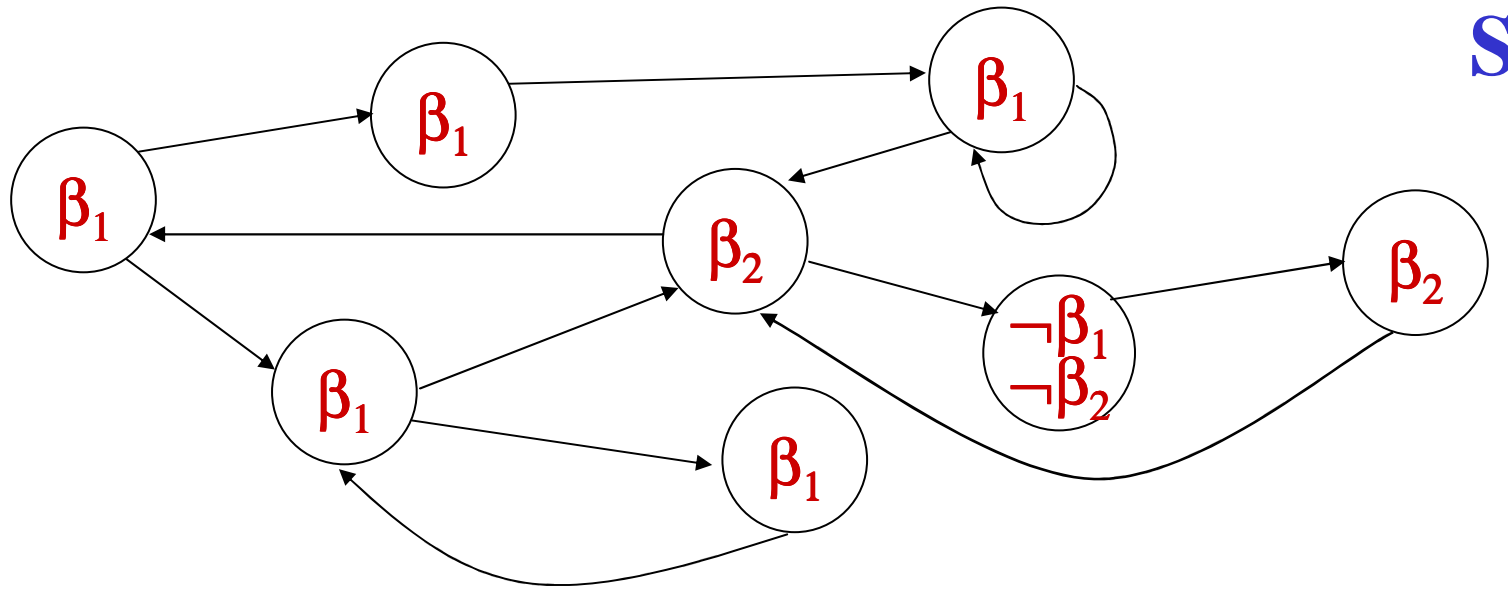
- **Stage $i + 1$:**
 - For every $t \in S$:
 - If $\alpha = \mathbf{EU}(\beta_1, \beta_2)$ then
 $\alpha \in \mathbf{Labels}(t)$ *iff*
 - $\beta_2 \in \mathbf{Labels}(t)$ or
 - $\beta_1 \in \mathbf{Labels}(t)$ and $\mathbf{EU}(\beta_1, \beta_2) \in \mathbf{Labels}(s)$
for some s with $t \rightarrow s$.

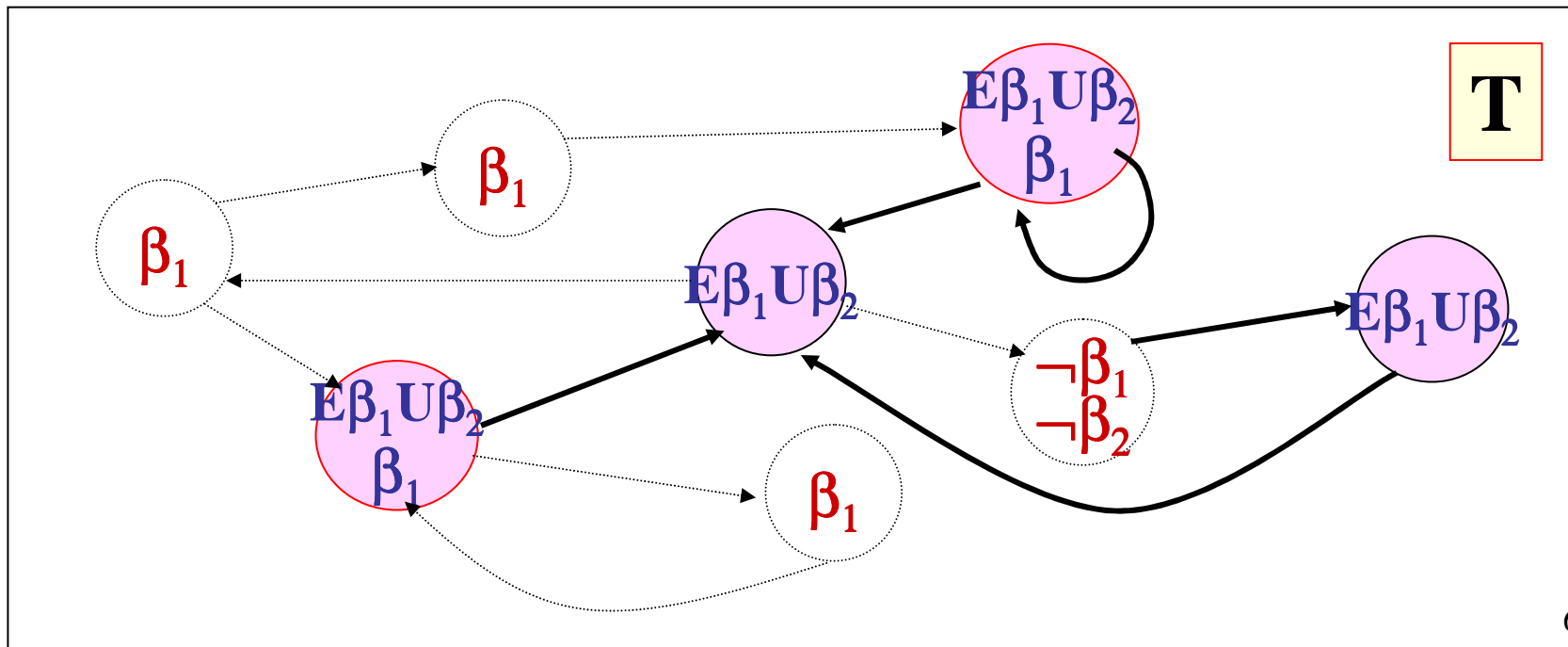
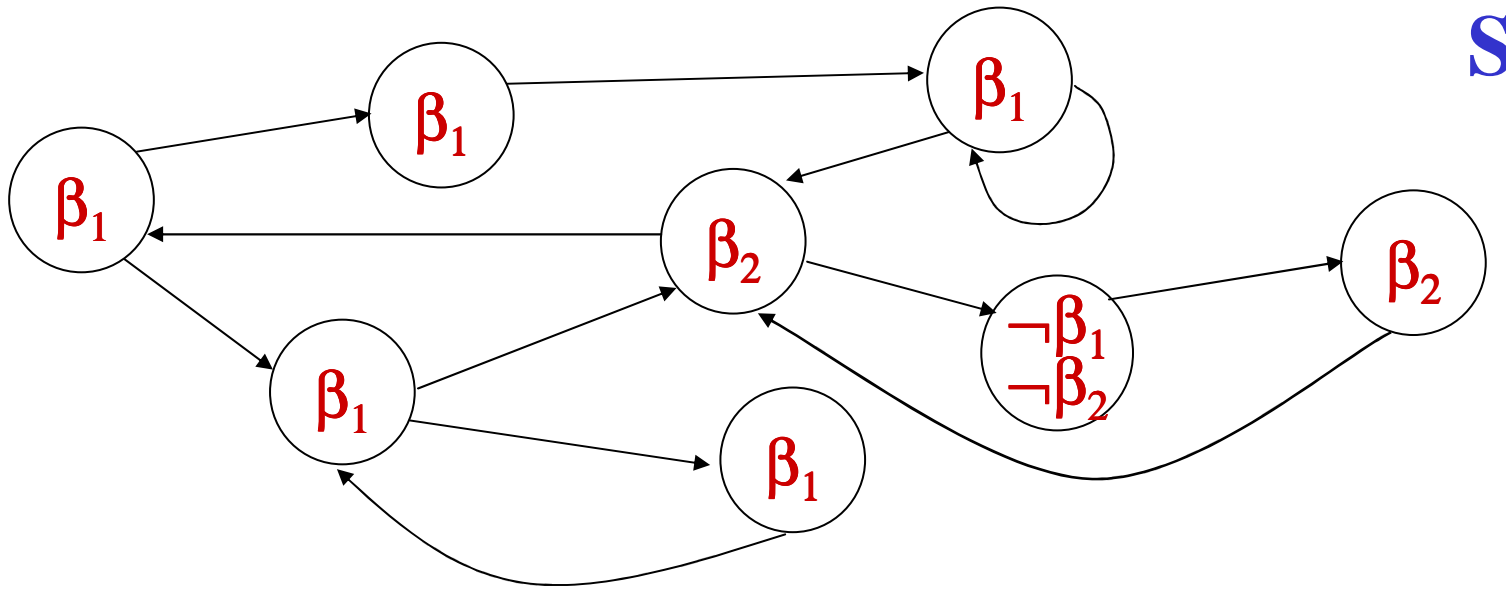
The Labels Function

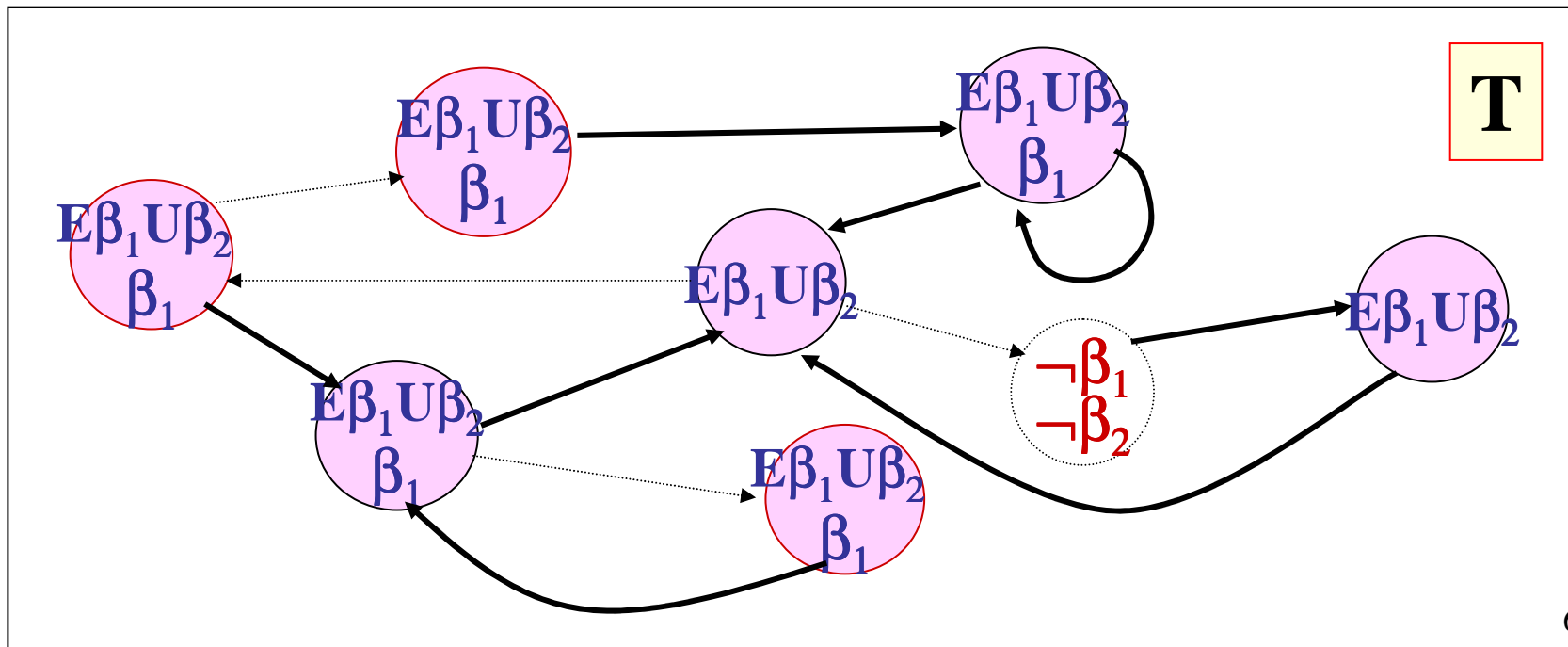
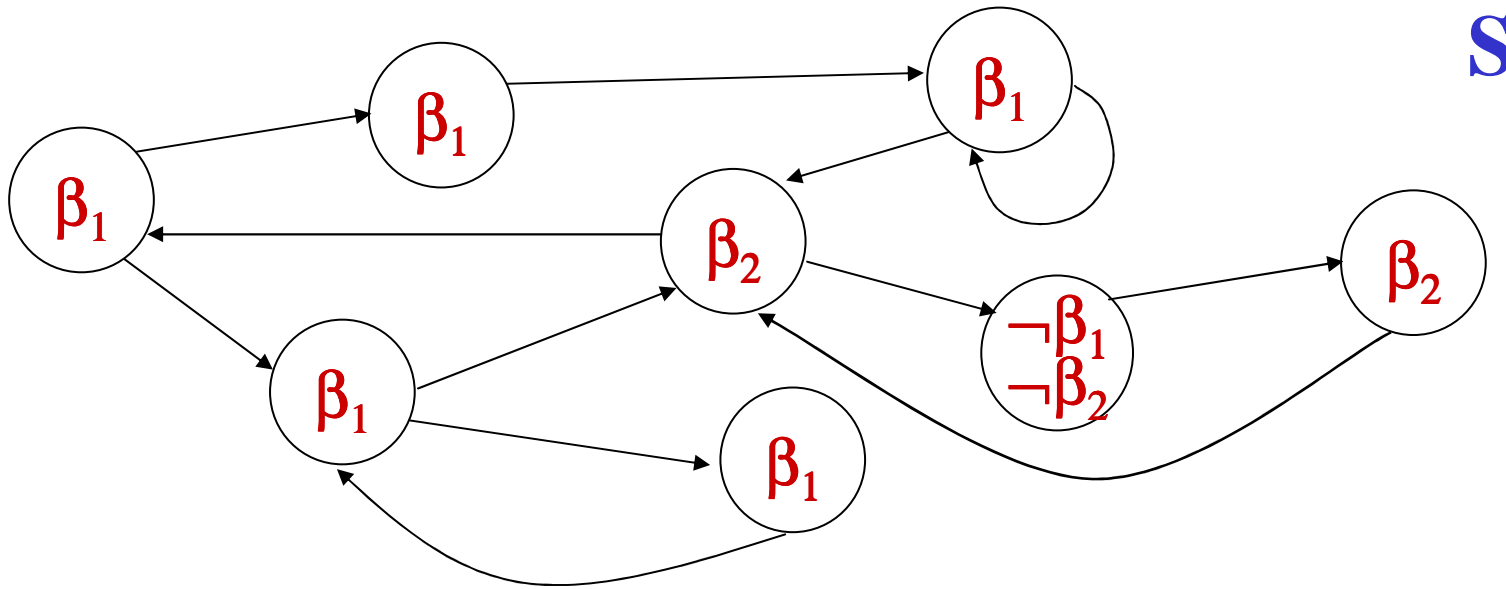
- Collect in **T** all the states satisfying β_2
 - all these states do also satisfy $\mathbf{EU}(\beta_1, \beta_2)$.
- Traverse backward \rightarrow from states in **T** and label with $\mathbf{EU}(\beta_1, \beta_2)$ all the states **t** satisfying β_1 and reaching at least a state **s** labeled with $\mathbf{EU}(\beta_1, \beta_2)$.

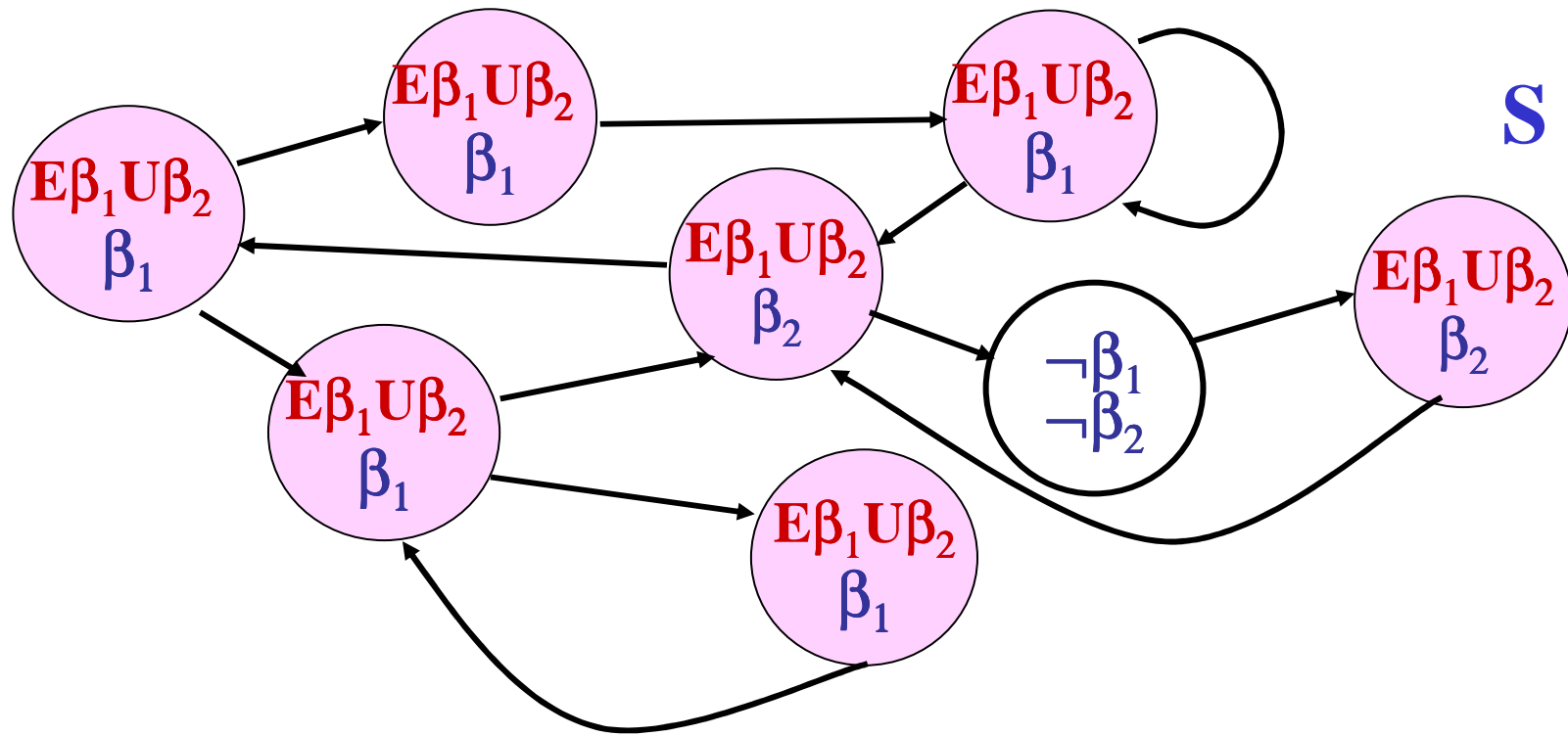
**If $s \in T$, t with $t \rightarrow s$ and $\beta_1 \in \text{Labels}(t)$
then $\mathbf{EU}(\beta_1, \beta_2) \in \text{Labels}(t)$**

Recall that: $\mathbf{EU}(\beta_1, \beta_2) = (\beta_2 \vee (\beta_1 \wedge \mathbf{EXEU}(\beta_1, \beta_2)))$









Computing the labeling for $\text{EU}(\beta_1, \beta_2)$

Algorithm Check_EU(β_1, β_2)

$\mathbf{T} := \{\mathbf{s} \mid \beta_2 \in \text{Labels}(\mathbf{s})\};$

Complexity: $O(|M|)$

for each $\mathbf{s} \in \mathbf{T}$ do

$\text{Labels}(\mathbf{s}) := \text{Labels}(\mathbf{s}) \cup \{\text{EU}(\beta_1, \beta_2)\};$

while $\mathbf{T} \neq \emptyset$ do

 choose $\mathbf{s} \in \mathbf{T};$

$\mathbf{T} := \mathbf{T} \setminus \{\mathbf{s}\};$

 for each $\mathbf{t} \in \mathbf{S}$ with $\mathbf{t} \rightarrow \mathbf{s}$ do

 if $\text{EU}(\beta_1, \beta_2) \notin \text{Labels}(\mathbf{t})$ and $\beta_1 \in \text{Labels}(\mathbf{t})$ then

$\text{Labels}(\mathbf{t}) := \text{Labels}(\mathbf{t}) \cup \{\text{EU}(\beta_1, \beta_2)\};$

$\mathbf{T} := \mathbf{T} \cup \{\mathbf{t}\};$

The Labels Function

- **Stage $i + 1$:**
 - For every $t \in S$:
 - If $\alpha = \mathbf{EG}(\beta)$ then
 $\alpha \in \mathbf{Labels}(t)$ *iff*
 - $\beta \in \mathbf{Labels}(t)$ and $\mathbf{EG}(\beta) \in \mathbf{Labels}(s)$ for
some s with $t \rightarrow s$.

Property of $EG(\beta)$

Let $\mathbf{M}' = (\mathbf{S}', \mathbf{R}', \mathbf{L}')$ be the sub-graph of \mathbf{M} where

- $\mathbf{S}' = \{ s \mid \mathbf{M}, s \models \beta \}$
- $\mathbf{R}' = \mathbf{R}|_{\mathbf{S}' \times \mathbf{S}'}$ (the restriction of \mathbf{R} to \mathbf{S}')
- $\mathbf{L}' = \mathbf{L}|_{\mathbf{S}'}$ (the restriction of \mathbf{L} to \mathbf{S}')

Lemma: $\mathbf{M}, s \models EG(\beta)$ *iff*

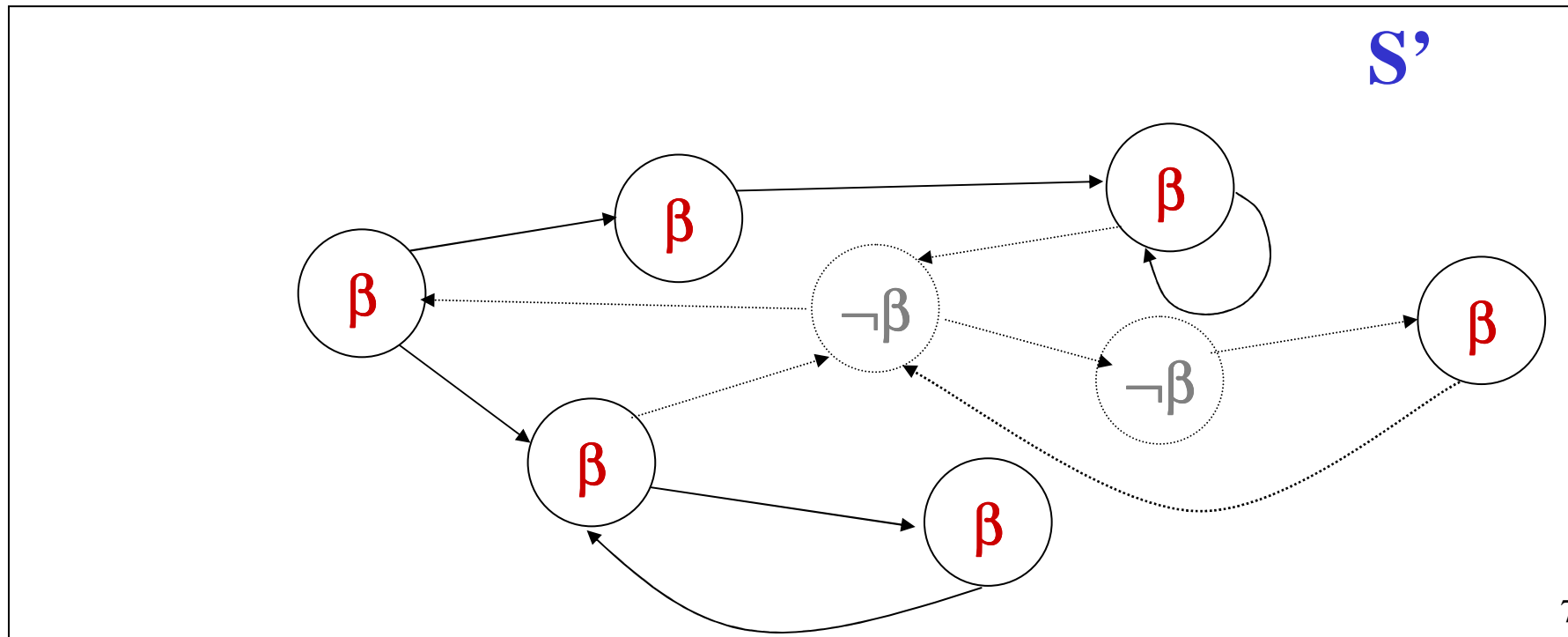
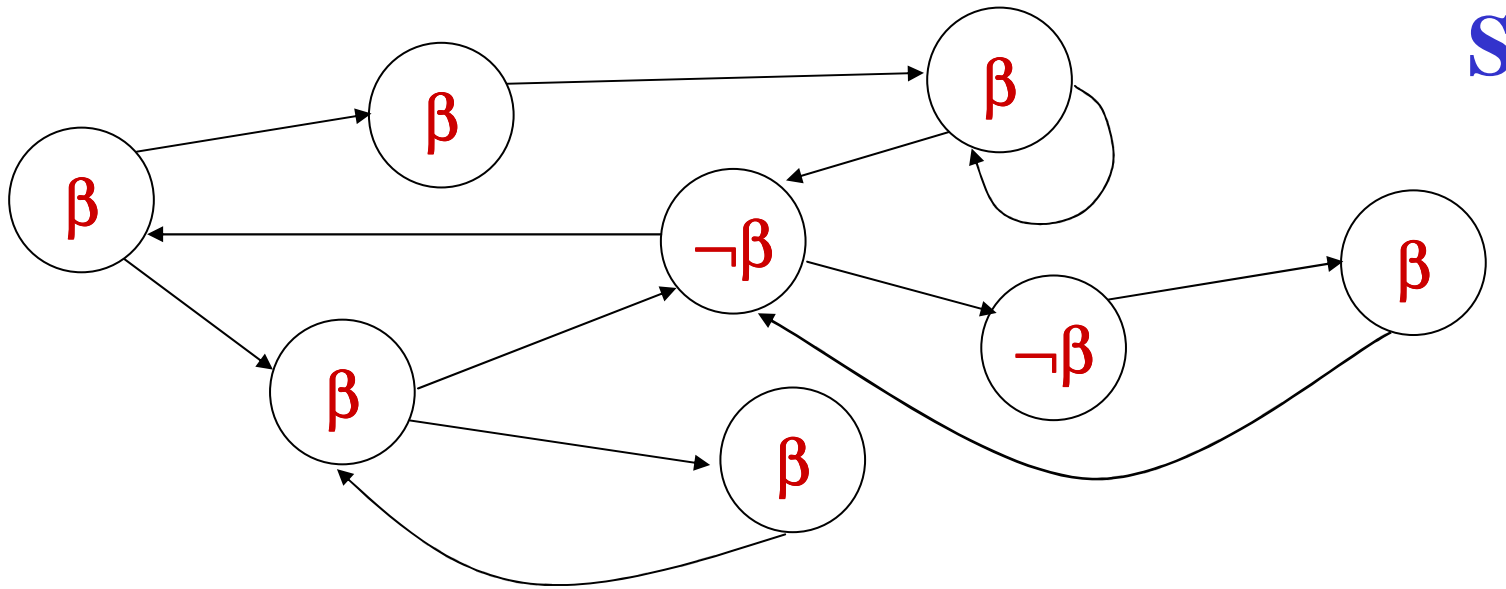
1. $s \in \mathbf{S}'$ and
2. *there exists a path* in \mathbf{M}' leading from s to a *non-trivial strongly connected component* \mathbf{C} of the graph $(\mathbf{S}', \mathbf{R}')$.

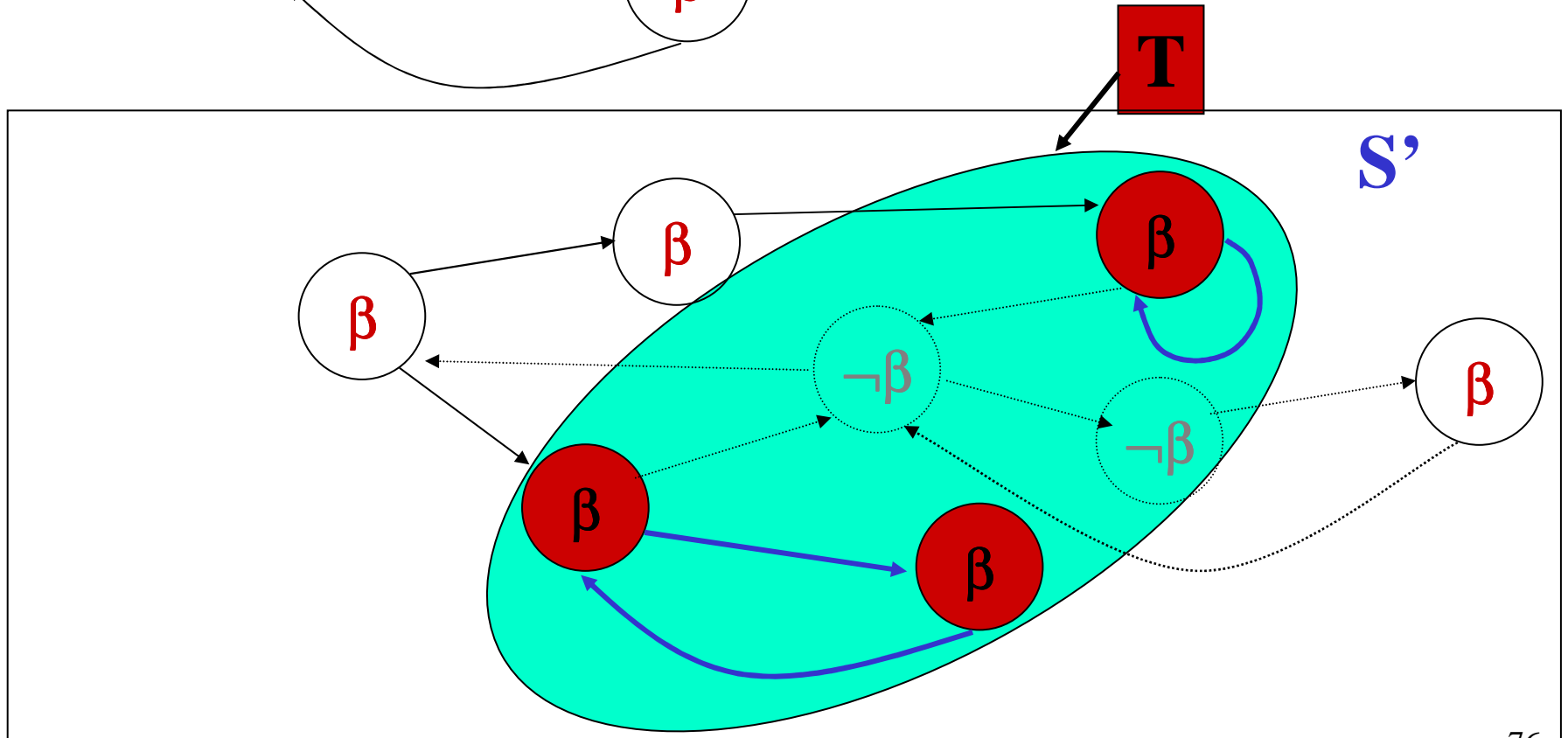
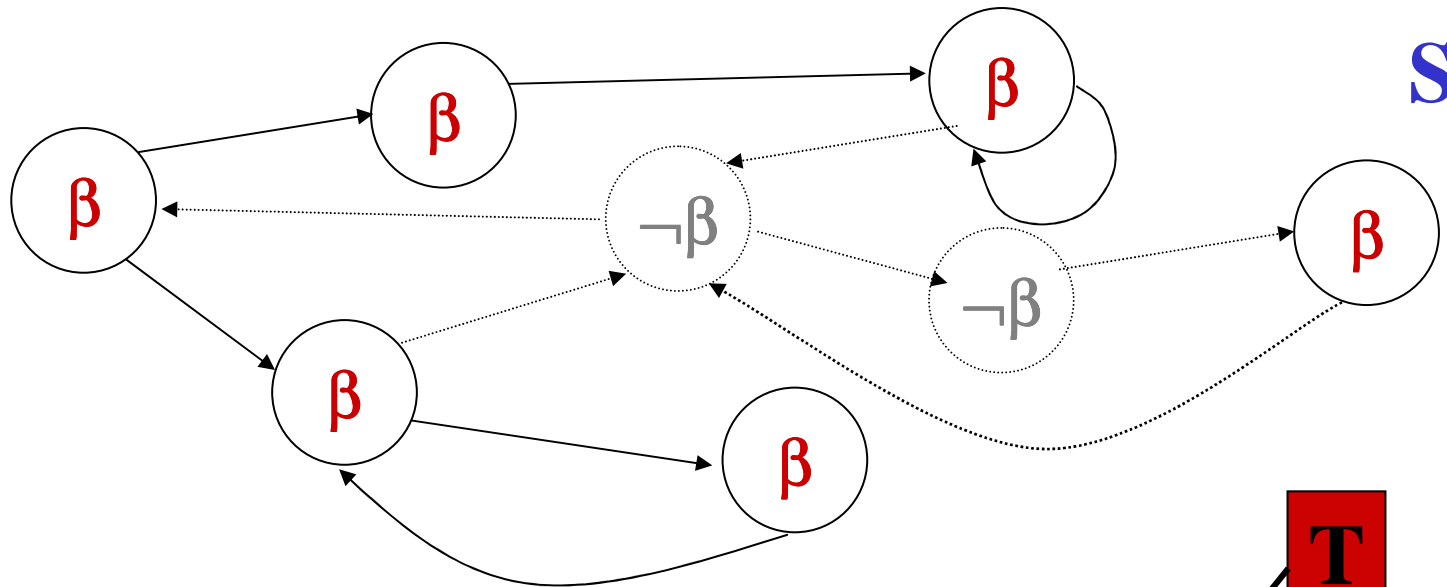
The Labels Function

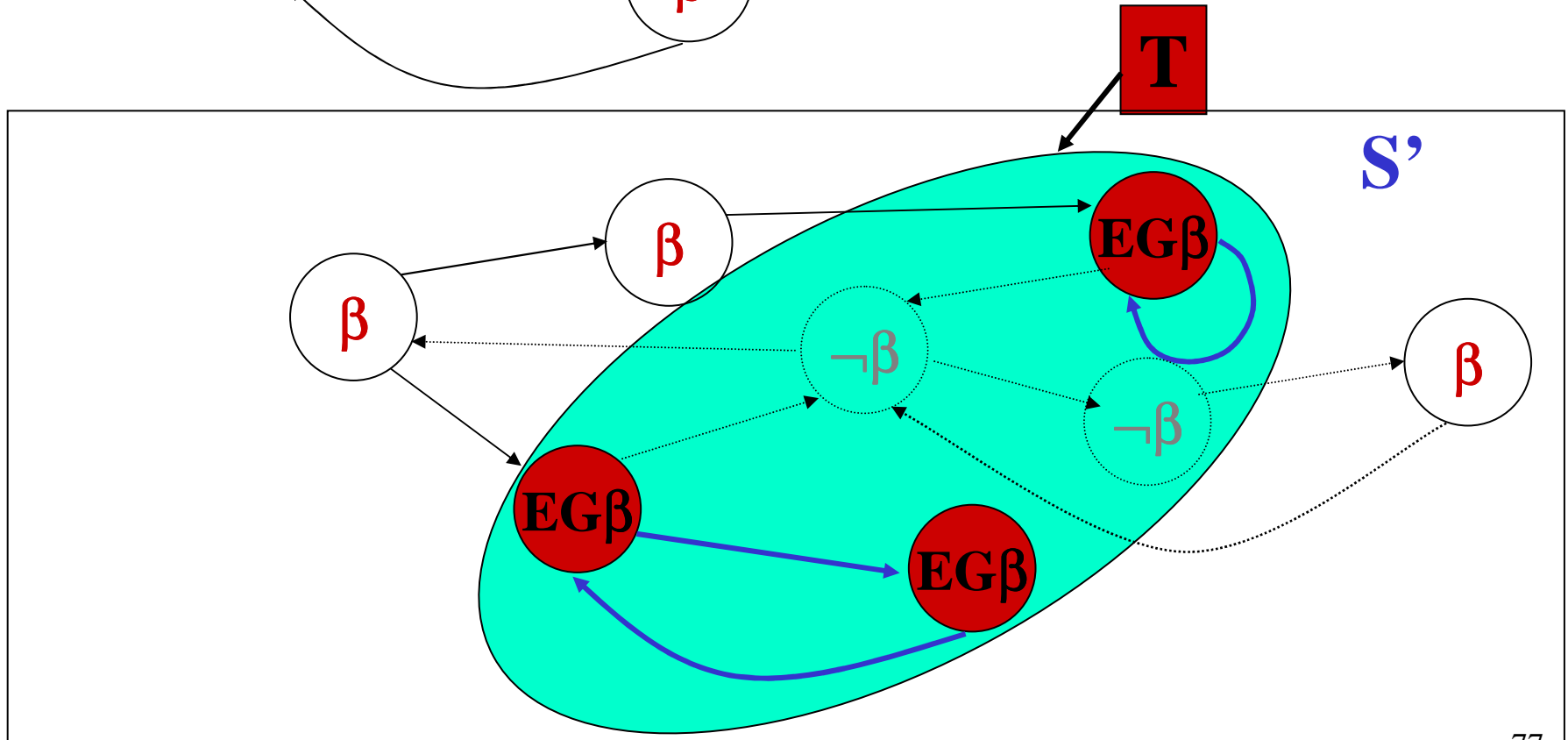
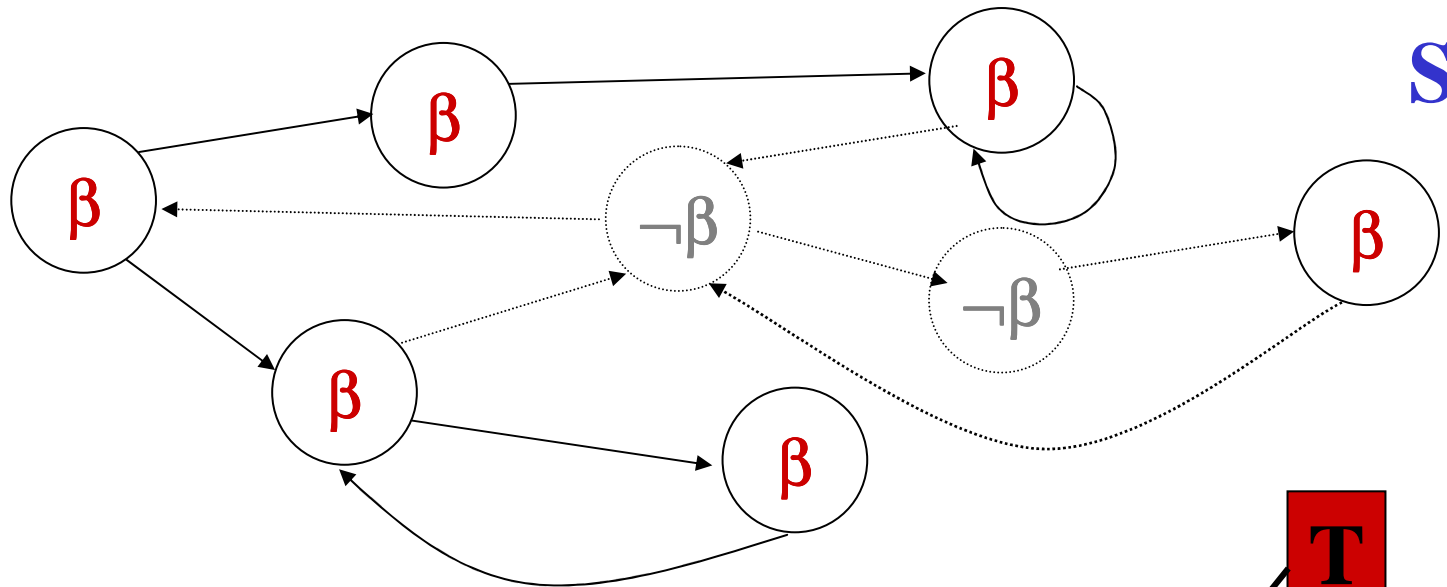
- Compute the subgraph S' whose states satisfy β
- Take *non-trivial strongly connected components* of S'
 - all the states in these components do satisfy $EG(\beta)$.
- Traverse backward \rightarrow' and label with $EG(\beta)$ the *states t reaching at least a state s* labeled with $EG(\beta)$ (note that both t and s belong to S').

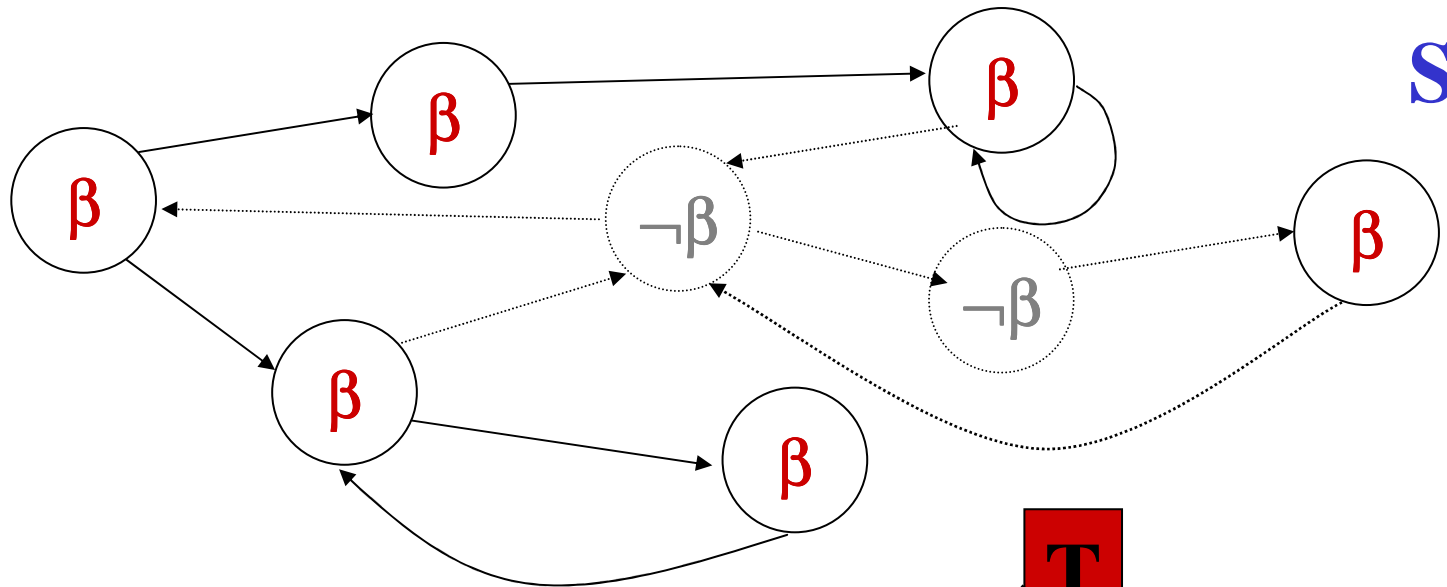
If $t \in S'$ and $R(t,s)$ then $EG(\beta) \in \text{Labels}(t)$

Recall that: $EG \beta = \beta \wedge EXEG \beta$

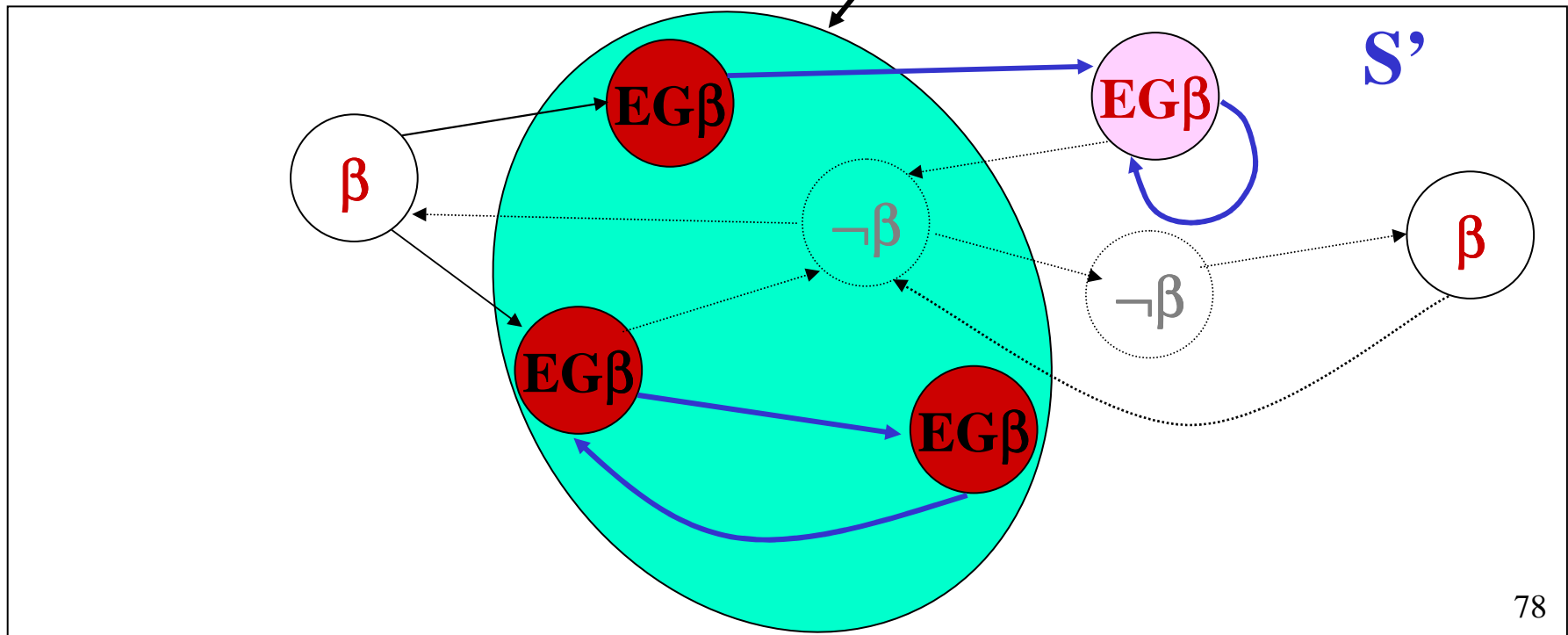


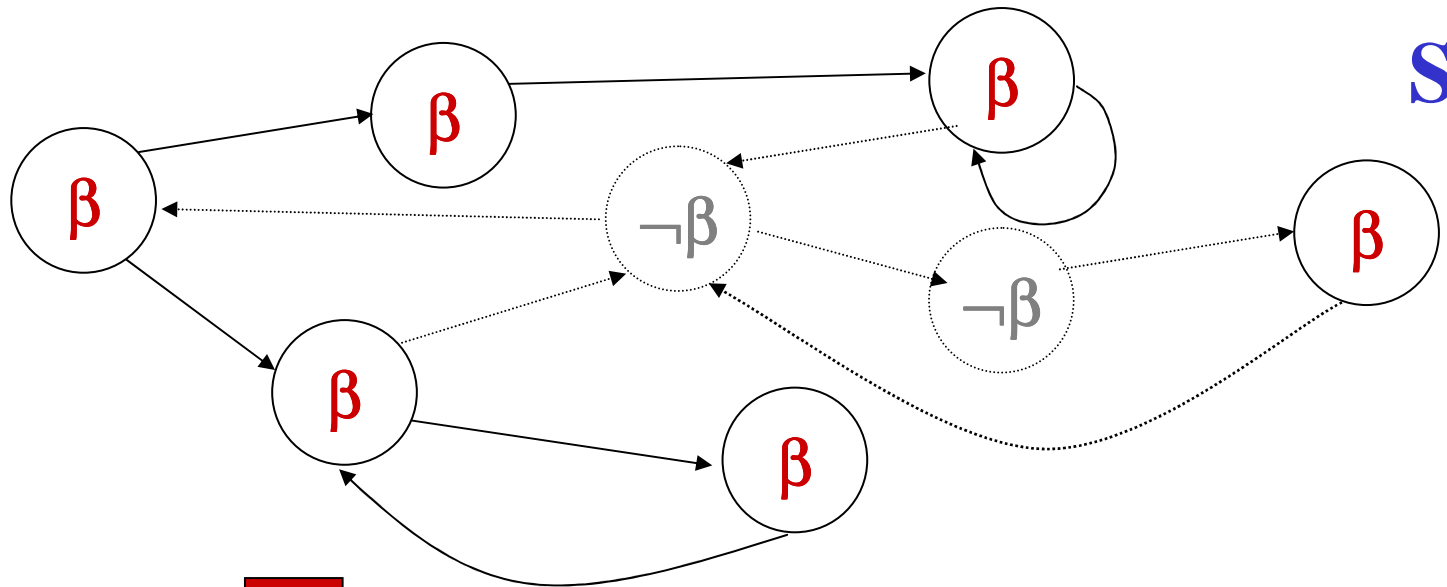




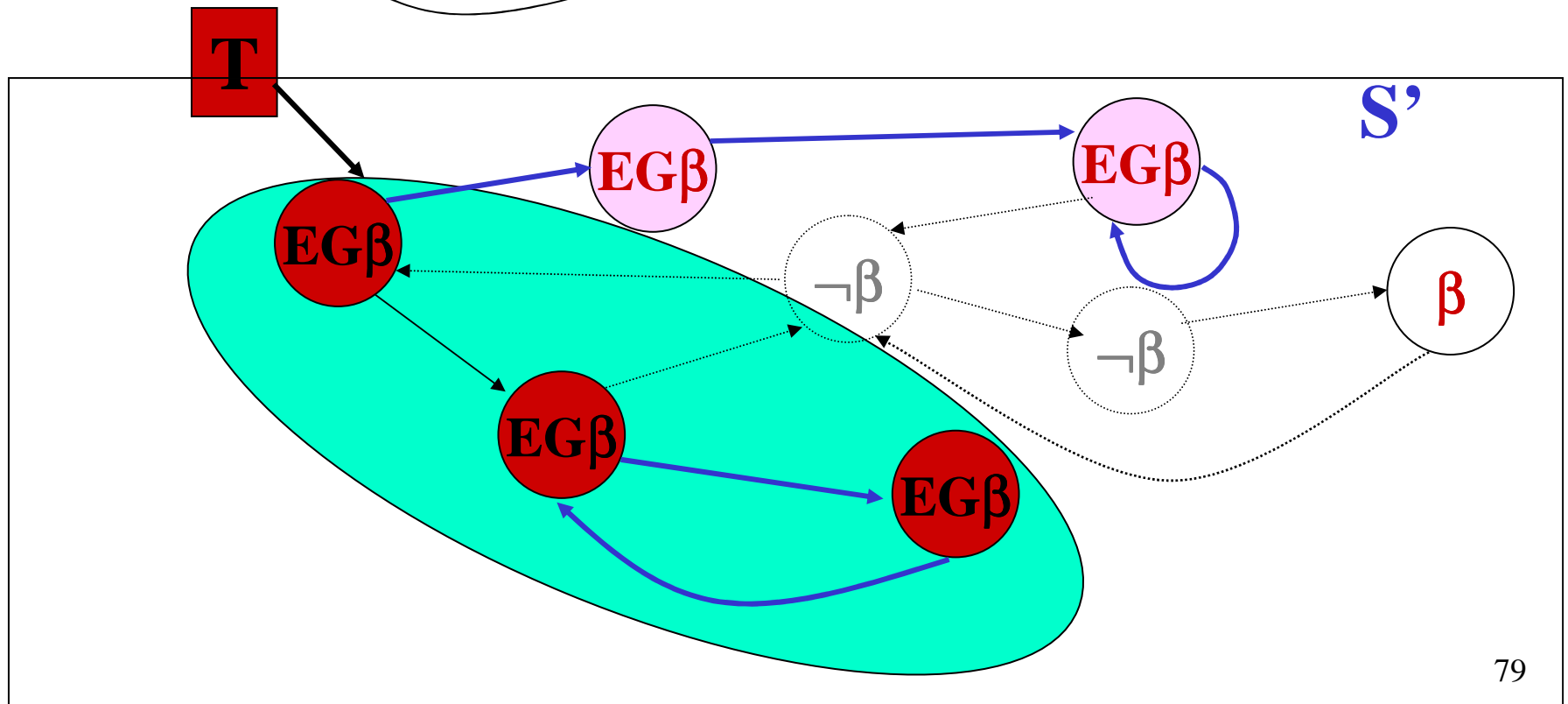


T

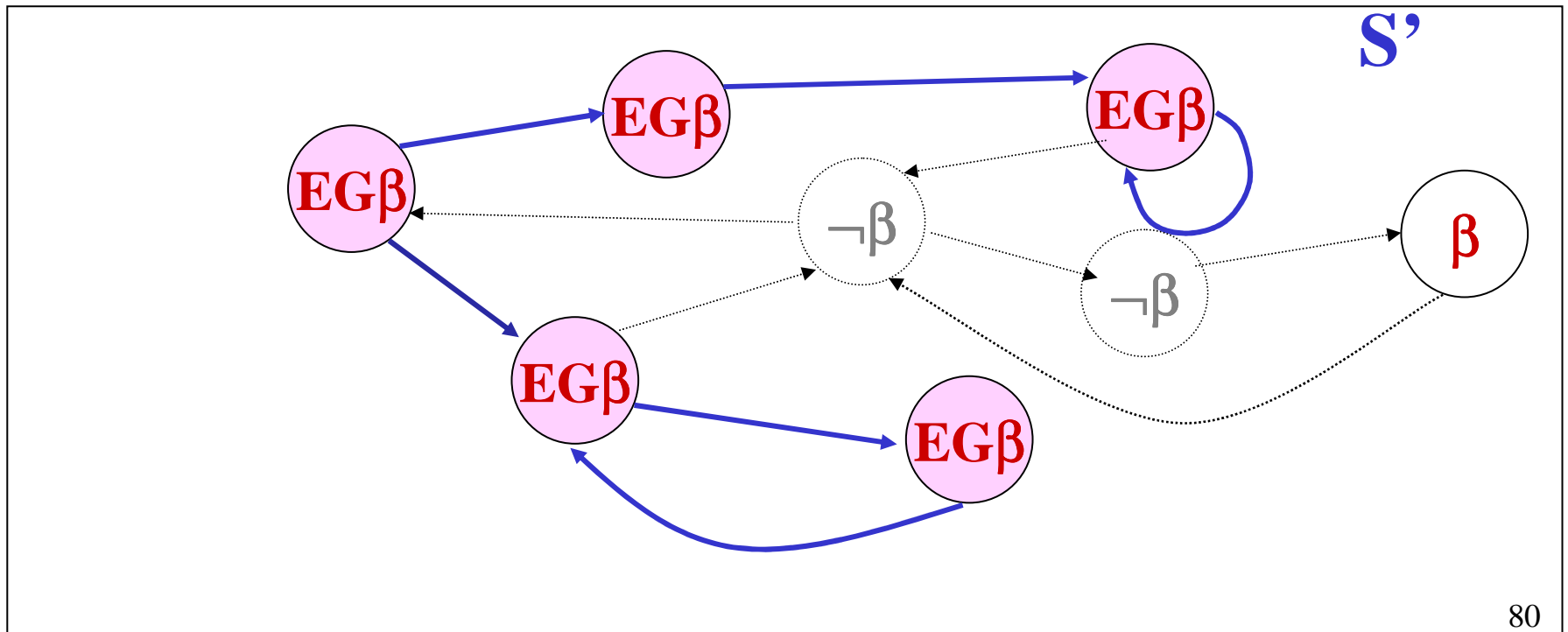
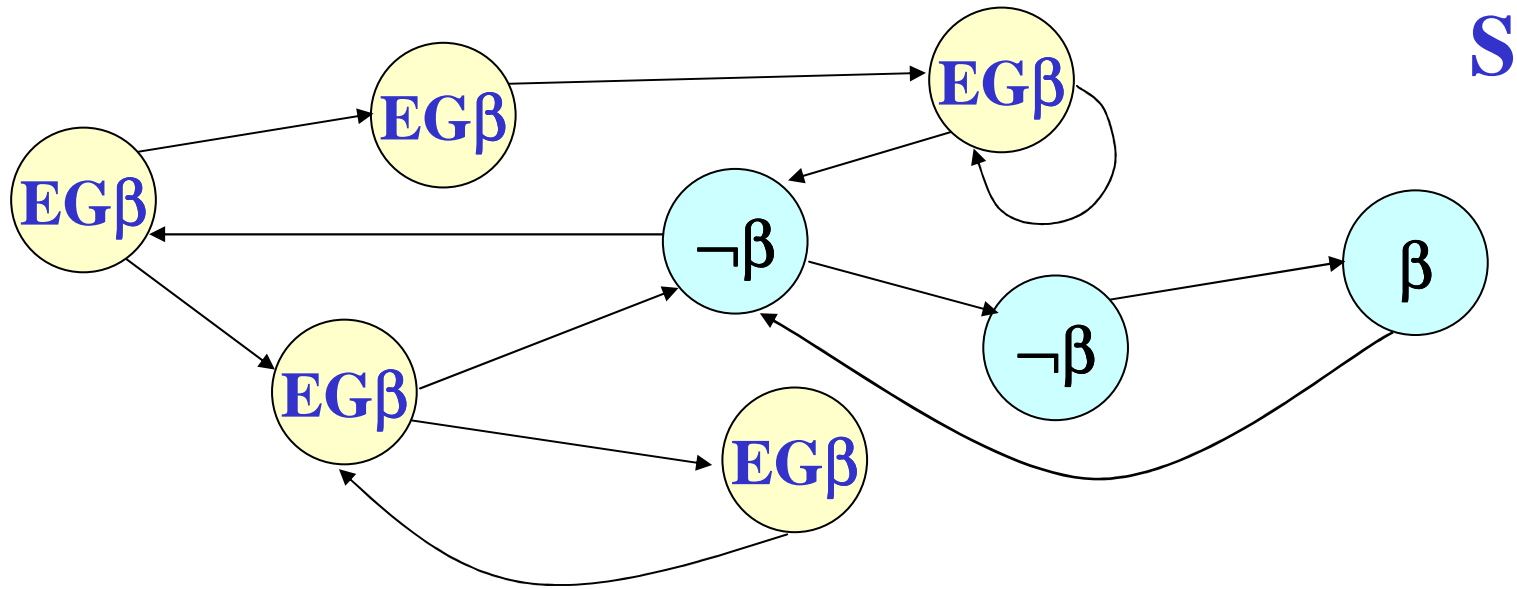




S



S'



Computing the labeling for $EG(\beta)$

Algorithm Check_EG(β)

Complexity: $O(|M|)$

$S' := \{s \mid \beta \in \text{Labels}(s)\};$

$\text{SCC} := \{C \mid C \text{ is a non trivial SCC of } S'\};$

$T := \bigcup_{C \in \text{SCC}} \{s \mid s \in C\};$

for each $s \in T$ do $\text{Labels}(s) := \text{Labels}(s) \cup \{EG(\beta)\};$

while $T \neq \emptyset$ do

 chose $s \in T;$

$T := T \setminus \{s\};$

 for each $t \in S'$ with $t \rightarrow s$ do

 if $EG(\beta) \notin \text{Labels}(t)$ then

$\text{Labels}(t) := \text{Labels}(t) \cup \{EG(\beta)\};$

$T := T \cup \{t\};$

CTL model checking

- The algorithms just presented show that the *model checking problem* for *CTL* can be solved in *time linear* in the size of System **M** and the size of the Property ϕ , namely:

in time $O(|\mathbf{M}| \cdot |\phi|)$

where $|\mathbf{M}|$ is the *size of the graph* underlying **M** and $|\phi|$ is the *number of subformulae* of ϕ .

Fixed point characterization

- We will redefine the labeling function in terms of *fixed point computation*.
- This is a *nice* and *elegant* algorithmic account.
- It will be used when *efficient symbolic approach* will be introduced.

Partial Orders

- A binary relation \subseteq on a set A is a *partial order* iff \subseteq is *reflexive*, *anti-symmetric* and *transitive*.
- The pair $\langle A, \subseteq \rangle$ is called a *partially ordered set* (or *poset*).
- **Example:** If S is any set and \subseteq is the ordinary subset relation, then $\langle 2^S, \subseteq \rangle$ is a *partially ordered set*.

Upper Bounds

Given $\langle A, \sqsubseteq \rangle$ and $A' \subseteq A$

- $a \in A$ is an *upper bound* of A' iff $\forall a' \in A', a' \sqsubseteq a$
- $a \in A$ is a *least upper bound (lub)* of A' , written $\sqcup A'$, iff
 - a is an *upper bound* of A' and
 - $\forall a' \in A$, if a' is an *upper bound* of A' , then $a \sqsubseteq a'$

Lower Bounds

Given $\langle A, \sqsubseteq \rangle$ and $A' \subseteq A$

- $a \in A$ is a *lower bound* of A' iff $\forall a' \in A', a \sqsubseteq a'$
- $a \in A$ is a *greatest lower bound (glb)* of A' , written $\sqcap A'$, iff
 - a is a *lower bound* of A' and
 - $\forall a' \in A$, if a' is a *lower bound* of A' , then $a' \sqsubseteq a$

Complete Lattice

A *poset* $\langle A, \sqsubseteq \rangle$ is a *complete lattice* if, for each $A' \subseteq A$, the *greatest lower bound* $\sqcap A'$ and the *least upper bound* $\sqcup A'$ do exist.

A *complete lattice* $\langle A, \sqsubseteq \rangle$ has a unique *greatest element* $\sqcup A = \top$ and also a unique *least element* $\sqcap A = \perp$.

Complete Lattice

The *poset* $\langle 2^S, \subseteq \rangle$ is a *complete lattice* where *intersection* \cap and *union* \cup correspond to \prod and \sqcup , respectively.

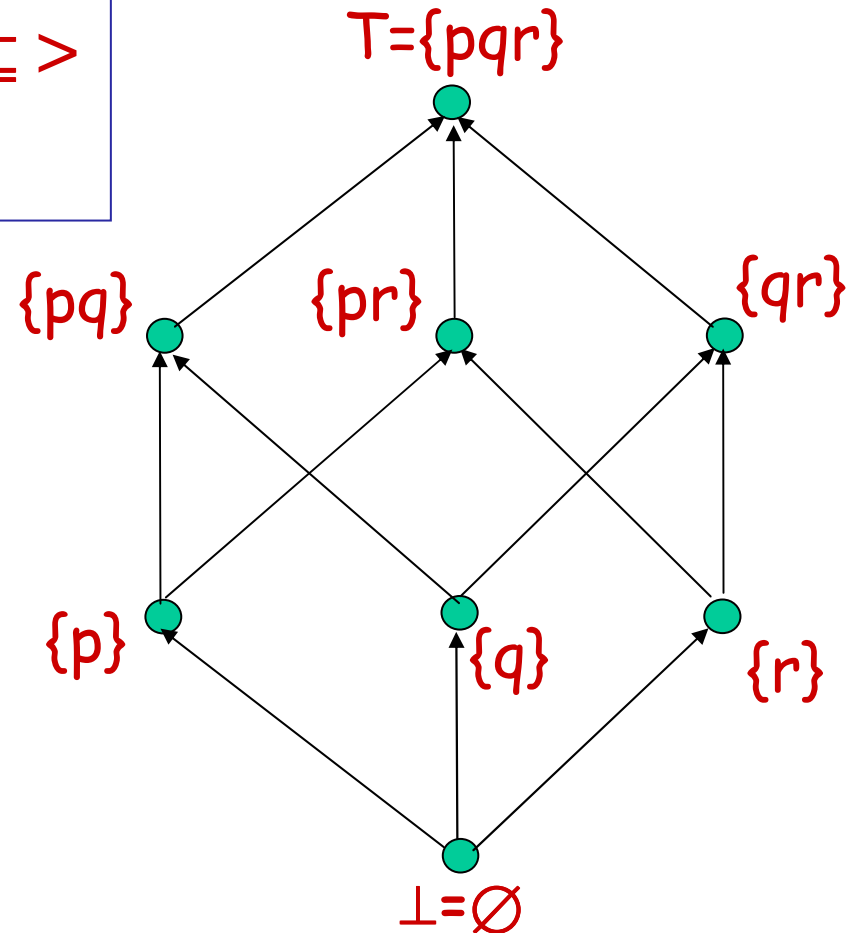
Any two subset of S have a *least upper* and a *greatest lower bound*.

Example: $S = \{a, b, c, d\}$. For $\{a, c\}$ and $\{b, c\}$ the *glb* is $\{c\}$, while the *lub* is $\{a, b, c\}$.

There is a unique *greatest element* $\cup 2^S = S$ and a unique *least element* $\cap 2^S = \emptyset$.

Example of a complete lattice

The complete lattice $\langle 2^S, \subseteq \rangle$
when S is the set $\{p, q, r\}$.



Monotonic functions

- A function $F: A \rightarrow A$ is *monotonic* if for each $a, b \in A$, $a \sqsubseteq b$ implies $F(a) \sqsubseteq F(b)$.
- In other words, a function F is monotonic if it *preserves the ordering* \sqsubseteq .

Fixed points

- Given a function $\mathbf{F}: \mathbf{A} \rightarrow \mathbf{A}$, an element $\mathbf{a} \in \mathbf{A}$ is a *fixed point* of \mathbf{F} if $\mathbf{F}(\mathbf{a}) = \mathbf{a}$.
- $\mathbf{a} \in \mathbf{A}$ is called the *least fixed point* of \mathbf{F} ($\mu \mathbf{x}.\mathbf{F}(\mathbf{x})$), if for all $\mathbf{a}' \in \mathbf{A}$ such that $\mathbf{F}(\mathbf{a}') = \mathbf{a}'$, then $\mathbf{a} \sqsubseteq \mathbf{a}'$.
- $\mathbf{a} \in \mathbf{A}$ is called the *greatest fixed point* of \mathbf{F} ($\nu \mathbf{x}.\mathbf{F}(\mathbf{x})$), if for all $\mathbf{a}' \in \mathbf{A}$ such that $\mathbf{F}(\mathbf{a}') = \mathbf{a}'$, then $\mathbf{a}' \sqsubseteq \mathbf{a}$.

Tarski's Fixed Point theorem

THEOREM: Let $\langle A, \sqsubseteq \rangle$ be a *complete lattice*, and $F: A \rightarrow A$ a monotonic function. Then F has a *least* and a *greatest fixed point* given, respectively, by:

- $\mu x.F(x) = \sqcap \{x \in A \mid F(x) \sqsubseteq x\}$
- $\nu x.F(x) = \sqcup \{x \in A \mid x \sqsubseteq F(x)\}$

Fixed point in finite lattices

Let $\langle A, \sqsubseteq \rangle$ be a *finite complete lattice*, and $F: A \rightarrow A$ be a monotonic function.

The *least element* of A

Then the *least fixed point* for F is obtained as

$$\mu x.F(x) = F^m(\perp)$$

for some m , where $F^0(\perp) = \perp$, and $F^{n+1}(\perp) = F(F^n(\perp))$.

Moreover, the *greatest fixed point* for F is obtained as

$$\nu x.F(x) = F^k(\top)$$

for some k , where $F^0(\top) = \top$, and $F^{n+1}(\top) = F(F^n(\top))$.

The *greatest element* of A

Generic fixed point algorithm

Algorithm `Compute_lfp(F:function)`

$\mathbf{X}_0 := \perp;$

$\mathbf{X}_1 := \mathbf{F}(\mathbf{X}_0);$

$j=1;$

while $\mathbf{X}_j \neq \mathbf{X}_{j-1}$

$j := j+1;$

$\mathbf{X}_j := \mathbf{F}(\mathbf{X}_{j-1});$

return \mathbf{X}_j

CTL and complete lattices

- Given a Kripke structure $M = \langle S, S_0, R, L, AP \rangle$. We will then consider the *poset* $\langle 2^S, \subseteq \rangle$.
- $\langle 2^S, \subseteq \rangle$ is clearly a *complete lattice* (with respect to intersection and union).
- We will identify a *CTL formula* with the *set of states* which *satisfy it*.
- In this way we can define *temporal operators* as *functions* on the *complete lattice* $\langle 2^S, \subseteq \rangle$.

Denotation of a CTL formula

- Given a formula ϕ , let us define its *denotation* (in \mathbf{M}), in symbols $\llbracket \phi \rrbracket$, as the set of states satisfying the formula:

$$\llbracket \phi \rrbracket = \{ s \mid \mathbf{M}, s \models \phi \}$$

- We could then define the cpo $\langle \mathbf{CTL}, \sqsubseteq \rangle$ by:

$$\phi \sqsubseteq \psi \text{ iff } \llbracket \phi \rrbracket \subseteq \llbracket \psi \rrbracket$$

Denotation of a CTL formula

- Given the *denotation* of a formula

$$|[\phi]| = \{ s \mid M, s \models \phi \}$$

- We could then define the cpo $\langle CTL, \sqsubseteq \rangle$ by:

$$\phi \sqsubseteq \psi \text{ iff } |[\phi]| \subseteq |[\psi]|$$

- Then $|[\perp]| = \emptyset$; $|[\top]| = S$;

- $|[p]| = \{ s \mid p \in L(s) \}$;

- $|[\neg\phi]| = S \setminus |[\phi]|$;

- $|[\phi \vee \psi]| = |[\phi]| \cup |[\psi]|$;

- $|[\phi \wedge \psi]| = |[\phi]| \cap |[\psi]|$;

CTL is closed under *conjunction* and *disjunction*, therefore for any pair of formulae the *upper* and *lower bound* do exist.

Denotation of a CTL formula

- Given a formula ϕ , let us define its *denotation* (in \mathbf{M}), in symbols $\llbracket \phi \rrbracket$, as the set of states satisfying the formula:

$$\llbracket \phi \rrbracket = \{ s \mid \mathbf{M}, s \models \phi \}$$

-
- $\llbracket \mathbf{EX}\phi \rrbracket = \{ s \mid \exists t. (t \in \llbracket \phi \rrbracket \cap \mathbf{R}(s)) \}$
- for the other **temporal operators** we would need to use **fixed points....**

Fixed point characterization of $\mathbf{EU}(\beta_1, \beta_2)$

- $\mathbf{EU}(\beta_1, \beta_2) \equiv \beta_2 \vee (\beta_1 \wedge \mathbf{EX} \mathbf{EU}(\beta_1, \beta_2))$
- $\llbracket \mathbf{EU}(\beta_1, \beta_2) \rrbracket = \mu \mathbf{Z} . (\llbracket \beta_2 \rrbracket \cup (\llbracket \beta_1 \rrbracket \cap \llbracket \mathbf{EX} \mathbf{Z} \rrbracket))$
- $\llbracket \mathbf{EU}(\beta_1, \beta_2) \rrbracket =$
 $\mu \mathbf{Z} . (\llbracket \beta_2 \rrbracket \cup (\llbracket \beta_1 \rrbracket \cap \{ s \mid \exists t \in \mathbf{Z} \cap \mathbf{R}(s) \}))$

Fixed point characterization of $\mathbf{EU}(\beta_1, \beta_2)$

Lemma: Let

$$\mathbf{F}(\mathbf{Z}) = ([\beta_2] \cup ([\beta_1] \cap \{s \mid \exists t \in \mathbf{Z} \cap \mathbf{R}(s)\}))$$

then \mathbf{F} is a *monotonic function*, i.e.

$$\mathbf{Z}_1 \subseteq \mathbf{Z}_2 \text{ implies } \mathbf{F}(\mathbf{Z}_1) \subseteq \mathbf{F}(\mathbf{Z}_2)$$

Fixed point characterization of $\mathbf{EU}(\beta_1, \beta_2)$

Theorem:

$$\llbracket \mathbf{EU}(\beta_1, \beta_2) \rrbracket = \mu \mathbf{Z}. (\llbracket \beta_2 \rrbracket \cup (\llbracket \beta_1 \rrbracket \cap \{ s \mid \exists t \in \mathbf{Z} \cap \mathbf{R}(s) \}))$$

in other words:

$$\mu \mathbf{Z}. (\llbracket \beta_2 \rrbracket \cup (\llbracket \beta_1 \rrbracket \cap \{ s \mid \exists t \in \mathbf{Z} \cap \mathbf{R}(s) \})) \subseteq \llbracket \mathbf{EU}(\beta_1, \beta_2) \rrbracket$$

and

$$\llbracket \mathbf{EU}(\beta_1, \beta_2) \rrbracket \subseteq \mu \mathbf{Z}. (\llbracket \beta_2 \rrbracket \cup (\llbracket \beta_1 \rrbracket \cap \{ s \mid \exists t \in \mathbf{Z} \cap \mathbf{R}(s) \}))$$

Computing fixed point for $EU(\beta_1, \beta_2)$

Algorithm Compute_EU(β_1, β_2)

$X_0 := \llbracket \perp \rrbracket$; /* i.e. $X_0 := \emptyset$ */

$X_1 := \llbracket \beta_2 \rrbracket \cup (\llbracket \beta_1 \rrbracket \cap X_0)$; /* i.e. $X_1 := \llbracket \beta_2 \rrbracket$ */

$j=1$;

while $X_j \neq X_{j-1}$

$j := j+1$; $T := X_{j-1}$; $X := \emptyset$

 while $T \neq \emptyset$ do

 chose $s \in T$;

$T := T \setminus \{s\}$;

 forall t such that $s \in R(t)$ do

$X := X \cup \{t\}$;

$X_j := \llbracket \beta_2 \rrbracket \cup (\llbracket \beta_1 \rrbracket \cap X)$

This computes
 $X = EX X_{j-1}$

Computing fixed point for $\mathbf{EU}(\beta_1, \beta_2)$

To compute $\llbracket \mathbf{EU}(\beta_1, \beta_2) \rrbracket$ we can *construct inductively the set* of states \mathbf{X}_j as follows:

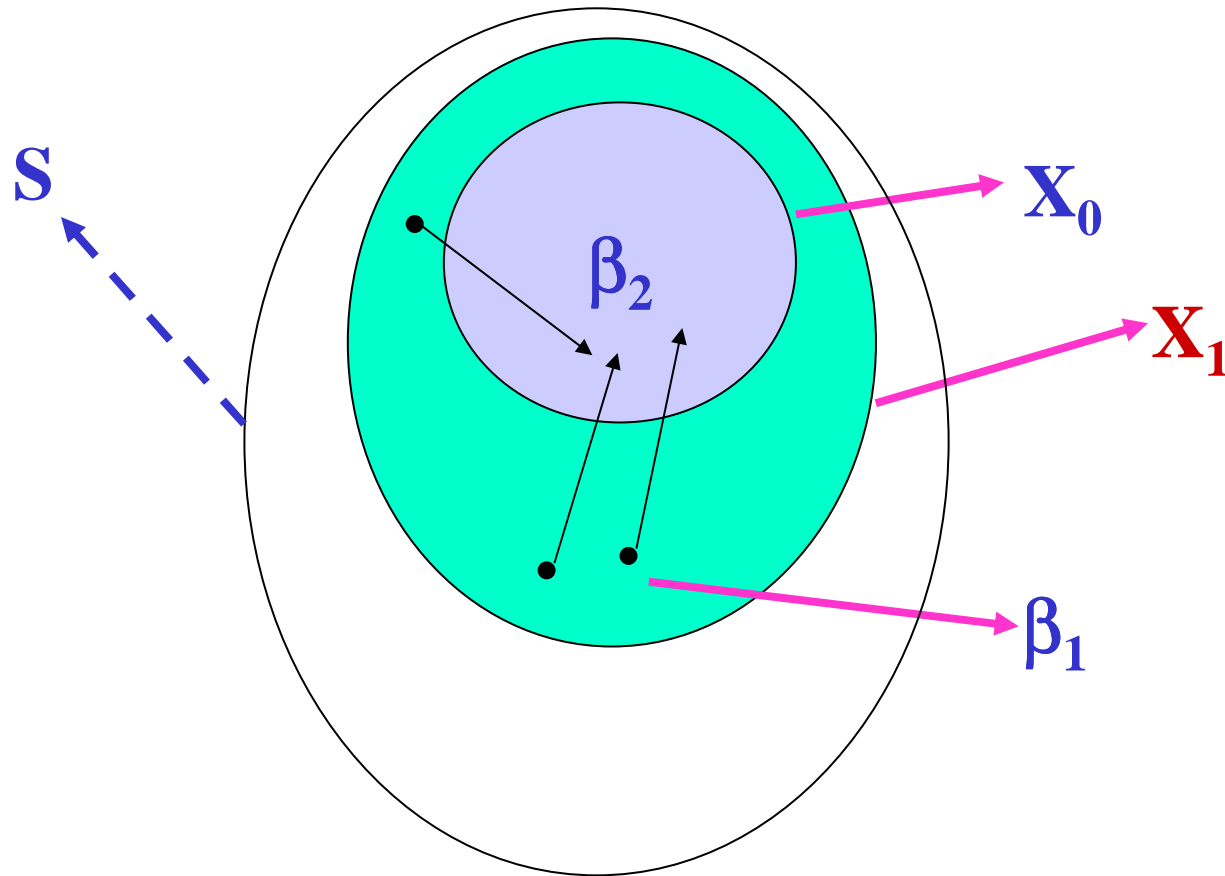
- $\mathbf{X}_1 = \llbracket \beta_2 \rrbracket$.
- $\mathbf{X}_{j+1} = \mathbf{X}_j \cup \{ s \mid s \in \llbracket \beta_1 \rrbracket \text{ and } \mathbf{R}(s, t) \text{ for some } t \in \mathbf{X}_j \}$

$\llbracket \mathbf{EU}(\beta_1, \beta_2) \rrbracket$ is then the set \mathbf{X} such that $\mathbf{X} = \mathbf{X}_n$ for n such that $\mathbf{X}_{n+1} = \mathbf{X}_n$.

Notice that n *must exist* by *Tarski's Theorem* since $\mathbf{X}_j \subseteq \mathbf{X}_{j+1} \subseteq \mathbf{S}$ (and \mathbf{S} is finite!)

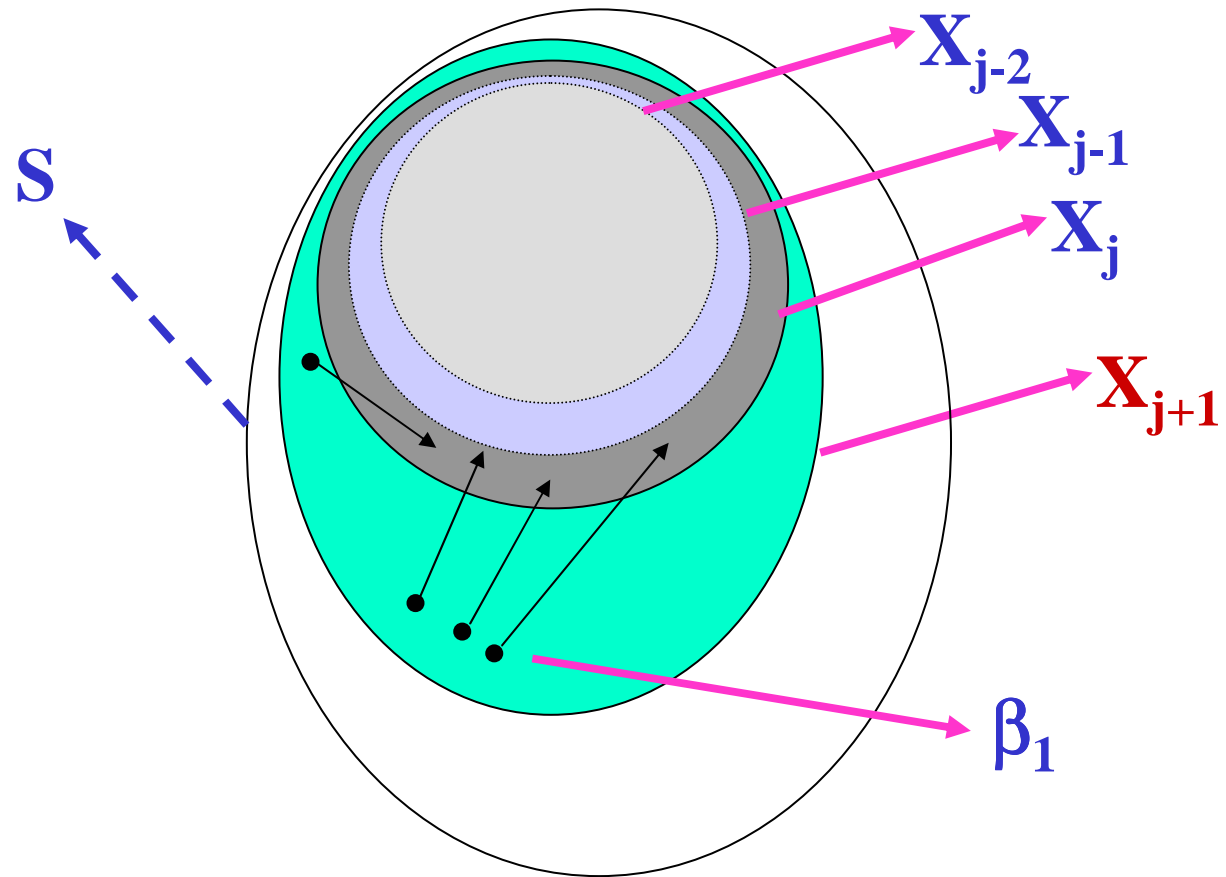
From X_0 to X_1

$EU(\beta_1, \beta_2)$



From X_j to X_{j+1}

$EU(\beta_1, \beta_2)$



Computing fixed point for $EU(\beta_1, \beta_2)$

Algorithm $\text{Compute_EU}(\beta_1, \beta_2)$

$X_1 := \llbracket \beta_2 \rrbracket$;

$j=1$;

repeat

$j := j+1$; $T := X := X_{j-1}$;

 while $T \neq \emptyset$ do

 chose $s \in T$;

$T := T \setminus \{s\}$;

 forall t such that $s \in R(t)$ do

 if $t \in \llbracket \beta_1 \rrbracket$ then /* $t \in \llbracket \beta_1 \rrbracket \cap EX X_{j-1}$ */

$X := X \cup \{t\}$;

$X_j = X$;

until $X_{j-1} = X_j$

Fixed point characterization of $\mathbf{EG}(\beta)$

- $\mathbf{EG}(\beta) \equiv \beta \wedge \mathbf{EX} \mathbf{EG}(\beta)$
- $|\mathbf{[EG}(\beta)\mathbf{]}| = \nu \mathbf{Z} . (|\mathbf{[\beta]}| \cap |\mathbf{[EX Z]}|)$
- $|\mathbf{[EG}(\beta)\mathbf{]}| =$
 $\nu \mathbf{Z} . (|\mathbf{[\beta]}| \cap \{ s \mid \exists t \in \mathbf{Z} \cap \mathbf{R}(s) \})$

Computing the fixed point for $EG(\beta)$

Algorithm $\text{Compute_EG}(\beta)$

$X_0 := \llbracket T \rrbracket$; /* i.e. $X_0 := S$ */

$X_1 := \llbracket \beta \rrbracket \cap X_0$; /* i.e. $X_1 := \llbracket \beta \rrbracket$ */

$j=1$;

while $X_j \neq X_{j-1}$

$j := j+1$; $T := X_{j-1}$; $X := \emptyset$;

 while $T \neq \emptyset$ do

 chose $s \in T$;

$T := T \setminus \{s\}$;

 forall t such that $s \in R(t)$

$X_j := X_j \cup \{t\}$;

$X_j := \llbracket \beta \rrbracket \cap X_j$

$X = EX X_{j-1}$

The Labels function

- To compute $[[\mathbf{EG}\beta]]$ we can *construct inductively the set* of states \mathbf{X}_j as follows:

$$- \mathbf{X}_1 = [[\beta]].$$

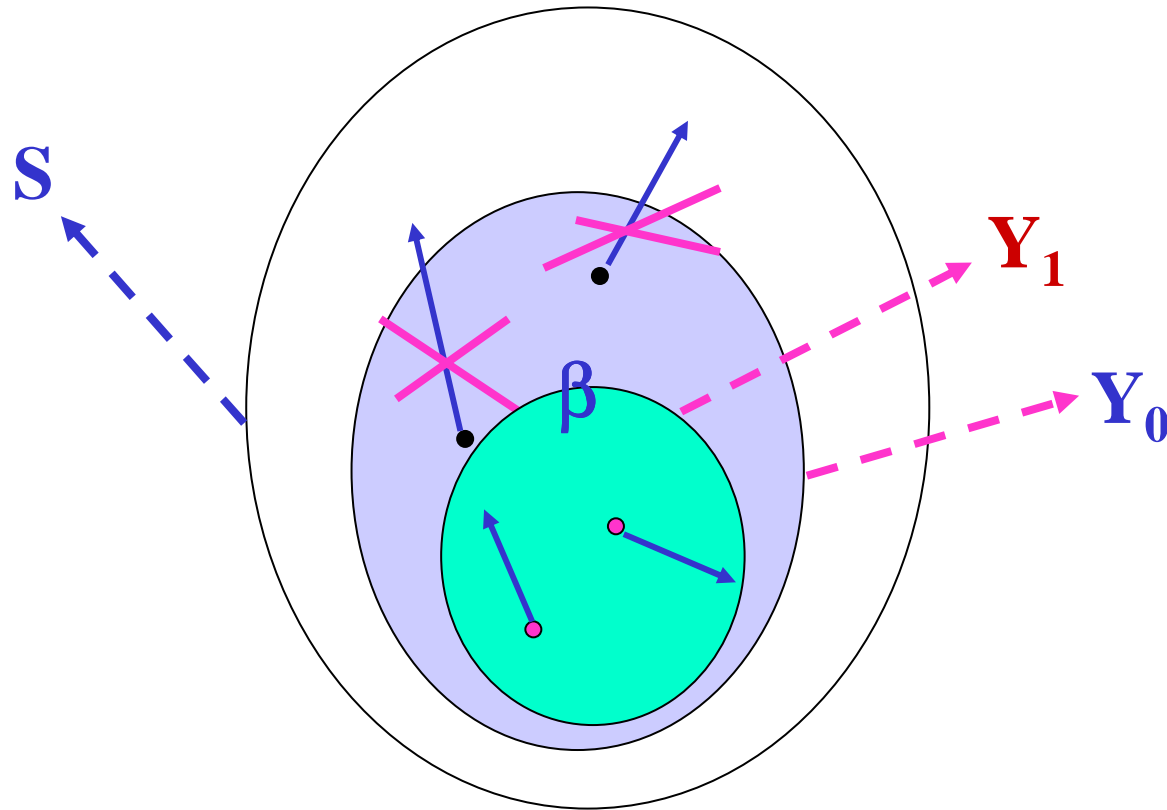
$$- \mathbf{X}_{j+1} = \mathbf{X}_j - \{s \mid s \in \mathbf{X}_j \text{ and} \\ \textit{there does not exist } t \in \mathbf{X}_j \\ \text{such that } \mathbf{R}(s, t)\}$$

$[[\mathbf{EG}\beta]]$ is then the set \mathbf{X} such that $\mathbf{X} = \mathbf{X}_n$ for \mathbf{m} such that $\mathbf{X}_{m+1} = \mathbf{X}_m$.

- Notice that \mathbf{m} *must exist* by *Tarski's Theorem* since $\emptyset \subseteq \mathbf{X}_{j+1} \subseteq \mathbf{X}_j$

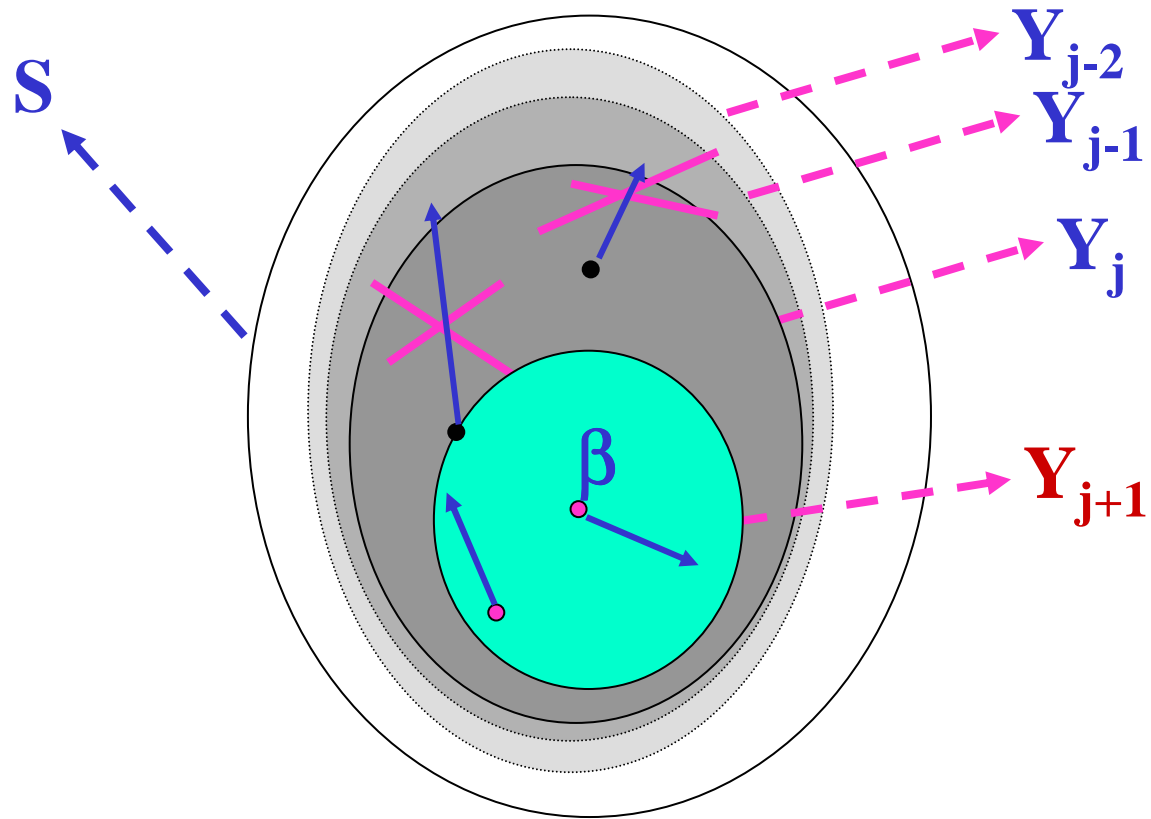
From Y_0 to Y_1

EG β



From Y_j to Y_{j+1}

EG β



Computing the fixed point for $EG(\beta)$

Algorithm Compute_EG(β)

$X_1 := \llbracket \beta \rrbracket$;

$j=1$;

repeat

$j := j+1$; $T := X_j := X_{j-1}$;

 while $T \neq \emptyset$ do

 chose $s \in T$;

$T := T \setminus \{s\}$;

 if for all $t \in R(s)$, $t \notin X_{j-1}$ then

$X_j := X_j - \{s\}$;

until $X_j = X_{j-1}$;