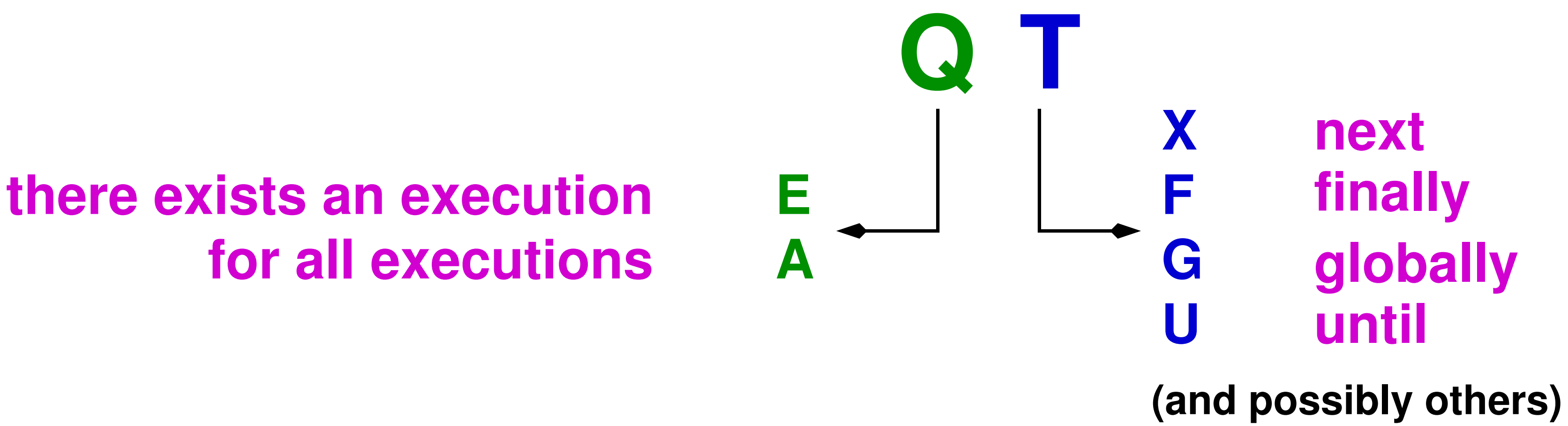
CTL: Overview

We now define CTL (Computation-Tree Logic) as a syntactic restriction of CTL*.

Operators are restricted to the following form:



CTL: Syntax

We define a minimal syntax first. Later we define additional operators with the help of the minimal syntax.

Let *AP* be a set of atomic propositions: The set of CTL formulas over *AP* is as follows:

if $a \in AP$, then a is a CTL formula;

if ϕ_1, ϕ_2 are CTL formulas, then so are

$$\neg\phi_1, \qquad \phi_1 \vee \phi_2, \qquad \mathbf{EX}\,\phi_1, \qquad \mathbf{EG}\,\phi_1, \qquad \phi_1\,\mathbf{EU}\,\phi_2$$

It is easy to see that every CTL formula is also a CTL* formula.

Previously, we defined the satisfaction relationship between valuation trees and CTL* formulae. Since each state of a Kripke structure has a clearly defined computation tree, we may just as well say that a *state* satisfies a CTL/CTL* formula, meaning that its computation tree does.

Let $\mathcal{K}$ be a Kripke structure, let s one of its states, and let ϕ be a CTL formula. On the following slide, we define a set $[\![\phi]\!]_{\mathcal{K}}$ in such a way that $s \in [\![\phi]\!]_{\mathcal{K}}$ iff $\mathcal{T}_{\mathcal{K}}(s) \models \phi$.

CTL: Semantics

Let $\mathcal{K} = (S, \to, r, AP, \nu)$ be a Kripke structure.

We define the semantic of every CTL formula ϕ over AP w.r.t. $\mathcal{K}$ as a set of states $[\![\phi]\!]_\mathcal{K}$, as follows:

$$
\begin{aligned}
[\![a]\!]_\mathcal{K} &= \{\, s \mid a \in \nu(s) \,\} \qquad \text{for } a \in AP \\
[\![\neg\phi_1]\!]_\mathcal{K} &= S \setminus [\![\phi_1]\!]_\mathcal{K} \\
[\![\phi_1 \vee \phi_2]\!]_\mathcal{K} &= [\![\phi_1]\!]_\mathcal{K} \cup [\![\phi_2]\!]_\mathcal{K} \\
[\![\mathbf{EX}\,\phi_1]\!]_\mathcal{K} &= \{\, s \mid \text{there is a } t \text{ s.t. } s \to t \text{ and } t \in [\![\phi_1]\!]_\mathcal{K} \,\} \\
[\![\mathbf{EG}\,\phi_1]\!]_\mathcal{K} &= \{\, s \mid \text{there is a run } \rho \text{ with } \rho(0) = s \\
&\qquad \text{and } \rho(i) \in [\![\phi_1]\!]_\mathcal{K} \text{ for all } i \geq 0 \,\} \\
[\![\phi_1\,\mathbf{EU}\,\phi_2]\!]_\mathcal{K} &= \{\, s \mid \text{there is a run } \rho \text{ with } \rho(0) = s \text{ and } k \geq 0 \text{ s.t.} \\
&\qquad \rho(i) \in [\![\phi_1]\!]_\mathcal{K} \text{ for all } i < k \text{ and } \rho(k) \in [\![\phi_2]\!]_\mathcal{K} \,\}
\end{aligned}
$$

We say that $\mathcal{K}$ satisfies ϕ (denoted $\mathcal{K} \models \phi$) iff $r \in [\![\phi]\!]_{\mathcal{K}}$.

The local model-checking problem is to check whether $\mathcal{K} \models \phi$.

The global model-checking problem is to compute $[\![\phi]\!]_{\mathcal{K}}$.

We declare two formulas equivalent (written $\phi_1 \equiv \phi_2$) iff for every Kripke structure $\mathcal{K}$ we have $[\![\phi_1]\!]_{\mathcal{K}} = [\![\phi_2]\!]_{\mathcal{K}}$.

In the following, we omit the index $\mathcal{K}$ from $[\![\cdot]\!]_{\mathcal{K}}$ if $\mathcal{K}$ is understood.

CTL: Extended syntax

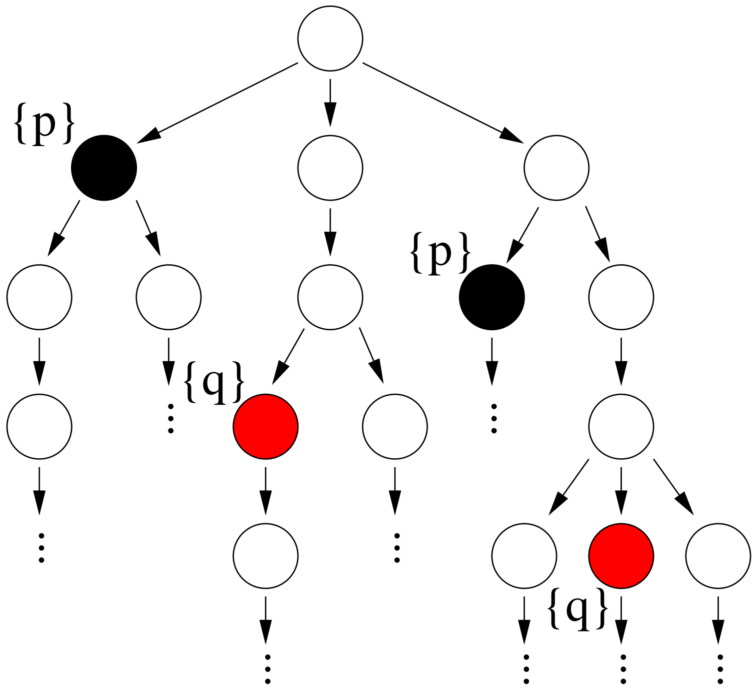
$$
\begin{aligned}
\phi_1 \wedge \phi_2 &\equiv \neg(\neg\phi_1 \vee \neg\phi_2) \\
\mathbf{true} &\equiv a \vee \neg a \\
\mathbf{false} &\equiv \neg\mathbf{true} \\
\phi_1\ \mathbf{EW}\ \phi_2 &\equiv \mathbf{EG}\,\phi_1 \vee (\phi_1\ \mathbf{EU}\ \phi_2) \\
\mathbf{EF}\,\phi &\equiv \mathbf{true}\ \mathbf{EU}\ \phi
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{AX}\,\phi &\equiv \neg\,\mathbf{EX}\,\neg\phi \\
\mathbf{AG}\,\phi &\equiv \neg\,\mathbf{EF}\,\neg\phi \\
\mathbf{AF}\,\phi &\equiv \neg\,\mathbf{EG}\,\neg\phi \\
\phi_1\ \mathbf{AW}\ \phi_2 &\equiv \neg(\neg\phi_2\,\mathbf{EU}\,\neg(\phi_1 \vee \phi_2)) \\
\phi_1\ \mathbf{AU}\ \phi_2 &\equiv \mathbf{AF}\,\phi_2 \wedge (\phi_1\ \mathbf{AW}\ \phi_2)
\end{aligned}
$$

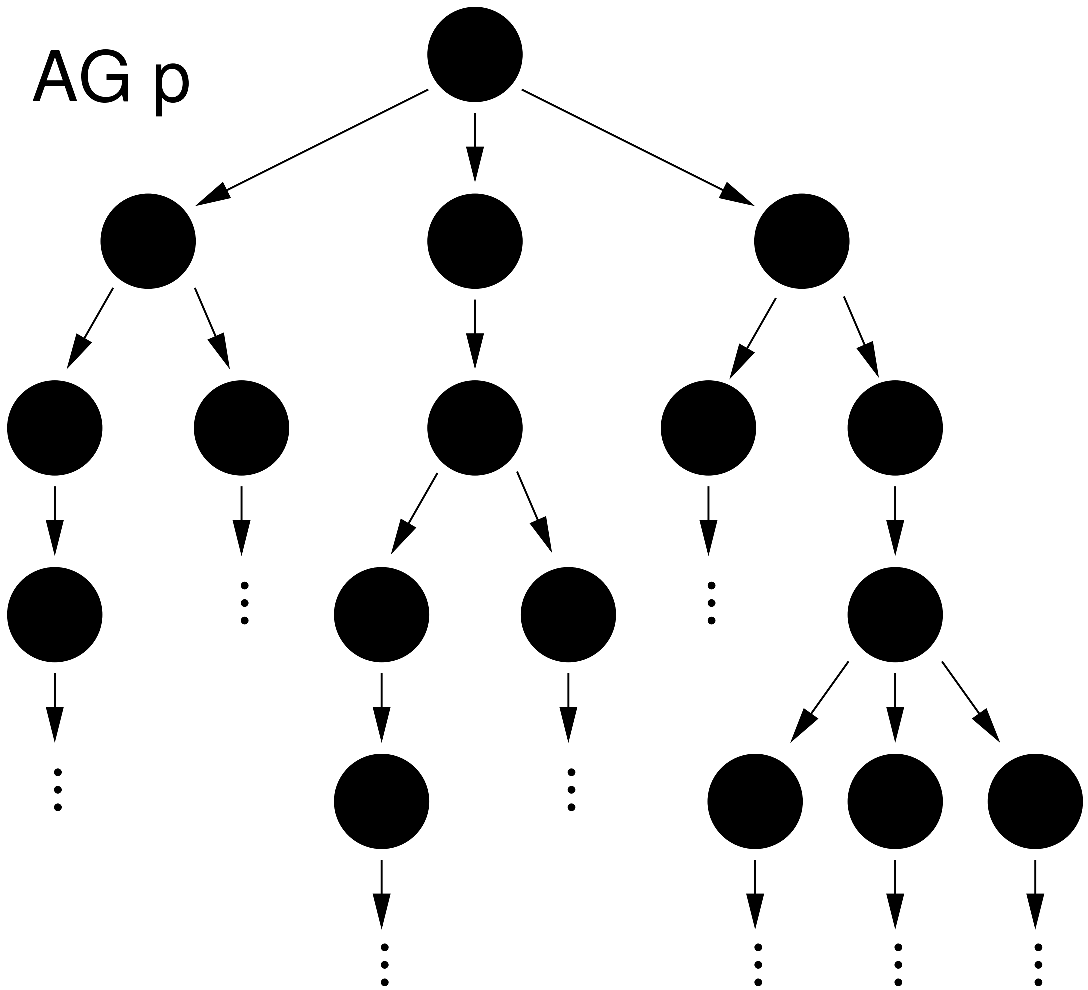
Other logical and temporal operators (e.g. $\rightarrow$, $\mathbf{ER}$, $\mathbf{AR}$), ... may also be defined.

CTL: Examples

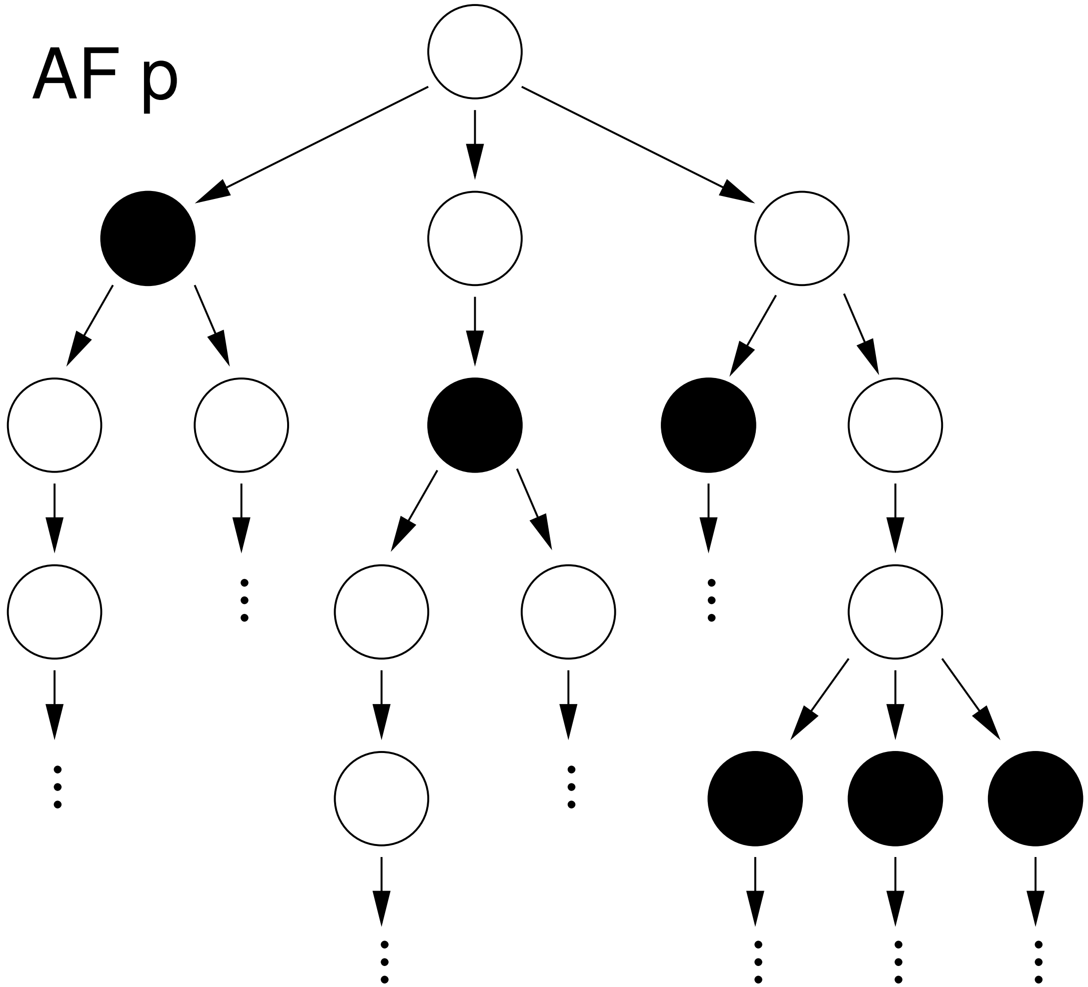
We use the following computation tree as a running example (with varying distributions of red and black states):

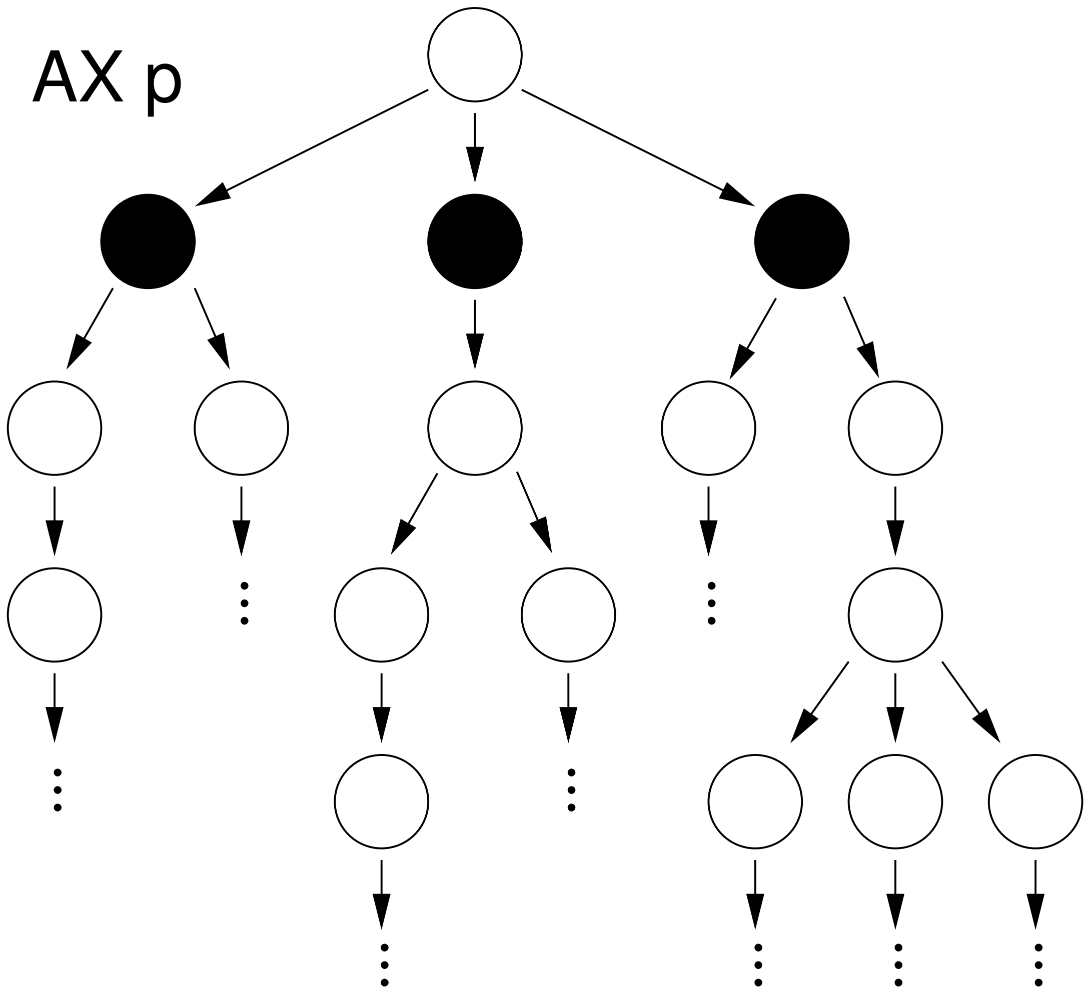


In the following slides, the topmost state satisfies the given formula if the black states satisfy *p* and the red states satisfy *q*.

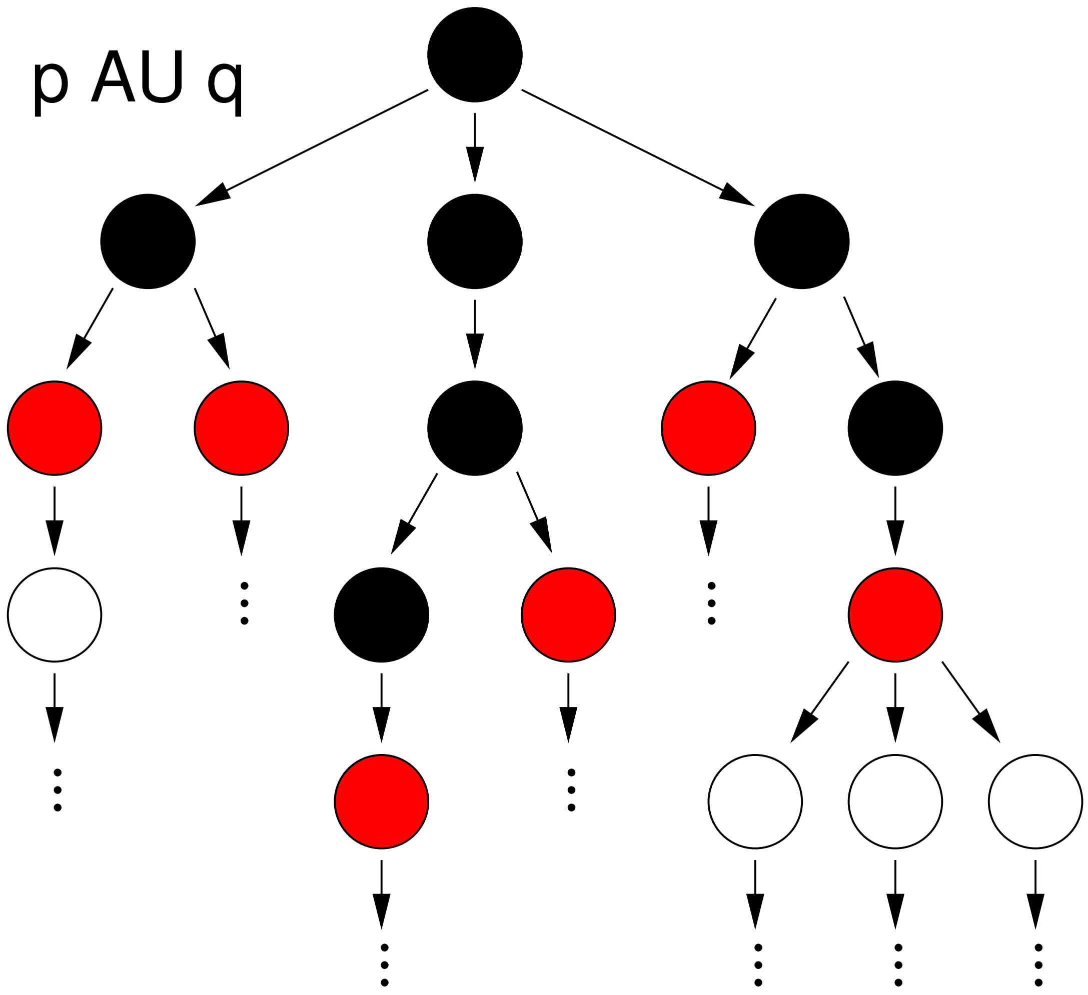

AG p

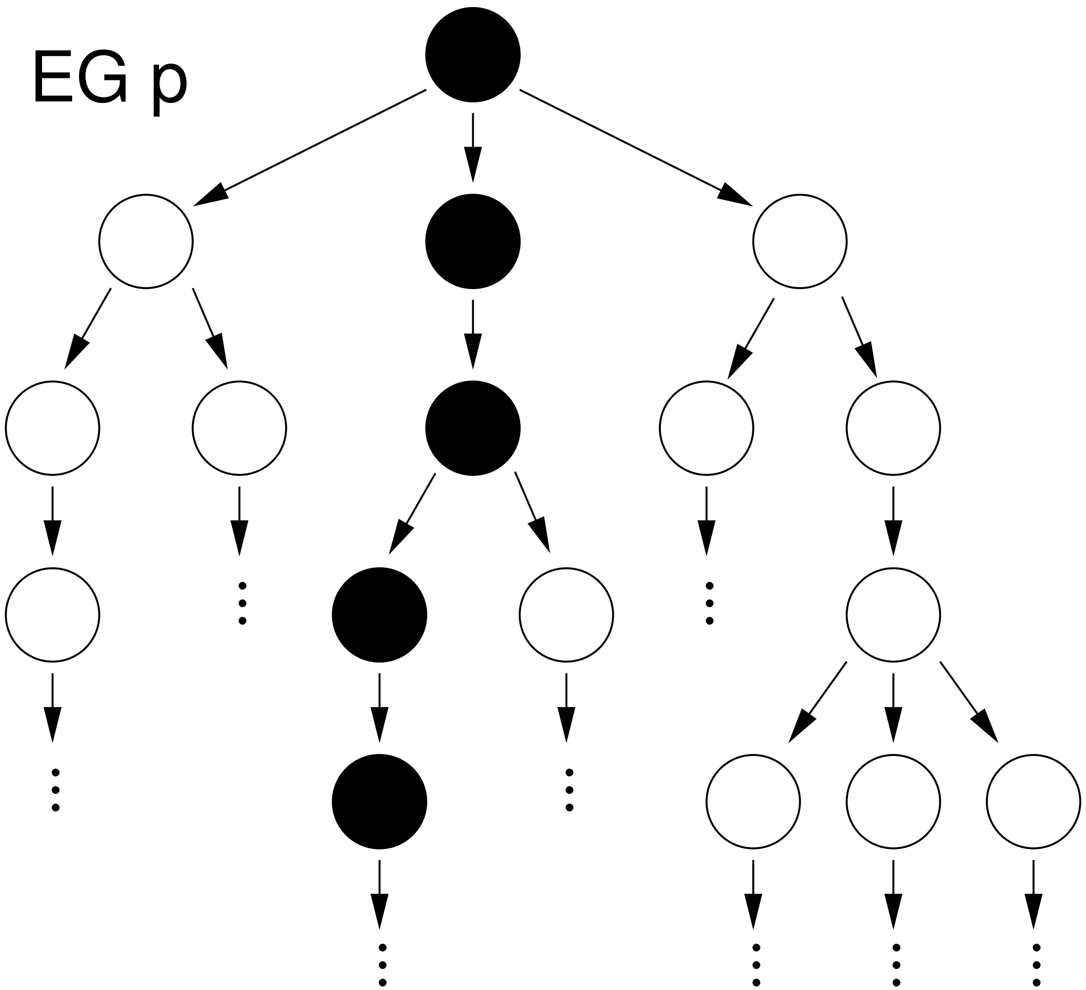
AF p

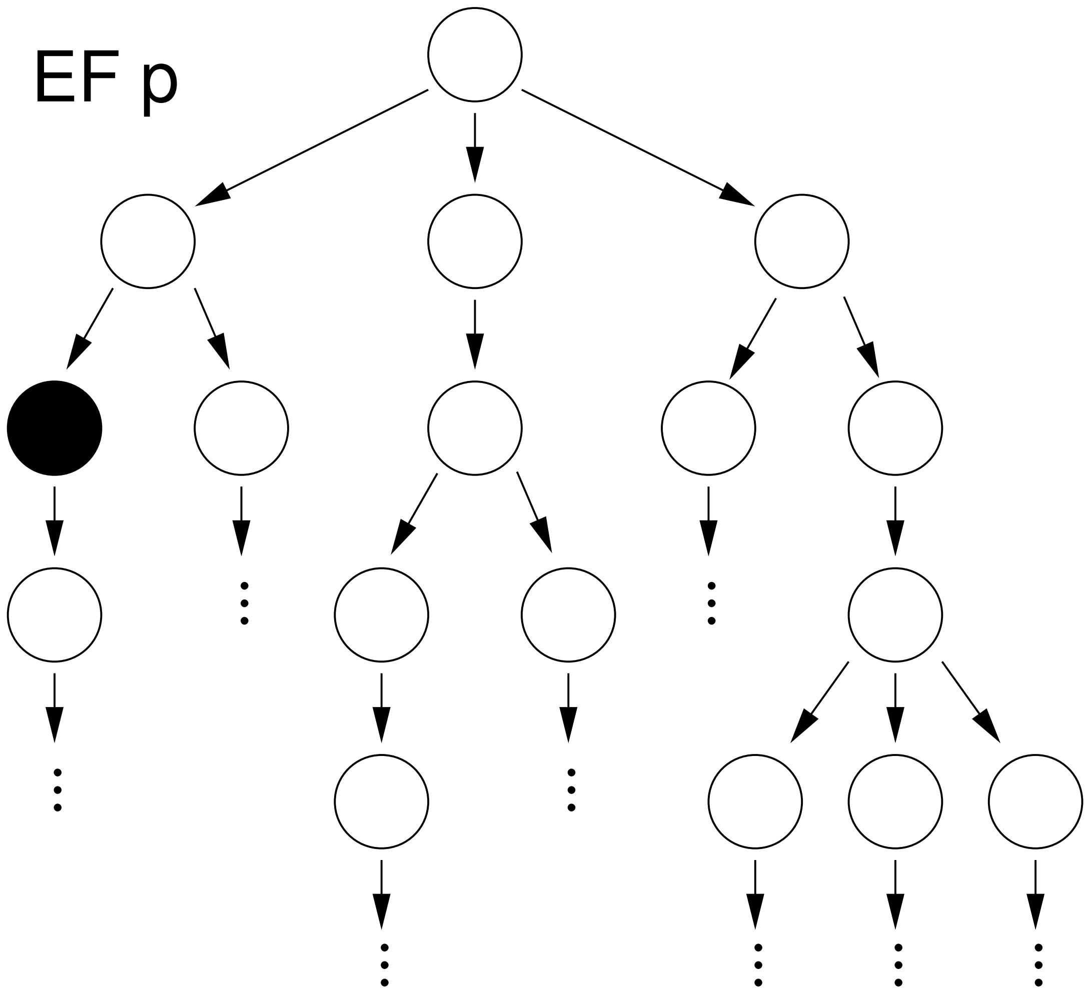


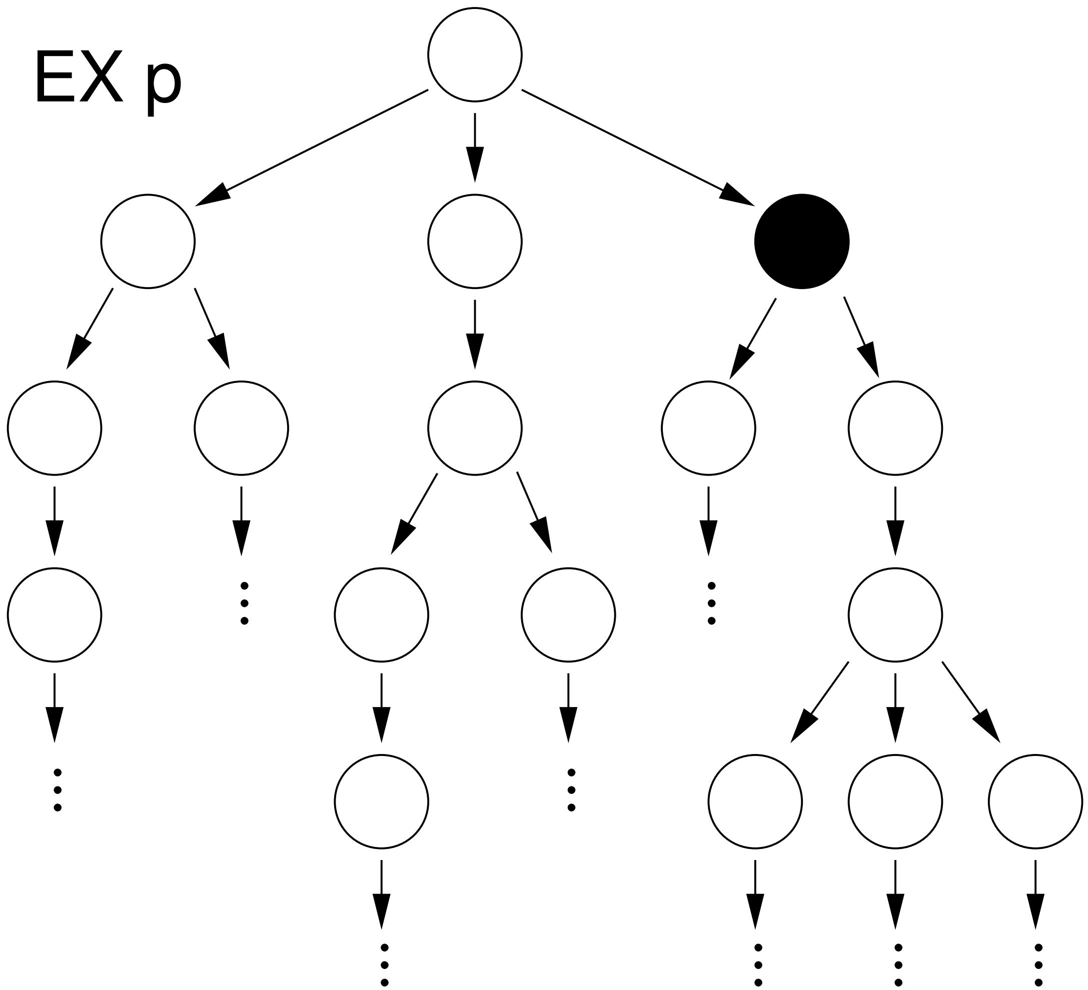

AX p

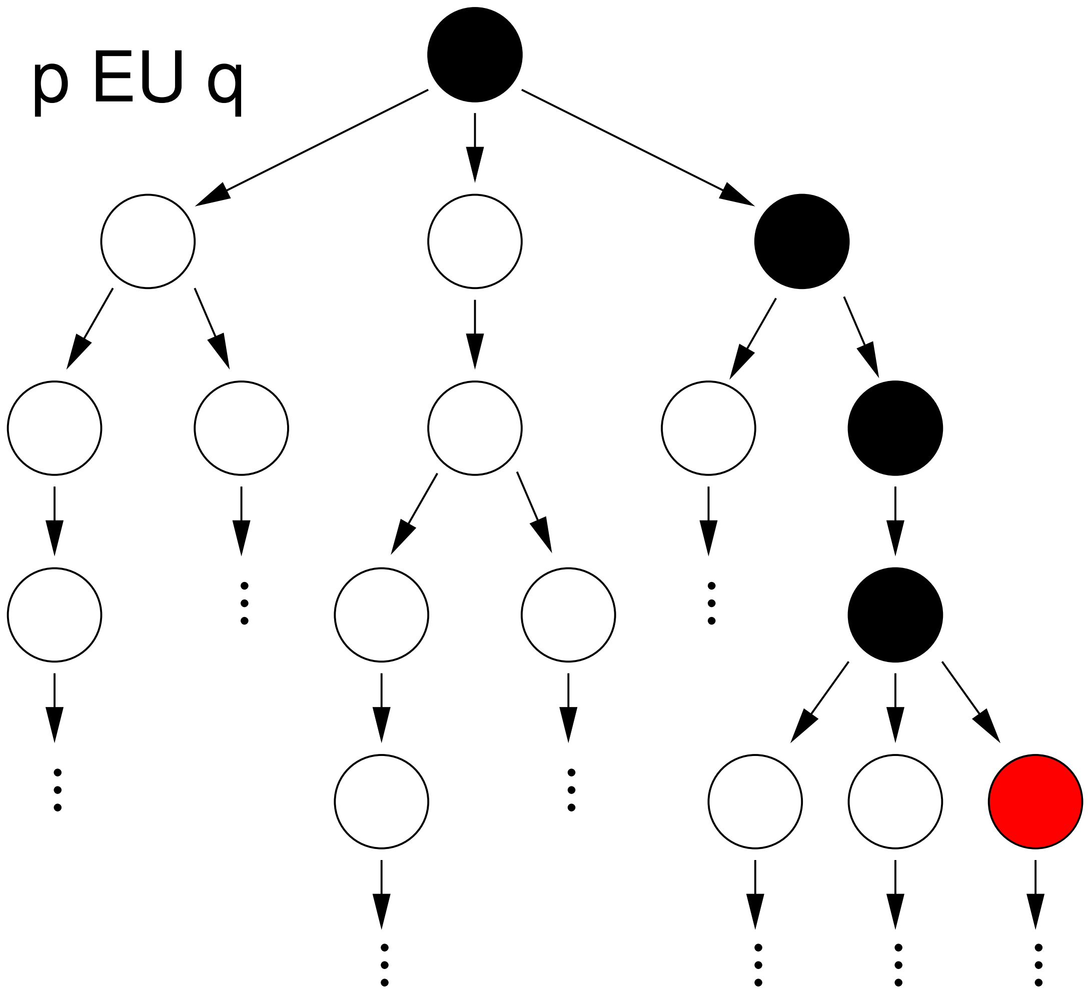
p AU q



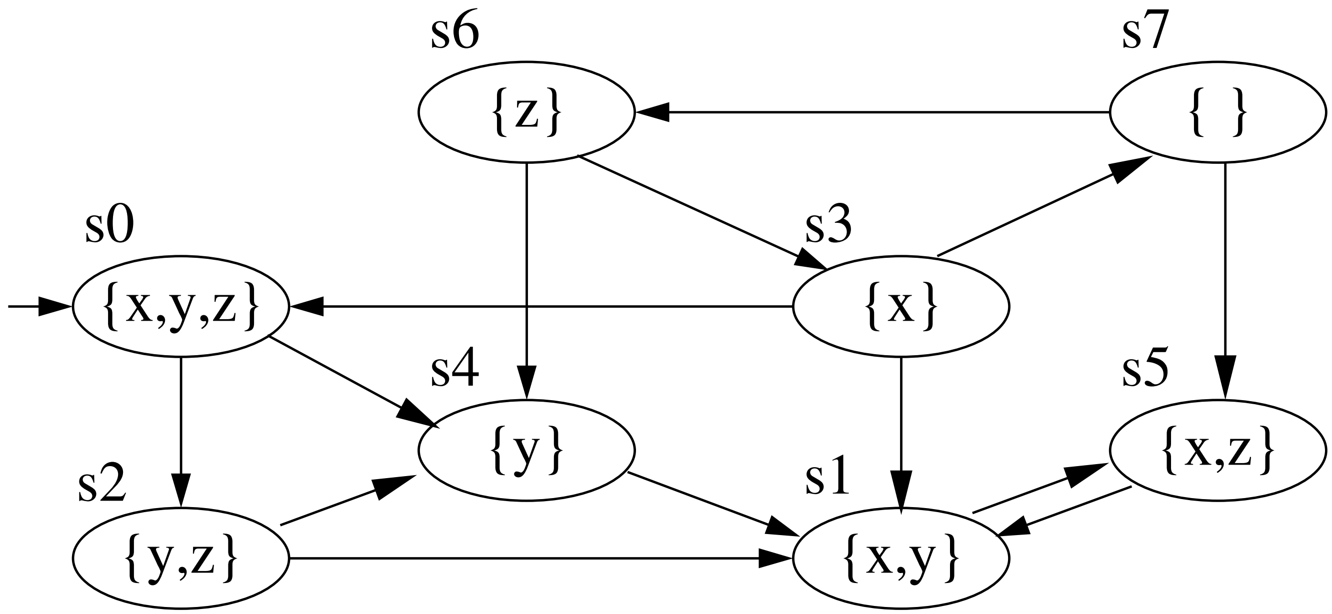

EG p


EF p


EX p


p EU q

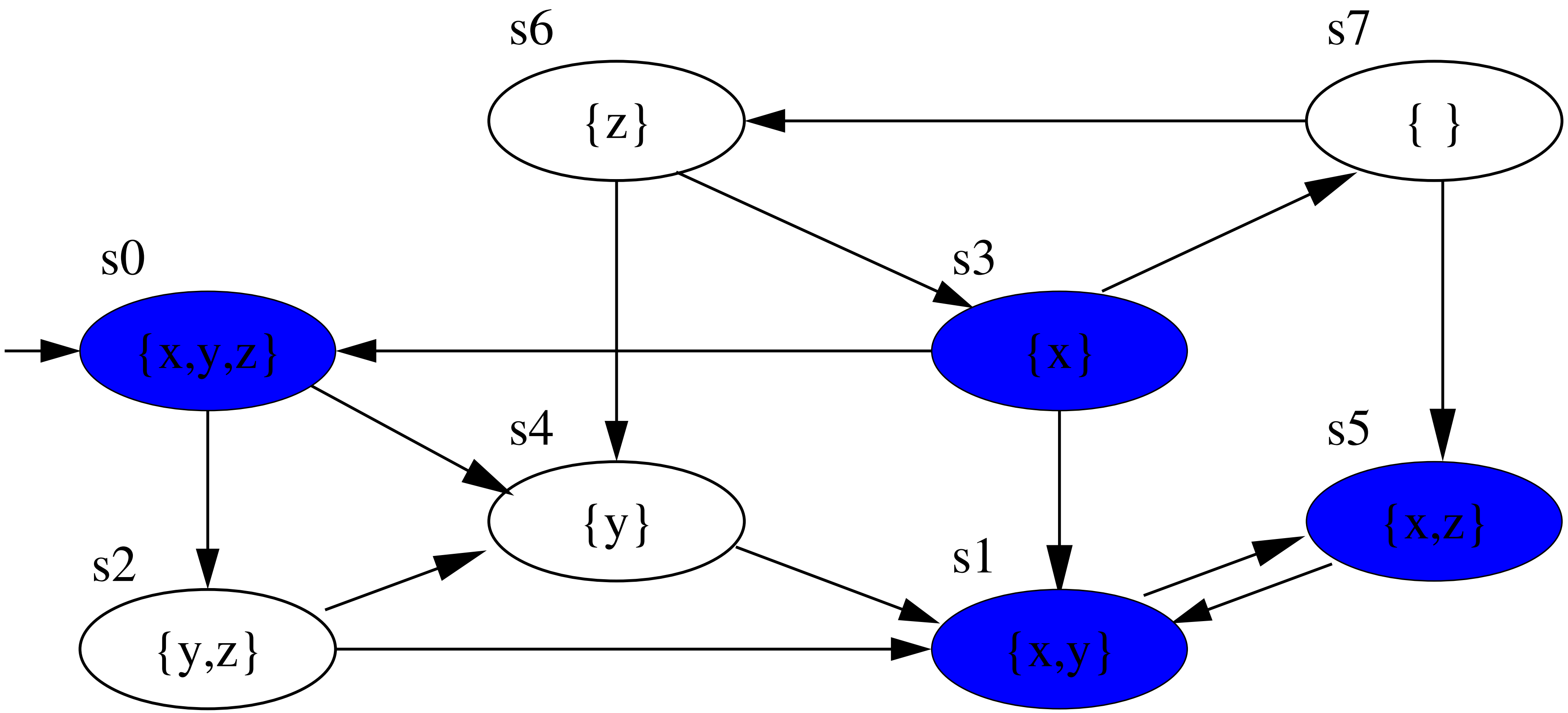
Solving nested formulas: Is $s_0 \in [\![\mathbf{AF}\ \mathbf{AG}\ x]\!]$?



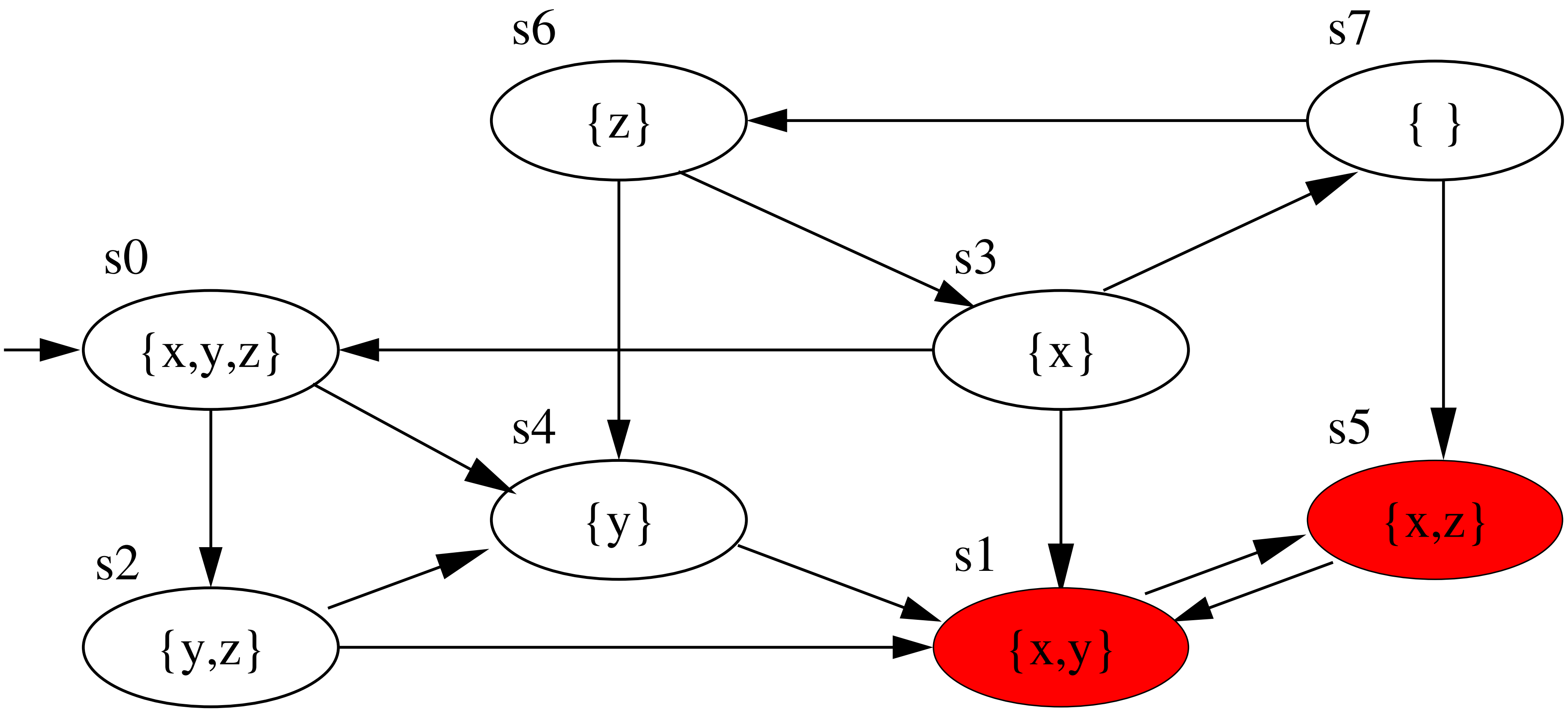
To compute the semantics of formulas with nested operators, we first compute the states satisfying the innermost formulas; then we use those results to solve progressively more complex formulas.

In this example, we compute $[\![x]\!]$, $[\![\mathbf{AG}\ x]\!]$, and $[\![\mathbf{AF}\ \mathbf{AG}\ x]\!]$, in that order.

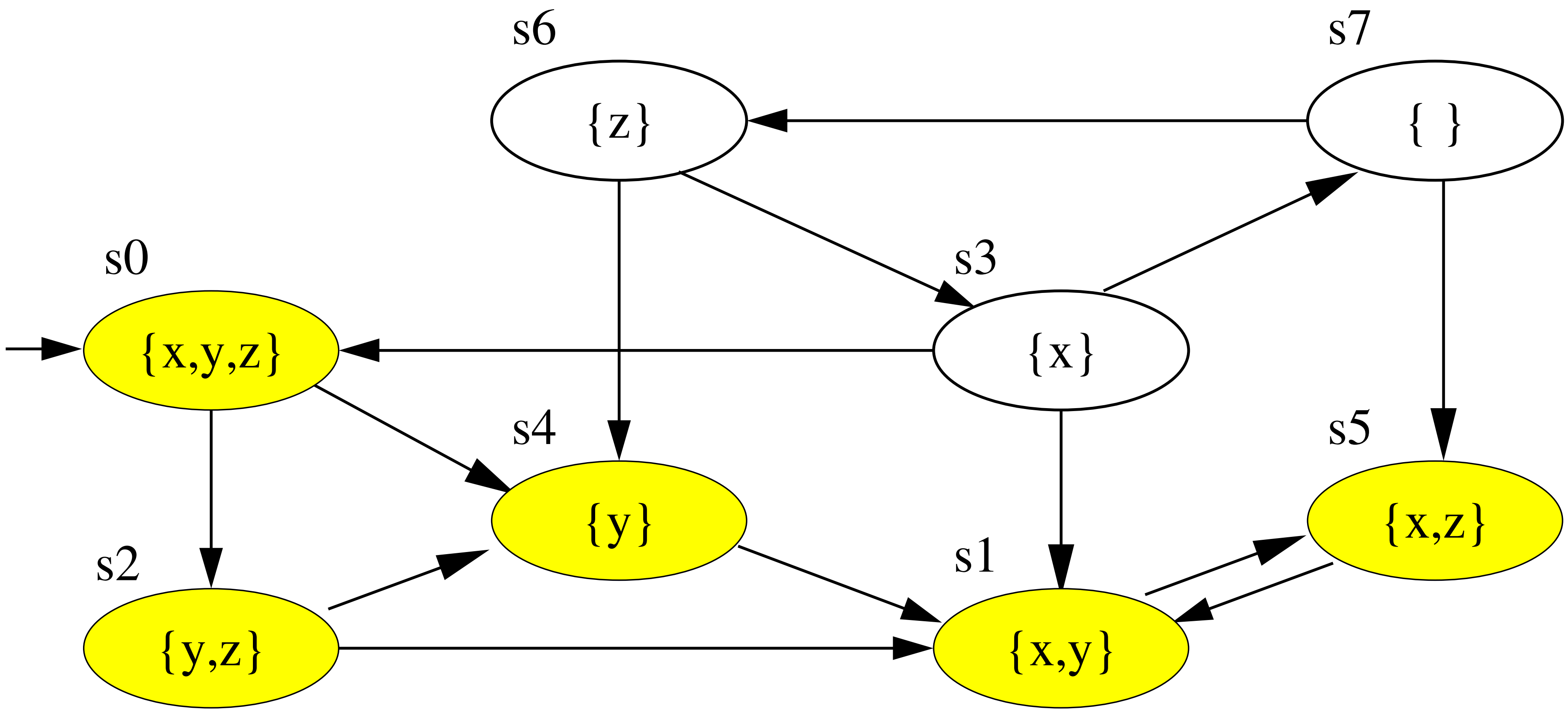
Bottom-up method (1): Compute $[\![x]\!]$



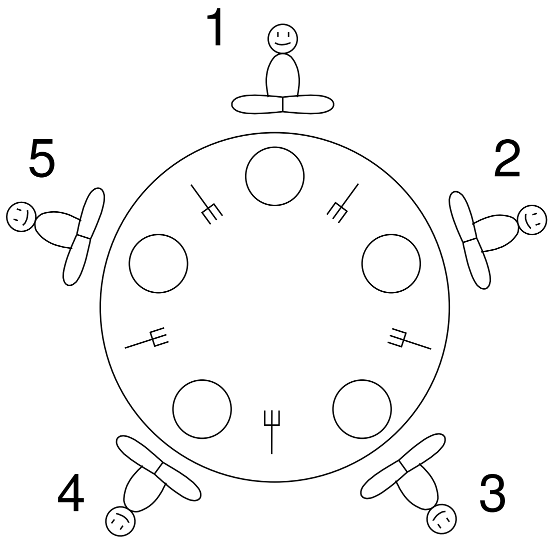
Bottom-up method (2): Compute $[\![\mathbf{AG}\ x]\!]$



Bottom-up method (3): Compute $[\![\mathbf{AF}\ \mathbf{AG}\ x]\!]$



Example: Dining Philosophers



Five philosophers are sitting around a table, taking turns at thinking and eating.

We shall express a couple of properties in CTL. Let us assume the following atomic propositions:

$e_i \mathrel{\widehat{=}}$ philosopher i is currently eating

Properties of the Dining Philosophers

"Philosophers 1 and 4 will never eat at the same time."

Properties of the Dining Philosophers

"Philosophers 1 and 4 will never eat at the same time."

$$\mathbf{AG}\,\neg(e_1 \wedge e_4)$$

"It is possible that Philosopher 3 never eats."

Properties of the Dining Philosophers

"Philosophers 1 and 4 will never eat at the same time."

$$\mathbf{AG}\,\neg(e_1 \wedge e_4)$$

"It is possible that Philosopher 3 never eats."

$$\mathbf{EG}\,\neg e_3$$

"From every situation on the table it is possible to reach a state where only philosopher 2 is eating."

Properties of the Dining Philosophers

"Philosophers 1 and 4 will never eat at the same time."

$$\mathbf{AG}\,\neg(e_1 \wedge e_4)$$

"It is possible that Philosopher 3 never eats."

$$\mathbf{EG}\,\neg e_3$$

"From every situation on the table it is possible to reach a state where only philosopher 2 is eating."

$$\mathbf{AG}\,\mathbf{EF}(\neg e_1 \wedge e_2 \wedge \neg e_3 \wedge \neg e_4)$$

Part 10: Algorithms for CTL

CTL-Model-Checking

In the following, let $\mathcal{K} = (S, \rightarrow, r, AP, \nu)$ be a Kripke structure (where S is finite) and ϕ a CTL formula over AP.

We shall solve the *global* model-checking problem for CTL, i.e. to compute $[\![\phi]\!]_{\mathcal{K}}$ (all states of $\mathcal{K}$ whose computation tree satisfies ϕ).

Our solution works “bottom-up”, i.e. it considers simple subformulae first, and then successively more complex ones.

The solution shown here considers only the minimal syntax. For additional efficiency one could extend it by treating some cases of the extended syntax more directly.

The ‘bottom-up’ algorithm for CTL

The algorithm reducs ϕ step by step to a single atomic proposition. Reminder: $[\![p]\!]_{\mathcal{K}} = \{\, s \mid p \in \nu(s) \,\}$ for $p \in AP$. In the following, we abbreviate this set as $\mu(p)$.

1. Check whether $\phi = p$, where $p \in AP$. If yes, output $\mu(p)$ and stop.

2. Otherwise, ϕ contains some subformula ψ of the form $\neg p$, $p \vee q$, $\mathbf{EX}\, p$, $\mathbf{EG}\, p$, or $p\, \mathbf{EU}\, q$, where $p, q \in AP$. Compute $[\![\psi]\!]_{\mathcal{K}}$ using the algorithms on the following slides.

3. Let $p' \notin AP$ be a “fresh” atomic proposition. Add p' to AP and set $\mu(p') := [\![\psi]\!]_{\mathcal{K}}$. Replace all occurrences of ψ in ϕ by p' and continue at step 1.

Computation of $[\![\psi]\!]_{\mathcal{K}}$: simple cases

Case 1: $\psi \equiv \neg p, \quad p \in AP$

By definition, $[\![\psi]\!]_{\mathcal{K}} = S \setminus \mu(p)$.

Case 2: $\psi \equiv p \vee q, \quad p, q \in AP$

Then $[\![\psi]\!]_{\mathcal{K}} = \mu(p) \cup \mu(q)$.

Case 3: $\psi \equiv \mathbf{EX}\, p, \quad p \in AP$

In the following, let $pre(X)$, for $X \subseteq S$, denote the set

$$pre(X) := \{\, s \mid \exists t \in X \colon s \rightarrow t \,\}.$$

Then by definition $[\![\psi]\!]_{\mathcal{K}} = pre(\mu(p))$.

Computation of $[\![\psi]\!]_{\mathcal{K}}$: $\mathbf{EU}$ and $\mathbf{EG}$

We shall first define $\mathbf{EU}$ and $\mathbf{EG}$ in terms of fixed points.

$\mathbf{EU}$ is characterized by a smallest fixed point: We first assume that no state satisfies the EU formula and then, one by one, identify those that do satisfy it after all.

By contrast, $\mathbf{EG}$ can be characterized by a largest fixed point: We first assume that all states satisfy a given EG formula and then, one by one, eliminate those that do not.

Based on this, we then derive algorithms for EG and EU.

Computation of $[\![\psi]\!]_{\mathcal{K}}$: $\mathbf{EG}$

Case 4: $\psi \equiv \mathbf{EG}\, p$, $\quad p \in AP$

Lemma 1: $[\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$ is the largest solution (w.r.t. $\subseteq$) of the equation

$$X = \mu(p) \cap \mathit{pre}(X).$$

Proof: We proceed in two steps:

1. We show that $[\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$ *is* indeed a solution of the equation, i.e.

$$[\![\mathbf{EG}\, p]\!]_{\mathcal{K}} = \mu(p) \cap \mathit{pre}([\![\mathbf{EG}\, p]\!]_{\mathcal{K}}).$$

Reminder: $[\![\mathbf{EG}\, p]\!]_{\mathcal{K}} = \{\, s \mid \exists \rho\colon \rho(0) = s \wedge \forall i \geq 0\colon \rho(i) \in \mu(p)\,\}$.

"$\Rightarrow$" Let $s \in [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$ and ρ a "witness" path. Then obviously $s \in \mu(p)$.
Moreover, $\rho(1) \in [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$ (because of ρ^1), hence $s \in \mathit{pre}([\![\mathbf{EG}\, p]\!]_{\mathcal{K}})$.

Continuation of the proof of Lemma 1:

1. "$\Leftarrow$" Let $s \in \mu(p) \cap pre([\![\mathbf{EG}\, p]\!]_{\mathcal{K}})$. Then s has a direct successor t, where a path ρ starts proving that $t \in [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$. Thus, $s\rho$ is a path witnessing that $s \in [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$.

2. We show that $[\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$ is indeed the *largest* solution, i.e., if M is a solution of the equation, then $M \subseteq [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$.

Let $M \subseteq S$ be a solution of the equation, i.e. $M = \mu(p) \cap pre(M)$, and let $s \in M$. We shall show $s \in [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$.

- Since $s \in M$, we have $s \in \mu(p)$ and $s \in pre(M)$.
- Since $s \in pre(M)$, there exist $s_1 \in M$ with $s \to s_1$.
- Repeating this argument, we can construct an infinite path $\rho = ss_1 \cdots$ in which all states are contained in $\mu(p)$. Therefore, $s \in [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$.

Lemma 2: Consider the sequence $S,\ \pi(S),\ \pi(\pi(S)),\ \ldots$, i.e. $\left(\pi^i(S)\right)_{i\geq 0}$,
where $\pi(X) := \mu(p) \cap pre(X)$.
For all $i \geq 0$ we have $\pi^i(S) \supseteq [\![\mathbf{EG}\,p]\!]_{\mathcal{K}}$.

We state the following two facts:

(1) π is *monotone*: if $X \supseteq X'$, then $\pi(X) \supseteq \pi(X')$.

(2) The sequence is *descending*: $S \supseteq \pi(S) \supseteq \pi(\pi(S)) \ldots$ (follows from (1)).

Proof of Lemma 2: (induction over *i*)
Base: $i = 0$: obvious.
Step: $i \to i + 1$:

$$\begin{aligned} \pi^{i+1}(S) &= \mu(p) \cap pre(\pi^i(S)) \\ &\supseteq \mu(p) \cap pre([\![\mathbf{EG}\,\varphi]\!]_{\mathcal{K}}) \quad \text{(i.h. and monotonicity)} \\ &= [\![\mathbf{EG}\,p]\!]_{\mathcal{K}} \end{aligned}$$

Lemma 3: There exists an index i such that $\pi^i(S) = \pi^{i+1}(S)$, and $[\![\mathbf{EG}\, p]\!]_{\mathcal{K}} = \pi^i(S)$.

Proof: Since S is finite, the descending sequence must reach a fixed point, say after i steps. Then we have $\pi^i(S) = \pi(\pi^i(S)) = \mu(p) \cap pre(\pi^i(S))$. Therefore, $\pi^i(S)$ is a solution of the equation from Lemma (1).

Because of Lemma 1, we have $\pi^i(S) \subseteq [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$.
Because of Lemma 2, we have $\pi^i(S) \supseteq [\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$.

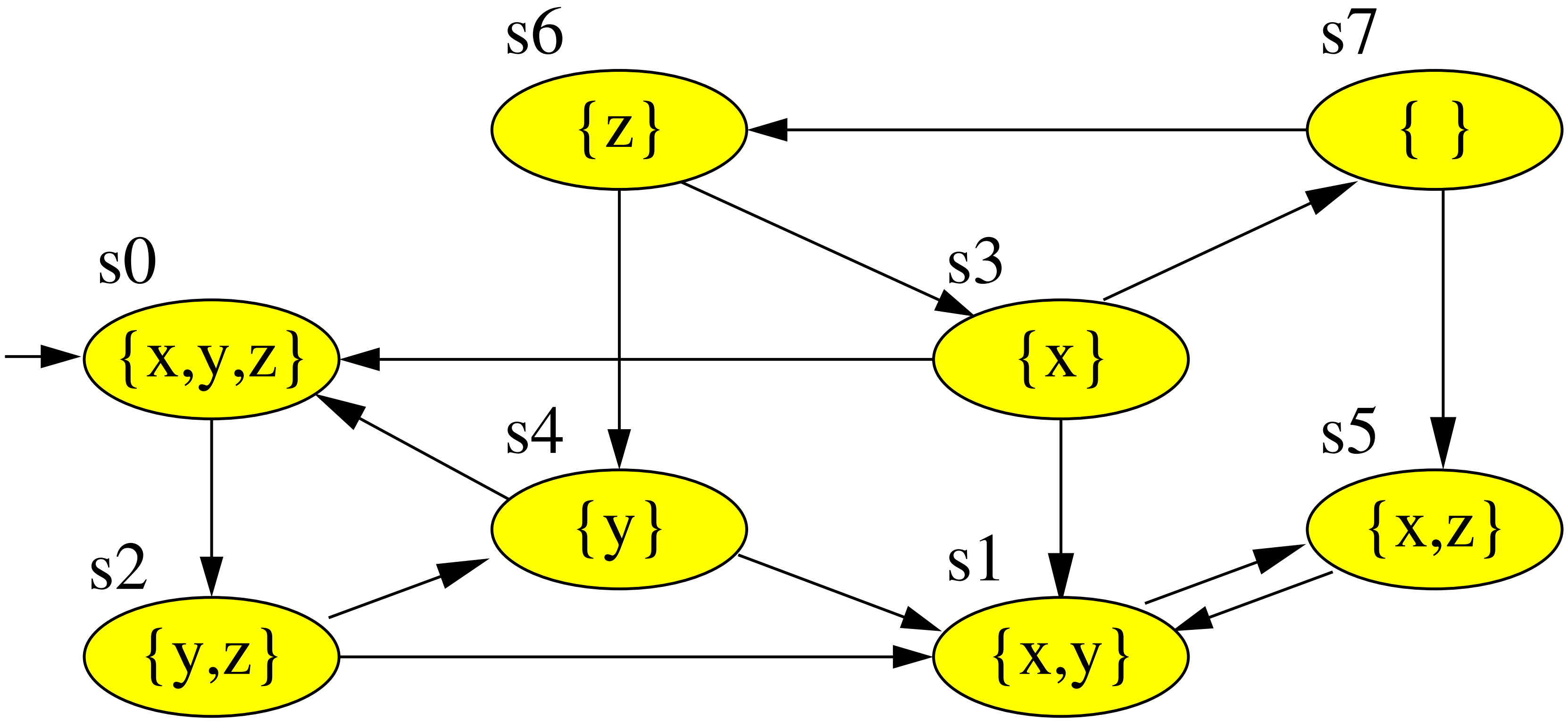
An algorithm for $\mathbf{EG}$

Lemma 3 gives us a strategy for computing $[\![\mathbf{EG}\, p]\!]_{\mathcal{K}}$: compute the sequence $S,\ \pi(S),\ \cdots$ until a fixed point is reached.

For practicality, one would start immediately with $X := \mu(p)$. Then, in each round, one eliminates those states having no successors in X.

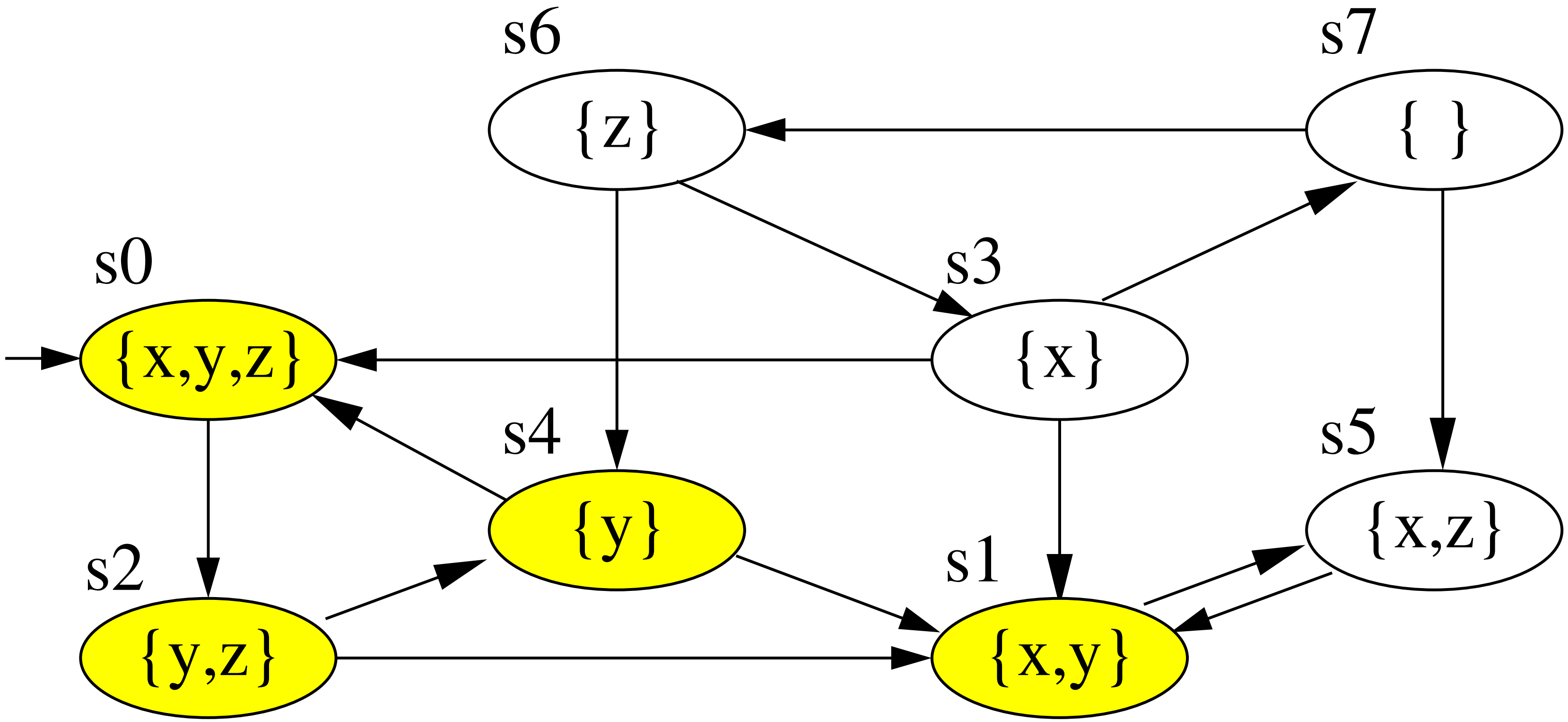
This can be efficiently implemented in $\mathcal{O}(|\mathcal{K}|)$ time (“reference counting”).

Example: Computation of $[\![\mathbf{EG}\ y]\!]_{\mathcal{K}}$ (1/4)



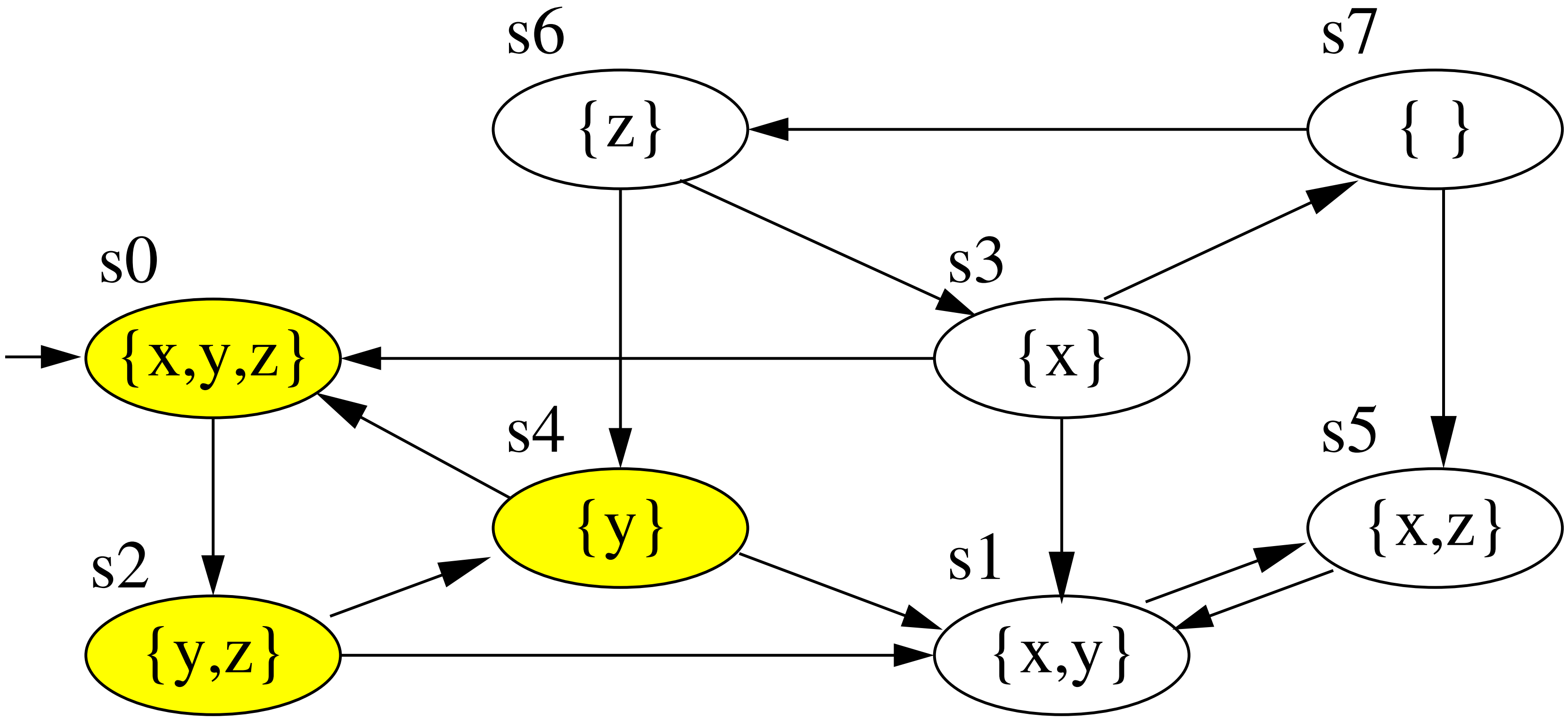
$$\pi^0(S) \;=\; S$$

Example: Computation of $[\![\mathbf{EG}\ y]\!]_{\mathcal{K}}$ (2/4)



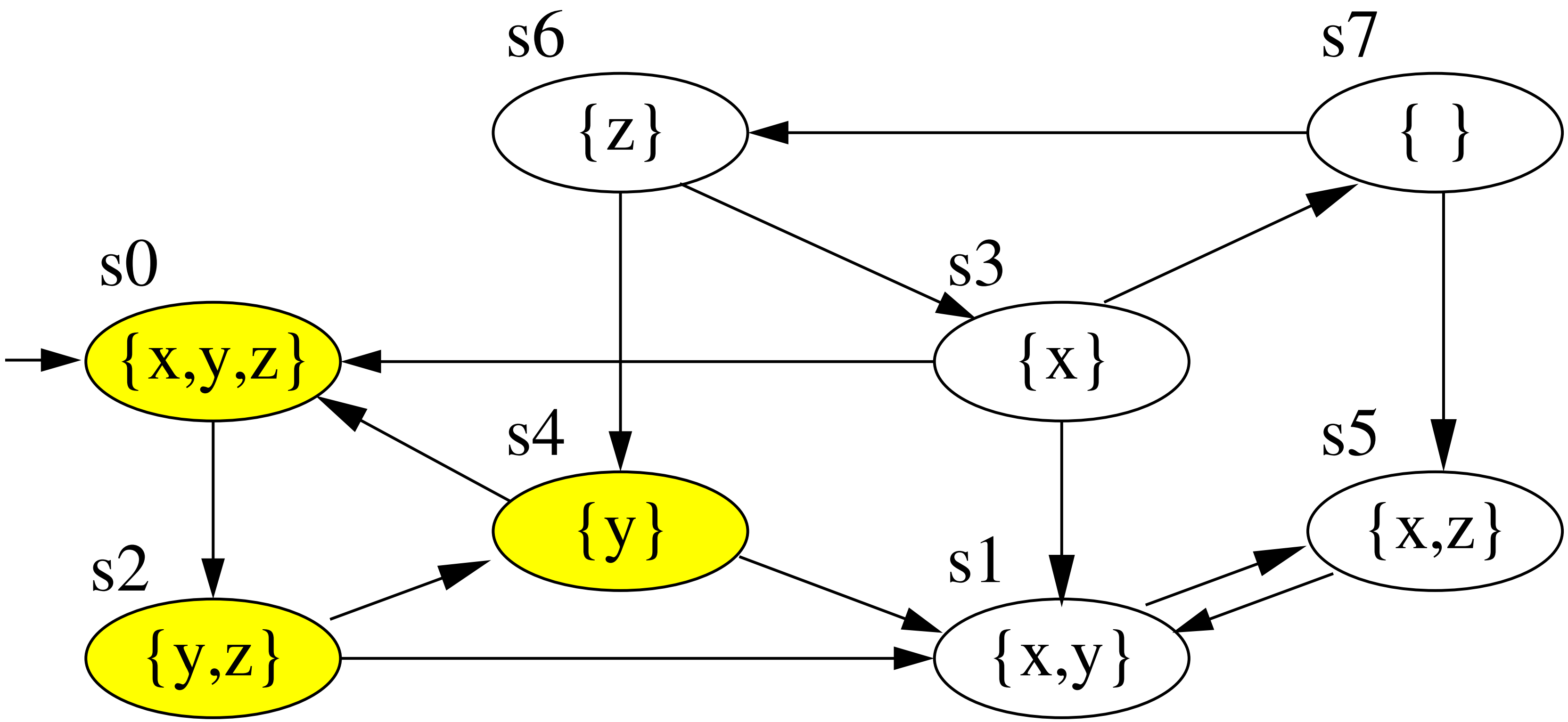
$$\pi^1(S) \;=\; \mu(y) \cap pre(S)$$

Example: Computation of $[\![\mathbf{EG}\ y]\!]_{\mathcal{K}}$ (3/4)



$$\pi^2(S) \;=\; \mu(y) \cap pre(\pi^1(S))$$

Example: Computation of $[\![\mathbf{EG}\ y]\!]_{\mathcal{K}}$ (4/4)



$\pi^3(S) \;=\; \mu(y) \cap pre(\pi^2(S)) \;=\; \pi^2(S)$: $\quad [\![\mathbf{EG}\ y]\!]_{\mathcal{K}} = \{s_0, s_2, s_4\}$

Computation of $\mathbf{EU}$

Case 5: $\psi \equiv p\,\mathbf{EU}\,q$, $\quad p, q \in AP$

Analogous to $\mathbf{EG}$ (proofs omitted):

Lemma 4: $[\![p\,\mathbf{EU}\,q]\!]_{\mathcal{K}}$ is the smallest solution (w.r.t. $\subseteq$) of the equation

$$X = \mu(q) \cup (\mu(p) \cap pre(X)).$$

Lemma 5: $[\![p\,\mathbf{EU}\,q]\!]_{\mathcal{K}}$ is the fixed point of the sequence

$\emptyset,\ \xi(\emptyset),\ \xi(\xi(\emptyset)),\ \ldots$where $\quad \xi(X) := \mu(q) \cup (\mu(p) \cap pre(X))$

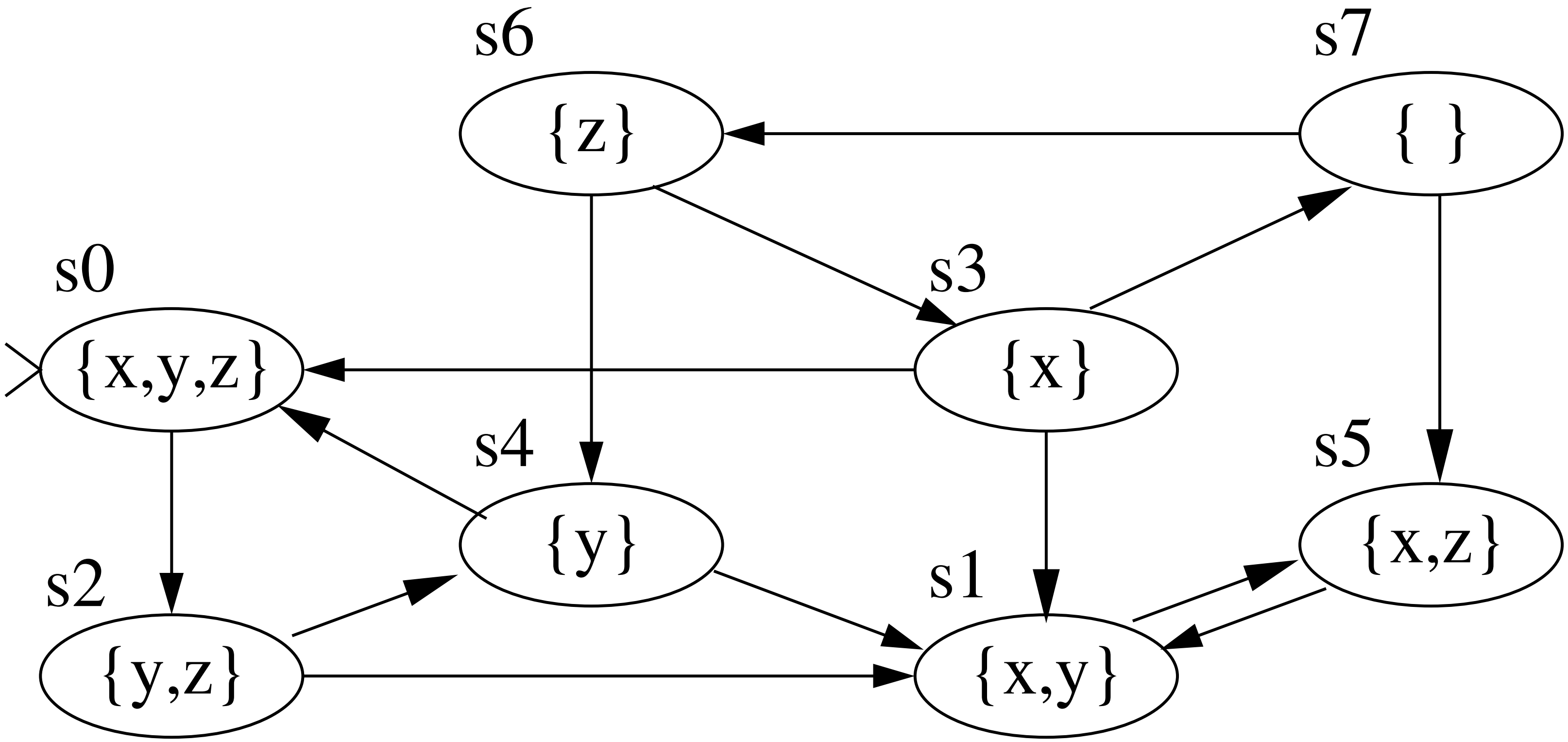
An algorithm for $\mathbf{EU}$

Lemma 5 proposes a strategy: Compute the sequence $\emptyset,\ \xi(\emptyset),\ \cdots$ until a fixed point is reached.

In practice one would start with $X := \mu(q)$. Then, in each step, one can add those direct predecessors that are in $\mu(p)$.

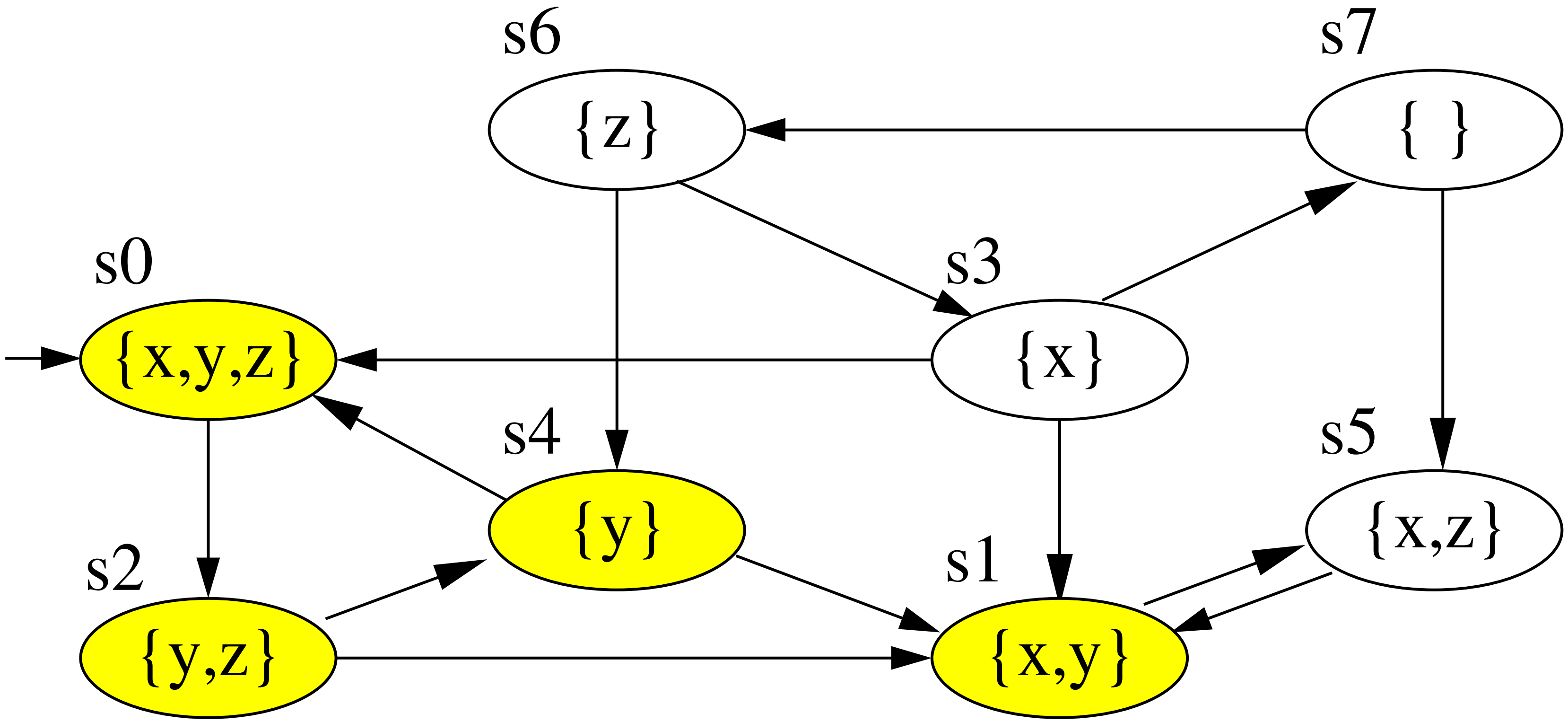
Can be done efficiently in $\mathcal{O}(|\mathcal{K}|)$ time (multiple backwards DFS).

Example: Computation of $[\![z \ \mathbf{EU} \ y]\!]_{\mathcal{K}}$ (1/4)



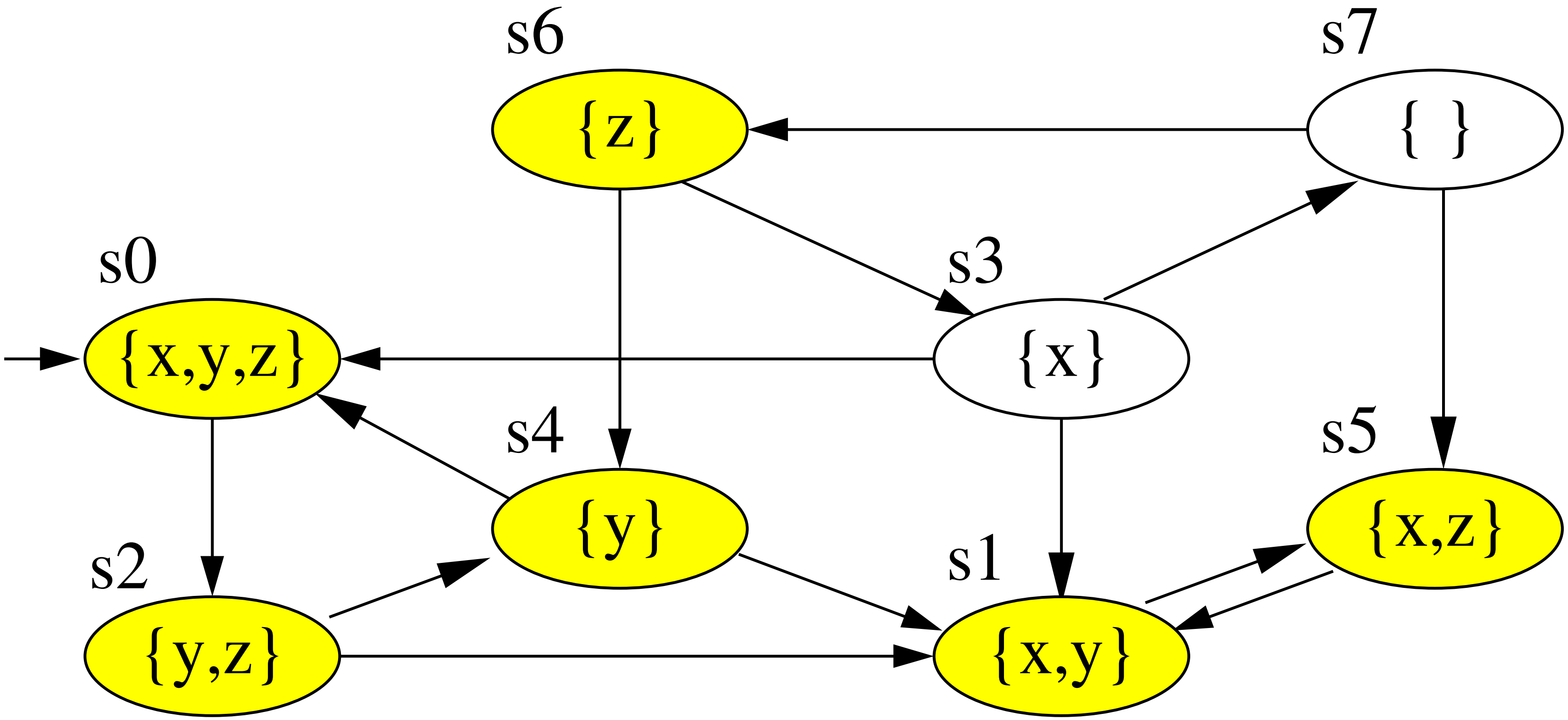
$$\xi^0(\emptyset) \;=\; \emptyset$$

Example: Computation of $[\![z \ \mathbf{EU} \ y]\!]_{\mathcal{K}}$ (2/4)



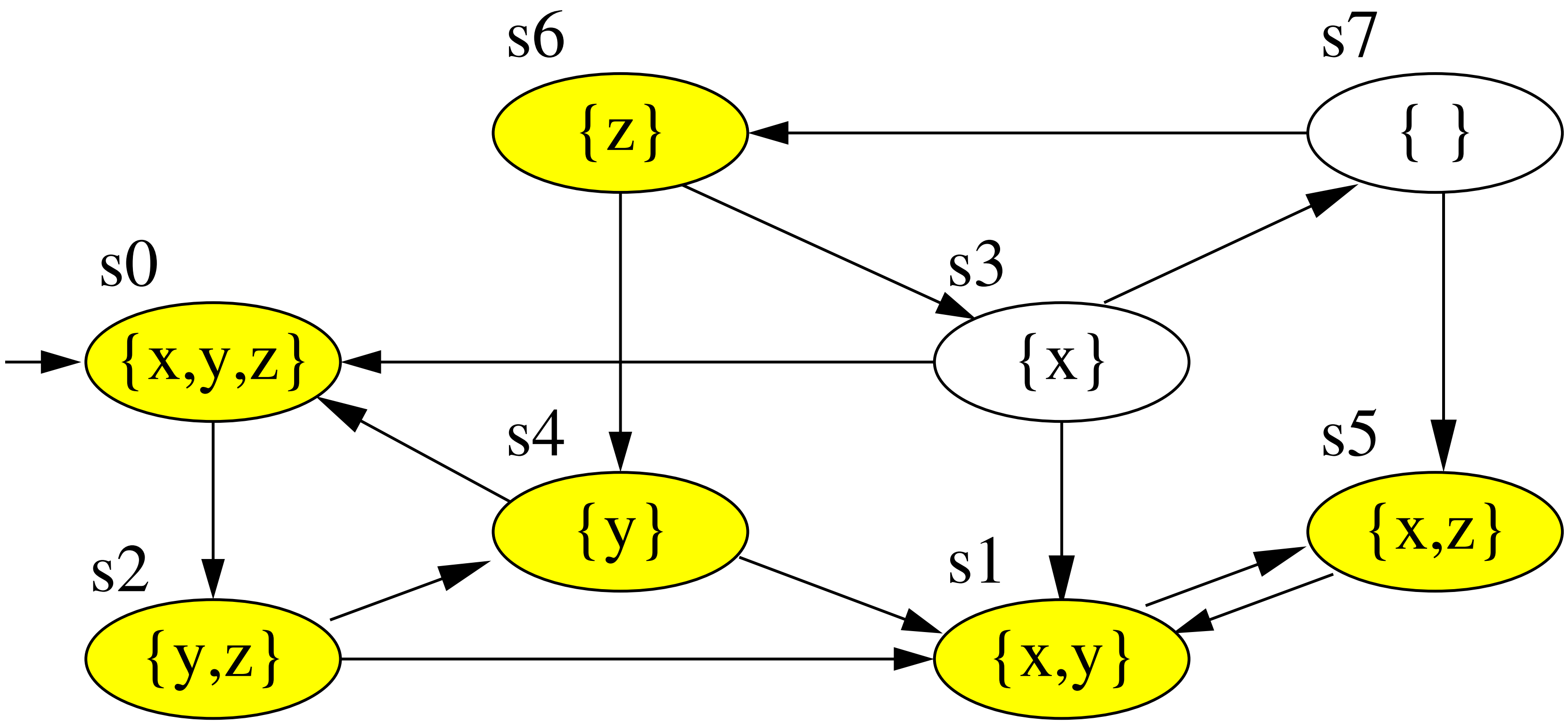
$$\xi^1(\emptyset) \;=\; \mu(y) \cup (\mu(z) \cap pre(\xi^0(\emptyset)))$$

Example: Computation of $[\![z \ \mathbf{EU} \ y]\!]_{\mathcal{K}}$ (3/4)



$$\xi^2(\emptyset) \;=\; \mu(y) \cup (\mu(z) \cap pre(\xi^1(\emptyset)))$$

Example: Computation of $[\![z\ \mathbf{EU}\ y]\!]_{\mathcal{K}}$ (4/4)



$$\xi^3(\emptyset) \;=\; \mu(y) \cup (\mu(z) \cap pre(\xi^2(\emptyset))) \;=\; \xi^2(\emptyset)$$

$[\![z\ \mathbf{EU}\ y]\!]_{\mathcal{K}} = \{s_0, s_1, s_2, s_4, s_5, s_6\}$