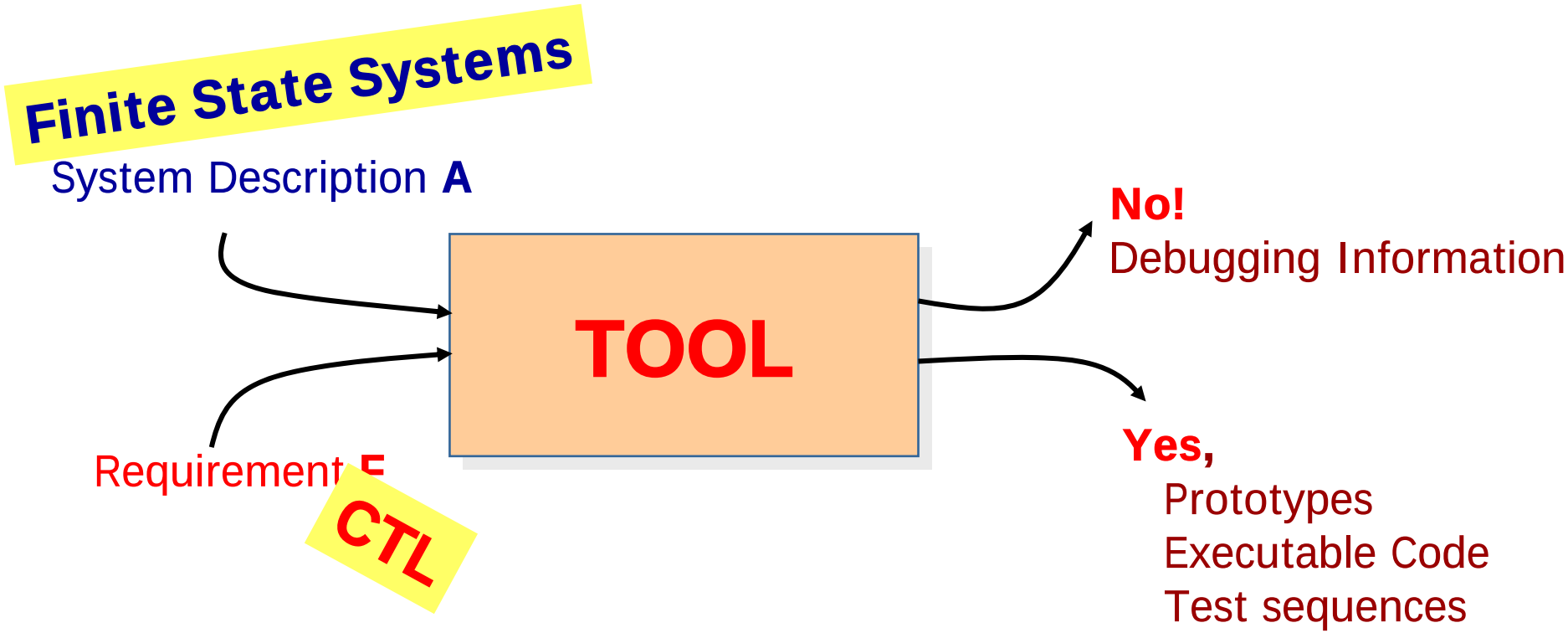
Finite State Model Checking

BRICS
Basic Research
in Computer Science

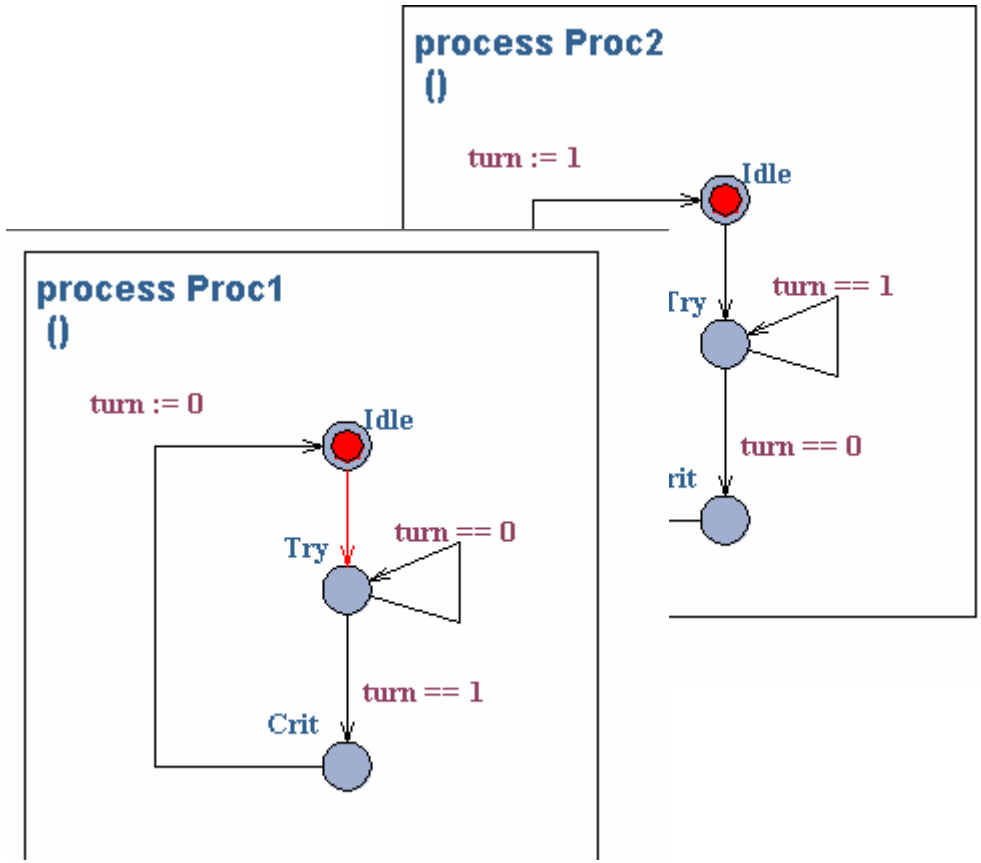
Finite State Model Checking



Tools: **visualSTATE, SPIN**, Statemate, Verilog, Formalcheck,...

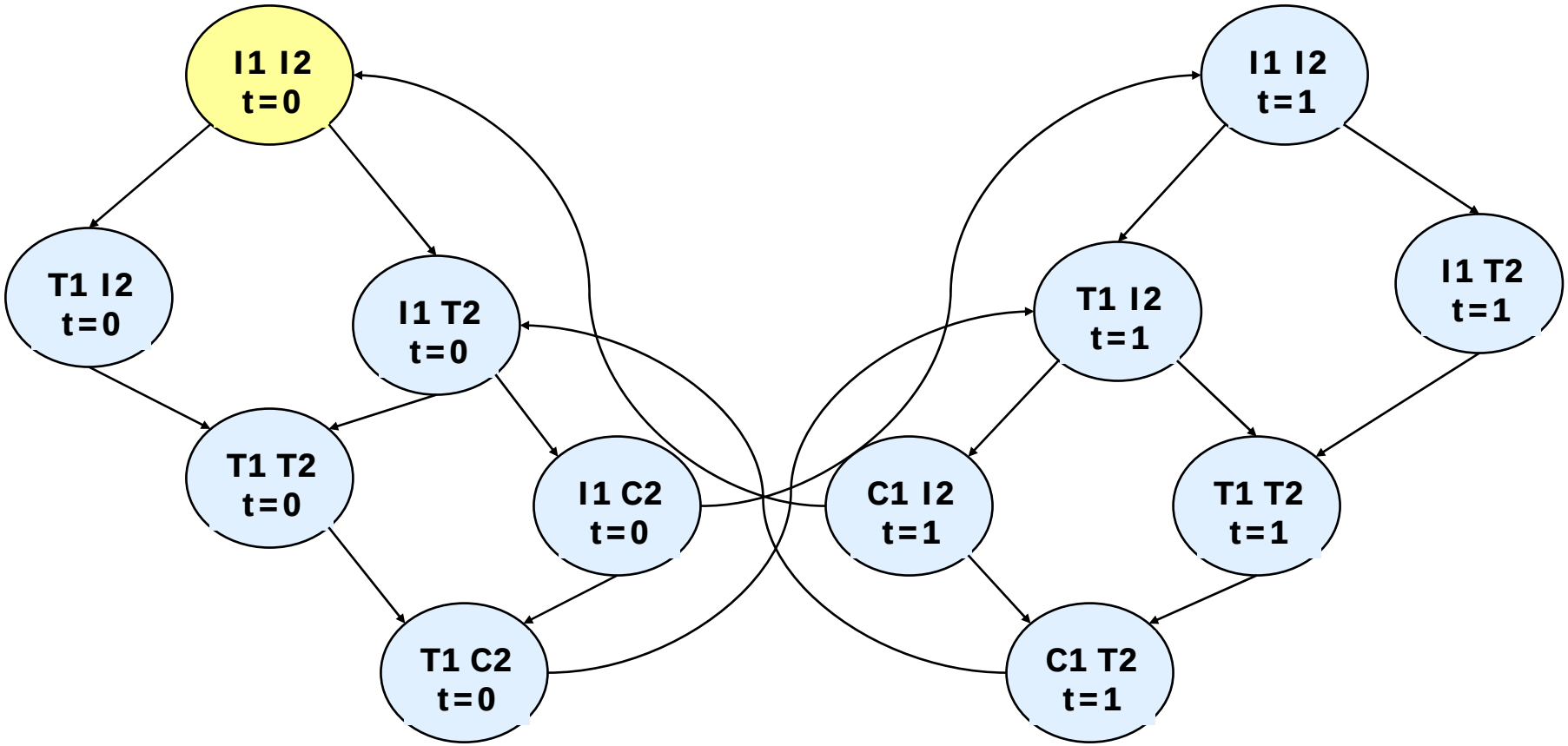
From Programs to Networks

```
P1 :: while True do
      T1 : wait(turn=1)
      C1 : turn:=0
      endwhile
||
P2 :: while True do
      T2 : wait(turn=0)
      C2 : turn:=1
      endwhile
```



Mutual Exclusion Program

From Network Models to *Kripke Structures*



CTL Models =

Kripke Structures

A CTL-model is a triple $\mathcal{M} = (S, R, Label)$ where

- S is a non-empty set of states,
- $R \subseteq S \times S$ is a total relation on S, which relates to $s \in S$ its possible successor states,
- $Label : S \longrightarrow 2^{AP}$, assigns to each state $s \in S$ the atomic propositions $Label(s)$ that are valid in s.

Computation Tree Logic, CTL

Clarke & Emerson 1980

Syntax

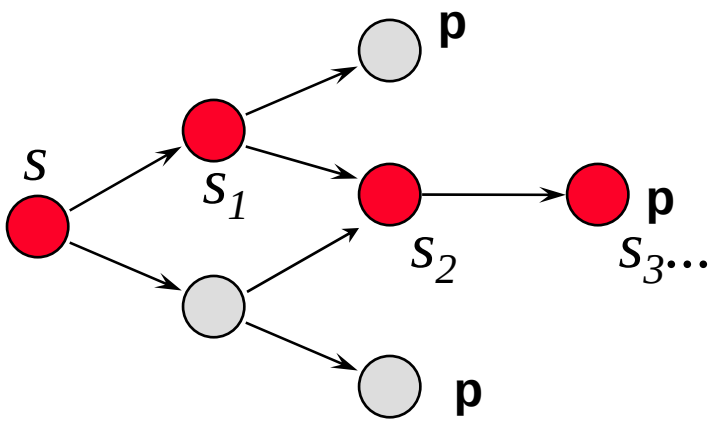
$$\phi \quad ::= \quad p \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{EX}\,\phi \mid \mathsf{E}[\phi\,\mathsf{U}\,\phi] \mid \mathsf{A}[\phi\,\mathsf{U}\,\phi].$$

- EX (pronounced "for some path next")
- E (pronounced "for some path")
- A (pronounced "for all paths") and
- U (pronounced "until").

Path

Definition 20. (Path)

A *path* is an infinite sequence of states $s_0\, s_1\, s_2 \ldots$ such that $(s_i, s_{i+1}) \in R$ for all $i \geqslant 0$.



The set of path starting in s

$$P_{\mathcal{M}}(s)$$

Formal Semantics

(satisfaction relation $\models$)

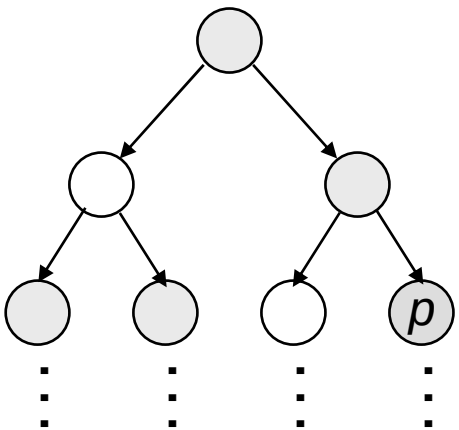
$$
\begin{array}{ll}
s \models p & \text{iff } p \in \mathit{Label}(s) \\
s \models \neg \phi & \text{iff } \neg (s \models \phi) \\
s \models \phi \vee \psi & \text{iff } (s \models \phi) \vee (s \models \psi) \\
s \models \mathsf{EX}\, \phi & \text{iff } \exists \sigma \in P_{\mathcal{M}}(s).\, \sigma[1] \models \phi \\
s \models \mathsf{E}\,[\phi\, \mathsf{U}\, \psi] & \text{iff } \exists \sigma \in P_{\mathcal{M}}(s).(\exists j \geqslant 0.\, \sigma[j] \models \psi \wedge (\forall 0 \leqslant k < j.\, \sigma[k] \models \phi)) \\
s \models \mathsf{A}\,[\phi\, \mathsf{U}\, \psi] & \text{iff } \forall \sigma \in P_{\mathcal{M}}(s).(\exists j \geqslant 0.\, \sigma[j] \models \psi \wedge (\forall 0 \leqslant k < j.\, \sigma[k] \models \phi)).
\end{array}
$$

CTL, Derived Operators

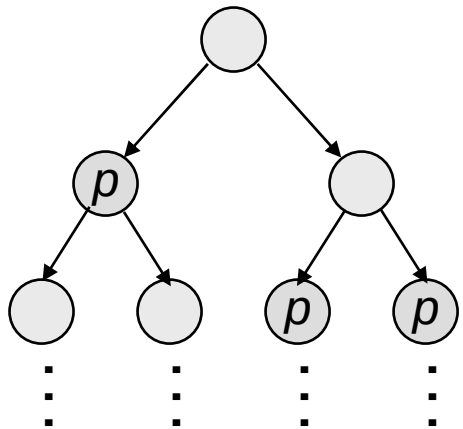
$$\mathsf{EF}\,\phi \equiv \mathsf{E}\,[\mathrm{true}\,\mathsf{U}\,\phi]$$ *possible*

$$\mathsf{AF}\,\phi \equiv \mathsf{A}\,[\mathrm{true}\,\mathsf{U}\,\phi].$$ *inevitable*

EF p



AF p



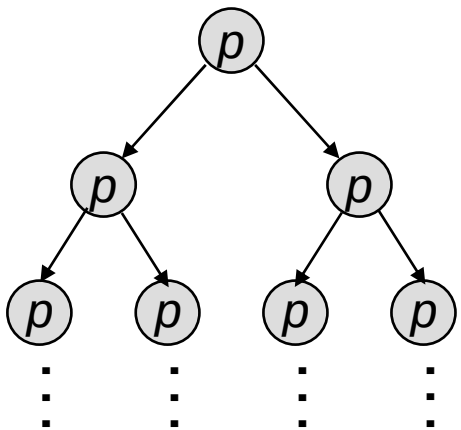
CTL, Derived Operators

$$\mathsf{EG}\,\phi \equiv \neg\,\mathsf{AF}\,\neg\,\phi$$ *potentially always*

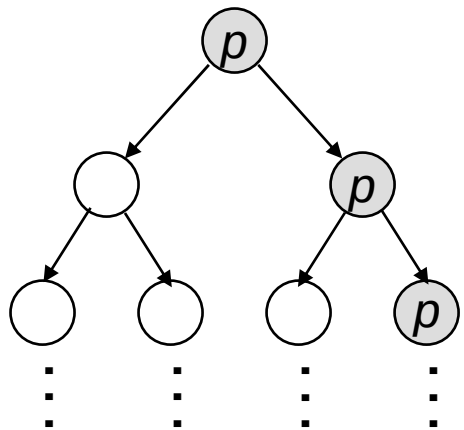
$$\mathsf{AG}\,\phi \equiv \neg\,\mathsf{EF}\,\neg\,\phi$$ *always*

$$\mathsf{AX}\,\phi \equiv \neg\,\mathsf{EX}\,\neg\,\phi.$$

AG p



EG p



Theorem

All operators are derivable from

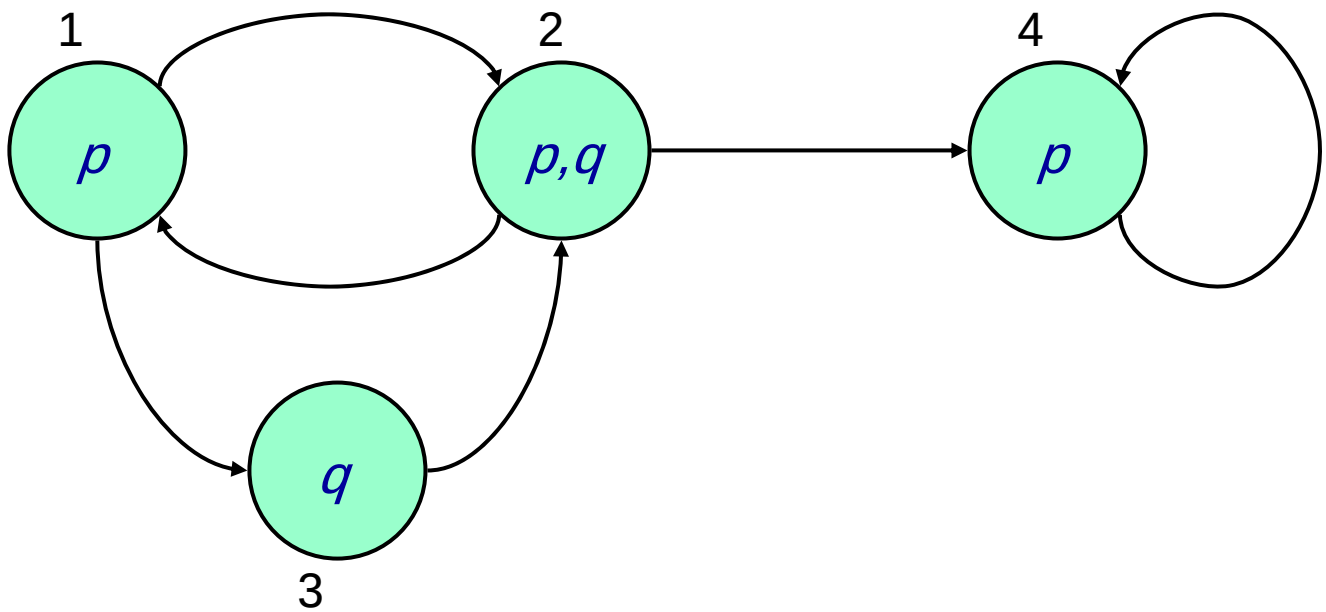
- EX f
- EG f
- E[f U g]

and boolean connectives

$$A[f\ U\ g] \equiv \neg E[\neg g U (\neg f \wedge \neg g)] \wedge \neg EG \neg g$$

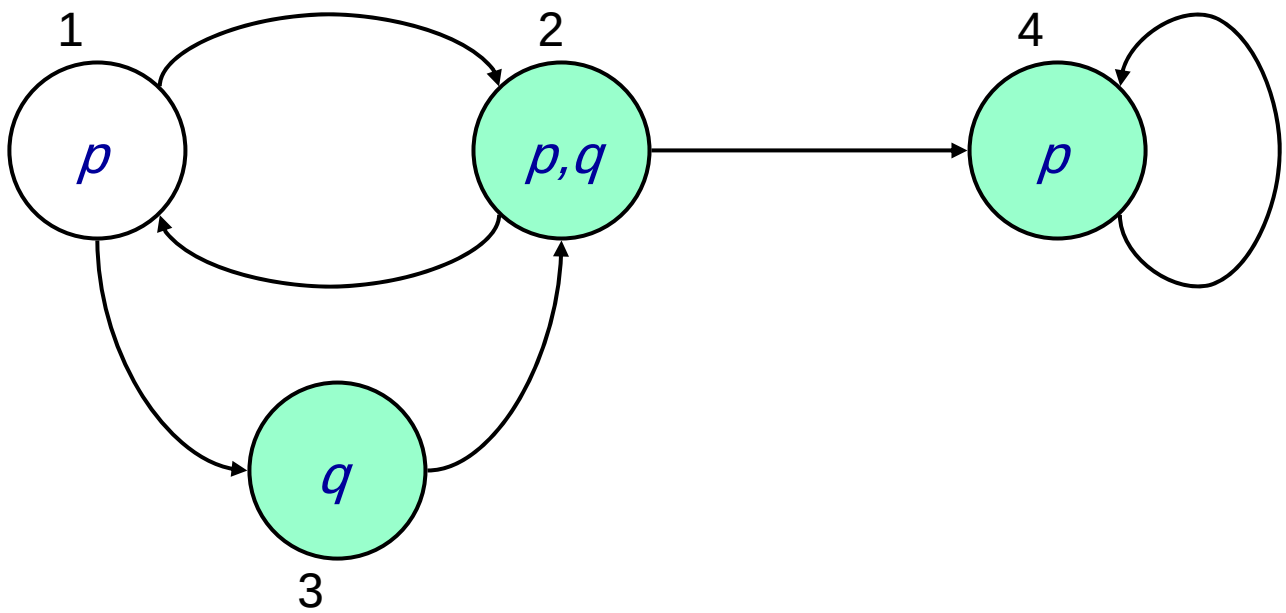
Example

EX *p*



Example

AX *p*



Properties of MUTEX example ?

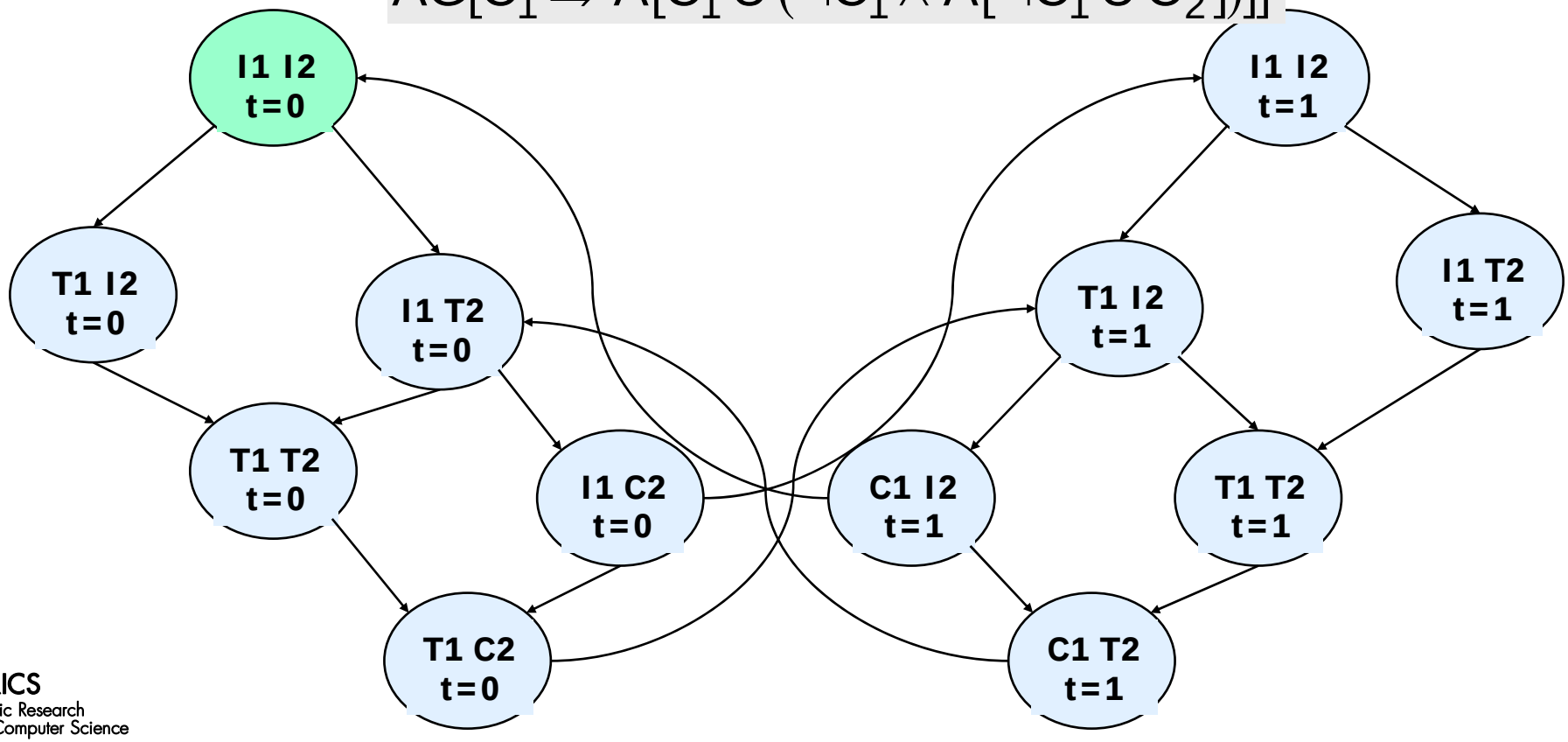
$$AG\,\neg(C_1 \wedge C_2)$$

$$AG[\,T_1 \Rightarrow AF(C_1)]$$

$$EG[\neg C_1]$$

$$AG[C_1 \Rightarrow A[C_1\,U\,(\neg C_1 \wedge A[\neg C_1\,U\,C_2])]]$$

HOW to DECIDE IN GENERAL



CTL Model Checking Algorithms

BRICS
Basic Research
in Computer Science

Fixpoint Characterizations

$$\mathsf{EF}\, p \quad \equiv \quad p \vee \mathsf{EX}\mathsf{EF}\, p$$

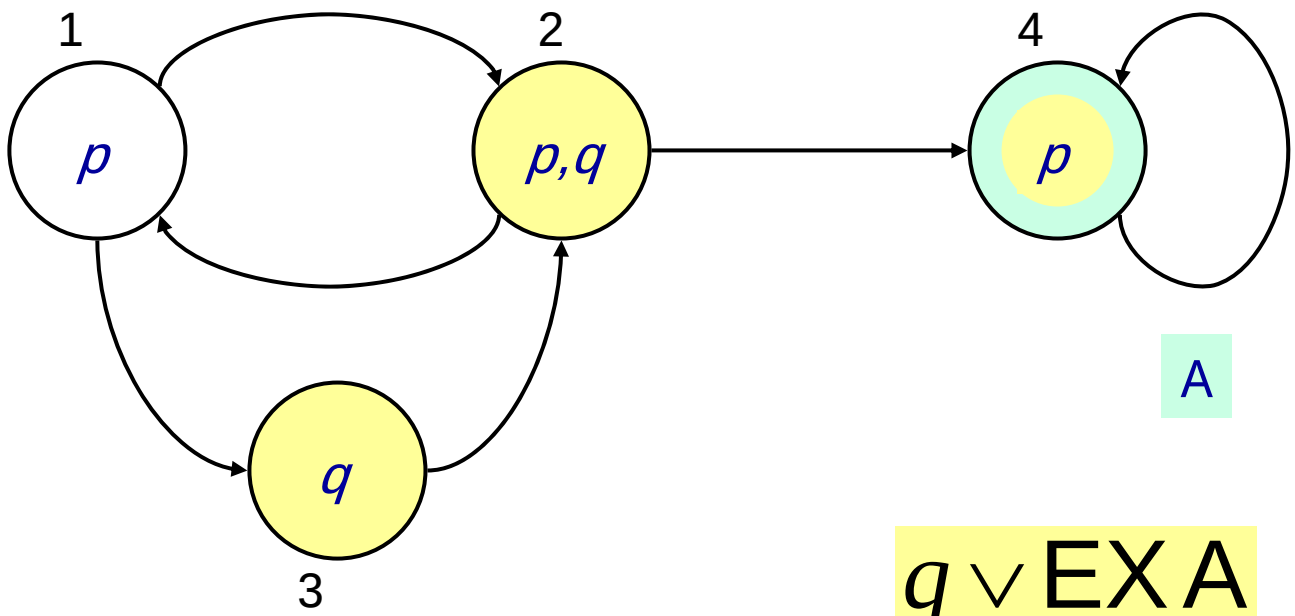
or let A **be the set of states satisfying** EF p **then**

$$\mathsf{A} \quad \equiv \quad p \vee \mathsf{EX}\,\mathsf{A}$$

in fact A **is the smallest such set (the least fixpoint)**

Example

EF q



A

$$q \vee \mathrm{EX}\, \mathrm{A}$$

Fixed points of monotonic functions

- Let τ be a function $2^S \to 2^S$
- Say τ is *monotonic* when

$$x \subseteq y \text{ implies } \tau(x) \subseteq \tau(y)$$

- Fixed point of τ is y such that

$$\tau(y) = y$$

- If τ monotonic, then it has
  - least fixed point $\mu y.\ \tau(y)$
  - greatest fixed point $\nu y.\ \tau(y)$

Iteratively computing fixed points

- Suppose S is finite
  - The least fixed point $\mu y.\ \tau(y)$ is the limit of

$$\text{false} \subseteq \tau(\text{false}) \subseteq \tau(\tau(\text{false})) \subseteq \cdots$$

  - The greatest fixed point $\nu y.\ \tau(y)$ is the limit of

$$\text{true} \supseteq \tau(\text{true}) \supseteq \tau(\tau(\text{true})) \supseteq \cdots$$

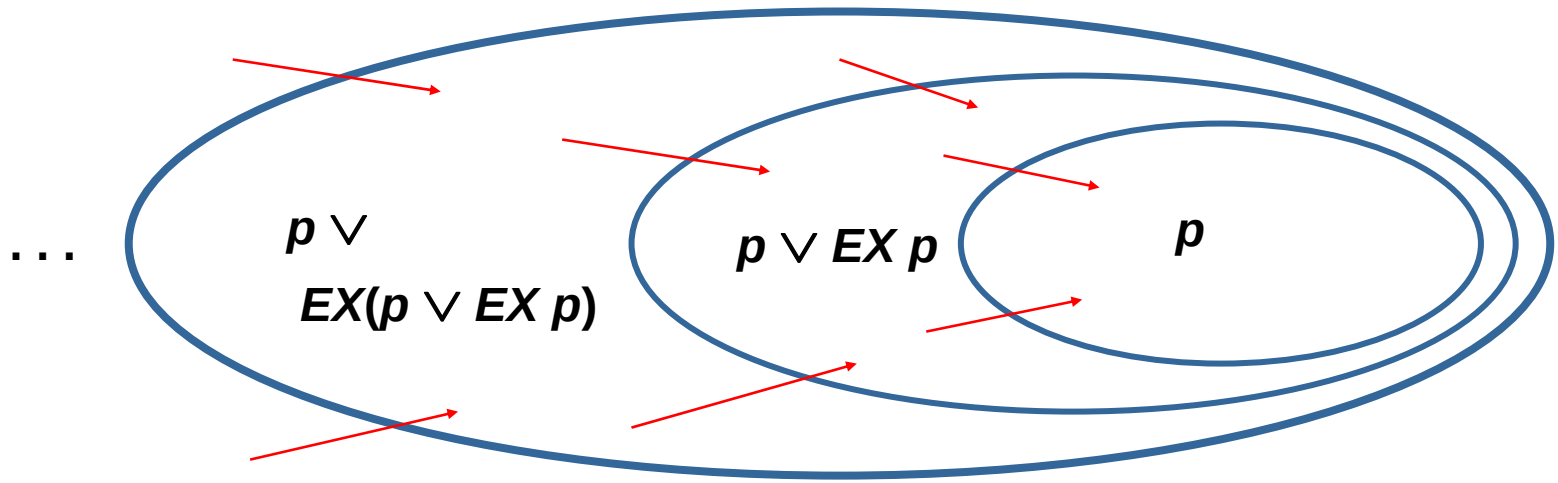
Note, since S is finite, convergence is finite

Example: *EF p*

- *EF p* is characterized by

$$EF\ \ p = \mu y.(p \vee EX\ \ y)$$

- Thus, it is the limit of the increasing series...

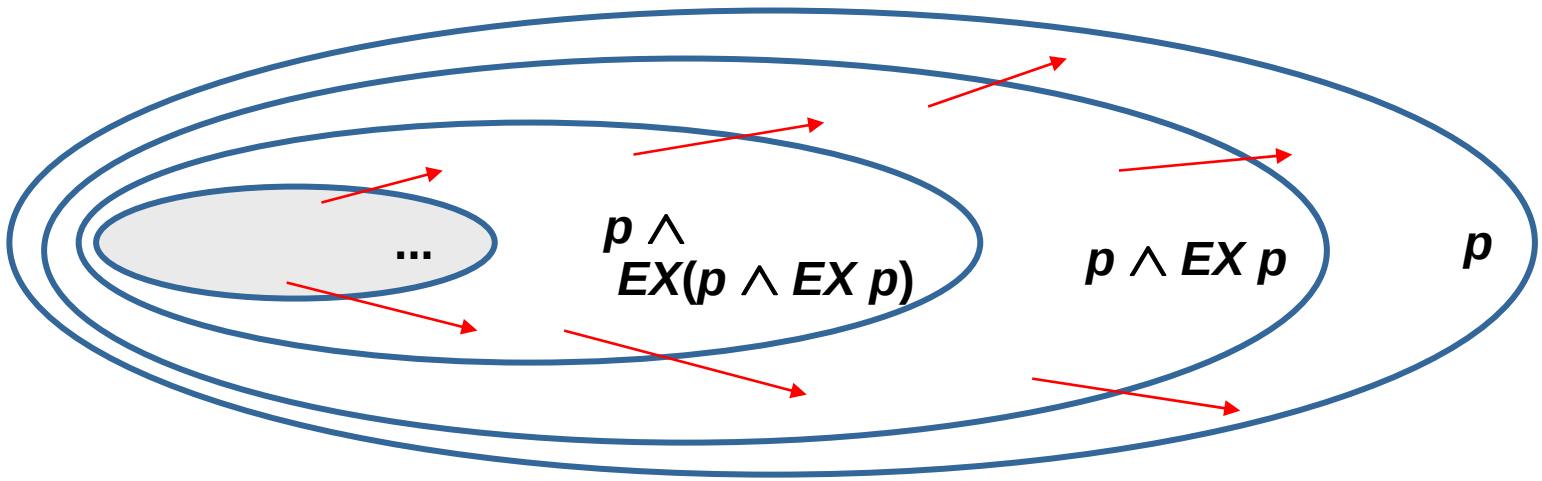


Example: *EG p*

- *EG p* is characterized by

$$EG\ p = \nu y.\ (p \wedge EX\ y)$$

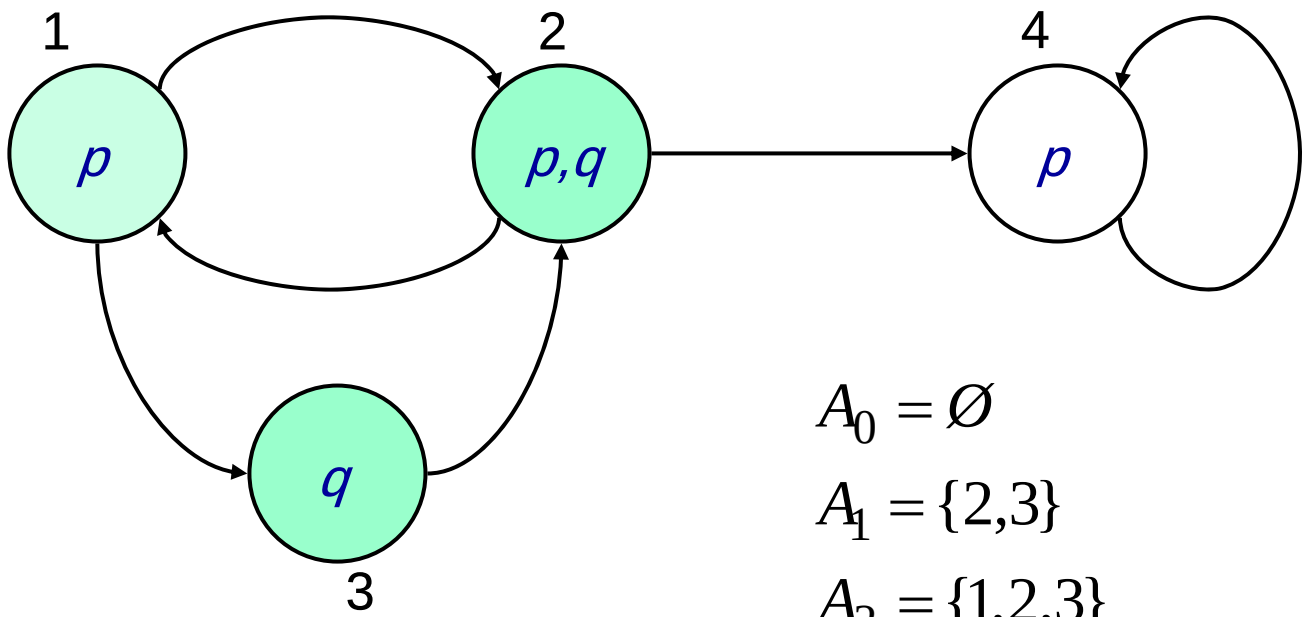
- Thus, it is the limit of the decreasing series...



Example, continued

EF *q*

$$EF\ q = \mu y.(q \vee EX\ y)$$



$$A_0 = \emptyset$$
$$A_1 = \{2,3\}$$
$$A_2 = \{1,2,3\}$$
$$A_3 = \{1,2,3\}$$

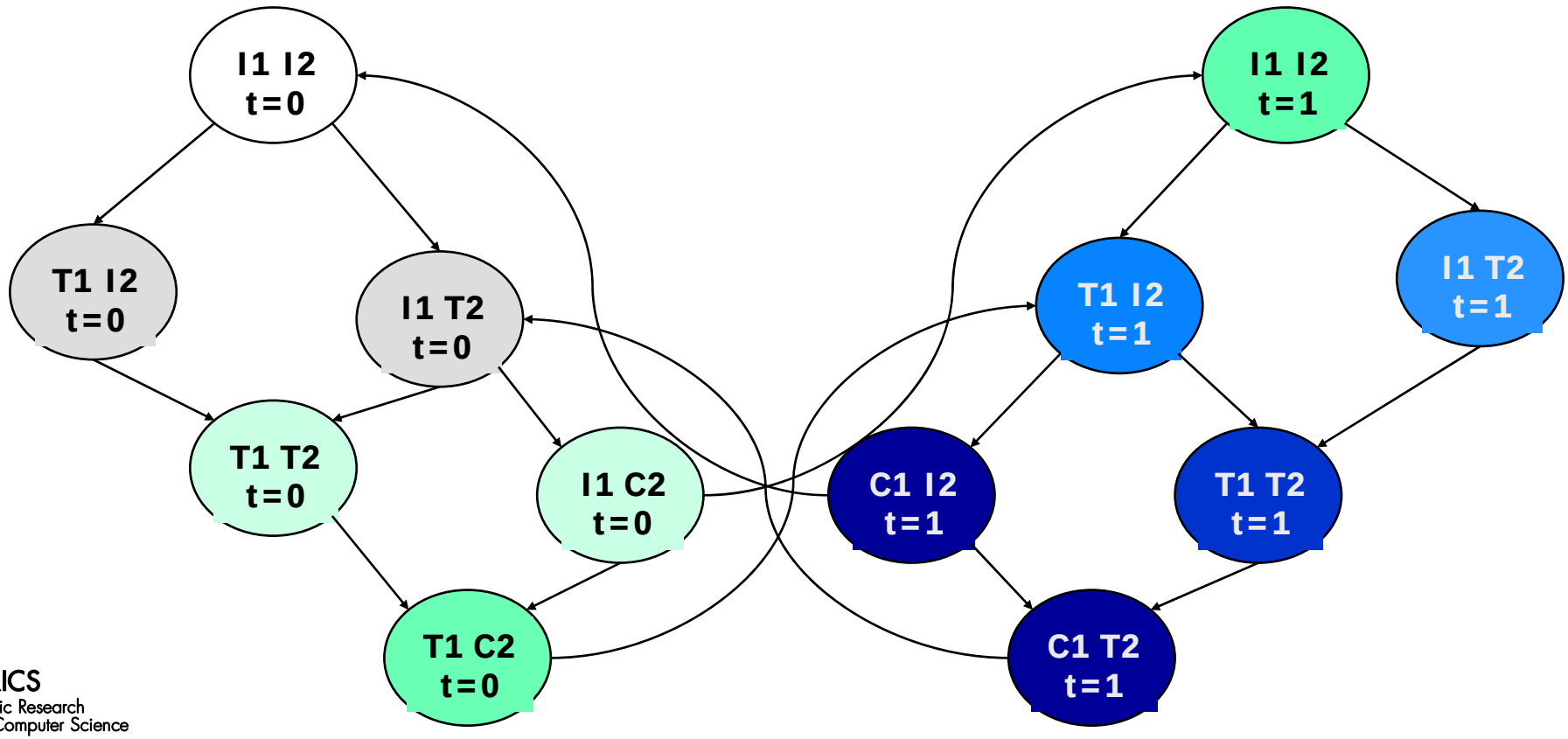
Remaining operators

$$
\begin{aligned}
AF\ p &= \mu y.(p \vee AX\ y) \\
AG\ p &= \nu y.(p \wedge AX\ y) \\
E(p\,U\,q) &= \mu y.(q \vee (p \wedge EX\ y)) \\
A(p\,U\,q) &= \mu y.(q \vee (p \wedge AX\ y))
\end{aligned}
$$

Properties of MUTEX example ?

$$AG[\, T_1 \Rightarrow AF(C_1)]$$

$$AF(C_1)]$$



$$
\begin{array}{l}
\textbf{function } \mathit{Sat}\,(\phi : \mathit{Formula}) : \textbf{set of } \mathit{State}; \\
(*\ \text{precondition: true}\ *) \\
\textbf{begin} \\
\qquad \textbf{if } \phi = \text{true} \longrightarrow \textbf{return } S \\
\qquad [] \; \phi = \text{false} \longrightarrow \textbf{return } \varnothing \\
\qquad [] \; \phi \in AP \longrightarrow \textbf{return } \{\, s \mid \phi \in \mathit{Label}(s) \,\} \\
\qquad [] \; \phi = \neg\, \phi_1 \longrightarrow \textbf{return } S - \mathit{Sat}(\phi_1) \\
\qquad [] \; \phi = \phi_1 \;\vee\; \phi_2 \longrightarrow \textbf{return } (\mathit{Sat}(\phi_1) \cup \mathit{Sat}(\phi_2)) \\
\qquad [] \; \phi = \mathsf{EX}\,\phi_1 \longrightarrow \textbf{return } \{\, s \in S \mid (s, s') \in R \;\wedge\; s' \in \mathit{Sat}(\phi_1) \,\} \\
\qquad [] \; \phi = \mathsf{E}\,[\phi_1\,\mathsf{U}\,\phi_2] \longrightarrow \textbf{return } \mathit{Sat}_{EU}(\phi_1, \phi_2) \\
\qquad [] \; \phi = \mathsf{A}\,[\phi_1\,\mathsf{U}\,\phi_2] \longrightarrow \textbf{return } \mathit{Sat}_{AU}(\phi_1, \phi_2) \\
\qquad \textbf{fi} \\
(*\ \text{postcondition: } \mathit{Sat}(\phi) = \{\, s \mid \mathcal{M}, s \models \phi \,\}\ *) \\
\textbf{end}
\end{array}
$$

function $Sat_{EU}\,(\phi, \psi : Formula)$: **set of** $State$;
(* precondition: true *)
begin var Q, Q' : **set of** $State$;
    $Q, Q' := Sat(\psi), \varnothing$;
    do $Q \neq Q' \longrightarrow$
        $Q' := Q$;
        $Q := Q \cup (\{\, s \mid \exists\, s' \in Q.\, (s, s') \in R \,\} \cap Sat(\phi))$
    od;
    return Q
(* postcondition: $Sat_{EU}(\phi, \psi) = \{\, s \in S \mid \mathcal{M}, s \models \mathsf{E}\,[\phi\,\mathsf{U}\,\psi] \,\}$ *)
end

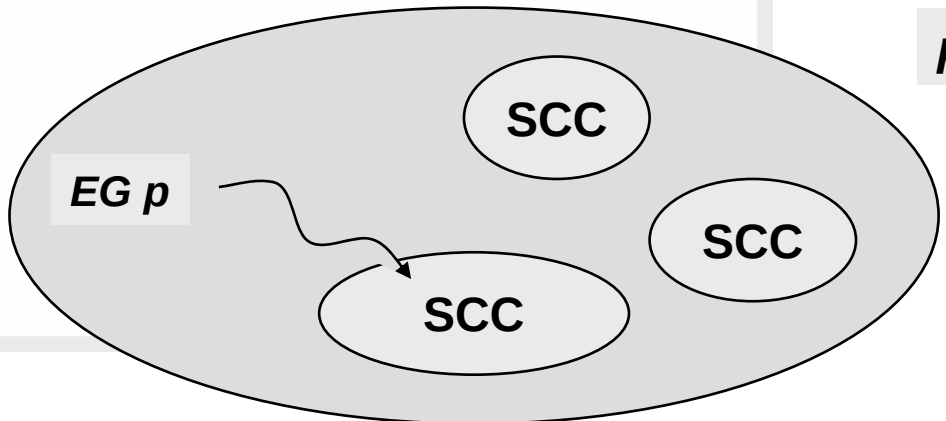
Table 3.4: Labelling procedure for $\mathsf{E}\,[\phi\,\mathsf{U}\,\psi]$

function $Sat_{AU}(\phi, \psi : Formula)$: **set of** $State$;
(* precondition: true *)
begin var Q, Q' : **set of** $State$;
$Q, Q' := Sat(\psi), \varnothing$;
do $Q \neq Q' \longrightarrow$
$Q' := Q$;
$Q := Q \cup (\{s \mid \forall s'.(s, s') \in R \Rightarrow s' \in Q\} \cap Sat(\phi))$
od;
return Q
(* postcondition: $Sat_{AU}(\phi, \psi) = \{ s \in S \mid \mathcal{M}, s \models \mathsf{A}[\phi \,\mathsf{U}\, \psi] \}$ *)
end

Table 3.5: Labelling procedure for $\mathsf{A}[\phi \,\mathsf{U}\, \psi]$

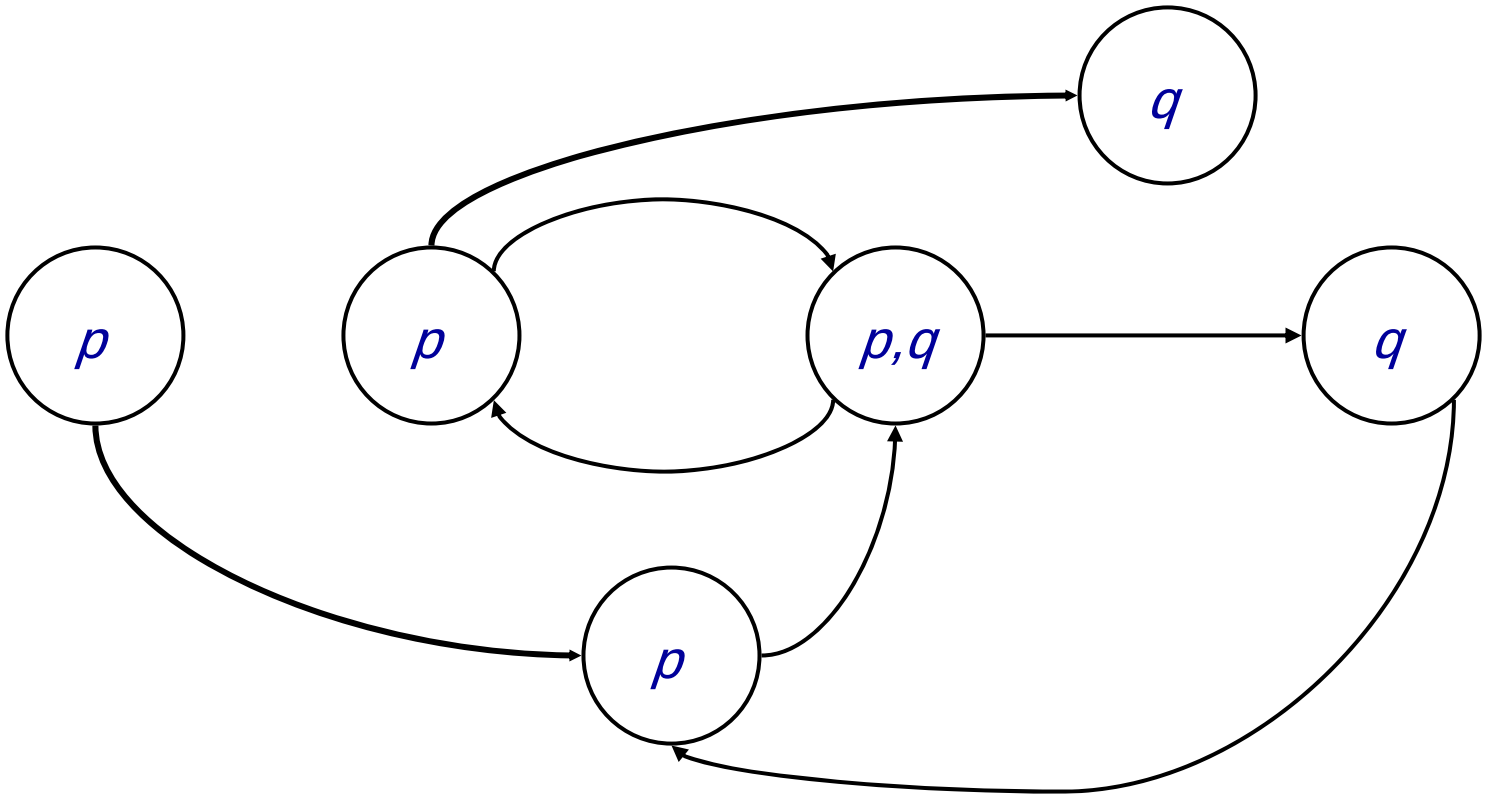
More Efficient Check

```
procedure CheckEG(f1)
begin
    S' := { s | f1 ∈ label(s) };
    SCC := { C | C is a nontrivial SCC of S' };
    T := ⋃C∈SCC { s | s ∈ C };
    for every s ∈ T do label(s) := label(s) ∪ { EG f1 };
    while T ≠ ∅ do
    begin
        choose s ∈ T;
        T := T \ {s};
        for every t such that t ∈ S' and R(t, s) do
        begin
            if EG f1 ∉ label(t) do
            begin
                label(t) := label(t) ∪ { EG f1 };
                T := T ∪ {t}
            end
    end
    end
end
```



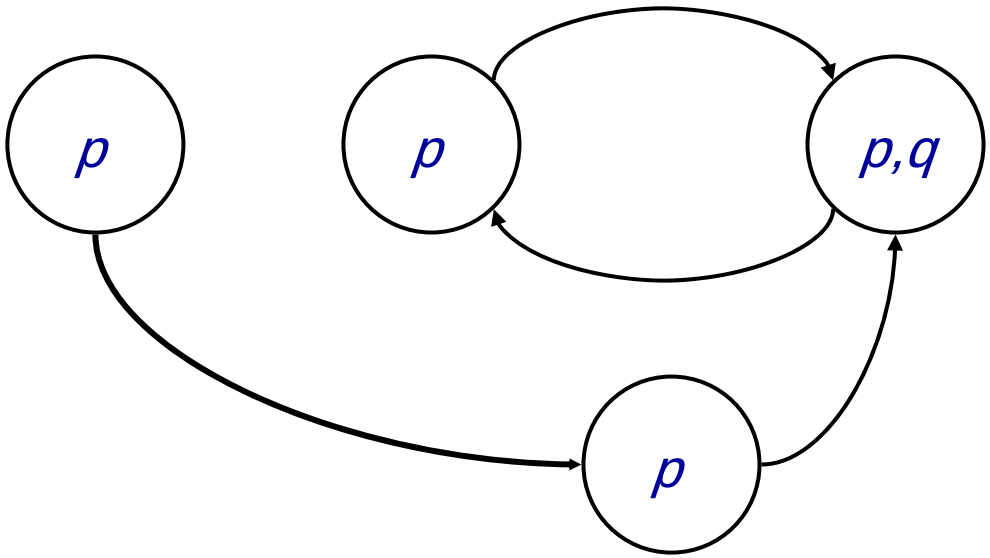
Example

EG *p*



Example

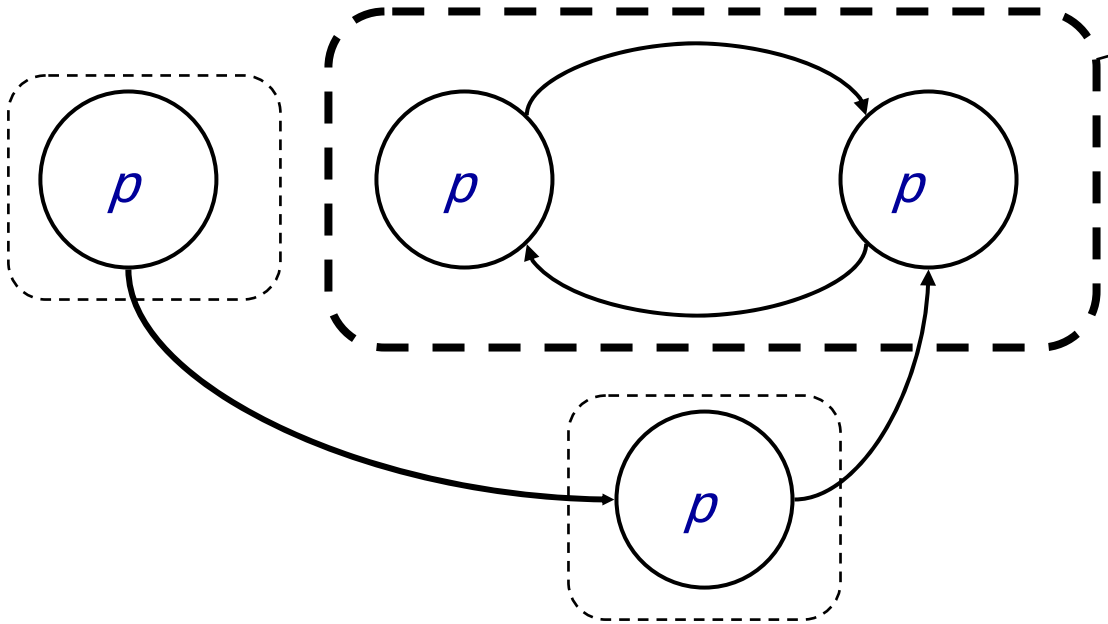
EG *p*



Reduced Model

Example

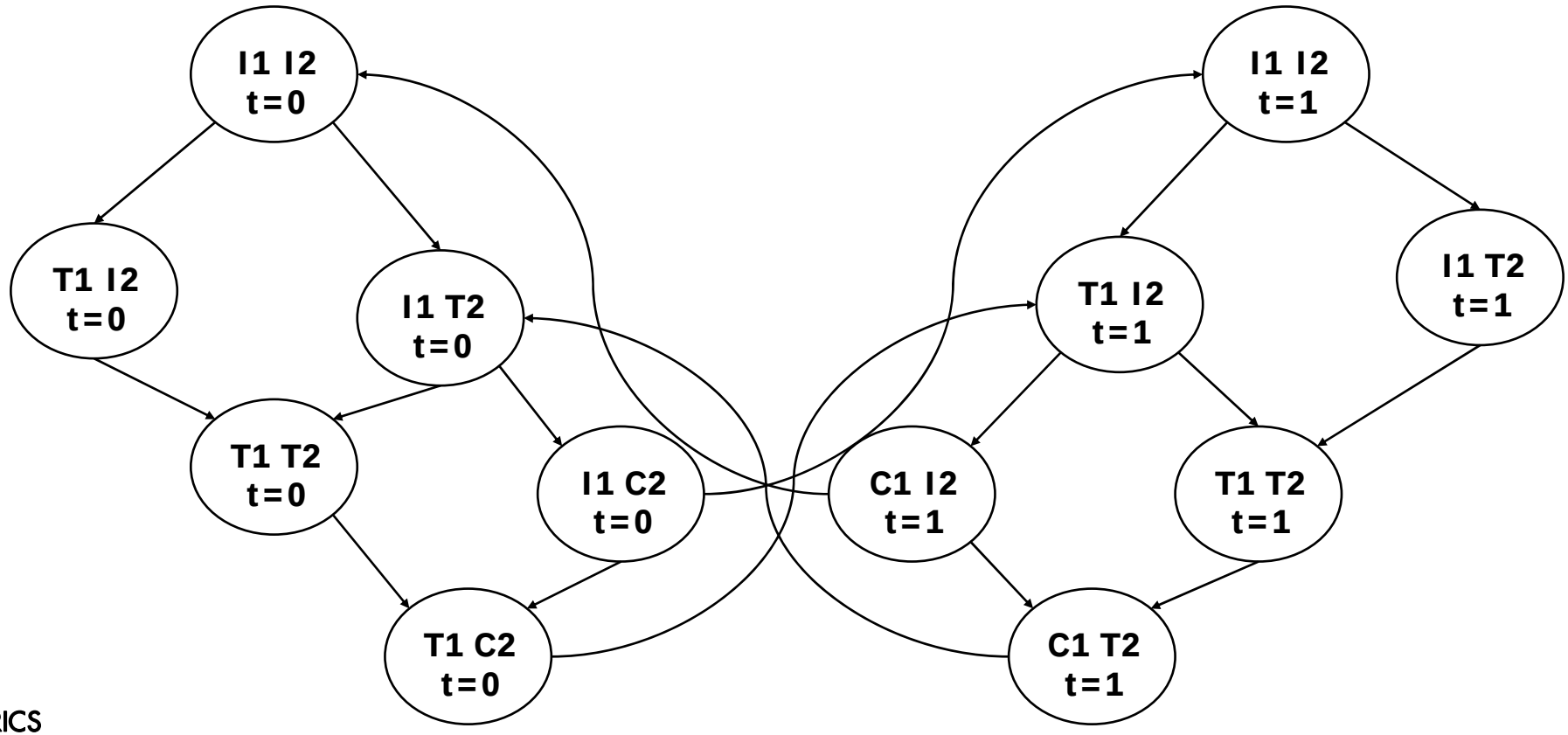
EG *p*



Non trivial Strongly Connected Component

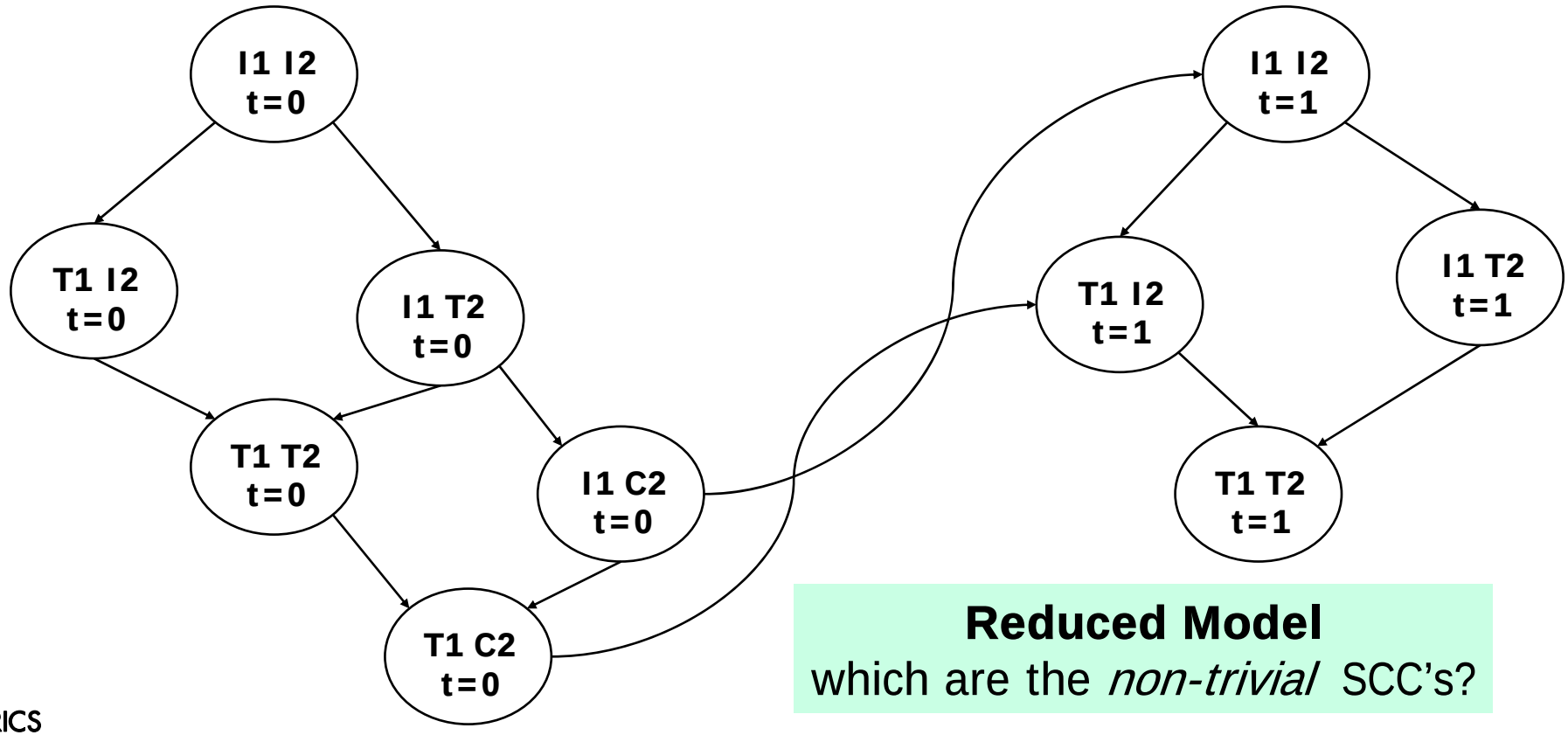
Properties of MUTEX example ?

$EG[\neg C_1]$



Properties of MUTEX example ?

$EG[\neg C_1]$



Reduced Model

which are the *non-trivial* SCC's?

Complexity

The worst-case time complexity of checking whether system-model sys satisfies the CTL*-formula* ϕ *is* $\mathcal{O}(|\ S_{sys}\ |^2 \times |\ \phi\ |)$

However S_{sys} may be EXPONENTIAL in number of parallel components!

--

FIXPOINT COMPUTATIONS may be carried out using

ROBDD's

(Reduced Ordered Binary Decision Diagrams)

Bryant, 86