Introduction to Software Verification

Orna Grumberg

Lectures Material
winter 2017-18

Lecture 4

Model Checking

Automated formal verification:

A different approach to formal verification

Formal Verification

Given

- a model of a (hardware or software) system and
- a formal specification

does the system model satisfy the specification?

Not decidable!

To enable automation, we restrict the problem to a decidable one:

- **Finite-state** reactive systems
- **Propositional** temporal logics

Properties in Propositional Temporal Logic - Examples

- **mutual exclusion:**
  always $\neg(cs_1 \wedge cs_2)$

- **non starvation:**
  always (request $\Rightarrow$ **eventually** granted)

- **communication protocols:**
  ($\neg$ get-message) **until** send-message

Finite State Systems - Examples

- Hardware designs
- Controllers (elevator, traffic-light)
- Communication protocols (when ignoring the message content)
- High level (abstracted) description of non finite state systems

Model Checking [CE81,QS82]

An efficient procedure that receives:

- A finite-state model describing a system
- A temporal logic formula describing a property

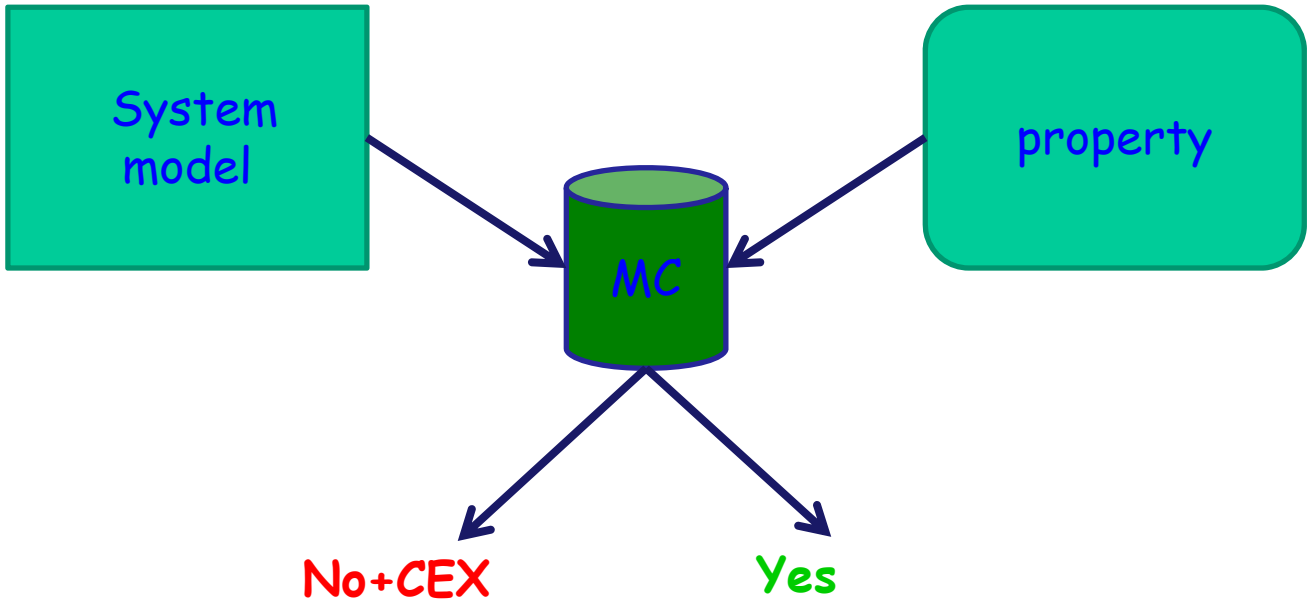
It returns

yes, if the system has the property

no + Counterexample, otherwise

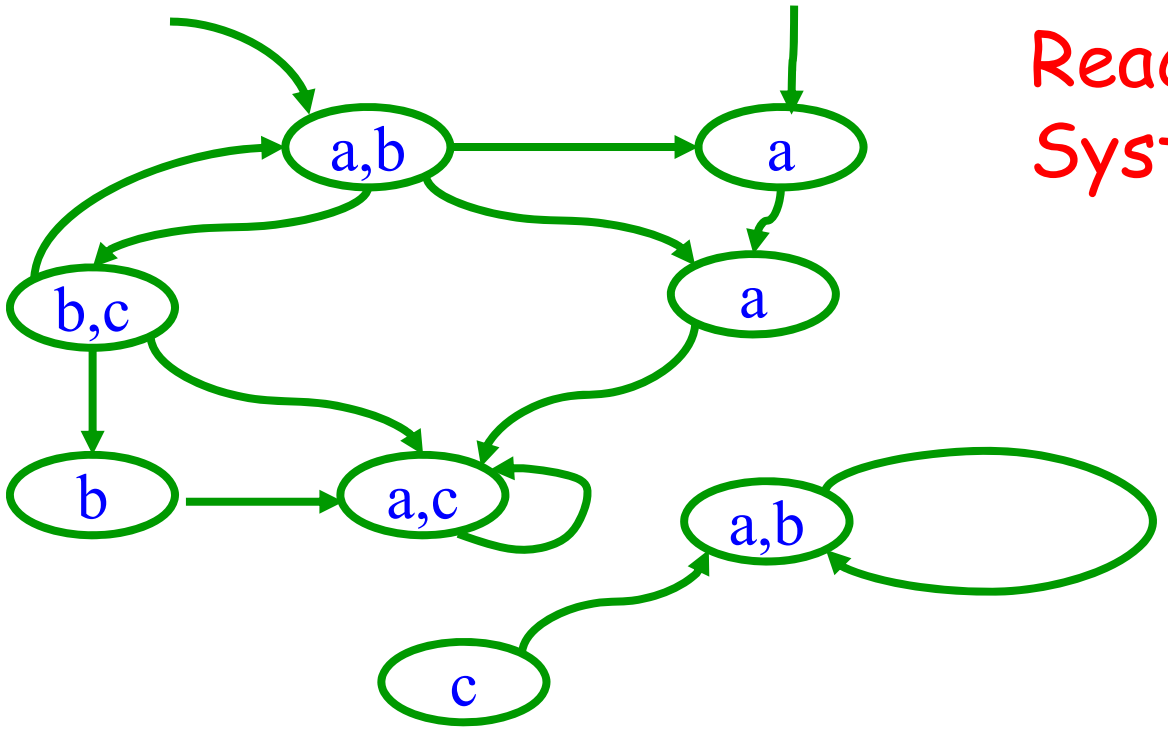
Model Checking

- Given a system and a specification, does the system satisfy the specification.



Model of a system

Kripke structure / transition system



Reactive Systems

Labeled by **atomic propositions AP**
(critical section, variable value…)

Kripke Structure $M=(S,R,L,S_0)$

Given AP – finite set of atomic proposition

- S – (finite) set of states
- $R \subseteq S \times S$ – total transition relation

  For every $s \in S$ there exists $s' \in S$ such that $(s,s') \in R$.

  Totality means that every path is infinite
- $L: S \rightarrow 2^{AP}$ - labeling function that associates every state with the atomic propositions true in that state
- $S_0 \subseteq S$ – set of initial states (optional)

$\pi = s_0, s_1, \ldots$ is a path in M from a state s if

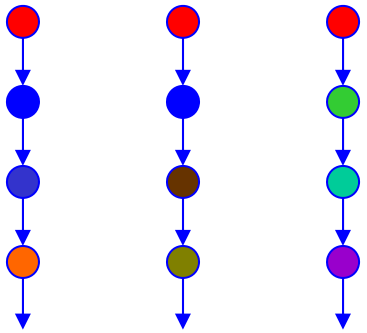
- $s = s_0$ and
- $R(s_i, s_{i+1})$ for every $i \in N$

π^i – the suffix of π starting at s_i

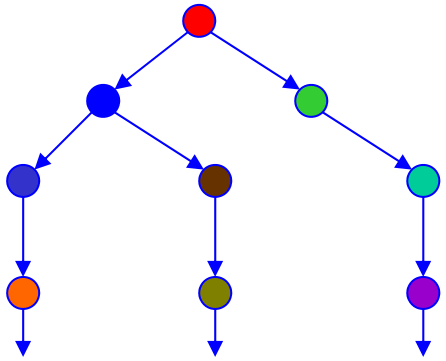
Temporal Logics

Express properties of event orderings in time

- **Linear** Time
  - Every moment has a unique successor
  - Infinite sequences (words)
  - Linear Time Temporal Logic (LTL)



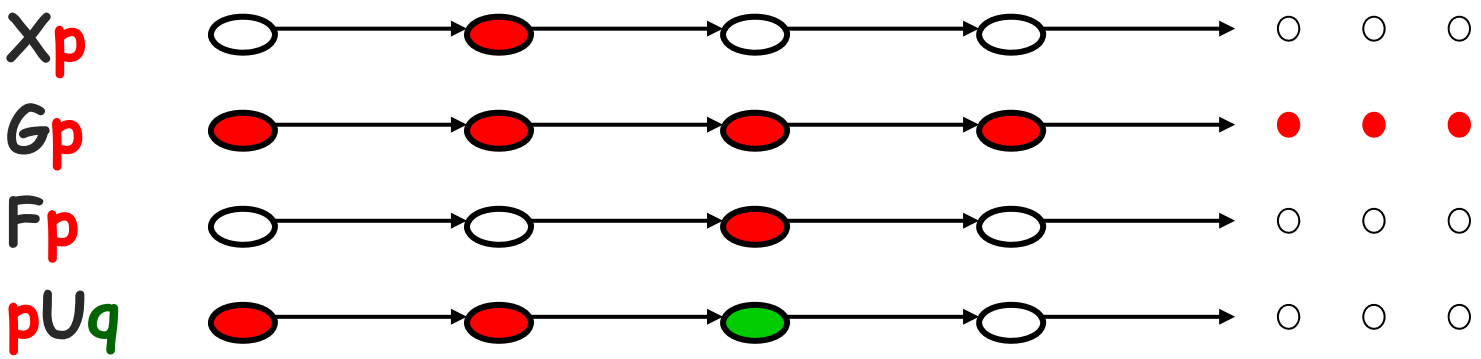
- **Branching** Time
  - Every moment has several successors
  - Infinite tree
  - Computation Tree Logic (CTL)



Propositional Temporal Logic

AP – a set of atomic propositions, $p \in AP$

Temporal operators:



Path quantifiers: **A** for **all** path

E there **exists** a path

Example to demonstrate:

- Building a model from a program
- Properties
- Model checking

Mutual Exclusion Example

- Two processes with a joint Boolean signal sem
- Each process P_i has a variable v_i describing its state:
  - $v_i = N$ Non critical
  - $v_i = T$ Trying
  - $v_i = C$ Critical

Mutual Exclusion Example

- Each process runs the following program:

P_i :: while (true) {

Atomic action

if (v_i == N) v_i = T;

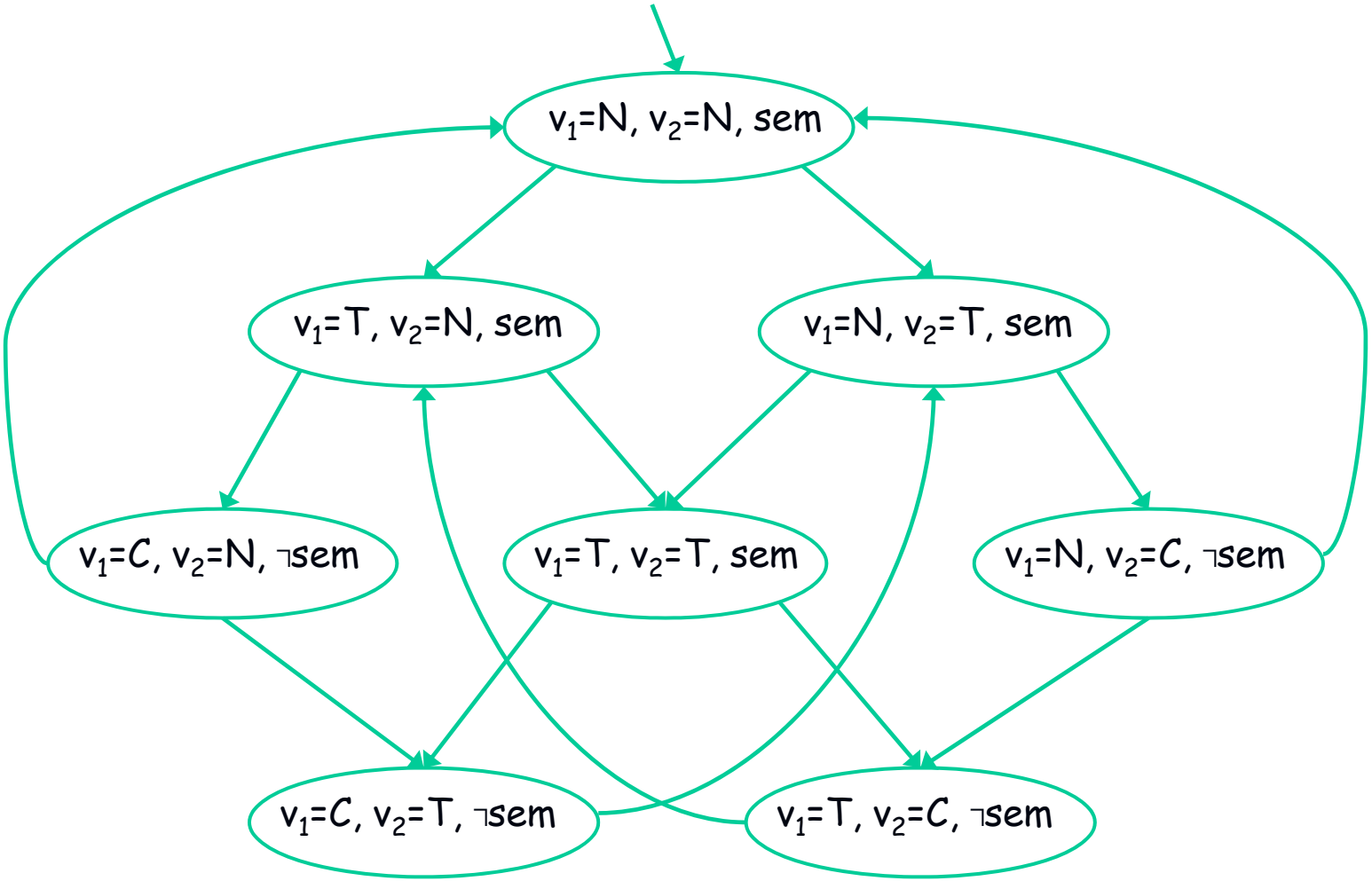
else if (v_i == T && sem) { v_i = C; sem = 0; }

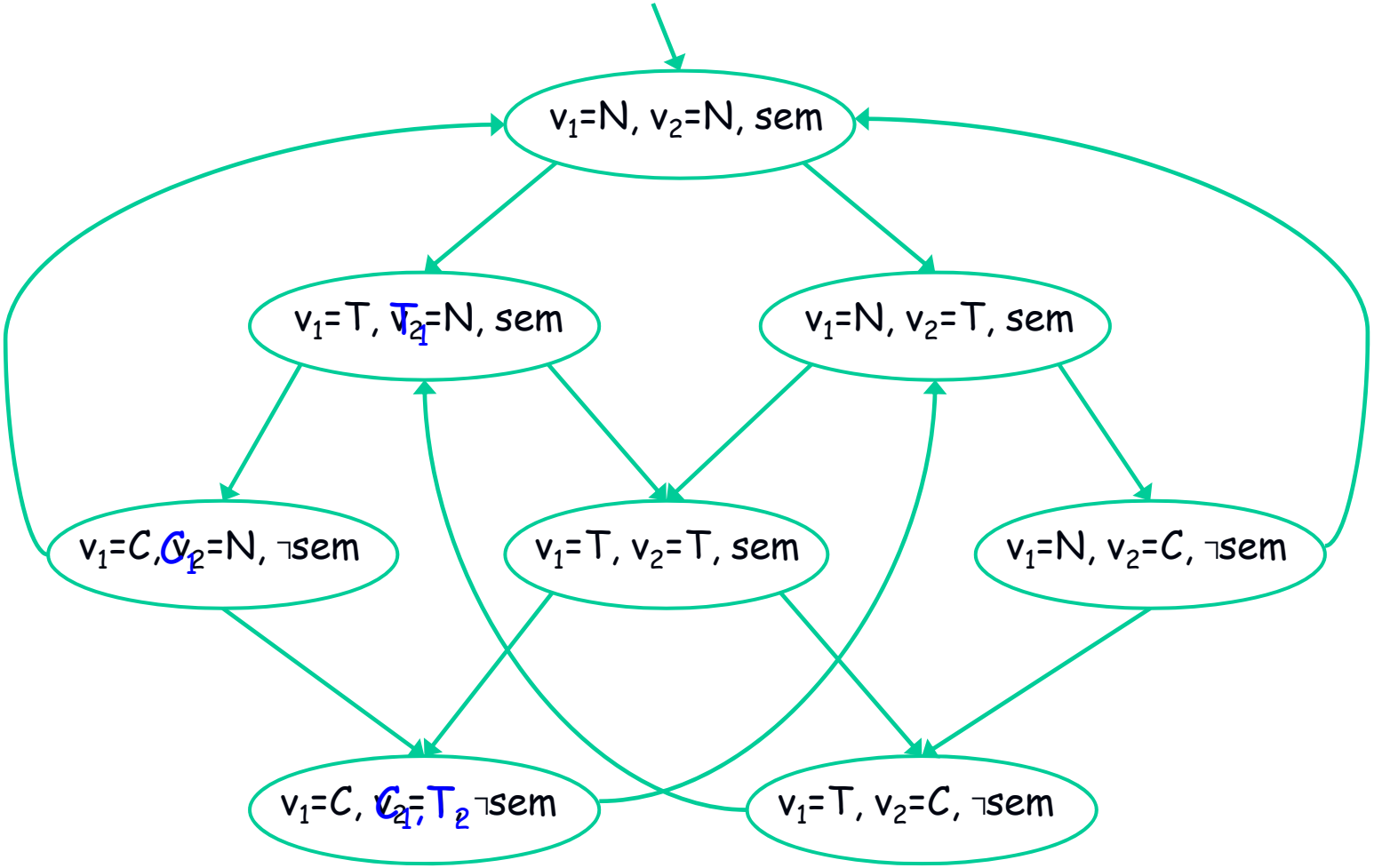
else if (v_i == C) {v_i = N; sem = 1; }

}

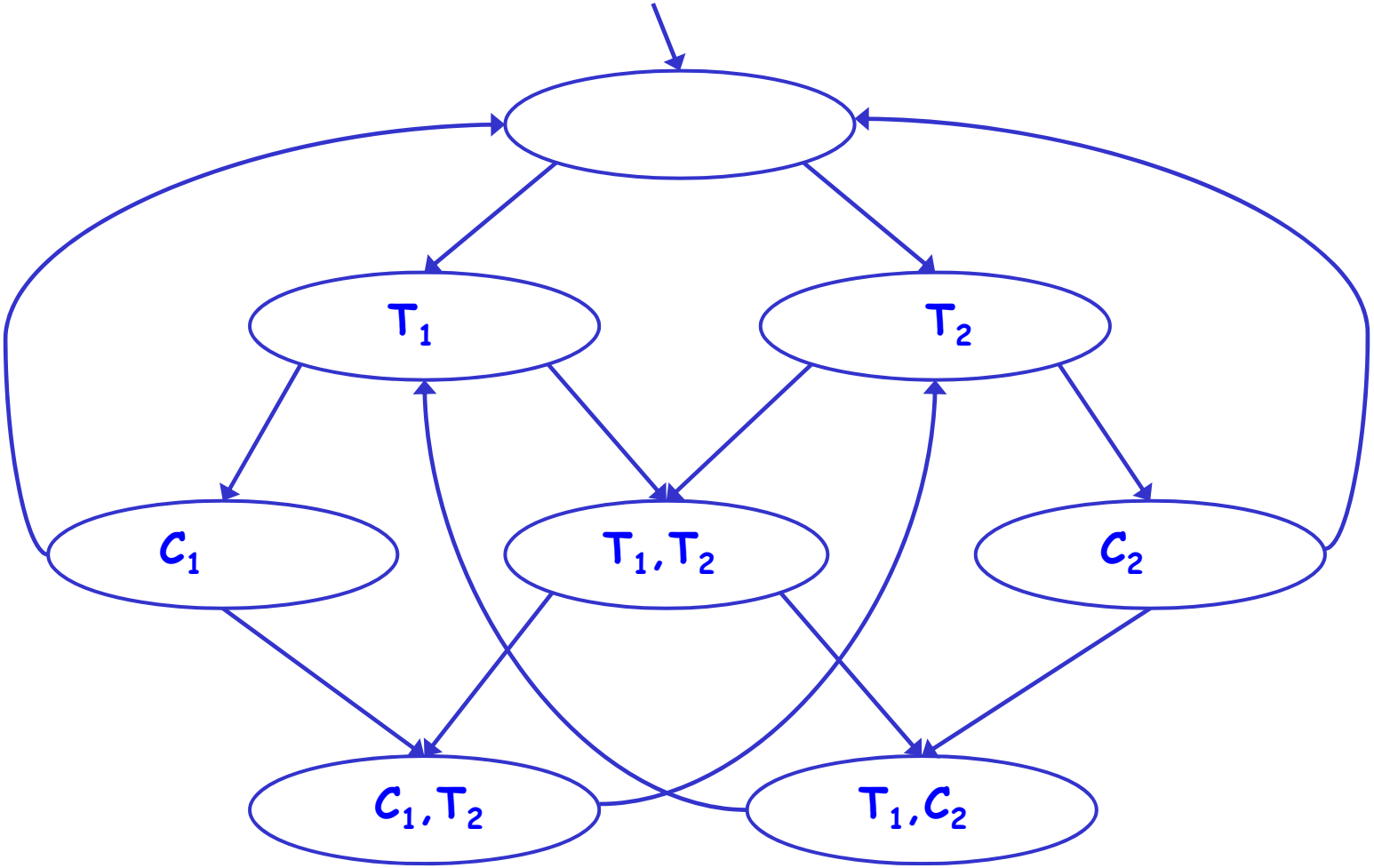
- The full program is: $P_1 || P_2$
- Initial state: (v_1=N, v_2=N, sem)
- The execution is interleaving

Mutual Exclusion Example

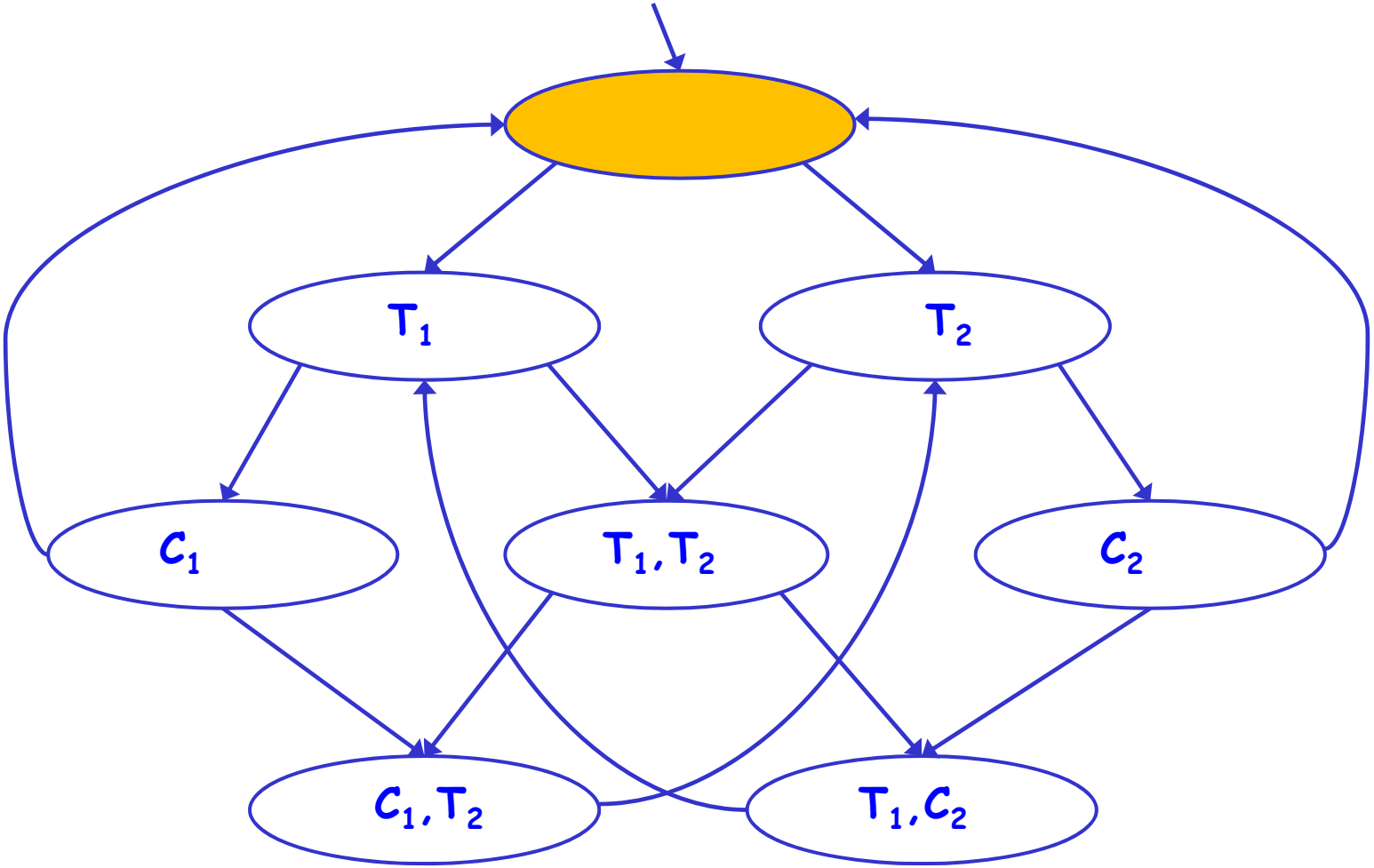




- We define atomic propositions: $AP=\{C_1, C_2, T_1, T_2\}$
- A state is marked with T_i if $v_i=T$
- A state is marked with C_i if $v_i=C$

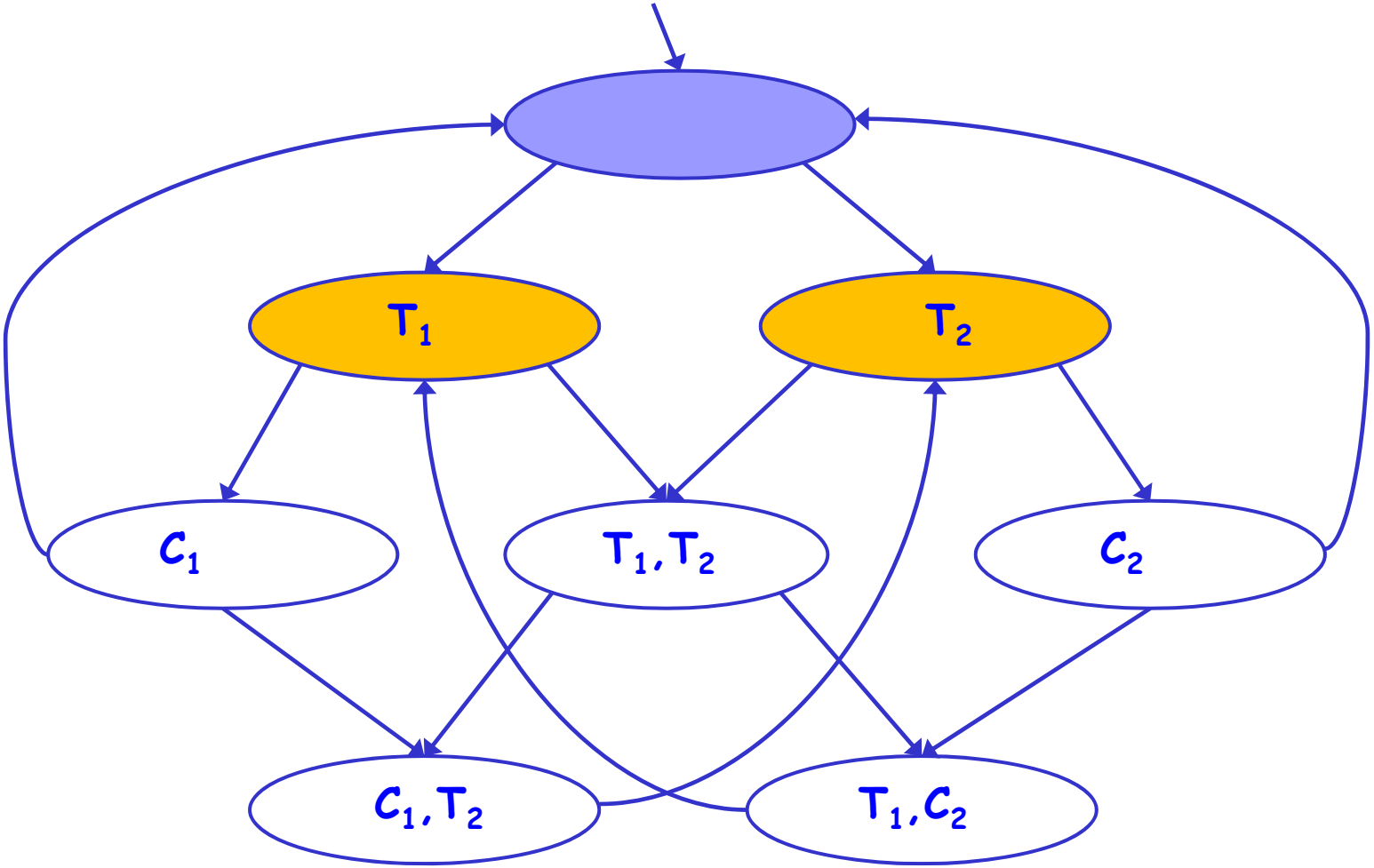


- Property 1: **AG**¬($C_1 \wedge C_2$)



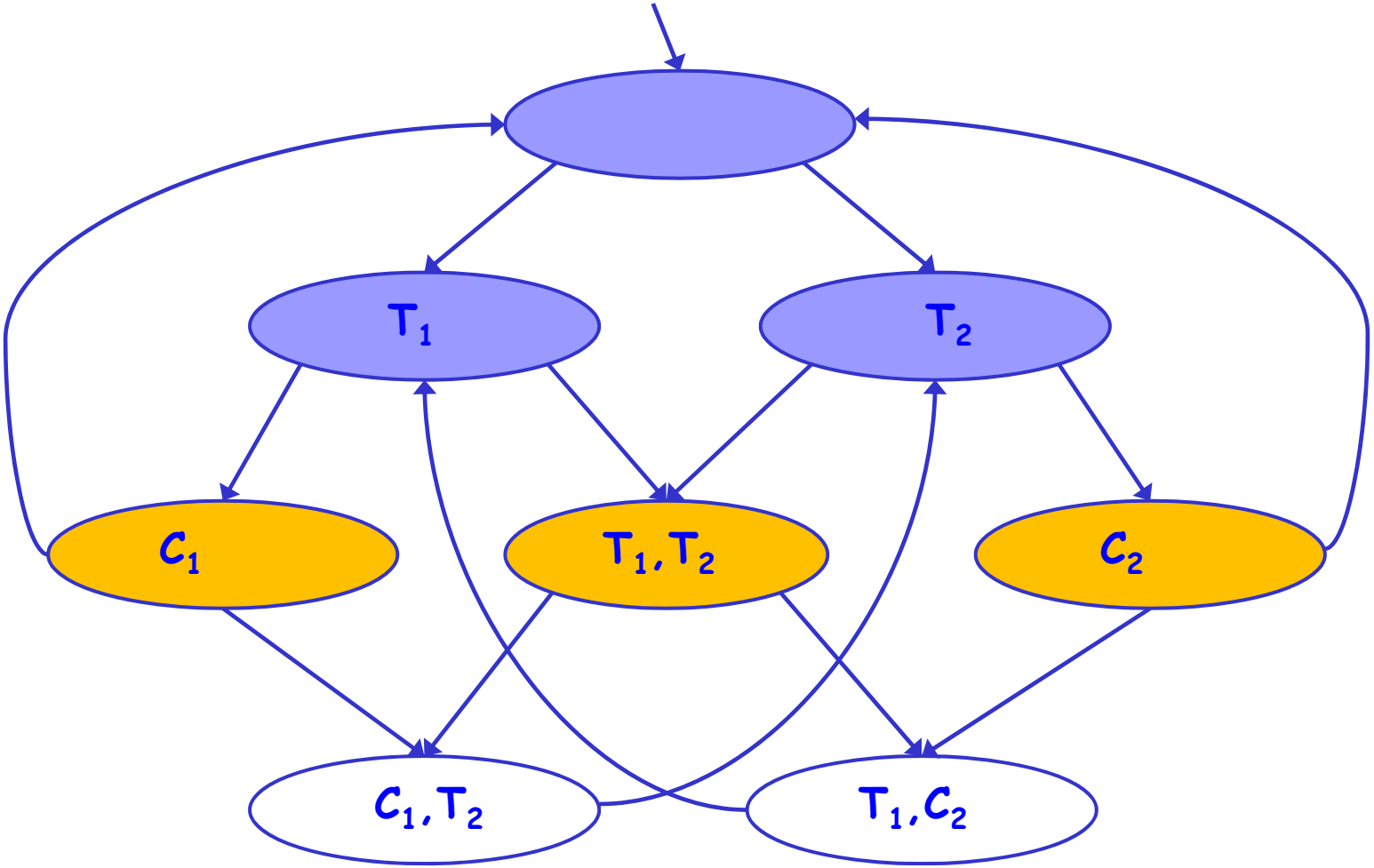
- Property 1: $\mathbf{AG}\neg(C_1 \wedge C_2)$

S_0



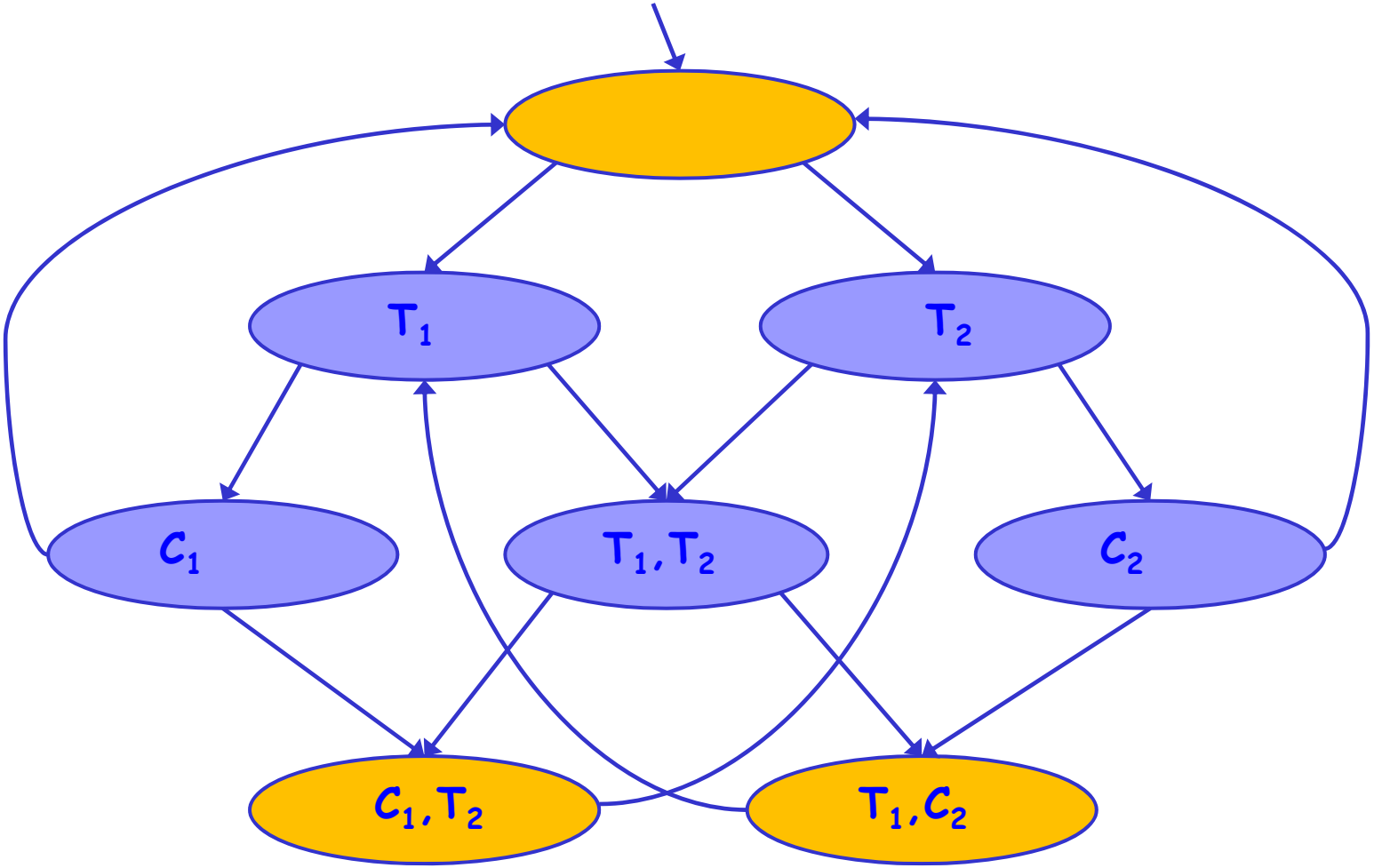
- Property 1: $\mathbf{AG}\neg(C_1 \wedge C_2)$

S_1



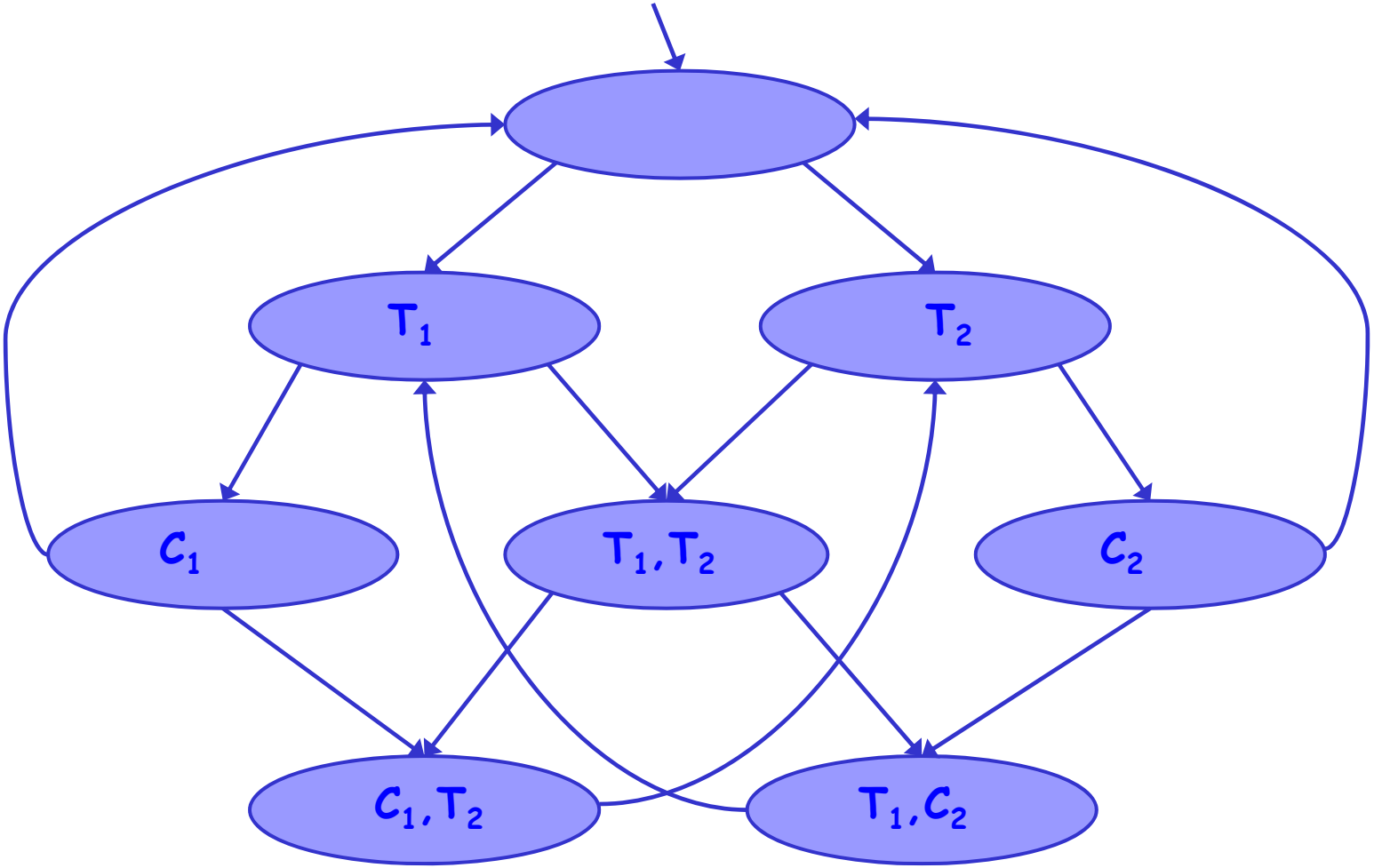
- Property 1: $\mathbf{AG}\neg(C_1 \wedge C_2)$

S_2



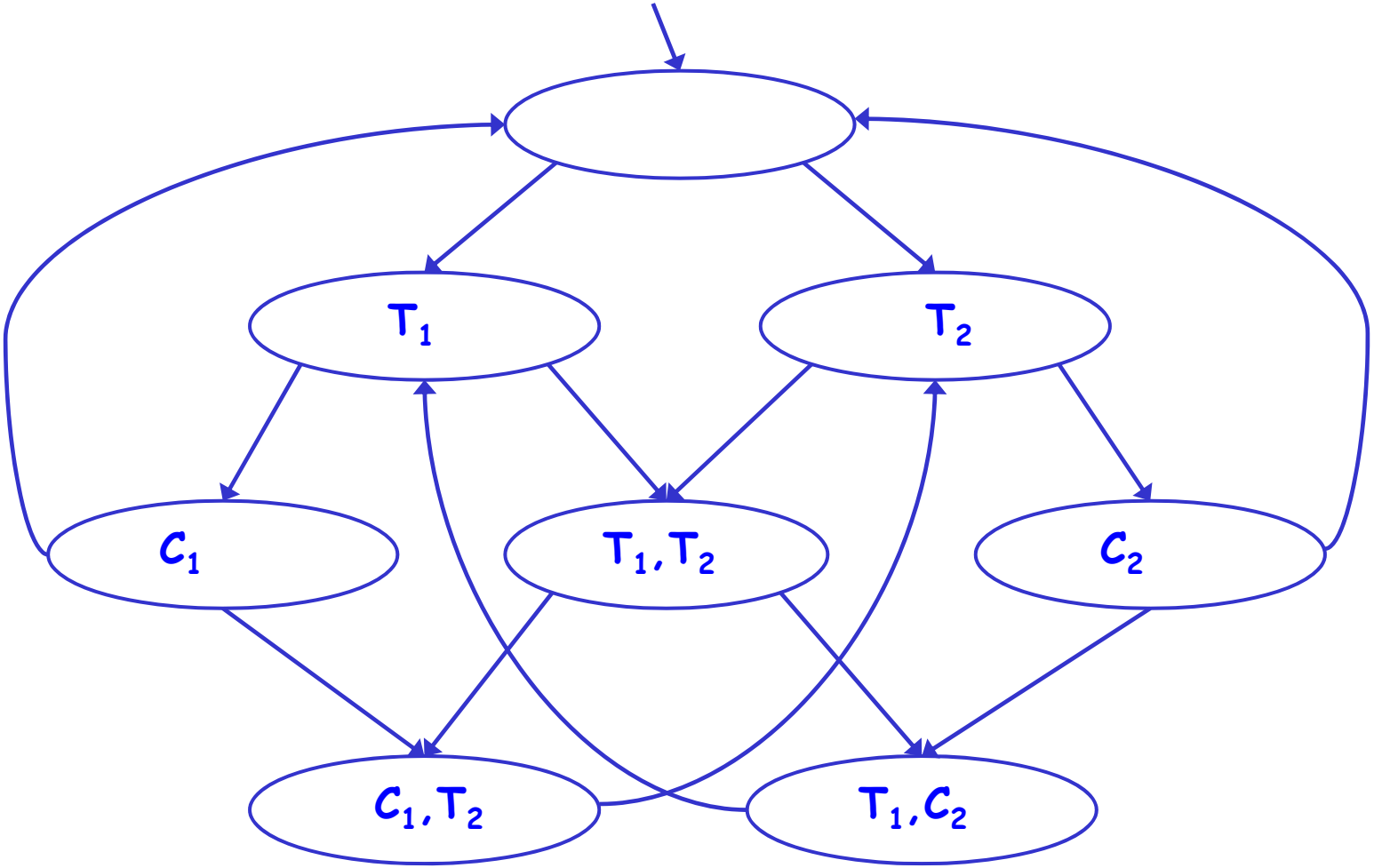
- Property 1: **AG**¬($C_1 \wedge C_2$)

S_3

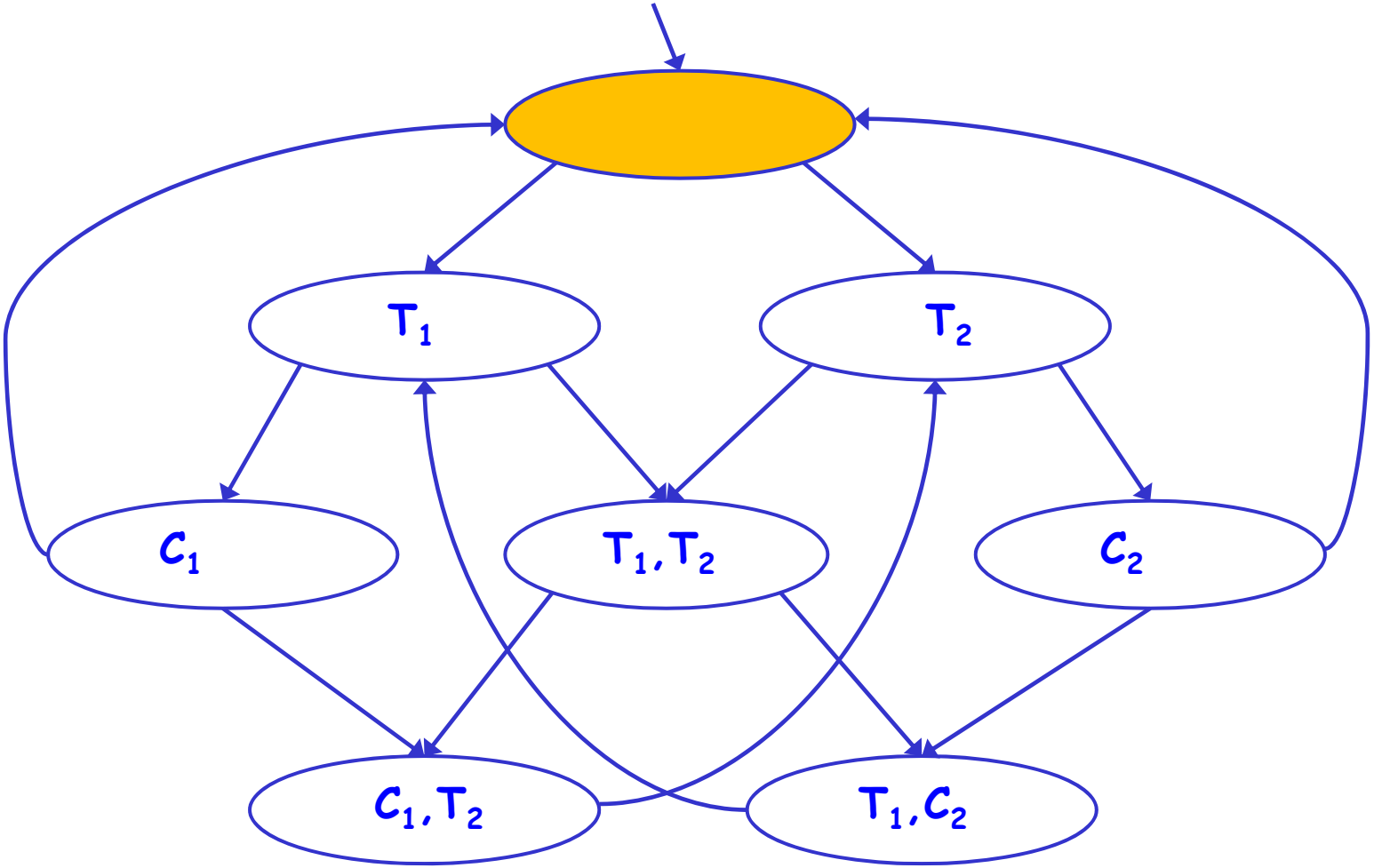


- $M \models AG \neg (C_1 \wedge C_2)$

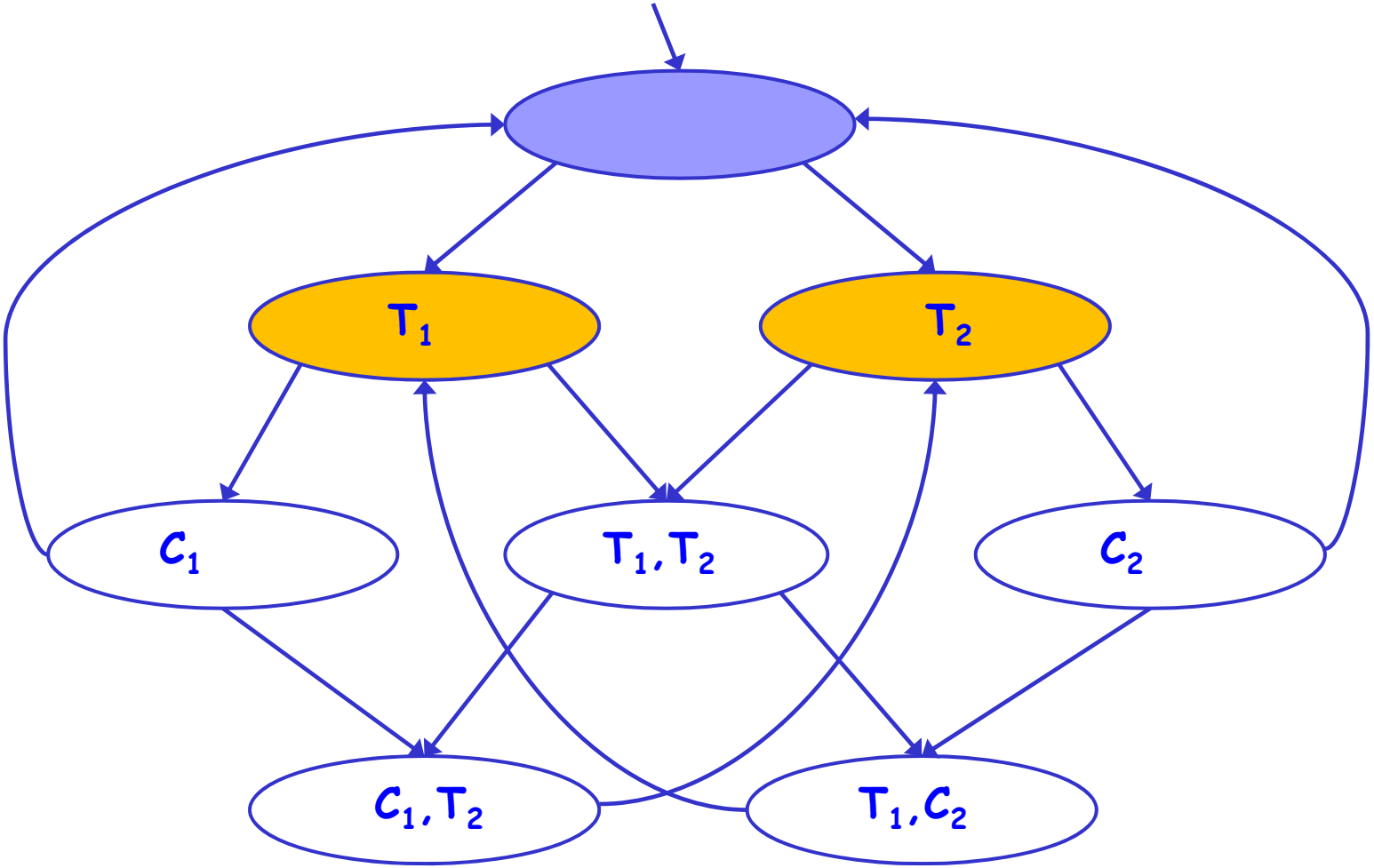
$$S_4 \subseteq S_0 \cup S_1 \cup S_2 \cup S_3$$



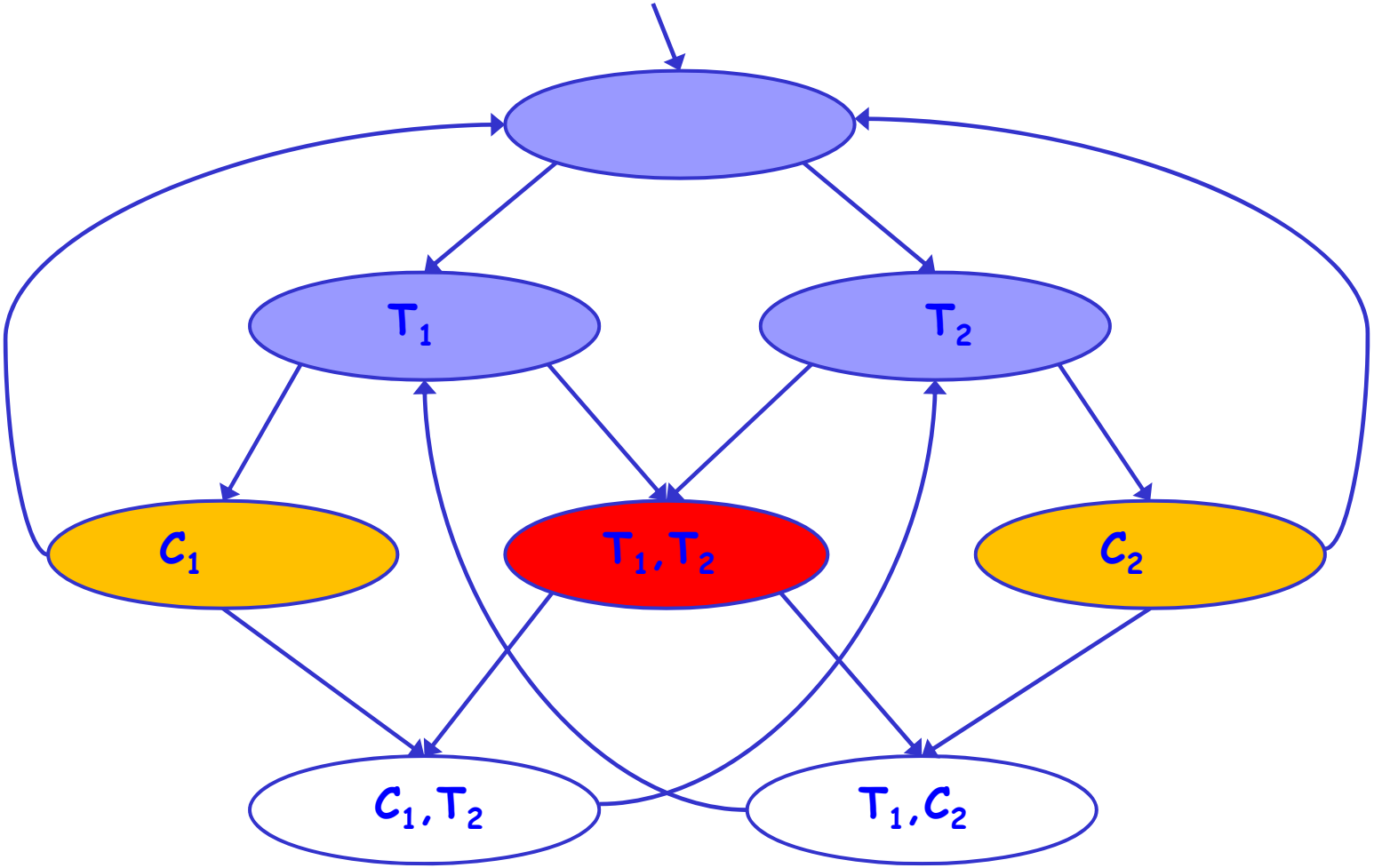
- Property 2: $AG\neg(T_1 \wedge T_2)$



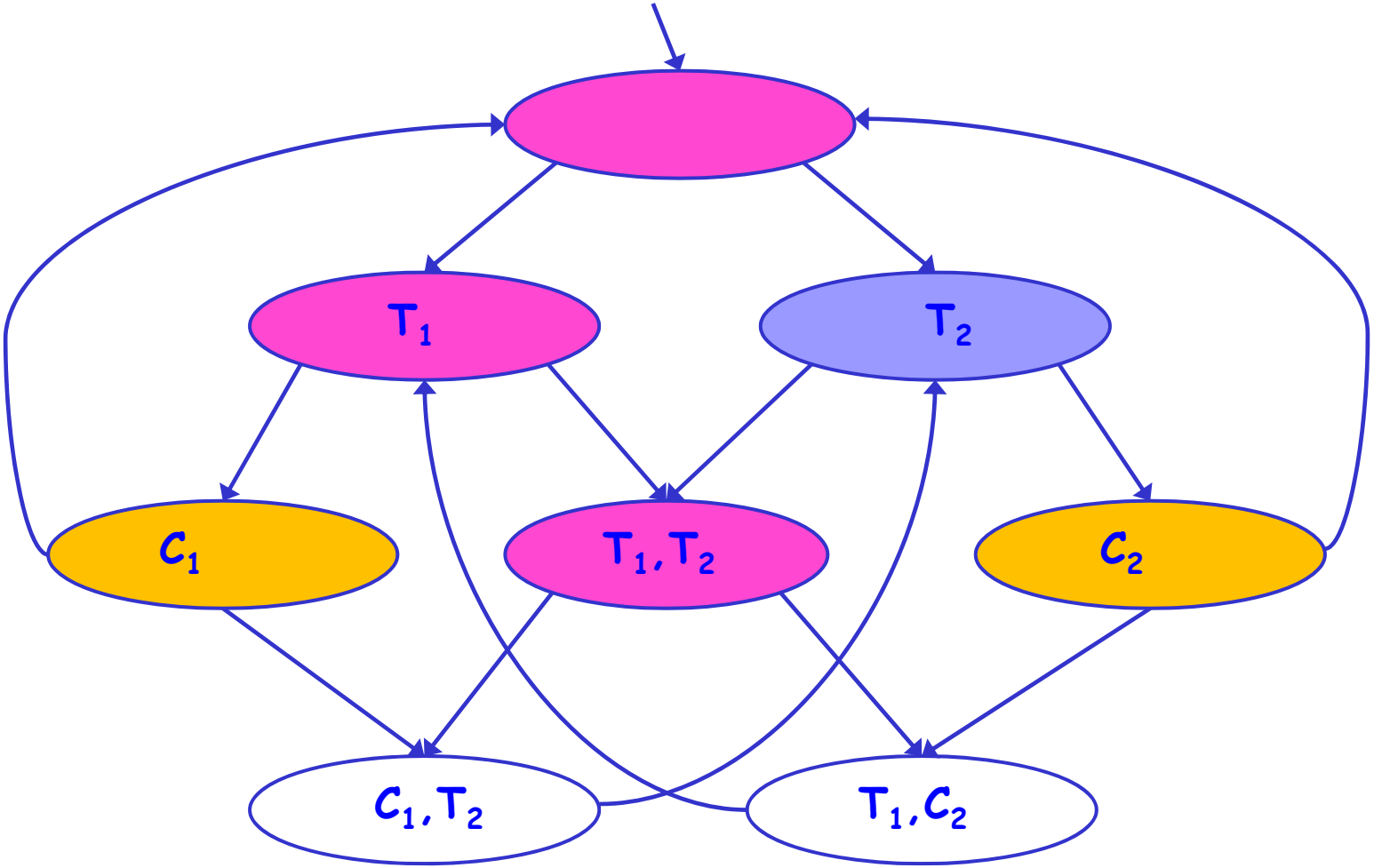
- Property 2: $AG\neg(T_1 \wedge T_2)$



- Property 2: $AG\neg(T_1 \wedge T_2)$

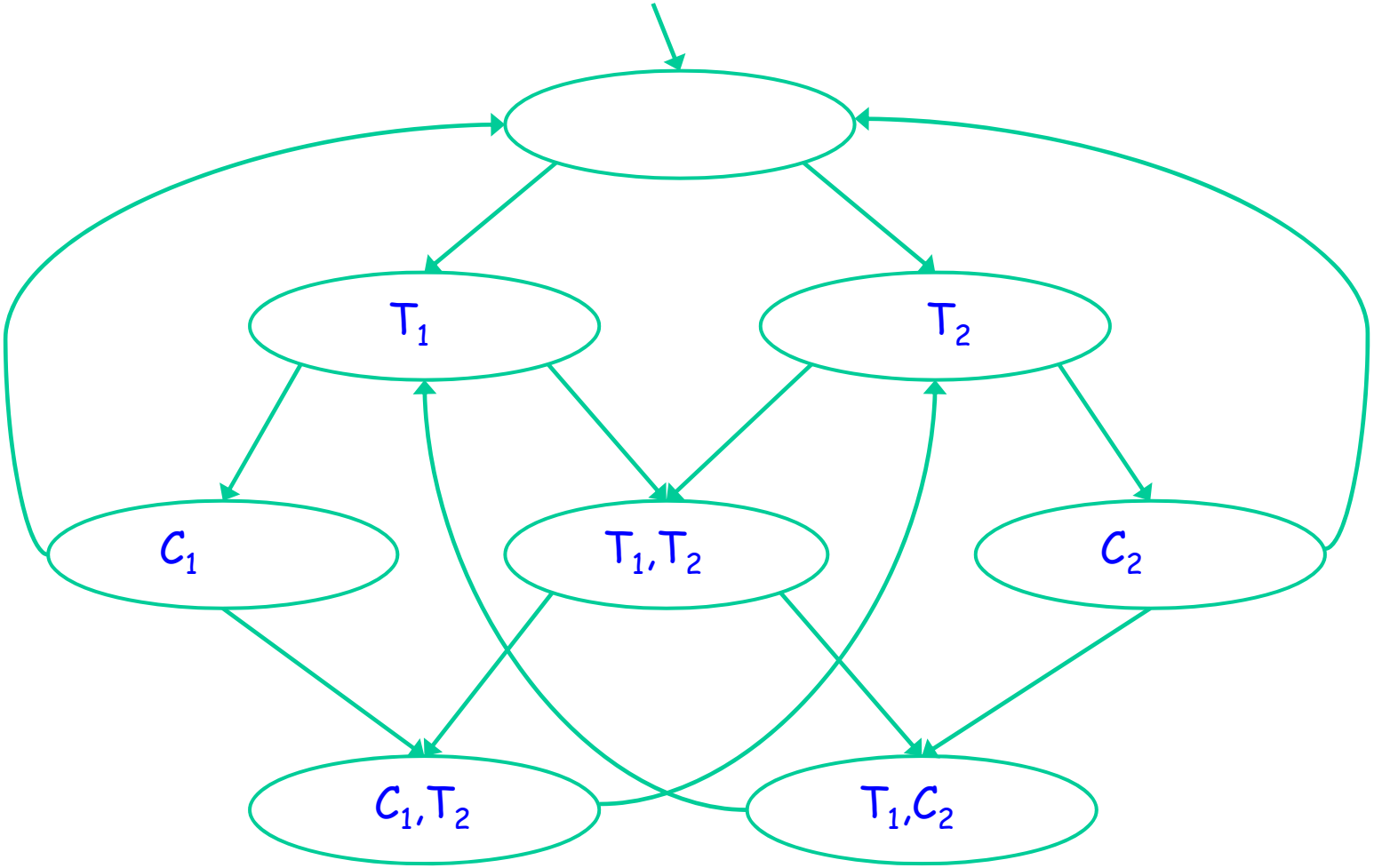


- $M \not\models AG \neg (T_1 \wedge T_2)$
- *A violating state has been found*



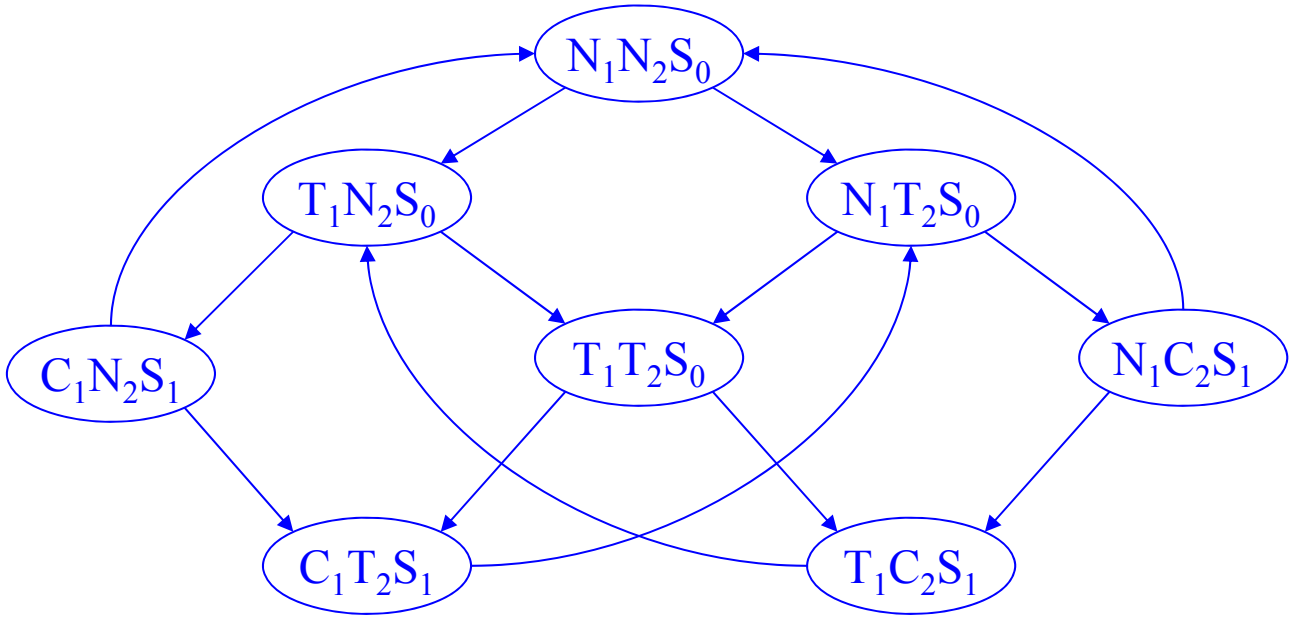
- $M \not\models AG \neg (T_1 \wedge T_2)$

Model checker returns a counterexample



- Property 3: $\mathbf{AG}((T_1 \rightarrow \mathbf{F}C_1) \wedge (T_2 \rightarrow \mathbf{F}C_2))$ ✗

Mutual Exclusion Example



$$M \models AG\ EF\ (N_1 \wedge N_2 \wedge S_0)$$

No matter where you are there is always a way to get to the initial state (restart)

Temporal logics

We present 3 (propositional) temporal logics:

- CTL*
- CTL
- LTL

CTL and LTL can be described as sub-logics of CTL*

CTL*

State formulas:

- $p \in AP$
- $\neg g_1$, $g_1 \vee g_2$, $g_1 \wedge g_2$ where g_1, g_2 are state formulas
- Ef, Af where f is a path formula

Path formulas:

- Every state formula g is a path formula
- $\neg f_1$, $f_1 \vee f_2$, $f_1 \wedge f_2$, Xf_1, Gf_1, Ff_1, $f_1 U f_2$ where f_1, f_2 are path formulas

CTL* - set of all state formulas

Semantics of CTL*

$\pi = s_0, s_1, \ldots$ is a path in M if $R(s_i, s_{i+1})$ for every i.

π^i – the suffix of π starting at s_i.

State formulas:

- $M,s \models p \Leftrightarrow p \in L(s)$
- $M,s \models Ef \Leftrightarrow$ there is a path π from s s.t. $M, \pi \models f$
- $M,s \models Af \Leftrightarrow$ for every path π from s, $M, \pi \models f$

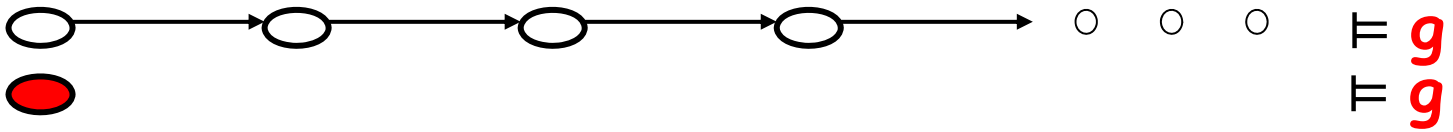
Semantics of CTL*

$\pi = s_0, s_1, \ldots$ is a path in M if $R(s_i, s_{i+1})$ for every i.

π^i – the suffix of π starting at s_i.

Path formulas:

- M, $\pi \models g$, where g is a state formula $\Leftrightarrow$ M, $s_0 \models g$



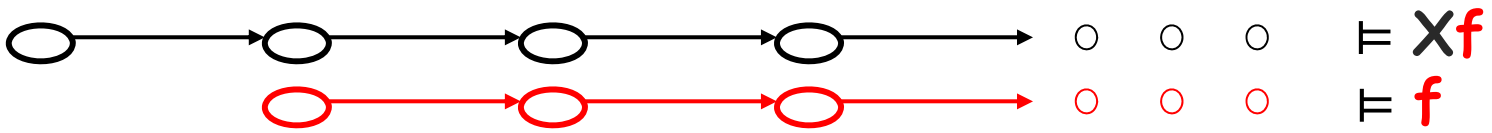
Semantics of CTL*

$\pi = s_0, s_1, \ldots$ is a path in M if $R(s_i, s_{i+1})$ for every i.

π^i – the suffix of π starting at s_i.

Path formulas:

- M, $\pi \models g$, where g is a state formula $\Leftrightarrow$ M, $s_0 \models g$
- M, $\pi \models Xf \Leftrightarrow$ M, $\pi^1 \models f$



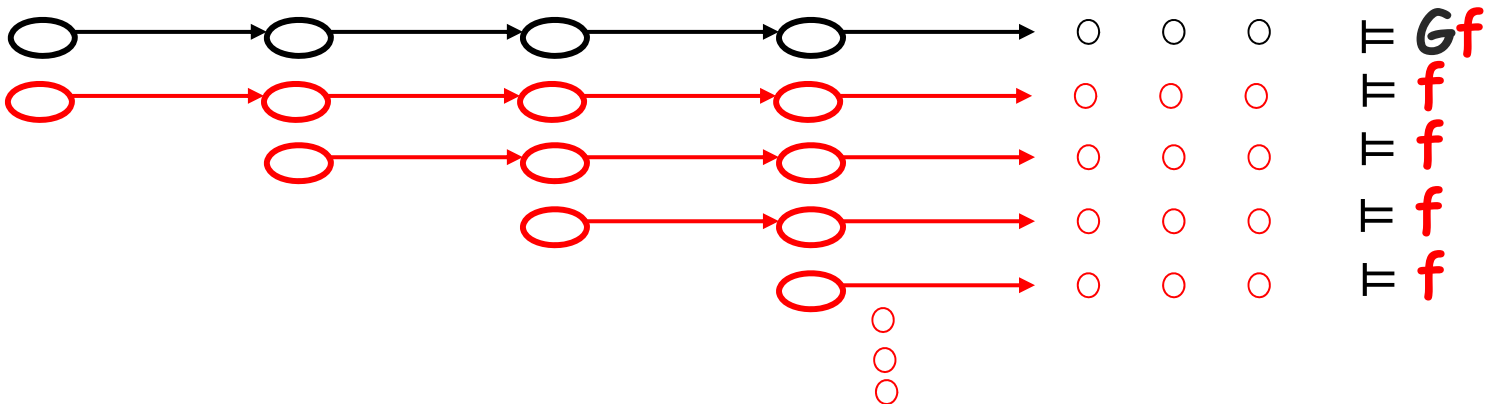
Semantics of CTL*

$\pi = s_0, s_1, \ldots$ is a path in M if $R(s_i, s_{i+1})$ for every i.

π^i – the suffix of π starting at s_i.

Path formulas:

- M, $\pi \models g$, where g is a state formula $\Leftrightarrow$ M, $s_0 \models g$
- M, $\pi \models Xf \Leftrightarrow$ M, $\pi^1 \models f$
- M, $\pi \models Gf \Leftrightarrow$ for every $k \geq 0$, M, $\pi^k \models f$



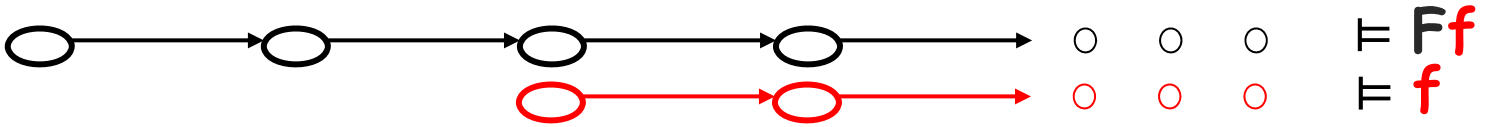
Semantics of CTL*

$\pi = s_0, s_1, \ldots$ is a path in M if $R(s_i, s_{i+1})$ for every i.

π^i – the suffix of π starting at s_i.

Path formulas:

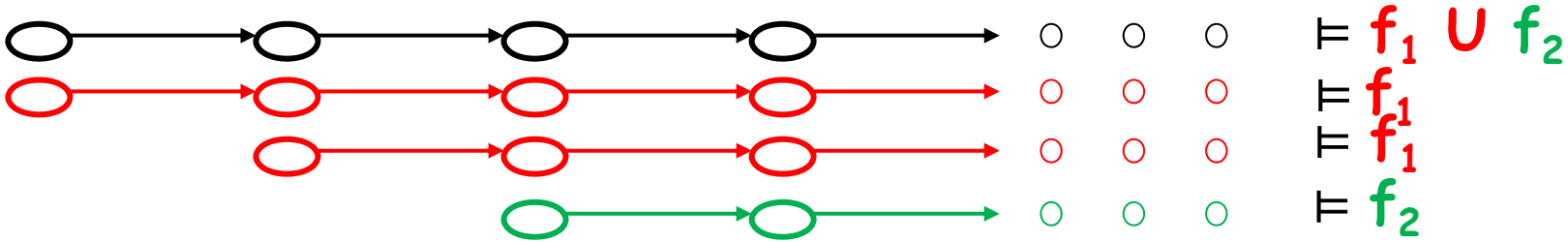
- $M, \pi \models g$, where g is a state formula $\Leftrightarrow M, s_0 \models g$
- $M, \pi \models Xf \Leftrightarrow M, \pi^1 \models f$
- $M, \pi \models Gf \Leftrightarrow$ for every $k \geq 0$, $M, \pi^k \models f$
- $M, \pi \models Ff \Leftrightarrow$ there exists $k \geq 0$, s.t. $M, \pi^k \models f$



Semantics of CTL*

$\pi = s_0, s_1, \ldots$ is a path in M if $R(s_i, s_{i+1})$ for every i.

π^i – the suffix of π starting at s_i.



- M, $\pi \models Gf \Leftrightarrow$ for every $k \geq 0$, M, $\pi^k \models f$
- M, $\pi \models Ff \Leftrightarrow$ there exists $k \geq 0$, s.t. M, $\pi^k \models f$
- M, $\pi \models f_1 \, U \, f_2 \Leftrightarrow$ there exists $k \geq 0$, s.t. M, $\pi^k \models f_2$ and for every $0 \leq j < k$, M, $\pi^j \models f_1$

Semantics of CTL*

$\pi = s_0, s_1, \ldots$ is a path in M if $R(s_i, s_{i+1})$ for every i.

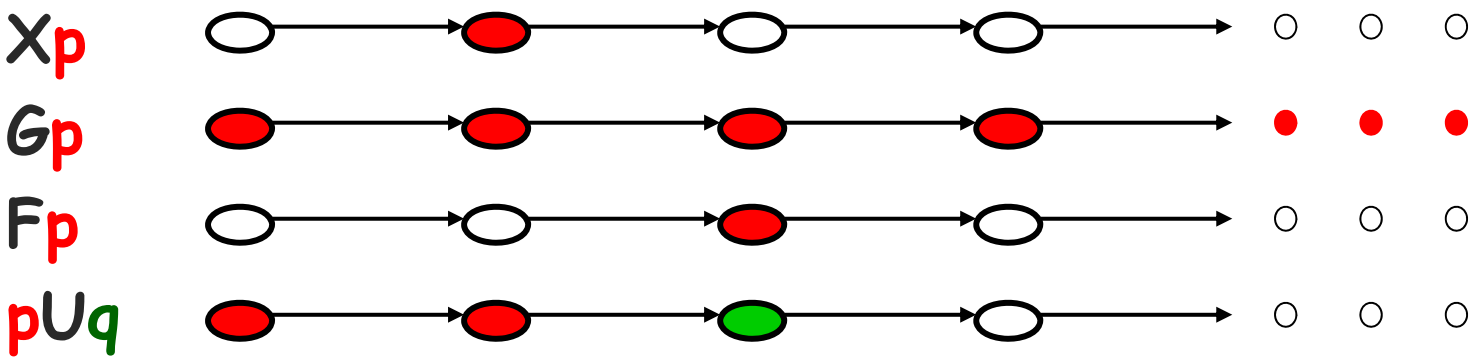
π^i – the suffix of π starting at s_i.

Path formulas:

- $M, \pi \models g$, where g is a state formula $\Leftrightarrow M, s_0 \models g$
- $M, \pi \models Xf \Leftrightarrow M, \pi^1 \models f$
- $M, \pi \models Gf \Leftrightarrow$ for every $k \geq 0$, $M, \pi^k \models f$
- $M, \pi \models Ff \Leftrightarrow$ there exists $k \geq 0$, s.t. $M, \pi^k \models f$
- $M, \pi \models f_1 \, U \, f_2 \Leftrightarrow$ there exists $k \geq 0$, s.t. $M, \pi^k \models f_2$ and for every $0 \leq j < k$, $M, \pi^j \models f_1$

Semantics of Path Formulas

If p,q are state formulas then:



But in the general case, they can be path formulas

Semantics of CTL*

$M \models g \Leftrightarrow$ for every initial state s: $M,s \models g$

Examples

LTL/CTL/CTL*

LTL – state formulas of the form $A\psi$

ψ - path formula, contains **no** path **quantifiers**

- interpreted over infinite computation paths

CTL – state formulas where path quantifiers and temporal operators appear in pairs:

AG, AU, AF, AX, EG, EU, EF, EX

- interpreted over infinite computation trees

CTL* - Allows any combination of temporal operators and path quantifiers. Includes both LTL and CTL

LTL/CTL/CTL*

LTL – state formulas of the form $A\psi$

ψ - path formula, contains **no** path **quantifiers**

- interpreted over infinite computation paths

CTL – state formulas where path quantifiers and temporal operators appear in pairs:

AG, AU, AF, AX, EG, EU, EF, EX

- interpreted over infinite computation trees

CTL* - Allows any combination of temporal operators and path quantifiers. Includes both LTL and CTL

LTL

State formulas:

- Af where f is a path formula

Path formulas:

- $p \in AP$
- $\neg f_1$, $f_1 \vee f_2$, $f_1 \wedge f_2$, Xf_1, Gf_1, Ff_1, $f_1 U f_2$ where f_1, f_2 are path formulas

LTL - set of all state formulas

CTL

CTL - set of all state formulas

- $p \in AP$
- $\neg g_1$, $g_1 \vee g_2$, $g_1 \wedge g_2$
- **AX** g_1, **AG** g_1, **AF** g_1, **A** g_1 **U** g_2,
- **EX** g_1, **EG** g_1, **EF** g_1, **E** g_1 **U** g_2,

  where g_1, g_2 are state formulas

Semantics of CTL

Recall: path $\pi = s_0, s_1, \ldots$

- $M,s \models p \iff p \in L(s)$ for $p \in AP$

- $M,s \models \varphi_1 \vee \varphi_2 \iff M,s \models \varphi_1$ or $M,s \models \varphi_2$
- $M,s \models EX\varphi \iff$ there is s' s.t. $R(s,s')$ and $M,s' \models \varphi$
- $M,s \models EG\varphi \iff$ there is a path π from s, s.t. for every $i \geq 0$, $M,s_i \models \varphi$

Semantics of CTL

- $M,s \models E[\varphi_1 U \varphi_2] \Leftrightarrow$ there is a path π from s
  and there is $k \geq 0$ s.t. $M,s_k \models \varphi_2$
  and for every $k > i \geq 0$, $M,s_i \models \varphi_1$

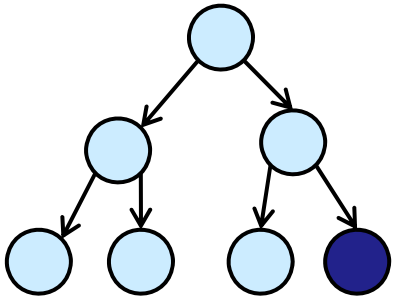
- $M,s \models AG\varphi \Leftrightarrow$ for every path π from s
  and for every $i \geq 0$, $M,s_i \models \varphi$

- $M,s \models AF\varphi \Leftrightarrow$ for every path π from s
  there exists $i \geq 0$ s.t. $M,s_i \models \varphi$

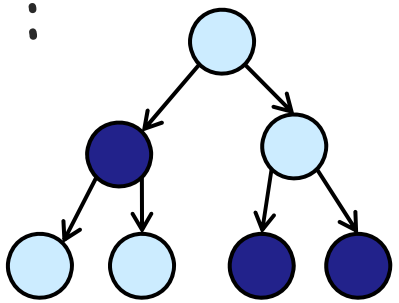
Illustration of CTL Semantics

EFp :

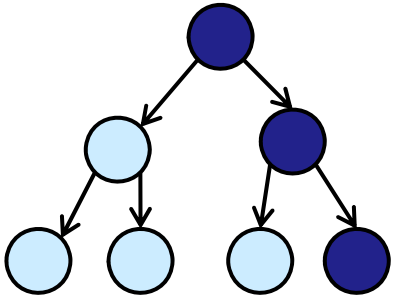
"exists reachable state s.t."



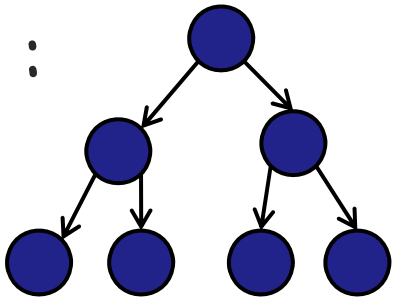
AFp :



EGp :



AGp :



"all reachable states…."

Property types

	Universal	Existential
Safety	AGp	EGp
Liveness	AFp	EFp

Examples (LTL)

1. $AG\ \neg(start \wedge \neg ready)$
2. $AG\ (req \rightarrow F\ ack)$
3. $A\ GF\ enbled$
4. $A\ FG\ deadlock$
5. $A\ (GF\ enbled \rightarrow GF\ running)$

Cannot express existential properties: "from any state the system can…"

Examples (CTL)

1. EF (start$\wedge\neg$ready)
2. AG (req $\rightarrow$ AF ack)
3. AG (AF enbled)
4. AF (AG deadlock)
5. AG (EF restart)
6. AG (non_critical $\rightarrow$ EX tryring)
7. AG (try $\rightarrow$ A[try U succeed])

Equivalence

- Path formulas ψ_1, ψ_2 are equivalent if:
  For every M and path π
  $$M, \pi \models \psi_1 \text{ iff } M, \pi \models \psi_2$$

- State formulas φ_1, φ_2 are equivalent if:
  For every M and state s
  $$M, s \models \varphi_1 \text{ iff } M, s \models \varphi_2$$

Expressiveness

$\neg$, $\vee$, X, U, E suffice to express all CTL*:

- $Ff \equiv \text{true}\ U\ f$
- $Gf \equiv \neg F(\neg f)$
- $Af \equiv \neg E(\neg f)$

In CTL: EX, EG, EU are sufficient

- $A[pUq] \equiv (\neg EG\ \neg q) \wedge \neg E[\neg q\ U\ (\neg p \wedge \neg q)]$

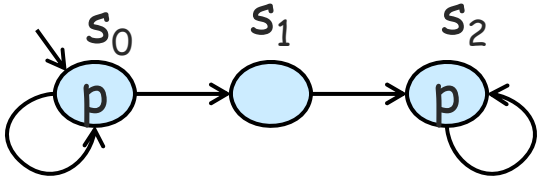
LTL vs. CTL

- A (FG p) has no equivalent in CTL

  "in all paths, p globally holds from some point on"

- Failed attempts:

  AFAGP : "in every path there is a point from which all reachable states satisfy p."



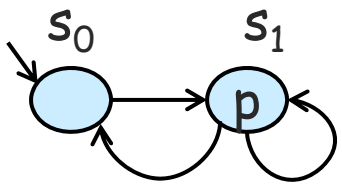
All paths satisfy FGp

- $s_0,s_0,s_0,\ldots$
- $s_0,s_0,\ldots s_0,s_1,s_2,s_2,s_2\ldots$

But first one does not sat FAGp

LTL vs. CTL

- A (FG p) has no equivalent in CTL
  "in all paths, p globally holds from some point on"
- What about AFEGP ?
  "in every path there is a point from which there is a path where p globally holds"



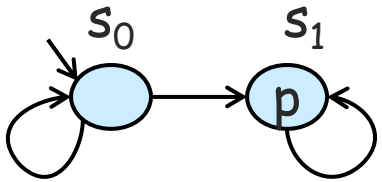
All paths satisfy FEGp
- since s_1 sat EGp
But s_0,s_1,s_0,s_1,s_0,s_1... does not sat FGp

LTL vs. CTL

- AG (EFp) has no equivalent in LTL
  - " all reachable states can reach p"

- Failed attempt:

  AGFp : "in all paths, p holds infinitely many times."



All reachable states (s_0, s_1) satisfy EFp

But $s_0,s_0,s_0,\ldots$ does not satisfy GFp

LTL and CTL vs. CTL*

- E (GFp) has no equivalent in LTL or CTL