Tecniche di Specifica e di Verifica

Linear Time Temporal Logic

Temporal Logics: The context

- ***Kripke Structures*** model systems.
- ***Temporal logics*** model the dynamic behavioral properties of systems.
  - **Linear Time**
  - Branching Time
- ***Model checking*** can be used to determine if a system has the desired behavioral property.

Properties of computations: local properties

Refer to **immediate successors** or **predecessors** of the current state.

Examples:

Some/every immediate successor state satises the property φ:

- The system may enable the process ***i*** at the next state.
- If the light was red at the previous state and is orange now, it must turn green at the next state.

Some/every immediate predecessor satises the property φ (usually expressed as conditionals):

- If the process ***i*** is currently enabled, the scheduler must have disabled the process ***j*** at the previous state.
- If train is entering the tunnel now, the semaphore must have been switched red on the other end at the previous moment.

Local properties can be iterated a fixed number of times, but ***not indefinitely***.

Universal properties of computations: invariance, safety

Invariance properties are properties that **must always hold** along the computation, while ***safety properties*** describe events that **must never happen** along the computation.

Invariance:

- The greatest common divisor of **X** and **Y** remains the same throughout the execution.

Safety:

- No ***deadlock*** will ever occur.
- At least one process will be enabled at any moment of time.
- Not more than one process will ever be in its ***critical section*** (e.g., not more than one train will ever be in the tunnel) at the same time.
- A resource will never be available to two or more processes simultaneously.

Also, **partial correctness** properties:

- If a ***pre-condition*** **P** holds at all initial states, then a ***post-condition*** **Q** will/must hold at all accepting (terminating) states.

Existential properties of computations: eventualities, liveness

Eventuality, liveness properties: those that **will (must) happen sometime** during the computation.

Examples:

- The execution of the program will terminate.
- If the train has entered the tunnel, it will eventually leave it.
- Once a printing job is activated, eventually it will be completed.
- If a message is sent, eventually it will be delivered.

Also, **total correctness** properties:

- If a ***pre-condition*** **P** holds at the initial state, then the computation will reach an accepting (terminating) state, where the ***post-condition*** **Q** will hold.

Properties of computations: fairness, precedence

Fairness properties: All processes will be treated "***fairly***" by the operating system (the scheduler, etc.)

Examples:

- ***Weak fairness***: Every continuous request is eventually granted.
- ***Strong fairness***: If a request is repeated infinitely often then it is eventually granted.
- ***Impartiality***: Every process is scheduled infinitely often.
- ***Precedence***: The event α will occur before the event β, which may or may not occur at all.
  - If the train has entered the tunnel, it will eventually leave it (before any other train has entered it).

Reachability properties in transition systems

- All important properties of computations can be expressed in terms of ***reachability*** or ***non-reachability*** of states with specic atomic properties.
- For instance, ***eventuality*** is just ***reachability*** of a "**good state**", while ***safety*** is ***non-reachability*** of "**bad states**", ***fairness*** corresponds to ***repeated reachability***, etc.
- More generally, we may interested in ***reachability*** of a state or a set of states along some or all paths starting from a given state (or, set of states); this is called **forward reachability**.
- Likewise we may be interested in the states from which a state (or a set of states) is reachable; this is calles **backward reachability**.

Linear time temporal logics.

- ***LTL (Linear Time Temporal Logic)***
  - **Syntax**
  - **Semantics**
  - The Model Checking Problem.
  - Its solution.

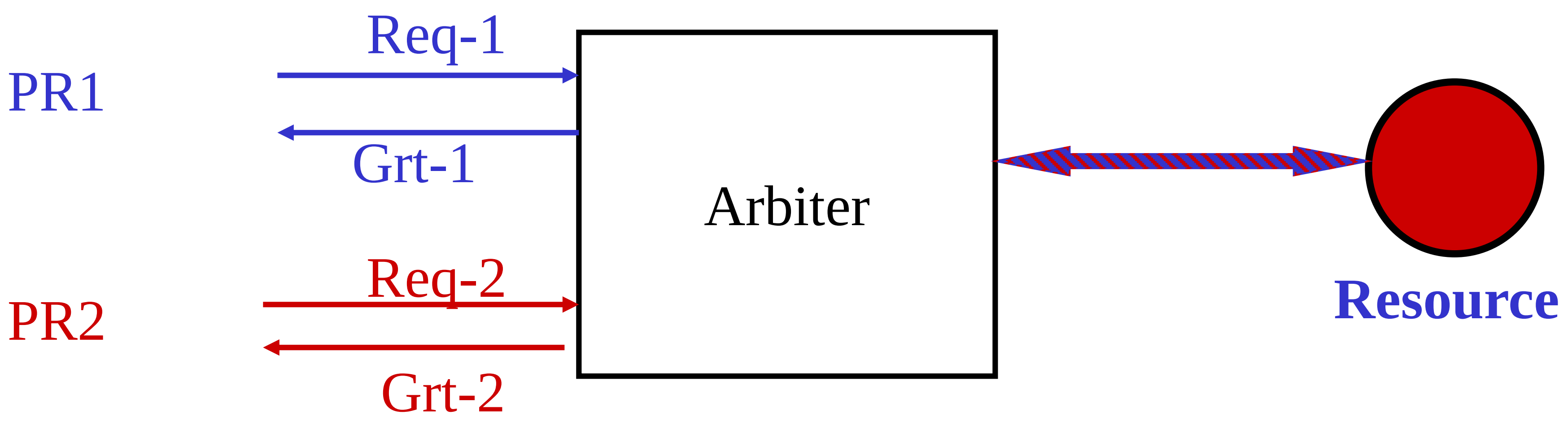
The Application

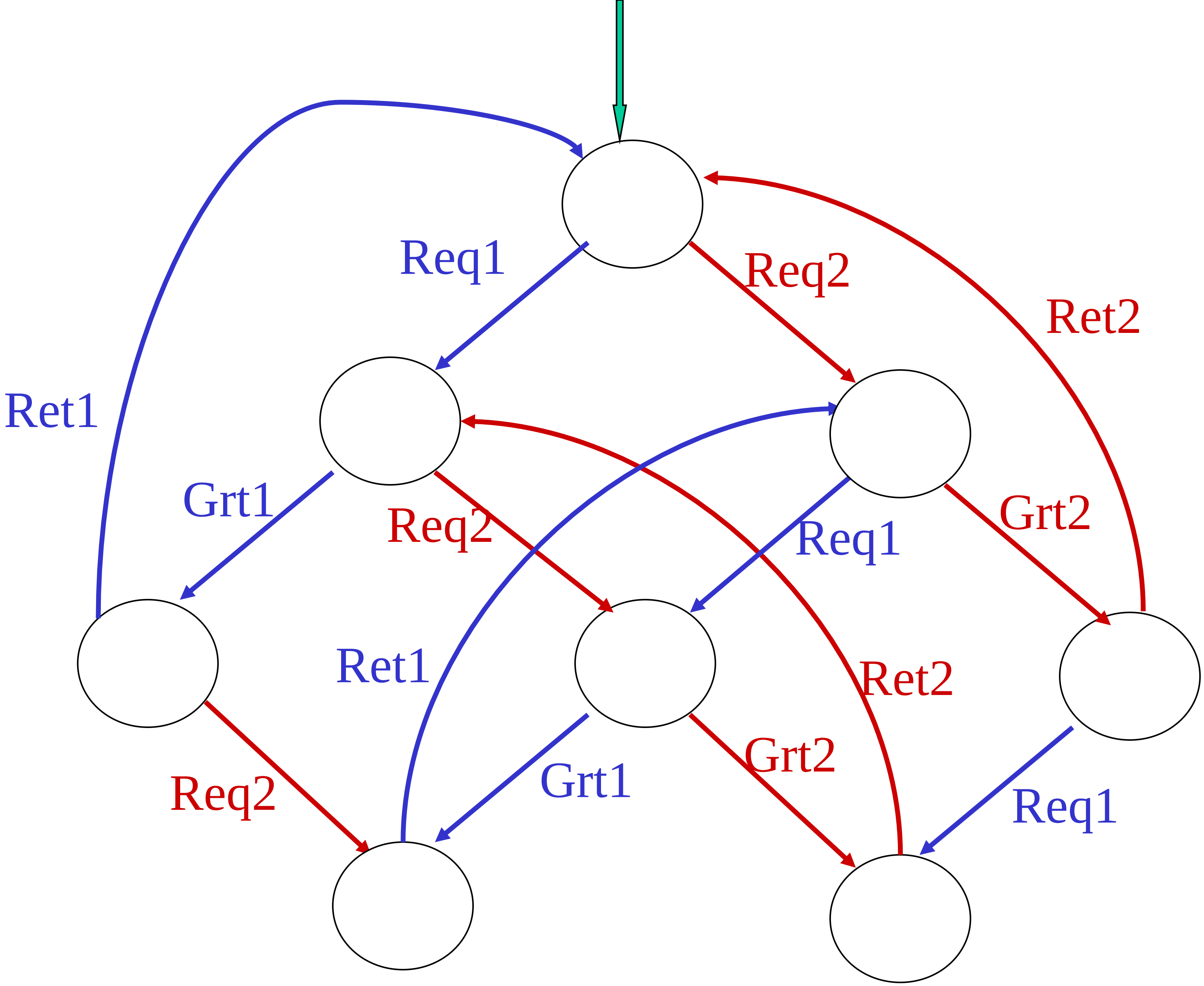
- Model a system to be verified as a Kripke structure:
  - Transition System **$TS = (S, S_0, R)$**
  - **AP** = A finite set of atomic propositions.
    - Basic assertions about the system
  - **$L : S \longrightarrow 2^{AP}$** = The set of subsets of **AP**.
  - **$p \in L(s)$ ---- p** is true at **s**.
  - **$p \notin L(s)$ ---- p** is not true at **s**.
- **$K = (S, S_0, R, AP, L)$** ---- Kripke Structure

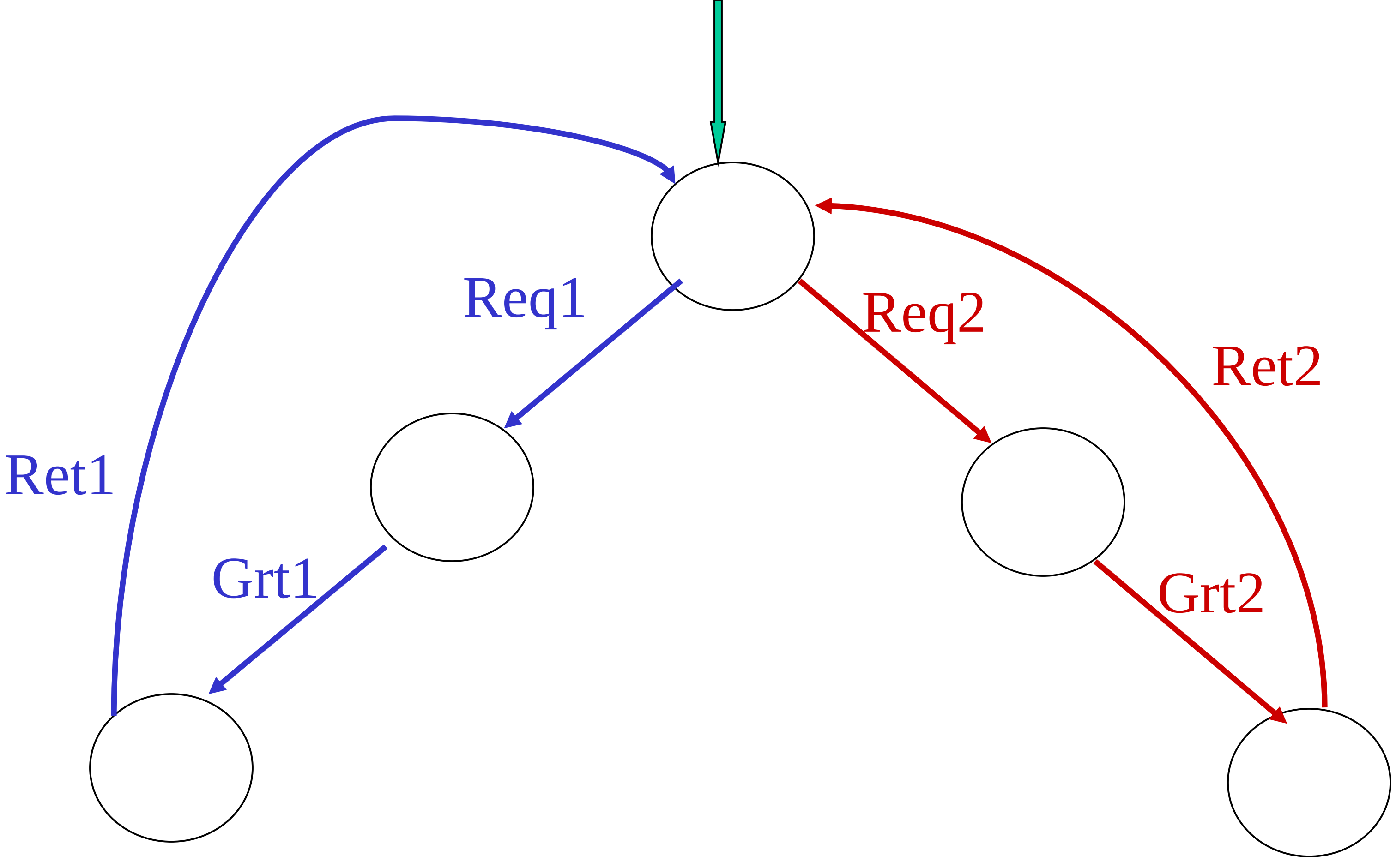
The Application

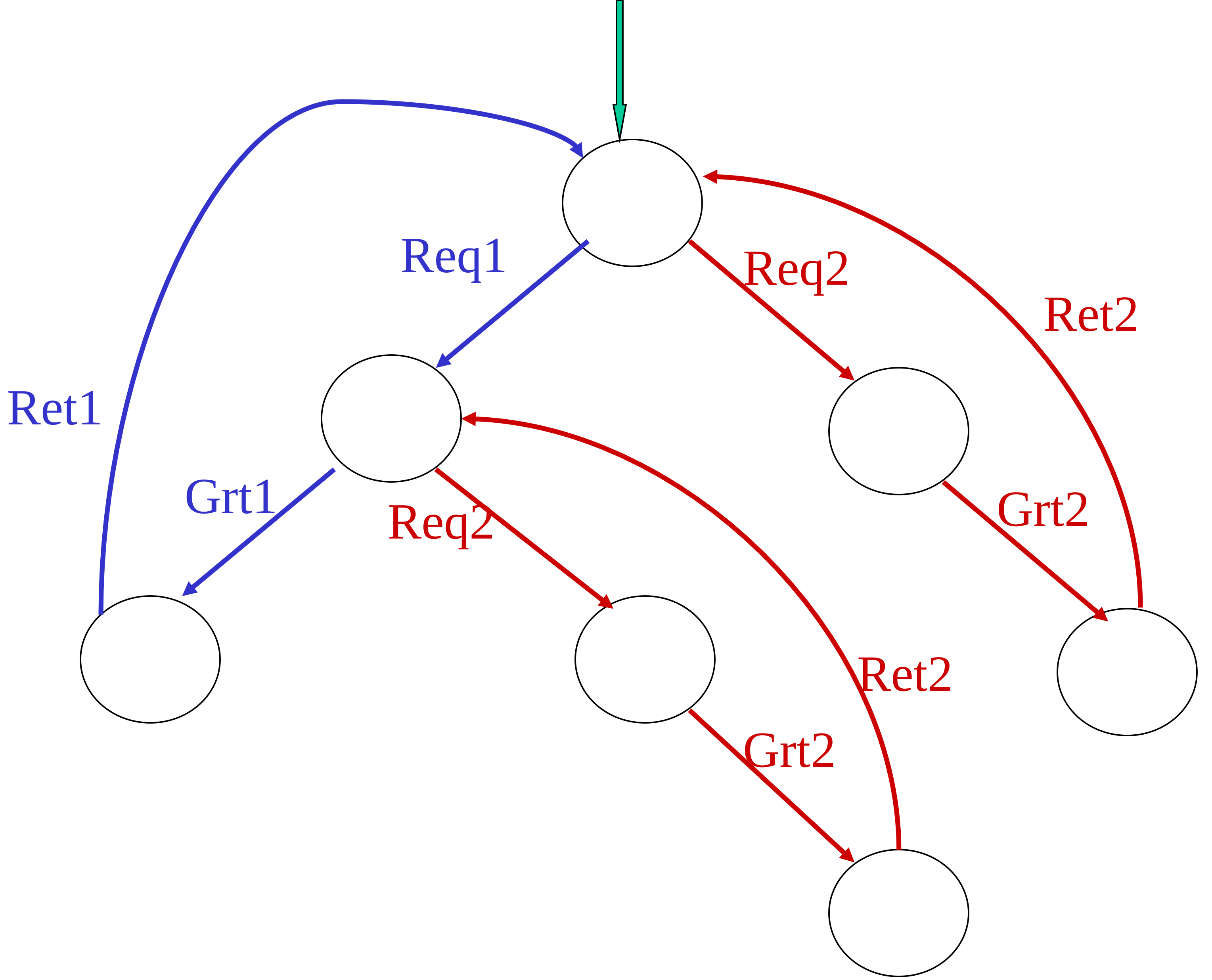
- ***The computations*** of the Kripke structure **K** will be the ***models*** for **LTL** formulas.
- ***The property*** to be verified is captured as an LTL formula φ.
- The modeled system **K** has the property φ ***iff every computation of K is a model of*** φ.
- We need to verify (***model check***) whether:
  - $\mathbf{K} \models \varphi$

An Example

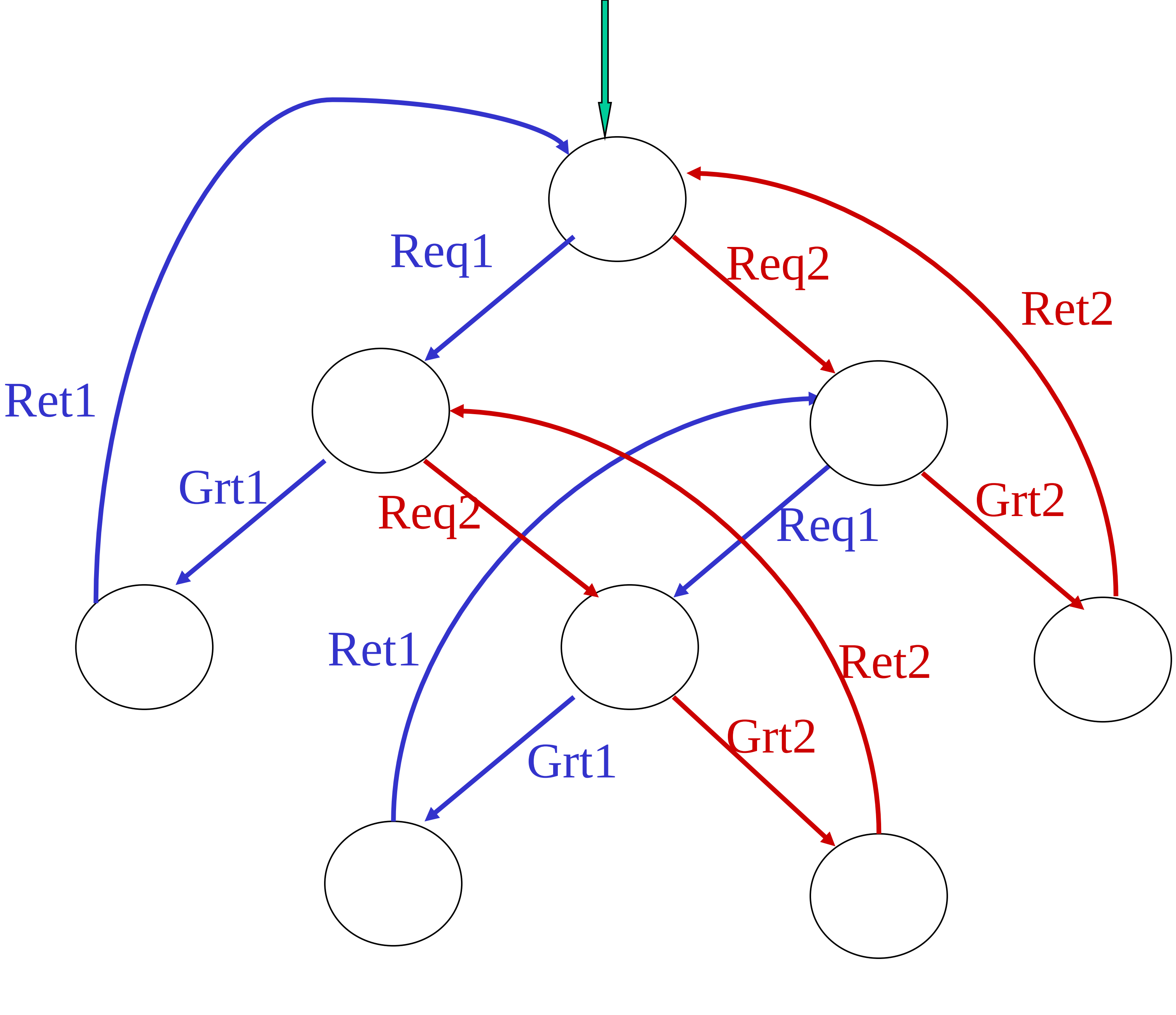


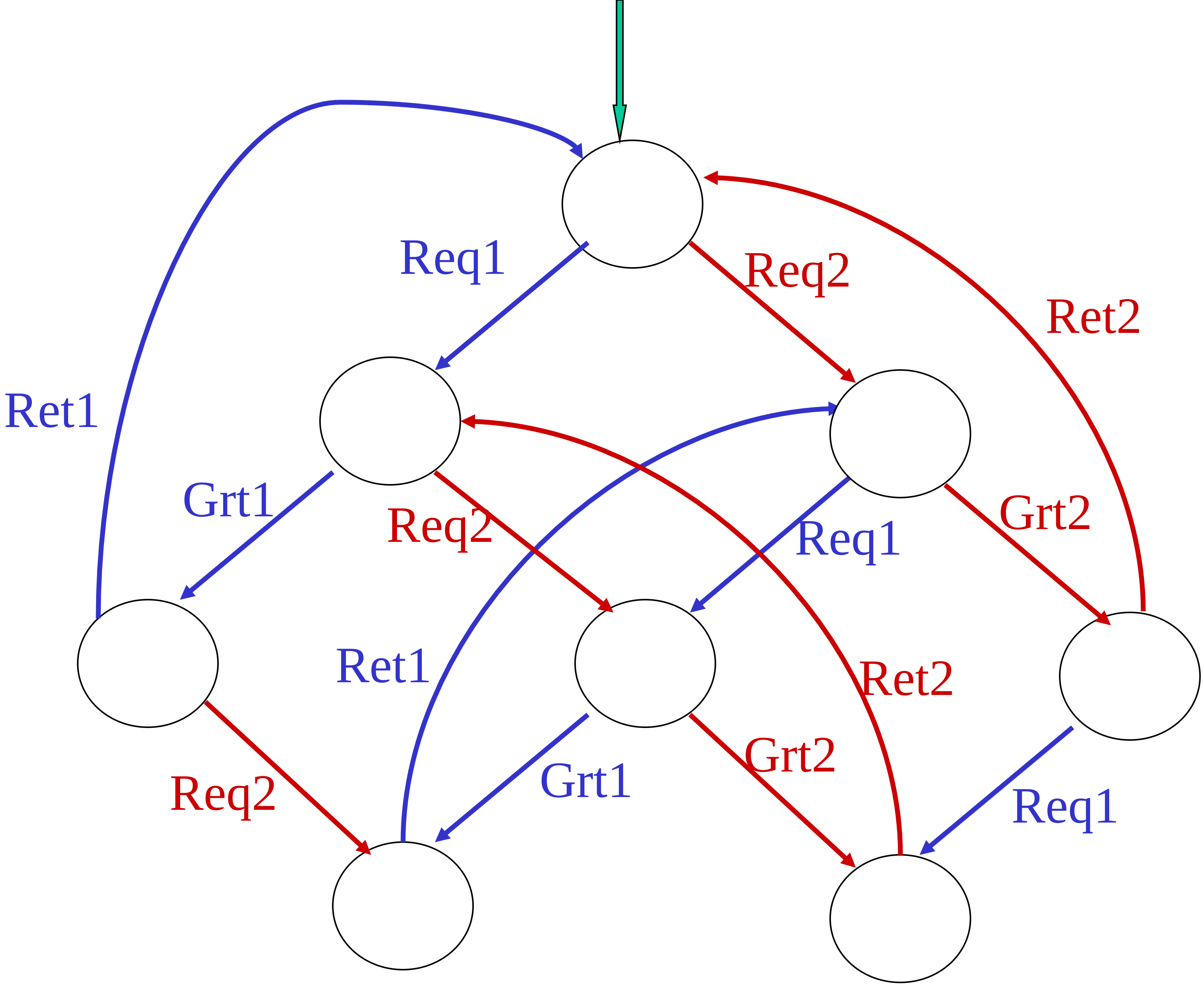

Req1
Req2
Ret2
Ret1
Grt1
Req2
Req1
Grt2
Ret1
Ret2
Req2
Grt1
Grt2
Req1


Req1
Req2
Ret2
Ret1
Grt1
Grt2

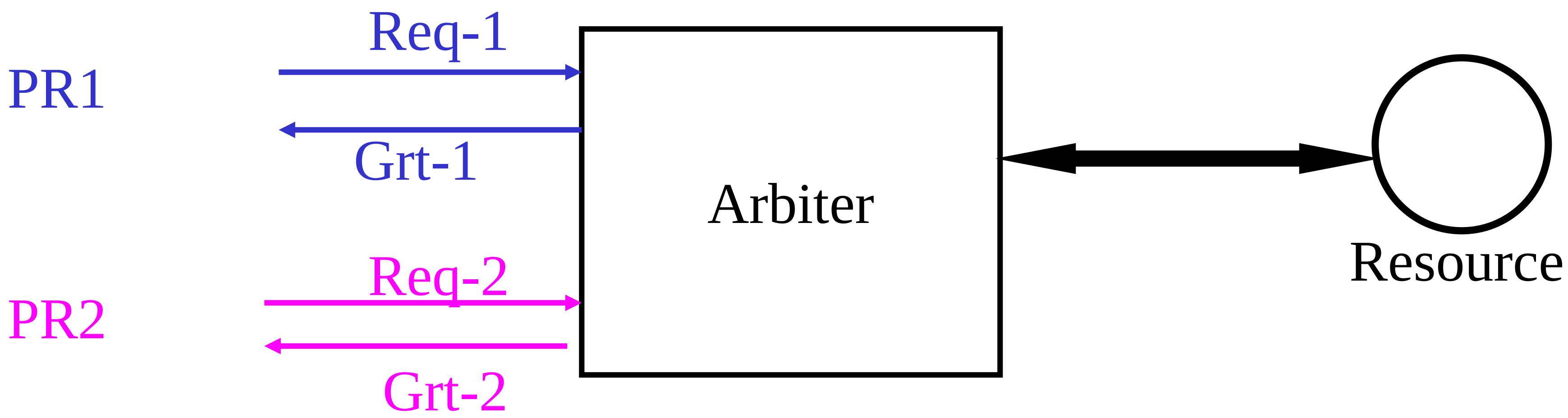


Req1
Req2
Ret2
Ret1
Grt1
Req2
Grt2
Ret2
Grt2


Req1
Req2
Ret2
Ret1
Grt1
Req2
Req1
Grt2
Ret1
Ret2
Grt2
Grt1


Req1
Req2
Ret2
Ret1
Grt1
Req2
Grt2
Req1
Ret1
Ret2
Grt2
Req2
Grt1
Req1

A set of Atomic Propositions



R1 – Process 1 is ***idle***

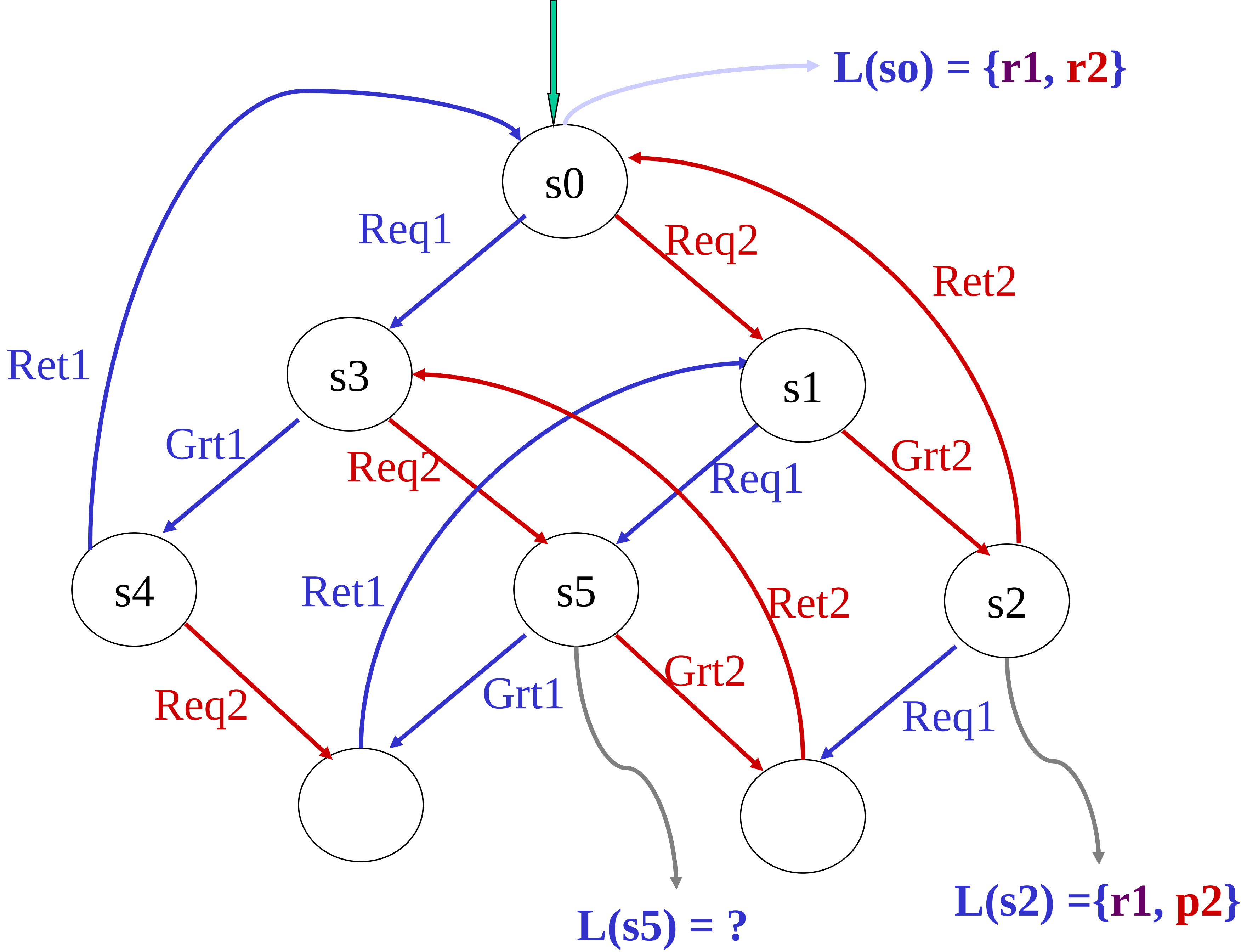
W1– Process 1 is ***waiting***

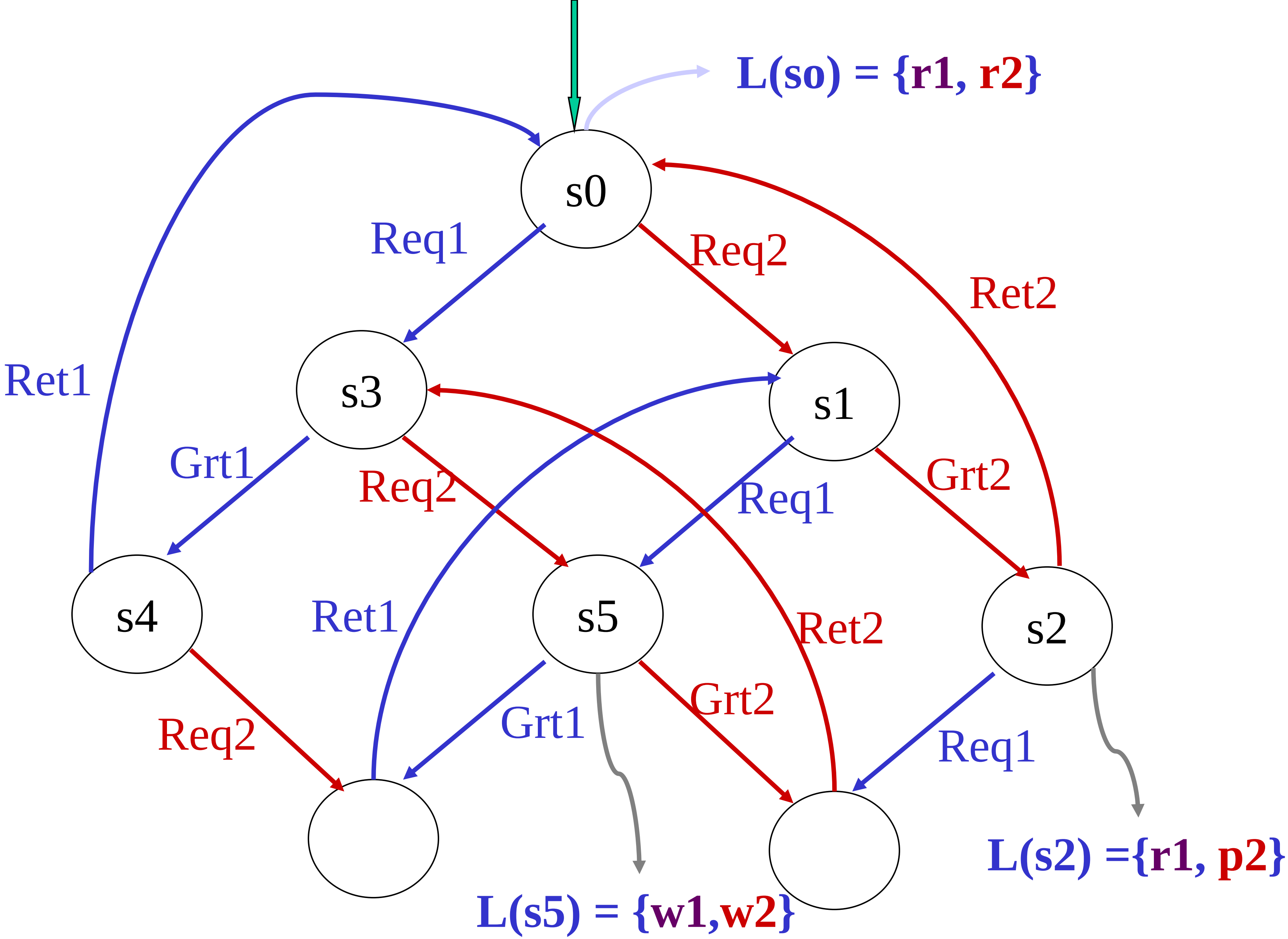
P1 – Process 1 is ***using*** the resource.

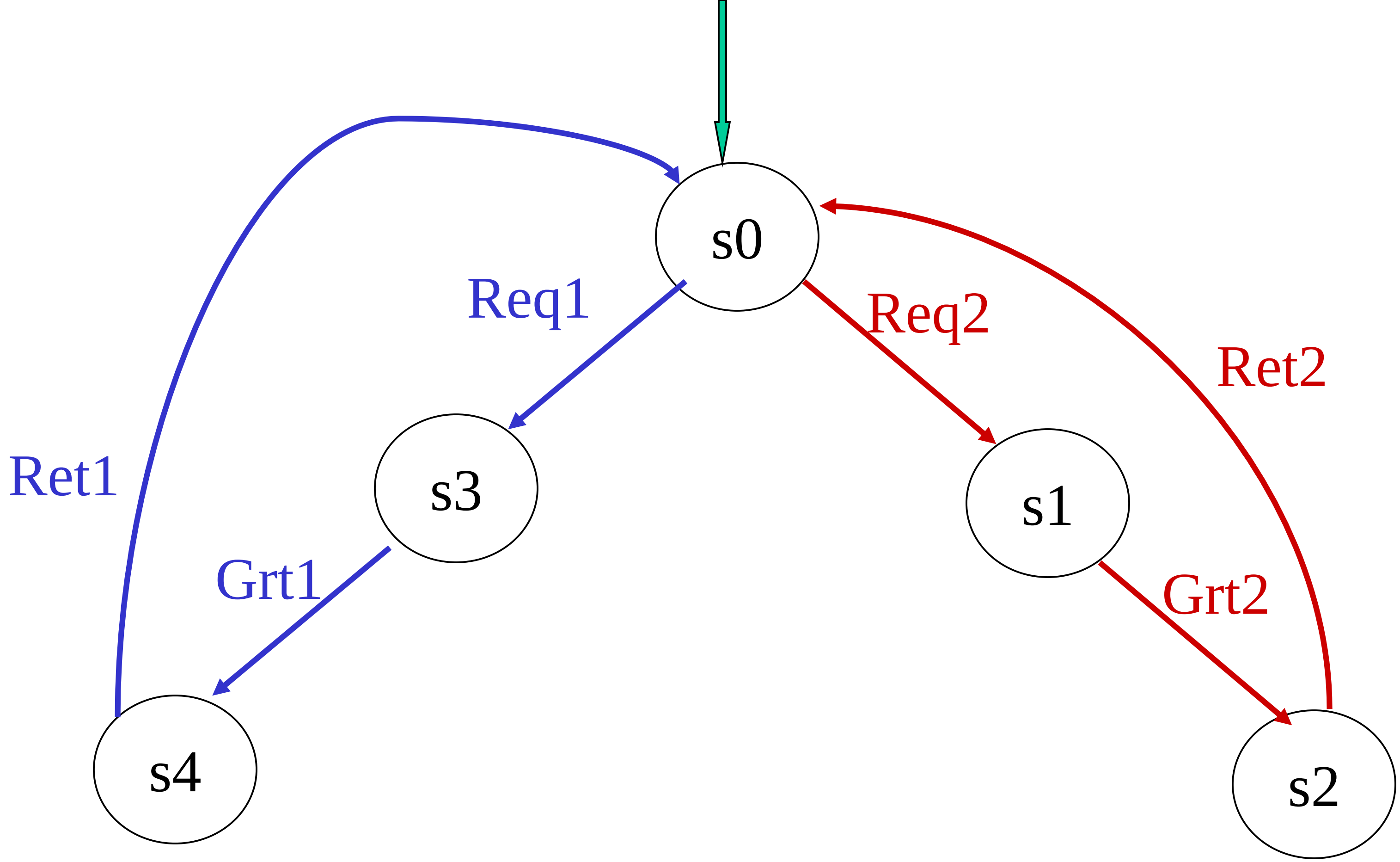
AP = { R1, W1, P1, R2, W2, P2}

The context

- Model a system to be verified as a Kripke structure:
  - Transition system **TS = (S, S_0, R)**
  - AP = A finite set of atomic propositions.
    - Basic assertions about the system
  - **L : S** $\longrightarrow$ $\mathbf{2^{AP}}$ = The set of subsets of AP.
  - $\mathbf{p} \in \mathbf{L(s)}$ ---- **p** is true at **s**
  - $\mathbf{p} \notin \mathbf{L(s)}$ ---- **p** is not true at **s**.
- **K = (S, S_0, R, AP, L)** ---- Kripke structure


L(so) = {r1, r2}
s0
Req1
Req2
Ret2
Ret1
s3
s1
Grt1
Req2
Grt2
Req1
s4
Ret1
s5
Ret2
s2
Req2
Grt1
Grt2
Req1
L(s5) = ?
L(s2) ={r1, p2}


L(so) = {r1, r2}
s0
Req1
Req2
Ret2
Ret1
s3
s1
Grt1
Req2
Req1
Grt2
s4
Ret1
s5
Ret2
s2
Req2
Grt1
Grt2
Req1
L(s2) ={r1, p2}
L(s5) = {w1,w2}



s0 s3 s4 s0 s1 s2 s0 s3 …

{r1, r2} {w1, r2} {u1,r2} {r1, r2} {r1,w2} {r1,p2} {r1,r2} {w1,r2}...

Assertions about a computation

s0	s3	s4	s0	s1	s2	s0	s3 …
{r1, r2}	**{w1, r2}**	**{p1,r2}**	**{r1, r2}**	**{r1,w2}**	**{r1,p2}**	**{r1,r2}**	**{w1,r2}...**

- If at some stage Process 1 is **waiting** then at some later stage it is **printing** (i.e. using the resource).
- **At no stage** are both processes using the resource.
- If a process is waiting then it does so **until** it starts to use the resource.
- **There is a stage** at which both processes are waiting.

The Application

- **$K = (S, S_0, R, AP, L)$**
- Every computation (sequence of states) can be viewed as a sequence of subsets of **AP**.
- **$s_0\ s_1\ s_2 \ldots.$** ---- **$L(s_0)\ L(s_1)\ L(s_2)$** ….
- These **AP-computations** will be the models for the formulas of LTL.

- Verification :
  - Every **AP**-computation of **K** is a model of φ

Linear Time Temporal Logic (LTL)

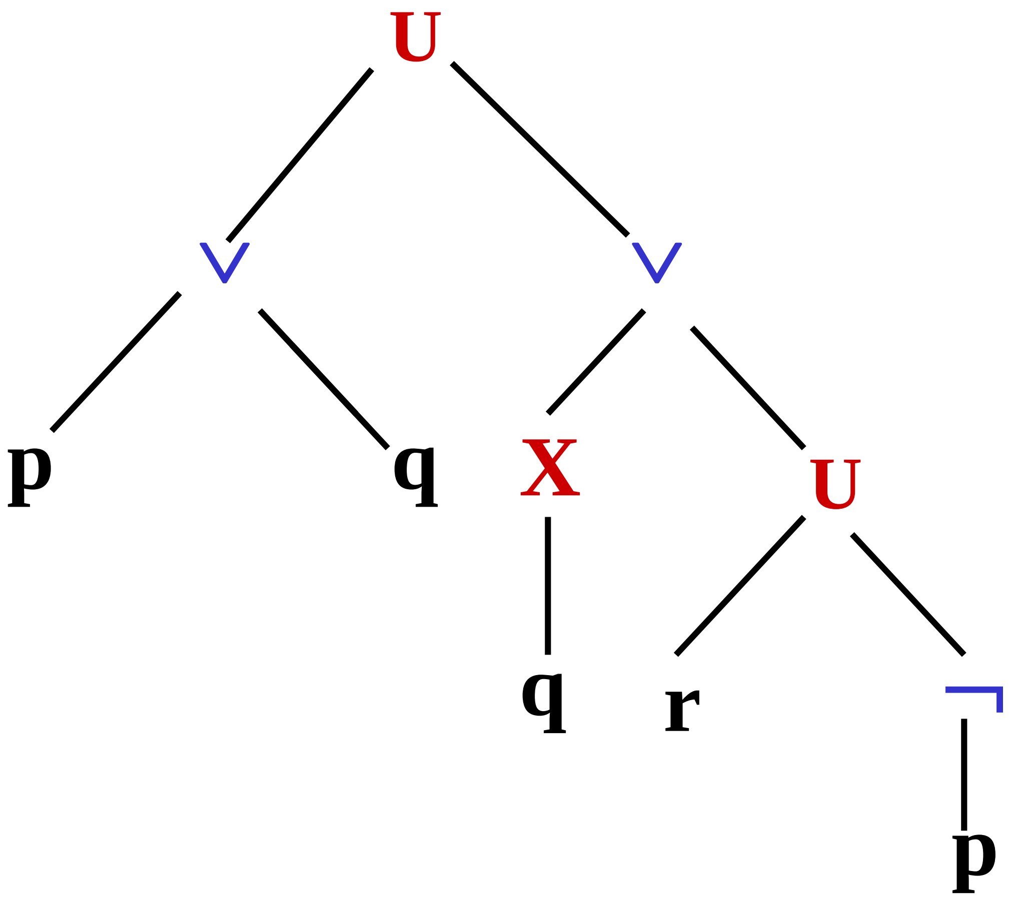
- Syntax :
  - **AP = {p_0, p_1, …p_n}**, a finite set of Atomic Propositions.
- Formulas :
  - Every **p_i** in AP is a ***LTL formula***.
  - If φ is a formula then $\neg\varphi$ is a ***LTL formula***.
  - If φ_1 and φ_2 are formulas then $(\varphi_1 \vee \varphi_2)$ is a ***LTL formula***.
  - If φ is a formula then $\mathbf{X}\,\varphi$, $\mathbf{F}\,\varphi$ and $\mathbf{G}\,\varphi$ are ***LTL formulae*** (**Next**, **Eventually**, **Always**).
  - If φ_1 and φ_2 are formulas then $(\varphi_1\ \mathbf{U}\ \varphi_2)$ is a ***LTL formula*** (Until).

Formulas

$\mathbf{LTL} ::= p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\,\varphi \mid \mathbf{F}\,\varphi \mid \mathbf{G}\,\varphi \mid \varphi_1\,\mathbf{U}\,\varphi_2$

- p ; $p \vee q$; $(\neg p \vee q) \vee \neg(r \vee q)$
- $X q$; $X(p \vee q)$; $X((\neg p \vee q) \vee X\neg(r \vee q))$
- $(p \vee q)\, U\, (X r \vee (\neg q\, U\, (X \neg p)))$

$$(p \vee q)\ \mathbf{U}\ (\mathbf{X}\, q \vee (r\ \mathbf{U} \neg p))$$



Semantics

- **AP** = A finite set of atomic propositions.
- $\boldsymbol{\Sigma = 2^{AP}}$ = The set of subsets of **AP**
- $\boldsymbol{AP = \{ p, q, r \}}$
- $\boldsymbol{\Sigma = \{ \phi, \{p\}, \{q\}, \{r\}, \{p,q\}, \{p,r\}, \{q,r\}, \{p,q,r\} \}}$

- $\boldsymbol{\Sigma^\omega}$ = The set of ***infinite sequences*** over $\boldsymbol{\Sigma}$.

Semantics

- **$AP = \{p, q, r\}$ $\Sigma = 2^{AP}$**
- **$\Sigma = \{\phi, \{ p \}, \{ q \}, \ldots., \{ p, q, r \} \}$**

$$\begin{array}{llllll} \sigma: & \{ p,r \} & \{ q \} & \varnothing & \{ p, q, r \} & \{ r \}\ldots \\ & | & | & | & | & | \\ \textit{path}: & 0 \longrightarrow & 1 \longrightarrow & 2 \longrightarrow & 3 \longrightarrow & 4 \;\ldots \end{array}$$

- At stage **0** of **σ**, **p** and **r** are true but not **q**; at stage **2** of **σ** no member of **AP** is true….

Semantics

- Σ^{ω} = The set of infinite sequences over Σ.
- $\sigma \in \Sigma^{\omega}$ --- A model
- $\sigma(i)$ ---- i-th position of σ
- $\{p\}$ $\{q,r\}$ $\varnothing$ $\{r, q\}$ $\{p, q, r\}$..............
- 0 1 2 3 4 ....
- $\sigma(0) = \{p\}$ $\sigma(2) = \varnothing$ $\sigma(3) = ?$

Semantics

- **AP** $\Sigma = 2^{AP}$
- Σ^{ω} = The set of infinite sequences over Σ.
- $\sigma \in \Sigma^{\omega}$ --- A model
- $\sigma(i)$ ---- **i**-th position of σ
- φ, a formula.
- $\sigma(i) \models \varphi$
  - $\sigma(i)$ ***satisfies*** φ
  - φ is true in the **i**-th position of σ

Semantics

LTL ::= $p \mid \neg \varphi \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\,\varphi \mid \mathbf{F}\,\varphi \mid \mathbf{G}\,\varphi \mid \varphi_1\,\mathbf{U}\,\varphi_2$

- $\sigma = \Gamma_0\ \Gamma_1\ \Gamma_2 \ldots\ldots \Gamma_i\ \ \Gamma_{i+1} \ldots\ldots\ldots\ldots$
- Each Γ_j is a subset of **AP**.
- $\boxed{\sigma(i) \models p \ \textit{iff}\ \ p \in \Gamma_i}$

Semantics

LTL ::= $p \mid \neg \varphi \mid \varphi_1 \vee \varphi_2 \mid X\,\varphi \mid F\,\varphi \mid G\,\varphi \mid \varphi_1\, U\, \varphi_2$

- **AP = {p, q, r}**
- **σ = {p,q} {r} ∅ {q, r} {p, q, r} …..**

0 1 2 3 4

- **σ(0)** ***satisfies*** **q**
- **σ(1)** ***satisfies*** **r**
- **σ(2)** does ***not*** *satisfy* **q !**

Semantics

LTL ::= p | $\neg \varphi$ | $\varphi_1 \vee \varphi_2$ | $X \varphi$ | $F \varphi$ | $G \varphi$ | $\varphi_1 U \varphi_2$

$\sigma = \Gamma_0 \; \Gamma_1 \; \Gamma_2 \; \ldots.. \; \Gamma_i \; \Gamma_{i+1} \; \ldots\ldots\ldots..$

Each Γ_j is a subset of **AP**.

- $\boxed{\sigma(i) \models \neg \varphi \;\; \textit{iff} \;\; \sigma(i) \not\models \varphi}$

Semantics

LTL ::= *p* | ¬ φ | φ₁ ∨ φ₂ | X φ | F φ | G φ | φ₁ U φ₂

$\sigma = \quad \Gamma_0 \ \Gamma_1 \ \Gamma_2 \ \ldots\ldots \ \Gamma_i \quad \Gamma_{i+1} \ldots\ldots\ldots\ldots$

Each Γ_j is a subset of **AP**.

- $\sigma(i) \models \varphi_1 \vee \varphi_2$ ***iff*** $\sigma(i) \models \varphi_1$ ***OR*** $\sigma(i) \models \varphi_2$

Semantics

LTL ::= *p* | $\neg \varphi$ | $\varphi_1 \vee \varphi_2$ | **X** φ | **F** φ | **G** φ | φ_1 **U** φ_2

AP = {p, q, r}

σ **= {p, q} {r}** $\varnothing$ **{q, r} {p, q, r} ….**

0 1 2 3 4

- **σ(0)** ***satisfies*** $\neg$**r** ; **σ(0)** does ***not satisfy*** **r**
- **σ(1)** ***satisfies*** **p** $\vee$ **r** ; **σ(1)** ***satisfies*** **r**
- **σ(2)** ***satisfies*** $\neg$**(p** $\vee$ **r)** **?**

Semantics

LTL ::= $p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\,\varphi \mid \mathbf{F}\,\varphi \mid \mathbf{G}\,\varphi \mid \varphi_1\,\mathbf{U}\,\varphi_2$

AP = {p, q, r}

σ = {p, q} {r} $\emptyset$ {q, r} {p, q, r} …..

0 1 2 3 4

- **σ(2)** ***satisfies*** $\neg$(**p** $\vee$ **r**) ? ***Yes!***
- **σ(2)** does ***not satisfy*** **p** $\vee$ **r**

Semantics

LTL ::= *p* | ¬ φ | $\varphi_1 \vee \varphi_2$ | X φ | F φ | G φ | φ_1 U φ_2

- **$\sigma = \Gamma_0 \quad \Gamma_1 \quad \Gamma_2 \ldots\ldots \Gamma_i \quad \Gamma_{i+1} \ldots\ldots\ldots\ldots$**

$$\begin{array}{cc} \Gamma_i & \Gamma_{i+1} \\ | & | \\ X\varphi & \varphi \end{array}$$

- **$\sigma(i) \models X\,\varphi$ *iff* $\sigma(i+1) \models \varphi$**

Semantics

LTL ::= *p* | $\neg\ \varphi$ | $\varphi_1 \vee \varphi_2$ | X φ | F φ | G φ | φ_1 U φ_2

AP = {p, q, r}

σ = {p,q} {r} ∅ {q,r} {p,q,r} …..

0 1 2 3 4

- **σ(2)** ***satisfies*** **X r** ; **σ(3)** ***satisfies*** **r**
- **σ(0)** ***satisfies*** **X(p ∨ r)** ; **σ(1)** ***satisfies*** **r**
- **σ(1)** does ***not satisfy*** **X(p ∨ r)**
  - **σ(2)** does ***not satisfy*** **p ∨ r**

Semantics

LTL ::= ***p*** **|** $\neg \varphi$ **|** $\varphi_1 \vee \varphi_2$ **|** **X** φ **|** **F** φ **|** **G** φ **|** φ_1 **U** φ_2

AP = {p, q, r}

σ = {p,q}   {r}   ∅   {q,r}   {p,q,r} …..

0   1   2   3   4

- **σ(1)** ***satisfies*** **X(X ¬ p)**   ***iff***
  - **σ(2)** ***satisfies*** **X ¬ p**   ***iff***
  - **σ(3)** ***satisfies*** **¬ p**   ***iff***
  - **σ(3)** does ***not satisfy*** **p**

Semantics

LTL ::= p | $\neg\,\varphi$ | $\varphi_1 \vee \varphi_2$ | $X\,\varphi$ | $F\,\varphi$ | $G\,\varphi$ | $\varphi_1\, U\, \varphi_2$

- $\sigma = \Gamma_0 \quad \Gamma_1 \quad \Gamma_2 \ldots\ldots \underset{\underset{F\varphi}{|}}{\Gamma_i} \ldots\ldots \Gamma_{j-1}\; \underset{\underset{\varphi}{|}}{\Gamma_j} \ldots\ldots\ldots$

- $\sigma(i) \models F\,\varphi$ ***iff*** $\sigma(j) \models \varphi$ ***for some*** $j \geq i$

Semantics

LTL ::= *p* | $\neg \varphi$ | $\varphi_1 \vee \varphi_2$ | X φ | F φ | G φ | φ_1 U φ_2

AP = {p, q, r}

σ = {p,q}   {r} $\varnothing$ {q,r}   {p,q,r} .....

0   1   2   3   4

- **σ(0)** ***satisfies*** **F(X p)**   this is true since
  - **σ(3)** ***satisfies*** **X p**   ***iff***
  - **σ(4)** ***satisfies*** **p**   is true since

Semantics

LTL ::= *p* | ¬ φ | $\varphi_1 \vee \varphi_2$ | X φ | F φ | G φ | φ_1 U φ_2

- $\sigma = \Gamma_0 \quad \Gamma_1 \quad \Gamma_2 \ldots\ldots \Gamma_i \quad \Gamma_{i+1} \ldots\ldots\ldots\ldots \Gamma_j \ldots\ldots$

$\qquad\qquad\qquad\qquad\qquad\qquad G\varphi \quad \varphi \quad \ldots\ \varphi \ldots\ldots \varphi \ \ldots\ldots$

- $\sigma(i) \models G\,\varphi$ ***iff*** $\sigma(j) \models \varphi$ ***for all*** $j \geq i$

Semantics

$\mathbf{LTL} ::= p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\,\varphi \mid \mathbf{F}\,\varphi \mid \mathbf{G}\,\varphi \mid \varphi_1\, \mathbf{U}\, \varphi_2$

$$\begin{array}{llllllll}
\sigma = \Gamma_0 & \Gamma_1 \ldots & \Gamma_i & \Gamma_{i+1} \ldots & \Gamma_{k-1} & \Gamma_k \ldots \\
 & & \varphi_1 \ldots & \varphi_1 \ldots & \varphi_1 & \varphi_2
\end{array}$$

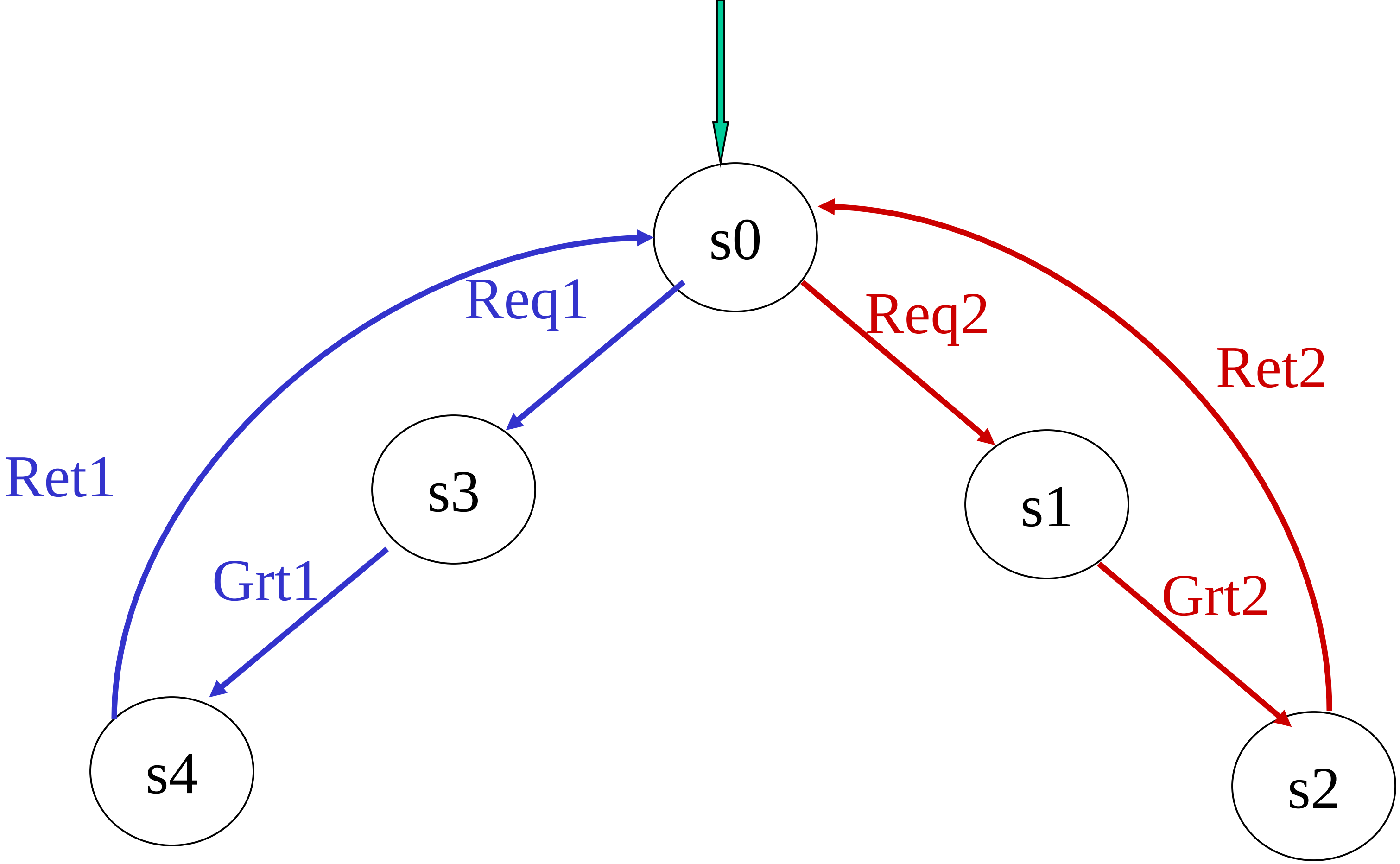
- $\sigma(i) \models \varphi_1\, \mathbf{U}\, \varphi_2$ ***iff*** there exists $\mathbf{k \geq i}$ s.t.
  - $\sigma(k) \models \varphi_2$
  - $\sigma(j) \models \varphi_1$ for every $\mathbf{i \leq j < k}$

Semantics

LTL ::= p | $\neg\varphi$ | $\varphi_1 \vee \varphi_2$ | $X\varphi$ | $F\varphi$ | $G\varphi$ | $\varphi_1 \, U \, \varphi_2$

- **k** could be arbitrarily greater than **i**.
- **k = i** is allowed and there is **no** $i \le j < k$

- $\sigma(i) \models \varphi_1 \, U \, \varphi_2$ ***iff*** there exists $k \ge i$ s.t.
  - $\sigma(k) \models \varphi_2$
  - $\sigma(j) \models \varphi_1$ for every $i \le j < k$



s0 s3 s4 s0 s1 s2 s0 s3 …

{r1,r2} {w1,r2} {p1,r2} {r1,r2} {r1,w2} {r1,p2} {r1,r2} {w1,r2}...

An Example

AP = {r1, w1, p1, r2, w2, p2}

{r1,r2}	**{w1,r2}**	**{p1,r2}**	**{r1,r2}**	**{r1,w2}**	**{r1,p2}**	**{r1,r2}**	**{w1,r2}...**
0	1	2	3	4	5	6	7

- **σ(1)** ***satisfies*** **(r2 U w2)** ;
  - **σ(4)** ***satisfies*** **w2** and
  - **σ(1), σ(2), σ(3)** ***satisfy*** **r2**.

An Example

AP = {r1, w1, p1, r2, w2, p2}

{r1,r2}	**{w1,r2}**	**{p1,r2}**	**{r1,r2}**	**{r1,w2}**	**{r1,p2}**	**{r1,r2}**	**{w1,r2}...**
0	1	2	3	4	5	6	7

- **σ(1)** does ***not satisfy*** **(r2 U p2)** ;
  - **σ(5)** ***satisfies*** **p2** and
  - **σ(1), σ(2), σ(3)** ***satisfy*** **r2**.
  - but **σ(4)** does ***not satisfy*** **r2 !**

An Example

AP = {r1, w1, p1, r2, w2, p2}

{r1,r2} {w1,r2} {p1,r2} {r1,r2} {r1,w2} {r1,p2} {r1,r2} {w1,r2}...

0 1 2 3 4 5 6 7

- **σ(1)** ***does satisfy*** **((r2 ∨ w2) U p2)** ;
  - **σ(5)** ***satisfies*** **p2** and
  - **σ(1), σ(2), σ(3)** ***satisfy*** **r2, hence also (r2 ∨ w2)**.
  - **σ(4)** ***satisfies*** **w2, hence also (r2 ∨ w2) !**

Models

- **AP** $\Sigma^{AP} = 2$
- Σ^{ω} = The set of infinite sequences over Σ.
- $\sigma \in \Sigma^{\omega}$
- φ an **LTL** formula.

- A path σ is a ***model*** of φ ($\sigma \models \varphi$) ***iff***
  - $\sigma(0) \models \varphi$

Validity in LTL

- **AP** $\Sigma^{AP} = 2$
- Σ^{ω} = The set of infinite sequences over Σ.
- $\sigma \in \Sigma^{\omega}$
- φ an **LTL** formula.

- φ is ***LTL-valid*** **(** $\models \varphi$ **)** ***iff for every*** $\sigma \in \Sigma^{\omega}$
  - $\sigma \models \varphi$

Basic LTL Language

We will use the *reduced LTL language*

LTL ::= $p \mid \neg \varphi \mid \varphi_1 \vee \varphi_2 \mid X\, \varphi \mid \varphi_1\, U\, \varphi_2$

- $\varphi_1 \wedge \varphi_2$ --- $\neg(\neg\varphi_1 \vee \neg\varphi_2)$ (**and**)
- $\varphi_1 \supset \varphi_2$ --- $\neg\varphi_1 \vee \varphi_2$ (**implies**)
- $\varphi_1 \equiv \varphi_2$ ---- $(\varphi_1 \supset \varphi_2) \wedge (\varphi_2 \supset \varphi_1)$ (**iff**)
- **AP** = $\{p_1, p_2, \ldots, p_n\}$
- $\top$ ---- $p_1 \vee \neg p_1$ (**true**)

- **Fact : In *every model* σ, *at every* i,**
  - $\sigma(i) \models \top$

Derived Operators

- **LTL ::=** $p \mid \neg\,\varphi \mid \varphi_1 \vee \varphi_2 \mid X\,\varphi \mid \varphi_1\, U\, \varphi_2$
- $F\varphi \equiv (\top\, U\, \varphi)$ (future ; diamond: $\Diamond$)

- **We gave the following semantics** :
  - $\sigma(i) \models F\varphi$ ***iff*** **there exists** $k \geq i$ such that $\sigma(k) \models \varphi$.

Derived Operators

- **We gave the following semantics :**
  - $\boldsymbol{\sigma(i)} \models \mathbf{F\varphi}$ ***iff*** **there exists k** $\geq$ **i** such that $\boldsymbol{\sigma(k)} \models \varphi$.

Proof of $\mathbf{F\varphi \equiv (\top\ U\ \varphi)}$

$\boldsymbol{\sigma(i)} \models (\top\ \mathbf{U}\ \varphi)$ ***iff***

$\exists\, j \geq i,\ \boldsymbol{\sigma(j)} \models \varphi$ and $\forall i \leq k < j,\ \boldsymbol{\sigma(k)} \models \top$ ***iff***

$\exists\, j \geq i,\ \boldsymbol{\sigma(j)} \models \varphi$ ***iff***

$\boldsymbol{\sigma(i)} \models \mathbf{F\varphi}$

Derived Operators

- **LTL ::=** $p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\,\varphi \mid \varphi_1\, \mathbf{U}\, \varphi_2$
- $\mathbf{F}\varphi \equiv (\top\, \mathbf{U}\, \varphi)$
- $\mathbf{G}\varphi \equiv \neg\mathbf{F}\neg\varphi$ (invariant; box: $\Box$)

- **We gave the following semantics :**
  - $\sigma(i) \models \mathbf{G}\,\varphi$ ***iff*** **for every** $k \geq i$,
    $\sigma(k) \models \varphi$.

Derived Operators

- **LTL ::=** $p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid X\,\varphi \mid \varphi_1\, U\, \varphi_2$
- $(\psi\, R\, \varphi) \equiv \neg(\neg\psi\, U\, \neg\varphi)$ (Releases)
- $G\,\varphi \equiv (\bot\, R\, \varphi)$

– $\sigma(i) \models (\psi\, R\, \varphi)$ ***iff***

for each $k \geq i$ (*if* for each $i \leq j < k$ $\sigma(j) \not\models \psi$ then $\sigma(k) \models \varphi$)

Derived Operators

- **LTL ::=** $p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \mathbf{X}\,\varphi \mid \varphi_1\, \mathbf{U}\, \varphi_2$
- $(\psi\ \mathbf{R}\ \varphi) \equiv \neg(\neg\psi\ \mathbf{U}\ \neg\varphi)$ (Releases)
- $\mathbf{G}\,\varphi \equiv (\perp \mathbf{R}\ \varphi)$

– $\sigma(i) \models (\psi\ \mathrm{R}\ \varphi)$ ***iff***

for each $k \geq i$ (for some $i \leq j < k$ $\sigma(j) \models \psi$ or $\sigma(k) \models \varphi$)

Derived Operators

- **LTL ::= p | $\neg\varphi$ | $\varphi_1 \vee \varphi_2$ | $\mathbf{X}\,\varphi$ | $\varphi_1\,\mathbf{U}\,\varphi_2$**
- **$(\psi\ \mathbf{W}\ \varphi)$** (Unless)

Give the semantics according to the following intuition:

- **$(\psi\ \mathbf{W}\ \varphi)$**: if ψ must be true unless φ occurs (notice that φ may never occur).

Show that: $(\psi\ \mathbf{W}\ \varphi) \equiv \mathbf{G}\,\psi \vee (\psi\ \mathbf{U}\ \varphi)$

Show that: $\neg(\psi\ \mathbf{U}\ \varphi) \equiv ((\neg\varphi \wedge \psi)\ \mathbf{W}\ (\neg\psi \wedge \neg\varphi))$

Derived Operators

- **LTL ::= p | $\neg\,\varphi$ | $\varphi_1 \vee \varphi_2$ | X φ | φ_1 U φ_2**
- **(ψ B φ)** (Before)

Give the semantics according to the following intuition:

- **(ψ B φ)**: if φ ever occurs, then ψ must occur before φ.

Show that: **$(\psi \text{ B } \varphi) \equiv \neg\,(\neg\psi \text{ U } \varphi)$**

Some important validities

- $(\psi\ \mathbf{U}\ \varphi) \equiv \varphi \vee (\ \psi \wedge \mathbf{X}\ (\psi\ \mathbf{U}\ \varphi)\ )$
- $(\psi\ \mathbf{R}\ \varphi) \equiv \varphi \wedge (\ \psi \vee \mathbf{X}\ (\psi\ \mathbf{R}\ \varphi)\ ) \equiv$

  $\equiv (\varphi \wedge \psi) \vee (\ \varphi \wedge \mathbf{X}\ (\psi\ \mathbf{R}\ \varphi)\ )$
- $\mathbf{F}\varphi \ \equiv\ \varphi \vee \mathbf{X}\ \mathbf{F}\ \varphi$
- $\mathbf{G}\varphi \ \equiv\ \varphi \wedge \mathbf{X}\ \mathbf{G}\ \varphi$

LTL: Some examples

- **Safety**: "it *never happens* that both A and B are printing at the same time"

  $$\mathbf{G}(\neg (P_A \wedge P_B))$$

- **Liveness:** "*whenever* A is waiting, it will *eventually* print in the future"

  $$\mathbf{G}(W_A \supset \mathbf{F}\, P_A)$$

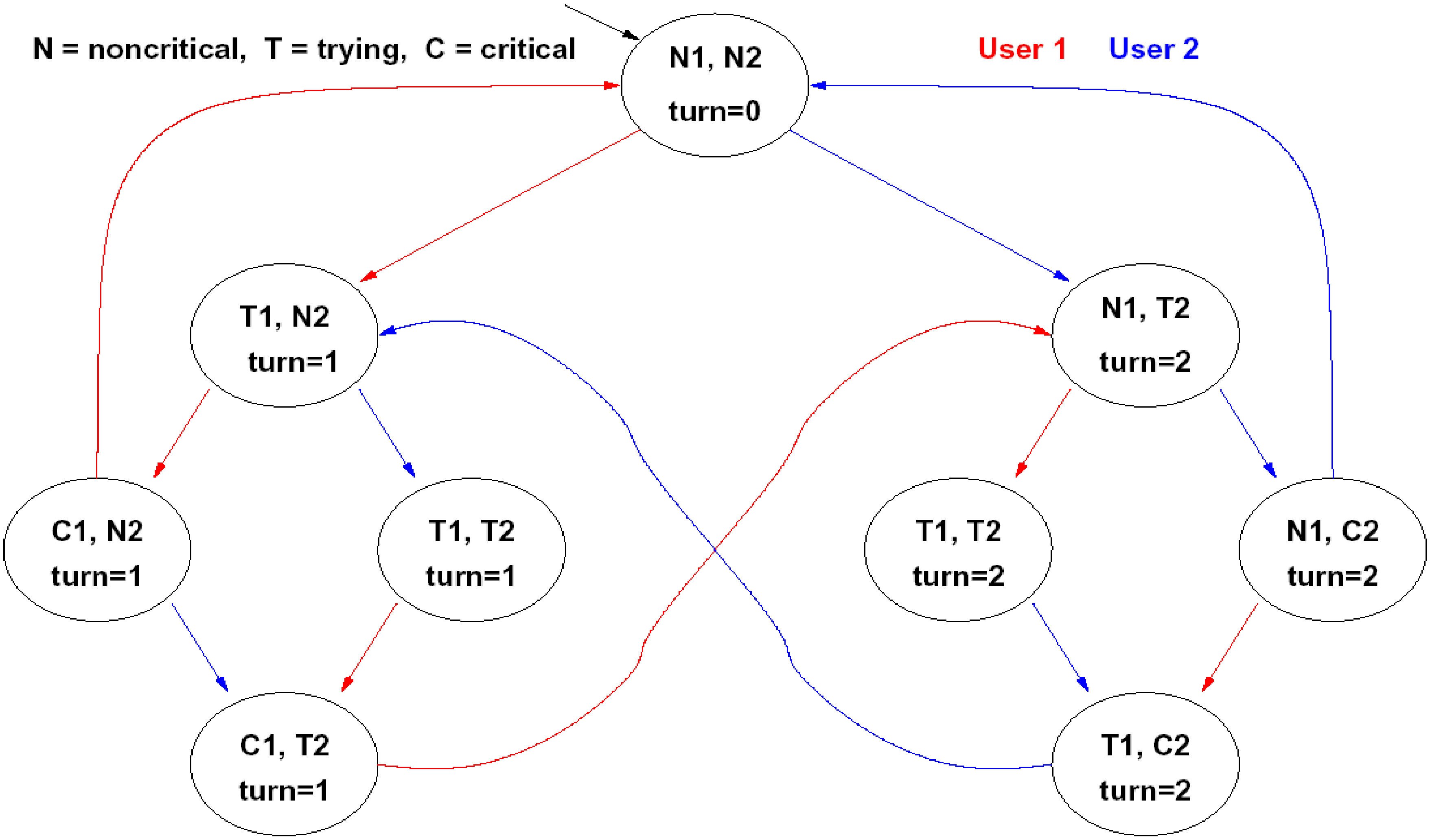
- **Fairness:** "A is *infinitely often* idle"

  $$\mathbf{GF}\, R_A$$

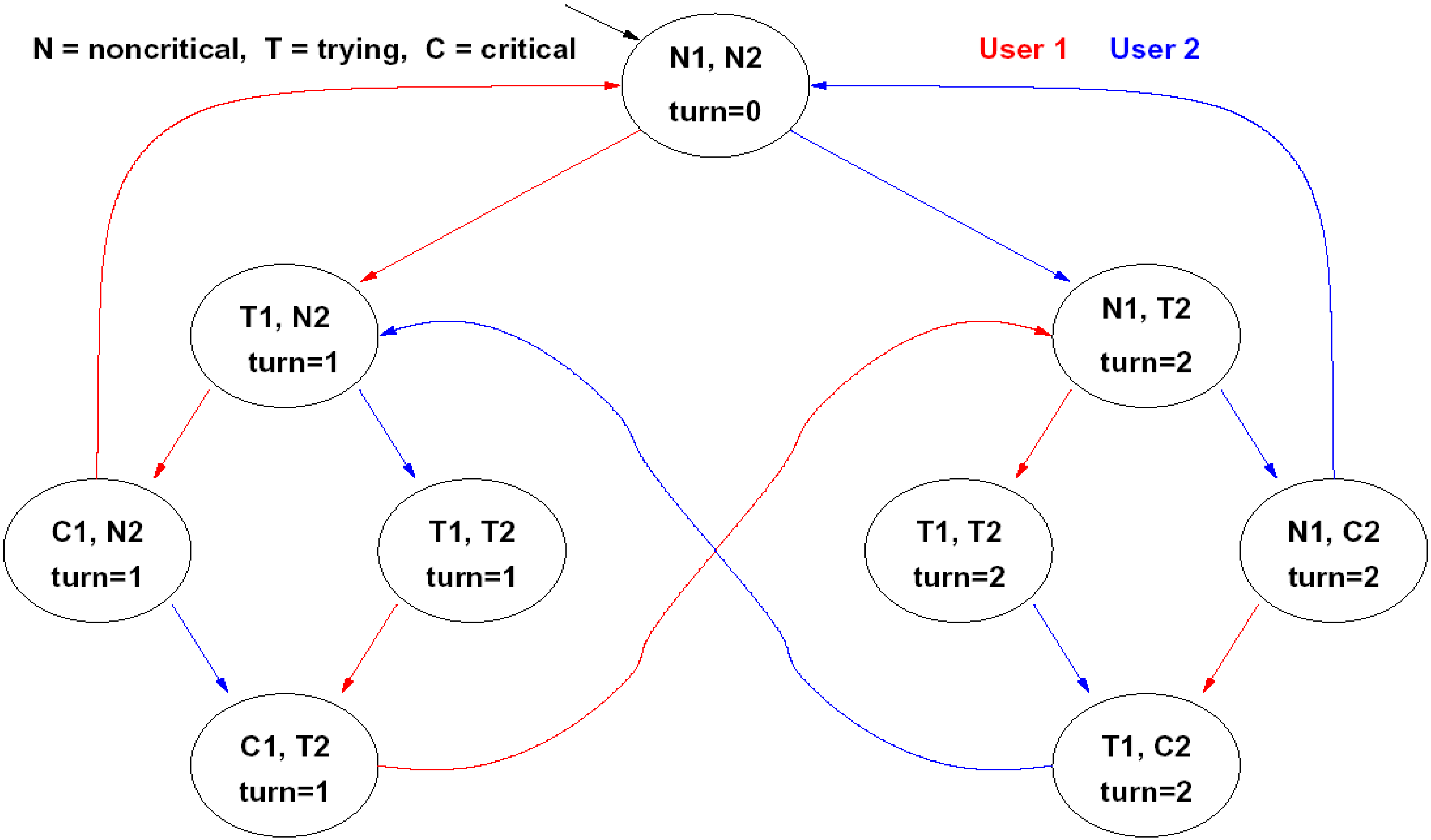
- **Strong fairness:** "if A is *infinitely often* waiting, then it will *infinitely often* printing"

  $$\mathbf{GF}\, W_A \supset \mathbf{GF}\, P_A$$

Example: mutual esclusion



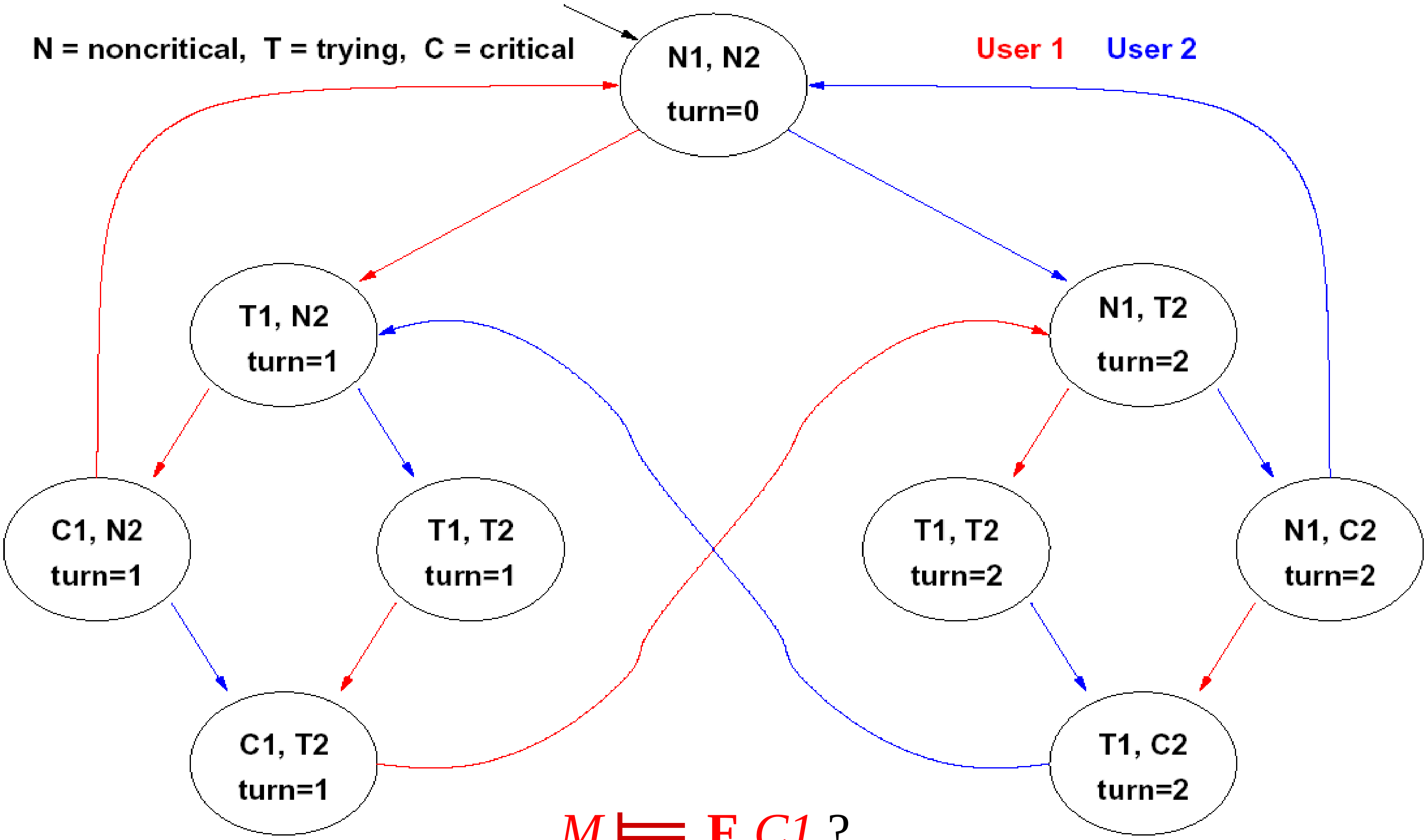
Example: mutual esclusion (safety)



$M \models \mathbf{G}\, \neg(C1 \wedge C2)$? [$M \models \neg\, \mathbf{F}\, (C1 \wedge C2)$?]

YES: There is no reachable state in which both C1 and C2 hold!

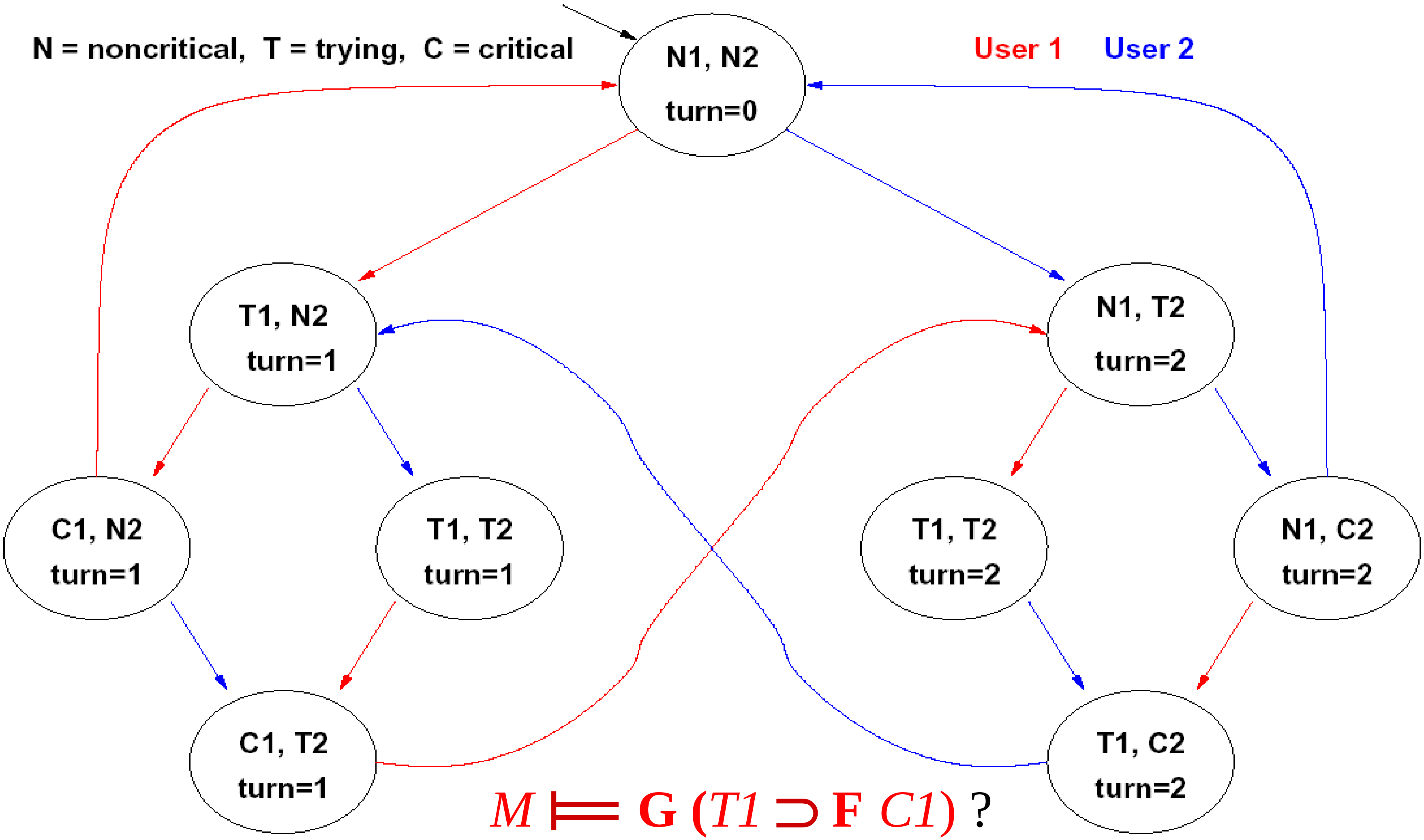
Example: mutual exclusion (liveness)



$M \models \mathbf{F}\ C1$?

NO: there is an infinite (cyclic) solution in which C1 never holds!

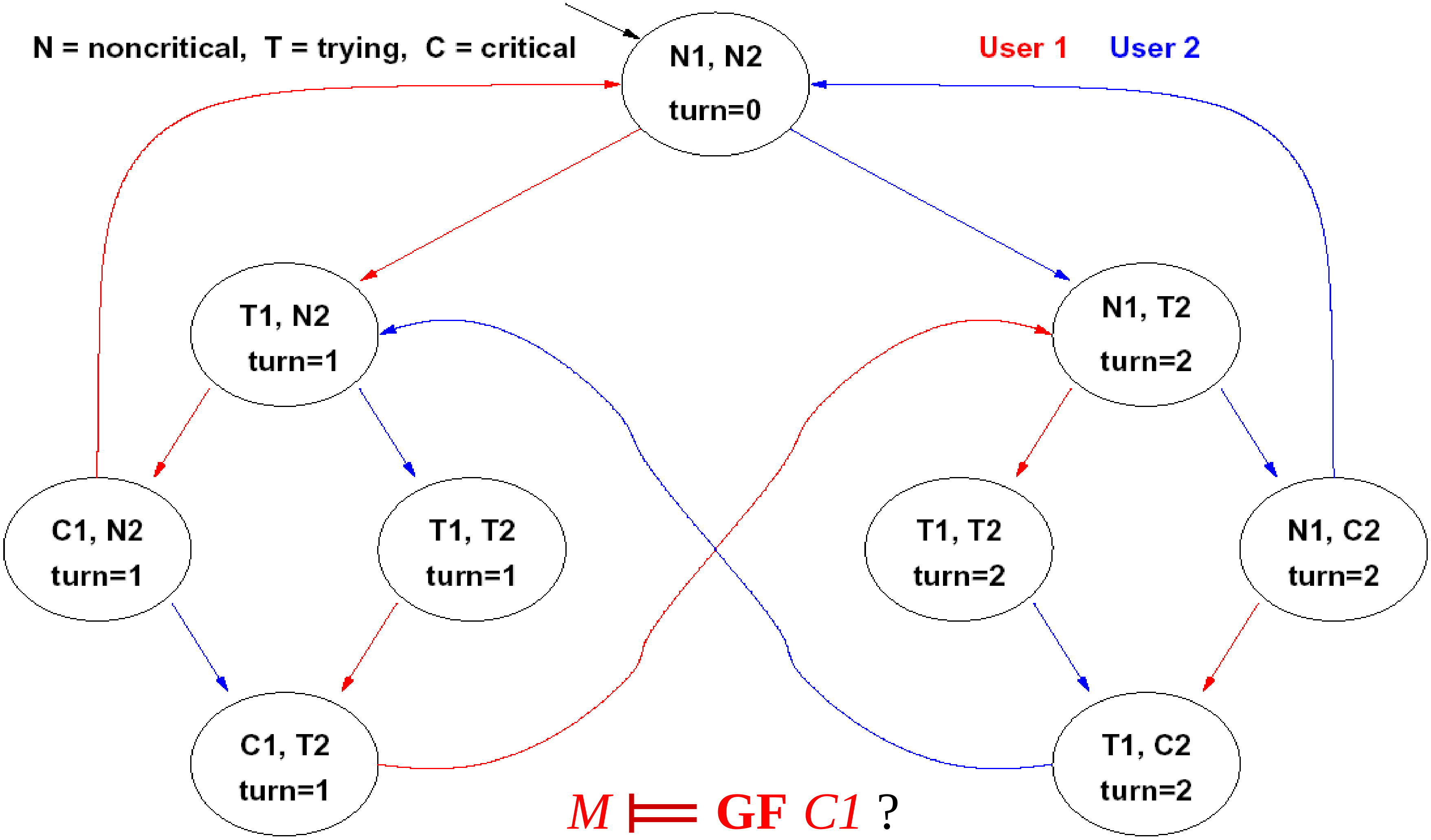
Example: mutual exclusion (liveness)



$M \models \mathbf{G}\,(T1 \supset \mathbf{F}\ C1)$?

YES: every path starting from each state where *T1* holds passes through a state where *C1* holds!

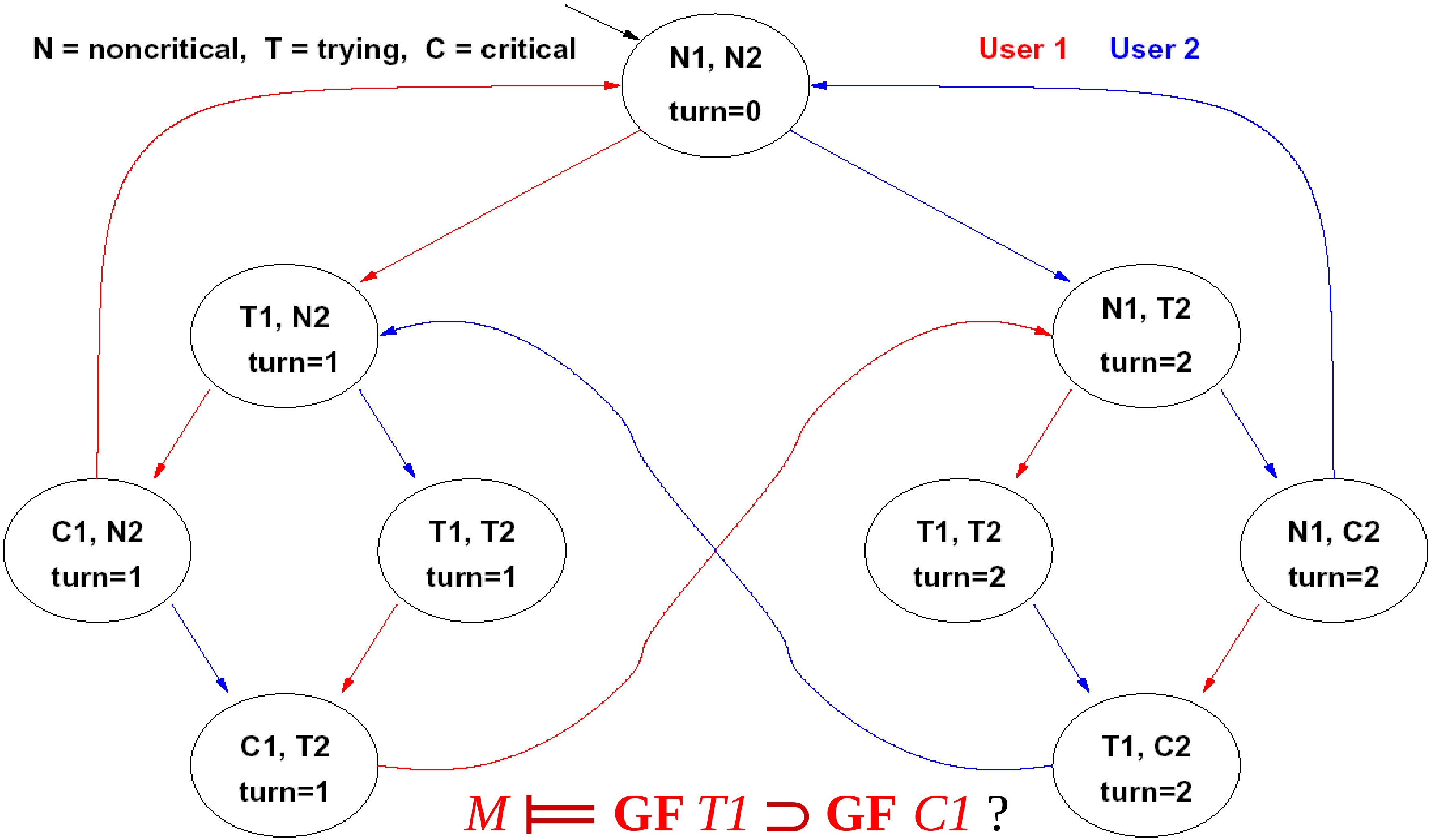
Example: mutual exclusion (fairness)



$M \models \mathbf{GF}\ C1$?

NO: e.g., in the initial state, there is an infinite (cyclic) solution in which *C1* never holds!

Example: mutual exclusion (strong fairness)



$M \models \mathbf{GF}\ T1 \supset \mathbf{GF}\ C1$?

YES: every path which visits $T1$ infinitely often also visits $C1$ infinitely often (see liveness prop. in previous example)!

Model Checking

- **$K = (S, S_0, R, AP, L)$** (the system)
- φ, an **LTL** formula. (the property)
- $K \models \varphi$ ***iff*** **every AP-computation** of **K** is a model of φ.
- Determining this is the ***model checking problem***.
- A solution to this problem can be automated!