FORMAL METHODS

LECTURE III: LINEAR TEMPORAL LOGIC

Alessandro Artale

Faculty of Computer Science – Free University of Bolzano
Room 2.03

artale@inf.unibz.it http://www.inf.unibz.it/∼artale/

Some material (text, figures) displayed in these slides is courtesy of:
M. Benerecetti, A. Cimatti, M. Fisher, F. Giunchiglia, M. Pistore, M. Roveri, R.Sebastiani.

Summary of Lecture III

- Introducing Temporal Logics.
- Intuitions beyond Linear Temporal Logic.
- LTL: Syntax and Semantics.
- LTL in Computer Science.
- LTL Interpreted over Kripke Models.
- LTL and Model Checking: Intuitions.

An Introduction to Temporal Logics

In classical logic, formulae are evaluated within a single fixed world.

For example, a proposition such as “it is Monday” must be either *true* or *false*.

Propositions are then combined using constructs such as ‘$\wedge$’, ‘$\neg$’, etc.

But, most (not just computational) systems are dynamic.

In temporal logics, evaluation takes place within a set of worlds. Thus, “it is Monday” may be satisfied in some worlds, but not in others.

An Introduction to Temporal Logics (Cont.)

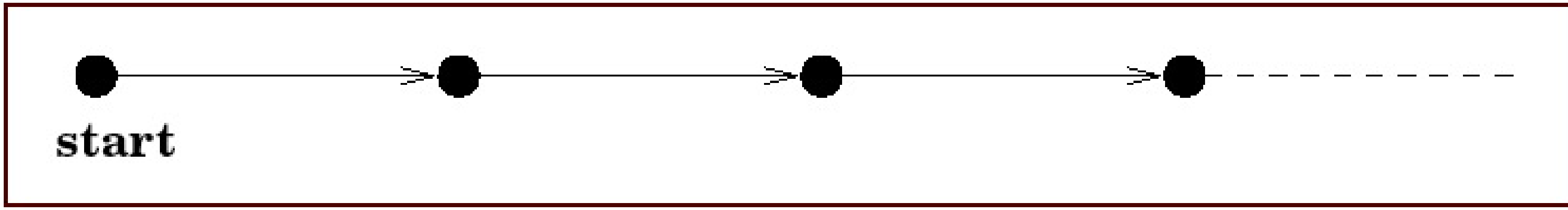
The set of worlds correspond to moments in time.

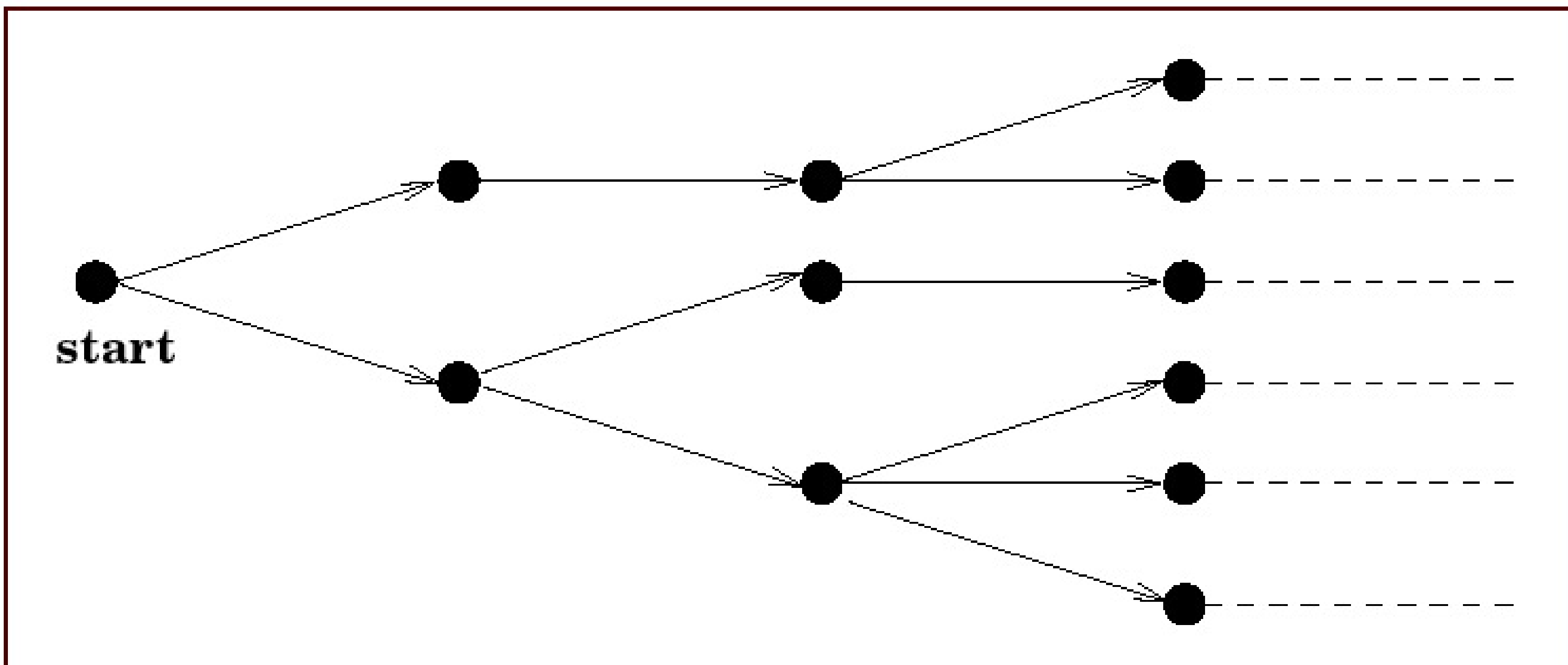
How we navigate between these worlds depends on our particular view of time.

The particular model of time is captured by a temporal accessibility relation between worlds.

Essentially, temporal logic extends classical propositional logic with a set of temporal operators that navigate between worlds using this accessibility relation.

Typical Models of Time





Summary

- Introducing Temporal Logics.
- Intuitions beyond Linear Temporal Logic.
- LTL: Syntax and Semantics.
- LTL in Computer Science.
- LTL Interpreted over Kripke Models.
- LTL and Model Checking: Intuitions.

Linear Temporal Logic (LTL): Intuitions

Consider the simple **Linear Temporal Logic** (LTL) where the accessibility relation characterises a discrete, linear model isomorphic to the Natural Numbers.

Typical temporal operators used are

$\bigcirc\varphi$	φ is true in the *next* moment in time
$\Box\varphi$	φ is true in *all* future moments
$\Diamond\varphi$	φ is true in *some* future moment
$\varphi\,\mathcal{U}\,\psi$	φ is true *until* ψ is true

Examples:

$$\Box((\neg passport \vee \neg ticket) \Rightarrow \bigcirc\neg board\_flight)$$

Computational Example

$$\Box(requested \Rightarrow \Diamond received)$$

$$\Box(received \Rightarrow \bigcirc processed)$$

$$\Box(processed \Rightarrow \Diamond\Box done)$$

From the above we should be able to infer that it is *not* the case that the system continually re-sends a request, but never sees it completed ($\Box\neg done$); i.e. the statement

$$\Box requested \wedge \Box\neg done$$

should be inconsistent.

Summary

- Introducing Temporal Logics.
- Intuitions beyond Linear Temporal Logic.
- LTL: Syntax and Semantics.
- LTL in Computer Science.
- LTL Interpreted over Kripke Models.
- LTL and Model Checking: Intuitions.

LTL: Syntax

Countable set Σ of *atomic propositions*: $p, q, \ldots$ the set $\textsc{Form}$ of formulas is:

$$
\begin{array}{lcll}
\varphi, \psi & \rightarrow & p \mid & \text{(atomic proposition)} \\
 & & \top \mid & \text{(true)} \\
 & & \bot \mid & \text{(false)} \\
 & & \neg\varphi \mid & \text{(complement)} \\
 & & \varphi \wedge \psi \mid & \text{(conjunction)} \\
 & & \varphi \vee \psi \mid & \text{(disjunction)} \\
 & & \bigcirc\varphi \mid & \text{(next time)} \\
 & & \Box\varphi \mid & \text{(always)} \\
 & & \Diamond\varphi \mid & \text{(sometime)} \\
 & & \varphi\, \mathcal{U}\, \psi & \text{(until)}
\end{array}
$$

Temporal Semantics

We interpret our temporal formulae in a discrete, linear model of time. Formally, this structure is represented by

$$\mathcal{M} = \langle \mathbb{N}, \mathcal{I} \rangle$$

where

- $\mathcal{I} : \mathbb{N} \mapsto 2^{\Sigma}$
  maps each Natural number (representing a moment in time) to a set of propositions.

The semantics of a temporal formula is provided by the *satisfaction* relation:

$$\models : (\mathcal{M} \times \mathbb{N} \times \textsc{Form}) \rightarrow \{\mathbf{true}, \mathbf{false}\}$$

Semantics: The Propositional Aspect

We start by defining when an atomic proposition is true at a time point "i"

$$\langle \mathcal{M}, i \rangle \models p \quad \textbf{iff} \quad p \in \mathcal{I}(i) \qquad (\text{for } p \in \Sigma)$$

The semantics for the classical operators is as expected:

$$\langle \mathcal{M}, i \rangle \models \neg\varphi \quad \textbf{iff} \quad \langle \mathcal{M}, i \rangle \not\models \varphi$$

$$\langle \mathcal{M}, i \rangle \models \varphi \wedge \psi \quad \textbf{iff} \quad \langle \mathcal{M}, i \rangle \models \varphi \text{ and } \langle \mathcal{M}, i \rangle \models \psi$$

$$\langle \mathcal{M}, i \rangle \models \varphi \vee \psi \quad \textbf{iff} \quad \langle \mathcal{M}, i \rangle \models \varphi \text{ or } \langle \mathcal{M}, i \rangle \models \psi$$

$$\langle \mathcal{M}, i \rangle \models \varphi \Rightarrow \psi \quad \textbf{iff} \quad \text{if } \langle \mathcal{M}, i \rangle \models \varphi \text{ then } \langle \mathcal{M}, i \rangle \models \psi$$

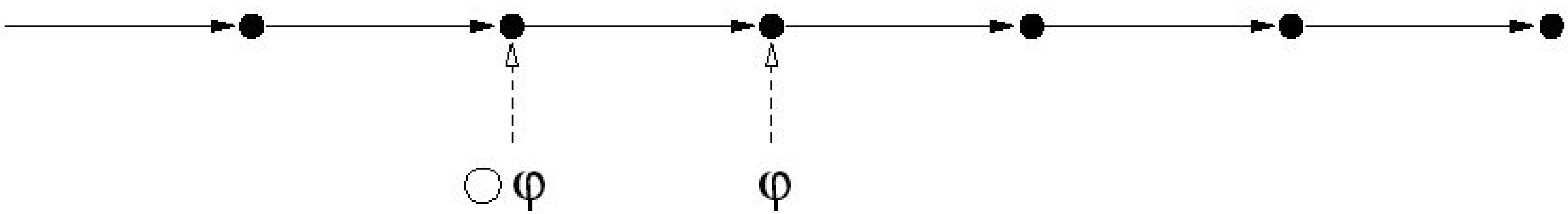
$$\mathcal{M}, i \models \top$$

$$\mathcal{M}, i \not\models \bot$$

Temporal Operators: ‘next’

$$\langle \mathcal{M}, i \rangle \models \bigcirc \varphi \quad \textbf{iff} \quad \langle \mathcal{M}, i+1 \rangle \models \varphi$$

This operator provides a constraint on the next moment in time.



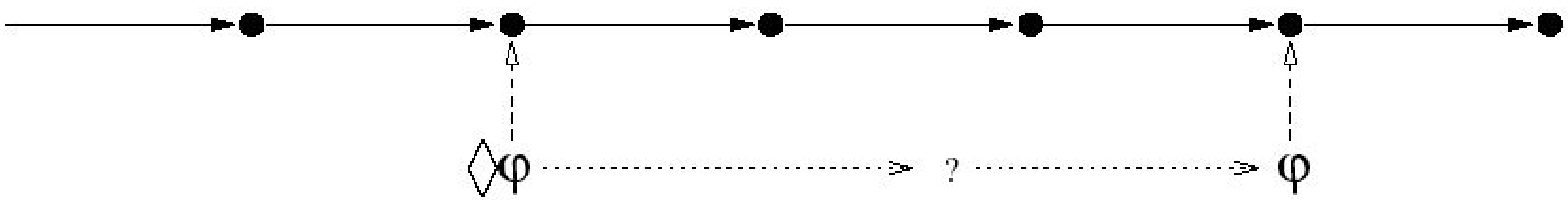
Examples:

$$\begin{array}{rcl} (sad \wedge \neg rich) & \Rightarrow & \bigcirc sad \\ ((x=0) \wedge add3) & \Rightarrow & \bigcirc (x=3) \end{array}$$

Temporal Operators: 'sometime'

$$\langle \mathcal{M}, i\rangle \models \Diamond\varphi \quad \textbf{iff} \quad \text{there exists } j.\ (j \geq i) \wedge \langle \mathcal{M}, j\rangle \models \varphi$$

N.B. while we can be sure that φ *will* be true either now or in the future, we can not be sure exactly *when* it will be true.



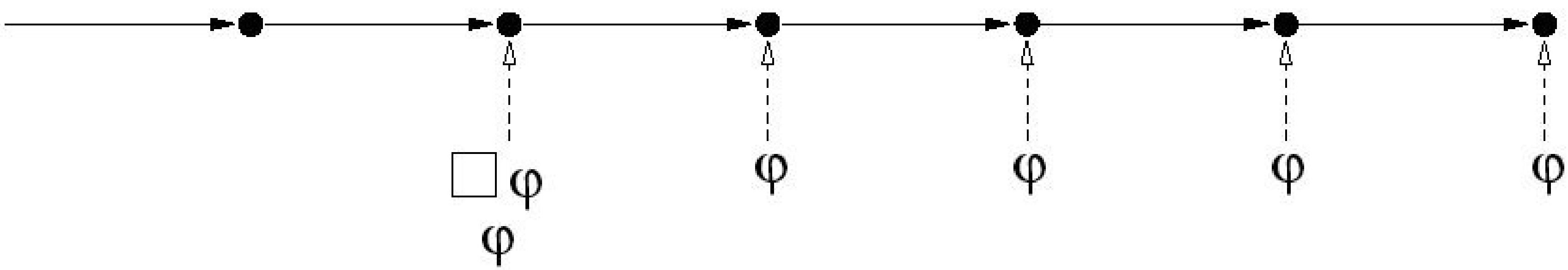
Examples:

$$\begin{aligned}
(\neg resigned \wedge sad) &\Rightarrow \Diamond famous \\
sad &\Rightarrow \Diamond happy \\
send &\Rightarrow \Diamond receive
\end{aligned}$$

Temporal Operators: ‘always’

$$\langle \mathcal{M}, i \rangle \models \Box\varphi \quad \textbf{iff} \quad \text{for all } j. \text{ if } (j \geq i) \text{ then } \langle \mathcal{M}, j \rangle \models \varphi$$

This can represent invariant properties.

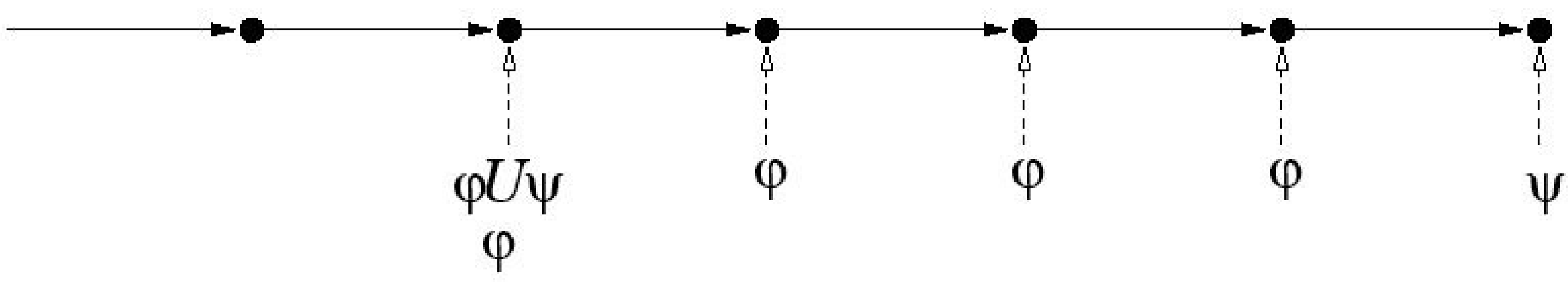


Examples:

$$\textit{lottery-win} \Rightarrow \Box \textit{rich}$$

Temporal Operators: 'until'

$$\langle \mathcal{M}, i\rangle \models \varphi\, \mathcal{U}\, \psi \quad \textbf{iff} \quad \text{there exists } j.\ (j \geq i) \wedge \langle \mathcal{M}, j\rangle \models \psi\ \wedge$$
$$\text{for all } k.\ (i \leq k < j) \Rightarrow \langle \mathcal{M}, k\rangle \models \varphi$$



Examples:

$$
\begin{aligned}
\mathit{start\_lecture} &\Rightarrow \mathit{talk}\ \mathcal{U}\ \mathit{end\_lecture}\\
\mathit{born} &\Rightarrow \mathit{alive}\ \mathcal{U}\ \mathit{dead}\\
\mathit{request} &\Rightarrow \mathit{reply}\ \mathcal{U}\ \mathit{acknowledgement}
\end{aligned}
$$

Satisfiability and Validity

A structure $\mathcal{M} = \langle \mathbb{N}, I \rangle$ is a **model** of ϕ, if

$$\langle \mathcal{M}, i \rangle \models \phi, \textit{ for some } i \in \mathbb{N}.$$

Similarly as in classical logic, an LTL formula ϕ can be **satisfiable**, **unsatisfiable** or **valid**. A formula ϕ is:

- **Satisfiable**, if there is model for ϕ.
- **Unsatisfiable**, if ϕ is not satisfiable.
- **Valid** (i.e., a **Tautology**):
  $\models \phi$ iff $\forall \mathcal{M}, \forall i \in \mathbb{N}.\ \langle \mathcal{M}, i \rangle \models \phi$.

Entailment and Equivalence

Similarly as in classical logic we can define the notions of **entailment** and **equivalence** between two LTL formulas

- **Entailment.**
  $\phi \models \psi$ iff $\forall \mathcal{M}, \forall i \in \mathbb{N}. \langle \mathcal{M}, i \rangle \models \phi \Rightarrow \langle \mathcal{M}, i \rangle \models \psi$
- **Equivalence.**
  $\phi \equiv \psi$ iff $\forall \mathcal{M}, \forall i \in \mathbb{N}. \langle \mathcal{M}, i \rangle \models \phi \Leftrightarrow \langle \mathcal{M}, i \rangle \models \psi$

Equivalences in LTL

The temporal operators $\Box$ and $\Diamond$ are duals

$$\neg \Box \varphi \equiv \Diamond \neg \varphi$$

$\Diamond$ (and then $\Box$) can be rewritten in terms of $\mathcal{U}$

$$\Diamond \varphi \equiv \top \, \mathcal{U} \, \varphi$$

All the temporal operators can be rewritten using the “Until” and “Next” operators

Equivalences in LTL (Cont.)

$\Diamond$ distributes over $\vee$ while $\Box$ distributes over $\wedge$

$$\Diamond(\varphi \vee \psi) \equiv \Diamond\varphi \vee \Diamond\psi$$
$$\Box(\varphi \wedge \psi) \equiv \Box\varphi \wedge \Box\psi$$

The following equivalences are useful for generating formulas in Negated Normal Form.

$$\neg\bigcirc\varphi \equiv \bigcirc\neg\varphi$$

$$\neg(\varphi\,\mathcal{U}\,\psi) \equiv (\neg\psi\,\mathcal{U}\,(\neg\varphi \wedge \neg\psi)) \vee \Box\neg\psi$$

LTL Vs. FOL

Linear Temporal Logic can be thought of as

> *a specific decidable (PSPACE-complete) fragment of classical first-order logic*

We just map each proposition to a unary predicate in FOL. In general, the following satisfiability preserving mapping ($\leadsto$) holds:

$$
\begin{array}{lcl}
p & \leadsto & p(t) \\
\bigcirc p & \leadsto & p(t+1) \\
\Diamond p & \leadsto & \exists t'.\ (t' \geq t) \wedge p(t') \\
\Box p & \leadsto & \forall t'.\ (t' \geq t) \Rightarrow p(t')
\end{array}
$$

LTL Alternative Notation

Alternative notations are used for temporal operators.

$$\Diamond \quad \rightsquigarrow \quad F \quad \text{sometime in the Future}$$
$$\Box \quad \rightsquigarrow \quad G \quad \text{Globally in the future}$$
$$\bigcirc \quad \rightsquigarrow \quad X \quad \text{neXtime}$$

Summary

- Introducing Temporal Logics.
- Intuitions beyond Linear Temporal Logic.
- LTL: Syntax and Semantics.
- LTL in Computer Science.
- LTL Interpreted over Kripke Models.
- LTL and Model Checking: Intuitions.

Temporal Logic in Computer Science

Temporal logic was originally developed in order to represent tense in natural language.

Within Computer Science, it has achieved a significant role in the formal specification and verification of concurrent reactive systems.

Much of this popularity has been achieved as a number of useful concepts can be formally, and concisely, specified using temporal logics, e.g.

- *safety properties*
- *liveness properties*
- *fairness properties*

Safety Properties

Safety:

"something bad will not happen"

Typical examples:

$\Box\neg(reactor\_temp > 1000)$

$\Box\neg((x = 0) \wedge \bigcirc\bigcirc\bigcirc(y = z/x))$

and so on.....

Usually: $\Box\neg \ldots.$

Liveness Properties

Liveness:

"something good will happen"

Typical examples:

$\Diamond rich$

$\Diamond(x > 5)$

$\Box(start \Rightarrow \Diamond terminate)$

$\Box(Trying \Rightarrow \Diamond Critical)$

and so on.....

Usually: $\Diamond \ldots$

Fairness Properties

Often only really useful when scheduling processes, responding to messages, etc.

Strong Fairness:

> "if something is attempted/requested infinitely often, then it will be successful/allocated infinitely often"

Typical example:

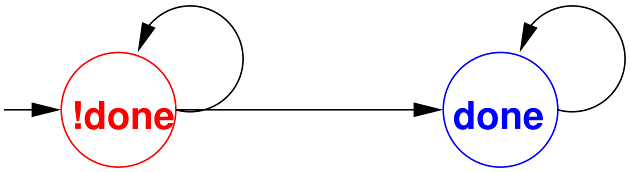
$$\Box\Diamond ready \Rightarrow \Box\Diamond run$$

Summary

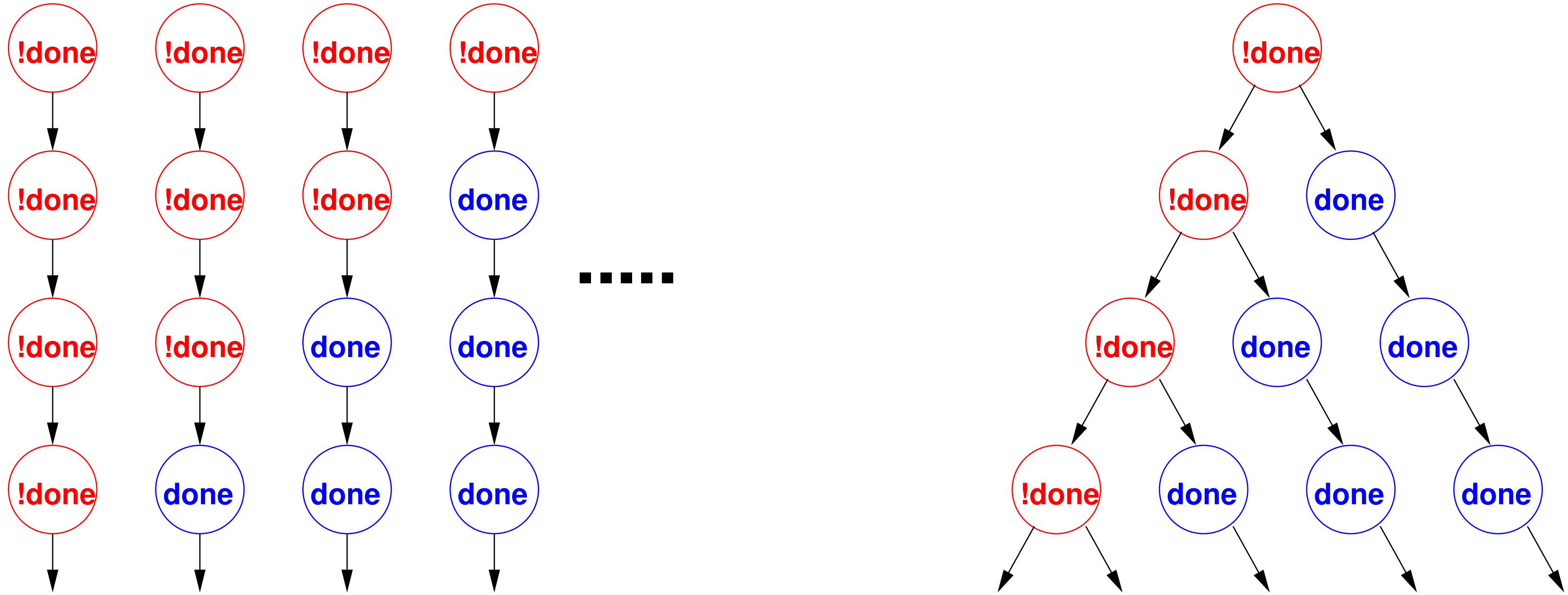
- Introducing Temporal Logics.
- Intuitions beyond Linear Temporal Logic.
- LTL: Syntax and Semantics.
- LTL in Computer Science.
- LTL Interpreted over Kripke Models.
- LTL and Model Checking: Intuitions.

Kripke Models and Linear Structures

Consider the following Kripke structure:



Its paths/computations can be seen as a set of linear structures, and thus as a computation tree (**unraveling**):



Path-Semantics for LTL

- LTL formulae are evaluated over the set $\mathbb{N}$ of Natural Numbers.
- Paths in Kripke structures are infinite and linear sequences of states. Thus, they are isomorphic to the Natural Numbers:
  $\pi = s_0 \longrightarrow s_1 \longrightarrow \cdots \longrightarrow s_i \longrightarrow s_{i+1} \longrightarrow \cdots$
- We want to interpret LTL formulas over Kripke structures: $\langle \mathcal{KM}, s \rangle \models \phi$
- Given a Kripke structure, $\mathcal{KM} = (S, I, R, AP, L)$, a path π in $\mathcal{KM}$, a state $s \in S$, and an LTL formula ϕ, we define:
  1. $\langle \mathcal{KM}, \pi \rangle \models \phi$, and then
  2. $\langle \mathcal{KM}, s \rangle \models \phi$

  Based on the LTL semantics over the Natural Numbers.

Path-Semantics for LTL (Cont.)

- We first extract an LTL structure, $\mathcal{M}_\pi = (\pi, I_\pi)$, from the Kripke structure $\mathcal{KM}$, such that:
  - π is a path in $\mathcal{KM}$
  - I_π is the restriction of L to states in π:

$$\forall s \in \pi \textit{ and } \forall p \in AP,\ p \in I_\pi(s) \textit{ iff } p \in L(s)$$

- Given a Kripke structure, $\mathcal{KM} = (S, I, R, AP, L)$, a path π in $\mathcal{KM}$, a state $s \in S$, and an LTL formula ϕ:
  1. $\langle \mathcal{KM}, \pi \rangle \models \phi$ iff $\langle \mathcal{M}_\pi, s_0 \rangle \models \phi$
     with s_0 initial state of π
  2. $\langle \mathcal{KM}, s \rangle \models \phi$ iff $\langle \mathcal{KM}, \pi \rangle \models \phi$
     for **all** paths π starting at s.

LTL Model Checking Definition

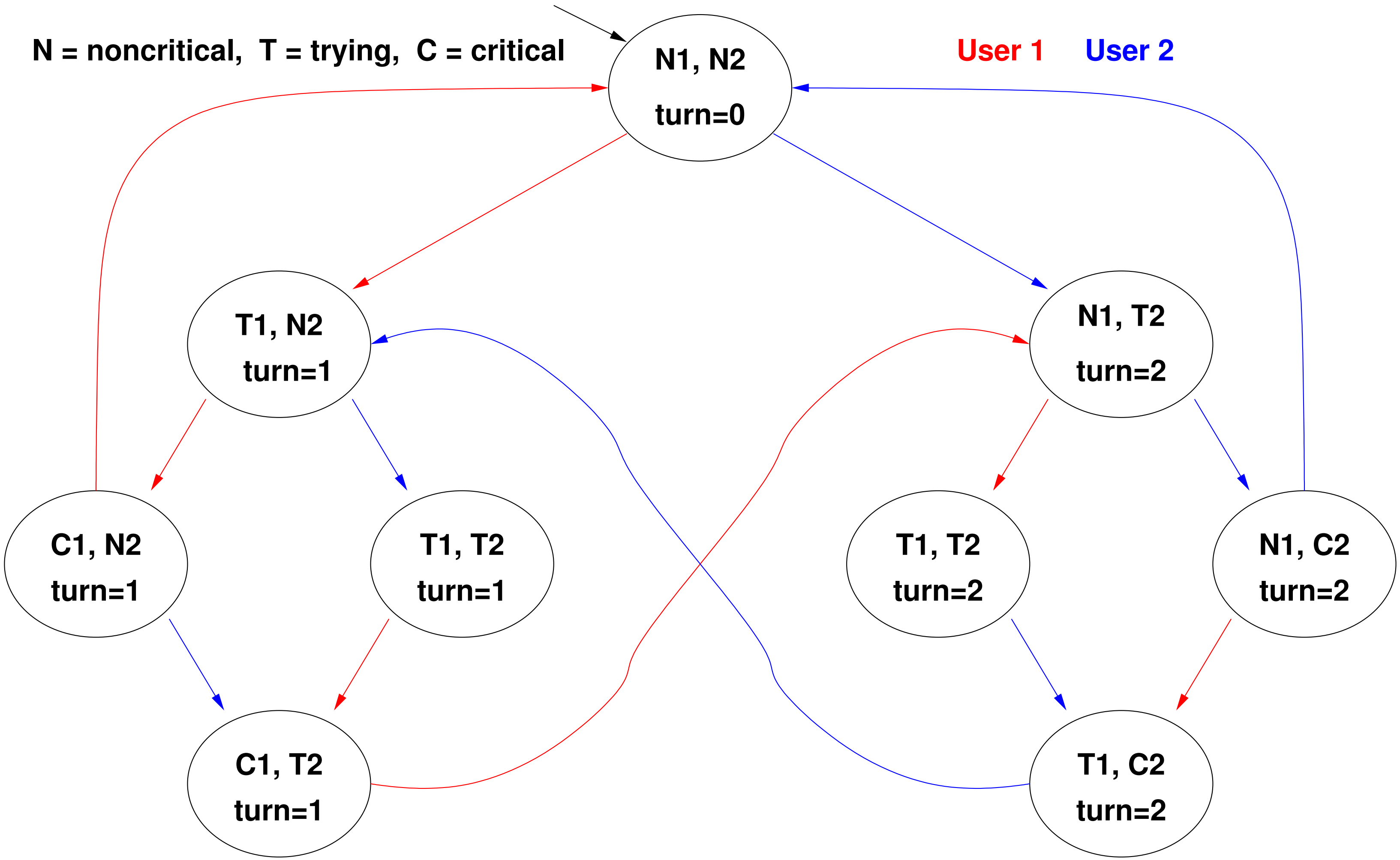
Given a Kripke structure, $\mathcal{KM} = (S, I, R, AP, L)$, the LTL model checking problem, $\mathcal{KM} \models \phi$:

> Checks if $\langle \mathcal{KM}, s_0 \rangle \models \phi$, for every $s_0 \in I$, initial state of the Kripke structure $\mathcal{KM}$

Summary

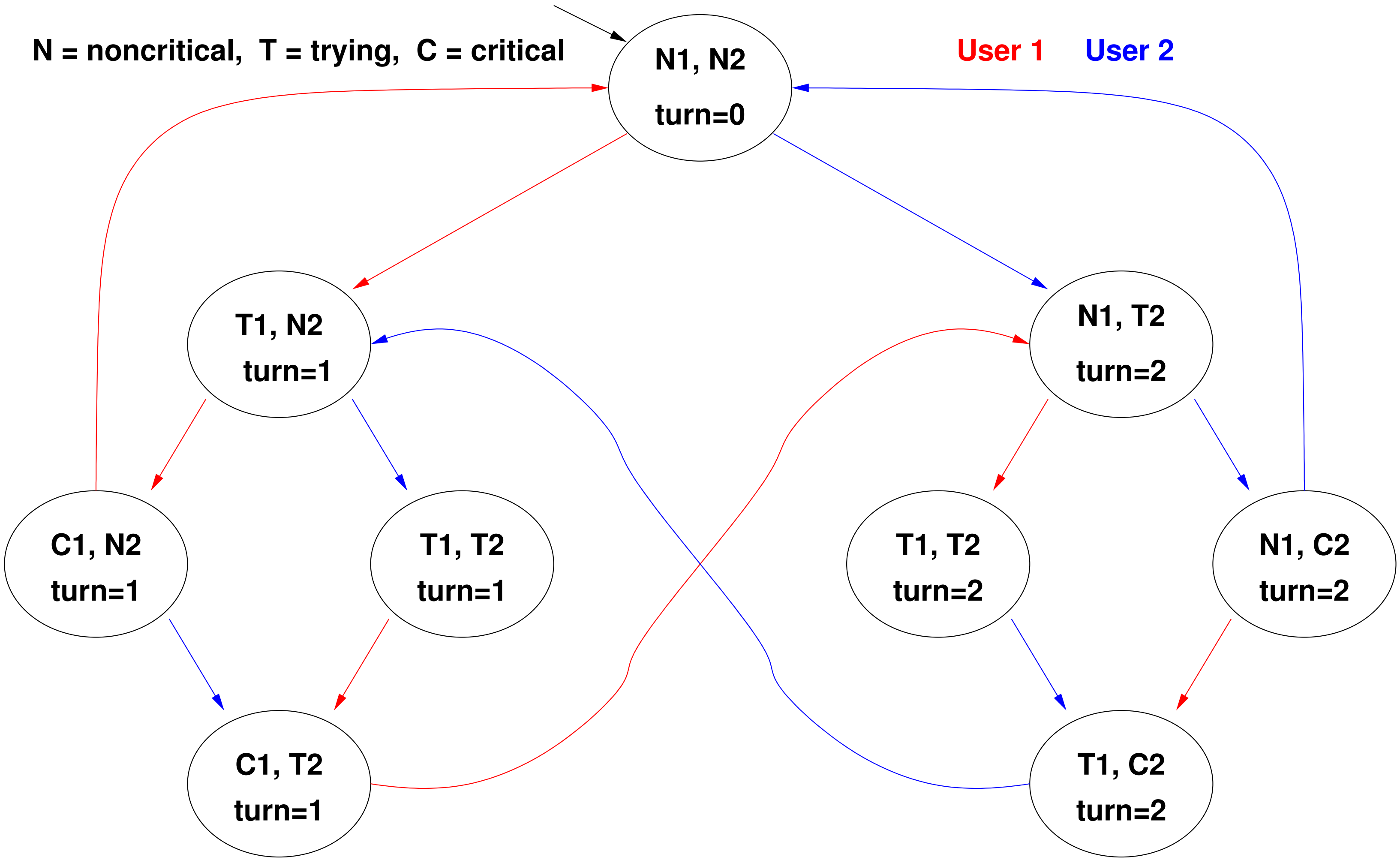
- Introducing Temporal Logics.
- Intuitions beyond Linear Temporal Logic.
- LTL: Syntax and Semantics.
- LTL in Computer Science.
- LTL Interpreted over Kripke Models.
- LTL and Model Checking: Intuitions.

Example 1: mutual exclusion (safety)



$$\mathcal{KM} \models \Box\neg(C_1 \wedge C_2)\ ?$$

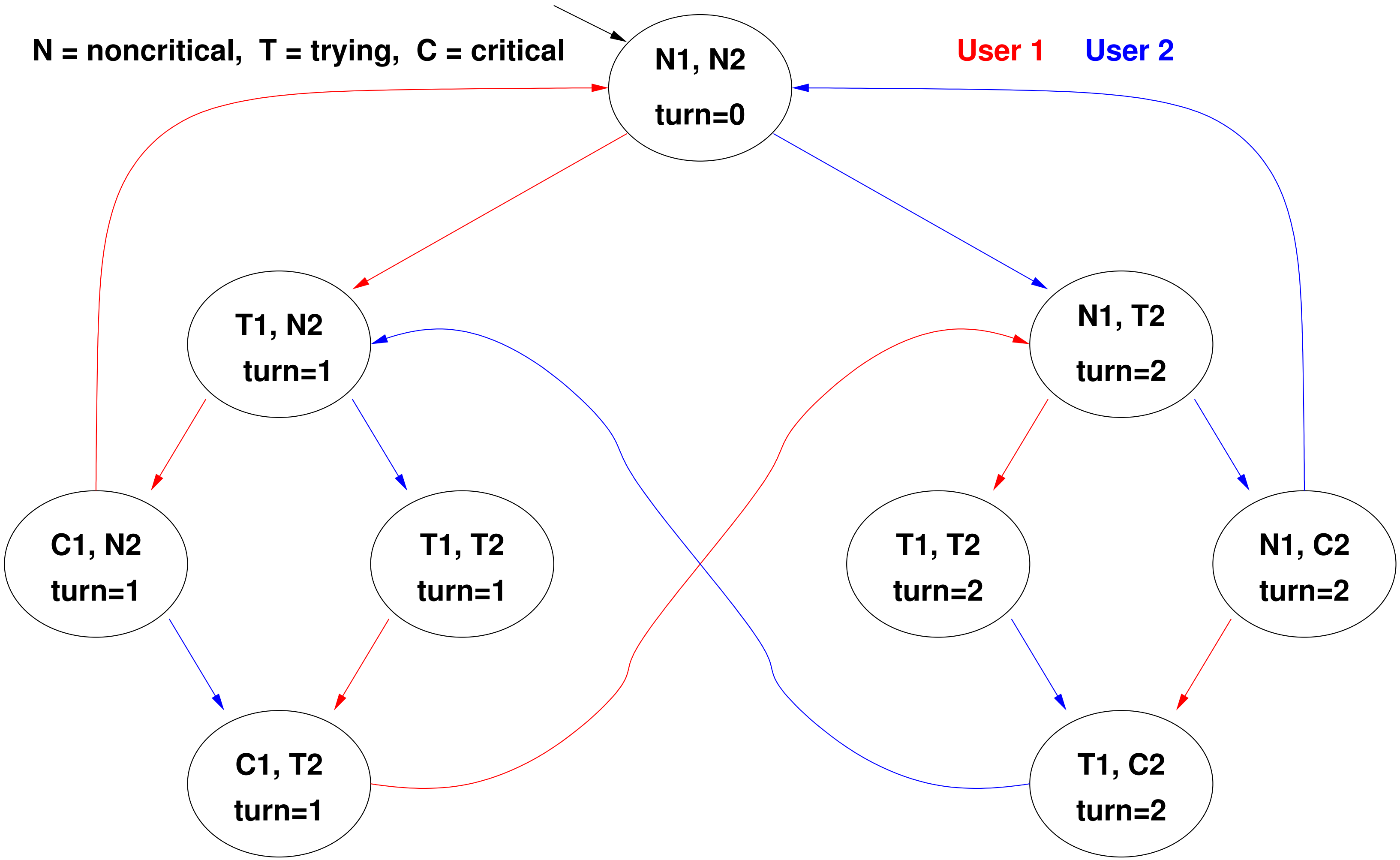
Example 1: mutual exclusion (safety)



$$\mathcal{KM} \models \Box\neg(C_1 \wedge C_2)\ ?$$

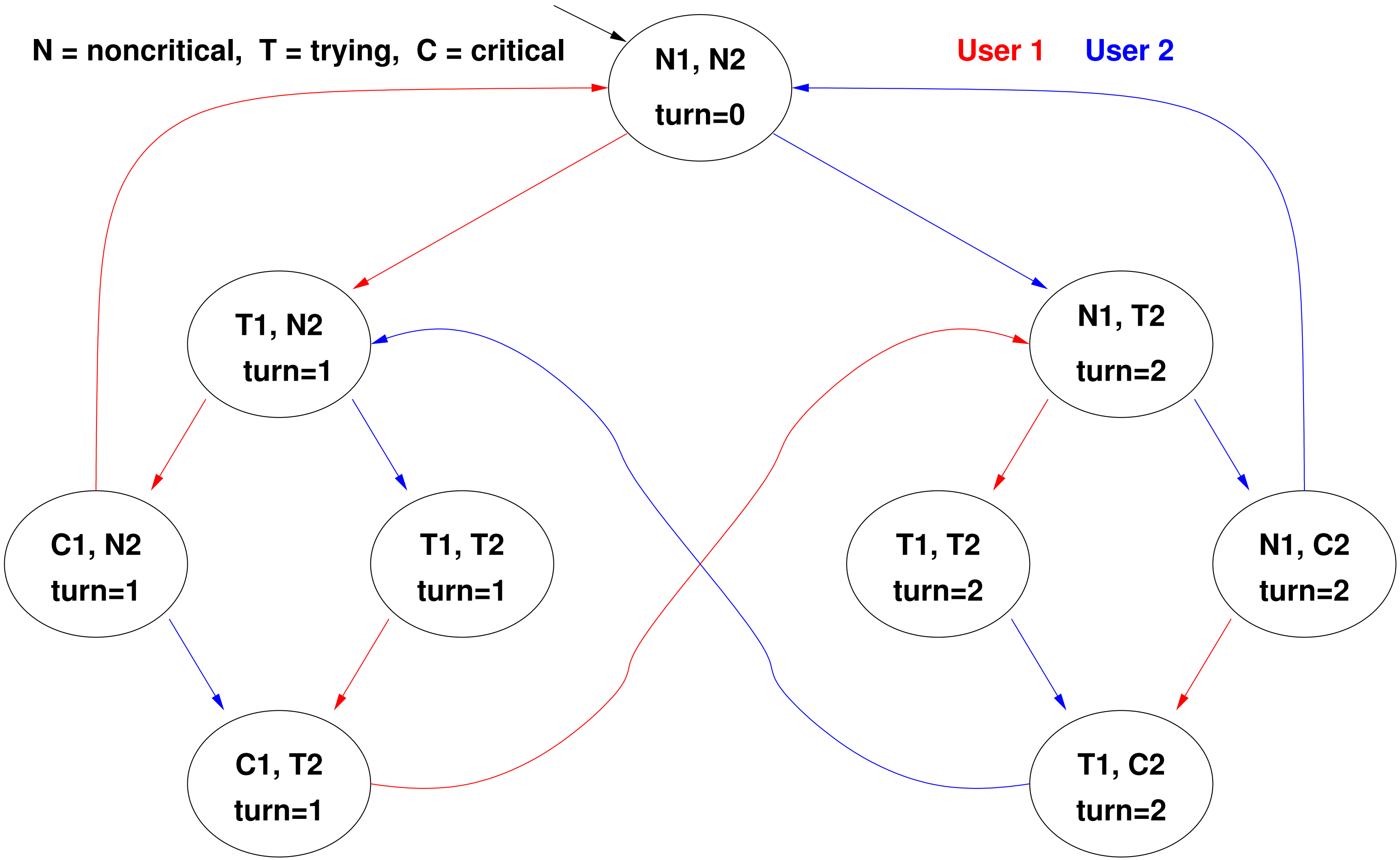
YES: There is no reachable state in which $(C_1 \wedge C_2)$ holds!

Example 2: mutual exclusion (liveness)



$$\mathcal{KM} \models \Diamond C_1 \ ?$$

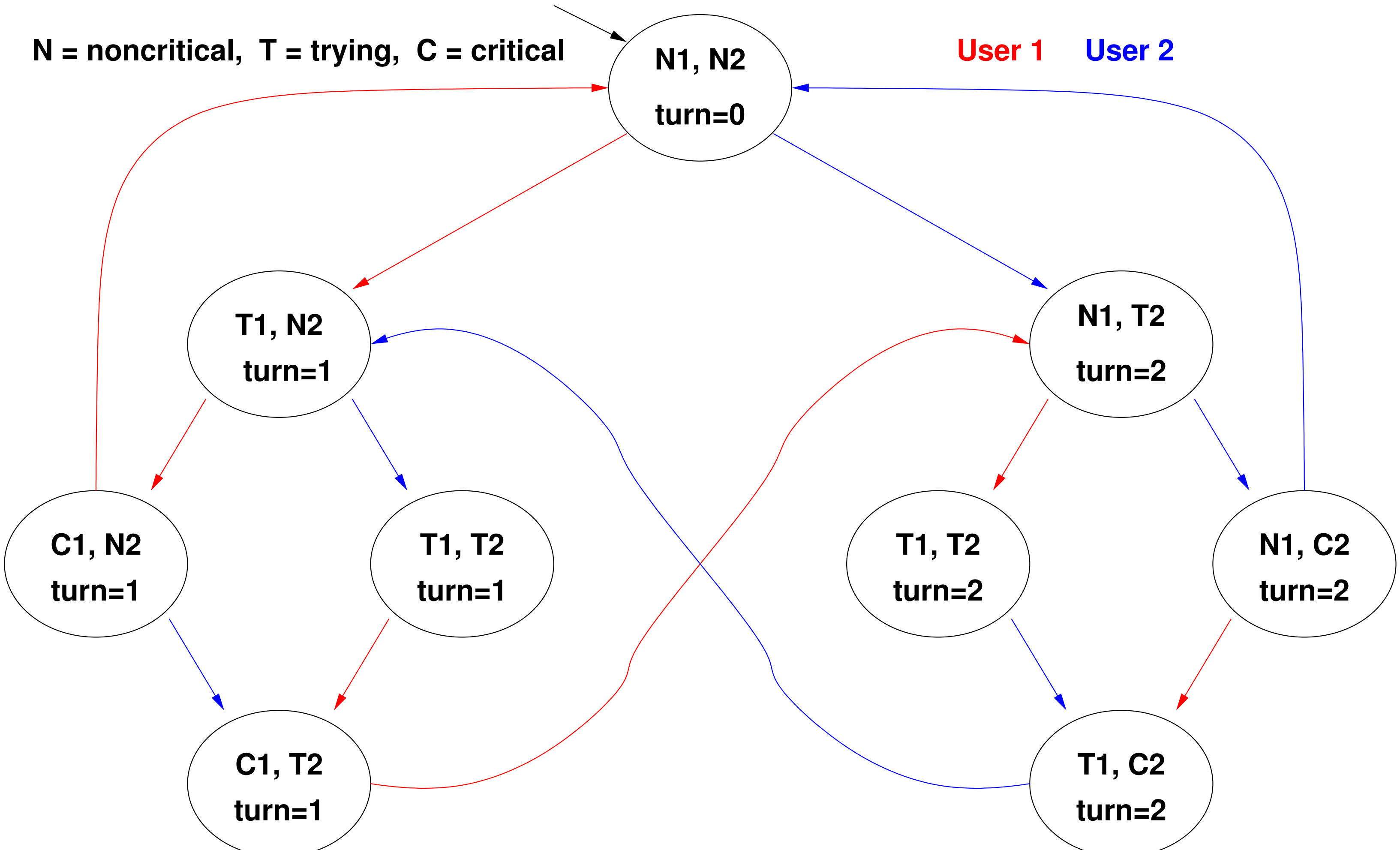
Example 2: mutual exclusion (liveness)



$$\mathcal{KM} \models \Diamond C_1 \ ?$$

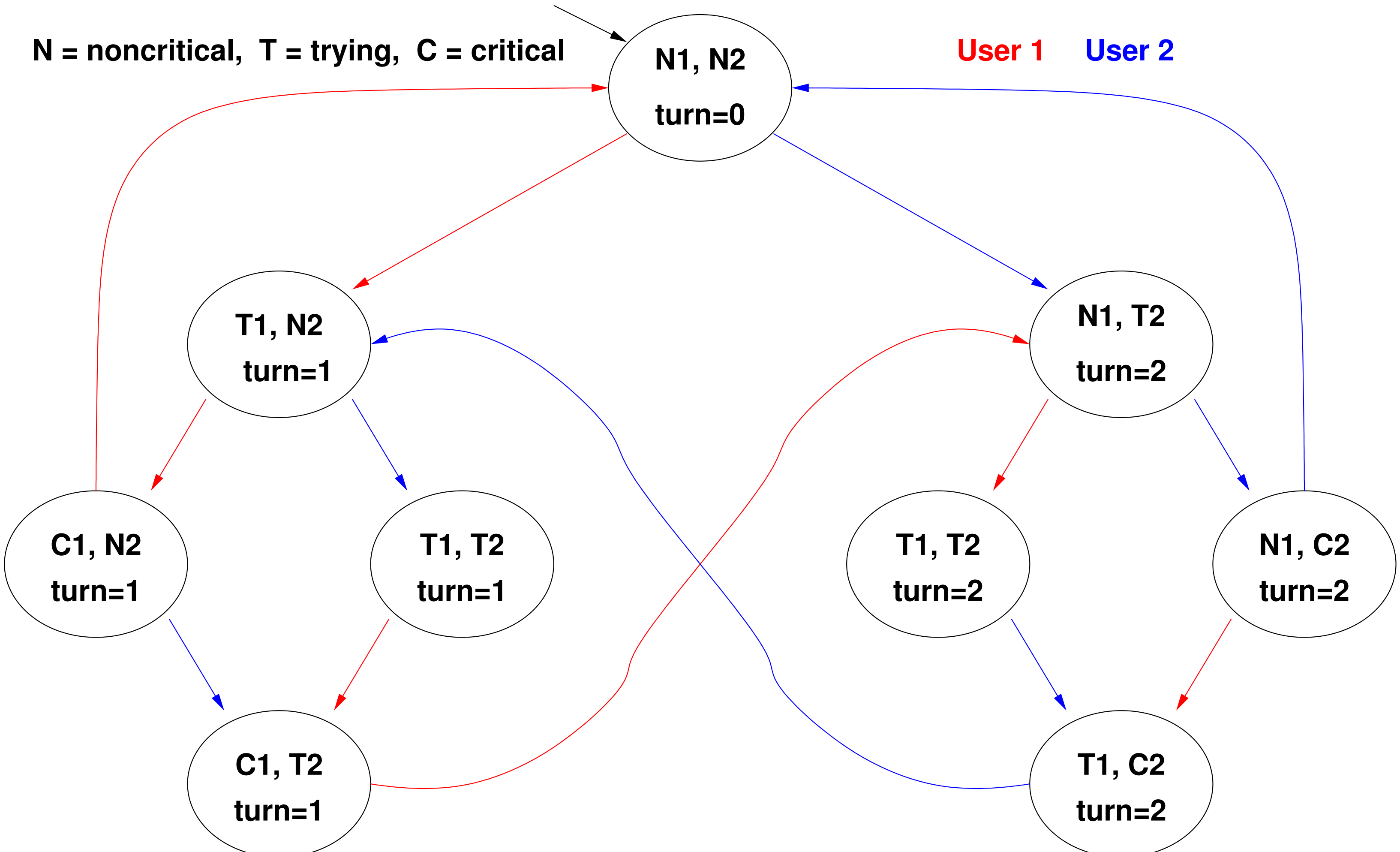
NO: the blue cyclic path is a counterexample!

Example 3: mutual exclusion (liveness)



$$\mathcal{KM} \models \Box(T_1 \Rightarrow \Diamond C_1) \ ?$$

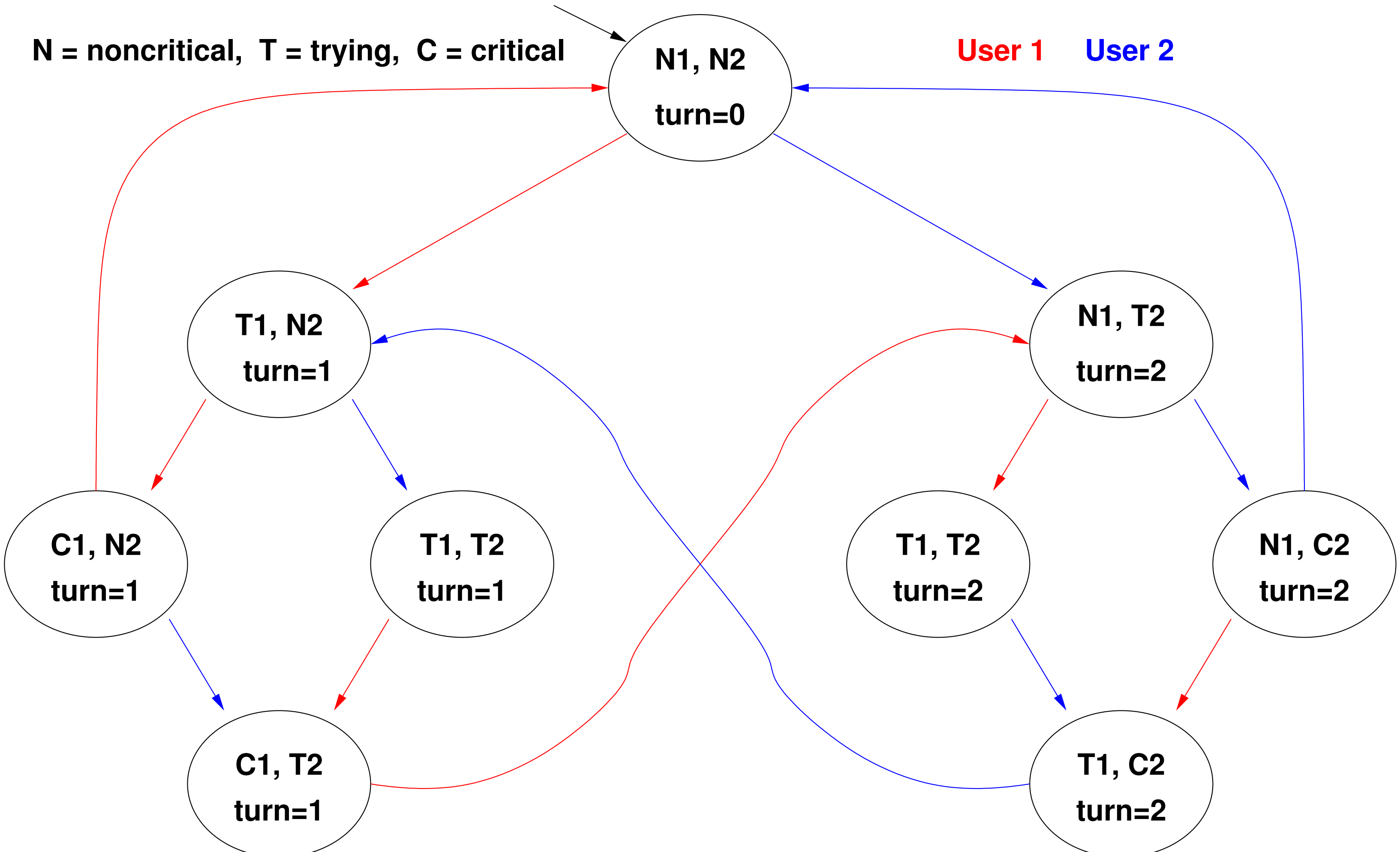
Example 3: mutual exclusion (liveness)



$$\mathcal{KM} \models \Box(T_1 \Rightarrow \Diamond C_1)\ ?$$

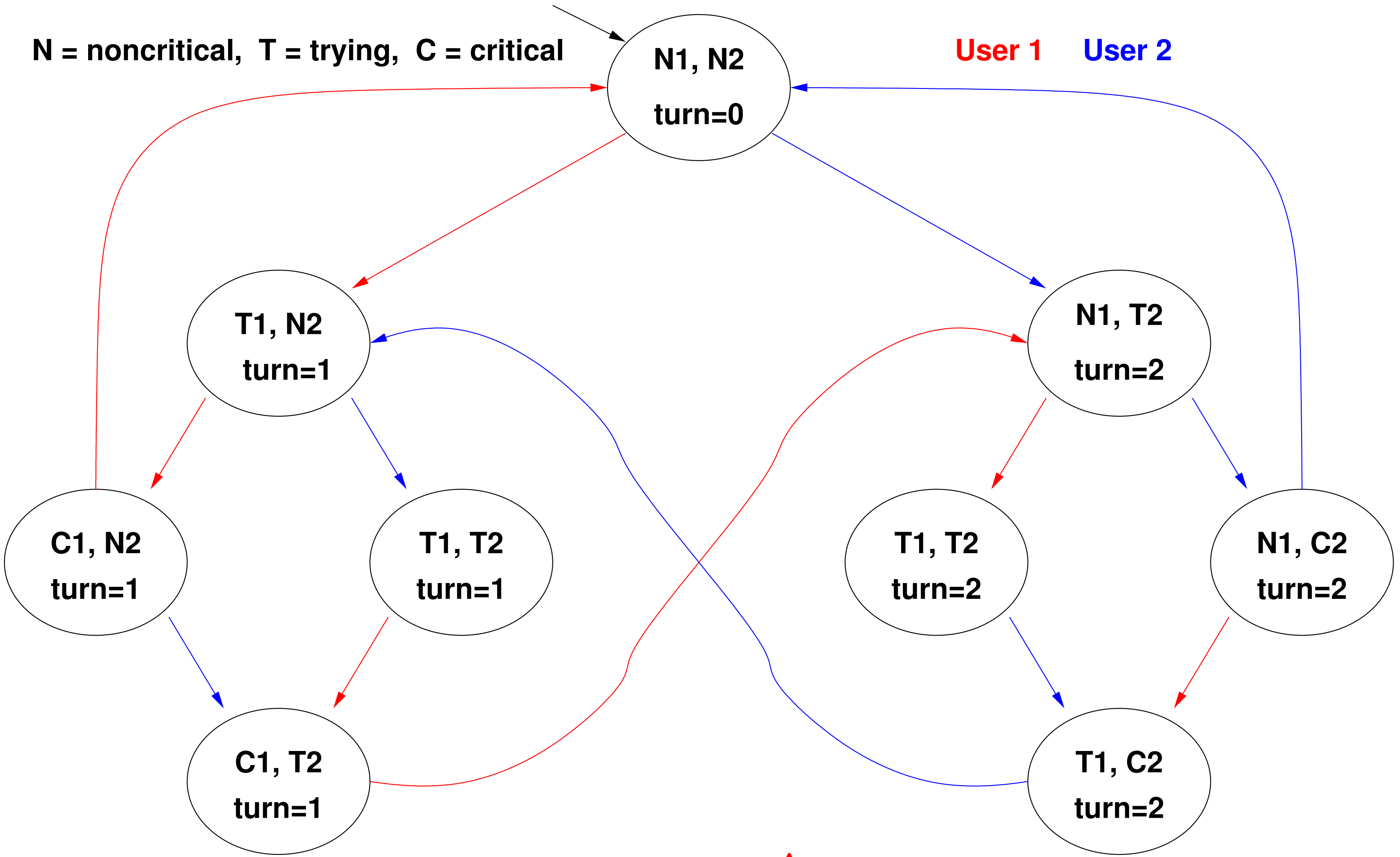
YES: in every path if T_1 holds afterwards C_1 holds!

Example 4: mutual exclusion (fairness)



$$\mathcal{KM} \models \Box\Diamond C_1 \;?$$

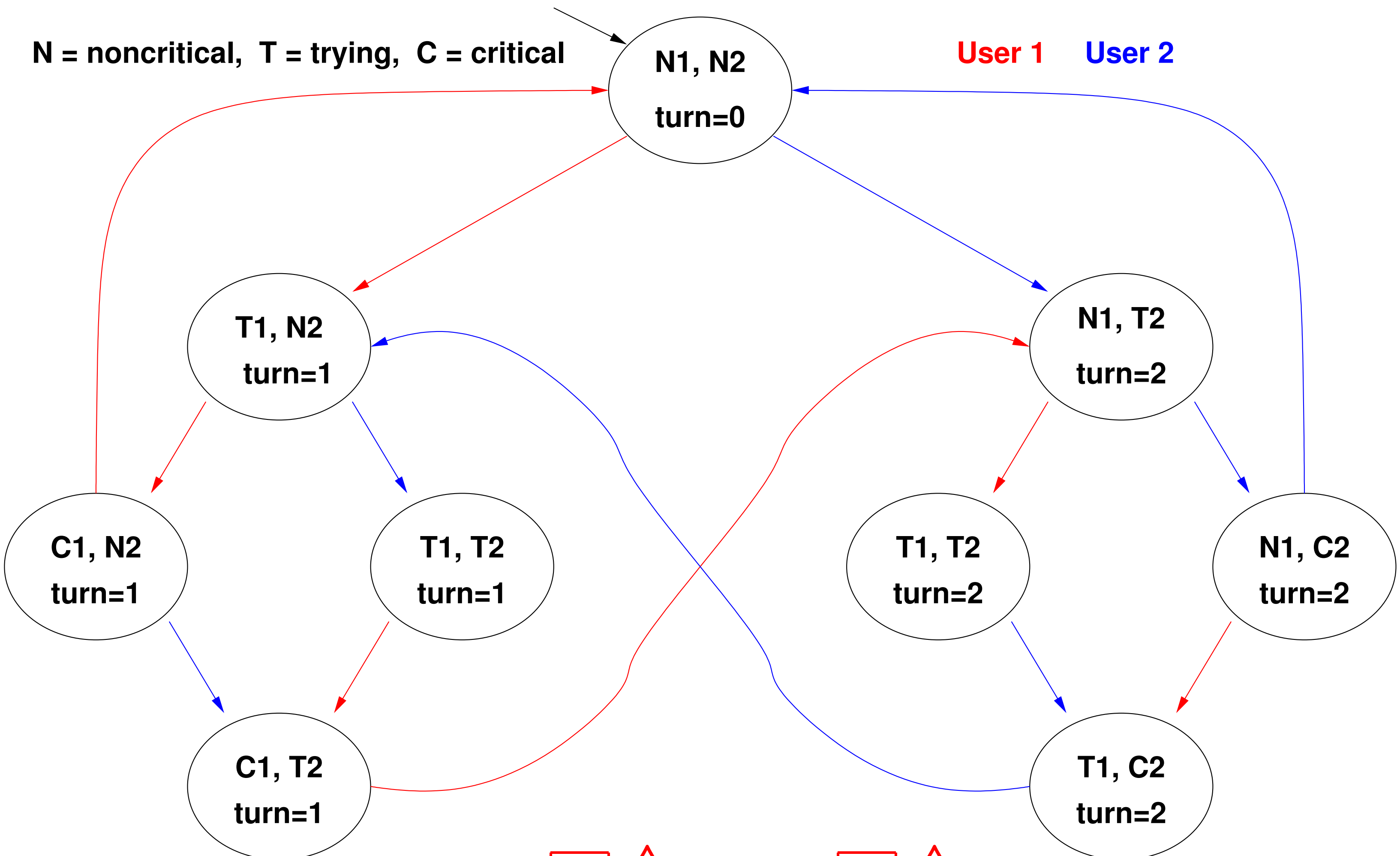
Example 4: mutual exclusion (fairness)



$$\mathcal{KM} \models \Box \Diamond C_1 \; ?$$

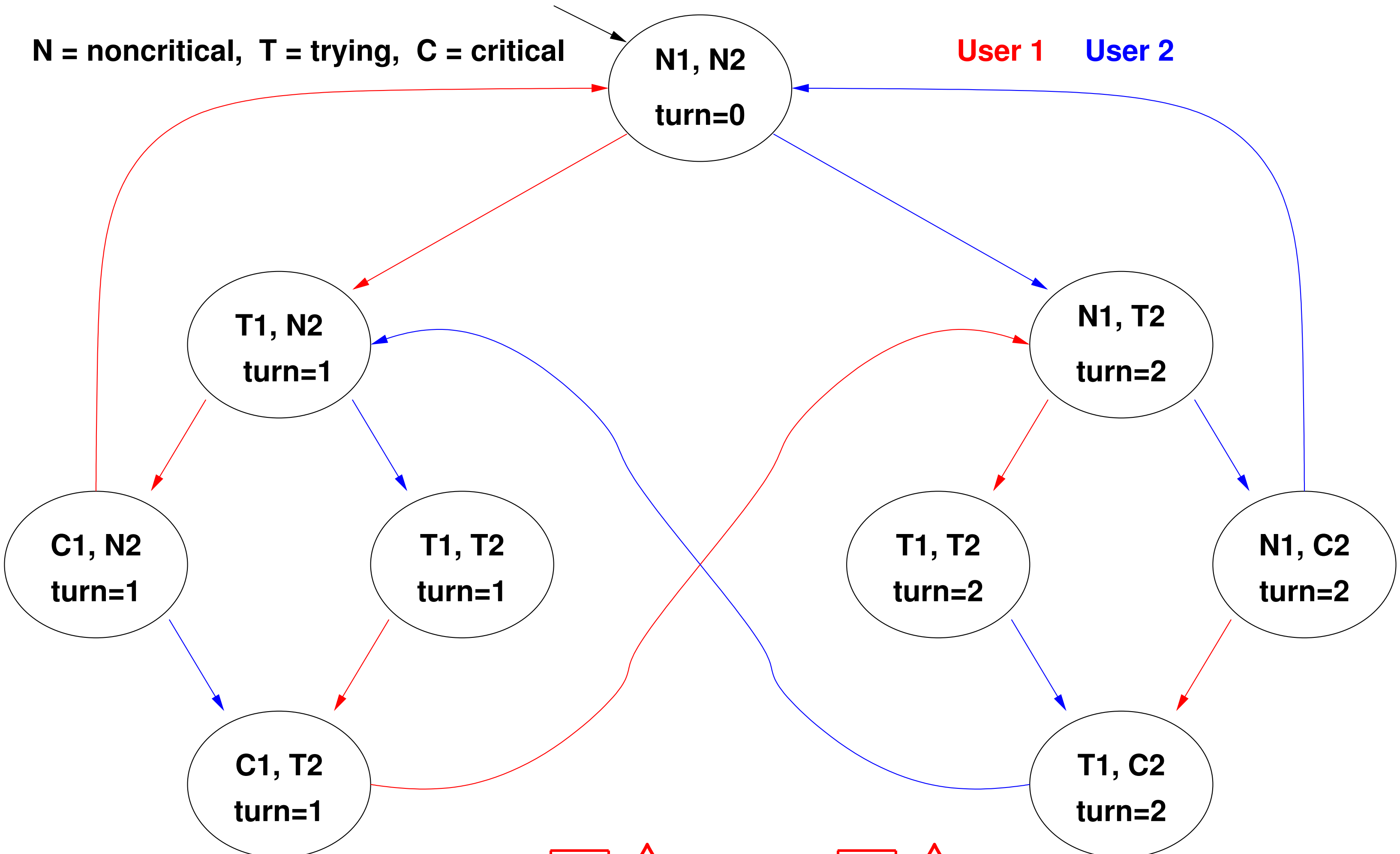
NO: the blue cyclic path is a counterexample!

Example 4: mutual exclusion (strong fairness)



$$\mathcal{KM} \models \Box\Diamond T_1 \Rightarrow \Box\Diamond C_1 \ ?$$

Example 4: mutual exclusion (strong fairness)



$$\mathcal{KM} \models \Box\Diamond T_1 \Rightarrow \Box\Diamond C_1 \ ?$$

YES: every path which visits T_1 infinitely often also visits C_1 infinitely often!

Summary of Lecture III

- Introducing Temporal Logics.
- Intuitions beyond Linear Temporal Logic.
- LTL: Syntax and Semantics.
- LTL in Computer Science.
- LTL Interpreted over Kripke Models.
- LTL and Model Checking: Intuitions.