

LINEAR TEMPORAL LOGIC (LTL)

Slides by Alessandro Artale
<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.*

- p. 1/??

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.

- p. 2/??

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.

– p. 3/??

An Introduction to Temporal Logics (Cont.)

The set of worlds correspond to **moments in time**.

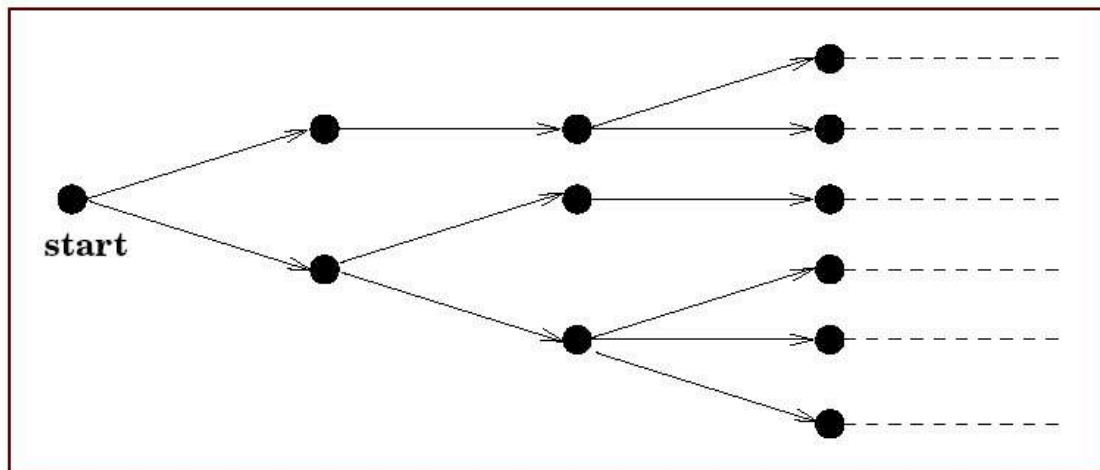
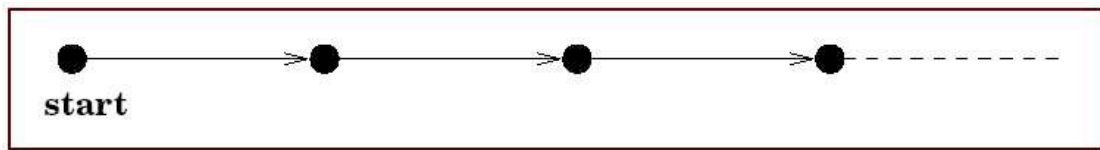
How we navigate between these worlds depends on our particular view of time.

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.

– p. 4/??

Typical Models of Time



- p. 5??

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.

- p. 6??

Linear Temporal Logic (LTL): Intuitions

Consider a simple 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 <i>next</i> moment in time
$\square\varphi$	φ is true in <i>all</i> future moments
$\diamond\varphi$	φ is true in <i>some</i> future moment
$\varphi \mathcal{U} \psi$	φ is true <i>until</i> ψ is true

Examples:

$$\square((\neg passport \vee \neg ticket) \Rightarrow \bigcirc\neg board_flight)$$

- p. 7??

Computational Example

$$\square(requested \Rightarrow \diamond received)$$

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

$$\square(processed \Rightarrow \diamond \square done)$$

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 ($\square\neg done$); i.e. the statement

$$\square requested \wedge \square\neg done$$

should be inconsistent.

- p. 8??

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.

– p. 9??

LTL: Syntax

Countable set Σ of *atomic propositions*: p, q, \dots the set FORM of formulas is:

$\varphi, \psi \rightarrow p$		(atomic proposition)
\top		(true)
\perp		(false)
$\neg\varphi$		(complement)
$\varphi \wedge \psi$		(conjunction)
$\varphi \vee \psi$		(disjunction)
$\bigcirc\varphi$		(next time)
$\square\varphi$		(always)
$\diamond\varphi$		(sometime)
$\varphi \mathcal{U} \psi$		(until)

– p. 10??

Temporal Semantics

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

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

where

- $I : \mathbb{N} \mapsto 2^\Sigma$
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:

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

- p. 11/??

Semantics: The Propositional Aspect

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

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

The semantics for the classical operators is as expected:

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

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

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

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

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

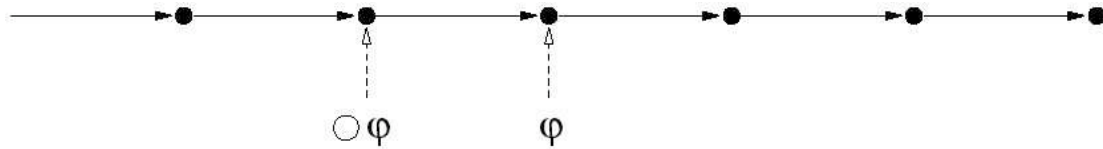
$$\mathcal{M}, i \not\models \perp$$

- p. 12/??

Temporal Operators: 'next'

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

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



Examples:

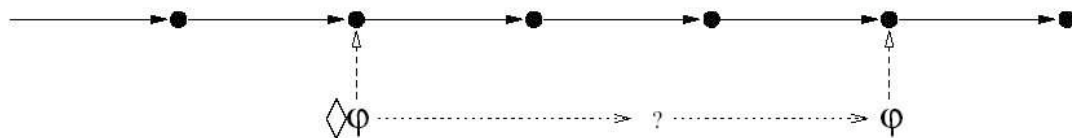
$$\begin{aligned} (sad \wedge \neg rich) &\Rightarrow \bigcirc sad \\ ((x = 0) \wedge add3) &\Rightarrow \bigcirc (x = 3) \end{aligned}$$

- p. 13/??

Temporal Operators: 'sometime'

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

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:

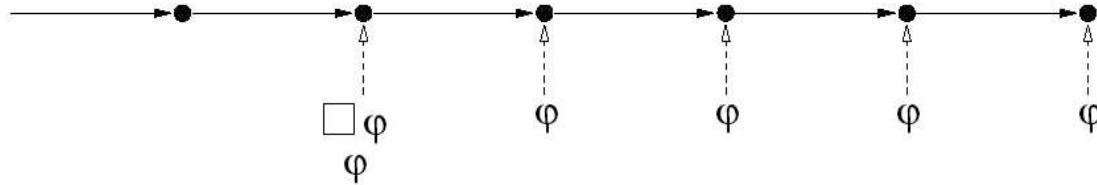
$$\begin{aligned} (\neg resigned \wedge sad) &\Rightarrow \diamond famous \\ sad &\Rightarrow \diamond happy \\ send &\Rightarrow \diamond receive \end{aligned}$$

- p. 14/??

Temporal Operators: 'always'

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

This can represent invariant properties.



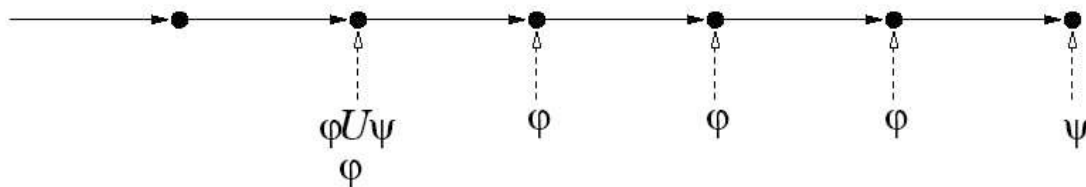
Examples:

lottery-win $\Rightarrow \Box$ *rich*

- p. 15/??

Temporal Operators: 'until'

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



Examples:

start_lecture \Rightarrow *talk* \mathcal{U} *end_lecture*

born \Rightarrow *alive* \mathcal{U} *dead*

request \Rightarrow *reply* \mathcal{U} *acknowledgement*

- p. 16/??

Satisfiability and Validity

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

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

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

- p. 17??

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 $\forall \mathcal{M}, \forall i \in \mathbb{N}. \langle \mathcal{M}, i \rangle \models \phi \Leftrightarrow \langle \mathcal{M}, i \rangle \models \psi$

- p. 18??

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

– p. 19??

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.

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

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

– p. 20??

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 (\rightsquigarrow) holds:

$$\begin{array}{lll} p & \rightsquigarrow & p(t) \\ \bigcirc p & \rightsquigarrow & p(t+1) \\ \diamond p & \rightsquigarrow & \exists t'. (t' \geq t) \wedge p(t') \\ \square p & \rightsquigarrow & \forall t'. (t' \geq t) \Rightarrow p(t') \end{array}$$

- p. 21/??

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.

- p. 22/??

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*

- p. 23/??

Safety Properties

Safety:

“something bad will not happen”

Typical examples:

$$\square \neg (\text{reactor_temp} > 1000)$$

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

and so on.....

Usually: $\square \neg \dots$

- p. 24/??

Liveness Properties

Liveness:

“something good will happen”

Typical examples:

$\diamond rich$

$\diamond (x > 5)$

$\square (start \Rightarrow \diamond terminate)$

$\square (Trying \Rightarrow \diamond Critical)$

and so on.....

Usually: \diamond

- p. 25/??

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:

$\square \diamond ready \Rightarrow \square \diamond run$

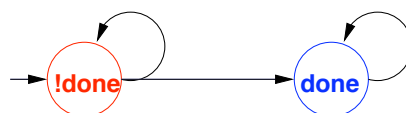
- p. 26/??

Summary

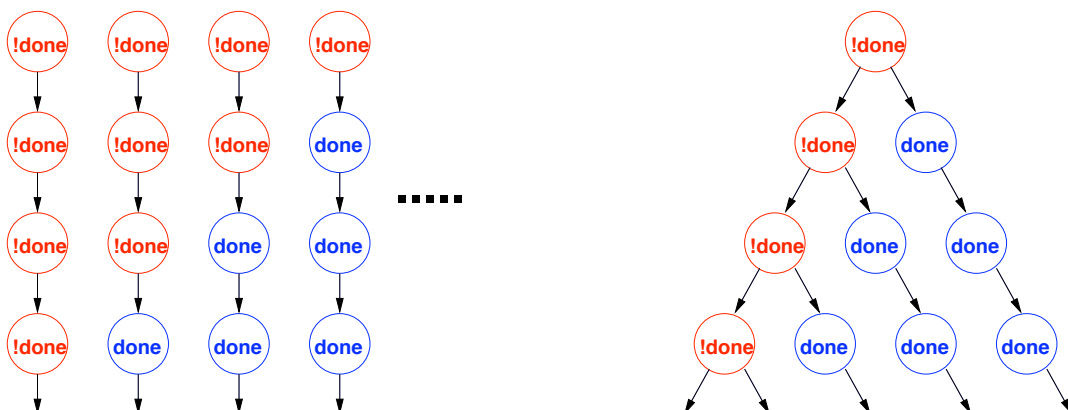
- Introducing Temporal Logics.
- Intuitions beyond Linear Temporal Logic.
- LTL: Syntax and Semantics.
- LTL in Computer Science.
- **LTL Interpreted over Kripke Models.**
- LTL and Model Checking: Intuitions.

Kripke Models and Linear Structures

Consider the following Kripke structure:



Its paths/computations can be seen as a set of linear structures (computation tree):



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:

$$\pi = s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_i \rightarrow s_{i+1} \rightarrow \cdots$$

- We want to interpret LTL formulas over Kripke structures.
- 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.

- p. 29/??

Path-Semantics for LTL (Cont.)

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

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

- 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 .

- p. 30/??

LTL Model Checking Definition

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

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

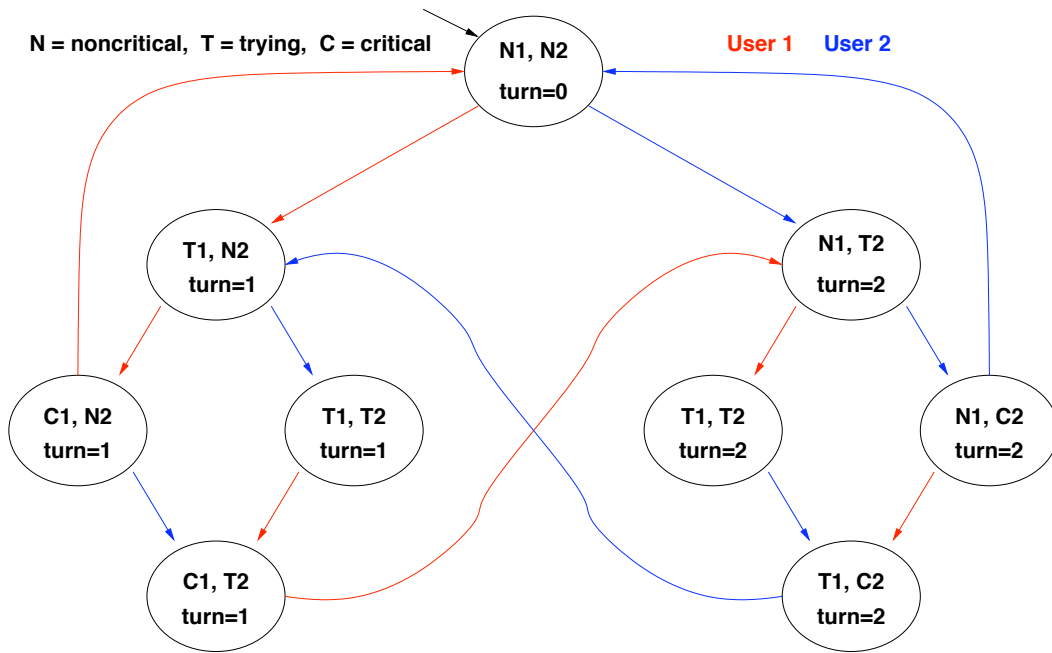
- p. 31/??

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.

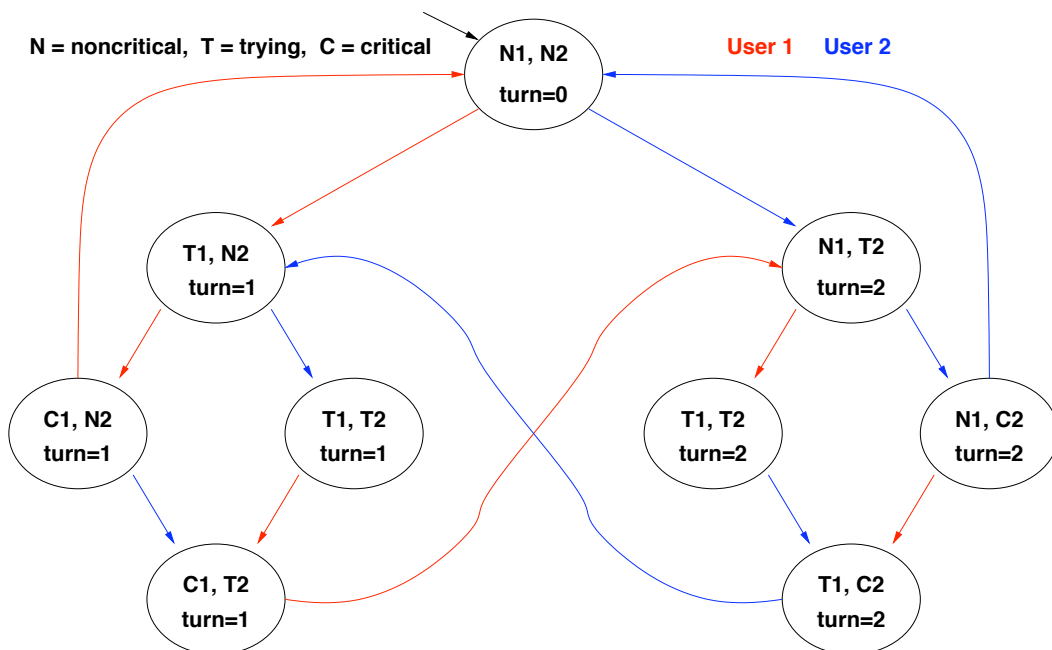
- p. 32/??

Example 1: mutual exclusion (safety)



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

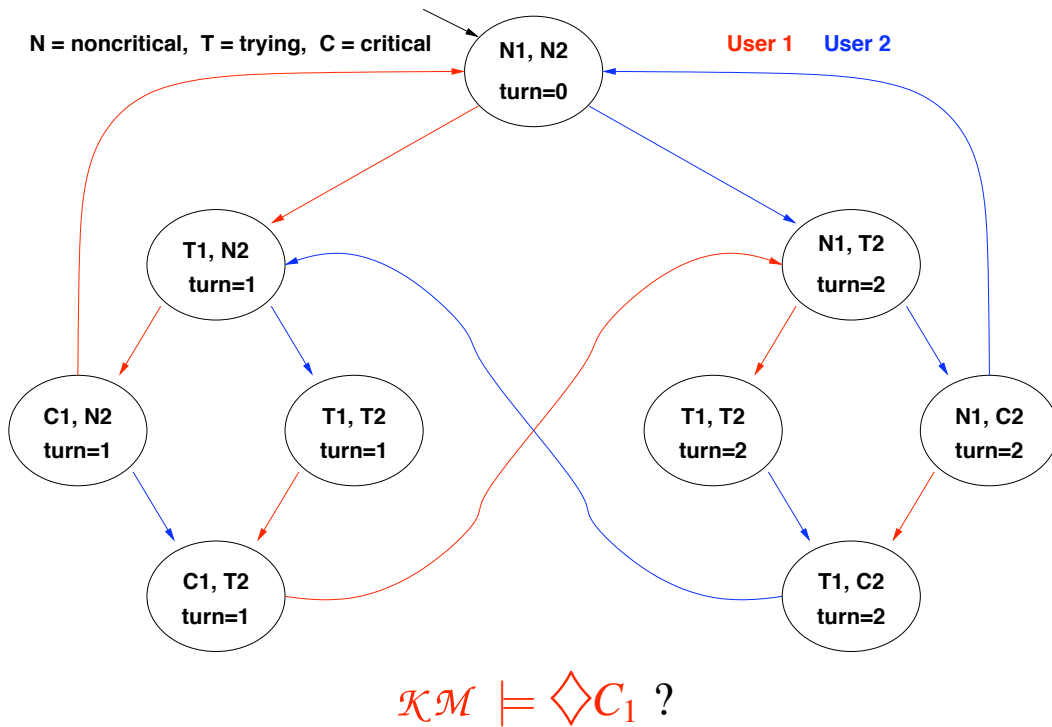
Example 1: mutual exclusion (safety)



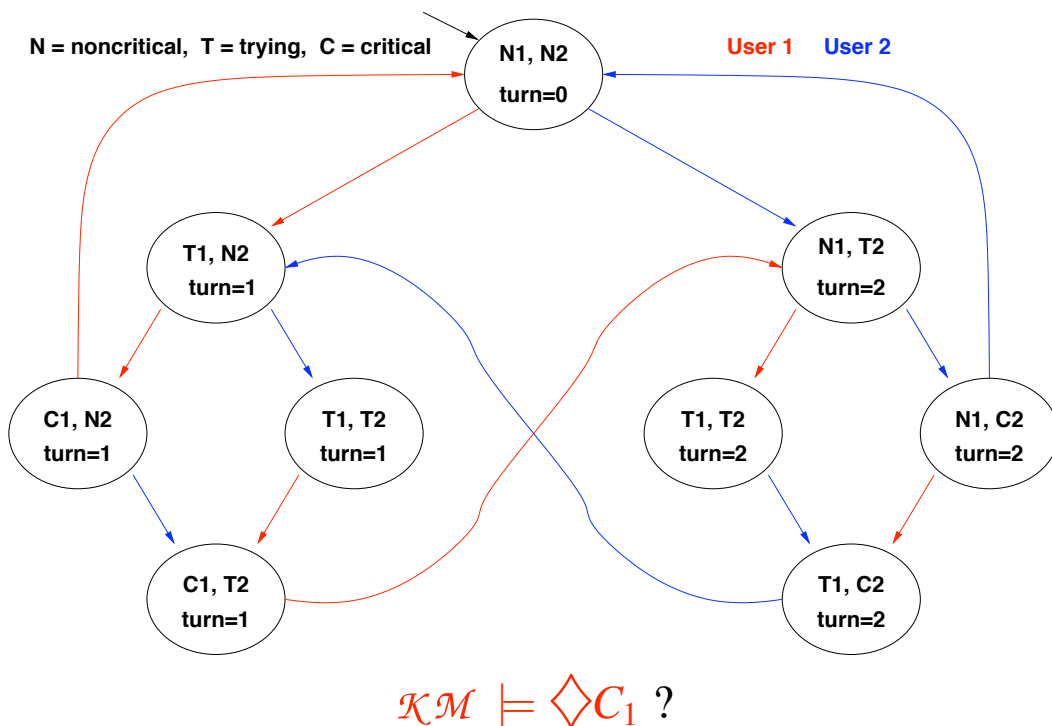
$$\mathcal{KM} \models \square \neg (C_1 \wedge C_2) ?$$

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

Example 2: mutual exclusion (liveness)

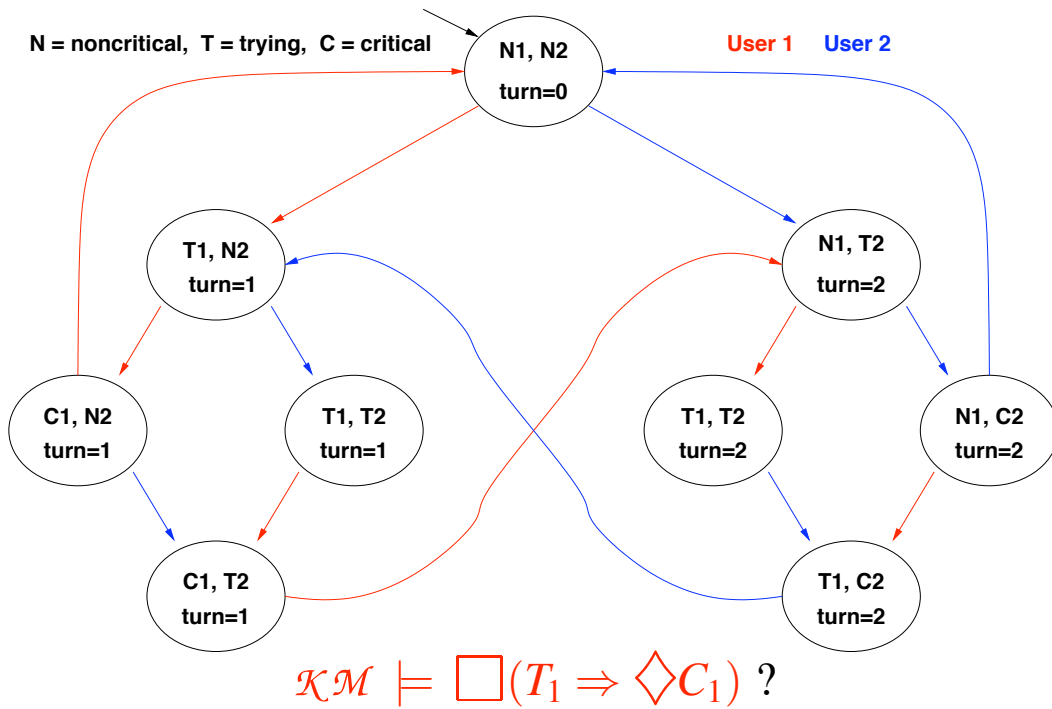


Example 2: mutual exclusion (liveness)



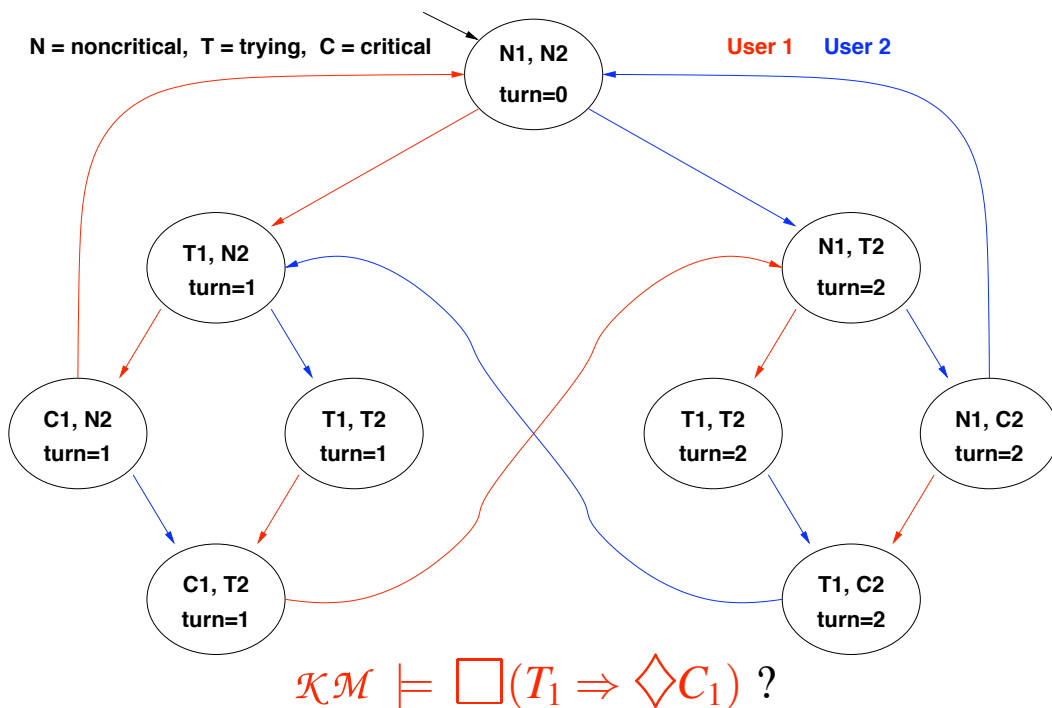
NO: the blue cyclic path is a counterexample!

Example 3: mutual exclusion (liveness)



- p. 35/??

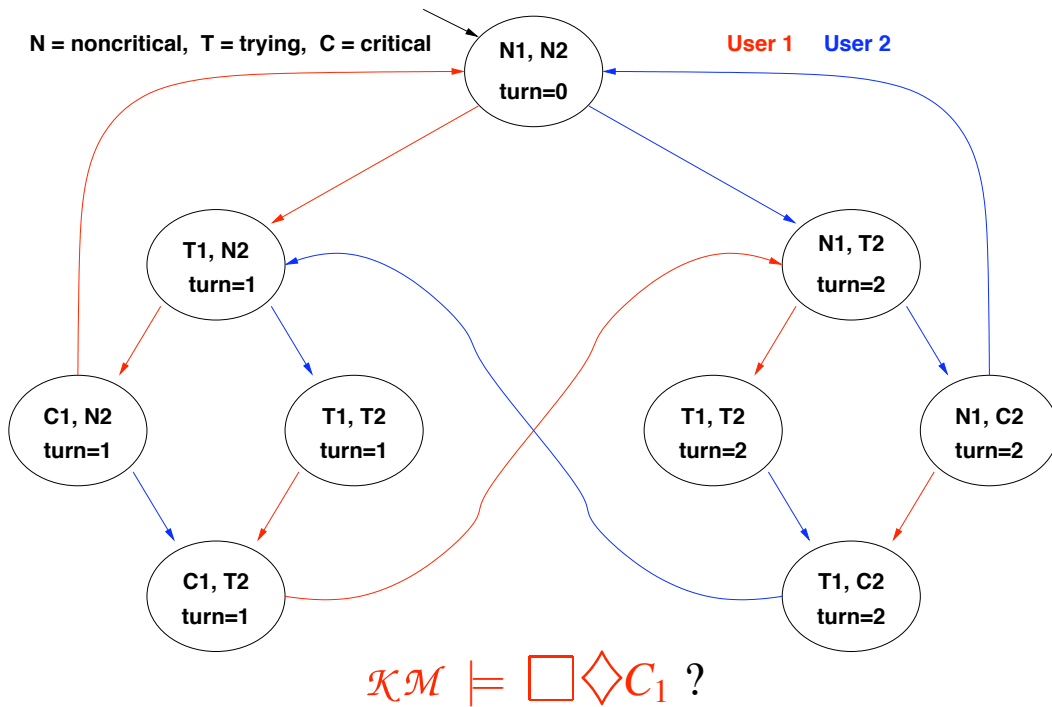
Example 3: mutual exclusion (liveness)



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

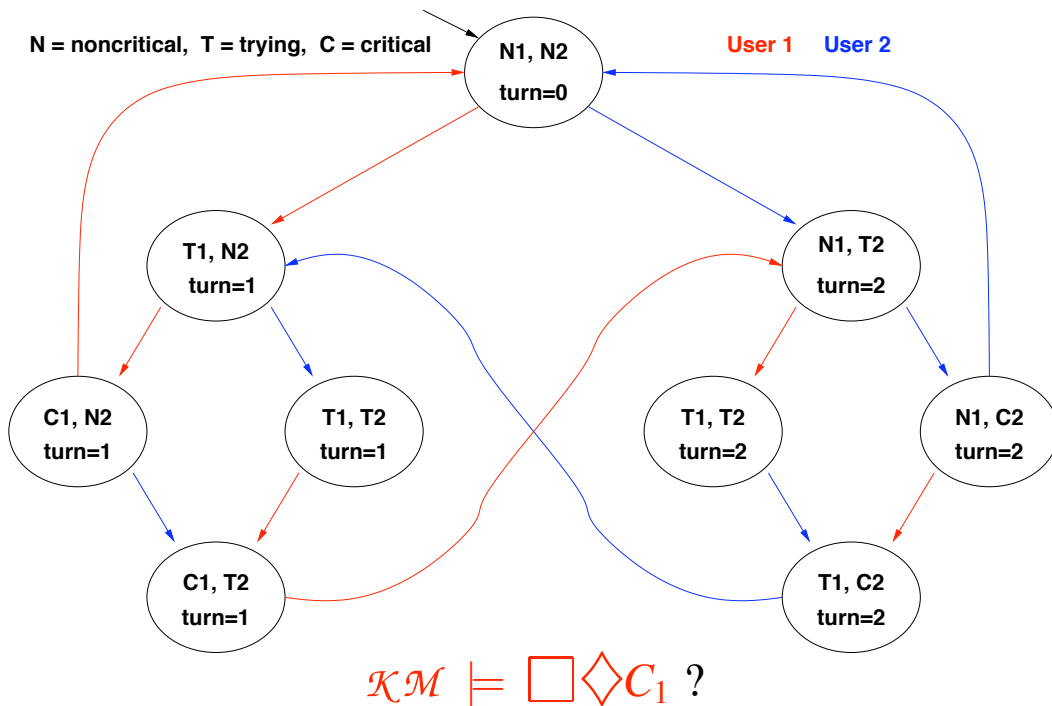
- p. 35/??

Example 4: mutual exclusion (fairness)



- p. 36??

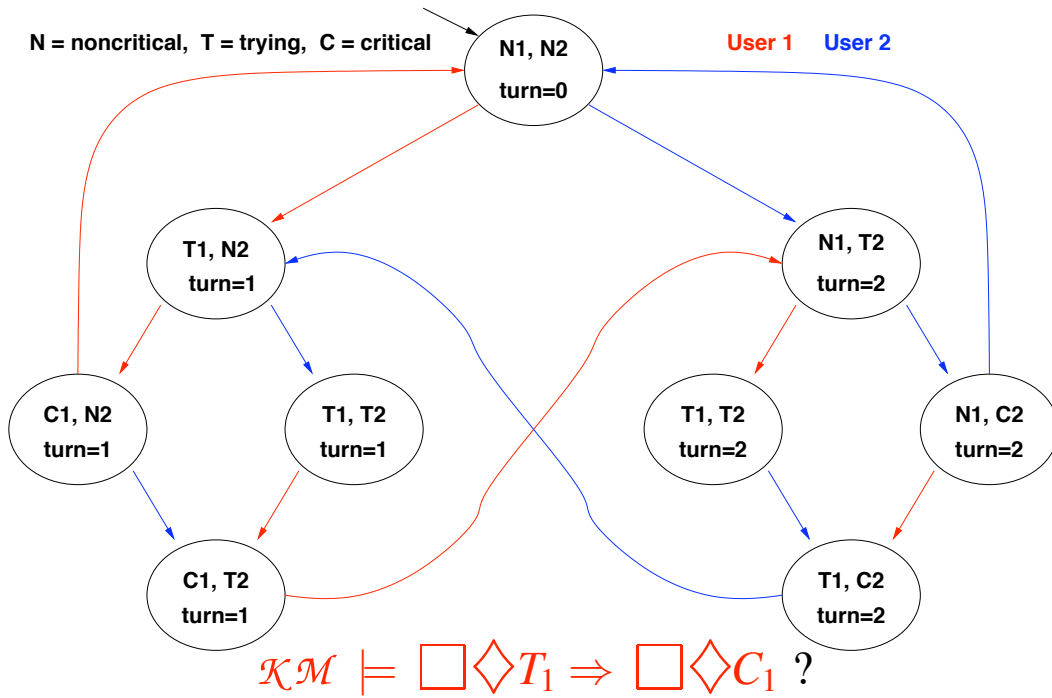
Example 4: mutual exclusion (fairness)



NO: the blue cyclic path is a counterexample!

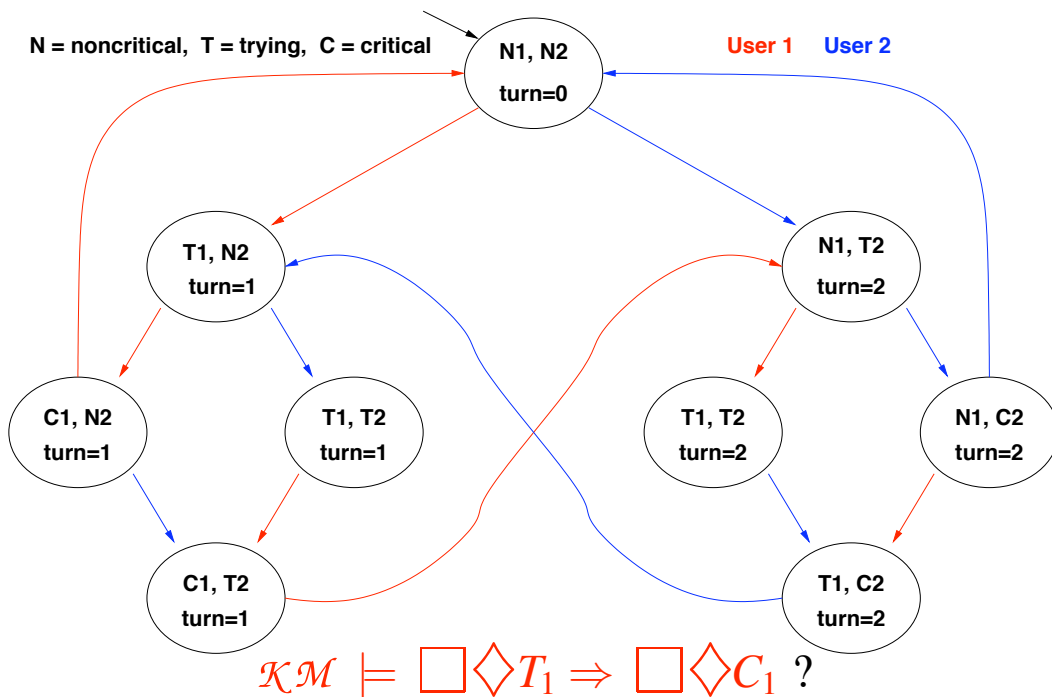
- p. 36??

Example 4: mutual exclusion (strong fairness)



- p. 37??

Example 4: mutual exclusion (strong fairness)



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

- p. 37??

LTL Alternative Notation

Alternative notations are used for temporal operators.

\diamond	\rightsquigarrow	F	sometime in the Future
\square	\rightsquigarrow	G	Globally in the future
\bigcirc	\rightsquigarrow	X	neXtime

- p. 38/??

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.

- p. 39/??