Linear-Time Temporal Logic

Bow-Yaw Wang

Institute of Information Science
Academia Sinica, Taiwan

November 16, 2021

Outline

Linear-time Temporal Logic (LTL)

- Linear-time temporal logic (LTL) models time as an infinite sequence of states.
  - Such an infinite sequence of states is called a computation path or simply path.
- LTL allows us to describe temporal properties about computation paths.
  - For instance, event P eventually happens, or event P happens until event Q does, etc.

Outline

Syntax of LTL

- Consider a fixed set $\texttt{Atoms}$ of atomic formulae $p, q, r, \ldots$.

Definition

Linear-time temporal logic has the following syntax:

$$\begin{aligned}\phi \quad ::= \quad & \top \mid \bot \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \implies \phi) \mid \\ & (\mathsf{X}\phi) \mid (\mathsf{F}\phi) \mid (\mathsf{G}\phi) \mid (\phi \,\mathsf{U}\, \phi) \mid (\phi \,\mathsf{W}\, \phi) \mid (\phi \,\mathsf{R}\, \phi)\end{aligned}$$

where $p \in \texttt{Atoms}$ is an atomic formula.

- The connectives $\mathsf{X}, \mathsf{F}, \mathsf{G}, \mathsf{U}, \mathsf{R}$, and W are temporal connectives.
- Informally, X means "neXt state," F means "some Future state," G means "all future states (Globally)," U means "Until," R means "Release," and W means "Weak-until."

Convention and Examples

- By convention, binding powers of LTL connectives are:

strongest		→	weakest
$\neg, \mathsf{X}, \mathsf{F}, \mathsf{G}$	$\mathsf{U}, \mathsf{R}, \mathsf{W}$	$\wedge, \vee$	$\Longrightarrow$

- Examples:

$$\mathsf{F}p \wedge \mathsf{G}q \Longrightarrow p \,\mathsf{W}\, r \qquad \mathsf{F}(p \Longrightarrow \mathsf{G}r) \vee \neg q \,\mathsf{U}\, p$$
$$p \,\mathsf{W}\, (q \,\mathsf{W}\, r) \qquad \mathsf{GF}p \Longrightarrow \mathsf{F}(q \vee s)$$

- Non-examples:

$$\cancel{\mathsf{U}r} \qquad \cancel{p\mathsf{G}q}$$

- A subformula of an LTL formula ϕ is a formula ψ whose parse tree is a subtree of ϕ's parse tree.

Outline

Semantics of LTL I

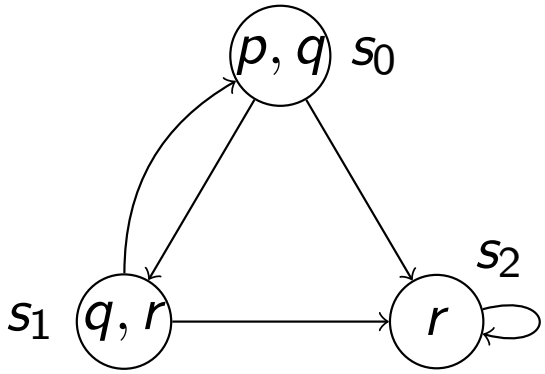
- Recall that LTL allows us to describe properties about computation paths.
- We will formalize computation paths by transition systems.

Definition

A transition system $\mathcal{M} = (S, \rightarrow, L)$ consists of a set S of states, a total transition relation $\rightarrow \subseteq S \times S$, and a labelling function $L : S \rightarrow 2^{\mathtt{Atoms}}$.

- Instead of $(s, s') \in \rightarrow$, we will write $s \rightarrow s'$.
- A transition relation $\rightarrow \subseteq S \times S$ is total if for every $s \in S$ there is an $s' \in S$ such that $s \rightarrow s'$.
- For any $s \in S$, $L(s)$ contains the set of atomic formulae which are true in s.
- A transition system is also called a model.

Semantics of LTL II



- Let $\mathcal{M} = (S, \rightarrow, L)$ with $S = \{s_0, s_1, s_2\}$, $\rightarrow = \{(s_0, s_1), (s_0, s_2), (s_1, s_0), (s_1, s_2), (s_2, s_2)\}$, and $L(s_0) = \{p, q\}$, $L(s_1) = \{q, r\}$, and $L(s_2) = \{r\}$.
- We can represent the transition system $\mathcal{M}$ as a directed graph.
- Note that transition relations must be total.
  - If the self loop at s_2 were removed, $\mathcal{M}$ would not be a transition system.
  - In order to model a state s without any outgoing transition, we add a new state s_d with a self loop and $s \rightarrow s_d$.

Computation Path and Suffix

Definition

A path in a model $\mathcal{M} = (S, \to, L)$ is an infinite sequence of states $s_0, s_1, \ldots, s_n, \ldots$ such that $s_i \to s_{i+1}$ for every $i \geq 0$. We also write $s_0 \to s_1 \to \cdots$ for the path $s_0, s_1, \ldots, s_n, \ldots$.

- For instance, $s_1 \to s_0 \to s_1 \to s_2 \to s_2 \to \cdots$ is a path in the example.

Definition

Let $\pi = s_0 \to s_1 \to \cdots$ be a path in a model $\mathcal{M} = (S, \to, L)$. The i-suffix π^i is the suffix $s_i \to s_{i+1} \to \cdots$ of π.

- Let $\pi \stackrel{\triangle}{=} s_1 \to s_0 \to s_1 \to s_2 \to s_2 \to \cdots$. We have
  - $\pi^0 \stackrel{\triangle}{=} \pi$;
  - $\pi^1 \stackrel{\triangle}{=} s_0 \to s_1 \to s_2 \to s_2 \to \cdots$;
  - $\pi^2 \stackrel{\triangle}{=} s_1 \to s_2 \to s_2 \to \cdots$;
  - $\pi^3 \stackrel{\triangle}{=} s_2 \to s_2 \to \cdots$; etc.
- Suffixes of a path are paths.

Definition

Let $\mathcal{M} = (S, \to, L)$ be a model, $\pi = s_0 \to s_1 \to \cdots$ a path in $\mathcal{M}$, and ϕ an LTL formula. Define the satisfaction relation $\pi \vDash \phi$ as follows.

1. $\pi \vDash \top$; $\pi \nvDash \bot$; and $\pi \vDash p$ if $p \in L(s_0)$;
2. $\pi \vDash \neg\phi$ if $\pi \nvDash \phi$; $\pi \vDash \phi \wedge \psi$ if $\pi \vDash \phi$ and $\pi \vDash \psi$; $\pi \vDash \phi \vee \psi$ if $\pi \vDash \phi$ or $\pi \vDash \psi$;
3. $\pi \vDash \phi \implies \psi$ if $\pi \vDash \psi$ whenever $\pi \vDash \phi$;
4. $\pi \vDash \mathsf{X}\phi$ if $\pi^1 \vDash \phi$;
5. $\pi \vDash \mathsf{G}\phi$ if $\pi^i \vDash \phi$ for every $i \geq 0$;
6. $\pi \vDash \mathsf{F}\phi$ if $\pi^i \vDash \phi$ for some $i \geq 0$;
7. $\pi \vDash \phi \,\mathsf{U}\, \psi$ if there is some $i \geq 0$ such that $\pi^i \vDash \psi$ and for every $0 \leq j < i$, $\pi^j \vDash \phi$;
8. $\pi \vDash \phi \,\mathsf{W}\, \psi$ if either there is some $i \geq 0$ such that $\pi^i \vDash \psi$ and for every $0 \leq j < i$ we have $\pi^j \vDash \phi$; or for every $k \geq 0$ we have $\pi^k \vDash \phi$;
9. $\pi \vDash \phi \,\mathsf{R}\, \psi$ if either there is some $i \geq 0$ such that $\pi^i \vDash \phi$ and for every $0 \leq j \leq i$ we have $\pi^j \vDash \psi$; or for every $k \geq 0$ we have $\pi^k \vDash \psi$.

$\pi \vDash \phi$ II

1. $\top$ is always true; $\bot$ is always false; and $\pi \vDash p$ if p is true at the start of π;
2. $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, and $\phi \Longrightarrow \psi$ have the usual semantics;
3. $\mathsf{X}\phi$ is true if ϕ is true at the 1-suffix of π;
4. $\mathsf{G}\phi$ is true if ϕ is always true in the future;
   - Note that "present" is a part of "future."
5. $\mathsf{F}\phi$ is true if ϕ is true for some future;
6. $\phi \,\mathsf{U}\, \psi$ is true if ϕ is true until exclusively ψ is true;
   - Note that ψ must be true in the future.
7. $\phi \,\mathsf{W}\, \psi$ is true if $\phi \,\mathsf{U}\, \psi$ or ϕ is always true;
   - Note that $\phi \,\mathsf{W}\, \psi$ does not require ψ to be true in the future.
8. $\phi \,\mathsf{R}\, \psi$ is true if ψ is true until inclusively ϕ releases ψ, or ψ is always true.
   - Note that $\phi \,\mathsf{R}\, \psi$ does not require ϕ to be true in the future.

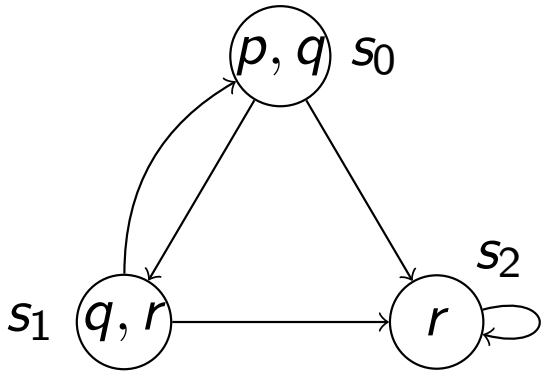
$\mathcal{M}, s \models \phi$

Definition

Let $\mathcal{M} = (S, \rightarrow, L)$ be a model, $s \in S$, and ϕ an LTL formula. $\underline{M, s \models \phi}$ if for every path π of $\mathcal{M}$ starting at s, we have $\pi \models \phi$.

- Think of the model $\mathcal{M}$ as the description of a program and s a state of the program.
- $\mathcal{M}, s \models \phi$ holds if for every possible computation path satisfies the LTL formula ϕ.
- Particularly, consider programs that depends on user inputs and the initial state s_i of such a program.
- $\mathcal{M}, s_i \models \phi$ holds if all executions of the program satisfy ϕ no matter what user inputs are.
- This is a very strong statement.
  - Much stronger than testing programs.
  - Testing can only falsify properties; it cannot prove properties.

Examples I



- $\mathcal{M}, s_0 \models p \wedge q$;
- $\mathcal{M}, s_0 \models \neg r$;
- $\mathcal{M}, s_0 \models \top$;
- $\mathcal{M}, s_0 \models \mathsf{X} r$;
- $\mathcal{M}, s_0 \not\models \mathsf{X}(q \wedge r)$;
- $\mathcal{M}, s_0 \models \mathsf{G}\neg(p \wedge r)$;
- $\mathcal{M}, s_0 \models \mathsf{GF} r$;
- $\mathcal{M}, s_0 \models \mathsf{GF} p \implies \mathsf{GF} r$;
- $\mathcal{M}, s_0 \not\models \mathsf{GF} r \implies \mathsf{GF} p$.

Outline

Patterns of Specifications

- It is impossible to go to a state where *started* holds but *ready* does not: $\mathsf{G}\neg(\mathit{started} \wedge \neg \mathit{ready})$
- For any state, if *request* holds then it will be *acknowledged* eventually: $\mathsf{G}(\mathit{request} \implies \mathsf{F}\mathit{acknowledged})$
- *enabled* occurs infinitely often: $\mathsf{GF}\mathit{enabled}$
- *stable* will eventually occurs permanently: $\mathsf{FG}\mathit{stable}$
- If a process is *enabled* infinitely often, it is *running* infinitely often: $\mathsf{GF}\mathit{enabled} \implies \mathsf{GF}\mathit{running}$

Outline

Semantically Equivalence I

Definition

Let ϕ and ψ be LTL formulae. ϕ and ψ are semantically equivalent (written $\phi \equiv \psi$) if for all paths π, $\pi \vDash \phi$ iff $\pi \vDash \psi$.

- Semantically equivalent propositional formulae are still equivalent, for example:

$$\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi \qquad \neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi.$$

- F and G are dual; X is dual to itself:

$$\neg \mathsf{G}\phi \equiv \mathsf{F}\neg\phi \qquad \neg \mathsf{F}\phi \equiv \mathsf{G}\neg\phi \qquad \neg \mathsf{X}\phi \equiv \mathsf{X}\neg\phi.$$

- U and R are dual:

$$\neg(\phi \,\mathsf{U}\, \psi) \equiv \neg\phi \,\mathsf{R}\, \neg\psi \qquad \neg(\phi \,\mathsf{R}\, \psi) \equiv \neg\phi \,\mathsf{U}\, \neg\psi.$$

Semantically Equivalence II

Lemma

$\neg(\phi \,\mathsf{U}\, \psi) \equiv \neg\phi \,\mathsf{R}\, \neg\psi$.

Proof.

Consider any path $\pi = s_0 \to s_1 \to \cdots$.
Suppose $\pi \vDash \neg(\phi \,\mathsf{U}\, \psi)$. If $\pi^i \nvDash \psi$ for all $i \geq 0$, then $\pi \vDash \neg\phi \,\mathsf{R}\, \neg\psi$ by definition. Otherwise, for every $i \geq 0$ such that $\pi^i \vDash \psi$, there is $j < i$ that $\pi^j \nvDash \phi$. Let i_0 be minimal that $\pi^{i_0} \vDash \psi$. Then $\pi^k \vDash \neg\psi$ for every $k < i_0$ by minimality and there is $j_0 < i_0$ that $\pi^{j_0} \vDash \neg\phi$. Hence $\pi \vDash \neg\phi \,\mathsf{R}\, \neg\psi$.
Conversely, suppose $\pi \vDash \neg\phi \,\mathsf{R}\, \neg\psi$. If $\pi^i \vDash \neg\psi$ for all $i \geq 0$, $\pi \vDash \neg(\phi \,\mathsf{U}\, \psi)$ by definition. Otherwise, let i_0 be minimal that $\pi^{i_0} \vDash \neg\phi$. Then $\pi^k \vDash \phi$ for all $k < i_0$ by minimality. Assume $\pi^j \vDash \psi$ for some $j \leq i_0$. Let j_0 be minimal that $\pi^{j_0} \vDash \psi$. Then $\pi^k \vDash \neg\psi$ and $\pi^k \vDash \phi$ for all $k < j_0$ but $\pi^{j_0} \vDash \psi$ and $\pi^{j_0} \vDash \phi$. We have $\phi \nvDash \neg\phi \,\mathsf{R}\, \neg\psi$. A contradiction. Hence $\pi^j \nvDash \psi$ for all $j \leq i_0$. And $\pi \vDash \neg(\phi \,\mathsf{U}\, \psi)$. □

Semantically Equivalence III

- F distributes over $\vee$ and G over $\wedge$ (why?):

$$\mathsf{F}(\phi \vee \psi) \equiv \mathsf{F}\phi \vee \mathsf{F}\psi \qquad \mathsf{G}(\phi \wedge \psi) \equiv \mathsf{G}\phi \wedge \mathsf{G}\psi.$$

  - What about F over $\wedge$ and G over $\vee$?

$$\mathsf{F}(\phi \wedge \psi) \stackrel{?}{\equiv} \mathsf{F}\phi \wedge \mathsf{F}\psi \qquad \mathsf{G}(\phi \vee \psi) \stackrel{?}{\equiv} \mathsf{G}\phi \vee \mathsf{G}\psi.$$

- F and G in U and R:

$$\mathsf{F}\phi \equiv \top \,\mathsf{U}\, \phi \qquad \mathsf{G}\phi \equiv \bot \,\mathsf{R}\, \phi.$$

- U is equivalent to W and F:

$$\phi \,\mathsf{U}\, \psi \equiv \phi \,\mathsf{W}\, \psi \wedge \mathsf{F}\psi.$$

- W and R are closely related:

$$\phi \,\mathsf{W}\, \psi \equiv \psi \,\mathsf{R}\, (\phi \vee \psi) \qquad \phi \,\mathsf{R}\, \psi \equiv \psi \,\mathsf{W}\, (\phi \wedge \psi)$$

Semantically Equivalence IV

Lemma

$\phi \,\mathsf{R}\, \psi \equiv (\phi \wedge \psi) \,\mathsf{R}\, \psi$.

Proof.

Consider any path $\pi = s_0 \to s_1 \to \cdots$.
Suppose $\pi \vDash \phi \,\mathsf{R}\, \psi$. If $\pi^i \vDash \psi$ for all $i \geq 0$, $\pi \vDash (\phi \wedge \psi) \,\mathsf{R}\, \psi$ as well.
Otherwise, there is $i \geq 0$ such that $\pi^i \vDash \phi$ and $\pi^j \vDash \psi$ for all $0 \leq j \leq i$.
Thus, $\pi^i \vDash \phi \wedge \psi$ and $\pi^j \vDash \psi$ for all $0 \leq j \leq i$. Hence $\pi \vDash (\phi \wedge \psi) \,\mathsf{R}\, \psi$.
Conversely, suppose $\pi \vDash (\phi \wedge \psi) \,\mathsf{R}\, \psi$. If $\pi^i \vDash \psi$ for all $i \geq 0$, $\pi \vDash \phi \,\mathsf{R}\, \psi$ as well. Otherwise, there is $i \geq 0$ that $\pi^i \vDash \phi \wedge \psi$ and $\pi^j \vDash \psi$ for all $0 \leq j \leq i$.
Thus, $\pi^i \vDash \phi$ and $\pi^j \vDash \psi$ for all $0 \leq j \leq i$. That is, $\pi \vDash \phi \,\mathsf{R}\, \psi$. □

Semantically Equivalence V

Lemma

$\phi \,\mathsf{W}\, \psi \equiv \psi \,\mathsf{R}\, (\phi \vee \psi)$.

Proof.

Consider any path $\pi = s_0 \to s_1 \to \cdots$.
Suppose $\pi \vDash \phi \,\mathsf{W}\, \psi$. If $\pi^i \vDash \phi$ for all $i \geq 0$, $\pi^i \vDash \phi \vee \psi$ for all $i \geq 0$.
$\pi \vDash \psi \,\mathsf{R}\, (\phi \vee \psi)$ by definition. Otherwise, there is $i \geq 0$ that $\phi^i \vDash \psi$ and $\pi^j \vDash \phi$ for all $0 \leq j < i$. Then $\pi^j \vDash \phi \vee \psi$ for all $0 \leq j \leq i$. $\pi \vDash \psi \,\mathsf{R}\, (\phi \vee \psi)$.
Conversely, suppose $\pi \vDash \psi \,\mathsf{R}\, (\phi \vee \psi)$. Consider whether $\pi^i \vDash \psi$ for some $i \geq 0$. If $\pi^i \nvDash \psi$ for all $i \geq 0$, $\pi^i \vDash \phi$ for all $i \geq 0$ since $\pi \vDash \psi \,\mathsf{R}\, (\phi \vee \psi)$.
Hence $\pi \vDash \phi \,\mathsf{W}\, \psi$. Otherwise, let i_0 be minimal that $\pi^{i_0} \vDash \psi$. Assume $\pi^j \nvDash \phi$ for some $0 \leq j < i_0$. Let $j_0 < i_0$ be minimal that $\pi^{j_0} \nvDash \phi$. Then $\pi^{j_0} \nvDash \phi \vee \psi$ and $\pi^k \nvDash \psi$ for all $0 \leq k \leq j_0$.$\pi \nvDash \psi \,\mathsf{R}\, (\phi \vee \psi)$, a contradiction.
Thus, $\pi^j \vDash \phi$ for all $0 \leq j < i_0$. Hence $\pi \vDash \phi \,\mathsf{W}\, \psi$. □

Semantically Equivalence VI

Lemma

1. $\neg(\phi \,\mathsf{R}\, \psi) \equiv \neg\phi \,\mathsf{U}\, \neg\psi$.
2. $\phi \,\mathsf{R}\, \psi \equiv \psi \,\mathsf{W}\, (\phi \wedge \psi)$.

Proof.

1. $\neg(\phi \,\mathsf{R}\, \psi) \equiv \neg(\neg\neg\phi \,\mathsf{R}\, \neg\neg\psi) \equiv \neg(\neg(\neg\phi \,\mathsf{U}\, \neg\psi)) \equiv \neg\phi \,\mathsf{U}\, \neg\psi$.
2. $\psi \,\mathsf{W}\, (\phi \wedge \psi) \equiv (\phi \wedge \psi) \,\mathsf{R}\, (\psi \vee (\phi \wedge \psi)) \equiv (\phi \wedge \psi) \,\mathsf{R}\, \psi \equiv \phi \,\mathsf{R}\, \psi$.

Semantically Equivalence VII

Theorem

$\phi \,\mathsf{U}\, \psi \equiv \neg(\neg\psi \,\mathsf{U}\, (\neg\phi \wedge \neg\psi)) \wedge \mathsf{F}\psi.$

Proof.

$$
\begin{aligned}
& \phi \,\mathsf{U}\, \psi \\
\equiv\ & \phi \,\mathsf{W}\, \psi \wedge \mathsf{F}\psi \\
\equiv\ & \psi \,\mathsf{R}\, (\phi \vee \psi) \wedge \mathsf{F}\psi \\
\equiv\ & \neg\neg(\psi \,\mathsf{R}\, (\phi \vee \psi)) \wedge \mathsf{F}\psi \\
\equiv\ & \neg(\neg\psi \,\mathsf{U}\, \neg(\phi \vee \psi)) \wedge \mathsf{F}\psi \\
\equiv\ & \neg(\neg\psi \,\mathsf{U}\, (\neg\phi \wedge \neg\psi)) \wedge \mathsf{F}\psi.
\end{aligned}
$$

Outline

Adequate Sets of LTL Connectives I

- An adequate set of connectives in a logic can express any connective in the same logic.
  - For instance, $\{\bot, \wedge, \neg\}$ is an adequate set of connectives in propositional logic.
- By semantical equivalences in LTL, we have the following adequate sets:
  - $\{\mathsf{U}, \mathsf{X}\}$. Recall $\phi \,\mathsf{R}\, \psi \equiv \neg(\neg\phi \,\mathsf{U}\, \neg\psi)$ and $\phi \,\mathsf{W}\, \psi \equiv \psi \,\mathsf{R}\, (\phi \vee \psi)$.
  - $\{\mathsf{R}, \mathsf{X}\}$. Recall that $\phi \,\mathsf{U}\, \psi \equiv \neg(\neg\phi \,\mathsf{R}\, \neg\psi)$ and $\phi \,\mathsf{W}\, \psi \equiv \psi \,\mathsf{R}\, (\phi \vee \psi)$.
  - $\{\mathsf{W}, \mathsf{X}\}$. Recall that $\phi \,\mathsf{R}\, \psi \equiv \psi \,\mathsf{W}\, (\phi \wedge \psi)$ and $\phi \,\mathsf{U}\, \psi \equiv \neg(\neg\phi \,\mathsf{R}\, \neg\psi)$.
- Note that X is independent of other connectives.

Adequate Sets of LTL Connectives II

- Consider the fragment of LTL without negation and X.
- We have the following adequate sets:
  - $\{\mathsf{U}, \mathsf{R}\}$ since $\phi \,\mathsf{W}\, \psi \equiv \psi \,\mathsf{R}\, (\phi \vee \psi)$, $\mathsf{F}\phi \equiv \top \,\mathsf{U}\, \phi$, and $\mathsf{G}\phi \equiv \bot \,\mathsf{R}\, \phi$.
  - $\{\mathsf{U}, \mathsf{W}\}$ since $\phi \,\mathsf{R}\, \psi \equiv \psi \,\mathsf{W}\, (\phi \wedge \psi)$, $\mathsf{F}\phi \equiv \top \,\mathsf{U}\, \phi$, and $\mathsf{G}\phi \equiv \bot \,\mathsf{R}\, \phi$.
  - $\{\mathsf{U}, \mathsf{G}\}$ since $\phi \,\mathsf{W}\, \psi \equiv \phi \,\mathsf{U}\, \psi \vee \mathsf{G}\phi$, $\phi \,\mathsf{R}\, \psi \equiv \psi \,\mathsf{W}\, (\phi \wedge \psi)$, and $\mathsf{F}\phi \equiv \top \,\mathsf{U}\, \phi$.
  - $\{\mathsf{R}, \mathsf{F}\}$ since $\phi \,\mathsf{W}\, \psi \equiv \psi \,\mathsf{R}\, (\phi \vee \psi)$, $\phi \,\mathsf{U}\, \psi \equiv \phi \,\mathsf{W}\, \psi \wedge \mathsf{F}\psi$, and $\mathsf{G}\phi \equiv \bot \,\mathsf{R}\, \phi$.
  - $\{\mathsf{W}, \mathsf{F}\}$ since $\phi\mathsf{U} \equiv \phi \,\mathsf{W}\, \psi \wedge \mathsf{F}\phi$, $\phi \,\mathsf{R}\, \psi \equiv \psi \,\mathsf{W}\, (\phi \wedge \psi)$, and $\mathsf{G}\phi \equiv \bot \,\mathsf{R}\, \phi$.
- Note that $\{\mathsf{R}, \mathsf{G}\}$, $\{\mathsf{W}, \mathsf{G}\}$, and $\{\mathsf{U}, \mathsf{F}\}$ are not adequate.
  - F cannot be defined by $\{\mathsf{R}, \mathsf{G}\}$ nor $\{\mathsf{W}, \mathsf{G}\}$.
  - G cannot be defined by $\{\mathsf{U}, \mathsf{F}\}$.

Outline

Outline

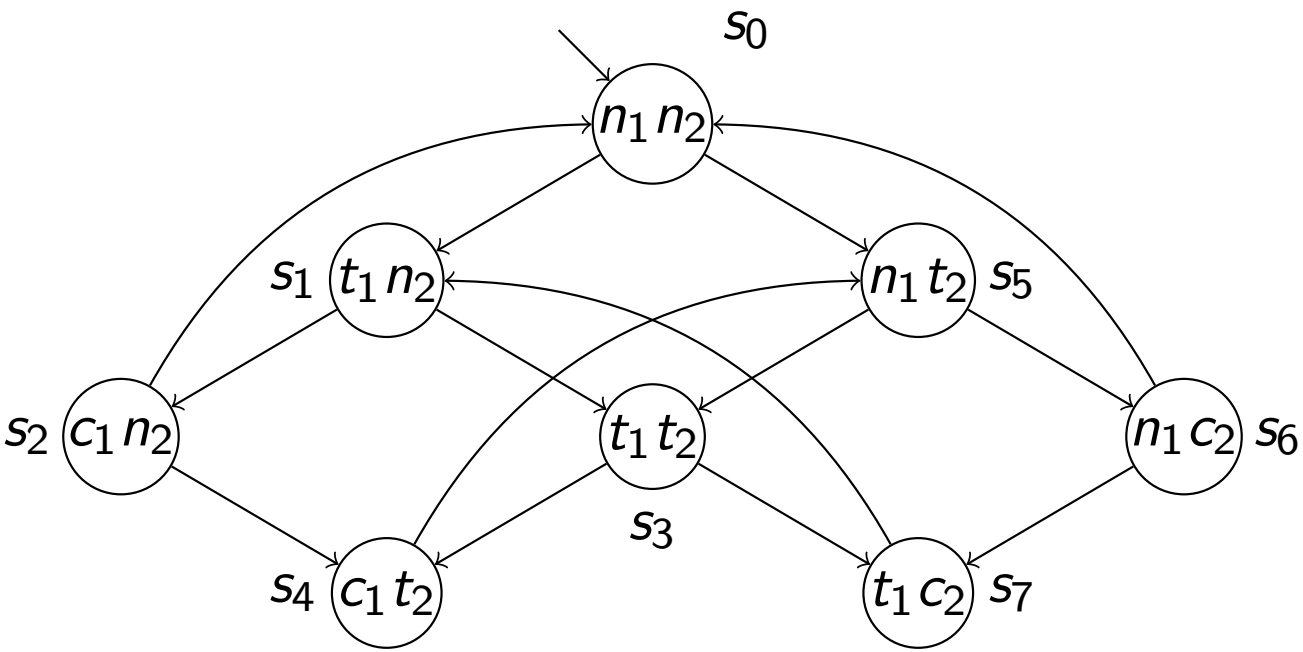
Mutual Exclusion I

- When two concurrent processes share a resource (printer, disk, etc), they sometimes need to access the resource exclusively.
- Critical sections are portions of process codes that have exclusive access to a shared resource.
- For efficiency, critical sections should be as small as possible.
- Moreover, at most one process can enter its critical section at any time.
- We will design a simple protocol to ensure mutually exclusive access to critical sections and verify our solution.

Mutual Exclusion II

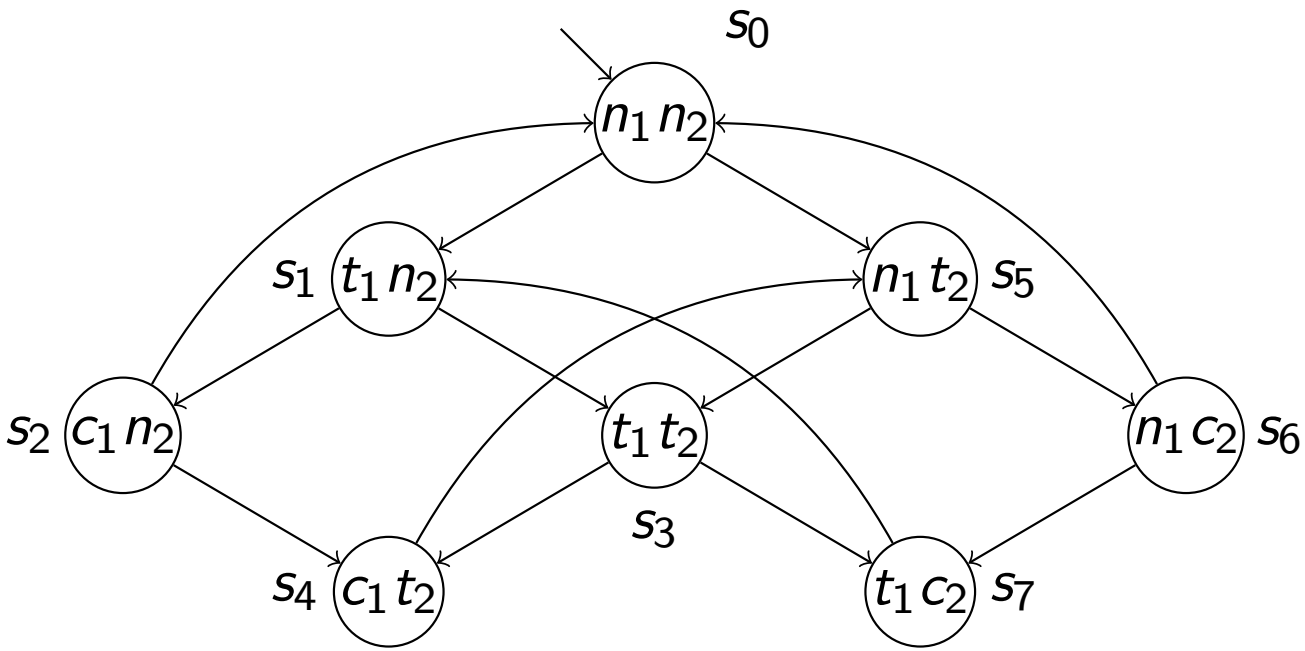
- Let us first try to specify our requirements informally.
- A protocol solving the mutual exclusion problem must ensure the following property:
  Safety: at most one process can enter its critical section at any time.
- Moreover, a protocol should not prevent any process from entering critical sections permanently:
  Liveness: When a process requests to enter its critical section, it will eventually be permitted to do so.
  Non-blocking: A process can always request to enter its critical section.

Mutual Exclusion: First Attempt I



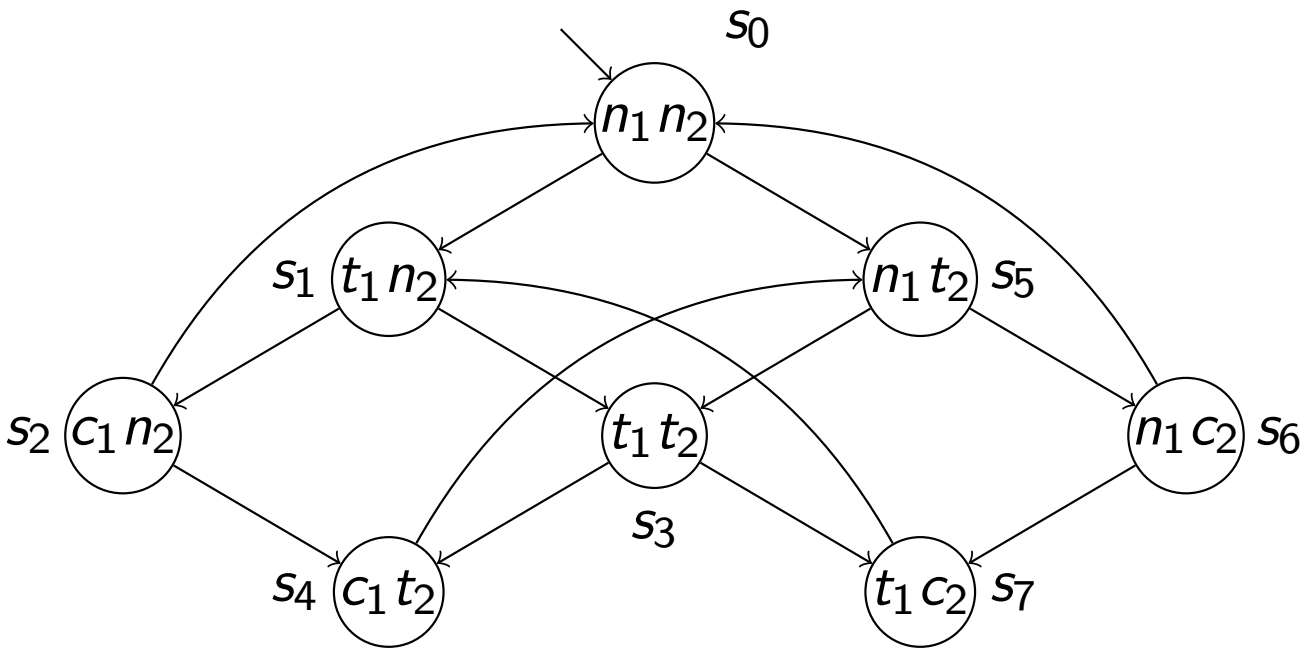
- Consider two processes P_1 and P_2.
- P_1 has three states: non-critical state (n_1), trying state (t_1), and critical state (c_1). Similarly, P_2 has states n_2, t_2, and c_2.
  - Local states are modeled as atomic formulae.
- Each process has transitions $n \to t \to c \to n \to t \to c \to \cdots$.

Mutual Exclusion: First Attempt II



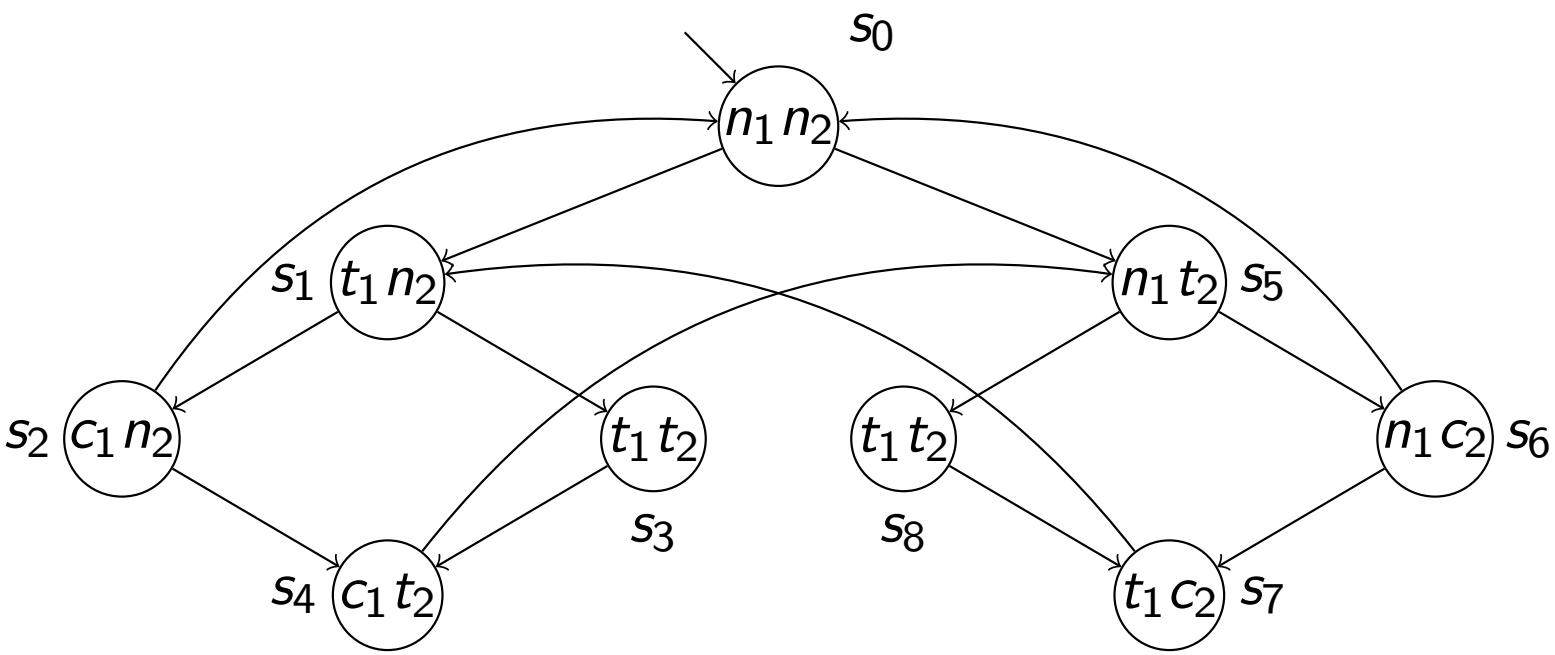
- The system starts with both processes at non-critical states (s_0).
- Exactly one process makes a transition at any time.
  - This is called an asynchronous interleaving model.

Mutual Exclusion: First Attempt III



- Safety. The property is expressed by $\mathsf{G}\neg(c_1 \wedge c_2)$ in LTL. It holds.
- Liveness. This is expressed by $\mathsf{G}(t_1 \implies \mathsf{F}c_1)$ in LTL. The property is not satisfied due to the path $s_0 \to s_1 \to s_3 \to s_7 \to s_1 \to s_3 \to s_7 \cdots$.
- Non-blocking. We would like to express: when a process at n, there is a successor at t. This is not expressible in LTL.

Mutual Exclusion: Second Attempt



- Liveness does not hold because the state s_3 does not record which process enters the trying state first.
- In our second design, we use two states to record which process enters the trying state first.
  - s_3 remembers P_1 enters t_1 first; s_8 remembers P_2 enters t_2 first.
- One can verify all three properties are satisfied.