LTL Model Checking

November 17, 2026

1 Semantics

As you probably know, we can model various discrete systems with directed graphs (nodes and arrows). The model-checking community adds labels and calls them *labelled transition systems* (LTSs). Let us see an example.

1.1 Beverage vending machine

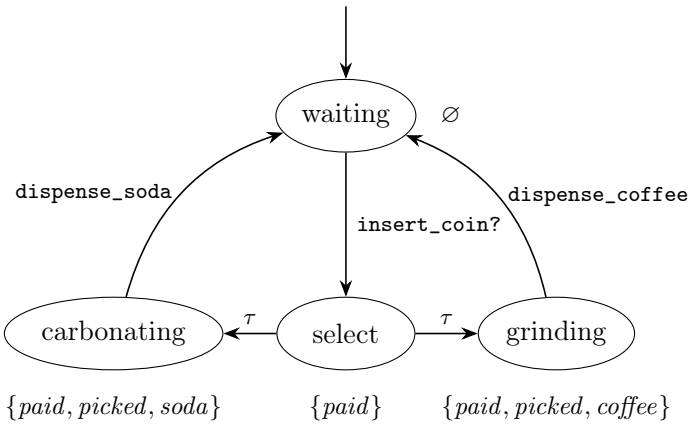


Figure 1: A labelled transition system for a simplified beverage vending machine.

Exercise 1 (Definition of LTS). Complete the definition by matching the columns. A *labelled transition system* (LTS) is a structure $\langle S, \to, L \rangle$, where[1]:

Symbol	**Meaning**
a) S	( ) is the transition relation
b) $\longrightarrow \subseteq S \times S$	( ) is the labeling function
c) $L \colon S \to \mathcal{P}(P)$	( ) is a (finite) set of states

Note: We also assume that every state s has at least one successor $s \longrightarrow s'$, so that every path through the model is infinite [1].

Exercise 2 (Reading and drawing LTSs).

[1]Here, P is the set of *atomic propositions* and $\mathcal{P}$ represents the powerset operation. The arrow $\to$ for functions should not be confused with $\longrightarrow$. L indicates which propositions are valid at which state.

Part 1 — *Reading an LTS.* Use the definition above and Figure 1 to complete the following description of the vending-machine LTS. Parts of the exercise have been complete as examples to guide you.

a) The set of states is $S = \{waiting, _____, _____, _____\}$.

b) The initial state is $_____$.

c) The set of atomic propositions (the labels) is $P = \{paid, picked, coffee, soda\}$.

d) The label of the *grinding* state is $L(grinding) = \{paid, picked, coffee\}$.

e) The label of the *waiting* state is $L(waiting) = _____$.

f) The label of the *select* state is $L(select) = \{_____\}$.

g) The label of the *carbonating* state is $L(carbonating) = _____$.

h) The transition relation is

$$\longrightarrow = \{(waiting, select), _____, _____, _____, _____\}.$$

Part 2 — *Drawing an LTS.* Notice that `dispense_soda` and `insert_coin?` are not covered in our definition of LTS. Nevertheless, since the vending machine satisfies our definition, it is an LTS. The extra information makes it a special kind of LTS. Next, we will consider an LTS with only the required information:

$$\begin{aligned} S &= \{s_0, s_1, s_2\}, \\ \longrightarrow &= \{(s_0, s_1), (s_0, s_2), (s_1, s_0), (s_1, s_2), (s_2, s_2)\}, \\ L(s_0) = \{p, q\}, \qquad L(s_1) &= \{q, r\}, \qquad L(s_2) = \{r\}. \end{aligned}$$

Draw the corresponding labelled transition system.

Exercise 3 (Identifying paths and suffixes). A *path* of an LTS $\langle S, \to, L\rangle$ is a sequence $\pi\colon \mathbb{N} \to S$ such that[2] $\pi(i) \longrightarrow \pi(i+1)$ for every $i \in \mathbb{N}$.

Part 1 — *Recognizing paths.* For each sequence displayed below, indicate with a checkmark (✓) or a cross (×) whether it represents a valid path of the LTS that you drew. Convention: to indicate that a continues forever, we will use the pattern $a \longrightarrow \cdots \longrightarrow a \longrightarrow \cdots$.

[2] That is, $(\pi(i), \pi(i+1)) \in \longrightarrow$.

( ) $\pi_1 = s_0 \longrightarrow s_1 \longrightarrow s_0 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow \cdots$

( ) $\pi_2 = s_0 \longrightarrow s_1 \longrightarrow s_1 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow \cdots$

( ) $\pi_3 = s_1 \longrightarrow s_0 \longrightarrow s_0 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow \cdots$

( ) $\pi_4 = s_0 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow s_2 \longrightarrow \cdots$

( ) $\pi_5 = s_0 \longrightarrow s_1 \longrightarrow s_0 \longrightarrow s_1 \longrightarrow s_0 \longrightarrow s_1 \longrightarrow \cdots \longrightarrow s_0 \longrightarrow s_1 \longrightarrow \cdots$

Part 2 — *Suffixes of a path.* Given a sequence of states $\pi\colon \mathbb{N} \to S$ we denote its i-th *suffix* as π^i and it corresponds to the subsequence $\pi(i) \longrightarrow \pi(i+1) \longrightarrow \pi(i+2) \longrightarrow \cdots$. Notice that $\pi^0 = \pi$.

Draw the sequences π_1^3, π_4^1, and π_5^4.

2 Syntax

With LTSs and their paths, we can model the passing of time for a system. Now, we are going to use a language to describe the temporal properties of LTSs. If P is the set of atomic propositions, the following rules define the set of *linear temporal logic* (LTL) *formulas*:

1. If $p \in P$, then p is an LTL formula.
2. Truth ($\top$) and Falsity ($\bot$) are LTL formulas.
3. If φ and ψ are LTL formulas, then
   $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, and $\varphi \leftrightarrow \psi$ are LTL formulas.
4. If φ is an LTL formula, then $\mathrm{X}\,\varphi$ (“next”) is an LTL formula.
5. If φ is an LTL formula, then $\mathrm{F}\,\varphi$ (“eventually”) is an LTL formula.
6. If φ is an LTL formula, then $\mathrm{G}\,\varphi$ (“always” or “globally”) is an LTL formula.
7. If φ and ψ are LTL formulas, then $\varphi\,\mathrm{U}\,\psi$ (“until”) is an LTL formula.
8. If φ and ψ are LTL formulas, then $\varphi\,\mathrm{W}\,\psi$ (“weak until”) is an LTL formula.

We can abbreviate these rules with the notation

$$\begin{aligned}\varphi ::= \top \mid \bot \mid p \mid \neg\varphi \mid \mathrm{X}\,\varphi \mid \mathrm{F}\,\varphi \mid \mathrm{G}\,\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \\ \mid \varphi \to \varphi \mid \varphi \leftrightarrow \varphi \mid \varphi\,\mathrm{U}\,\varphi \mid \varphi\,\mathrm{W}\,\varphi, \qquad \text{where } p \in P.\end{aligned}$$

We use parentheses to make formulas unambiguous. When parentheses are omitted, the operators have the following precedence, from strongest to weakest:

$$\{\neg, \mathrm{X}, \mathrm{F}, \mathrm{G}\} \;>\; \{\mathrm{U}, \mathrm{W}\} \;>\; \{\wedge, \vee\} \;>\; \{\to, \leftrightarrow\}.$$

Exercise 4 (Recognizing LTL formulas). Assume that $p, q, r \in P$. Indicate with a checkmark or a cross whether each of the following strings is an LTL formula.

( ) $\mathrm{G}\big(p \to \mathrm{F}(q \wedge \mathrm{X}\, r)\big)$

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

( ) $\mathrm{G}(\mathrm{U}\ p\ \ \mathrm{F}\, q)$

( ) $\mathrm{X}\, p \wedge (\mathrm{F}(q \vee r)$

3 LTL semantics

Let $\mathcal{M} = \langle S, \longrightarrow, L\rangle$ be an LTS and let $\pi = s_0 \longrightarrow s_1 \longrightarrow \cdots$ be a path in $\mathcal{M}$. We define the satisfaction relation $\pi \models \varphi$ (read "the LTL formula φ holds on π") recursively as follows:

$$
\begin{aligned}
\pi &\models \top, & \pi &\not\models \bot,\\
\pi &\models p &&\iff p \in L(s_0),\\
\pi &\models \neg\varphi &&\iff \pi \not\models \varphi,\\
\pi &\models \varphi \wedge \psi &&\iff \pi \models \varphi \text{ and } \pi \models \psi,\\
\pi &\models \varphi \vee \psi &&\iff \pi \models \varphi \text{ or } \pi \models \psi,\\
\pi &\models \varphi \to \psi &&\iff \pi \not\models \varphi \text{ or } \pi \models \psi,\\
\pi &\models \varphi \leftrightarrow \psi &&\iff (\pi \models \varphi \text{ iff } \pi \models \psi),\\
\pi &\models \mathrm{X}\,\varphi &&\iff \pi^1 \models \varphi,\\
\pi &\models \mathrm{F}\,\varphi &&\iff \text{there is an } i \in \mathbb{N} \text{ such that } \pi^i \models \varphi,\\
\pi &\models \mathrm{G}\,\varphi &&\iff \text{for every } i \in \mathbb{N},\ \pi^i \models \varphi,\\
\pi &\models \varphi \,\mathrm{U}\, \psi &&\iff \text{there is an } i \in \mathbb{N} \text{ such that } \pi^i \models \psi \text{ and } \pi^j \models \varphi \text{ for every } j < i,\\
\pi &\models \varphi \,\mathrm{W}\, \psi &&\iff \pi \models \varphi \,\mathrm{U}\, \psi \text{ or } \pi^i \models \varphi \text{ for every } i \in \mathbb{N}.
\end{aligned}
$$

Exercise 5 (Matching semantics). Assume that $p, q, r \in P$. Match each formula with the corresponding sequence that satisfies it. We abuse notation and represent the paths with sequences of the labels on the corresponding state. Notice that, absence of the proposition indicates that its negation is true.

a) $p \wedge q \wedge r$

b) $\neg p \wedge \neg q \wedge \neg r$

c) $\mathrm{X}(p \wedge \neg q \wedge \neg r)$

d) $\mathrm{F}(\neg p \wedge q \wedge \neg r)$

e) $\mathrm{G}\big((p \wedge q) \vee r\big)$

f) $r \,\mathrm{U}\, (q \wedge \neg p)$

g) $p \,\mathrm{W}\, \bot$

( ) $\rho_1 = \{p, q\} \longrightarrow \{r\} \longrightarrow \{p, r\} \longrightarrow \{q, r\} \longrightarrow \{q, r\} \longrightarrow \cdots \longrightarrow \{q, r\} \longrightarrow \cdots$

( ) $\rho_2 = \{p, q\} \longrightarrow \{p\} \longrightarrow \varnothing \longrightarrow \varnothing \longrightarrow \varnothing \longrightarrow \cdots \longrightarrow \varnothing \longrightarrow \cdots$

( ) $\rho_3 = \{p\} \longrightarrow \{p, q\} \longrightarrow \{p\} \longrightarrow \{p, r\} \longrightarrow \{p\} \longrightarrow \cdots \longrightarrow \{p\} \longrightarrow \cdots$

( ) $\rho_4 = \{p, q, r\} \longrightarrow \varnothing \longrightarrow \varnothing \longrightarrow \varnothing \longrightarrow \varnothing \longrightarrow \cdots$

( ) $\rho_5 = \{p, r\} \longrightarrow \{p, r\} \longrightarrow \{p, r\} \longrightarrow \{q, r\} \longrightarrow \varnothing \longrightarrow \cdots$

( ) $\rho_6 = \varnothing \longrightarrow \{p, q\} \longrightarrow \{p, q\} \longrightarrow \{p, q\} \longrightarrow \{p, q\} \longrightarrow \cdots$

( ) $\rho_7 = \{p\} \longrightarrow \{p, q\} \longrightarrow \{q\} \longrightarrow \varnothing \longrightarrow \varnothing \longrightarrow \cdots$

4 The model-checking problem

So far, the satisfaction relation $\models$ associates a single path with a formula. However, we want to make sure that our whole system respects a property (why?). To do so, we require that *every* possible execution of the LTS satisfies it.

Let $\mathcal{M} = \langle S, \longrightarrow, L \rangle$ be an LTS, let $s \in S$ and let φ be an LTL formula. We write

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

if $\pi \models \varphi$ for every path π of $\mathcal{M}$ with $\pi(0) = s$. When $\mathcal{M}$ is clear from the context we abbreviate $\mathcal{M}, s \models \varphi$ by $s \models \varphi$.

The *model-checking problem* is then: given $\mathcal{M}$, s and φ, decide whether $\mathcal{M}, s \models \varphi$ holds.

Notice that a single path π starting at s with $\pi \not\models \varphi$ is enough to refute $\mathcal{M}, s \models \varphi$; such a path is called a *counterexample*. Notice also that $\mathcal{M}, s \not\models \varphi$ is *not* the same as $\mathcal{M}, s \models \neg\varphi$: the former says that a path starting at s does not satisfy φ, the latter, that *all* of them do.

Exercise 6 (Model checking LTSs).

Part 1 — *Three-state LTS.* Consider the LTS $\mathcal{M}$ of Figure 2, (solution to the "Drawing an LTS" exercise) with $S = \{s_0, s_1, s_2\}$ and $P = \{p, q, r\}$.

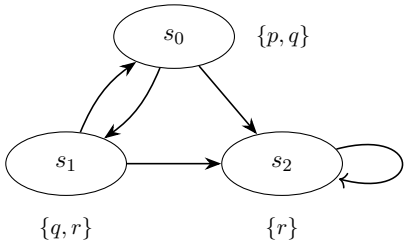


Figure 2: The 3-state LTS $\mathcal{M}$ for the exercises below.

Indicate with a checkmark (✓) or a cross (×) whether each of the following statements holds. For each statement that you mark with a cross, write down a counterexample path.

( ) $\mathcal{M}, s_0 \models p \wedge q$

( ) $\mathcal{M}, s_0 \models \neg r$

( ) $\mathcal{M}, s_0 \models \mathrm{X}\, r$

( ) $\mathcal{M}, s_0 \models \mathrm{X}\, q$

( ) $\mathcal{M}, s_0 \models \mathrm{G}\, \neg(p \wedge r)$

( ) $\mathcal{M}, s_2 \models \mathrm{G}\, r$

( ) $\mathcal{M}, s_0 \models \mathrm{G}\, r$

( ) $\mathcal{M}, s_0 \models q\, \mathrm{U}\, r$

( ) $\mathcal{M}, s_0 \models \mathrm{F}\, \mathrm{G}\, r$

( ) $\mathcal{M}, s_1 \models \mathrm{F}(\neg q \wedge r) \rightarrow \mathrm{F}\, \mathrm{G}\, r$

( ) $\mathcal{M}, s_0 \models \mathrm{G}\, \mathrm{F}\, p \rightarrow \mathrm{G}\, \mathrm{F}\, r$

( ) $\mathcal{M}, s_0 \models \mathrm{G}\, \mathrm{F}\, r \rightarrow \mathrm{G}\, \mathrm{F}\, p$

Part 2 — *Beverage vending machine.* Repeat the same activity for the beverage vending machine $\mathcal{B}$ of Figure 1, starting in the *waiting* state.

( ) $\mathcal{B}, \mathit{waiting} \models \mathrm{G}(\mathit{picked} \rightarrow \mathit{paid})$

( ) $\mathcal{B}, \mathit{waiting} \models \mathrm{G}(\mathit{paid} \rightarrow \mathrm{F}\ \mathit{picked})$

( ) $\mathcal{B}, \mathit{waiting} \models \mathrm{G}\,\mathrm{F}\ \mathit{picked}$

( ) $\mathcal{B}, \mathit{waiting} \models \mathrm{G}\,\mathrm{F}\ \mathit{coffee}$

( ) $\mathcal{B}, \mathit{waiting} \models \mathrm{F}\ \mathit{coffee}$

Exercise 7 (Practical specification patterns). Which properties about systems do we want to check in practice? Several patterns appear frequently and recognising them is useful. In the following exercise the atomic propositions describe a real system, so we write them in sans serif, as in busy or $\mathsf{requested}$.

Part 1 — *Matching.* Match each statement with the LTL formula that formalises it. Write your candidate formalisation first next to the statement. Then verify whether your candidate appears in the column of solutions.

a) busy

b) $\mathrm{G}\,\mathsf{ready}$

c) $\mathrm{G}\,\mathrm{F}\,\mathsf{enabled}$

d) $\mathsf{busy}\ \mathrm{U}\ \mathsf{acknowledged}$

e) $\mathrm{G}\,\neg(\mathsf{started} \wedge \neg\mathsf{ready})$

f) $\mathrm{X}\,\mathsf{busy}$

g) $\mathrm{F}\,\mathrm{G}\,\mathsf{deadlock}$

h) $\mathrm{G}(\mathsf{requested} \rightarrow \mathrm{F}\,\mathsf{acknowledged})$

i) $\mathrm{F}\,\mathsf{acknowledged}$

j) $\mathrm{G}\,\neg\mathsf{deadlock}$

k) $\mathrm{G}\,\mathrm{F}\,\mathsf{enabled} \rightarrow \mathrm{G}\,\mathrm{F}\,\mathsf{running}$

l) $\mathrm{G}\big(\mathsf{floor2} \wedge \mathsf{up} \wedge \mathsf{button5} \rightarrow (\mathsf{up}\ \mathrm{U}\ \mathsf{floor5})\big)$

( ) The system starts busy.

( ) Every request is eventually acknowledged.

( ) The system is busy in the second state of the execution.

( ) The process can be enabled infinitely often.

( ) The system is ready at every moment.

( ) It is impossible to reach a state in which the system has started but is not ready.

( ) The system is never deadlocked.

( ) An acknowledgement arrives at some point.

( ) The process eventually becomes permanently deadlocked.

( ) The system stays busy until an acknowledgement arrives (and an acknowledgement does arrive).

( ) If the process is enabled infinitely often, then it runs infinitely often.

( ) An upwards-travelling lift at the second floor does not change its direction while it carries passengers who want to go to the fifth floor.

Part 2 — *Beyond LTL.* Try now to formalise the three statements below. For each one, write your best candidate formula, and then use the hint to decide whether the candidate really says what the English statement says.

1. From any state it is possible to reach a restart state.

   Hint. A tempting candidate is $\mathrm{G}\,\mathrm{F}\,\mathsf{restart}$. Why would that not correspond to the desired property? Recall that $\mathcal{M}, s \models \varphi$ asks for every path starting at s.

2. The lift can remain idle on the third floor with its doors closed.

   Hint. The word "can" asks for *one* path with a given property. Which paths do the statements $\mathcal{M}, s \models \varphi$ and $\mathcal{M}, s \models \neg\varphi$ talk about. Does either of the two options give you "there is a path"?

3. It is possible to reach a state in which the system has started but is not ready (the negation of statement 6 of Part 1).

   Hint. The intuitive candidate is the negation of formula (e), that is, $\mathrm{F}(\mathsf{started} \wedge \neg\mathsf{ready})$. Compare what $\mathcal{M}, s \models \mathrm{F}(\mathsf{started} \wedge \neg\mathsf{ready})$ says with what $\mathcal{M}, s \not\models \mathrm{G}\,\neg(\mathsf{started} \wedge \neg\mathsf{ready})$ says. Which of the two is "it is possible"? And is the second one an LTL formula at all?

5 Computation Tree logic

From the previous exercise, you can see that we cannot express all interesting properties in LTL. Below, we write the definition of the syntax and semantics of a logic that addresses this issue, Computation Tree Logic (CTL).

5.1 CTL syntax

CTL distinguishes between *state formulas*, which are evaluated at a state, and *path formulas*, which describe a path starting at that state. For $p \in P$, their syntax is given by

$$\begin{aligned} \varphi &::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists\psi \mid \forall\psi, && \text{state formulas,} \\ \psi &::= \bigcirc\varphi \mid \varphi \,\mathrm{U}\, \varphi \mid \Box\varphi \mid \Diamond\varphi, && \text{path formulas.} \end{aligned}$$

Here, $\forall$ will be interpreted as "on every path" and $\exists$, "on some path". The symbols $\bigcirc$, $\Box$, $\Diamond$ mean "next", "always" and "eventually", respectively.[3] A different presentation of the same syntax appears below. Notice that, therefore, the concatenations $\forall\Box(\Diamond(\varphi))$ and $\exists\Diamond(\Box(\varphi))$ are not CTL formulas. The corresponding well-formed versions are $\forall\Box(\forall\Diamond(\varphi))$ and $\exists\Diamond(\exists\Box(\varphi))$.

$$\begin{aligned} \varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \\ \mid \forall\bigcirc(\varphi) \mid \exists\bigcirc(\varphi) \mid \forall\Diamond(\varphi) \mid \exists\Diamond(\varphi) \mid \forall\Box(\varphi) \mid \exists\Box(\varphi) \mid \forall(\varphi \,\mathrm{U}\, \varphi) \mid \exists(\varphi \,\mathrm{U}\, \varphi). \end{aligned}$$

[3] We changed notation to get you used to the two standard notations used in the literature.

5.2 CTL semantics

Let $\mathcal{M} = \langle S, \longrightarrow, L \rangle$ be an LTS and $s \in S$. We define when a CTL state formula φ holds at s, written $\mathcal{M}, s \models_{\mathsf{CTL}} \varphi$, as follows:

$$
\begin{aligned}
\mathcal{M}, s &\models_{\mathsf{CTL}} \top \\
\mathcal{M}, s &\models_{\mathsf{CTL}} p &&\iff p \in L(s), \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \neg\varphi &&\iff \mathcal{M}, s \not\models_{\mathsf{CTL}} \varphi, \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \varphi \wedge \chi &&\iff \mathcal{M}, s \models_{\mathsf{CTL}} \varphi \text{ and } \mathcal{M}, s \models_{\mathsf{CTL}} \chi, \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \forall\bigcirc(\varphi) &&\iff \mathcal{M}, s' \models_{\mathsf{CTL}} \varphi \text{ for every } s' \text{ with } s \longrightarrow s', \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \exists\bigcirc(\varphi) &&\iff \mathcal{M}, s' \models_{\mathsf{CTL}} \varphi \text{ for some } s' \text{ with } s \longrightarrow s'.
\end{aligned}
$$

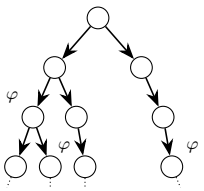
For the remaining operators, let π range over paths starting at s:

$$
\begin{aligned}
\mathcal{M}, s &\models_{\mathsf{CTL}} \forall\Diamond(\varphi) &&\iff \text{for every } \pi, \text{ some } i \in \mathbb{N} \text{ satisfies } \mathcal{M}, \pi(i) \models_{\mathsf{CTL}} \varphi, \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \exists\Diamond(\varphi) &&\iff \text{for some } \pi, \text{ some } i \in \mathbb{N} \text{ satisfies } \mathcal{M}, \pi(i) \models_{\mathsf{CTL}} \varphi, \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \forall\Box(\varphi) &&\iff \text{for every } \pi \text{ and every } i \in \mathbb{N}, \mathcal{M}, \pi(i) \models_{\mathsf{CTL}} \varphi, \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \exists\Box(\varphi) &&\iff \text{for some } \pi \text{ and every } i \in \mathbb{N}, \mathcal{M}, \pi(i) \models_{\mathsf{CTL}} \varphi, \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \forall(\varphi \,\mathrm{U}\, \chi) &&\iff \text{for every } \pi, \text{ some } i \in \mathbb{N} \text{ satisfies } \mathcal{M}, \pi(i) \models_{\mathsf{CTL}} \chi \\
& &&\quad \text{and } \mathcal{M}, \pi(j) \models_{\mathsf{CTL}} \varphi \text{ for every } j < i, \\
\mathcal{M}, s &\models_{\mathsf{CTL}} \exists(\varphi \,\mathrm{U}\, \chi) &&\iff \text{for some } \pi, \text{ some } i \in \mathbb{N} \text{ satisfies } \mathcal{M}, \pi(i) \models_{\mathsf{CTL}} \chi \\
& &&\quad \text{and } \mathcal{M}, \pi(j) \models_{\mathsf{CTL}} \varphi \text{ for every } j < i.
\end{aligned}
$$

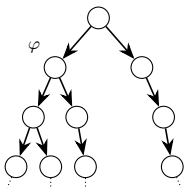
Exercise 8 (CTL operators). Match each computation-tree diagram with the CTL formula that it illustrates. A label φ or ψ marks a state where that formula holds; unlabelled states are irrelevant to the property.

a) $\exists\bigcirc(\varphi)$
b) $\forall\bigcirc(\varphi)$
c) $\exists\Diamond(\varphi)$
d) $\forall\Diamond(\varphi)$
e) $\exists\Box(\varphi)$
f) $\forall\Box(\varphi)$
g) $\exists(\varphi \,\mathrm{U}\, \psi)$
h) $\forall(\varphi \,\mathrm{U}\, \psi)$

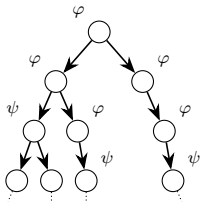
( )



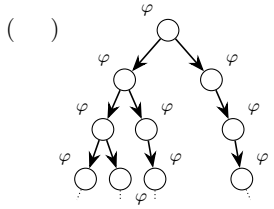
( )



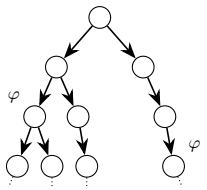
( )



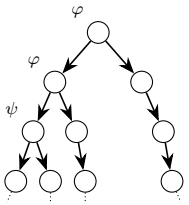
( )



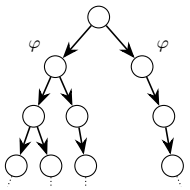
( )



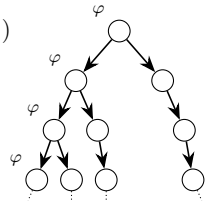
( )



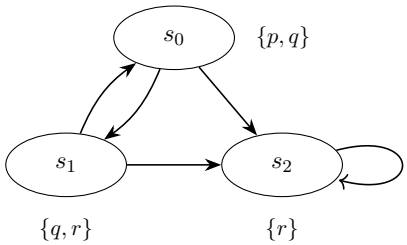
( )



( )



Exercise 9 (CTL model checking). Consider again the LTS $\mathcal{M}$ below.



Indicate with a checkmark (✓) or a cross (×) whether each statement holds.

( ) $\mathcal{M}, s_0 \models_{\mathsf{CTL}} \exists\bigcirc(q \wedge r)$

( ) $\mathcal{M}, s_0 \models_{\mathsf{CTL}} \forall\bigcirc(q \wedge r)$

( ) $\mathcal{M}, s_0 \models_{\mathsf{CTL}} \neg\exists\Diamond(p \wedge r)$

( ) $\mathcal{M}, s_2 \models_{\mathsf{CTL}} \exists\Box(r)$

( ) $\mathcal{M}, s_0 \models_{\mathsf{CTL}} \forall\Diamond(r)$

( ) $\mathcal{M}, s_0 \models_{\mathsf{CTL}} \exists((p \wedge q)\,\mathrm{U}\,r)$

( ) $\mathcal{M}, s_0 \models_{\mathsf{CTL}} \forall(p\,\mathrm{U}\,r)$

( ) $\mathcal{M}, s_0 \models_{\mathsf{CTL}} \forall\Box\big((p \vee q \vee r) \rightarrow \exists\Diamond(\exists\Box(r))\big)$

Exercise 10 (CTL specification patterns). Formalise each statement in CTL. If this is impossible, say why. Hint: recall that intermediate $\forall$ and $\exists$ will be necessary because some operators must be applied to state formulas.

1. Some reachable start state is not active.
2. Every request is eventually granted.
3. The process is enabled infinitely often on every path.
4. The process will eventually remain deadlocked forever.

5. From every state it is possible to reach a restart state.

6. A lift can always remain idle on the floor3 with its doorclosed.
   Hint: define $\alpha = \mathsf{floor3} \wedge \mathsf{idle} \wedge \mathsf{doorclosed}$ and from there formalise "can always remain".

7. A process can always request to enter its critical section next, from its noncritical one.

8. Processes need not enter their critical sections in strict sequence.
   Hint: If $\mathsf{critical}_1$ and $\mathsf{critical}_2$ represent the critical sections of processes 1 and 2, respectively, this could be expressed as "process 1 can enter its critical section twice without process 2 entering its critical section in between." You only need nested conjunctions and U operators for this.

9. On every path on which a process is enabled infinitely often, it runs infinitely often (on that same path).
   Hint: Why "$\forall\Box(\forall\Diamond(\mathsf{enabled})) \rightarrow \forall\Box(\forall\Diamond(\mathsf{running}))$" does not work?
   Does the LTL formula "$\mathrm{G\,F}\,\mathsf{enabled} \rightarrow \mathrm{G\,F}\,\mathsf{running}$" formalise it?

Comment (beyond this lecture). CTL* combines CTL path quantifiers with LTL path formulas. Its state formulas φ and path formulas ψ have the syntax

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall\psi \mid \exists\psi,$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid \bigcirc(\psi) \mid \Box(\psi) \mid \Diamond(\psi) \mid (\psi\ \mathrm{U}\ \psi).$$

Thus, the last property can be written $\forall\big(\Box(\Diamond(\mathsf{enabled})) \rightarrow \Box(\Diamond(\mathsf{running}))\big)$: the outer $\forall$ considers one path at a time, so both sides of the implication concern that same path. CTL* will not be studied in this course.

References

[1] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Second edition. Cambridge University Press, 2004.

[2] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.