

Semantics

Let $\pi = s_0, s_1, s_2 \dots$ be a sequence of states and F be an LTL formula. We define the notion F is true on π , denoted by $\pi \models F$, by induction on F as follows. For all $i = 0, 1, \dots$ denote by π_i the sequence of states $s_i, s_{i+1}, s_{i+2} \dots$ (note that $\pi_0 = \pi$).

1. $\pi \models \top$ and $\pi \not\models \perp$.
2. $\pi \models x = v$ if $s_0 \models x = v$.
3. $\pi \models F_1 \wedge \dots \wedge F_n$ if for all $j = 1, \dots, n$ we have $\pi \models F_j$;
 $\pi \models F_1 \vee \dots \vee F_n$ if for some $j = 1, \dots, n$ we have $\pi \models F_j$.
4. $\pi \models \neg F$ if $\pi \not\models F$.
5. $\pi \models F \rightarrow G$ if either $\pi \not\models F$ or $\pi \models G$;
 $\pi \models F \leftrightarrow G$ if either both $\pi \not\models F$ and $\pi \not\models G$ or both $\pi \models F$

and $\pi \models G$.

6. $\pi \models \bigcirc F$ if $\pi_1 \models F$;

$\pi \models \diamond F$ if for some $i = 0, 1, \dots$ we have $\pi_i \models F$;

$\pi \models \square F$ if for all $i = 0, 1, \dots$ we have $\pi_i \models F$.

7. $\pi \models F \mathbf{U} G$ if for some $k = 0, 1, \dots$ we have $\pi_k \models G$ and

$\pi_0 \models F, \dots, \pi_{k-1} \models F$;

$\pi \models F \mathbf{R} G$ if for all $k \geq 0$, either $\pi_k \models G$ or there exists $j < k$ such that $\pi_j \models F$.

Two LTL formulas F and G are called **equivalent**, denoted

$F \equiv G$, if for every path π we have $\pi \models F$ if and only if $\pi \models G$.

Semantics

For an LTL formula F we can consider at least two kinds of properties of K :

1. does F hold on **some** computation path for K from an initial state?
2. does F hold on **all** computation paths for K from an initial state?

Examples

- ▶ Reachability and safety properties.
- ▶ Mutual exclusion.
- ▶ Deadlock.
- ▶ Termination and finiteness.
- ▶ Fairness.
- ▶ **Responsiveness:** every request will be eventually acknowledged.
- ▶ Alternation.

Precedences of Connectives and Temporal Operators

Connective	Precedence
$\neg, \bigcirc, \diamond, \square$	5
U, R	4
\wedge, \vee	3
\rightarrow	2
\leftrightarrow	1

Expressing Some Properties

1. F never holds at two consecutive states.
2. If F holds at a state s , it also holds at all states after s .
3. F holds at at most one state.
4. F holds at at least two states.
5. If A holds at a state s_i , then B must hold at at least one of the two states just before s_i , that is s_{i-1} and s_{i-2} .
6. A happens infinitely often.

Meaning of Some Formulas

$$\square \diamond F$$

$$\diamond \square F$$

$$\square (F \rightarrow \bigcirc F)$$

$$\neg F \mathbf{U} \square F$$

$$F \mathbf{U} \neg F$$

$$\diamond F \wedge \square (F \rightarrow \bigcirc F)$$

Formalization: Variables and Domains

variable	domain	explanation
st_coffee	{0, 1}	drink storage contains coffee
st_beer	{0, 1}	drink storage contains beer
disp	{ <i>none, beer, coffee</i> }	content of drink dispenser
coins	{0, 1, 2, 3}	number of coins in the slot
customer	{ <i>none, student, prof</i> }	customer

Transitions

1. *Recharge* which results in the drink storage having both beer and coffee.
2. *Customer_arrives*, after which a customer appears at the machine.
3. *Customer_leaves*, after which the customer leaves.
4. *Coin_insert*, when the customer inserts a coin in the machine.
5. *Dispense_beer*, when the customer presses the button to get a can of beer.
6. *Dispense_coffee*, when the customer presses the button to get a cup of coffee.
7. *Take_drink*, when the customer removes a drink from the dispenser.

Reasoning About Transitions

Consider the following properties:

1. “one cannot get two beers in a row without inserting a coin”.
2. “If we never have two recharge transitions in a row, then the next transition after a recharge must be a customer arrival”.

Note that they are about transitions, not about states.

How can one represent these properties?

Reasoning About Transitions

Consider the following properties:

1. “one cannot get two beers in a row without inserting a coin”.
2. “If we never have two recharge transitions in a row, then the next transition after a recharge must be a customer arrival”.

Note that they are about transitions, not about states.

How can one represent these properties?

Introduce a state variable denoting the next transition.

Symbolic Representation

Recharge $\stackrel{\text{def}}{=}$ $\text{tr} = \text{Recharge} \wedge \text{customer} = \text{none} \wedge$
 $\text{st_coffee}' \wedge \text{st_beer}' \wedge$
 $\text{only}(\text{st_coffee}, \text{st_beer}, \text{tr}).$

Customer_arrives $\stackrel{\text{def}}{=}$ $\text{tr} = \text{Customer_arrives} \wedge \text{customer} = \text{none} \wedge$
 $\text{customer}' \neq \text{none} \wedge$
 $\text{only}(\text{customer}, \text{tr})$

Coin_insert $\stackrel{\text{def}}{=}$ $\text{tr} = \text{Coin_insert} \wedge \text{customer} \neq \text{none} \wedge \text{coins} \neq 3 \wedge$
 $(\text{coins} = 0 \rightarrow \text{coins}' = 1) \wedge (\text{coins} = 1 \rightarrow \text{coins}' = 2) \wedge$
 $(\text{coins} = 2 \rightarrow \text{coins}' = 3) \wedge$
 $\text{only}(\text{coins}, \text{tr}).$