

Meaning of Some Formulas

$$\Box \Diamond F$$

$$\Diamond \Box F$$

$$\Box (F \rightarrow \bigcirc F)$$

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

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

$$\Diamond F \wedge \Box (F \rightarrow \bigcirc F)$$

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

Representing Temporal Properties of Transitions

1. One cannot claim two beers without inserting a coin in between getting them.
2. One can get two beers (from the drink dispenser) without inserting a coin in between getting them.
3. If we never have two recharge transitions in a row, then the next transition after a recharge must be a customer arrival.
4. The value of `customer` can only be changed as a result of either *Customer_arrives* or *Customer_leaves*.
5. For every two transitions *Customer_arrives* there should be a transition *Customer_leaves* between them.
6. If somebody inserts a coin twice and then gets a beer, then the amount of coins in the slot will not change.
7. If the system is recharged from time to time, then after each *Dispense_beer* the customer will leave.

Putting it All Together

When we design a system, we would like to be sure that it will satisfy all requirements, such as safety. **How?**

What we can do:

- ▶ formally represent transition systems (the symbolic representation);
- ▶ express the desired properties of the systems in temporal logic.

What is missing?

Model Checking

Given

1. A symbolic representation of a transition system;
2. A temporal formula F ,

check if every (some) computation of the system satisfies this formula, preferably in a **fully automatic way**.

Equivalences: Unwinding Properties

$$\diamond F \equiv F \vee \bigcirc \diamond F$$

$$\square F \equiv F \wedge \bigcirc \square F$$

$$F \mathbf{U} G \equiv G \vee (F \wedge \bigcirc (F \mathbf{U} G))$$

$$F \mathbf{R} G \equiv G \wedge (F \vee \bigcirc (F \mathbf{R} G))$$

Equivalences: Negation of Temporal Operators

$$\neg \bigcirc F \equiv \bigcirc \neg F$$

$$\neg \diamond F \equiv \square \neg F$$

$$\neg \square F \equiv \diamond \neg F$$

$$\neg(F \mathbf{U} G) \equiv \neg F \mathbf{R} \neg G$$

$$\neg(F \mathbf{R} G) \equiv \neg F \mathbf{U} \neg G$$

Expressing Temporal Operators Using \mathbf{U}

$$\diamond F \equiv \top \mathbf{U} F$$

$$\square F \equiv \neg(\top \mathbf{U} \neg F)$$

$$F \mathbf{R} G \equiv \neg(\neg F \mathbf{U} \neg G).$$

Other Equivalences

$$\diamond(F \vee G) \equiv \diamond F \vee \diamond G$$

$$\square(F \wedge G) \equiv \square F \wedge \square G$$

But

$$\square(F \vee G) \not\equiv \square F \vee \square G$$

$$\diamond(F \wedge G) \not\equiv \diamond F \wedge \diamond G$$