Outline

State-Changing Systems

Our main interest from now on is modelling state-changing systems.

Informally	
At each time moment, the system is in a particular state.	
The system state is changing in time. There are actions (controlled or not) that change the state.	

State-Changing Systems

Our main interest from now on is modelling state-changing systems.

Informally	Formally
At each time moment, the system is in a particular state.	This state can be characterized by values of some variables, called the state variables.
The system state is changing in time. There are actions (controlled or not) that change the state.	Actions change values of state variables.

Reasoning About State-Changing Systems

1. Build a formal model of this state-changing system which describes the behaviour of the system, or some abstraction thereof.

Reasoning About State-Changing Systems

1. Build a formal model of this state-changing system which describes the behaviour of the system, or some abstraction thereof.
2. Using a logic to specify and verify properties of the system.

Vending Machine Example

Consider an example state-changing system: a vending machine which dispenses drinks in a university department.

- The machine has several components, including at least the following: a storage space for storing and preparing drinks, a box for dispensing drinks and a coin slot for putting coins in.
- When the machine is operating, it goes through several states depending on the behavior of the current customer.

Vending Machine Example

Consider an example state-changing system: a vending machine which dispenses drinks in a university department.

- The machine has several components, including at least the following: a storage space for storing and preparing drinks, a box for dispensing drinks and a coin slot for putting coins in.
- When the machine is operating, it goes through several states depending on the behavior of the current customer.
- Each action undertaken by the customer or by the machine itself may change the state of the machine. For example, when the customer inserts a coin in the coin slot, the amount of money stored in the slot changes.

Vending Machine Example

Consider an example state-changing system: a vending machine which dispenses drinks in a university department.

- The machine has several components, including at least the following: a storage space for storing and preparing drinks, a box for dispensing drinks and a coin slot for putting coins in.
- When the machine is operating, it goes through several states depending on the behavior of the current customer.
- Each action undertaken by the customer or by the machine itself may change the state of the machine. For example, when the customer inserts a coin in the coin slot, the amount of money stored in the slot changes.
- Actions which may change the state of the system are called transitions.

Modeling State-Changing Systems

To build a formal model of a particular state-changing system, we should define

1. What are the state variables.
2. What are the possible values of the state variables.
3. What are the transitions and how they change the values of the state variables.

Modeling State-Changing Systems

To build a formal model of a particular state-changing system, we should define

1. What are the state variables.
2. What are the possible values of the state variables.
3. What are the transitions and how they change the values of the state variables.

A state can be identified with the set of pairs *(variable,value)*, or with a function from variables to values.

Transition Systems

A transition system is a tuple $\mathbb{S} = (S, \mathit{In}, T, \mathcal{X}, \mathit{dom}, L)$, where

1. S is a finite non-empty set, called the set of states of $\mathbb{S}$.
2. $\mathit{In} \subseteq S$ is a non-empty set of states, called the set of initial states of M.

Transition Systems

A transition system is a tuple $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, where

1. S is a finite non-empty set, called the set of states of $\mathbb{S}$.
2. $In \subseteq S$ is a non-empty set of states, called the set of initial states of M.
3. $T \subseteq S \times S$ is a set of pairs of states, called the transition relation of $\mathbb{S}$.

Transition Systems

A transition system is a tuple $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, where

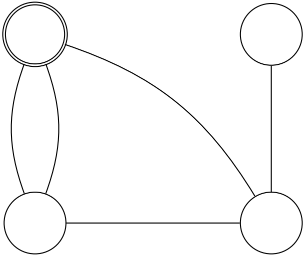
1. S is a finite non-empty set, called the set of states of $\mathbb{S}$.
2. $In \subseteq S$ is a non-empty set of states, called the set of initial states of M.
3. $T \subseteq S \times S$ is a set of pairs of states, called the transition relation of $\mathbb{S}$.

$\mathcal{X}$, dom and L will be explained later.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.

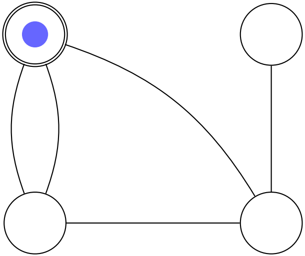


We denote the initial state(s) using double lines.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.

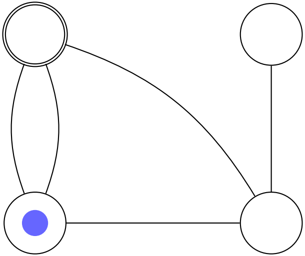


We denote the initial state(s) using double lines.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.

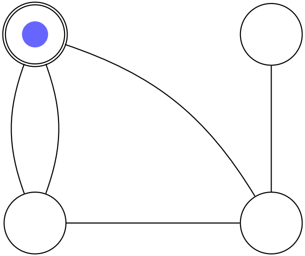


We denote the initial state(s) using double lines.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.

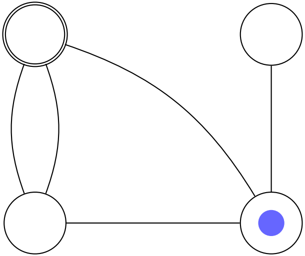


We denote the initial state(s) using double lines.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.

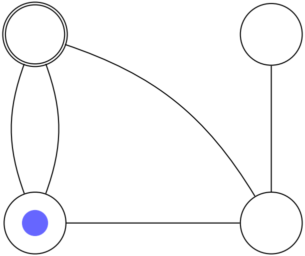


We denote the initial state(s) using double lines.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.

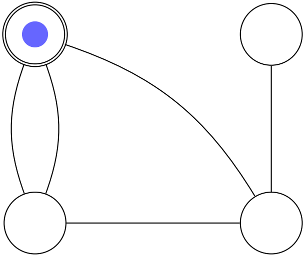


We denote the initial state(s) using double lines.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.



We denote the initial state(s) using double lines.

Transition Systems

A transition system is a tuple $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, where

1. S is a finite non-empty set, called the set of states of $\mathbb{S}$.
2. $In \subseteq S$ is a non-empty set of states, called the set of initial states of M.
3. $T \subseteq S \times S$ is a set of pairs of states, called the transition relation of $\mathbb{S}$.

Transition Systems

A transition system is a tuple $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, where

1. S is a finite non-empty set, called the set of states of $\mathbb{S}$.
2. $In \subseteq S$ is a non-empty set of states, called the set of initial states of M.
3. $T \subseteq S \times S$ is a set of pairs of states, called the transition relation of $\mathbb{S}$.
4. $\mathcal{X}$ is a finite set, its members are called state variables.

Transition Systems

A transition system is a tuple $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, where

1. S is a finite non-empty set, called the set of states of $\mathbb{S}$.
2. $In \subseteq S$ is a non-empty set of states, called the set of initial states of M.
3. $T \subseteq S \times S$ is a set of pairs of states, called the transition relation of $\mathbb{S}$.
4. $\mathcal{X}$ is a finite set, its members are called state variables.
5. dom is a mapping from $\mathcal{X}$ such that for every state variable $v \in \mathcal{X}$, $dom(v)$ is a non-empty set, called the domain for v.

Transition Systems

A transition system is a tuple $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, where

1. S is a finite non-empty set, called the set of states of $\mathbb{S}$.
2. $In \subseteq S$ is a non-empty set of states, called the set of initial states of M.
3. $T \subseteq S \times S$ is a set of pairs of states, called the transition relation of $\mathbb{S}$.
4. $\mathcal{X}$ is a finite set, its members are called state variables.
5. dom is a mapping from $\mathcal{X}$ such that for every state variable $v \in \mathcal{X}$, $dom(v)$ is a non-empty set, called the domain for v.
6. L is a function mapping states in S into interpretations, called the labeling function of $\mathbb{S}$. It will be explained later.

Transition Systems

A transition system is a tuple $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, where

1. S is a finite non-empty set, called the set of states of $\mathbb{S}$.
2. $In \subseteq S$ is a non-empty set of states, called the set of initial states of M.
3. $T \subseteq S \times S$ is a set of pairs of states, called the transition relation of $\mathbb{S}$.
4. $\mathcal{X}$ is a finite set, its members are called state variables.
5. dom is a mapping from $\mathcal{X}$ such that for every state variable $v \in \mathcal{X}$, $dom(v)$ is a non-empty set, called the domain for v.
6. L is a function mapping states in S into interpretations, called the labeling function of $\mathbb{S}$. It will be explained later.

The transition system is said to be finite-state if for every state variable v, the domain $dom(v)$ for this variable is finite.

We will only study finite-state transition systems.

Labeling Function

Note this part of the definition:

4. $\mathcal{X}$ is a finite set, its members are called state variables.
5. *dom* is a mapping from $\mathcal{X}$ such that for every state variable $v \in \mathcal{X}$, $dom(v)$ is a non-empty set, called the domain for v.

Labeling Function

Note this part of the definition:

4. $\mathcal{X}$ is a finite set, its members are called state variables.
5. *dom* is a mapping from $\mathcal{X}$ such that for every state variable $v \in \mathcal{X}$, $dom(v)$ is a non-empty set, called the domain for v.

That is, for a transition system $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, the set of variables $\mathcal{X}$ and the mapping *dom* defines an instance of propositional logic of finite domains.

Labeling Function

Note this part of the definition:

4. $\mathcal{X}$ is a finite set, its members are called state variables.
5. *dom* is a mapping from $\mathcal{X}$ such that for every state variable $v \in \mathcal{X}$, $dom(v)$ is a non-empty set, called the domain for v.

That is, for a transition system $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, the set of variables $\mathcal{X}$ and the mapping *dom* defines an instance of propositional logic of finite domains.

Denote the set of all interpretations for this instance of PLFD by $\mathbb{I}$. Then the labelling function L is a mapping $L : S \to \mathbb{I}$, that is, it maps every state to an interpretation.

Labeling Function

Note this part of the definition:

4. $\mathcal{X}$ is a finite set, its members are called state variables.
5. *dom* is a mapping from $\mathcal{X}$ such that for every state variable $v \in \mathcal{X}$, $dom(v)$ is a non-empty set, called the domain for v.

That is, for a transition system $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$, the set of variables $\mathcal{X}$ and the mapping *dom* defines an instance of propositional logic of finite domains.

Denote the set of all interpretations for this instance of PLFD by $\mathbb{I}$.
Then the labelling function L is a mapping $L : S \to \mathbb{I}$, that is, it maps every state to an interpretation.
This means that

1. for every variable $v \in \mathcal{X}$ and every state $s \in S$, we have $L(s)(v) \in dom(v)$;
2. for every formula A of this instance of PLFD and every state $s \in S$, either $L(s) \models A$ or $L(s) \not\models A$.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

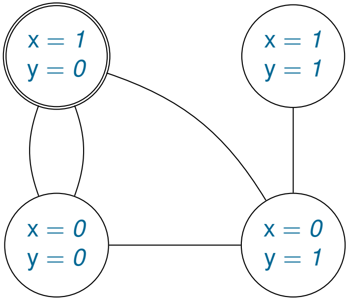
- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.

State Transition Graph

State Transition Graph of a transition system $\mathbb{S}$:

- The nodes are the states of $\mathbb{S}$.
- The arcs are elements of the transition relation: there is an arc from a state s to a state s' if and only if $(s, s') \in T$.

Assume two boolean-valued variables x, y.



We denote the initial state(s) using double lines.

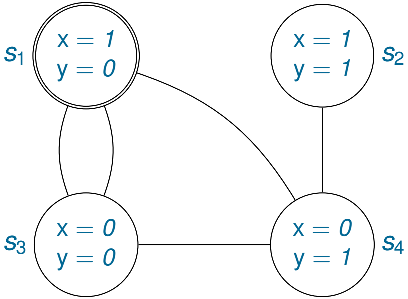
States as Interpretations

Essentially, in each state each variable has a value.

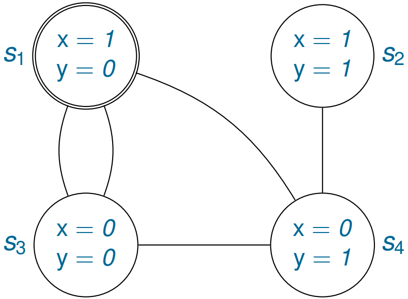
- If $L(s)(x) = v$ then we say that x has the value v in s and write $s(x) = v$.
- If $L(s) \models A$ then we say that s satisfies A or A is true in s and write $s \models A$.

In both cases, we identify s with $L(s)$.

States as Interpretations

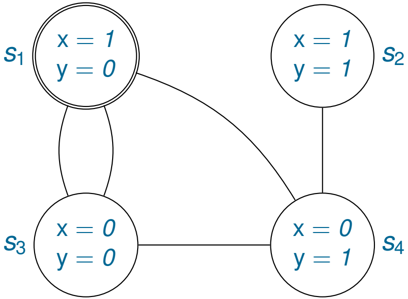


States as Interpretations



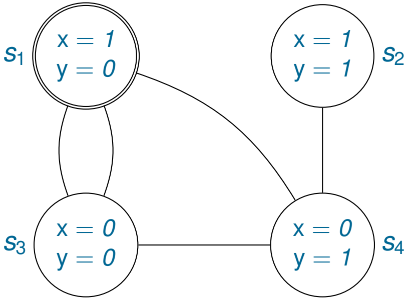
- $s_1 \models x$

States as Interpretations



- $s_1 \models \mathrm{x}$
- $s_2 \models \mathrm{x} \wedge \mathrm{y}$

States as Interpretations



- $s_1 \models x$
- $s_2 \models x \wedge y$
- $s_3 \models x \leftrightarrow y$

Transitions

When we model systems, we will usually represent the transition relation as a union of so-called transitions.

- A transition t is any set of pairs of states.
- A transition t is applicable to a state s if there exists a state s' such that $(s, s') \in t$.
- A transition t is deterministic if for every state s there exists at most one state s' such that $(s, s') \in t$.

Vending Machine

1. The vending machine contains a drink storage, a coin slot, and a drink dispenser. The drink storage stores drinks of two kinds: beer and coffee. We are only interested in whether a particular kind of drink is currently being stored or not, but not interested in the amount of it.
2. The coin slot can accommodate up to three coins.
3. The drink dispenser can store at most one drink. If it contains a drink, this drink should be removed before the next one can be dispensed.
4. A can of beer costs two coins. A cup of coffee costs one coin.
5. There are two kinds of customers: students and professors. Students drink only beer, professors drink only coffee.
6. From time to time the drink storage can be recharged.

Vending Machine

1. The vending machine contains a drink storage, a coin slot, and a drink dispenser. The drink storage stores drinks of two kinds: beer and coffee. We are only interested in whether a particular kind of drink is currently being stored or not, but not interested in the amount of it.
2. The coin slot can accommodate up to three coins.
3. The drink dispenser can store at most one drink. If it contains a drink, this drink should be removed before the next one can be dispensed.
4. A can of beer costs two coins. A cup of coffee costs one coin.
5. There are two kinds of customers: students and professors. Students drink only beer, professors drink only coffee.
6. From time to time the drink storage can be recharged.

Formalization: Variables and Domains

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

Transitions for the Vending Machine

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.

Symbolic Representation of Sets of States

Let $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$ be a finite-state transition system. Then every formula F defines a set states:

$$\{s \mid s \models F\}.$$

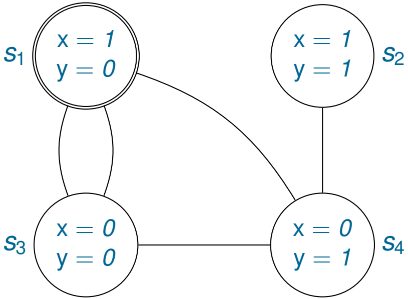
Symbolic Representation of Sets of States

Let $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$ be a finite-state transition system. Then every formula F defines a set states:

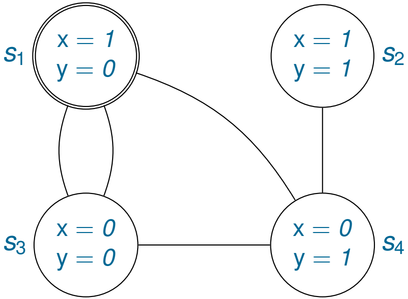
$$\{s \mid s \models F\}.$$

We say that F (symbolically) represent this set of states.

Symbolic Representation of Sets of States

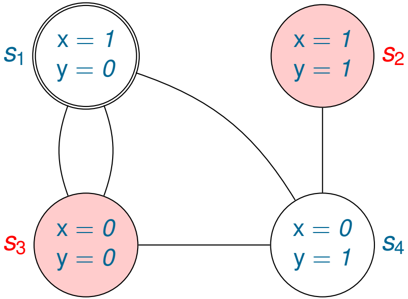


Symbolic Representation of Sets of States



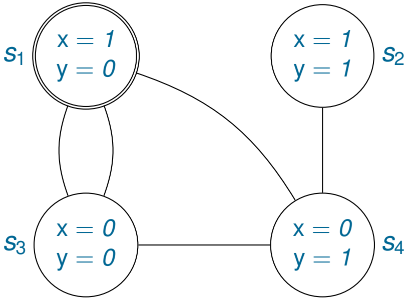
- $x \leftrightarrow y$

Symbolic Representation of Sets of States



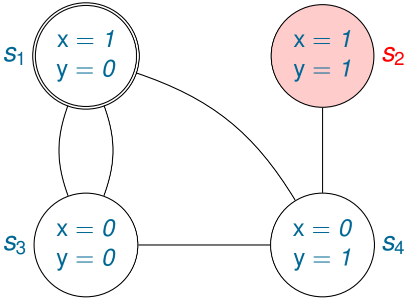
- $x \leftrightarrow y$ represents $\{s_2, s_3\}$

Symbolic Representation of Sets of States



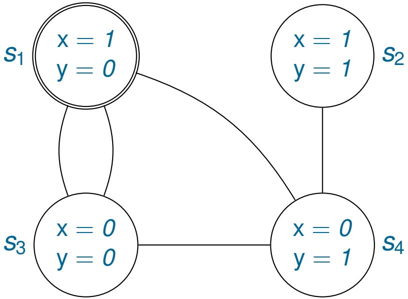
- $x \leftrightarrow y$ represents $\{s_2, s_3\}$
- $x \wedge y$

Symbolic Representation of Sets of States



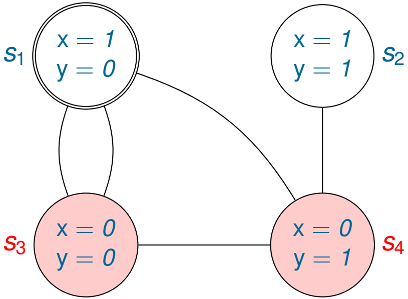
- $x \leftrightarrow y$ represents $\{s_2, s_3\}$
- $x \wedge y$ represents $\{s_2\}$

Symbolic Representation of Sets of States



- $x \leftrightarrow y$ represents $\{s_2, s_3\}$
- $x \wedge y$ represents $\{s_2\}$
- $\neg x$

Symbolic Representation of Sets of States



- $x \leftrightarrow y$ represents $\{s_2, s_3\}$
- $x \land y$ represents $\{s_2\}$
- $\neg x$ represents $\{s_3, s_4\}$

Example

Let us represent the set of states in which the machine is ready to dispense a drink. In every such state, a drink should be available, the drink dispenser empty, and the coin slot contain enough coins.

Example

Let us represent the set of states in which the machine is ready to dispense a drink. In every such state, a drink should be available, the drink dispenser empty, and the coin slot contain enough coins.

This can be expressed by:

$$
\begin{aligned}
&(\mathrm{st\_coffee} \vee \mathrm{st\_beer}) \wedge \\
&\mathrm{disp} = \mathit{none} \wedge \\
&((\mathrm{coins} = \mathit{1} \wedge \mathrm{st\_coffee}) \vee \mathrm{coins} = \mathit{2} \vee \mathrm{coins} = \mathit{3}).
\end{aligned}
$$

Symbolic Representation of Transitions

A transition is a relation on pairs of states. It brins the system to the current state and the next state. Formulas of PLFD can only express properties of a single state. How can we represent transitions using formulas?

Symbolic Representation of Transitions

A transition is a relation on pairs of states. It brins the system to the current state and the next state. Formulas of PLFD can only express properties of a single state. How can we represent transitions using formulas?

- In addition to the set of propositional variables $\mathcal{X} = \{x_1, \ldots, x_n\}$, introduce a set of next state variables $\mathcal{X}' = \{x'_1, \ldots, x'_n\}$.

Symbolic Representation of Transitions

A transition is a relation on pairs of states. It brins the system to the current state and the next state. Formulas of PLFD can only express properties of a single state. How can we represent transitions using formulas?

- In addition to the set of propositional variables $\mathcal{X} = \{x_1, \ldots, x_n\}$, introduce a set of next state variables $\mathcal{X}' = \{x'_1, \ldots, x'_n\}$.
- Pairs of states as interpretations. For every variable $x \in \mathcal{X}$ define

$$\begin{array}{rcl} (s, s')(x) & \stackrel{\text{def}}{=} & s(x); \\ (s, s')(x') & \stackrel{\text{def}}{=} & s'(x). \end{array}$$

Symbolic Representation of Transitions

A transition is a relation on pairs of states. It brins the system to the current state and the next state. Formulas of PLFD can only express properties of a single state. How can we represent transitions using formulas?

- In addition to the set of propositional variables $\mathcal{X} = \{x_1, \ldots, x_n\}$, introduce a set of next state variables $\mathcal{X}' = \{x_1', \ldots, x_n'\}$.
- Pairs of states as interpretations. For every variable $x \in \mathcal{X}$ define

$$\begin{aligned}(s, s')(x) &\stackrel{\text{def}}{=} s(x);\\ (s, s')(x') &\stackrel{\text{def}}{=} s'(x).\end{aligned}$$

- Symbolic representation. Formula F of variables $\mathcal{X} \cup \mathcal{X}'$ represents a transition t if $t = \{(s, s') \mid (s, s') \models F\}$.

Example

The transition *Recharge*:

$$\mathrm{customer} = \mathit{none} \land \mathrm{st\_coffee}' \land \mathrm{st\_beer}'.$$

Example

The transition *Recharge*:

$$\text{customer} = none \land \text{st\_coffee}' \land \text{st\_beer}'.$$

But this formula includes describes a very strange transition after which, for example

- coins may appear in and disappear from the slot;
- dfrinks may appear in and disappear from the dispenser.
- ...

Frame Problem

One has to express explicitly, maybe for a large number of state variables, that the values of these variables do not change after a transition. For example,

$$\begin{array}{l}(\mathrm{coins} = 0 \leftrightarrow \mathrm{coins}' = 0) \wedge \\ (\mathrm{coins} = 1 \leftrightarrow \mathrm{coins}' = 1) \wedge \\ (\mathrm{coins} = 2 \leftrightarrow \mathrm{coins}' = 2) \wedge \\ (\mathrm{coins} = 3 \leftrightarrow \mathrm{coins}' = 3).\end{array}$$

This frame problem arises in artificial intelligence, knowledge representation, and reasoning about actions.

End of Lecture 17

Slides for lecture 17 end here . . .

Notation for the Frame Formula

Abbreviations (we assume $dom(x) = dom(y)$):

$$
\begin{aligned}
x \neq v &\stackrel{\text{def}}{=} \neg(x = v) \\
x = y &\stackrel{\text{def}}{=} \bigwedge_{v \in dom(x)} (x = v \leftrightarrow y = v).
\end{aligned}
$$

Notation for the Frame Formula

Abbreviations (we assume $dom(x) = dom(y)$):

$$
\begin{array}{lcl}
x \neq v & \stackrel{\text{def}}{=} & \neg(x = v) \\
x = y & \stackrel{\text{def}}{=} & \bigwedge_{v \in dom(x)} (x = v \leftrightarrow y = v).
\end{array}
$$

Let $\mathbb{S}$ be a transition system and $\{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ be a set of state variables of $\mathcal{L}(\mathbb{S})$. Define

$$
only(x_1, \ldots, x_n) \stackrel{\text{def}}{=} \bigwedge_{y \in \mathcal{X} \setminus \{x_1, \ldots, x_n\}} y = y'.
$$

This formula expresses that $x_1, \ldots, x_n$ are the only variables whose values can be changed by the transition.

Preconditions and Postconditions

When we represent a transition symbolically using a formula F of variables $\mathcal{X} \cup \mathcal{X}'$, the formula F is usually represented as the conjunction $F_1 \wedge F_2$ of two formulas:

1. F_1 expresses some conditions on the variables $\mathcal{X}$ which are necessary to execute the transition (precondition);

Preconditions and Postconditions

When we represent a transition symbolically using a formula F of variables $\mathcal{X} \cup \mathcal{X}'$, the formula F is usually represented as the conjunction $F_1 \wedge F_2$ of two formulas:

1. F_1 expresses some conditions on the variables $\mathcal{X}$ which are necessary to execute the transition (precondition);
2. F_2 expresses some conditions relating variables in $\mathcal{X}$ to those in $\mathcal{X}'$, i.e., conditions which show how the values of the variables after the transition relate to their values before the transition (postcondition).

Transitions for the Vending Machine

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.

Transitions: Symbolic Representation

st_coffee	$\{0, 1\}$	
st_beer	$\{0, 1\}$	*Recharge*
disp	$\{none, beer, coffee\}$	*Customer_arrives*
coins	$\{0, 1, 2, 3\}$	*Customer_leaves*
customer	$\{none, student, prof\}$	*Coin_insert*

$$\mathit{Recharge} \stackrel{\text{def}}{=} \begin{array}{l} \text{customer} = \mathit{none} \wedge \\ \text{st\_coffee}' \wedge \text{st\_beer}' \wedge \\ \mathit{only}(\text{st\_coffee}, \text{st\_beer}). \end{array}$$

Transitions: Symbolic Representation

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{none, beer, coffee\}$
coins	$\{0, 1, 2, 3\}$
customer	$\{none, student, prof\}$

Recharge
Customer_arrives
Customer_leaves
Coin_insert

$$\begin{aligned} \textit{Recharge} \quad \stackrel{\text{def}}{=} \quad & \text{customer} = \textit{none} \wedge \\ & \text{st\_coffee}' \wedge \text{st\_beer}' \wedge \\ & \textit{only}(\text{st\_coffee}, \text{st\_beer}). \end{aligned}$$

Transitions: Symbolic Representation

st_coffee	$\{0, 1\}$	
st_beer	$\{0, 1\}$	*Recharge*
disp	$\{\mathit{none}, \mathit{beer}, \mathit{coffee}\}$	*Customer_arrives*
coins	$\{\mathit{0}, \mathit{1}, \mathit{2}, \mathit{3}\}$	*Customer_leaves*
customer	$\{\mathit{none}, \mathit{student}, \mathit{prof}\}$	*Coin_insert*

$$\begin{aligned}
\mathit{Recharge} &\stackrel{\text{def}}{=} \text{customer} = \mathit{none} \wedge \\
&\quad\ \text{st_coffee}' \wedge \text{st_beer}' \wedge \\
&\quad\ \mathit{only}(\text{st_coffee}, \text{st_beer}). \\
\mathit{Customer_arrives} &\stackrel{\text{def}}{=} \text{customer} = \mathit{none} \wedge \text{customer}' \neq \mathit{none} \wedge \\
&\quad\ \mathit{only}(\text{customer})
\end{aligned}$$

Transitions: Symbolic Representation

st_coffee	$\{0, 1\}$	
st_beer	$\{0, 1\}$	*Recharge*
disp	$\{\mathit{none}, \mathit{beer}, \mathit{coffee}\}$	*Customer_arrives*
coins	$\{\mathit{0}, \mathit{1}, \mathit{2}, \mathit{3}\}$	*Customer_leaves*
customer	$\{\mathit{none}, \mathit{student}, \mathit{prof}\}$	*Coin_insert*

$$\begin{aligned}
\mathit{Recharge} &\stackrel{\text{def}}{=} \text{customer} = \mathit{none} \wedge \\
&\quad \text{st_coffee}' \wedge \text{st_beer}' \wedge \\
&\quad \mathit{only}(\text{st_coffee}, \text{st_beer}). \\
\mathit{Customer_arrives} &\stackrel{\text{def}}{=} \text{customer} = \mathit{none} \wedge \text{customer}' \neq \mathit{none} \wedge \\
&\quad \mathit{only}(\text{customer})
\end{aligned}$$

Transitions: Symbolic Representation

st_coffee	$\{0, 1\}$	
st_beer	$\{0, 1\}$	*Recharge*
disp	$\{none, beer, coffee\}$	*Customer_arrives*
coins	$\{0, 1, 2, 3\}$	*Customer_leaves*
customer	$\{none, student, prof\}$	*Coin_insert*

$$\begin{aligned}
\textit{Recharge} &\stackrel{\text{def}}{=} \text{customer} = \textit{none} \wedge \\
&\quad \text{st_coffee}' \wedge \text{st_beer}' \wedge \\
&\quad \textit{only}(\text{st_coffee}, \text{st_beer}). \\
\textit{Customer_arrives} &\stackrel{\text{def}}{=} \text{customer} = \textit{none} \wedge \text{customer}' \neq \textit{none} \wedge \\
&\quad \textit{only}(\text{customer}) \\
\textit{Customer_leaves} &\stackrel{\text{def}}{=} \text{customer} \neq \textit{none} \wedge \text{customer}' = \textit{none} \wedge \\
&\quad \textit{only}(\text{customer}).
\end{aligned}$$

Transitions: Symbolic Representation

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{\mathit{none}, \mathit{beer}, \mathit{coffee}\}$
coins	$\{\mathit{0}, \mathit{1}, \mathit{2}, \mathit{3}\}$
customer	$\{\mathit{none}, \mathit{student}, \mathit{prof}\}$

Recharge
Customer_arrives
Customer_leaves
Coin_insert

$$
\begin{aligned}
\mathit{Recharge} &\stackrel{\text{def}}{=} \text{customer} = \mathit{none} \wedge \\
&\quad \text{st\_coffee}' \wedge \text{st\_beer}' \wedge \\
&\quad \mathit{only}(\text{st\_coffee}, \text{st\_beer}). \\
\mathit{Customer\_arrives} &\stackrel{\text{def}}{=} \text{customer} = \mathit{none} \wedge \text{customer}' \neq \mathit{none} \wedge \\
&\quad \mathit{only}(\text{customer}) \\
\mathit{Customer\_leaves} &\stackrel{\text{def}}{=} \text{customer} \neq \mathit{none} \wedge \text{customer}' = \mathit{none} \wedge \\
&\quad \mathit{only}(\text{customer}).
\end{aligned}
$$

Transitions: Symbolic Representation

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{none, beer, coffee\}$
coins	$\{0, 1, 2, 3\}$
customer	$\{none, student, prof\}$

Recharge
Customer_arrives
Customer_leaves
Coin_insert

$$\begin{aligned}
\mathit{Recharge} &\stackrel{\text{def}}{=} \text{customer} = \mathit{none} \wedge \\
&\quad \text{st_coffee}' \wedge \text{st_beer}' \wedge \\
&\quad \mathit{only}(\text{st_coffee}, \text{st_beer}). \\
\mathit{Customer_arrives} &\stackrel{\text{def}}{=} \text{customer} = \mathit{none} \wedge \text{customer}' \neq \mathit{none} \wedge \\
&\quad \mathit{only}(\text{customer}) \\
\mathit{Customer_leaves} &\stackrel{\text{def}}{=} \text{customer} \neq \mathit{none} \wedge \text{customer}' = \mathit{none} \wedge \\
&\quad \mathit{only}(\text{customer}). \\
\mathit{Coin_insert} &\stackrel{\text{def}}{=} \text{customer} \neq \mathit{none} \wedge \text{coins} \neq \mathit{3} \wedge \\
&\quad (\text{coins} = \mathit{0} \rightarrow \text{coins}' = \mathit{1}) \wedge \\
&\quad (\text{coins} = \mathit{1} \rightarrow \text{coins}' = \mathit{2}) \wedge \\
&\quad (\text{coins} = \mathit{2} \rightarrow \text{coins}' = \mathit{3}) \wedge \\
&\quad \mathit{only}(\text{coins}).
\end{aligned}$$

Transitions

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{none, beer, coffee\}$
coins	$\{0, 1, 2, 3\}$
customer	$\{none, student, prof\}$

Dispense_beer
Dispense_coffee
Take_drink

Transitions

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{\mathit{none}, \mathit{beer}, \mathit{coffee}\}$
coins	$\{0, 1, 2, 3\}$
customer	$\{\mathit{none}, \mathit{student}, \mathit{prof}\}$

Dispense_beer
Dispense_coffee
Take_drink

$$
\begin{aligned}
\mathit{Dispense\_beer} \stackrel{\text{def}}{=}\ & \text{customer} = \mathit{student} \wedge \text{st\_beer} \wedge \\
& \text{disp} = \mathit{none} \wedge (\text{coins} = 2 \vee \text{coins} = 3) \wedge \\
& \text{disp}' = \mathit{beer} \wedge \\
& (\text{coins} = 2 \rightarrow \text{coins}' = 0) \wedge \\
& (\text{coins} = 3 \rightarrow \text{coins}' = 1) \wedge \\
& \mathit{only}(\text{st\_beer}, \text{disp}, \text{coins}).
\end{aligned}
$$

Transitions

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{\textit{none}, \textit{beer}, \textit{coffee}\}$
coins	$\{0, 1, 2, 3\}$
customer	$\{\textit{none}, \textit{student}, \textit{prof}\}$

Dispense_beer
Dispense_coffee
Take_drink

$$
\begin{aligned}
\textit{Dispense\_beer} \stackrel{\text{def}}{=}\ & \text{customer} = \textit{student} \wedge \text{st\_beer} \wedge \\
& \text{disp} = \textit{none} \wedge (\text{coins} = 2 \vee \text{coins} = 3) \wedge \\
& \text{disp}' = \textit{beer} \wedge \\
& (\text{coins} = 2 \rightarrow \text{coins}' = 0) \wedge \\
& (\text{coins} = 3 \rightarrow \text{coins}' = 1) \wedge \\
& \textit{only}(\text{st\_beer}, \text{disp}, \text{coins}).
\end{aligned}
$$

Transitions

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{none, beer, coffee\}$
coins	$\{0, 1, 2, 3\}$
customer	$\{none, student, prof\}$

Dispense_beer
Dispense_coffee
Take_drink

$$
\begin{aligned}
\mathit{Dispense\_beer} \stackrel{\text{def}}{=}\ & \text{customer} = \mathit{student} \wedge \text{st\_beer} \wedge \\
& \text{disp} = \mathit{none} \wedge (\text{coins} = 2 \vee \text{coins} = 3) \wedge \\
& \text{disp}' = \mathit{beer} \wedge \\
& (\text{coins} = 2 \rightarrow \text{coins}' = 0) \wedge \\
& (\text{coins} = 3 \rightarrow \text{coins}' = 1) \wedge \\
& \mathit{only}(\text{st\_beer}, \text{disp}, \text{coins}).
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Dispense\_coffee} \stackrel{\text{def}}{=}\ & \text{customer} = \mathit{prof} \wedge \text{st\_coffee} \wedge \\
& \text{disp} = \mathit{none} \wedge \text{coins} \neq 0 \wedge \\
& \text{disp}' = \mathit{coffee} \wedge \\
& (\text{coins} = 1 \rightarrow \text{coins}' = 0) \wedge \\
& (\text{coins} = 2 \rightarrow \text{coins}' = 1) \wedge \\
& (\text{coins} = 3 \rightarrow \text{coins}' = 2) \wedge \\
& \mathit{only}(\text{st\_coffee}, \text{disp}, \text{coins}).
\end{aligned}
$$

Transitions

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{none, beer, coffee\}$
coins	$\{0, 1, 2, 3\}$
customer	$\{none, student, prof\}$

Dispense_beer
Dispense_coffee
Take_drink

$$\begin{aligned}
\textit{Dispense_beer} \stackrel{\text{def}}{=}\ & \text{customer} = \textit{student} \wedge \text{st_beer} \wedge \\
& \text{disp} = \textit{none} \wedge (\text{coins} = \textit{2} \vee \text{coins} = \textit{3}) \wedge \\
& \text{disp}' = \textit{beer} \wedge \\
& (\text{coins} = \textit{2} \to \text{coins}' = \textit{0}) \wedge \\
& (\text{coins} = \textit{3} \to \text{coins}' = \textit{1}) \wedge \\
& \textit{only}(\text{st_beer}, \text{disp}, \text{coins}).
\end{aligned}$$

$$\begin{aligned}
\textit{Dispense_coffee} \stackrel{\text{def}}{=}\ & \text{customer} = \textit{prof} \wedge \text{st_coffee} \wedge \\
& \text{disp} = \textit{none} \wedge \text{coins} \neq \textit{0} \wedge \\
& \text{disp}' = \textit{coffee} \wedge \\
& (\text{coins} = \textit{1} \to \text{coins}' = \textit{0}) \wedge \\
& (\text{coins} = \textit{2} \to \text{coins}' = \textit{1}) \wedge \\
& (\text{coins} = \textit{3} \to \text{coins}' = \textit{2}) \wedge \\
& \textit{only}(\text{st_coffee}, \text{disp}, \text{coins}).
\end{aligned}$$

Transitions

st_coffee	$\{0, 1\}$
st_beer	$\{0, 1\}$
disp	$\{none, beer, coffee\}$
coins	$\{0, 1, 2, 3\}$
customer	$\{none, student, prof\}$

Dispense_beer
Dispense_coffee
Take_drink

$$\begin{aligned}
\mathit{Dispense_beer} \stackrel{\text{def}}{=}\ & \text{customer} = \mathit{student} \land \text{st_beer} \land \\
& \text{disp} = \mathit{none} \land (\text{coins} = 2 \lor \text{coins} = 3) \land \\
& \text{disp}' = \mathit{beer} \land \\
& (\text{coins} = 2 \rightarrow \text{coins}' = 0) \land \\
& (\text{coins} = 3 \rightarrow \text{coins}' = 1) \land \\
& \mathit{only}(\text{st_beer}, \text{disp}, \text{coins}).
\end{aligned}$$

$$\begin{aligned}
\mathit{Dispense_coffee} \stackrel{\text{def}}{=}\ & \text{customer} = \mathit{prof} \land \text{st_coffee} \land \\
& \text{disp} = \mathit{none} \land \text{coins} \neq 0 \land \\
& \text{disp}' = \mathit{coffee} \land \\
& (\text{coins} = 1 \rightarrow \text{coins}' = 0) \land \\
& (\text{coins} = 2 \rightarrow \text{coins}' = 1) \land \\
& (\text{coins} = 3 \rightarrow \text{coins}' = 2) \land \\
& \mathit{only}(\text{st_coffee}, \text{disp}, \text{coins}).
\end{aligned}$$

$$\begin{aligned}
\mathit{Take_drink} \stackrel{\text{def}}{=}\ & \text{customer} \neq \mathit{none} \land \text{disp} \neq \mathit{none} \land \\
& \text{disp}' = \mathit{none} \land \\
& \mathit{only}(\text{disp}).
\end{aligned}$$

Transitions

Model checkers often use the convention that the variables that can change are those variables x such that x' occurs in the problem. Under this convention we can remove $only(\ldots)$ from all transitions and change $Dispense\_beer$ and $Dispense\_coffee$ as follows:

$$
\begin{array}{lcl}
Dispense\_beer & \stackrel{\text{def}}{=} & \text{customer} = student \wedge \text{st\_beer} \wedge \\
& & \text{disp} = none \wedge (\text{coins} = 2 \vee \text{coins} = 3) \wedge \\
& & \text{disp}' = beer \wedge \\
& & (\text{coins} = 2 \rightarrow \text{coins}' = 0) \wedge \\
& & (\text{coins} = 3 \rightarrow \text{coins}' = 1) \wedge \\
& & \text{st\_beer}' = \text{st\_beer}'. \\
Dispense\_coffee & \stackrel{\text{def}}{=} & \text{customer} = prof \wedge \text{st\_coffee} \wedge \\
& & \text{disp} = none \wedge \text{coins} \neq 0 \wedge \\
& & \text{disp}' = coffee \wedge \\
& & (\text{coins} = 1 \rightarrow \text{coins}' = 0) \wedge \\
& & (\text{coins} = 2 \rightarrow \text{coins}' = 1) \wedge \\
& & (\text{coins} = 3 \rightarrow \text{coins}' = 2) \wedge \\
& & \text{st\_coffee}' = \text{st\_coffee}'.
\end{array}
$$

Temporal Properties of Transition Systems

1. There is no state in which professor and student are both customers.

Temporal Properties of Transition Systems

1. There is no state in which professor and student are both customers.
2. Students never drink coffee.

Temporal Properties of Transition Systems

1. There is no state in which professor and student are both customers.
2. Students never drink coffee.
3. The machine cannot dispense drinks forever without recharging.