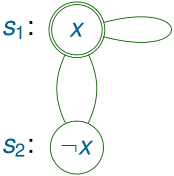
# Outline

# Computation Tree

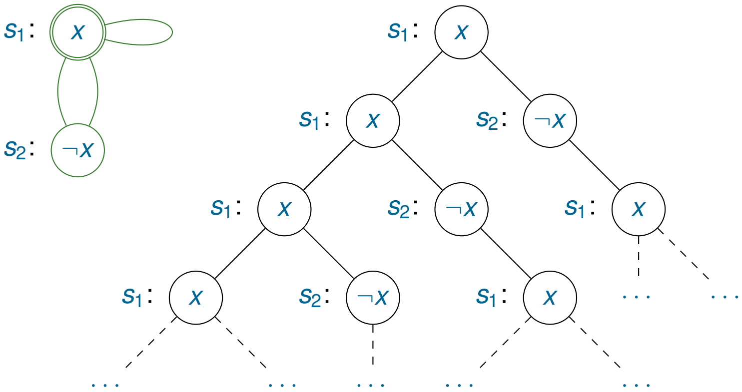
Let $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$ be a transition system and $s \in S$ be a state. The computation tree for $\mathbb{S}$ starting at $s$ is the following (possibly infinite) tree.

1. The nodes of the tree are labeled by states in $S$.
2. The root of the tree is labeled by $s$.
3. For every node $s'$ in the tree, its children are exactly such nodes $s'' \in S$ that $(s', s'') \in T$.

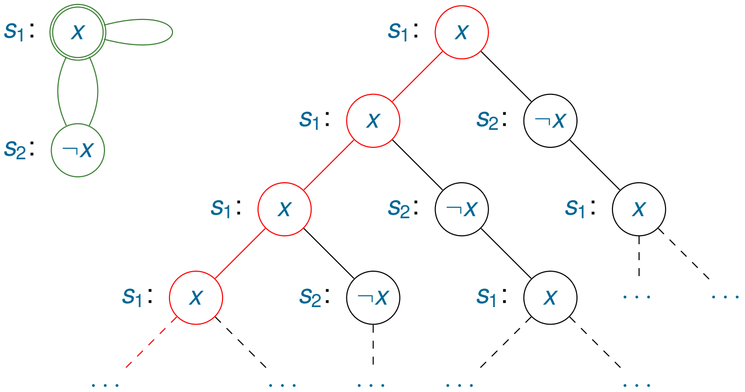
# Computation Trees and Paths



# Computation Trees and Paths

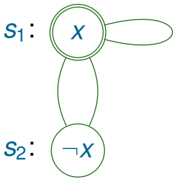


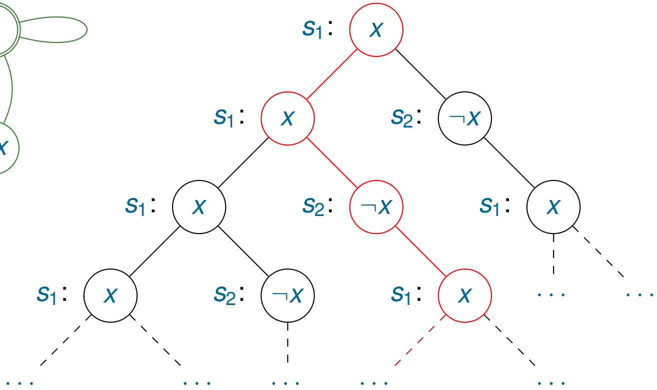
# Computation Trees and Paths




A computation path for $\mathbb{S}$: any branch $s_0, s_1, \ldots$ in the tree.

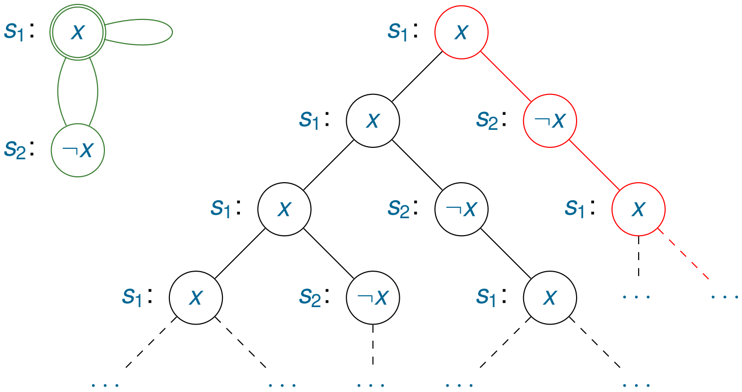
# Computation Trees and Paths




A computation path for $\mathbb{S}$: any branch $s_0, s_1, \ldots$ in the tree.

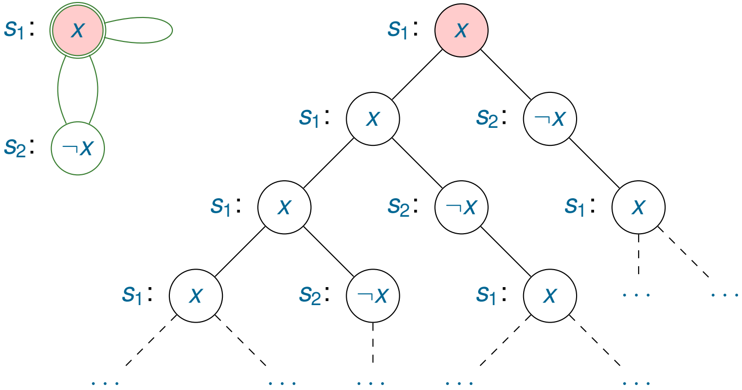
# Computation Trees and Paths




A computation path for $\mathbb{S}$: any branch $s_0, s_1, \dots$ in the tree.

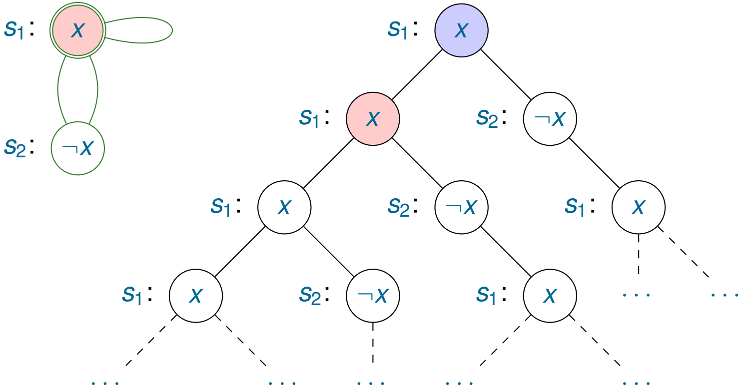
# Computation

Every path in the computation tree corresponds to a computation:



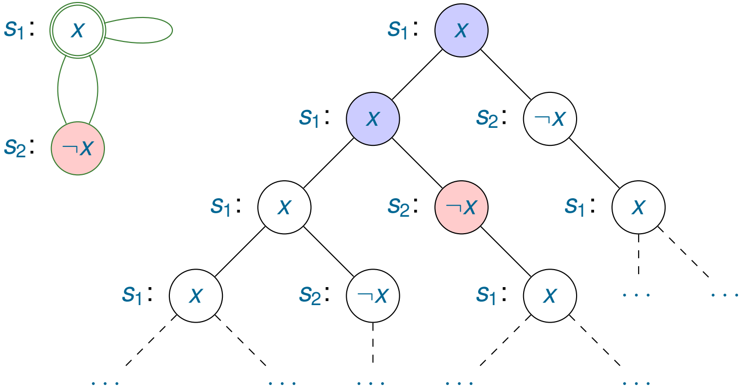
# Computation

Every path in the computation tree corresponds to a computation:



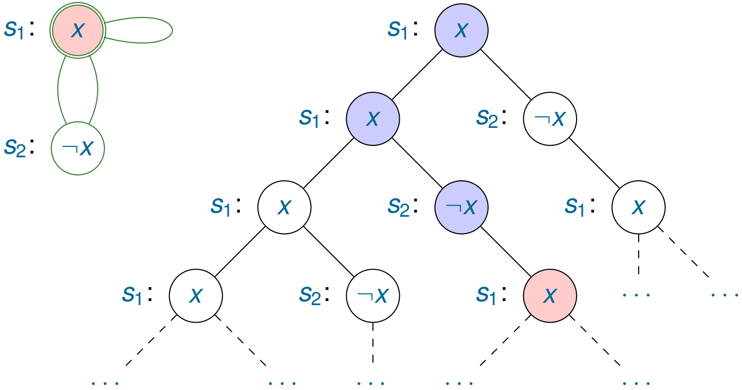
# Computation

Every path in the computation tree corresponds to a computation:



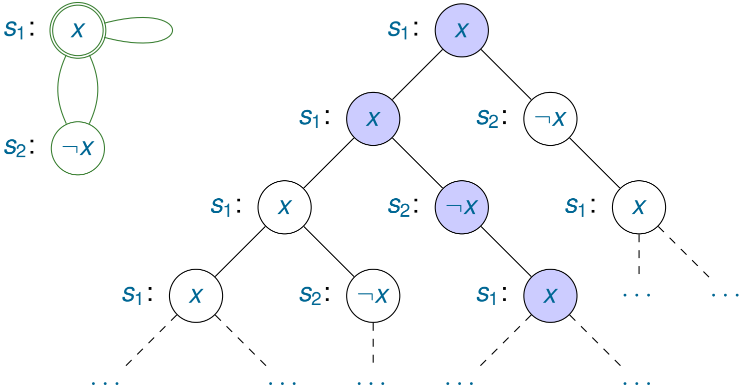
# Computation

Every path in the computation tree corresponds to a computation:



# Computation

Every path in the computation tree corresponds to a computation:



# Properties

- Computation paths for a transition system are exactly all branches in the computation trees for this transition system.

# Properties

- Computation paths for a transition system are exactly all branches in the computation trees for this transition system.
- Let $n$ be a node in a computation tree $C$ for $\mathbb{S}$ labeled by $s'$. Then the subtree of $C$ rooted at $s'$ is the computation tree for $\mathbb{S}$ starting at $s'$. In other words, every subtree of a computation tree rooted at some node is itself a computation tree.

# Properties

- Computation paths for a transition system are exactly all branches in the computation trees for this transition system.
- Let $n$ be a node in a computation tree $C$ for $\mathbb{S}$ labeled by $s'$. Then the subtree of $C$ rooted at $s'$ is the computation tree for $\mathbb{S}$ starting at $s'$. In other words, every subtree of a computation tree rooted at some node is itself a computation tree.
- For every transition system $\mathbb{S}$ and state $s$ there exists a unique computation tree for $\mathbb{S}$ starting at $s$, up to the order of children.

# LTL

Linear Temporal Logic is a logic for reasoning about properties of computation paths.

# LTL

Linear Temporal Logic is a logic for reasoning about properties of computation paths.

Formulas are built in the same way as in propositional logic, with the following additions:

1. If $F$ is a formula, then $\bigcirc F$, $\square F$, and $\lozenge F$ are formulas;
2. If $F$ and $G$ are formulas, then $F \,\mathcal{U}\, G$ and $F \,\mathcal{R}\, G$ are formulas.

# LTL

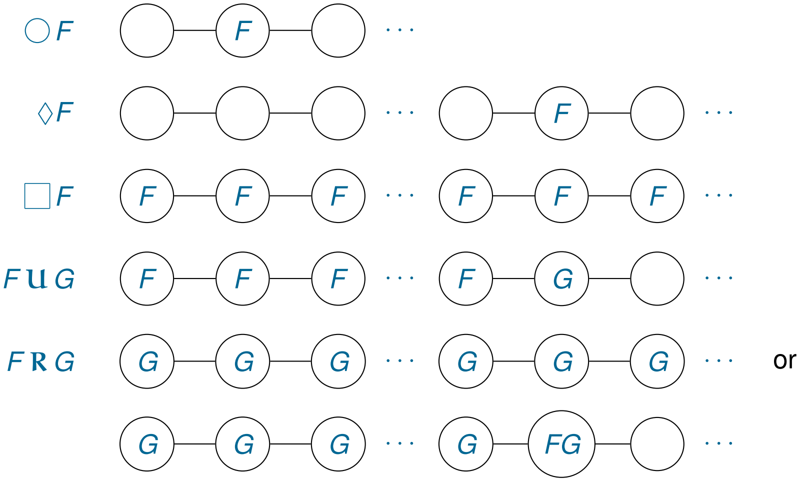
Linear Temporal Logic is a logic for reasoning about properties of computation paths.

Formulas are built in the same way as in propositional logic, with the following additions:

1. If $F$ is a formula, then $\bigcirc F$, $\square F$, and $\lozenge F$ are formulas;
2. If $F$ and $G$ are formulas, then $F \,\mathcal{U}\, G$ and $F \,\mathcal{R}\, G$ are formulas.

| | |
|---|---|
| $\bigcirc$ | next |
| $\square$ | always (in future) |
| $\lozenge$ | sometimes (in future) |
| $\mathcal{U}$ | until |
| $\mathcal{R}$ | release |

# Semantics (intuitive)



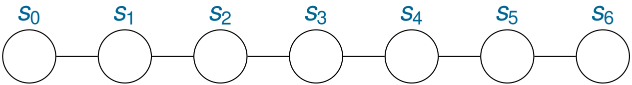
# Semantics

Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.

# Semantics

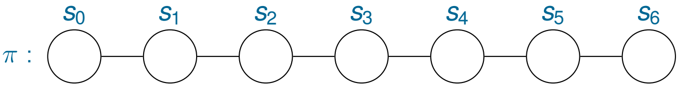
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and $F$ be an $\mathrm{LTL}$ formula.

$\pi$ :



# Semantics

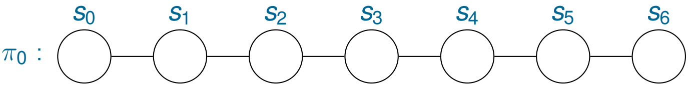
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and $F$ be an $\mathrm{LTL}$ formula.




We define the notion $F$ is true on $\pi$ (or $F$ holds on $\pi$), denoted by $\pi \models F$, by induction on $F$ as follows.

# Semantics

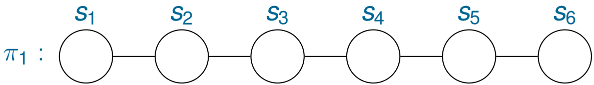
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and $F$ be an LTL formula.




We define the notion $F$ is true on $\pi$ (or $F$ holds on $\pi$), denoted by $\pi \models F$, by induction on $F$ as follows.
For all $i = 0, 1, \ldots$ denote by $\pi_i$ the sequence of states $s_i, s_{i+1}, s_{i+2} \ldots$ (note that $\pi_0 = \pi$).

# Semantics

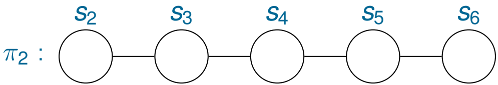
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and $F$ be an $\mathrm{LTL}$ formula.




We define the notion $F$ is true on $\pi$ (or $F$ holds on $\pi$), denoted by $\pi \models F$, by induction on $F$ as follows.
For all $i = 0, 1, \ldots$ denote by $\pi_i$ the sequence of states $s_i, s_{i+1}, s_{i+2} \ldots$ (note that $\pi_0 = \pi$).

# Semantics

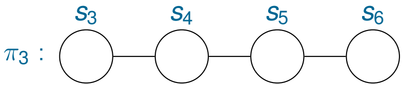
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and $F$ be an $\mathrm{LTL}$ formula.




We define the notion $F$ is true on $\pi$ (or $F$ holds on $\pi$), denoted by $\pi \models F$, by induction on $F$ as follows.
For all $i = 0, 1, \ldots$ denote by $\pi_i$ the sequence of states $s_i, s_{i+1}, s_{i+2} \ldots$ (note that $\pi_0 = \pi$).

# Semantics

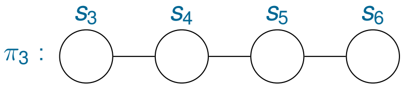
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and $F$ be an $\mathrm{LTL}$ formula.




We define the notion $F$ is true on $\pi$ (or $F$ holds on $\pi$), denoted by $\pi \models F$, by induction on $F$ as follows.
For all $i = 0, 1, \ldots$ denote by $\pi_i$ the sequence of states $s_i, s_{i+1}, s_{i+2} \ldots$ (note that $\pi_0 = \pi$).

# Semantics

Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and $F$ be an $\mathrm{LTL}$ formula.




We define the notion $F$ is true on $\pi$ (or $F$ holds on $\pi$), denoted by $\pi \models F$, by induction on $F$ as follows.
For all $i = 0, 1, \ldots$ denote by $\pi_i$ the sequence of states $s_i, s_{i+1}, s_{i+2} \ldots$ (note that $\pi_0 = \pi$).
To define $\pi \models F$ we will use $\pi_i \models G$ for some $i$ and $G$. We will sometimes (slightly informally) say that $G$ is true in $s_i$ or $G$ holds in $s_i$ to mean that $G$ is true on $\pi_i$.

# Semantics, formally

The semantics of propositional connectives is standard.

# Semantics, formally

The semantics of propositional connectives is standard.

Atomic formulas are true iff they are true in $s_0$.

# Semantics, formally

The semantics of propositional connectives is standard.

Atomic formulas are true iff they are true in $s_0$.

The semantics of formulas built using propositional connectives on $\pi$ is the same as in propositional logic where all subformulas are also evaluated on $\pi$.

# Semantics, formally

The semantics of propositional connectives is standard.

Atomic formulas are true iff they are true in $s_0$.

The semantics of formulas built using propositional connectives on $\pi$ is the same as in propositional logic where all subformulas are also evaluated on $\pi$.

1. $\pi \models \top$ and $\pi \not\models \bot$.

# Semantics, formally

The semantics of propositional connectives is standard.

Atomic formulas are true iff they are true in $s_0$.

The semantics of formulas built using propositional connectives on $\pi$ is the same as in propositional logic where all subformulas are also evaluated on $\pi$.

1. $\pi \models \top$ and $\pi \not\models \bot$.
2. $\pi \models x = v$ if $s_0 \models x = v$.

# Semantics, formally

The semantics of propositional connectives is standard.

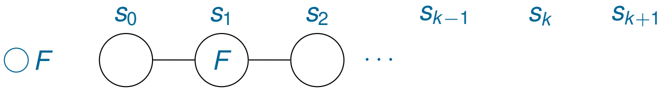
Atomic formulas are true iff they are true in $s_0$.

The semantics of formulas built using propositional connectives on $\pi$ is the same as in propositional logic where all subformulas are also evaluated on $\pi$.

1. $\pi \models \top$ and $\pi \not\models \bot$.
2. $\pi \models x = v$ if $s_0 \models x = v$.
3. $\pi \models F_1 \land \ldots \land F_n$ if for all $j = 1, \ldots, n$ we have $\pi \models F_j$;
   $\pi \models F_1 \lor \ldots \lor F_n$ if for some $j = 1, \ldots, n$ we have $\pi \models F_j$.
4. $\pi \models \neg F$ if $\pi \not\models F$.
5. $\pi \models F \to G$ 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 $\pi \models G$.

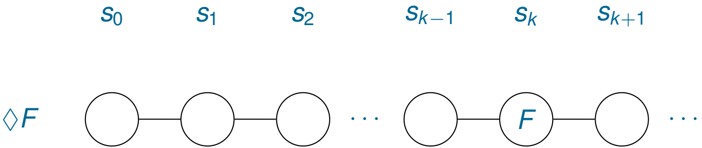
# Semantics of temporal operators

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



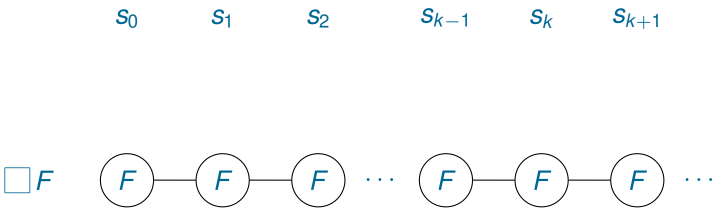
# Semantics of temporal operators

6. $\pi \models \bigcirc F$ if $\pi_1 \models F$;
   $\pi \models \Diamond F$ if for some $k \geq 0$ we have $\pi_k \models F$;



# Semantics of temporal operators

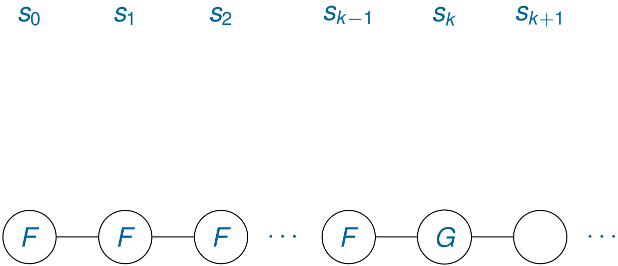
6. $\pi \models \bigcirc F$ if $\pi_1 \models F$;
   $\pi \models \Diamond F$ if for some $k \geq 0$ we have $\pi_k \models F$;
   $\pi \models \Box F$ if for all $i \geq 0$ we have $\pi_i \models F$.



# Semantics of temporal operators

6. $\pi \models \bigcirc F$ if $\pi_1 \models F$;
   $\pi \models \Diamond F$ if for some $k \geq 0$ we have $\pi_k \models F$;
   $\pi \models \Box F$ if for all $i \geq 0$ we have $\pi_i \models F$.
7. $\pi \models F \mathcal{U} G$ if for some $k \geq 0$ we have $\pi_k \models G$ and
   $\pi_0 \models F, \ldots, \pi_{k-1} \models F$;

$F \mathcal{U} G$

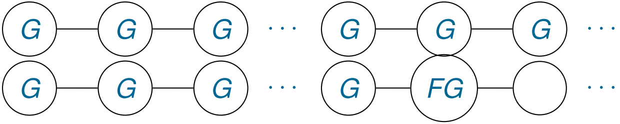


# Semantics of temporal operators

6. $\pi \models \bigcirc F$ if $\pi_1 \models F$;
   $\pi \models \Diamond F$ if for some $k \geq 0$ we have $\pi_k \models F$;
   $\pi \models \Box F$ if for all $i \geq 0$ we have $\pi_i \models F$.
7. $\pi \models F \,\mathcal{U}\, G$ if for some $k \geq 0$ we have $\pi_k \models G$ and
   $\pi_0 \models F, \ldots, \pi_{k-1} \models F$;
   $\pi \models F \,\mathcal{R}\, G$ if for all $k \geq 0$, either $\pi_k \models G$ or there exists $j < k$
   such that $\pi_j \models F$.

$s_0$ $s_1$ $s_2$ $s_{k-1}$ $s_k$ $s_{k+1}$

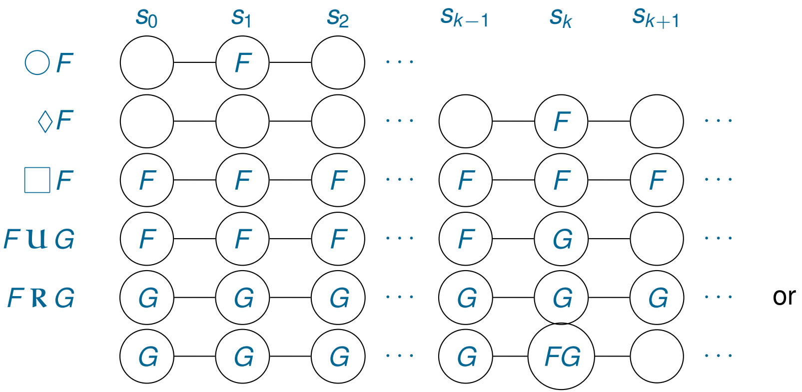
$F \,\mathcal{R}\, G$




or

# Semantics of temporal operators

6. $\pi \models \bigcirc F$ if $\pi_1 \models F$;
   $\pi \models \Diamond F$ if for some $k \geq 0$ we have $\pi_k \models F$;
   $\pi \models \Box F$ if for all $i \geq 0$ we have $\pi_i \models F$.
7. $\pi \models F \,\mathcal{U}\, G$ if for some $k \geq 0$ we have $\pi_k \models G$ and
   $\pi_0 \models F, \ldots, \pi_{k-1} \models F$;
   $\pi \models F \,\mathcal{R}\, G$ if for all $k \geq 0$, either $\pi_k \models G$ or there exists $j < k$
   such that $\pi_j \models F$.



# Standard properties???

Two $\mathrm{LTL}$ formulas $F$ and $G$ are called equivalent, denoted $F \equiv G$, if for every path $\pi$ we have $\pi \models F$ if and only if $\pi \models G$.

# Standard properties???

Two $\mathrm{LTL}$ formulas $F$ and $G$ are called equivalent, denoted $F \equiv G$, if for every path $\pi$ we have $\pi \models F$ if and only if $\pi \models G$.

We are not interested in satisfiability, validity etc. for temporal formulas.

# Standard properties???

Two $\mathrm{LTL}$ formulas $F$ and $G$ are called equivalent, denoted $F \equiv G$, if for every path $\pi$ we have $\pi \models F$ if and only if $\pi \models G$.

We are not interested in satisfiability, validity etc. for temporal formulas.

For an $\mathrm{LTL}$ formula $F$ we can consider two kinds of properties of $\mathbb{S}$:

1. does $F$ hold on some computation path for $\mathbb{S}$ from an initial state?
2. does $F$ hold on all computation paths for $\mathbb{S}$ from an initial state?

# Precedences of Connectives and Temporal Operators

| Connective | Precedence |
|---|---|
| $\neg, \bigcirc, \Diamond, \Box$ | 5 |
| $\mathbf{U}, \mathbf{R}$ | 4 |
| $\wedge, \vee$ | 3 |
| $\rightarrow$ | 2 |
| $\leftrightarrow$ | 1 |

# Expressing Some Properties

1. $F$ never holds in two consecutive states.

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \to \bigcirc\neg F)$

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \to \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \to \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\Box(F \to \Box F)$

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\square(F \to \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\square(F \to \square F)$
3. $F$ holds in at most one state.

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \rightarrow \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\Box(F \rightarrow \Box F)$
3. $F$ holds in at most one state. $\Box(F \rightarrow \bigcirc\Box\neg F)$

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \rightarrow \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\Box(F \rightarrow \Box F)$
3. $F$ holds in at most one state. $\Box(F \rightarrow \bigcirc\Box\neg F)$
4. $F$ holds in at least two states.

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\square(F \to \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\square(F \to \square F)$
3. $F$ holds in at most one state. $\square(F \to \bigcirc\square\neg F)$
4. $F$ holds in at least two states. $\Diamond(F \land \bigcirc\Diamond F)$

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \to \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$. $\Box(F \to \Box F)$
3. $F$ holds in at most one state. $\Box(F \to \bigcirc\Box\neg F)$
4. $F$ holds in at least two states. $\Diamond(F \wedge \bigcirc\Diamond F)$
5. Unless $s_i$ is the first state of the path, if $F$ holds in state $s_i$, then $G$ must hold in at least one of the two states just before $s_i$, that is $s_{i-1}$ and $s_{i-2}$.

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \to \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\Box(F \to \Box F)$
3. $F$ holds in at most one state. $\Box(F \to \bigcirc\Box\neg F)$
4. $F$ holds in at least two states. $\Diamond(F \land \bigcirc\Diamond F)$
5. Unless $s_i$ is the first state of the path, if $F$ holds in state $s_i$, then $G$ must hold in at least one of the two states just before $s_i$, that is $s_{i-1}$ and $s_{i-2}$. $(\bigcirc F \to G) \land \Box(\bigcirc\bigcirc F \to G \lor \bigcirc G)$

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \to \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\Box(F \to \Box F)$
3. $F$ holds in at most one state. $\Box(F \to \bigcirc\Box\neg F)$
4. $F$ holds in at least two states. $\Diamond(F \wedge \bigcirc\Diamond F)$
5. Unless $s_i$ is the first state of the path, if $F$ holds in state $s_i$, then $G$ must hold in at least one of the two states just before $s_i$, that is $s_{i-1}$ and $s_{i-2}$. $(\bigcirc F \to G) \wedge \Box(\bigcirc\bigcirc F \to G \vee \bigcirc G)$
6. $F$ happens infinitely often.

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \rightarrow \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\Box(F \rightarrow \Box F)$
3. $F$ holds in at most one state. $\Box(F \rightarrow \bigcirc\Box\neg F)$
4. $F$ holds in at least two states. $\Diamond(F \wedge \bigcirc\Diamond F)$
5. Unless $s_i$ is the first state of the path, if $F$ holds in state $s_i$, then $G$ must hold in at least one of the two states just before $s_i$, that is $s_{i-1}$ and $s_{i-2}$. $(\bigcirc F \rightarrow G) \wedge \Box(\bigcirc\bigcirc F \rightarrow G \vee \bigcirc G)$
6. $F$ happens infinitely often. $\Box\Diamond F$

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\square(F \rightarrow \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$. $\square(F \rightarrow \square F)$
3. $F$ holds in at most one state. $\square(F \rightarrow \bigcirc\square\neg F)$
4. $F$ holds in at least two states. $\Diamond(F \wedge \bigcirc\Diamond F)$
5. Unless $s_i$ is the first state of the path, if $F$ holds in state $s_i$, then $G$ must hold in at least one of the two states just before $s_i$, that is $s_{i-1}$ and $s_{i-2}$. $(\bigcirc F \rightarrow G) \wedge \square(\bigcirc\bigcirc F \rightarrow G \vee \bigcirc G)$
6. $F$ happens infinitely often. $\square\Diamond F$
7. $F$ holds in each even state and does not hold in each odd state (states are counted from $0$).

# Expressing Some Properties

1. $F$ never holds in two consecutive states. $\Box(F \to \bigcirc\neg F)$
2. If $F$ holds in a state $s$, it also holds in all states after $s$.
   $\Box(F \to \Box F)$
3. $F$ holds in at most one state. $\Box(F \to \bigcirc\Box\neg F)$
4. $F$ holds in at least two states. $\Diamond(F \land \bigcirc\Diamond F)$
5. Unless $s_i$ is the first state of the path, if $F$ holds in state $s_i$, then $G$ must hold in at least one of the two states just before $s_i$, that is $s_{i-1}$ and $s_{i-2}$. $(\bigcirc F \to G) \land \Box(\bigcirc\bigcirc F \to G \lor \bigcirc G)$
6. $F$ happens infinitely often. $\Box\Diamond F$
7. $F$ holds in each even state and does not hold in each odd state (states are counted from $0$). $F \land \Box(F \leftrightarrow \bigcirc\neg F)$.

# Not all “reasonable” properties are expressible in LTL

$p$ holds in all even states.

# End of Lecture 19

Slides for lecture 19 end here . . .

# Meaning of Some Formulas

- $\Diamond \Box F$;

# Meaning of Some Formulas

- $\Diamond \Box F$;
- $\Box (F \rightarrow \bigcirc F)$;

# Meaning of Some Formulas

- $\Diamond \Box F$;
- $\Box(F \to \bigcirc F)$;
- $\neg F\, \mathcal{U}\, \Box F$;

# Meaning of Some Formulas

- $\Diamond \Box F$;
- $\Box(F \to \bigcirc F)$;
- $\neg F \, \mathcal{U} \, \Box F$;
- $F \, \mathcal{U} \, \neg F$;

# Meaning of Some Formulas

- $\Diamond \Box F$;
- $\Box (F \rightarrow \bigcirc F)$;
- $\neg F \, \mathcal{U} \, \Box F$;
- $F \, \mathcal{U} \, \neg F$;
- $\Diamond F \wedge \Box (F \rightarrow \bigcirc F)$;

# Meaning of Some Formulas

- $\Diamond \Box F$;
- $\Box(F \rightarrow \bigcirc F)$;
- $\neg F \, \mathcal{U} \, \Box F$;
- $F \, \mathcal{U} \neg F$;
- $\Diamond F \land \Box(F \rightarrow \bigcirc F)$;
- $\Box \Diamond F$;

# Meaning of Some Formulas

- $\Diamond \Box F$;
- $\Box(F \to \bigcirc F)$;
- $\neg F \,\mathcal{U}\, \Box F$;
- $F \,\mathcal{U}\, \neg F$;
- $\Diamond F \land \Box(F \to \bigcirc F)$;
- $\Box \Diamond F$;
- $F \land \Box(F \leftrightarrow \neg \bigcirc F)$;

# Equivalences: Unwinding Properties

$$
\begin{array}{rcl}
\Diamond F & \equiv & F \vee \bigcirc \Diamond F \\
\Box F & \equiv & F \wedge \bigcirc \Box F \\
F \,\mathcal{U}\, G & \equiv & G \vee (F \wedge \bigcirc (F \,\mathcal{U}\, G)) \\
F \,\mathcal{R}\, G & \equiv & G \wedge (F \vee \bigcirc (F \,\mathcal{R}\, G))
\end{array}
$$

# Equivalences: Negation of Temporal Operators

$$
\begin{array}{rcl}
\neg \bigcirc F & \equiv & \bigcirc \neg F \\
\neg \Diamond F & \equiv & \Box \neg F \\
\neg \Box F & \equiv & \Diamond \neg F \\
\neg (F \,\mathbf{U}\, G) & \equiv & \neg F \,\mathbf{R}\, \neg G \\
\neg (F \,\mathbf{R}\, G) & \equiv & \neg F \,\mathbf{U}\, \neg G
\end{array}
$$

# Expressing Temporal Operators Using $\mathcal{U}$

$$
\begin{array}{rcl}
\Diamond F & \equiv & \top \,\mathcal{U}\, F \\
\Box F & \equiv & \neg(\top \,\mathcal{U}\, \neg F) \\
F \,\mathcal{R}\, G & \equiv & \neg(\neg F \,\mathcal{U}\, \neg G).
\end{array}
$$

Therefore, all operators can be expressed using $\bigcirc$ and $\mathcal{U}$.

# Other Equivalences

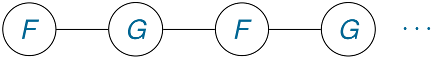
$$
\begin{aligned}
\Diamond(F \vee G) &\equiv \Diamond F \vee \Diamond G \\
\Box(F \wedge G) &\equiv \Box F \wedge \Box G
\end{aligned}
$$

But

$$
\begin{aligned}
\Box(F \vee G) &\not\equiv \Box F \vee \Box G \\
\Diamond(F \wedge G) &\not\equiv \Diamond F \wedge \Diamond G
\end{aligned}
$$

# How to Show that Two Formulas are not Equivalent?

Find a path that satisfies one of the formulas but not the other. For example for $\square(F \vee G)$ and $\square F \vee \square G$.



# 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

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 have 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.

# Reasoning About Transitions

Consider the following properties:

1. "one cannot have 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 have 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.

# Example

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

# Representing Temporal Properties of Transitions

1. One cannot have two beers without inserting a coin in between getting them.

# Representing Temporal Properties of Transitions

1. One cannot have two beers without inserting a coin in between getting them.

$$\Box(\mathrm{tr} = \mathit{Dispense\_beer} \rightarrow \bigcirc(\Box\,\mathrm{tr} \neq \mathit{Dispence\_beer} \ \vee \\ \mathrm{tr} \neq \mathit{Dispence\_beer}\ \mathcal{U}\ \mathrm{tr} = \mathit{Insert\_coin}))$$

# Representing Temporal Properties of Transitions

1. One cannot have two beers without inserting a coin in between getting them.

$$\Box(\mathrm{tr} = \mathit{Dispense\_beer} \rightarrow \bigcirc(\Box \mathrm{tr} \neq \mathit{Dispence\_beer} \ \vee \\ \mathrm{tr} \neq \mathit{Dispence\_beer}\ \mathcal{U}\ \mathrm{tr} = \mathit{Insert\_coin}))$$

2. If we never have two recharge transitions in a row, then the next transition after a recharge must be a customer arrival.

# Representing Temporal Properties of Transitions

1. One cannot have two beers without inserting a coin in between getting them.

$$\Box(\text{tr} = \textit{Dispense\_beer} \rightarrow \bigcirc(\Box \text{tr} \neq \textit{Dispence\_beer} \vee \text{tr} \neq \textit{Dispence\_beer}\, \mathcal{U}\, \text{tr} = \textit{Insert\_coin}))$$

2. If we never have two recharge transitions in a row, then the next transition after a recharge must be a customer arrival.

$$\Box(\text{tr} = \textit{Recharge} \rightarrow \bigcirc \text{tr} \neq \textit{Recharge}) \rightarrow$$
$$\Box(\text{tr} = \textit{Recharge} \rightarrow \bigcirc \text{tr} = \textit{Customer\_arrives})$$

# Representing Temporal Properties of Transitions

1. One cannot have two beers without inserting a coin in between getting them.

$$\Box(\mathrm{tr} = \mathit{Dispense\_beer} \rightarrow \bigcirc(\Box\mathrm{tr} \neq \mathit{Dispence\_beer} \vee \mathrm{tr} \neq \mathit{Dispence\_beer}\, \mathcal{U}\, \mathrm{tr} = \mathit{Insert\_coin}))$$

2. If we never have two recharge transitions in a row, then the next transition after a recharge must be a customer arrival.

$$\Box(\mathrm{tr} = \mathit{Recharge} \rightarrow \bigcirc\mathrm{tr} \neq \mathit{Recharge}) \rightarrow$$
$$\Box(\mathrm{tr} = \mathit{Recharge} \rightarrow \bigcirc\mathrm{tr} = \mathit{Customer\_arrives})$$

3. The value of customer can only be changed as a result of either *Customer_arrives* or *Customer_leaves*.

# Representing Temporal Properties of Transitions

1. One cannot have two beers without inserting a coin in between getting them.

$$\Box(\text{tr} = \textit{Dispense\_beer} \rightarrow \bigcirc(\Box\text{tr} \neq \textit{Dispence\_beer} \;\vee \\ \text{tr} \neq \textit{Dispence\_beer}\;\mathcal{U}\;\text{tr} = \textit{Insert\_coin}))$$

2. If we never have two recharge transitions in a row, then the next transition after a recharge must be a customer arrival.

$$\Box(\text{tr} = \textit{Recharge} \rightarrow \bigcirc\text{tr} \neq \textit{Recharge}) \rightarrow \\ \Box(\text{tr} = \textit{Recharge} \rightarrow \bigcirc\text{tr} = \textit{Customer\_arrives})$$

3. The value of customer can only be changed as a result of either *Customer_arrives* or *Customer_leaves*.

$$\Box(\bigwedge_{v \in \textit{dom}(\text{customer})}(\text{customer} = v \wedge \bigcirc\text{customer} \neq v) \rightarrow \\ \text{tr} = \textit{Customer\_arrives} \vee \text{tr} = \textit{Customer\_leaves})$$

# Representing Temporal Properties of Transitions

1. If somebody inserts a coin twice and then gets a beer, then the amount of coins in the slot will not change.

# Representing Temporal Properties of Transitions

1. If somebody inserts a coin twice and then gets a beer, then the amount of coins in the slot will not change.

$$\bigwedge_{v \in dom(\text{coin})} \Box(\text{customer} = v \wedge \text{tr} = \mathit{Coin\_insert} \wedge \bigcirc \text{tr} = \mathit{Coin\_insert} \wedge \bigcirc\bigcirc \text{tr} = \mathit{Dispense\_beer} \rightarrow \bigcirc\bigcirc\bigcirc \text{customer} = v)$$

# Representing Temporal Properties of Transitions

1. If somebody inserts a coin twice and then gets a beer, then the amount of coins in the slot will not change.

$$\bigwedge_{v \in dom(\text{coin})} \Box(\text{customer} = v \wedge \\ \text{tr} = \mathit{Coin\_insert} \wedge \\ \bigcirc \text{tr} = \mathit{Coin\_insert} \wedge \\ \bigcirc\bigcirc \text{tr} = \mathit{Dispense\_beer} \rightarrow \\ \bigcirc\bigcirc\bigcirc \text{customer} = v)$$

2. If the system is recharged from time to time, then after each *Dispense_beer* the customer will leave.

# Representing Temporal Properties of Transitions

1. If somebody inserts a coin twice and then gets a beer, then the amount of coins in the slot will not change.

$$\bigwedge_{v \in dom(\text{coin})} \Box(\text{customer} = v \wedge \\ \text{tr} = \mathit{Coin\_insert} \wedge \\ \bigcirc \text{tr} = \mathit{Coin\_insert} \wedge \\ \bigcirc\bigcirc \text{tr} = \mathit{Dispense\_beer} \rightarrow \\ \bigcirc\bigcirc\bigcirc \text{customer} = v)$$

2. If the system is recharged from time to time, then after each *Dispense_beer* the customer will leave.

$$\Box\Diamond \text{tr} = \mathit{Recharge} \rightarrow \\ \Box(\text{tr} = \mathit{Dispense\_beer} \rightarrow \Diamond \text{tr} = \mathit{Customer\_leaves})$$

# End of Lecture 20

Slides for lecture 20 end here . . .