CS:4350 Logic in Computer Science

Linear Temporal Logic

Cesare Tinelli

Spring 2022

THE UNIVERSITY OF IOWA

Credits

These slides are largely based on slides originally developed by **Andrei Voronkov** at the University of Manchester. Adapted by permission.

Outline

Computation Tree

Let $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$ be a transition system and $s_0 \in S$

Computation tree for $\mathbb{S}$ starting at s_0:

Defined as the (possibly infinite) tree C such that

1. every node of C is labeled by a state in S
2. the root of C is labeled by s_0
3. every node in the tree labeled by a state s has a child labeled by a state s' iff $(s, s') \in T$

Computation Tree

Let $\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$ be a transition system and $s_0 \in S$

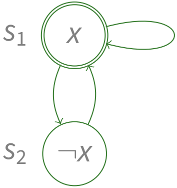
Computation tree for $\mathbb{S}$ starting at s_0:

Defined as the (possibly infinite) tree C such that

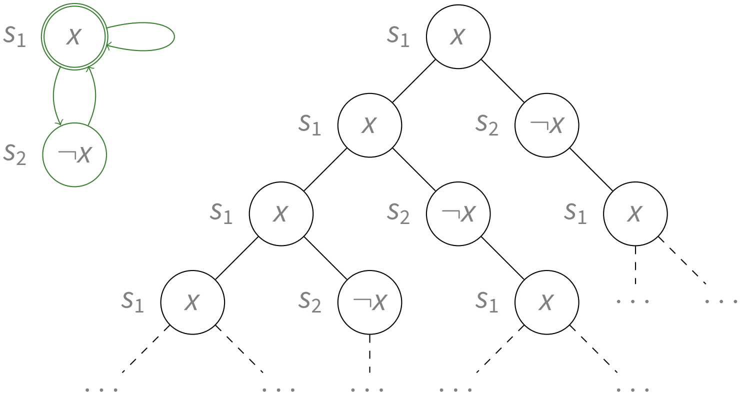
1. every node of C is labeled by a state in S
2. the root of C is labeled by s_0
3. every node in the tree labeled by a state s has a child labeled by a state s' iff $(s, s') \in T$

Computation path for $\mathbb{S}$ starting at s_0: any branch $s_0, s_1, \ldots$ in C

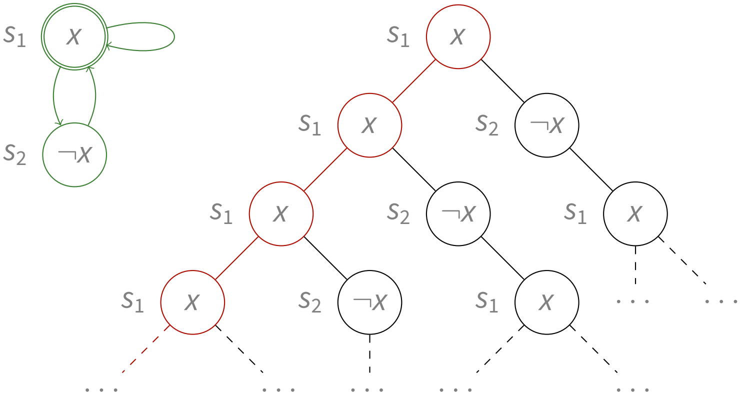
Computation Trees and Paths



Computation Trees and Paths

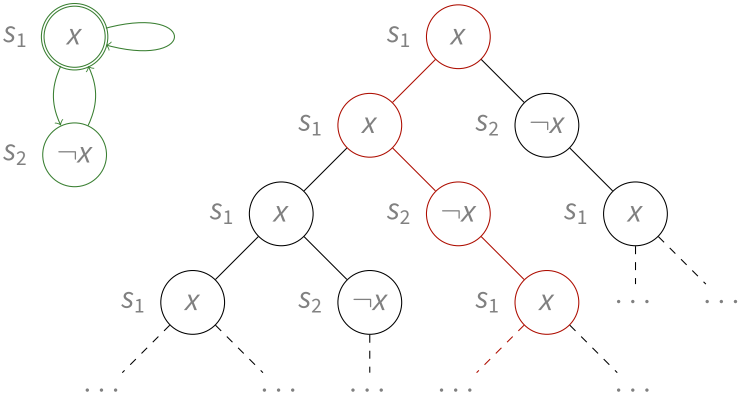


Computation Trees and Paths



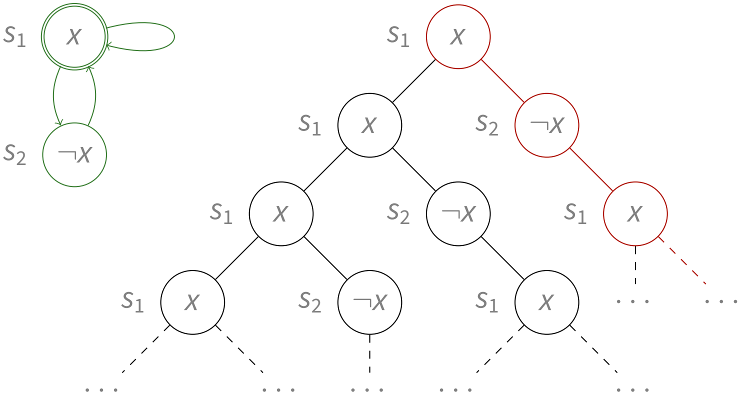
computation path

Computation Trees and Paths



computation path

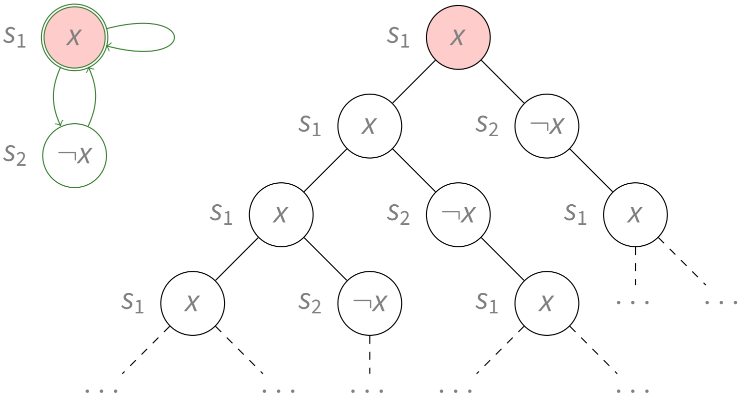
Computation Trees and Paths



computation path

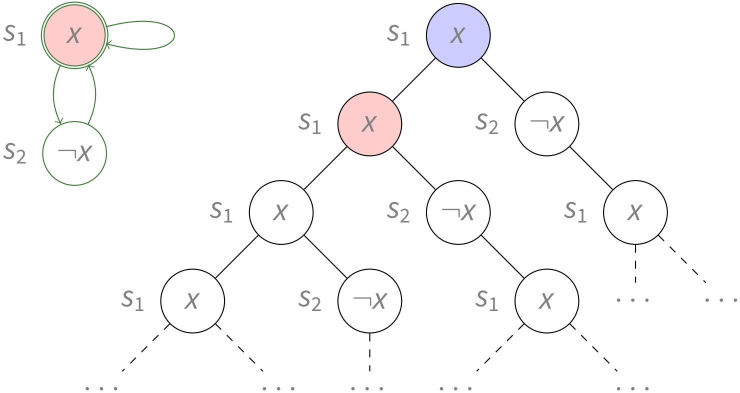
Computation

Every path in the computation tree corresponds to a computation:



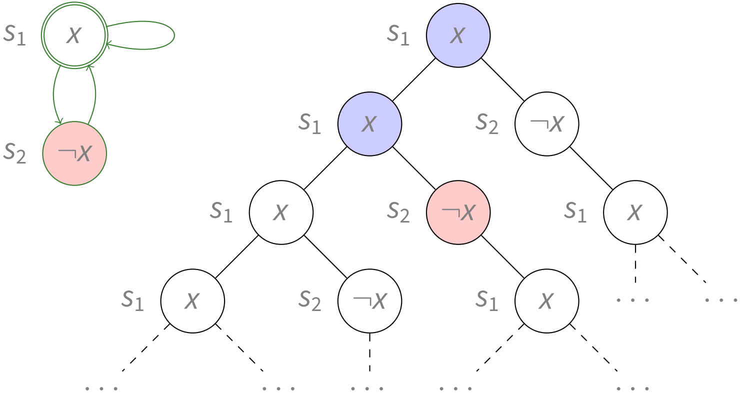
Computation

Every path in the computation tree corresponds to a computation:



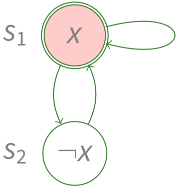
Computation

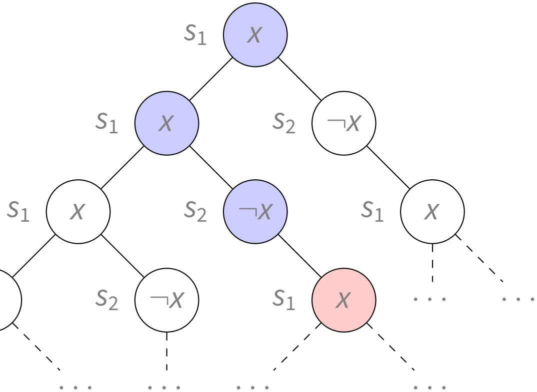
Every path in the computation tree corresponds to a computation:



Computation

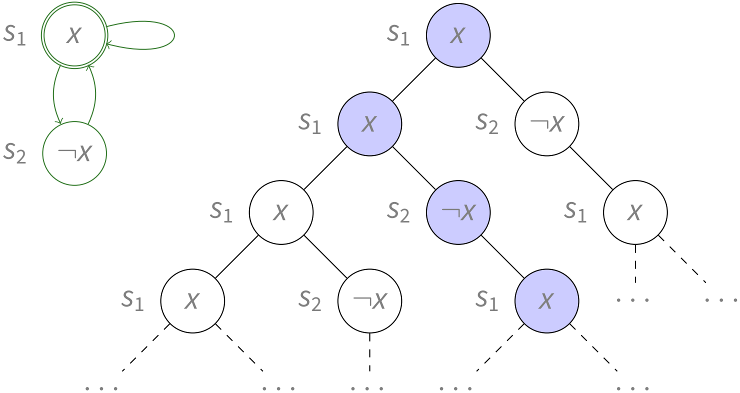
Every path in the computation tree corresponds to a computation:



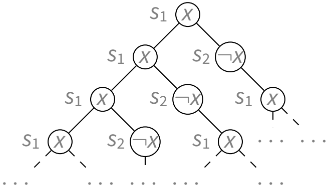


Computation

Every path in the computation tree corresponds to a computation:



Properties

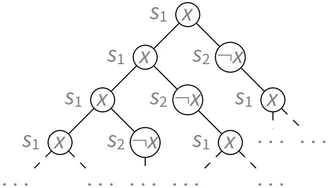


$\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$

1. The computation paths of $\mathbb{S}$ are exactly the branches in the computation trees for $\mathbb{S}$
2. If C is a computation tree for $\mathbb{S}$, the subtree of C rooted at a state s is the computation tree for $\mathbb{S}$ starting at s

   (every subtree of a computation tree is itself a computation tree)
3. For all $s \in S$, there is a unique computation tree for $\mathbb{S}$ starting at s

Properties



$\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$

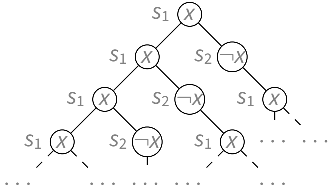
1. The computation paths of $\mathbb{S}$ are exactly the branches in the computation trees for $\mathbb{S}$

2. If C is a computation tree for $\mathbb{S}$, the subtree of C rooted at a state s is the computation tree for $\mathbb{S}$ starting at s

   (every subtree of a computation tree is itself a computation tree)

3. For all $s \in S$, there is a unique computation tree for $\mathbb{S}$ starting at s

Properties



$\mathbb{S} = (S, In, T, \mathcal{X}, dom, L)$

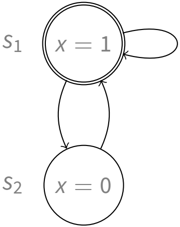
1. The computation paths of $\mathbb{S}$ are exactly the branches in the computation trees for $\mathbb{S}$

2. If C is a computation tree for $\mathbb{S}$, the subtree of C rooted at a state s is the computation tree for $\mathbb{S}$ starting at s

   (every subtree of a computation tree is itself a computation tree)

3. For all $s \in S$, there is a unique computation tree for $\mathbb{S}$ starting at s

Representing system paths with ω-regular expressions

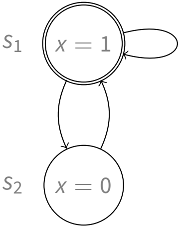


System paths

$s_1^\omega = s_1 s_1 s_1 \cdots$

$(s_1 s_2)^\omega = s_1 s_2 s_1 s_2 \cdots$

Representing system paths with ω-regular expressions



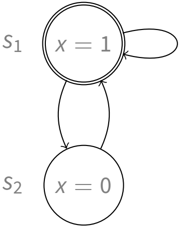
System paths

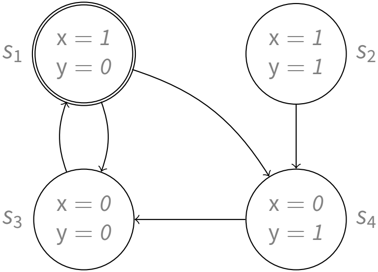
$s_1^\omega = s_1 s_1 s_1 \cdots$

$(s_1 s_2)^\omega = s_1 s_2 s_1 s_2 \cdots$

not a system path: $(s_2 s_1)^\omega$

Representing system paths with ω-regular expressions





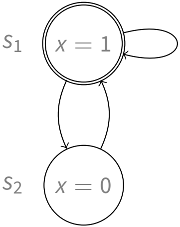
System paths

$s_1^\omega = s_1 s_1 s_1 \cdots$

$(s_1 s_2)^\omega = s_1 s_2 s_1 s_2 \cdots$

not a system path: $(s_2 s_1)^\omega$

Representing system paths with ω-regular expressions

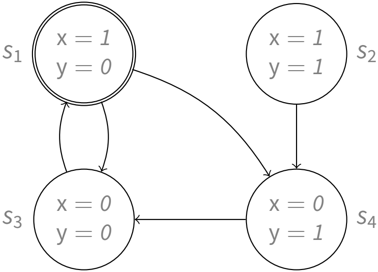


System paths

$s_1^\omega = s_1 s_1 s_1 \cdots$

$(s_1 s_2)^\omega = s_1 s_2 s_1 s_2 \cdots$

not a system path: $(s_2 s_1)^\omega$



$(s_1 s_3)^\omega$

$s_1 s_4 (s_3 s_1)^\omega$

$(s_1 s_4 s_3)^\omega$

$s_1 s_4 (s_3 s_1)^n s_4 (s_3 s_1)^\omega$ for all $n > 1$

$(s_1 s_4 s_3)^n s_1 (s_3 s_1)^\omega$ for all $n > 0$

$\ldots$

Linear Temporal Logic

Linear Temporal Logic (LTL) is a logic for reasoning about properties of computation paths

Linear Temporal Logic

Linear Temporal Logic (LTL) is a logic for reasoning about properties of computation paths

Formulas are built in the same way as in PLFD, 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

Linear Temporal Logic

Linear Temporal Logic (LTL) is a logic for reasoning about properties of computation paths

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

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

$\bigcirc$	**next**
$\Box$	**always** (in the future)
$\Diamond$	**eventually** (in the future)
$\mathcal{U}$	**until**
$\mathcal{R}$	**release**

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

- unary temporal operators have the same precedence as $\neg$
- binary temporal operators have higher precedence than binary Boolean connectives

Semantics (intuitive)

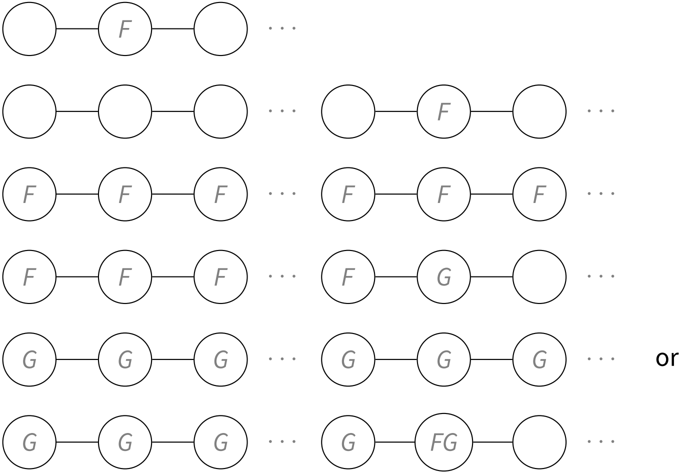
next: $\bigcirc F$

eventually: $\Diamond F$

always: $\Box F$

until: $F \, \mathcal{U} \, G$

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



Semantics

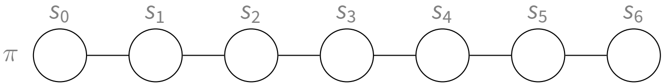
LTL formulas express properties of computations or computation paths

Semantics

LTL formulas express properties of computations or computation paths

$\pi = s_0, s_1, s_2, \ldots$, sequence of states
$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$, subsequence of π starting at $i \geq 0$

F, an LTL formula

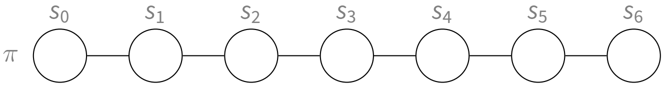


Semantics

LTL formulas express properties of computations or computation paths

$\pi = s_0, s_1, s_2, \ldots$, sequence of states
$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$, subsequence of π starting at $i \geq 0$

F, an LTL formula

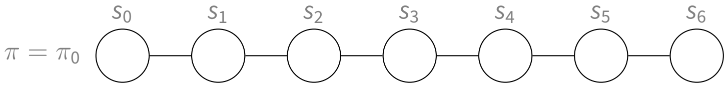


Semantics

LTL formulas express properties of computations or computation paths

$\pi = s_0, s_1, s_2, \ldots$, sequence of states
$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$, subsequence of π starting at $i \geq 0$

F, an LTL formula

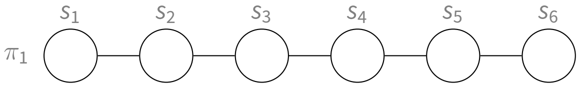


Semantics

LTL formulas express properties of computations or computation paths

$\pi = s_0, s_1, s_2, \ldots$, sequence of states
$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$, subsequence of π starting at $i \geq 0$

F, an LTL formula

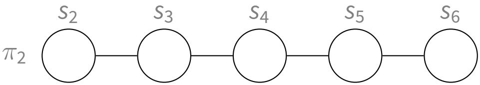


Semantics

LTL formulas express properties of computations or computation paths

$\pi = s_0, s_1, s_2, \ldots$, sequence of states
$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$, subsequence of π starting at $i \geq 0$

F, an LTL formula

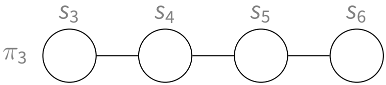


Semantics

LTL formulas express properties of computations or computation paths

$\pi = s_0, s_1, s_2, \ldots$, sequence of states
$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$, subsequence of π starting at $i \geq 0$

F, an LTL formula

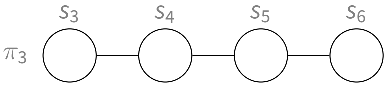


Semantics

LTL formulas express properties of computations or computation paths

$\pi = s_0, s_1, s_2, \ldots$, sequence of states
$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$, subsequence of π starting at $i \geq 0$

F, an LTL formula



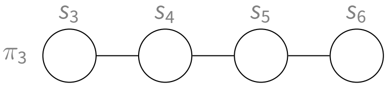
F holds on π or *π satisfies F*, written $\pi \models F$, iff *F holds on π_0*, written $\pi_0 \models F$, where $\pi_i \models F$ is defined for all $i \geq 0$ by induction on F

Semantics

LTL formulas express properties of computations or computation paths

$\pi = s_0, s_1, s_2, \ldots$, sequence of states
$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$, subsequence of π starting at $i \geq 0$

F, an LTL formula



F holds on π or π *satisfies F*, written $\pi \models F$, iff *F holds on* π_0, written $\pi_0 \models F$, where $\pi_i \models F$ is defined for all $i \geq 0$ by induction on F

We will informally say that *F holds in* s_i to mean that F holds on π_i

Semantics, formally

$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$

Atomic formulas hold on π_i iff they hold in s_i:

1. $\pi_i \models x = v$ if $s_i \models x = v$

$\vdots$

2. $\pi_i \models \top$ and $\pi_i \not\models \bot$
3. $\pi_i \models \neg F$ if $\pi_i \not\models F$
4. $\pi_i \models F_1 \wedge \cdots \wedge F_n$ if $\pi_i \models F_j$ for all $j = 1, \ldots, n$
   $\pi_i \models F_1 \vee \cdots \vee F_n$ if $\pi_i \models F_j$ for some $j = 1, \ldots, n$
5. $\pi_i \models F \rightarrow G$ if either $\pi_i \not\models F$ or $\pi_i \models G$
   $\pi_i \models F \leftrightarrow G$ if either both $\pi_i \not\models F$ and $\pi_i \not\models G$ or both $\pi_i \models F$ and $\pi_i \models G$

Semantics, formally

$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$

Atomic formulas hold on π_i iff they hold in s_i:

1. $\pi_i \models x = v$ if $s_i \models x = v$

The semantics of formulas whose top symbol is a propositional connective is the same as in PL, with all subformulas also evaluated on π_i:

2. $\pi_i \models \top$ and $\pi_i \not\models \bot$
3. $\pi_i \models \neg F$ if $\pi_i \not\models F$
4. $\pi_i \models F_1 \land \cdots \land F_n$ if $\pi_i \models F_j$ for all $j = 1, \ldots, n$
   $\pi_i \models F_1 \lor \cdots \lor F_n$ if $\pi_i \models F_j$ for some $j = 1, \ldots, n$
5. $\pi_i \models F \to G$ if either $\pi_i \not\models F$ or $\pi_i \models G$
   $\pi_i \models F \leftrightarrow G$ if either both $\pi_i \not\models F$ and $\pi_i \not\models G$ or both $\pi_i \models F$ and $\pi_i \models G$

Semantics, formally

$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$

Atomic formulas hold on π_i iff they hold in s_i:

1. $\pi_i \models x = v$ if $s_i \models x = v$

The semantics of formulas whose top symbol is a propositional connective is the same as in PL, with all subformulas also evaluated on π_i:

2. $\pi_i \models \top$ and $\pi_i \not\models \bot$
3. $\pi_i \models \neg F$ if $\pi_i \not\models F$
4. $\pi_i \models F_1 \wedge \cdots \wedge F_n$ if $\pi_i \models F_j$ for all $j = 1, \ldots, n$
   $\pi_i \models F_1 \vee \cdots \vee F_n$ if $\pi_i \models F_j$ for some $j = 1, \ldots, n$
5. $\pi_i \models F \rightarrow G$ if either $\pi_i \not\models F$ or $\pi_i \models G$
   $\pi_i \models F \leftrightarrow G$ if either both $\pi_i \not\models F$ and $\pi_i \not\models G$ or both $\pi_i \models F$ and $\pi_i \models G$

Semantics, formally

$$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$$

Atomic formulas hold on π_i iff they hold in s_i:

1. $\pi_i \models x = v$ if $s_i \models x = v$

The semantics of formulas whose top symbol is a propositional connective is the same as in PL, with all subformulas also evaluated on π_i:

2. $\pi_i \models \top$ and $\pi_i \not\models \bot$
3. $\pi_i \models \neg F$ if $\pi_i \not\models F$
4. $\pi_i \models F_1 \wedge \cdots \wedge F_n$ if $\pi_i \models F_j$ for all $j = 1, \ldots, n$
   $\pi_i \models F_1 \vee \cdots \vee F_n$ if $\pi_i \models F_j$ for some $j = 1, \ldots, n$
5. $\pi_i \models F \rightarrow G$ if either $\pi_i \not\models F$ or $\pi_i \models G$
   $\pi_i \models F \leftrightarrow G$ if either both $\pi_i \not\models F$ and $\pi_i \not\models G$ or both $\pi_i \models F$ and $\pi_i \models G$

Semantics, formally

$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$

Atomic formulas hold on π_i iff they hold in s_i:

1. $\pi_i \models x = v$ if $s_i \models x = v$

The semantics of formulas whose top symbol is a propositional connective is the same as in PL, with all subformulas also evaluated on π_i:

2. $\pi_i \models \top$ and $\pi_i \not\models \bot$
3. $\pi_i \models \neg F$ if $\pi_i \not\models F$
4. $\pi_i \models F_1 \wedge \cdots \wedge F_n$ if $\pi_i \models F_j$ for all $j = 1, \ldots, n$
   $\pi_i \models F_1 \vee \cdots \vee F_n$ if $\pi_i \models F_j$ for some $j = 1, \ldots, n$
5. $\pi_i \models F \rightarrow G$ if either $\pi_i \not\models F$ or $\pi_i \models G$
   $\pi_i \models F \leftrightarrow G$ if either both $\pi_i \not\models F$ and $\pi_i \not\models G$ or both $\pi_i \models F$ and $\pi_i \models G$

Semantics, formally

$\pi_i = s_i, s_{i+1}, s_{i+2}, \ldots$

Atomic formulas hold on π_i iff they hold in s_i:

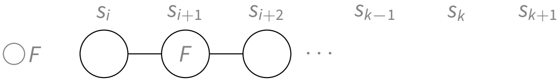
1. $\pi_i \models x = v$ if $s_i \models x = v$

The semantics of formulas whose top symbol is a propositional connective is the same as in PL, with all subformulas also evaluated on π_i:

2. $\pi_i \models \top$ and $\pi_i \not\models \bot$
3. $\pi_i \models \neg F$ if $\pi_i \not\models F$
4. $\pi_i \models F_1 \wedge \cdots \wedge F_n$ if $\pi_i \models F_j$ for all $j = 1, \ldots, n$
   $\pi_i \models F_1 \vee \cdots \vee F_n$ if $\pi_i \models F_j$ for some $j = 1, \ldots, n$
5. $\pi_i \models F \rightarrow G$ if either $\pi_i \not\models F$ or $\pi_i \models G$
   $\pi_i \models F \leftrightarrow G$ if either both $\pi_i \not\models F$ and $\pi_i \not\models G$ or both $\pi_i \models F$ and $\pi_i \models G$

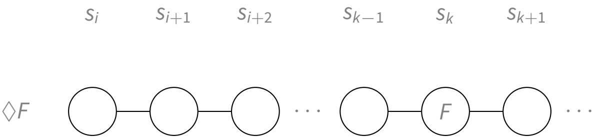
Semantics, formally

6. $\pi_i \models \bigcirc F$ if $\pi_{i+1} \models F$



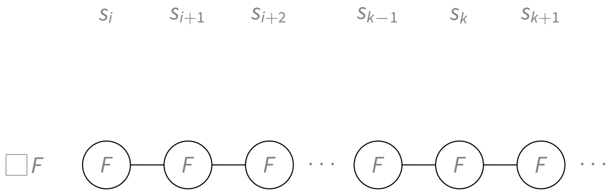
Semantics, formally

6. $\pi_i \models \bigcirc F$ if $\pi_{i+1} \models F$
   $\pi_i \models \Diamond F$ if for some $k \geq i$ we have $\pi_k \models F$



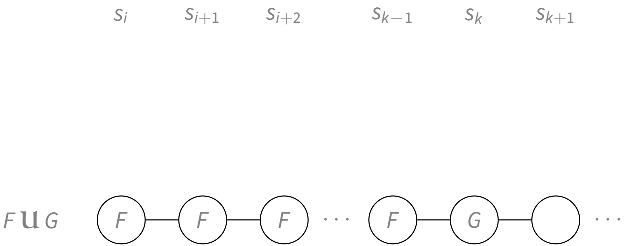
Semantics, formally

6. $\pi_i \models \bigcirc F$ if $\pi_{i+1} \models F$
   $\pi_i \models \Diamond F$ if for some $k \geq i$ we have $\pi_k \models F$
   $\pi_i \models \Box F$ if for all $k \geq i$ we have $\pi_k \models F$



Semantics, formally

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

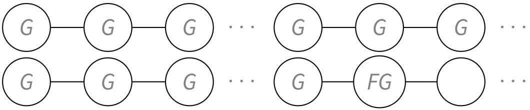


Semantics, formally

6. $\pi_i \models \bigcirc F$ if $\pi_{i+1} \models F$
   $\pi_i \models \Diamond F$ if for some $k \geq i$ we have $\pi_k \models F$
   $\pi_i \models \Box F$ if for all $k \geq i$ we have $\pi_k \models F$
7. $\pi_i \models F \, \mathbf{U} \, G$ if for some $k \geq i$ we have $\pi_k \models G$ and $\pi_i \models F, \ldots, \pi_{k-1} \models F$
   $\pi_i \models F \, \mathbf{R} \, G$ if either for all $k \geq i$ we have $\pi_i \models G$
   or for some $k \geq i$ and all $j = i, \ldots, k$ we have $\pi_j \models G$ and $\pi_k \models F$

s_i s_{i+1} s_{i+2} s_{k-1} s_k s_{k+1}

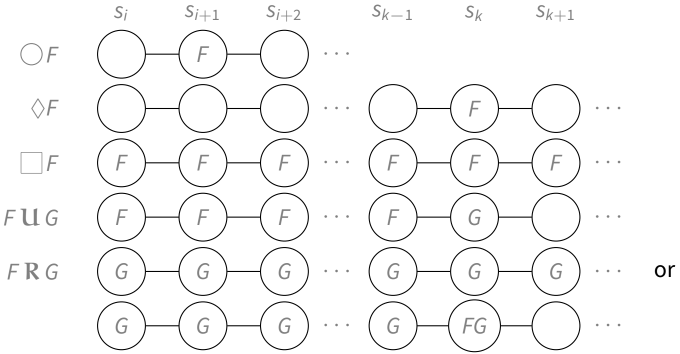
$F \, \mathbf{R} \, G$



or

Semantics, formally

6. $\pi_i \models \bigcirc F$ if $\pi_{i+1} \models F$
   $\pi_i \models \Diamond F$ if for some $k \geq i$ we have $\pi_k \models F$
   $\pi_i \models \Box F$ if for all $k \geq i$ we have $\pi_k \models F$
7. $\pi_i \models F \,\mathbf{U}\, G$ if for some $k \geq i$ we have $\pi_k \models G$ and $\pi_i \models F, \ldots, \pi_{k-1} \models F$
   $\pi_i \models F \,\mathbf{R}\, G$ if either for all $k \geq i$ we have $\pi_i \models G$
   or for some $k \geq i$ and all $j = i, \ldots, k$ we have $\pi_j \models G$ and $\pi_k \models F$



Example

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	...
p	1	1	1	1	1	1	1	1	0	0	1	1	1	1^ω
q	0	0	0	0	0	0	0	0	1	0	0	1	0	0^ω
$\bigcirc p$	1	1	1	1	1	1	1	0	0	1	1	1	1	1^ω
$\Diamond q$	1	1	1	1	1	1	1	1	1	1	1	1	0	0^ω
$\Box p$	0	0	0	0	0	0	0	0	0	0	1	1	1	1^ω
$p\,\mathbf{U}\,q$	1	1	1	1	1	1	1	1	1	0	1	1	0	0^ω
a	0	0	1	0	0	1	0	0	1	0	1	0	0	0^ω
b	1	1	1	1	1	1	0	1	1	1	1	0	1	1^ω
$a\,\mathbf{R}\,b$	1	1	1	1	1	1	0	1	1	1	1	0	0	0^ω

Notation: v^ω denotes the infinite repetition of v

Standard properties?

Two LTL formulas F and G are *equivalent*, written $F \equiv G$, if for every path π we have $\pi \models F$ iff $\pi \models G$

Standard properties?

Two LTL formulas F and G are *equivalent*, written $F \equiv G$, if for every path π we have $\pi \models F$ iff $\pi \models G$

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

Standard properties?

Two LTL formulas F and G are *equivalent*, written $F \equiv G$, if for every path π we have $\pi \models F$ iff $\pi \models G$

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

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

Meaning of Some Formulas

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathcal{U}$ (until)
$\mathcal{R}$ (release)

- $\Diamond \Box F$

Meaning of Some Formulas

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathbf{U}$ (until)
$\mathbf{R}$ (release)

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

Meaning of Some Formulas

$\Diamond$ (eventually)	$\bigcirc$ (next)
$\Box$ (always)	$\mathbf{U}$ (until)
$\mathbf{R}$ (release)	

- $\Diamond \Box F$
- $\Box (F \rightarrow \bigcirc F)$
- $\neg F \, \mathbf{U} \, \Box F$

Meaning of Some Formulas

$\Diamond$ (eventually)	$\bigcirc$ (next)
$\Box$ (always)	$\mathcal{U}$ (until)
$\mathcal{R}$ (release)	

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

Meaning of Some Formulas

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathbf{U}$ (until)
$\mathbf{R}$ (release)

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

Meaning of Some Formulas

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathcal{U}$ (until)
$\mathcal{R}$ (release)

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

Meaning of Some Formulas

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathcal{U}$ (until)
$\mathcal{R}$ (release)

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

Expressing Some Properties

$\Diamond$ (eventually)	$\bigcirc$ (next)
$\Box$ (always)	$\mathbf{U}$ (until)
$\mathbf{R}$ (release)	

1. F holds initially but not later
2. F never holds in two consecutive states
3. If F holds in a state s, it also holds in all states after s
4. F holds in at most one state
5. F holds in at least two states
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$ (eventually)	$\bigcirc$ (next)
$\Box$ (always)	$\mathbf{U}$ (until)
$\mathbf{R}$ (release)	

1. F holds initially but not later $F \land \bigcirc \Box \neg F$
2. F never holds in two consecutive states
3. If F holds in a state s, it also holds in all states after s
4. F holds in at most one state
5. F holds in at least two states
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

◊ (eventually) ○ (next)
□ (always) **U** (until)
R (release)

1. F holds initially but not later $F \land \bigcirc \Box \neg F$
2. F never holds in two consecutive states
3. If F holds in a state s, it also holds in all states after s
4. F holds in at most one state
5. F holds in at least two states
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathbf{U}$ (until)
$\mathbf{R}$ (release)

1. F holds initially but not later $F \land \bigcirc \Box \neg F$

2. F never holds in two consecutive states $\Box(F \rightarrow \bigcirc \neg F)$

3. If F holds in a state s, it also holds in all states after s

4. F holds in at most one state

5. F holds in at least two states

6. F happens infinitely often

7. F holds in each even state and does not hold in each odd state (states are counted from 0)

8. 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

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathbf{U}$ (until)
$\mathbf{R}$ (release)

1. F holds initially but not later $F \wedge \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \rightarrow \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s
4. F holds in at most one state
5. F holds in at least two states
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$	(eventually)	$\bigcirc$	(next)
$\Box$	(always)	$\mathcal{U}$	(until)
$\mathcal{R}$	(release)		

1. F holds initially but not later $F \wedge \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \rightarrow \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \rightarrow \Box F)$
4. F holds in at most one state
5. F holds in at least two states
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathbf{U}$ (until)
$\mathbf{R}$ (release)

1. F holds initially but not later $F \land \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \to \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \to \Box F)$
4. F holds in at most one state
5. F holds in at least two states
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathcal{U}$ (until)
$\mathcal{R}$ (release)

1. F holds initially but not later $F \wedge \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \rightarrow \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \rightarrow \Box F)$
4. F holds in at most one state $\Box(F \rightarrow \bigcirc \Box \neg F)$
5. F holds in at least two states
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$ (eventually)	$\bigcirc$ (next)
$\Box$ (always)	$\mathbf{U}$ (until)
$\mathbf{R}$ (release)	

1. F holds initially but not later $F \wedge \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \rightarrow \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \rightarrow \Box F)$
4. F holds in at most one state $\Box(F \rightarrow \bigcirc \Box \neg F)$
5. F holds in at least two states
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$	(eventually)	$\bigcirc$	(next)
$\Box$	(always)	$\mathcal{U}$	(until)
$\mathcal{R}$	(release)		

1. F holds initially but not later $F \land \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \to \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \to \Box F)$
4. F holds in at most one state $\Box(F \to \bigcirc \Box \neg F)$
5. F holds in at least two states $\Diamond(F \land \bigcirc \Diamond F)$
6. F happens infinitely often
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathbf{U}$ (until)
$\mathbf{R}$ (release)

1. F holds initially but not later $\quad F \land \bigcirc \Box \neg F$

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

3. If F holds in a state s, it also holds in all states after s $\quad \Box(F \to \Box F)$

4. F holds in at most one state $\quad \Box(F \to \bigcirc \Box \neg F)$

5. F holds in at least two states $\quad \Diamond(F \land \bigcirc \Diamond F)$

6. F happens infinitely often

7. F holds in each even state and does not hold in each odd state (states are counted from 0)

8. 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

$\Diamond$ (eventually)	$\bigcirc$ (next)
$\Box$ (always)	$\mathbf{U}$ (until)
$\mathbf{R}$ (release)	

1. F holds initially but not later $\quad F \land \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\quad \Box(F \to \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\quad \Box(F \to \Box F)$
4. F holds in at most one state $\quad \Box(F \to \bigcirc \Box \neg F)$
5. F holds in at least two states $\quad \Diamond(F \land \bigcirc \Diamond F)$
6. F happens infinitely often $\quad \Box \Diamond F$
7. F holds in each even state and does not hold in each odd state (states are counted from 0)
8. 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

$\Diamond$	(eventually)	$\bigcirc$	(next)
$\Box$	(always)	$\mathbf{U}$	(until)
$\mathbf{R}$	(release)		

1. F holds initially but not later $F \land \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \rightarrow \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \rightarrow \Box F)$
4. F holds in at most one state $\Box(F \rightarrow \bigcirc \Box \neg F)$
5. F holds in at least two states $\Diamond(F \land \bigcirc \Diamond F)$
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)
8. 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

$\Diamond$	(eventually)	$\bigcirc$	(next)
$\Box$	(always)	$\mathcal{U}$	(until)
$\mathcal{R}$	(release)		

1. F holds initially but not later $F \land \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \to \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \to \Box F)$
4. F holds in at most one state $\Box(F \to \bigcirc \Box \neg F)$
5. F holds in at least two states $\Diamond(F \land \bigcirc \Diamond F)$
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) $F \land \Box(F \leftrightarrow \bigcirc \neg F)$
8. 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

$\lozenge$ (eventually)	$\bigcirc$ (next)
$\square$ (always)	$\mathcal{U}$ (until)
$\mathcal{R}$ (release)	

1. F holds initially but not later $F \wedge \bigcirc \square \neg F$
2. F never holds in two consecutive states $\square(F \rightarrow \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\square(F \rightarrow \square F)$
4. F holds in at most one state $\square(F \rightarrow \bigcirc \square \neg F)$
5. F holds in at least two states $\lozenge(F \wedge \bigcirc \lozenge F)$
6. F happens infinitely often $\square \lozenge F$
7. F holds in each even state and does not hold in each odd state (states are counted from 0) $F \wedge \square(F \leftrightarrow \bigcirc \neg F)$
8. 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

$\Diamond$ (eventually) $\bigcirc$ (next)
$\Box$ (always) $\mathbf{U}$ (until)
$\mathbf{R}$ (release)

1. F holds initially but not later $F \wedge \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \rightarrow \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \rightarrow \Box F)$
4. F holds in at most one state $\Box(F \rightarrow \bigcirc \Box \neg F)$
5. F holds in at least two states $\Diamond(F \wedge \bigcirc \Diamond F)$
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) $F \wedge \Box(F \leftrightarrow \bigcirc \neg F)$
8. 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}
   $(\bigcirc F \rightarrow G) \wedge \Box(\bigcirc \bigcirc F \rightarrow \bigcirc G \vee G)$

Expressing Some Properties

$\Diamond$ (eventually)	$\bigcirc$ (next)
$\Box$ (always)	$\mathcal{U}$ (until)
$\mathcal{R}$ (release)	

1. F holds initially but not later $F \land \bigcirc \Box \neg F$
2. F never holds in two consecutive states $\Box(F \rightarrow \bigcirc \neg F)$
3. If F holds in a state s, it also holds in all states after s $\Box(F \rightarrow \Box F)$
4. F holds in at most one state $\Box(F \rightarrow \bigcirc \Box \neg F)$
5. F holds in at least two states $\Diamond(F \land \bigcirc \Diamond F)$
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) $F \land \Box(F \leftrightarrow \bigcirc \neg F)$
8. 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}
   $(\bigcirc F \rightarrow G) \land \Box(\bigcirc \bigcirc F \rightarrow \bigcirc G \lor G)$

Expressiveness of LTL

Not all *reasonable* properties are expressible in LTL

Example: p holds in all even states (and possibly in others)

Equivalences: Unwinding Properties

$\Diamond$	(eventually)	$\bigcirc$	(next)
$\Box$	(always)	$\mathbf{U}$	(until)
$\mathbf{R}$	(release)		

$$\Diamond F \equiv F \lor \bigcirc \Diamond F$$

$$\Box F \equiv F \land \bigcirc \Box F$$

$$F \,\mathbf{U}\, G \equiv G \lor (F \land \bigcirc (F \,\mathbf{U}\, G))$$

$$F \,\mathbf{R}\, G \equiv G \land (F \lor \bigcirc (F \,\mathbf{R}\, G))$$

Equivalences: Negation of Temporal Operators

$\Diamond$	(eventually)	$\bigcirc$	(next)
$\Box$	(always)	$\mathbf{U}$	(until)
$\mathbf{R}$	(release)		

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

$$\neg \Diamond F \equiv \Box \neg F$$

$$\neg \Box F \equiv \Diamond \neg F$$

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

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

Expressing Temporal Operators Using $\mathbf{U}$

$\Diamond$	(eventually)	$\bigcirc$	(next)
$\Box$	(always)	$\mathbf{U}$	(until)
$\mathbf{R}$	(release)		

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

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

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

Hence, all operators can be expressed using $\bigcirc$ and $\mathbf{U}$

Further Equivalences

$\Diamond$	(eventually)	$\bigcirc$	(next)
$\Box$	(always)	$\mathbf{U}$	(until)
$\mathbf{R}$	(release)		

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

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

But

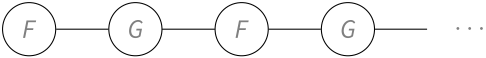
$$\Box(F \vee G) \not\equiv \Box F \vee \Box G$$

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

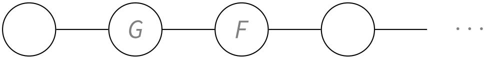
How to Show that Two Formulas are **not** Equivalent

Find a path that satisfies one of the formulas but not the other

Example 1: for $\Box(F \lor G)$ and $\Box F \lor \Box G$



Example 2: for $\Diamond(F \land G)$ and $\Diamond F \land \Diamond G$



Back to the Vending Machine

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

Talking about the vending machine in LTL, Examples

1. If the machine runs out of soda, it gets restocked immediately.
2. The machine eventually runs out of drinks.
3. The machine runs out of soda infinitely often.
4. Students never leave without a drink.
5. Professors sometimes leave a drink in the dispenser.
6. If students forget a coin in the coin slot, they (or other students) will use this coin to get a drink before any professor does the same.
7. If professors forget coins or their drink in the machine, a student will immediately arrive at the machine.
8. If there is a coin in the coin slot when a professor arrives, they will leave without getting a drink.
9. If a professor is currently at the machine, there will be no student at the machine for at least the next three transitions.
10. …

Transitions

1. *Restock* which results in the drink storage having both soda 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_soda*, when the customer presses the button to get a can of soda.
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 sodas in a row without inserting a coin.*
2. *If we never have two restock transitions in a row, then the next transition after a restock must be a customer arrival.*

Note that they are about transitions, not states

How can one represent these properties?

Introduce a state variable denoting the next transition

Reasoning About Transitions

Consider the following properties:

1. *One cannot have two sodas in a row without inserting a coin.*
2. *If we never have two restock transitions in a row, then the next transition after a restock must be a customer arrival.*

Note that they are about transitions, not states

How can one represent these properties?

Introduce a state variable denoting the next transition

Reasoning About Transitions

Consider the following properties:

1. *One cannot have two sodas in a row without inserting a coin.*
2. *If we never have two restock transitions in a row, then the next transition after a restock must be a customer arrival.*

Note that they are about transitions, not states

How can one represent these properties?

Introduce a state variable denoting the next transition

Reasoning About Transitions

Consider the following properties:

1. *One cannot have two sodas in a row without inserting a coin.*
2. *If we never have two restock transitions in a row, then the next transition after a restock must be a customer arrival.*

Note that they are about transitions, not states

How can one represent these properties?

Introduce a state variable denoting the next transition

Example

tr with domain $\{\, \mathit{restock}, \mathit{customer\_arrives}, \mathit{coin\_insert}, \ldots \,\}$

$$
\begin{aligned}
\mathit{Restock} \;&\stackrel{\text{def}}{=}\; \mathrm{tr} = \mathit{restock} \wedge \mathrm{customer} = \mathit{none} \wedge \\
&\quad \mathrm{st\_coffee}' \wedge \mathrm{st\_soda}' \wedge \\
&\quad \mathit{only}(\mathrm{st\_coffee}, \mathrm{st\_soda}, \mathrm{tr}) \\
\mathit{Customer\_arrives} \;&\stackrel{\text{def}}{=}\; \mathrm{tr} = \mathit{customer\_arrives} \wedge \mathrm{customer} = \mathit{none} \wedge \\
&\quad \mathrm{customer}' \neq \mathit{none} \wedge \\
&\quad \mathit{only}(\mathrm{customer}, \mathrm{tr}) \\
\mathit{Coin\_insert} \;&\stackrel{\text{def}}{=}\; \mathrm{tr} = \mathit{coin\_insert} \wedge \\
&\quad \mathrm{customer} \neq \mathit{none} \wedge \mathrm{coins} \neq \mathit{3} \wedge \\
&\quad (\mathrm{coins} = \mathit{0} \rightarrow \mathrm{coins}' = \mathit{1}) \wedge \\
&\quad (\mathrm{coins} = \mathit{1} \rightarrow \mathrm{coins}' = \mathit{2}) \wedge \\
&\quad (\mathrm{coins} = \mathit{2} \rightarrow \mathrm{coins}' = \mathit{3}) \wedge \\
&\quad \mathit{only}(\mathrm{coins}, \mathrm{tr}) \\
\ldots
\end{aligned}
$$

Representing Temporal Properties of Transitions

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

Representing Temporal Properties of Transitions

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

$$\square(\text{tr} = \textit{dispense\_soda} \rightarrow \bigcirc(\square(\text{tr} \neq \textit{dispense\_soda}) \vee (\text{tr} \neq \textit{dispense\_soda})\, \mathbf{U}\, (\text{tr} = \textit{insert\_coin})))$$

Representing Temporal Properties of Transitions

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

$$\Box(\text{tr} = \textit{dispense\_soda} \rightarrow \bigcirc(\Box(\text{tr} \neq \textit{dispense\_soda}) \vee (\mathbf{tr} \neq \textit{dispense\_soda})\,\mathbf{U}\,(\text{tr} = \textit{insert\_coin})))$$

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

Representing Temporal Properties of Transitions

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

$$\Box(\mathrm{tr} = \mathit{dispense\_soda} \rightarrow \bigcirc(\Box(\mathrm{tr} \neq \mathit{dispense\_soda}) \vee (\mathrm{tr} \neq \mathit{dispense\_soda})\, \mathbf{U}\, (\mathrm{tr} = \mathit{insert\_coin})))$$

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

$$\Box(\mathrm{tr} = \mathit{restock} \rightarrow \bigcirc \mathrm{tr} \neq \mathit{restock}) \rightarrow \Box(\mathrm{tr} = \mathit{restock} \rightarrow \bigcirc \mathrm{tr} = \mathit{customer\_arrives})$$

Representing Temporal Properties of Transitions

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

$$\Box(\mathrm{tr} = \mathit{dispense\_soda} \rightarrow \bigcirc(\Box(\mathrm{tr} \neq \mathit{dispense\_soda}) \vee (\mathrm{tr} \neq \mathit{dispense\_soda})\,\mathcal{U}\,(\mathrm{tr} = \mathit{insert\_coin})))$$

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

$$\Box(\mathrm{tr} = \mathit{restock} \rightarrow \bigcirc\mathrm{tr} \neq \mathit{restock}) \rightarrow \Box(\mathrm{tr} = \mathit{restock} \rightarrow \bigcirc\mathrm{tr} = \mathit{customer\_arrives})$$

3. The value of customer changes only as a result of either *Customer_arrives* or *Customer_leaves*:

Representing Temporal Properties of Transitions

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

$$\Box(\mathrm{tr} = \mathit{dispense\_soda} \rightarrow \bigcirc(\Box(\mathrm{tr} \neq \mathit{dispense\_soda}) \vee (\mathrm{tr} \neq \mathit{dispense\_soda})\,\mathcal{U}\,(\mathrm{tr} = \mathit{insert\_coin})))$$

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

$$\Box(\mathrm{tr} = \mathit{restock} \rightarrow \bigcirc\mathrm{tr} \neq \mathit{restock}) \rightarrow \Box(\mathrm{tr} = \mathit{restock} \rightarrow \bigcirc\mathrm{tr} = \mathit{customer\_arrives})$$

3. The value of customer changes only as a result of either *Customer_arrives* or *Customer_leaves*:

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

Representing Temporal Properties of Transitions

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

Representing Temporal Properties of Transitions

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

$$\bigwedge_{v \in dom(\mathrm{coin})} \Box(\mathrm{coin} = v \wedge \mathrm{tr} = \mathit{coin\_insert} \wedge \bigcirc \mathrm{tr} = \mathit{coin\_insert} \wedge \bigcirc\bigcirc \mathrm{tr} = \mathit{dispense\_soda} \rightarrow \bigcirc\bigcirc\bigcirc \mathrm{coin} = v)$$

Representing Temporal Properties of Transitions

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

$$\bigwedge_{v \in dom(\mathrm{coin})} \Box(\mathrm{coin} = v \wedge \\ \mathrm{tr} = coin\_insert \wedge \\ \bigcirc \mathrm{tr} = coin\_insert \wedge \\ \bigcirc\bigcirc \mathrm{tr} = dispense\_soda \rightarrow \\ \bigcirc\bigcirc\bigcirc \mathrm{coin} = v)$$

2. If the system is occasionally restocked, then after each *dispense_soda* the customer will leave:

Representing Temporal Properties of Transitions

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

$$\bigwedge_{v \in dom(\mathrm{coin})} \Box(\mathrm{coin} = v \wedge \mathrm{tr} = \mathit{coin\_insert} \wedge \bigcirc \mathrm{tr} = \mathit{coin\_insert} \wedge \bigcirc\bigcirc \mathrm{tr} = \mathit{dispense\_soda} \rightarrow \bigcirc\bigcirc\bigcirc \mathrm{coin} = v)$$

2. If the system is occasionally restocked, then after each *dispense_soda* the customer will leave:

$$\Box\Diamond \mathrm{tr} = \mathit{restock} \rightarrow \Box(\mathrm{tr} = \mathit{dispense\_soda} \rightarrow \Diamond \mathrm{tr} = \mathit{customer\_leaves})$$

Exercise, Dimmable Lamp

Device A lamp with two buttons that can be

- off
- on at medium intensity
- on at full intensity

Actions

1. pushing the first button (set): switches light from off to medium intensity or from medium to full intensity
2. pushing the second button (reset): switches light off
3. doing nothing (none): results just in time passing

Constraints

1. Pushing the first button has no effect if done immediately after a reset
2. Pushing the second button has no effect if done immediately after a set
3. It is impossible to push both buttons at the same time

Exercise, Dimmable Lamp

Device A lamp with two buttons that can be

- off
- on at medium intensity
- on at full intensity

state variable	domain	explanation
a	{ *set*, *reset*, *none* }	actions/transitions
s	{ *off*, *on1*, *on2* }	lamp status
st	{ 0, 1 }	time counter for set
rt	{ 0, 1 }	time counter for reset

Actions

1. pushing the first button (set): switches light from off to medium intensity or from medium to full intensity
2. pushing the second button (reset): switches light off
3. doing nothing (none): results just in time passing

Constraints

1. Pushing the first button has no effect if done immediately after a reset
2. Pushing the second button has no effect if done immediately after a set
3. It is impossible to push both buttons at the same time

Exercise, Modeling device as a transition system

Initial state formula

$$\text{s} = \textit{off} \wedge \text{st} = \textit{1} \wedge \text{rt} = \textit{1}$$

Transition formulas

Exercise, Modeling device as a transition system

Initial state formula

$$\mathrm{s} = \mathit{off} \wedge \mathrm{st} = \mathit{1} \wedge \mathrm{rt} = \mathit{1}$$

Transition formulas

$$\begin{aligned} \mathit{Set} \quad \stackrel{\mathrm{def}}{=} \quad & \mathrm{a} = \mathit{set} \wedge \mathrm{rt} = \mathit{1} \wedge \\ & (\mathrm{s} = \mathit{off} \wedge \mathrm{s}' = \mathit{on1} \ \vee \ \mathrm{s} \neq \mathit{off} \wedge \mathrm{s}' = \mathit{on2}) \wedge \\ & \mathrm{st}' = \mathit{0} \wedge \mathit{only}(\mathrm{s}, \mathrm{st}, \mathrm{a}) \end{aligned}$$

Exercise, Modeling device as a transition system

Initial state formula

$$\mathsf{s} = \mathit{off} \wedge \mathsf{st} = \mathit{1} \wedge \mathsf{rt} = \mathit{1}$$

Transition formulas

$$\begin{aligned}
\mathit{Set} \stackrel{\text{def}}{=}\ & \mathsf{a} = \mathit{set} \wedge \mathsf{rt} = \mathit{1} \wedge \\
& (\mathsf{s} = \mathit{off} \wedge \mathsf{s}' = \mathit{on1} \ \vee\ \mathsf{s} \neq \mathit{off} \wedge \mathsf{s}' = \mathit{on2}) \wedge \\
& \mathsf{st}' = \mathit{0} \wedge \mathit{only}(\mathsf{s}, \mathsf{st}, \mathsf{a})
\end{aligned}$$

$$\begin{aligned}
\mathit{Reset} \stackrel{\text{def}}{=}\ & \mathsf{a} = \mathit{reset} \wedge \mathsf{st} = \mathit{1} \wedge \\
& \mathsf{s}' = \mathit{off} \wedge \mathsf{rt}' = \mathit{0} \wedge \mathit{only}(\mathsf{s}, \mathsf{rt}, \mathsf{a})
\end{aligned}$$

Exercise, Modeling device as a transition system

Initial state formula

$$\mathrm{s} = \textit{off} \wedge \mathrm{st} = \textit{1} \wedge \mathrm{rt} = \textit{1}$$

Transition formulas

$$\begin{aligned}
\textit{Set} \;&\stackrel{\text{def}}{=}\; \mathrm{a} = \textit{set} \wedge \mathrm{rt} = \textit{1} \wedge \\
&\quad (\mathrm{s} = \textit{off} \wedge \mathrm{s}' = \textit{on1} \;\vee\; \mathrm{s} \neq \textit{off} \wedge \mathrm{s}' = \textit{on2}) \wedge \\
&\quad \mathrm{st}' = \textit{0} \wedge \textit{only}(\mathrm{s}, \mathrm{st}, \mathrm{a})
\end{aligned}$$

$$\begin{aligned}
\textit{Reset} \;&\stackrel{\text{def}}{=}\; \mathrm{a} = \textit{reset} \wedge \mathrm{st} = \textit{1} \wedge \\
&\quad \mathrm{s}' = \textit{off} \wedge \mathrm{rt}' = \textit{0} \wedge \textit{only}(\mathrm{s}, \mathrm{rt}, \mathrm{a})
\end{aligned}$$

$$\begin{aligned}
\textit{None} \;&\stackrel{\text{def}}{=}\; \mathrm{a} = \textit{none} \wedge \\
&\quad \mathrm{st}' = \textit{1} \wedge \mathrm{rt}' = \textit{1} \wedge \textit{only}(\mathrm{st}, \mathrm{rt}, \mathrm{a})
\end{aligned}$$

Exercise, Temporal properties about the lamp

1. The lamp is initially off.
2. Resetting when the lamp is on turns it off.
3. Resetting always turns the lamp off.
4. Setting when the lamp is off turns it on.
5. Setting when the lamp is half-on turns it fully on.
6. A reset cannot immediately follow a set and vice versa.
7. Setting when the lamp is fully on has no effect on the light.
8. The lamp is initially off and stays off until the first set.
9. Once off, the lamp stays off until the next set.
10. Two consecutive set actions are enough to turn the lamp fully on.
11. If the lamp is on at any point, it must have been turned on some time before.
12. If the lamp is on, it will eventually be off.
13. The lamp will be on repeatedly.
14. At some point the lamp will burn and stay permanently off.
15. If set occurs infinitely often the lamp will be on infinitely often.

Exercise, formalization of properties

1. $s = \mathit{off}$
2. $\Box(a = \mathit{reset} \land s \neq \mathit{off} \rightarrow \bigcirc s = \mathit{off})$
3. $\Box(a = \mathit{reset} \rightarrow \bigcirc s = \mathit{off})$
4. $\Box(a = \mathit{set} \land s = \mathit{off} \rightarrow \bigcirc s \neq \mathit{off})$
5. $\Box(a = \mathit{set} \land s = \mathit{on1} \rightarrow \bigcirc s = \mathit{on2})$
6. $\Box(a = \mathit{set} \rightarrow \bigcirc a \neq \mathit{reset}) \land \Box(a = \mathit{reset} \rightarrow \bigcirc a \neq \mathit{set})$
7. $\Box(a = \mathit{set} \land s = \mathit{on2} \rightarrow \bigcirc s = \mathit{on2})$
8. $a = \mathit{set}\ \mathbf{R}\ s = \mathit{off}$
9. $\Box(s = \mathit{off} \rightarrow a = \mathit{set}\ \mathbf{R}\ s = \mathit{off})$
10. $\Box(a = \mathit{set} \land \bigcirc a = \mathit{set} \rightarrow \bigcirc\bigcirc s = \mathit{on2})$, also $\Box(a = \mathit{set} \rightarrow \bigcirc(a = \mathit{set} \rightarrow \bigcirc s = \mathit{on2}))$
11. $\neg(a \neq \mathit{set}\ \mathbf{U}\ s \neq \mathit{off})$
12. $\Box(s \neq \mathit{off} \rightarrow \Diamond s = \mathit{off})$
13. $\Box(\Diamond s \neq \mathit{off})$
14. $\Diamond(\Box s = \mathit{off})$
15. $\Box\Diamond a \neq \mathit{set} \rightarrow \Box\Diamond s \neq \mathit{off}$

Exercise, formalization of properties

1. $s = \mathit{off}$
2. $\Box(a = \mathit{reset} \wedge s \neq \mathit{off} \rightarrow \bigcirc s = \mathit{off})$
3. $\Box(a = \mathit{reset} \rightarrow \bigcirc s = \mathit{off})$
4. $\Box(a = \mathit{set} \wedge s = \mathit{off} \rightarrow \bigcirc s \neq \mathit{off})$
5. $\Box(a = \mathit{set} \wedge s = \mathit{on1} \rightarrow \bigcirc s = \mathit{on2})$
6. $\Box(a = \mathit{set} \rightarrow \bigcirc a \neq \mathit{reset}) \wedge \Box(a = \mathit{reset} \rightarrow \bigcirc a \neq \mathit{set})$
7. $\Box(a = \mathit{set} \wedge s = \mathit{on2} \rightarrow \bigcirc s = \mathit{on2})$
8. $a = \mathit{set} \, \mathbf{R} \, s = \mathit{off}$
9. $\Box(s = \mathit{off} \rightarrow a = \mathit{set} \, \mathbf{R} \, s = \mathit{off})$
10. $\Box(a = \mathit{set} \wedge \bigcirc a = \mathit{set} \rightarrow \bigcirc\bigcirc s = \mathit{on2})$, also
    $\Box(a = \mathit{set} \rightarrow \bigcirc(a = \mathit{set} \rightarrow \bigcirc s = \mathit{on2}))$
11. $\neg(a \neq \mathit{set} \, \mathbf{U} \, s \neq \mathit{off})$
12. $\Box(s \neq \mathit{off} \rightarrow \Diamond s = \mathit{off})$
13. $\Box(\Diamond s \neq \mathit{off})$
14. $\Diamond(\Box s = \mathit{off})$
15. $\Box\Diamond a \neq \mathit{set} \rightarrow \Box\Diamond s \neq \mathit{off}$

Which of these properties are satisfied by every execution path of the transition system?