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.

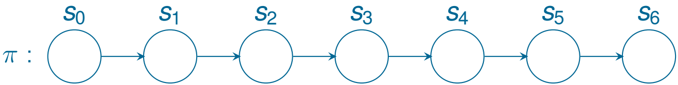
We define the notion F is true on π (or F holds on π), denoted by $\pi \models F$, by induction on F as follows.

For all $i = 0, 1, \ldots$ denote by π_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 π_i.

Semantics

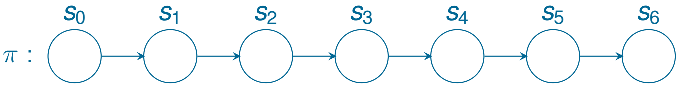
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and F be an LTL formula.



We define the notion F is true on π (or F holds on π), denoted by $\pi \models F$, by induction on F as follows.
For all $i = 0, 1, \ldots$ denote by π_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 π_i.

Semantics

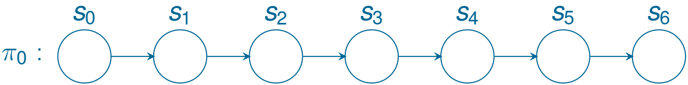
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and F be an LTL formula.



We define the notion F is true on π (or F holds on π), denoted by $\pi \models F$, by induction on F as follows.
For all $i = 0, 1, \ldots$ denote by π_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 π_i.

Semantics

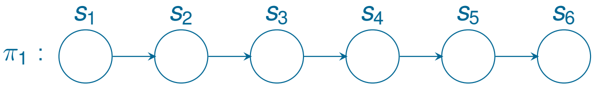
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and F be an LTL formula.



We define the notion F is true on π (or F holds on π), denoted by $\pi \models F$, by induction on F as follows.
For all $i = 0, 1, \ldots$ denote by π_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 π_i.

Semantics

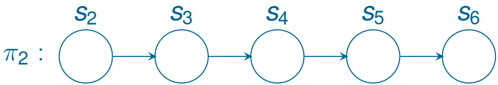
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and F be an LTL formula.



We define the notion F is true on π (or F holds on π), denoted by $\pi \models F$, by induction on F as follows.
For all $i = 0, 1, \ldots$ denote by π_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 π_i.

Semantics

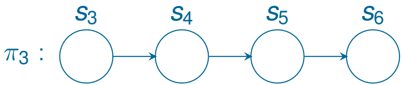
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and F be an LTL formula.



We define the notion F is true on π (or F holds on π), denoted by $\pi \models F$, by induction on F as follows.
For all $i = 0, 1, \ldots$ denote by π_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 π_i.

Semantics

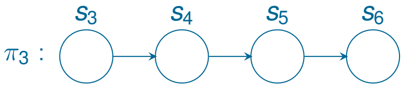
Unlike propositonal formulas, LTL formulas express properties of computations or computation paths.
Let $\pi = s_0, s_1, s_2 \ldots$ be a sequence of states and F be an LTL formula.



We define the notion F is true on π (or F holds on π), denoted by $\pi \models F$, by induction on F as follows.
For all $i = 0, 1, \ldots$ denote by π_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 π_i.

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 π (or F holds on π), denoted by $\pi \models F$, by induction on F as follows.
For all $i = 0, 1, \ldots$ denote by π_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 π_i.

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 π is the same as in propositional logic where all subformulas are also evaluated on π.

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

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 π is the same as in propositional logic where all subformulas are also evaluated on π.

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

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 π is the same as in propositional logic where all subformulas are also evaluated on π.

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

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 π is the same as in propositional logic where all subformulas are also evaluated on π.

1. $\pi \models \top$ and $\pi \not\models \bot$.
2. $\pi \models x = v$ if $s_0 \models x = v$.
3. $\pi \models F_1 \wedge \ldots \wedge F_n$ if for all $j = 1, \ldots, n$ we have $\pi \models F_j$;
   $\pi \models F_1 \vee \ldots \vee F_n$ 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, formally

The semantics of propositional connectives is standard.

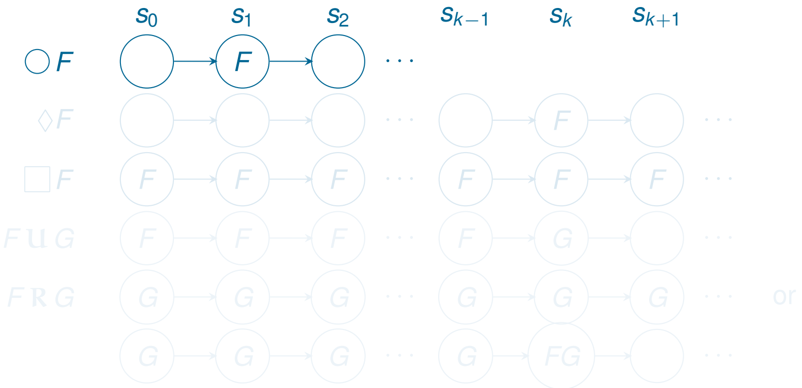
Atomic formulas are true iff they are true in s_0.

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

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. $\pi \models F \rightarrow G$ if either $\pi \not\models F$ or $\pi \models G$;
   $\pi \models F \leftrightarrow G$ if either both $\pi \not\models F$ and $\pi \not\models G$ or both $\pi \models F$ and $\pi \models G$.

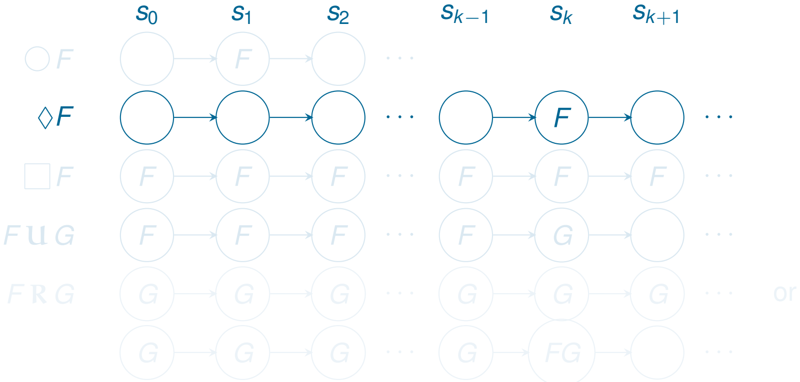
Semantics of temporal operators

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. $\pi \models F \mathbf{U} G$ if for some $k \geq 0$ we have $\pi_k \models G$ and $\pi_0 \models F, \ldots, \pi_{k-1} \models F$;
   $\pi \models F \mathbf{R} G$ if for all $k \geq 0$, either $\pi_k \models G$ or there exists $j < k$ such that $\pi_j \models F$.



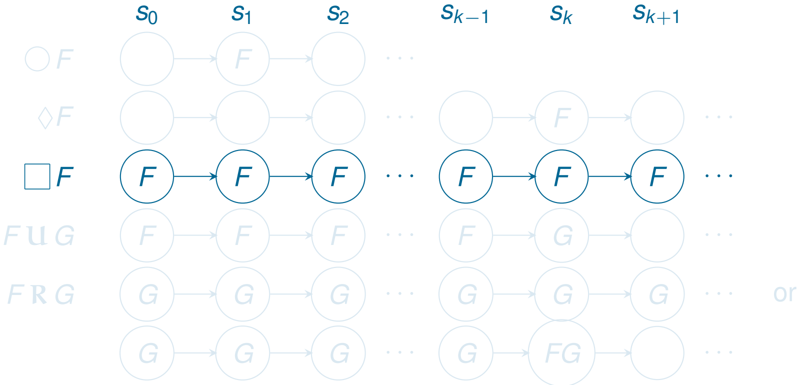
Semantics of temporal operators

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 $\pi_j \models F$.



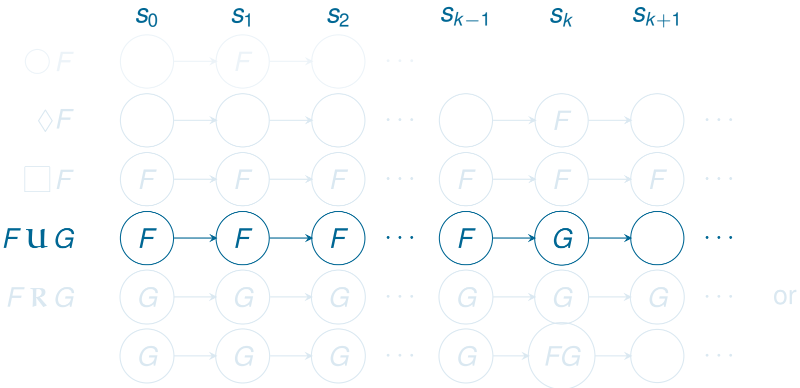
Semantics of temporal operators

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 $\pi_j \models F$.



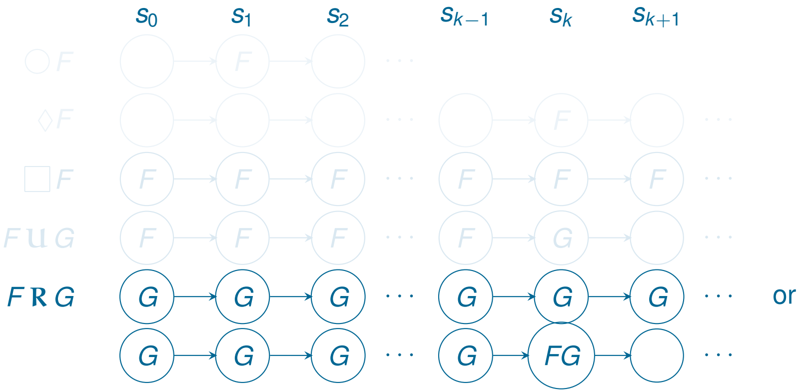
Semantics of temporal operators

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. $\pi \models F \mathbf{U} G$ if for some $k \geq 0$ we have $\pi_k \models G$ and $\pi_0 \models F, \ldots, \pi_{k-1} \models F$;
   $\pi \models F \mathbf{R} G$ if for all $k \geq 0$, either $\pi_k \models G$ or there exists $j < k$ such that $\pi_j \models F$.



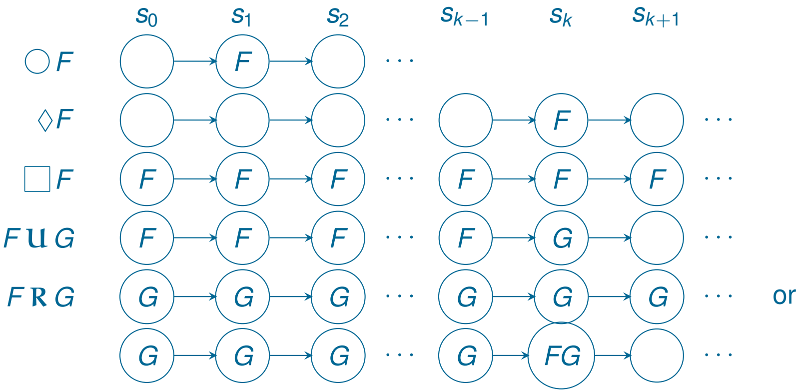
Semantics of temporal operators

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 $\pi_j \models F$.



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. $\pi \models F \mathbf{U} G$ if for some $k \geq 0$ we have $\pi_k \models G$ and $\pi_0 \models F, \ldots, \pi_{k-1} \models F$;
   $\pi \models F \mathbf{R} G$ if for all $k \geq 0$, either $\pi_k \models G$ or there exists $j < k$ such that $\pi_j \models F$.



Standard properties???

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

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

Standard properties???

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

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

Standard properties???

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

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?
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
$\mathcal{U}, \mathcal{R}$	4
$\wedge, \vee$	3
$\rightarrow$	2
$\leftrightarrow$	1

Expressing Some Properties

1. F never holds in two consecutive states.
2. If F holds in a state s, it also holds at all states after s.
3. F holds in most one state.
4. F holds in least two states.
5. 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}.
6. F happens infinitely often.

Expressing Some Properties

1. F never holds in two consecutive states.
2. If F holds in a state s, it also holds at all states after s.
3. F holds in most one state.
4. F holds in least two states.
5. 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}.
6. F happens infinitely often.

Expressing Some Properties

1. F never holds in two consecutive states.
2. If F holds in a state s, it also holds at all states after s.
3. F holds in most one state.
4. F holds in least two states.
5. 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}.
6. F happens infinitely often.

Expressing Some Properties

1. F never holds in two consecutive states.
2. If F holds in a state s, it also holds at all states after s.
3. F holds in most one state.
4. F holds in least two states.
5. 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}.
6. F happens infinitely often.

Expressing Some Properties

1. F never holds in two consecutive states.
2. If F holds in a state s, it also holds at all states after s.
3. F holds in most one state.
4. F holds in least two states.
5. 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}.
6. F happens infinitely often.

Expressing Some Properties

1. F never holds in two consecutive states.
2. If F holds in a state s, it also holds at all states after s.
3. F holds in most one state.
4. F holds in least two states.
5. 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}.
6. F happens infinitely often.

Meaning of Some Formulas

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

Meaning of Some Formulas

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

Meaning of Some Formulas

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

Meaning of Some Formulas

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

Meaning of Some Formulas

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

Meaning of Some Formulas

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

Meaning of Some Formulas

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

Equivalences: Unwinding Properties

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

Equivalences: Negation of Temporal Operators

$$
\begin{aligned}
\neg \bigcirc F &\equiv \bigcirc \neg F \\
\neg \Diamond F &\equiv \Box \neg F \\
\neg \Box F &\equiv \Diamond \neg F \\
\neg (F \, \mathcal{U} \, G) &\equiv \neg F \, \mathcal{R} \, \neg G \\
\neg (F \, \mathcal{R} \, G) &\equiv \neg F \, \mathcal{U} \, \neg G
\end{aligned}
$$

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 $\Box(F \lor G)$ and $\Box F \lor \Box G$.