CS228 : Logic for Computer Science

Module 5: Temporal Logic

Lecture 24: LTL Equivalences, and on to CTL

Instructor: S. Akshay

IIT Bombay, India

Recap: LTL syntax

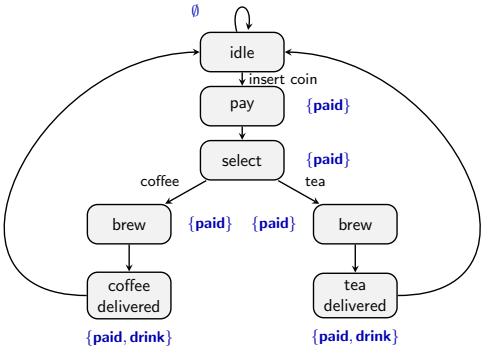
$$\varphi ::= \top \mid \bot \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$$
$$\mid (X\varphi) \mid (F\varphi) \mid (G\varphi) \mid (\varphi U \varphi)$$

Recap: LTL syntax

$$\varphi ::= \top \mid \bot \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$$
$$\mid (X\varphi) \mid (F\varphi) \mid (G\varphi) \mid (\varphi U \varphi)$$

Examples (atomic propositions **paid** and **drink**):

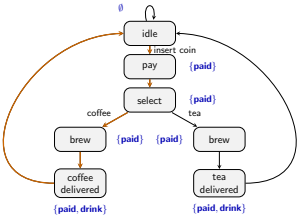
- "If a drink is delivered now, the user has to pay again in the *next* state."
  drink $\rightarrow X \neg$ **paid**
- "The machine *eventually* delivers a drink."
  F **drink**
- "Whenever a drink is delivered, it has been paid for."
  G (**drink** $\rightarrow$ **paid**)
- "No drink is delivered *until* the user has paid."
  $\neg$**drink** U **paid**



Recap: LTL semantics

- A path in a model $M = (S, \to, L)$ is an infinite sequence $\pi = s_1 \to s_2 \to \ldots$ of states with $s_i \to s_{i+1}$ for all $i \geq 1$; π^i is the suffix of π starting at s_i.

A path in this model: $\pi =$ idle, pay, select, brew, coffee delivered, idle, ...

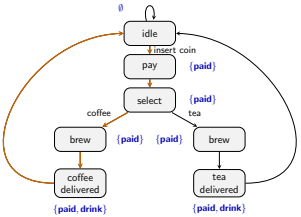


Recap: LTL semantics

- A path in a model $M = (S, \to, L)$ is an infinite sequence $\pi = s_1 \to s_2 \to \ldots$ of states with $s_i \to s_{i+1}$ for all $i \geq 1$; π^i is the suffix of π starting at s_i.
- $\pi \models \varphi$ is defined by induction on φ:

1. $\pi \models \top, \pi \not\models \bot$
2. $\pi \models p$ iff $p \in L(s_1)$
3. $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$; $\wedge, \vee, \to$ are as usual!
4. $\pi \models X\ \varphi$ iff $\pi^2 \models \varphi$.
5. $\pi \models F\ \varphi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \varphi$.
6. $\pi \models G\ \varphi$ iff $\forall i \geq 1$, $\pi^i \models \varphi$.
7. $\pi \models \varphi\ U\ \psi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \psi$ and $\forall j \in \{1, \ldots, i-1\}$, $\pi^j \models \varphi$.

A path in this model: $\pi =$ idle, pay, select, brew, coffee delivered, idle, ...



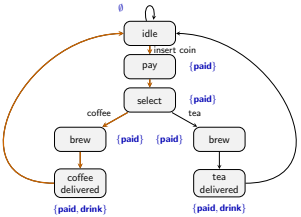
Recap: LTL semantics

- A path in a model $M = (S, \to, L)$ is an infinite sequence $\pi = s_1 \to s_2 \to \ldots$ of states with $s_i \to s_{i+1}$ for all $i \geq 1$; π^i is the suffix of π starting at s_i.
- $\pi \models \varphi$ is defined by induction on φ:

1. $\pi \models \top, \pi \not\models \bot$
2. $\pi \models p$ iff $p \in L(s_1)$
3. $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$; $\wedge, \vee, \to$ are as usual!
4. $\pi \models X\ \varphi$ iff $\pi^2 \models \varphi$.
5. $\pi \models F\ \varphi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \varphi$.
6. $\pi \models G\ \varphi$ iff $\forall i \geq 1$, $\pi^i \models \varphi$.
7. $\pi \models \varphi\ U\ \psi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \psi$ and $\forall j \in \{1, \ldots, i-1\}$, $\pi^j \models \varphi$.

Exercise 24.1 Does the orange path π satisfy

1. $\neg$**drink** U **paid**?
2. F **drink**?



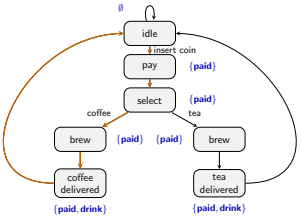
Recap: LTL semantics

- A path in a model $M = (S, \to, L)$ is an infinite sequence $\pi = s_1 \to s_2 \to \ldots$ of states with $s_i \to s_{i+1}$ for all $i \geq 1$; π^i is the suffix of π starting at s_i.
- $\pi \models \varphi$ is defined by induction on φ:

1. $\pi \models \top, \pi \not\models \bot$
2. $\pi \models p$ iff $p \in L(s_1)$
3. $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$; $\wedge, \vee, \to$ are as usual!
4. $\pi \models X\ \varphi$ iff $\pi^2 \models \varphi$.
5. $\pi \models F\ \varphi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \varphi$.
6. $\pi \models G\ \varphi$ iff $\forall i \geq 1$, $\pi^i \models \varphi$.
7. $\pi \models \varphi\ U\ \psi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \psi$ and $\forall j \in \{1, \ldots, i-1\}$, $\pi^j \models \varphi$.

Exercise 24.1 Does the orange path π satisfy

1. $\neg$**drink** U **paid**? Yes: **paid** holds at $\pi^2 =$ pay, and **drink** fails at $\pi^1 =$ idle.
2. F **drink**?



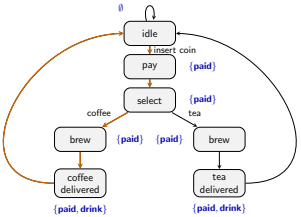
Recap: LTL semantics

- A path in a model $M = (S, \to, L)$ is an infinite sequence $\pi = s_1 \to s_2 \to \ldots$ of states with $s_i \to s_{i+1}$ for all $i \geq 1$; π^i is the suffix of π starting at s_i.
- $\pi \models \varphi$ is defined by induction on φ:

1. $\pi \models \top, \pi \not\models \bot$
2. $\pi \models p$ iff $p \in L(s_1)$
3. $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$; $\wedge, \vee, \to$ are as usual!
4. $\pi \models X\ \varphi$ iff $\pi^2 \models \varphi$.
5. $\pi \models F\ \varphi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \varphi$.
6. $\pi \models G\ \varphi$ iff $\forall i \geq 1$, $\pi^i \models \varphi$.
7. $\pi \models \varphi\ U\ \psi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \psi$ and $\forall j \in \{1, \ldots, i-1\}$, $\pi^j \models \varphi$.

Exercise 24.1 Does the orange path π satisfy

1. $\neg$**drink** U **paid**? Yes: **paid** holds at π^2 = pay, and **drink** fails at π^1 = idle.
2. F **drink**? Yes: **drink** holds at π^5 = coffee delivered.

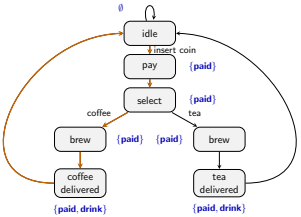


Recap: LTL semantics

- A path in a model $M = (S, \to, L)$ is an infinite sequence $\pi = s_1 \to s_2 \to \ldots$ of states with $s_i \to s_{i+1}$ for all $i \geq 1$; π^i is the suffix of π starting at s_i.
- $\pi \models \varphi$ is defined by induction on φ:

1. $\pi \models \top, \pi \not\models \bot$
2. $\pi \models p$ iff $p \in L(s_1)$
3. $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$; $\wedge, \vee, \to$ are as usual!
4. $\pi \models X\ \varphi$ iff $\pi^2 \models \varphi$.
5. $\pi \models F\ \varphi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \varphi$.
6. $\pi \models G\ \varphi$ iff $\forall i \geq 1$, $\pi^i \models \varphi$.
7. $\pi \models \varphi\ U\ \psi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \psi$ and $\forall j \in \{1, \ldots, i-1\}$, $\pi^j \models \varphi$.

For a model M and a state s: $M, s \models \varphi$ if *every* path π of M starting from s satisfies φ.



Recap: LTL semantics

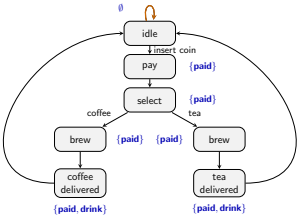
- A path in a model $M = (S, \to, L)$ is an infinite sequence $\pi = s_1 \to s_2 \to \ldots$ of states with $s_i \to s_{i+1}$ for all $i \geq 1$; π^i is the suffix of π starting at s_i.
- $\pi \models \varphi$ is defined by induction on φ:

1. $\pi \models \top, \pi \not\models \bot$
2. $\pi \models p$ iff $p \in L(s_1)$
3. $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$; $\wedge, \vee, \to$ are as usual!
4. $\pi \models X\ \varphi$ iff $\pi^2 \models \varphi$.
5. $\pi \models F\ \varphi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \varphi$.
6. $\pi \models G\ \varphi$ iff $\forall i \geq 1$, $\pi^i \models \varphi$.
7. $\pi \models \varphi\ U\ \psi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \psi$ and $\forall j \in \{1, \ldots, i-1\}$, $\pi^j \models \varphi$.

For a model M and a state s: $M, s \models \varphi$ if *every* path π of M starting from s satisfies φ.

Exercise 24.1 (continued)

3. In the vending machine, does $M, idle \models F$ **drink** hold?



Recap: LTL semantics

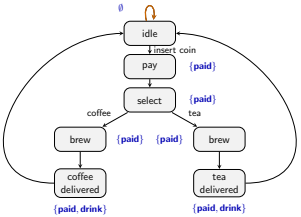
- A path in a model $M = (S, \to, L)$ is an infinite sequence $\pi = s_1 \to s_2 \to \ldots$ of states with $s_i \to s_{i+1}$ for all $i \geq 1$; π^i is the suffix of π starting at s_i.
- $\pi \models \varphi$ is defined by induction on φ:

1. $\pi \models \top, \pi \not\models \bot$
2. $\pi \models p$ iff $p \in L(s_1)$
3. $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$; $\wedge, \vee, \to$ are as usual!
4. $\pi \models X\ \varphi$ iff $\pi^2 \models \varphi$.
5. $\pi \models F\ \varphi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \varphi$.
6. $\pi \models G\ \varphi$ iff $\forall i \geq 1$, $\pi^i \models \varphi$.
7. $\pi \models \varphi\ U\ \psi$ iff $\exists i \geq 1$, s.t. $\pi^i \models \psi$ and $\forall j \in \{1, \ldots, i-1\}$, $\pi^j \models \varphi$.

For a model M and a state s: $M, s \models \varphi$ if *every* path π of M starting from s satisfies φ.

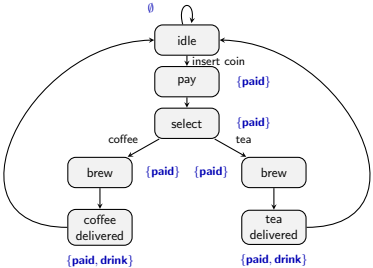
Exercise 24.1 (continued)

3. In the vending machine, does $M, idle \models F$ **drink** hold? No: the path idle, idle, ... never delivers a drink.



Two ways to say the same thing?

"A drink is never delivered unless it is paid for."



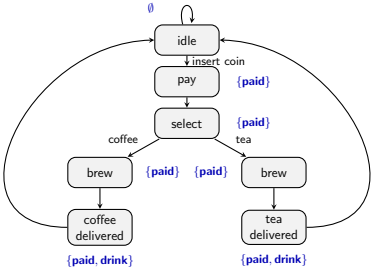
Two ways to say the same thing?

"A drink is never delivered unless it is paid for."

Two ways to write it:

(a) $G(\textbf{drink} \rightarrow \textbf{paid})$

(b) $\neg F(\textbf{drink} \wedge \neg\textbf{paid})$



Two ways to say the same thing?

"A drink is never delivered unless it is paid for."

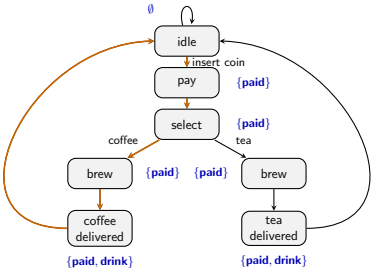
Two ways to write it:

(a) $G(\textbf{drink} \rightarrow \textbf{paid})$

(b) $\neg F(\textbf{drink} \wedge \neg \textbf{paid})$

Exercise 24.2 Do (a) and (b) agree (both true or both false) on

1. the orange path π?
2. the path idle, idle, . . . ?
3. a path of a faulty machine that delivers a free drink (a state with **drink** but not **paid**)?



Two ways to say the same thing?

"A drink is never delivered unless it is paid for."

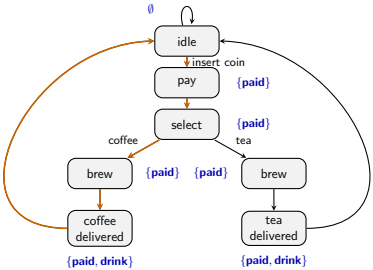
Two ways to write it:

(a) $G(\textbf{drink} \rightarrow \textbf{paid})$

(b) $\neg F(\textbf{drink} \wedge \neg \textbf{paid})$

Exercise 24.2 Do (a) and (b) agree (both true or both false) on

1. the orange path π? Both true: no state has **drink** without **paid**.
2. the path idle, idle, . . . ?
3. a path of a faulty machine that delivers a free drink (a state with **drink** but not **paid**)?



Two ways to say the same thing?

"A drink is never delivered unless it is paid for."

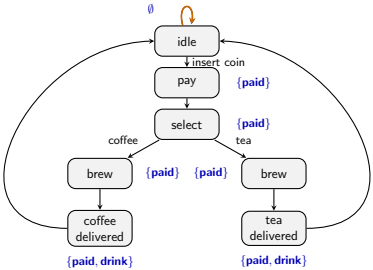
Two ways to write it:

(a) $G(\textbf{drink} \rightarrow \textbf{paid})$

(b) $\neg F(\textbf{drink} \wedge \neg\textbf{paid})$

Exercise 24.2 Do (a) and (b) agree (both true or both false) on

1. the orange path π? Both true: no state has **drink** without **paid**.
2. the path idle, idle, ...? Both true: no state has **drink**.
3. a path of a faulty machine that delivers a free drink (a state with **drink** but not **paid**)?



Two ways to say the same thing?

"A drink is never delivered unless it is paid for."

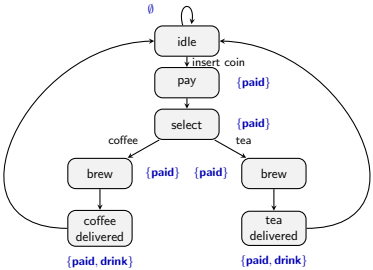
Two ways to write it:

(a) $G(\textbf{drink} \rightarrow \textbf{paid})$

(b) $\neg F(\textbf{drink} \wedge \neg \textbf{paid})$

Exercise 24.2 Do (a) and (b) agree (both true or both false) on

1. the orange path π? Both true: no state has **drink** without **paid**.
2. the path idle, idle, . . . ? Both true: no state has **drink**.
3. a path of a faulty machine that delivers a free drink (a state with **drink** but not **paid**)? Both false: (a) fails at that state, and $F(\textbf{drink} \wedge \neg \textbf{paid})$ holds, so (b) fails too.



Two ways to say the same thing?

"A drink is never delivered unless it is paid for."

Two ways to write it:

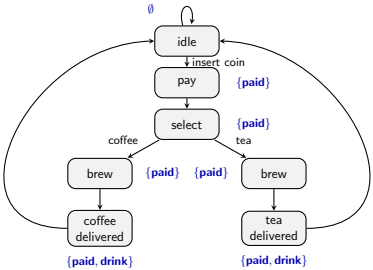
(a) $G(\textbf{drink} \rightarrow \textbf{paid})$

(b) $\neg F(\textbf{drink} \wedge \neg\textbf{paid})$

Exercise 24.2 Do (a) and (b) agree (both true or both false) on

1. the orange path π? Both true: no state has **drink** without **paid**.
2. the path idle, idle, ...? Both true: no state has **drink**.
3. a path of a faulty machine that delivers a free drink (a state with **drink** but not **paid**)? Both false: (a) fails at that state, and $F(\textbf{drink} \wedge \neg\textbf{paid})$ holds, so (b) fails too.

Same answer on every path of every model? That is what we call *equivalence*.



Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

- As in prop logic, if two formulae φ, ψ are equivalent and if φ occurs as a subformula of some formula χ, then in χ we can replace φ by ψ without changing its meaning.

Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

- As in prop logic, if two formulae φ, ψ are equivalent and if φ occurs as a subformula of some formula χ, then in χ we can replace φ by ψ without changing its meaning.
- All propositional equivalences carry over to LTL, e.g., $\neg\neg\varphi \equiv \varphi$ and $\varphi \to \psi \equiv \neg\varphi \vee \psi$.

Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

- As in prop logic, if two formulae φ, ψ are equivalent and if φ occurs as a subformula of some formula χ, then in χ we can replace φ by ψ without changing its meaning.
- All propositional equivalences carry over to LTL, e.g., $\neg\neg\varphi \equiv \varphi$ and $\varphi \to \psi \equiv \neg\varphi \vee \psi$. **Why?**

Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

- As in prop logic, if two formulae φ, ψ are equivalent and if φ occurs as a subformula of some formula χ, then in χ we can replace φ by ψ without changing its meaning.
- All propositional equivalences carry over to LTL, e.g., $\neg\neg\varphi \equiv \varphi$ and $\varphi \to \psi \equiv \neg\varphi \vee \psi$. **Why?** On a path, $\neg, \wedge, \vee, \to$ mean exactly what they mean in propositional logic.

Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

- As in prop logic, if two formulae φ, ψ are equivalent and if φ occurs as a subformula of some formula χ, then in χ we can replace φ by ψ without changing its meaning.
- All propositional equivalences carry over to LTL, e.g., $\neg\neg\varphi \equiv \varphi$ and $\varphi \to \psi \equiv \neg\varphi \vee \psi$. **Why?** On a path, $\neg, \wedge, \vee, \to$ mean exactly what they mean in propositional logic.

Duals

- $\wedge, \vee$ are duals of each other: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$, $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$,

Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

- As in prop logic, if two formulae φ, ψ are equivalent and if φ occurs as a subformula of some formula χ, then in χ we can replace φ by ψ without changing its meaning.
- All propositional equivalences carry over to LTL, e.g., $\neg\neg\varphi \equiv \varphi$ and $\varphi \to \psi \equiv \neg\varphi \vee \psi$. **Why?** On a path, $\neg, \wedge, \vee, \to$ mean exactly what they mean in propositional logic.

Duals

- $\wedge, \vee$ are duals of each other: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$, $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$,
- So are F, G! $\neg G\varphi \equiv F\neg\varphi$, $\neg F\varphi \equiv G\neg\varphi$.

Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

- As in prop logic, if two formulae φ, ψ are equivalent and if φ occurs as a subformula of some formula χ, then in χ we can replace φ by ψ without changing its meaning.
- All propositional equivalences carry over to LTL, e.g., $\neg\neg\varphi \equiv \varphi$ and $\varphi \to \psi \equiv \neg\varphi \vee \psi$. **Why?** On a path, $\neg, \wedge, \vee, \to$ mean exactly what they mean in propositional logic.

Duals

- $\wedge, \vee$ are duals of each other: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$, $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$,
- So are F, G! $\neg G\varphi \equiv F\neg\varphi$, $\neg F\varphi \equiv G\neg\varphi$. What is the dual of X?

Equivalence and properties

Equivalence

LTL formulas φ and ψ are (semantically) equivalent, denoted $\varphi \equiv \psi$ if for all models M and all paths π in M we have:

$$\pi \models \varphi \text{ iff } \pi \models \psi$$

- As in prop logic, if two formulae φ, ψ are equivalent and if φ occurs as a subformula of some formula χ, then in χ we can replace φ by ψ without changing its meaning.
- All propositional equivalences carry over to LTL, e.g., $\neg\neg\varphi \equiv \varphi$ and $\varphi \to \psi \equiv \neg\varphi \vee \psi$. **Why?** On a path, $\neg, \wedge, \vee, \to$ mean exactly what they mean in propositional logic.

Duals

- $\wedge, \vee$ are duals of each other: $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$, $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$,
- So are F, G! $\neg G\varphi \equiv F\neg\varphi$, $\neg F\varphi \equiv G\neg\varphi$. What is the dual of X? X is its own dual: $\neg X\varphi \equiv X\neg\varphi$.

Proving equivalences

Claim: $\neg G\varphi \equiv F\neg\varphi$. Proof for an arbitrary model M and path π in M:

$\pi \models \neg G\varphi$

Proving equivalences

Claim: $\neg G\varphi \equiv F\neg\varphi$. Proof for an arbitrary model M and path π in M:

$$\pi \models \neg G\varphi$$

$$\iff \pi \not\models G\varphi \qquad \text{semantics of } \neg$$

Proving equivalences

Claim: $\neg G\varphi \equiv F\neg\varphi$. Proof for an arbitrary model M and path π in M:

$$\pi \models \neg G\varphi$$

$\iff$ $\pi \not\models G\varphi$ semantics of $\neg$

$\iff$ not $(\forall i \geq 1,\ \pi^i \models \varphi)$ semantics of G

Proving equivalences

Claim: $\neg G\varphi \equiv F\neg\varphi$. Proof for an arbitrary model M and path π in M:

$$\pi \models \neg G\varphi$$

$\iff$ $\pi \not\models G\varphi$ — semantics of $\neg$

$\iff$ not $(\forall i \geq 1,\ \pi^i \models \varphi)$ — semantics of G

$\iff$ $\exists i \geq 1,\ \pi^i \not\models \varphi$ — negating $\forall$

Proving equivalences

Claim: $\neg G\varphi \equiv F\neg\varphi$. Proof for an arbitrary model M and path π in M:

$$
\begin{aligned}
& \pi \models \neg G\varphi && \\
\iff\ & \pi \not\models G\varphi && \text{semantics of } \neg \\
\iff\ & \text{not } (\forall i \geq 1,\ \pi^i \models \varphi) && \text{semantics of } G \\
\iff\ & \exists i \geq 1,\ \pi^i \not\models \varphi && \text{negating } \forall \\
\iff\ & \exists i \geq 1,\ \pi^i \models \neg\varphi && \text{semantics of } \neg
\end{aligned}
$$

Proving equivalences

Claim: $\neg G\varphi \equiv F\neg\varphi$. Proof for an arbitrary model M and path π in M:

$$
\begin{aligned}
& \pi \models \neg G\varphi && \\
\iff\ & \pi \not\models G\varphi && \text{semantics of } \neg \\
\iff\ & \text{not } (\forall i \geq 1,\ \pi^i \models \varphi) && \text{semantics of } G \\
\iff\ & \exists i \geq 1,\ \pi^i \not\models \varphi && \text{negating } \forall \\
\iff\ & \exists i \geq 1,\ \pi^i \models \neg\varphi && \text{semantics of } \neg \\
\iff\ & \pi \models F\neg\varphi && \text{semantics of } F
\end{aligned}
$$

Proving equivalences

Claim: $\neg G\varphi \equiv F\neg\varphi$. Proof for an arbitrary model M and path π in M:

$$
\begin{aligned}
& \pi \models \neg G\varphi && \\
\iff\ & \pi \not\models G\varphi && \text{semantics of } \neg \\
\iff\ & \text{not } (\forall i \geq 1,\ \pi^i \models \varphi) && \text{semantics of } G \\
\iff\ & \exists i \geq 1,\ \pi^i \not\models \varphi && \text{negating } \forall \\
\iff\ & \exists i \geq 1,\ \pi^i \models \neg\varphi && \text{semantics of } \neg \\
\iff\ & \pi \models F\neg\varphi && \text{semantics of } F
\end{aligned}
$$

Exercise 24.3 Prove the other duals in the same way:

1. $\neg F\varphi \equiv G\neg\varphi$
2. $\neg X\varphi \equiv X\neg\varphi$

Using equivalences to rewrite a specification

Exercise 24.4 Show that (a) $G(\textbf{drink} \rightarrow \textbf{paid})$ and (b) $\neg F(\textbf{drink} \wedge \neg \textbf{paid})$ from the earlier slide are equivalent, using the duals and propositional equivalences.

Using equivalences to rewrite a specification

Exercise 24.4 Show that (a) $G(\textbf{drink} \rightarrow \textbf{paid})$ and (b) $\neg F(\textbf{drink} \wedge \neg \textbf{paid})$ from the earlier slide are equivalent, using the duals and propositional equivalences.

Solution: start from (b) and push the negation inwards

$\neg F(\textbf{drink} \wedge \neg \textbf{paid})$

Using equivalences to rewrite a specification

Exercise 24.4 Show that (a) $G(\textbf{drink} \rightarrow \textbf{paid})$ and (b) $\neg F(\textbf{drink} \wedge \neg\textbf{paid})$ from the earlier slide are equivalent, using the duals and propositional equivalences.

Solution: start from (b) and push the negation inwards

$$\neg F(\textbf{drink} \wedge \neg\textbf{paid})$$

$$\equiv G\,\neg(\textbf{drink} \wedge \neg\textbf{paid}) \qquad \text{dual: } \neg F\varphi \equiv G\neg\varphi$$

Using equivalences to rewrite a specification

Exercise 24.4 Show that (a) $G(\textbf{drink} \rightarrow \textbf{paid})$ and (b) $\neg F(\textbf{drink} \wedge \neg\textbf{paid})$ from the earlier slide are equivalent, using the duals and propositional equivalences.

Solution: start from (b) and push the negation inwards

$$\neg F(\textbf{drink} \wedge \neg\textbf{paid})$$

$$\equiv G\,\neg(\textbf{drink} \wedge \neg\textbf{paid}) \qquad \text{dual: } \neg F\varphi \equiv G\neg\varphi$$

$$\equiv G\,(\neg\textbf{drink} \vee \neg\neg\textbf{paid}) \qquad \text{De Morgan, inside } G$$

Using equivalences to rewrite a specification

Exercise 24.4 Show that (a) $G(\textbf{drink} \to \textbf{paid})$ and (b) $\neg F(\textbf{drink} \wedge \neg\textbf{paid})$ from the earlier slide are equivalent, using the duals and propositional equivalences.

Solution: start from (b) and push the negation inwards

$$\begin{aligned}
& \neg F(\textbf{drink} \wedge \neg\textbf{paid}) \\
\equiv\ & G\,\neg(\textbf{drink} \wedge \neg\textbf{paid}) && \text{dual: } \neg F\varphi \equiv G\neg\varphi \\
\equiv\ & G\,(\neg\textbf{drink} \vee \neg\neg\textbf{paid}) && \text{De Morgan, inside } G \\
\equiv\ & G\,(\neg\textbf{drink} \vee \textbf{paid}) && \neg\neg\varphi \equiv \varphi
\end{aligned}$$

Using equivalences to rewrite a specification

Exercise 24.4 Show that (a) $G(\textbf{drink} \to \textbf{paid})$ and (b) $\neg F(\textbf{drink} \wedge \neg\textbf{paid})$ from the earlier slide are equivalent, using the duals and propositional equivalences.

Solution: start from (b) and push the negation inwards

$$
\begin{aligned}
& \neg F(\textbf{drink} \wedge \neg\textbf{paid}) && \\
\equiv\ & G\,\neg(\textbf{drink} \wedge \neg\textbf{paid}) && \text{dual: } \neg F\varphi \equiv G\neg\varphi \\
\equiv\ & G\,(\neg\textbf{drink} \vee \neg\neg\textbf{paid}) && \text{De Morgan, inside } G \\
\equiv\ & G\,(\neg\textbf{drink} \vee \textbf{paid}) && \neg\neg\varphi \equiv \varphi \\
\equiv\ & G\,(\textbf{drink} \to \textbf{paid}) && \varphi \to \psi \equiv \neg\varphi \vee \psi
\end{aligned}
$$

Using equivalences to rewrite a specification

Exercise 24.4 Show that (a) $G(\textbf{drink} \to \textbf{paid})$ and (b) $\neg F(\textbf{drink} \wedge \neg\textbf{paid})$ from the earlier slide are equivalent, using the duals and propositional equivalences.

Solution: start from (b) and push the negation inwards

$$
\begin{aligned}
& \neg F(\textbf{drink} \wedge \neg\textbf{paid}) && \\
\equiv\ & G\,\neg(\textbf{drink} \wedge \neg\textbf{paid}) && \text{dual: } \neg F\varphi \equiv G\neg\varphi \\
\equiv\ & G\,(\neg\textbf{drink} \vee \neg\neg\textbf{paid}) && \text{De Morgan, inside } G \\
\equiv\ & G\,(\neg\textbf{drink} \vee \textbf{paid}) && \neg\neg\varphi \equiv \varphi \\
\equiv\ & G\,(\textbf{drink} \to \textbf{paid}) && \varphi \to \psi \equiv \neg\varphi \vee \psi
\end{aligned}
$$

Each step replaces a subformula by an equivalent one, which is allowed by the substitution remark on the definition slide.

Distributivity

Exercise 24.5 Which of these equivalences hold? Prove the ones that do, unfolding the semantics as on the previous slides.

1. $F(\varphi \vee \psi) \equiv F\varphi \vee F\psi$
2. $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$
3. $G(\varphi \wedge \psi) \equiv G\varphi \wedge G\psi$
4. $G(\varphi \vee \psi) \equiv G\varphi \vee G\psi$

Distributivity

Exercise 24.5 Which of these equivalences hold? Prove the ones that do, unfolding the semantics as on the previous slides.

1. $F(\varphi \vee \psi) \equiv F\varphi \vee F\psi$ ✓
2. $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$
3. $G(\varphi \wedge \psi) \equiv G\varphi \wedge G\psi$ ✓
4. $G(\varphi \vee \psi) \equiv G\varphi \vee G\psi$

- F distributes over $\vee$ and G over $\wedge$.

Distributivity

Exercise 24.5 Which of these equivalences hold? Prove the ones that do, unfolding the semantics as on the previous slides.

1. $F(\varphi \vee \psi) \equiv F\varphi \vee F\psi$ ✓
2. $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$ ×
3. $G(\varphi \wedge \psi) \equiv G\varphi \wedge G\psi$ ✓
4. $G(\varphi \vee \psi) \equiv G\varphi \vee G\psi$ ×

- F distributes over $\vee$ and G over $\wedge$.

Distributivity

Exercise 24.5 Which of these equivalences hold? Prove the ones that do, unfolding the semantics as on the previous slides.

1. $F(\varphi \vee \psi) \equiv F\varphi \vee F\psi$ ✓
2. $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$ ×
3. $G(\varphi \wedge \psi) \equiv G\varphi \wedge G\psi$ ✓
4. $G(\varphi \vee \psi) \equiv G\varphi \vee G\psi$ ×

- F distributes over $\vee$ and G over $\wedge$.

Exercise 24.6 How do we show that (2) and (4) do *not* hold?

How do we show a non-equivalence?

The same attempt for $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$

$$
\begin{aligned}
& \pi \models F(\varphi \wedge \psi) \\
\iff & \exists i \geq 1,\ (\pi^i \models \varphi \text{ and } \pi^i \models \psi) \\
\implies & (\exists i \geq 1,\ \pi^i \models \varphi) \text{ and } (\exists j \geq 1,\ \pi^j \models \psi) \\
\iff & \pi \models F\varphi \wedge F\psi
\end{aligned}
$$

How do we show a non-equivalence?

The same attempt for $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$

$\pi \models F(\varphi \wedge \psi)$

$\iff$ $\exists i \geq 1,\ (\pi^i \models \varphi$ and $\pi^i \models \psi)$

$\implies$ $(\exists i \geq 1,\ \pi^i \models \varphi)$ and $(\exists j \geq 1,\ \pi^j \models \psi)$

$\iff$ $\pi \models F\varphi \wedge F\psi$

$\impliedby$ fails: the two witnesses i and j can be different

How do we show a non-equivalence?

The same attempt for $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$

$\pi \models F(\varphi \wedge \psi)$

$\iff$ $\exists i \geq 1,\ (\pi^i \models \varphi$ and $\pi^i \models \psi)$

$\implies$ $(\exists i \geq 1,\ \pi^i \models \varphi)$ and $(\exists j \geq 1,\ \pi^j \models \psi)$

$\iff$ $\pi \models F\varphi \wedge F\psi$

$\impliedby$ fails: the two witnesses i and j can be different

Counterexample: φ holds only at s_1 and ψ only at s_2.

How do we show a non-equivalence?

The same attempt for $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$

$\pi \models F(\varphi \wedge \psi)$

$\iff$ $\exists i \geq 1,\ (\pi^i \models \varphi$ and $\pi^i \models \psi)$

$\implies$ $(\exists i \geq 1,\ \pi^i \models \varphi)$ and $(\exists j \geq 1,\ \pi^j \models \psi)$

$\iff$ $\pi \models F\varphi \wedge F\psi$

$\Longleftarrow$ fails: the two witnesses i and j can be different

Counterexample: φ holds only at s_1 and ψ only at s_2.

This is not a proof that $F(\varphi \wedge \psi) \not\equiv F\varphi \wedge F\psi$: it only shows that *this* attempt at proving the equivalence failed.

How do we show a non-equivalence?

The same attempt for $F(\varphi \wedge \psi) \equiv F\varphi \wedge F\psi$

$$\pi \models F(\varphi \wedge \psi)$$
$$\iff \exists i \geq 1,\ (\pi^i \models \varphi \text{ and } \pi^i \models \psi)$$
$$\implies (\exists i \geq 1,\ \pi^i \models \varphi) \text{ and } (\exists j \geq 1,\ \pi^j \models \psi)$$
$$\iff \pi \models F\varphi \wedge F\psi$$
$$\impliedby \text{fails: the two witnesses } i \text{ and } j \text{ can be different}$$

Counterexample: φ holds only at s_1 and ψ only at s_2.

This is not a proof that $F(\varphi \wedge \psi) \not\equiv F\varphi \wedge F\psi$: it only shows that *this* attempt at proving the equivalence failed.

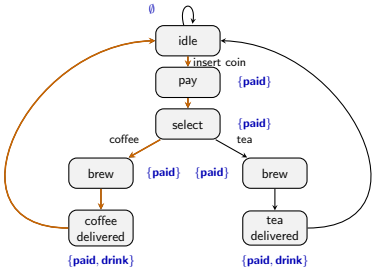
To show a non-equivalence $\varphi \not\equiv \psi$ we need to exhibit

- one model M and one path π in M,
- on which they differ: $\pi \models \varphi$ but $\pi \not\models \psi$ (or the other way round)!

Solution to Exercise 24.6

$F(\varphi \wedge \psi) \not\equiv F\varphi \wedge F\psi$: take the vending machine, $\varphi = \neg$paid, $\psi =$ drink

$\pi =$ idle, pay, select, brew, coffee, idle, ...

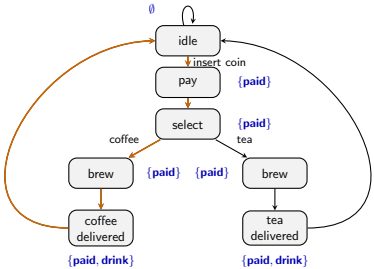


Solution to Exercise 24.6

$F(\varphi \wedge \psi) \not\equiv F\varphi \wedge F\psi$: take the vending machine, $\varphi = \neg$paid, $\psi =$ drink

$\pi =$ idle, pay, select, brew, coffee, idle, ...

- $\pi^1 =$ idle $\models \neg$**paid**, so $\pi \models F\neg$**paid**

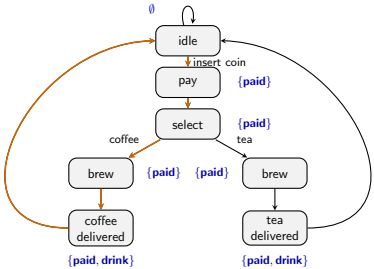


Solution to Exercise 24.6

$F(\varphi \wedge \psi) \not\equiv F\varphi \wedge F\psi$: take the vending machine, $\varphi = \neg$paid, $\psi =$ drink

$\pi =$ idle, pay, select, brew, coffee, idle, ...

- $\pi^1 =$ idle $\models \neg$**paid**, so $\pi \models F\neg$**paid**
- $\pi^5 =$ coffee delivered $\models$ **drink**, so $\pi \models F$ **drink**

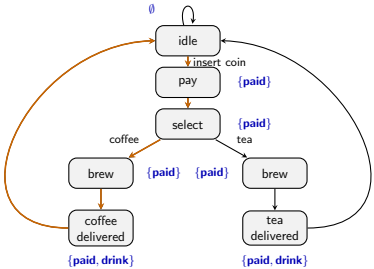


Solution to Exercise 24.6

$F(\varphi \wedge \psi) \not\equiv F\varphi \wedge F\psi$: take the vending machine, $\varphi = \neg$paid, $\psi =$ drink

$\pi =$ idle, pay, select, brew, coffee, idle, ...

- $\pi^1 =$ idle $\models \neg$**paid**, so $\pi \models F\neg$**paid**
- $\pi^5 =$ coffee delivered $\models$ **drink**, so $\pi \models F$ **drink**
- no state has $\neg$**paid** and **drink** (a delivered state has **paid**), so $\pi \not\models F(\neg$**paid** $\wedge$ **drink**$)$



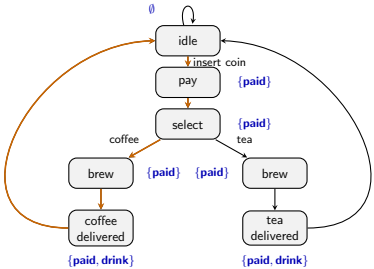
Solution to Exercise 24.6

$F(\varphi \wedge \psi) \not\equiv F\varphi \wedge F\psi$: take the vending machine, $\varphi = \neg$paid, $\psi =$ drink

$\pi =$ idle, pay, select, brew, coffee, idle, ...

- $\pi^1 =$ idle $\models \neg$**paid**, so $\pi \models F\neg$**paid**
- $\pi^5 =$ coffee delivered $\models$ **drink**, so $\pi \models F$ **drink**
- no state has $\neg$**paid** and **drink** (a delivered state has **paid**), so $\pi \not\models F(\neg$**paid** $\wedge$ **drink**$)$

Hence $\pi \models F\varphi \wedge F\psi$ but $\pi \not\models F(\varphi \wedge \psi)$.



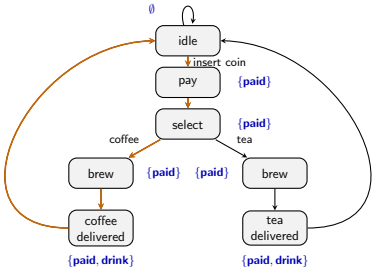
Solution to Exercise 24.6

$F(\varphi \wedge \psi) \not\equiv F\varphi \wedge F\psi$: take the vending machine, $\varphi = \neg$paid, $\psi =$ drink

$\pi =$ idle, pay, select, brew, coffee, idle, ...

- $\pi^1 =$ idle $\models \neg$**paid**, so $\pi \models F\neg$**paid**
- $\pi^5 =$ coffee delivered $\models$ **drink**, so $\pi \models F$ **drink**
- no state has $\neg$**paid** and **drink** (a delivered state has **paid**), so $\pi \not\models F(\neg$**paid** $\wedge$ **drink**$)$

Hence $\pi \models F\varphi \wedge F\psi$ but $\pi \not\models F(\varphi \wedge \psi)$.



Adapt the above idea to show the non-equivalence (4) as well: $G(\varphi \vee \psi) \not\equiv G\varphi \vee G\psi$.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$: so can $\rightarrow$.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$: so can $\rightarrow$.

How few connectives do we really need?

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$: so can $\rightarrow$.

Definition: adequate (called *universal* in ProblemSheet1)

A set of connectives is adequate/universal for a logic, if all other connectives in that logic can be expressed by combinations within this set. E.g., for prop logic $\{\neg, \vee\}$ and $\{\top, \neg, \wedge\}$ are adequate.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$: so can $\rightarrow$.

Definition: adequate (called *universal* in ProblemSheet1)

A set of connectives is adequate/universal for a logic, if all other connectives in that logic can be expressed by combinations within this set. E.g., for prop logic $\{\neg, \vee\}$ and $\{\top, \neg, \wedge\}$ are adequate.

Exercise 24.7 Show that $\{\top, \neg, \wedge, X, U\}$ is adequate for LTL.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \to \psi \equiv \neg\varphi \vee \psi$: so can $\to$.

Definition: adequate (called *universal* in ProblemSheet1)

A set of connectives is adequate/universal for a logic, if all other connectives in that logic can be expressed by combinations within this set. E.g., for prop logic $\{\neg, \vee\}$ and $\{\top, \neg, \wedge\}$ are adequate.

Exercise 24.7 Show that $\{\top, \neg, \wedge, X, U\}$ is adequate for LTL.

- First observe that all non-temporal connectives are done.
- Then, for F, write it in terms of U. Finally, write G in terms of F.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$: so can $\rightarrow$.

Definition: adequate (called *universal* in ProblemSheet1)

A set of connectives is adequate/universal for a logic, if all other connectives in that logic can be expressed by combinations within this set. E.g., for prop logic $\{\neg, \vee\}$ and $\{\top, \neg, \wedge\}$ are adequate.

Exercise 24.7 Show that $\{\top, \neg, \wedge, X, U\}$ is adequate for LTL.

- First observe that all non-temporal connectives are done.
- Then, for F, write it in terms of U. Finally, write G in terms of F.

We still write the others in the logic for simplicity.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \to \psi \equiv \neg\varphi \vee \psi$: so can $\to$.

Definition: adequate (called *universal* in ProblemSheet1)

A set of connectives is adequate/universal for a logic, if all other connectives in that logic can be expressed by combinations within this set. E.g., for prop logic $\{\neg, \vee\}$ and $\{\top, \neg, \wedge\}$ are adequate.

Exercise 24.7 Show that $\{\top, \neg, \wedge, X, U\}$ is adequate for LTL.

- First observe that all non-temporal connectives are done.
- Then, for F, write it in terms of U. Finally, write G in terms of F.

We still write the others in the logic for simplicity. Sometimes we use even more!

- Release: the dual of Until

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \to \psi \equiv \neg\varphi \vee \psi$: so can $\to$.

Definition: adequate (called *universal* in ProblemSheet1)

A set of connectives is adequate/universal for a logic, if all other connectives in that logic can be expressed by combinations within this set. E.g., for prop logic $\{\neg, \vee\}$ and $\{\top, \neg, \wedge\}$ are adequate.

Exercise 24.7 Show that $\{\top, \neg, \wedge, X, U\}$ is adequate for LTL.

- First observe that all non-temporal connectives are done.
- Then, for F, write it in terms of U. Finally, write G in terms of F.

We still write the others in the logic for simplicity. Sometimes we use even more!

- Release: the dual of Until $\varphi \; R \; \psi := \neg(\neg\varphi \; U \; \neg\psi)$

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \to \psi \equiv \neg\varphi \vee \psi$: so can $\to$.

Definition: adequate (called *universal* in ProblemSheet1)

A set of connectives is adequate/universal for a logic, if all other connectives in that logic can be expressed by combinations within this set. E.g., for prop logic $\{\neg, \vee\}$ and $\{\top, \neg, \wedge\}$ are adequate.

Exercise 24.7 Show that $\{\top, \neg, \wedge, X, U\}$ is adequate for LTL.

- First observe that all non-temporal connectives are done.
- Then, for F, write it in terms of U. Finally, write G in terms of F.

We still write the others in the logic for simplicity. Sometimes we use even more!

- Release: the dual of Until $\varphi \; R \; \psi := \neg(\neg\varphi \; U \; \neg\psi)$
- Weak-Until: $\varphi \; W \; \psi := \varphi \; U \; \psi \vee \; G \; \varphi$.

Do we need all these connectives?

- $G\varphi \equiv \neg F \neg \varphi$: G can be written with F and $\neg$.
- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$: $\vee$ can be written with $\neg$ and $\wedge$.
- $\varphi \to \psi \equiv \neg\varphi \vee \psi$: so can $\to$.

Definition: adequate (called *universal* in ProblemSheet1)

A set of connectives is adequate/universal for a logic, if all other connectives in that logic can be expressed by combinations within this set. E.g., for prop logic $\{\neg, \vee\}$ and $\{\top, \neg, \wedge\}$ are adequate.

Exercise 24.7 Show that $\{\top, \neg, \wedge, X, U\}$ is adequate for LTL.

- First observe that all non-temporal connectives are done.
- Then, for F, write it in terms of U. Finally, write G in terms of F.

We still write the others in the logic for simplicity. Sometimes we use even more!

- Release: the dual of Until $\varphi \; R \; \psi := \neg(\neg\varphi \; U \; \neg\psi)$
- Weak-Until: $\varphi \; W \; \psi := \varphi \; U \; \psi \vee \; G \; \varphi$. E.g., $\neg$**drink** W **paid**, "no drink unless the user pays": **paid** here does not need to hold.

What can LTL not say?

Exercise 24.8 Give an example of a property that cannot be written in LTL.

What can LTL not say?

Exercise 24.8 Give an example of a property that cannot be written in LTL.

Hint: from every state, can **enabled** still be reached?

What can LTL not say?

Exercise 24.8 Give an example of a property that cannot be written in LTL.

Hint: from every state, can **enabled** still be reached?

LTL formulas talk about *all* paths: we cannot say "there is a path"...

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.
- Implicitly quantifies universally over paths.

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.
- Implicitly quantifies universally over paths.
- Problem: LTL cannot assert existence of path.

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.
- Implicitly quantifies universally over paths.
- Problem: LTL cannot assert existence of path.
- Soln: To check if there exists a path satisfying φ, just check if all paths satisfy $\neg\varphi$.

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.
- Implicitly quantifies universally over paths.
- Problem: LTL cannot assert existence of path.
- Soln: To check if there exists a path satisfying φ, just check if all paths satisfy $\neg\varphi$.
- But: What if you mix universal and existential path quantifiers? then just taking negation won't work!

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.
- Implicitly quantifies universally over paths.
- Problem: LTL cannot assert existence of path.
- Soln: To check if there exists a path satisfying φ, just check if all paths satisfy $\neg\varphi$.
- But: What if you mix universal and existential path quantifiers? then just taking negation won't work!
  - E.g., "from every state, we can reach a state where a drink is delivered": for *all* states, *there exists* a path to **drink**.

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.
- Implicitly quantifies universally over paths.
- Problem: LTL cannot assert existence of path.
- Soln: To check if there exists a path satisfying φ, just check if all paths satisfy $\neg\varphi$.
- But: What if you mix universal and existential path quantifiers? then just taking negation won't work!
  - E.g., "from every state, we can reach a state where a drink is delivered": for *all* states, *there exists* a path to **drink**.

Computation Tree Logic (CTL)

Introduce two quantifiers A to express 'all paths' and E for 'exists a path'.

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.
- Implicitly quantifies universally over paths.
- Problem: LTL cannot assert existence of path.
- Soln: To check if there exists a path satisfying φ, just check if all paths satisfy $\neg\varphi$.
- But: What if you mix universal and existential path quantifiers? then just taking negation won't work!
  - E.g., "from every state, we can reach a state where a drink is delivered": for *all* states, *there exists* a path to **drink**.

Computation Tree Logic (CTL)

Introduce two quantifiers A to express 'all paths' and E for 'exists a path'.

- There exists a reachable state in which q holds: $E\ F\ q$

Limitations of linear-time logic: no branching!

- LTL is evaluated on paths. A state of a system satisfies an LTL formula if all paths from the state satisfy it.
- Implicitly quantifies universally over paths.
- Problem: LTL cannot assert existence of path.
- Soln: To check if there exists a path satisfying φ, just check if all paths satisfy $\neg\varphi$.
- But: What if you mix universal and existential path quantifiers? then just taking negation won't work!
  - E.g., "from every state, we can reach a state where a drink is delivered": for *all* states, *there exists* a path to **drink**.

Computation Tree Logic (CTL)

Introduce two quantifiers A to express 'all paths' and E for 'exists a path'.

- There exists a reachable state in which q holds: $E\ F\ q$
- Whenever a state satisfying p is reached, there is a path where q holds continuously after that: $A\ G\ (p \rightarrow E\ G\ q)$

CTL Syntax

$$\varphi ::= \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid$$
$$AX\varphi \mid EX\varphi \mid AF\varphi \mid EF\varphi \mid AG\varphi \mid EG\varphi \mid A[\varphi \ U \ \varphi] \mid E[\varphi \ U \ \varphi]$$

where $p \in AP$ atomic propositions.

CTL Syntax

$$\varphi ::= \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid$$
$$AX\varphi \mid EX\varphi \mid AF\varphi \mid EF\varphi \mid AG\varphi \mid EG\varphi \mid A[\varphi\ U\ \varphi] \mid E[\varphi\ U\ \varphi]$$

where $p \in AP$ atomic propositions.

- $A\psi$ means along all paths the temporal formula ψ holds.
- $E\psi$ means there exists a path in which the temporal formula ψ holds.

CTL Syntax

$$\varphi ::= \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid$$
$$AX\varphi \mid EX\varphi \mid AF\varphi \mid EF\varphi \mid AG\varphi \mid EG\varphi \mid A[\varphi\ U\ \varphi] \mid E[\varphi\ U\ \varphi]$$

where $p \in AP$ atomic propositions.

- $A\psi$ means along all paths the temporal formula ψ holds.
- $E\psi$ means there exists a path in which the temporal formula ψ holds.
- All temporal formula must have A or E in front and all occurrences of A or E must be followed by temporal formulas

CTL Syntax

$$\varphi ::= \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \rightarrow \varphi) \mid$$
$$AX\varphi \mid EX\varphi \mid AF\varphi \mid EF\varphi \mid AG\varphi \mid EG\varphi \mid A[\varphi\ U\ \varphi] \mid E[\varphi\ U\ \varphi]$$

where $p \in AP$ atomic propositions.

- $A\psi$ means along all paths the temporal formula ψ holds.
- $E\psi$ means there exists a path in which the temporal formula ψ holds.
- All temporal formula must have A or E in front and all occurrences of A or E must be followed by temporal formulas
- Similar binding as for LTL: unary connectives ($\neg$, AX, EX, AF, EF, AG, EG) bind most tightly

CTL Syntax

$$\varphi ::= \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \to \varphi) \mid$$
$$AX\varphi \mid EX\varphi \mid AF\varphi \mid EF\varphi \mid AG\varphi \mid EG\varphi \mid A[\varphi\ U\ \varphi] \mid E[\varphi\ U\ \varphi]$$

where $p \in AP$ atomic propositions.

- $A\psi$ means along all paths the temporal formula ψ holds.
- $E\psi$ means there exists a path in which the temporal formula ψ holds.
- All temporal formula must have A or E in front and all occurrences of A or E must be followed by temporal formulas
- Similar binding as for LTL: unary connectives ($\neg$, AX, EX, AF, EF, AG, EG) bind most tightly

Exercise 24.9 Which of the following are well-formed CTL formulas? Draw their parse tree.

1. $AG\ (q \to EG\ r)$
2. $EFG\ r$
3. $F[r\ Uq]$
4. $A[p\ U\ A[q\ U\ r]]$
5. $A[(r\ U\ q) \wedge (p\ U\ r)]$
6. $A[(r\ \wedge\ q)\ U\ (\neg p \to\ r)]$

CTL Syntax

$$\varphi ::= \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \to \varphi) \mid$$
$$AX\varphi \mid EX\varphi \mid AF\varphi \mid EF\varphi \mid AG\varphi \mid EG\varphi \mid A[\varphi\ U\ \varphi] \mid E[\varphi\ U\ \varphi]$$

where $p \in AP$ atomic propositions.

- $A\psi$ means along all paths the temporal formula ψ holds.
- $E\psi$ means there exists a path in which the temporal formula ψ holds.
- All temporal formula must have A or E in front and all occurrences of A or E must be followed by temporal formulas
- Similar binding as for LTL: unary connectives ($\neg$, AX, EX, AF, EF, AG, EG) bind most tightly

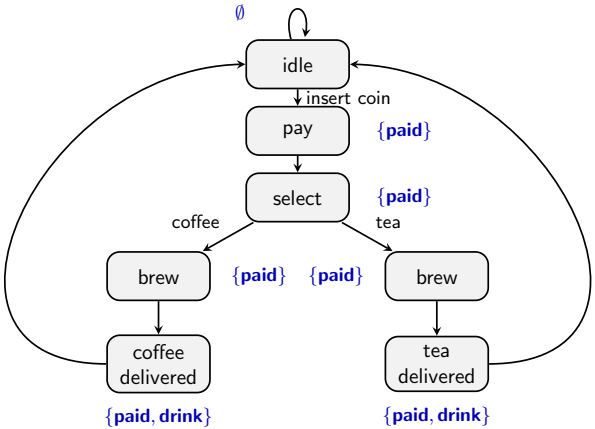
Exercise 24.9 Which of the following are well-formed CTL formulas? Draw their parse tree.

1. $AG\ (q \to EG\ r)$ ✓
2. $EFG\ r$ ×
3. $F[r\ Uq]$ ×
4. $A[p\ U\ A[q\ U\ r]]$ ✓
5. $A[(r\ U\ q) \wedge (p\ U\ r)]$ ×
6. $A[(r\ \wedge\ q)\ U\ (\neg p \to\ r)]$ ✓

CTL on the vending machine

Exercise 24.10 In the vending machine, which of these hold at *idle*? (E: some path from the state, A: all paths from the state.)

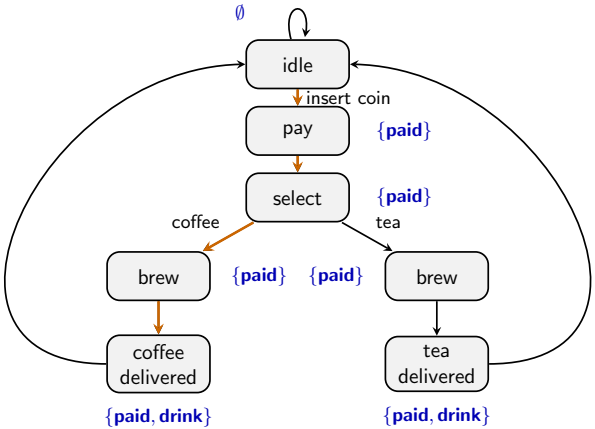
1. EF **drink**?
2. AF **drink**?
3. $EG\ \neg$**drink**?
4. $AG\ EF$ **drink**?



CTL on the vending machine

Exercise 24.10 In the vending machine, which of these hold at *idle*? (E: some path from the state, A: all paths from the state.)

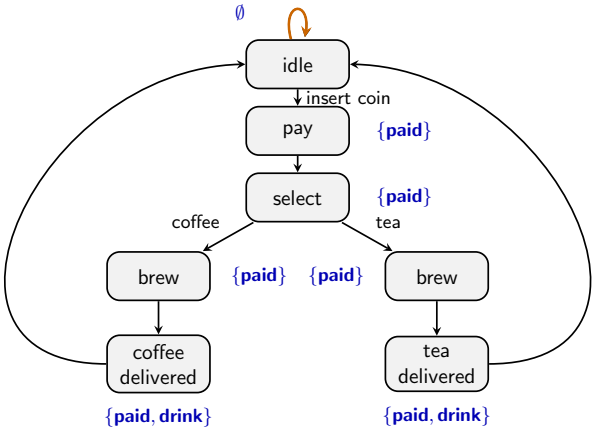
1. EF **drink**? ✓ some path reaches a delivered state.
2. AF **drink**?
3. $EG\ \neg$**drink**?
4. $AG\ EF$ **drink**?



CTL on the vending machine

Exercise 24.10 In the vending machine, which of these hold at *idle*? (E: some path from the state, A: all paths from the state.)

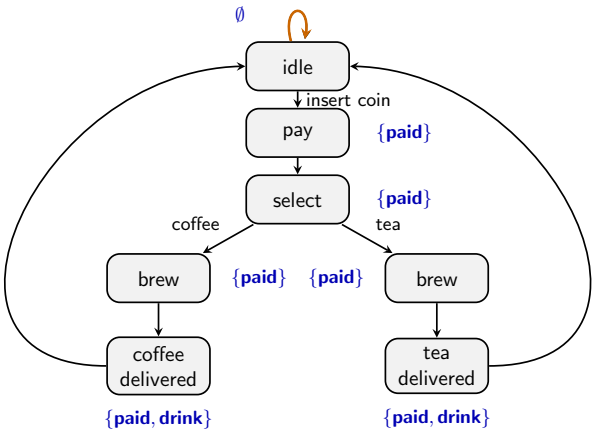
1. EF **drink**? ✓ some path reaches a delivered state.
2. AF **drink**? × the path idle, idle, . . . never delivers one.
3. $EG\,\neg$**drink**?
4. $AG\,EF$ **drink**?



CTL on the vending machine

Exercise 24.10 In the vending machine, which of these hold at *idle*? (E: some path from the state, A: all paths from the state.)

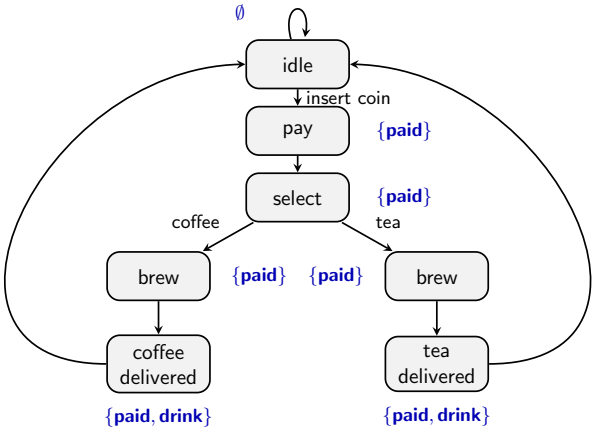
1. EF **drink**? ✓ some path reaches a delivered state.
2. AF **drink**? × the path idle, idle, . . . never delivers one.
3. $EG\,\neg$**drink**? ✓ the same path, idle, idle, . . . , has no drink.
4. $AG\,EF$ **drink**?



CTL on the vending machine

Exercise 24.10 In the vending machine, which of these hold at *idle*? (E: some path from the state, A: all paths from the state.)

1. EF **drink**? ✓ some path reaches a delivered state.
2. AF **drink**? × the path idle, idle, . . . never delivers one.
3. $EG\,\neg$**drink**? ✓ the same path, idle, idle, . . . , has no drink.
4. $AG\,EF$ **drink**? ✓ from every state, we can reach a state where a drink is delivered.



Semantics of CTL

- CTL formulae are interpreted over transition systems directly.
- In Exercise 24.10 you evaluated formulas at a *state* of the machine: A and E ranged over the paths from that state. The definition below makes this general.

Semantics of CTL

- CTL formulae are interpreted over transition systems directly.
- In Exercise 24.10 you evaluated formulas at a *state* of the machine: A and E ranged over the paths from that state. The definition below makes this general.

Let $M = (S, \to, L)$ be a model, $s \in S$, φ a CTL formula. Then the satisfaction relation $M, s \models \varphi$ is defined inductively, as follows,

Semantics of CTL

Let $M = (S, \rightarrow, L)$ be a model, $s \in S$, φ a CTL formula. Then the satisfaction relation $M, s \models \varphi$ is defined inductively, as follows,

Semantics of CTL

Let $M = (S, \to, L)$ be a model, $s \in S$, φ a CTL formula. Then the satisfaction relation $M, s \models \varphi$ is defined inductively, as follows,

- $M, s \models p$ iff $p \in L(s)$
- $\top, \bot, \neg, \vee, \wedge, \to$ defined as usual, e.g., $M, s \models \neg\varphi$ iff $M, s \not\models \varphi$.

Semantics of CTL

Let $M = (S, \to, L)$ be a model, $s \in S$, φ a CTL formula. Then the satisfaction relation $M, s \models \varphi$ is defined inductively, as follows,

- $M, s \models p$ iff $p \in L(s)$
- $M, s \models AX\varphi$ iff $\forall s_1, s \to s_1$, we have $M, s_1 \models \varphi$.
- $\top, \bot, \neg, \vee, \wedge, \to$ defined as usual
- $M, s \models EX\varphi$ iff $\exists s_1, s \to s_1$, we have $M, s_1 \models \varphi$.

If the outermost connective starts with E (resp. A), then for some (resp. all) path(s) starting from s, 'LTL formula' obtained by removing the top E (resp. A) is satisfied.

Semantics of CTL

Let $M = (S, \to, L)$ be a model, $s \in S$, φ a CTL formula. Then the satisfaction relation $M, s \models \varphi$ is defined inductively, as follows,

- $M, s \models p$ iff $p \in L(s)$
- $M, s \models AX\varphi$ iff $\forall s_1, s \to s_1$, we have $M, s_1 \models \varphi$.
- $M, s \models AG\varphi$ iff $\forall$ paths $s = s_1 \to s_2 \ldots$, $\forall s_i$, we have $M, s_i \models \varphi$.
- $\top, \bot, \neg, \vee, \wedge, \to$ defined as usual
- $M, s \models EX\varphi$ iff $\exists s_1, s \to s_1$, we have $M, s_1 \models \varphi$.
- $M, s \models EG\varphi$ iff $\exists$ path $s = s_1 \to s_2 \ldots$, $\forall s_i$, we have $M, s_i \models \varphi$.

If the outermost connective starts with E (resp. A), then for some (resp. all) path(s) starting from s, 'LTL formula' obtained by removing the top E (resp. A) is satisfied.

Semantics of CTL

Let $M = (S, \to, L)$ be a model, $s \in S$, φ a CTL formula. Then the satisfaction relation $M, s \models \varphi$ is defined inductively, as follows,

- $M, s \models p$ iff $p \in L(s)$
- $M, s \models AX\varphi$ iff $\forall s_1, s \to s_1$, we have $M, s_1 \models \varphi$.
- $M, s \models AG\varphi$ iff $\forall$ paths $s = s_1 \to s_2 \ldots$, $\forall s_i$, we have $M, s_i \models \varphi$.
- $M, s \models AF\varphi$ iff $\forall$ paths $s = s_1 \to s_2 \ldots$, $\exists s_i$ in path with $M, s_i \models \varphi$.
- $\top, \bot, \neg, \vee, \wedge, \to$ defined as usual
- $M, s \models EX\varphi$ iff $\exists s_1, s \to s_1$, we have $M, s_1 \models \varphi$.
- $M, s \models EG\varphi$ iff $\exists$ path $s = s_1 \to s_2 \ldots$, $\forall s_i$, we have $M, s_i \models \varphi$.
- $M, s \models EF\varphi$ iff $\exists$ path $s = s_1 \to s_2 \ldots$, $\exists s_i$ in path with $M, s_i \models \varphi$.

If the outermost connective starts with E (resp. A), then for some (resp. all) path(s) starting from s, 'LTL formula' obtained by removing the top E (resp. A) is satisfied.

Semantics of CTL

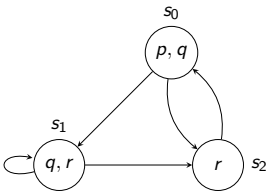
Let $M = (S, \to, L)$ be a model, $s \in S$, φ a CTL formula. Then the satisfaction relation $M, s \models \varphi$ is defined inductively, as follows,

- $M, s \models p$ iff $p \in L(s)$
- $M, s \models AX\varphi$ iff $\forall s_1, s \to s_1$, we have $M, s_1 \models \varphi$.
- $M, s \models AG\varphi$ iff $\forall$ paths $s = s_1 \to s_2 \ldots$, $\forall s_i$, we have $M, s_i \models \varphi$.
- $M, s \models AF\varphi$ iff $\forall$ paths $s = s_1 \to s_2 \ldots$, $\exists s_i$ in path with $M, s_i \models \varphi$.
- $M, s \models A[\varphi\ U \psi]$ iff $\forall$ paths $s = s_1 \to s_2 \ldots$, $\exists s_i$ such that $M, s_i \models \psi$ and $\forall j < i$, $M, s_j \models \varphi$.
- $\top, \bot, \neg, \vee, \wedge, \to$ defined as usual
- $M, s \models EX\varphi$ iff $\exists s_1, s \to s_1$, we have $M, s_1 \models \varphi$.
- $M, s \models EG\varphi$ iff $\exists$ path $s = s_1 \to s_2 \ldots$, $\forall s_i$, we have $M, s_i \models \varphi$.
- $M, s \models EF\varphi$ iff $\exists$ path $s = s_1 \to s_2 \ldots$, $\exists s_i$ in path with $M, s_i \models \varphi$.
- $M, s \models E[\varphi\ U \psi]$ iff $\exists$ path $s = s_1 \to s_2 \ldots$, $\exists s_i$ such that $M, s_i \models \psi$ and $\forall j < i$, $M, s_j \models \varphi$.

If the outermost connective starts with E (resp. A), then for some (resp. all) path(s) starting from s, 'LTL formula' obtained by removing the top E (resp. A) is satisfied.

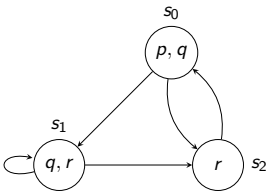
An example: Consider M from before...

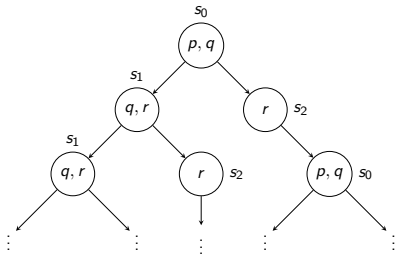
From any given state, we can unwind the system as an infinite tree of computation paths.



An example: Consider M from before...

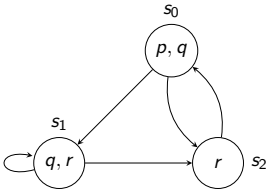
From any given state, we can unwind the system as an infinite tree of computation paths.

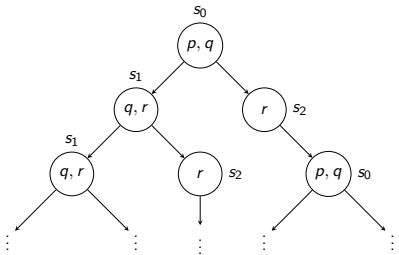




An example: Consider M from before...

From any given state, we can unwind the system as an infinite tree of computation paths.



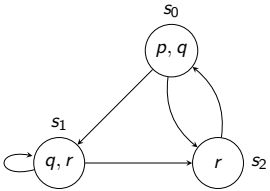


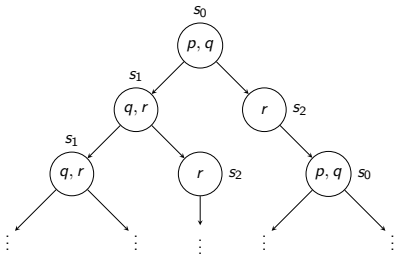
Exercise 24.11 Which of the following hold? Why or why not? Justify!

1. $M, s_0 \models (p \wedge q) \wedge \neg r$
2. $M, s_0 \models EX(q \wedge r)$
3. $M, s_0 \models AX(q \wedge r)$
4. $M, s_0 \models \neg EF(p \wedge r)$

An example: Consider M from before...

From any given state, we can unwind the system as an infinite tree of computation paths.



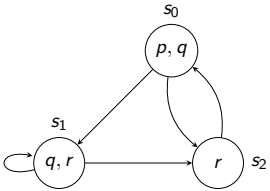


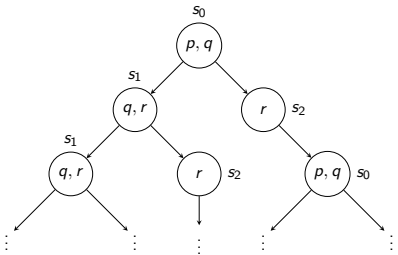
Exercise 24.11 Which of the following hold? Why or why not? Justify!

1. $M, s_0 \models (p \wedge q) \wedge \neg r$
2. $M, s_0 \models EX(q \wedge r)$
3. $M, s_0 \models AX(q \wedge r)$
4. $M, s_0 \models \neg EF(p \wedge r)$
5. $M, s_1 \models EG(q \wedge r)$
6. $M, s_0 \models AF\ r$
7. $M, s_0 \models E[(p \wedge q)\ U\ (q \wedge r)]$
8. $M, s_0 \models A[(p \wedge q)\ U\ (q \wedge r)]$

An example: Consider M from before...

From any given state, we can unwind the system as an infinite tree of computation paths.



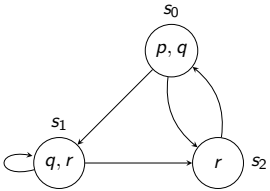


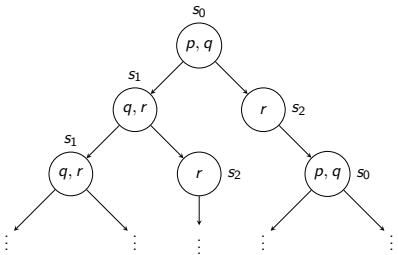
Exercise 24.11 Which of the following hold? Why or why not? Justify!

1. $M, s_0 \models (p \wedge q) \wedge \neg r$ ✓
2. $M, s_0 \models EX(q \wedge r)$ ✓
3. $M, s_0 \models AX(q \wedge r)$ ×
4. $M, s_0 \models \neg EF(p \wedge r)$ ✓
5. $M, s_1 \models EG(q \wedge r)$ ✓
6. $M, s_0 \models AF\ r$ ✓
7. $M, s_0 \models E[(p \wedge q)\ U\ (q \wedge r)]$ ✓
8. $M, s_0 \models A[(p \wedge q)\ U\ (q \wedge r)]$ ×

An example: Consider M from before...

From any given state, we can unwind the system as an infinite tree of computation paths.





Exercise 24.11 Which of the following hold? Why or why not? Justify!

1. $M, s_0 \models (p \wedge q) \wedge \neg r$ ✓
2. $M, s_0 \models EX(q \wedge r)$ ✓
3. $M, s_0 \models AX(q \wedge r)$ ×
4. $M, s_0 \models \neg EF(p \wedge r)$ ✓
5. $M, s_1 \models EG(q \wedge r)$ ✓
6. $M, s_0 \models AF\ r$ ✓
7. $M, s_0 \models E[(p \wedge q)\ U\ (q \wedge r)]$ ✓
8. $M, s_0 \models A[(p \wedge q)\ U\ (q \wedge r)]$ ×

Some of these were also in the LTL lecture last time. Which ones?