CSC410

AZADEH FARZAN

FALL 2020

Model Checking

Overview

- Nontraditional use of nontraditional logic!
- Checking whether a formula is satisfied in a finite domain.
  - Model: finite-state transition system
  - Logic: Propositional Temporal Logic.
  - Verification Procedure: exhaustively search of the state space to determine the truth of specification.

Why Model checking?

- Doesn't aim too high!
  - Originally restricted to finite-state systems.
  - applicable to systems with "short" descriptions.
  - control-oriented systems such as hardware, protocols, ...
- Fully automatic with low computational complexity.
- Can be viewed as an elaborate debugging tool: counterexamples.

First Step:
We need a formal model!

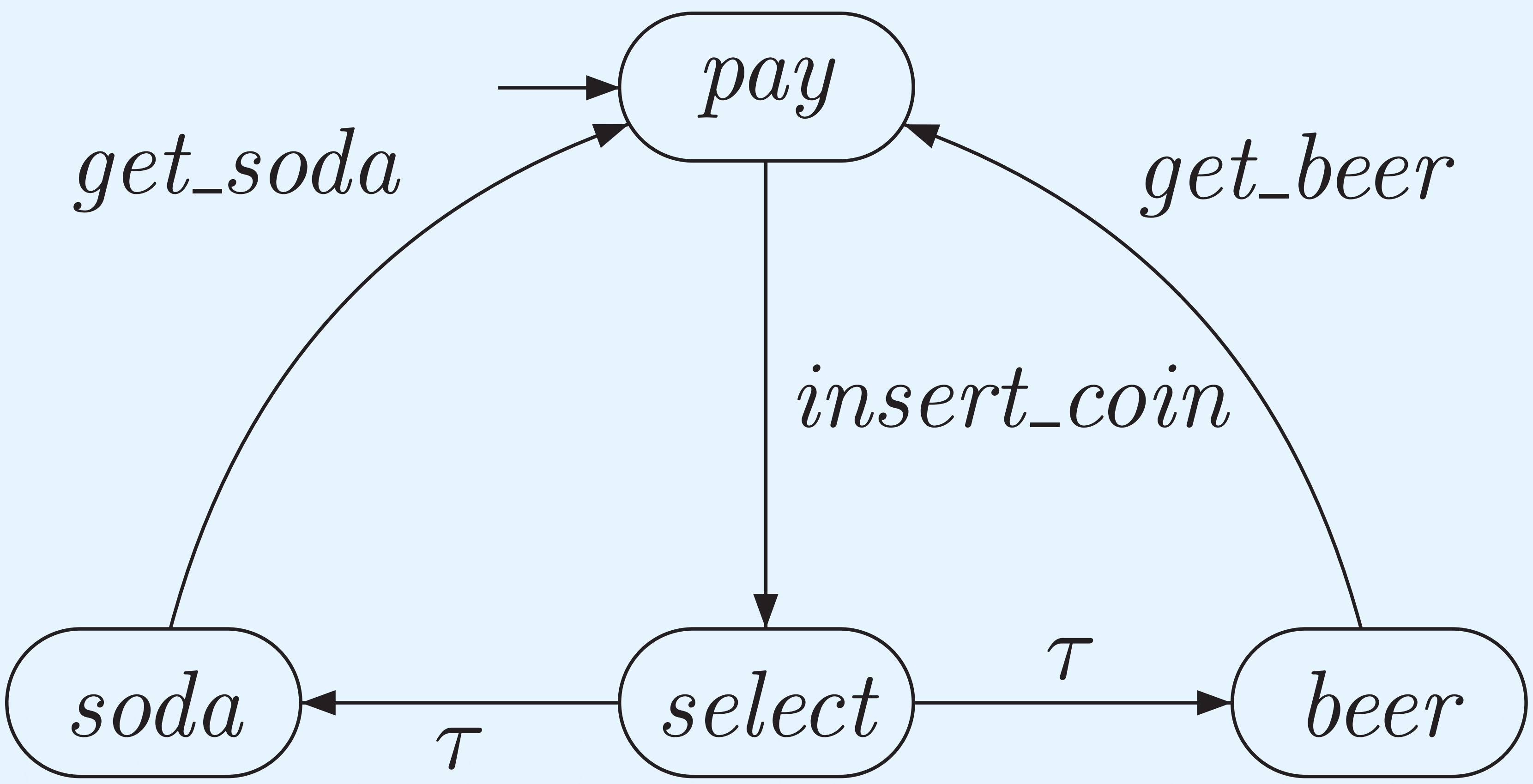
Labeled Transition Systems

A *transition system* TS is a tuple $(S, Act, \rightarrow, I, AP, L)$ where

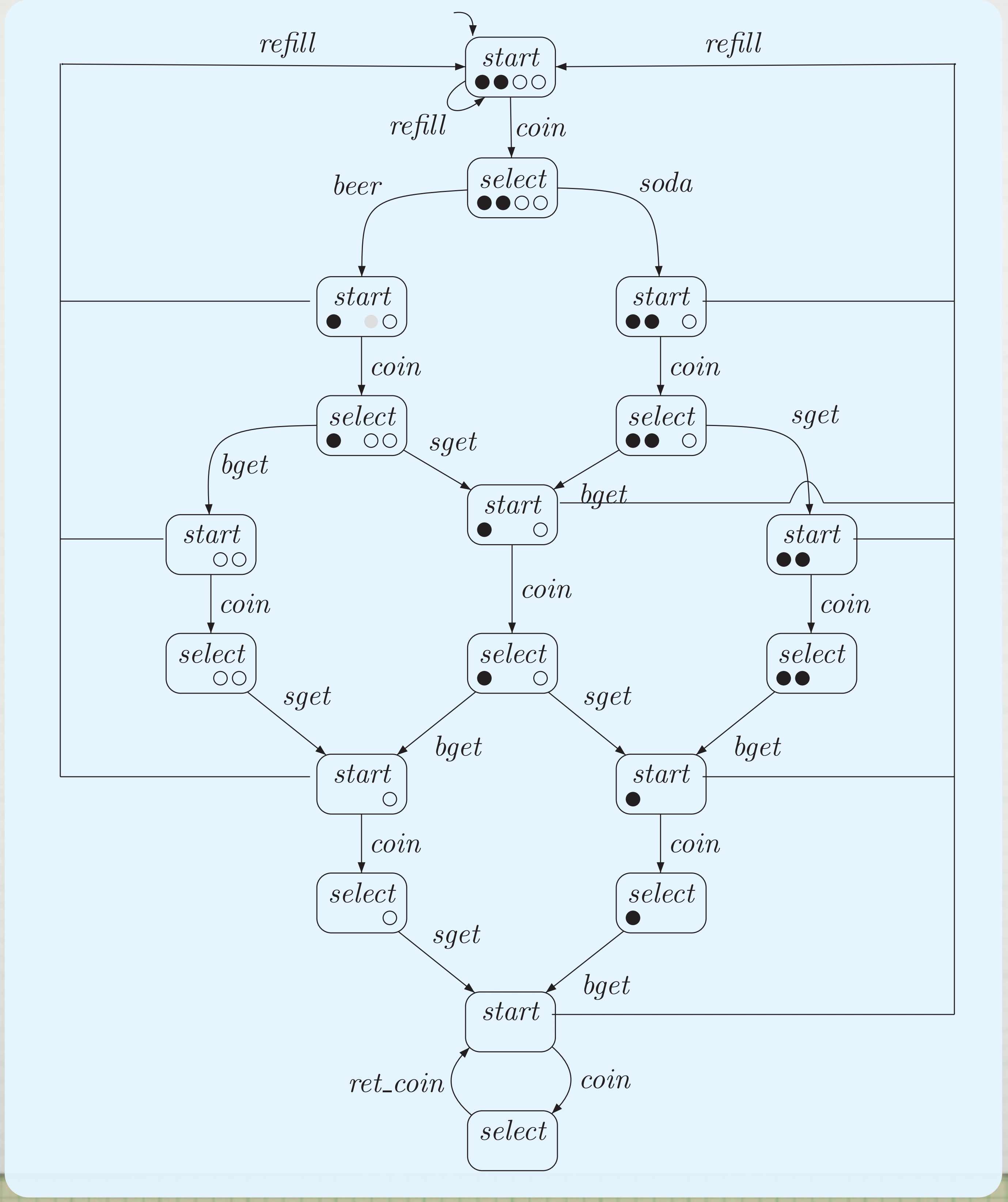
- S is a set of states,
- Act is a set of actions,
- $\longrightarrow \subseteq \ S \times Act \times S$ is a transition relation,
- $I \subseteq S$ is a set of initial states,
- AP is a set of atomic propositions, and
- $L : S \rightarrow 2^{AP}$ is a labeling function.

TS is called *finite* if S, Act, and AP are finite.

Example

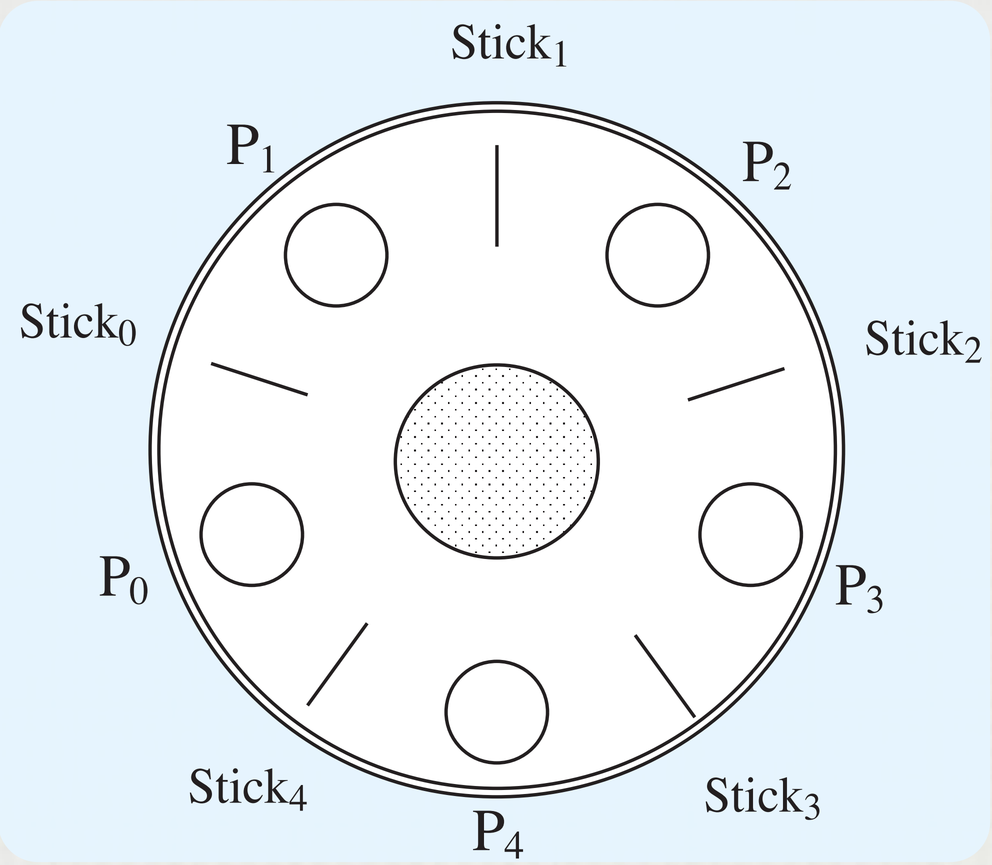


Example


refill
refill
start
refill
coin
select
beer
soda
start
start
coin
coin
select
select
sget
sget
bget
start
bget
start
start
coin
coin
coin
select
select
select
sget
sget
bget
bget
start
start
coin
coin
select
select
sget
bget
start
ret_coin
coin
select

Second Step:
We need a formal Specification!

Example: Dining Philosophers



There are 5 philosophers at a table sharing 5 chopsticks for eating.

Each philosopher needs two chopsticks to eat.

At each point in time at most one of two neighbouring philosophers can eat.

Classic deadlock scenario example!

Reachability

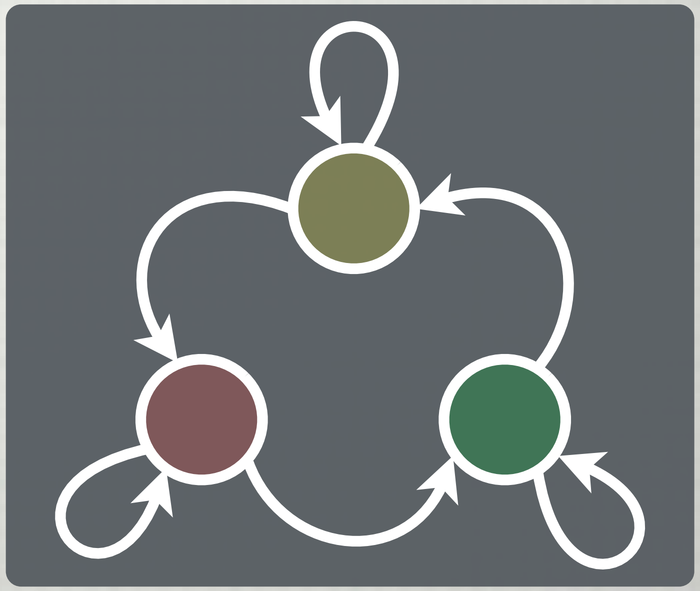
Problem: given an TS, and a target set T, is T reachable from Q_0.

Solution? Depth First Search, in O(n+m) time.

What if we are interested in more sophisticated properties?

Suggest a non-reachability property for philosophers!

The light will always eventually turn green.



Option 1 for properties beyond reachability ...

One TS as a Spec for Another TS!

Given a TS M for the model and a TS S for the specification:

Question: Is every behaviour of M a behaviour of S?

$$L(M) \subseteq L(S)$$

Solvable in PSpace: linear in M and exponential in S.

Best choice: new logic!

Alternative: Temporal Logic

- Language for describing properties of infinite sequences.
- Extension of propositional logic.
- Uses temporal operators to describe sequencing properties.

Linear Temporal Logic

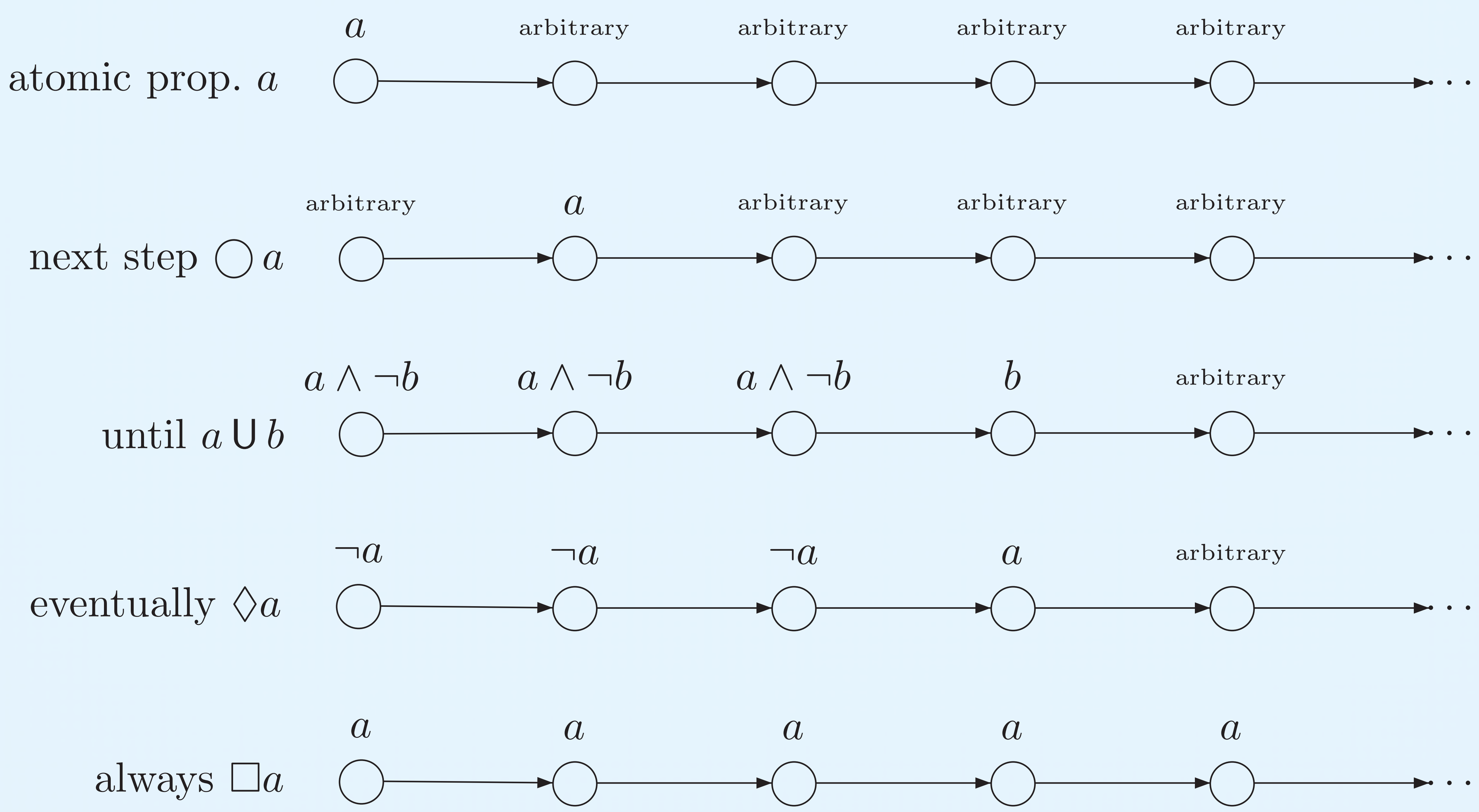
LTL Syntax

$$\varphi ::= \text{true} \;\Big|\; a \;\Big|\; \varphi_1 \wedge \varphi_2 \;\Big|\; \neg\varphi \;\Big|\; \boxed{\bigcirc\, \varphi \;\Big|\; \varphi_1 \,\mathsf{U}\, \varphi_2}$$

$a \in AP$

$$\Diamond\varphi \overset{\text{def}}{=} \text{true}\,\mathsf{U}\,\varphi \qquad\qquad \Box\varphi \overset{\text{def}}{=} \neg\Diamond\neg\varphi$$

LTL: Intuition



LTL Semantics

LTL is interpreted over paths.

These paths are (infinite) words labeled with subset of the atomic propositions (AP) that are true at each letter.

$$
\begin{array}{llll}
\sigma & \models & \text{true} & \\
\sigma & \models & a & \text{iff} \quad a \in A_0 \quad (\text{i.e., } A_0 \models a) \\
\sigma & \models & \varphi_1 \wedge \varphi_2 & \text{iff} \quad \sigma \models \varphi_1 \text{ and } \sigma \models \varphi_2 \\
\sigma & \models & \neg\varphi & \text{iff} \quad \sigma \not\models \varphi \\
\sigma & \models & \bigcirc\varphi & \text{iff} \quad \sigma[1\ldots] = A_1 A_2 A_3 \ldots \models \varphi \\
\sigma & \models & \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} \quad \exists j \geqslant 0.\ \sigma[j\ldots] \models \varphi_2 \ \text{ and } \ \sigma[i\ldots] \models \varphi_1, \text{ for all } 0 \leqslant i < j
\end{array}
$$

LTL's $\models$ is the smallest relation satisfying the above rules.

$$
\begin{array}{llll}
\sigma & \models & \Diamond\varphi & \text{iff} \quad \exists j \geqslant 0.\ \sigma[j\ldots] \models \varphi \\
\sigma & \models & \Box\varphi & \text{iff} \quad \forall j \geqslant 0.\ \sigma[j\ldots] \models \varphi
\end{array}
$$

$$
\begin{array}{llll}
\sigma & \models & \Box\Diamond\varphi & \text{iff} \quad \overset{\infty}{\exists}\ j.\ \sigma[j\ldots] \models \varphi \\
\sigma & \models & \Diamond\Box\varphi & \text{iff} \quad \overset{\infty}{\forall}\ j.\ \sigma[j\ldots] \models \varphi
\end{array}
$$

More Examples in Class