



**PSC 2023/24** (375AA, 9CFU)

Principles for Software Composition

Roberto Bruni

<http://www.di.unipi.it/~bruni/>

<http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/start>

22a - Temporal logic

# Testing

how do you guarantee that your code is correct?

*testing* can show the presence of bugs

not their absence

coverage of all cases: difficult to achieve

especially in concurrent systems!  
(because of nondeterminism)

# Formal logics

what does it mean to be correct? to satisfy some properties

how are these properties expressed? in some syntax

*formal logics* serve to express properties about programs

safety: something bad will not happen

liveness: something good will happen

*model checking* are certain properties satisfied  
(by a model of the program)?

# Temporal logics

notion of time (discrete, infinite)

properties of states (atomic proposition)

*linear operators* at the next instant

always

never

eventually

*path quantifiers* (nondeterministic systems)

for all possible futures

in a possible future

# Modal logics

notion of time (discrete, infinite)

properties of states (atomic proposition)

*modal operators* at the next step  
at any next step  
(like HM logic)

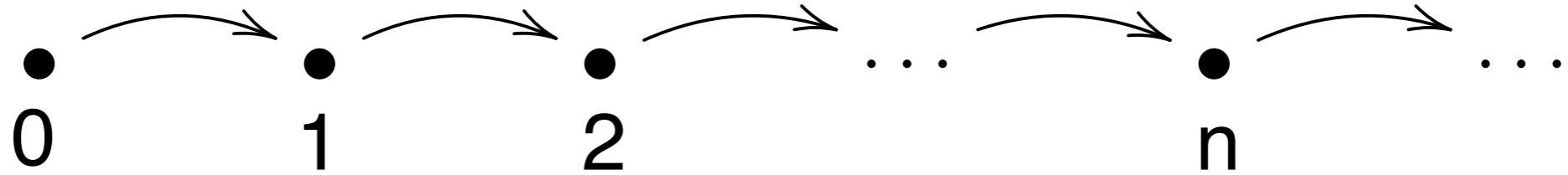
*fix point operators* recursively defined formulas  
minimal / maximal fixpoint  
(meaning of a formula:  
the set of states where it holds)

# LTL

## Linear temporal logic

# Linear Temporal Logic

models



syntax

$\psi$	$::=$	<b>tt</b>   <b>ff</b>   $\neg\psi$   $\psi_0 \wedge \psi_1$   $\psi_0 \vee \psi_1$
		$p$ atomic proposition $p \in P$
		$O\psi$ NEXT: $\psi$ holds at the next instant of time
		$F\psi$ FINALLY: $\psi$ holds sometimes in the future
		$G\psi$ GLOBALLY: $\psi$ holds always in the future
		$\psi_0 U \psi_1$ UNTIL: $\psi_0$ holds until $\psi_1$ is true

$O\psi$  sometimes written  $X\psi$  or  $N\psi$

# Linear Structure

$$S : P \rightarrow \wp(\mathbb{N})$$

set of atomic propositions

$S(p)$  is the set of time instants  
in which  $p$  holds

$$S(p) = \{n \mid p \text{ holds at } n\}$$

**Shift**  $S^k : P \rightarrow \wp(\mathbb{N})$

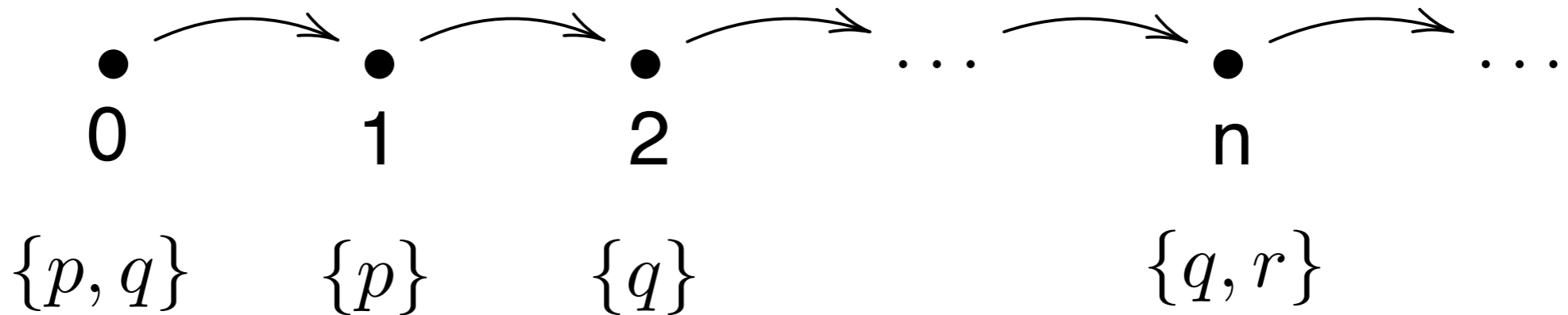
$$S^k(p) = \{n - k \mid n \geq k \wedge n \in S(p)\}$$

$$S^k(p) = \{m \mid m + k \in S(p)\}$$



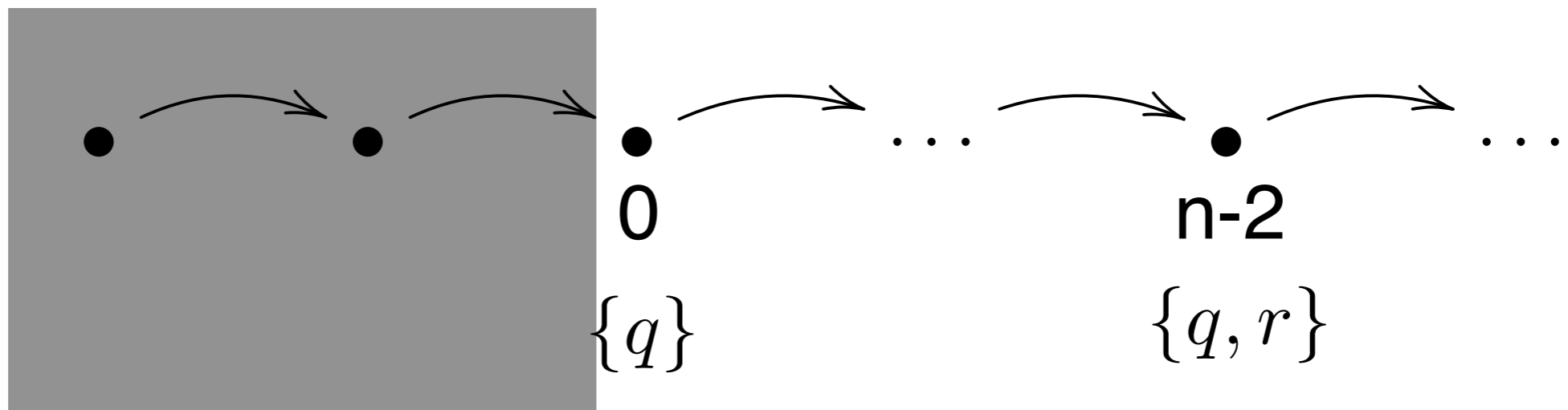
# Example

$$S : P \rightarrow \wp(\mathbb{N})$$

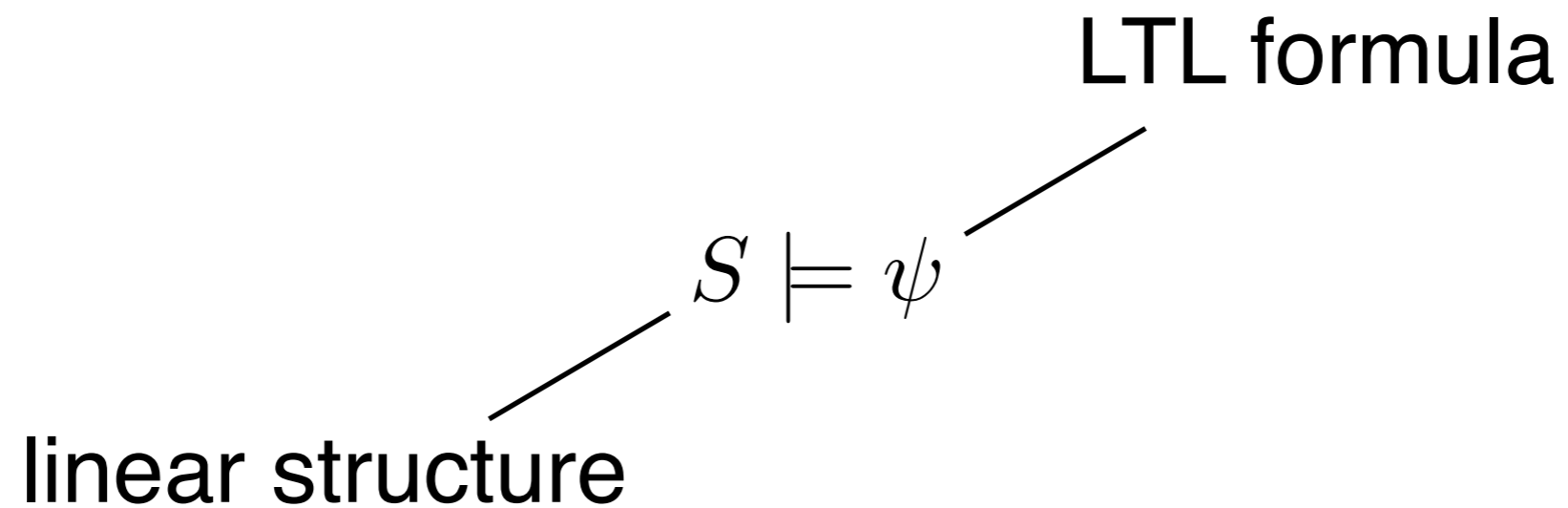


$$S(p) = \{0, 1, \dots\} \quad S(q) = \{0, 2, n, \dots\} \quad S(r) = \{n, \dots\}$$

$$S^2 : P \rightarrow \wp(\mathbb{N})$$



# LTL: satisfaction



# LTL: satisfaction

$$S \models \mathbf{tt}$$

current time: 0

$$S \models \neg\psi \quad \text{iff } S \not\models \psi$$

$$S \models \psi_0 \wedge \psi_1 \quad \text{iff } S \models \psi_0 \text{ and } S \models \psi_1$$

$$S \models \psi_0 \vee \psi_1 \quad \text{iff } S \models \psi_0 \text{ or } S \models \psi_1$$

$$S \models p \quad \text{iff } 0 \in S(p)$$

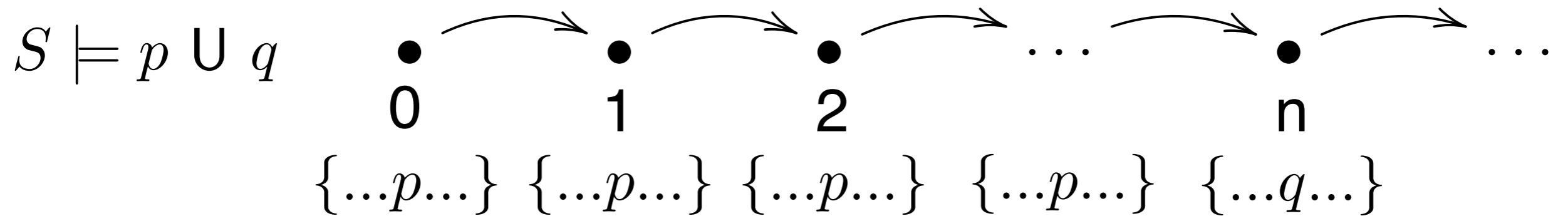
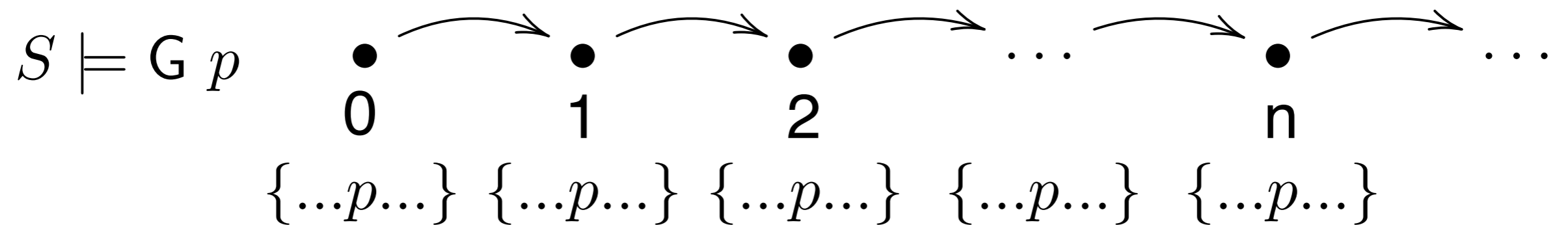
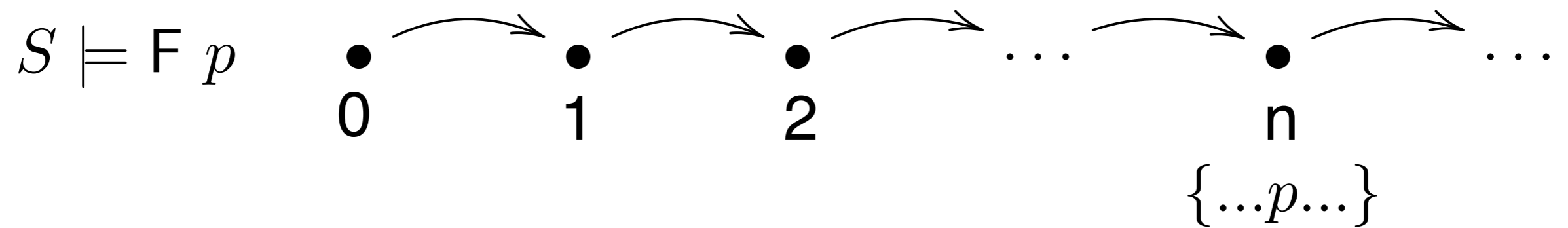
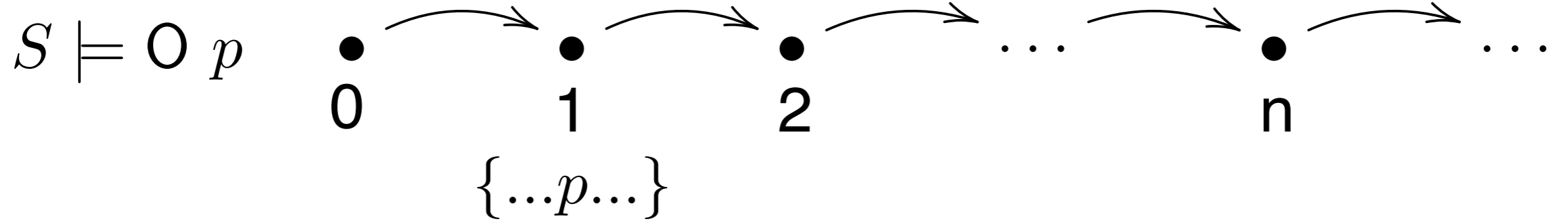
$$S \models \mathbf{O}\psi \quad \text{iff } S^1 \models \psi$$

$$S \models \mathbf{F}\psi \quad \text{iff } \exists k \in \mathbb{N}. S^k \models \psi$$

$$S \models \mathbf{G}\psi \quad \text{iff } \forall k \in \mathbb{N}. S^k \models \psi$$

$$S \models \psi_0 \mathbf{U}\psi_1 \quad \text{iff } \exists k \in \mathbb{N}. S^k \models \psi_1 \text{ and } \forall i < k. S^i \models \psi_0$$

# Examples



# LTL: equivalent formulas

$$\psi_0 \equiv \psi_1 \quad \text{iff} \quad \forall S. S \models \psi_0 \Leftrightarrow S \models \psi_1$$

$$F \psi \equiv \mathbf{tt} \ U \ \psi$$

$$\begin{aligned} G \psi &\equiv \neg(F \neg\psi) \\ &\equiv \neg(\mathbf{tt} \ U \ \neg\psi) \end{aligned}$$

$$\psi_0 \Rightarrow \psi_1 \triangleq \psi_1 \vee \neg\psi_0$$

# Examples

$$G \neg error$$

error will never arise

$$press \Rightarrow F error$$

if you press now, an error will arise in the future

$$G F enter$$

enter will happen infinitely often (fairness)

$$F G idle$$

the system will stay idle from some time in the future onward

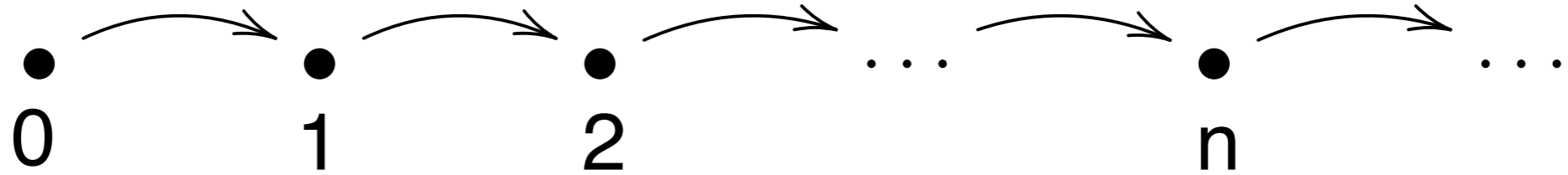
$$G (req \Rightarrow (req U eval))$$

whenever a request is made, it holds until evaluated

# LTL automata-like models

# LTL, again

models



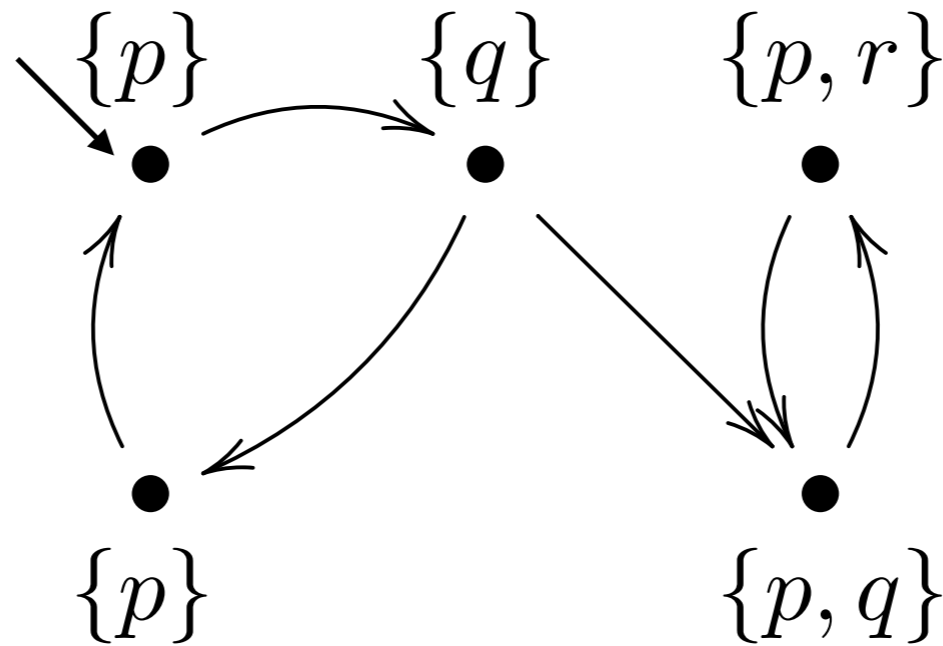
syntax

$\psi$	$::=$	<b>tt</b>   <b>ff</b>   $\neg\psi$   $\psi_0 \wedge \psi_1$   $\psi_0 \vee \psi_1$
		$p$ atomic proposition $p \in P$
		$O\psi$ NEXT: $\psi$ holds at the next instant of time
		$F\psi$ FINALLY: $\psi$ holds sometimes in the future
		$G\psi$ GLOBALLY: $\psi$ holds always in the future
		$\psi_0 U \psi_1$ UNTIL: $\psi_0$ holds until $\psi_1$ is true

$O\psi$  sometimes written  $X\psi$  or  $N\psi$



# Automata-like models

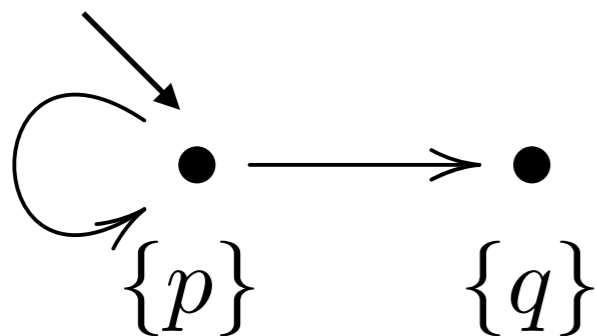


the formula must be satisfied along all (infinite) traces

(if we enter a deadlock state, the last state is repeated forever)



# Exercise



$$\not\models F q \quad \text{⊗} \{p\} \{p\} \{p\} \dots$$

$$\not\models G p \quad \text{⊗} \{p\} \{q\} \{q\} \dots$$

$$\not\models p \cup q \quad \text{⊗} \{p\} \{p\} \{p\} \dots$$

$$\models q \cup p \quad \text{✓}$$

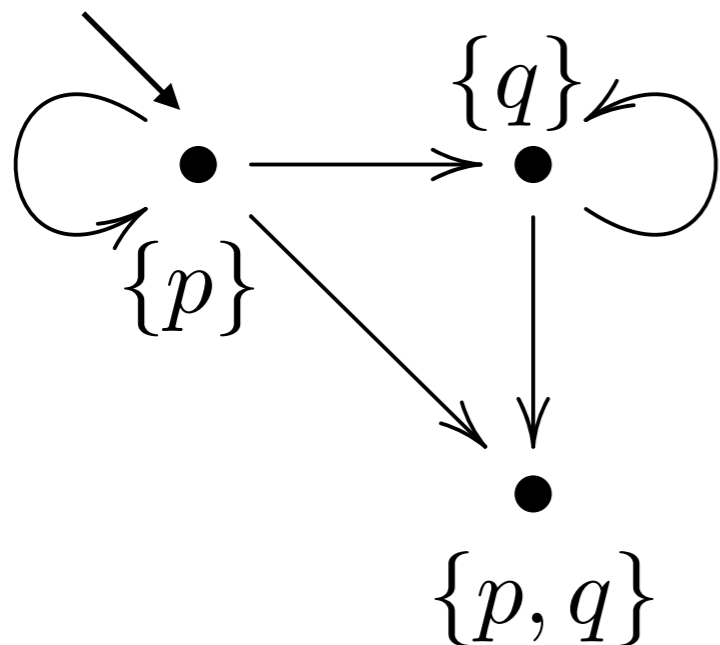
$$\models G(q \Rightarrow G q) \quad \text{✓}$$

the formula must be satisfied along all (infinite) traces

(if we enter a deadlock state, the last state is repeated forever)



# Exercise



$$\not\models G(q \cup p) \quad \times \{p\} \cdots \{p\} \{q\} \{q\} \cdots$$

$$\models G p \vee F q \quad \checkmark$$

$$\not\models F q \Rightarrow \neg G p \quad \times \{p\} \{p, q\} \{p, q\} \cdots$$

$$\models G(q \Rightarrow O q) \quad \checkmark$$

the formula must be satisfied along all (infinite) traces

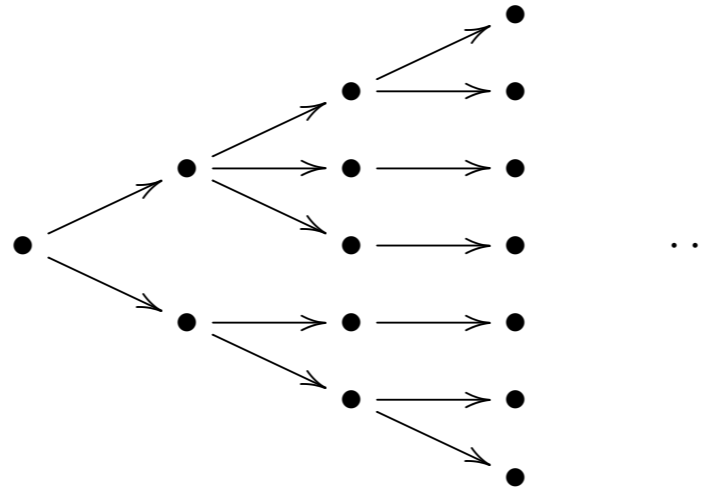
(if we enter a deadlock state, the last state is repeated forever)

# CTL\*, CTL

## Computational tree logic

# Computational Tree Logic

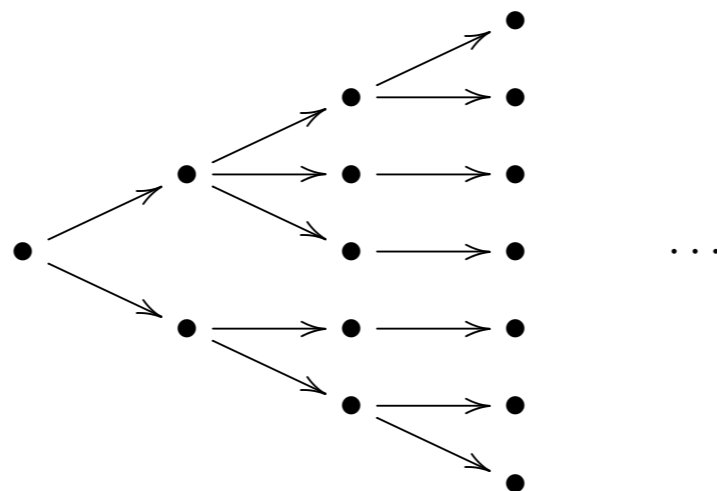
models



syntax (CTL\*)

$\psi$	$::=$	$\mathbf{tt} \mid \mathbf{ff} \mid \neg\psi \mid \psi_0 \wedge \psi_1 \mid \psi_0 \vee \psi_1$	classical ops
		$p \mid \mathbf{O}\psi \mid \mathbf{F}\psi \mid \mathbf{G}\psi \mid \psi_0 \mathbf{U}\psi_1$	linear ops
		$\mathbf{E}\psi$	<b>POSSIBLY:</b> there is a path that satisfies
		$\mathbf{A}\psi$	<b>ALWAYS:</b> every path satisfies $\psi$

# Infinite Tree



$T = (V, \rightarrow)$  directed graph

tree

$v_0 \in V$  root: a distinguished vertex (no incoming arc)

exactly one directed path from  $v_0$  to any other vertex  $v \in V$

infinite

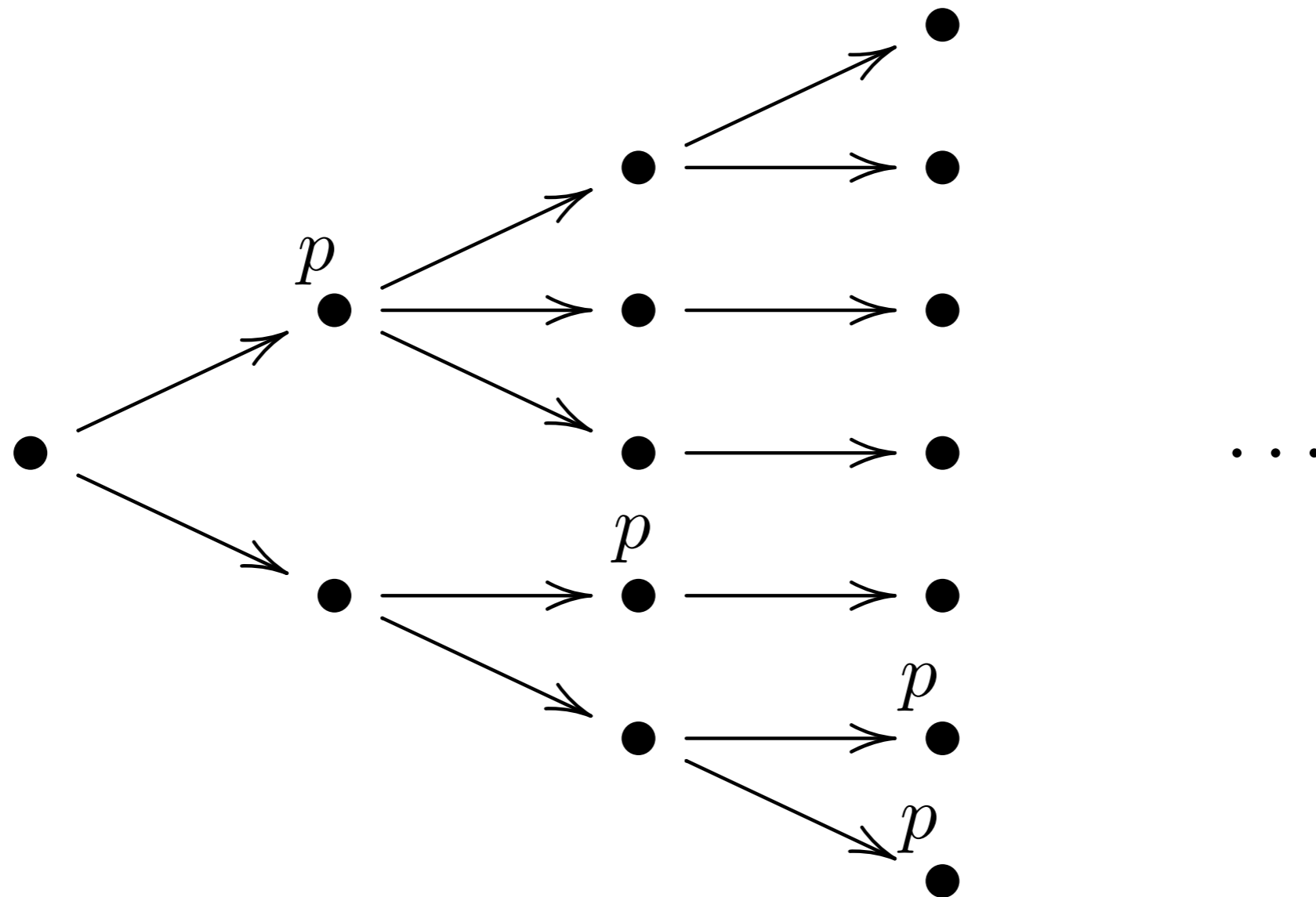
every node has a child

# Branching Structure

$T = (V, \rightarrow)$  infinite tree

$S : P \rightarrow \wp(V)$

$$S(p) = \{x \in V \mid x \text{ satisfies } p\}$$



# Infinite Path

$T = (V, \rightarrow)$        $S : P \rightarrow \wp(V)$       branching structure

infinite path     $T = (V, \rightarrow)$        $\pi : \mathbb{N} \rightarrow V$       ( $\pi = v_0 v_1 \dots$ )

such that  $\forall k \in \mathbb{N}. v_k \rightarrow v_{k+1}$

path shifting

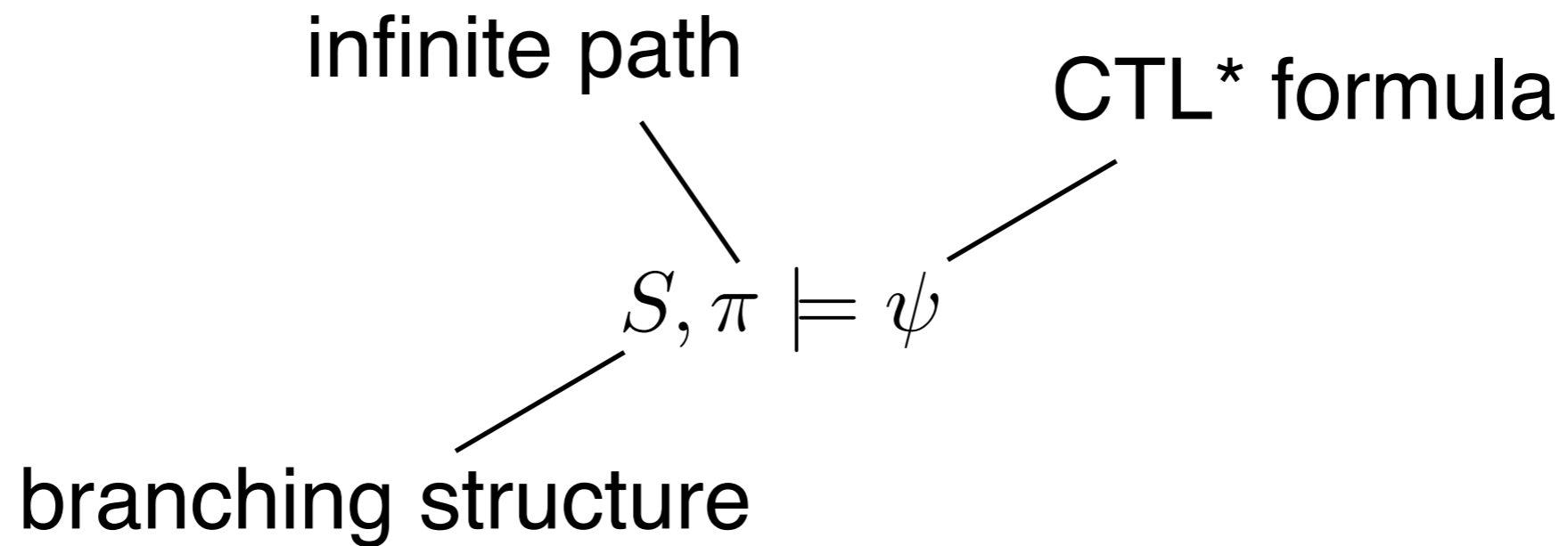
$$\pi = v_0 v_1 \dots \qquad \pi^k = v_k v_{k+1} \dots$$

$$\pi : \mathbb{N} \rightarrow V \qquad \pi^k : \mathbb{N} \rightarrow V$$

$$\pi^k(i) = \pi(k + i)$$



# CTL\*: satisfaction



# CTL\*: satisfaction

$$S, \pi \models \mathbf{tt}$$

$$S, \pi \models \neg\psi \quad \text{iff } S, \pi \not\models \psi$$

$$S, \pi \models \psi_0 \wedge \psi_1 \quad \text{iff } S, \pi \models \psi_0 \text{ and } S, \pi \models \psi_1$$

$$S, \pi \models \psi_0 \vee \psi_1 \quad \text{iff } S, \pi \models \psi_0 \text{ or } S, \pi \models \psi_1$$

$$S, \pi \models p \quad \text{iff } \pi(0) \in S(p)$$

$$S, \pi \models \mathbf{O}\psi \quad \text{iff } S, \pi^1 \models \psi$$

state ops

$$S, \pi \models \mathbf{F}\psi \quad \text{iff } \exists k \in \mathbb{N}. S, \pi^k \models \psi$$

$$S, \pi \models \mathbf{G}\psi \quad \text{iff } \forall k \in \mathbb{N}. S, \pi^k \models \psi$$

$$S, \pi \models \psi_0 \mathbf{U} \psi_1 \quad \text{iff } \exists k \in \mathbb{N}. S, \pi^k \models \psi_1 \text{ and } \forall i < k. S, \pi^i \models \psi_0$$

---

$$S, \pi \models \mathbf{E}\psi \quad \text{iff } \exists \pi'. \pi'(0) = \pi(0) \text{ and } S, \pi' \models \psi$$

path ops

$$S, \pi \models \mathbf{A}\psi \quad \text{iff } \forall \pi'. \pi'(0) = \pi(0) \text{ implies } S, \pi' \models \psi$$

# CTL\*: equivalent formulas

$$\psi_0 \equiv \psi_1 \quad \text{iff} \quad \forall S. \forall \pi. S, \pi \models \psi_0 \Leftrightarrow S, \pi \models \psi_1$$

$$A \psi \equiv \neg(E \neg \psi)$$

$$A A \psi \equiv A \psi$$

$$A E \psi \equiv E \psi$$

LTL formulas as CTL\* ones

$\psi$

$A \psi$

# Examples

$$E O \psi$$

analogous to HML formula  $\diamond\psi$

$$A G p$$

$p$  holds at any reachable state

$$E F p$$

$p$  holds at some reachable state

$$A F p$$

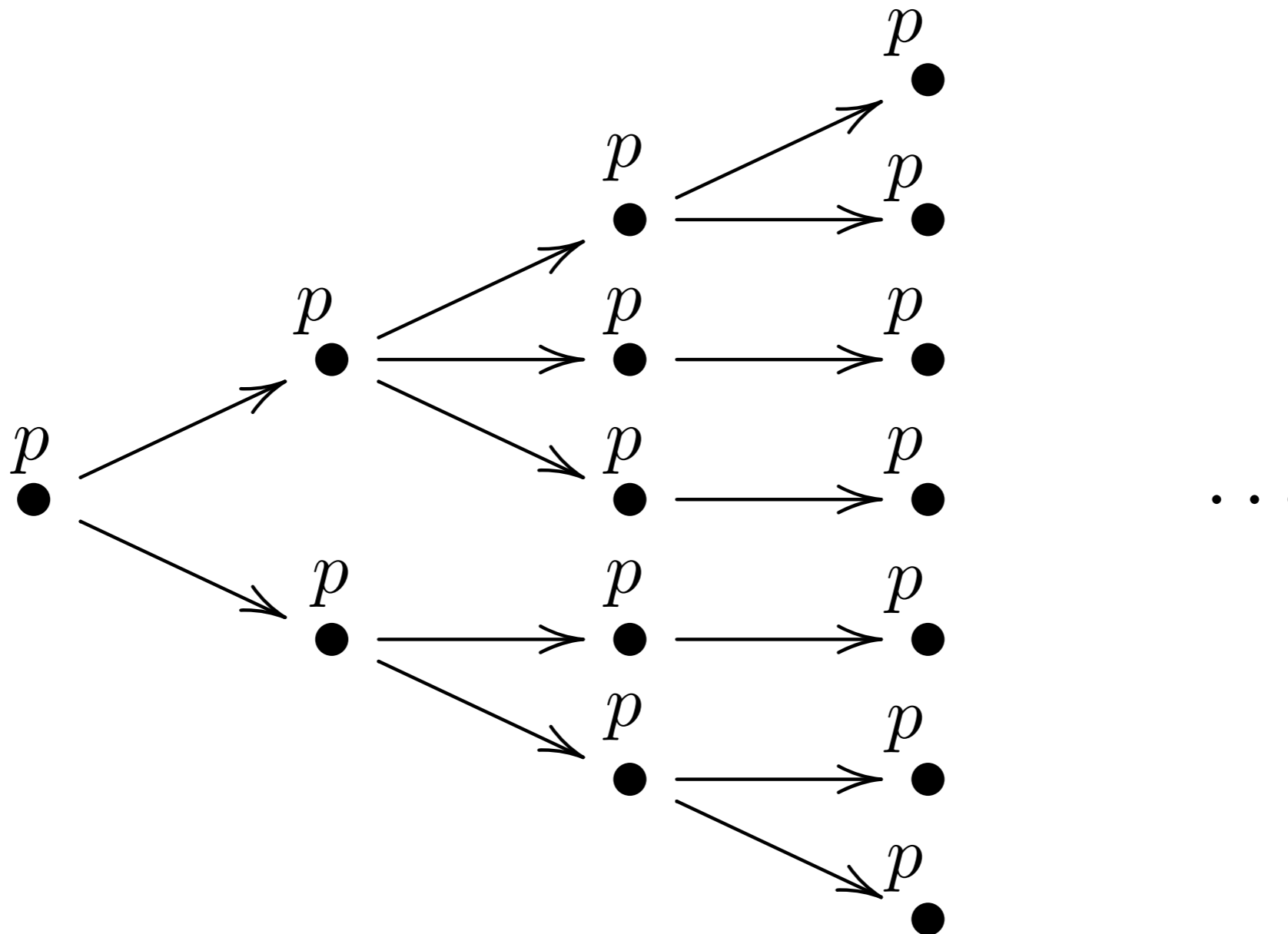
on every path there is a state where  $p$  holds

$$E (p U q)$$

there is a path where  $p$  holds until  $q$

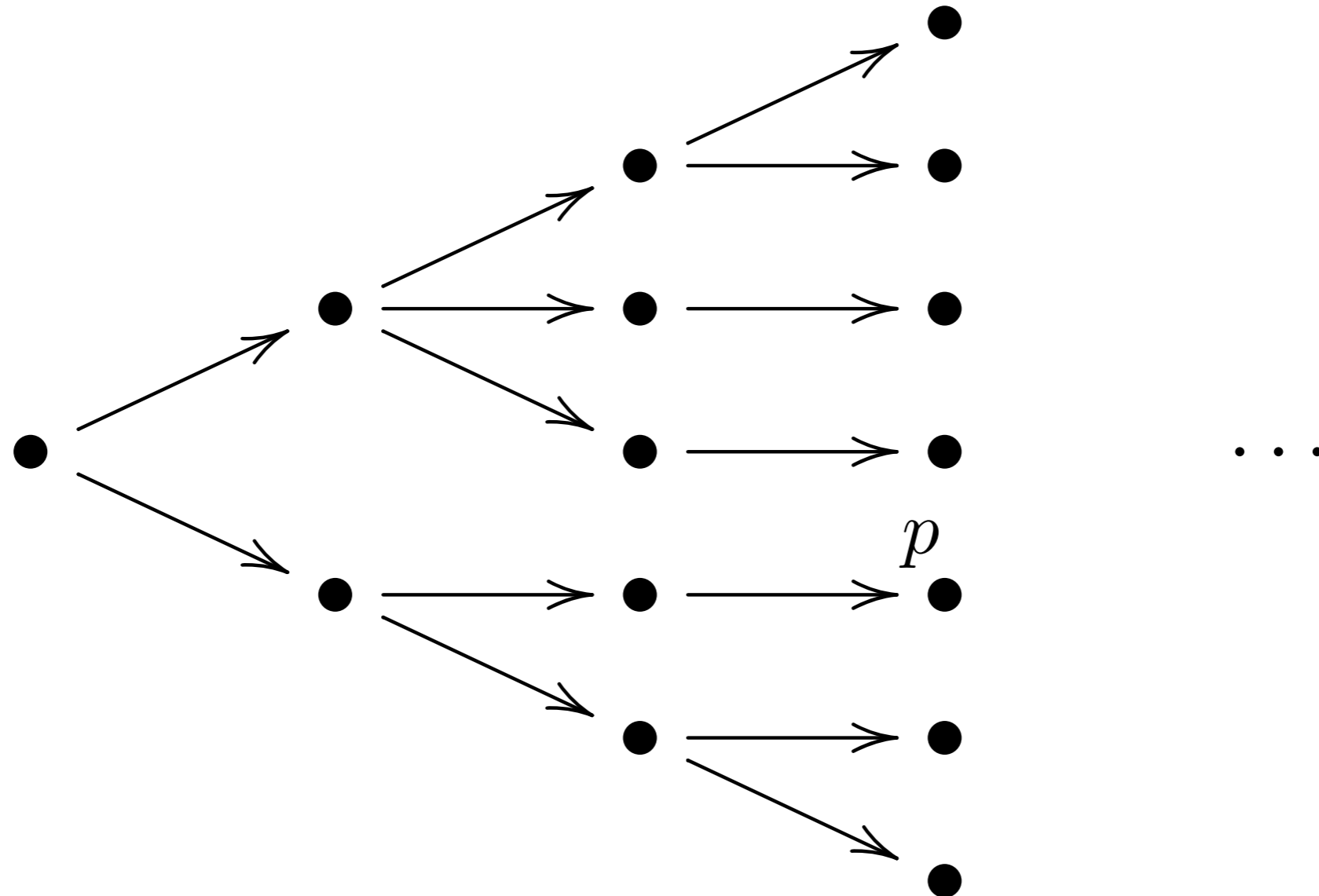
# Example

A G p



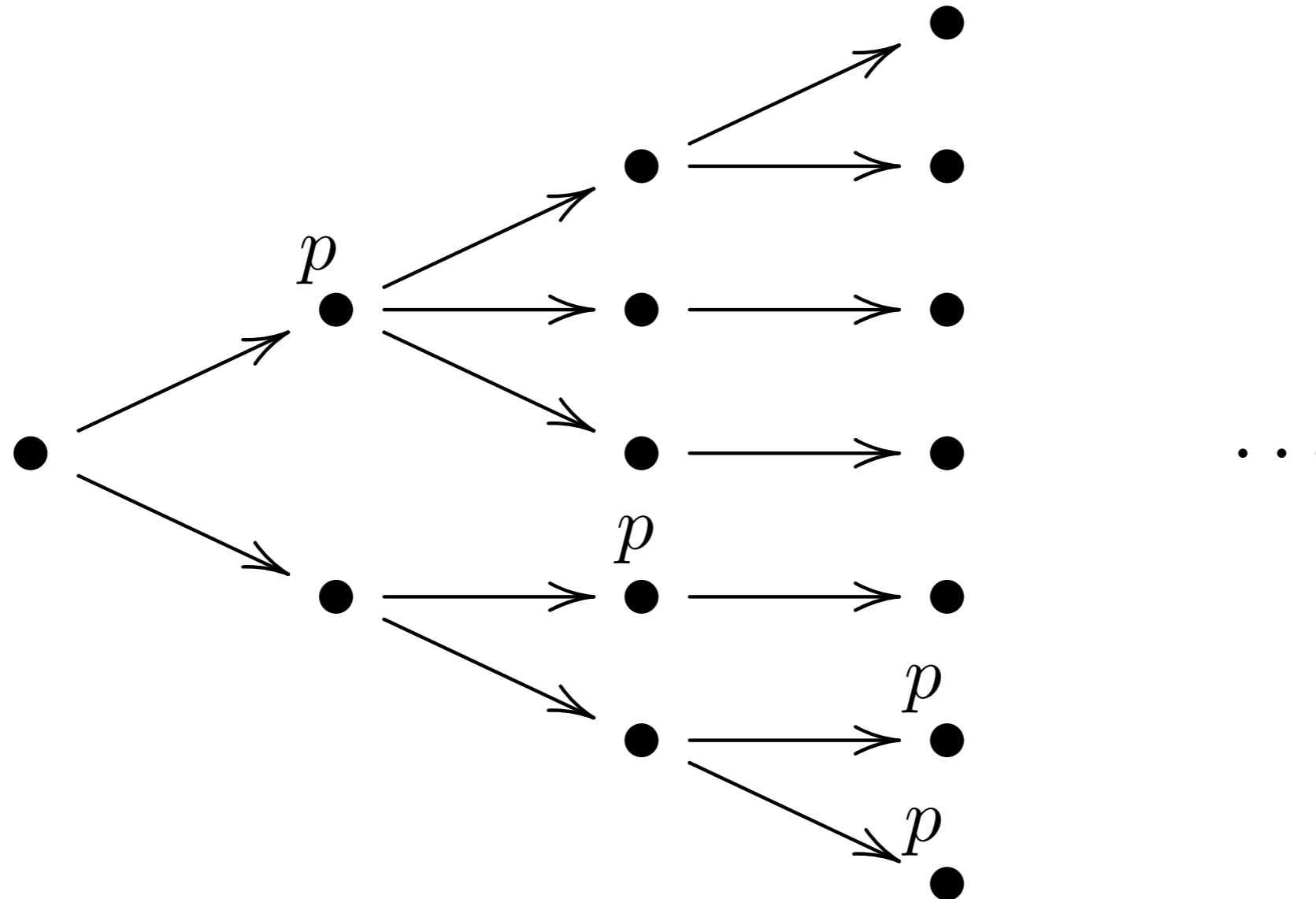
# Example

$E F p$



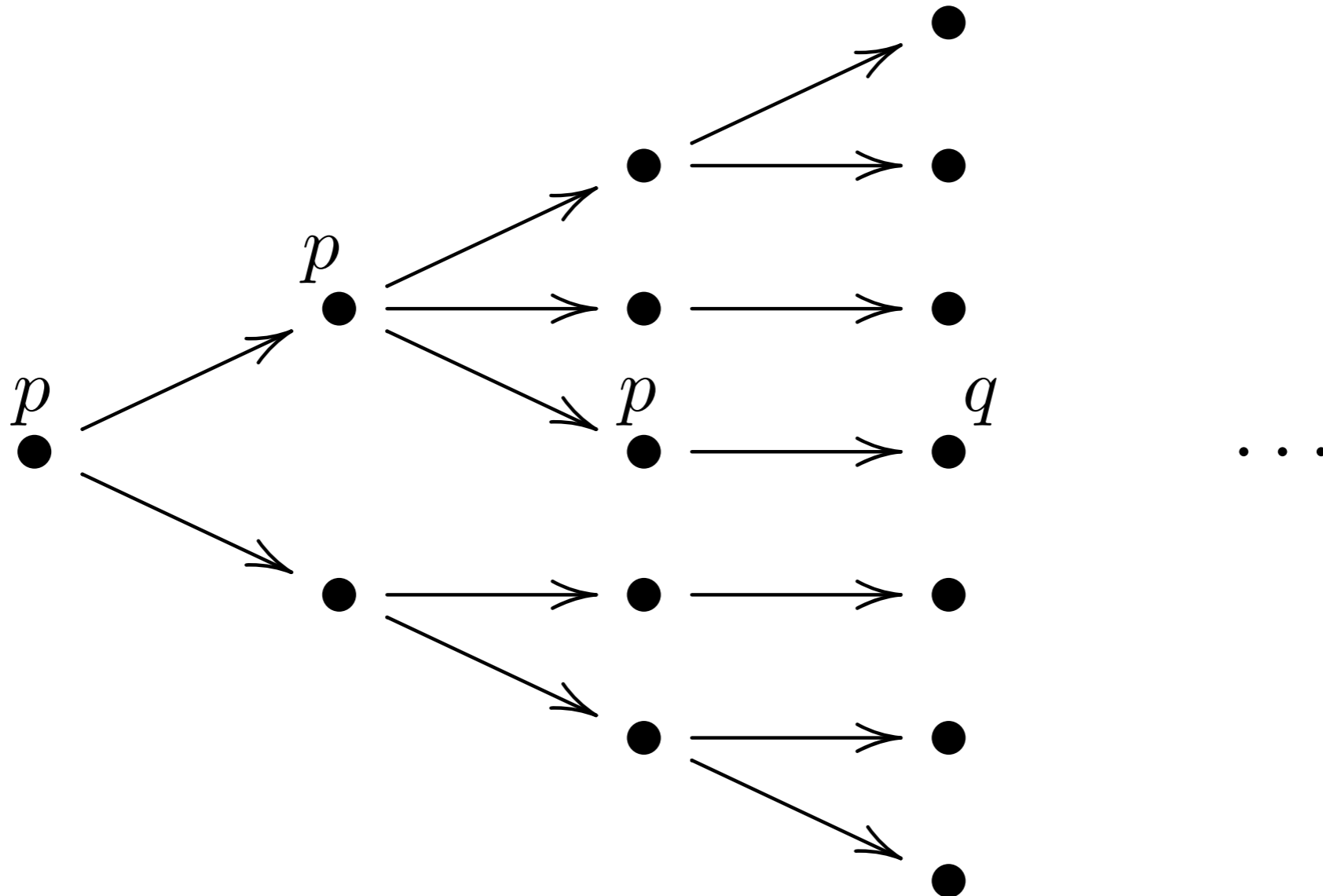
# Example

A F  $p$



# Example

$$E (p \cup q)$$





CTL

# CTL formulas

each path op (A/E) appears immediately before a linear op

each linear op (O/F/G/U) appears immediately after a path op

$E O \psi$

$E F \psi$

$E G \psi$

$E (\psi_0 U \psi_1)$

$A O \psi$

$A F \psi$

$A G \psi$

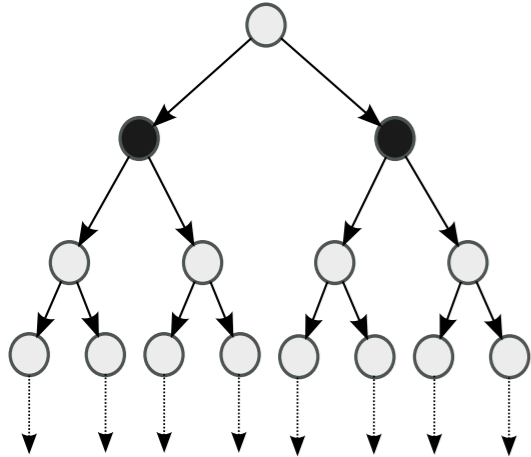
$A (\psi_0 U \psi_1)$

$A G F \psi$

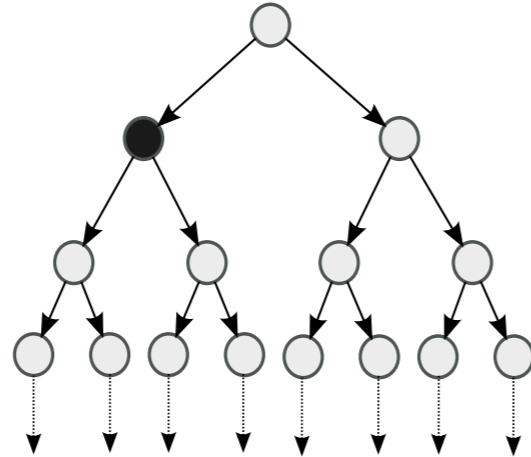
CTL\*, not CTL

# CTL formulas

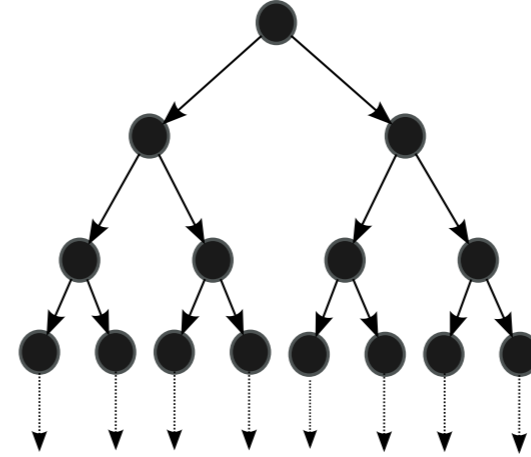
$AO \emptyset$



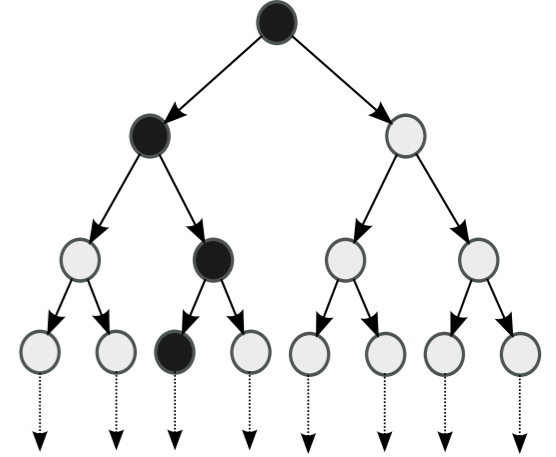
$EO \emptyset$



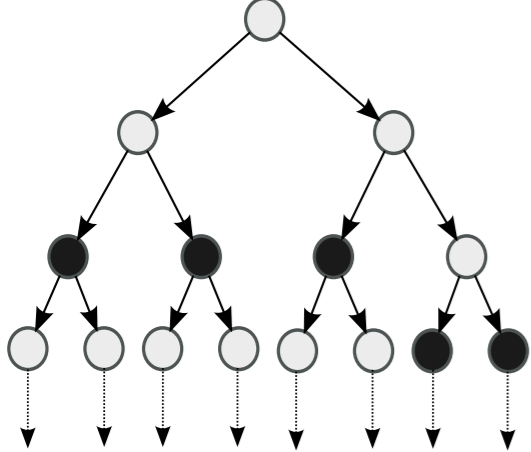
$AG \emptyset$



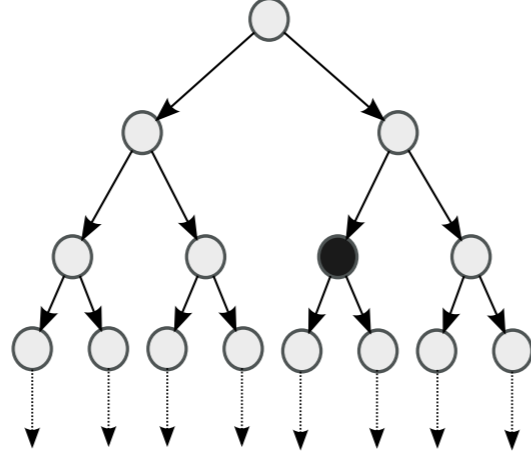
$EG \emptyset$



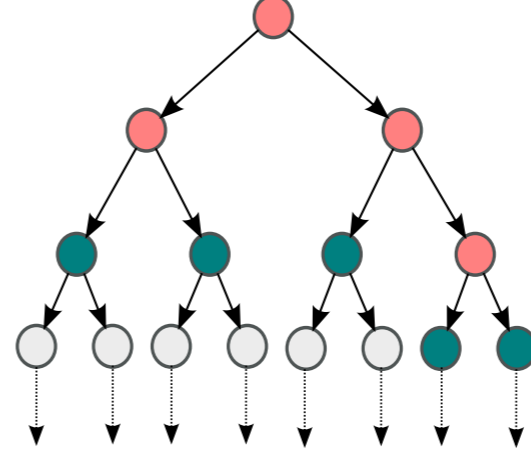
$AF \emptyset$



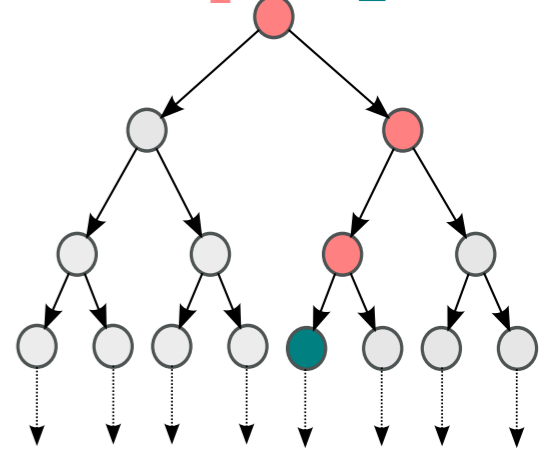
$EF \emptyset$



$A[\phi_1 U \phi_2]$



$E[\phi_1 U \phi_2]$



# CTL: minimal set of ops

$\neg \cdot \quad \cdot \vee \cdot \quad \text{EO} \cdot \quad \text{EG} \cdot \quad \text{E}(\cdot \text{U} \cdot)$

$$\text{AO } \psi \equiv \neg(\text{EO } \neg\psi)$$

$$\text{AF } \psi \equiv \neg(\text{EG } \neg\psi)$$

$$\text{EF } \psi \equiv \text{E}(\text{tt U } \psi)$$

$$\begin{aligned} \text{AG } \psi &\equiv \neg(\text{EF } \neg\psi) \\ &\equiv \neg\text{E}(\text{tt U } \neg\psi) \end{aligned}$$

$$\text{A } (\psi_0 \text{ U } \psi_1) \equiv \neg(\text{EG } \neg\psi_1 \vee \text{E}(\neg\psi_1 \text{ U } \neg(\psi_0 \vee \psi_1)))$$

# Expressiveness

