Branching-Time Logic

Bow-Yaw Wang

Institute of Information Science
Academia Sinica, Taiwan

November 16, 2021

Outline

1. Branching-time logic
   - Syntax of CTL
   - Semantics of computation tree logic
   - Practical patterns of specifications
   - Important equivalences between CTL formulae
   - Adequate sets of CTL connectives
2. CTL* and the expressive powers of LTL and CTL

Branching-time Logic I

- In LTL, we specify properties about paths.
- A model satisfies an LTL formula if all its paths from the initial state satisfy the formula.
- Sometimes, we may wish to consider properties about states.
  - Recall that non-blocking asks if a process can always enter the trying state when it is at the non-critical state.
- Branching-time logic allows us to specify such properties.
- We can ask if
  - all paths from a state satisfy certain properties; and
  - a path from a state satisfies certain properties.

Outline

1. Branching-time logic
   - Syntax of CTL
   - Semantics of computation tree logic
   - Practical patterns of specifications
   - Important equivalences between CTL formulae
   - Adequate sets of CTL connectives
2. CTL* and the expressive powers of LTL and CTL

Computation Tree Logic (CTL)

- Computation Tree Logic (CTL) is a branching-time logic.
- As usual, we fix a set $\texttt{Atoms}$ of atomic formulae $p, q, \ldots$.

Definition

Computation Tree Logic (CTL) has the following syntax:

$$\begin{array}{rcl} \phi & ::= & \bot \mid \top \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \implies \phi) \mid \mathsf{AX}\phi \mid \mathsf{EX}\phi \mid \\ & & \mathsf{AF}\phi \mid \mathsf{EF}\phi \mid \mathsf{AG}\phi \mid \mathsf{EG}\phi \mid \mathsf{A}[\phi\,\mathsf{U}\,\phi] \mid \mathsf{E}[\phi\,\mathsf{U}\,\phi] \end{array}$$

where $p \in \texttt{Atoms}$ is an atomic formula.

- Observe that CTL temporal connectives are either A (for all paths) or E (there exists) followed by an LTL temporal connectives $\mathsf{X}, \mathsf{F}, \mathsf{G}, \mathsf{U}$.

Convention and Examples

- By convention, binding powers of CTL connectives are:

strongest	$\rightarrow$	weakest
$\neg, \mathsf{AX}, \mathsf{EX}, \mathsf{AF}, \mathsf{EF}, \mathsf{AG}, \mathsf{EG}$	$\wedge, \vee$	$\Longrightarrow, \mathsf{AU}, \mathsf{EU}$

- Examples:

$$\mathsf{AG}(q \Longrightarrow \mathsf{EG}r) \quad \mathsf{EFE}[r\,\mathsf{U}\,q] \quad \mathsf{A}[p\,\mathsf{U}\,\mathsf{EF}r]$$
$$\mathsf{EFEG}p \Longrightarrow \mathsf{AF}r \quad \mathsf{A}[p_1\,\mathsf{U}\,\mathsf{A}[p_2\,\mathsf{U}\,p_3]] \quad \mathsf{E}[\mathsf{A}[p_1\,\mathsf{U}\,p_2]\,\mathsf{U}\,p_3]$$
$$\mathsf{AG}(p \Longrightarrow \mathsf{A}[p\,\mathsf{U}\,(\neg p \wedge \mathsf{A}[\neg p\,\mathsf{U}\,q])])$$

- Non-examples:

$$\cancel{\mathsf{EFG}r} \quad \cancel{\mathsf{A}\neg\mathsf{G}\neg p} \quad \cancel{\mathsf{F}[r\,\mathsf{U}\,q]}$$
$$\cancel{\mathsf{EF}(r\,\mathsf{U}\,q)} \quad \cancel{\mathsf{AEF}r} \quad \cancel{\mathsf{A}[(r\,\mathsf{U}\,q) \wedge (p\,\mathsf{U}\,r)]}$$

- A subformula of a CTL formula ϕ is any formula ψ whose parse tree is a subtree of ϕ's parse tree.

Outline

1. Branching-time logic
   - Syntax of CTL
   - Semantics of computation tree logic
   - Practical patterns of specifications
   - Important equivalences between CTL formulae
   - Adequate sets of CTL connectives
2. CTL* and the expressive powers of LTL and CTL

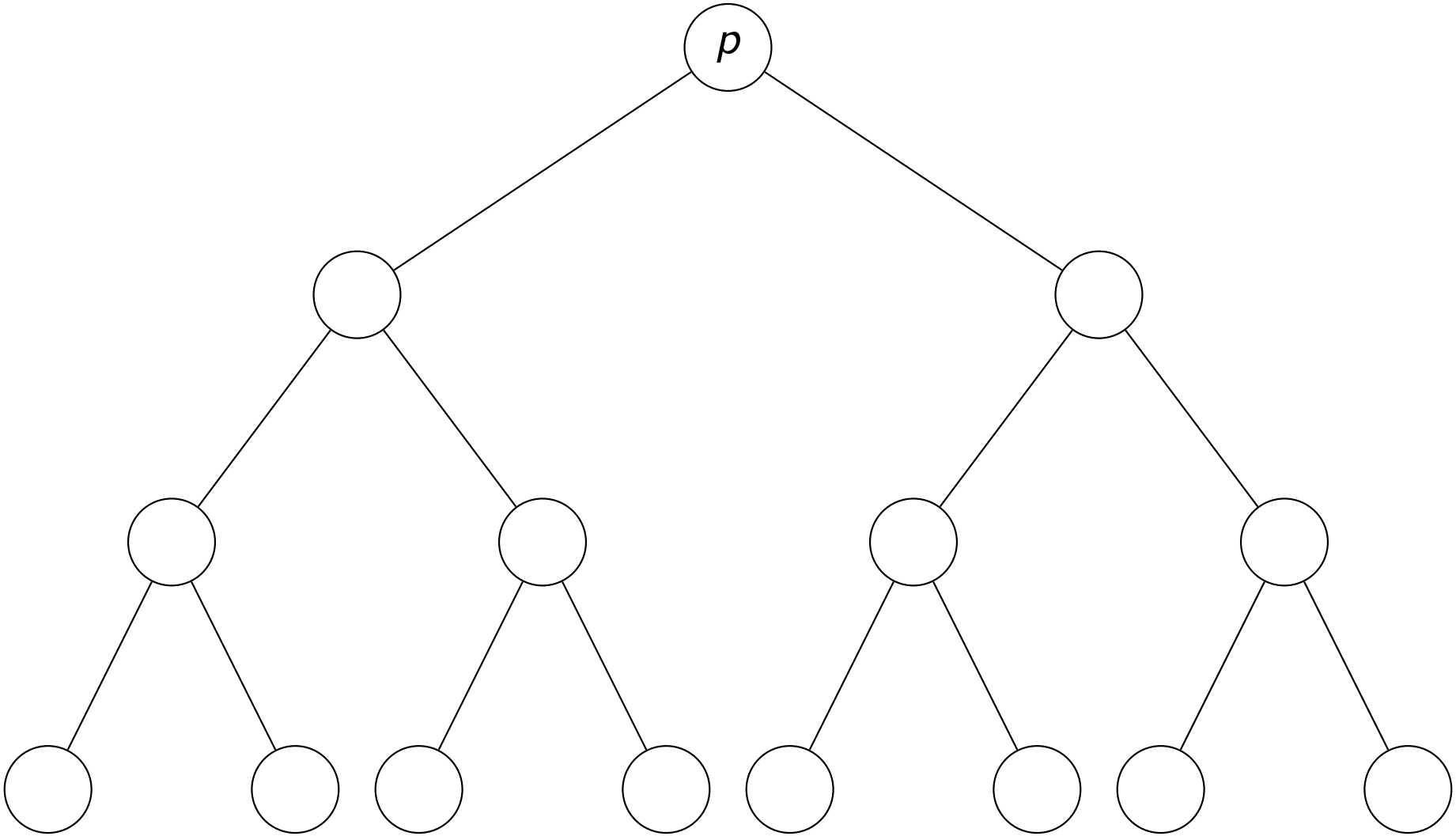
$\mathcal{M}, s \vDash \phi$

Definition

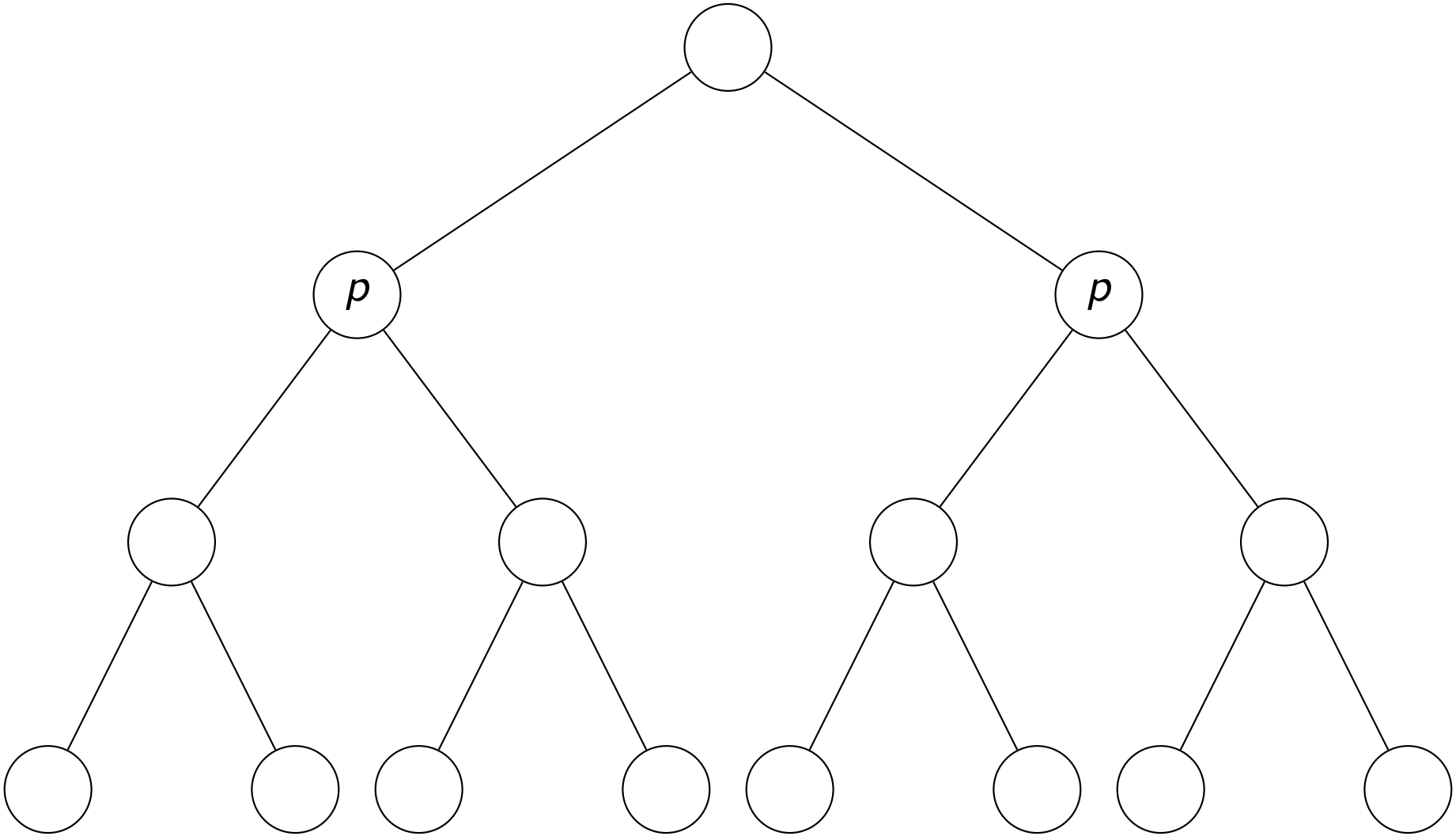
Let $\mathcal{M} = (S, \rightarrow, L)$ be a transition system, $s \in S$, and ϕ a CTL formula. $\underline{\mathcal{M}, s \vDash \phi}$ is defined as follows.

1. $\mathcal{M}, s \vDash \top$; $\mathcal{M}, s \nvDash \bot$; $\mathcal{M}, s \vDash p$ if $p \in L(s)$; $\mathcal{M}, s \vDash \neg\phi$ if $\mathcal{M}, s \nvDash \phi$;
2. $\mathcal{M}, s \vDash \phi \wedge \psi$ if $\mathcal{M}, s \vDash \phi$ and $\mathcal{M}, s \vDash \psi$; $\mathcal{M}, s \vDash \phi \vee \psi$ if $\mathcal{M}, s \vDash \phi$ or $\mathcal{M}, s \vDash \psi$;
3. $\mathcal{M}, s \vDash \phi \implies \psi$ if $\mathcal{M}, s \vDash \psi$ whenever $\mathcal{M}, s \vDash \phi$;
4. $\mathcal{M}, s \vDash \mathsf{AX}\phi$ if for all s' that $s \rightarrow s'$ we have $\mathcal{M}, s' \vDash \phi$;
5. $\mathcal{M}, s \vDash \mathsf{EX}\phi$ if for some s' that $s \rightarrow s'$ we have $\mathcal{M}, s' \vDash \phi$;
6. $\mathcal{M}, s \vDash \mathsf{AG}\phi$ if for all paths $s_0(= s) \rightarrow s_1 \rightarrow \cdots$ we have $\mathcal{M}, s_i \vDash \phi$ for every $i \geq 0$;
7. $\mathcal{M}, s \vDash \mathsf{EG}\phi$ if for some path $s_0(= s) \rightarrow s_1 \rightarrow \cdots$ we have $\mathcal{M}, s_i \vDash \phi$ for every $i \geq 0$;
8. $\mathcal{M}, s \vDash \mathsf{AF}\phi$ if for all paths $s_0(= s) \rightarrow s_1 \rightarrow \cdots$ we have $\mathcal{M}, s_i \vDash \phi$ for some $i \geq 0$;
9. $\mathcal{M}, s \vDash \mathsf{EF}\phi$ if for some path $s_0(= s) \rightarrow s_1 \rightarrow \cdots$ we have $\mathcal{M}, s_i \vDash \phi$ for some $i \geq 0$;
10. $\mathcal{M}, s \vDash \mathsf{A}[\phi \,\mathsf{U}\, \psi]$ if for all paths $s_0(= s) \rightarrow s_1 \rightarrow \cdots$ there is $i \geq 0$ that $\mathcal{M}, s_i \vDash \psi$ and for every $0 \leq j < i$ we have $\mathcal{M}, s_j \vDash \phi$;
11. $\mathcal{M}, s \vDash \mathsf{E}[\phi \,\mathsf{U}\, \psi]$ if for some path $s_0(= s) \rightarrow s_1 \rightarrow \cdots$ there is $i \geq 0$ that $\mathcal{M}, s_i \vDash \psi$ and for every $0 \leq j < i$ we have $\mathcal{M}, s_j \vDash \phi$.

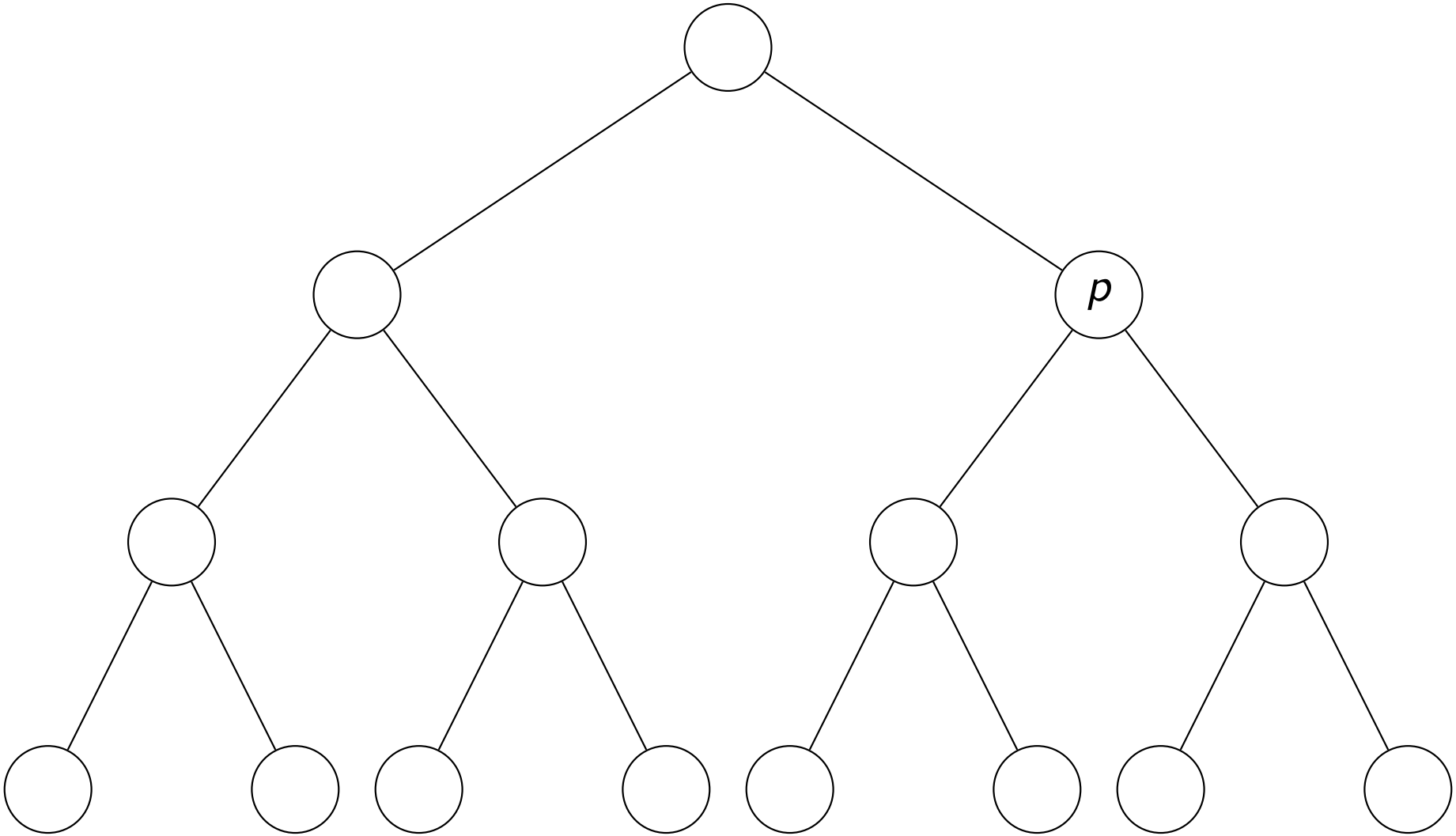
Visualization – $\mathcal{M}, s \vDash p$



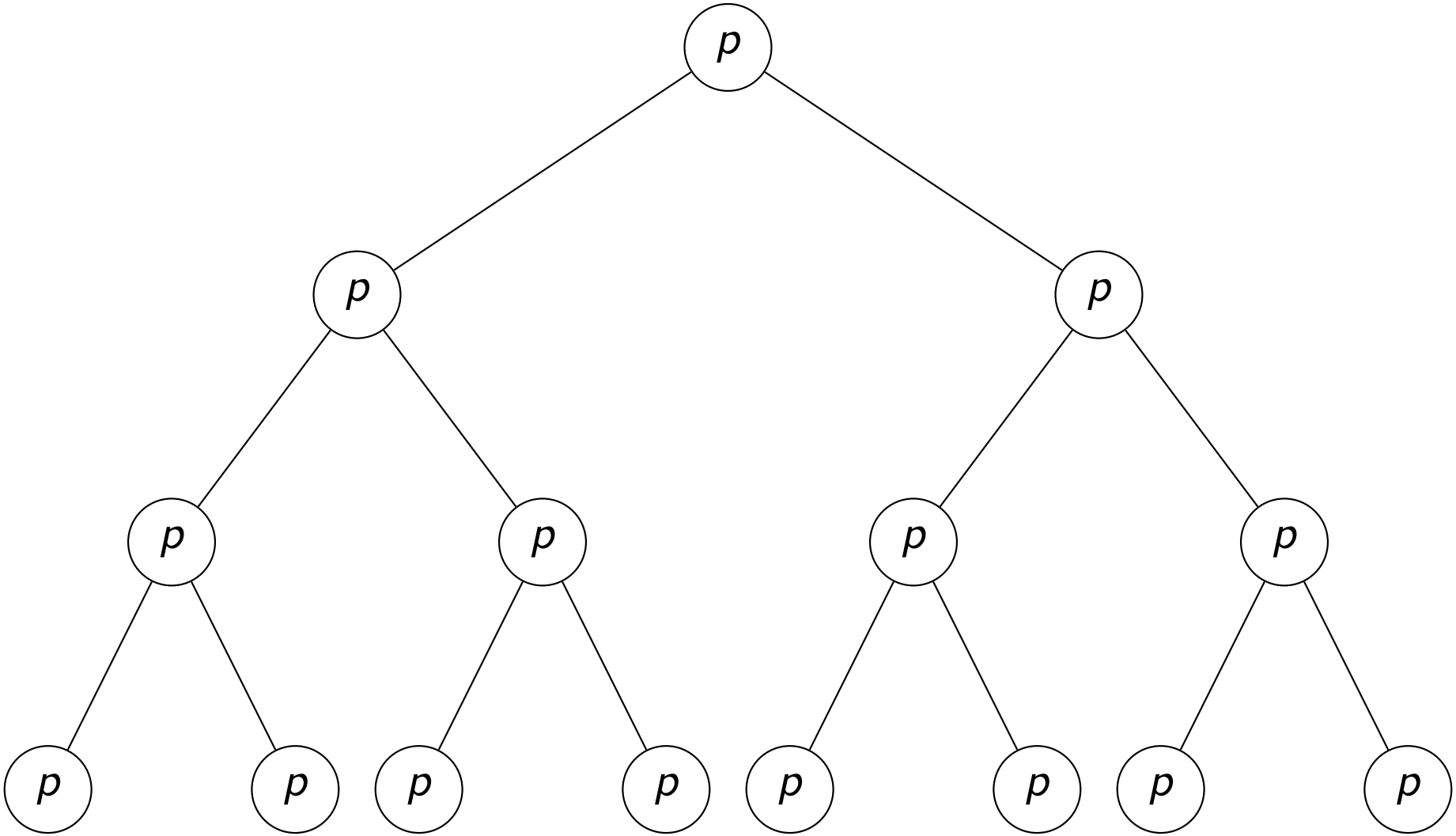
Visualization – $\mathcal{M}, s \vDash \mathsf{AX}p$



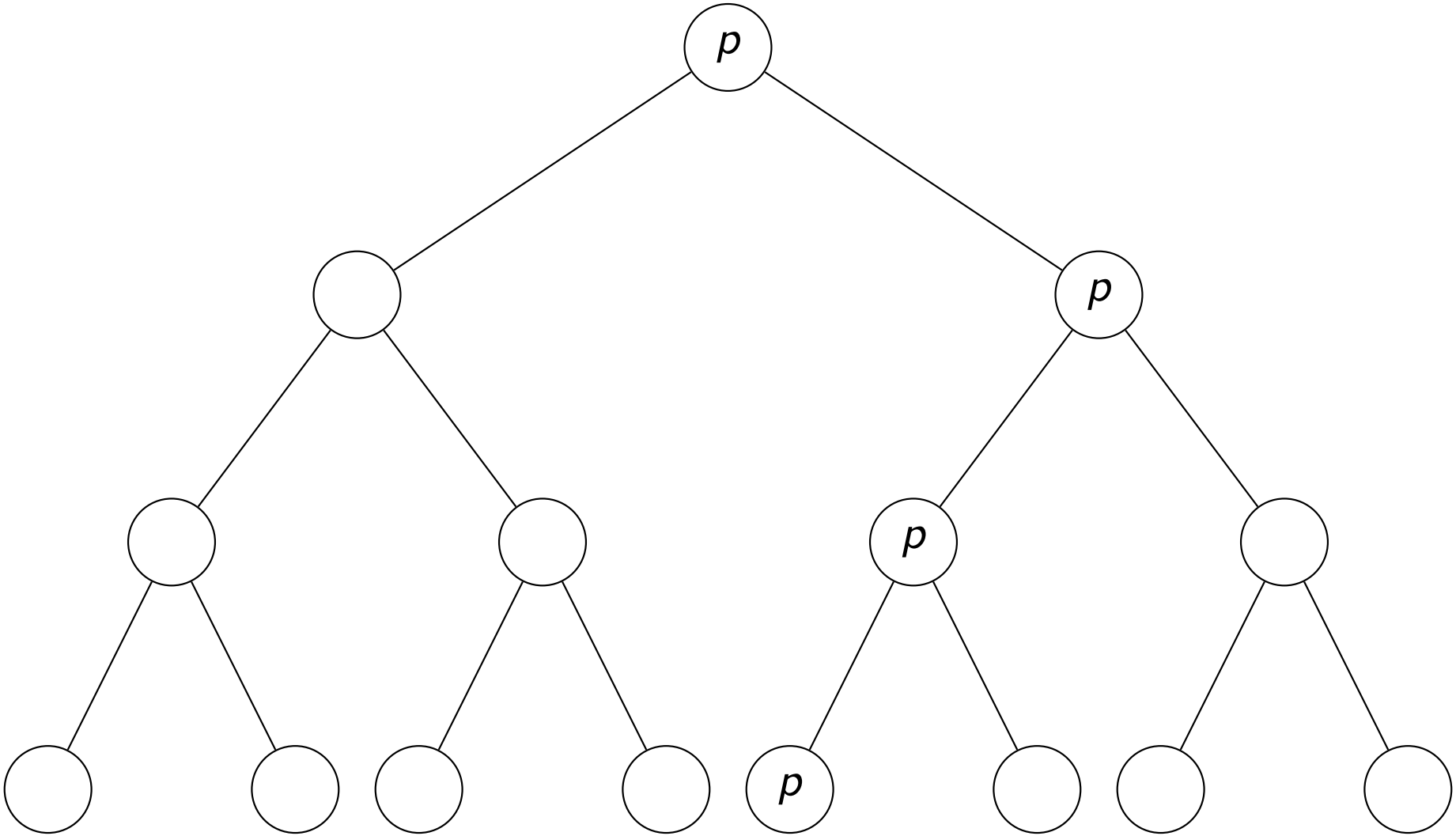
Visualization – $\mathcal{M}, s \vDash \mathsf{EX}p$



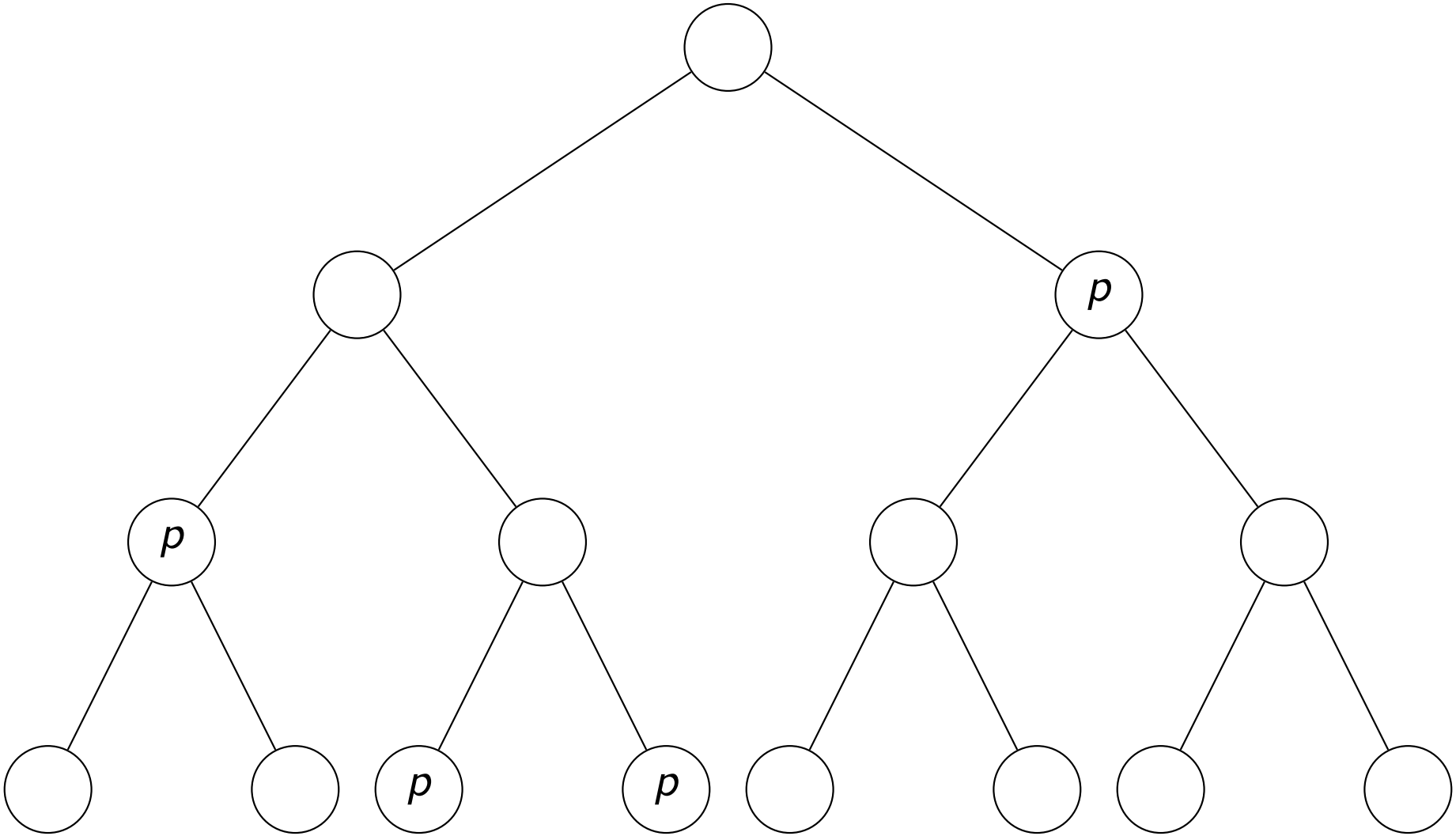
Visualization – $\mathcal{M}, s \models \mathsf{AG}p$



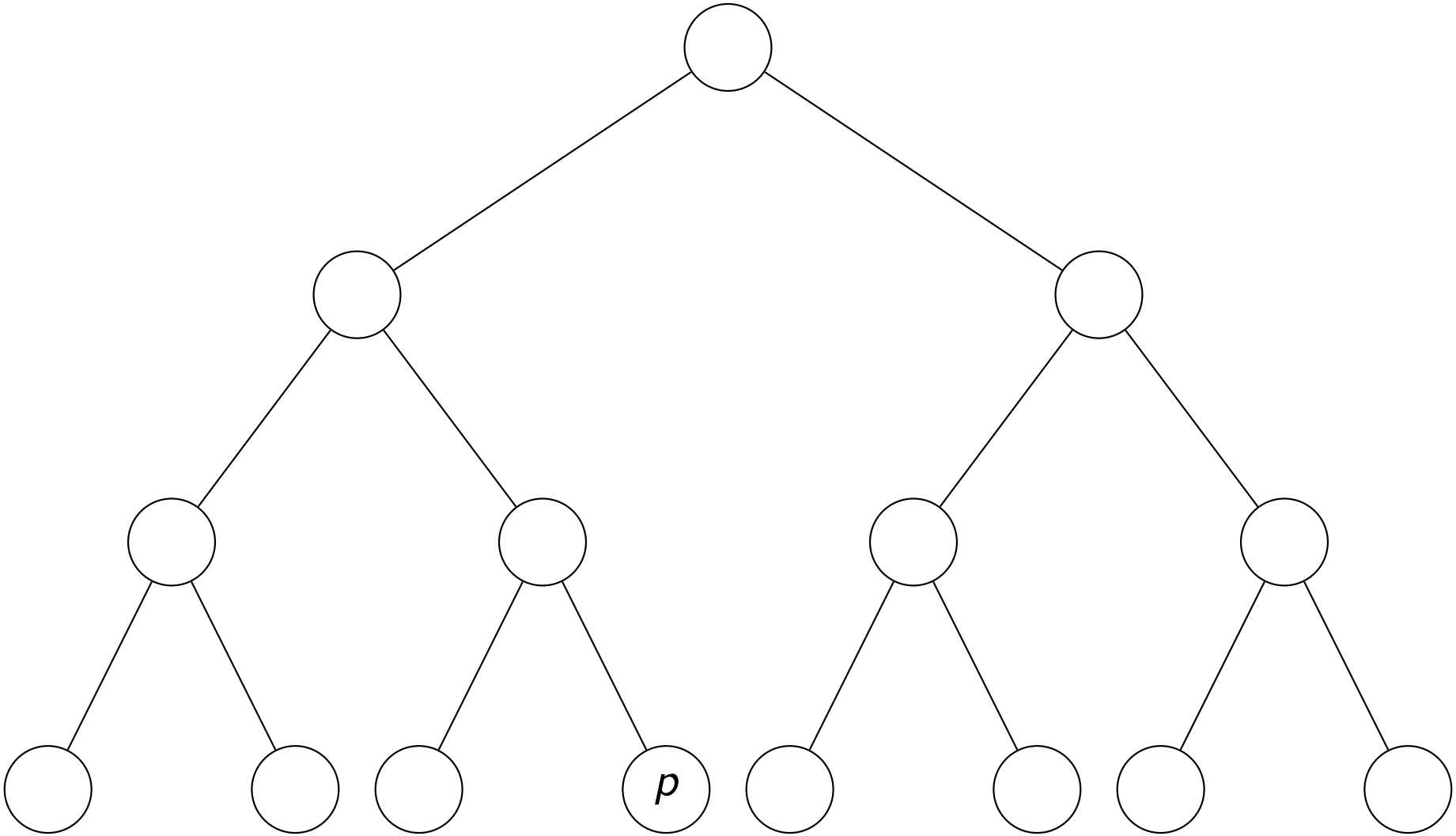
Visualization – $\mathcal{M}, s \models \mathsf{EG}p$



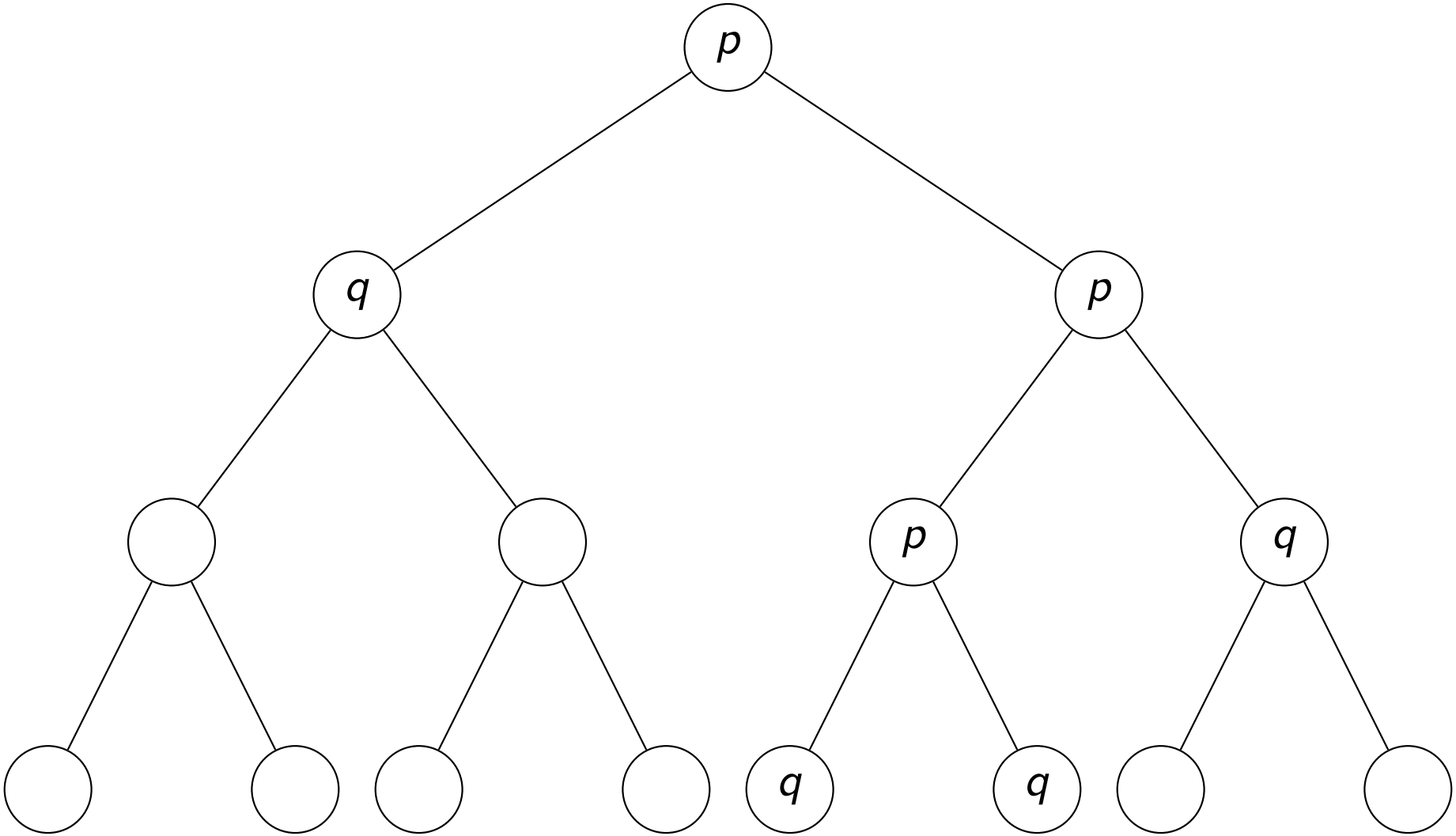
Visualization – $\mathcal{M}, s \models \mathsf{AF}p$



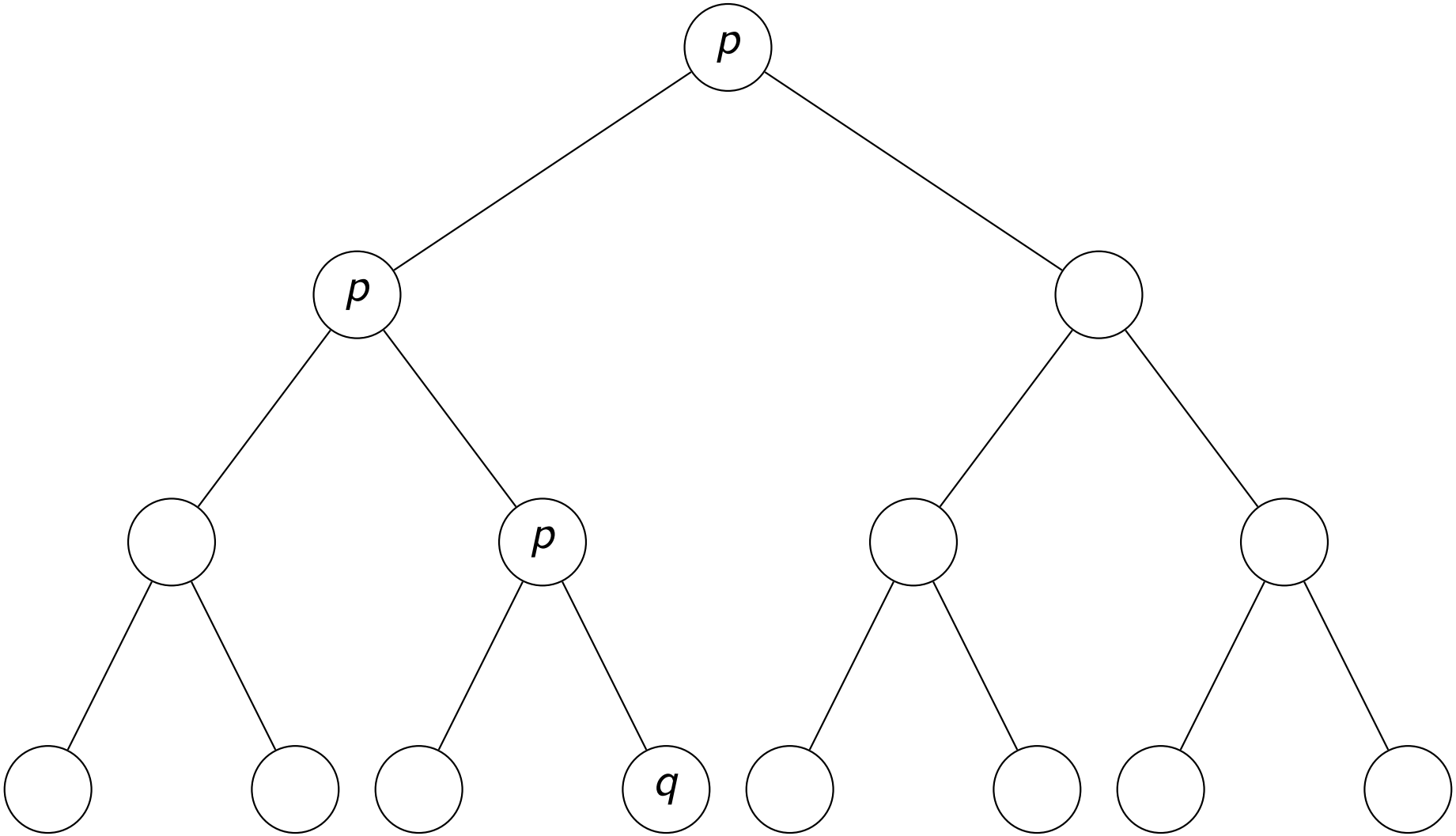
Visualization – $\mathcal{M}, s \vDash \mathsf{EF}p$



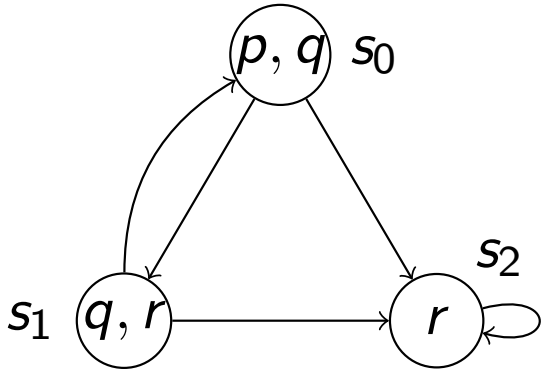
Visualization – $\mathcal{M}, s \vDash \mathsf{A}[p \,\mathsf{U}\, q]$



Visualization – $\mathcal{M}, s \vDash \mathsf{E}[p \,\mathsf{U}\, q]$

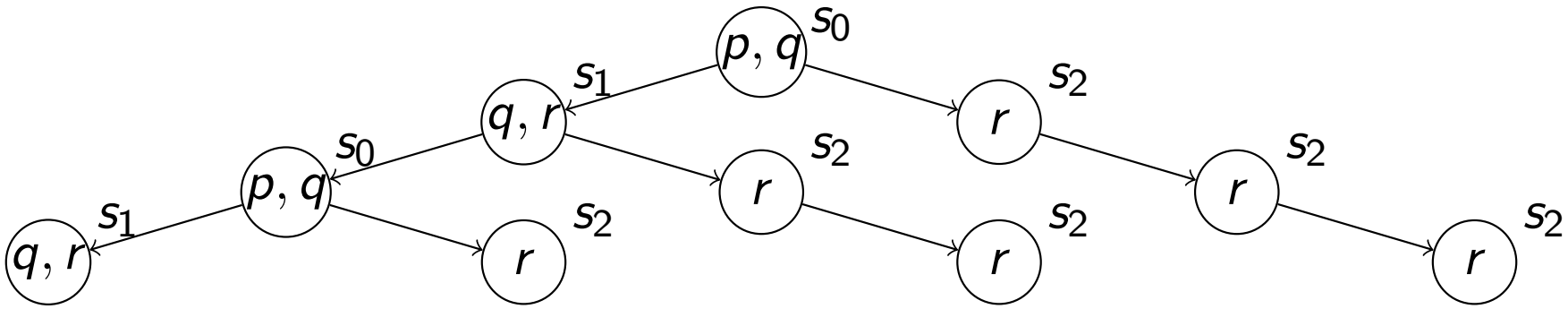


Example I



- Recall the transition system $\mathcal{M}$ above.
- We will give examples of $\mathcal{M}, s \models \phi$.
- To do so, we first "unroll" $\mathcal{M}$ from its initial state s_0.

Example II



- $\mathcal{M}, s_0 \models p \wedge q$; $\mathcal{M}, s_0 \models \neg r$; $\mathcal{M}, s_0 \models \top$;
- $\mathcal{M}, s_0 \models \mathsf{EX}(q \wedge r)$; $\mathcal{M}, s_0 \models \neg\mathsf{AX}(q \wedge r)$;
- $\mathcal{M}, s_0 \models \neg\mathsf{EF}(p \wedge r)$ since a state with p, r is not reachable from s_0;
- $\mathcal{M}, s_2 \models \mathsf{EG}r$ since $s_2 \to s_2 \to \cdots$;
- $\mathcal{M}, s_0 \models \mathsf{AF}r$ since r is always reachable from s_0;
- $\mathcal{M}, s_0 \models \mathsf{E}[(p \wedge q)\,\mathsf{U}\,r]$ since $s_0 \to s_2 \to s_2 \to \cdots$;
- $\mathcal{M}, s_0 \models \mathsf{A}[p\,\mathsf{U}\,r]$ since $s_0 \to s_1 \to \cdots$ and $s_0 \to s_2 \to \cdots$;
- $\mathcal{M}, s_0 \models \mathsf{AG}(p \vee q \vee r \implies \mathsf{EFEG}r)$.

Outline

1. Branching-time logic
   - Syntax of CTL
   - Semantics of computation tree logic
   - Practical patterns of specifications
   - Important equivalences between CTL formulae
   - Adequate sets of CTL connectives
2. CTL* and the expressive powers of LTL and CTL

Patterns of Specifications

- It is possible to get to a state where *started* holds but *ready* doesn't: $\mathsf{EF}(\mathit{started} \wedge \neg \mathit{ready})$;
- For any state, if a *requested* occurs, it will be *acknowledged* eventually: $\mathsf{AG}(\mathit{requested} \implies \mathsf{AF}\mathit{acknowledged})$;
- A process is *enabled* infinitely often on all paths: $\mathsf{AG}(\mathsf{AF}\mathit{enabled})$;
- A process will eventually be *stable* permanently: $\mathsf{AF}(\mathsf{AG}\mathit{stable})$;
- From any state it is possible to get to a *restart* state: $\mathsf{AG}(\mathsf{EF}\mathit{restart})$;
- Process P_1 can always request to enter its critical section: $\mathsf{AG}(n_1 \implies \mathsf{EX}t_1)$.
  - Recall that non-blocking is not expressible in LTL.
- Consider the following property: "if a process is *enabled* infinitely often, it is *running* infinitely often. This is not expressed by $\mathsf{AGAF}\mathit{enabled} \implies \mathsf{AGAF}\mathit{running}$.
  - In fact, this property is not expressible in CTL.
  - We can express the property by $\mathsf{GF}\mathit{enabled} \implies \mathsf{GF}\mathit{running}$ in LTL.

Outline

1. Branching-time logic
   - Syntax of CTL
   - Semantics of computation tree logic
   - Practical patterns of specifications
   - Important equivalences between CTL formulae
   - Adequate sets of CTL connectives
2. CTL* and the expressive powers of LTL and CTL

Semantic Equivalences

Definition

Let ϕ and ψ be CTL formulae. ϕ and ψ are <u>semantically equivalent</u> (written $\phi \equiv \psi$) if for all models $\mathcal{M}$ and all state s in $\mathcal{M}$, $\mathcal{M}, s \models \phi$ iff $\mathcal{M}, s \models \psi$.

- Since A is a universal quantifier and E an existential quantifier, it is not hard to see the following semantically equivalent formulae:

$$\neg \mathsf{AF}\phi \equiv \mathsf{EG}\neg\phi \qquad \neg \mathsf{EF}\phi \equiv \mathsf{AG}\neg\phi \qquad \neg \mathsf{AX}\phi \equiv \mathsf{EX}\neg\phi.$$

- Moreover, by similar arguments in LTL, we have

$$\mathsf{AF}\phi \equiv \mathsf{A}[\top \mathsf{U}\, \phi] \qquad \mathsf{EF}\phi \equiv \mathsf{E}[\top \mathsf{U}\, \phi].$$

- Finally, we have

$$\mathsf{AG}\phi \equiv \phi \wedge \mathsf{AXAG}\phi \qquad \mathsf{EG}\phi \equiv \phi \wedge \mathsf{EXEG}\phi$$
$$\mathsf{AF}\phi \equiv \phi \vee \mathsf{AXAF}\phi \qquad \mathsf{EF}\phi \equiv \phi \vee \mathsf{EXEF}\phi$$
$$\mathsf{A}[\phi \mathsf{U}\, \psi] \equiv \psi \vee (\phi \wedge \mathsf{AXA}[\phi \mathsf{U}\, \psi])$$
$$\mathsf{E}[\phi \mathsf{U}\, \psi] \equiv \psi \vee (\phi \wedge \mathsf{EXE}[\phi \mathsf{U}\, \psi]).$$

Outline

1. Branching-time logic
   - Syntax of CTL
   - Semantics of computation tree logic
   - Practical patterns of specifications
   - Important equivalences between CTL formulae
   - Adequate sets of CTL connectives
2. CTL* and the expressive powers of LTL and CTL

Adequate Sets of CTL Connectives I

- $\{\mathsf{AU}, \mathsf{EU}, \mathsf{AX}\}$ is an adequate set of CTL connectives.
  - Observe that

$$\mathsf{AX}\phi \equiv \neg\mathsf{EX}\neg\phi$$

$$\begin{array}{rclcrcl} \mathsf{AG}\phi & \equiv & \neg\mathsf{EF}\neg\phi & \qquad & \mathsf{EG}\phi & \equiv & \neg\mathsf{AF}\neg\phi \\ \mathsf{AF}\phi & \equiv & \mathsf{A}[\top\,\mathsf{U}\,\phi] & & \mathsf{EF}\phi & \equiv & \mathsf{E}[\top\,\mathsf{U}\,\phi]. \end{array}$$

- $\{\mathsf{EG}, \mathsf{EU}, \mathsf{EX}\}$ is an adequate set of CTL connectives.
  - Recall $\phi\,\mathsf{U}\,\psi \equiv \neg(\neg\psi\,\mathsf{U}\,(\neg\phi \wedge \neg\psi)) \wedge \mathsf{F}\psi$ in LTL.
  - Observe that $\mathsf{A}[\phi\,\mathsf{U}\,\psi] \equiv \neg(\mathsf{E}[\neg\psi\,\mathsf{U}\,(\neg\phi \wedge \neg\psi)] \vee \mathsf{EG}\neg\psi)$.

$$\begin{array}{rcl} \mathsf{A}[\phi\,\mathsf{U}\,\psi] & \equiv & \mathsf{A}[\neg(\neg\psi\,\mathsf{U}\,(\neg\phi \wedge \neg\psi)) \wedge \mathsf{F}\psi] \\ & \equiv & \neg\mathsf{E}\neg[\neg(\neg\psi\,\mathsf{U}\,(\neg\phi \wedge \neg\psi)) \wedge \mathsf{F}\psi] \\ & \equiv & \neg\mathsf{E}[(\neg\psi\,\mathsf{U}\,(\neg\phi \wedge \neg\psi)) \vee \mathsf{G}\neg\psi] \\ & \equiv & \neg(\mathsf{E}[\neg\psi\,\mathsf{U}\,(\neg\phi \wedge \neg\psi)] \vee \mathsf{EG}\neg\psi). \end{array}$$

  - Strictly speaking, we need CTL^* for the proof.

Adequate Sets of CTL Connectives II

- More generally, we have

Theorem

A set of CTL temporal connectives is adequate iff it contains at least one of $\{\mathsf{AX}, \mathsf{EX}\}$*, at least one of* $\{\mathsf{EG}, \mathsf{AF}, \mathsf{AU}\}$*, and* EU*.*

- We can also define $\mathsf{AR}, \mathsf{ER}, \mathsf{AW}$, and EW by EU:

$$\mathsf{A}[\phi \,\mathsf{R}\, \psi] \triangleq \neg\mathsf{E}[\neg\phi \,\mathsf{U}\, \neg\psi] \qquad \mathsf{E}[\phi \,\mathsf{R}\, \psi] \triangleq \neg\mathsf{A}[\neg\phi \,\mathsf{U}\, \neg\psi]$$

$$\mathsf{A}[\phi \,\mathsf{W}\, \psi] \triangleq \mathsf{A}[\psi \,\mathsf{R}\, (\phi \vee \psi)] \qquad \mathsf{E}[\phi \,\mathsf{W}\, \psi] \triangleq \mathsf{E}[\psi \,\mathsf{R}\, (\phi \vee \psi)]$$

Outline

LTL and CTL

- In LTL, we have temporal connectives $\mathsf{X}, \mathsf{F}, \mathsf{G}, \mathsf{U}$.
- In CTL, we have temporal connectives $\mathsf{AX}, \mathsf{EX}, \mathsf{AF}, \mathsf{EF}, \mathsf{AG}, \mathsf{EG}, \mathsf{AU}, \mathsf{EU}$.
- Observe that CTL temporal connectives are LTL connectives prefixed by path quantifiers A or E.
- Consider the property: for some path, if p occurs eventually, then q occurs eventually.
- Using LTL, one would write $\mathsf{F}p \implies \mathsf{F}q$.
  - But LTL considers all paths from a specified state.
- Using CTL, one would write $\mathsf{EF}p \implies \mathsf{EF}q$.
  - But this is not what we want either.
- How about $\mathsf{E}(\mathsf{F}p \implies \mathsf{F}q)$?
  - But this is not in LTL nor CTL!
- We need a more expressive logic.

The Syntax of CTL*

Definition

CTL* has the following syntax:

- <u>state formulae</u>

$$\phi ::= \top \mid p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid \mathsf{A}[\alpha] \mid \mathsf{E}[\alpha]$$

  where $p \in \mathtt{Atoms}$ is an atomic formula

- <u>path formulae</u>

$$\alpha ::= \phi \mid (\neg\alpha) \mid (\alpha \wedge \alpha) \mid (\alpha \,\mathsf{U}\, \alpha) \mid (\mathsf{G}\alpha) \mid (\mathsf{F}\alpha) \mid (\mathsf{X}\alpha)$$

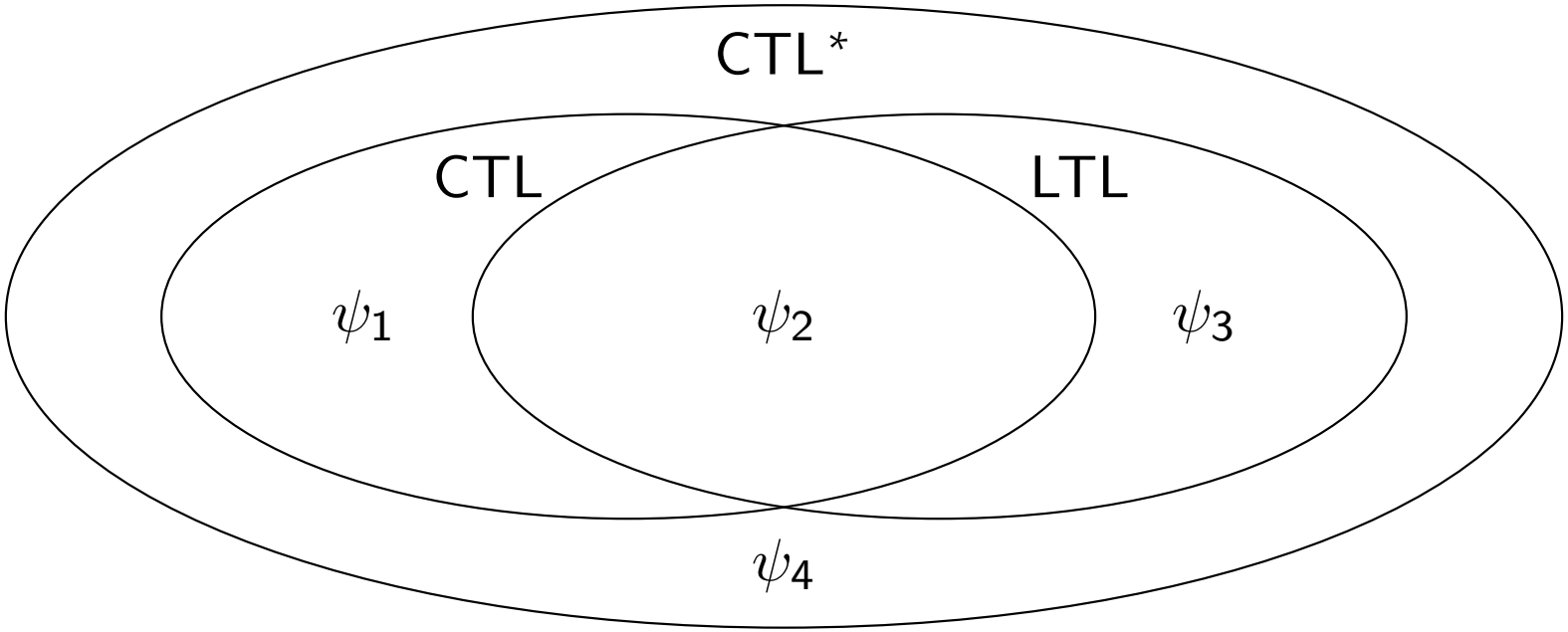
LTL, CTL, and CTL*

- An LTL formula is in fact a CTL* path formula.
- Since we consider all paths in LTL, an LTL formula α is in fact the CTL* state formula $\mathsf{A}[\alpha]$.
- We obtain CTL by restricting path formulae to

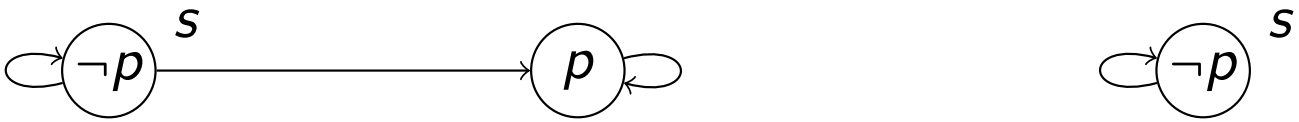
$$\alpha ::= (\phi \,\mathsf{U}\, \phi) \mid (\mathsf{G}\phi) \mid (\mathsf{F}\phi) \mid (\mathsf{X}\phi)$$

- Hence both LTL and CTL are subclasses of CTL*.

Expressive Powers of LTL, CTL, and CTL*



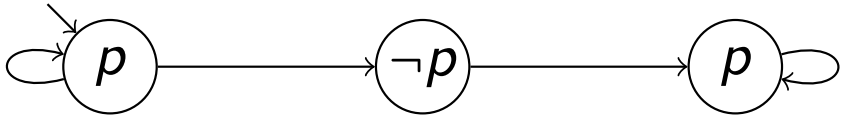
- $\psi_1 = \mathsf{AGEF}p$.



- $\psi_2 = \mathsf{AG}(p \implies \mathsf{AF}q)(\mathsf{CTL}) = \mathsf{G}(p \implies \mathsf{F}q)(\mathsf{LTL})$.
- $\psi_3 = \mathsf{GF}p \implies \mathsf{F}q$.
- $\psi_4 = \mathsf{E}[\mathsf{GF}p]$.

More Examples

- FGp and AFAGp are not equivalent.



- XF$p \equiv$ FXp (LTL) and AXAFp (CTL) are equivalent. But AFAX$p \not\equiv$ AXAFp.
- So, between LTL and CTL, which one is better?
  - In 1980's, many debates (and papers) discussed the question.
  - Now, people generally believe they are just different: one is not better than the other.
  - Try to compare LTL and CTL.