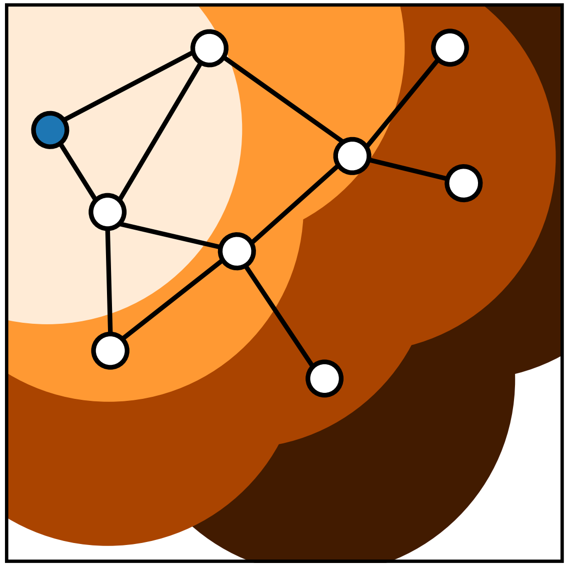
Discrete Event Systems
Verification of Finite Automata (Part 2)



Lana Josipović
Digital Systems and Design Automation Group
dynamo.ethz.ch

ETH Zurich (D-ITET)

December 7, 2023

Most materials from Lothar Thiele and Romain Jacob

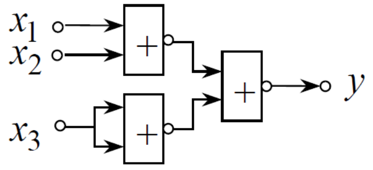
Last week in

Discrete Event Systems

Verification Scenarios

Example

$y = (x_1 + x_2) \cdot x_3$



Comparison of specification and implementation

reference system → data structure

system under test → data structure

→ comparison

Proving properties

"The device can always be switched off."

property → fixed-point calculation

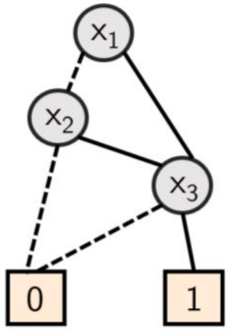
system under test → data structure → fixed-point calculation

Comparison using BDDs

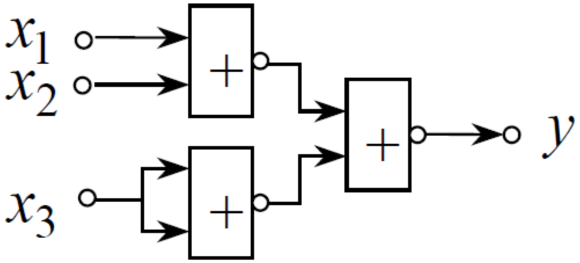
- Boolean (combinatorial) circuits: Compare specification and implementation, or compare two implementations.
- Method:
  - Representation of the two systems in ROBDDs, e.g., by applying the ***APPLY*** operator repeatedly.
  - Compare the structures of the ROBDDs.
- Example:

$$y = (x_1 + x_2) \cdot x_3$$

APPLY

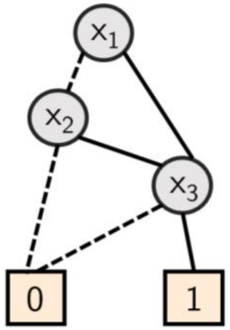


compare



$$y = \overline{\overline{x_1 + x_2} + \overline{x_3 + x_3}}$$

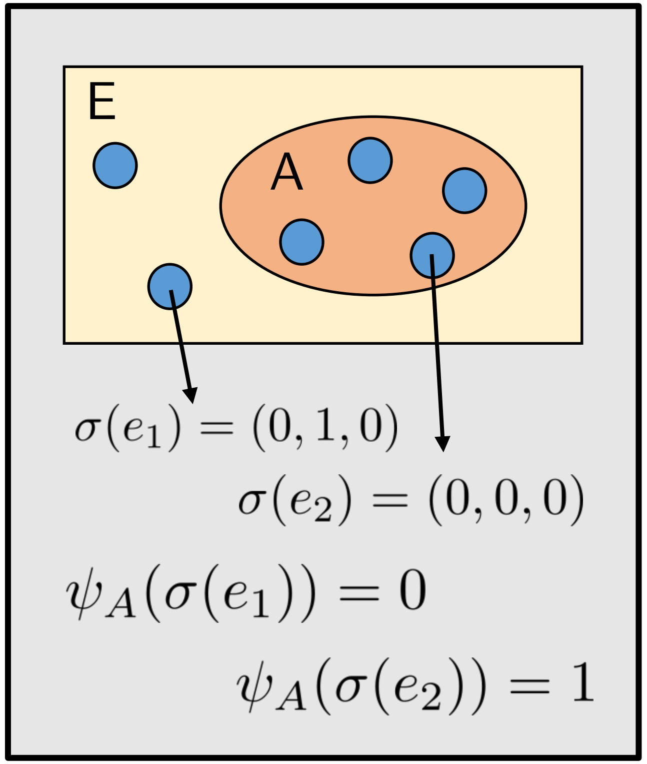
APPLY



Sets and Relations

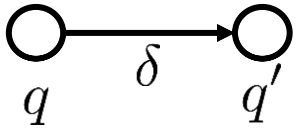
characteristic function of subset A

- Representation of a subset $A \subseteq E$:
  - Binary encoding $\sigma(e)$ of all elements $e \in E$
  - Subset A is represented by $a \in A \Leftrightarrow \psi_A(\sigma(a))$



- Relation function: describe state transitions

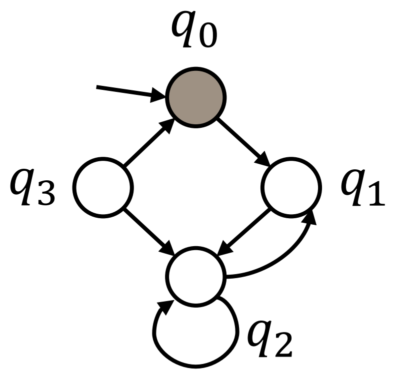
$$\psi_\delta(\sigma(q), \sigma(q')) = \psi_\delta(q, q')$$



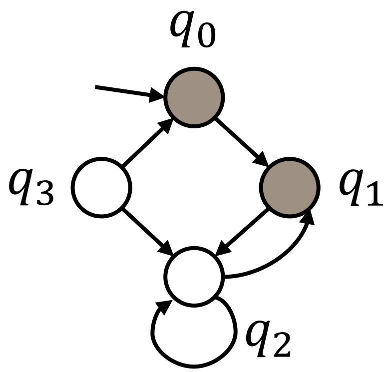
Reachability of States

- Problem: Is a state $q \in Q$ reachable by a sequence of state transitions?
- Method:
  - Represent set of states and the transformation relation as ROBDDs.
  - Use these representations to transform from one set of states to another. Set Q_i corresponds to the set of states reachable after i transitions.
  - Iterate the transformation until a fixed-point is reached, i.e., until the set of states does not change anymore (steady-state).
- Example:

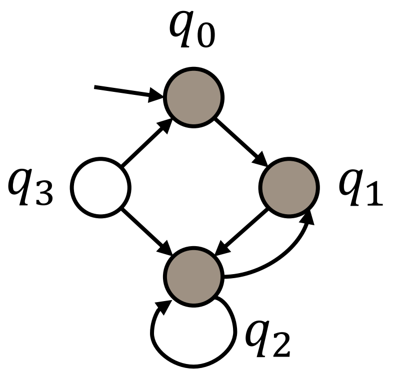
$Q_0 = \{q_0\}$



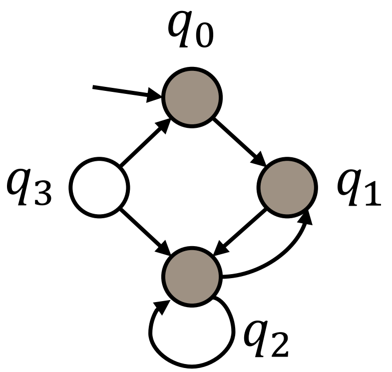
$Q_1 = \{q_0, q_1\}$



$Q_2 = \{q_0, q_1, q_2\}$



$Q_3 = \{q_0, q_1, q_2\}$



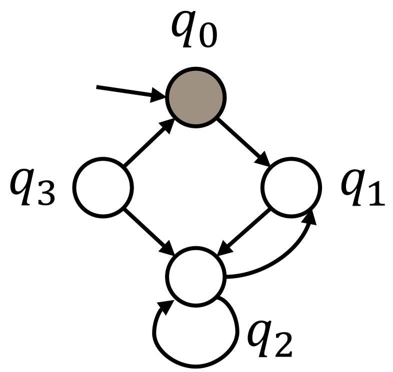
Reachability of States

State encoding — Characteristic function: 1 if state in set, 0 otherwise

$\sigma(q)$	x_1	x_0	ψ_{Q_0}	ψ_{Q_1}	ψ_{Q_2}	ψ_{Q_3}
q_0	0	0	1	1	1	1
q_1	0	1	0	1	1	1
q_2	1	0	0	0	1	1
q_3	1	1	0	0	0	0

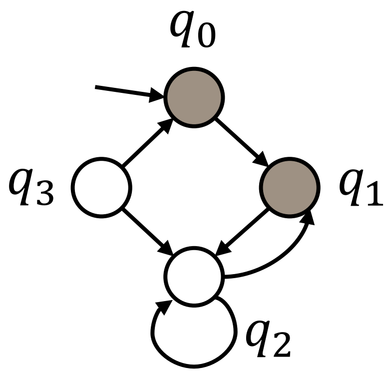
$\psi_{Q_0}(q) = \overline{x_1} \cdot \overline{x_0}$

$Q_0 = \{q_0\}$



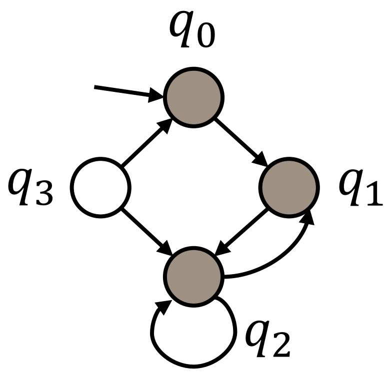
$\psi_{Q_1}(q) = \overline{x_1}$

$Q_1 = \{q_0, q_1\}$



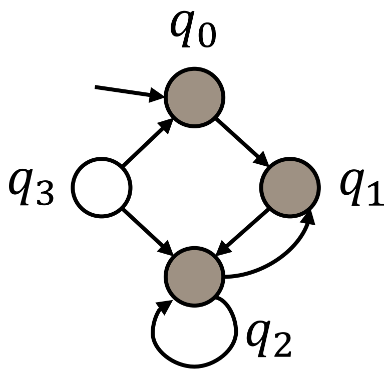
$\psi_{Q_2}(q) = \overline{x_1} + \overline{x_0}$

$Q_2 = \{q_0, q_1, q_2\}$



$\psi_{Q_3}(q) = \overline{x_1} + \overline{x_0}$

$Q_3 = \{q_0, q_1, q_2\}$



This week in
Discrete Event Systems

	Efficient state representation	- Set of states as Boolean function - Binary Decision Diagram representation
	Computing reachability	- Leverage efficient state representation - Explore successor sets of states
Today	Proving properties	- Temporal logic (CTL) - Encoding as reachability problem

Temporal Logic

- Verify properties of a finite automaton, for example
  - Can we always reset the automaton?
  - Is every request followed by an acknowledgement?
  - Are both outputs always equivalent?

Formula	Examples
Atomic proposition	The printer is busy. The light is on.
Boolean logic	$\phi_1 + \phi_2$; $\neg\phi_1$

Temporal Logic

- Verify properties of a finite automaton, for example
  - Can we always reset the automaton?
  - Is every request followed by an acknowledgement?
  - Are both outputs always equivalent?

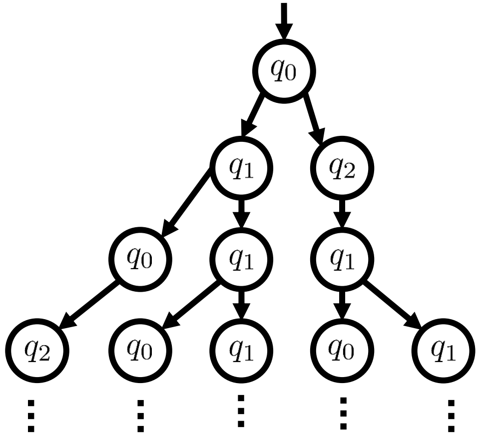
Formula	Examples
Atomic proposition	The printer is busy. The light is on.
Boolean logic	$\phi_1 + \phi_2$; $\neg\phi_1$
CTL logic	EX ϕ_1

- Specification of the query in a formula of temporal logic.
- We use a simple form called Computation Tree Logic (CTL).

There exists other logics (e.g. LTL, CTL*)

- Let us start with a minimal set of operators.
  - Any atomic proposition is a CTL formula.
  - CTL formula are constructed by composition of other CTL formula.

Formulation of CTL properties



Based on atomic propositions (ϕ) and quantifiers

$A\phi$	→ «**All** ϕ»,	ϕ holds on all paths	Quantifiers over paths
$E\phi$	→ «**E**xists ϕ»,	ϕ holds on at least one path	
$X\phi$	→ «Ne**X**t ϕ»,	ϕ holds on the next state	Path-specific quantifiers
$F\phi$	→ «**F**inally ϕ»,	ϕ holds at some state along the path	
$G\phi$	→ «**G**lobally ϕ»,	ϕ holds on all states along the path	
$\phi_1 U \phi_2$	→ «ϕ_1**U**ntil ϕ_2»,	ϕ_1 holds until ϕ_2 holds implies that ϕ_2 has to hold eventually	

CTL quantifiers work in pairs: we need one of each! {A,E} {X,F,G,U}ϕ

Over paths:	Path-specific:
$A\phi$ → **All** ϕ	$X\phi$ → Ne**X**t ϕ
$E\phi$ → **Exists** ϕ	$F\phi$ → **Finally** ϕ
	$G\phi$ → **Globally** ϕ
	$\phi_1 U \phi_2$ → ϕ_1 **Until** ϕ_2

Formulation of CTL properties

Based on atomic propositions (ϕ) and quantifiers

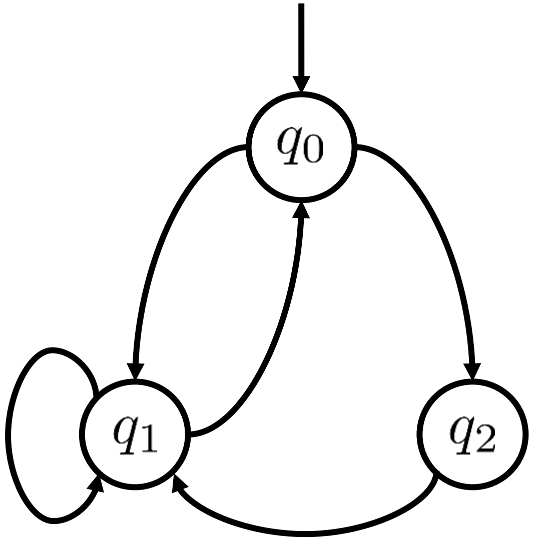
$A\phi$	→ «**A**ll ϕ»,	ϕ holds on all paths	Quantifiers over paths
$E\phi$	→ «**E**xists ϕ»,	ϕ holds on at least one path	
$X\phi$	→ «Ne**X**t ϕ»,	ϕ holds on the next state	Path-specific quantifiers
$F\phi$	→ «**F**inally ϕ»,	ϕ holds at some state along the path	
$G\phi$	→ «**G**lobally ϕ»,	ϕ holds on all states along the path	
$\phi_1 U \phi_2$	→ «ϕ_1**U**ntil ϕ_2»,	ϕ_1 holds until ϕ_2 holds implies that ϕ_2 has to hold eventually	

CTL quantifiers work in pairs: we need one of each! $\{A,E\}\ \{X,F,G,U\}\phi$

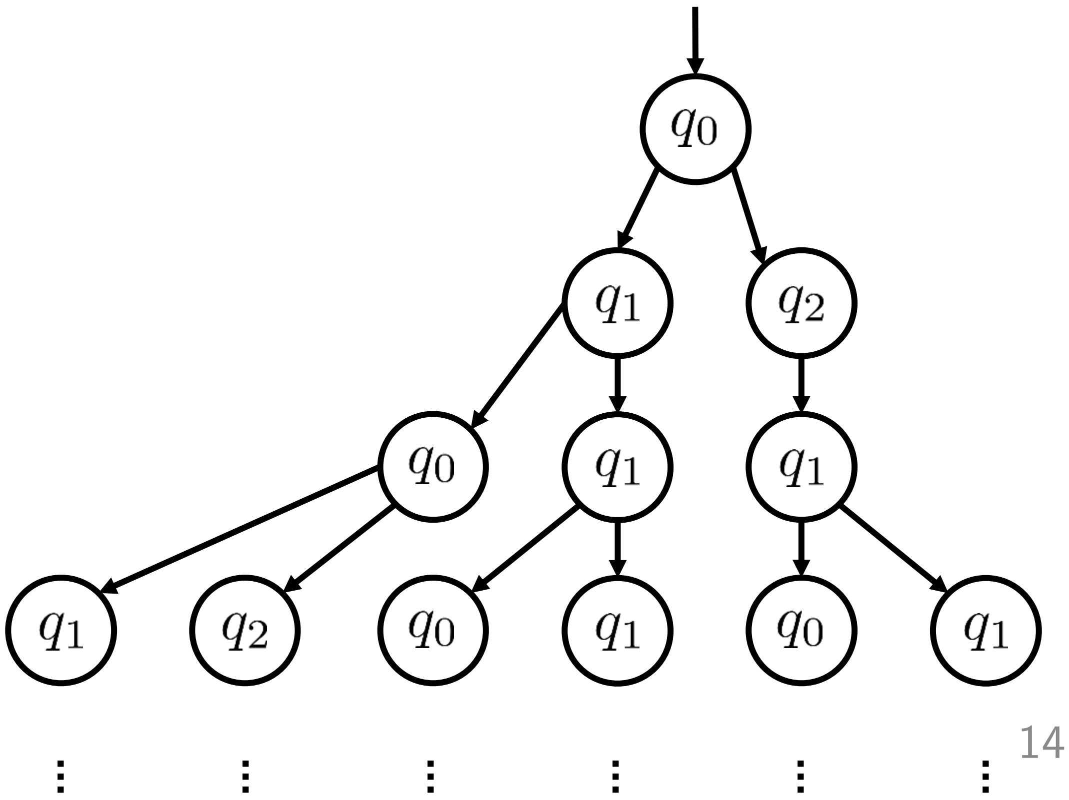
CTL works on computation trees

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Automaton



Computation tree

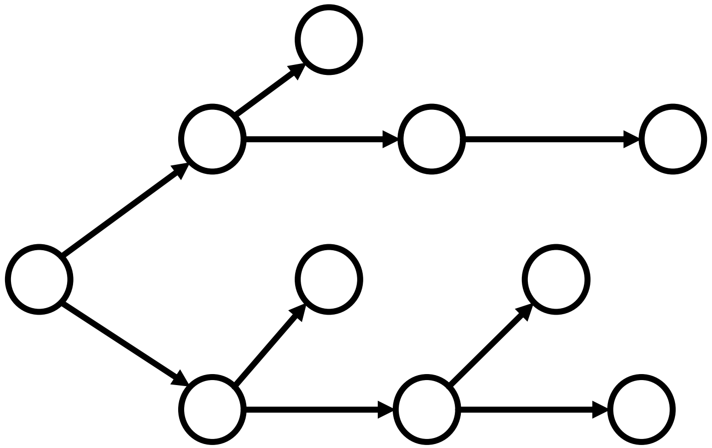


Over paths:
$A\phi \to$ **A**ll ϕ
$E\phi \to$ **E**xists ϕ

Path-specific:
$X\phi \to$ Ne**X**t ϕ
$F\phi \to$ **F**inally ϕ
$G\phi \to$ **G**lobally ϕ
$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

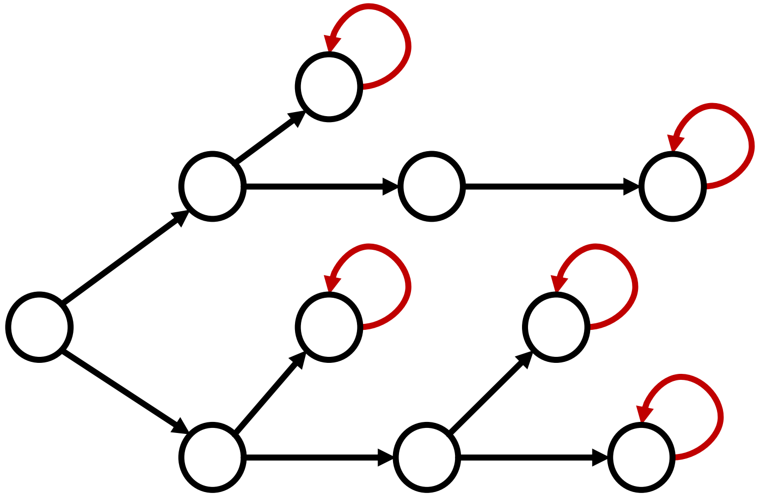
CTL works on computation trees

Automaton of interest



Requires fully-defined transition functions

Automaton to work with



Each state has at least one successor (can be itself)

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

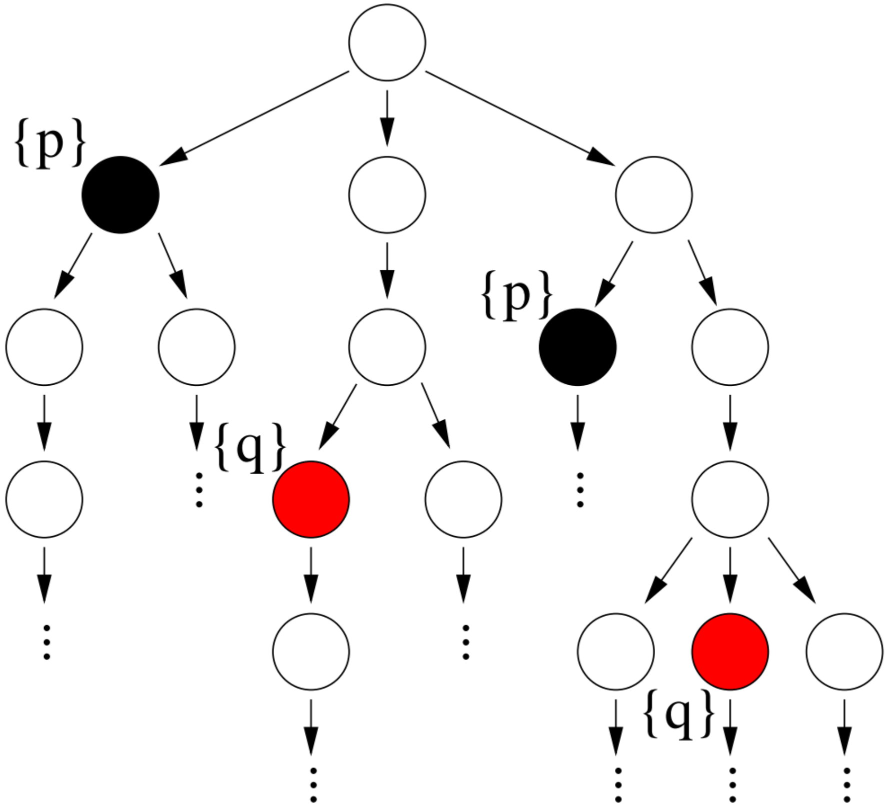
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Visualizing CTL formula

- We use this computation tree as a running example.

- We suppose that the black and red states satisfy atomic properties p and q, respectively.

- The topmost state is the initial state; in the examples, it always satisfies the given formula.



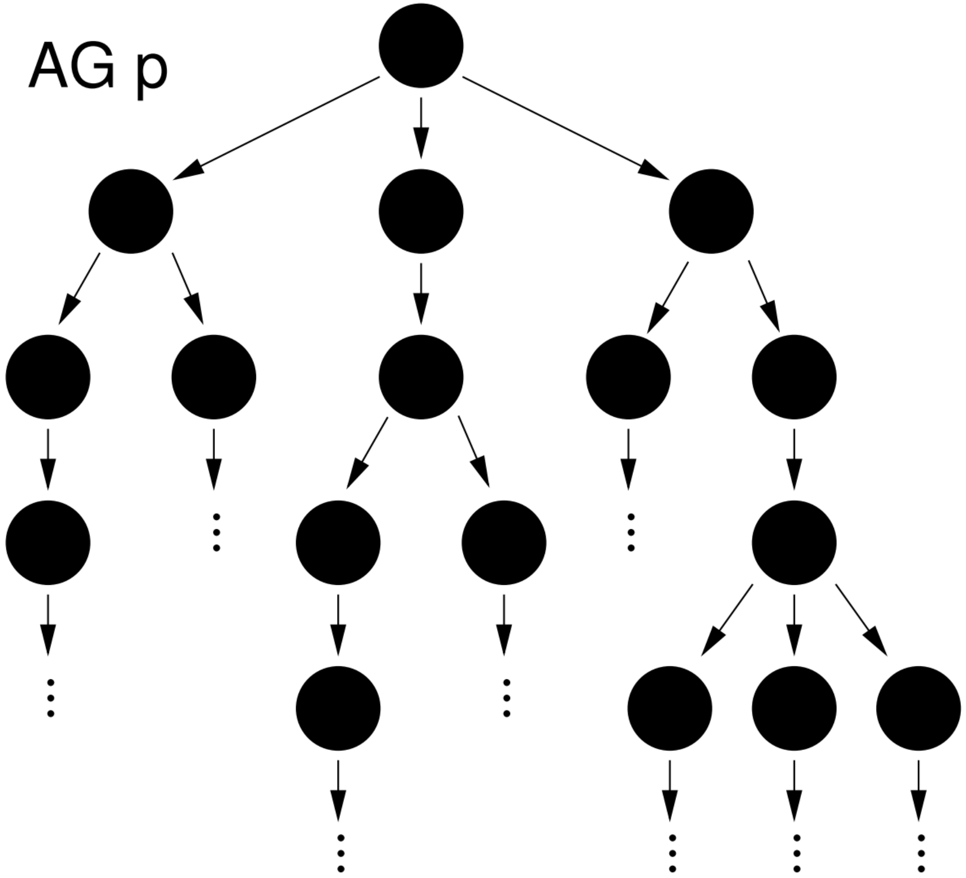
M satisfies $\phi \quad \Leftrightarrow \quad q_0 \vDash \phi$ where q_0 is the initial state of M

Over paths:
$A\phi \rightarrow$ **All** ϕ
$E\phi \rightarrow$ **Exists** ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

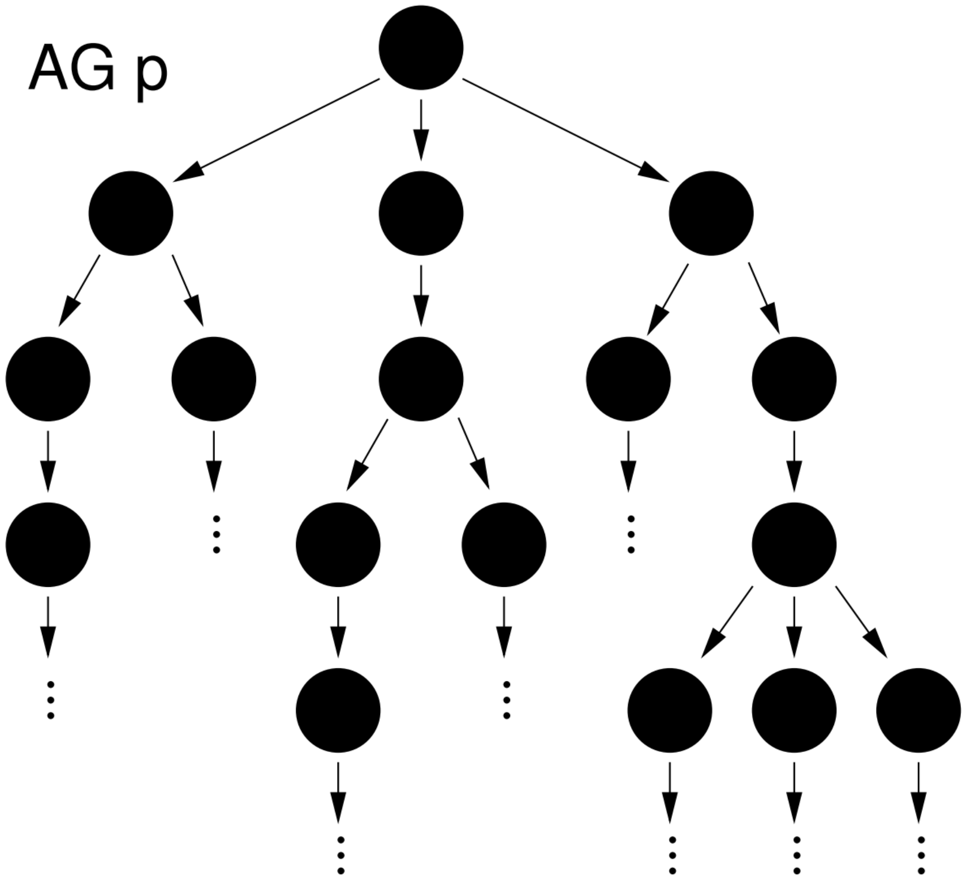
Visualizing CTL formula

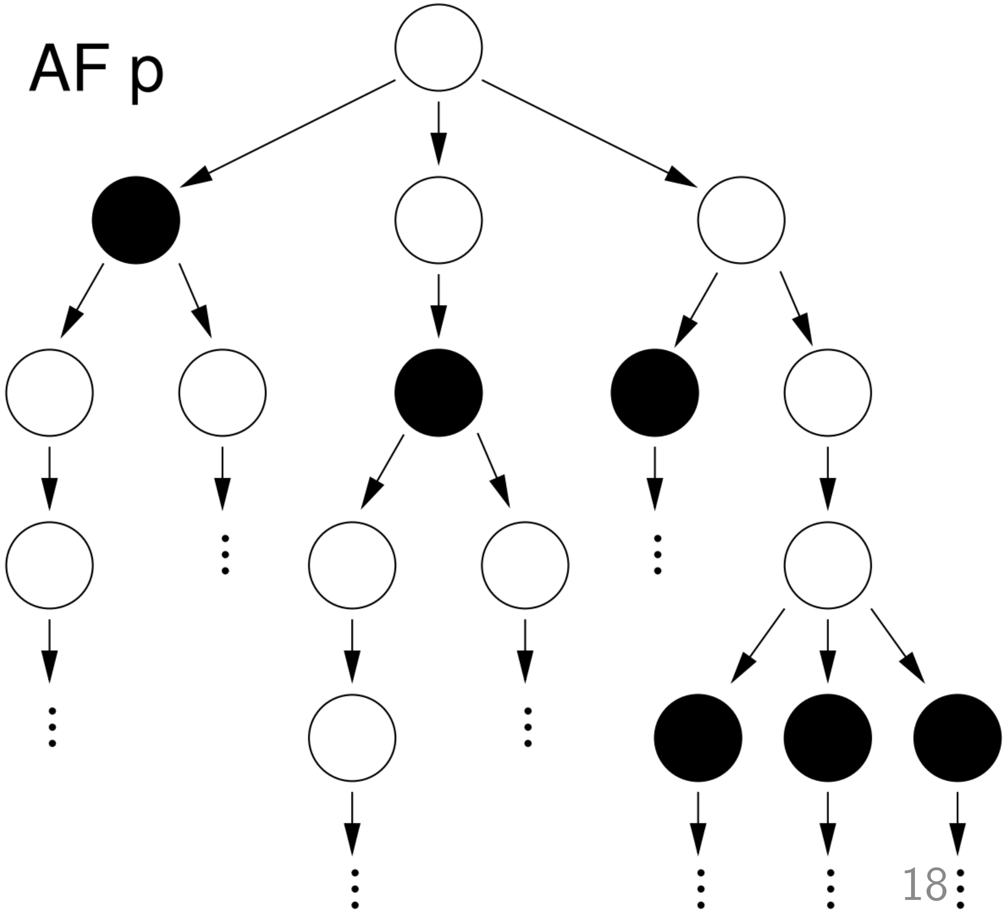
AG p



Over paths:	Path-specific:
$A\phi \rightarrow$ **All** ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **Exists** ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Visualizing CTL formula



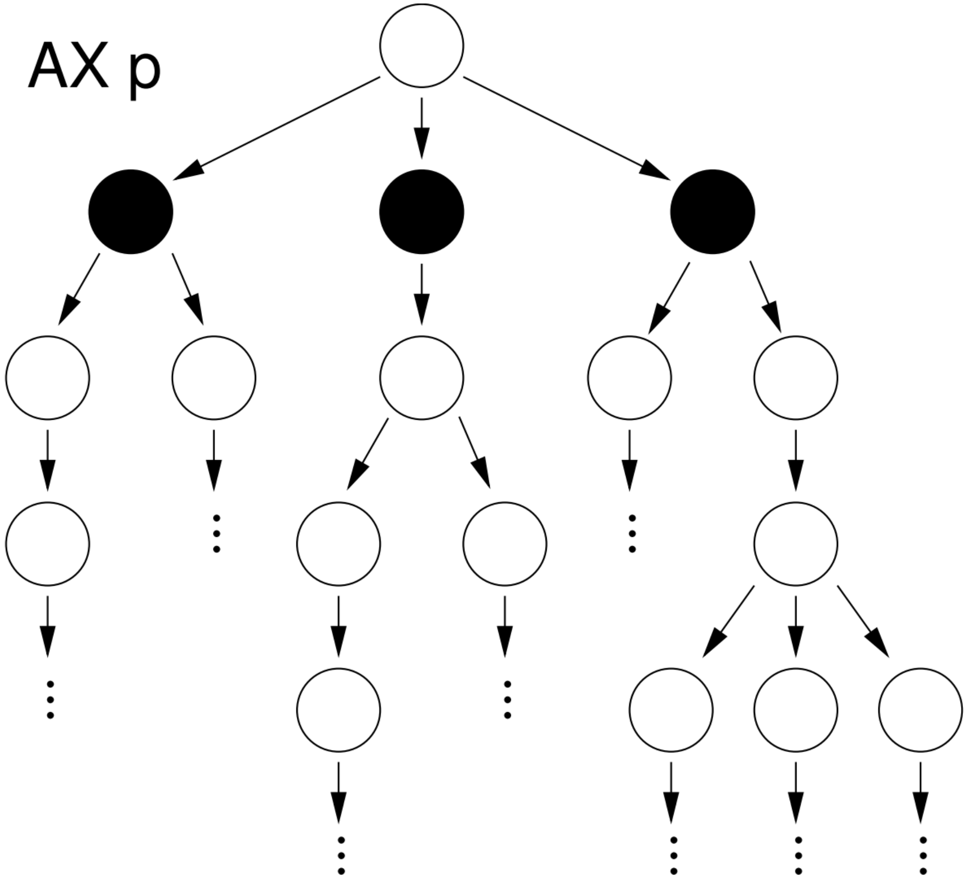


Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Visualizing CTL formula

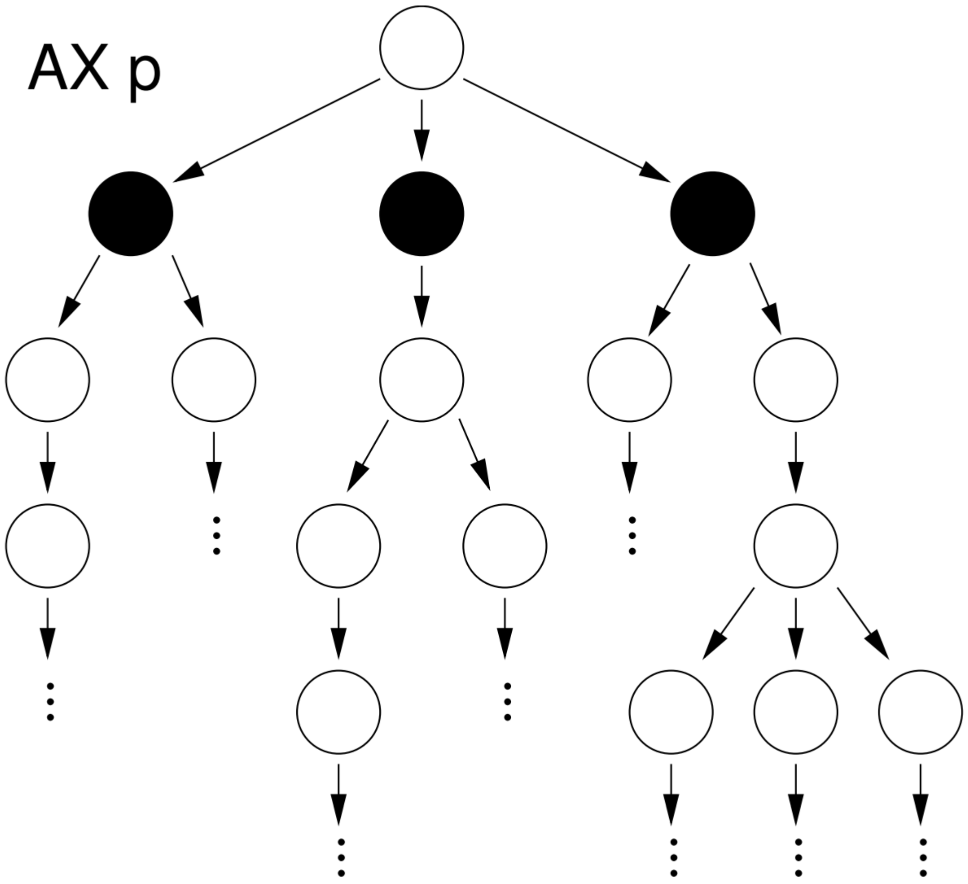
AX p



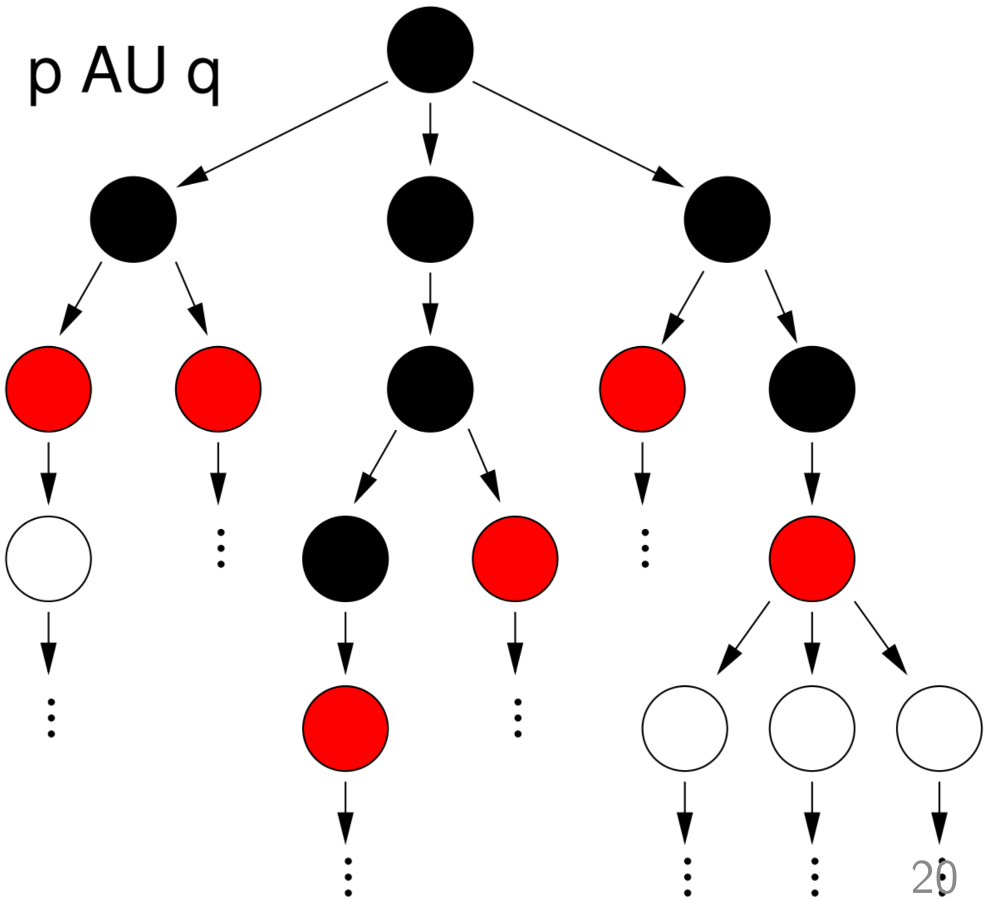
Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Visualizing CTL formula

AX p



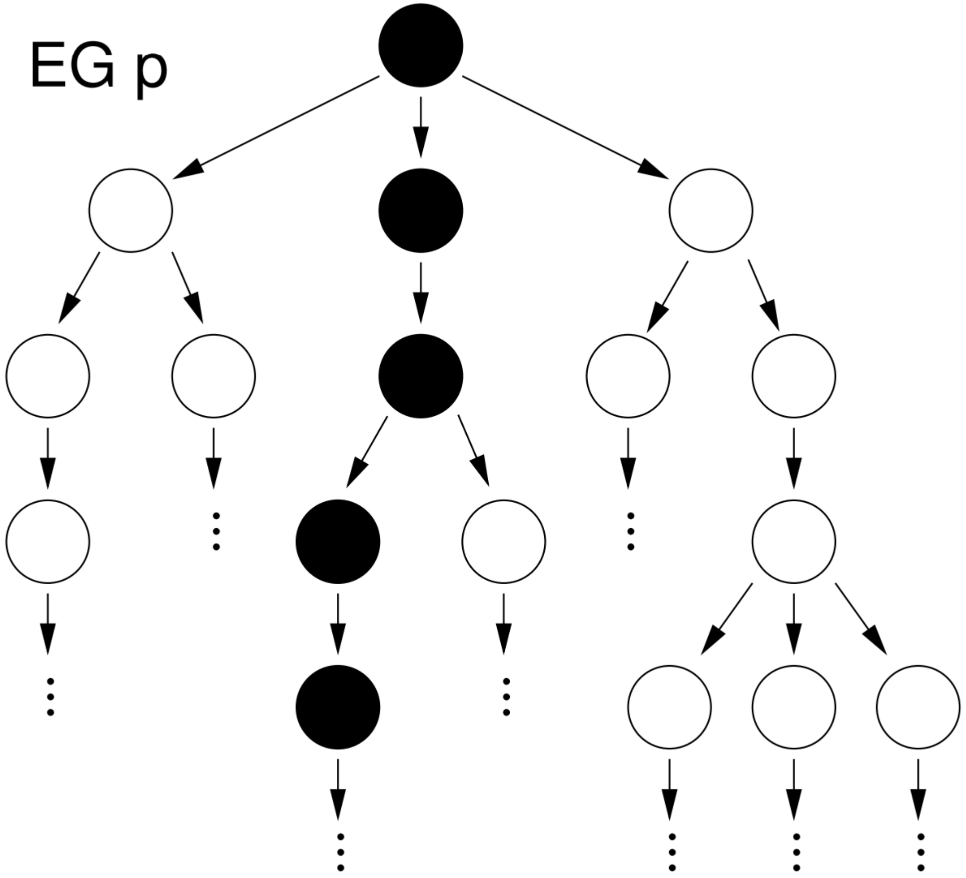
p AU q



Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

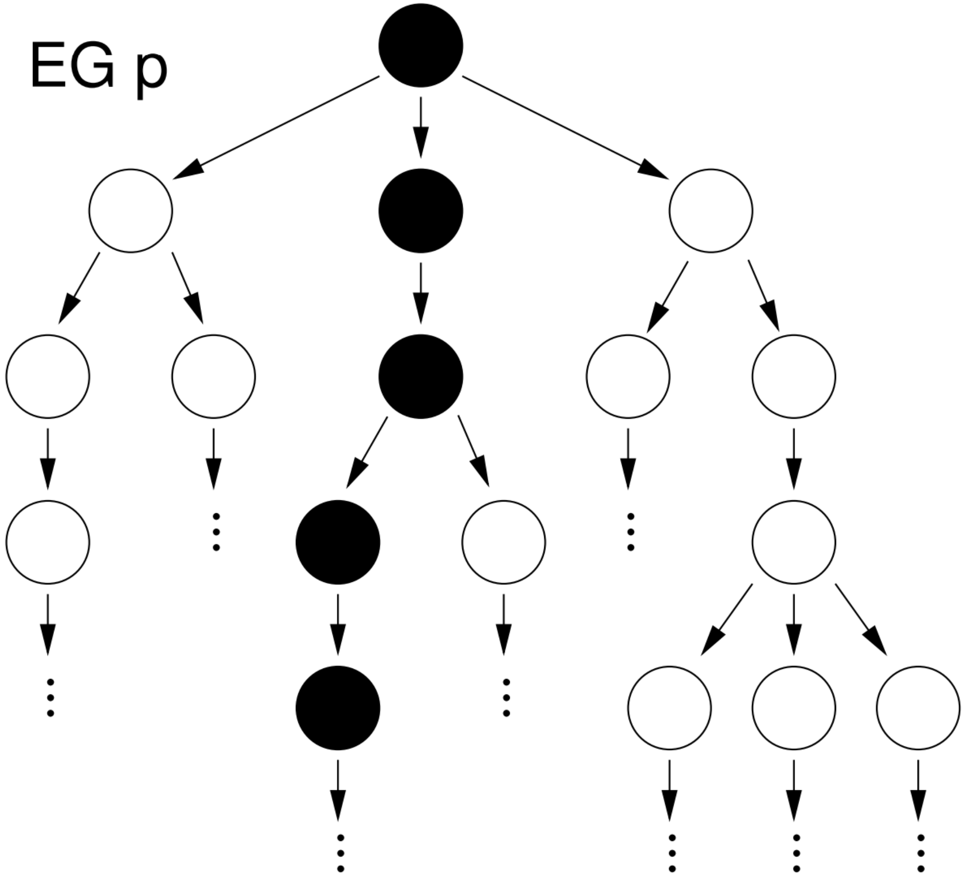
Visualizing CTL formula



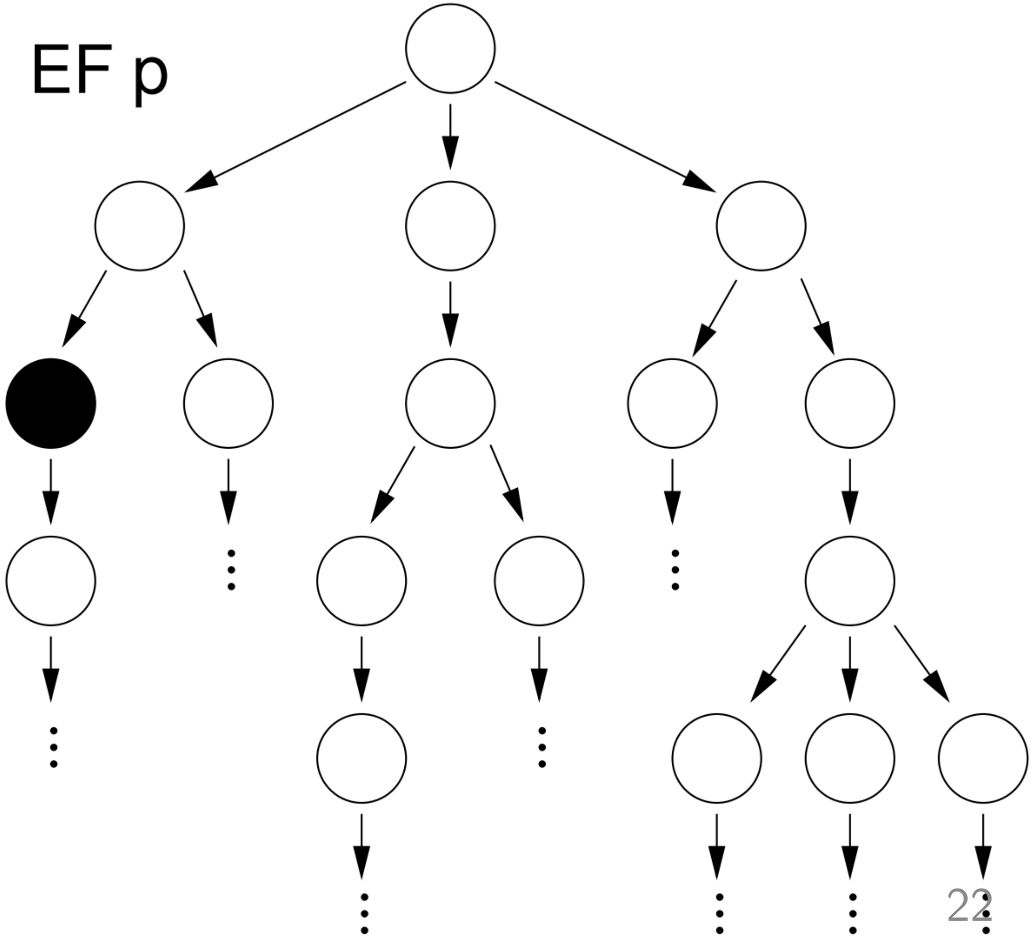
Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Visualizing CTL formula

EG p



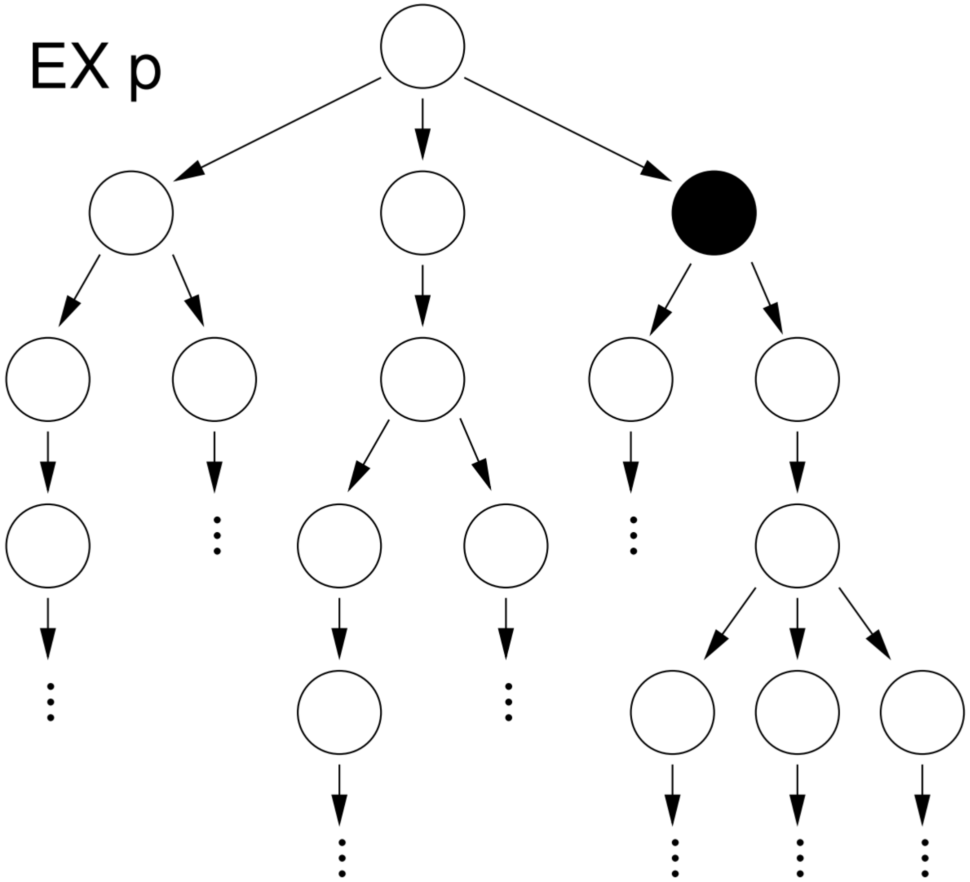
EF p



Over paths:	Path-specific:
$A\phi \rightarrow$ **All** ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **Exists** ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Visualizing CTL formula

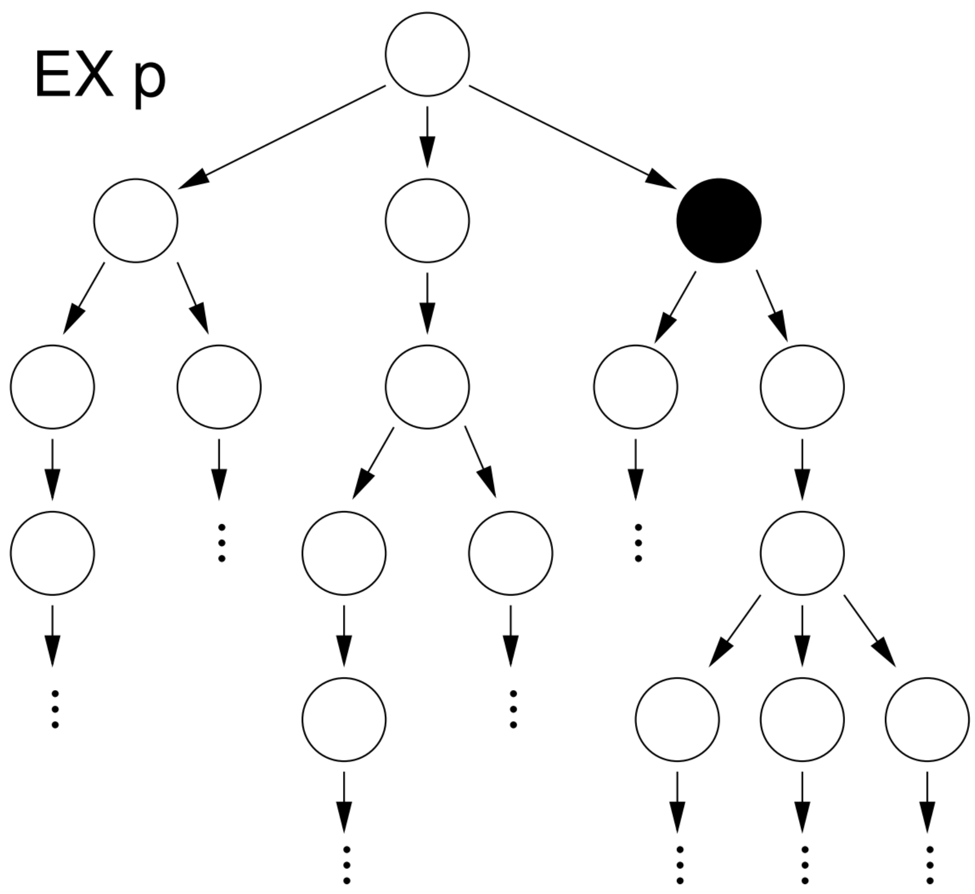
EX p

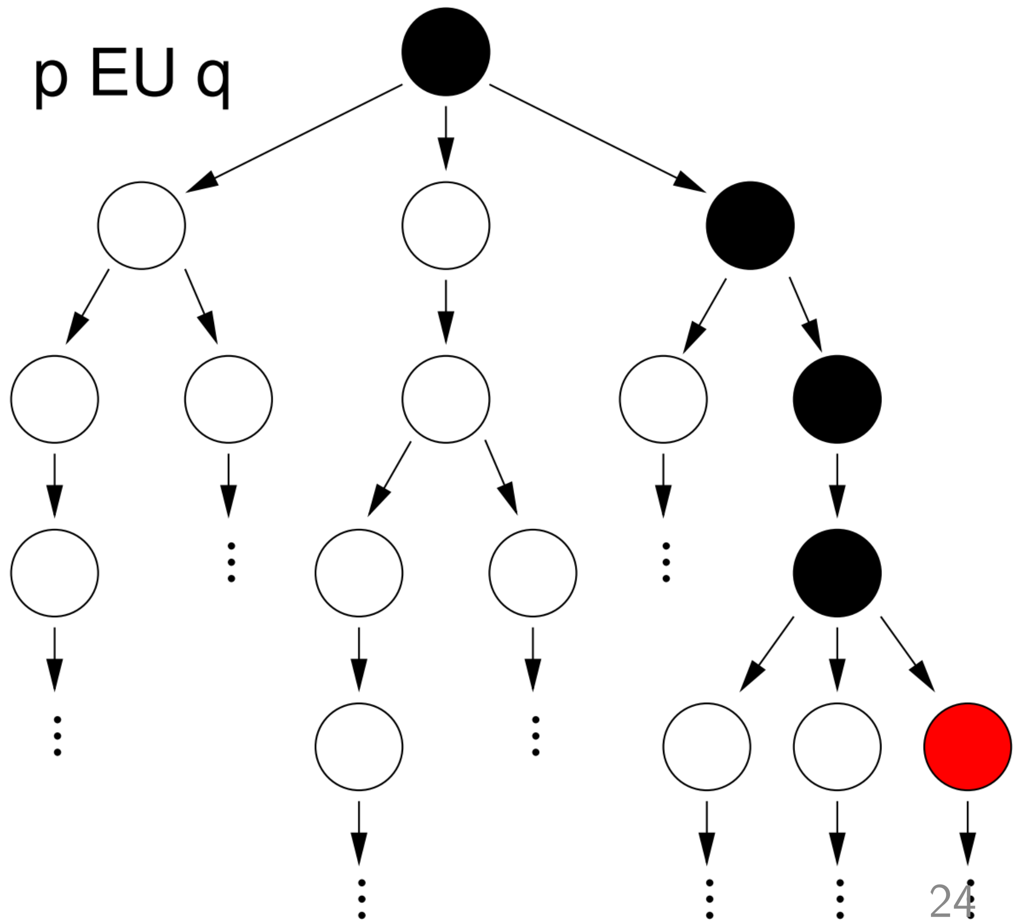


Over paths:
$A\phi \to$ **A**ll ϕ
$E\phi \to$ **E**xists ϕ

Path-specific:
$X\phi \to$ Ne**X**t ϕ
$F\phi \to$ **F**inally ϕ
$G\phi \to$ **G**lobally ϕ
$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

Visualizing CTL formula





Formulation of CTL properties

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Can be more than one pair

AG ϕ_1 where ϕ_1 = EF ϕ_2 $\equiv$ AG EF ϕ_2

A and F are convenient, but not necessary

E,G,X,U are sufficient to define the whole logic.

$$\mathrm{AF}\phi \equiv \neg\mathrm{EG}(\neg\phi)$$

$$\mathrm{AG}\phi \equiv \neg\mathrm{EF}(\neg\phi)$$

$$\mathrm{AX}\phi \equiv \neg\mathrm{EX}(\neg\phi)$$

$$\mathrm{EF}\phi \equiv \mathrm{true}\,\mathrm{EU}\phi$$

No need to know that one ▸

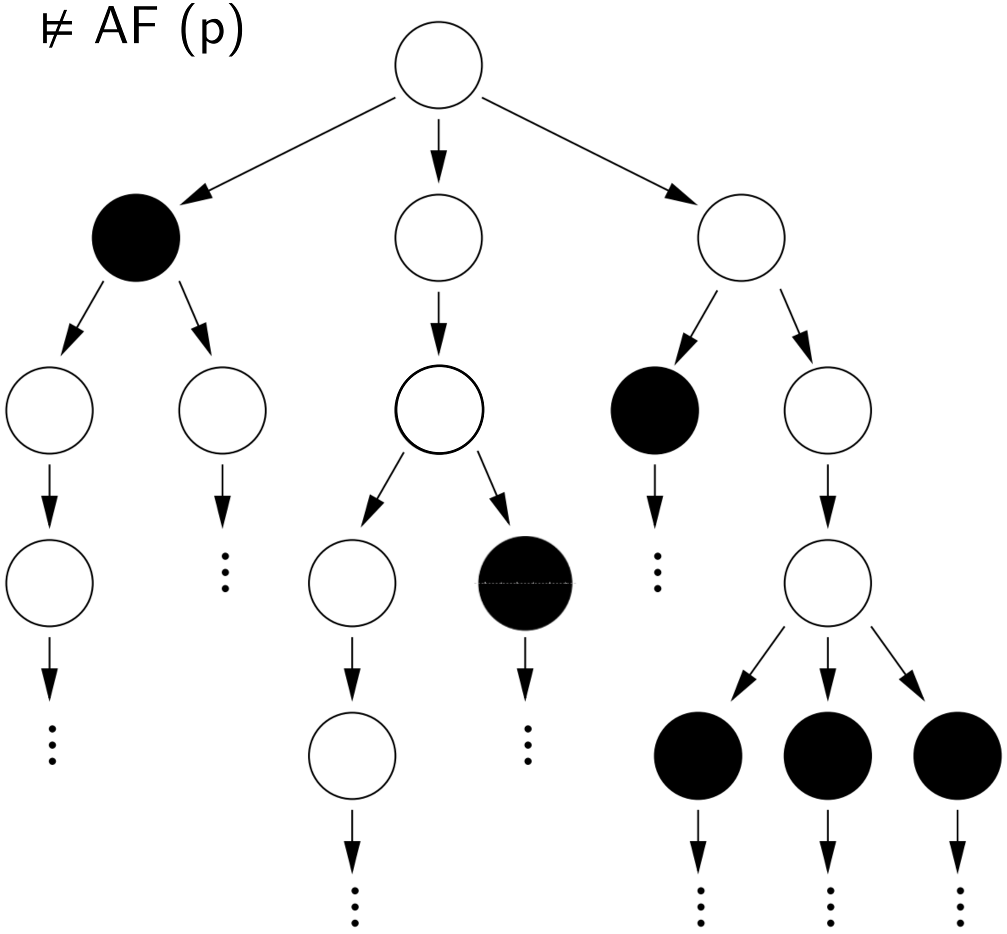
$$\phi_1 \mathrm{AU} \phi_2 \equiv \neg([(\neg\phi_1)\mathrm{EU}\neg(\phi_1+\phi_2)] + \mathrm{EG}(\neg\phi_2))$$

Over paths:
$A\phi \rightarrow$ **All** ϕ
$E\phi \rightarrow$ **Exists** ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Intuition for "AF p = ¬ EG (¬ p)"

$\not\models$ AF (p)

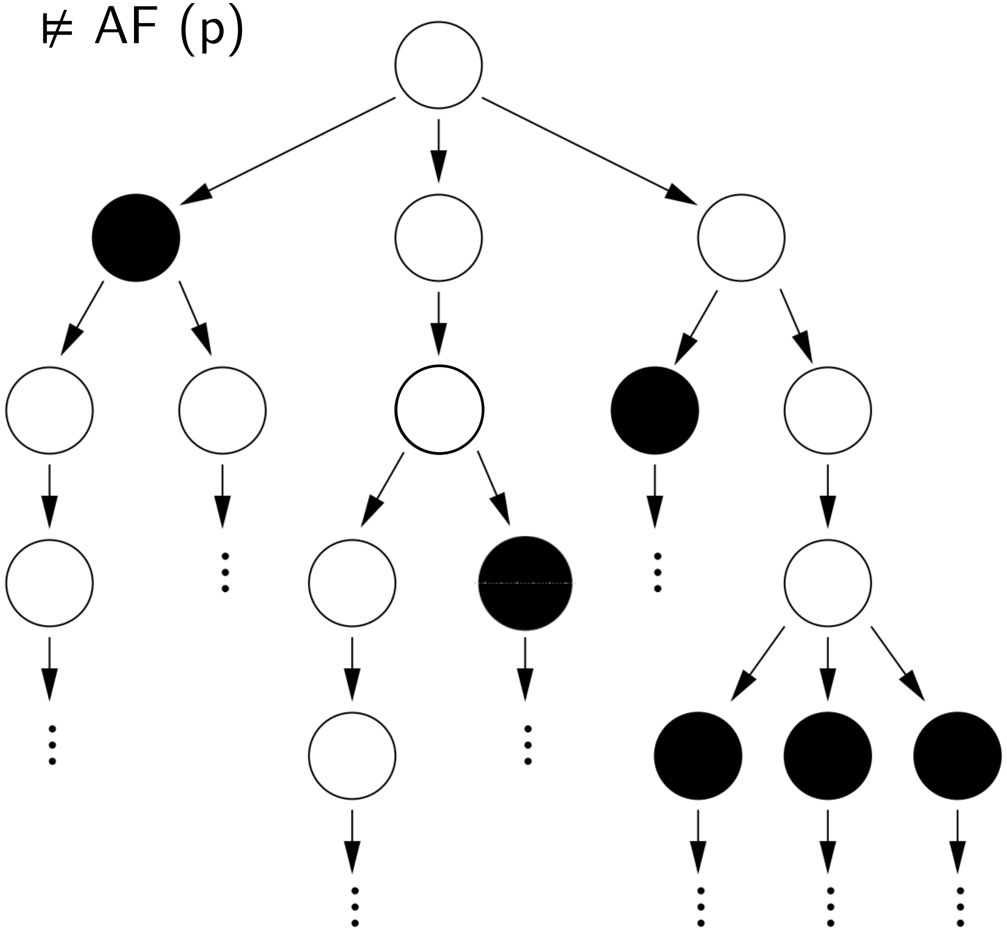


Intuition for “AF p = ¬ EG (¬ p)”

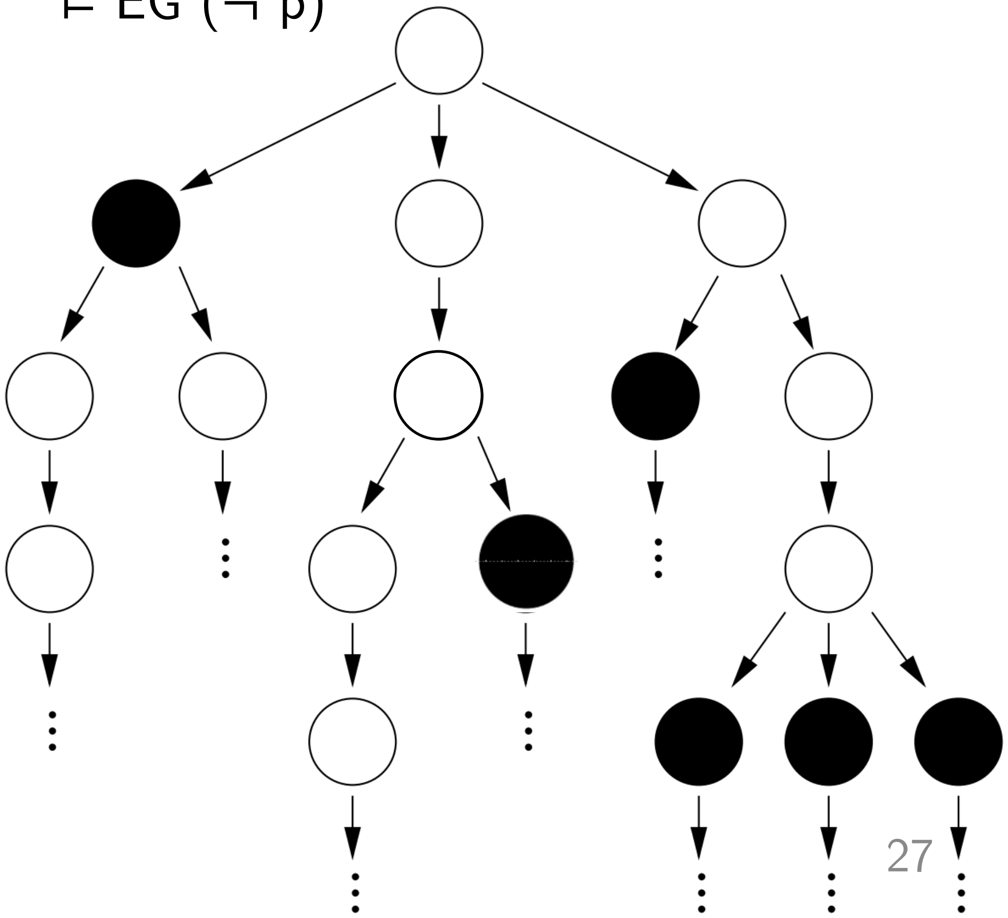
Over paths:
$A\phi \rightarrow$ **All** ϕ
$E\phi \rightarrow$ **Exists** ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

$\not\models$ AF (p)



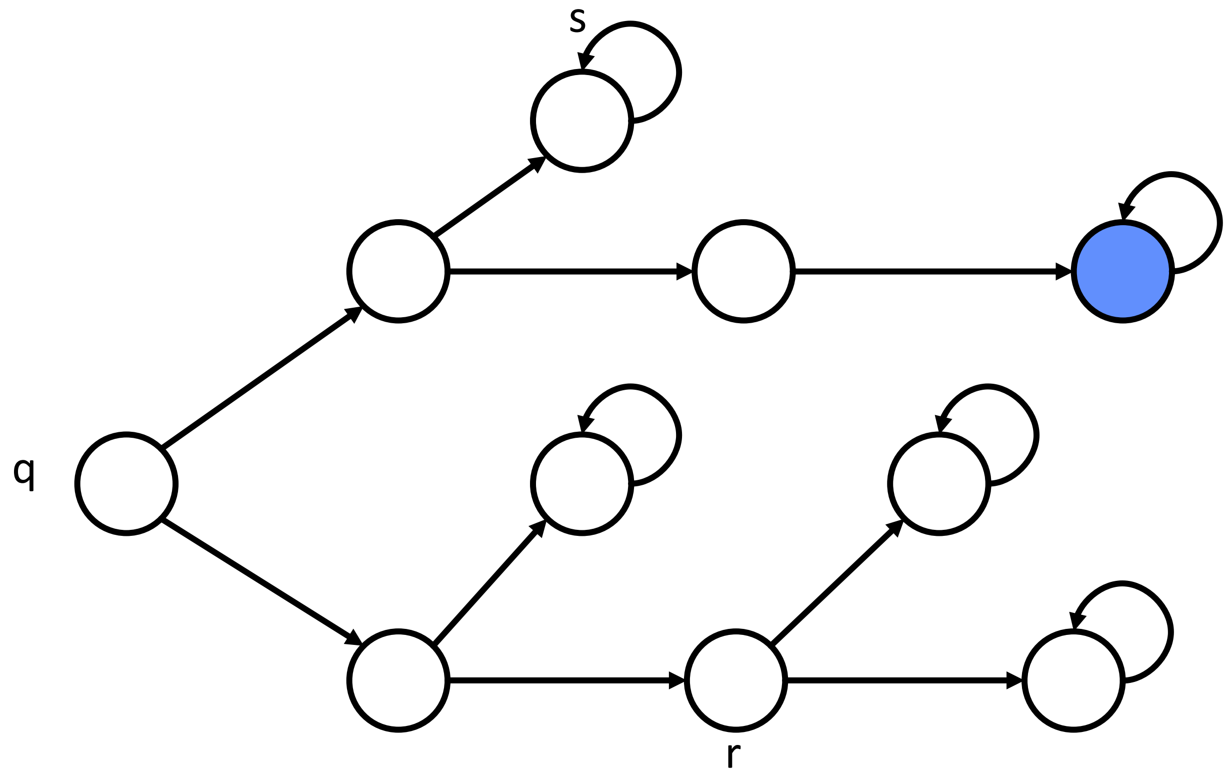
$\models$ EG (¬ p)

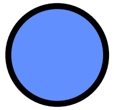


Evaluating a CTL formula

EF ϕ : "There exists a path along which at some state ϕ holds."

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



 $\models \phi$

$q \models EF\ \phi$

$r \models ?$

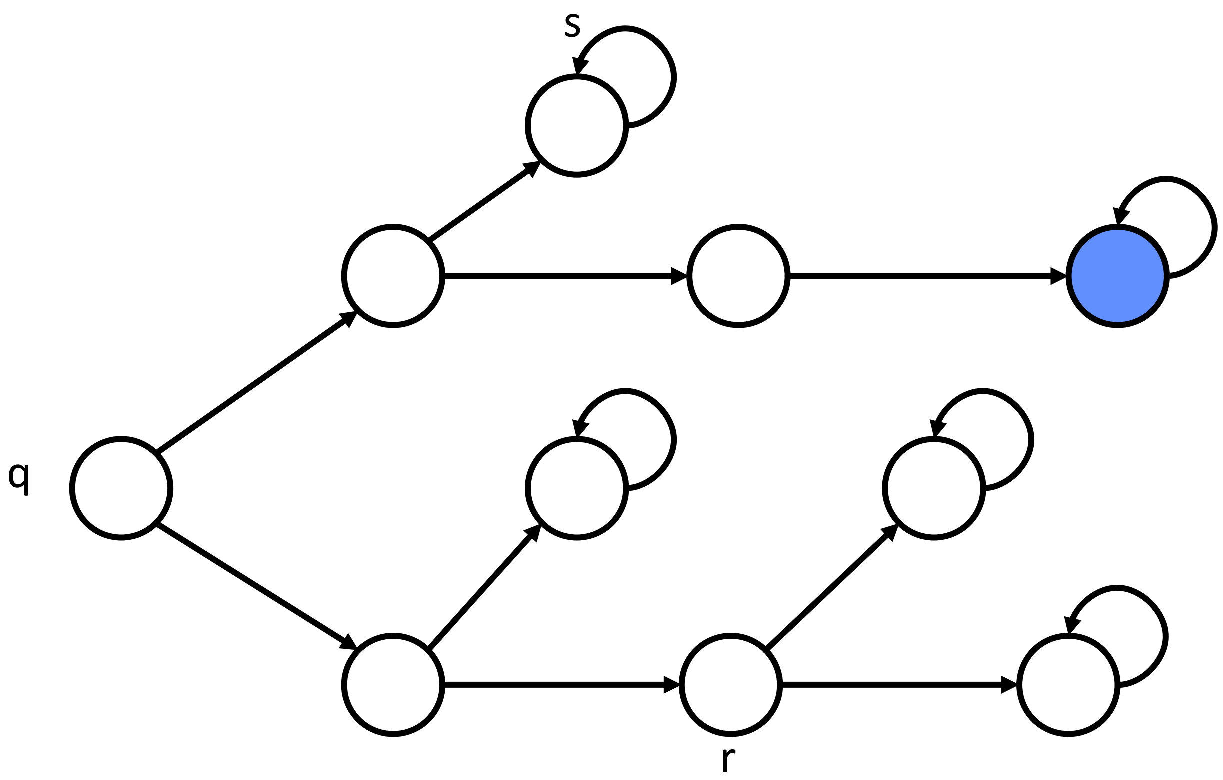
$s \models ?$

Evaluating a CTL formula

EF ϕ : "There exists a path along which at some state ϕ holds."

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



$\models \phi$

$q \models EF\ \phi$

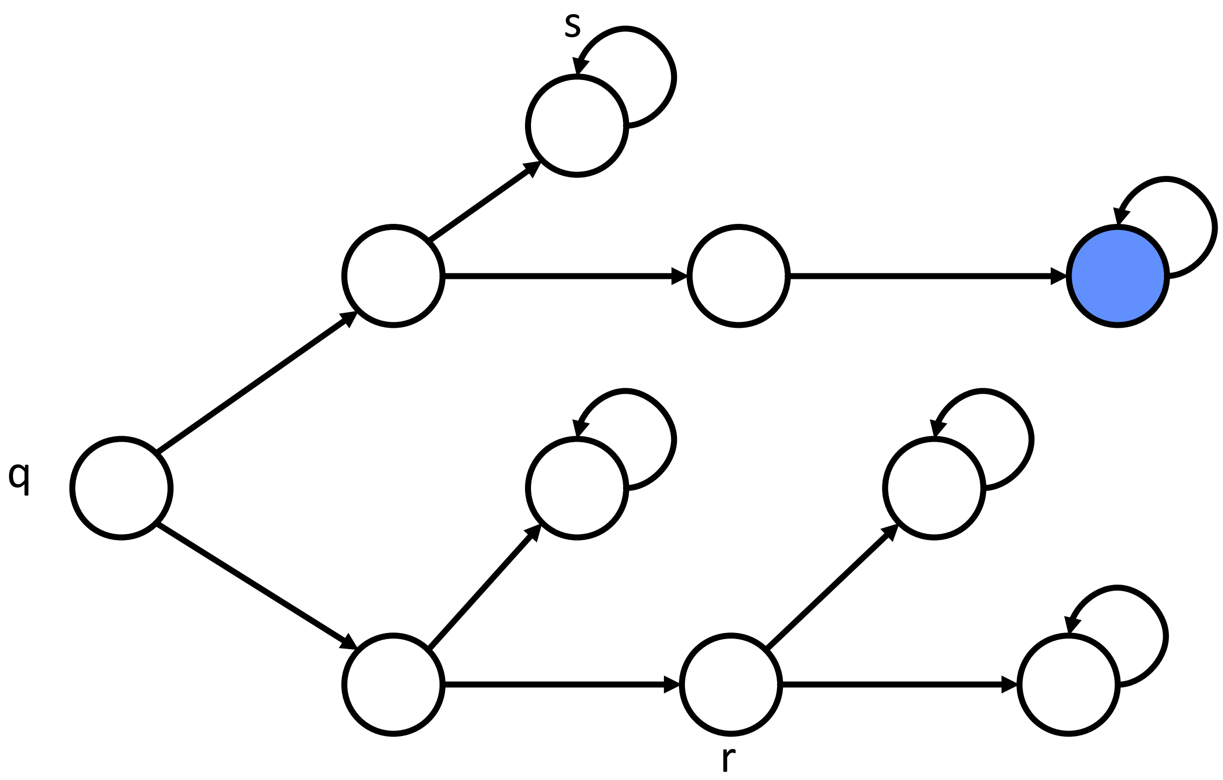
$r \not\models EF\ \phi$

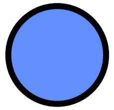
$s \models ?$

Evaluating a CTL formula

EF ϕ : "There exists a path along which at some state ϕ holds."

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2





$\vDash \phi$

$q \vDash EF\ \phi$

$r \nvDash EF\ \phi$

$s \nvDash EF\ \phi$

Over paths:

$A\phi \rightarrow$ **All** ϕ

$E\phi \rightarrow$ **Exists** ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

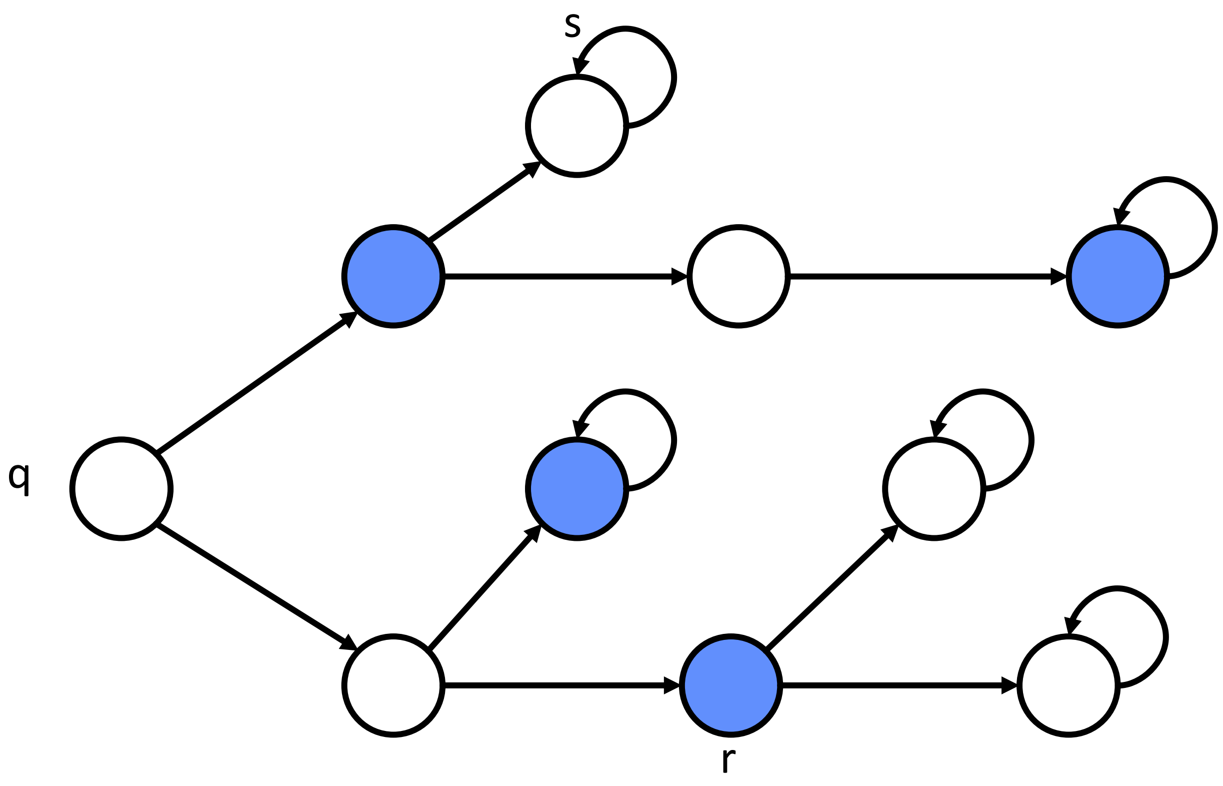
$F\phi \rightarrow$ **F**inally ϕ

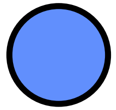
$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Evaluating a CTL formula

AF ϕ : "On all paths, at some state ϕ holds."





$\vDash \phi$

$q \vDash AF\ \phi$

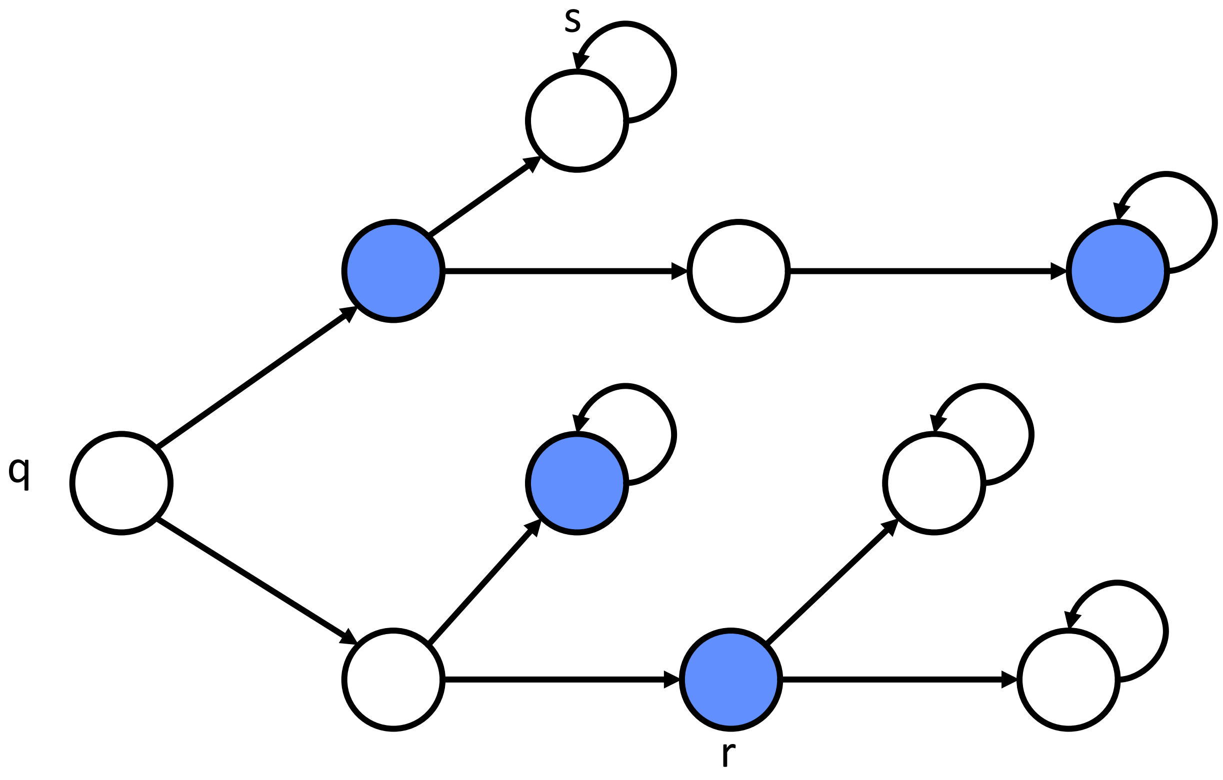
$r \vDash ?$

$s \vDash ?$

Evaluating a CTL formula

AF ϕ : "On all paths, at some state ϕ holds."

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



$\models \phi$

$q \models AF\ \phi$

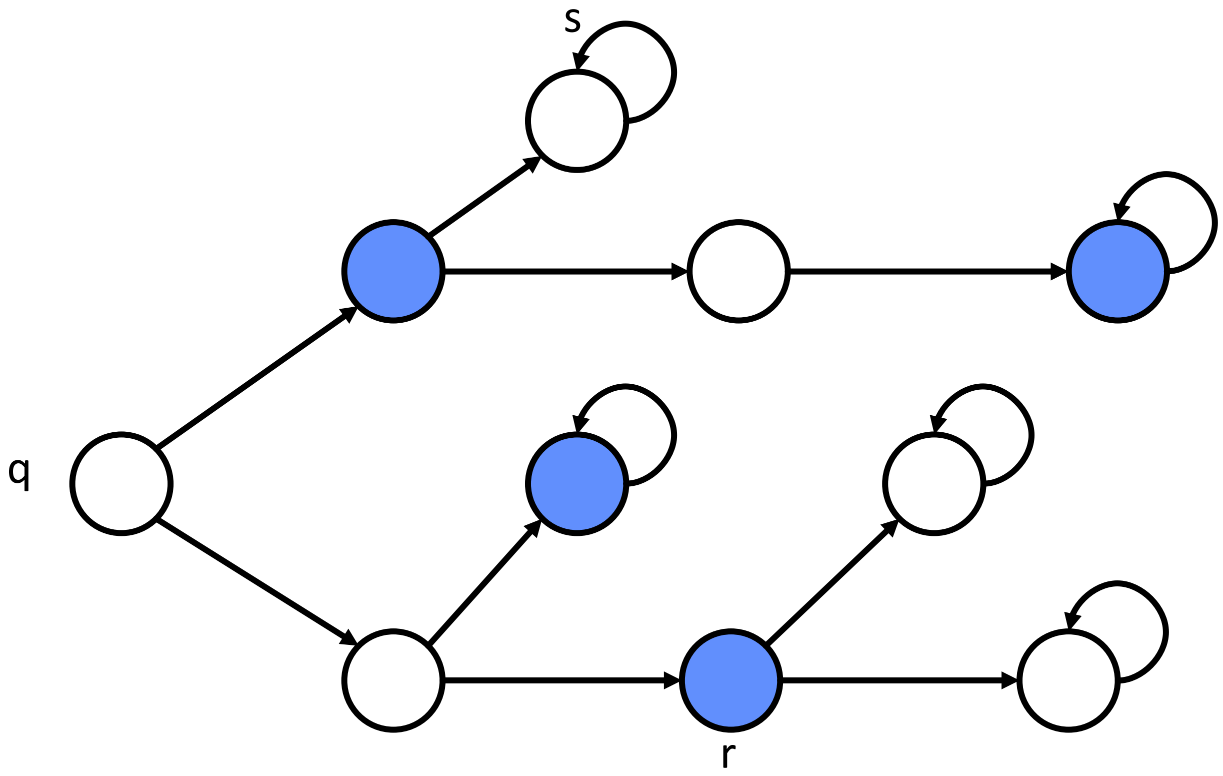
$r \models AF\ \phi$

$s \models ?$

Over paths:	Path-specific:
$A\phi \rightarrow$ **All** ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **Exists** ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Evaluating a CTL formula

AF ϕ : "On all paths, at some state ϕ holds."



$\vDash \phi$

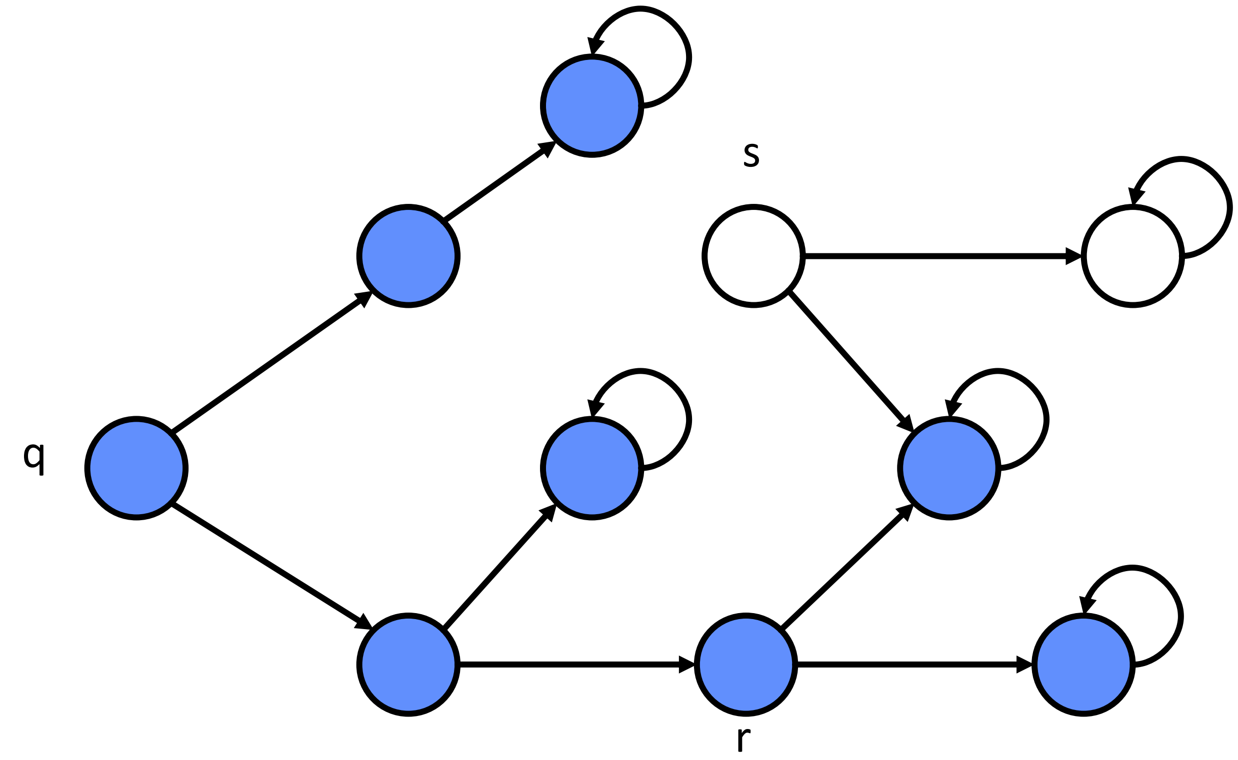
$q \vDash AF\,\phi$

$r \vDash AF\,\phi$

$s \nvDash AF\,\phi$

Evaluating a CTL formula
AG ϕ : "On all paths, for all states ϕ holds."

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



$\models \phi$

$q \models AG\ \phi$

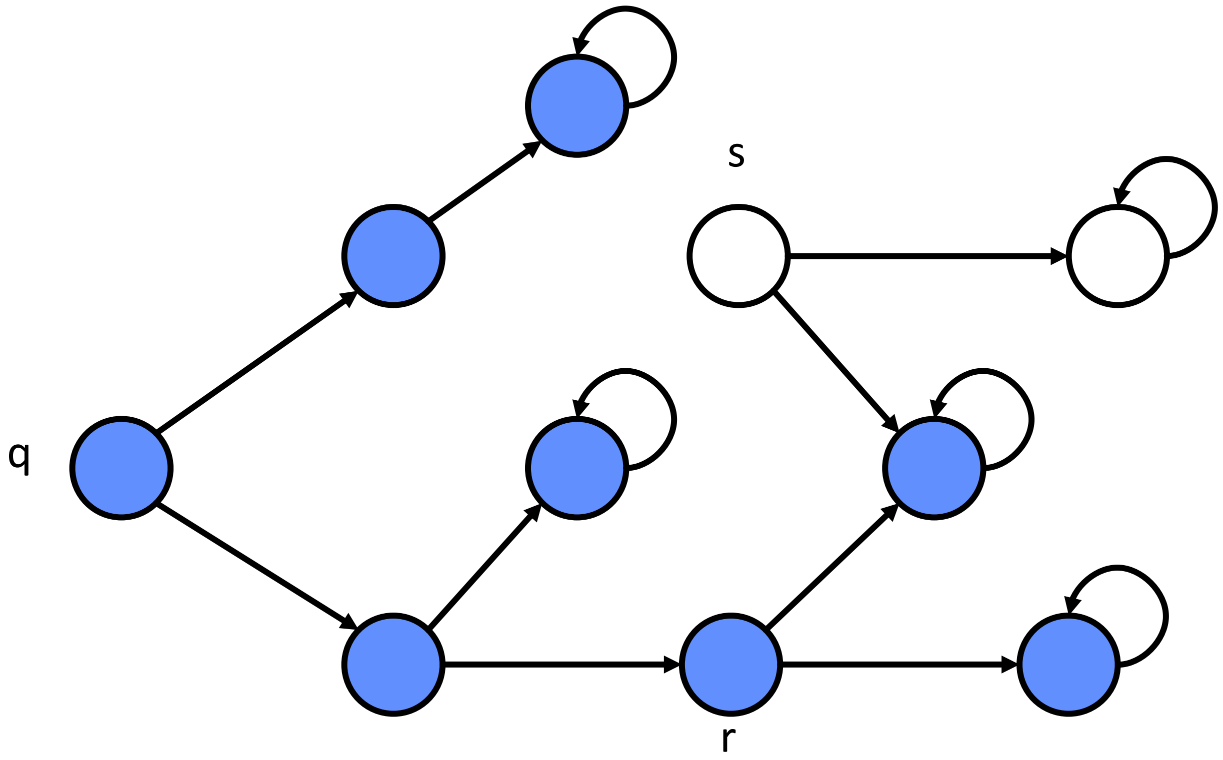
$r \models ?$

$s \models ?$

Evaluating a CTL formula

AG ϕ : "On all paths, for all states ϕ holds."

Over paths:	Path-specific:
$A\phi$ → **A**ll ϕ	$X\phi$ → Ne**X**t ϕ
$E\phi$ → **E**xists ϕ	$F\phi$ → **F**inally ϕ
	$G\phi$ → **G**lobally ϕ
	$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2



● $\models \phi$

$q \models AG\ \phi$

$r \models AG\ \phi$

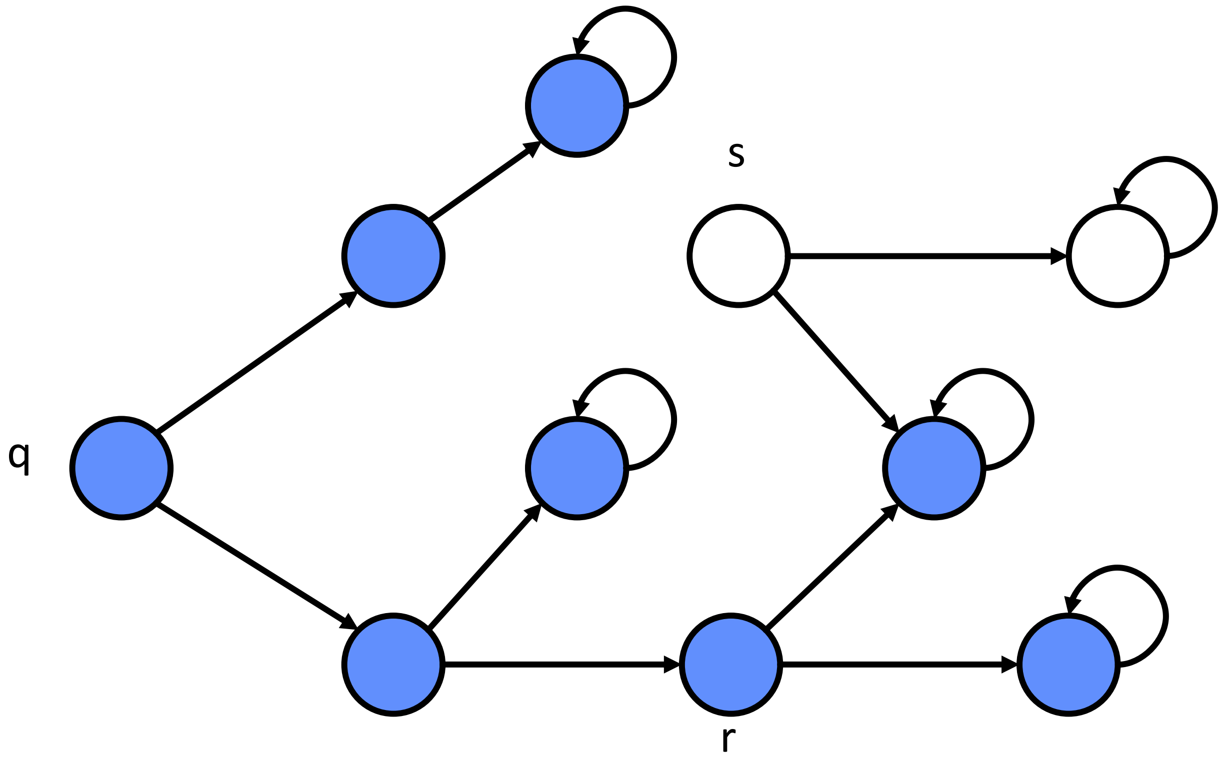
$s \models$?

Evaluating a CTL formula

AG ϕ : "On all paths, for all states ϕ holds."

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



● $\vDash \phi$

$q \vDash AG\ \phi$

$r \vDash AG\ \phi$

$s \nvDash AG\ \phi$

Evaluating a CTL formula

EG ϕ : "There exists a path along which for all states ϕ holds."

Over paths:

$A\phi \rightarrow$ **All** ϕ

$E\phi \rightarrow$ **Exists** ϕ

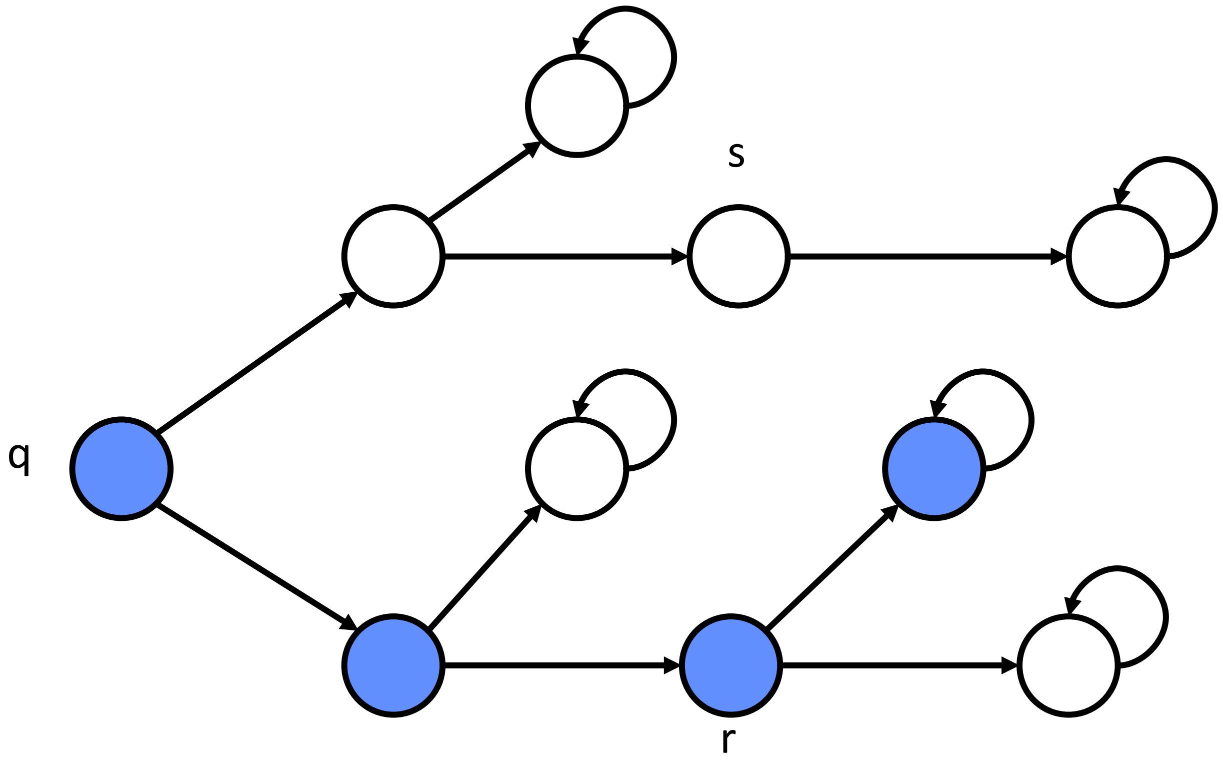
Path-specific:

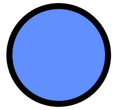
$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2




$\models \phi$

$q \models EG\ \phi$

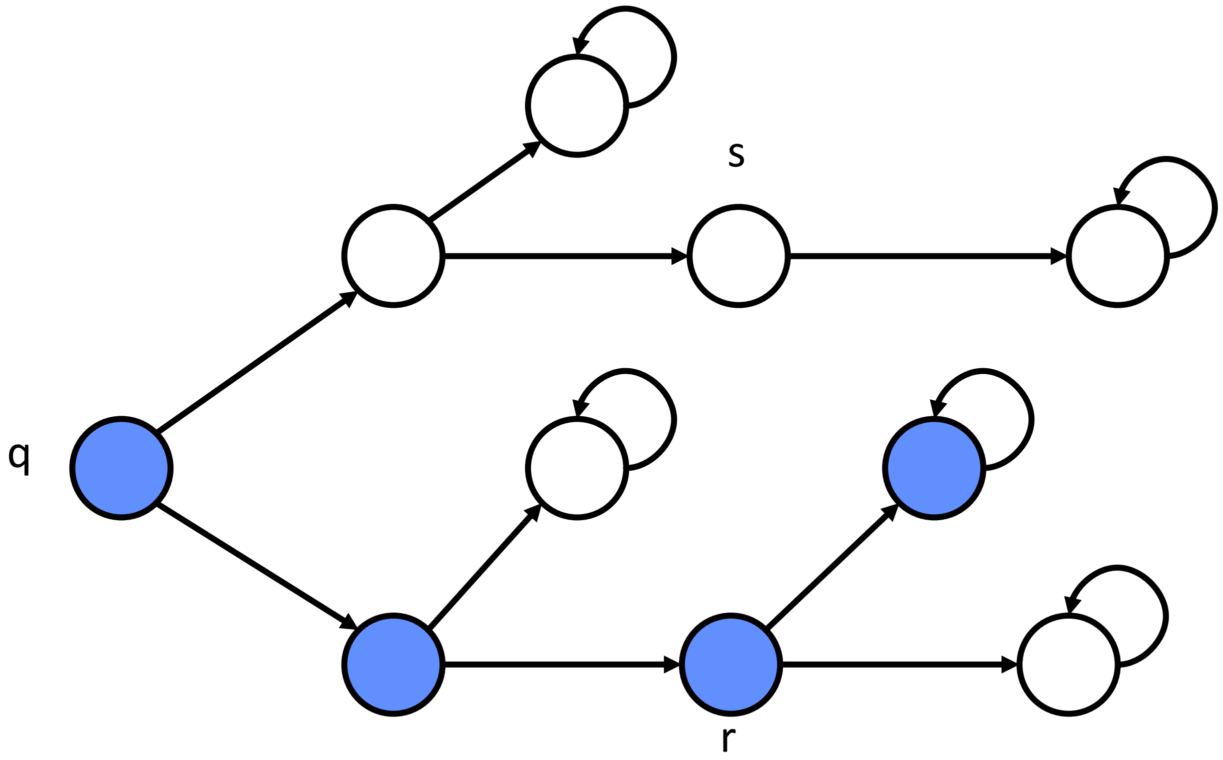
$r \models ?$

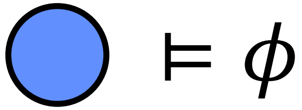
$s \models ?$

Evaluating a CTL formula

EG ϕ : "There exists a path along which for all states ϕ holds."

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2





$q \models EG\ \phi$

$r \models EG\ \phi$

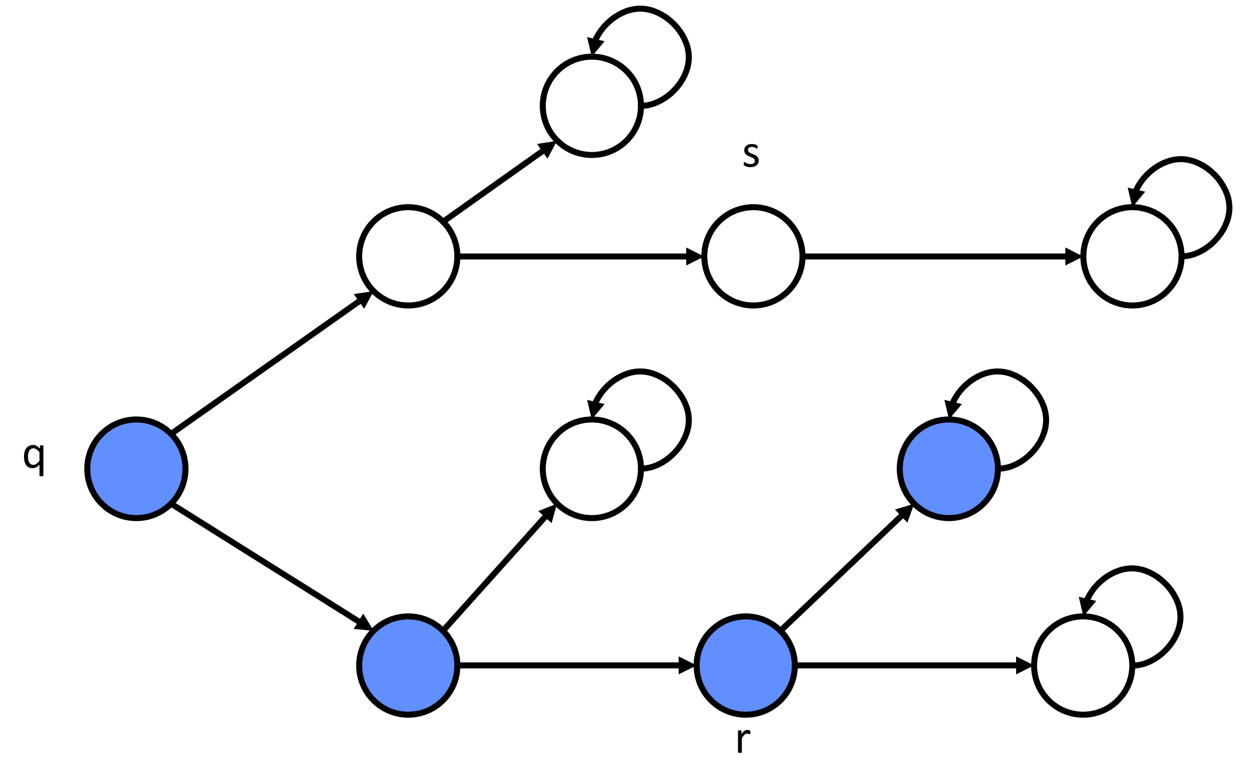
$s \models ?$

Evaluating a CTL formula

EG ϕ : "There exists a path along which for all states ϕ holds."

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



$\vDash \phi$

$q \vDash EG\ \phi$

$r \vDash EG\ \phi$

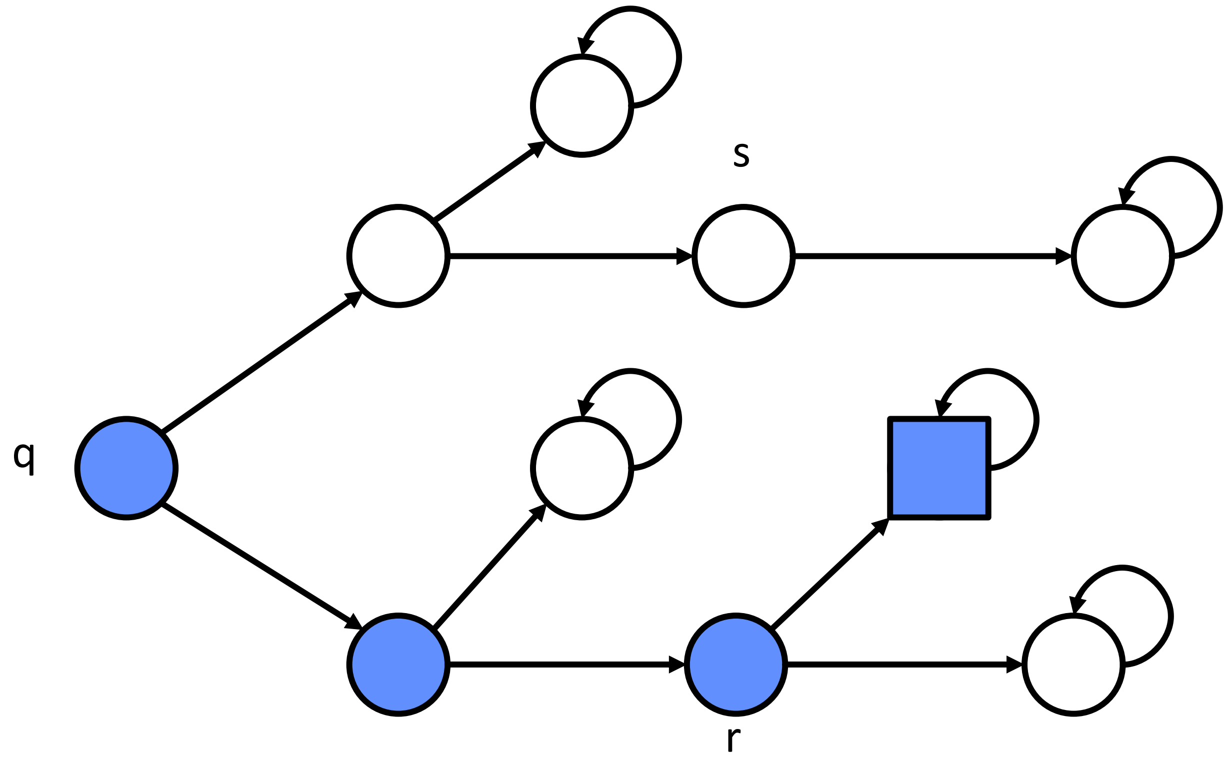
$s \nvDash EG\ \phi$

Evaluating a CTL formula

$\phi EU\Psi$: "There exists a path along which ϕ holds until Ψ holds."

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



$\blacksquare \models \Psi$

$\bullet \models \phi$

$q \models \phi EU\Psi$

$r \models ?$

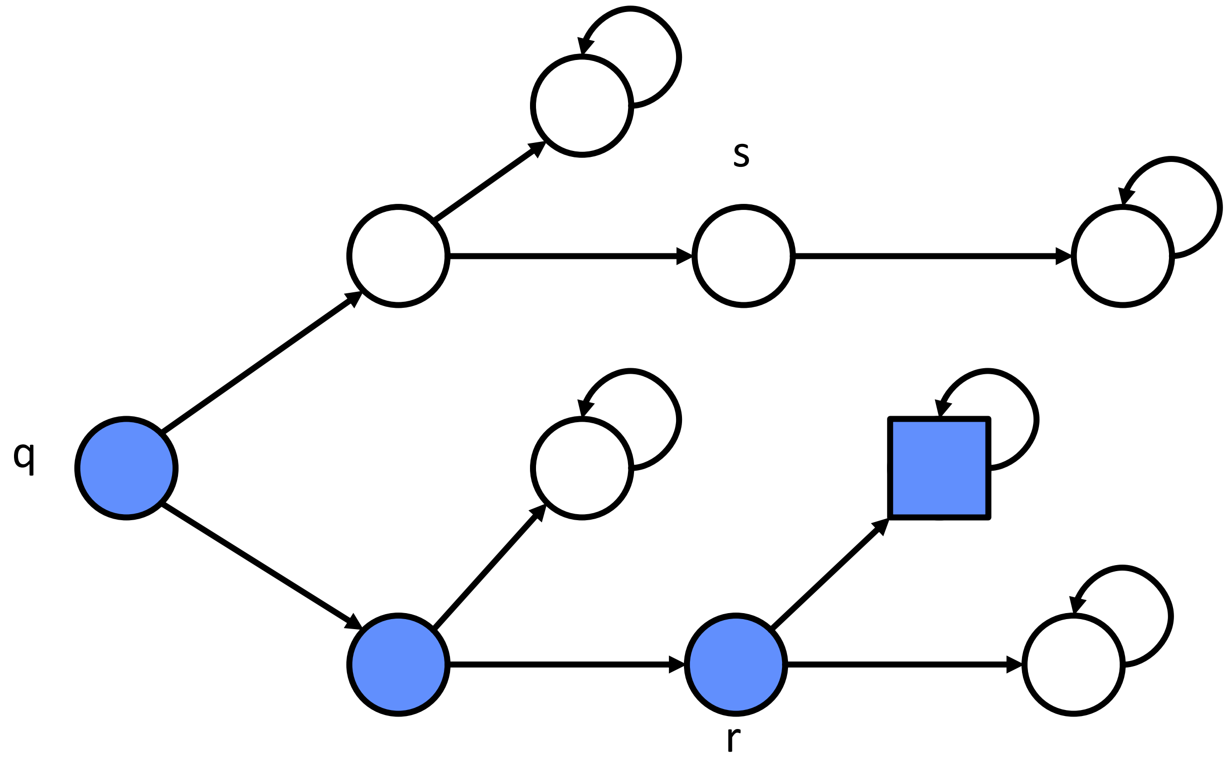
$s \models ?$

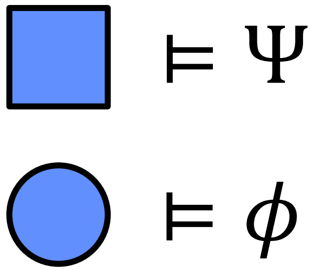
Evaluating a CTL formula

$\phi EU\Psi$: "There exists a path along which ϕ holds until Ψ holds."

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



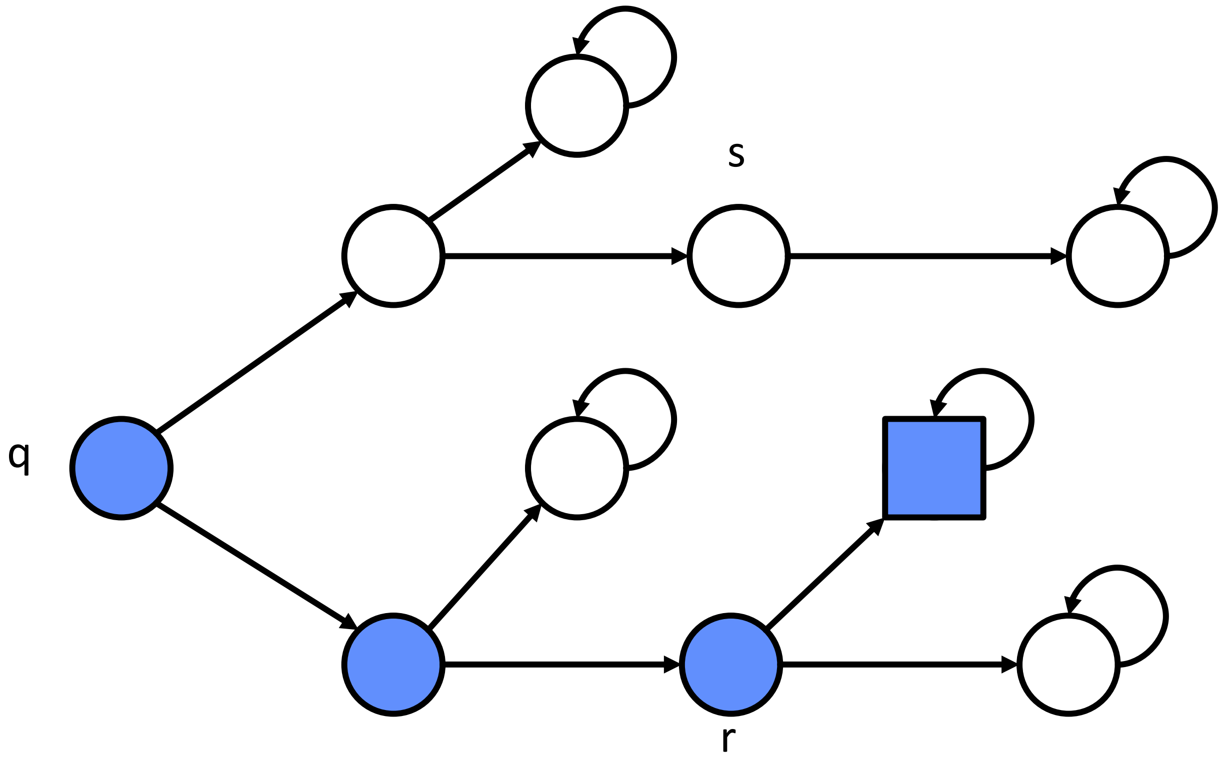


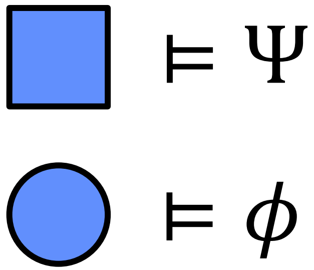
$q \vDash \phi EU\Psi$
$r \vDash \phi EU\Psi$
$s \vDash$?

Evaluating a CTL formula

$\phi E U \Psi$: "There exists a path along which ϕ holds until Ψ holds."

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2





$q \models \phi E U \Psi$

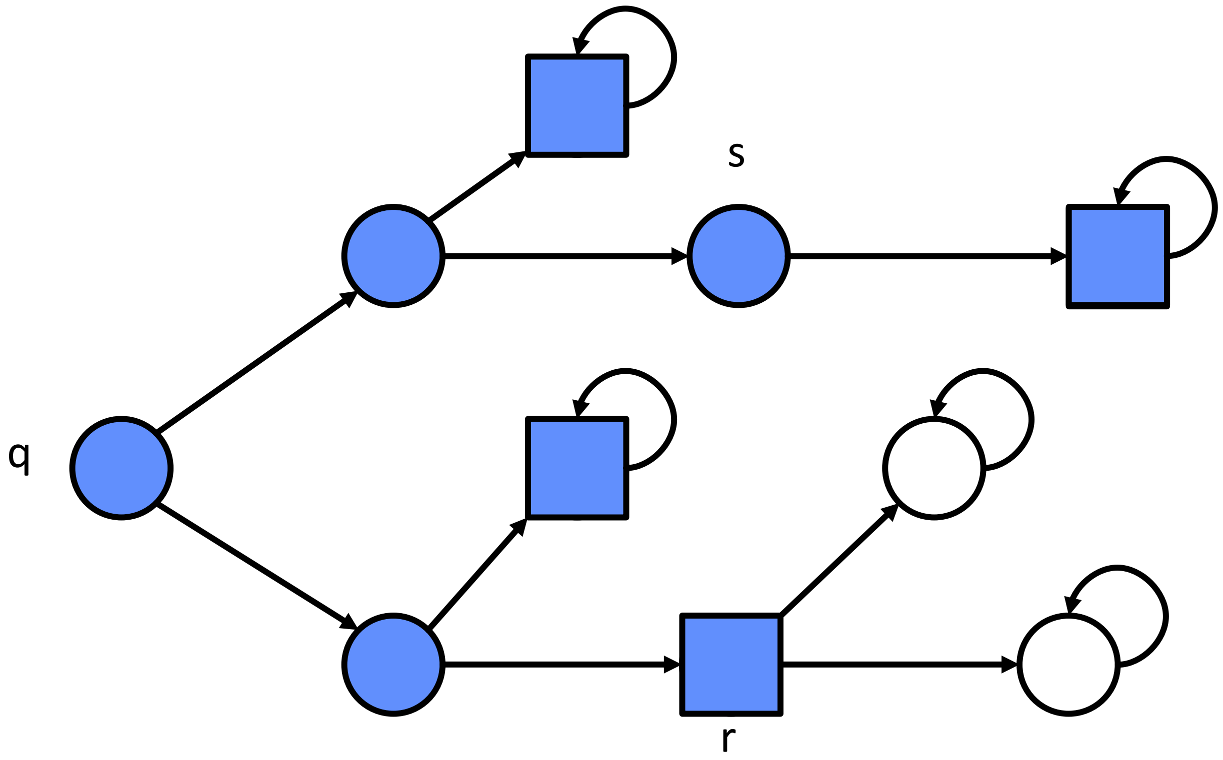
$r \models \phi E U \Psi$

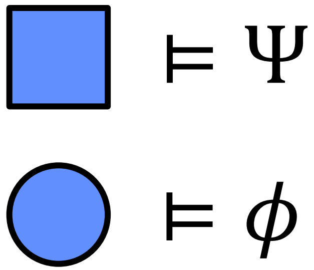
$s \not\models \phi E U \Psi$

Evaluating a CTL formula

$\phi \mathrm{AU} \Psi$: "On all paths, ϕ holds until Ψ holds."

Over paths:	Path-specific:
$\mathrm{A}\phi \rightarrow$ **A**ll ϕ	$\mathrm{X}\phi \rightarrow$ Ne**X**t ϕ
$\mathrm{E}\phi \rightarrow$ **E**xists ϕ	$\mathrm{F}\phi \rightarrow$ **F**inally ϕ
	$\mathrm{G}\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 \mathrm{U} \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2





$q \models \phi \mathrm{AU} \Psi$

$r \models ?$

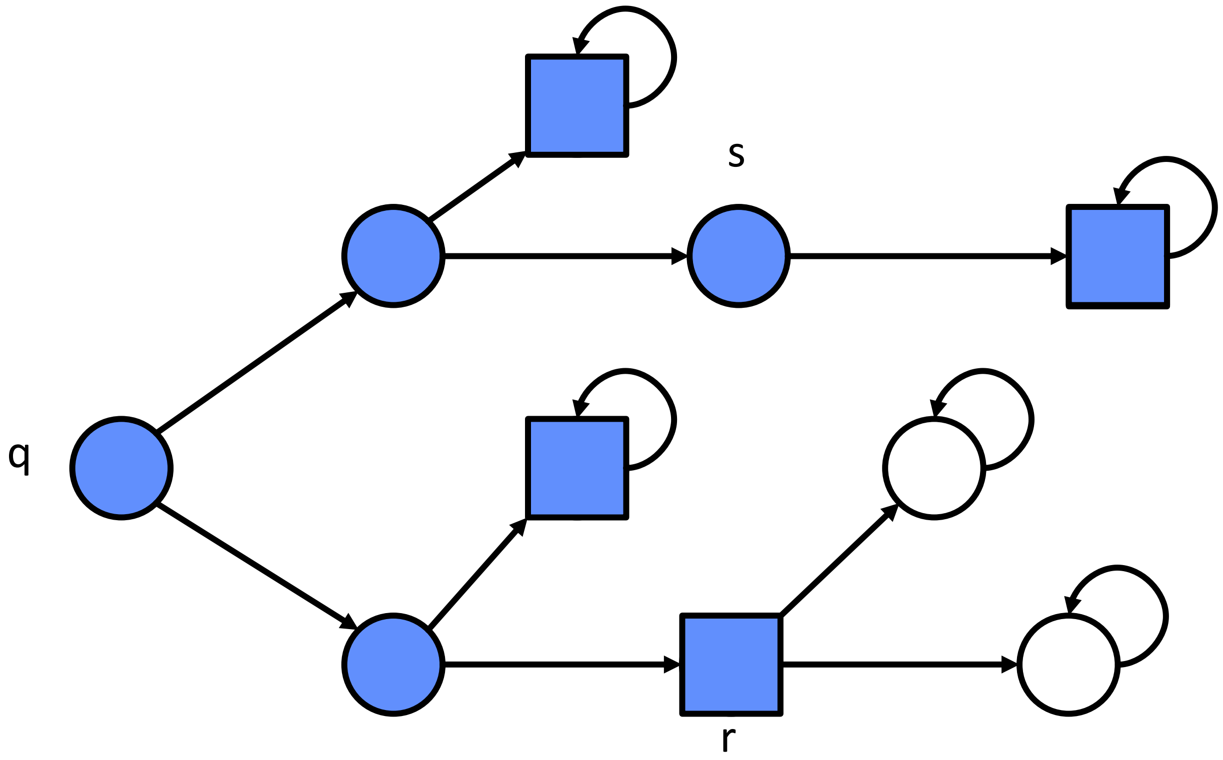
$s \models ?$

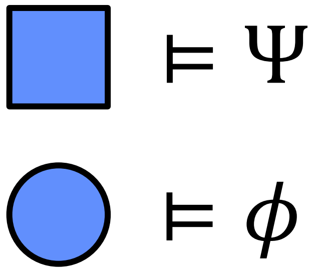
Evaluating a CTL formula

$\phi A U \Psi$: "On all paths, ϕ holds until Ψ holds."

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



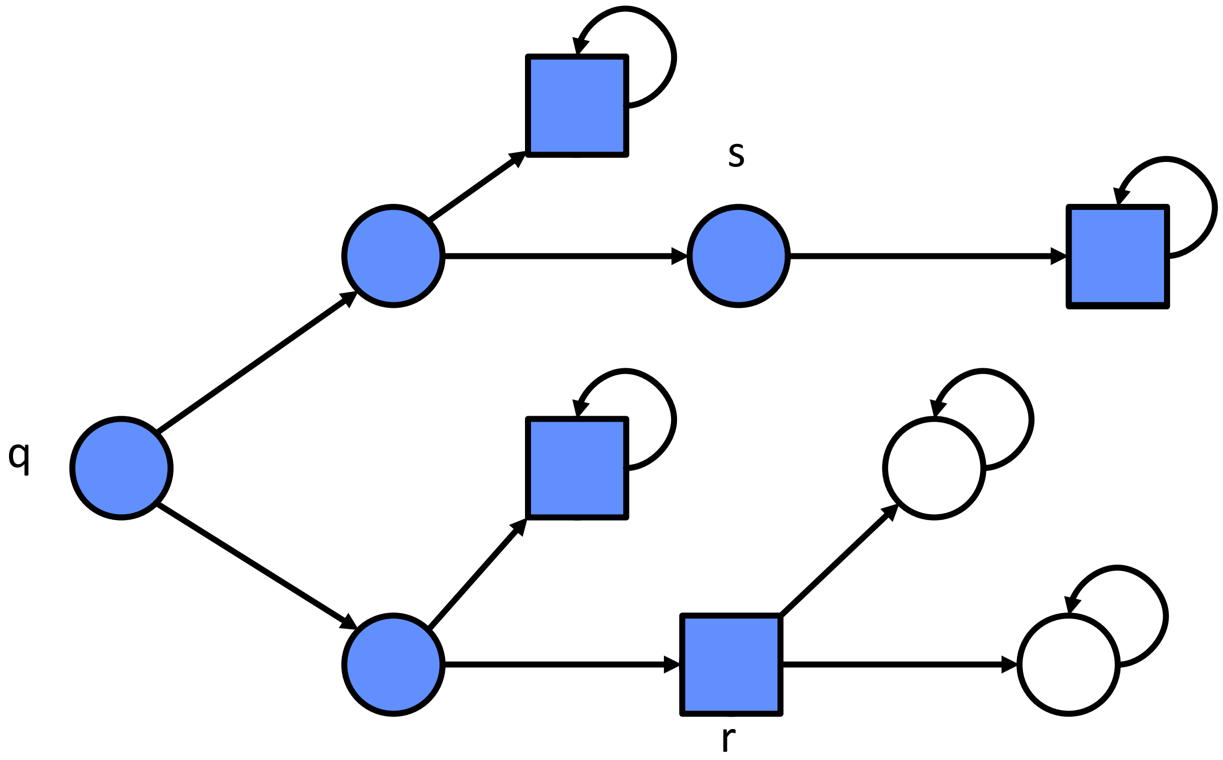


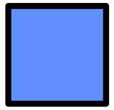
q $\vDash \phi A U \Psi$
r $\vDash \phi A U \Psi$
s $\vDash$?

Evaluating a CTL formula

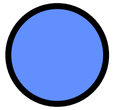
$\phi A U \Psi$: "On all paths, ϕ holds until Ψ holds."

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2





$\vDash \Psi$



$\vDash \phi$

$q \vDash \phi A U \Psi$

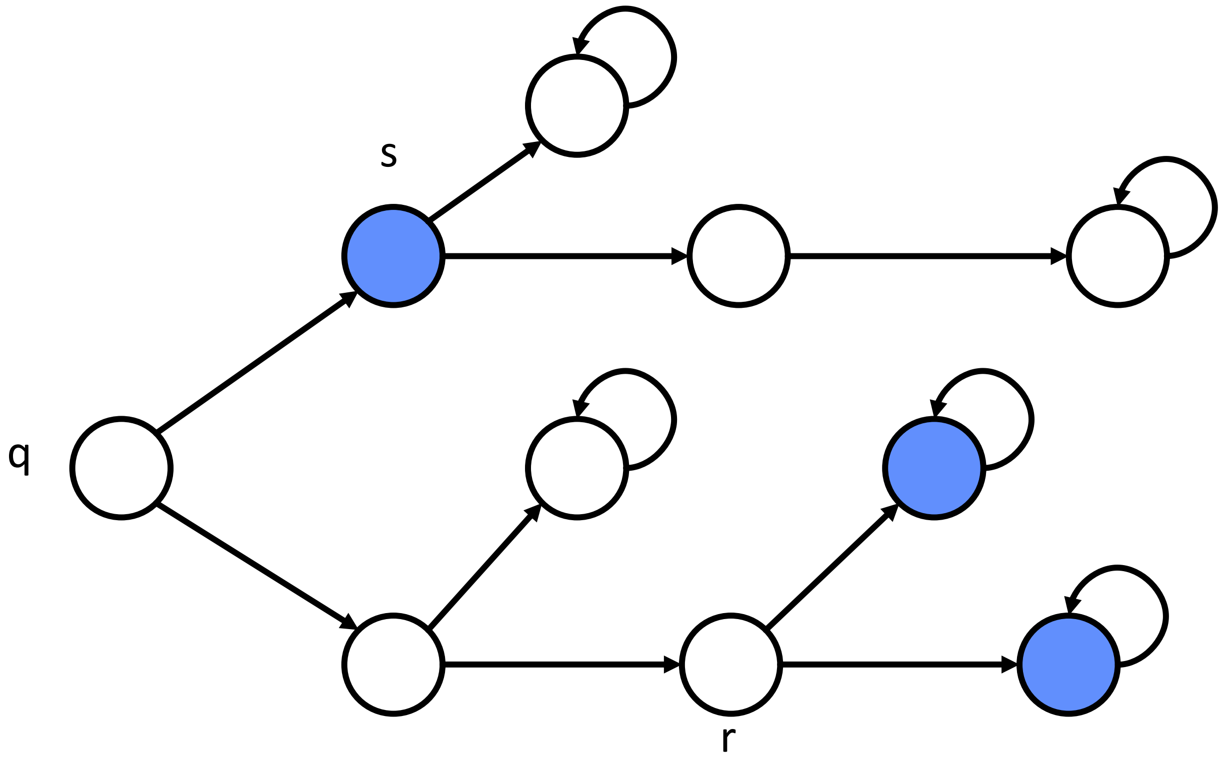
$r \vDash \phi A U \Psi$

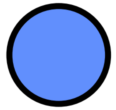
$s \vDash \phi A U \Psi$

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Evaluating a CTL formula

$EX\phi$: "There exists a path along which the next state satisfies ϕ."





$q \vDash EX\phi$

$r \vDash ?$

$s \vDash ?$

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

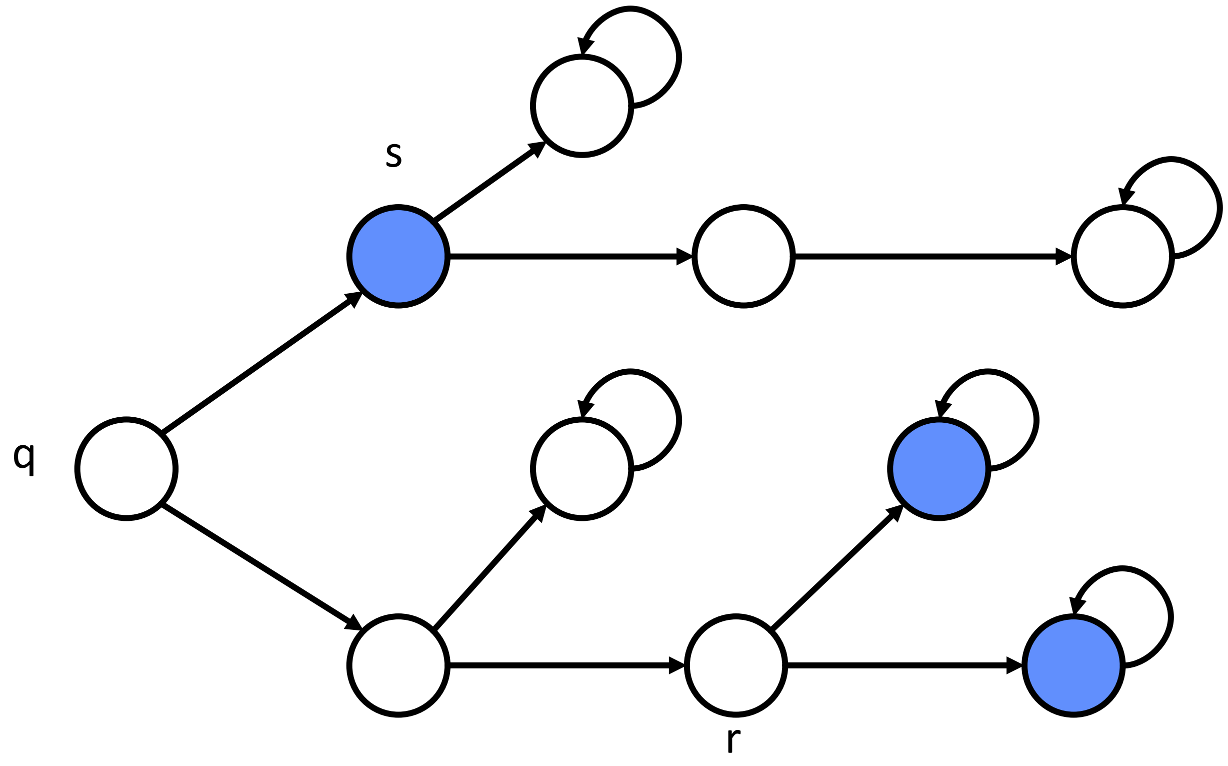
$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Evaluating a CTL formula

$EX\phi$: "There exists a path along which the next state satisfies ϕ."



$\vDash \phi$

$q \vDash EX\phi$

$r \vDash EX\phi$

$s \vDash ?$

Evaluating a CTL formula

EXϕ : "There exists a path along which the next state satisfies ϕ."

Over paths:

Aϕ → **A**ll ϕ

Eϕ → **E**xists ϕ

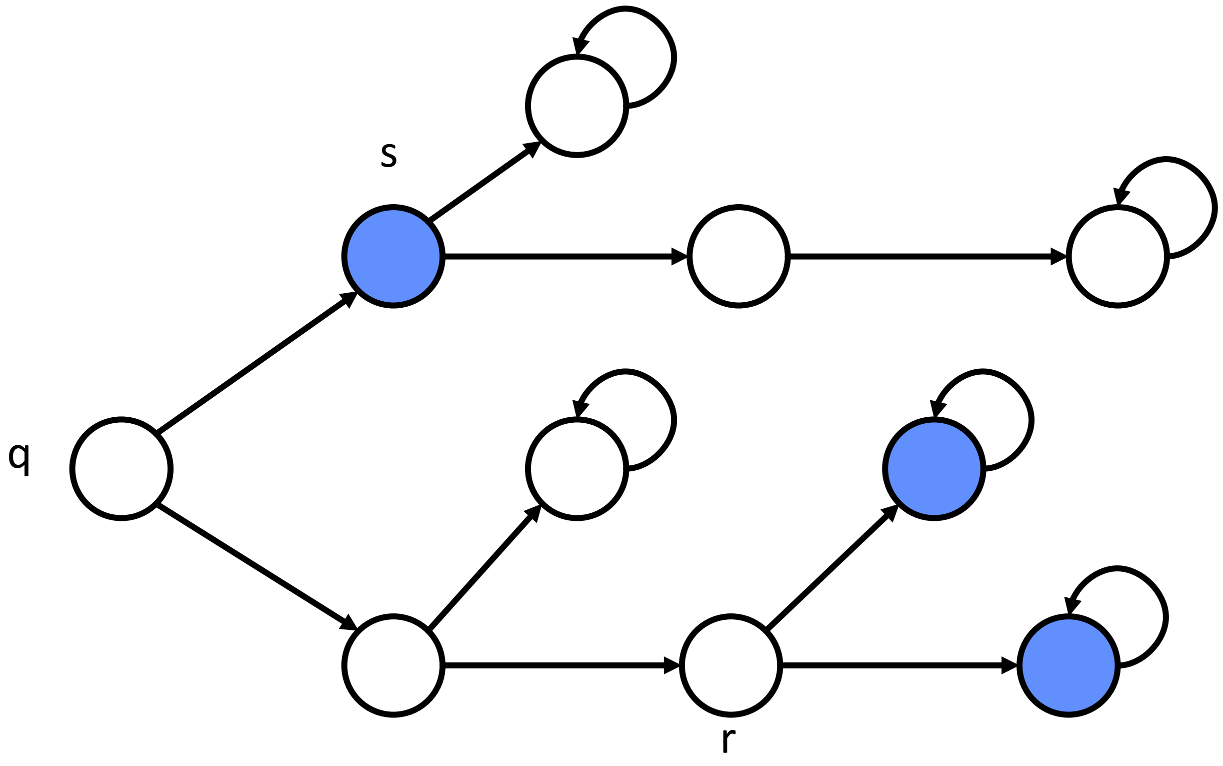
Path-specific:

Xϕ → Ne**X**t ϕ

Fϕ → **F**inally ϕ

Gϕ → **G**lobally ϕ

ϕ_1Uϕ_2 → ϕ_1**U**ntil ϕ_2



$\vDash \phi$

q $\vDash$ EXϕ

r $\vDash$ EXϕ

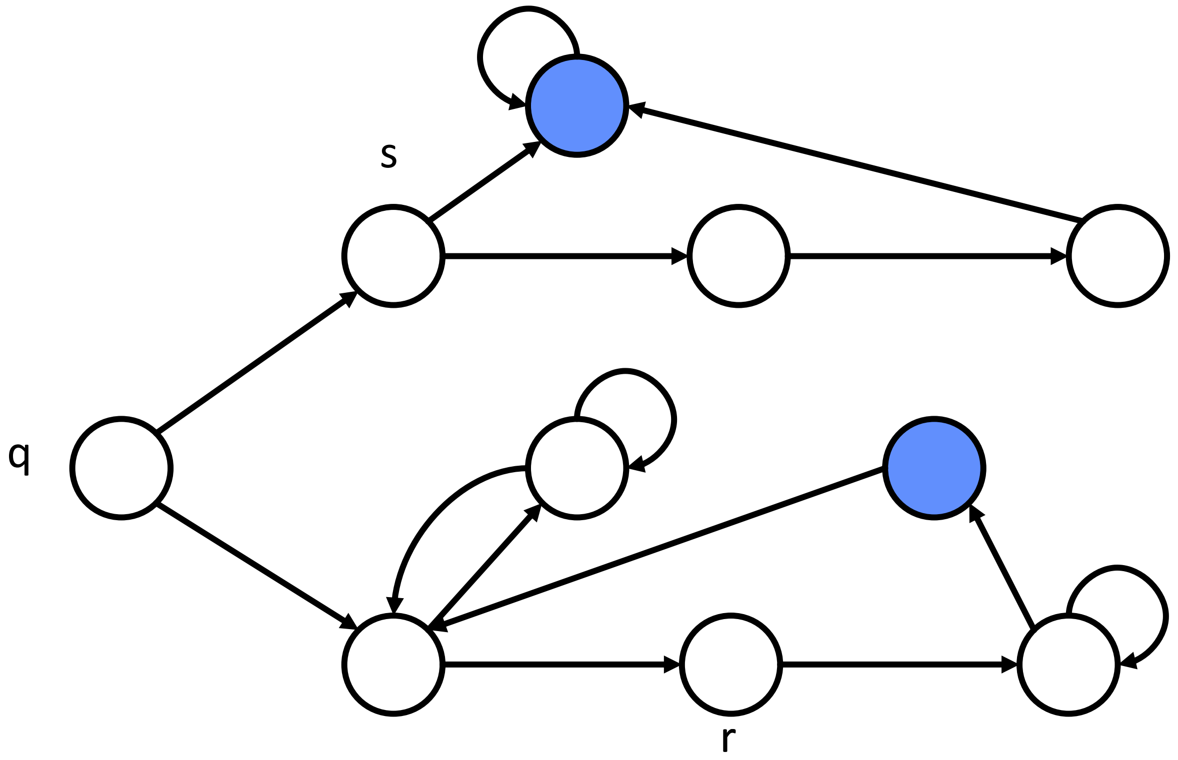
s $\nvDash$ EXϕ

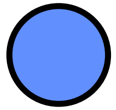
Evaluating a CTL formula

AG EF ϕ : "On all paths and for all states, there exists a path along which at some state ϕ holds."

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2





$q \models$ AG EF ϕ

$r \models$?

$s \models$?

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

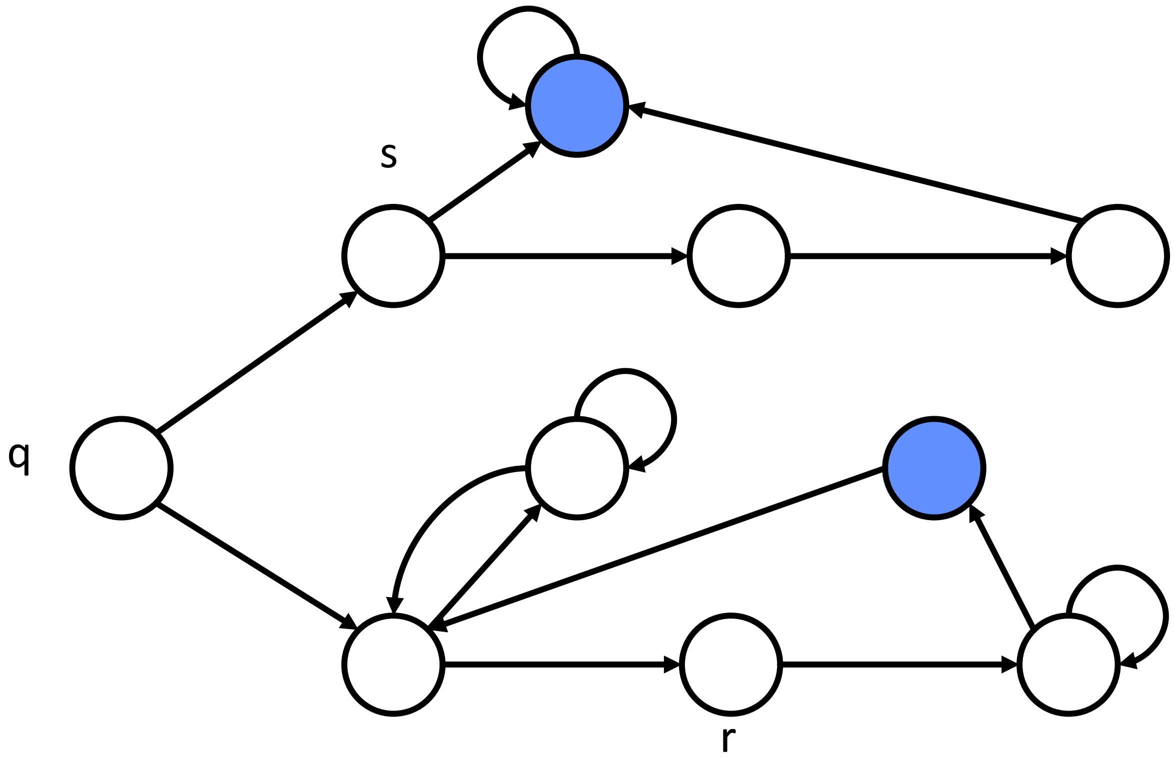
$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Evaluating a CTL formula

AG EF ϕ : "On all paths and for all states, there exists a path along which at some state ϕ holds."



$\bullet \models \phi$

$q \models$ AG EF ϕ

$r \models$ AG EF ϕ

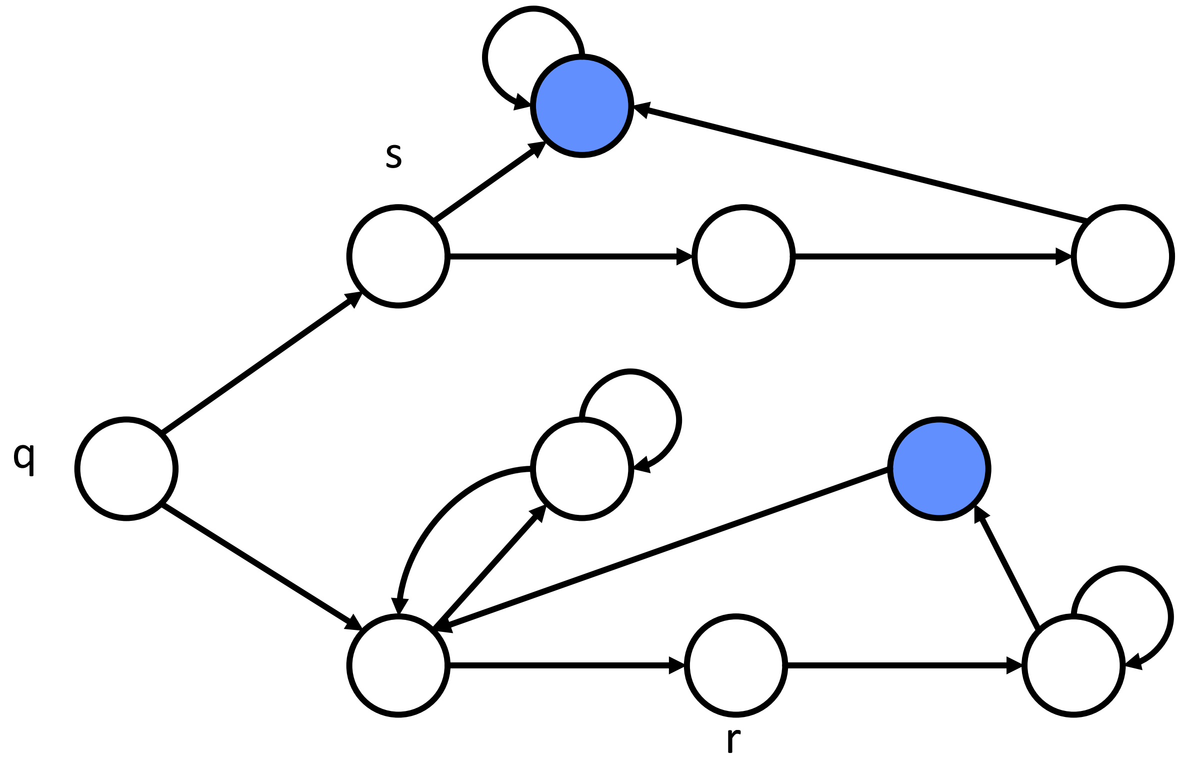
$s \models$?

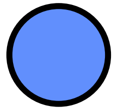
Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Evaluating a CTL formula

AG EF ϕ : "On all paths and for all states, there exists a path along which at some state ϕ holds."



 $\vDash \phi$

q $\vDash$ AG EF ϕ

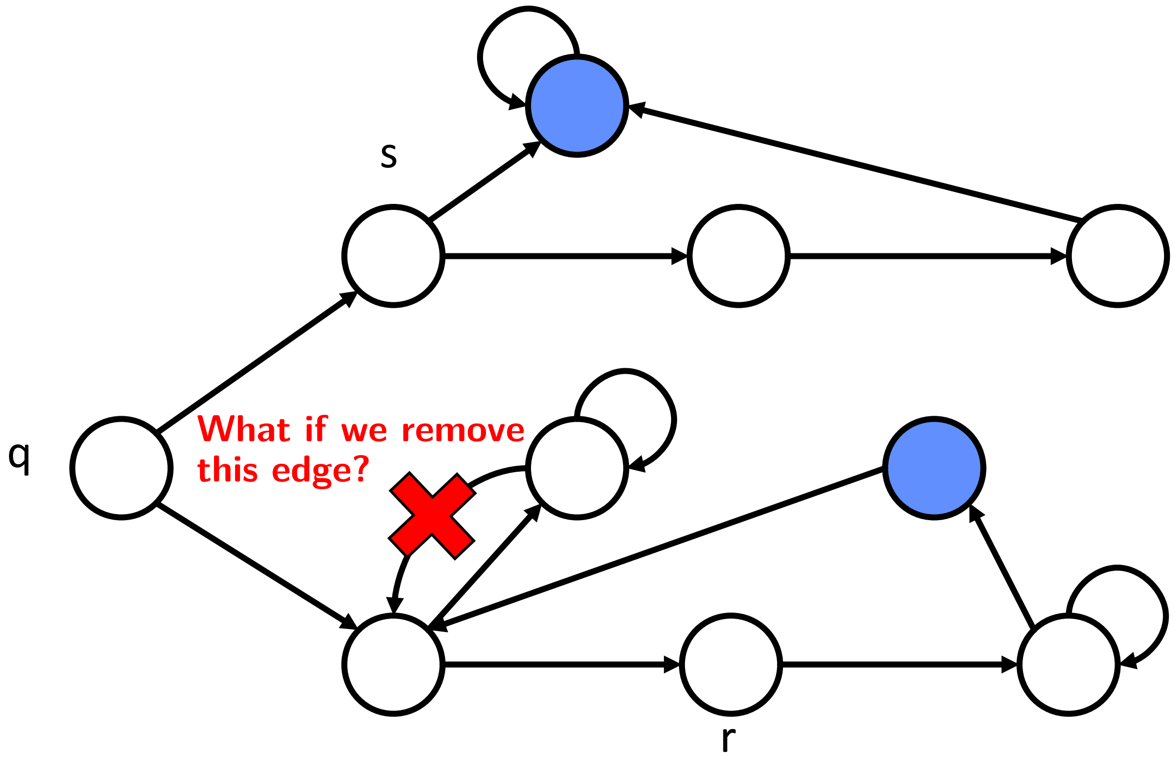
r $\vDash$ AG EF ϕ

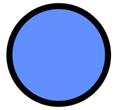
s $\vDash$ AG EF ϕ

Evaluating a CTL formula

AG EF ϕ : "On all paths and for all states, there exists a path along which at some state ϕ holds."

Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2





q ⊨ ?

r ⊨ ?

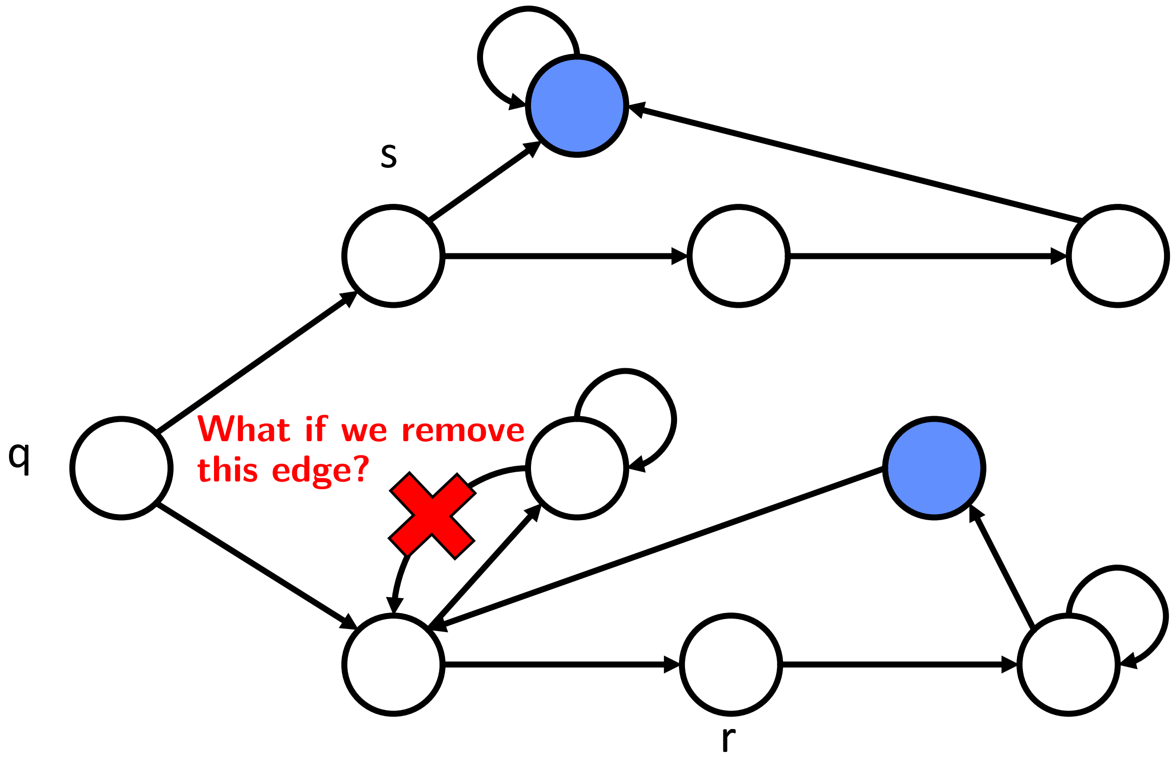
s ⊨ ?

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Evaluating a CTL formula

AG EF ϕ : "On all paths and for all states, there exists a path along which at some state ϕ holds."



$\bullet \models \phi$

$q \not\models AG\ EF\ \phi$

$r \not\models AG\ EF\ \phi$

$s \models AG\ EF\ \phi$

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

Over paths:

$A\phi \rightarrow$ **All** ϕ

$E\phi \rightarrow$ **Exists** ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p

Over paths:
$A\phi \to$ **A**ll ϕ
$E\phi \to$ **E**xists ϕ

Path-specific:
$X\phi \to$ Ne**X**t ϕ
$F\phi \to$ **F**inally ϕ
$G\phi \to$ **G**lobally ϕ
$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.

Over paths:

$A\phi \rightarrow$ **All** ϕ

$E\phi \rightarrow$ **Exists** ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.
- EF p

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.
- EF p — It's possible I may like chocolate someday, at least for one day.

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.
- EF p — It's possible I may like chocolate someday, at least for one day.
- AF EG p

Over paths:

$A\phi \rightarrow$ **All** ϕ

$E\phi \rightarrow$ **Exists** ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

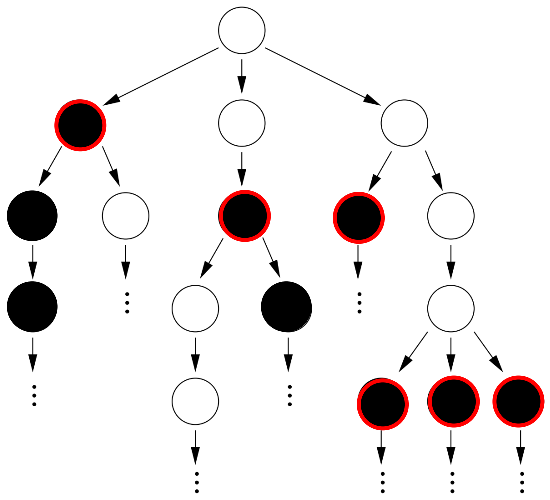
$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.
- EF p — It's possible I may like chocolate someday, at least for one day.
- AF EG p — There will be always sometime in the future (AF) that I may suddenly start liking chocolate for the rest of time (EG).



Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.
- EF p — It's possible I may like chocolate someday, at least for one day.
- AF EG p — There will be always sometime in the future (AF) that I may suddenly start liking chocolate for the rest of time (EG).
- EG AF p

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

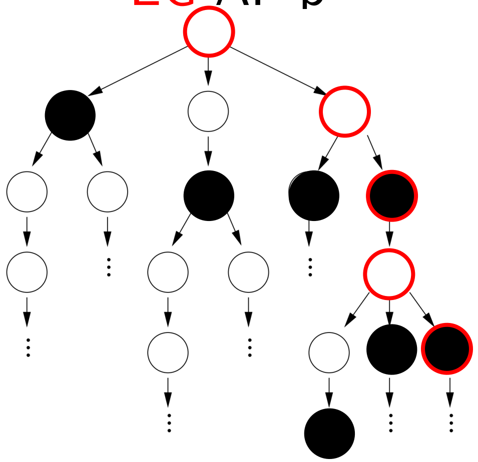
$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.
- EF p — It's possible I may like chocolate someday, at least for one day.
- AF EG p — There will be always sometime in the future (AF) that I may suddenly start liking chocolate for the rest of time (EG).
- EG AF p — This is a critical time in my life. Depending on what happens (E), it's possible that for the rest of time (G), there will always be some time in the future (AF) when I will like chocolate. However, if the wrong thing happens next, then all bets are off and there's no guarantee about whether I will ever like chocolate.



Over paths:	Path-specific:
$A\phi$ → **All** ϕ	$X\phi$ → Ne**X**t ϕ
$E\phi$ → **Exists** ϕ	$F\phi$ → **F**inally ϕ
	$G\phi$ → **G**lobally ϕ
	$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.
- EF p — It's possible I may like chocolate someday, at least for one day.
- AF EG p — There will be always sometime in the future (AF) that I may suddenly start liking chocolate for the rest of time (EG).
- EG AF p — This is a critical time in my life. Depending on what happens (E), it's possible that for the rest of time (G), there will always be some time in the future (AF) when I will like chocolate. However, if the wrong thing happens next, then all bets are off and there's no guarantee about whether I will ever like chocolate.
- p AU q

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Interpreting a CTL formula

Encoding	Proposition
p	I like chocolate
q	It's warm outside

- AG p — I will like chocolate from now on, no matter what happens.
- EF p — It's possible I may like chocolate someday, at least for one day.
- AF EG p — There will be always sometime in the future (AF) that I may suddenly start liking chocolate for the rest of time (EG).
- EG AF p — This is a critical time in my life. Depending on what happens (E), it's possible that for the rest of time (G), there will always be some time in the future (AF) when I will like chocolate. However, if the wrong thing happens next, then all bets are off and there's no guarantee about whether I will ever like chocolate.
- p AU q — No matter what happens, I will like chocolate from now on. But when it gets warm outside, I don’t know whether I still like it. And it will get warm outside someday.

Over paths:
$A\phi \to$ **A**ll ϕ
$E\phi \to$ **E**xists ϕ

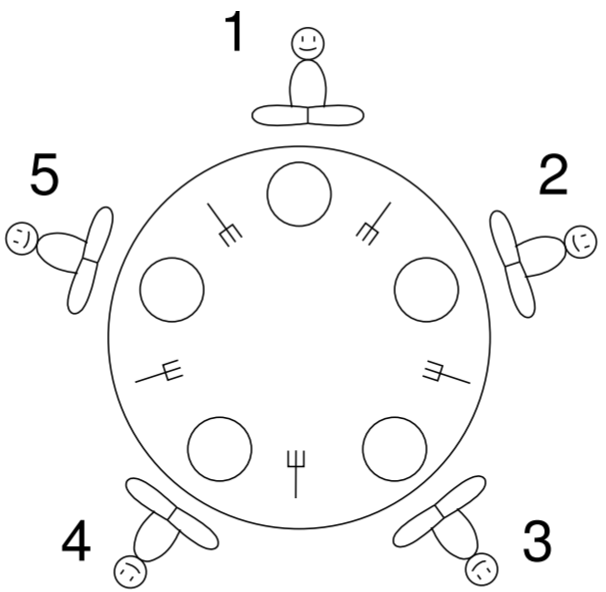
Path-specific:
$X\phi \to$ Ne**X**t ϕ
$F\phi \to$ **F**inally ϕ
$G\phi \to$ **G**lobally ϕ
$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

Specifying using a CTL formula

Famous problem

Dining Philosophers

- Five philosophers are sitting around a table, taking turns at thinking and eating.
- Each needs two forks to eat.
- They put down forks only once they have eaten.
- There are only five forks.



Atomic proposition

e_i : Philosopher i is currently eating.

Specifying using a CTL formula

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

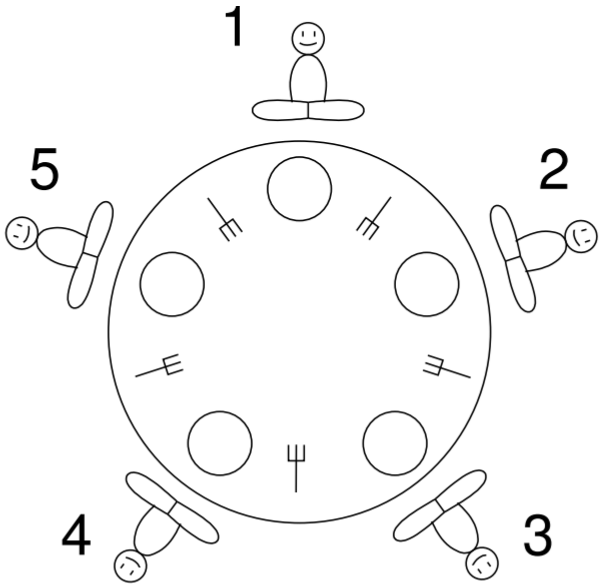
$X\phi \rightarrow$ Ne**X**t ϕ

$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

- "Philosophers 1 and 4 will never eat at the same time."
- "Every philosopher will get infinitely many turns to eat."
- "Philosopher 2 will be the first to eat."



Over paths:
$A\phi \to$ **A**ll ϕ
$E\phi \to$ **E**xists ϕ

Path-specific:
$X\phi \to$ Ne**X**t ϕ
$F\phi \to$ **F**inally ϕ
$G\phi \to$ **G**lobally ϕ
$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

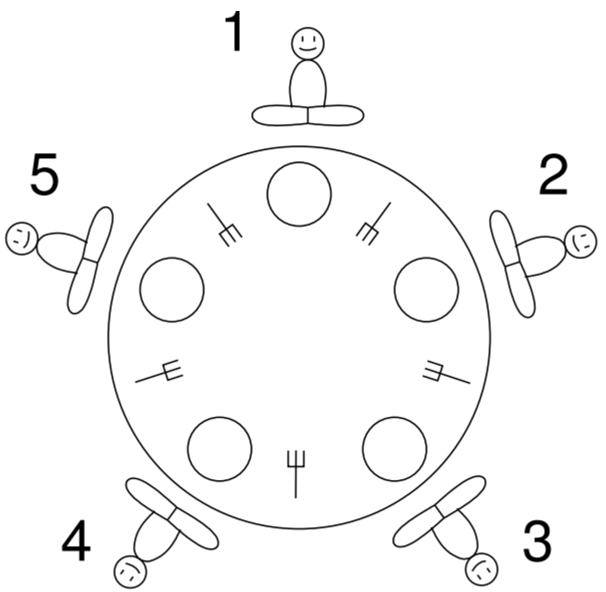
Specifying using a CTL formula

- "Philosophers 1 and 4 will never eat at the same time."

$$\mathrm{AG}\neg(e_1 \cdot e_4)$$

- "Every philosopher will get infinitely many turns to eat."

- "Philosopher 2 will be the first to eat."



Specifying using a CTL formula

Over paths:	Path-specific:
$A\phi$ → **A**ll ϕ	$X\phi$ → Ne**X**t ϕ
$E\phi$ → **E**xists ϕ	$F\phi$ → **F**inally ϕ
	$G\phi$ → **G**lobally ϕ
	$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

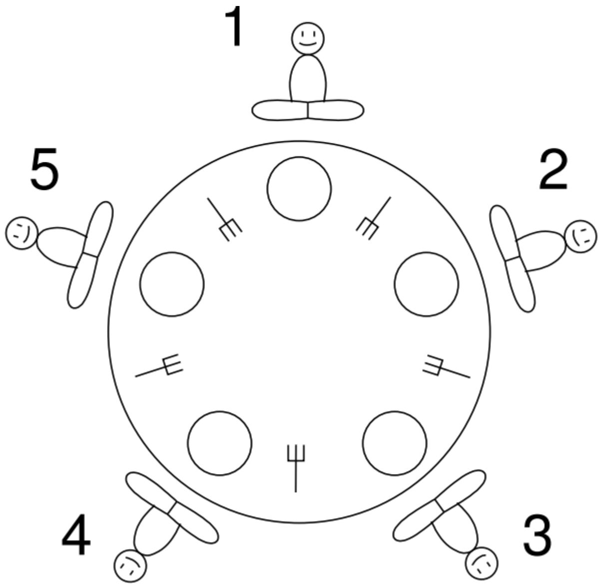
- "Philosophers 1 and 4 will never eat at the same time."

$$\mathrm{AG}\neg(e_1 \cdot e_4)$$

- "Every philosopher will get infinitely many turns to eat."

$$\mathrm{AG}(\mathrm{AF}e_1 \cdot \mathrm{AF}e_2 \cdot \mathrm{AF}e_3 \cdot \mathrm{AF}e_4 \cdot \mathrm{AF}e_5)$$

- "Philosopher 2 will be the first to eat."



Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Specifying using a CTL formula

- "Philosophers 1 and 4 will never eat at the same time."

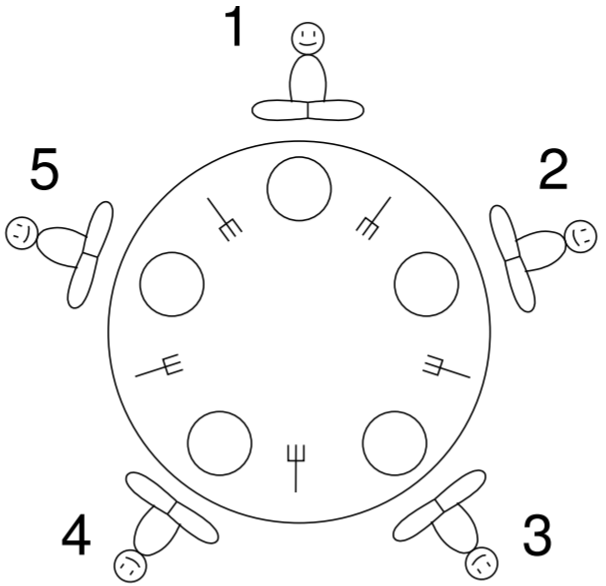
$$\mathrm{AG}\neg(e_1 \cdot e_4)$$

- "Every philosopher will get infinitely many turns to eat."

$$\mathrm{AG}(\mathrm{AF}e_1 \cdot \mathrm{AF}e_2 \cdot \mathrm{AF}e_3 \cdot \mathrm{AF}e_4 \cdot \mathrm{AF}e_5)$$

- "Philosopher 2 will be the first to eat."

$$\neg(e_1 + e_3 + e_4 + e_5)\,\mathrm{AU}\,e_2$$



Over paths:
$A\phi \to$ **A**ll ϕ
$E\phi \to$ **E**xists ϕ

Path-specific:
$X\phi \to$ Ne**X**t ϕ
$F\phi \to$ **F**inally ϕ
$G\phi \to$ **G**lobally ϕ
$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

Computing a CTL formula

1. Define $[\![\phi]\!]$ as the set of all states of the finite automaton for which CTL formula ϕ is true.
2. A finite automaton with initial state q_0 satisfies ϕ iff

$$q_0 \in [\![\phi]\!]$$

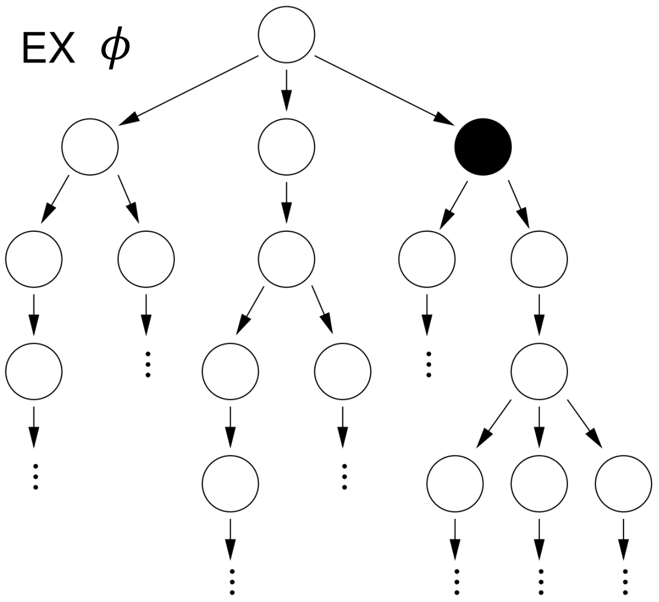
- Now, we can use our "trick": computing with sets of states!
  - $\psi_{[\![\phi]\!]}(q)$ is true if the state q is in the set $[\![\phi]\!]$, i.e., it is a state for which the CTL formula is true.
  - Therefore, we can also say

$$q_0 \in [\![\phi]\!] \equiv \psi_{[\![\phi]\!]}(q_0)$$

characteristic function of the set $[\![\phi]\!]$

Computing a CTL formula: EX ϕ

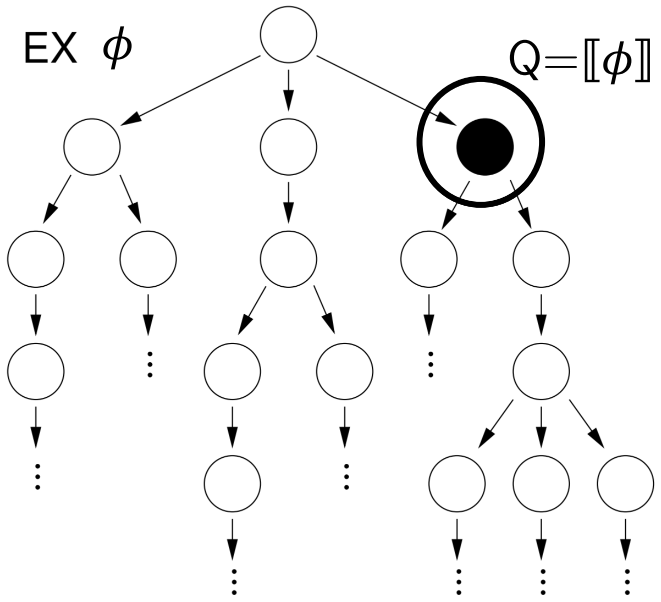
Over paths:	Path-specific:
$A\phi$ → **A**ll ϕ	$X\phi$ → Ne**X**t ϕ
$E\phi$ → **E**xists ϕ	$F\phi$ → **F**inally ϕ
	$G\phi$ → **G**lobally ϕ
	$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2



Computing a CTL formula: EX ϕ

Over paths:	Path-specific:
$A\phi$ → **All** ϕ	$X\phi$ → Ne**X**t ϕ
$E\phi$ → **Exists** ϕ	$F\phi$ → **F**inally ϕ
	$G\phi$ → **G**lobally ϕ
	$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

- Suppose that Q is the set of states for which the formula ϕ is true.



Sets	$Q = [\![\phi]\!]$
Characteristic functions	$\psi_Q(q)$

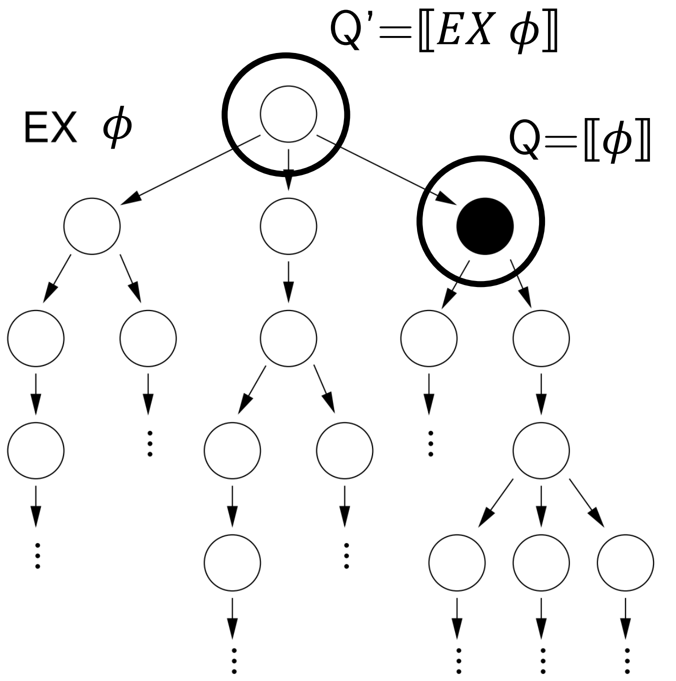
Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: EX ϕ

- Suppose that Q is the set of states for which the formula ϕ is true.
- Q' is the set of predecessor states of Q, i.e., the set of states that lead in one transition to a state in Q:

$$Q' = Pre(Q, \delta) = \{q' \mid \exists q : \psi_\delta(q', q) \cdot \psi_Q(q)\}$$



Sets $\quad Q = [\![\phi]\!] \longrightarrow Q' = [\![\mathrm{EX}\phi]\!] = \mathrm{Pre}([\![\phi]\!], \delta)$

Characteristic functions $\quad \psi_Q(q) \longrightarrow \psi_{Q'}(q') = (\exists q \; : \; \psi_Q(q) \cdot \psi_\delta(q', q))$

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

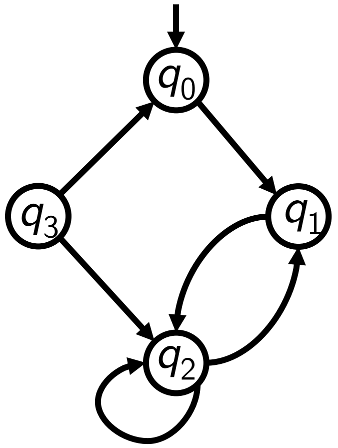
$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: EX ϕ

- Example for EX ϕ : Compute EX q_2



1. Define $[\![EX\ q_2]\!]$: set of all states of the finite automaton for which CTL formula $\mathrm{EX}\, q_2$ is true.

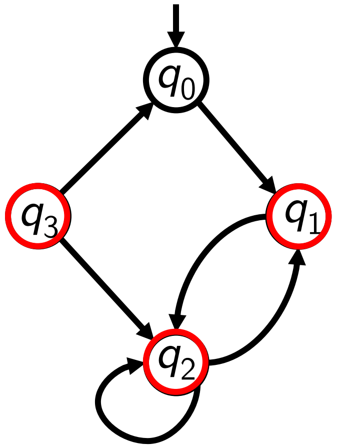
$[\![q_2]\!] = \{q_2\}$

Over paths:
$A\phi \rightarrow$ **All** ϕ
$E\phi \rightarrow$ **Exists** ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: EX ϕ

- Example for EX ϕ : Compute EX q_2



1. Define $[\![EX\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EX q_2 is true.

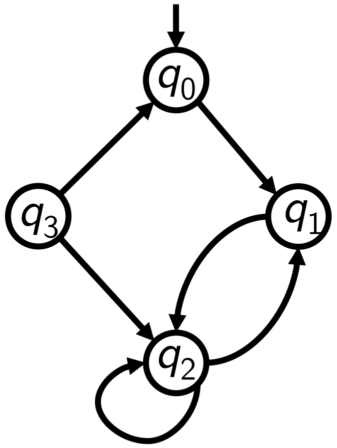
$[\![q_2]\!] = \{q_2\}$

$$Q' = [\![EX\ q_2]\!] = \underbrace{Pre(\{q_2\}, \delta)}_{\{q' \mid \exists q : \psi_\delta(q', q) \cdot \psi_Q(q)\}} = \{q_1, q_2, q_3\}$$

Over paths:	Path-specific:
$A\phi$ → **A**ll ϕ	$X\phi$ → Ne**X**t ϕ
$E\phi$ → **E**xists ϕ	$F\phi$ → **F**inally ϕ
	$G\phi$ → **G**lobally ϕ
	$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

Computing a CTL formula: EX ϕ

- Example for EX ϕ : Compute EX q_2



1. Define $[\![EX\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EX q_2 is true.

$[\![q_2]\!] = \{q_2\}$

$Q' = [\![EX\ q_2]\!] = Pre(\{q_2\}, \delta) = \{q_1, q_2, q_3\}$

$\{q' \mid \exists q : \psi_\delta(q', q) \cdot \psi_Q(q)\}$

2. A finite automaton with initial state q_0 satisfies EX q_2 iff $q_0 \in [\![EX\ q_2]\!]$

As $q_0 \notin [\![EX\ q_2]\!] = \{q_1, q_2, q_3\}$, the CTL formula EX q_2 is not true.

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

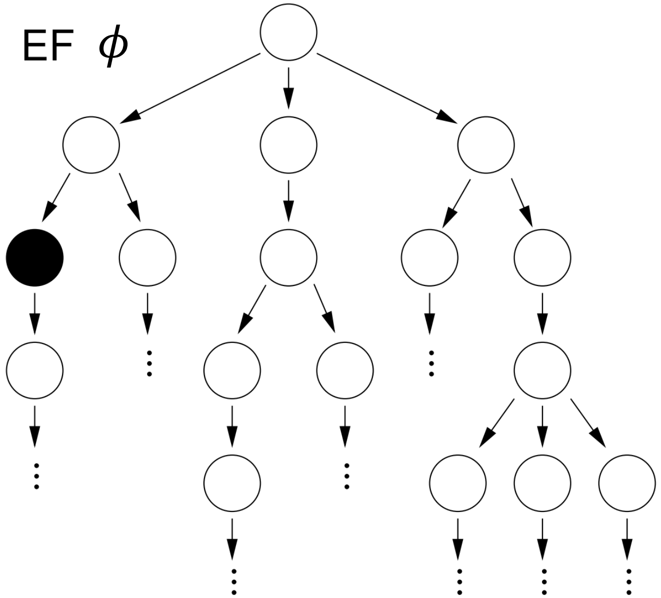
Computing a CTL formula: EF ϕ

- Start with the set of states for which the formula ϕ is true.
- Add to this set the set of predecessor states. Repeat for the resulting set of states,…, until we reach a fixed point.

$$Q_0 = [\![\phi]\!]$$

$$Q_i = Q_{i-1} \cup \mathrm{Pre}(Q_{i-1}, \delta)$$ for all $i > 1$ until a fixed point Q' is reached

$$[\![\mathrm{EF}\phi]\!] = Q'$$

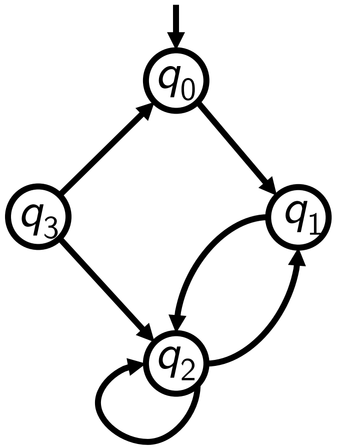


Computing a CTL formula: EF ϕ

Over paths:
Aϕ → **A**ll ϕ
Eϕ → **E**xists ϕ

Path-specific:
Xϕ → Ne**X**t ϕ
Fϕ → **F**inally ϕ
Gϕ → **G**lobally ϕ
ϕ_1Uϕ_2 → ϕ_1 **U**ntil ϕ_2

- Example for EFϕ: Compute EF q_2



1. Define $[\![EF\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EF q_2 is true.

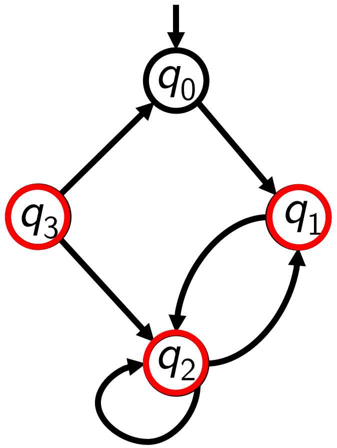
$$Q_0 = [\![q_2]\!] = \{q_2\}$$

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: EF ϕ

- Example for EFϕ: Compute EF q_2



1. Define $[\![EF\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EF q_2 is true.

$$Q_0 = [\![q_2]\!] = \{q_2\}$$

$$Q_1 = \{q_2\} \cup \underline{\mathrm{Pre}(\{q_2\}, \delta)} = \{q_1, q_2, q_3\}$$

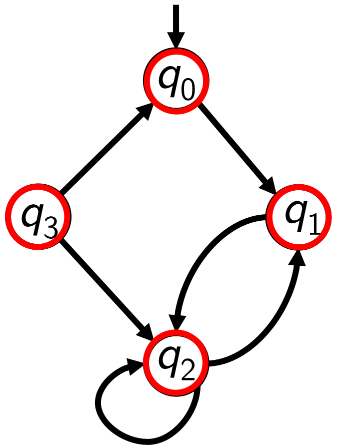
$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_1, q_2, q_3\}$$

Over paths:
$A\phi \to$ **A**ll ϕ
$E\phi \to$ **E**xists ϕ

Path-specific:
$X\phi \to$ Ne**X**t ϕ
$F\phi \to$ **F**inally ϕ
$G\phi \to$ **G**lobally ϕ
$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: EF ϕ

- Example for EFϕ: Compute EF q_2



1. Define $[\![EF\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EF q_2 is true.

$$Q_0 = [\![q_2]\!] = \{q_2\}$$

$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_1, q_2, q_3\}$$

$$Q_1 = \{q_2\} \cup \mathrm{Pre}(\{q_2\}, \delta) = \{q_1, q_2, q_3\}$$

$$Q_2 = \{q_1, q_2, q_3\} \cup \mathrm{Pre}(\{q_1, q_2, q_3\}, \delta) = \{q_0, q_1, q_2, q_3\}$$

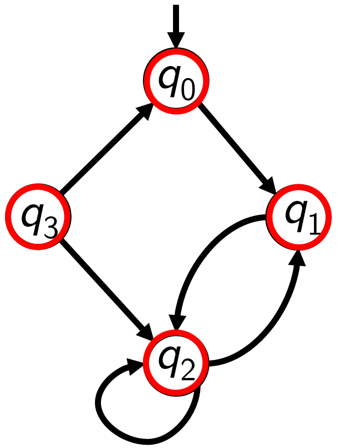
Over paths:
$A\phi$ → **A**ll ϕ
$E\phi$ → **E**xists ϕ

Path-specific:
$X\phi$ → Ne**X**t ϕ
$F\phi$ → **F**inally ϕ
$G\phi$ → **G**lobally ϕ
$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

Computing a CTL formula: EF ϕ

- Example for EFϕ: Compute EF q_2

1. Define $[\![EF\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EF q_2 is true.



$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_1, q_2, q_3\}$$

$$Q_0 = [\![q_2]\!] = \{q_2\}$$

$$Q_1 = \{q_2\} \cup \text{Pre}(\{q_2\}, \delta) = \{q_1, q_2, q_3\}$$

$$Q_2 = \{q_1, q_2, q_3\} \cup \text{Pre}(\{q_1, q_2, q_3\}, \delta) = \{q_0, q_1, q_2, q_3\}$$

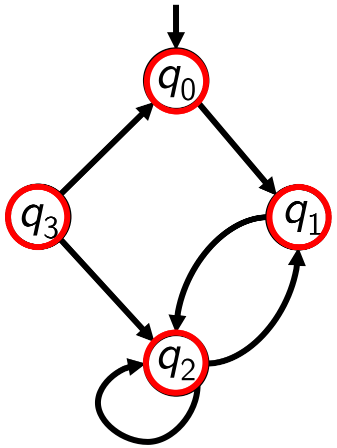
$$Q_3 = \{q_0, q_1, q_2, q_3\} \cup \text{Pre}(\{q_0, q_1, q_2, q_3\}, \delta) = \{q_0, q_1, q_2, q_3\}$$

$$[\![\text{EF} q_2]\!] = Q_3 = \{q_0, q_1, q_2, q_3\}$$

Computing a CTL formula: EF ϕ

Over paths:	Path-specific:
$A\phi$ → **All** ϕ	$X\phi$ → Ne**X**t ϕ
$E\phi$ → **Exists** ϕ	$F\phi$ → **F**inally ϕ
	$G\phi$ → **G**lobally ϕ
	$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

- Example for EFϕ: Compute EF q_2



1. Define $[\![EF\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EF q_2 is true.

$$Q_0 = [\![q_2]\!] = \{q_2\}$$

$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_1, q_2, q_3\}$$

$$Q_1 = \{q_2\} \cup \mathrm{Pre}(\{q_2\}, \delta) = \{q_1, q_2, q_3\}$$

$$Q_2 = \{q_1, q_2, q_3\} \cup \mathrm{Pre}(\{q_1, q_2, q_3\}, \delta) = \{q_0, q_1, q_2, q_3\}$$

$$Q_3 = \{q_0, q_1, q_2, q_3\} \cup \underline{\mathrm{Pre}(\{q_0, q_1, q_2, q_3\}, \delta)} = \{q_0, q_1, q_2, q_3\}$$

$$[\![\mathrm{EF} q_2]\!] = Q_3 = \{q_0, q_1, q_2, q_3\}$$

2. A finite automaton with initial state q_0 satisfies EF q_2 iff $q_0 \in [\![EF\ q_2]\!]$

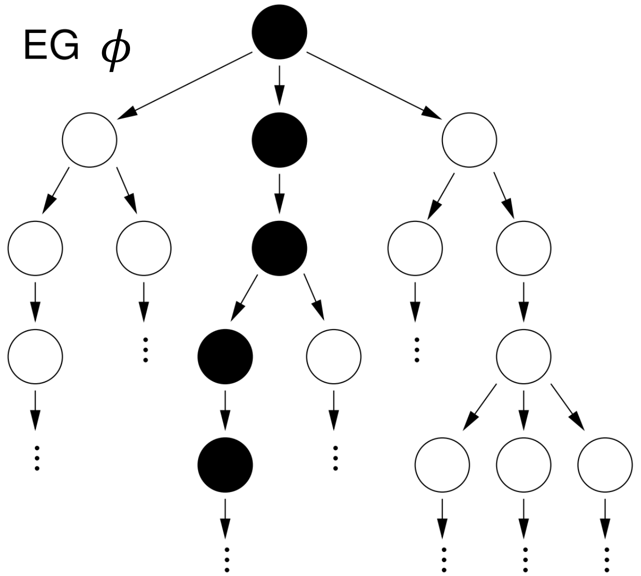
As $q_0 \in [\![\mathrm{EF} q_2]\!] = \{q_0, q_1, q_2, q_3\}$, the CTL formula EF q_2 is true.

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: EG ϕ

- Start with the set of states for which the formula ϕ is true.
- Cut this set with the set of predecessor states. Repeat for the resulting set of states,..., until we reach a fixed point.



$$Q_0 = [\![\phi]\!]$$

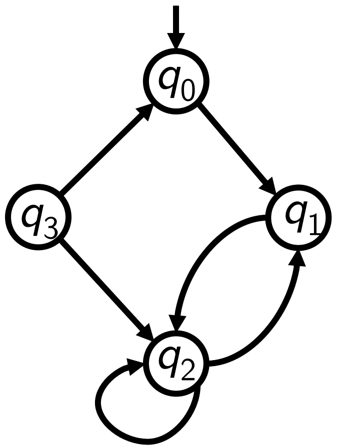
$$Q_i = Q_{i-1} \cap \mathrm{Pre}(Q_{i-1}, \delta)$$ for all $i > 1$ until a fixed point Q' is reached

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: EG ϕ

- Example for EG ϕ: Compute EG q_2



1. Define $[\![EG\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EG q_2 is true.

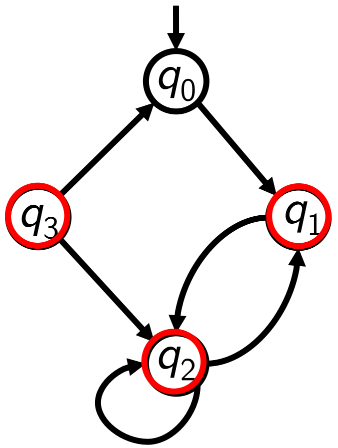
$$Q_0 = [\![q_2]\!] = \{q_2\}$$

Computing a CTL formula: EG ϕ

Over paths:
$A\phi \to$ **All** ϕ
$E\phi \to$ **Exists** ϕ

Path-specific:
$X\phi \to$ Ne**X**t ϕ
$F\phi \to$ **F**inally ϕ
$G\phi \to$ **G**lobally ϕ
$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

- Example for EG ϕ: Compute EG q_2



1. Define $[\![EG\ q_2]\!]$: set of all states of the finite automaton for which CTL formula EG q_2 is true.

$$Q_0 = [\![q_2]\!] = \{q_2\}$$

$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_1, q_2, q_3\}$$

$$Q_1 = \{q_2\} \cap \underline{\mathrm{Pre}(\{q_2\}, \delta)} = \{q_2\}$$

$$[\![\mathrm{EG} q_2]\!] = Q_2 = \{q_2\}$$

2. A finite automaton with initial state q_0 satisfies EG q_2 iff $q_0 \in [\![EG\ q_2]\!]$

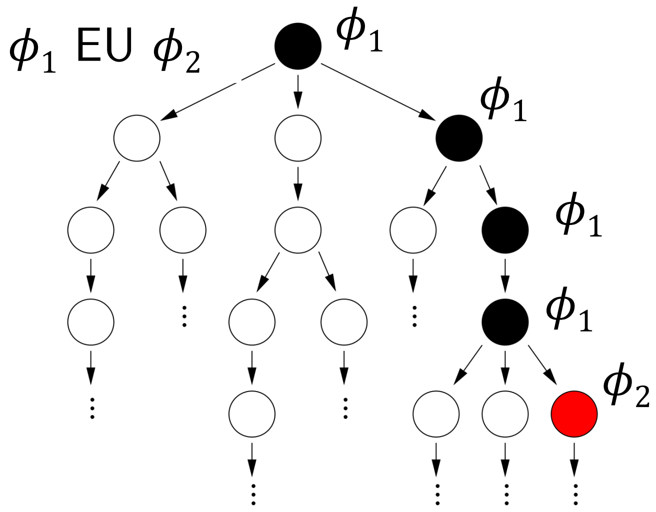
As $q_0 \notin [\![\mathrm{EG} q_2]\!] = \{q_2\}$, the CTL formula EG q_2 is not true.

Over paths:
Aϕ → **A**ll ϕ
Eϕ → **E**xists ϕ

Path-specific:
Xϕ → Ne**X**t ϕ
Fϕ → **F**inally ϕ
Gϕ → **G**lobally ϕ
ϕ_1Uϕ_2 → ϕ_1**U**ntil ϕ_2

Computing a CTL formula: $\phi_1 EU \phi_2$

- Start with the set of states for which the formula ϕ_2 is true.
- Add to this set the set of predecessor states for which the formula ϕ_1 is true. Repeat for the resulting set of states we do the same,…, until we reach a fixed point.
- Like EF ϕ_2; the only difference is that, on our path backwards, we always make sure that also ϕ_1 holds.



$$Q_0 = [\![\phi_2]\!]$$

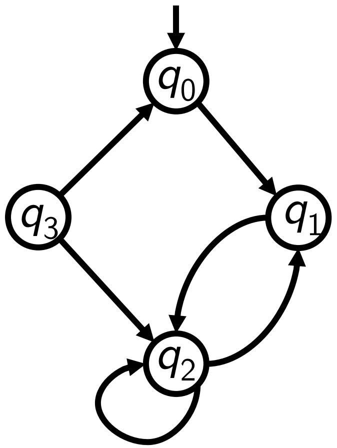
$$Q_i = Q_{i-1} \cup (\mathrm{Pre}(Q_{i-1}, \delta) \cap [\![\phi_1]\!])$$ for all $i > 1$ until a fixed point is reached

Over paths:
$A\phi \rightarrow$ **All** ϕ
$E\phi \rightarrow$ **Exists** ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: $\phi_1 EU \phi_2$

- Example for $\phi_1 EU \phi_2$: Compute $q_0\, EU\, q_1$



1. Define $[\![q_0\, EU\, q_1]\!]$: set of all states of the finite automaton for which CTL formula $q_0\, EU\, q_1$ is true.

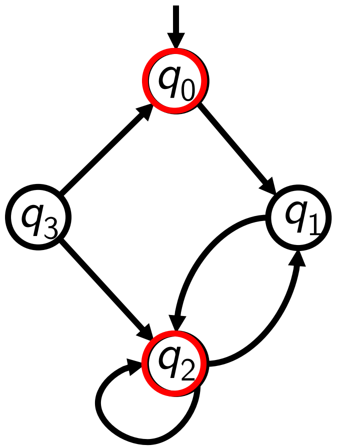
$$Q_0 = [\![q_1]\!] = \{q_1\}$$

Computing a CTL formula: $\phi_1 EU \phi_2$

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

- Example for $\phi_1 EU \phi_2$: Compute $q_0\, EU\, q_1$



1. Define $[\![q_0\, EU\, q_1]\!]$: set of all states of the finite automaton for which CTL formula $q_0\, EU\, q_1$ is true.

$$Q_0 = [\![q_1]\!] = \{q_1\}$$

$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_0, q_2\}$$

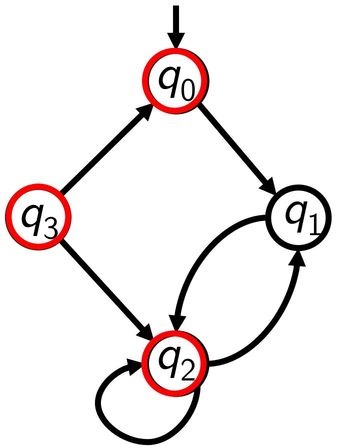
$$Q_1 = \{q_1\} \cup (\mathrm{Pre}(\{q_1\}, \delta) \cap \{q_0\}) = \{q_0, q_1\}$$

Computing a CTL formula: $\phi_1 EU \phi_2$

Over paths:
$A\phi \rightarrow$ **All** ϕ
$E\phi \rightarrow$ **Exists** ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

- Example for $\phi_1 EU \phi_2$: Compute $q_0\ EU\ q_1$



1. Define $[\![q_0\ EU\ q_1]\!]$: set of all states of the finite automaton for which CTL formula $q_0\ EU\ q_1$ is true.

$$Q_0 = [\![q_1]\!] = \{q_1\}$$

$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_0, q_2\}$$

$$Q_1 = \{q_1\} \cup (\mathrm{Pre}(\{q_1\}, \delta) \cap \{q_0\}) = \{q_0, q_1\}$$

$$Q_2 = \{q_0, q_1\} \cup (\mathrm{Pre}(\{q_0, q_1\}, \delta) \cap \{q_0\}) = \{q_0, q_1\}$$

$$[\![q_0 \mathrm{EU} q_1]\!] = Q_2 = \{q_0, q_1\}$$

$$\{q_0, q_2, q_3\}$$

2. A finite automaton with initial state q_0 satisfies $q_0\ EU\ q_1$ iff $q_0 \in [\![q_0\ EU\ q_1]\!]$

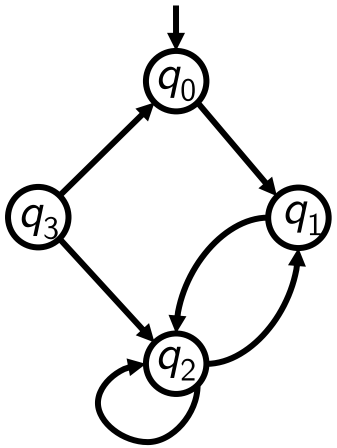
As $q_0 \in [\![q_0 \mathrm{EU} q_1]\!] = \{q_0, q_1\}$, the CTL formula q_0 EU q_1 is true.

Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Computing a CTL formula: $\phi_1 EU \phi_2$

- Example for $\phi_1 EU \phi_2$: Compute $q_0\ EU\ q_1$



1. Define $[\![q_0\ EU\ q_1]\!]$: set of all states of the finite automaton for which CTL formula $q_0\ EU\ q_1$ is true.

$$Q_0 = [\![q_1]\!] = \{q_1\}$$

$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_0, q_2\}$$

$$Q_1 = \{q_1\} \cup (\mathrm{Pre}(\{q_1\}, \delta) \cap \{q_0\}) = \{q_0, q_1\}$$

$$Q_2 = \{q_0, q_1\} \cup (\mathrm{Pre}(\{q_0, q_1\}, \delta) \cap \{q_0\}) = \{q_0, q_1\}$$

$$[\![q_0 \mathrm{EU} q_1]\!] = Q_2 = \{q_0, q_1\}$$

$$\{q_0, q_2, q_3\}$$

2. A finite automaton with initial state q_0 satisfies $q_0\ EU\ q_1$ iff $q_0 \in [\![q_0\ EU\ q_1]\!]$

As $q_0 \in [\![q_0 \mathrm{EU} q_1]\!] = \{q_0, q_1\}$, the CTL formula q_0 EU q_1 is true.

Compute other CTL expressions as:
$\mathrm{AF}\phi \equiv \neg\mathrm{EG}(\neg\phi)$ $\quad \mathrm{AG}\phi \equiv \neg\mathrm{EF}(\neg\phi)$ $\quad \mathrm{AX}\phi \equiv \neg\mathrm{EX}(\neg\phi)$

So... what is model checking exactly?

Model checking is an algorithm which takes two inputs

- a DES model $\boldsymbol{M}$
- a formula $\boldsymbol{\phi}$

Finite automaton
Petri net
Kripke machine
...

CTL, LTL, ...

It explores the state space of $\boldsymbol{M}$ such as to either

- prove that $\boldsymbol{M} \models \boldsymbol{\phi}$, or
- return a trace where the formula does not hold in $\boldsymbol{M}$.

a counter-example

Extremely useful!

- Debugging the model
- Searching a specific execution sequence

Efficient state representation

- Set of states as Boolean function
- Binary Decision Diagram representation

Computing reachability

- Leverage efficient state representation
- Explore successor sets of states

Today

Proving properties

- Temporal logic (CTL)
- Encoding as reachability problem

Conclusion and perspectives

Next week(s)

Petri Nets

- asynchronous DES model
- tailored model concurrent distributed systems
- capture an infinite state space with a finite model

a computer
a network

How they work?
How to use them for modeling systems?
How to verify them?

Your turn to practice!
after the break

1. Familiarise yourself with CTL logic
   and how to compute sets of states
   satisfying a given formula

2. Convert a concrete problem into
   a state reachability question
   (adapted from state-of-the-art research!)

Any feedback?
Please fill out this short (anonymous) form!

The form will be available throughout the lecture—feel free to provide feedback at any point.

https://forms.gle/7VUaidEVreS9uswa9

Thanks for your attention and see you next week! ☺