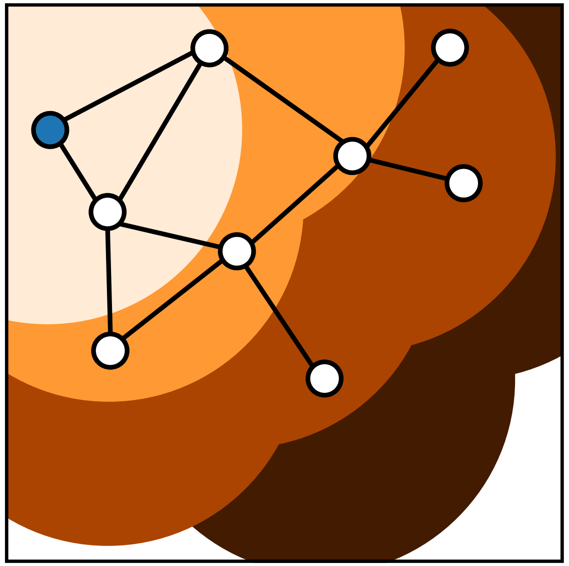
Discrete Event Systems
Verification of Finite Automata (Part 2)



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

ETH Zurich (D-ITET)

December 1, 2022

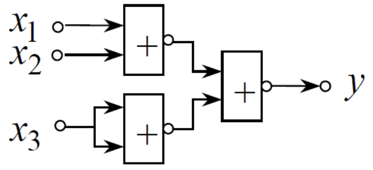
Most materials from Lothar Thiele and Romain Jacob

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 → comparison

system under test → data structure → comparison

Proving properties

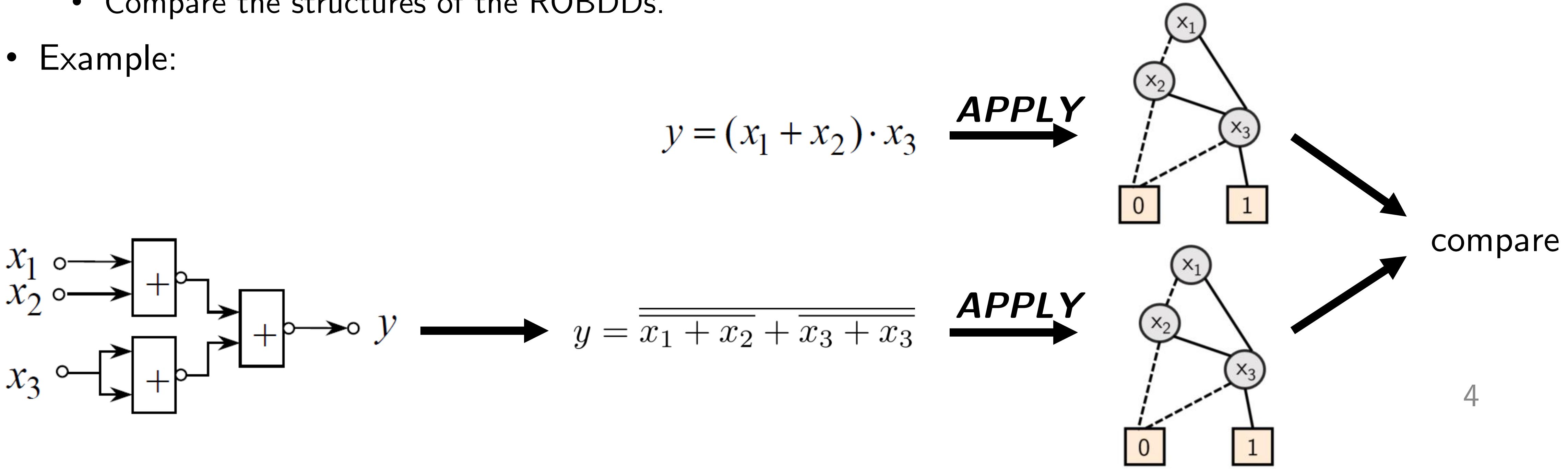
"The device can always be switched off."

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:



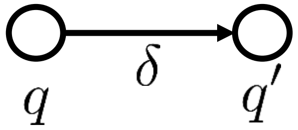
Sets and Relations

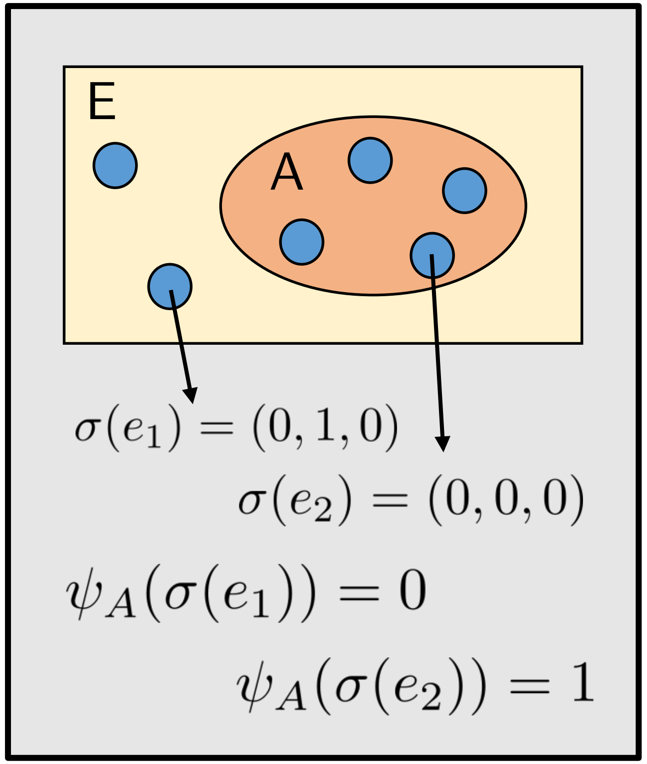
characteristic function of subset A

- 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

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

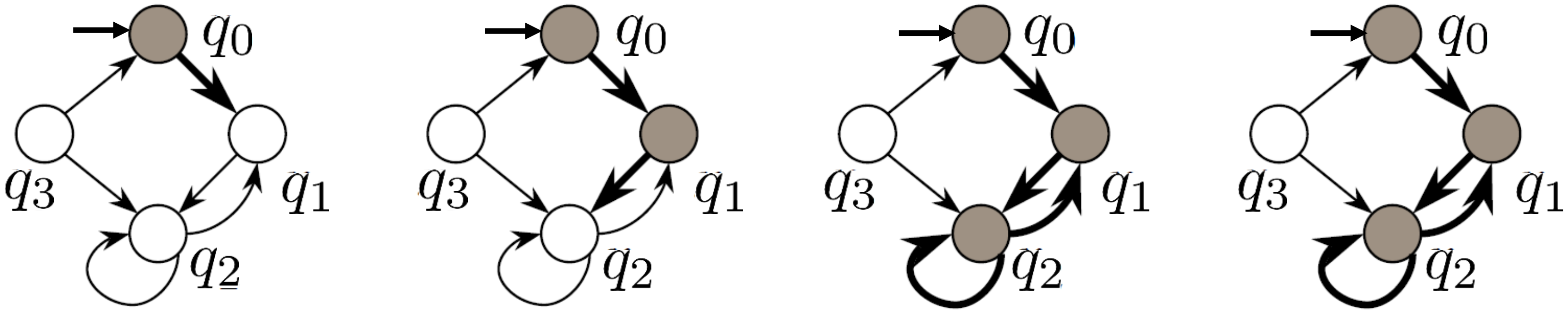




Reachability of States

- 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:

$Q_0 = \{q_0\}$ $Q_1 = \{q_0, q_1\}$ $Q_2 = \{q_0, q_1, q_2\}$ $Q_3 = \{q_0, q_1, q_2\}$



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?

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

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

- 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?

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

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?

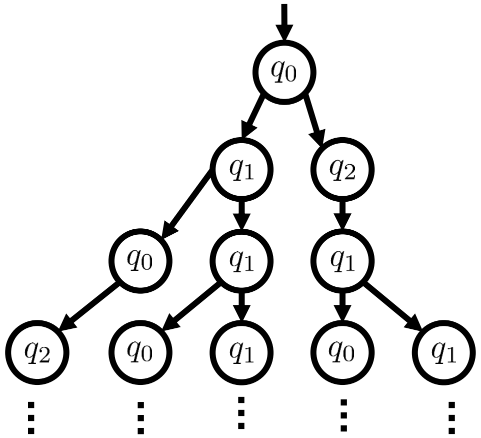
- Specification of the query in a formula of temporal logic.
- We use a simple form called Computation Tree Logic (CTL).

- 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.

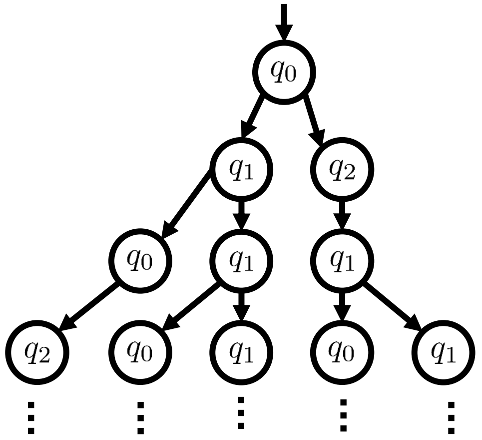
Formula	Examples
Atomic proposition	The printer is busy. The light is on.
Boolean logic	$\phi_1 + \phi_2$; $\neg\phi_1$
CTL logic	EX ϕ_1

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

Formulation of CTL properties



Formulation of CTL properties

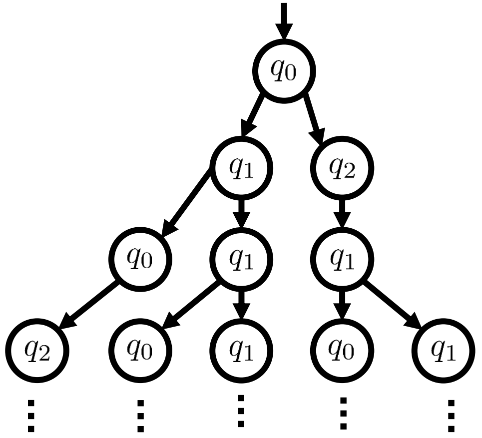


Based on atomic propositions (ϕ) and quantifiers

Aϕ	→ «**All** ϕ»,	ϕ holds on all paths
Eϕ	→ «**Exists** ϕ»,	ϕ holds on at least one path

Quantifiers over paths

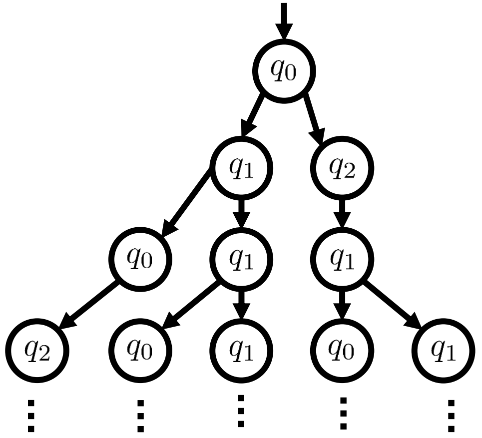
Formulation of CTL properties



Based on atomic propositions (ϕ) and quantifiers

Aϕ	→ «**A**ll ϕ»,	ϕ holds on all paths	Quantifiers over paths
Eϕ	→ «**E**xists ϕ»,	ϕ holds on at least one path	
Xϕ	→ «Ne**X**t ϕ»,	ϕ holds on the next state	Path-specific quantifiers
Fϕ	→ «**F**inally ϕ»,	ϕ holds at some state along the path	
Gϕ	→ «**G**lobally ϕ»,	ϕ holds on all states along the path	
ϕ_1Uϕ_2	→ «ϕ_1**U**ntil ϕ_2»,	ϕ_1 holds until ϕ_2 holds implies that ϕ_2 has to hold eventually	

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! $\{A,E\}$ $\{X,F,G,U\}\phi$

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

Formulation of CTL properties

Based on atomic propositions (ϕ) and quantifiers

Quantifiers over paths

$A\phi$	→ «**A**ll ϕ»,	ϕ holds on all paths
$E\phi$	→ «**E**xists ϕ»,	ϕ holds on at least one path

Path-specific quantifiers

$X\phi$	→ «Ne**X**t ϕ»,	ϕ holds on the next state
$F\phi$	→ «**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! $\{A,E\}\ \{X,F,G,U\}\phi$

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

$X\phi \rightarrow$ Ne**X**t ϕ

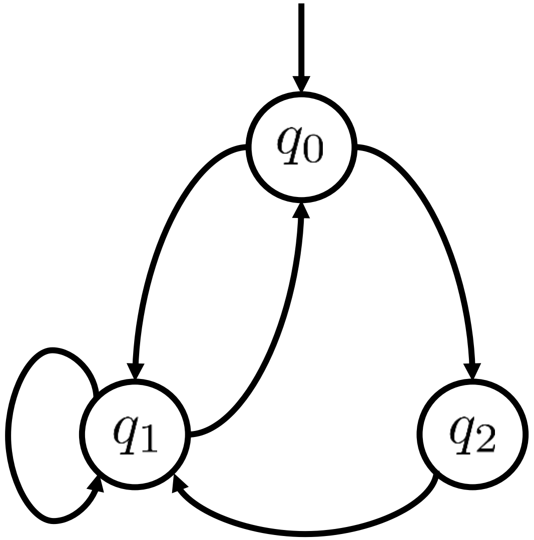
$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

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

CTL works on computation trees

Automaton



CTL works on computation trees

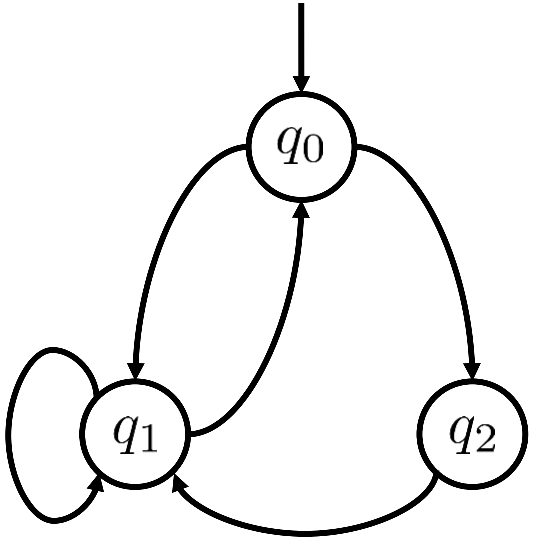
Over paths:

- $A\phi \rightarrow$ **All** ϕ
- $E\phi \rightarrow$ **Exists** ϕ

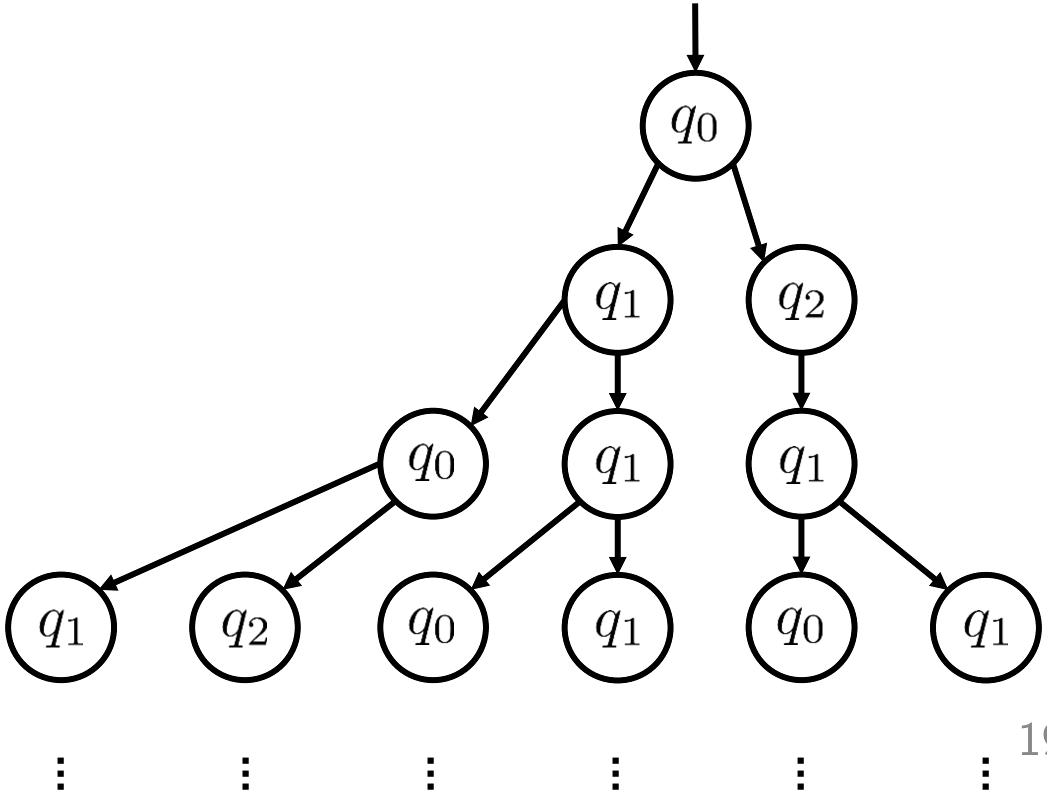
Path-specific:

- $X\phi \rightarrow$ 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

Automaton



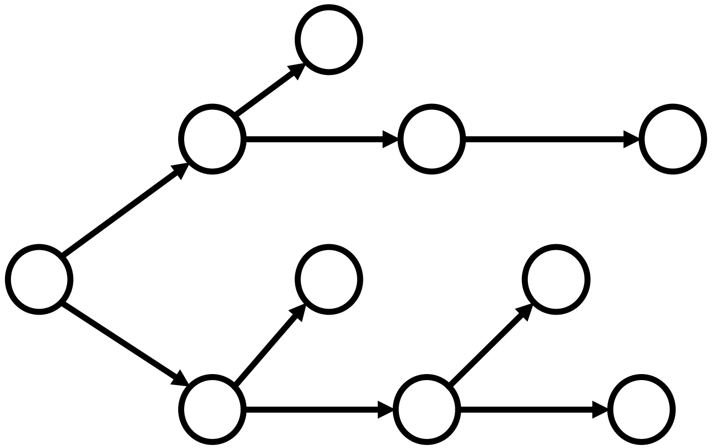
Computation tree



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

CTL works on computation trees

Automaton of interest



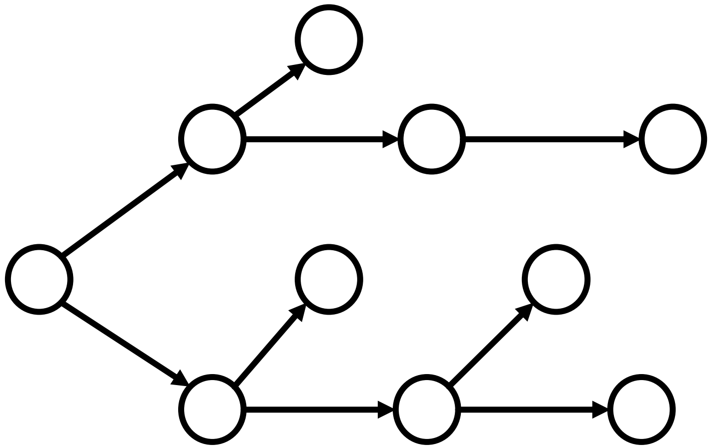
Requires fully-defined transition functions

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

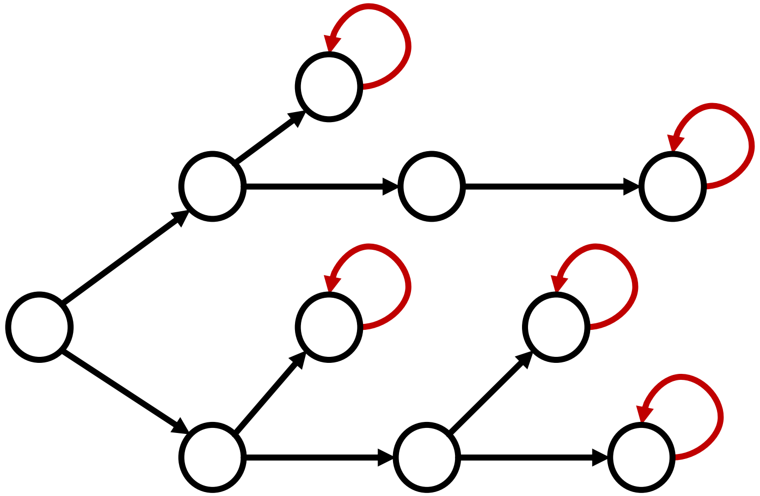
CTL works on computation trees

Automaton of interest



Requires fully-defined transition functions

Automaton to work with



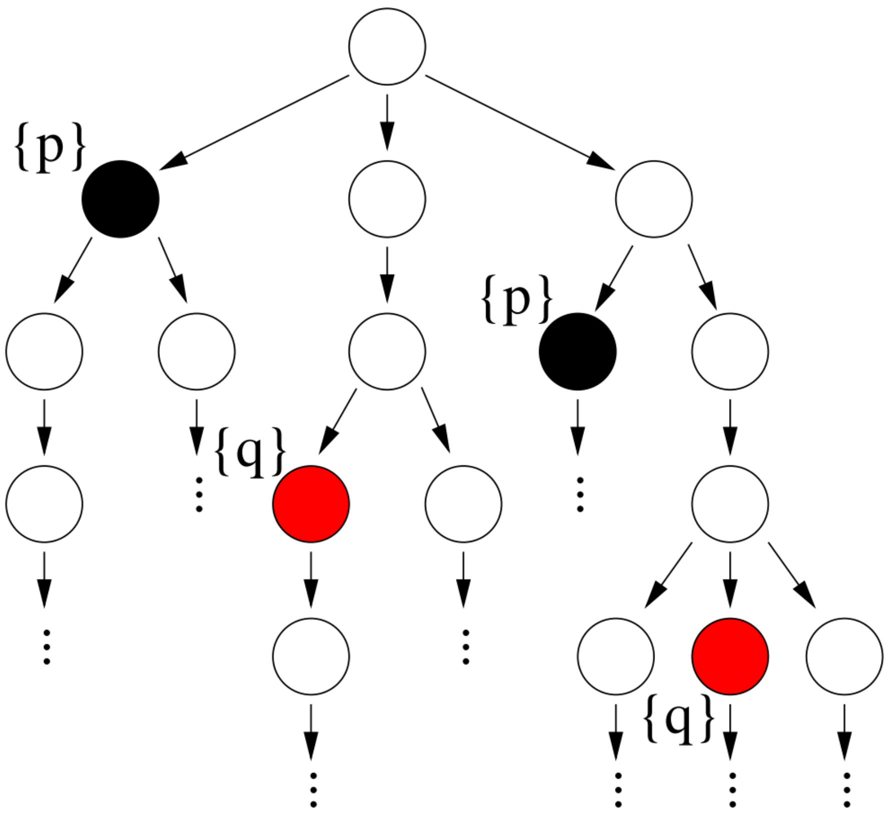
Each state has at least one successor (can be itself)

Over paths:
$A\phi$ → **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

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.

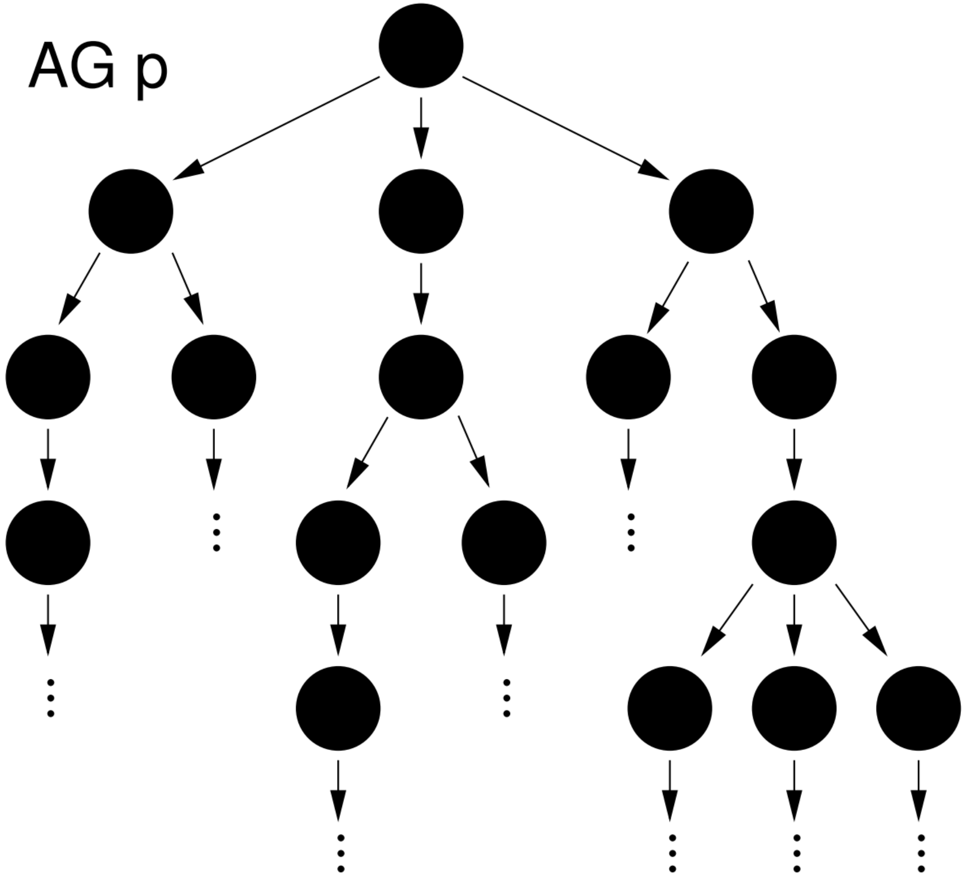


M satisfies ϕ $\Leftrightarrow$ $q_0 \vDash \phi$ where q_0 is the initial state of M

Visualizing CTL formula

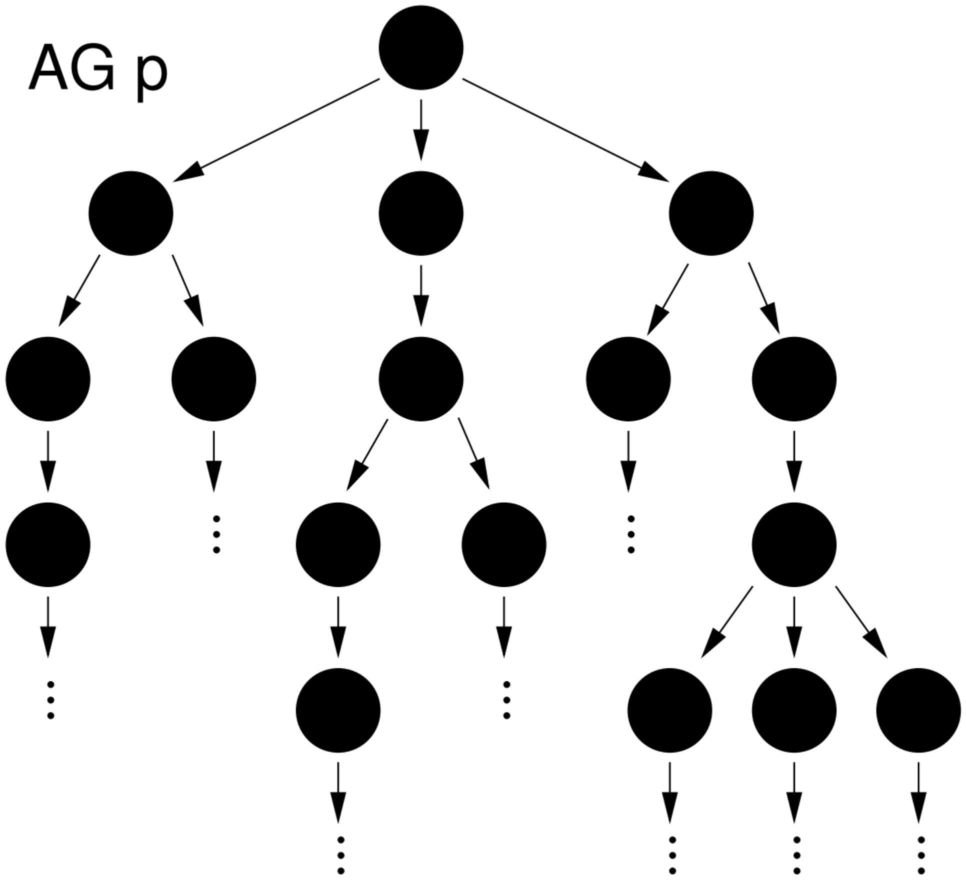
Over paths:	Path-specific:
$A\phi \to$ **A**ll ϕ	$X\phi \to$ Ne**X**t ϕ
$E\phi \to$ **E**xists ϕ	$F\phi \to$ **F**inally ϕ
	$G\phi \to$ **G**lobally ϕ
	$\phi_1 U \phi_2 \to \phi_1$ **U**ntil ϕ_2

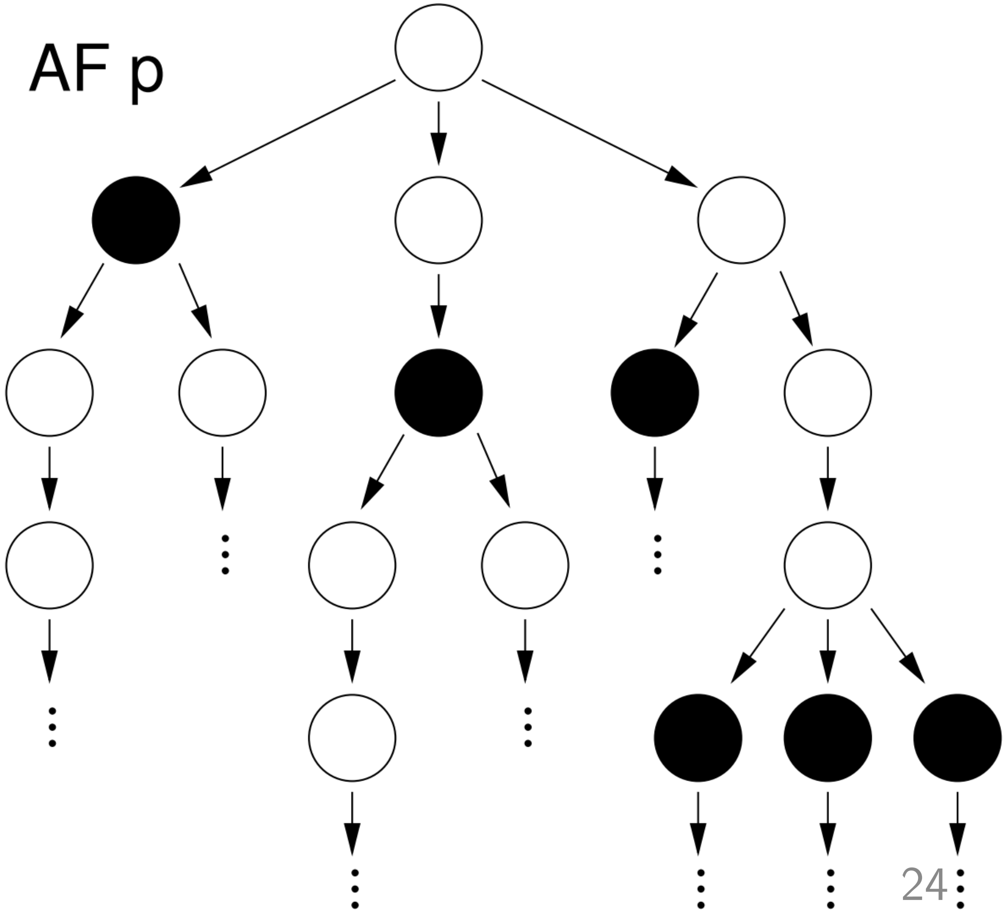
AG p



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

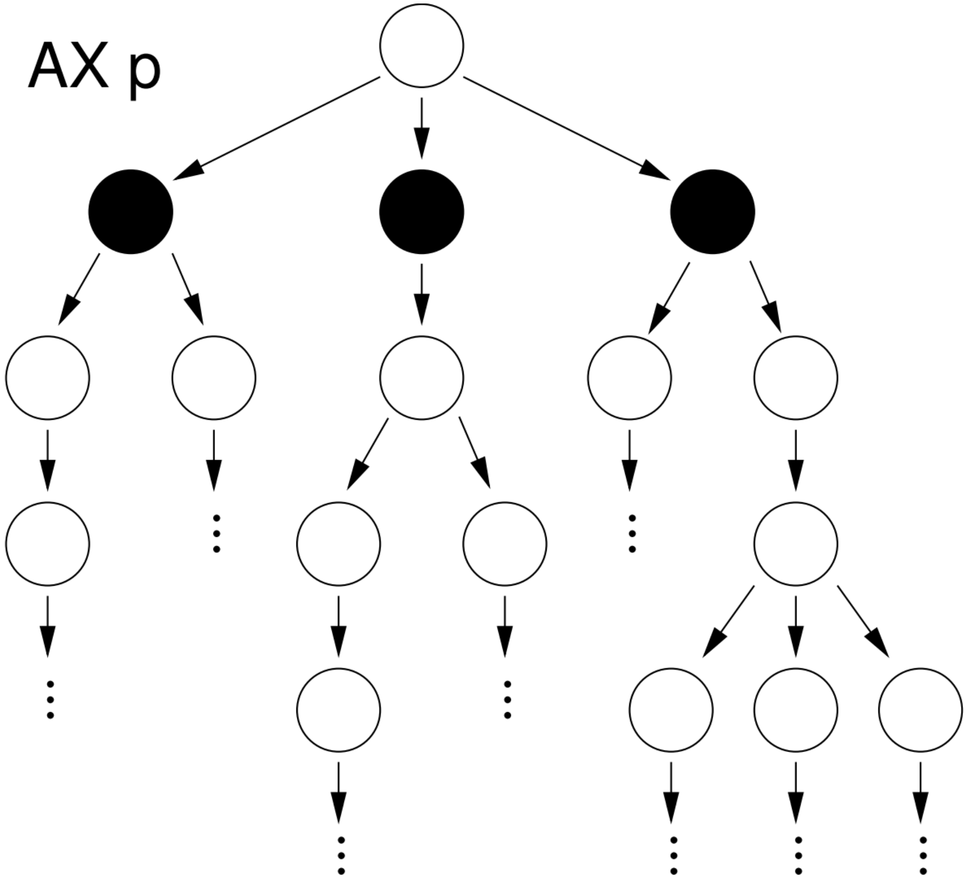
$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

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

Visualizing CTL formula

AX p

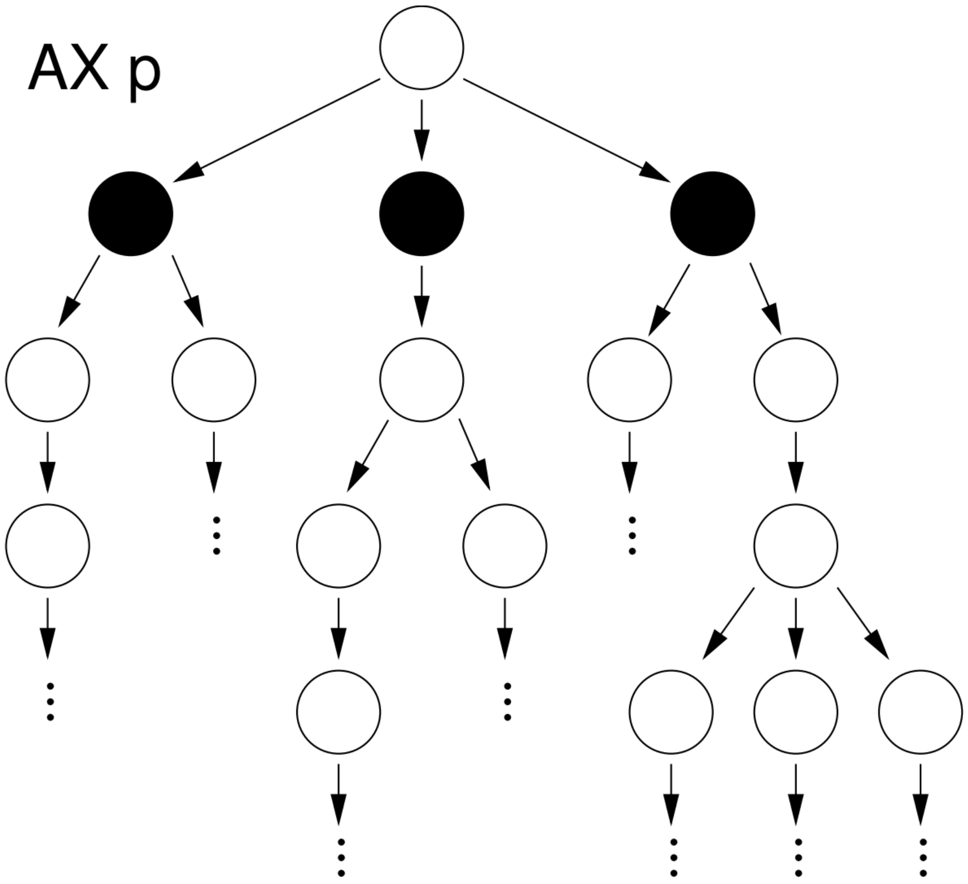


Visualizing CTL formula

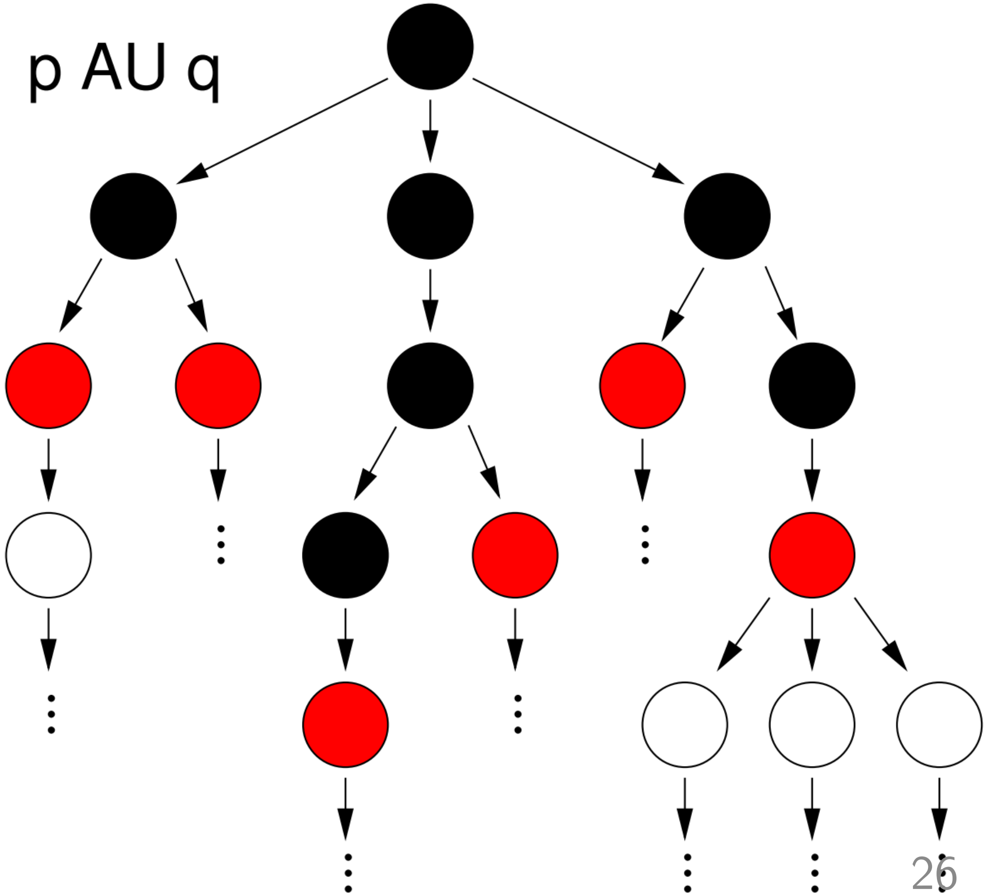
Over paths:
Aϕ → **All** ϕ
Eϕ → **Exists** ϕ

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

AX p



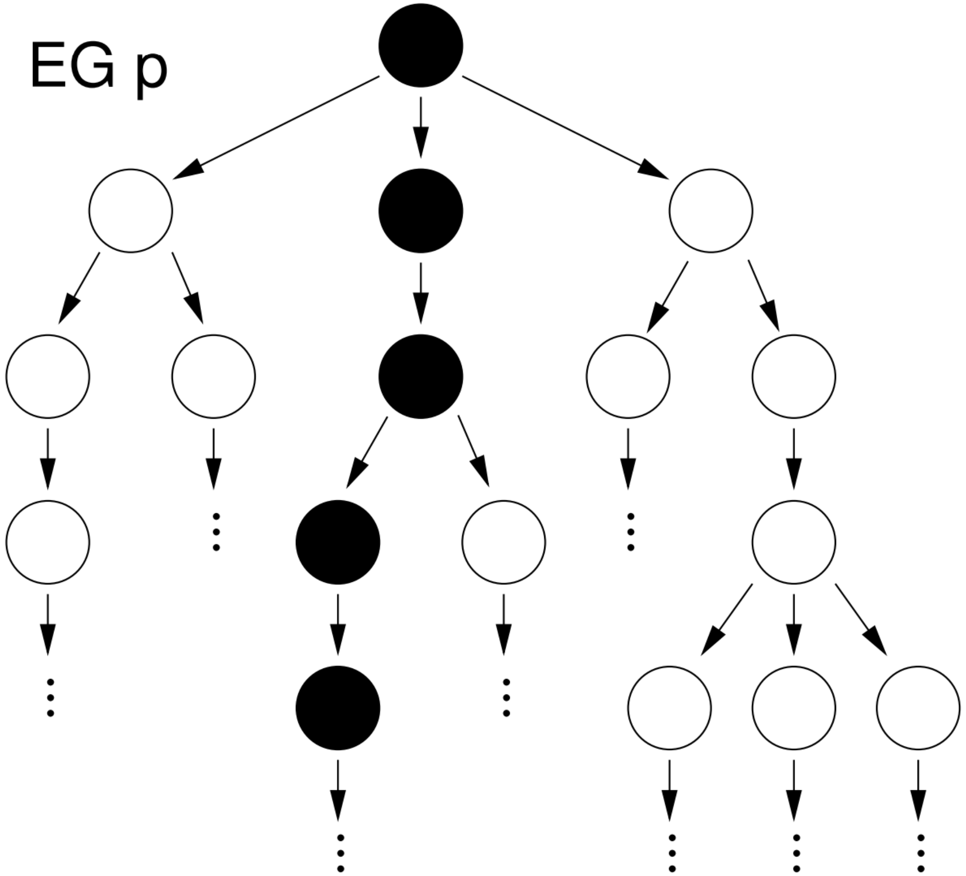
p AU q



Visualizing CTL formula

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

EG p

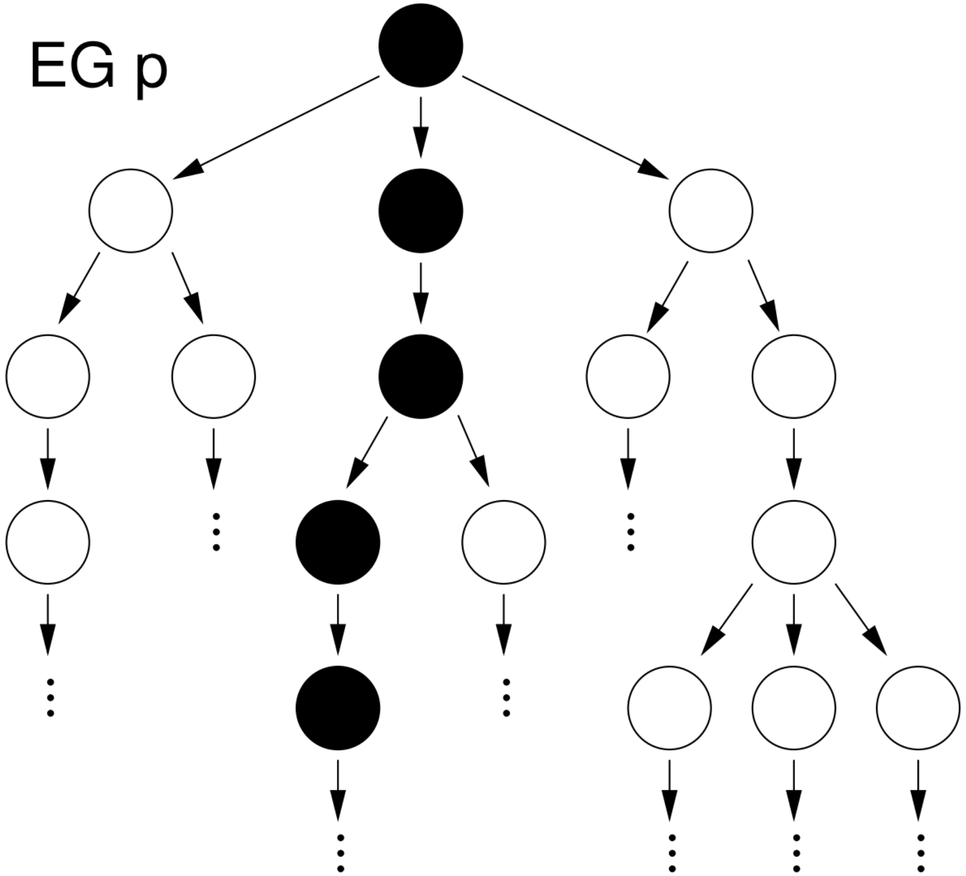


Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

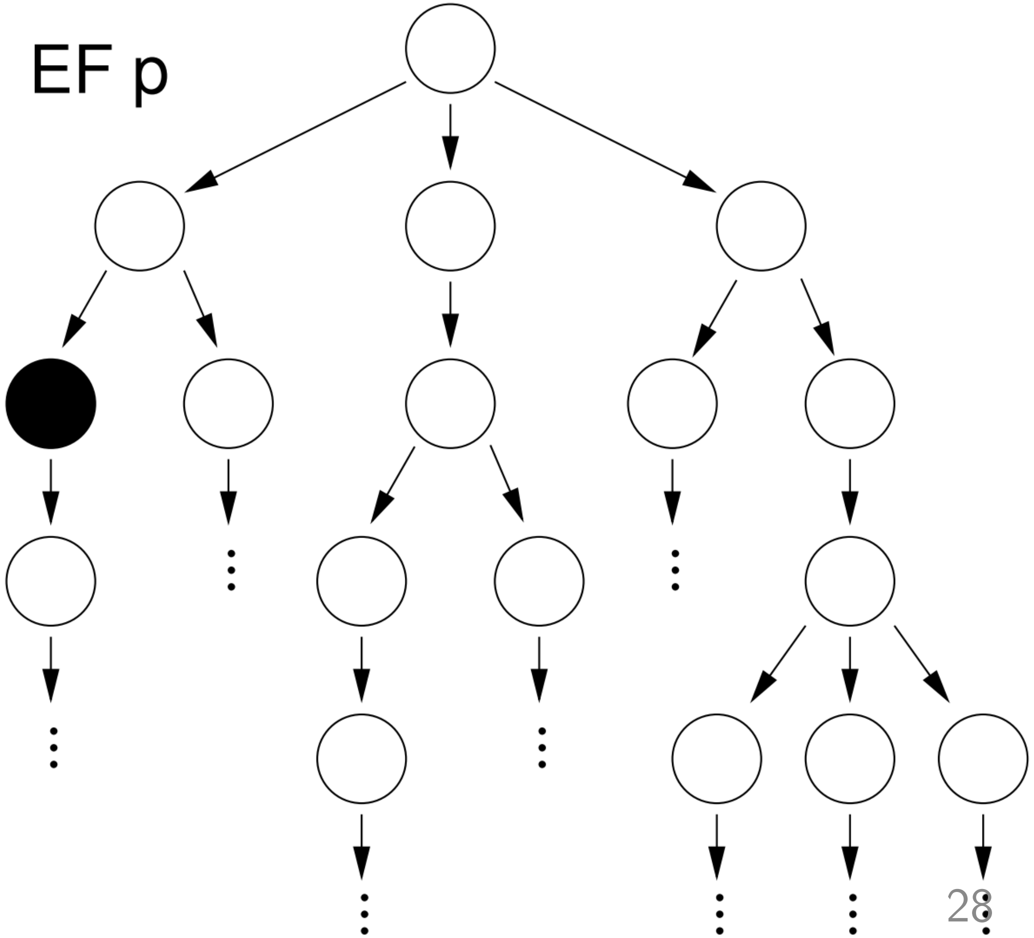
Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Visualizing CTL formula

EG p



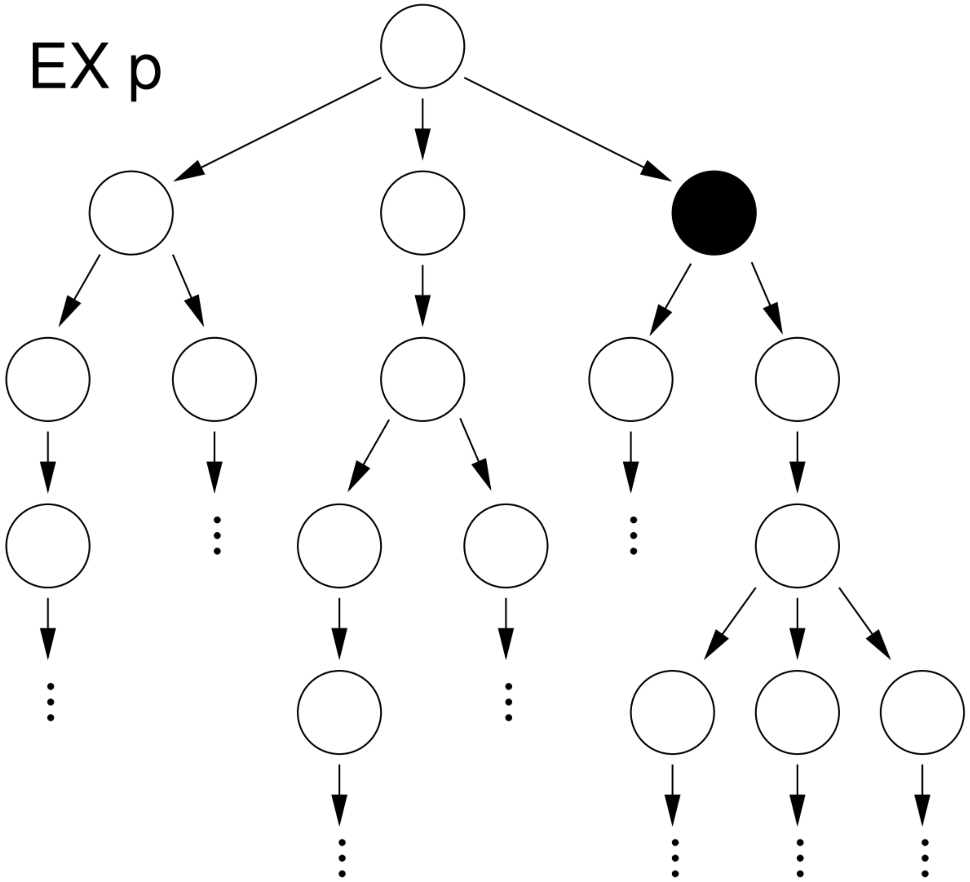
EF p



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

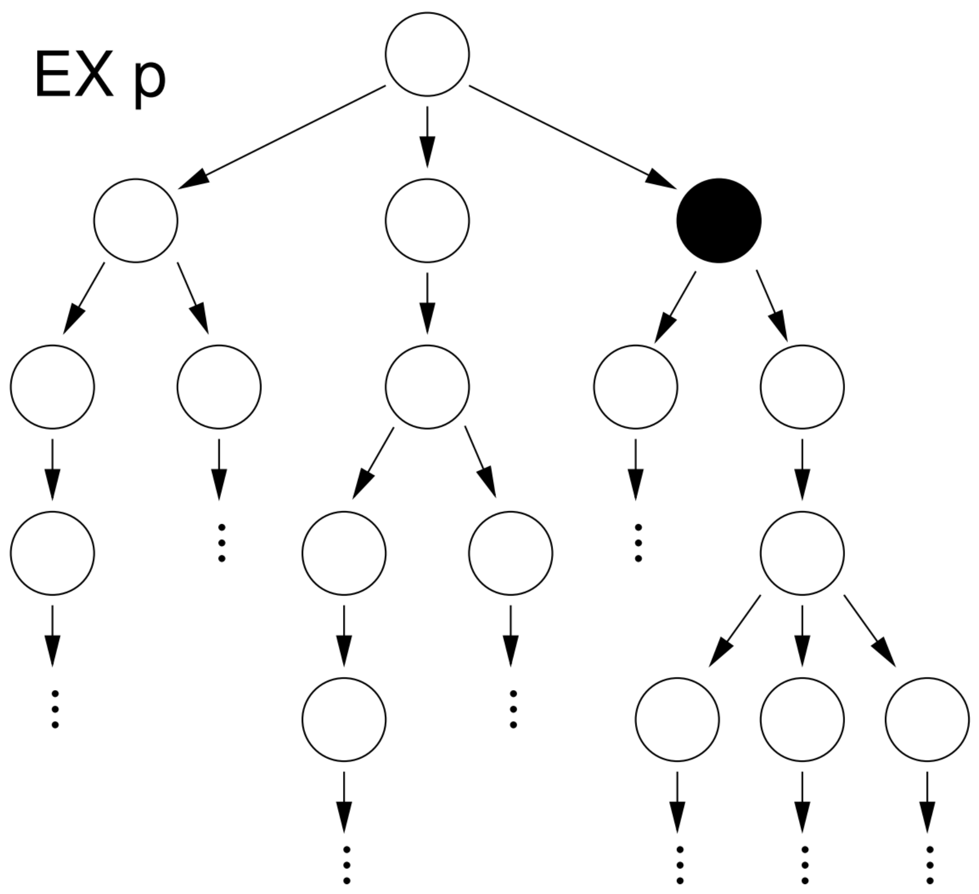
Visualizing CTL formula

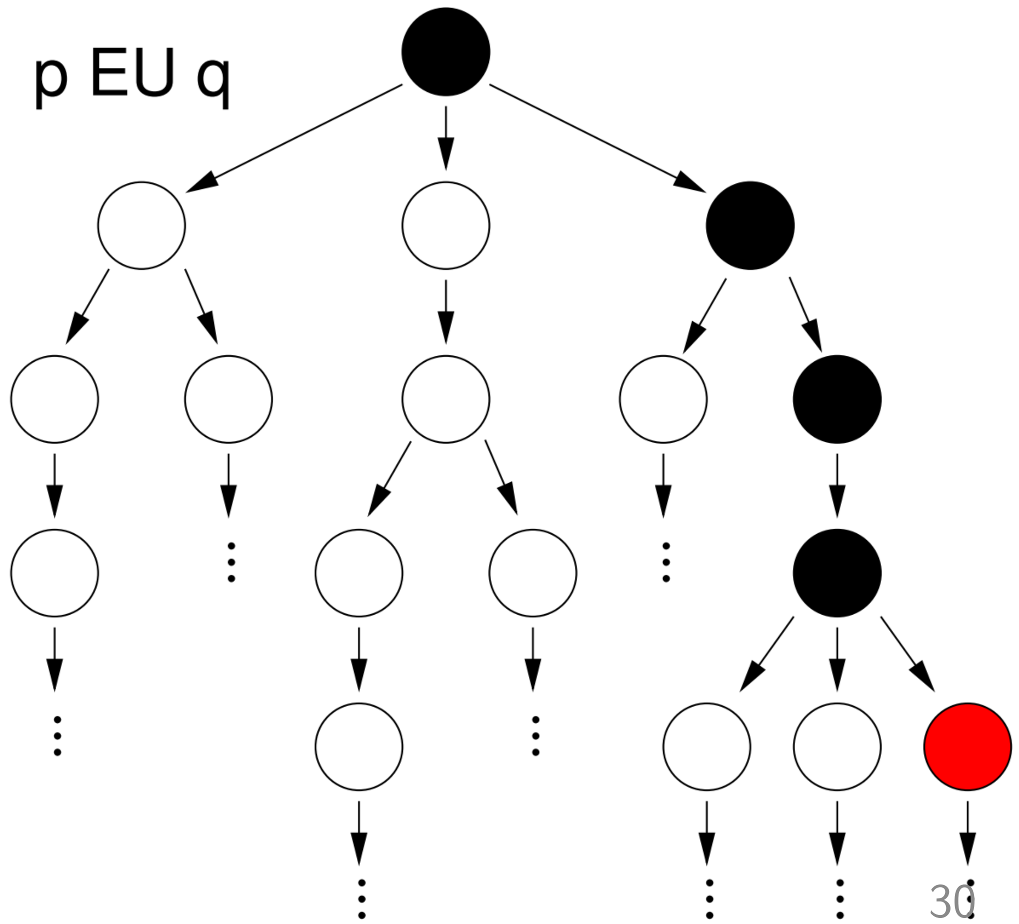
EX p



Visualizing CTL formula

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





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

Formulation of CTL properties

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.

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

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

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

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

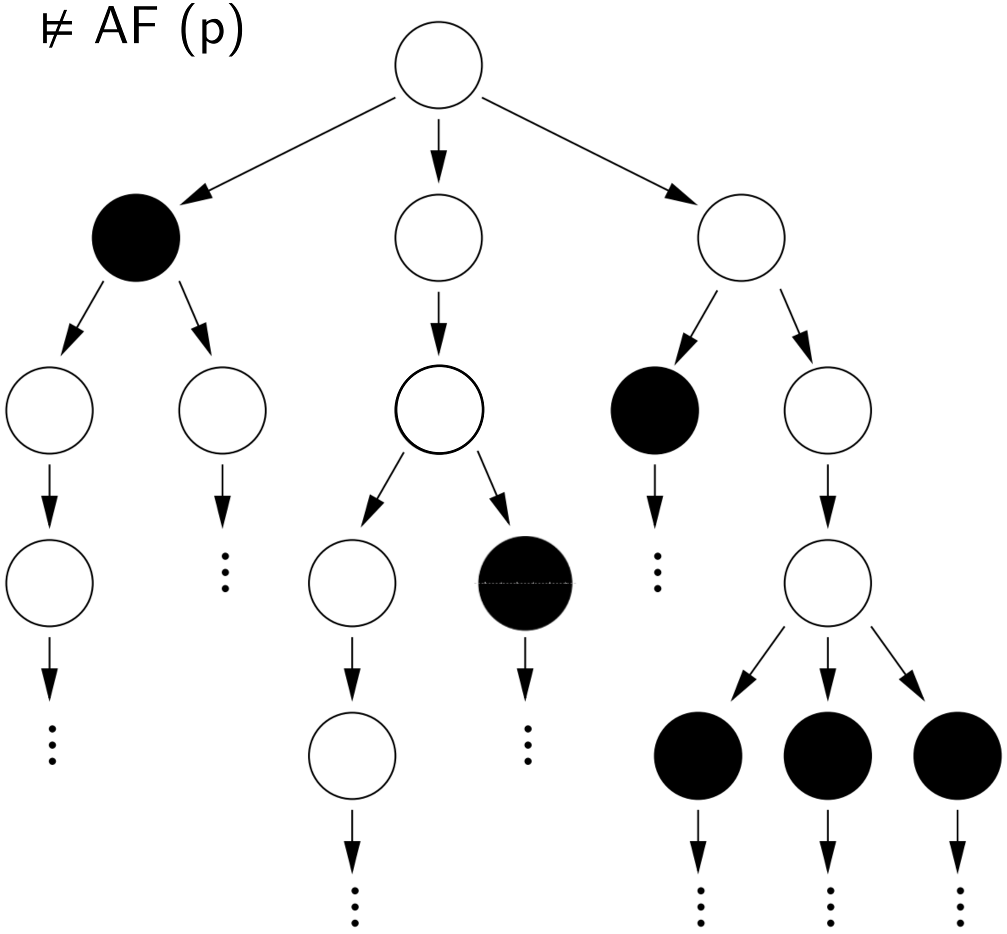
No need to know that one ▸

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

Intuition for "AF p = ¬ EG (¬ p)"

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

$\not\models$ AF (p)

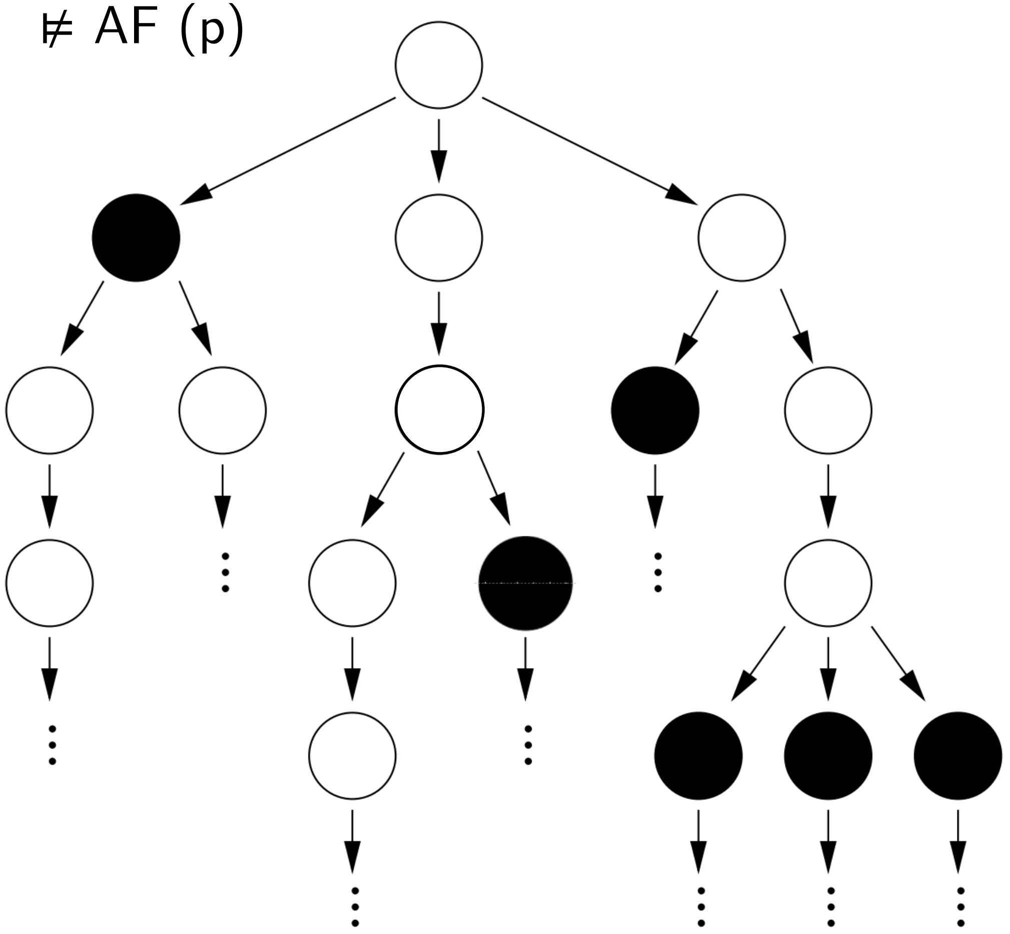


Over paths:
$A\phi \rightarrow$ **A**ll ϕ
$E\phi \rightarrow$ **E**xists ϕ

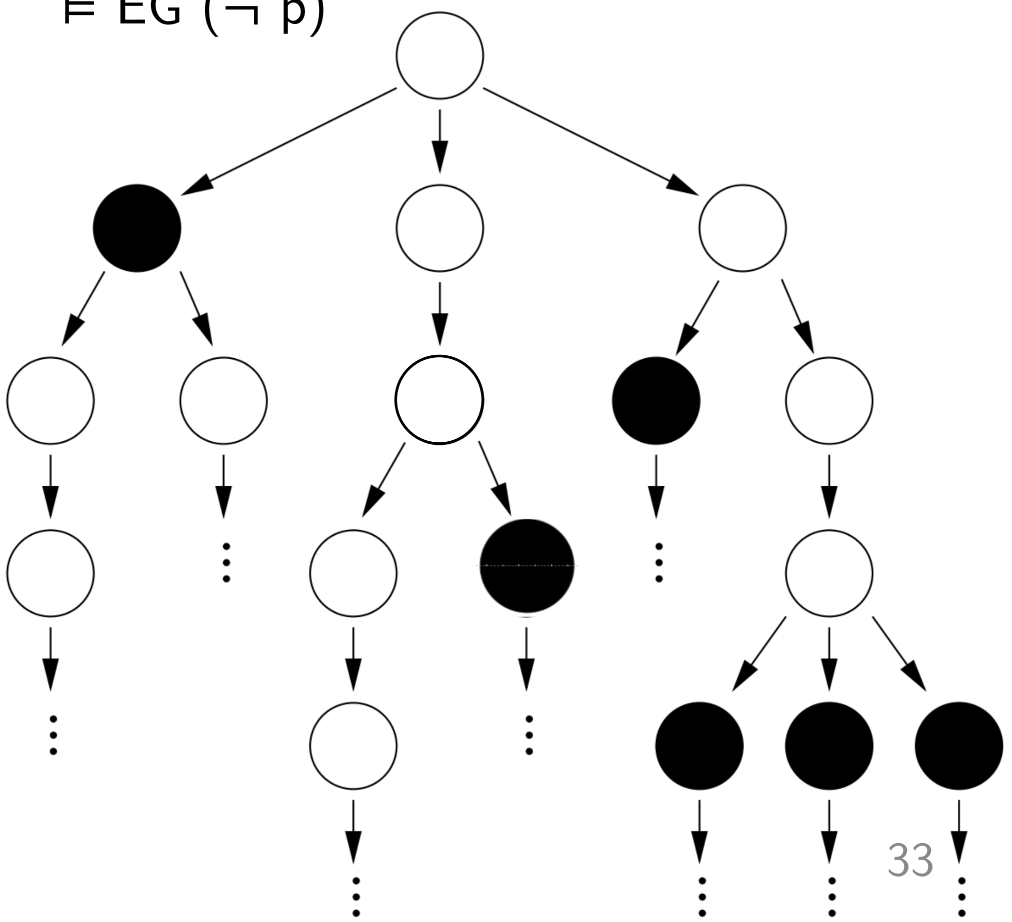
Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Intuition for "AF p = ¬ EG (¬ p)"

$\not\models$ AF (p)



$\models$ EG (¬ p)



Over paths:

$A\phi$ → **All** ϕ

$E\phi$ → **Exists** 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

Interpreting CTL formula

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

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 CTL formula

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

- AG p
- EF p
- AF EG p
- EG AF p
- p AU q

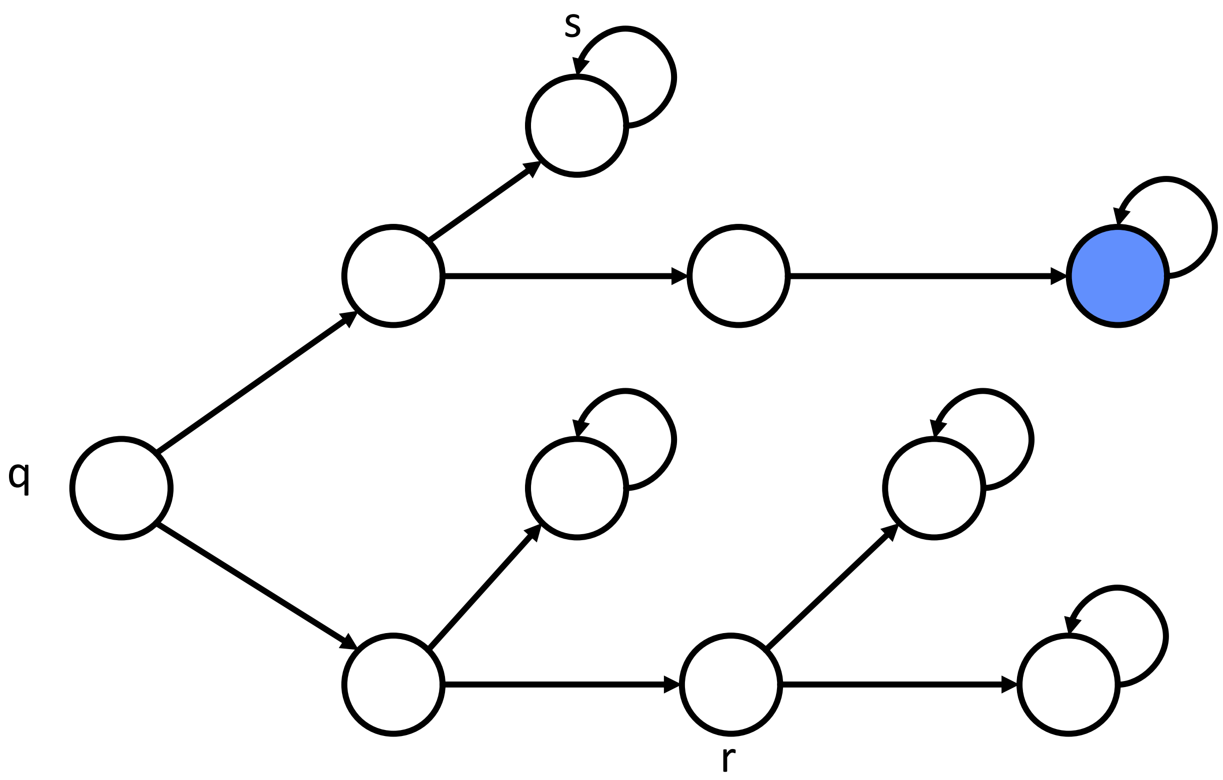
EF ϕ : "There exists a path along which at some state ϕ holds."

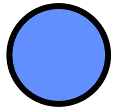
Over paths:

- $A\phi$ → **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



 $\vDash \phi$

$q \vDash EF\ \phi$

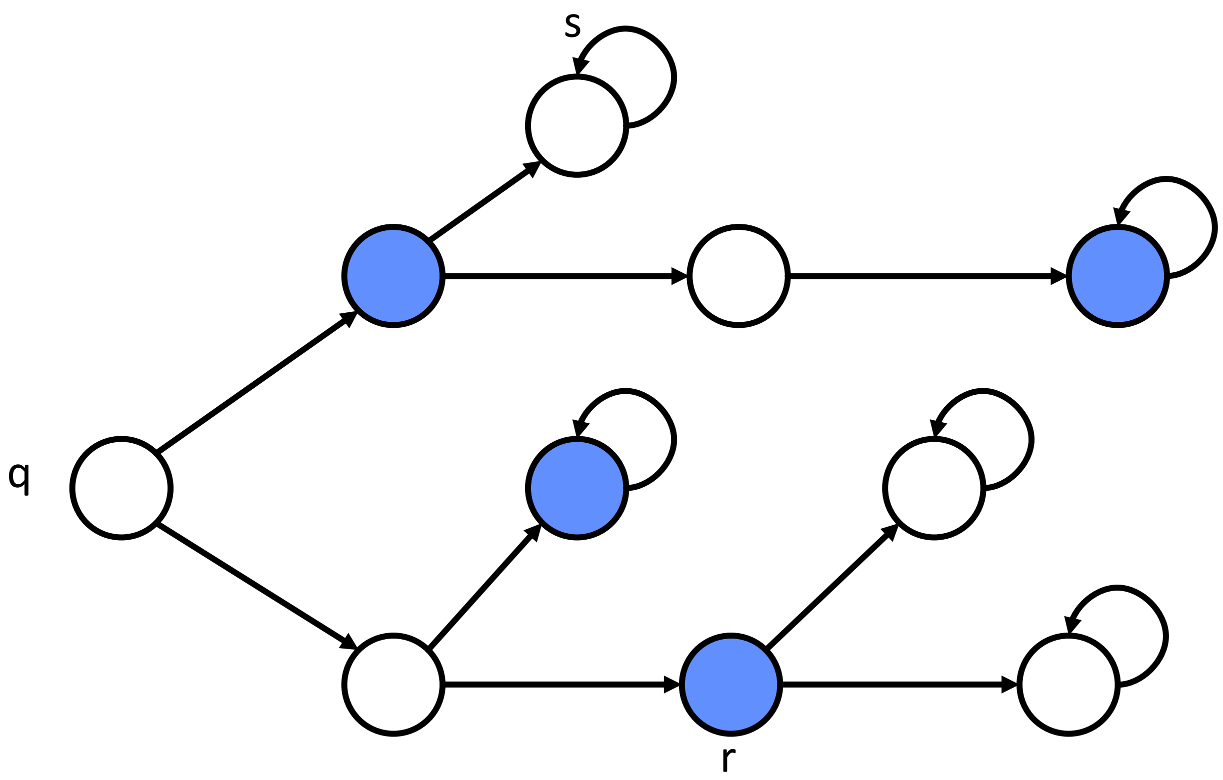
$r \vDash ?$

$s \vDash ?$

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

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



$\models \phi$

$q \models AF\ \phi$

$r \models ?$

$s \models ?$

Over paths:

$A\phi \rightarrow$ **A**ll ϕ

$E\phi \rightarrow$ **E**xists ϕ

Path-specific:

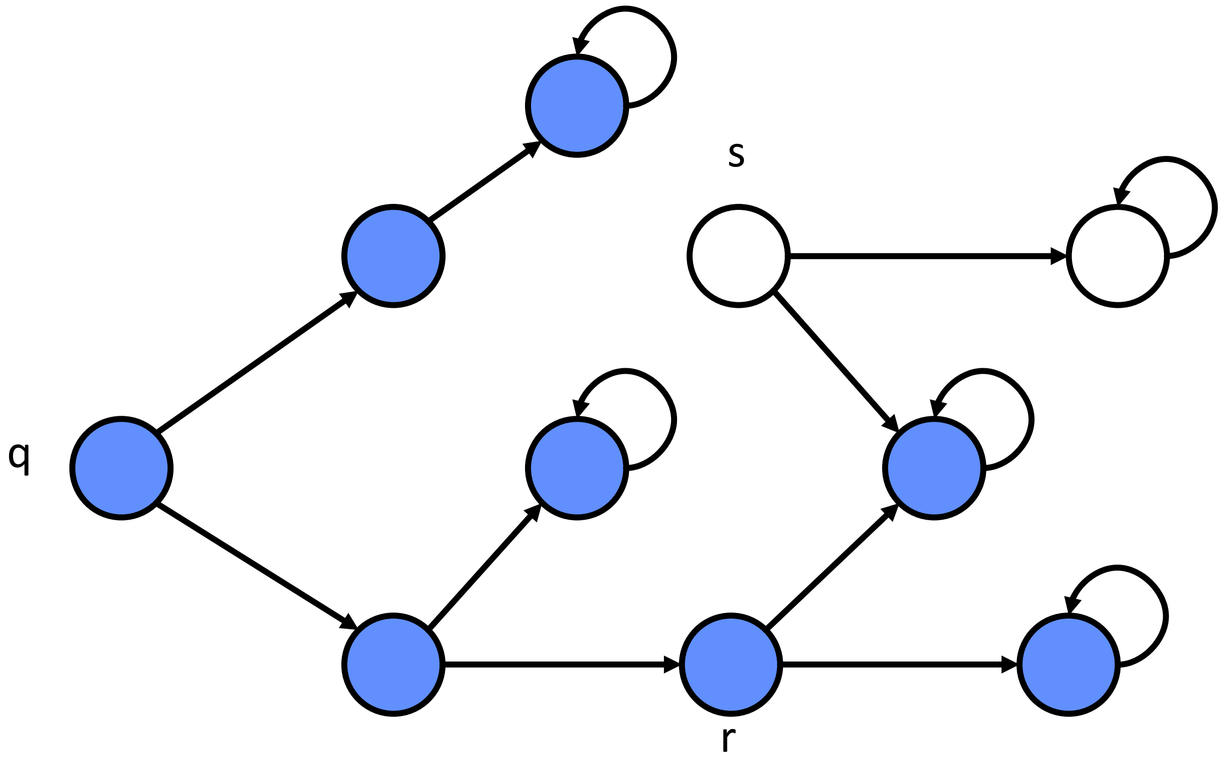
$X\phi \rightarrow$ Ne**X**t ϕ

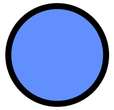
$F\phi \rightarrow$ **F**inally ϕ

$G\phi \rightarrow$ **G**lobally ϕ

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

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





$\vDash \phi$

$q \vDash AG\ \phi$

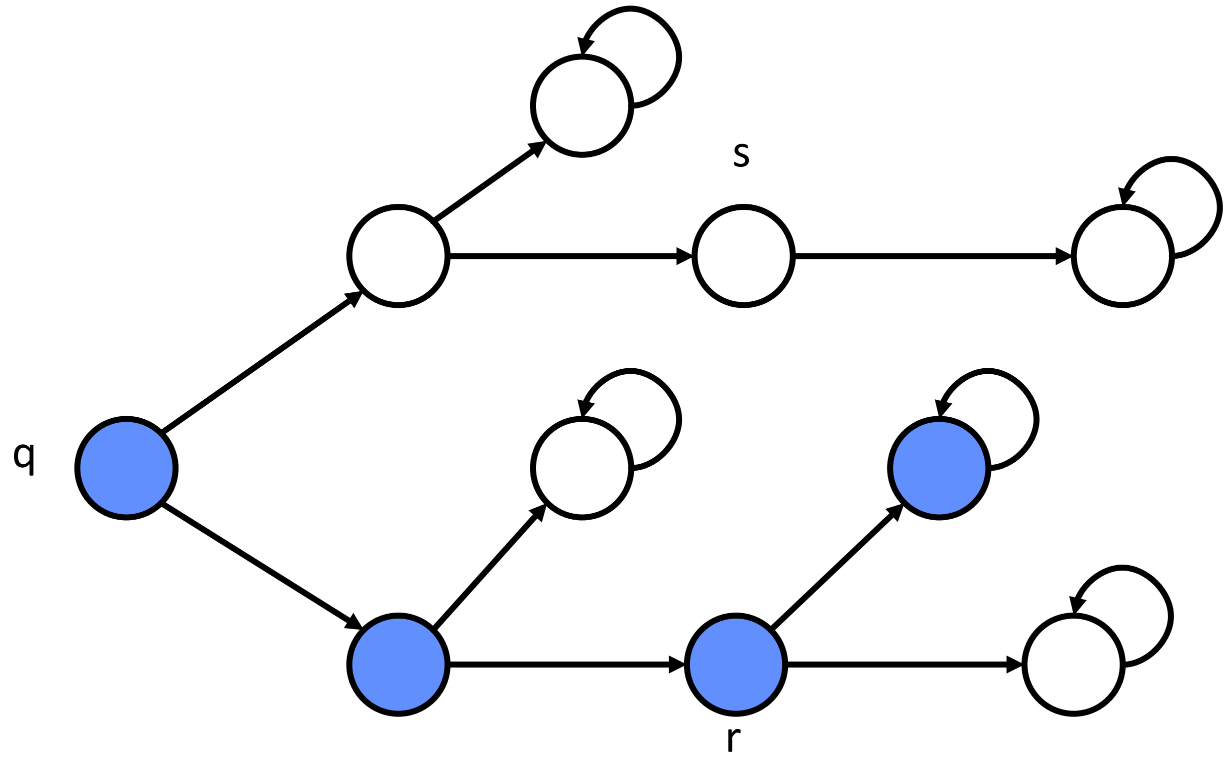
$r \vDash ?$

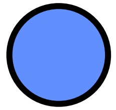
$s \vDash ?$

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

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





$q \models EG\ \phi$

$r \models ?$

$s \models ?$

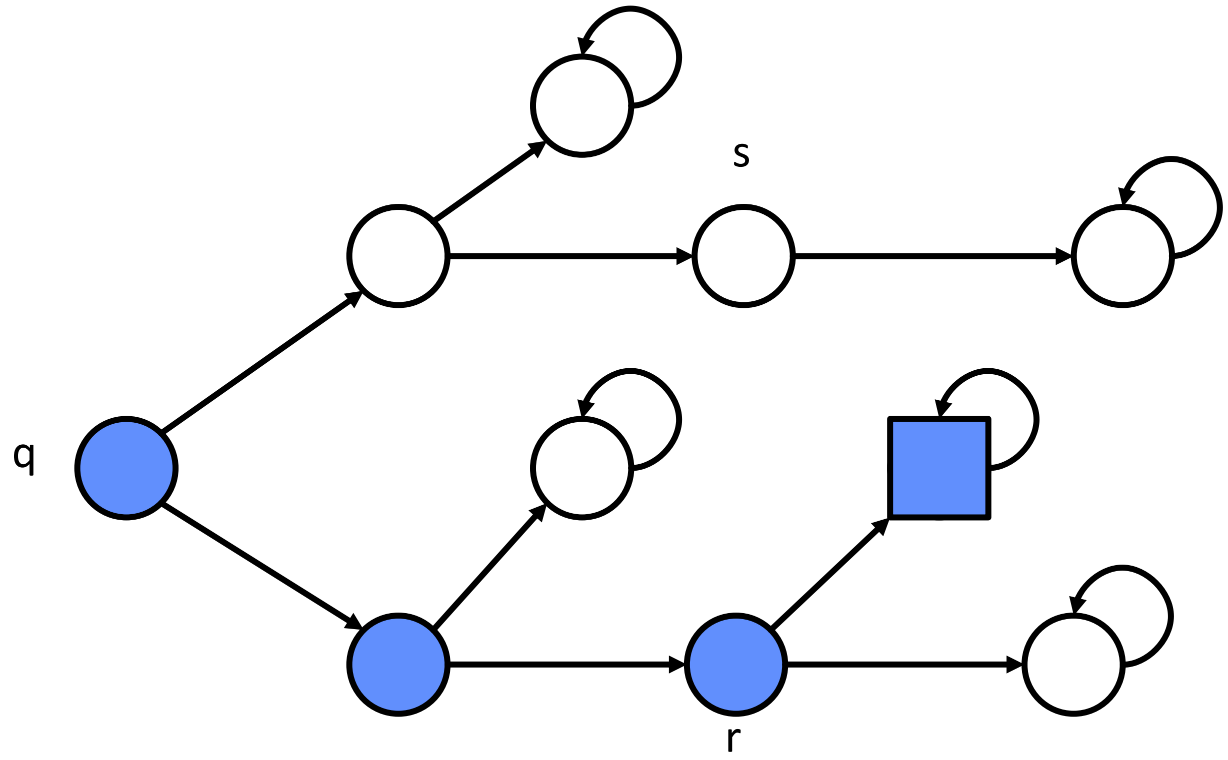
Over paths:

- $A\phi \rightarrow$ **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

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



$\blacksquare \models \Psi$

$\bullet \models \phi$

$q \models \phi EU\Psi$

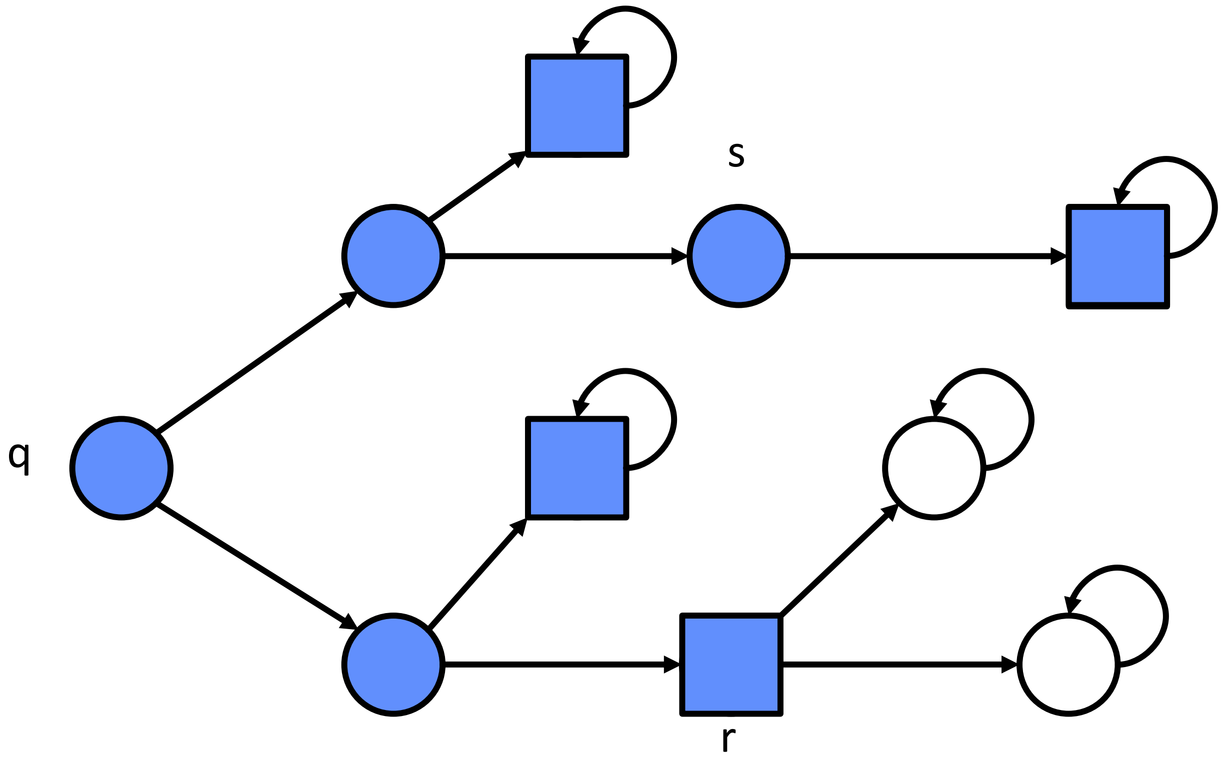
$r \models ?$

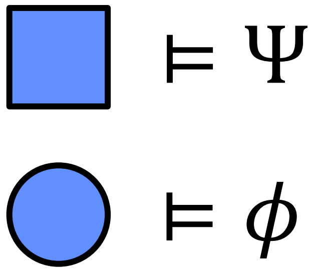
$s \models ?$

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

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



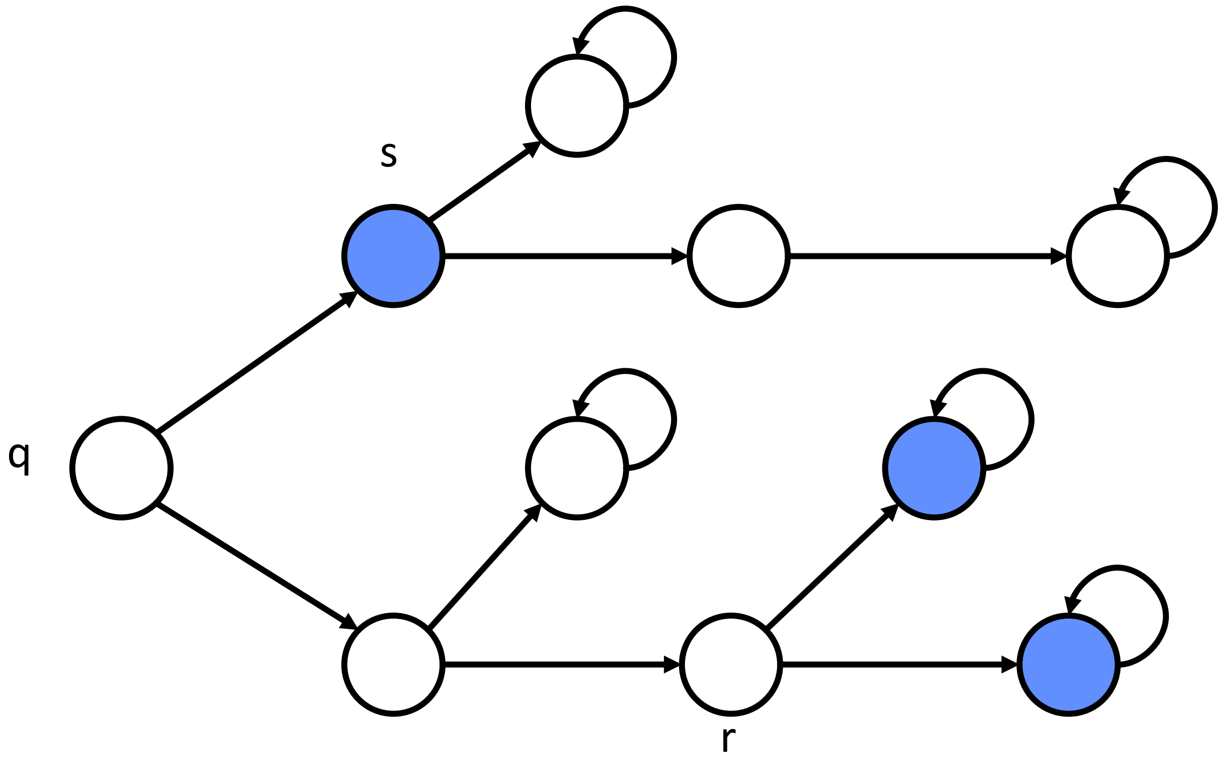


$q \vDash \phi AU\Psi$
$r \vDash ?$
$s \vDash ?$

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

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



● $\models \phi$

$q \models EX\phi$

$r \models$?

$s \models$?

Over paths:

$A\phi$ → **A**ll ϕ

$E\phi$ → **E**xists ϕ

Path-specific:

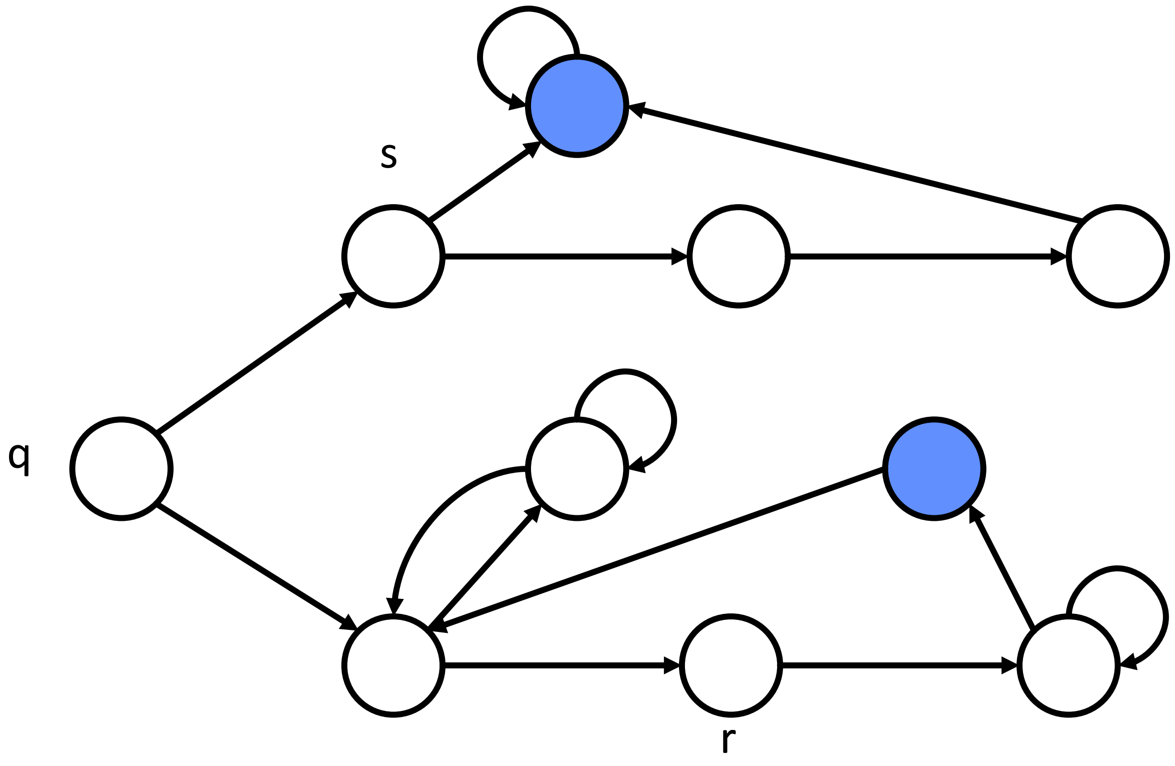
$X\phi$ → Ne**X**t ϕ

$F\phi$ → **F**inally ϕ

$G\phi$ → **G**lobally ϕ

$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

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



$\vDash \phi$

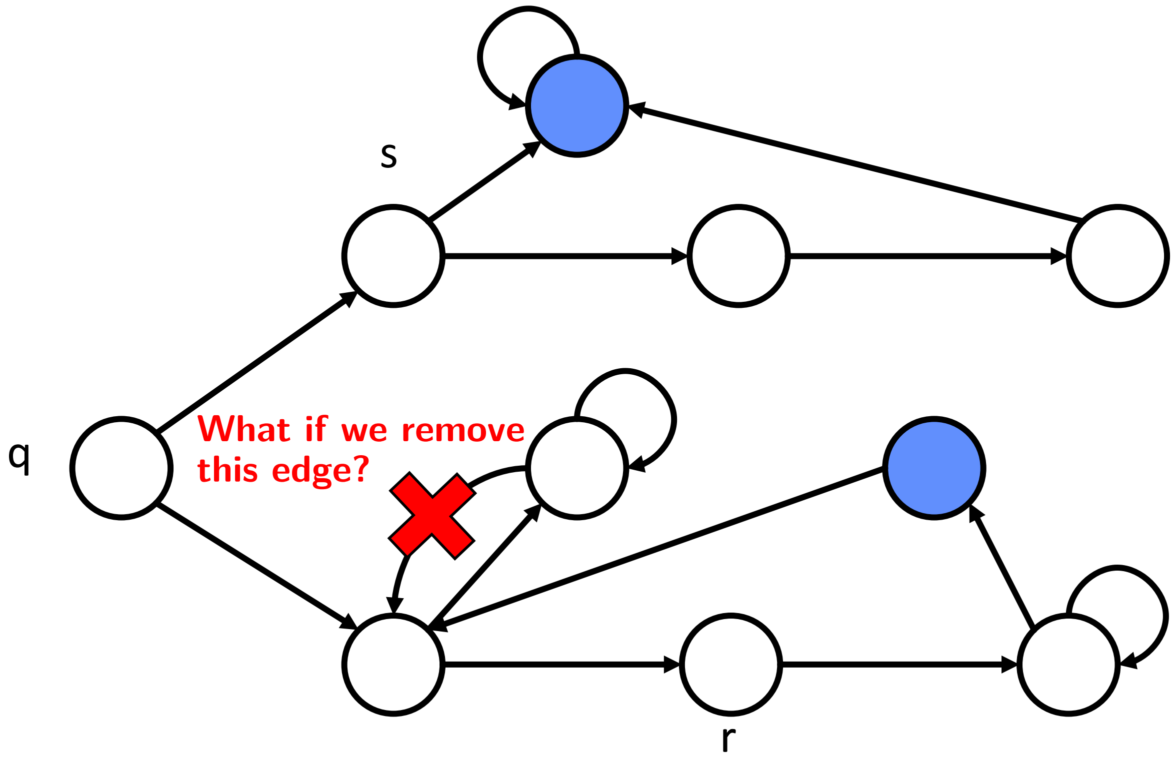
q $\vDash$ AG EF ϕ

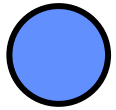
r $\vDash$?

s $\vDash$?

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





$q \vDash ?$

$r \vDash ?$

$s \vDash ?$

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

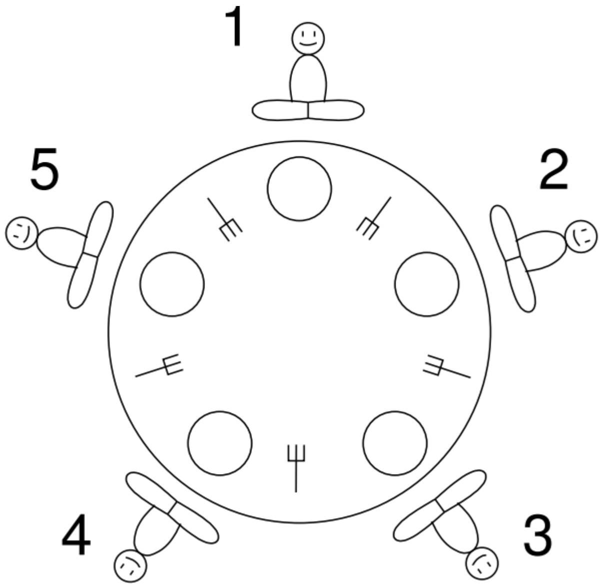
Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

Specifying using 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

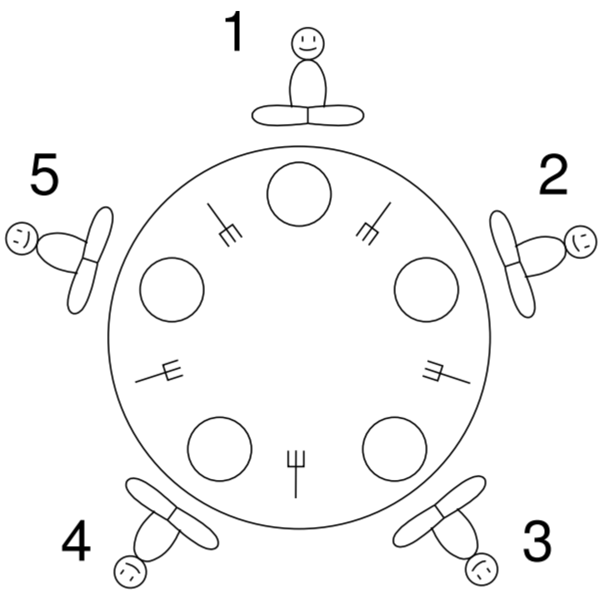
e_i : Philosopher i is currently eating.

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 CTL formula

- "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:
$A\phi$ → **All** ϕ
$E\phi$ → **Exists** 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 CTL formula

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

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

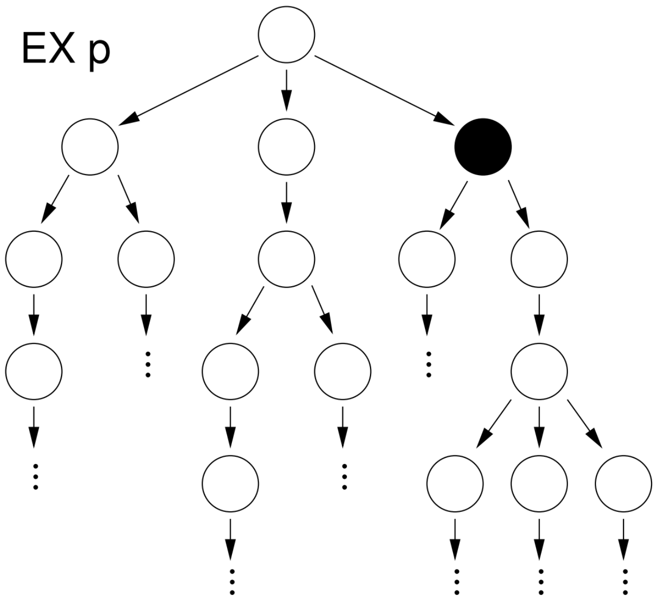
- Now, we can use our "trick": computing with sets of states!
  - $\psi_{[\![\phi]\!]}(q)$ is true if the state q is in the set $[\![\phi]\!]$, i.e., it is a state for which the CTL formula is true.
  - Therefore, we can also say

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

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

Computing CTL formula: EX ϕ

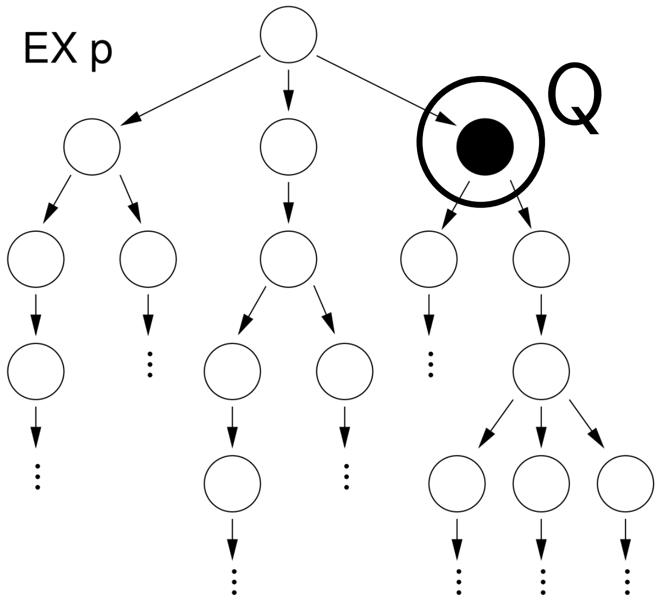
Over paths:	Path-specific:
$A\phi \rightarrow$ **A**ll ϕ	$X\phi \rightarrow$ Ne**X**t ϕ
$E\phi \rightarrow$ **E**xists ϕ	$F\phi \rightarrow$ **F**inally ϕ
	$G\phi \rightarrow$ **G**lobally ϕ
	$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2



Computing 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 initial states for which the formula ϕ is true.



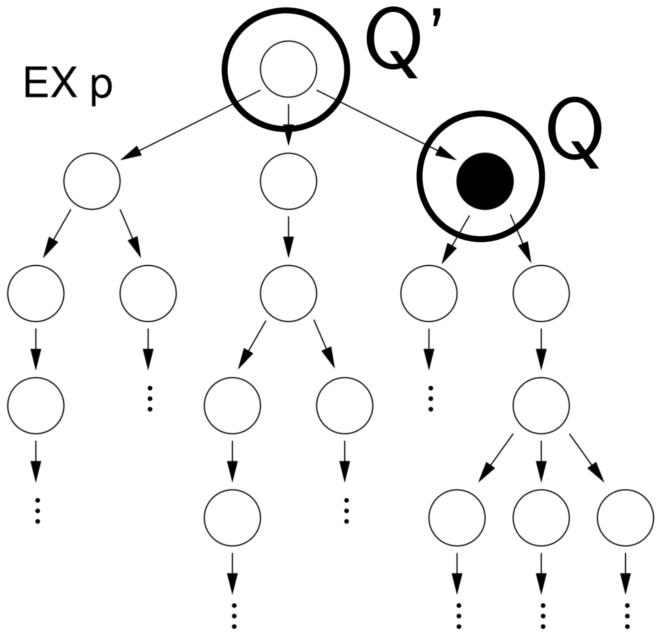
Sets	$Q = [\![\phi]\!]$
Characteristic functions	$\psi_Q(q)$

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

Computing CTL formula: EX ϕ

- Suppose that Q is the set of initial 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	$Q = [\![\phi]\!]$ ⟶	$Q' = [\![\mathrm{EX}\phi]\!] = \mathrm{Pre}([\![\phi]\!], \delta)$
Characteristic functions	$\psi_Q(q)$ ⟶	$\psi_{Q'}(q') = (\exists q \ : \ \psi_Q(q) \cdot \psi_\delta(q', q))$

Over paths:

$A\phi$ → **A**ll ϕ

$E\phi$ → **E**xists ϕ

Path-specific:

$X\phi$ → Ne**X**t ϕ

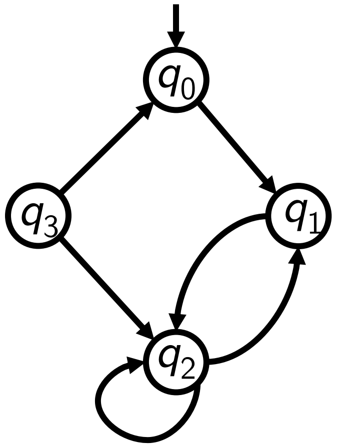
$F\phi$ → **F**inally ϕ

$G\phi$ → **G**lobally ϕ

$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

Computing CTL formula: EX ϕ

- Example for EX ϕ : Compute EX q_2



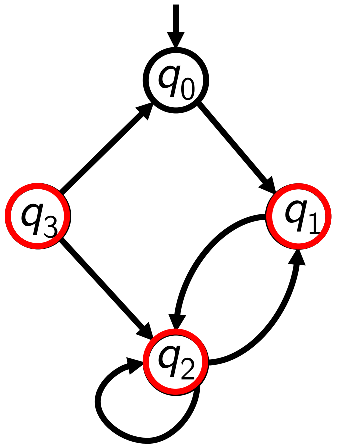
$$[\![q_2]\!] = \{q_2\}$$

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 CTL formula: EX ϕ

- Example for EX ϕ : Compute EX q_2



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

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

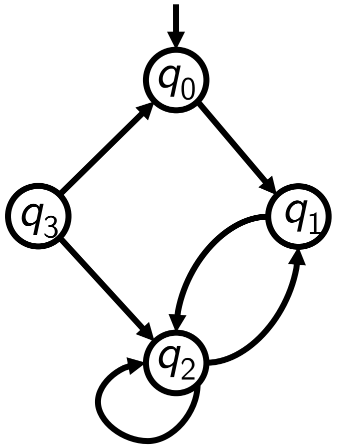
$$\{q' \mid \exists q : \psi_\delta(q', q) \cdot \psi_Q(q)\}$$

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 CTL formula: EX ϕ

- Example for EX ϕ : Compute EX q_2



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

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

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

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

Computing CTL formula: EF ϕ

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

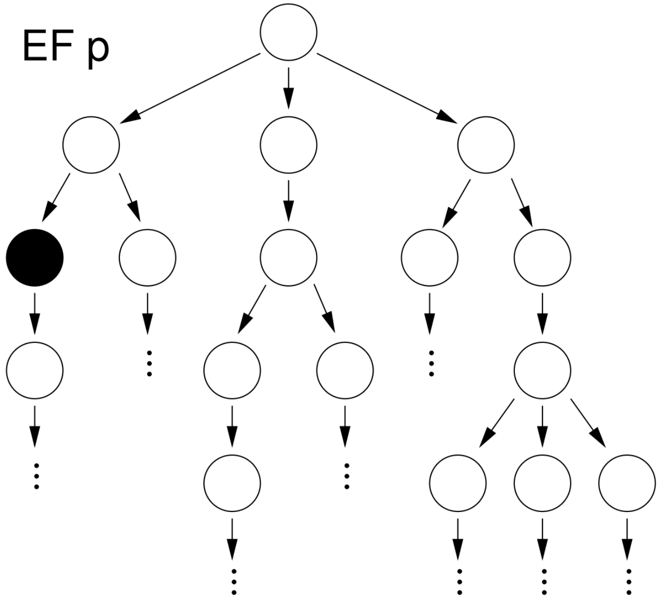
Path-specific:
$X\phi \rightarrow$ Ne**X**t ϕ
$F\phi \rightarrow$ **F**inally ϕ
$G\phi \rightarrow$ **G**lobally ϕ
$\phi_1 U \phi_2 \rightarrow \phi_1$ **U**ntil ϕ_2

- Start with the set of initial 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

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



Over paths:

$A\phi$ → **All** ϕ

$E\phi$ → **Exists** ϕ

Path-specific:

$X\phi$ → Ne**X**t ϕ

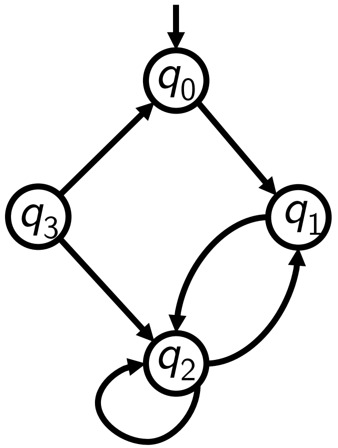
$F\phi$ → **F**inally ϕ

$G\phi$ → **G**lobally ϕ

$\phi_1 U \phi_2$ → ϕ_1 **U**ntil ϕ_2

Computing CTL formula: EF ϕ

- Example for EFϕ: Compute EF q_2



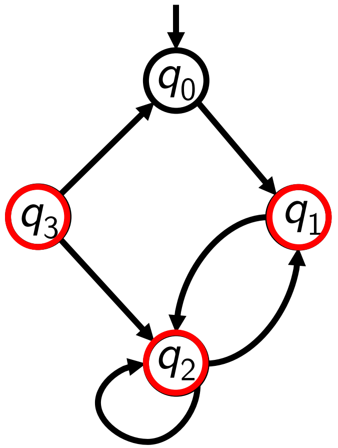
$$Q_0 = [\![q_2]\!] = \{q_2\}$$

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 CTL formula: EF ϕ

- Example for EFϕ: Compute EF q_2



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

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

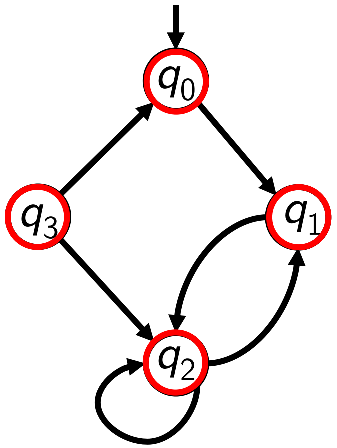
$$\{q' \mid \exists q \text{ with } \psi_Q(q) \cdot \psi_\delta(q', q)\} = \{q_1, q_2, q_3\}$$

Over paths:
$A\phi \rightarrow$ **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 CTL formula: EF ϕ

- Example for EFϕ: Compute EF q_2



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

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

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

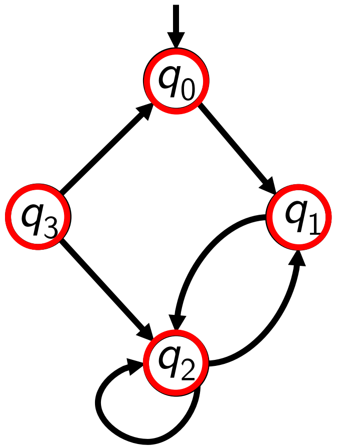
$$Q_2 = \{q_1, q_2, q_3\} \cup \mathrm{Pre}(\{q_1, q_2, q_3\}, \delta) = \{q_0, q_1, q_2, q_3\}$$

Over paths:
$A\phi \to$ **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 CTL formula: EF ϕ

- Example for EFϕ: Compute EF q_2



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

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

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

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

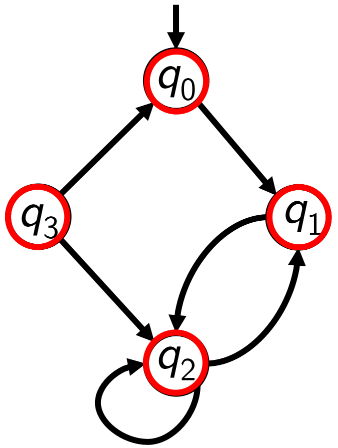
$$Q_3 = \{q_0, q_1, q_2, q_3\} \cup \text{Pre}(\{q_0, q_1, q_2, q_3\}, \delta) = \{q_0, q_1, q_2, q_3\}$$

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

Computing CTL formula: EF 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

- Example for EFϕ: Compute EF q_2



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

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

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

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

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

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

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:
Aϕ → **All** ϕ
Eϕ → **Exists** ϕ

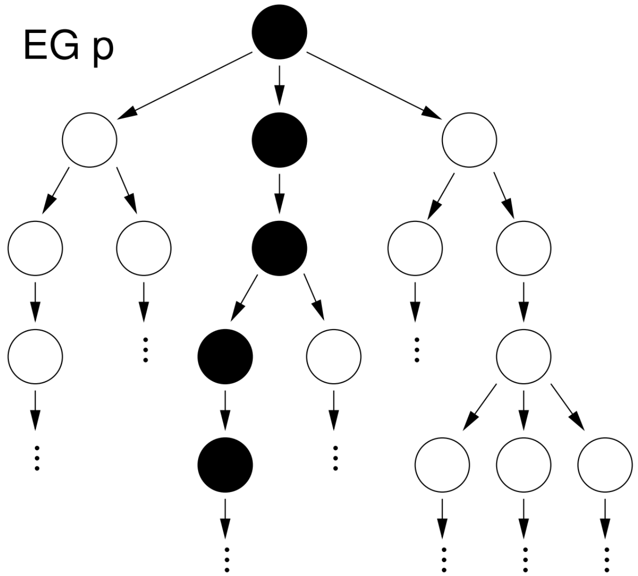
Path-specific:
Xϕ → Ne**X**t ϕ
Fϕ → **F**inally ϕ
Gϕ → **G**lobally ϕ
ϕ_1Uϕ_2 → ϕ_1**U**ntil ϕ_2

Computing CTL formula: EG ϕ

- Start with the set of initial 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

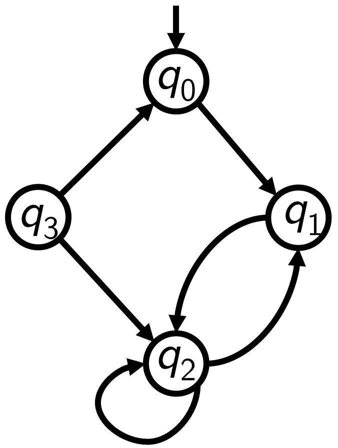


Over paths:
$A\phi \to$ **A**ll ϕ
$E\phi \to$ **E**xists ϕ

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 CTL formula: EG ϕ

- Example for EG ϕ: Compute EG q_2



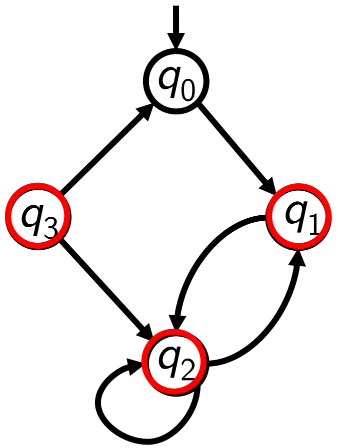
$$Q_0 = [\![q_2]\!] = \{q_2\}$$

Over paths:
Aϕ → **All** ϕ
Eϕ → **Exists** 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 CTL formula: EG ϕ

- Example for EG ϕ: Compute EG q_2



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

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

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

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

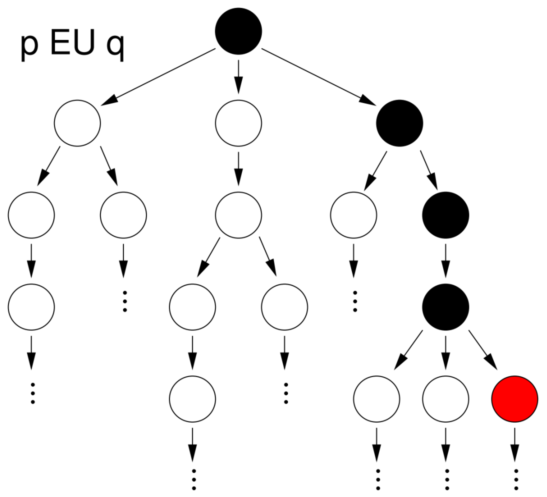
As $q_0 \notin [\![\mathrm{EG}q_2]\!] = \{q_2\}$, the CTL formula EG q_2 is not true.

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 CTL formula: $\phi_1 EU \phi_2$

- Start with the set of initial 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.



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

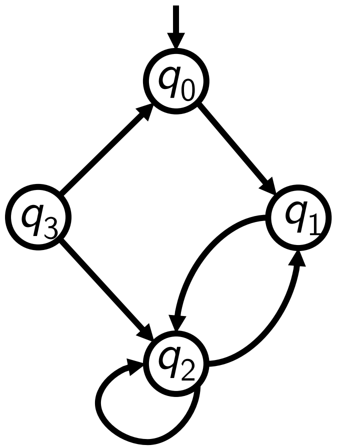
$$Q_i = Q_{i-1} \cup (\mathrm{Pre}(Q_{i-1}, \delta) \cap [\![\phi_1]\!])$$ for all $i > 1$ until a fixed point is reached

Over paths:
$A\phi \rightarrow$ **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 CTL formula: $\phi_1 EU \phi_2$

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



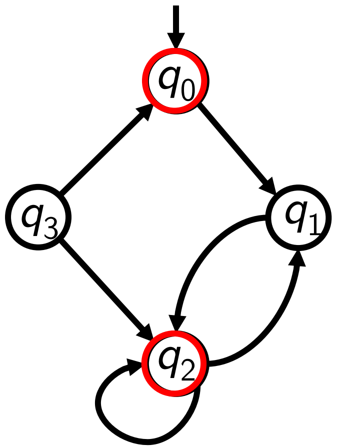
$$Q_0 = [\![q_1]\!] = \{q_1\}$$

Computing 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 $q_0\ EU\ q_1$



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

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

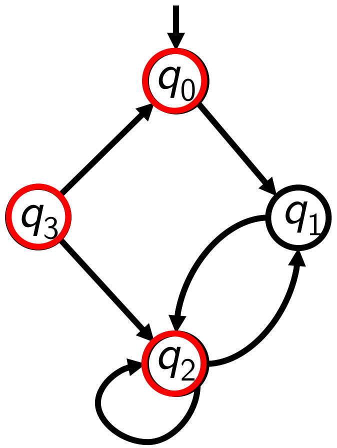
$$Q_1 = \{q_1\} \cup (\text{Pre}(\{q_1\}, \delta) \cap \{q_0\}) = \{q_0, q_1\}$$

Computing CTL formula: $\phi_1 EU \phi_2$

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

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



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

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

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

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

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

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

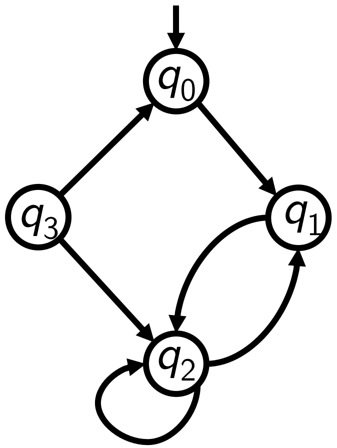
As $q_0 \in [\![q_0 \mathrm{EU} q_1]\!] = \{q_0, q_1\}$, the CTL formula q_0 EU q_1 is true.

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 CTL formula: $\phi_1 EU \phi_2$

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



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

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

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

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

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

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

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 automato
Petri nets
Kripke machine
...

CTL, LTL, ...

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

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

So... what is model-checking exactly?

Model-checking is an algorithm
which takes two inputs

- a DES model $\boldsymbol{M}$ — Finite automato, Petri nets, Kripke machine, ...
- a formula $\boldsymbol{\phi}$ — CTL, LTL, ...

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

- prove that $\boldsymbol{M} \vDash \boldsymbol{\phi}$, 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

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!)

	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

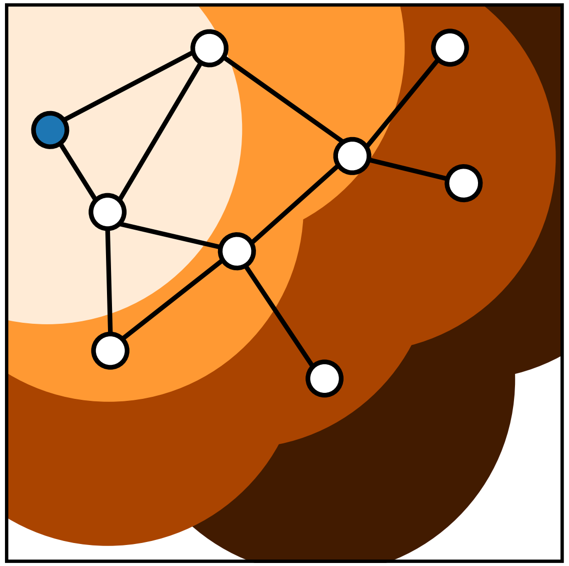
Next week(s)

Petri Nets

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

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

Thanks for your attention and see you next week! ☺



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

ETH Zurich (D-ITET)

December 1, 2022

Most materials from Lothar Thiele and Romain Jacob