

# Model Checking CTL

Sean Raffi

November 18, 2008

## Recall ...

Last week we presented an algorithm that given:

- a CTL formula  $\phi$
- a model  $\mathcal{M} = (S, \rightarrow, L)$

computes the states in  $S$  satisfying  $\phi$ .

## Recall ...

Last week we presented an algorithm that given:

- a CTL formula  $\phi$
- a model  $\mathcal{M} = (S, \rightarrow, L)$

computes the states in  $S$  satisfying  $\phi$ .

## Recall ...

Last week we presented an algorithm that given:

- a CTL formula  $\phi$
- a model  $\mathcal{M} = (S, \rightarrow, L)$

computes the states in  $S$  satisfying  $\phi$ .

## Recall ...

Last week we presented an algorithm that given:

- a CTL formula  $\phi$
- a model  $\mathcal{M} = (S, \rightarrow, L)$

computes the states in  $S$  satisfying  $\phi$ .

## Notation

We'll denote such a set as  $[[\phi]]$ .

## Recap: Top-level Function

$$SAT(\lambda) = \begin{cases} \emptyset & \text{if } \lambda \text{ is } \perp \\ \{s \in S \mid s \in L(S)\} & \text{if } \lambda \text{ is atomic} \\ S \setminus SAT(\phi) & \text{if } \lambda \text{ is } \neg\phi \\ SAT(\phi) \cup SAT(\kappa) & \text{if } \lambda \text{ is } \phi \vee \kappa \\ SAT_{EX}(\phi) & \text{if } \lambda \text{ is EX } \phi \\ SAT_{AF}(\phi) & \text{if } \lambda \text{ is AF } \phi \\ SAT_{EU}(\phi, \kappa) & \text{if } \lambda \text{ is E}[\phi \text{ U } \kappa] \end{cases}$$

# Rewrite: Top-level Function

$$\llbracket \lambda \rrbracket = \begin{cases} \emptyset & \text{if } \lambda \text{ is } \perp \\ \{s \in S \mid s \in L(S)\} & \text{if } \lambda \text{ is atomic} \\ S \setminus \llbracket \phi \rrbracket & \text{if } \lambda \text{ is } \neg\phi \\ \llbracket \phi \rrbracket \cup \llbracket \kappa \rrbracket & \text{if } \lambda \text{ is } \phi \vee \kappa \\ SAT_{EX}(\phi) & \text{if } \lambda \text{ is EX } \phi \\ SAT_{AF}(\phi) & \text{if } \lambda \text{ is AF } \phi \\ SAT_{EU}(\phi, \kappa) & \text{if } \lambda \text{ is E}[\phi \text{ U } \kappa] \end{cases}$$

## SAT<sub>EX</sub> interesting but easy?

- $\llbracket \text{EX } \phi \rrbracket = \text{pre}_\exists(\llbracket \phi \rrbracket)$
- $\mathcal{M}, s \models \text{EX } \phi \iff \exists s' [s \rightarrow s' \wedge \mathcal{M}, s' \models \phi]$

## SAT<sub>EX</sub> interesting but easy?

- $\llbracket \text{EX } \phi \rrbracket = \text{pre}_\exists(\llbracket \phi \rrbracket)$
- $\mathcal{M}, s \models \text{EX } \phi \iff \exists s' [s \rightarrow s' \wedge \mathcal{M}, s' \models \phi]$

## SAT<sub>EX</sub> interesting but easy?

- $\llbracket \text{EX } \phi \rrbracket = \text{pre}_\exists(\llbracket \phi \rrbracket)$
- $\mathcal{M}, s \models \text{EX } \phi \iff \exists s' [s \rightarrow s' \wedge \mathcal{M}, s' \models \phi]$

## SAT<sub>EX</sub> interesting but easy?

- $\llbracket \text{EX } \phi \rrbracket = \text{pre}_\exists(\llbracket \phi \rrbracket)$
- $\mathcal{M}, s \models \text{EX } \phi \iff \exists s' [s \rightarrow s' \wedge \mathcal{M}, s' \models \phi]$

## SAT<sub>EX</sub> interesting but easy?

- $\llbracket \text{EX } \phi \rrbracket = \text{pre}_\exists(\llbracket \phi \rrbracket)$
- $\mathcal{M}, s \models \text{EX } \phi \iff \exists s' [s \rightarrow s' \wedge \mathcal{M}, s' \models \phi]$

## Other Operators

- Correctness of  $SAT_{AF}$ ,  $SAT_{EU}$ ,  $SAT_{EG}$  not as obvious!
- Why?
- Need to iterate labeling procedure until no further change.
  - Does it ever end?
  - If so, is it correct?

## Other Operators

- Correctness of  $SAT_{AF}$ ,  $SAT_{EU}$ ,  $SAT_{EG}$  not as obvious!
- Why?
- Need to iterate labeling procedure until no further change.
  - Does it ever end?
  - If so, is it correct?

## Other Operators

- Correctness of  $SAT_{AF}$ ,  $SAT_{EU}$ ,  $SAT_{EG}$  not as obvious!
- Why?
- Need to iterate labeling procedure until no further change.
  - Does it ever end?
  - If so, is it correct?

## Other Operators

- Correctness of  $SAT_{AF}$ ,  $SAT_{EU}$ ,  $SAT_{EG}$  not as obvious!
- Why?
- Need to iterate labeling procedure until no further change.
  - Does it ever end?
  - If so, is it correct?

## Other Operators

- Correctness of  $SAT_{AF}$ ,  $SAT_{EU}$ ,  $SAT_{EG}$  not as obvious!
- Why?
- Need to iterate labeling procedure until no further change.
  - Does it ever end?
  - If so, is it correct?

# The Plan

- We will prove the correctness of  $SAT_{EU}$ .
  - Computes in finite time.
  - Terminates with the correct answer.
- Will prove correctness of  $SAT_{EG}$ , time permitting.

# The Plan

- We will prove the correctness of  $SAT_{EU}$ .
  - Computes in finite time.
  - Terminates with the correct answer.
- Will prove correctness of  $SAT_{EG}$ , time permitting.

# The Plan

- We will prove the correctness of  $SAT_{EU}$ .
  - Computes in finite time.
  - Terminates with the correct answer.
- Will prove correctness of  $SAT_{EG}$ , time permitting.

# The Plan

- We will prove the correctness of  $SAT_{EU}$ .
  - Computes in finite time.
  - Terminates with the correct answer.
- Will prove correctness of  $SAT_{EG}$ , time permitting.

# Intuition

## The semantics of $E[\phi \cup \kappa]$

$$\mathcal{M}, s_0 \models E[\phi \cup \kappa]$$

$$\iff$$

$$\exists \langle s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rangle, i \geq 0 \\ [\mathcal{M}, s_i \models \kappa \wedge \forall 0 \leq j < i [\mathcal{M}, s_j \models \phi]]$$

# Intuition

The semantics of  $E[\phi \cup \kappa]$

$$\mathcal{M}, s_0 \models E[\phi \cup \kappa]$$

$$\iff$$

$$\exists \langle s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rangle, i \geq 0 \\ [\mathcal{M}, s_i \models \kappa \wedge \forall 0 \leq j < i [\mathcal{M}, s_j \models \phi]]$$

# Intuition

## The semantics of $E[\phi \cup \kappa]$

$$\mathcal{M}, s_0 \models E[\phi \cup \kappa]$$

$$\iff$$

$$\exists \langle s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rangle, i \geq 0$$

$$[\mathcal{M}, s_i \models \kappa \wedge \forall 0 \leq j < i [\mathcal{M}, s_j \models \phi]]$$

# Intuition

The semantics of  $E[\phi \cup \kappa]$

$$\begin{aligned} \mathcal{M}, s_0 \models E[\phi \cup \kappa] \\ \iff \\ \exists \langle s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rangle, i \geq 0 \\ [\mathcal{M}, s_i \models \kappa \wedge \forall 0 \leq j < i [\mathcal{M}, s_j \models \phi]] \end{aligned}$$

# Intuition

## Equivalences

$$\begin{aligned} E[\phi \cup \kappa] &\equiv \kappa \vee (\phi \wedge EX E[\phi \cup \kappa]) \\ \llbracket EX \psi \rrbracket &= \llbracket pre_{\exists}(\llbracket \psi \rrbracket) \rrbracket \end{aligned}$$

# Intuition

## Equivalences

$$\begin{aligned} E[\phi \text{ U } \kappa] &\equiv \kappa \vee (\phi \wedge EX E[\phi \text{ U } \kappa]) \\ \llbracket EX \psi \rrbracket &= \llbracket \text{pre}_{\exists}(\llbracket \psi \rrbracket) \rrbracket \end{aligned}$$

# Intuition

## Equivalences

$$E[\phi \text{ U } \kappa] \equiv \kappa \vee (\phi \wedge EX E[\phi \text{ U } \kappa])$$

$$\llbracket EX \psi \rrbracket = \llbracket pre_{\exists}(\llbracket \psi \rrbracket) \rrbracket$$

# Intuition

## Equivalences

$$E[\phi \text{ U } \kappa] \equiv \kappa \vee (\phi \wedge EX E[\phi \text{ U } \kappa])$$

$$\llbracket EX \psi \rrbracket = \llbracket pre_{\exists}(\llbracket \psi \rrbracket) \rrbracket$$

# Intuition

## Equivalences

$$E[\phi \text{ U } \kappa] \equiv \kappa \vee (\phi \wedge \text{EX } E[\phi \text{ U } \kappa])$$

$$\llbracket \text{EX } \psi \rrbracket = \llbracket \text{pre}_{\exists}(\llbracket \psi \rrbracket) \rrbracket$$

## Recall ...

**function**  $SAT_{EU}(\phi, \kappa)$

```
1:  $X \leftarrow S$ 
2:  $Y \leftarrow SAT(\kappa)$ 
3:  $W \leftarrow SAT(\phi)$ 
4: repeat
5:    $X \leftarrow Y$ 
6:    $Y \leftarrow Y \cup (W \cap pre_{\exists}(Y))$ 
7: until  $X = Y$ 
8: return  $Y$ 
```

## Recall ...

**function**  $SAT_{EU}(\phi, \kappa)$

```
1:  $X \leftarrow S$   
2:  $Y \leftarrow SAT(\kappa)$   
3:  $W \leftarrow SAT(\phi)$   
4: repeat  
5:    $X \leftarrow Y$   
6:    $Y \leftarrow Y \cup (W \cap pre_{\exists}(Y))$   
7: until  $X = Y$   
8: return  $Y$ 
```

## Recall ...

**function**  $SAT_{EU}(\phi, \kappa)$

1:  $X \leftarrow S$

2:  $Y \leftarrow SAT(\kappa)$

3:  $W \leftarrow SAT(\phi)$

4: **repeat**

5:    $X \leftarrow Y$

6:    $Y \leftarrow Y \cup (W \cap pre_{\exists}(Y))$

7: **until**  $X = Y$

8: **return**  $Y$

## Recall ...

**function**  $SAT_{EU}(\phi, \kappa)$

1:  $X \leftarrow S$

2:  $Y \leftarrow SAT(\kappa)$

3:  $W \leftarrow SAT(\phi)$

4: **repeat**

5:    $X \leftarrow Y$

6:    $Y \leftarrow Y \cup (W \cap pre_{\exists}(Y))$

7: **until**  $X = Y$

8: **return**  $Y$

## Recall ...

**function**  $SAT_{EU}(\phi, \kappa)$

1:  $X \leftarrow S$

2:  $Y \leftarrow SAT(\kappa)$

3:  $W \leftarrow SAT(\phi)$

4: **repeat**

5:  $X \leftarrow Y$

6:  $Y \leftarrow Y \cup (W \cap pre_{\exists}(Y))$

7: **until**  $X = Y$

8: **return**  $Y$

## Recall ...

**function**  $SAT_{EU}(\phi, \kappa)$

1:  $X \leftarrow S$

2:  $Y \leftarrow SAT(\kappa)$

3:  $W \leftarrow SAT(\phi)$

4: **repeat**

5:    $X \leftarrow Y$

6:    $Y \leftarrow Y \cup (W \cap pre_{\exists}(Y))$

7: **until**  $X = Y$

8: **return**  $Y$

## Recall ...

**function**  $SAT_{EU}(\phi, \kappa)$

```
1:  $X \leftarrow S$   
2:  $Y \leftarrow SAT(\kappa)$   
3:  $W \leftarrow SAT(\phi)$   
4: repeat  
5:    $X \leftarrow Y$   
6:    $Y \leftarrow Y \cup (W \cap pre_{\exists}(Y))$   
7: until  $X = Y$   
8: return  $Y$ 
```

## Recall ...

**function**  $SAT_{EU}(\phi, \kappa)$

```
1:  $X \leftarrow S$   
2:  $Y \leftarrow SAT(\kappa)$   
3:  $W \leftarrow SAT(\phi)$   
4: repeat  
5:    $X \leftarrow Y$   
6:    $Y \leftarrow Y \cup (W \cap pre_{\exists}(Y))$   
7: until  $X = Y$   
8: return  $Y$ 
```

# Intuition

## Equivalences

$$\mathbf{E}[\phi \mathbf{U} \kappa] \equiv \kappa \vee (\phi \wedge \mathbf{EX} \mathbf{E}[\phi \mathbf{U} \kappa])$$

$$\llbracket \mathbf{EX} \psi \rrbracket = \llbracket \text{pre}_{\exists}(\llbracket \psi \rrbracket) \rrbracket$$

$$\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$$

# Intuition

## Equivalences

$$\mathbf{E}[\phi \mathbf{U} \kappa] \equiv \kappa \vee (\phi \wedge \mathbf{EX} \mathbf{E}[\phi \mathbf{U} \kappa])$$

$$\llbracket \mathbf{EX} \psi \rrbracket = \llbracket \text{pre}_{\exists}(\llbracket \psi \rrbracket) \rrbracket$$

$$\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$$

# Intuition

## Equivalences

$$\begin{aligned} \mathbf{E}[\phi \mathbf{U} \kappa] &\equiv \kappa \vee (\phi \wedge \mathbf{EX} \mathbf{E}[\phi \mathbf{U} \kappa]) \\ \llbracket \mathbf{EX} \psi \rrbracket &= \llbracket \text{pre}_{\exists}(\llbracket \psi \rrbracket) \rrbracket \\ \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) \end{aligned}$$

# Intuition

## Equivalences

$$\begin{aligned} \mathbf{E}[\phi \mathbf{U} \kappa] &\equiv \kappa \vee (\phi \wedge \mathbf{EX} \mathbf{E}[\phi \mathbf{U} \kappa]) \\ \llbracket \mathbf{EX} \psi \rrbracket &= \llbracket \text{pre}_{\exists}(\llbracket \psi \rrbracket) \rrbracket \\ \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) \end{aligned}$$

# Intuition

## Equivalences

$$\begin{aligned} E[\phi \text{ U } \kappa] &\equiv \kappa \vee (\phi \wedge EX E[\phi \text{ U } \kappa]) \\ \llbracket EX \psi \rrbracket &= \llbracket \text{pre}_{\exists}(\llbracket \psi \rrbracket) \rrbracket \\ \llbracket E[\phi \text{ U } \kappa] \rrbracket &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket E[\phi \text{ U } \kappa] \rrbracket) \end{aligned}$$

## Circularity!

How do we solve this?

# Definitions

Let  $S$  be a set of states and  $F : P(S) \rightarrow P(S)$ , a function on the power set of  $S$ .

**Monotone Functions:** We say that  $F$  is *monotone* iff  $X \subseteq Y$  implies  $F(X) \subseteq F(Y)$  for all subsets  $X$  and  $Y$  of  $S$ .

**Fixed Points:** A subset  $X$  of  $S$  is called a *fixed point* of  $F$  iff  $F(X) = X$ .

# Example

$S$  defined as  $\{s_0, s_1\}$

$F(X)$  defined as  $X \cup \{s_0\}, \forall X \subseteq S$

*Is it monotonous?*

# Why monotone functions?

- They always have a least and greatest fixed point.
- The semantics of EG, AF, and EU can be expressed as greatest or least fixed points of monotonous functions.
- These fixed points are "easily computed".
- The procedures for  $\text{SAT}_{\text{EG}}$  and  $\text{SAT}_{\text{EU}}$  code to such fixed point computations, so they're correct by #2.

*Requires some proof, right?*

# Knaster-Tarski Theorem

Let  $S$  be a set  $\{s_0, s_1, \dots, s_n\}$  with  $n+1$  elements.

If  $F : P(S) \rightarrow P(S)$  is a monotone function, then  $F^{n+1}(\emptyset)$  is the least fixed point of  $F$  and  $F^{n+1}(S)$  is the greatest fixed point of  $F$ .

*(Notation note:  $F^i$  is just shorthand for  $F$  applied  $i$  times)*

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the **least** fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the **least** fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the **least** fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the **least** fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the *least* fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the *least* fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  

$$G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the *least* fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the *least* fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \mathbf{U} \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the *least* fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists} \llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  
 $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X)) \\ G(\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the **least** fixed point.

## Recall ...

- $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists \llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket)$
- Define  $G: S \rightarrow S$  to be the function  

$$G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists(X))$$
- We have that:

$$\begin{aligned} G(X) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists(X)) \\ G(\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket) &= \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists(\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket)) \\ &= \llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket \end{aligned}$$

- Implies that  $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket$  is a fixed point of  $G$ .
- Turns out that it is the **least** fixed point.

### Theorem (3.7.3)

Let  $S$  be the *finite* set of states  $S = \{s_0, s_1, \dots, s_n\}$  and  $G: S \rightarrow S$  be the function  $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$ . Then,  $G$  is monotone,  $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket$  is the least fixed point of  $G$ , and we have that  $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket = G^{n+1}(\emptyset)$ .

### Theorem (3.7.3)

Let  $S$  be the *finite* set of states  $S = \{s_0, s_1, \dots, s_n\}$  and  $G: S \rightarrow S$  be the function  $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$ . Then,  $G$  is monotone,  $\llbracket \mathbf{E}[\phi \ U \ \kappa] \rrbracket$  is the least fixed point of  $G$ , and we have that  $\llbracket \mathbf{E}[\phi \ U \ \kappa] \rrbracket = G^{n+1}(\emptyset)$ .

### Remark

Notice that  $S$  has  $n + 1$  elements.

### Theorem (3.7.3)

Let  $S$  be the *finite* set of states  $S = \{s_0, s_1, \dots, s_n\}$  and  $G: S \rightarrow S$  be the function  $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_\exists(X))$ . Then,  $G$  is monotone,  $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket$  is the least fixed point of  $G$ , and we have that  $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket = G^{n+1}(\emptyset)$ .

Proof: Assume  $S = \{s_0, s_1, \dots, s_n\}$  and that  $G$  is defined as above.

### Claim

$G$  is monotone. That is,  $X \subseteq Y \implies G(X) \subseteq G(Y)$ .

- Assume  $X \subseteq Y$ .
- Show  $\forall s \in G(X)[s \in G(Y)]$ .
- Take any  $s' \in G(X)$ .
  - Case 1:  $s' \in \llbracket \kappa \rrbracket$ . Clearly,  $s' \in G(Y)$ .
  - Case 2:  $s' \in \llbracket \phi \rrbracket \cap \text{pre}_\exists(X)$ . Then,  $s' \in \llbracket \phi \rrbracket \wedge s' \in \text{pre}_\exists(X)$ . Show that  $s' \in \text{pre}_\exists(X) \implies s' \in \text{pre}_\exists(Y)$ .

### Theorem (3.7.3)

Let  $S$  be the *finite* set of states  $S = \{s_0, s_1, \dots, s_n\}$  and  $G: S \rightarrow S$  be the function  $G(X) = \llbracket \kappa \rrbracket \cup (\llbracket \phi \rrbracket \cap \text{pre}_{\exists}(X))$ . Then,  $G$  is monotone,  $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket$  is the least fixed point of  $G$ , and we have that  $\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket = G^{n+1}(\emptyset)$ .

### Claim

$\llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket$  is the least fixed point of  $G$ .

- $S$  has  $n + 1$  states. Then,  $LFP(G) = G^{n+1}(\emptyset)$  by K-T.
- Suffices to show  $G^{n+1}(\emptyset) = \llbracket \mathbf{E}[\phi \text{ U } \kappa] \rrbracket$ .

## The correctness of $SAT_{EU}$

- Change  $Y \leftarrow Y \cup (W \cap \text{pre}_{\exists}(Y))$  to  $Y \leftarrow \llbracket \kappa \rrbracket \cup (W \cap \text{pre}_{\exists}(Y))$ .
- Does not change result of the algorithm.
- “Clear” that  $SAT_{EU}$  is just computing the least fixed point of  $G$  using  $K - T$ . ???



## The correctness of $SAT_{EU}$

- Change  $Y \leftarrow Y \cup (W \cap \text{pre}_{\exists}(Y))$  to  $Y \leftarrow \llbracket \kappa \rrbracket \cup (W \cap \text{pre}_{\exists}(Y))$ .
- Does not change result of the algorithm.
- “Clear” that  $SAT_{EU}$  is just computing the least fixed point of  $G$  using  $K - T$ . ???



## The correctness of $SAT_{EU}$

- Change  $Y \leftarrow Y \cup (W \cap \text{pre}_{\exists}(Y))$  to  $Y \leftarrow \llbracket \kappa \rrbracket \cup (W \cap \text{pre}_{\exists}(Y))$ .
- Does not change result of the algorithm.
- “Clear” that  $SAT_{EU}$  is just computing the least fixed point of  $G$  using  $K - T$ . ???

