Master in Artificial Intelligence and Robotics (AIRO)
Electives in AI
Reasoning Agents

Fabio Patrizi

Sapienza University of Rome, Italy
patrizi@diag.uniroma1.it

A.Y. 2021-2022

CTL Model Checking*

*Slides originally by A. Artale (https://www.inf.unibz.it/~artale/), here restyled and modified

References:

1. A fully detailed presentation of the topics discussed in these slides can be found in [CGP99]

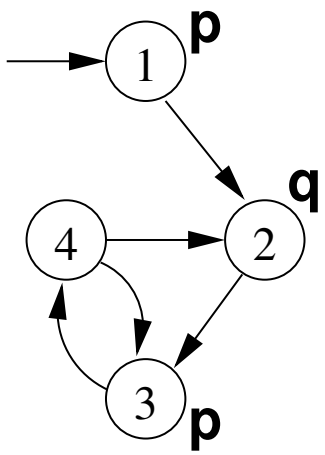
Summary

- CTL Model Checking: General Ideas
- CTL Model Checking: The Labeling Algorithm
- Labeling Algorithm in Details
- CTL Model Checking: Theoretical Issues

CTL Model Checking

CTL Model Checking is an analysis technique, where:

- The world is modeled as a TS $\mathcal{T}$:



- The property is expressed as a CTL formula φ, e.g.:

$$\mathbf{AG}(p \rightarrow \mathbf{AF}q)$$

- The algorithm checks whether $\mathcal{T} \models \varphi$

CTL Model Checking Algorithm: General Ideas

Given $\mathcal{T}$ and φ, the algorithm proceeds as follows:

1. Compute the set of states where the formula holds (called the *denotation* of φ)

$$[\![\varphi]\!] := \{s \in S : \mathcal{T}, s \models \varphi\}$$

2. Check whether $s_0 \in [\![\varphi]\!]$

CTL Model Checking Algorithm: General Ideas

To compute $[\![\varphi]\!]$ proceed "bottom-up" on the structure of the formula, computing $[\![\varphi_i]\!]$ for each subformula φ_i of φ.

For example, to compute $[\![\mathbf{AG}(p \rightarrow \mathbf{AF}q)]\!]$ we need to compute:

- $[\![q]\!]$,
- $[\![\mathbf{AF}q]\!]$,
- $[\![p]\!]$,
- $[\![p \rightarrow \mathbf{AF}q]\!]$,
- $[\![\mathbf{AG}(p \rightarrow \mathbf{AF}q)]\!]$

CTL Model Checking Algorithm: General Ideas

To compute each $[\![\varphi_i]\!]$ for generic subformulas:

- Handle boolean operators by standard set operations;
- Handle temporal operators **AX**, **EX** by computing pre-images;
- Handle temporal operators **AG**, **EG**, **AF**, **EF**, **AU**, **EU**, by applying fixpoint operators.

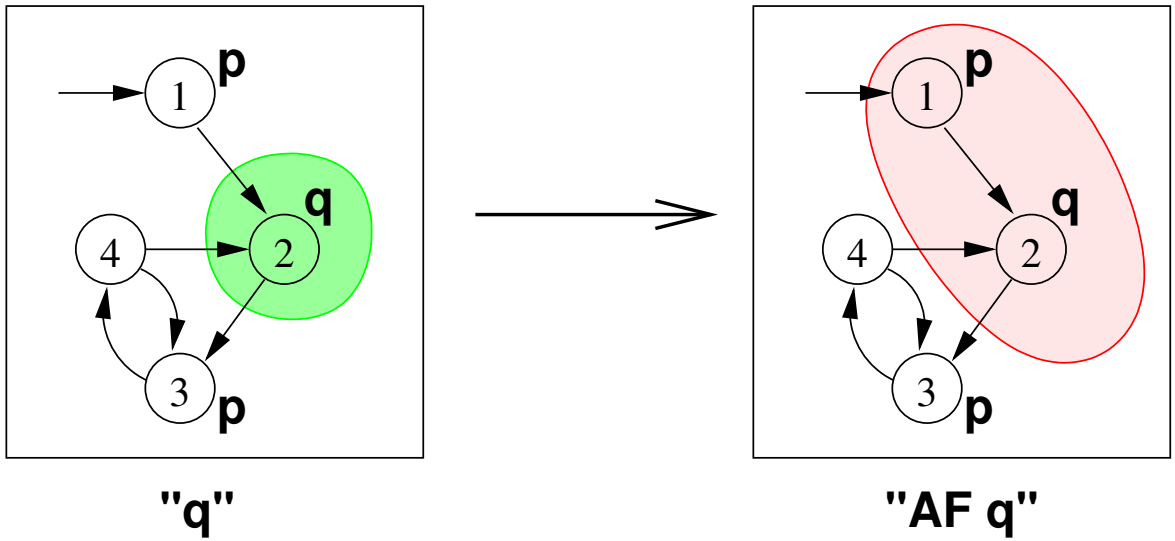
Summary

- CTL Model Checking: General Ideas.
- CTL Model Checking: The Labeling Algorithm.
- Labeling Algorithm in Details.
- CTL Model Checking: Theoretical Issues.

The Labeling Algorithm: General Idea

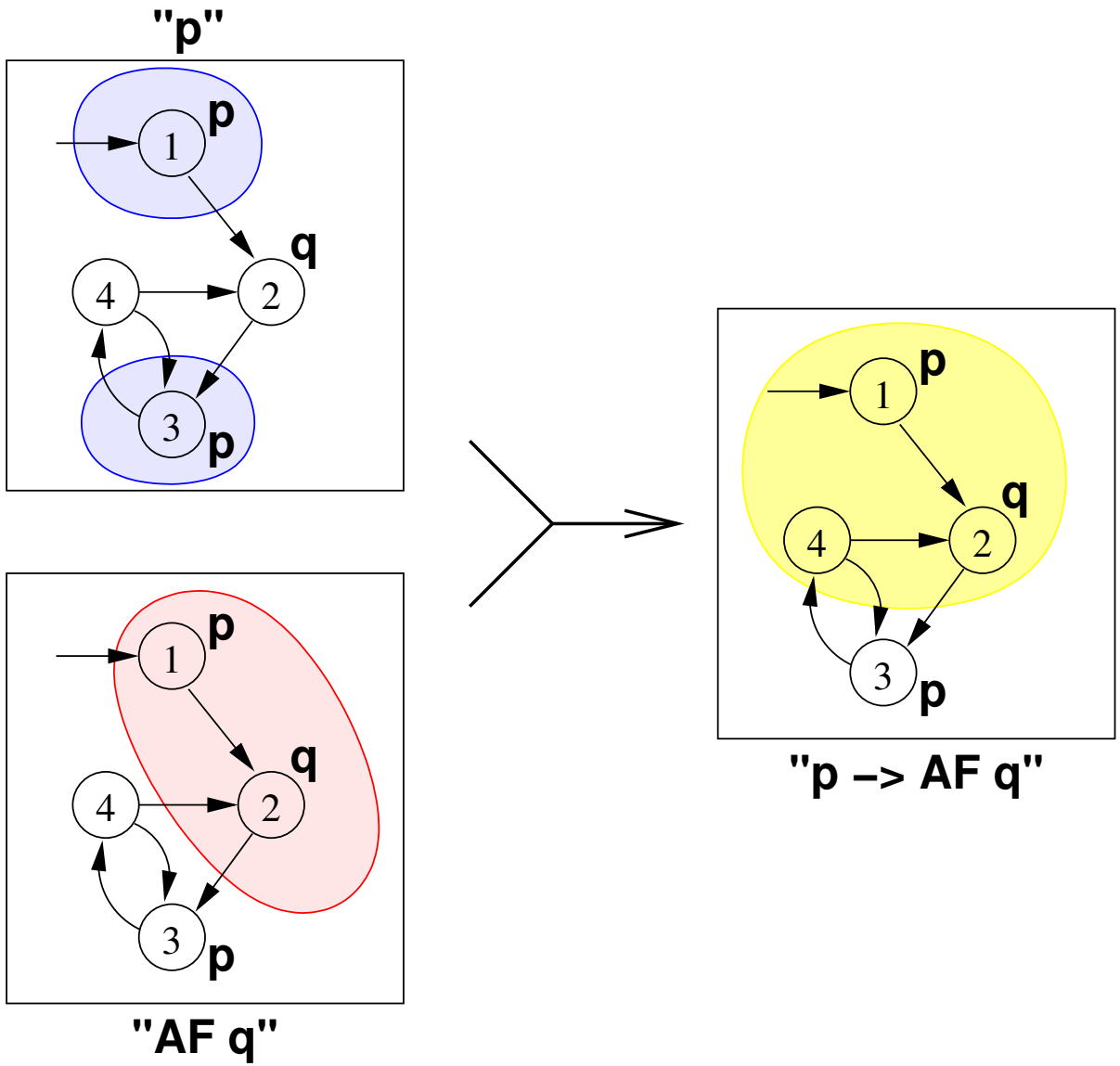
- The Labeling Algorithm given a TS and a CTL formula outputs the set of states satisfying the formula.
- Main Idea: Label the states of the TS with the subformulas of φ satisfied there.

The Labeling Algorithm: An Example

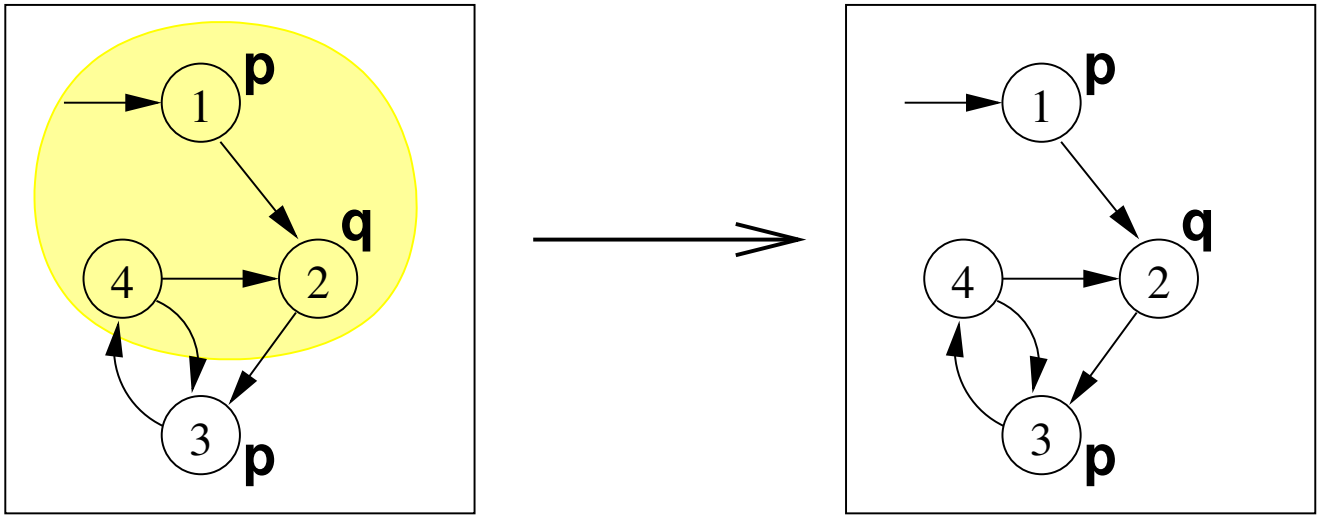


- $\mathbf{AF}q \equiv (q \vee \mathbf{AX}(\mathbf{AF}q))$
- $[\![\mathbf{AF}q]\!]$ can be computed as the union of:
  - $[\![q]\!] = \{2\}$
  - $[\![q \vee \mathbf{AX}q]\!] = \{2\} \cup \{1\} = \{1,2\}$
  - $[\![q \vee \mathbf{AX}(q \vee \mathbf{AX}q)]\!] = \{2\} \cup \{1\} = \{1,2\}$ (fixpoint).

The Labeling Algorithm: An Example



The Labeling Algorithm: An Example



"p –> AF q" "AG(p –> AF q)"

- $\mathbf{AG}\varphi \equiv (\varphi \wedge \mathbf{AX}(\mathbf{AG}\varphi))$
- $[\![\mathbf{AG}\varphi]\!]$ can be computed as the intersection of:
  - $[\![\varphi]\!] = \{1,2,4\}$
  - $[\![\varphi \wedge \mathbf{AX}\varphi]\!] = \{1,2,4\} \cap \{1,3\} = \{1\}$
  - $[\![\varphi \wedge \mathbf{AX}(\varphi \wedge \mathbf{AX}\varphi)]\!] = \{1,2,4\} \cap \{\} = \{\}$ (fixpoint)

The Labeling Algorithm: An Example

- The set of states where the formula holds is empty, thus:
  - The initial state does not satisfy the property;
  - $\mathcal{T} \not\models \mathbf{AG}(p \rightarrow \mathbf{AF}q)$.

- Counterexample: A lazo-shaped path: $1, 2, \{3, 4\}^{\omega}$ (satisfying $\mathbf{EF}(p \wedge \mathbf{EG}\neg q)$)

Summary

- CTL Model Checking: General Ideas.
- CTL Model Checking: The Labeling Algorithm.
- Labeling Algorithm in Details.
- CTL Model Checking: Theoretical Issues.

The Labeling Algorithm: General Schema

- Assume φ in terms of $\neg$, $\wedge$, **EX**, **EU**, **EG** (minimal set of operators)
- The Labeling algorithm takes a CTL formula and a TS as input and returns the *denotation* of φ:
  1. For every $\varphi_i \in Sub(\varphi)$, find $[\![\varphi_i]\!]$;
  2. Compute $[\![\varphi]\!]$ starting from $[\![\varphi_i]\!]$;
  3. Check whether $s_0 \in [\![\varphi]\!]$.
- Subformulas $Sub(\varphi)$ of φ are checked bottom-up
- To compute each $[\![\varphi_i]\!]$: if the main operator of φ_i is a
  - *Boolean Operator*: apply standard set operations;
  - *Temporal Operator*: apply recursive rules until a fixpoint is reached.

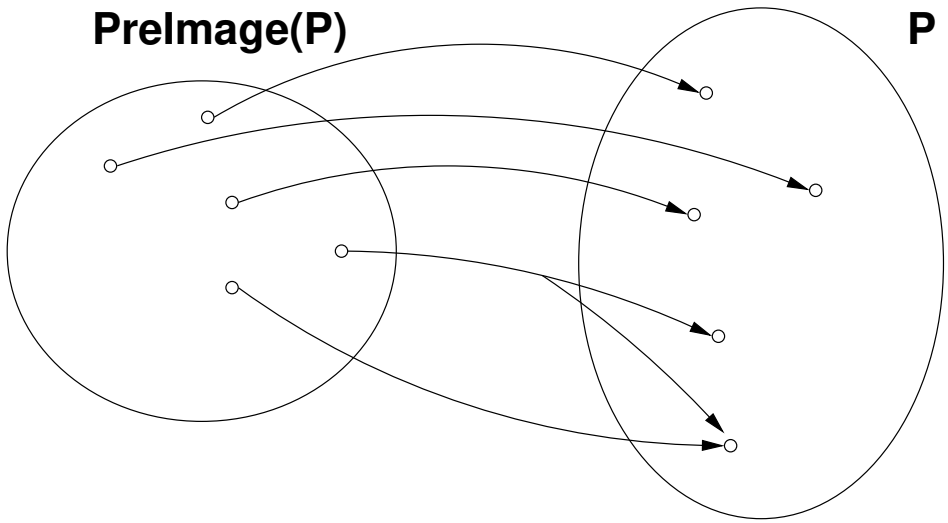
Denotation of Formulas: The Boolean Case

For a TS $\mathcal{T} = (P, A, S, s_0, \rightarrow, \lambda)$, we have:

- $[\![p]\!] = \{s \mid p \in \lambda(s)\}$ (time $O(|S|)$)
- $[\![\neg\varphi_1]\!] = S \setminus [\![\varphi_1]\!]$ (time $O(|S|)$)
- $[\![\varphi_1 \wedge \varphi_2]\!] = [\![\varphi_1]\!] \cap [\![\varphi_2]\!]$ (time $O(|S|)$)
- Time $O(|S|)$ for all operations

Denotation of Formulas: The **EX** Case

- $[\![\mathbf{EX}\varphi]\!] = \{s \in S \mid \exists s'.s \to s' \text{ and } s' \in [\![\varphi]\!]\}$
- $[\![\mathbf{EX}\varphi]\!]$ is called the (existential) pre-image of $[\![\varphi]\!]$ ($\textsc{Pre}([\![\varphi]\!])$)
- Key step of Labeling Algorithm



- Time $O(|\delta|) = O(|S|^2)$

Denotation of Formulas: The **EG** Case

- The following equivalence holds:
  $$\mathbf{EG}\varphi \equiv \varphi \wedge \mathbf{EX}(\mathbf{EG}\varphi)$$
- $[\![\mathbf{EG}\varphi]\!]$ can be computed through following recursive definition:
  $$[\![\mathbf{EG}\varphi]\!] = [\![\varphi]\!] \cap \textsc{Pre}([\![\mathbf{EG}\varphi]\!])$$

Denotation of Formulas: The **EG** Case

- Can compute $X := [\![\mathbf{EG}\varphi]\!]$ as follows:
  - $X_1 := [\![\varphi]\!]$
  - $X_2 := X_1 \cap \textsc{Pre}(X_1)$
  - $\vdots$
  - $X_{j+1} := X_j \cap \textsc{Pre}(X_j)$
  - $\vdots$
- When $X_n = X_{n+1}$ we reach a fixpoint and stop
- Since $X_{j+1} \subseteq X_j$, fixpoint exists (Tarski-Knaster theorem)
- Time: $O(|S| \cdot (|S| + |\delta|)) = O(|S|^3)$

Denotation of Formulas: The **EU** Case

- The following equivalence holds:
  $$(\varphi \mathbf{EU} \psi) \equiv \psi \vee (\varphi \wedge \mathbf{EX}(\varphi \mathbf{EU} \psi))$$

- $[\![\varphi \mathbf{EU} \psi]\!]$ can be computed through following recursive definition:
  $$[\![(\varphi \mathbf{EU} \psi)]\!] = [\![\psi]\!] \cup ([\![\varphi]\!] \cap \textsc{Pre}([\![(\varphi \mathbf{EU} \psi)]\!]))$$

Denotation of Formulas: The **EU** Case

- Can compute $X := [\![(\varphi \mathbf{EU} \psi)]\!]$ as follows:
  - $X_1 := [\![\psi]\!]$
  - $X_2 := X_1 \cup ([\![\varphi]\!] \cap \textsc{Pre}(X_1))$
  - $\vdots$
  - $X_{j+1} := X_j \cup ([\![\varphi]\!] \cap \textsc{Pre}(X_j))$
  - $\vdots$
- When $X_n = X_{n+1}$ we reach a fixpoint and stop
- Since $X_{j+1} \supseteq X_j$, fixpoint exists (Tarski-Knaster theorem)
- Time: $O(|S| \cdot (|S| + |\delta|)) = O(|S|^3)$

The Pseudo-Code

Input:

- TS $\mathcal{T} = (P, A, S, s_0, \rightarrow, \lambda)$
- CTL formula φ

FUNCTION Label(φ) {

case φ **of**

an atom p:	**return** $\{s \in S \mid p \in \lambda(s)\}$;
$\neg\varphi_1$:	**return** $S \backslash \text{Label}(\varphi_1)$;
$\varphi_1 \wedge \varphi_2$:	**return** $\text{Label}(\varphi_1) \cap \text{Label}(\varphi_2)$;
EXφ_1:	**return** PRE(Label(φ_1));
(φ_1 **EU** φ_2):	**return** Label_EU(Label(φ_1),Label(φ_2));
EGφ_1:	**return** Label_EG(Label(φ_1));

end case

}

PreImage

$[\![\mathbf{EX}\varphi]\!] = \textsc{Pre}([\![\varphi]\!]) = \{s \in S \mid \exists s'.s \to s' \text{ and } s' \in [\![\varphi]\!]\}$

$\textsc{Function Pre}([\![\varphi]\!])\{$

var X;

$X := \{\}$;

for each $s' \in [\![\varphi]\!]$ **do**

for each $s \in S$ **such that** $s \to s'$ **do**

$X := X \cup \{s\}$;

return X

$\}$

Label_EG

$[\![\mathbf{EG}\varphi]\!] = [\![\varphi]\!] \cap \textsc{Pre}([\![\mathbf{EG}\varphi]\!])$

$\textsc{Function Label\_EG}([\![\varphi]\!])\{$

var X, X_{old};
$X := [\![\varphi]\!]$;
$X_{old} := \emptyset$;
while $X \neq X_{old}$
begin
$X_{old} := X$;
$X := X \cap \textsc{Pre}(X)$
end
return X

$\}$

Label_EU

$[\![(\varphi \mathbf{EU} \psi)]\!] = [\![\psi]\!] \cup ([\![\varphi]\!] \cap \textsc{Pre}([\![(\varphi \mathbf{EU} \psi)]\!]))$

$\textsc{Function Label\_EU}([\![\varphi]\!],[\![\psi]\!])\{$

var X, X_{old};
$X := [\![\psi]\!]$;
$X_{old} := S$;
while $X \neq X_{old}$
begin
$X_{old} := X$;
$X := X \cup ([\![\varphi]\!] \cap \textsc{Pre}(X))$
end
return X

$\}$

Summary

- CTL Model Checking: General Ideas.
- CTL Model Checking: The Labeling Algorithm.
- Labeling Algorithm in Details.
- CTL Model Checking: Theoretical Issues.

Correctness and Termination

- The Labeling Algorithm proceeds recursively on the structure of φ
- For the Boolean and **EX** operators, termination and correctness follow from semantics
- Need to prove correctness and termination for **EG** and **EU** operators

Monotone Functions and Fixpoints

Definition (Monotone Function)

Let S be a set and $F : 2^S \to 2^S$ a function

1. F is *monotone* if $X \subseteq Y$ implies $F(X) \subseteq F(Y)$
2. A set $X \subseteq S$ s.t. $F(X) = X$ of S is called a *fixpoint of* F
3. $X \subseteq S$ is a *least fixpoint* (LFP) of F, written $X = \mu X.F(X)$, if, for every other fixpoint Y of F, $X \subseteq Y$
4. $X \subseteq S$ is a *greatest fixpoint* (GFP) of F, written $X = \nu X.F(X)$, if, for every other fixpoint Y of F, $Y \subseteq X$

Example. Let $S = \{s_0, s_1\}$ and $F(X) = X \cup \{s_0\}$.

Tarski-Knaster Theorem

Notation: $F^i(X)$ means applying F i-times, i.e., $F(F(\ldots F(X)\ldots))$.

Theorem (Tarski-Knaster '55)

If $F : 2^S \to 2^S$ is a monotone function, over finite S s.t. $|S| = n$, then:

1. $\mu X.F(X) \equiv F^n(\emptyset)$
2. $\nu X.F(X) \equiv F^n(S)$

Correctness and Termination: **EG** Case

The function $\textsc{Label\_EG}$ computes:

$$[\![\mathbf{EG}\varphi]\!] = [\![\varphi]\!] \cap \textsc{Pre}([\![\mathbf{EG}\varphi]\!])$$

applying the semantic equivalence:

$$\mathbf{EG}\varphi \equiv \varphi \wedge \mathbf{EX}(\mathbf{EG}\varphi)$$

$[\![\mathbf{EG}\varphi]\!]$ is the fixpoint of the function:

$$F(X) = [\![\varphi]\!] \cap \textsc{Pre}(X)$$

Correctness and Termination: **EG** Case

Theorem

Let $F(X) = [\![\varphi]\!] \cap \textsc{Pre}(X)$, *with* $X \subseteq S$ *and* $|S| = n$. *Then:*

1. F *is monotone;*
2. $[\![\mathbf{EG}\varphi]\!]$ *is the greatest fixpoint of* F.

Correctness and Termination: **EU** Case

The function LABEL_EU computes:

$$[\![(\varphi \mathbf{EU} \psi)]\!] = [\![\psi]\!] \cup ([\![\varphi]\!] \cap \mathrm{PRE}([\![(\varphi \mathbf{EU} \psi)]\!]))$$

applying the semantic equivalence:

$$(\varphi \mathbf{EU} \psi) \equiv \psi \vee (\varphi \wedge \mathbf{EX}(\varphi \mathbf{EU} \psi))$$

$[\![(\varphi \mathbf{EU} \psi)]\!]$ is the fixpoint of the function:

$$F(X) = [\![\psi]\!] \cup ([\![\varphi]\!] \cap \mathrm{PRE}(X))$$

Correctness and Termination: **EU** Case

Theorem

Let $F(X) = [\![\psi]\!] \cup ([\![\varphi]\!] \cap \textsc{Pre}(X))$, *with* $X \subseteq S$ *and* $|S| = n$. *Then:*

1. F *is monotone;*
2. $[\![(\varphi \mathbf{EU} \psi)]\!]$ *is the least fixpoint of* F.

Complexity of CTL Model Checking

- The Labeling Algorithm operates in time: $O(|\varphi| \cdot |S| \cdot (|S| + |\delta|))$
- Since $|\delta| = O(|S|^2)$, we have:

$$O(|\varphi| \cdot |S|^3)$$

- Can be further optimized to:

$$O(|\varphi| \cdot |S|^2)$$

Theorem

CTL Model Checking can be solved in polynomial time wrt $|\varphi| + |S|$

References I

Edmund M Clarke, Orna Grumberg, and Doron A. Peled.

Model checking.

MIT Press, London, Cambridge, 1999.