# COMP424/524-06A
# Topics in Software Engineering

Part I – *Model Checking Algorithms*

## 16. Model Checking CTL

*Robi Malik*

THE UNIVERSITY OF WAIKATO *Te Whare Wānanga o Waikato*
DEPARTMENT OF COMPUTER SCIENCE
TARI ROROHIKO

## Checking CTL Formulas

***Given:***

- Kripke-Structure $K$
- CTL Formula $\varphi$

***Task:***

- Identify the states of $K$ where $\varphi$ is true.

$p$

$\neg\varphi \quad \varphi \wedge \psi \quad \ldots$

$\mathbf{AX}\,\varphi \quad \mathbf{EX}\,\varphi$

$\mathbf{AG}\,\varphi \quad \mathbf{EG}\,\varphi$

$\mathbf{AF}\,p \quad \mathbf{EF}\,\varphi$

$\mathbf{A}(\varphi\,\mathbf{U}\,\psi)$

$\mathbf{E}(\varphi\,\mathbf{U}\,\psi)$



## The Marking Algorithm

**Idea:**

Given a CTL formula, e.g.,

$$\mathbf{AG}\,(L_2 \Rightarrow \mathbf{AF}(\neg L_1 \wedge \neg L_2))$$

mark each state of $K$ with all subformulas that are satisfied in that state.

Start with the innermost subformulas.



## Marking Algorithm (1)

**Marking for propositions**

- Given by Kripke-Structure

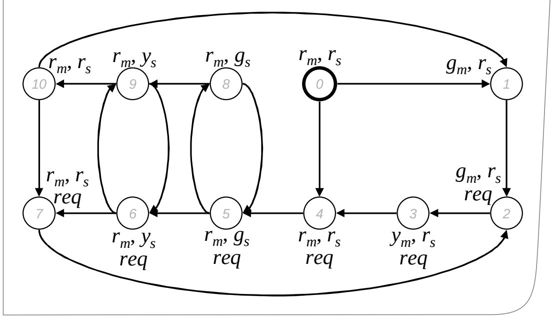
$\neg \quad \wedge \quad \vee \quad \Rightarrow \quad \Leftrightarrow$

**Marking for propositional formulas**

1. Apply the algorithm to all subformulas
2. Evaluate the connectives on each state, e.g., all states not marked $\varphi$ are marked $\neg\varphi$.



## Example: $\neg(req \Rightarrow g_s)$






## Marking Algorithm (2)

**Marking for $\mathbf{EX}\,\varphi$**

1. Apply marking algorithm to $\varphi$.
2. Mark states $\mathbf{EX}\,\varphi$, if they have at least one successor marked $\varphi$.

**Marking for $\mathbf{AX}\,\varphi$**

1. Apply marking algorithm to $\varphi$.
2. Mark states $\mathbf{AX}\,\varphi$, if all their successors are marked $\varphi$.

## **Marking Algorithm** (3)

**<u>Marking for EF φ</u>**

1. Apply marking algorithm to φ.
2. Mark states **EF** φ,
   if they are marked φ.
3. Mark states **EF** φ, if they have at least one successor already marked **EF** φ.
4. Repeat step 3.



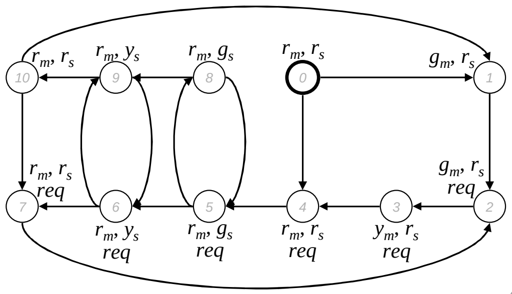
## **Marking Algorithm** (4)

**<u>Marking for AF φ</u>**

1. Apply marking algorithm to φ.
2. Mark states **AF** φ,
   if they are marked φ.
3. Mark states **AF** φ, if all their successors are already marked **AF** φ.
4. Repeat step 3.



## **Example:** $req \Rightarrow$ **AF** $g_s$





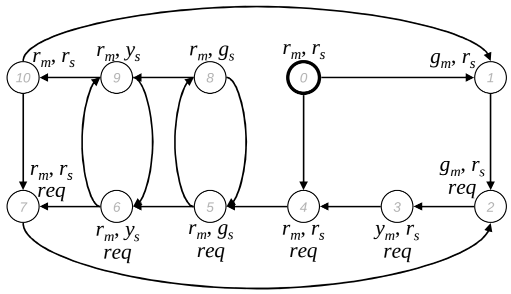

## **Marking Algorithm** (4)

**<u>Marking for A(φ U ψ)</u>**

1. Apply marking algorithm to φ and ψ.
2. Mark states **A**(φ **U** ψ),
   if they are marked ψ.
3. Mark states **A**(φ **U** ψ),
   if they are marked φ and all their successors are marked **A**(φ **U** ψ).
4. Repeat step 3.



## **Example: A** ($g_m$ **U** $req$)






## **Other Operators?**

*Use equivalences …*

$$\mathbf{AX}\ \varphi \equiv \neg\mathbf{EX}\ \neg\varphi$$
$$\mathbf{AG}\ \varphi \equiv \neg\mathbf{EF}\ \neg\varphi$$
$$\mathbf{EG}\ \varphi \equiv \neg\mathbf{AF}\ \neg\varphi$$
$$\mathbf{AF}\ \varphi \equiv \mathbf{A}\ (\text{true}\ \mathbf{U}\ \varphi)$$
$$\mathbf{EF}\ \varphi \equiv \mathbf{E}\ (\text{true}\ \mathbf{U}\ \varphi)$$

*… or try direct algorithms.*

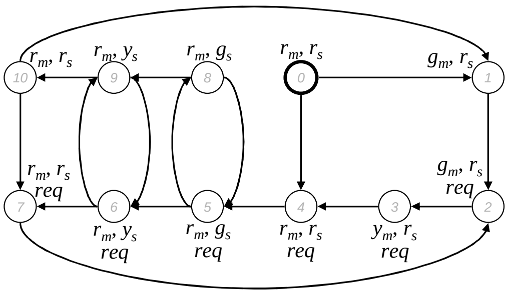
## Marking Algorithm (5)

**Marking for AG φ**

1. Apply marking algorithm to φ.
2. Mark states **AG** φ, if they are marked φ.
3. *Unmark* states **AG** φ, if they have a successor that is *not* marked **AG** φ.
4. Repeat step 3.



## Example: AG ¬($req \Rightarrow g_s$)






## Another Way of Doing It

Note:

**EF** φ ≡ φ ∨ **EX** **EF** φ

To solve this equation calculate …

$$X_0 = \text{false}$$
$$X_1 = \varphi \vee \mathbf{EX}\, X_0$$
$$X_2 = \varphi \vee \mathbf{EX}\, X_1$$
$$\ldots$$
$$X_n = \varphi \vee \mathbf{EX}\, X_n$$

Terminates at ***Least Fixed Point***



## And Similarly …

Note:

**AG** φ ≡ φ ∧ **AX** **AG** φ

To solve this equation calculate …

$$X_0 = \text{true}$$
$$X_1 = \varphi \wedge \mathbf{AX}\, X_0$$
$$X_2 = \varphi \wedge \mathbf{AX}\, X_1$$
$$\ldots$$
$$X_n = \varphi \wedge \mathbf{AX}\, X_n$$

***Greatest Fixed Point***



## Fixed Point Characterisations

$$\mathbf{AG}\,\varphi \equiv \varphi \wedge \mathbf{AX}\,\mathbf{AG}\,\varphi \equiv \text{gfp}\ \lambda X.(\varphi \wedge \mathbf{AX}\ X)$$
$$\mathbf{AF}\,\varphi \equiv \varphi \vee \mathbf{AX}\,\mathbf{AF}\,\varphi \equiv \text{lfp}\ \lambda X.(\varphi \vee \mathbf{AX}\ X)$$
$$\mathbf{EG}\,\varphi \equiv \varphi \wedge \mathbf{EX}\,\mathbf{EG}\,\varphi \equiv \text{gfp}\ \lambda X.(\varphi \wedge \mathbf{EX}\ X)$$
$$\mathbf{EF}\,\varphi \equiv \varphi \wedge \mathbf{EX}\,\mathbf{EF}\,\varphi \equiv \text{lfp}\ \lambda X.(\varphi \vee \mathbf{EX}\ X)$$
$$\mathbf{A}(\varphi\ \mathbf{U}\ \psi) \equiv \psi \vee (\varphi \wedge \mathbf{AX}\ \mathbf{A}\,(\varphi\ \mathbf{U}\ \psi))$$
$$\equiv \text{lfp}\ \lambda X.(\psi \vee (\varphi \wedge \mathbf{AX}\ X))$$
$$\mathbf{E}(\varphi\ \mathbf{U}\ \psi) \equiv \psi \vee (\varphi \wedge \mathbf{EX}\ \mathbf{E}\,(\varphi\ \mathbf{U}\ \psi))$$
$$\equiv \text{lfp}\ \lambda X.(\psi \vee (\varphi \wedge \mathbf{EX}\ X))$$



## Reading

Bérard et. al.:
Chapter 3 – Model Checking