Unit-10: Algorithms for CTL

B. Srivathsan

Chennai Mathematical Institute

NPTEL-course

July - November 2015

Module 3:
Final algorithm

CTL model-checking problem

Given transition system M and a CTL formula ϕ, find all states of M that satisfy ϕ

CTL model-checking problem

Given transition system M and a CTL formula ϕ, find all states of M that satisfy ϕ

- **Module 1:** Every CTL formula can be written using **EX, EU, EG**
- **Module 2:** Labelling algorithms for **EX, EU, EG**

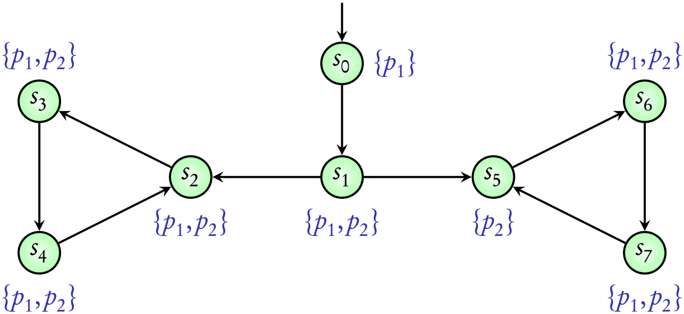
Coming next: Generic algorithm for a CTL formula

State formulae

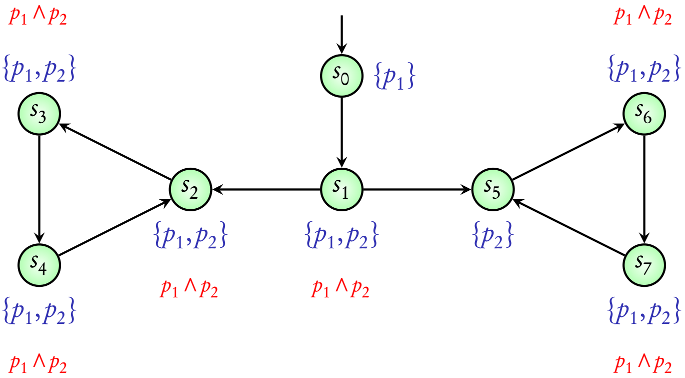
$$\phi := \text{true} \mid p_i \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid E\,X\,\phi \mid E\,(\phi_1\, U\, \phi_2) \mid E\,G\,\phi$$

$p_i \in AP$ ϕ, ϕ_1, ϕ_2 : State formulae

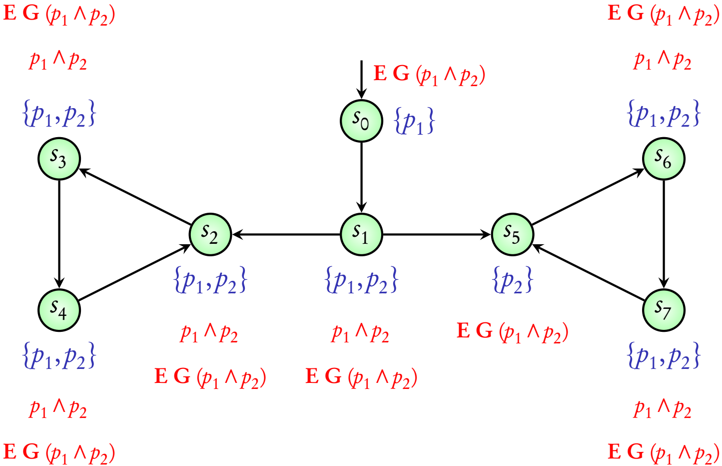
$\mathbf{E\,X\,E\,G}\,(p_1 \wedge p_2)$



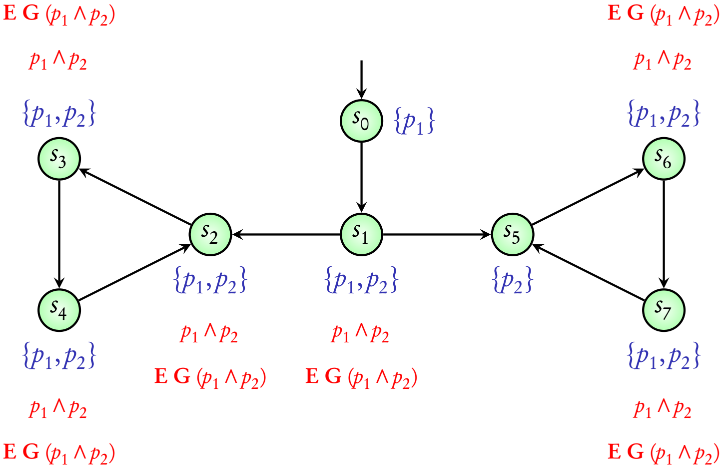
$\mathbf{E\,X\,E\,G}\,(p_1 \wedge p_2)$



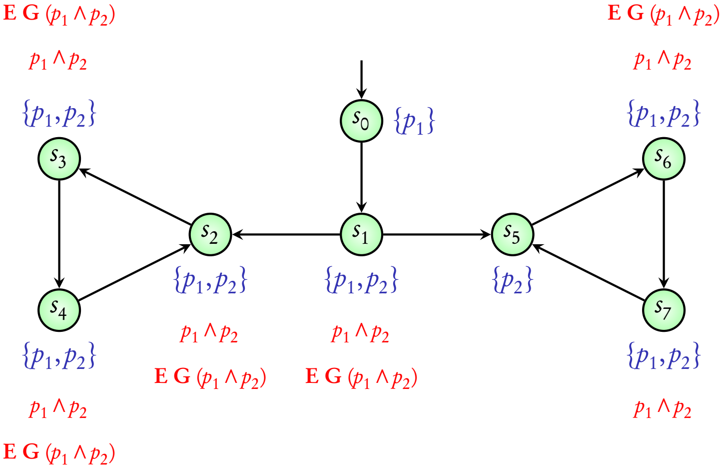
$\mathbf{E\,X\,E\,G}\,(p_1 \wedge p_2)$



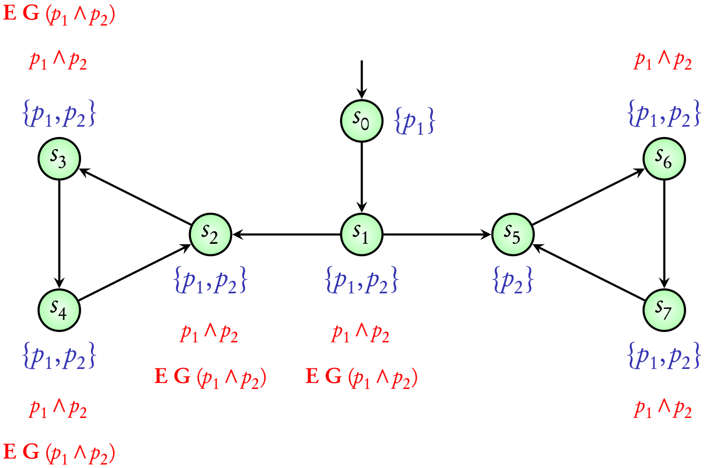
$\mathbf{E\,X\,E\,G}\,(p_1 \wedge p_2)$



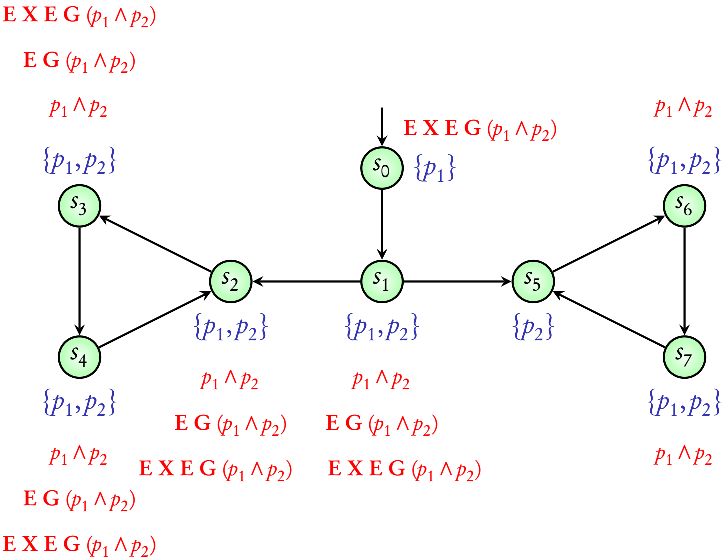
$\mathbf{E\,X\,E\,G}\,(p_1 \wedge p_2)$



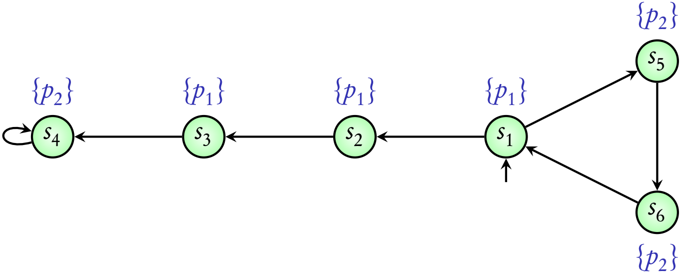
$\mathbf{E\,X\,E\,G}\,(p_1 \wedge p_2)$



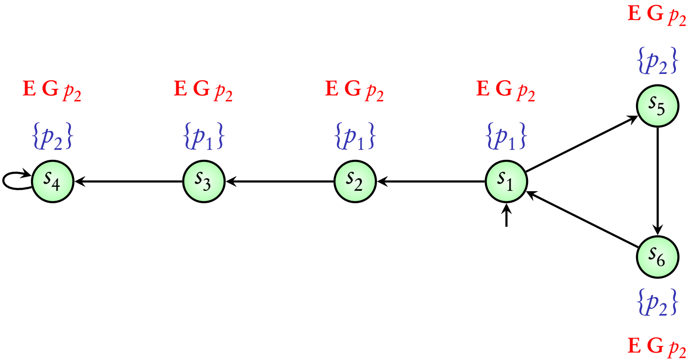
$\mathbf{E\,X\,E\,G}\,(p_1 \wedge p_2)$



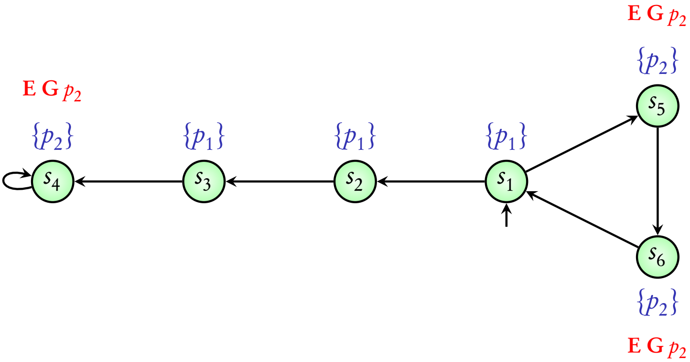
$\mathbf{E}\, p_1 \,\mathbf{U}\, (\mathbf{E}\,\mathbf{G}\, p_2)$



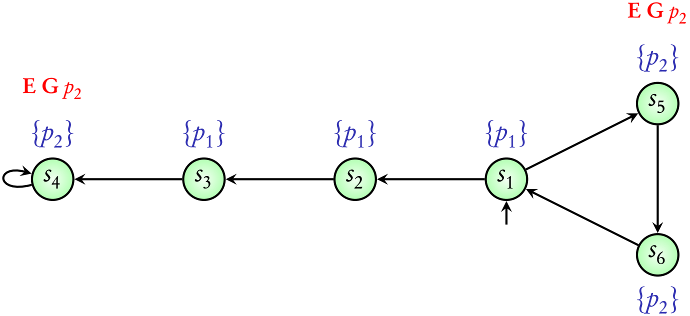
$\mathbf{E}\, p_1\, \mathbf{U}\, (\mathbf{E}\, \mathbf{G}\, p_2)$



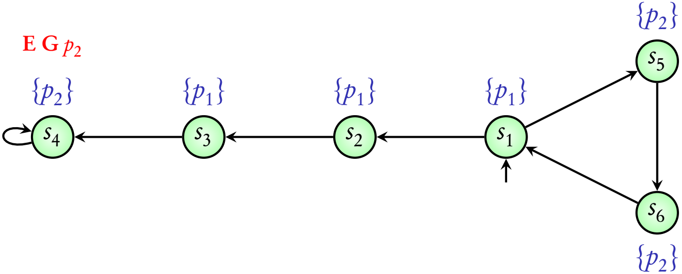
$\mathbf{E}\, p_1\, \mathbf{U}\, (\mathbf{E}\, \mathbf{G}\, p_2)$



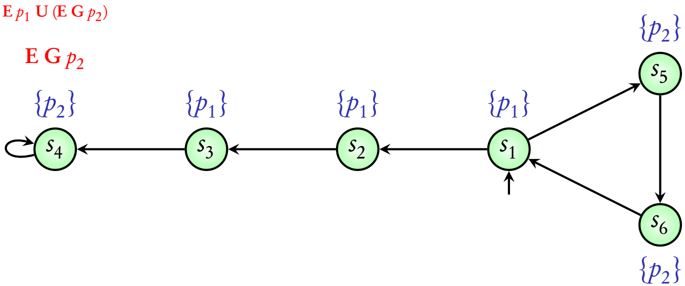
$\mathbf{E}\, p_1\, \mathbf{U}\, (\mathbf{E}\, \mathbf{G}\, p_2)$



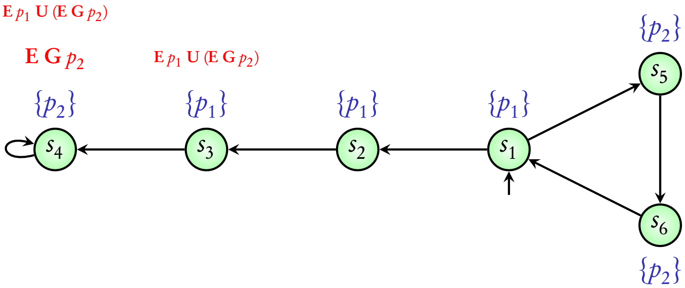
$\mathbf{E}\, p_1\, \mathbf{U}\, (\mathbf{E}\, \mathbf{G}\, p_2)$



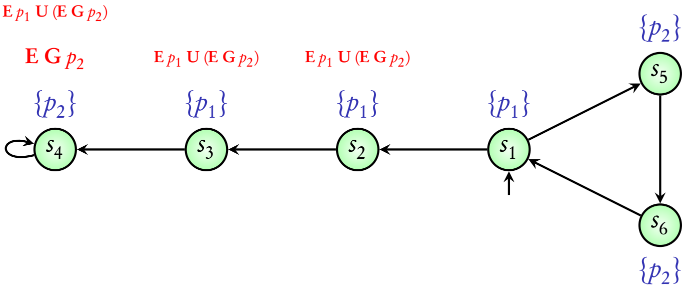
$\mathbf{E}\, p_1\, \mathbf{U}\, (\mathbf{E}\, \mathbf{G}\, p_2)$



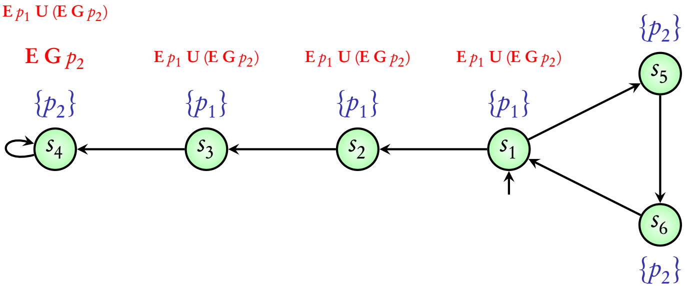
$\mathbf{E}\, p_1\, \mathbf{U}\, (\mathbf{E}\, \mathbf{G}\, p_2)$



$\mathbf{E}\, p_1\, \mathbf{U}\, (\mathbf{E}\, \mathbf{G}\, p_2)$



$\mathbf{E}\, p_1\, \mathbf{U}\, (\mathbf{E}\, \mathbf{G}\, p_2)$



function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

end function

function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

case

end case
end function

function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

case

ϕ is *true* : **return** S

end case
end function

function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

case

ϕ is *true* : **return** S

ϕ is p_i : **return** {states containing p_i }

end case
end function

function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

case

ϕ is *true* : **return** S

ϕ is p_i : **return** {states containing p_i }

ϕ is $\phi_1 \wedge \phi_2$: **return** SAT(ϕ_1) $\cap$ SAT(ϕ_2)

end case

end function

function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

case

ϕ is *true* : **return** S

ϕ is p_i : **return** {states containing p_i }

ϕ is $\phi_1 \wedge \phi_2$: **return** SAT(ϕ_1) ∩ SAT(ϕ_2)

ϕ is $\neg\phi_1$: **return** $S -$ SAT(ϕ_1)

end case

end function

function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

case

ϕ is *true* : **return** S

ϕ is p_i : **return** {states containing p_i }

ϕ is $\phi_1 \wedge \phi_2$: **return** SAT(ϕ_1) $\cap$ SAT(ϕ_2)

ϕ is $\neg\phi_1$: **return** S – SAT(ϕ_1)

ϕ is **E X** ϕ_1 : **return** $\text{SAT}_{\text{EX}}(\phi_1)$ /* procedure seen in Module 2 */

end case

end function

function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

case

ϕ is *true* : **return** S

ϕ is p_i : **return** {states containing p_i }

ϕ is $\phi_1 \wedge \phi_2$: **return** SAT(ϕ_1) $\cap$ SAT(ϕ_2)

ϕ is $\neg\phi_1$: **return** S – SAT(ϕ_1)

ϕ is **E X** ϕ_1 : **return** $\text{SAT}_{\text{EX}}(\phi_1)$ /* procedure seen in Module 2 */

ϕ is **E** (ϕ_1 **U** ϕ_2) : **return** $\text{SAT}_{\text{EU}}(\phi_1, \phi_2)$ /* procedure seen in Module 2 */

end case

end function

function SAT(ϕ)

/* Input: **Transition system** M with state set S, **CTL formula** ϕ in ENF */

/* Output: Set of states satisfying ϕ */

case

ϕ is *true* : **return** S

ϕ is p_i : **return** {states containing p_i }

ϕ is $\phi_1 \wedge \phi_2$: **return** SAT(ϕ_1) $\cap$ SAT(ϕ_2)

ϕ is $\neg\phi_1$: **return** $S -$ SAT(ϕ_1)

ϕ is **E X** ϕ_1 : **return** $\text{SAT}_{\text{EX}}(\phi_1)$ /* procedure seen in Module 2 */

ϕ is **E** (ϕ_1 **U** ϕ_2) : **return** $\text{SAT}_{\text{EU}}(\phi_1, \phi_2)$ /* procedure seen in Module 2 */

ϕ is **E G** ϕ_1 : **return** $\text{SAT}_{\text{EG}}(\phi_1)$ /* procedure seen in Module 2 */

end case

end function

CTL model-checking algorithm

Reference: Logic in Computer Science, *by Huth and Ryan* - Section 3.6.1