— Advanced Logic —

Linear Temporal Logic
Computation Tree Logic

Daniel Gebler

VU University Amsterdam

March 11, 2013

Overview

Linear temporal logic (LTL):

- describes properties of paths (individual executions)
- no modalities to reason about branching

Computation tree logic (CTL):

- is a branching-time logic
- time has a tree structure (multiple possible futures)
- has modalities for reasoning about the branching structure

Linear Temporal Logic (LTL)

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \mathsf{U} \phi \mid \mathsf{X}\, \phi$$

where $p \in \Omega$

LTL formulas have meaning on individual computation paths:

- let $\pi = s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$ a path; write π^i for $s_i \rightarrow s_{i+1} \rightarrow \ldots$

The path π satisfies ϕ, $\pi \models \phi$, is defined by:

1. $\pi \models p$ iff $s_1 \in V(p)$
2. $\pi \models \top$; $\pi \models \neg\phi$ iff $\pi \not\models \phi$; $\pi \models \phi_1 \wedge \phi_2$ iff $\pi \models \phi_1$ and $\pi \models \phi_2$

Linear Temporal Logic (LTL)

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underbrace{\mathsf{U}}_{\text{until}} \phi \mid \mathsf{X}\, \phi$$

where $p \in \Omega$

LTL formulas have meaning on individual computation paths:

- let $\pi = s_1 \to s_2 \to s_3 \to \ldots$ a path; write π^i for $s_i \to s_{i+1} \to \ldots$

The path π satisfies ϕ, $\pi \models \phi$, is defined by:

1. $\pi \models p$ iff $s_1 \in V(p)$
2. $\pi \models \top$; $\quad \pi \models \neg\phi$ iff $\pi \not\models \phi$; $\quad \pi \models \phi_1 \wedge \phi_2$ iff $\pi \models \phi_1$ and $\pi \models \phi_2$

Linear Temporal Logic (LTL)

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \; \underset{\text{until}}{\mathsf{U}} \; \phi \mid \overset{\text{next}}{\mathsf{X}} \; \phi$$

where $p \in \Omega$

LTL formulas have meaning on individual computation paths:

- let $\pi = s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \ldots$ a path; write π^i for $s_i \rightarrow s_{i+1} \rightarrow \ldots$

The path π satisfies ϕ, $\pi \models \phi$, is defined by:

1. $\pi \models p$ iff $s_1 \in V(p)$
2. $\pi \models \top$; $\quad \pi \models \neg\phi$ iff $\pi \not\models \phi$; $\quad \pi \models \phi_1 \wedge \phi_2$ iff $\pi \models \phi_1$ and $\pi \models \phi_2$

Linear Temporal Logic (LTL)

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{until}}{\mathsf{U}} \phi \mid \overset{\text{next}}{\mathsf{X}}\ \phi$$

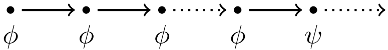
where $p \in \Omega$

LTL formulas have meaning on individual computation paths:

- let $\pi = s_1 \to s_2 \to s_3 \to \ldots$ a path; write π^i for $s_i \to s_{i+1} \to \ldots$

The path π satisfies ϕ, $\pi \models \phi$, is defined by:

1. $\pi \models p$ iff $s_1 \in V(p)$
2. $\pi \models \top$; $\pi \models \neg\phi$ iff $\pi \not\models \phi$; $\pi \models \phi_1 \wedge \phi_2$ iff $\pi \models \phi_1$ and $\pi \models \phi_2$
3. $\pi \models \phi \mathsf{U} \psi$ (ϕ is true until ψ is true)



formally: for some $i \geq 1$, $\pi^i \models \psi$ and for all $j < i$, $\pi^j \models \phi$

Linear Temporal Logic (LTL)

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{until}}{\mathsf{U}} \phi \mid \overset{\text{next}}{\mathsf{X}}\, \phi$$

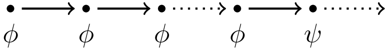
where $p \in \Omega$

LTL formulas have meaning on individual computation paths:

- let $\pi = s_1 \to s_2 \to s_3 \to \dots$ a path; write π^i for $s_i \to s_{i+1} \to \dots$

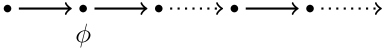
The path π satisfies ϕ, $\pi \models \phi$, is defined by:

1. $\pi \models p$ iff $s_1 \in V(p)$
2. $\pi \models \top$; $\pi \models \neg\phi$ iff $\pi \not\models \phi$; $\pi \models \phi_1 \wedge \phi_2$ iff $\pi \models \phi_1$ and $\pi \models \phi_2$
3. $\pi \models \phi \,\mathsf{U}\, \psi$ (ϕ is true until ψ is true)



formally: for some $i \geq 1$, $\pi^i \models \psi$ and for all $j < i$, $\pi^j \models \phi$

4. $\pi \models \mathsf{X}\, \phi$ (ϕ is true in the next moment in time)



formally: $\pi^2 \models \phi$

LTL: Extended

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{until}}{\mathsf{U}} \phi \mid \overset{\text{next}}{\mathsf{X}} \phi \mid \mathsf{F}\ \phi \mid \mathsf{G}\ \phi$$

where $p \in \Omega$

LTL: Extended

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{until}}{\mathsf{U}} \phi \mid \overset{\text{next}}{\mathsf{X}}\ \phi \mid \underset{\text{finally}}{\mathsf{F}}\ \phi \mid \mathsf{G}\ \phi$$

where $p \in \Omega$

LTL: Extended

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{until}}{\mathsf{U}} \phi \mid \overset{\text{next}}{\mathsf{X}} \phi \mid \underset{\text{finally}}{\mathsf{F}} \phi \mid \overset{\text{globally}}{\mathsf{G}} \phi$$

where $p \in \Omega$

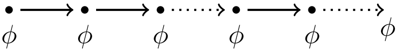
LTL: Extended

Linear temporal logic (LTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{until}}{\mathsf{U}} \phi \mid \overset{\text{next}}{\mathsf{X}}\, \phi \mid \underset{\text{finally}}{\mathsf{F}}\, \phi \mid \overset{\text{globally}}{\mathsf{G}}\, \phi$$

where $p \in \Omega$

1. $\pi \models \mathsf{G}\ \phi$ iff for all $i \geq 1$, $\pi^i \models \phi$



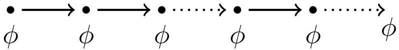
LTL: Extended

Linear temporal logic (LTL) is defined by:

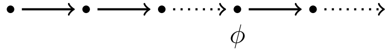
$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{until}}{\mathsf{U}} \phi \mid \overset{\text{next}}{\mathsf{X}} \phi \mid \underset{\text{finally}}{\mathsf{F}} \phi \mid \overset{\text{globally}}{\mathsf{G}} \phi$$

where $p \in \Omega$

1. $\pi \models \mathsf{G}\ \phi$ iff for all $i \geq 1$, $\pi^i \models \phi$



2. $\pi \models \mathsf{F}\ \phi$ iff for some $i \geq 1$, $\pi^i \models \phi$



LTL: Extended

Linear temporal logic (LTL) is defined by:

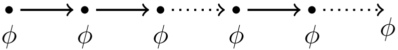
next globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ U } \phi \mid \text{X } \phi \mid \text{F } \phi \mid \text{G } \phi$$

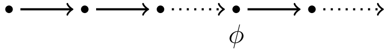
until finally

where $p \in \Omega$

1. $\pi \models \text{G } \phi$ iff for all $i \geq 1$, $\pi^i \models \phi$



2. $\pi \models \text{F } \phi$ iff for some $i \geq 1$, $\pi^i \models \phi$



The modalities F and G can be defined:

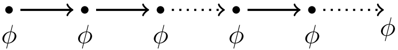
LTL: Extended

Linear temporal logic (LTL) is defined by:

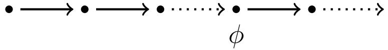
$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{until}}{\mathsf{U}} \phi \mid \overset{\text{next}}{\mathsf{X}} \phi \mid \underset{\text{finally}}{\mathsf{F}} \phi \mid \overset{\text{globally}}{\mathsf{G}} \phi$$

where $p \in \Omega$

1. $\pi \models \mathsf{G}\ \phi$ iff for all $i \geq 1$, $\pi^i \models \phi$



2. $\pi \models \mathsf{F}\ \phi$ iff for some $i \geq 1$, $\pi^i \models \phi$



The modalities F and G can be defined:

$$\mathsf{F}\ = \top\ \mathsf{U}\ \phi$$

$$\mathsf{G}\ \phi = \neg\mathsf{F}\ \neg\phi = \neg(\top\ \mathsf{U}\ \neg\phi)$$

LTL: Extended

Linear temporal logic (LTL) is defined by:

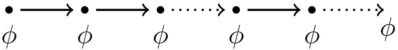
next globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \,\mathsf{U}\, \phi \mid \mathsf{X}\, \phi \mid \mathsf{F}\, \phi \mid \mathsf{G}\, \phi$$

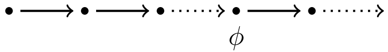
until finally

where $p \in \Omega$

1. $\pi \models \mathsf{G}\ \phi$ iff for all $i \geq 1$, $\pi^i \models \phi$



2. $\pi \models \mathsf{F}\ \phi$ iff for some $i \geq 1$, $\pi^i \models \phi$



The modalities F and G can be defined:

$$\mathsf{F} \ = \top \,\mathsf{U}\, \phi$$

$$\mathsf{G}\ \phi = \neg\mathsf{F}\ \neg\phi = \neg(\top \,\mathsf{U}\, \neg\phi)$$

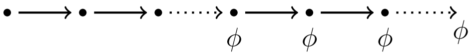
Binding strength: $\neg$, X , F , G stronger than U than $\wedge, \vee$ than $\rightarrow, \leftrightarrow$

LTL: Examples

- F G ϕ:

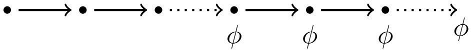
LTL: Examples

- F G ϕ: from some point on, ϕ holds forever



LTL: Examples

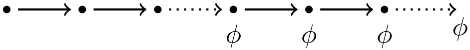
- F G ϕ : from some point on, ϕ holds forever



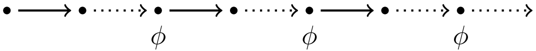
- G F ϕ :

LTL: Examples

- F G ϕ : from some point on, ϕ holds forever



- G F ϕ : always eventually ϕ (in every suffix, at some point ϕ holds)

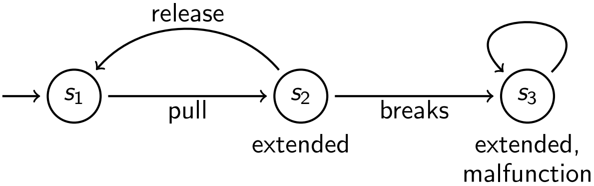


LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.

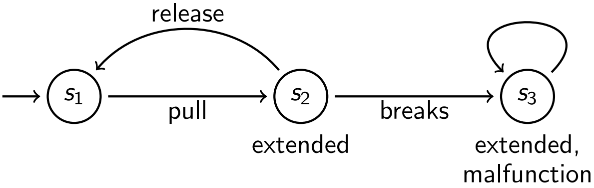
LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$? \models \mathrm{X}$ extended

$? \models \mathrm{X}\ \mathrm{X}$ extended

$? \models \mathrm{F}$ extended

$? \models \mathrm{G}$ extended

$? \models \mathrm{G}\ \mathrm{F}$ extended

$? \models \mathrm{F}\ \mathrm{G}$ extended

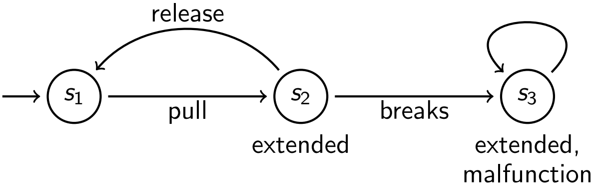
$? \models \neg\, \mathrm{F}\ \mathrm{G}$ extended

$? \models \mathrm{G}\ (\neg \text{extended} \rightarrow \mathrm{X}\ \text{extended})$

$? \models \mathrm{G}\ (\text{extended} \rightarrow \mathrm{X}\ \neg \text{extended})$

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



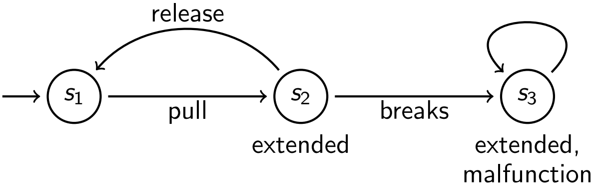
Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models$ X extended
? $\models$ X X extended
? $\models$ F extended
? $\models$ G extended
? $\models$ G F extended

? $\models$ F G extended
? $\models$ ¬ F G extended
? $\models$ G (¬extended → X extended)
? $\models$ G (extended → X ¬extended)

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models \mathrm{X}$ extended

$\mathfrak{M}, s_2, s_3 \models \mathrm{X}\ \mathrm{X}$ extended

$? \models \mathrm{F}$ extended

$? \models \mathrm{G}$ extended

$? \models \mathrm{G}\ \mathrm{F}$ extended

$? \models \mathrm{F}\ \mathrm{G}$ extended

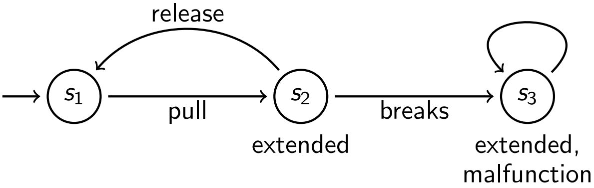
$? \models \neg\, \mathrm{F}\ \mathrm{G}$ extended

$? \models \mathrm{G}\ (\neg \text{extended} \rightarrow \mathrm{X}\ \text{extended})$

$? \models \mathrm{G}\ (\text{extended} \rightarrow \mathrm{X}\ \neg\text{extended})$

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models \mathrm{X}$ extended

$\mathfrak{M}, s_2, s_3 \models \mathrm{X}\ \mathrm{X}$ extended

$\mathfrak{M}, s_1, s_2, s_3 \models \mathrm{F}$ extended

$? \models \mathrm{G}$ extended

$? \models \mathrm{G}\ \mathrm{F}$ extended

$? \models \mathrm{F}\ \mathrm{G}$ extended

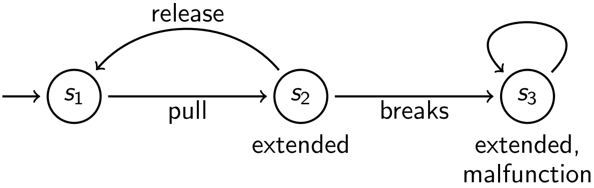
$? \models \neg\, \mathrm{F}\ \mathrm{G}$ extended

$? \models \mathrm{G}\ (\neg\text{extended} \rightarrow \mathrm{X}\ \text{extended})$

$? \models \mathrm{G}\ (\text{extended} \rightarrow \mathrm{X}\ \neg\text{extended})$

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models \mathsf{X}$ extended

$\mathfrak{M}, s_2, s_3 \models \mathsf{X}\ \mathsf{X}$ extended

$\mathfrak{M}, s_1, s_2, s_3 \models \mathsf{F}$ extended

$\mathfrak{M}, s_3 \models \mathsf{G}$ extended

$? \models \mathsf{G}\ \mathsf{F}$ extended

$? \models \mathsf{F}\ \mathsf{G}$ extended

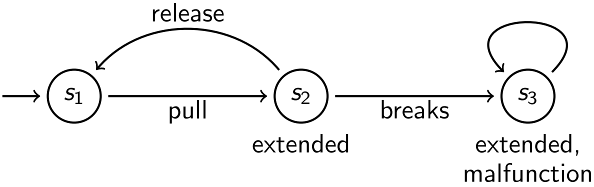
$? \models \neg\, \mathsf{F}\ \mathsf{G}$ extended

$? \models \mathsf{G}\ (\neg\text{extended} \rightarrow \mathsf{X}\ \text{extended})$

$? \models \mathsf{G}\ (\text{extended} \rightarrow \mathsf{X}\ \neg\text{extended})$

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models \text{X extended}$

$\mathfrak{M}, s_2, s_3 \models \text{X X extended}$

$\mathfrak{M}, s_1, s_2, s_3 \models \text{F extended}$

$\mathfrak{M}, s_3 \models \text{G extended}$

$\mathfrak{M}, s_1, s_2, s_3 \models \text{G F extended}$

$? \models \text{F G extended}$

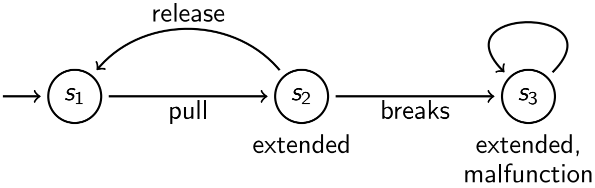
$? \models \neg\,\text{F G extended}$

$? \models \text{G } (\neg\text{extended} \rightarrow \text{X extended})$

$? \models \text{G } (\text{extended} \rightarrow \text{X } \neg\text{extended})$

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models \mathrm{X}$ extended

$\mathfrak{M}, s_2, s_3 \models \mathrm{X}\ \mathrm{X}$ extended

$\mathfrak{M}, s_1, s_2, s_3 \models \mathrm{F}$ extended

$\mathfrak{M}, s_3 \models \mathrm{G}$ extended

$\mathfrak{M}, s_1, s_2, s_3 \models \mathrm{G}\ \mathrm{F}$ extended

$\mathfrak{M}, s_3 \models \mathrm{F}\ \mathrm{G}$ extended

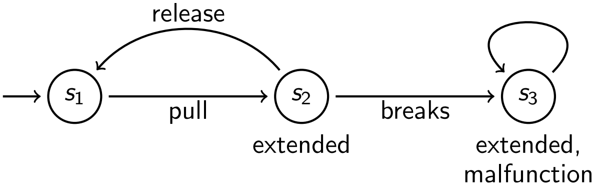
$? \models \neg\,\mathrm{F}\ \mathrm{G}$ extended

$? \models \mathrm{G}\ (\neg\text{extended} \rightarrow \mathrm{X}\ \text{extended})$

$? \models \mathrm{G}\ (\text{extended} \rightarrow \mathrm{X}\ \neg\text{extended})$

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models \mathsf{X}$ extended

$\mathfrak{M}, s_2, s_3 \models \mathsf{X}\ \mathsf{X}$ extended

$\mathfrak{M}, s_1, s_2, s_3 \models \mathsf{F}$ extended

$\mathfrak{M}, s_3 \models \mathsf{G}$ extended

$\mathfrak{M}, s_1, s_2, s_3 \models \mathsf{G}\ \mathsf{F}$ extended

$\mathfrak{M}, s_3 \models \mathsf{F}\ \mathsf{G}$ extended

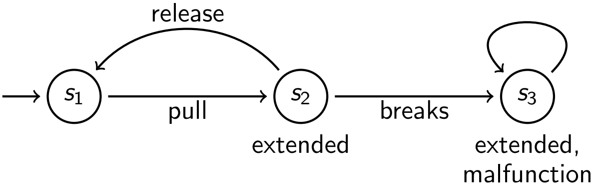
$\mathfrak{M}, s_1, s_2, s_3 \not\models \neg\, \mathsf{F}\ \mathsf{G}$ extended

$? \models \mathsf{G}\ (\neg\text{extended} \rightarrow \mathsf{X}\ \text{extended})$

$? \models \mathsf{G}\ (\text{extended} \rightarrow \mathsf{X}\ \neg\text{extended})$

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models \mathsf{X}\ \text{extended}$

$\mathfrak{M}, s_2, s_3 \models \mathsf{X}\ \mathsf{X}\ \text{extended}$

$\mathfrak{M}, s_1, s_2, s_3 \models \mathsf{F}\ \text{extended}$

$\mathfrak{M}, s_3 \models \mathsf{G}\ \text{extended}$

$\mathfrak{M}, s_1, s_2, s_3 \models \mathsf{G}\ \mathsf{F}\ \text{extended}$

$\mathfrak{M}, s_3 \models \mathsf{F}\ \mathsf{G}\ \text{extended}$

$\mathfrak{M}, s_1, s_2, s_3 \not\models \neg\, \mathsf{F}\ \mathsf{G}\ \text{extended}$

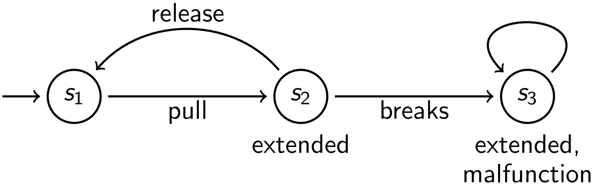
$? \models \mathsf{G}\ (\neg\text{extended} \to \mathsf{X}\ \text{extended})$

$? \models \mathsf{G}\ (\text{extended} \to \mathsf{X}\ \neg\text{extended})$

Note that: $\mathfrak{M} \not\models \mathsf{F}\ \mathsf{G}\ \text{extended}$ and $\mathfrak{M} \not\models \neg\, \mathsf{F}\ \mathsf{G}\ \text{extended}$!

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$\mathfrak{M}, s_1, s_3 \models$ X extended

$\mathfrak{M}, s_2, s_3 \models$ X X extended

$\mathfrak{M}, s_1, s_2, s_3 \models$ F extended

$\mathfrak{M}, s_3 \models$ G extended

$\mathfrak{M}, s_1, s_2, s_3 \models$ G F extended

$\mathfrak{M}, s_3 \models$ F G extended

$\mathfrak{M}, s_1, s_2, s_3 \not\models \neg$ F G extended

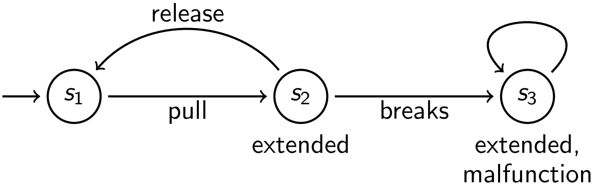
$\mathfrak{M} \models$ G ($\neg$extended $\rightarrow$ X extended)

? $\models$ G (extended $\rightarrow$ X $\neg$extended)

Note that: $\mathfrak{M} \not\models$ F G extended and $\mathfrak{M} \not\models \neg$ F G extended !

LTL: Models

$\mathfrak{M}, s \models \phi$ if ϕ is satisfied on every path starting at s.
$\mathfrak{M} \models \phi$ if ϕ is satisfied on every path starting from the initial state.



Which of the states satisfies the following?

$$\mathfrak{M}, s_1, s_3 \models \text{X extended}$$

$$\mathfrak{M}, s_2, s_3 \models \text{X X extended}$$

$$\mathfrak{M}, s_1, s_2, s_3 \models \text{F extended}$$

$$\mathfrak{M}, s_3 \models \text{G extended}$$

$$\mathfrak{M}, s_1, s_2, s_3 \models \text{G F extended}$$

$$\mathfrak{M}, s_3 \models \text{F G extended}$$

$$\mathfrak{M}, s_1, s_2, s_3 \not\models \neg\,\text{F G extended}$$

$$\mathfrak{M} \models \text{G } (\neg\text{extended} \rightarrow \text{X extended})$$

$$\mathfrak{M}, s_1, s_2, s_3 \not\models \text{G } (\text{extended} \rightarrow \text{X } \neg\text{extended})$$

Note that: $\mathfrak{M} \not\models \text{F G extended}$ and $\mathfrak{M} \not\models \neg\,\text{F G extended}$!

LTL: Equivalence of Formulas

LTL formulas ϕ and ψ are semantically equivalent, denoted by $\phi \equiv \psi$, if they are true for the same paths

LTL: Equivalence of Formulas

LTL formulas ϕ and ψ are semantically equivalent, denoted by $\phi \equiv \psi$, if they are true for the same paths

Which of the following are semantically equivalent?

$$\mathsf{X}\ (\phi \vee \psi) \equiv \mathsf{X}\ \phi \vee \mathsf{X}\ \psi$$

$$\mathsf{X}\ (\phi \wedge \psi) \equiv \mathsf{X}\ \phi \wedge \mathsf{X}\ \psi$$

$$\mathsf{F}\ (\phi \wedge \psi) \equiv \mathsf{F}\ \phi \wedge \mathsf{F}\ \psi$$

$$\mathsf{F}\ (\phi \vee \psi) \equiv \mathsf{F}\ \phi \vee \mathsf{F}\ \psi$$

$$\mathsf{G}\ (\phi \wedge \psi) \equiv \mathsf{G}\ \phi \wedge \mathsf{F}\ \psi$$

$$\mathsf{G}\ (\phi \vee \psi) \equiv \mathsf{G}\ \phi \vee \mathsf{F}\ \psi$$

$$\rho\ \mathsf{U}\ (\phi \vee \psi) \equiv (\rho\ \mathsf{U}\ \phi) \vee (\rho\ \mathsf{U}\ \psi)$$

$$\rho\ \mathsf{U}\ (\phi \wedge \psi) \equiv (\rho\ \mathsf{U}\ \phi) \wedge (\rho\ \mathsf{U}\ \psi)$$

$$\mathsf{F}\ \mathsf{F}\ \phi \equiv \mathsf{F}\ \phi$$

$$\mathsf{G}\ \mathsf{G}\ \phi \equiv \mathsf{G}\ \phi$$

$$\mathsf{F}\ \mathsf{G}\ \phi \equiv \mathsf{G}\ \mathsf{F}\ \phi$$

$$\neg \mathsf{F}\ \phi \equiv \mathsf{G}\ \neg \phi$$

$$\neg \mathsf{G}\ \phi \equiv \mathsf{F}\ \neg \phi$$

$$\mathsf{F}\ \phi \equiv \phi \vee \mathsf{X}\ (\mathsf{F}\ \phi)$$

$$\mathsf{G}\ \phi \equiv \phi \wedge \mathsf{X}\ (\mathsf{G}\ \phi)$$

$$\phi\ \mathsf{U}\ \psi \equiv \phi\ \mathsf{U}\ (\phi\ \mathsf{U}\ \psi)$$

LTL: Equivalence of Formulas

LTL formulas ϕ and ψ are semantically equivalent, denoted by $\phi \equiv \psi$, if they are true for the same paths

Which of the following are semantically equivalent?

$\mathsf{X}\ (\phi \vee \psi) \equiv \mathsf{X}\ \phi \vee \mathsf{X}\ \psi$

$\mathsf{X}\ (\phi \wedge \psi) \equiv \mathsf{X}\ \phi \wedge \mathsf{X}\ \psi$

~~$\mathsf{F}\ (\phi \wedge \psi) \equiv \mathsf{F}\ \phi \wedge \mathsf{F}\ \psi$~~

$\mathsf{F}\ (\phi \vee \psi) \equiv \mathsf{F}\ \phi \vee \mathsf{F}\ \psi$

$\mathsf{G}\ (\phi \wedge \psi) \equiv \mathsf{G}\ \phi \wedge \mathsf{F}\ \psi$

$\mathsf{G}\ (\phi \vee \psi) \equiv \mathsf{G}\ \phi \vee \mathsf{F}\ \psi$

$\rho\ \mathsf{U}\ (\phi \vee \psi) \equiv (\rho\ \mathsf{U}\ \phi) \vee (\rho\ \mathsf{U}\ \psi)$

$\rho\ \mathsf{U}\ (\phi \wedge \psi) \equiv (\rho\ \mathsf{U}\ \phi) \wedge (\rho\ \mathsf{U}\ \psi)$

$\mathsf{F}\ \mathsf{F}\ \phi \equiv \mathsf{F}\ \phi$

$\mathsf{G}\ \mathsf{G}\ \phi \equiv \mathsf{G}\ \phi$

$\mathsf{F}\ \mathsf{G}\ \phi \equiv \mathsf{G}\ \mathsf{F}\ \phi$

$\neg \mathsf{F}\ \phi \equiv \mathsf{G}\ \neg \phi$

$\neg \mathsf{G}\ \phi \equiv \mathsf{F}\ \neg \phi$

$\mathsf{F}\ \phi \equiv \phi \vee \mathsf{X}\ (\mathsf{F}\ \phi)$

$\mathsf{G}\ \phi \equiv \phi \wedge \mathsf{X}\ (\mathsf{G}\ \phi)$

$\phi\ \mathsf{U}\ \psi \equiv \phi\ \mathsf{U}\ (\phi\ \mathsf{U}\ \psi)$

LTL: Equivalence of Formulas

LTL formulas ϕ and ψ are semantically equivalent, denoted by $\phi \equiv \psi$, if they are true for the same paths

Which of the following are semantically equivalent?

$\mathsf{X}\,(\phi \vee \psi) \equiv \mathsf{X}\,\phi \vee \mathsf{X}\,\psi$

$\mathsf{X}\,(\phi \wedge \psi) \equiv \mathsf{X}\,\phi \wedge \mathsf{X}\,\psi$

~~$\mathsf{F}\,(\phi \wedge \psi) \equiv \mathsf{F}\,\phi \wedge \mathsf{F}\,\psi$~~

$\mathsf{F}\,(\phi \vee \psi) \equiv \mathsf{F}\,\phi \vee \mathsf{F}\,\psi$

$\mathsf{G}\,(\phi \wedge \psi) \equiv \mathsf{G}\,\phi \wedge \mathsf{F}\,\psi$

~~$\mathsf{G}\,(\phi \vee \psi) \equiv \mathsf{G}\,\phi \vee \mathsf{F}\,\psi$~~

$\rho\;\mathsf{U}\;(\phi \vee \psi) \equiv (\rho\;\mathsf{U}\;\phi) \vee (\rho\;\mathsf{U}\;\psi)$

$\rho\;\mathsf{U}\;(\phi \wedge \psi) \equiv (\rho\;\mathsf{U}\;\phi) \wedge (\rho\;\mathsf{U}\;\psi)$

$\mathsf{F}\,\mathsf{F}\,\phi \equiv \mathsf{F}\,\phi$

$\mathsf{G}\,\mathsf{G}\,\phi \equiv \mathsf{G}\,\phi$

$\mathsf{F}\,\mathsf{G}\,\phi \equiv \mathsf{G}\,\mathsf{F}\,\phi$

$\neg\mathsf{F}\,\phi \equiv \mathsf{G}\,\neg\phi$

$\neg\mathsf{G}\,\phi \equiv \mathsf{F}\,\neg\phi$

$\mathsf{F}\,\phi \equiv \phi \vee \mathsf{X}\,(\mathsf{F}\,\phi)$

$\mathsf{G}\,\phi \equiv \phi \wedge \mathsf{X}\,(\mathsf{G}\,\phi)$

$\phi\;\mathsf{U}\;\psi \equiv \phi\;\mathsf{U}\;(\phi\;\mathsf{U}\;\psi)$

LTL: Equivalence of Formulas

LTL formulas ϕ and ψ are semantically equivalent, denoted by $\phi \equiv \psi$, if they are true for the same paths

Which of the following are semantically equivalent?

$$\mathsf{X}\ (\phi \vee \psi) \equiv \mathsf{X}\ \phi \vee \mathsf{X}\ \psi$$

$$\mathsf{X}\ (\phi \wedge \psi) \equiv \mathsf{X}\ \phi \wedge \mathsf{X}\ \psi$$

~~$\mathsf{F}\ (\phi \wedge \psi) \equiv \mathsf{F}\ \phi \wedge \mathsf{F}\ \psi$~~

$$\mathsf{F}\ (\phi \vee \psi) \equiv \mathsf{F}\ \phi \vee \mathsf{F}\ \psi$$

$$\mathsf{G}\ (\phi \wedge \psi) \equiv \mathsf{G}\ \phi \wedge \mathsf{F}\ \psi$$

~~$\mathsf{G}\ (\phi \vee \psi) \equiv \mathsf{G}\ \phi \vee \mathsf{F}\ \psi$~~

$$\rho\ \mathsf{U}\ (\phi \vee \psi) \equiv (\rho\ \mathsf{U}\ \phi) \vee (\rho\ \mathsf{U}\ \psi)$$

~~$\rho\ \mathsf{U}\ (\phi \wedge \psi) \equiv (\rho\ \mathsf{U}\ \phi) \wedge (\rho\ \mathsf{U}\ \psi)$~~

$$\mathsf{F}\ \mathsf{F}\ \phi \equiv \mathsf{F}\ \phi$$

$$\mathsf{G}\ \mathsf{G}\ \phi \equiv \mathsf{G}\ \phi$$

$$\mathsf{F}\ \mathsf{G}\ \phi \equiv \mathsf{G}\ \mathsf{F}\ \phi$$

$$\neg \mathsf{F}\ \phi \equiv \mathsf{G}\ \neg \phi$$

$$\neg \mathsf{G}\ \phi \equiv \mathsf{F}\ \neg \phi$$

$$\mathsf{F}\ \phi \equiv \phi \vee \mathsf{X}\ (\mathsf{F}\ \phi)$$

$$\mathsf{G}\ \phi \equiv \phi \wedge \mathsf{X}\ (\mathsf{G}\ \phi)$$

$$\phi\ \mathsf{U}\ \psi \equiv \phi\ \mathsf{U}\ (\phi\ \mathsf{U}\ \psi)$$

LTL: Equivalence of Formulas

LTL formulas ϕ and ψ are semantically equivalent, denoted by $\phi \equiv \psi$, if they are true for the same paths

Which of the following are semantically equivalent?

$\mathsf{X}\ (\phi \lor \psi) \equiv \mathsf{X}\ \phi \lor \mathsf{X}\ \psi$

$\mathsf{X}\ (\phi \land \psi) \equiv \mathsf{X}\ \phi \land \mathsf{X}\ \psi$

~~$\mathsf{F}\ (\phi \land \psi) \equiv \mathsf{F}\ \phi \land \mathsf{F}\ \psi$~~

$\mathsf{F}\ (\phi \lor \psi) \equiv \mathsf{F}\ \phi \lor \mathsf{F}\ \psi$

$\mathsf{G}\ (\phi \land \psi) \equiv \mathsf{G}\ \phi \land \mathsf{F}\ \psi$

~~$\mathsf{G}\ (\phi \lor \psi) \equiv \mathsf{G}\ \phi \lor \mathsf{F}\ \psi$~~

$\rho\ \mathsf{U}\ (\phi \lor \psi) \equiv (\rho\ \mathsf{U}\ \phi) \lor (\rho\ \mathsf{U}\ \psi)$

~~$\rho\ \mathsf{U}\ (\phi \land \psi) \equiv (\rho\ \mathsf{U}\ \phi) \land (\rho\ \mathsf{U}\ \psi)$~~

$\mathsf{F}\ \mathsf{F}\ \phi \equiv \mathsf{F}\ \phi$

$\mathsf{G}\ \mathsf{G}\ \phi \equiv \mathsf{G}\ \phi$

~~$\mathsf{F}\ \mathsf{G}\ \phi \equiv \mathsf{G}\ \mathsf{F}\ \phi$~~

$\neg \mathsf{F}\ \phi \equiv \mathsf{G}\ \neg \phi$

$\neg \mathsf{G}\ \phi \equiv \mathsf{F}\ \neg \phi$

$\mathsf{F}\ \phi \equiv \phi \lor \mathsf{X}\ (\mathsf{F}\ \phi)$

$\mathsf{G}\ \phi \equiv \phi \land \mathsf{X}\ (\mathsf{G}\ \phi)$

$\phi\ \mathsf{U}\ \psi \equiv \phi\ \mathsf{U}\ (\phi\ \mathsf{U}\ \psi)$

Mutual Exclusion

- multiple processes
- a shared resource that can only be used by one process at a time

process Q

shared resource

process P

Mutual Exclusion

- multiple processes
- a shared resource that can only be used by one process at a time

Q

non critical

...

C_Q critical section

...

non critical

P

non critical

...

C_P critical section

...

non critical

To solve conflicts: processes agree on a negotiation protocol.

- mutual exclusion: never more than one process in the critical section

Mutual Exclusion

- multiple processes
- a shared resource that can only be used by one process at a time

Q
non critical
...
C_Q critical section
...
non critical

P
non critical
...
C_P critical section
...
non critical

To solve conflicts: processes agree on a negotiation protocol.

- mutual exclusion: never more than one process in the critical section

$$\mathsf{G}\ \neg(C_Q \wedge C_P)$$

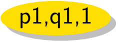
Mutual Exclusion: Attempt 1

- boolean variable *free* = 1

```
P
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

```
Q
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



p1,q1,1

Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

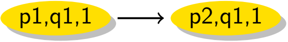
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

```
P
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

```
Q
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:

$$\text{p1,q1,1} \longrightarrow \text{p2,q1,1} \longrightarrow C_P\text{,q1,0}$$

Mutual Exclusion: Attempt 1

- boolean variable $free = 1$

```
P
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

```
Q
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:

$$\text{p1,q1,1} \longrightarrow \text{p2,q1,1} \longrightarrow C_P\text{,q1,0} \longrightarrow \text{p4,q1,0}$$

Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

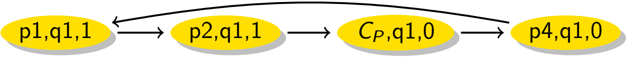
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

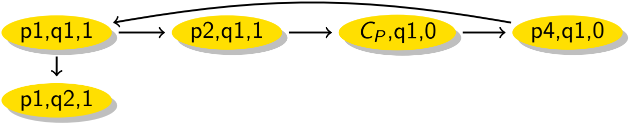
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

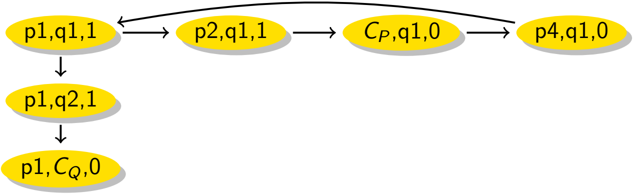
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



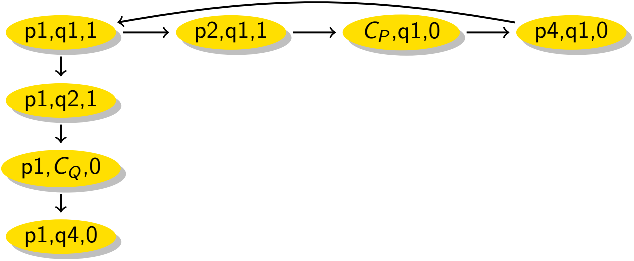
Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

```
P
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

```
Q
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

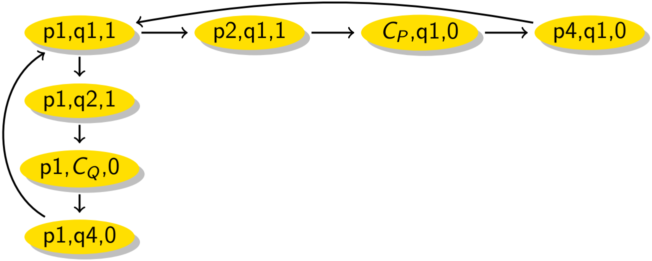
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

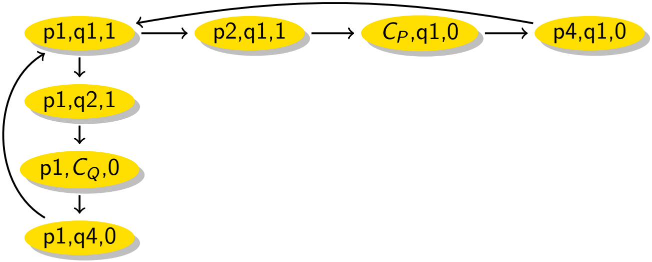
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

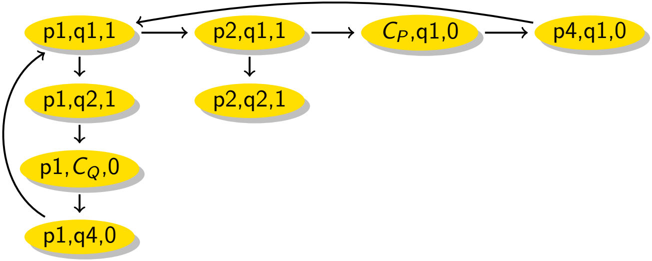
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

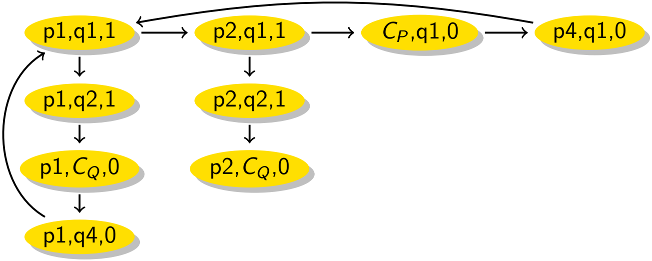
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



Mutual Exclusion: Attempt 1

- boolean variable $free = 1$

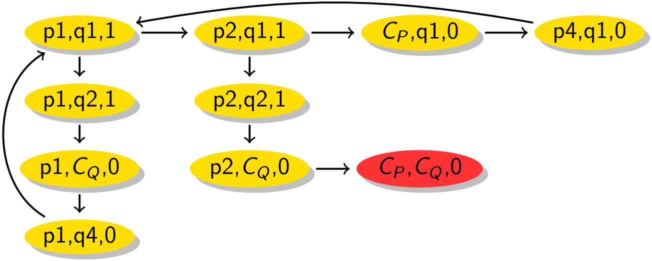
P

```
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

Q

```
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:



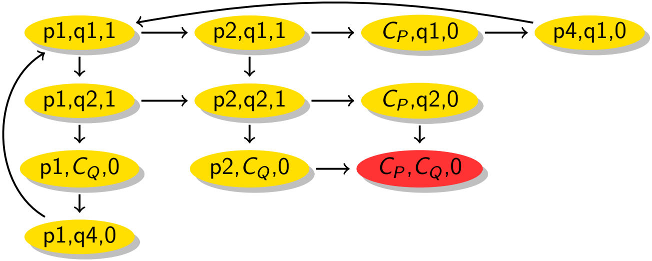
Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

```
P
loop forever
  p1: wait for $free = 1$
  p2: $free = 0$
  $C_P$: critical section
  p4: $free = 1$
```

```
Q
loop forever
  q1: wait for $free = 1$
  q2: $free = 0$
  $C_Q$: critical section
  q4: $free = 1$
```

For such a program we compute the state space:

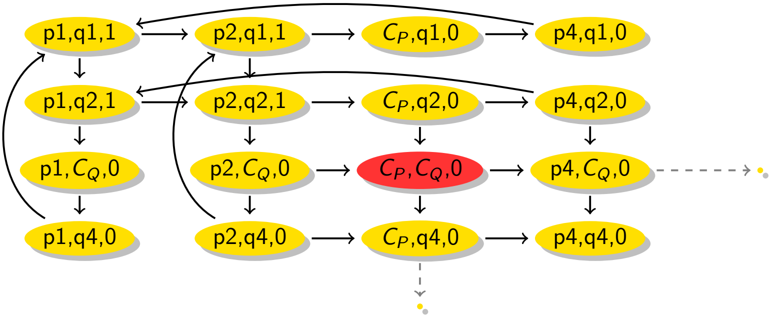


Mutual Exclusion: Attempt 1

- boolean variable *free* $= 1$

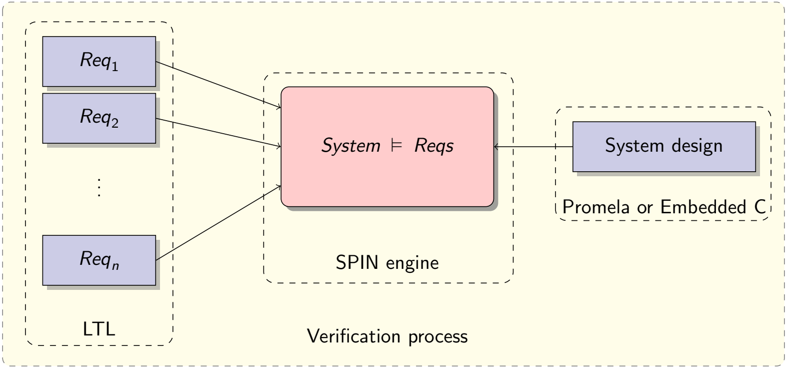
P	*Q*
loop forever p1: wait for $free = 1$ p2: $free = 0$ C_P: critical section p4: $free = 1$	loop forever q1: wait for $free = 1$ q2: $free = 0$ C_Q: critical section q4: $free = 1$

For such a program we compute the state space:



Model Checking

1. Formalize the system design
2. Formalize the validation requirements
3. Validate: system meets requirements



Req_1

Req_2

⋮

Req_n

LTL

$System \models Reqs$

SPIN engine

System design

Promela or Embedded C

Verification process

Mutual Exclusion: Peterson

- boolean variables $x = 0$, $y = 0$, $t = 0$

P

```
loop forever
  p1: $x = 1$
  p2: $turn = 1$
  p3: wait for $y = 0$ or $t = 0$
  $C_P$: critical section
  p4: $x = 0$
```

Q

```
loop forever
  q1: $y = 1$
  q2: $turn = 0$
  q3: wait for $x = 0$ or $t = 1$
  $C_Q$: critical section
  q4: $y = 0$
```

LTL: Applications

Safety properties

- “nothing bad ever happens”

$$\mathsf{G}\ \neg(\text{reactor temperature} > 1000)$$

- invariant: “*a* is always false”

Liveness properties

- “something good will eventually happen”

$$\mathsf{G}\ (\text{ordered} \rightarrow \mathsf{F}\ \text{delivered})$$

- termination: “the system will eventually terminate”
- response: “if action *a* occurs then *b* eventually will occur”

Deadlock freeness

- deadlock state: “a state where no actions are possible”
- no deadlocks: there is always some next state

$$\mathsf{G}\ (\neg\text{terminated} \rightarrow \mathsf{X}\ \top)$$

Industrial Case Studies I

Figure: After Flood Disaster (1953), Maeslant Barrier (Maeslantkering)

Industrial Case Studies: Flood Control

Verification of the interface between BOS and BESW:

- Beslis- en Ondersteunend Systeem (BOS)
- BEsturingsSysteem Waterweg (BESW)
- BOS takes the decision to move the barrier
- BESW performs this task

Even deadlocks were found in BESW!

Industrial Case Studies II

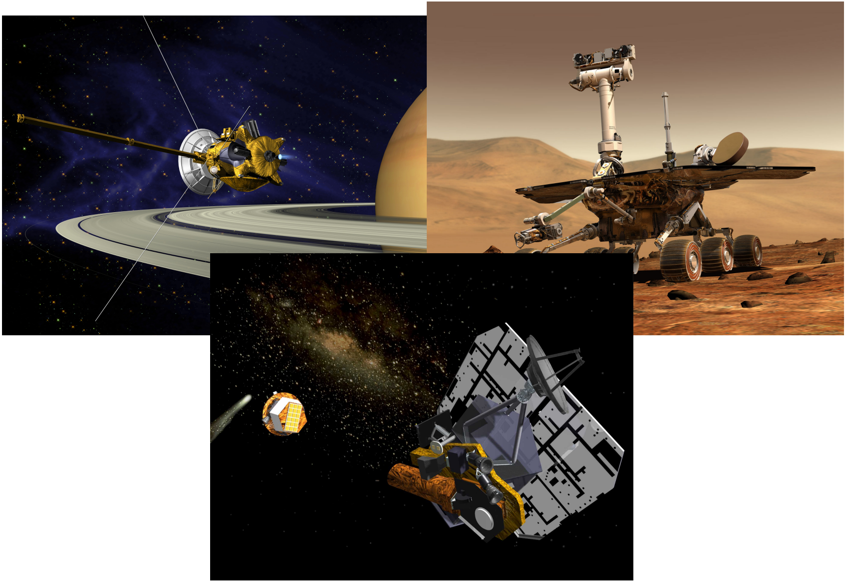
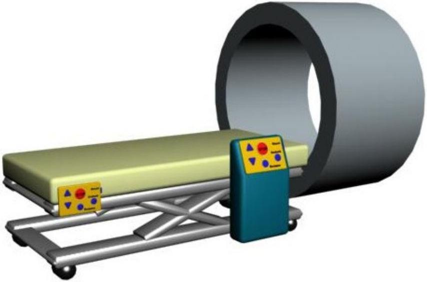
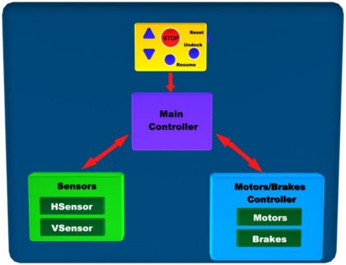


Figure: NASA Mission Critical Software: Cassini, Mars Rovers, Deep Impact

Industrial Case Studies III



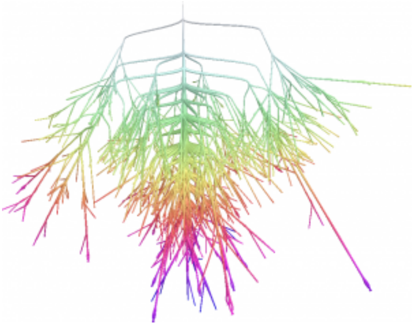


Main Controller

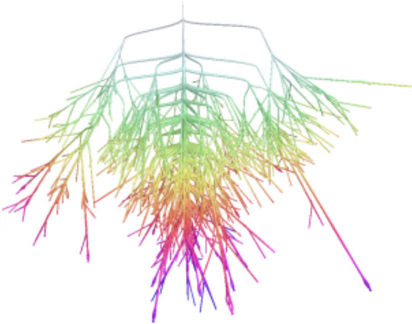
Sensors
- HSensor
- VSensor

Motors/Brakes Controller
- Motors
- Brakes

State Space Explosion

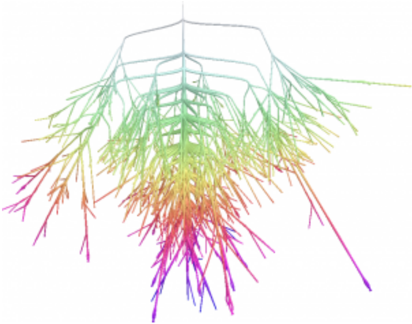


State Space Explosion



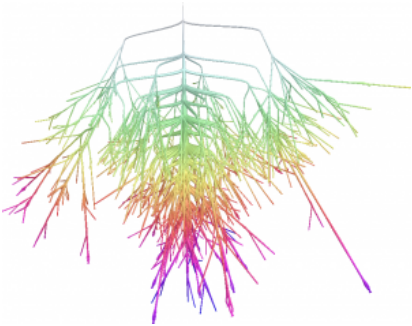
- Assume A_1, A_2, $\ldots$ are a processes each having 10 states

State Space Explosion



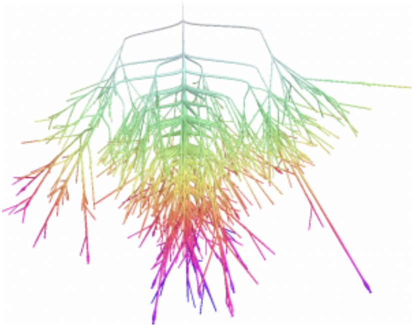
- Assume A_1, A_2, . . . are a processes each having 10 states
- Then A_1 and A_2 together have 100 states.

State Space Explosion



- Assume A_1, A_2, . . . are a processes each having 10 states
- Then A_1 and A_2 together have 100 states.
- Then A_1, . . . , A_n together have 10^n states.

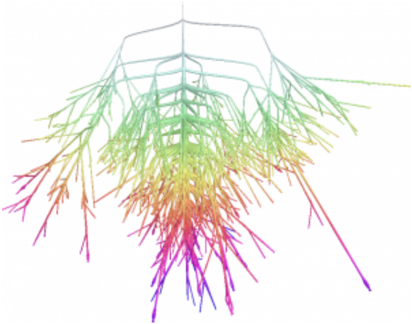
State Space Explosion



- Assume A_1, A_2, . . . are a processes each having 10 states
- Then A_1 and A_2 together have 100 states.
- Then A_1, . . . , A_n together have 10^n states.

This is the state space explosion problem.

State Space Explosion



- Assume A_1, A_2, . . . are a processes each having 10 states
- Then A_1 and A_2 together have 100 states.
- Then A_1, . . . , A_n together have 10^n states.

This is the state space explosion problem.

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi$$

where $p \in \Omega$

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \ \underbrace{\mathsf{EU}}_{\text{exists until}} \ \phi \mid \mathsf{EG}\ \phi \mid \mathsf{EX}\ \phi$$

where $p \in \Omega$

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

exists globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \,\mathsf{EU}\, \phi \mid \mathsf{EG}\, \phi \mid \mathsf{EX}\, \phi$$

exists until

where $p \in \Omega$

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{exists until}}{\mathsf{EU}} \phi \mid \overset{\text{exists globally}}{\mathsf{EG}} \phi \mid \underset{\text{exists next}}{\mathsf{EX}} \phi$$

where $p \in \Omega$

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi\ \underset{\text{exists until}}{\mathsf{EU}}\ \phi \mid \overset{\text{exists globally}}{\mathsf{EG}}\ \phi \mid \underset{\text{exists next}}{\mathsf{EX}}\ \phi$$

where $p \in \Omega$

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$
2. $\mathfrak{M}, s \models \phi\ \mathsf{EU}\ \psi$ (ϕ until ψ holds on some path starting from s)

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \; \mathsf{EU} \; \phi \mid \mathsf{EG} \; \phi \mid \mathsf{EX} \; \phi$$

exists globally

exists until

exists next

where $p \in \Omega$

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$
2. $\mathfrak{M}, s \models \phi \; \mathsf{EU} \; \psi$ (ϕ until ψ holds on some path starting from s)

   iff there is a path $s = s_1 \rightarrow s_2 \rightarrow \ldots$, such that
   for some $i \geq 1$, $\mathfrak{M}, s_i \models \psi$ and for all $j < i$, $\mathfrak{M}, s_j \models \phi$

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

exists globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi$$

exists until

exists next

where $p \in \Omega$

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$
2. $\mathfrak{M}, s \models \phi \text{ EU } \psi$ (ϕ until ψ holds on some path starting from s)

   iff there is a path $s = s_1 \to s_2 \to \ldots$, such that
   for some $i \geq 1$, $\mathfrak{M}, s_i \models \psi$ and for all $j < i$, $\mathfrak{M}, s_j \models \phi$
3. $\mathfrak{M}, s \models \text{EG } \phi$ (ϕ holds globally on some path starting from s)

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

exists globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \ \mathsf{EU}\ \phi \mid \mathsf{EG}\ \phi \mid \mathsf{EX}\ \phi$$

where $p \in \Omega$

exists until

exists next

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$
2. $\mathfrak{M}, s \models \phi \ \mathsf{EU}\ \psi$ (ϕ until ψ holds on some path starting from s)

   iff there is a path $s = s_1 \rightarrow s_2 \rightarrow \ldots$, such that
   for some $i \geq 1$, $\mathfrak{M}, s_i \models \psi$ and for all $j < i$, $\mathfrak{M}, s_j \models \phi$
3. $\mathfrak{M}, s \models \mathsf{EG}\ \phi$ (ϕ holds globally on some path starting from s)

   iff there is a path $s = s_1 \rightarrow s_2 \rightarrow \ldots$ such that
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \underset{\text{exists until}}{\mathsf{EU}} \phi \mid \overset{\text{exists globally}}{\mathsf{EG}}\ \phi \mid \underset{\text{exists next}}{\mathsf{EX}}\ \phi$$

where $p \in \Omega$

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$
2. $\mathfrak{M}, s \models \phi\ \mathsf{EU}\ \psi$ (ϕ until ψ holds on some path starting from s)

   iff there is a path $s = s_1 \rightarrow s_2 \rightarrow \ldots$, such that
   for some $i \geq 1$, $\mathfrak{M}, s_i \models \psi$ and for all $j < i$, $\mathfrak{M}, s_j \models \phi$
3. $\mathfrak{M}, s \models \mathsf{EG}\ \phi$ (ϕ holds globally on some path starting from s)

   iff there is a path $s = s_1 \rightarrow s_2 \rightarrow \ldots$ such that
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$
4. $\mathfrak{M}, s \models \mathsf{EX}\ \phi$ (ϕ holds in some next state)

Computation Tree Logic (CTL)

Computation Tree Logic (CTL) is defined by: exists globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi$$

where $p \in \Omega$ exists until exists next

The formula ϕ holds model $\mathfrak{M}$ at state s, $\mathfrak{M}, s \models \phi$, is defined by:

1. as usual: $\mathfrak{M}, s \models \top$, $\mathfrak{M}, s \models p$, $\mathfrak{M}, s \models \neg\phi$, $\mathfrak{M}, s \models \phi_1 \wedge \phi_2$
2. $\mathfrak{M}, s \models \phi \text{ EU } \psi$ (ϕ until ψ holds on some path starting from s)

   iff there is a path $s = s_1 \rightarrow s_2 \rightarrow \ldots$, such that
   for some $i \geq 1$, $\mathfrak{M}, s_i \models \psi$ and for all $j < i$, $\mathfrak{M}, s_j \models \phi$
3. $\mathfrak{M}, s \models \text{EG } \phi$ (ϕ holds globally on some path starting from s)

   iff there is a path $s = s_1 \rightarrow s_2 \rightarrow \ldots$ such that
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$
4. $\mathfrak{M}, s \models \text{EX } \phi$ (ϕ holds in some next state)

   iff $(M, s_2) \models \phi$ for some s_2 such that $s \rightarrow s_2$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \ \mathsf{EU}\ \phi \mid \mathsf{EG}\ \phi \mid \mathsf{EX}\ \phi \mid \phi \ \mathsf{AU}\ \phi \mid \mathsf{AG}\ \phi \mid \mathsf{AX}\ \phi$$

where $p \in \Omega$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi \mid \phi \text{ AU } \phi \mid \text{AG } \phi \mid \text{AX } \phi$

always until

where $p \in \Omega$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \ \mathsf{EU}\ \phi \mid \mathsf{EG}\ \phi \mid \mathsf{EX}\ \phi \mid \phi \ \mathsf{AU}\ \phi \mid \mathsf{AG}\ \phi \mid \mathsf{AX}\ \phi$$

always until

where $p \in \Omega$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \ \mathsf{EU}\ \phi \mid \mathsf{EG}\ \phi \mid \mathsf{EX}\ \phi \mid \phi\ \overset{\text{always until}}{\mathsf{AU}}\ \phi \mid \overset{\text{always globally}}{\mathsf{AG}}\ \phi \mid \overset{\text{always next}}{\mathsf{AX}}\ \phi$$

where $p \in \Omega$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi \mid \phi \text{ AU } \phi \mid \text{AG } \phi \mid \text{AX } \phi$

where $p \in \Omega$

always globally

always until

always next

1. $\mathfrak{M}, s \models \text{AG } \phi$ (ϕ holds globally on all paths starting from s)

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi \mid \phi \text{ AU } \phi \mid \text{AG } \phi \mid \text{AX } \phi$

where $p \in \Omega$

always until

always next

1. $\mathfrak{M}, s \models \text{AG } \phi$ (ϕ holds globally on all paths starting from s)
   iff for all paths $s = s_1 \to s_2 \to \ldots$ we have:
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi \mid \phi \text{ AU } \phi \mid \text{AG } \phi \mid \text{AX } \phi$$

always until

always next

where $p \in \Omega$

1. $\mathfrak{M}, s \models \text{AG } \phi$ (ϕ holds globally on all paths starting from s)
   iff for all paths $s = s_1 \to s_2 \to \ldots$ we have:
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$

   $\text{AG } \phi = \neg\text{EF } \neg\phi$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \; \mathsf{EU} \; \phi \mid \mathsf{EG} \; \phi \mid \mathsf{EX} \; \phi \mid \phi \; \mathsf{AU} \; \phi \mid \mathsf{AG} \; \phi \mid \mathsf{AX} \; \phi$$

always until

always next

where $p \in \Omega$

1. $\mathfrak{M}, s \models \mathsf{AG} \; \phi$ (ϕ holds globally on all paths starting from s)
   iff for all paths $s = s_1 \rightarrow s_2 \rightarrow \ldots$ we have:
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$ $\qquad$ $\mathsf{AG} \; \phi = \neg \mathsf{EF} \; \neg\phi$
2. $\mathfrak{M}, s \models \mathsf{AX} \; \phi$ (ϕ holds in all next states)

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \;\mathsf{EU}\; \phi \mid \mathsf{EG}\;\phi \mid \mathsf{EX}\;\phi \mid \phi \;\mathsf{AU}\; \phi \mid \mathsf{AG}\;\phi \mid \mathsf{AX}\;\phi$$

where $p \in \Omega$

always until

always next

1. $\mathfrak{M}, s \models \mathsf{AG}\;\phi$ (ϕ holds globally on all paths starting from s)
   iff for all paths $s = s_1 \rightarrow s_2 \rightarrow \ldots$ we have:
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$

   $\mathsf{AG}\;\phi = \neg\mathsf{EF}\;\neg\phi$

2. $\mathfrak{M}, s \models \mathsf{AX}\;\phi$ (ϕ holds in all next states)
   iff $(M, s_2) \models \phi$ for all s_2 such that $s \rightarrow s_2$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \land \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi \mid \phi \text{ AU } \phi \mid \text{AG } \phi \mid \text{AX } \phi$$

where $p \in \Omega$

always until

always next

1. $\mathfrak{M}, s \models \text{AG } \phi$ (ϕ holds globally on all paths starting from s)
   iff for all paths $s = s_1 \rightarrow s_2 \rightarrow \ldots$ we have:
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$ $\qquad$ $\text{AG } \phi = \neg\text{EF } \neg\phi$

2. $\mathfrak{M}, s \models \text{AX } \phi$ (ϕ holds in all next states)
   iff $(M, s_2) \models \phi$ for all s_2 such that $s \rightarrow s_2$ $\qquad$ $\text{AX } \phi = \neg\text{EX } \neg\phi$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi\ \mathsf{EU}\ \phi \mid \mathsf{EG}\ \phi \mid \mathsf{EX}\ \phi \mid \phi\ \mathsf{AU}\ \phi \mid \mathsf{AG}\ \phi \mid \mathsf{AX}\ \phi$$

where $p \in \Omega$

always until

always next

1. $\mathfrak{M}, s \models \mathsf{AG}\ \phi$ (ϕ holds globally on all paths starting from s)
   iff for all paths $s = s_1 \to s_2 \to \ldots$ we have:
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$ $\mathsf{AG}\ \phi = \neg\mathsf{EF}\ \neg\phi$
2. $\mathfrak{M}, s \models \mathsf{AX}\ \phi$ (ϕ holds in all next states)
   iff $(M, s_2) \models \phi$ for all s_2 such that $s \to s_2$ $\mathsf{AX}\ \phi = \neg\mathsf{EX}\ \neg\phi$
3. $\mathfrak{M}, s \models \phi\ \mathsf{AU}\ \psi$ (ϕ until ψ holds on all paths starting from s)

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \land \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi \mid \phi \text{ AU } \phi \mid \text{AG } \phi \mid \text{AX } \phi$$

where $p \in \Omega$

always until

always next

1. $\mathfrak{M}, s \models \text{AG } \phi$ (ϕ holds globally on all paths starting from s)

   iff for all paths $s = s_1 \to s_2 \to \ldots$ we have:
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$ $\text{AG } \phi = \neg\text{EF } \neg\phi$

2. $\mathfrak{M}, s \models \text{AX } \phi$ (ϕ holds in all next states)

   iff $(M, s_2) \models \phi$ for all s_2 such that $s \to s_2$ $\text{AX } \phi = \neg\text{EX } \neg\phi$

3. $\mathfrak{M}, s \models \phi \text{ AU } \psi$ (ϕ until ψ holds on all paths starting from s)

   iff for all paths $s = s_1 \to s_2 \to \ldots$ we have:
   for some $i \geq 1$, $\mathfrak{M}, s_i \models \psi$ and for all $j < i$, $\mathfrak{M}, s_j \models \phi$

CTL: Extensions

Computation Tree Logic (CTL) is defined by:

always globally

$$\phi ::= p \mid \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \text{ EU } \phi \mid \text{EG } \phi \mid \text{EX } \phi \mid \phi \text{ AU } \phi \mid \text{AG } \phi \mid \text{AX } \phi$$

where $p \in \Omega$

always until

always next

1. $\mathfrak{M}, s \models \text{AG } \phi$ (ϕ holds globally on all paths starting from s)

   iff for all paths $s = s_1 \to s_2 \to \ldots$ we have:
   for all $i \geq 1$, $\mathfrak{M}, s_i \models \phi$ $\qquad$ $\text{AG } \phi = \neg\text{EF } \neg\phi$

2. $\mathfrak{M}, s \models \text{AX } \phi$ (ϕ holds in all next states)

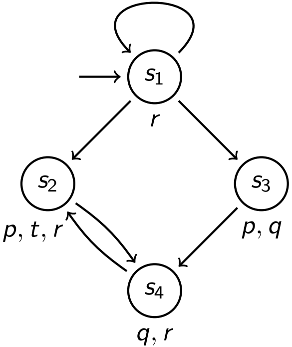
   iff $(M, s_2) \models \phi$ for all s_2 such that $s \to s_2$ $\qquad$ $\text{AX } \phi = \neg\text{EX } \neg\phi$

3. $\mathfrak{M}, s \models \phi \text{ AU } \psi$ (ϕ until ψ holds on all paths starting from s)

   iff for all paths $s = s_1 \to s_2 \to \ldots$ we have:
   for some $i \geq 1$, $\mathfrak{M}, s_i \models \psi$ and for all $j < i$, $\mathfrak{M}, s_j \models \phi$

$$\phi \text{ AU } \psi = \neg(\neg\psi \text{ EU } (\neg\phi \wedge \neg\psi)) \wedge \neg\text{EG } \neg\psi$$

CTL: Examples



Which of the states satisfies the following?

$? \models \mathrm{AF}\ t$

$? \models \neg\mathrm{EG}\ r$

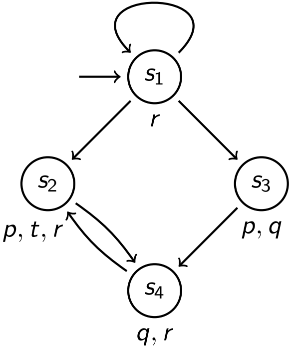
$? \models t\ \mathrm{EU}\ q$

$? \models \mathrm{EX}\ q$

$? \models \mathrm{AX}\ q$

$? \models \mathrm{EF}\ q$

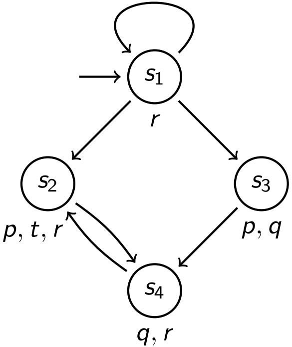
CTL: Examples



Which of the states satisfies the following?

$$
\begin{aligned}
\mathfrak{M}, s_2, s_3, s_4 &\models \text{AF } t \\
? &\models \neg\text{EG } r \\
? &\models t \text{ EU } q \\
? &\models \text{EX } q \\
? &\models \text{AX } q \\
? &\models \text{EF } q
\end{aligned}
$$

CTL: Examples



Which of the states satisfies the following?

$$\mathfrak{M}, s_2, s_3, s_4 \models \text{AF } t$$

$$\mathfrak{M}, s_3 \models \neg\text{EG } r$$

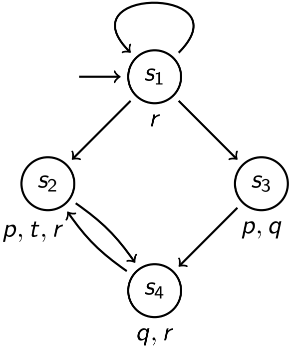
$$? \models t \text{ EU } q$$

$$? \models \text{EX } q$$

$$? \models \text{AX } q$$

$$? \models \text{EF } q$$

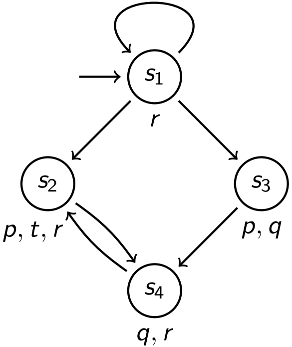
CTL: Examples



Which of the states satisfies the following?

$$
\begin{aligned}
\mathfrak{M}, s_2, s_3, s_4 &\models \text{AF } t \\
\mathfrak{M}, s_3 &\models \neg\text{EG } r \\
\mathfrak{M}, s_2, s_3, s_4 &\models t \text{ EU } q \\
? &\models \text{EX } q \\
? &\models \text{AX } q \\
? &\models \text{EF } q
\end{aligned}
$$

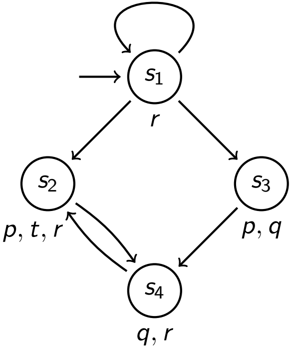
CTL: Examples



Which of the states satisfies the following?

$$\mathfrak{M}, s_2, s_3, s_4 \models \mathsf{AF}\ t$$
$$\mathfrak{M}, s_3 \models \neg\mathsf{EG}\ r$$
$$\mathfrak{M}, s_2, s_3, s_4 \models t\ \mathsf{EU}\ q$$
$$\mathfrak{M}, s_1, s_2, s_3 \models \mathsf{EX}\ q$$
$$? \models \mathsf{AX}\ q$$
$$? \models \mathsf{EF}\ q$$

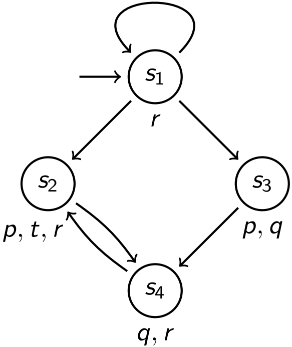
CTL: Examples



Which of the states satisfies the following?

$$\begin{aligned}
\mathfrak{M}, s_2, s_3, s_4 &\models \text{AF } t \\
\mathfrak{M}, s_3 &\models \neg\text{EG } r \\
\mathfrak{M}, s_2, s_3, s_4 &\models t \text{ EU } q \\
\mathfrak{M}, s_1, s_2, s_3 &\models \text{EX } q \\
\mathfrak{M}, s_2, s_3 &\models \text{AX } q \\
? &\models \text{EF } q
\end{aligned}$$

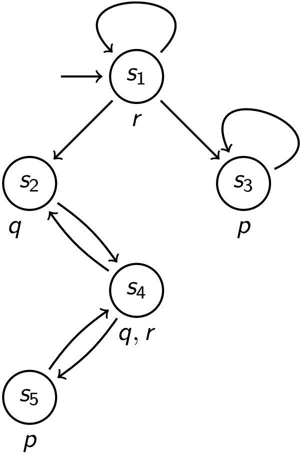
CTL: Examples



Which of the states satisfies the following?

$$\mathfrak{M}, s_2, s_3, s_4 \models \mathsf{AF}\ t$$
$$\mathfrak{M}, s_3 \models \neg\mathsf{EG}\ r$$
$$\mathfrak{M}, s_2, s_3, s_4 \models t\ \mathsf{EU}\ q$$
$$\mathfrak{M}, s_1, s_2, s_3 \models \mathsf{EX}\ q$$
$$\mathfrak{M}, s_2, s_3 \models \mathsf{AX}\ q$$
$$\mathfrak{M}, s_1, s_2, s_3, s_4 \models \mathsf{EF}\ q$$

CTL: Examples



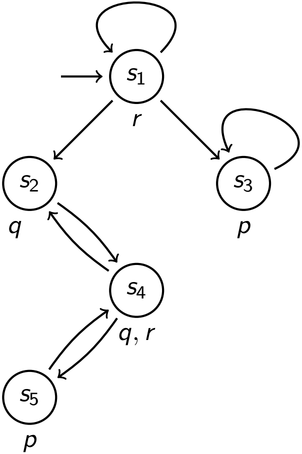
Which of the states satisfies the following?

$? \models \mathrm{AG}\ (\mathrm{EF}\ p)$

$? \models \mathrm{AG}\ ((q \vee r)\ \mathrm{AU}\ p)$

$? \models \mathrm{AG}\ (\mathrm{EF}\ (q \wedge r))$

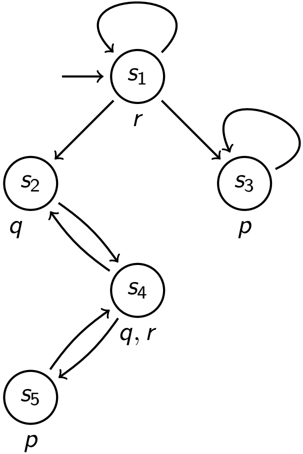
CTL: Examples



Which of the states satisfies the following?

$$\mathfrak{M}, s_1, s_2, s_3, s_4, s_5 \models \mathsf{AG}\ (\mathsf{EF}\ p)$$
$$? \models \mathsf{AG}\ ((q \vee r)\ \mathsf{AU}\ p)$$
$$? \models \mathsf{AG}\ (\mathsf{EF}\ (q \wedge r))$$

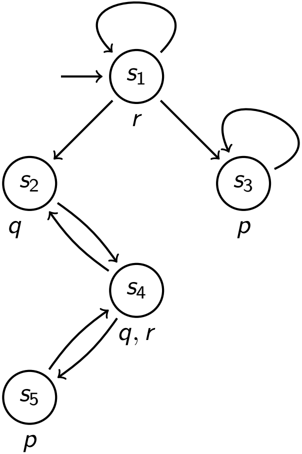
CTL: Examples



Which of the states satisfies the following?

$$\mathfrak{M}, s_1, s_2, s_3, s_4, s_5 \models \mathsf{AG}\ (\mathsf{EF}\ p)$$
$$\mathfrak{M}, s_3 \models \mathsf{AG}\ ((q \vee r)\ \mathsf{AU}\ p)$$
$$? \models \mathsf{AG}\ (\mathsf{EF}\ (q \wedge r))$$

CTL: Examples



Which of the states satisfies the following?

$$\mathfrak{M}, s_1, s_2, s_3, s_4, s_5 \models \mathsf{AG}\ (\mathsf{EF}\ p)$$

$$\mathfrak{M}, s_3 \models \mathsf{AG}\ ((q \vee r)\ \mathsf{AU}\ p)$$

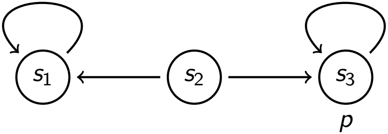
$$\mathfrak{M}, s_2, s_4, s_5 \models \mathsf{AG}\ (\mathsf{EF}\ (q \wedge r))$$

CTL vs LTL

CTL vs LTL

- a CTL formula necessitating E cannot be expressed in LTL

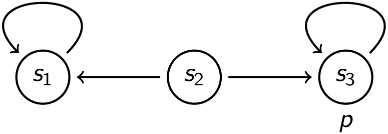
$$\mathrm{EX}\ p$$



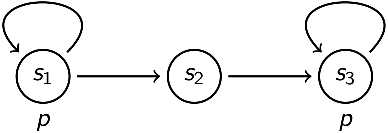
CTL vs LTL

- a CTL formula necessitating E cannot be expressed in LTL

$$\text{EX}\ p$$



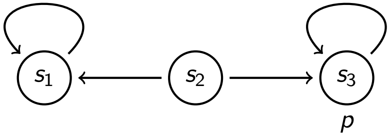
- the CTL formula AF AG p cannot be expressed in LTL



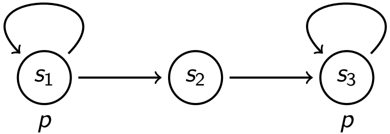
CTL vs LTL

- a CTL formula necessitating E cannot be expressed in LTL

$$\mathrm{EX}\ p$$



- the CTL formula AF AG p cannot be expressed in LTL



- the LTL formula G F $p \rightarrow$ F q cannot be expressed in CTL