Overview

- Introduction
- Modelling parallel systems
- Linear Time Properties
- Regular Properties
- Linear Temporal Logic (LTL)
- Computation-Tree Logic
- **Equivalences and Abstraction**
  - bisimulation
  - CTL, CTL*-equivalence
  - computing the bisimulation quotient
  - abstraction stutter steps ⟵
  - simulation relations

Classification of implementation relations

STUTTER5.4-CL

Classification of implementation relations

- linear vs. branching time
  - ∗ linear time: trace relations
  - ∗ branching time: (bi)simulation relations
- (nonsymmetric) preorders vs. equivalences:
  - ∗ preorders: trace inclusion, simulation
  - ∗ equivalences: trace equivalence, bisimulation
- strong vs. weak relations
  - ∗ strong: reasoning about all transitions
  - ∗ weak: abstraction from stutter steps

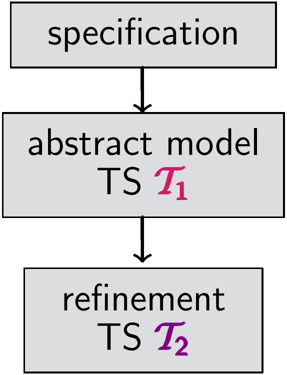
Classification of implementation relations

- linear vs. branching time
  - ∗ linear time: trace relations
  - ∗ branching time: (bi)simulation relations
- (nonsymmetric) preorders vs. equivalences:
  - ∗ preorders: trace inclusion, simulation
  - ∗ equivalences: trace equivalence, bisimulation
- strong vs. weak relations
  - ∗ strong: reasoning about all transitions
  - ∗ weak: abstraction from stutter steps

Classification of implementation relations

- linear vs. branching time
  - ∗ linear time: trace relations
  - ∗ branching time: (bi)simulation relations
- (nonsymmetric) preorders vs. equivalences:
  - ∗ preorders: trace inclusion, simulation
  - ∗ equivalences: trace equivalence, bisimulation
- strong vs. weak relations
  - ∗ strong: reasoning about all transitions
  - ∗ weak: abstraction from stutter steps

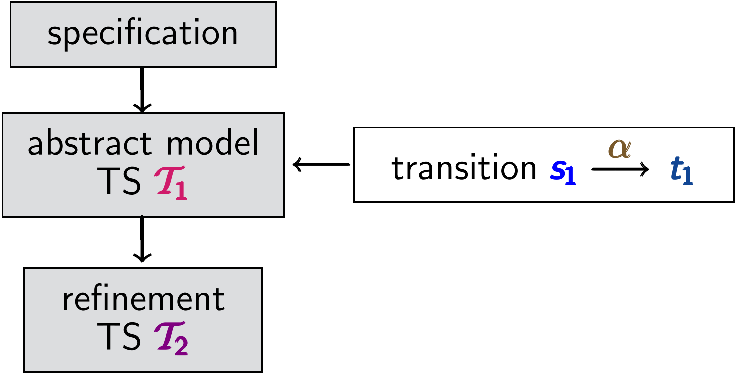
Design by stepwise refinement


specification

abstract model
TS $\mathcal{T}_1$

refinement
TS $\mathcal{T}_2$

Design by stepwise refinement



specification

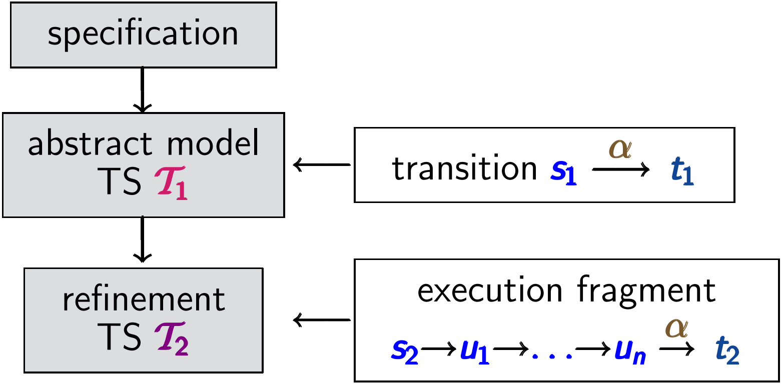
abstract model
TS $\mathcal{T}_1$

transition $s_1 \xrightarrow{\alpha} t_1$

refinement
TS $\mathcal{T}_2$

Design by stepwise refinement

STUTTER5.4-1



specification

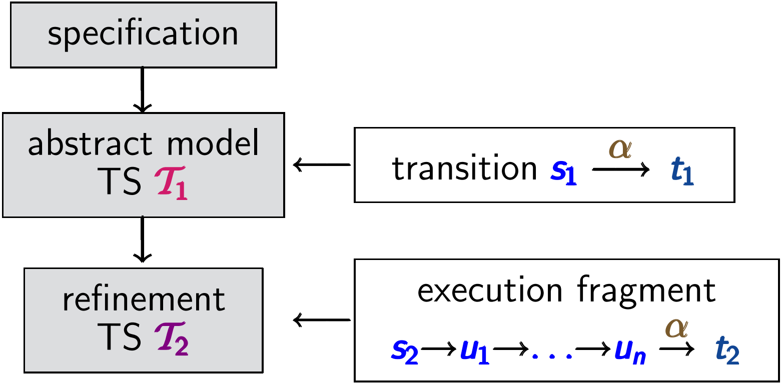
abstract model
TS $\mathcal{T}_1$

transition $s_1 \xrightarrow{\alpha} t_1$

refinement
TS $\mathcal{T}_2$

execution fragment
$s_2 \longrightarrow u_1 \longrightarrow \ldots \longrightarrow u_n \xrightarrow{\alpha} t_2$

Design by stepwise refinement



specification

abstract model TS $\mathcal{T}_1$

transition $s_1 \xrightarrow{\alpha} t_1$

refinement TS $\mathcal{T}_2$

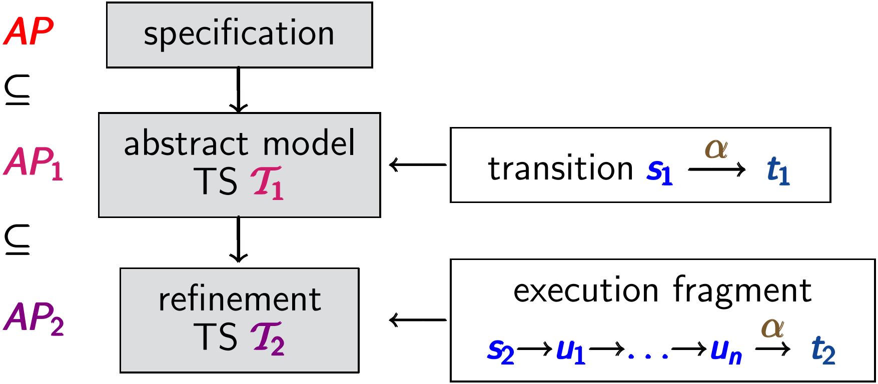
execution fragment

$s_2 \longrightarrow u_1 \longrightarrow \ldots \longrightarrow u_n \xrightarrow{\alpha} t_2$

internal computation prior to the execution of action α

- access on auxiliary variables of $\mathcal{T}_2$
- no access on variables of $\mathcal{T}_1$

Design by stepwise refinement



AP

$\subseteq$

AP_1

$\subseteq$

AP_2

specification

abstract model TS $\mathcal{T}_1$

refinement TS $\mathcal{T}_2$

transition $s_1 \xrightarrow{\alpha} t_1$

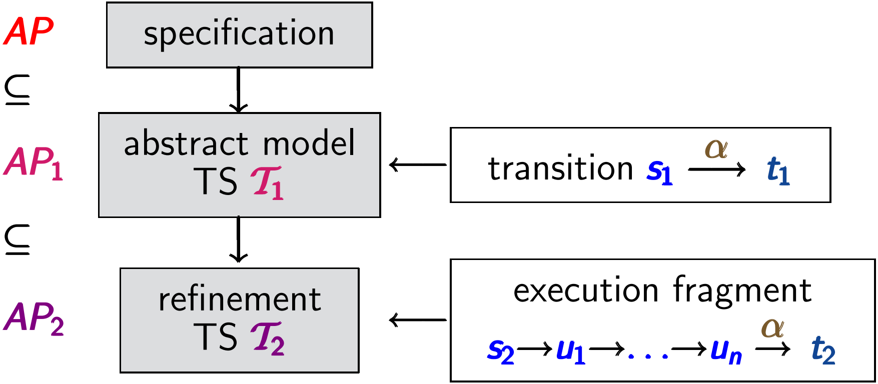
execution fragment

$s_2 \longrightarrow u_1 \longrightarrow \ldots \longrightarrow u_n \xrightarrow{\alpha} t_2$

internal computation prior to the execution of action α

- access on auxiliary variables of $\mathcal{T}_2$
- no access on variables of $\mathcal{T}_1$

Design by stepwise refinement



AP specification

$\subseteq$

AP_1 abstract model TS $\mathcal{T}_1$ ← transition $s_1 \xrightarrow{\alpha} t_1$

$\subseteq$

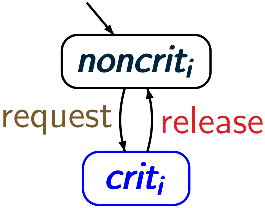
AP_2 refinement TS $\mathcal{T}_2$ ← execution fragment $s_2 \to u_1 \to \ldots \to u_n \xrightarrow{\alpha} t_2$

internal computation prior to the execution of action α

- access on auxiliary variables of $\mathcal{T}_2$
- no access on variables of $\mathcal{T}_1$

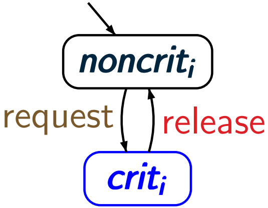
$s_2 \to u_1 \to \ldots \to u_n$: stutter steps w.r.t. AP_1 (or AP)

Mututal exclusion (with arbiter)

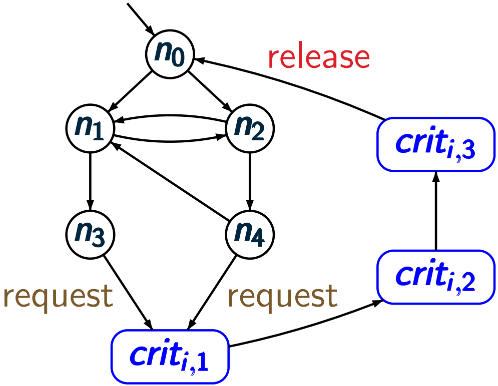


abstract representation
for process P_i

Mututal exclusion (with arbiter)



abstract representation
for process P_i



refined representation
for process P_i

Example: abstraction from stutter steps

process P

```
LOOP FOREVER
    $x := y \text{ MOD } 3$
    $y := (x + y) \text{ MOD } 3$
    $z := (2y - x) \text{ DIV } 3$
END LOOP
```

Example: abstraction from stutter steps

process $P \rightsquigarrow$ transition system $\mathcal{T}_P$

```
$\ell_0$  LOOP FOREVER
$\ell_1$          $x := y$ MOD 3
$\ell_2$          $y := (x + y)$ MOD 3
$\ell_3$          $z := (2y - x)$ DIV 3
$\ell_4$  END LOOP
```

Example: abstraction from stutter steps

process $P \rightsquigarrow$ transition system $\mathcal{T}_P$

```
$\ell_0$  LOOP FOREVER
$\ell_1$      $x := y$ MOD 3
$\ell_2$      $y := (x + y)$ MOD 3
$\ell_3$      $z := (2y - x)$ DIV 3
$\ell_4$  END LOOP
```

CTL* property: does $\mathcal{T}_P \models \forall\Box\Diamond(z = 1)$ hold ?

Example: abstraction from stutter steps

process $P \rightsquigarrow$ transition system $\mathcal{T}_P$ over $AP = Eval(z)$

```
$\ell_0$   LOOP FOREVER
$\ell_1$         $x := y$ MOD 3
$\ell_2$         $y := (x + y)$ MOD 3
$\ell_3$         $z := (2y - x)$ DIV 3
$\ell_4$   END LOOP
```

CTL* property: does $\mathcal{T}_P \models \forall\Box\Diamond(z = 1)$ hold ?

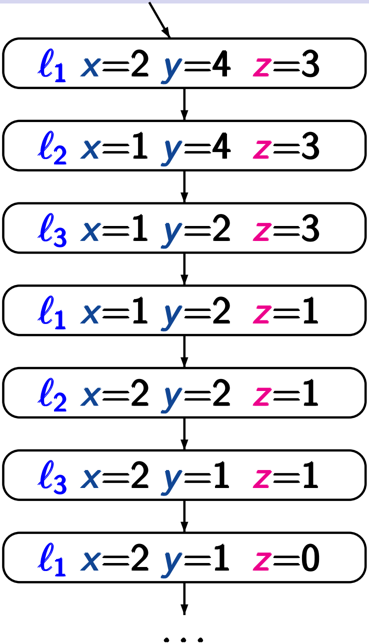
Example: abstraction from stutter steps

process $P \rightsquigarrow$ transition system $\mathcal{T}_P$ over $AP = \mathit{Eval}(z)$

```
$\ell_0$  LOOP FOREVER
$\ell_1$      $x := y$ MOD 3             ⟵ stutter step
$\ell_2$      $y := (x + y)$ MOD 3       ⟵ stutter step
$\ell_3$      $z := (2y - x)$ DIV 3      ⟵ visible action
$\ell_4$  END LOOP
```

CTL* property: does $\mathcal{T}_P \models \forall\Box\Diamond(z = 1)$ hold ?

Transition system for process P



ℓ_1 $x=2$ $y=4$ $z=3$

ℓ_2 $x=1$ $y=4$ $z=3$

ℓ_3 $x=1$ $y=2$ $z=3$

ℓ_1 $x=1$ $y=2$ $z=1$

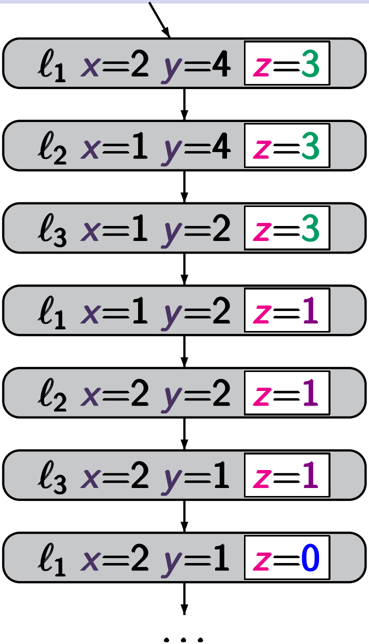
ℓ_2 $x=2$ $y=2$ $z=1$

ℓ_3 $x=2$ $y=1$ $z=1$

ℓ_1 $x=2$ $y=1$ $z=0$

$\dots$

Analysis by abstraction from stutter steps



ℓ_1 $x{=}2$ $y{=}4$ $\boxed{z{=}3}$

ℓ_2 $x{=}1$ $y{=}4$ $\boxed{z{=}3}$

ℓ_3 $x{=}1$ $y{=}2$ $\boxed{z{=}3}$

ℓ_1 $x{=}1$ $y{=}2$ $\boxed{z{=}1}$

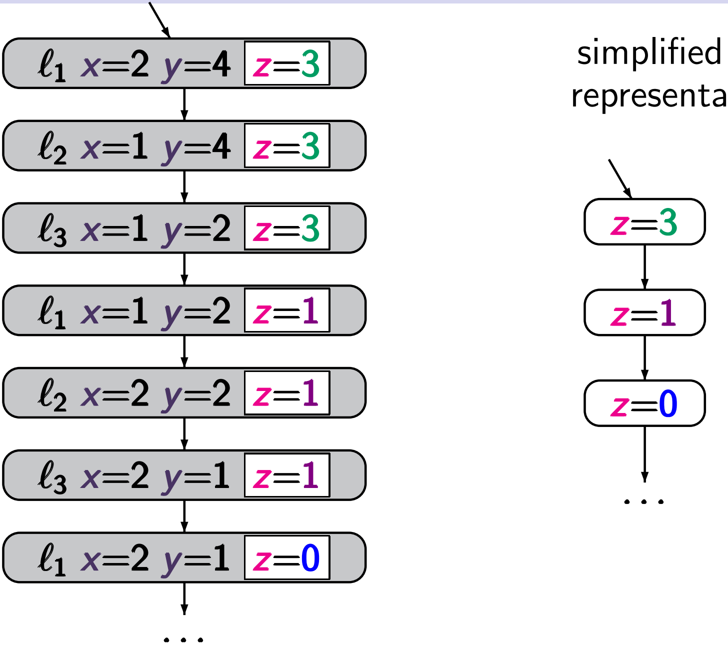
ℓ_2 $x{=}2$ $y{=}2$ $\boxed{z{=}1}$

ℓ_3 $x{=}2$ $y{=}1$ $\boxed{z{=}1}$

ℓ_1 $x{=}2$ $y{=}1$ $\boxed{z{=}0}$

$\cdots$

Analysis by abstraction from stutter steps



simplified TS representation

Overview

- Introduction
- Modelling parallel systems
- Linear Time Properties
- Regular Properties
- Linear Temporal Logic (LTL)
- Computation-Tree Logic (CTL)
- **Equivalences and Abstraction**
  - bisimulation, CTL/CTL*-equivalence
  - computing the bisimulation quotient
  - abstraction stutter steps
    - stutter LT relations ⟵
    - stutter bisimulation
  - simulation relations

Remind: trace relations

STUTTER5.4-5-REMIND

Remind: trace relations

trace equivalence for paths

π_1, π_2 are trace equivalent iff $\mathit{trace}(\pi_1) = \mathit{trace}(\pi_2)$

Remind: trace relations

trace equivalence for paths

π_1, π_2 are trace equivalent iff $\mathit{trace}(\pi_1) = \mathit{trace}(\pi_2)$

trace inclusion for TS: $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$

$\forall \pi_1 \in \mathit{Traces}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Traces}(\mathcal{T}_2)$

s.t. π_1, π_2 are trace equivalent

Remind: trace relations

trace equivalence for paths

π_1, π_2 are trace equivalent iff $trace(\pi_1) = trace(\pi_2)$

trace inclusion for TS: $Traces(\mathcal{T}_1) \subseteq Traces(\mathcal{T}_2)$

$\forall \pi_1 \in Traces(\mathcal{T}_1)\ \exists \pi_2 \in Traces(\mathcal{T}_2)$

s.t. π_1, π_2 are trace equivalent

trace equivalence for TS:

$$Traces(\mathcal{T}_1) \subseteq Traces(\mathcal{T}_2) \quad \wedge \quad Traces(\mathcal{T}_2) \subseteq Traces(\mathcal{T}_1)$$

Remind: trace relations

trace equivalence for paths

π_1, π_2 are trace equivalent iff $trace(\pi_1) = trace(\pi_2)$

trace inclusion for TS: $Traces(\mathcal{T}_1) \subseteq Traces(\mathcal{T}_2)$

$\forall \pi_1 \in Traces(\mathcal{T}_1)\ \exists \pi_2 \in Traces(\mathcal{T}_2)$

s.t. π_1, π_2 are trace equivalent

trace equivalence for TS:

$$Traces(\mathcal{T}_1) \subseteq Traces(\mathcal{T}_2) \quad \wedge \quad Traces(\mathcal{T}_2) \subseteq Traces(\mathcal{T}_1)$$

$Traces(\mathcal{T}_1) \subseteq Traces(\mathcal{T}_2)$ iff for each **LT** property E:

$\mathcal{T}_2 \models E$ implies $\mathcal{T}_1 \models E$

Remind: trace relations

trace equivalence for paths

π_1, π_2 are trace equivalent iff $\mathit{trace}(\pi_1) = \mathit{trace}(\pi_2)$

trace inclusion for TS: $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$

$\forall \pi_1 \in \mathit{Traces}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Traces}(\mathcal{T}_2)$
s.t. π_1, π_2 are trace equivalent

trace equivalence for TS:

$$\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2) \quad \wedge \quad \mathit{Traces}(\mathcal{T}_2) \subseteq \mathit{Traces}(\mathcal{T}_1)$$

$\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$ iff for each **LT** property E:
$\mathcal{T}_2 \models E$ implies $\mathcal{T}_1 \models E$

↑

trace equivalent TS satisfy the same **LTL** formulas

Stutter equivalence for paths

Stutter equivalence for paths

stutter equivalence for infinite path fragments:

Stutter equivalence for paths

stutter equivalence for infinite path fragments:

$\pi_1 \triangleq \pi_2$ iff there exists an infinite word $A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^{\omega}$ s.t. the traces of π_1 and π_2 are of the form

$$A_0 \ldots A_0\ A_1 \ldots A_1\ A_2 \ldots A_2 \ldots$$

Stutter equivalence for paths

stutter equivalence for infinite path fragments:

$\pi_1 \triangleq \pi_2$ iff there exists an infinite word $A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^{\omega}$ s.t. the traces of π_1 and π_2 are of the form

$$A_0^{n_0}\ A_1^{n_1}\ A_2^{n_2} \ldots$$

where $n_0, n_1, n_2, \ldots$ are natural numbers ≥ 1

Stutter equivalence for paths

stutter equivalence for infinite path fragments:

$\pi_1 \triangleq \pi_2$ iff there exists an infinite word $A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^{\omega}$ s.t. the traces of π_1 and π_2 are of the form

$${A_0}^{+}\ {A_1}^{+}\ {A_2}^{+} \ldots$$

Stutter equivalence for paths

stutter equivalence for infinite path fragments:

$\pi_1 \triangleq \pi_2$ iff there exists an infinite word $A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^{\omega}$ s.t. the traces of π_1 and π_2 are of the form

$$A_0{}^{+}\ A_1{}^{+}\ A_2{}^{+} \ldots$$

stutter equivalence for finite path fragments:

$\hat{\pi}_1 \triangleq \hat{\pi}_2$ iff there exists a finite word $A_0\, A_1\, A_2 \ldots A_n \in \left(2^{AP}\right)^{+}$ s.t. the traces of $\hat{\pi}_1$ and $\hat{\pi}_2$ are in

$$A_0{}^{+}A_1{}^{+}A_2{}^{+} \ldots A_n{}^{+}$$

Stutter trace relations for TS

stutter equivalence for infinite path fragments:

$\pi_1 \triangleq \pi_2$ iff there exists an infinite word $A_0 \, A_1 \, A_2 \ldots \in \left(2^{AP}\right)^\omega$ s.t. the traces of π_1 and π_2 are of the form

$$A_0{}^+ \, A_1{}^+ \, A_2{}^+ \ldots$$

Stutter trace relations for TS

stutter equivalence for infinite path fragments:

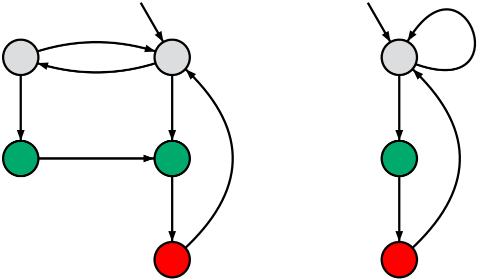
$\pi_1 \triangleq \pi_2$ iff there exists an infinite word $A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^{\omega}$ s.t. the traces of π_1 and π_2 are of the form $A_0^{\,+}\, A_1^{\,+}\, A_2^{\,+} \ldots$

stutter trace inclusion for transition systems:

$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2$ iff for all paths π_1 of $\mathcal{T}_1$ there exists a path π_2 of $\mathcal{T}_2$ s.t. $\pi_1 \triangleq \pi_2$

Example: stutter trace inclusion ⊴

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2)$$
$$\text{s.t. } \pi_1 \triangleq \pi_2$$



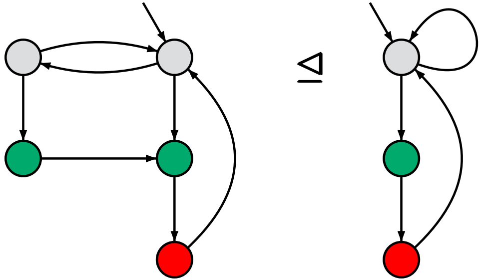
⚪ = ∅

🔴 = $\{a\}$

🟢 = $\{b\}$

Example: stutter trace inclusion $\trianglelefteq$

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2) \quad \text{s.t. } \pi_1 \overset{\Delta}{=} \pi_2$$



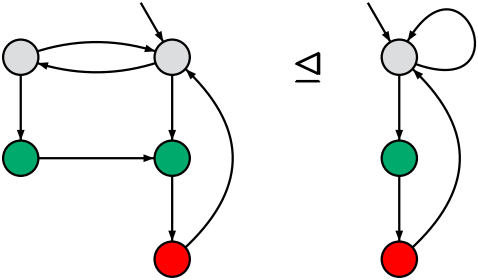
⚪ = ∅

🔴 = {*a*}

🟢 = {*b*}

Example: stutter trace inclusion $\trianglelefteq$

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2) \quad \text{s.t. } \pi_1 \triangleq \pi_2$$



⚪ $= \varnothing$

🔴 $= \{a\}$

🟢 $= \{b\}$

all traces have the form $(\varnothing^+\{b\}^+\{a\}^+)^\omega$

or $(\varnothing^+\{b\}^+\{a\}^+)^*\varnothing^\omega$

Stutter trace inclusion and LTL

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2)$$
$$\text{s.t. } \pi_1 \triangleq \pi_2$$

Does stutter trace inclusion preserve **LTL** properties?

Stutter trace inclusion and LTL

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2)$$
$$\text{s.t. } \pi_1 \triangleq \pi_2$$

Does stutter trace inclusion preserve **LTL** properties?

↑

i.e., for all **LTL** formulas φ:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \ \wedge\ \mathcal{T}_2 \models \varphi \quad \text{implies} \quad \mathcal{T}_1 \models \varphi$$

Stutter trace inclusion and LTL

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2)$$

$$\text{s.t. } \pi_1 \triangleq \pi_2$$

Does stutter trace inclusion preserve **LTL** properties?

i.e., for all **LTL** formulas φ:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \ \wedge \ \mathcal{T}_2 \models \varphi \quad \text{implies} \quad \mathcal{T}_1 \models \varphi$$

answer: **no**

Stutter trace inclusion and LTL

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2)$$

$$\text{s.t. } \pi_1 \triangleq \pi_2$$

Does stutter trace inclusion preserve **LTL** properties?

i.e., for all **LTL** formulas φ:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \ \wedge\ \mathcal{T}_2 \models \varphi \quad \text{implies} \quad \mathcal{T}_1 \models \varphi$$

answer: **no**

Example: **LTL** formulas of the form $\bigcirc a$

Stutter trace inclusion and $\mathbf{LTL}_{\setminus\bigcirc}$

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2)$$
$$\text{s.t. } \pi_1 \triangleq \pi_2$$

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ are TS without terminal states and φ an $\mathbf{LTL}_{\setminus\bigcirc}$ formula. Then:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \wedge \quad \mathcal{T}_2 \models \varphi \quad \text{implies} \quad \mathcal{T}_1 \models \varphi$$

Stutter trace inclusion and $\mathbf{LTL}_{\setminus\bigcirc}$

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{iff} \quad \forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2)$$
$$\text{s.t. } \pi_1 \triangleq \pi_2$$

Let $\mathcal{T}_1$ and $\mathcal{T}_2$ are TS without terminal states and φ an $\mathbf{LTL}_{\setminus\bigcirc}$ formula. Then:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \ \wedge\ \mathcal{T}_2 \models \varphi \quad \text{implies} \quad \mathcal{T}_1 \models \varphi$$

where $\mathbf{LTL}_{\setminus\bigcirc} = \mathbf{LTL}$ without the next operator $\bigcirc$

Stutter trace equivalence $\triangleq$ for TS

Stutter trace equivalence $\triangleq$ for TS

stutter trace inclusion $\mathcal{T}_1 \trianglelefteq \mathcal{T}_2$

$$\forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2) \text{ s.t. } \pi_1 \triangleq \pi_2$$

Stutter trace equivalence $\triangleq$ for TS

stutter trace inclusion $\mathcal{T}_1 \trianglelefteq \mathcal{T}_2$

$$\forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2) \text{ s.t. } \pi_1 \triangleq \pi_2$$

stutter trace equivalence

$$\mathcal{T}_1 \triangleq \mathcal{T}_2 \quad \text{iff} \quad \mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{and} \quad \mathcal{T}_2 \trianglelefteq \mathcal{T}_1$$

Stutter trace equivalence $\triangleq$ for TS

stutter trace inclusion $\mathcal{T}_1 \trianglelefteq \mathcal{T}_2$

$$\forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2) \text{ s.t. } \pi_1 \triangleq \pi_2$$

stutter trace equivalence

$$\mathcal{T}_1 \triangleq \mathcal{T}_2 \quad \text{iff} \quad \mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \quad \text{and} \quad \mathcal{T}_2 \trianglelefteq \mathcal{T}_1$$

↑

kernel of $\trianglelefteq$, i.e.,
coarsest equivalence that refines $\trianglelefteq$

Stutter trace equivalence $\triangleq$ for TS

stutter trace inclusion $\mathcal{T}_1 \trianglelefteq \mathcal{T}_2$

$$\forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1) \ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2) \text{ s.t. } \pi_1 \triangleq \pi_2$$

For all $\mathbf{LTL}_{\setminus \bigcirc}$ formulas φ:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \ \wedge \ \mathcal{T}_2 \models \varphi \ \text{ implies } \ \mathcal{T}_1 \models \varphi$$

stutter trace equivalence

$$\mathcal{T}_1 \triangleq \mathcal{T}_2 \ \text{ iff } \ \mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \ \text{ and } \ \mathcal{T}_2 \trianglelefteq \mathcal{T}_1$$

kernel of $\trianglelefteq$, i.e.,
coarsest equivalence that refines $\trianglelefteq$

Stutter trace equivalence $\triangleq$ for TS

stutter trace inclusion $\mathcal{T}_1 \trianglelefteq \mathcal{T}_2$

$$\forall \pi_1 \in \mathit{Paths}(\mathcal{T}_1)\ \exists \pi_2 \in \mathit{Paths}(\mathcal{T}_2) \text{ s.t. } \pi_1 \triangleq \pi_2$$

For all $\mathbf{LTL}_{\setminus\bigcirc}$ formulas φ:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \ \wedge\ \mathcal{T}_2 \models \varphi \ \text{ implies } \ \mathcal{T}_1 \models \varphi$$

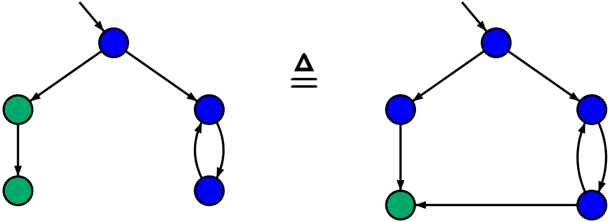
stutter trace equivalence

$$\mathcal{T}_1 \triangleq \mathcal{T}_2 \ \text{ iff } \ \mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \ \text{ and } \ \mathcal{T}_2 \trianglelefteq \mathcal{T}_1$$

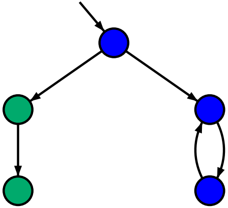
If $\mathcal{T}_1 \triangleq \mathcal{T}_2$ then $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathbf{LTL}_{\setminus\bigcirc}$ equivalent.

Correct or wrong?

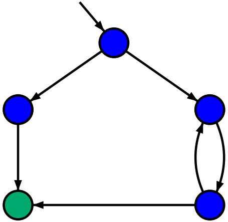
STUTTER5.4-13A



Correct or wrong?

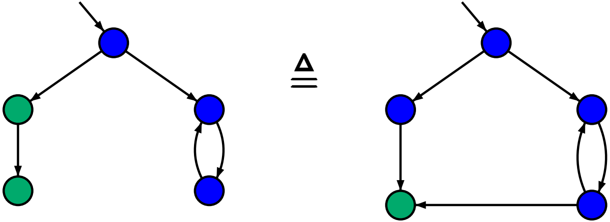


$\triangleq$



correct

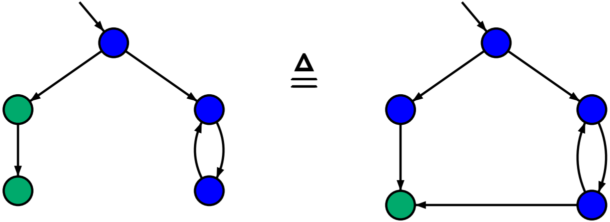
Correct or wrong?



correct

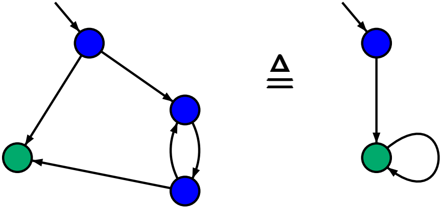
The traces of $\mathcal{T}_1$ and $\mathcal{T}_2$ have the form 🔵$^+$🟢$^+$ or 🔵$^\omega$

Correct or wrong?



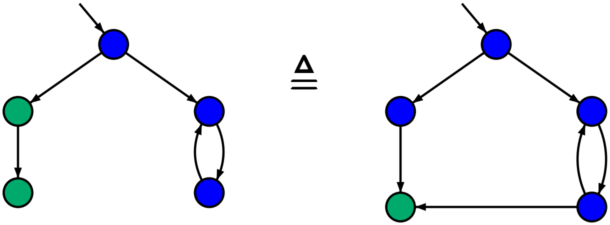
correct

The traces of $\mathcal{T}_1$ and $\mathcal{T}_2$ have the form 🔵$^+$🟢$^+$ or 🔵$^\omega$



Correct or wrong?

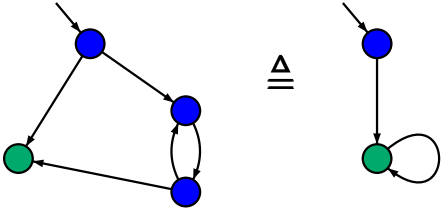
STUTTER5.4-13A



$\triangleq$

correct

The traces of $\mathcal{T}_1$ and $\mathcal{T}_2$ have the form 🔵$^+$🟢$^+$ or 🔵$^\omega$

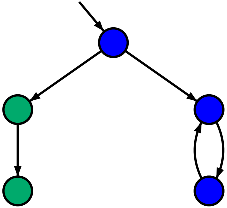


$\triangleq$

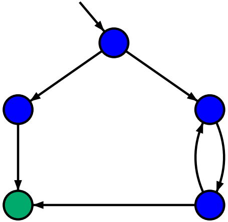
wrong

Correct or wrong?

STUTTER5.4-13A

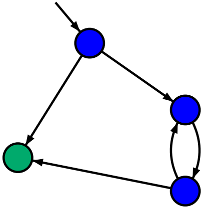


$\triangleq$

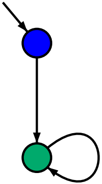


correct

The traces of $\mathcal{T}_1$ and $\mathcal{T}_2$ have the form ●$^+$●$^+$ or ●$^\omega$



$\triangleq$



wrong

$\mathcal{T}_1$ has a finite trace ●$^+$●, while $\mathcal{T}_2$ has not

Correct or wrong?

If $\mathcal{T}_1$ and $\mathcal{T}_2$ are TS over AP then:

$$\mathcal{T}_1 \sim \mathcal{T}_2 \quad \text{implies} \quad \mathcal{T}_1 \triangleq \mathcal{T}_2$$

Correct or wrong?

If $\mathcal{T}_1$ and $\mathcal{T}_2$ are TS over AP then:

$$\mathcal{T}_1 \sim \mathcal{T}_2 \quad \text{implies} \quad \mathcal{T}_1 \triangleq \mathcal{T}_2$$

bisimulation equivalence

stutter trace equivalence

Correct or wrong?

If $\mathcal{T}_1$ and $\mathcal{T}_2$ are TS over AP then:

$$\mathcal{T}_1 \sim \mathcal{T}_2 \quad \text{implies} \quad \mathcal{T}_1 \triangleq \mathcal{T}_2$$

bisimulation equivalence

stutter trace equivalence

correct

Correct or wrong?

If $\mathcal{T}_1$ and $\mathcal{T}_2$ are TS over AP then:

$$\mathcal{T}_1 \sim \mathcal{T}_2 \quad \text{implies} \quad \mathcal{T}_1 \triangleq \mathcal{T}_2$$

bisimulation equivalence

stutter trace equivalence

correct, as

- $\mathcal{T}_1 \sim \mathcal{T}_2$ implies $\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$
- trace equivalent paths are stutter trace equivalent

Correct or wrong?

If $\mathcal{T}_1$ and $\mathcal{T}_2$ are TS over AP then:

$$\mathcal{T}_1 \sim \mathcal{T}_2 \quad \text{implies} \quad \mathcal{T}_1 \triangleq \mathcal{T}_2$$

bisimulation equivalence

stutter trace equivalence

correct, as

- $\mathcal{T}_1 \sim \mathcal{T}_2$ implies $\mathit{Traces}(\mathcal{T}_1) = \mathit{Traces}(\mathcal{T}_2)$
- trace equivalent paths are stutter trace equivalent

obviously: $\mathit{Traces}(\mathcal{T}_1) \subseteq \mathit{Traces}(\mathcal{T}_2)$ implies $\mathcal{T}_1 \trianglelefteq \mathcal{T}_2$

Stutter-insensitive LT properties

Stutter-insensitive LT properties

stutter equivalence for infinite words

Stutter-insensitive LT properties

stutter equivalence for infinite words σ_1, $\sigma_2 \in \left(2^{AP}\right)^{\omega}$:

Stutter-insensitive LT properties

stutter equivalence for infinite words $\sigma_1, \sigma_2 \in \left(2^{AP}\right)^\omega$:

$\sigma_1 \triangleq \sigma_2$ iff there exists an infinite word $A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^\omega$ s.t. σ_1 and σ_2 are in ${A_0}^+ {A_1}^+ {A_2}^+ \ldots$

Stutter-insensitive LT properties

stutter equivalence for infinite words $\sigma_1, \sigma_2 \in \left(2^{AP}\right)^\omega$:

$\sigma_1 \triangleq \sigma_2$ iff there exists an infinite word $A_0\,A_1\,A_2 \ldots \in \left(2^{AP}\right)^\omega$ s.t. σ_1 and σ_2 are in $A_0{}^+A_1{}^+A_2{}^+\ldots$

Let $E \subseteq \left(2^{AP}\right)^\omega$ be an LT property. E is called stutter-insensitive iff for all $\sigma_1, \sigma_2 \in \left(2^{AP}\right)^\omega$:

if $\sigma_1 \in E$ and $\sigma_1 \triangleq \sigma_2$ then $\sigma_2 \in E$

Stutter-insensitive LT properties

stutter equivalence for infinite words $\sigma_1, \sigma_2 \in \left(2^{AP}\right)^\omega$:

$\sigma_1 \triangleq \sigma_2$ iff there exists an infinite word $A_0\, A_1\, A_2 \ldots \in \left(2^{AP}\right)^\omega$ s.t. σ_1 and σ_2 are in $A_0{}^+ A_1{}^+ A_2{}^+ \ldots$

Let $E \subseteq \left(2^{AP}\right)^\omega$ be an LT property. E is called stutter-insensitive iff for all $\sigma_1, \sigma_2 \in \left(2^{AP}\right)^\omega$:

if $\sigma_1 \in E$ and $\sigma_1 \triangleq \sigma_2$ then $\sigma_2 \in E$

Example: if φ is an $\mathbf{LTL}_{\setminus\bigcirc}$ formula then $E = \mathit{Words}(\varphi)$ is stutter-insensitive

Stutter-insensitive LT properties

Let $\mathcal{T}_1$, $\mathcal{T}_2$ be two TS and E a stutter-insensitive LT-property. Then:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \text{ and } \mathcal{T}_2 \models E \quad \text{implies} \quad \mathcal{T}_1 \models E$$

Stutter-insensitive LT properties

Let $\mathcal{T}_1$, $\mathcal{T}_2$ be two TS and E a stutter-insensitive LT-property. Then:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \text{ and } \mathcal{T}_2 \models E \ \text{ implies } \ \mathcal{T}_1 \models E$$

Let $\mathcal{T}_1$, $\mathcal{T}_2$ be two TS and φ an $\mathbf{LTL}_{\setminus\bigcirc}$ formula.

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \text{ and } \mathcal{T}_2 \models \varphi \ \text{ implies } \ \mathcal{T}_1 \models \varphi$$

Stutter-insensitive LT properties

Let $\mathcal{T}_1$, $\mathcal{T}_2$ be two TS and E a stutter-insensitive LT-property. Then:

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \text{ and } \mathcal{T}_2 \models E \quad \text{implies} \quad \mathcal{T}_1 \models E$$

Let $\mathcal{T}_1$, $\mathcal{T}_2$ be two TS and φ an $\mathbf{LTL}_{\setminus \bigcirc}$ formula.

$$\mathcal{T}_1 \trianglelefteq \mathcal{T}_2 \text{ and } \mathcal{T}_2 \models \varphi \quad \text{implies} \quad \mathcal{T}_1 \models \varphi$$

remind: if φ is an $\mathbf{LTL}_{\setminus \bigcirc}$ formula then

$$E = \mathit{Words}(\varphi) \text{ is stutter-insensitive}$$

Overview

- Introduction
- Modelling parallel systems
- Linear Time Properties
- Regular Properties
- Linear Temporal Logic (LTL)
- Computation-Tree Logic (CTL)
- **Equivalences and Abstraction**
  - bisimulation, CTL/CTL*-equivalence
  - computing the bisimulation quotient
  - abstraction stutter steps
    - stutter LT relations
    - stutter bisimulation ⟵
  - simulation relations

Stutter bisimulation

Stutter bisimulation

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS, possibly with terminal states.

Stutter bisimulation

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS,
possibly with terminal states.

A *stutter bisimulation* for $\mathcal{T}$ is ....

Stutter bisimulation

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS,
possibly with terminal states.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S s.t.

Stutter bisimulation

Let $\mathcal{T} = (S, Act, \to, S_0, AP, L)$ be a TS,
possibly with terminal states.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) labeling condition

(2) simulation condition up to stuttering
"s_2 can mimick all transitions of of s_1"

(3) simulation condition up to stuttering
"s_1 can mimick all transitions of of s_2"

Stutter bisimulation

Let $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$ be a TS,
possibly with terminal states.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) labeling condition: $L(s_1) = L(s_2)$

(2) simulation condition up to stuttering
"s_2 can mimick all transitions of of s_1"

(3) simulation condition up to stuttering
"s_1 can mimick all transitions of of s_2"

Stutter bisimulation

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS, possibly with terminal states.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S s.t. for all $(s_1, s_2) \in \mathcal{R}$:

(1) labeling condition: $L(s_1) = L(s_2)$

(2) simulation condition up to stuttering
"s_2 can mimick all transitions of of s_1"

(3) simulation condition up to stuttering
"s_1 can mimick all transitions of of s_2"

Simulation condition

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S s.t. for all $(s_1, s_2) \in \mathcal{R}$:

$\vdots$ $\qquad\qquad$ $\vdots$

(2) simulation condition up to stuttering

$$\begin{tikzcd}
s_1 \arrow[r, phantom, "-\mathcal{R}-"] \arrow[d] & s_2 \\
s_1' &
\end{tikzcd}$$

Simulation condition

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S s.t. for all $(s_1, s_2) \in \mathcal{R}$:

$\vdots \qquad \vdots$

(2) simulation condition up to stuttering

$$\begin{tikzcd}
s_1 \arrow[r, dash, "\mathcal{R}"] \arrow[d] & s_2 \\
s_1' &
\end{tikzcd}$$

with $(s_1', s_2) \notin \mathcal{R}$

Simulation condition

STUTTER5.4-DEF-STUTTER-BIS

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S s.t. for all $(s_1, s_2) \in \mathcal{R}$:

$\vdots \qquad\qquad \vdots$

(2) simulation condition up to stuttering

$$\begin{tikzcd}
s_1 \arrow[r, phantom, "\text{-}\mathcal{R}\text{-}"] \arrow[d] & s_2 \\
s_1' &
\end{tikzcd}$$

can be completed to

$$\begin{tikzcd}
s_1 \arrow[r, phantom, "\text{-}\mathcal{R}\text{-}"] \arrow[dddd] & s_2 \arrow[d] \\
& u_1 \arrow[d] \\
& \vdots \arrow[d] \\
& u_n \arrow[d] \\
s_1' \arrow[r, phantom, "\text{-}\mathcal{R}\text{-}"] & s_2'
\end{tikzcd}$$

with $(s_1', s_2) \notin \mathcal{R}$

Simulation condition

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S s.t. for all $(s_1, s_2) \in \mathcal{R}$:

$\vdots$ $\vdots$

(2) simulation condition up to stuttering

$$\begin{tikzcd}
s_1 \arrow[r, dash, "\mathcal{R}" description] \arrow[d] & s_2 \\
s_1' &
\end{tikzcd}$$

with $(s_1', s_2) \notin \mathcal{R}$

can be completed to

$$\begin{tikzcd}
s_1 \arrow[r, dash, "\mathcal{R}" description] \arrow[dddd] & s_2 \arrow[d] & \\
 & u_1 \arrow[d] \arrow[dr, dashed, no head] & \\
 & \vdots \arrow[d] & s_1 \text{ -}\mathcal{R}\text{- } u_i \\
 & u_n \arrow[d] \arrow[ur, dashed, no head] & \\
s_1' \arrow[r, dash, "\mathcal{R}" description] & s_2' &
\end{tikzcd}$$

Stutter bisimulation for a TS

Let $\mathcal{T}$ be a transition system wih state space S.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S such that for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L(s_1) = L(s_2)$

(2) for each transition $s_1 \rightarrow s_1'$ with $(s_1', s_2) \notin \mathcal{R}$ there exists a path fragment $s_2\, u_1\, u_2\, \ldots\, u_n\, s_2'$ s.t. ...

(3) ...

Stutter bisimulation for a TS

Let $\mathcal{T}$ be a transition system wih state space S.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S such that for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L(s_1) = L(s_2)$

(2) for each transition $s_1 \rightarrow s_1'$ with $(s_1', s_2) \notin \mathcal{R}$ there exists a path fragment $s_2\, u_1\, u_2\, \ldots\, u_n\, s_2'$ s.t. $n \geq 0$ and $(s_1, u_i) \in \mathcal{R}$ for $1 \leq i \leq n$

(3) …

Stutter bisimulation for a TS

Let $\mathcal{T}$ be a transition system wih state space S.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S such that for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L(s_1) = L(s_2)$

(2) for each transition $s_1 \rightarrow s_1'$ with $(s_1', s_2) \notin \mathcal{R}$ there exists a path fragment $s_2\, u_1\, u_2\, \ldots\, u_n\, s_2'$ s.t. $n \geq 0$ and $(s_1, u_i) \in \mathcal{R}$ for $1 \leq i \leq n$

(3) symmetric condition

Stutter bisimulation for a TS

Let $\mathcal{T}$ be a transition system wih state space S.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S such that for all $(s_1, s_2) \in \mathcal{R}$:

(1) $L(s_1) = L(s_2)$

(2) for each transition $s_1 \rightarrow s_1'$ with $(s_1', s_2) \notin \mathcal{R}$
there exists a path fragment $s_2\, u_1\, u_2\, \ldots\, u_n\, s_2'$
s.t. $n \geq 0$ and $(s_1, u_i) \in \mathcal{R}$ for $1 \leq i \leq n$

(3) for each transition $s_2 \rightarrow s_2'$ with $(s_1, s_2') \notin \mathcal{R}$
there exists a path fragment $s_1\, v_1\, v_2\, \ldots\, v_n\, s_1'$
s.t. $n \geq 0$ and $(v_i, s_2) \in \mathcal{R}$ for $1 \leq i \leq n$

Stutter bisimulation equivalence $\approx_{\mathcal{T}}$

Stutter bisimulation equivalence $\approx_{\mathcal{T}}$

Let $\mathcal{T}$ be a transition system wih state space S.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S such that for all $(s_1, s_2) \in \mathcal{R}$:

(1) labeling condition

(2) and (3) mutual simulation condition

Stutter bisimulation equivalence $\approx_{\mathcal{T}}$

Let $\mathcal{T}$ be a transition system wih state space S.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S such that for all $(s_1, s_2) \in \mathcal{R}$:

(1) labeling condition

(2) and (3) mutual simulation condition

stutter bisimulation equivalence $\approx_{\mathcal{T}}$:

Stutter bisimulation equivalence $\approx_{\mathcal{T}}$

Let $\mathcal{T}$ be a transition system wih state space S.

A *stutter bisimulation* for $\mathcal{T}$ is a binary relation $\mathcal{R}$ on S such that for all $(s_1, s_2) \in \mathcal{R}$:

(1) labeling condition

(2) and (3) mutual simulation condition

stutter bisimulation equivalence $\approx_{\mathcal{T}}$:

$s_1 \approx_{\mathcal{T}} s_2$ iff there exists a stutter bisimulation $\mathcal{R}$ for $\mathcal{T}$ s.t. $(s_1, s_2) \in \mathcal{R}$

$\approx_{\mathcal{T}}$ is an equivalence

$\approx_{\mathcal{T}}$ is an equivalence

symmetry: if $s_1 \approx_{\mathcal{T}} s_2$ then $s_2 \approx_{\mathcal{T}} s_1$

$\approx_{\mathcal{T}}$ is an equivalence

symmetry: if $s_1 \approx_{\mathcal{T}} s_2$ then $s_2 \approx_{\mathcal{T}} s_1$

proof:

if $\mathcal{R}$ is a stutter bisimulation with $(s_1, s_2) \in \mathcal{R}$ then

$$\mathcal{R}^{-1} = \{(t_2, t_1) : (t_1, t_2) \in \mathcal{R}\}$$

is a stutter bisimulation that contains (s_2, s_1).

$\approx_{\mathcal{T}}$ is an equivalence

symmetry: if $s_1 \approx_{\mathcal{T}} s_2$ then $s_2 \approx_{\mathcal{T}} s_1$

reflexivity: $s \approx_{\mathcal{T}} s$ for all states s

$\approx_{\mathcal{T}}$ is an equivalence

symmetry: if $s_1 \approx_{\mathcal{T}} s_2$ then $s_2 \approx_{\mathcal{T}} s_1$

reflexivity: $s \approx_{\mathcal{T}} s$ for all states s

proof:

$\mathcal{R} = \{(s, s) : s \in S\}$ is a stutter bisimulation

$\approx_{\mathcal{T}}$ is an equivalence

symmetry: if $s_1 \approx_{\mathcal{T}} s_2$ then $s_2 \approx_{\mathcal{T}} s_1$

reflexivity: $s \approx_{\mathcal{T}} s$ for all states s

transitivity: $s_1 \approx_{\mathcal{T}} s_2$ and $s_2 \approx_{\mathcal{T}} s_3$ implies $s_1 \approx_{\mathcal{T}} s_3$

$\approx_{\mathcal{T}}$ is an equivalence

symmetry: if $s_1 \approx_{\mathcal{T}} s_2$ then $s_2 \approx_{\mathcal{T}} s_1$

reflexivity: $s \approx_{\mathcal{T}} s$ for all states s

transitivity: $s_1 \approx_{\mathcal{T}} s_2$ and $s_2 \approx_{\mathcal{T}} s_3$ implies $s_1 \approx_{\mathcal{T}} s_3$

Proof: Let $\mathcal{R}_{1,2}$ and $\mathcal{R}_{2,3}$ be stutter bisimulations s.t.

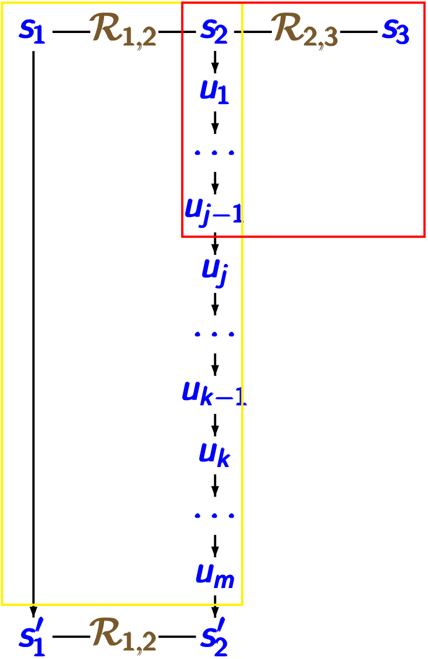
$$(s_1, s_2) \in \mathcal{R}_{1,2}, \ (s_2, s_3) \in \mathcal{R}_{2,3}$$

Show that $\mathcal{R} = \mathcal{R}_{1,2} \circ \mathcal{R}_{2,3}$ is a stutter bisimulation.

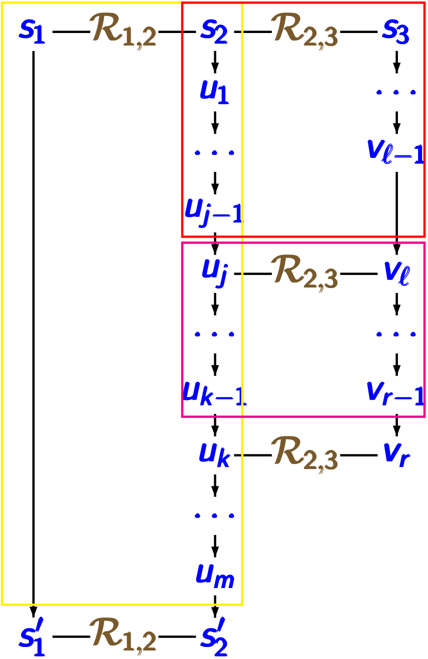
$$\begin{tikzcd}
s_1 \arrow[r, dash] \arrow[d] & \mathcal{R}_{1,2} \arrow[r, dash] & s_2 \arrow[r, dash] & \mathcal{R}_{2,3} \arrow[r, dash] & s_3 \\
s_1' & & & &
\end{tikzcd}$$

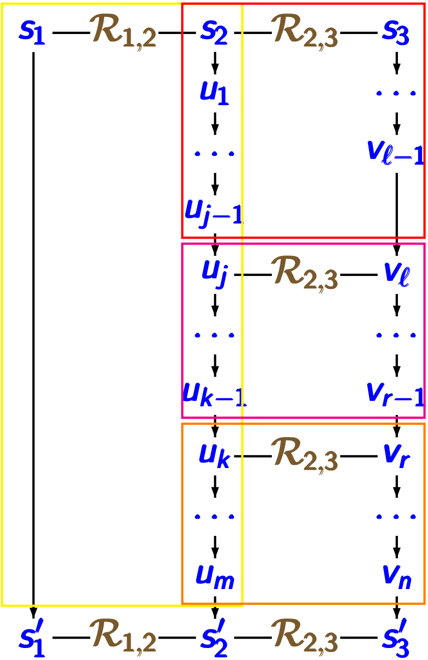
$$\begin{tikzcd}
s_1 \arrow[r, dash] \arrow[dddddddddd] & \mathcal{R}_{1,2} \arrow[r, dash] & s_2 \arrow[r, dash] \arrow[d] & \mathcal{R}_{2,3} \arrow[r, dash] & s_3 \\
 & & u_1 \arrow[d] & & \\
 & & \cdots \arrow[d] & & \\
 & & u_{j-1} \arrow[d] & & \\
 & & u_j \arrow[d] & & \\
 & & \cdots \arrow[d] & & \\
 & & u_{k-1} \arrow[d] & & \\
 & & u_k \arrow[d] & & \\
 & & \cdots \arrow[d] & & \\
 & & u_m \arrow[d] & & \\
s'_1 \arrow[r, dash] & \mathcal{R}_{1,2} \arrow[r, dash] & s'_2 & &
\end{tikzcd}$$

$$\begin{tikzcd}
s_1 \arrow[r, dash] \arrow[dddddddddd] & \mathcal{R}_{1,2} \arrow[r, dash] & s_2 \arrow[r, dash] \arrow[d] & \mathcal{R}_{2,3} \arrow[r, dash] & s_3 \\
 & & u_1 \arrow[d] & & \\
 & & \cdots \arrow[d] & & \\
 & & u_{j-1} \arrow[d] & & \\
 & & u_j \arrow[d] & & \\
 & & \cdots \arrow[d] & & \\
 & & u_{k-1} \arrow[d] & & \\
 & & u_k \arrow[d] & & \\
 & & \cdots \arrow[d] & & \\
 & & u_m \arrow[d] & & \\
s'_1 \arrow[r, dash] & \mathcal{R}_{1,2} \arrow[r, dash] & s'_2 & &
\end{tikzcd}$$


s_1 — $\mathcal{R}_{1,2}$ — s_2 — $\mathcal{R}_{2,3}$ — s_3
u_1
$\dots$
u_{j-1}
u_j
$\dots$
u_{k-1}
u_k
$\dots$
u_m
s_1' — $\mathcal{R}_{1,2}$ — s_2'

$$\begin{tikzcd}
s_1 \arrow[r, dash] \arrow[dddddddddd] & \mathcal{R}_{1,2} \arrow[r, dash] & s_2 \arrow[r, dash] \arrow[d] & \mathcal{R}_{2,3} \arrow[r, dash] & s_3 \arrow[d] \\
 & & u_1 \arrow[d] & & \cdots \arrow[d] \\
 & & \cdots \arrow[d] & & v_{\ell-1} \arrow[dd] \\
 & & u_{j-1} \arrow[d] & & \\
 & & u_j \arrow[r, dash] \arrow[d] & \mathcal{R}_{2,3} \arrow[r, dash] & v_\ell \\
 & & \cdots \arrow[d] & & \\
 & & u_{k-1} \arrow[d] & & \\
 & & u_k \arrow[d] & & \\
 & & \cdots \arrow[d] & & \\
 & & u_m \arrow[d] & & \\
s_1' \arrow[r, dash] & \mathcal{R}_{1,2} \arrow[r, dash] & s_2' & &
\end{tikzcd}$$


s_1 — $\mathcal{R}_{1,2}$ — s_2 — $\mathcal{R}_{2,3}$ — s_3
u_1
$\dots$
u_{j-1}
$\dots$
$v_{\ell-1}$
u_j — $\mathcal{R}_{2,3}$ — v_ℓ
$\dots$
$\dots$
u_{k-1}
v_{r-1}
u_k — $\mathcal{R}_{2,3}$ — v_r
$\dots$
u_m
s_1' — $\mathcal{R}_{1,2}$ — s_2'

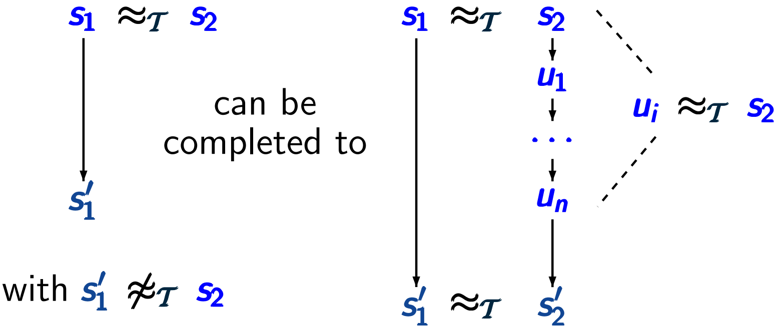

s_1
$\mathcal{R}_{1,2}$
s_2
$\mathcal{R}_{2,3}$
s_3
u_1
$\dots$
u_{j-1}
$\dots$
$v_{\ell-1}$
u_j
$\mathcal{R}_{2,3}$
v_ℓ
$\dots$
$\dots$
u_{k-1}
v_{r-1}
u_k
$\mathcal{R}_{2,3}$
v_r
$\dots$
$\dots$
u_m
v_n
s'_1
$\mathcal{R}_{1,2}$
s'_2
$\mathcal{R}_{2,3}$
s'_3

Stutter bisimulation equivalence

$\approx_{\mathcal{T}}$ is an equivalence on state space S of $\mathcal{T}$ such that for all states s_1, s_2 with $s_1 \approx_{\mathcal{T}} s_2$:

(1) $L(s_1) = L(s_2)$

(2) simulation condition up to stuttering



$s_1 \approx_{\mathcal{T}} s_2$, $s_1 \to s_1'$

can be completed to

with $s_1' \not\approx_{\mathcal{T}} s_2$

$s_1 \approx_{\mathcal{T}} s_2$, $s_2 \to u_1 \to \cdots \to u_n \to s_2'$, $s_1 \to s_1'$

$u_i \approx_{\mathcal{T}} s_2$

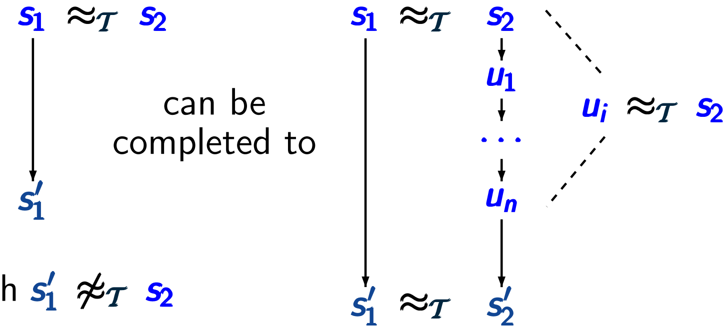
$s_1' \approx_{\mathcal{T}} s_2'$

Stutter bisimulation equivalence

$\approx_{\mathcal{T}}$ is the coarsest equivalence on state space S of $\mathcal{T}$ such that for all states s_1, s_2 with $s_1 \approx_{\mathcal{T}} s_2$:

(1) $L(s_1) = L(s_2)$

(2) simulation condition up to stuttering

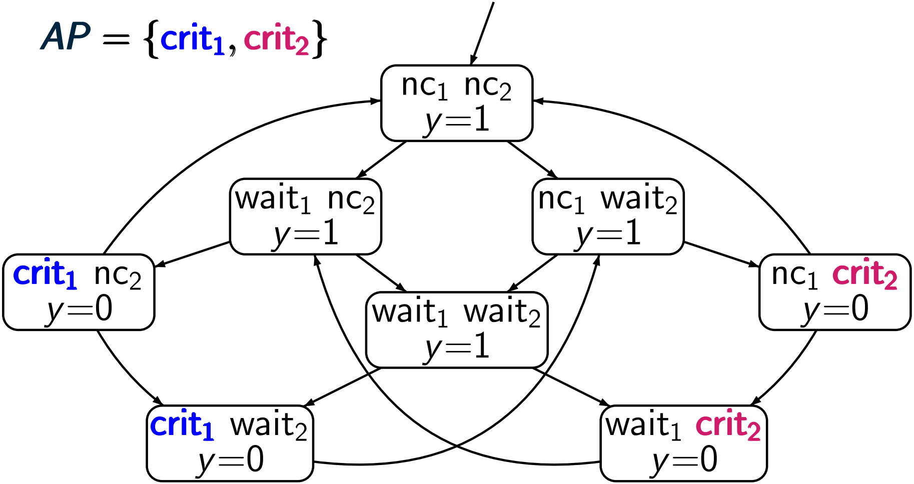


can be completed to

with $s_1' \not\approx_{\mathcal{T}} s_2$

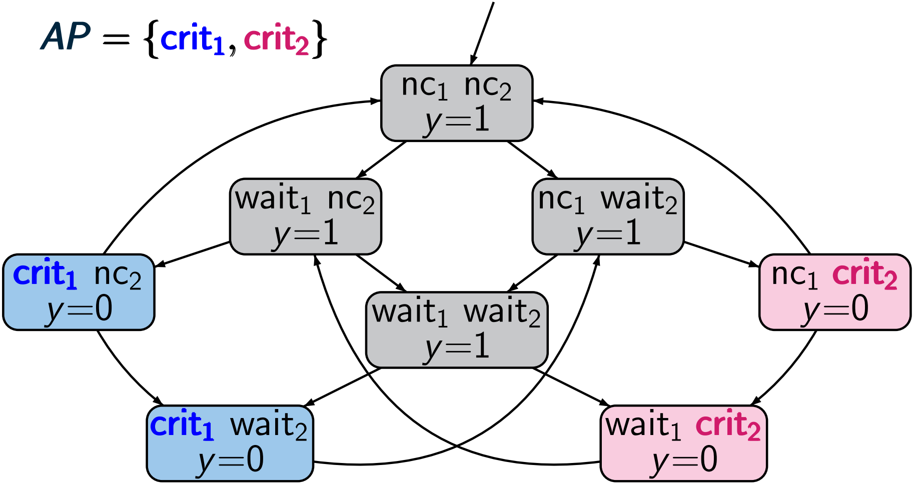
Example: mutual exclusion with semaphore

$AP = \{crit_1, crit_2\}$



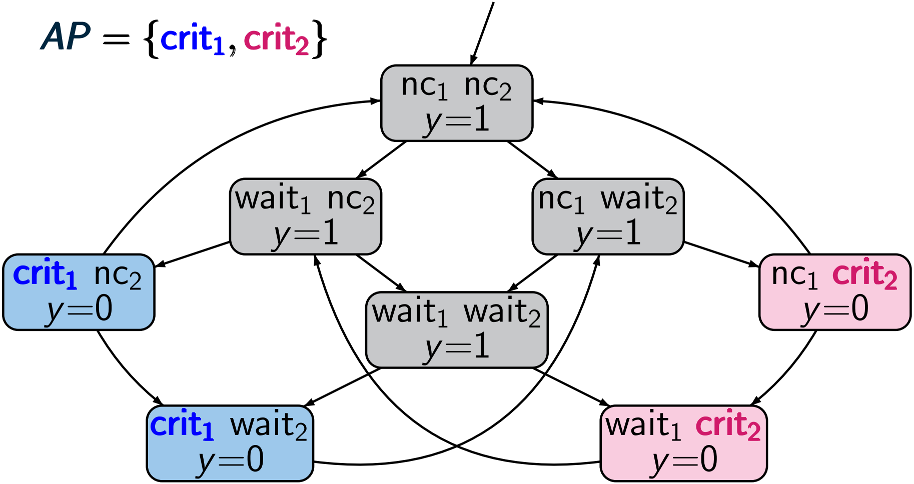
Example: mutual exclusion with semaphore

$AP = \{crit_1, crit_2\}$



Example: mutual exclusion with semaphore

$AP = \{\mathbf{crit_1}, \mathbf{crit_2}\}$



stutter bisimulation with three equivalence classes

Peterson algorithm

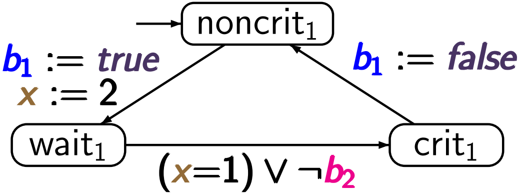
protocol for P_1

```
LOOP FOREVER
    noncritical section
    $b_1 := true$; $x := 2$
    AWAIT $(x{=}1) \lor \neg b_2$
    critical section
    $b_1 := false$
END LOOP
```

Peterson algorithm

protocol for P_1

```
LOOP FOREVER
    noncritical section
    $b_1 := true$; $x := 2$
    AWAIT $(x=1) \vee \neg b_2$
    critical section
    $b_1 := false$
END LOOP
```



Peterson algorithm

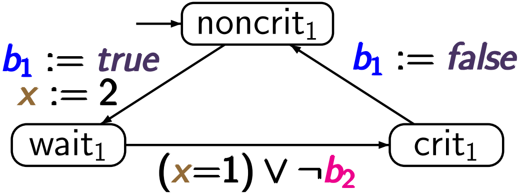
STUTTER5.4-7

protocol for P_1

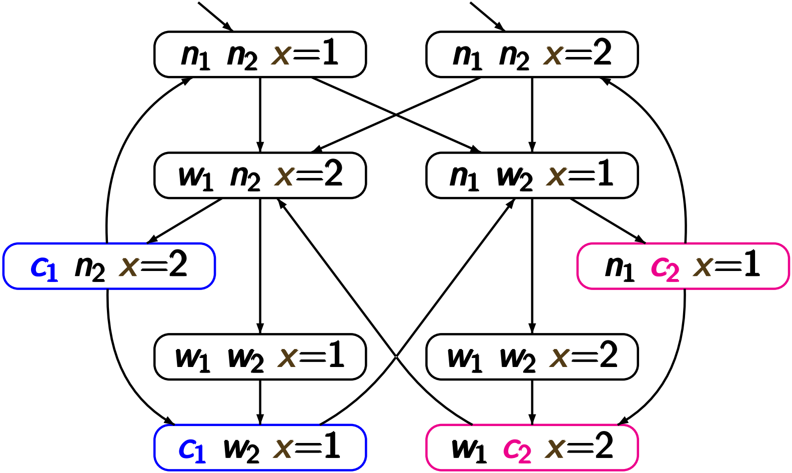
```
LOOP FOREVER
    noncritical section
    $b_1 := true$; $x := 2$
    AWAIT $(x=1) \lor \neg b_2$
    critical section
    $b_1 := false$
END LOOP
```

protocol for P_2

```
LOOP FOREVER
    noncritical section
    $b_2 := true$; $x := 1$
    AWAIT $(x=2) \lor \neg b_1$
    critical section
    $b_2 := false$
END LOOP
```

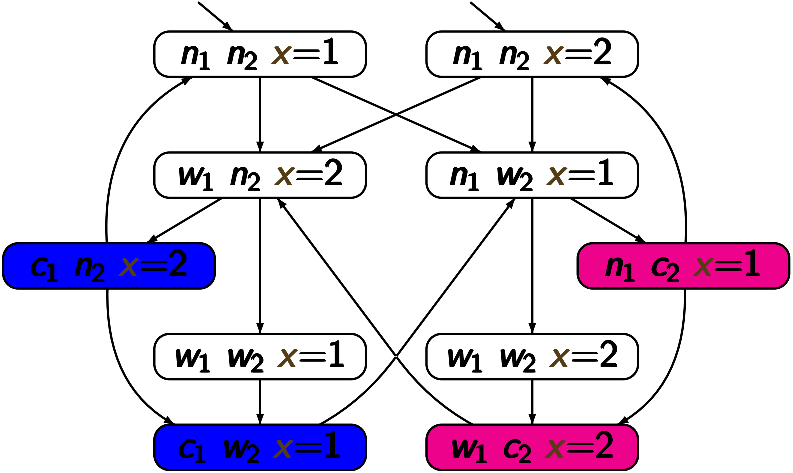


TS for the Peterson algorithm



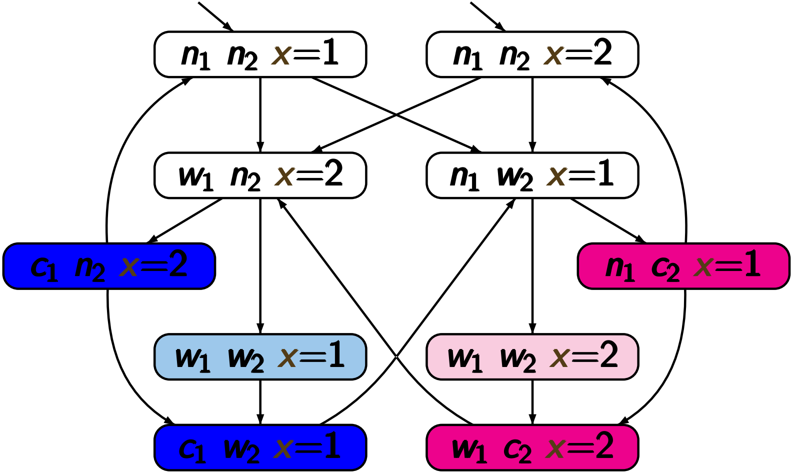
$AP = \{crit_1, crit_2\}$

TS for the Peterson algorithm



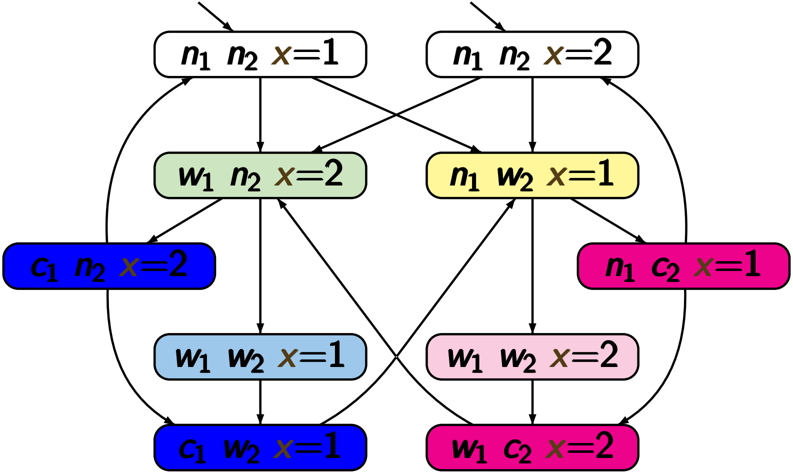
$AP = \{crit_1, crit_2\}$

TS for the Peterson algorithm



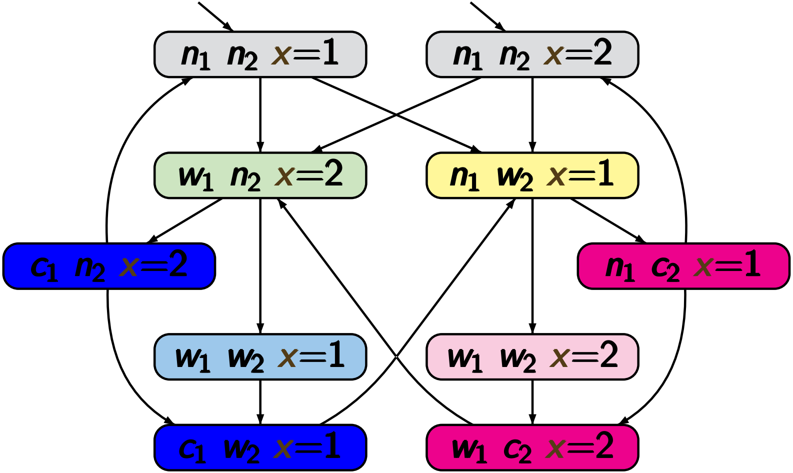
$AP = \{crit_1, crit_2\}$

TS for the Peterson algorithm



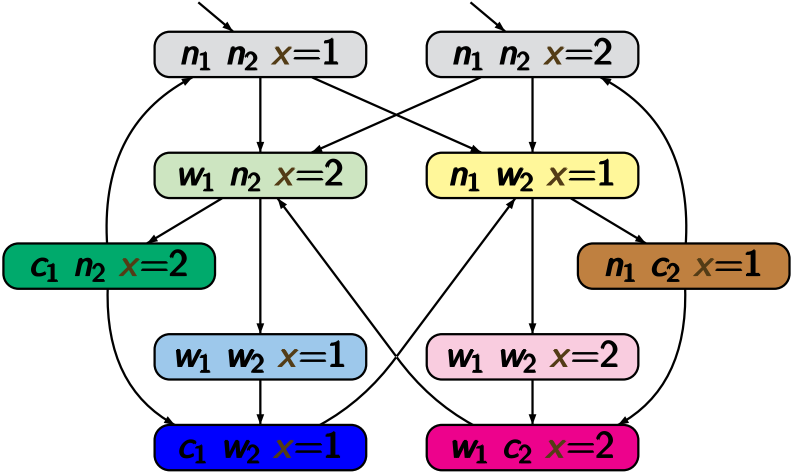
$AP = \{\text{crit}_1, \text{crit}_2\}$

TS for the Peterson algorithm



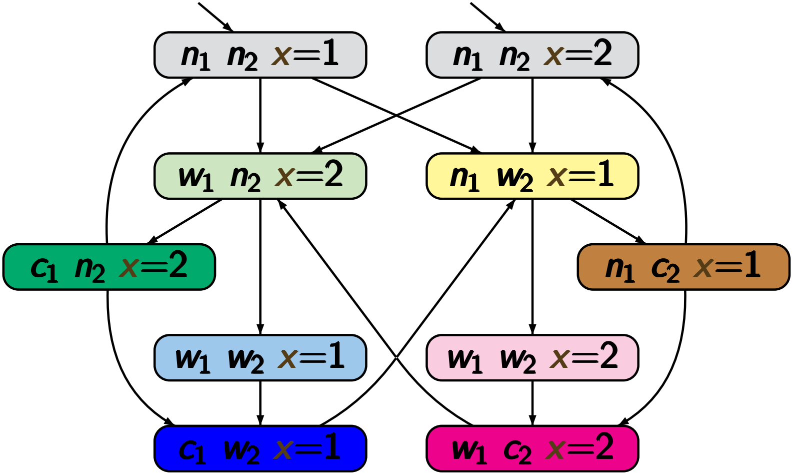
$AP = \{crit_1, crit_2\}$

TS for the Peterson algorithm



$AP = \{\text{crit}_1, \text{crit}_2\}$

TS for the Peterson algorithm



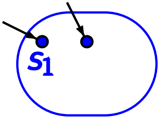
$AP = \{\text{crit}_1, \text{crit}_2\}$

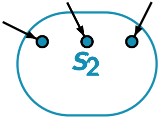
9 stutter bisimulation equivalence classes

Stutter bisimulation equivalence for two TS

Stutter bisimulation equivalence for two TS

transition system $\mathcal{T}_1$ transition system $\mathcal{T}_2$

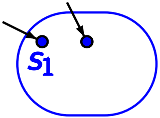


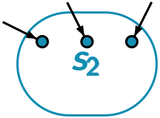


state space S_1 state space S_2

Stutter bisimulation equivalence for two TS

transition system $\mathcal{T}_1$ transition system $\mathcal{T}_2$





state space S_1 state space S_2

$\mathcal{T}_1 \approx \mathcal{T}_2$ iff there exists a stutter bisimulation $\mathcal{R}$ for $\mathcal{T} = \mathcal{T}_1 \uplus \mathcal{T}_2$ such that

Stutter bisimulation equivalence for two TS

transition system $\mathcal{T}_1$ transition system $\mathcal{T}_2$

state space S_1 state space S_2

$\mathcal{T}_1 \approx \mathcal{T}_2$ iff there exists a stutter bisimulation $\mathcal{R}$ for $\mathcal{T} = \mathcal{T}_1 \uplus \mathcal{T}_2$ such that

$\forall$ initial states s_1 of $\mathcal{T}_1$ $\exists$ initial state s_2 of $\mathcal{T}_2$ s.t. $s_1 \approx_{\mathcal{T}} s_2$

$\forall$ initial states s_2 of $\mathcal{T}_2$ $\exists$ initial state s_1 of $\mathcal{T}_1$ s.t. $s_1 \approx_{\mathcal{T}} s_2$

Stutter bisimulation equivalence for two TS

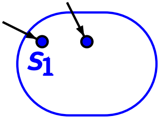
transition system $\mathcal{T}_1$ transition system $\mathcal{T}_2$

state space S_1 state space S_2

$\mathcal{T}_1 \approx \mathcal{T}_2$ iff there exists a stutter bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$

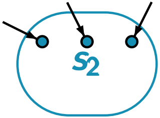
Stutter bisimulation equivalence for two TS

transition system $\mathcal{T}_1$



state space S_1

transition system $\mathcal{T}_2$



state space S_2

$\mathcal{T}_1 \approx \mathcal{T}_2$ iff there exists a stutter bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$, i.e., $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

Stutter bisimulation equivalence for two TS

transition system $\mathcal{T}_1$ transition system $\mathcal{T}_2$

state space S_1 state space S_2

$\mathcal{T}_1 \approx \mathcal{T}_2$ iff there exists a stutter bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$, i.e., $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

(1) if $(s_1, s_2) \in \mathcal{R}$ then $L_1(s_1) = L_2(s_2)$

Stutter bisimulation equivalence for two TS

transition system $\mathcal{T}_1$ transition system $\mathcal{T}_2$

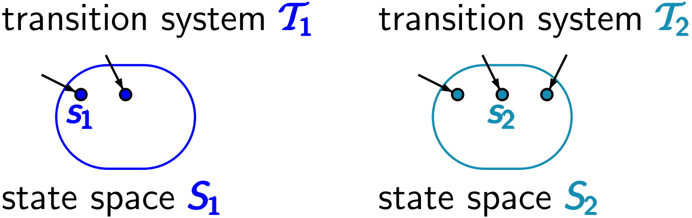
state space S_1 state space S_2

$\mathcal{T}_1 \approx \mathcal{T}_2$ iff there exists a stutter bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$, i.e., $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

(1) if $(s_1, s_2) \in \mathcal{R}$ then $L_1(s_1) = L_2(s_2)$

(2) and (3) $\ldots$

Stutter bisimulation equivalence for two TS



$\mathcal{T}_1 \approx \mathcal{T}_2$ iff there exists a stutter bisimulation $\mathcal{R}$ for $(\mathcal{T}_1, \mathcal{T}_2)$, i.e., $\mathcal{R} \subseteq S_1 \times S_2$ s.t.

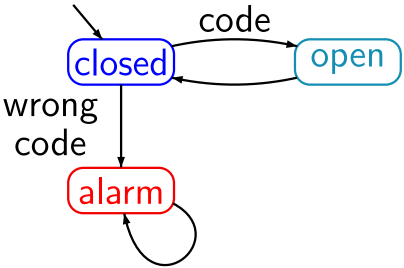
(1) if $(s_1, s_2) \in \mathcal{R}$ then $L_1(s_1) = L_2(s_2)$

(2) and (3) ...

(I) $\forall$ initial state s_1 of $\mathcal{T}_1$ $\exists$ initial state s_2 of $\mathcal{T}_2$ with $(s_1, s_2) \in \mathcal{R}$, and vice versa

Example: door opener

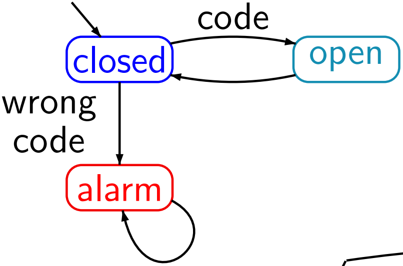
abstract model $\mathcal{T}_1$



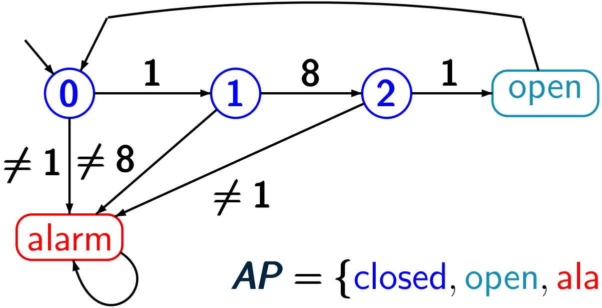
$AP = \{\text{closed}, \text{open}, \text{alarm}\}$

Example: door opener with code no. 181

abstract model $\mathcal{T}_1$



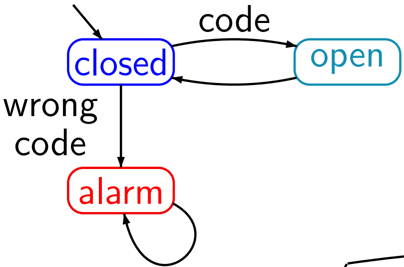
refinement TS $\mathcal{T}_2$



$AP = \{\text{closed}, \text{open}, \text{alarm}\}$

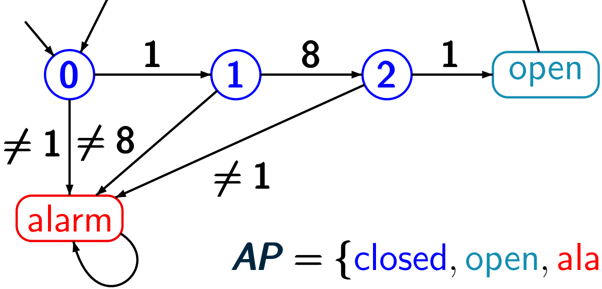
Example: door opener with code no. 181

abstract model $\mathcal{T}_1$



$\mathcal{T}_1 \not\approx \mathcal{T}_2$

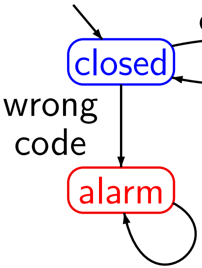
refinement TS $\mathcal{T}_2$



$AP = \{\text{closed}, \text{open}, \text{alarm}\}$

Example: door opener with code no. 181

abstract model $\mathcal{T}_1$

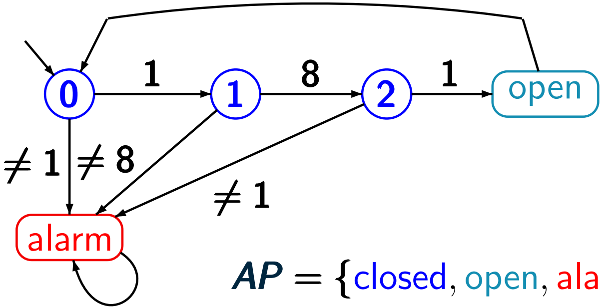


$$\mathcal{T}_1 \not\sim \mathcal{T}_2$$

abstraction from stutter steps:

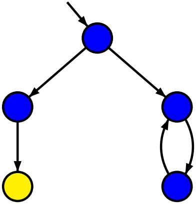
$$\mathcal{T}_1 \approx \mathcal{T}_2$$

refinement TS $\mathcal{T}_2$

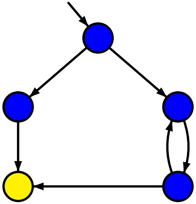


$AP = \{$closed, open, alarm$\}$

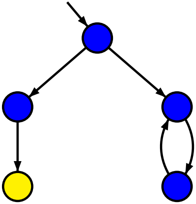
Correct or wrong?



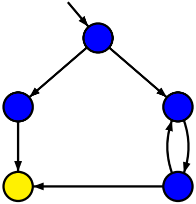
$\mathcal{T}_1 \approx \mathcal{T}_2$



Correct or wrong?

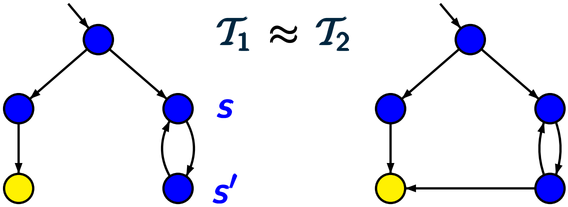


$\mathcal{T}_1 \approx \mathcal{T}_2$



wrong

Correct or wrong?

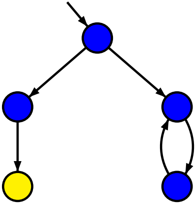


$\mathcal{T}_1 \approx \mathcal{T}_2$

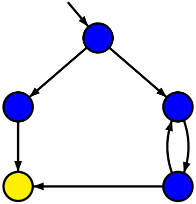
wrong

$\mathcal{T}_2$ does not contain an equivalent state to s and s'

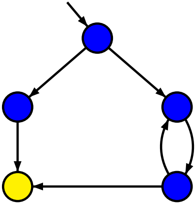
Correct or wrong?



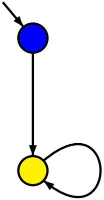
$\mathcal{T}_1 \approx \mathcal{T}_2$



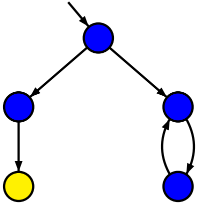
wrong



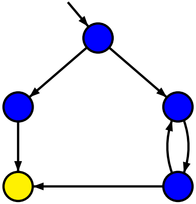
$\mathcal{T}_1 \approx \mathcal{T}_2$



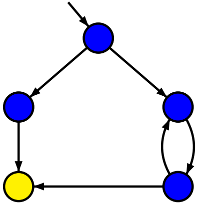
Correct or wrong?



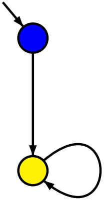
$\mathcal{T}_1 \approx \mathcal{T}_2$



wrong

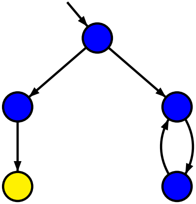


$\mathcal{T}_1 \approx \mathcal{T}_2$

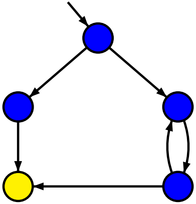


correct

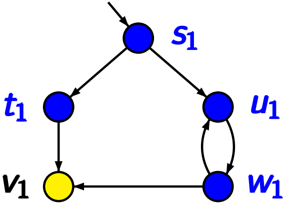
Correct or wrong?



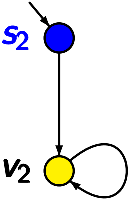
$\mathcal{T}_1 \approx \mathcal{T}_2$



wrong



$\mathcal{T}_1 \approx \mathcal{T}_2$



correct

stutter bisimulation for $(\mathcal{T}_1, \mathcal{T}_2)$:

$$\{(s_1, s_2), (t_1, s_2), (u_1, s_2), (w_1, s_2), (v_1, v_2)\}$$

Correct or wrong?

If $s_1 \sim_{\mathcal{T}} s_2$ then $s_1 \approx_{\mathcal{T}} s_2$

remind: $\sim_{\mathcal{T}}$ bisimulation equivalence for $\mathcal{T}$

$\approx_{\mathcal{T}}$ stutter bisimulation equivalence for $\mathcal{T}$

Correct or wrong?

If $s_1 \sim_{\mathcal{T}} s_2$ then $s_1 \approx_{\mathcal{T}} s_2$

correct

remind: $\sim_{\mathcal{T}}$ bisimulation equivalence for $\mathcal{T}$

$\approx_{\mathcal{T}}$ stutter bisimulation equivalence for $\mathcal{T}$

Correct or wrong?

If $s_1 \sim_{\mathcal{T}} s_2$ then $s_1 \approx_{\mathcal{T}} s_2$

correct

as $\sim_{\mathcal{T}}$ is a stutter bisimulation for $\mathcal{T}$

remind: $\sim_{\mathcal{T}}$ bisimulation equivalence for $\mathcal{T}$

$\approx_{\mathcal{T}}$ stutter bisimulation equivalence for $\mathcal{T}$

Correct or wrong?

If $s_1 \sim_{\mathcal{T}} s_2$ then $s_1 \approx_{\mathcal{T}} s_2$

correct

as $\sim_{\mathcal{T}}$ is a stutter bisimulation for $\mathcal{T}$

If $s_1 \approx_{\mathcal{T}} s_2$ then $s_1 \sim_{\mathcal{T}} s_2$

Correct or wrong?

If $s_1 \sim_{\mathcal{T}} s_2$ then $s_1 \approx_{\mathcal{T}} s_2$

correct

as $\sim_{\mathcal{T}}$ is a stutter bisimulation for $\mathcal{T}$

If $s_1 \approx_{\mathcal{T}} s_2$ then $s_1 \sim_{\mathcal{T}} s_2$

wrong

Correct or wrong?

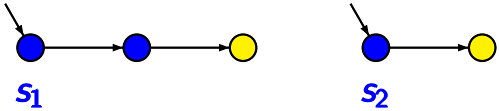
If $s_1 \sim_{\mathcal{T}} s_2$ then $s_1 \approx_{\mathcal{T}} s_2$

correct

as $\sim_{\mathcal{T}}$ is a stutter bisimulation for $\mathcal{T}$

If $s_1 \approx_{\mathcal{T}} s_2$ then $s_1 \sim_{\mathcal{T}} s_2$

wrong, e.g.:



Correct or wrong?

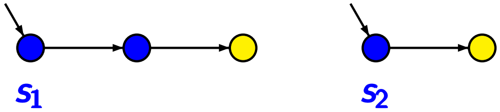
If $s_1 \sim_{\mathcal{T}} s_2$ then $s_1 \approx_{\mathcal{T}} s_2$

correct

as $\sim_{\mathcal{T}}$ is a stutter bisimulation for $\mathcal{T}$

If $s_1 \approx_{\mathcal{T}} s_2$ then $s_1 \sim_{\mathcal{T}} s_2$

wrong, e.g.:



$s_1 \approx_{\mathcal{T}} s_2$

$s_1 \not\sim_{\mathcal{T}} s_2$

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct, as $\approx_{\mathcal{T}}$ is a bisimulation for $\mathcal{T}$

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct, as $\approx_{\mathcal{T}}$ is a bisimulation for $\mathcal{T}$

(1) labeling condition: $\surd$

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct, as $\approx_{\mathcal{T}}$ is a bisimulation for $\mathcal{T}$

(1) labeling condition: $\surd$

(2) Suppose $s_1 \longrightarrow s_1'$.

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct, as $\approx_{\mathcal{T}}$ is a bisimulation for $\mathcal{T}$

(1) labeling condition: $\surd$

(2) Suppose $s_1 \rightarrow s_1'$. Then: $L(s_1) \neq L(s_1')$

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct, as $\approx_{\mathcal{T}}$ is a bisimulation for $\mathcal{T}$

(1) labeling condition: $\surd$

(2) Suppose $s_1 \rightarrow s_1'$. Then: $L(s_1) \neq L(s_1')$

$$\Longrightarrow \quad s_1 \not\approx_{\mathcal{T}} s_1'$$

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct, as $\approx_{\mathcal{T}}$ is a bisimulation for $\mathcal{T}$

(1) labeling condition: $\surd$

(2) Suppose $s_1 \rightarrow s_1'$. Then: $L(s_1) \neq L(s_1')$

$$\Longrightarrow \quad s_1 \not\approx_{\mathcal{T}} s_1'$$

$\Longrightarrow$ there is a path fragment $s_2 u_1 \ldots u_m s_2'$ with $m \geq 0$ and $s_1 \approx_{\mathcal{T}} u_i \wedge s_1' \approx_{\mathcal{T}} s_2'$

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct, as $\approx_{\mathcal{T}}$ is a bisimulation for $\mathcal{T}$

(1) labeling condition: $\surd$

(2) Suppose $s_1 \rightarrow s_1'$. Then: $L(s_1) \neq L(s_1')$

$$\Longrightarrow \quad s_1 \not\approx_{\mathcal{T}} s_1'$$

$\Longrightarrow$ there is a path fragment $s_2 u_1 \ldots u_m s_2'$ with $m \geq 0$ and $s_1 \approx_{\mathcal{T}} u_i \wedge s_1' \approx_{\mathcal{T}} s_2'$

$$\Longrightarrow \quad m{=}0.$$

Correct or wrong?

Let $\mathcal{T}$ be a transition system without stutter steps.
Then $s_1 \approx_{\mathcal{T}} s_2$ implies $s_1 \sim_{\mathcal{T}} s_2$

correct, as $\approx_{\mathcal{T}}$ is a bisimulation for $\mathcal{T}$

(1) labeling condition: $\surd$

(2) Suppose $s_1 \rightarrow s_1'$. Then: $L(s_1) \neq L(s_1')$

$$\Longrightarrow \quad s_1 \not\approx_{\mathcal{T}} s_1'$$

$\Longrightarrow$ there is a path fragment $s_2 u_1 \ldots u_m s_2'$ with $m \geq 0$ and $s_1 \approx_{\mathcal{T}} u_i \wedge s_1' \approx_{\mathcal{T}} s_2'$

$\Longrightarrow$ $m=0$. Hence: $s_2 \rightarrow s_2'$ and $s_1' \approx_{\mathcal{T}} s_2'$

Stutter bisimulation quotient

Stutter bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

Stutter bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/\!\approx \;=\; (S/\!\approx_{\mathcal{T}}, Act', \rightarrow_{\approx}, S_0', AP, L')$$

Stutter bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/\!\approx \; = \; (S/\!\approx_{\mathcal{T}}, Act', \rightarrow_{\approx}, S'_0, AP, L')$$

- state space: $S/\!\approx_{\mathcal{T}}$ ⟵ set of stutter bisimulation equivalence classes

Stutter bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/{\approx} \;=\; (S/{\approx_{\mathcal{T}}}, Act', \rightarrow_{\approx}, S'_0, AP, L')$$

- state space: $S/{\approx_{\mathcal{T}}}$
- initial states: $S'_0 = \{[s] : s \in S_0\}$

$[s] = [s]_{\approx_{\mathcal{T}}} = \{s' \in S : s \approx_{\mathcal{T}} s'\}$

equivalence class of state s

Stutter bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/\!\approx \;=\; (S/\!\approx_{\mathcal{T}}, Act', \rightarrow_{\approx}, S_0', AP, L')$$

- state space: $S/\!\approx_{\mathcal{T}}$
- initial states: $S_0' = \{[s] : s \in S_0\}$
- labeling: $L'([s]) = L(s)$

$[s] = [s]_{\approx_{\mathcal{T}}} = \{s' \in S : s \approx_{\mathcal{T}} s'\}$

equivalence class of state s

Stutter bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/{\approx} \; = \; (S/{\approx_{\mathcal{T}}}, Act', \rightarrow_{\approx}, S_0', AP, L')$$

- state space: $S/{\approx_{\mathcal{T}}}$
- initial states: $S_0' = \{[s] : s \in S_0\}$
- labeling: $L'([s]) = L(s)$
- transition relation:

$$\frac{s \rightarrow s' \;\wedge\; s \not\approx_{\mathcal{T}} s'}{[s] \rightarrow_{\approx} [s']}$$

Stutter bisimulation quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/\!\approx \;=\; (S/\!\approx_{\mathcal{T}}, Act', \rightarrow_{\approx}, S_0', AP, L')$$

- state space: $S/\!\approx_{\mathcal{T}}$
- initial states: $S_0' = \{[s] : s \in S_0\}$
- labeling: $L'([s]) = L(s)$
- transition relation: ⟵ $\boxed{\text{actions irrelevant}}$

$$\frac{s \rightarrow s' \;\wedge\; s \not\approx_{\mathcal{T}} s'}{[s] \rightarrow_{\approx} [s']}$$

Equivalence of $\mathcal{T}$ and its quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/\!\approx \;=\; (S/\!\approx_{\mathcal{T}}, Act', \rightarrow_{\approx}, S_0', AP, L')$$

where $S_0' = \big\{[s] : s \in S_0\big\}$ and $L'([s]) = L(s)$

transition relation:

$$\frac{s \rightarrow s' \;\wedge\; s \not\approx_{\mathcal{T}} s'}{[s] \rightarrow_{\approx} [s']}$$

Equivalence of $\mathcal{T}$ and its quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/\approx \; = \; (S/\approx_{\mathcal{T}}, Act', \rightarrow_{\approx}, S_0', AP, L')$$

where $S_0' = \big\{ [s] : s \in S_0 \big\}$ and $L'([s]) = L(s)$

transition relation:

$$\frac{s \rightarrow s' \;\wedge\; s \not\approx_{\mathcal{T}} s'}{[s] \rightarrow_{\approx} [s']}$$

$$\boxed{\mathcal{T} \;\approx\; \mathcal{T}/\approx}$$

Equivalence of $\mathcal{T}$ and its quotient

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a TS.

stutter bisimulation quotient of $\mathcal{T}$:

$$\mathcal{T}/{\approx} \; = \; (S/{\approx_{\mathcal{T}}}, Act', \rightarrow_{\approx}, S_0', AP, L')$$

where $S_0' = \big\{ [s] : s \in S_0 \big\}$ and $L'([s]) = L(s)$

transition relation:

$$\frac{s \rightarrow s' \;\wedge\; s \not\approx_{\mathcal{T}} s'}{[s] \rightarrow_{\approx} [s']}$$

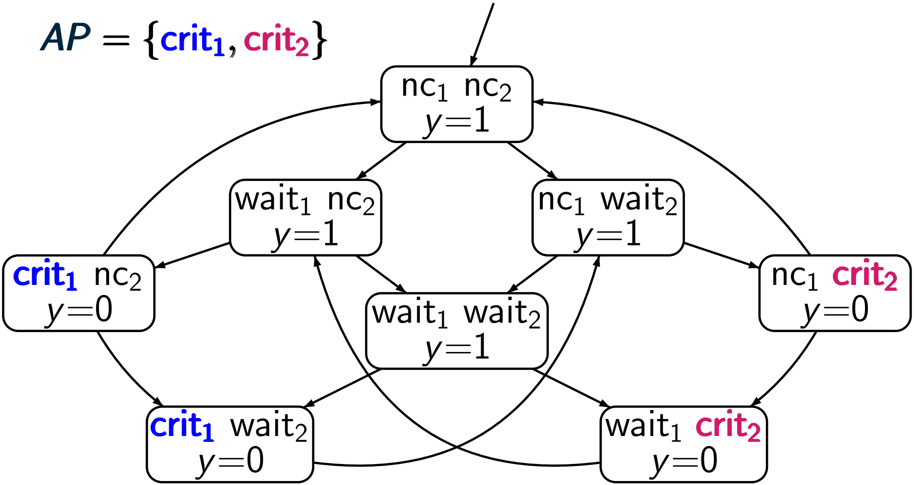
$$\boxed{\mathcal{T} \;\approx\; \mathcal{T}/{\approx}}$$

proof: $\mathcal{R} = \big\{ (s, [s]) \, : \, s \in S \big\}$

is a stutter bisimulation for $(\mathcal{T}, \mathcal{T}/{\approx})$

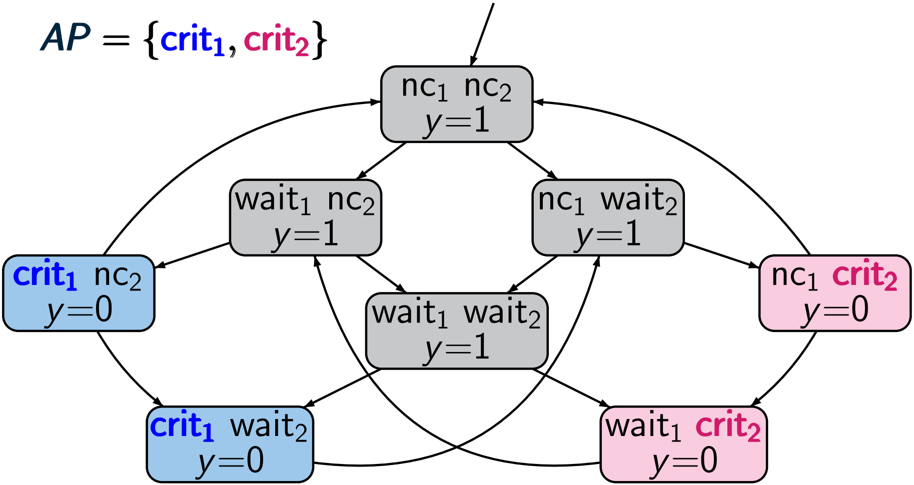
Example: mutual exclusion with semaphore

$AP = \{\mathsf{crit}_1, \mathsf{crit}_2\}$



Example: mutual exclusion with semaphore

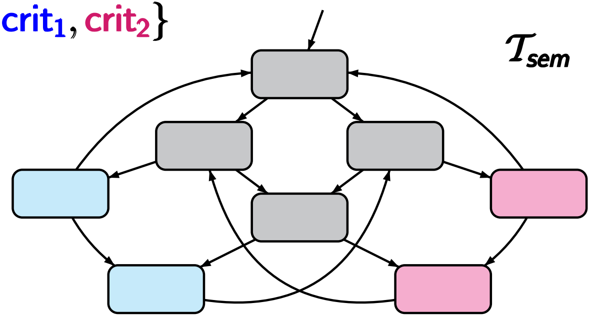
$AP = \{crit_1, crit_2\}$



stutter bisimulation with three equivalence classes

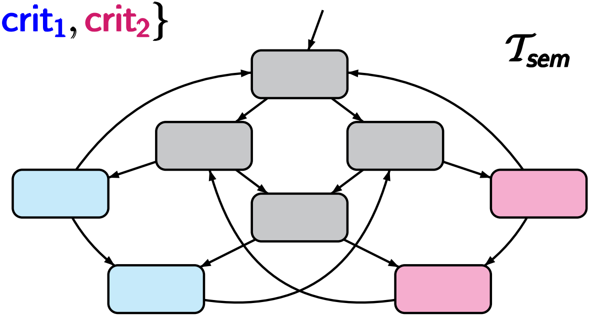
Example: mutual exclusion with semaphore

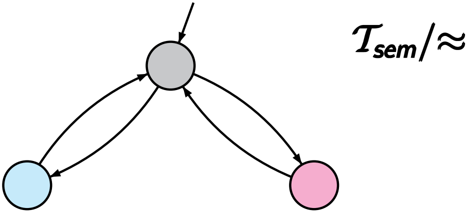
$AP = \{crit_1, crit_2\}$



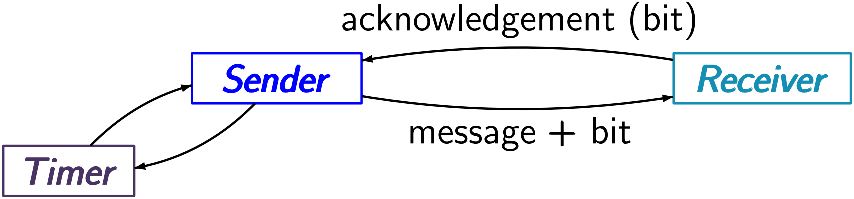
Example: mutual exclusion with semaphore STUTTER5.4-16B

$AP = \{crit_1, crit_2\}$





Alternating bit protocol



acknowledgement (bit)

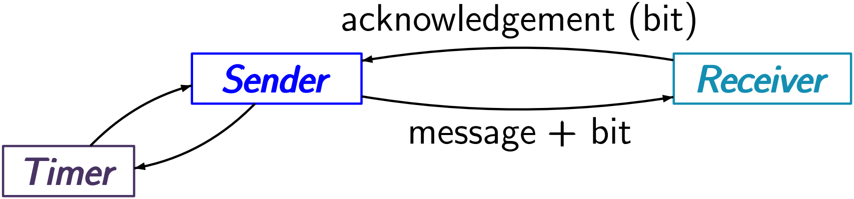
Sender

Receiver

message + bit

Timer

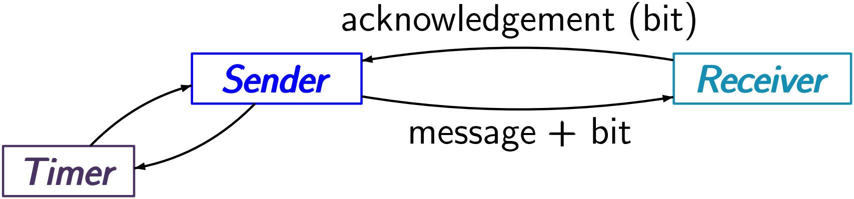
Alternating bit protocol



- formalization by a closed channel system

$$[\mathit{Sender} \,|\, \mathit{Timer} \,|\, \mathit{Receiver}]$$

Alternating bit protocol


acknowledgement (bit)

Sender

Receiver

Timer

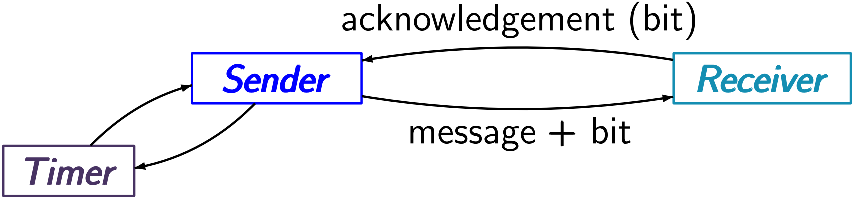
message + bit

- formalization by a closed channel system

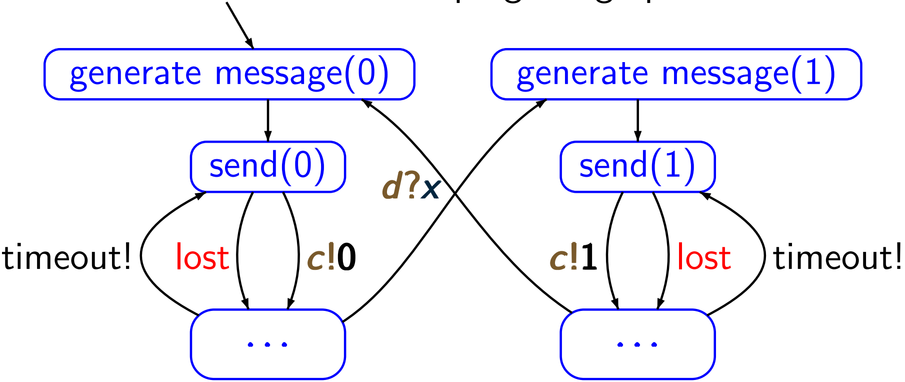
  $$[\mathit{Sender} \,|\, \mathit{Timer} \,|\, \mathit{Receiver}]$$

- TS with about $\mathbf{2^{30}}$ states
  for channels of capacity **10**

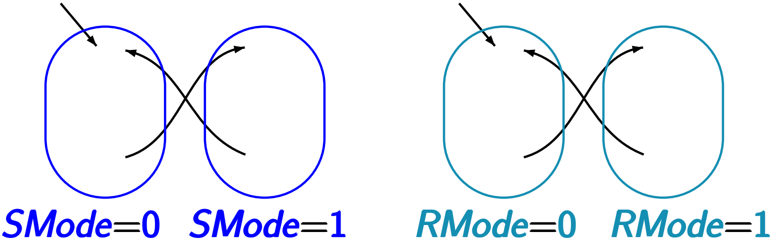
Alternating bit protocol



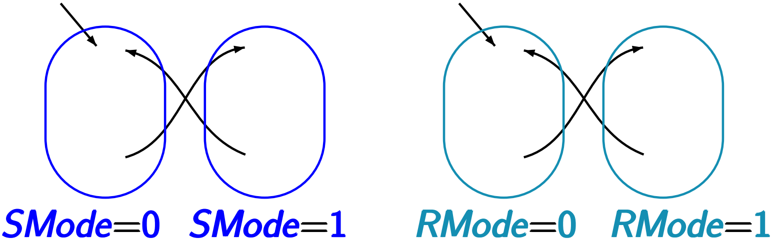
program graph for sender



Alternating bit protocol



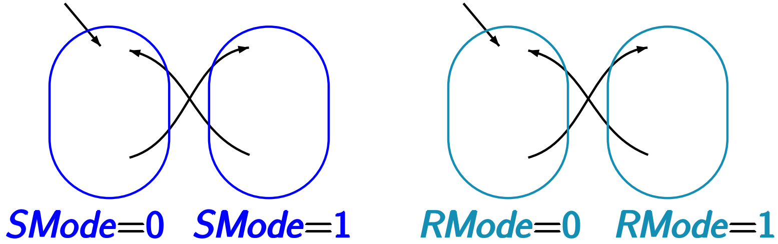
Alternating bit protocol



$$AP = \{SMode{=}0, SMode{=}1, RMode{=}0, RMode{=}1\}$$

$$\Phi \;\; = \forall\Box\Diamond SMode{=}0 \wedge \forall\Box\Diamond SMode{=}1$$

Alternating bit protocol

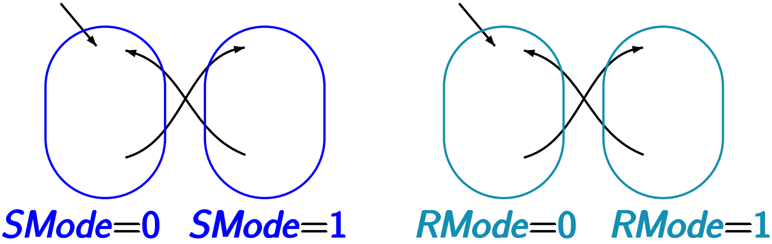


$$AP = \{SMode{=}0, SMode{=}1, RMode{=}0, RMode{=}1\}$$

$$\Phi \quad = \forall\Box\Diamond SMode{=}0 \wedge \forall\Box\Diamond SMode{=}1$$

$$ABP \not\models \Phi$$

Alternating bit protocol



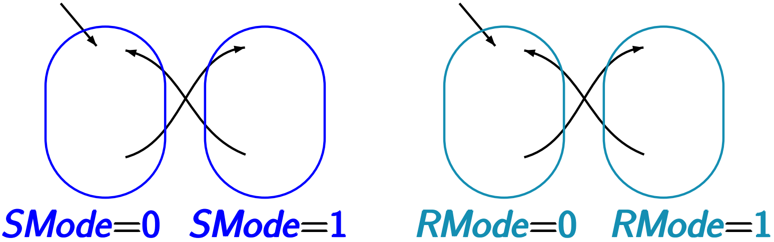
$$AP = \{SMode{=}0, SMode{=}1, RMode{=}0, RMode{=}1\}$$

$$\Phi = \forall\Box\Diamond SMode{=}0 \wedge \forall\Box\Diamond SMode{=}1$$

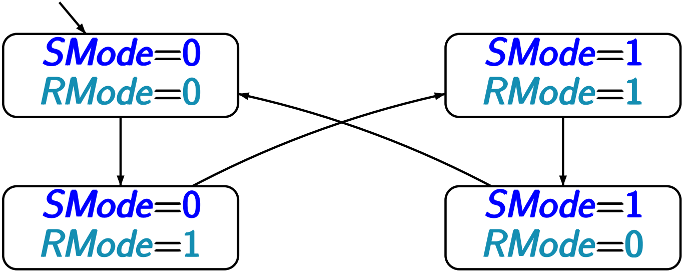
$$\boxed{ABP \not\models \Phi, \text{ but } \underset{\uparrow}{ABP/\approx} \models \Phi}$$

stutter bisimulation quotient

Alternating bit protocol



stutter bisimulation quotient:



Correct or wrong?

If $\mathcal{T}_1 \approx \mathcal{T}_2$ then $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathbf{LTL}_{\setminus\bigcirc}$-equivalent.

Correct or wrong?

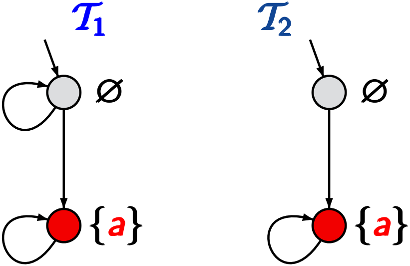
If $\mathcal{T}_1 \approx \mathcal{T}_2$ then $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathbf{LTL}_{\setminus\bigcirc}$-equivalent.

wrong.

Correct or wrong?

If $\mathcal{T}_1 \approx \mathcal{T}_2$ then $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathbf{LTL}_{\setminus\bigcirc}$-equivalent.

wrong.

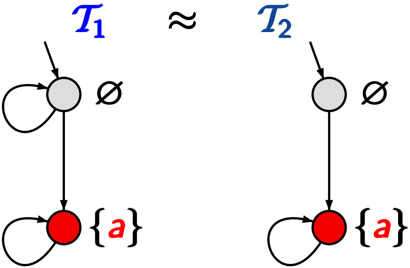


$AP = \{a\}$

Correct or wrong?

If $\mathcal{T}_1 \approx \mathcal{T}_2$ then $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathbf{LTL}_{\setminus\bigcirc}$-equivalent.

wrong.



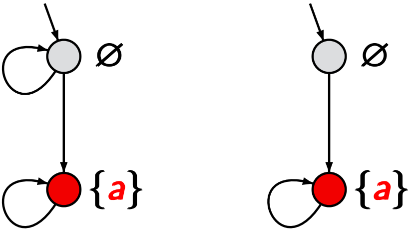
$AP = \{a\}$

Correct or wrong?

If $\mathcal{T}_1 \approx \mathcal{T}_2$ then $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\mathbf{LTL}_{\setminus \bigcirc}$-equivalent.

wrong.

$\mathcal{T}_1 \quad \approx \quad \mathcal{T}_2$



$\mathcal{T}_1 \not\models \Diamond a$ $\qquad$ $\mathcal{T}_2 \models \Diamond a$

$AP = \{a\}$

$\varnothing^\omega \in \mathit{Traces}(\mathcal{T}_1)$

$\varnothing^\omega \notin \mathit{Traces}(\mathcal{T}_2)$

Abstraction from stuttering: LT vs. BT

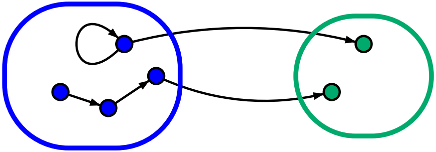
Stutter bisimulation/stutter trace equivalence

stutter trace equivalence: $\mathcal{T}_1 \triangleq \mathcal{T}_2$ iff

$$\forall \pi_1 \in Paths(\mathcal{T}_1)\ \exists \pi_2 \in Paths(\mathcal{T}_2) \text{ s.t. } \pi_1 \triangleq \pi_2$$

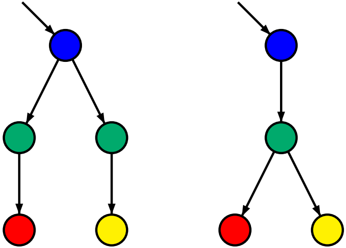
$$\forall \pi_2 \in Paths(\mathcal{T}_2)\ \exists \pi_1 \in Paths(\mathcal{T}_1) \text{ s.t. } \pi_1 \triangleq \pi_2$$

stutter bisimulation equivalence $\approx_{\mathcal{T}}$



$B \in S/\approx_{\mathcal{T}}$ $\qquad$ $C \in S/\approx_{\mathcal{T}}$

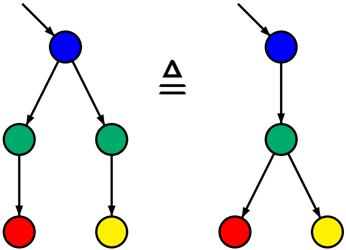
Stutter bisimulation/stutter trace equivalence



$\triangleq$ stutter trace equivalence

$\approx$ stutter bisimulation equivalence

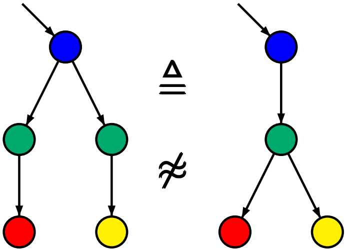
Stutter bisimulation/stutter trace equivalence



$\triangleq$ stutter trace equivalence

$\approx$ stutter bisimulation equivalence

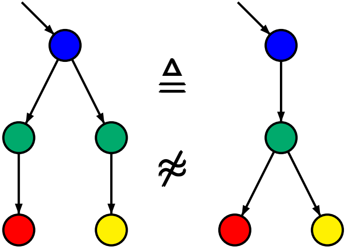
Stutter bisimulation/stutter trace equivalence

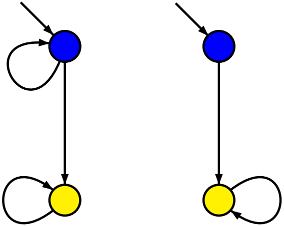


$\triangleq$ stutter trace equivalence

$\approx$ stutter bisimulation equivalence

Stutter bisimulation/stutter trace equivalence

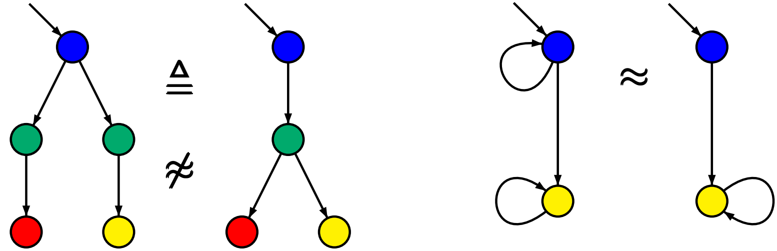




$\triangleq$ stutter trace equivalence

$\approx$ stutter bisimulation equivalence

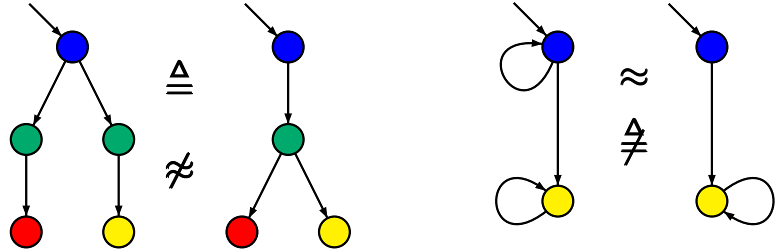
Stutter bisimulation/stutter trace equivalence



$\triangleq$ stutter trace equivalence

$\approx$ stutter bisimulation equivalence

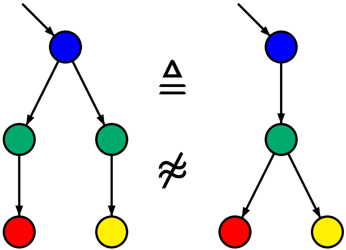
Stutter bisimulation/stutter trace equivalence

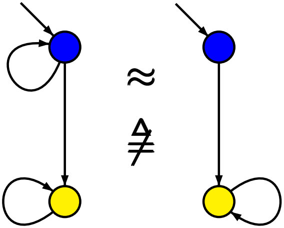


$\triangleq$ stutter trace equivalence

$\approx$ stutter bisimulation equivalence

Stutter bisimulation/stutter trace equivalence





$\triangleq$ stutter trace equivalence

$\approx$ stutter bisimulation equivalence

$\approx$ and $\triangleq$ are incomparable