# Overview

# Graph minimization under $\approx^{\mathsf{div}}$

# Graph minimization under $\approx^{\mathsf{div}}$



*given*: finite transition system $\mathcal{T}$

$\mathbf{CTL}^*_{\setminus \bigcirc}$ formula $\Phi$

*question*: does $\mathcal{T} \models \Phi$ hold **?**

# Graph minimization under $\approx^{\mathsf{div}}$



*given*: finite transition system $\mathcal{T}$

**CTL***$_{\setminus\bigcirc}$ formula $\Phi$

*question*: does $\mathcal{T} \models \Phi$ hold **?**

1. compute the quotient system $\mathcal{T}/\approx^{\mathsf{div}}$

2. apply a **CTL*** model checker to $\mathcal{T}/\approx^{\mathsf{div}}$ and $\Phi$

# Graph minimization under $\approx^{\mathsf{div}}$



*given*: finite transition system $\mathcal{T}$
$\mathbf{CTL}^*_{\setminus \bigcirc}$ formula $\Phi$

*question*: does $\mathcal{T} \models \Phi$ hold ?

1. compute the quotient system $\mathcal{T}/\approx^{\mathsf{div}}$

2. apply a **CTL*** model checker to $\mathcal{T}/\approx^{\mathsf{div}}$ and $\Phi$

$\uparrow$

Remind: $\mathcal{T} \models \Phi$ iff $\mathcal{T}/\approx^{\mathsf{div}} \models \Phi$

# Graph minimization under $\approx^{\mathsf{div}}$



*given*: finite transition system $\mathcal{T}$

**CTL***$_{\setminus \bigcirc}$ formula $\Phi$

*question*: does $\mathcal{T} \models \Phi$ hold **?**

1. compute the quotient system $\mathcal{T}/\approx^{\mathsf{div}}$

1.1. construct the divergence-sensitive expansion $\overline{\mathcal{T}}$

2. apply a **CTL*** model checker to $\mathcal{T}/\approx^{\mathsf{div}}$ and $\Phi$

$\uparrow$

Remind: $\mathcal{T} \models \Phi$ iff $\mathcal{T}/\approx^{\mathsf{div}} \models \Phi$

## Graph minimization under $\approx^{\text{div}}$



*given*: finite transition system $\mathcal{T}$

**CTL***$_{\setminus\bigcirc}$ formula $\Phi$

*question*: does $\mathcal{T} \models \Phi$ hold ?

1. compute the quotient system $\mathcal{T}/\approx^{\text{div}}$

   1.1. construct the divergence-sensitive expansion $\overline{\mathcal{T}}$

   1.2. compute quotient system $\overline{\mathcal{T}}/\approx$

2. apply a **CTL*** model checker to $\mathcal{T}/\approx^{\text{div}}$ and $\Phi$

Remind: $\mathcal{T} \models \Phi$ iff $\mathcal{T}/\approx^{\text{div}} \models \Phi$

*given*: finite transition system $\mathcal{T}$

$\mathbf{CTL}^*_{\setminus\bigcirc}$ formula $\Phi$

*question*: does $\mathcal{T} \models \Phi$ hold ?

1. compute the quotient system $\mathcal{T}/{\approx^{\mathsf{div}}}$

| |
|---|
| 1.1. construct the divergence-sensitive expansion $\overline{\mathcal{T}}$ |
| 1.2. compute quotient system $\overline{\mathcal{T}}/{\approx}$ |
| 1.3. derive $\mathcal{T}/{\approx^{\mathsf{div}}}$ from $\overline{\mathcal{T}}/{\approx}$ |

2. apply a **CTL*** model checker to $\mathcal{T}/{\approx^{\mathsf{div}}}$ and $\Phi$

$\uparrow$

Remind: $\mathcal{T} \models \Phi$ iff $\mathcal{T}/{\approx^{\mathsf{div}}} \models \Phi$

# Computing the stutter bisimulation quotient

# Computing the stutter bisimulation quotient 

*input*: finite transition system $\mathcal{T}$ over $AP$
with state space $S$

*goal*: computation of $S/\!\approx_{\mathcal{T}}$

stutter bisimulation equivalence classes
without divergence

# Computing the stutter bisimulation quotient



*input*: finite transition system $\mathcal{T}$ over $AP$
with state space $S$

*goal*: computation of $S/\!\approx_{\mathcal{T}}$

$\uparrow$

| stutter bisimulation equivalence classes without divergence |
|---|

algorithm by Groote/Vaandrager:

- iterative partitioning refinement approach

# Computing the stutter bisimulation quotient



*input*: finite transition system $\mathcal{T}$ over $AP$
with state space $S$

*goal*: computation of $S/{\approx_{\mathcal{T}}}$

stutter bisimulation equivalence classes
without divergence

algorithm by Groote/Vaandrager:

- iterative partitioning refinement approach
- initial partition $\mathcal{B}_{AP}$

# Computing the stutter bisimulation quotient



*input*: finite transition system $\mathcal{T}$ over $AP$
with state space $S$

*goal*: computation of $S/{\approx_{\mathcal{T}}}$

$\uparrow$

stutter bisimulation equivalence classes
without divergence

algorithm by Groote/Vaandrager:

- iterative partitioning refinement approach
- initial partition $\mathcal{B}_{AP}$
- refinement relies on a $Pre^*_{\mathcal{B}}$-operator

# The operator $\mathsf{Pre}^*_{\mathcal{B}}$



Let $\mathcal{B}$ be a partition and $\mathsf{C}$ be a block of $\mathcal{B}$.

# The operator $\mathsf{Pre}^*_{\mathcal{B}}$



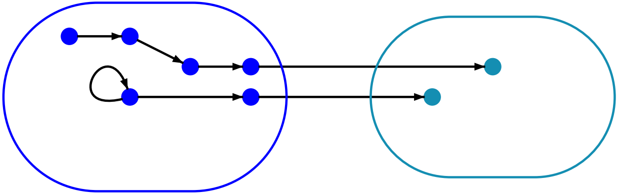
Let $\mathcal{B}$ be a partition and $C$ be a block of $\mathcal{B}$.

$\mathsf{Pre}^*_{\mathcal{B}}(C)$ = set of states $s_0$ that have a path fragment

$$s_0\, s_1\, \ldots\, s_{m-1}\, s_m$$

such that $m \geq 0$ and $s_0, s_1, \ldots, s_{m-1}$ belong to the same block of $\mathcal{B}$ and $s_m \in C$

$\mathsf{Pre}^*_{\mathcal{B}}(C)$ = set of states $s_0$ that have a path fragment $s_0 \ldots s_{m-1}\, s_m$ s.t. $m \geq 0$, $s_0, s_1, \ldots, s_{m-1}$ belong to the same block of $\mathcal{B}$ and $s_m \in C$

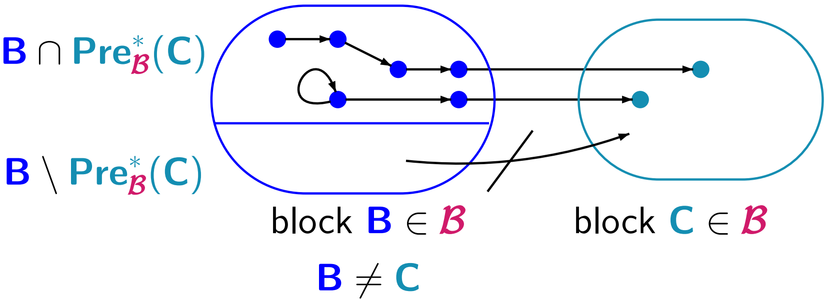


block $B \in \mathcal{B}$ block $C \in \mathcal{B}$

$B \neq C$

# The operator $\mathrm{Pre}^*_{\mathcal{B}}$

$\mathrm{Pre}^*_{\mathcal{B}}(C)$ = set of states $s_0$ that have a path fragment $s_0 \ldots s_{m-1}\, s_m$ s.t. $m \geq 0$, $s_0, s_1, \ldots, s_{m-1}$ belong to the same block of $\mathcal{B}$ and $s_m \in C$



# Computing the stutter bisimulation quotient

# Computing the stutter bisimulation quotient 

*input*: finite TS $\mathcal{T}$ over $AP$ with state space $S$

*goal*: computation of $S/\approx_{\mathcal{T}}$

# Computing the stutter bisimulation quotient



*input*: finite TS $\mathcal{T}$ over $AP$ with state space $S$

*goal*: computation of $S/\!\approx_{\mathcal{T}}$

set of stutter bisimulation equivalence classes

# Computing the stutter bisimulation quotient



*input*: finite TS $\mathcal{T}$ over $AP$ with state space $S$

*goal*: computation of $S/\approx_{\mathcal{T}}$

set of stutter bisimulation equivalence classes

algorithm by Groote/Vaandrager:

- iterative partitioning refinement approach

# Computing the stutter bisimulation quotient



*input*: finite TS $\mathcal{T}$ over $AP$ with state space $S$

*goal*: computation of $S/\approx_{\mathcal{T}}$

set of stutter bisimulation equivalence classes

algorithm by Groote/Vaandrager:

- iterative partitioning refinement approach
- initial partition $\mathcal{B}_0 = \mathcal{B}_{AP}$

# Computing the stutter bisimulation quotient

*input*: finite TS $\mathcal{T}$ over $AP$ with state space $S$

*goal*: computation of $S/\approx_{\mathcal{T}}$

set of stutter bisimulation equivalence classes

algorithm by Groote/Vaandrager:

- iterative partitioning refinement approach
- initial partition $\mathcal{B}_0 = \mathcal{B}_{AP}$
- uses splitter pairs, i.e., pairs $(B, C)$ of blocks of the current partition $\mathcal{B}_i$

# Computing the stutter bisimulation quotient

*input*: finite TS $\mathcal{T}$ over $AP$ with state space $S$

*goal*: computation of $S/{\approx_{\mathcal{T}}}$

set of stutter bisimulation equivalence classes

algorithm by Groote/Vaandrager:

- iterative partitioning refinement approach
- initial partition $\mathcal{B}_0 = \mathcal{B}_{AP}$
- uses splitter pairs, i.e., pairs $(B, C)$ of blocks of the current partition $\mathcal{B}_i$ such that $B \cap Pre^*_{\mathcal{B}_i}(C)$ and $B \setminus Pre^*_{\mathcal{B}_i}(C)$ are nonempty

# Groote/Vaandrager algorithm (pseudo-code)

# Groote/Vaandrager algorithm (pseudo-code)



```
𝓑 := 𝓑_AP;

WHILE 𝓑 can be refined DO




OD

return 𝓑
```

$\mathcal{B} := \mathcal{B}_{AP}$;

# Groote/Vaandrager algorithm (pseudo-code)



```
𝓑 := 𝓑_AP;
WHILE 𝓑 can be refined DO
    choose a splitter pair (B, C) for 𝓑


OD
return 𝓑
```

# Groote/Vaandrager algorithm (pseudo-code)

```
𝓑 := 𝓑_AP;
WHILE 𝓑 can be refined DO
    choose a splitter pair (B, C) for 𝓑
    remove B from 𝓑;

OD
return 𝓑
```

# Groote/Vaandrager algorithm (pseudo-code)

$\mathcal{B} := \mathcal{B}_{AP}$;

WHILE $\mathcal{B}$ can be refined DO

    choose a splitter pair $(B, C)$ for $\mathcal{B}$

    remove $B$ from $\mathcal{B}$;

    add $B \cap Pre^*_{\mathcal{B}}(C)$ and $B \setminus Pre^*_{\mathcal{B}}(C)$ to $\mathcal{B}$

OD

return $\mathcal{B}$

## Groote/Vaandrager algorithm (pseudo-code)



$\mathcal{B} := \mathcal{B}_{AP}$;

`WHILE` $\mathcal{B}$ can be refined `DO`

    choose a splitter pair $(B, C)$ for $\mathcal{B}$

    remove $B$ from $\mathcal{B}$;

    add $B \cap Pre^*_{\mathcal{B}}(C)$ and $B \setminus Pre^*_{\mathcal{B}}(C)$ to $\mathcal{B}$

`OD`

return $\mathcal{B}$

loop invariant:
$\mathcal{B}$ is finer than $\mathcal{B}_{AP}$ and coarser than $S/{\approx_{\mathcal{T}}}$

# Groote/Vaandrager algorithm (pseudo-code)



$\mathcal{B} := \mathcal{B}_{AP}$;

WHILE $\mathcal{B}$ can be refined DO

choose a splitter pair $(B, C)$ for $\mathcal{B}$

remove $B$ from $\mathcal{B}$;

add $B \cap Pre^*_{\mathcal{B}}(C)$ and $B \setminus Pre^*_{\mathcal{B}}(C)$ to $\mathcal{B}$

OD

return $\mathcal{B}$ $\longleftarrow$ $\boxed{\mathcal{B} = S/\approx_{\mathcal{T}}}$

loop invariant:
$\mathcal{B}$ is finer than $\mathcal{B}_{AP}$ and coarser than $S/\approx_{\mathcal{T}}$

# Complexity of the G/V algorithm

# Complexity of the G/V algorithm



$\mathcal{B} := \mathcal{B}_{AP}$;

WHILE $\mathcal{B}$ can be refined DO

  choose a splitter pair $(B, C)$ for $\mathcal{B}$

  remove $B$ from $\mathcal{B}$;

  add $B \cap Pre^*_{\mathcal{B}}(C)$ and $B \setminus Pre^*_{\mathcal{B}}(C)$ to $\mathcal{B}$

OD

return $\mathcal{B}$

## Complexity of the G/V algorithm



$\mathcal{B} := \mathcal{B}_{AP}$;

WHILE $\mathcal{B}$ can be refined DO

choose a splitter pair $(B, C)$ for $\mathcal{B}$

remove $B$ from $\mathcal{B}$;

add $B \cap Pre^*_{\mathcal{B}}(C)$ and $B \setminus Pre^*_{\mathcal{B}}(C)$ to $\mathcal{B}$

OD

return $\mathcal{B}$

# Complexity of the G/V algorithm

$\mathcal{B} := \mathcal{B}_{AP}$;
WHILE $\mathcal{B}$ can be refined DO
  choose a splitter pair $(B, C)$ for $\mathcal{B}$
  remove $B$ from $\mathcal{B}$;
  add $B \cap Pre^*_{\mathcal{B}}(C)$ and $B \setminus Pre^*_{\mathcal{B}}(C)$ to $\mathcal{B}$
OD
return $\mathcal{B}$

- search for splitter pair in time $\mathcal{O}(m)$

where $m$ = number of edges $\geq |S|$

# Complexity of the G/V algorithm

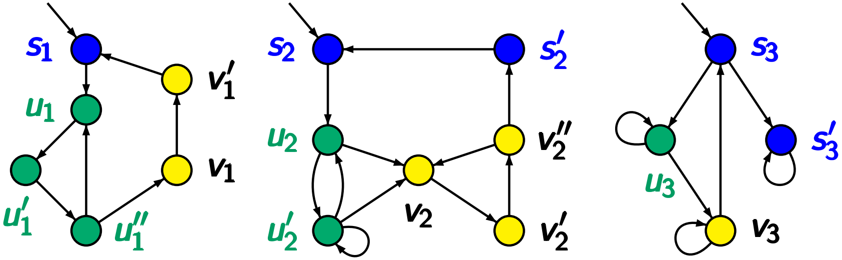
```
𝓑 := 𝓑_AP;
WHILE 𝓑 can be refined DO
    choose a splitter pair (B, C) for 𝓑
    remove B from 𝓑;
    add B ∩ Pre*_𝓑(C) and B \ Pre*_𝓑(C) to 𝓑
OD
return 𝓑
```

- search for splitter pair in time $\mathcal{O}(m)$
- complexity: $\mathcal{O}(|S| \cdot |AP| + |S| \cdot m)$

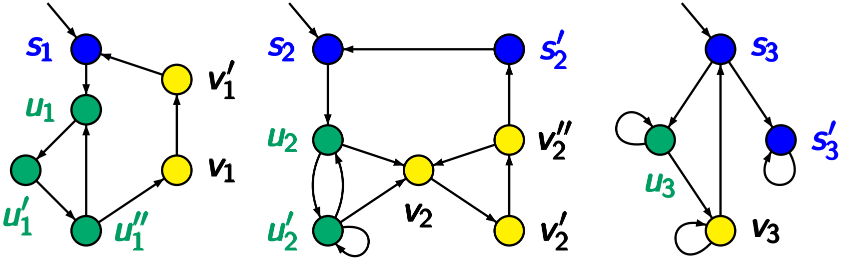
where $m$ = number of edges $\geq |S|$

# Example: computation of $S/{\approx_{\mathcal{T}}}$





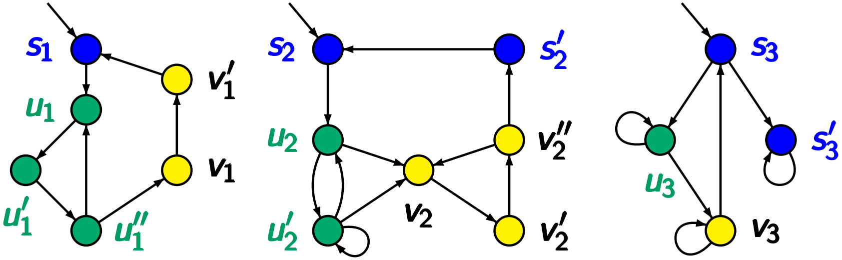
# Example: computation of $S/{\approx_{\mathcal{T}}}$




initial partition: $\{s_1, s_2, s_2', s_3, s_3'\}$,
$\{u_1, u_1', u_1'', u_2, u_2', u_3\}$,
$\{v_1, v_1', v_2, v_2', v_2'', v_3\}$

# Example: computation of $S/{\approx_\mathcal{T}}$

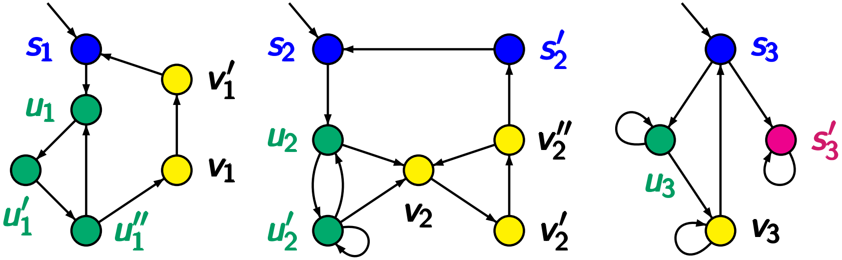





initial partition: $\{s_1, s_2, s_2', s_3, s_3'\}$,
$\{u_1, u_1', u_1'', u_2, u_2', u_3\}$,
$\{v_1, v_1', v_2, v_2', v_2'', v_3\}$

splitter pair $(B, C)$ with $B = \{s_1, \ldots, s_3'\}$, $C = \{u_1, \ldots\}$

# Example: computation of $S/\approx_{\mathcal{T}}$



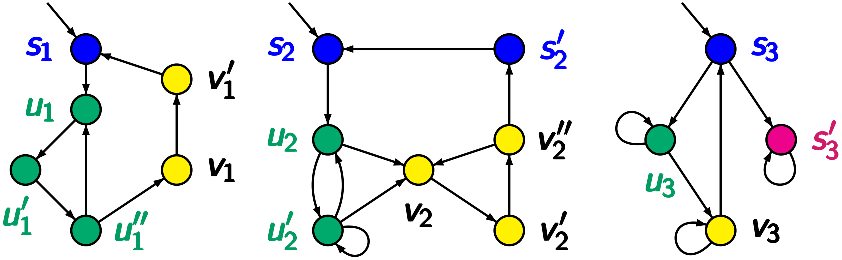

initial partition: $\{s_1, s_2, s_2', s_3, s_3'\}$,
$\{u_1, u_1', u_1'', u_2, u_2', u_3\}$,
$\{v_1, v_1', v_2, v_2', v_2'', v_3\}$

splitter pair $(B, C)$ with $B = \{s_1, \ldots, s_3'\}$, $C = \{u_1, \ldots\}$

$B \rightsquigarrow B_1 = \{s_1, s_2, s_2', s_3\}$ and $B_2 = \{s_3'\}$

# Example: computation of $S/\approx_{\mathcal{T}}$






initial partition: $\{s_1, s_2, s_2', s_3, s_3'\}$,
$\{u_1, u_1', u_1'', u_2, u_2', u_3\}$,
$\{v_1, v_1', v_2, v_2', v_2'', v_3\}$

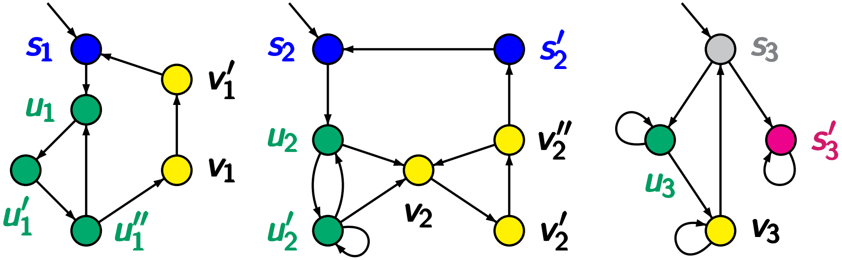
splitter pair $(B, C)$ with $B = \{s_1, \ldots, s_3'\}$, $C = \{u_1, \ldots\}$

$B \rightsquigarrow B_1 = \{s_1, s_2, s_2', s_3\}$ and $B_2 = \{s_3'\}$

splitter pair $(B_1, B_2)$:

# Example: computation of $S/{\approx_{\mathcal{T}}}$



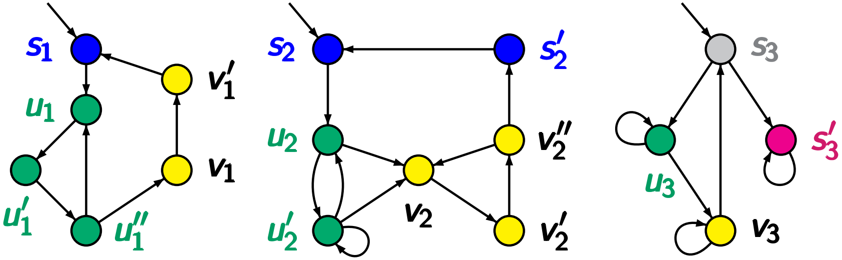



splitter pair $(B_1, B_2)$:

$$B_1 \rightsquigarrow B_3 = \{s_1, s_2, s_2'\} \text{ and } B_4 = \{s_3\}$$

# Example: computation of $S/\approx_{\mathcal{T}}$



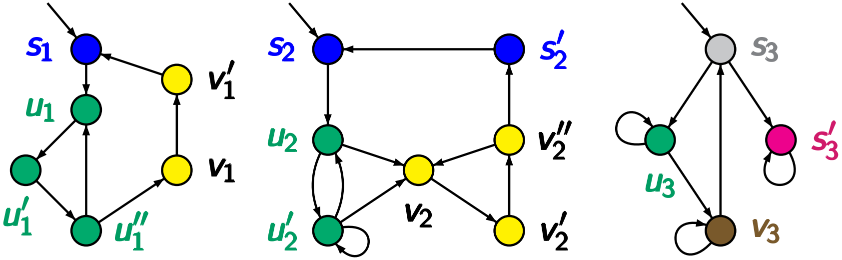



splitter pair $(B_1, B_2)$:

$$B_1 \rightsquigarrow B_3 = \{s_1, s_2, s_2'\} \text{ and } B_4 = \{s_3\}$$

splitter pair $(D, \{s_3\})$ where $D = \{v_1, v_1', \ldots, v_3\}$ ...

# Example: computation of $S/{\approx_{\mathcal{T}}}$



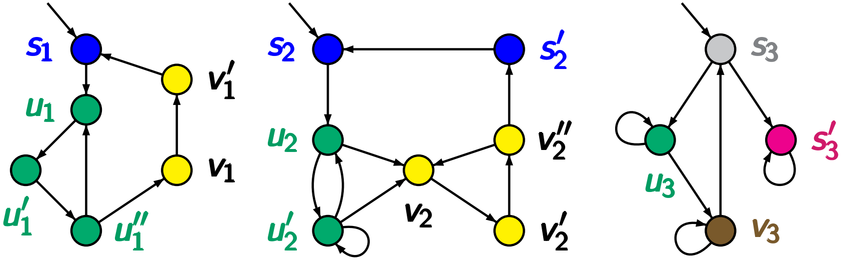

splitter pair $(B_1, B_2)$:

$B_1 \rightsquigarrow B_3 = \{s_1, s_2, s_2'\}$ and $B_4 = \{s_3\}$

splitter pair $(D, \{s_3\})$ where $D = \{v_1, v_1', \ldots, v_3\}$ ...

$D \rightsquigarrow D_1 = \{v_1, \ldots\}$ and $D_2 = \{v_3\}$

# Example: computation of $S/\approx_{\mathcal{T}}$




splitter pair $(B_1, B_2)$:

$B_1 \rightsquigarrow B_3 = \{s_1, s_2, s'_2\}$ and $B_4 = \{s_3\}$

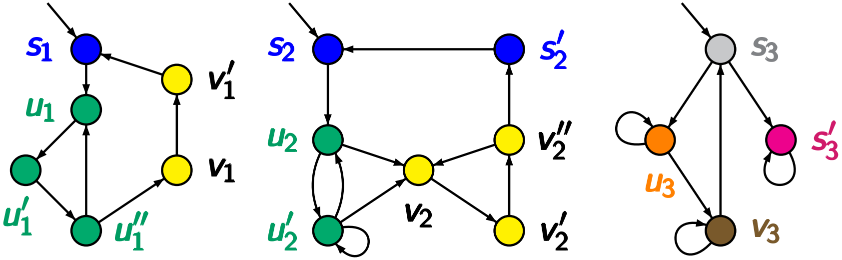
splitter pair $(D, \{s_3\})$ where $D = \{v_1, v'_1, \ldots, v_3\}$ ...

$D \rightsquigarrow D_1 = \{v_1, \ldots\}$ and $D_2 = \{v_3\}$

splitter pair $(C, \{v_3\})$ where $C = \{u_1, u'_1, \ldots, u_3\}$ ...

# Example: computation of $S/\!\approx_{\mathcal{T}}$






splitter pair $(B_1, B_2)$:

$B_1 \rightsquigarrow B_3 = \{s_1, s_2, s_2'\}$ and $B_4 = \{s_3\}$

splitter pair $(D, \{s_3\})$ where $D = \{v_1, v_1', \ldots, v_3\}$ ...

$D \rightsquigarrow D_1 = \{v_1, \ldots\}$ and $D_2 = \{v_3\}$

splitter pair $(C, \{v_3\})$ where $C = \{u_1, u_1', \ldots, u_3\}$ ...

$C \rightsquigarrow C_1 = \{u_1, \ldots\}$ and $C_2 = \{u_3\}$

# Computation of the quotient for $\approx^{\mathsf{div}}_{\mathcal{T}}$



*given*: finite transition system $\mathcal{T}$
with state space $S$

*goal*: compute $S/\approx^{\mathsf{div}}_{\mathcal{T}}$

# Computation of the quotient for $\approx_{\mathcal{T}}^{\mathsf{div}}$



*given*: finite transition system $\mathcal{T}$
with state space $S$

*goal*: compute $S/\approx_{\mathcal{T}}^{\mathsf{div}}$

via reduction to the problem of computing the quotient w.r.t. $\approx$

# Computation of the quotient for $\approx^{\text{div}}_{\mathcal{T}}$



*given*: finite transition system $\mathcal{T}$
with state space $S$

*goal*: compute $S/{\approx^{\text{div}}_{\mathcal{T}}}$

via reduction to the problem of computing the quotient w.r.t. $\approx$

1. construct the divergence-sensitive expansion $\overline{\mathcal{T}}$
with state space $\overline{S} = S \cup \ldots$

*given*: finite transition system $\mathcal{T}$
with state space $S$

*goal*: compute $S/\approx_{\mathcal{T}}^{\text{div}}$

via reduction to the problem of computing the quotient w.r.t. $\approx$

1. construct the divergence-sensitive expansion $\overline{\mathcal{T}}$ with state space $\overline{S} = S \cup \ldots$
2. compute the quotient $\overline{S}/\approx_{\overline{\mathcal{T}}}$ w.r.t. stutter bisimulation equivalence without divergence

# Computation of the quotient for $\approx^{\mathsf{div}}_{\mathcal{T}}$

*given*: finite transition system $\mathcal{T}$
with state space $S$

*goal*: compute $S/\approx^{\mathsf{div}}_{\mathcal{T}}$

via reduction to the problem of computing the quotient w.r.t. $\approx$

1. construct the divergence-sensitive expansion $\overline{\mathcal{T}}$ with state space $\overline{S} = S \cup \ldots$
2. compute the quotient $\overline{S}/\approx_{\overline{\mathcal{T}}}$ w.r.t. stutter bisimulation equivalence without divergence
3. derive $S/\approx^{\mathsf{div}}_{\mathcal{T}}$ from $\overline{S}/\approx_{\overline{\mathcal{T}}}$

# Stutter cycles

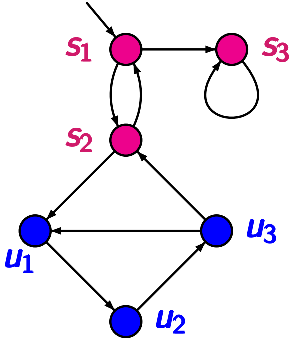
STUTTER5.4-63

# Stutter cycles



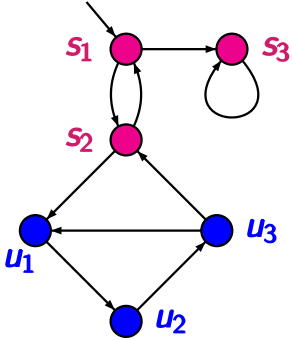
A stutter cycle is a cycle $s_0\, s_1\, \ldots\, s_n$ in $\mathcal{T}$ s.t.
$L(s_0) = L(s_i)$ for $i = 1, \ldots, n$.

A stutter cycle is a cycle $s_0\, s_1\, \ldots\, s_n$ in $\mathcal{T}$ s.t.
$L(s_0) = L(s_i)$ for $i = 1, \ldots, n$.



# Stutter cycles

A stutter cycle is a cycle $s_0\, s_1\, \ldots\, s_n$ in $\mathcal{T}$ s.t. $L(s_0) = L(s_i)$ for $i = 1, \ldots, n$.




stutter cycles, e.g.:

$s_1\, s_2\, s_1$

$s_3\, s_3$

$u_1\, u_2\, u_3\, u_1$

A stutter cycle is a cycle $s_0\, s_1\, \ldots\, s_n$ in $\mathcal{T}$ s.t.
$$L(s_0) = L(s_i) \text{ for } i = 1, \ldots, n.$$

If $s_0\, s_1\, \ldots\, s_n$ is a stutter cycle then
$$s_0 \approx_{\mathcal{T}}^{\mathsf{div}} s_1 \approx_{\mathcal{T}}^{\mathsf{div}} \ldots \approx_{\mathcal{T}}^{\mathsf{div}} s_n$$

A stutter cycle is a cycle $s_0\, s_1\, \ldots\, s_n$ in $\mathcal{T}$ s.t.

$$L(s_0) = L(s_i) \text{ for } i = 1, \ldots, n.$$

If $s_0\, s_1\, \ldots\, s_n$ is a stutter cycle then

$$s_0 \approx^{\text{div}}_{\mathcal{T}} s_1 \approx^{\text{div}}_{\mathcal{T}} \ldots \approx^{\text{div}}_{\mathcal{T}} s_n$$

*Proof:* show that

$$\mathcal{R} = \mathbf{id} \cup \{(s_i, s_j) : i, j{=}1, \ldots, n\}$$

is a divergence-sensitive stutter bisimulation.

Let $\mathcal{T}$ be a finite transition system.

If state $s$ is $\approx_{\mathcal{T}}^{\mathsf{div}}$-divergent then $s$ belongs to a stutter cycle.

Let $\mathcal{T}$ be a finite transition system.

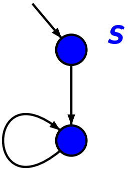
> If state $s$ is $\approx^{\mathsf{div}}_{\mathcal{T}}$-divergent then $s$ belongs to a stutter cycle.

**wrong.**

Let $\mathcal{T}$ be a finite transition system.

If state $s$ is $\approx_{\mathcal{T}}^{\mathsf{div}}$-divergent then $s$ belongs to a stutter cycle.

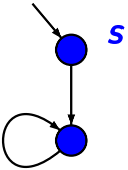
**wrong.**



Let $\mathcal{T}$ be a finite transition system.

If state $s$ is $\approx_{\mathcal{T}}^{\mathsf{div}}$-divergent then $s$ belongs to a stutter cycle.

**wrong.**

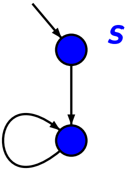



$s$ is $\approx_{\mathcal{T}}^{\mathsf{div}}$-divergent

Let $\mathcal{T}$ be a finite transition system.

> If state $s$ is $\approx^{\mathsf{div}}_{\mathcal{T}}$-divergent then $s$ belongs to a stutter cycle.

**wrong.**



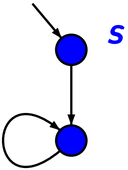

$s$ is $\approx^{\mathsf{div}}_{\mathcal{T}}$-divergent

> If state $s$ is $\approx^{\mathsf{div}}_{\mathcal{T}}$-divergent then $s$ can reach a stutter cycle by stutter steps within $[s]$.

Let $\mathcal{T}$ be a finite transition system.

If state $s$ is $\approx^{\text{div}}_{\mathcal{T}}$-divergent then $s$ belongs to a stutter cycle.

**wrong.**




$s$ is $\approx^{\text{div}}_{\mathcal{T}}$-divergent

State $s$ is $\approx^{\text{div}}_{\mathcal{T}}$-divergent iff $s$ can reach a stutter cycle by stutter steps within $[s]$.

# Correct or wrong?



If $s \longrightarrow s'$ is a stutter step, i.e., $L(s) = L(s')$, then $s \approx^{\mathsf{div}}_{\mathcal{T}} s'$.

# Correct or wrong?



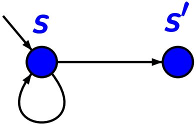
If $s \to s'$ is a stutter step, i.e., $L(s) = L(s')$, then $s \approx_{\mathcal{T}}^{\text{div}} s'$.

**wrong.**

# Correct or wrong?

If $s \to s'$ is a stutter step, i.e., $L(s) = L(s')$, then $s \approx^{\text{div}}_{\mathcal{T}} s'$.

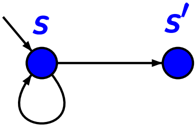
**wrong.**



# Correct or wrong?

If $s \to s'$ is a stutter step, i.e., $L(s) = L(s')$,
then $s \approx_{\mathcal{T}}^{\textsf{div}} s'$.

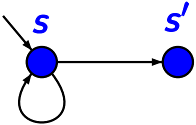
**wrong.**




$s$ divergent
$s'$ nondivergent

If $s \longrightarrow s'$ is a stutter step, i.e., $L(s) = L(s')$, then $s \approx^{\mathsf{div}}_{\mathcal{T}} s'$.

**wrong.**



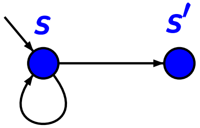

$s$ divergent

$s'$ nondivergent

$s \not\approx^{\mathsf{div}}_{\mathcal{T}} s'$

If $s \to s'$ is a stutter step, i.e., $L(s) = L(s')$, then $s \approx_{\mathcal{T}}^{\mathsf{div}} s'$.

**wrong.**




$s$ divergent

$s'$ nondivergent

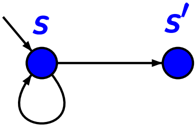
$s \not\approx_{\mathcal{T}}^{\mathsf{div}} s'$

1. If $s \to s'$ is a stutter step and $Post(s) = \{s'\}$ then $s \approx_{\mathcal{T}}^{\mathsf{div}} s'$.

# Correct or wrong?

If $s \to s'$ is a stutter step, i.e., $L(s) = L(s')$,
then $s \approx_{\mathcal{T}}^{\text{div}} s'$.

**wrong.**




$s$ divergent
$s'$ nondivergent

$s \not\approx_{\mathcal{T}}^{\text{div}} s'$

1. If $s \to s'$ is a stutter step and $Post(s) = \{s'\}$ then $s \approx_{\mathcal{T}}^{\text{div}} s'$.
2. If $s$ and $s'$ are on the same stutter cycle then $s \approx_{\mathcal{T}}^{\text{div}} s'$.

# Divergence-sensitive expansion

# Divergence-sensitive expansion



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

# Divergence-sensitive expansion



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

The divergence-sensitive expansion $\overline{\mathcal{T}}$ of $\mathcal{T}$ is a TS

# Divergence-sensitive expansion



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

The divergence-sensitive expansion $\overline{\mathcal{T}}$ of $\mathcal{T}$ is a TS

- that contains $\mathcal{T}$ as a subsystem

# Divergence-sensitive expansion



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

The divergence-sensitive expansion $\overline{\mathcal{T}}$ of $\mathcal{T}$ is a TS

- that contains $\mathcal{T}$ as a subsystem
- for all states $s_1, s_2 \in S$:

$$s_1 \approx^{\mathsf{div}}_{\mathcal{T}} s_2 \qquad \text{iff} \qquad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

# Divergence-sensitive expansion

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

The divergence-sensitive expansion $\overline{\mathcal{T}}$ of $\mathcal{T}$ is a TS

- that contains $\mathcal{T}$ as a subsystem
- for all states $s_1, s_2 \in S$:

$$\underbrace{s_1 \approx^{\text{div}}_{\mathcal{T}} s_2}_{\boxed{\text{stutter bis. equiv. with divergence in } \mathcal{T}}} \quad \text{iff} \quad \underbrace{s_1 \approx_{\overline{\mathcal{T}}} s_2}_{\boxed{\text{stutter bis. equiv. without divergence in } \overline{\mathcal{T}}}}$$

# Divergence-sensitive expansion



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

The divergence-sensitive expansion $\overline{\mathcal{T}}$ of $\mathcal{T}$ is a TS

- that contains $\mathcal{T}$ as a subsystem
- for all states $s_1, s_2 \in S$:

$$s_1 \approx_{\mathcal{T}}^{\text{div}} s_2 \qquad \text{iff} \qquad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

| | |
|---|---|
| stutter bis. equiv. with divergence in $\mathcal{T}$ | stutter bis. equiv. without divergence in $\overline{\mathcal{T}}$ |

# Divergence-sensitive expansion

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

Divergence-sensitive expansion of $\mathcal{T}$:

$$\overline{\mathcal{T}} \stackrel{\text{def}}{=} (S \cup \{div\}, Act', \rightarrow', S_0, \overline{AP}, \overline{L})$$

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

Divergence-sensitive expansion of $\mathcal{T}$:

$$\overline{\mathcal{T}} \stackrel{\text{def}}{=} (S \cup \{div\}, Act', \rightarrow', S_0, \overline{AP}, \overline{L})$$

where ***div*** is a new state, not contained in $S$

# Divergence-sensitive expansion



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

Divergence-sensitive expansion of $\mathcal{T}$:

$$\overline{\mathcal{T}} \stackrel{\text{def}}{=} (S \cup \{div\}, Act', \rightarrow', S_0, \overline{AP}, \overline{L})$$

- $\overline{AP} = AP \cup \{div\}$

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

Divergence-sensitive expansion of $\mathcal{T}$:

$$\overline{\mathcal{T}} \stackrel{\text{def}}{=} (S \cup \{div\}, Act', \to', S_0, \overline{AP}, \overline{L})$$

- $\overline{AP} = AP \cup \{div\}$
- labeling: $\overline{L(s)} = L(s)$ for all states $s$ in $\mathcal{T}$
  $\overline{L}(div) = \{div\}$

# Divergence-sensitive expansion



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

Divergence-sensitive expansion of $\mathcal{T}$:

$$\overline{\mathcal{T}} \stackrel{\text{def}}{=} (S \cup \{div\}, Act', \longrightarrow', S_0, \overline{AP}, \overline{L})$$

- $\overline{AP} = AP \cup \{div\}$
- labeling: $\overline{L(s)} = L(s)$ for all states $s$ in $\mathcal{T}$

  $\overline{L}(div) = \{div\}$
- transition relation $\longrightarrow'$ consists of all transitions in $\mathcal{T}$

  $+$ $s \longrightarrow' div$ for all states $s$ on a stutter cycle in $\mathcal{T}$

  $+$ self-loop $div \longrightarrow' div$

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

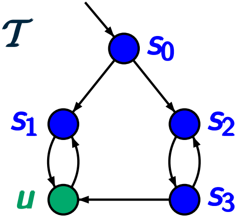
Divergence-sensitive expansion of $\mathcal{T}$:

$$\overline{\mathcal{T}} \stackrel{\text{def}}{=} (S \cup \{div\}, Act', \rightarrow', S_0, \overline{AP}, \overline{L})$$

- $\overline{AP} = AP \cup \{div\}$
- labeling: $\overline{L(s)} = L(s)$ for all states $s$ in $\mathcal{T}$
  $\overline{L}(div) = \{div\}$
- transition relation $\rightarrow'$ consists of all transitions in $\mathcal{T}$
  - $+$ $s \rightarrow' div$ for all states $s$ on a stutter cycle in $\mathcal{T}$
  - $+$ self-loop $div \rightarrow' div$
- actions names: irrelevant

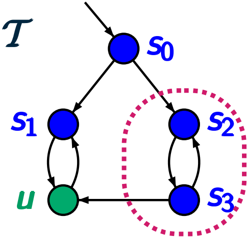
# Example: divergence-sensitive expansion





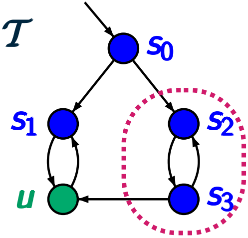

$$AP = \{a\}$$

# Example: divergence-sensitive expansion




$AP = \{a\}$

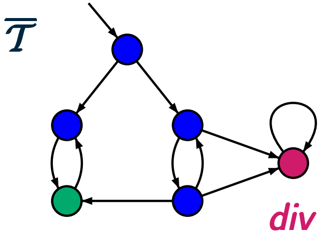
stutter cycle $s_2\, s_3\, s_2$



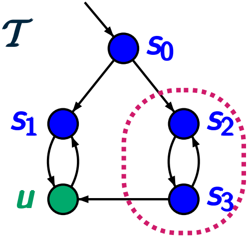

$AP = \{a\}$

stutter cycle $s_2\, s_3\, s_2$

divergence-sensitive expansion:



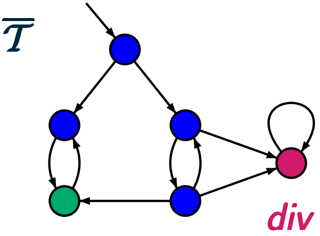
# Example: divergence-sensitive expansion



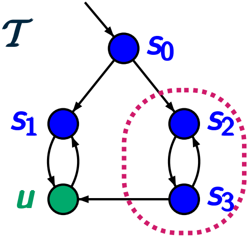

$AP = \{a\}$

stutter cycle $s_2\, s_3\, s_2$

divergence-sensitive expansion:



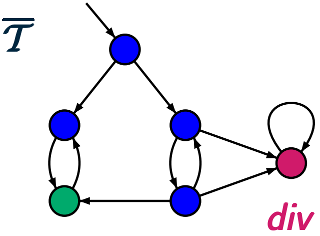

$\overline{AP} = \{a, div\}$




$AP = \{a\}$

stutter cycle $s_2\, s_3\, s_2$

divergence-sensitive expansion:

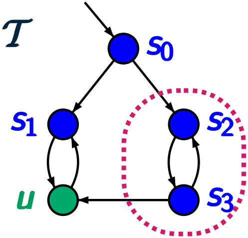



$\overline{AP} = \{a, div\}$

$\approx_{\overline{\mathcal{T}}}$-equivalence classes:

$\{s_0\}\ \{s_1\}\ \{s_2, s_3\}\ \{u\}\ \{div\}$

# Example: divergence-sensitive expansion



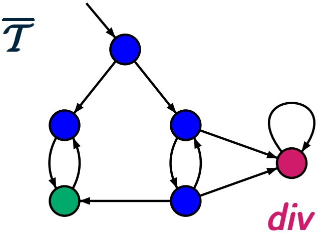

$AP = \{a\}$

stutter cycle $s_2\, s_3\, s_2$

$\approx_{\mathcal{T}}^{\mathbf{div}}$-equivalence classes:

$\{s_0\}\ \{s_1\}\ \{s_2, s_3\}\ \{u\}$

divergence-sensitive expansion:




$\overline{AP} = \{a, div\}$

$\approx_{\overline{\mathcal{T}}}$-equivalence classes:

$\{s_0\}\ \{s_1\}\ \{s_2, s_3\}\ \{u\}\ \{div\}$

# $\mathcal{T}$ and its divergence-sensitive expansion $\overline{\mathcal{T}}$ stutter5.4-67a

# $\mathcal{T}$ and its divergence-sensitive expansion $\overline{\mathcal{T}}$ 

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a finite TS. Then, for all states $s_1, s_2 \in S$:

$$s_1 \approx^{\mathsf{div}}_{\mathcal{T}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

Let $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$ be a finite TS. Then, for all states $s_1$, $s_2 \in S$:

$$s_1 \approx^{\text{div}}_{\mathcal{T}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

stutter bis. equiv. with divergence in $\mathcal{T}$

stutter bis. equiv. without divergence in $\overline{\mathcal{T}}$

# $\mathcal{T}$ and its divergence-sensitive expansion $\overline{\mathcal{T}}$ 

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a finite TS. Then, for all states $s_1, s_2 \in S$:

$$s_1 \approx^{\text{div}}_{\mathcal{T}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

"$\Longrightarrow$":

Let $\mathcal{T} = (S, Act, \rightarrow, S_0, AP, L)$ be a finite TS. Then, for all states $s_1$, $s_2 \in S$:

$$s_1 \approx^{\mathsf{div}}_{\mathcal{T}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

"$\Longrightarrow$": show that

$$\mathcal{R} = \approx^{\mathsf{div}}_{\mathcal{T}} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$

Let $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$ be a finite TS. Then, for all states $s_1$, $s_2 \in S$:

$$s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

"$\Longrightarrow$": show that

$$\mathcal{R} = \approx_{\mathcal{T}}^{\mathsf{div}} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$

"$\Longleftarrow$": show that

$$\mathcal{R} = \approx_{\overline{\mathcal{T}}} \cap (S \times S)$$

is a divergence-sensitive stutter bisimulation for $\mathcal{T}$

Let $\mathcal{T} = (S, Act, \longrightarrow, S_0, AP, L)$ be a $\boxed{\text{finite}}$ TS. Then, for all states $s_1, s_2 \in S$:

$$s_1 \approx^{\text{div}}_{\mathcal{T}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

"$\Longrightarrow$": show that

$$\mathcal{R} \;= \approx^{\text{div}}_{\mathcal{T}} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$

"$\Longleftarrow$": show that

$$\mathcal{R} \;= \approx_{\overline{\mathcal{T}}} \cap (S \times S)$$

is a divergence-sensitive stutter bisimulation for $\mathcal{T}$

# $\approx_{\mathcal{T}}$ is finer than $\approx_{\mathcal{T}}^{\mathsf{div}}$

# $\approx_{\mathcal{T}}$ is finer than $\approx_{\mathcal{T}}^{\mathrm{div}}$

Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Claim: $\mathcal{R} = \{(s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2\}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

*Proof:* show that for all states $s_1$, $s_2$ in $\mathcal{T}$ with $s_1 \approx_{\mathcal{T}} s_2$:

# $\approx_{\mathcal{T}}$ is finer than $\approx_{\mathcal{T}}^{\mathsf{div}}$

Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

*Proof:* show that for all states $s_1$, $s_2$ in $\mathcal{T}$ with $s_1 \approx_{\mathcal{T}} s_2$:

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

# $\approx_{\mathcal{T}}$ is finer than $\approx_{\mathcal{T}}^{\text{div}}$



Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

*Proof:* show that for all states $s_1$, $s_2$ in $\mathcal{T}$ with $s_1 \approx_{\mathcal{T}} s_2$:

(1) $L(s_1) = L(s_2)$ $\quad\checkmark$

Claim: $\mathcal{R} = \{(s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2\}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

*Proof:* show that for all states $s_1$, $s_2$ in $\mathcal{T}$ with $s_1 \approx_{\mathcal{T}} s_2$:

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

(2) if $s_1 \rightarrow t_1$ in $\mathcal{T}$ s.t. $s_1 \not\approx_{\mathcal{T}} t_1$ then there is a path fragment $s_2\, u_1 \ldots u_n t_2$ in $\mathcal{T}$ with $s_2 \approx_{\mathcal{T}} u_i$ for $1 \leq i \leq n$ and $t_1 \approx_{\mathcal{T}} t_2$

Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

*Proof:* show that for all states $s_1$, $s_2$ in $\mathcal{T}$ with $s_1 \approx_{\mathcal{T}} s_2$:

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

(2) if $s_1 \to t_1$ in $\mathcal{T}$ s.t. $s_1 \not\approx_{\mathcal{T}} t_1$ then there is a path fragment $s_2\, u_1 \ldots u_n t_2$ in $\mathcal{T}$ with $s_2 \approx_{\mathcal{T}} u_i$ for $1 \le i \le n$ and $t_1 \approx_{\mathcal{T}} t_2$

$\surd$ by definition of $\approx_{\mathcal{T}}$

Claim: $\mathcal{R} = \{(s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2\}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

*Proof:* show that for all states $s_1$, $s_2$ in $\mathcal{T}$ with $s_1 \approx_{\mathcal{T}} s_2$:

(1) $L(s_1) = L(s_2)$ $\quad\surd$

(2) if $s_1 \to t_1$ in $\mathcal{T}$ s.t. $s_1 \not\approx_{\mathcal{T}} t_1$ then there is a path fragment $s_2\, u_1 \ldots u_n t_2$ in $\mathcal{T}$ with $s_2 \approx_{\mathcal{T}} u_i$ for $1 \leq i \leq n$ and $t_1 \approx_{\mathcal{T}} t_2$

$\surd$ by definition of $\approx_{\mathcal{T}}$

(3) $s_1$ is $\mathcal{R}$-divergent iff $s_2$ is $\mathcal{R}$-divergent

## $\approx_{\mathcal{T}}$ is finer than $\approx_{\mathcal{T}}^{\mathsf{div}}$



Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

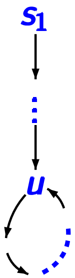
Proof of the divergence-sensitivity of $\mathcal{R}$:

suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent

Claim: $\mathcal{R} = \{(s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2\}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent

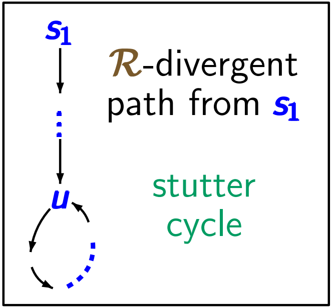



$\mathcal{R}$-divergent path from $s_1$

Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

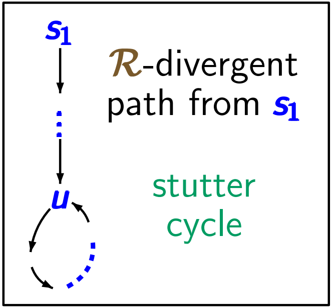
suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent



Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent

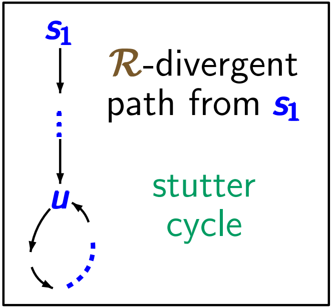



$u \approx_{\mathcal{T}} s_1 \approx_{\mathcal{T}} s_2$

Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent




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

$\downarrow$

*div*

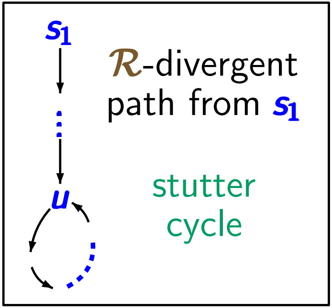
# $\approx_{\mathcal{T}}$ is finer than $\approx_{\mathcal{T}}^{\mathsf{div}}$

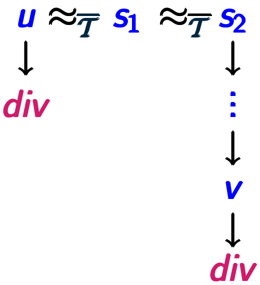


Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent

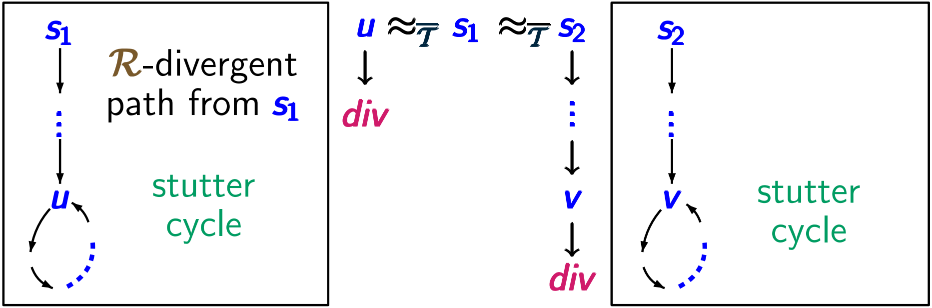


Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent



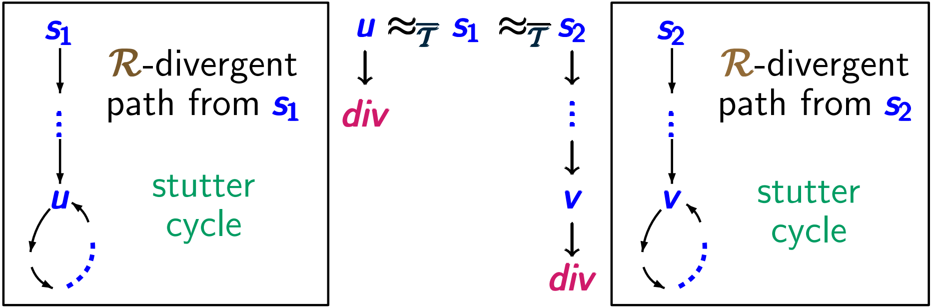
# $\approx_{\mathcal{T}}$ is finer than $\approx_{\mathcal{T}}^{\mathrm{div}}$



Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent



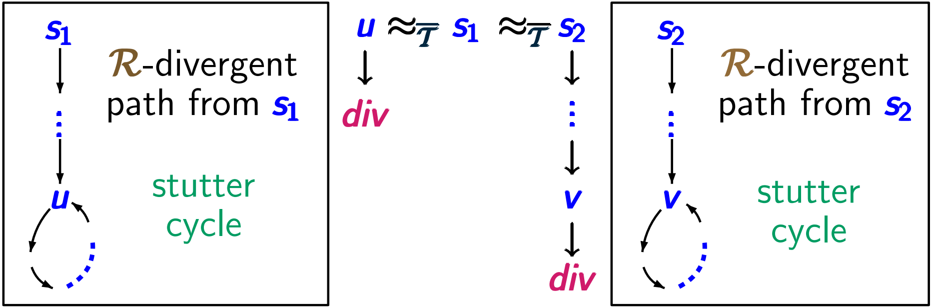
# $\approx_{\mathcal{T}}$ is finer than $\approx_{\mathcal{T}}^{\mathsf{div}}$



Claim: $\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}} s_2 \}$ is a divergence-sensitive stutter bisimulation for $\mathcal{T}$.

Proof of the divergence-sensitivity of $\mathcal{R}$:

suppose $s_1 \approx_{\mathcal{T}} s_2$, $s_1$ $\mathcal{R}$-divergent $\Longrightarrow$ $s_2$ $\mathcal{R}$-divergent



# $\approx_{\mathcal{T}}^{\text{div}}$ is finer than $\approx_{\mathcal{T}}$

$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \right\} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

# $\approx_{\mathcal{T}}^{\mathsf{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \right\} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

*Proof:*

$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \right\} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

*Proof:*

(1) labeling condition: $\surd$

# $\approx_{\mathcal{T}}^{\mathbf{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$

$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathbf{div}} s_2 \right\} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

*Proof:*

(1) labeling condition: $\checkmark$

(2) simulation condition:

$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx^{\text{div}}_{\mathcal{T}} s_2 \right\} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

*Proof:*

(1) labeling condition: $\checkmark$

(2) simulation condition:

- for the pair $(div, div) \in \mathcal{R}$: $\checkmark$

# $\approx_{\mathcal{T}}^{\mathsf{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \right\} \cup \{(div, div)\}$$
is a stutter bisimulation for $\overline{\mathcal{T}}$.

*Proof:*

(1) labeling condition: $\surd$

(2) simulation condition:

- for the pair $(div, div) \in \mathcal{R}$: $\surd$
- regard now a pair $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

# $\approx_{\mathcal{T}}^{\text{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\text{div}} s_2 \} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

# $\approx_{\mathcal{T}}^{\mathbf{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathbf{div}} s_2 \} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

show that:

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

can be completed to

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

$\mathcal{R}$

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

# $\approx^{\mathrm{div}}_{\mathcal{T}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx^{\mathrm{div}}_{\mathcal{T}} s_2 \} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $\boxed{s_1, s_2 \in S}$

show that:

$$\begin{tikzcd}
s_1 \arrow[d] & \approx^{\mathrm{div}}_{\mathcal{T}} & s_2 \\
s_1' & &
\end{tikzcd}$$

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

can be completed to

$$\begin{tikzcd}
s_1 \arrow[ddd] & \approx^{\mathrm{div}}_{\mathcal{T}} & s_2 \arrow[d] \arrow[dd, bend left, dash, "\approx^{\mathrm{div}}_{\mathcal{T}}"] \\
& & \vdots \arrow[d] \\
& & u_n \arrow[d] \\
s_1' & -\mathcal{R}- & s_2'
\end{tikzcd}$$

# $\approx^{\text{div}}_{\mathcal{T}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx^{\text{div}}_{\mathcal{T}} s_2 \} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

show that:

$$\begin{array}{ccc} s_1 & \approx^{\text{div}}_{\mathcal{T}} & s_2 \\ \downarrow & & \\ s_1' & & \end{array}$$

can be completed to

$$\begin{tikzcd} s_1 \arrow[ddd] & \approx^{\text{div}}_{\mathcal{T}} & s_2 \arrow[d] \arrow[dd, phantom, bend left=60, "\approx^{\text{div}}_{\mathcal{T}}"] \\ & & \vdots \arrow[d] \\ & & u_n \arrow[d] \\ s_1' & -\mathcal{R}- & s_2' \end{tikzcd}$$

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

$\boxed{\text{obvious if } s_1' \in S}$

# $\approx_{\mathcal{T}}^{\mathsf{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \} \cup \{(div, div)\}$
is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

show that:

$$\begin{tikzcd} s_1 \arrow[d] & \approx_{\mathcal{T}}^{\mathsf{div}} & s_2 \\ s_1' & & \end{tikzcd}$$

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

$\boxed{\text{consider } s_1' = div}$

can be completed to

$$\begin{tikzcd} s_1 \arrow[ddd] & \approx_{\mathcal{T}}^{\mathsf{div}} & s_2 \arrow[d] & \\ & & \vdots \arrow[d] & \approx_{\mathcal{T}}^{\mathsf{div}} \\ & & u_n \arrow[d] & \\ s_1' & -\mathcal{R}- & s_2' & \end{tikzcd}$$

# $\approx_{\mathcal{T}}^{\mathsf{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

$$\begin{tikzcd}
s_1 \arrow[d] & \approx_{\mathcal{T}}^{\mathsf{div}} & s_2 \\
div & &
\end{tikzcd}$$

# $\approx_{\mathcal{T}}^{\mathsf{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \right\} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

$$\begin{tikzcd}
s_1 \arrow[d] & \approx_{\mathcal{T}}^{\mathsf{div}} & s_2 \\
div & &
\end{tikzcd}$$

$s_1$ is on stutter cycle

# $\approx^{\mathsf{div}}_{\mathcal{T}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx^{\mathsf{div}}_{\mathcal{T}} s_2 \right\} \cup \{(div, div)\}$$
is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

$$\begin{tikzcd} s_1 \arrow[d] & \approx^{\mathsf{div}}_{\mathcal{T}} & s_2 \\ div & & \end{tikzcd}$$

$s_1$ is on stutter cycle

$$\Downarrow$$

$s_1, s_2$ $\approx^{\mathsf{div}}_{\mathcal{T}}$-divergent

$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\mathbf{div}} s_2 \right\} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

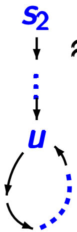
simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

$$\begin{array}{ccc} s_1 & \approx_{\mathcal{T}}^{\mathbf{div}} & s_2 \\ \downarrow & & \\ div & & \end{array}$$

$s_1$ is on stutter cycle

$\Downarrow$

$s_1, s_2$ $\approx_{\mathcal{T}}^{\mathbf{div}}$-divergent




$\approx_{\mathcal{T}}^{\mathbf{div}}$-divergent path from $s_2$

# $\approx_{\mathcal{T}}^{\text{div}}$ is finer than $\approx_{\overline{\mathcal{T}}}$



$$\mathcal{R} = \left\{ (s_1, s_2) \in S \times S : s_1 \approx_{\mathcal{T}}^{\text{div}} s_2 \right\} \cup \{(div, div)\}$$

is a stutter bisimulation for $\overline{\mathcal{T}}$.

simulation condition for $(s_1, s_2) \in \mathcal{R}$ where $s_1, s_2 \in S$

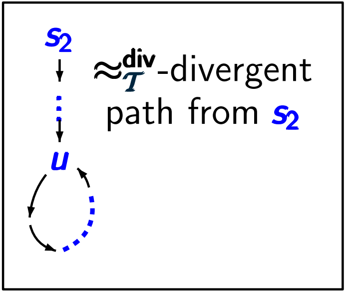
$s_1 \approx_{\mathcal{T}}^{\text{div}} s_2$

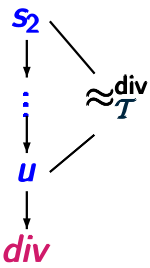
$s_1 \to div$

$s_1$ is on stutter cycle

$\Downarrow$

$s_1, s_2$ $\approx_{\mathcal{T}}^{\text{div}}$-divergent



# $\approx_{\mathcal{T}}^{\mathsf{div}}$ and $\approx_{\overline{\mathcal{T}}}$



For finite transition systems:

$$s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

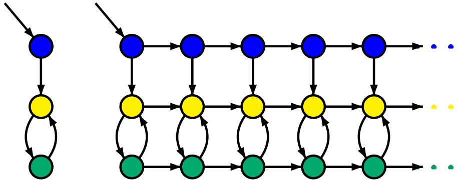
# $\approx_{\mathcal{T}}^{\mathsf{div}}$ and $\approx_{\mathcal{T}}$ for infinite systems

For finite transition systems:

$$s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \quad \text{iff} \quad s_1 \approx_{\mathcal{T}} s_2$$

**wrong** for infinite transition systems

$\mathcal{T}$



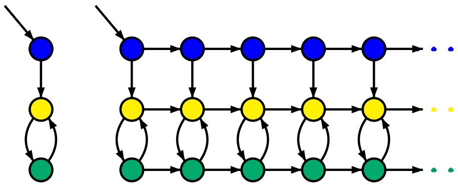
# $\approx_{\mathcal{T}}^{\mathsf{div}}$ and $\approx_{\overline{\mathcal{T}}}$ for infinite systems

For finite transition systems:

$$s_1 \approx_{\mathcal{T}}^{\mathsf{div}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

**wrong** for infinite transition systems

$\mathcal{T} \mathrel{\widehat{=}} \overline{\mathcal{T}}$



As there are no stutter cycles:

$\mathcal{T}$ agrees with the reachable part of $\overline{\mathcal{T}}$

# $\approx^{\mathsf{div}}_{\mathcal{T}}$ and $\approx_{\overline{\mathcal{T}}}$ for infinite systems

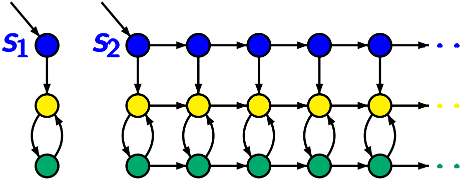


For finite transition systems:

$$s_1 \approx^{\mathsf{div}}_{\mathcal{T}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

**wrong** for infinite transition systems

$\mathcal{T} \mathrel{\widehat{=}} \overline{\mathcal{T}}$




As there are no stutter cycles:

$\mathcal{T}$ agrees with the reachable part of $\overline{\mathcal{T}}$

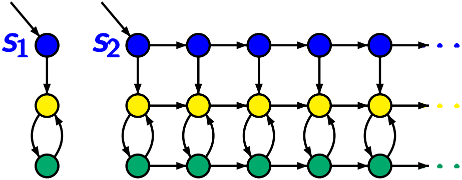
$s_1 \approx s_2$,

For finite transition systems:

$$s_1 \approx^{\mathsf{div}}_{\mathcal{T}} s_2 \quad \text{iff} \quad s_1 \approx_{\overline{\mathcal{T}}} s_2$$

**wrong** for infinite transition systems

$\mathcal{T} \,\widehat{=}\, \overline{\mathcal{T}}$




As there are no stutter cycles:

$\mathcal{T}$ agrees with the reachable part of $\overline{\mathcal{T}}$

$s_1 \approx s_2$, but $s_1 \not\approx^{\mathsf{div}} s_2$

# Computing $\mathcal{T}/\approx^{\mathsf{div}}$ for finite TS

# Computing $\mathcal{T}/{\approx^{\mathsf{div}}}$ for finite TS



| transition system $\mathcal{T}$ |
| --- |

atomic prop.: $AP$

# Computing $\mathcal{T}/{\approx^{\mathsf{div}}}$ for finite TS



transition system $\mathcal{T}$

$\downarrow$

divergence-sensitive expansion $\overline{\mathcal{T}}$

atomic prop.: $AP$

# Computing $\mathcal{T}/\approx^{\textsf{div}}$ for finite TS



$$\boxed{\text{transition system } \mathcal{T}} \longrightarrow \boxed{\text{divergence-sensitive expansion } \overline{\mathcal{T}}}$$

atomic prop.: $AP$

add trap state $div$

# Computing $\mathcal{T}/{\approx^{\mathsf{div}}}$ for finite TS



| | | |
|---|---|---|
| transition system $\mathcal{T}$ | $\downarrow$ | atomic prop.: $AP$ |
| divergence-sensitive expansion $\overline{\mathcal{T}}$ | | add trap state $div$<br>atomic prop.: $AP \cup \{div\}$ |

# Computing $\mathcal{T}/{\approx^{\mathbf{div}}}$ for finite TS



transition system $\mathcal{T}$

$\downarrow$

divergence-sensitive expansion $\overline{\mathcal{T}}$

$\downarrow$

quotient system $\overline{\mathcal{T}}/{\approx}$ w.r.t. stutter bisim.

atomic prop.: $AP$

add trap state $div$

atomic prop.: $AP \cup \{div\}$

# Computing $\mathcal{T}/\approx^{\text{div}}$ for finite TS



transition system $\mathcal{T}$

atomic prop.: $AP$

$\downarrow$

divergence-sensitive expansion $\overline{\mathcal{T}}$

add trap state $div$

atomic prop.: $AP \cup \{div\}$

$\downarrow$

quotient system $\overline{\mathcal{T}}/\approx$ w.r.t. stutter bisim.

$\downarrow$

quotient system $\mathcal{T}/\approx^{\text{div}}$ w.r.t. stutter bisim. with divergence

# Computing $\mathcal{T}/\approx^{\mathsf{div}}$ for finite TS



$$\boxed{\text{transition system } \mathcal{T}}$$
$$\downarrow$$
$$\boxed{\text{divergence-sensitive expansion } \overline{\mathcal{T}}}$$
$$\downarrow$$
$$\boxed{\text{quotient system } \overline{\mathcal{T}}/\approx \text{ w.r.t. stutter bisim.}}$$
$$\downarrow$$
$$\boxed{\text{quotient system } \mathcal{T}/\approx^{\mathsf{div}} \text{ w.r.t. stutter bisim. with divergence}}$$

atomic prop.: $AP$

add trap state *div*

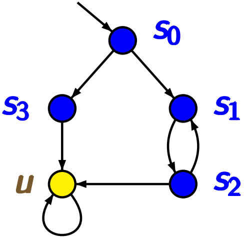
atomic prop.: $AP \cup \{div\}$

redirect $[s] \rightarrow div$

to $[s] \rightarrow [s]$

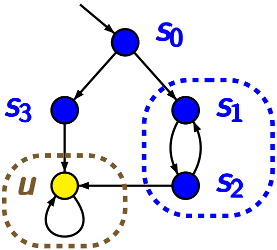
# Example: computing $\mathcal{T}/\approx^{\text{div}}$






$AP = \{blue\}$

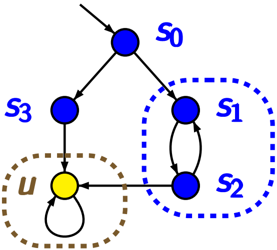
# Example: computing $\mathcal{T}/\approx^{\mathsf{div}}$



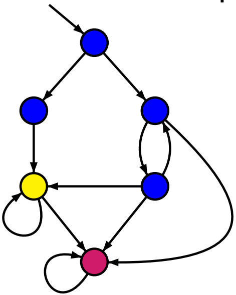

$AP = \{blue\}$

stutter cycles

$u\,u$ and $s_1\,s_2\,s_1$




div.-sensitive exp. $\overline{\mathcal{T}}$

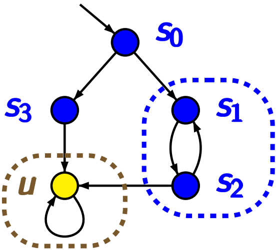


$AP = \{blue\}$

stutter cycles

$u\,u$ and $s_1\,s_2\,s_1$

# Example: computing $\mathcal{T}/\approx^{\text{div}}$

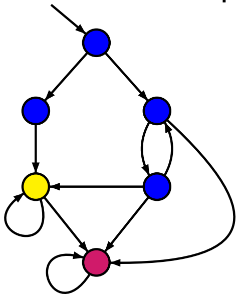



$AP = \{blue\}$

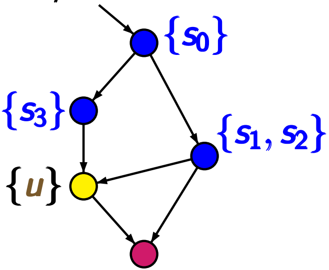
stutter cycles

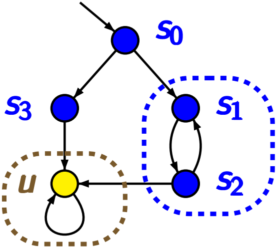
$u\,u$ and $s_1\,s_2\,s_1$

div.-sensitive exp. $\overline{\mathcal{T}}$



$\overline{\mathcal{T}}/\approx$



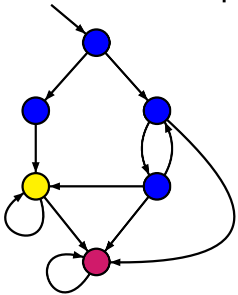



$AP = \{\mathit{blue}\}$

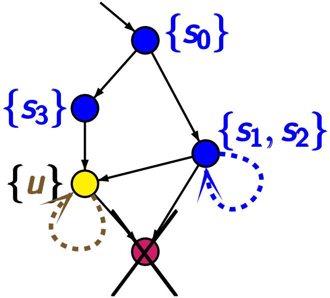
stutter cycles

$u\,u$ and $s_1\,s_2\,s_1$

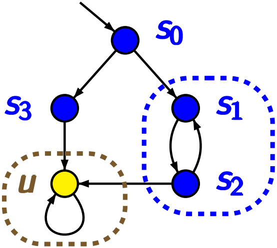
div.-sensitive exp. $\overline{\mathcal{T}}$



$\overline{\mathcal{T}}/\!\approx \;\rightsquigarrow\; \mathcal{T}/\!\approx^{\text{div}}$



# Example: computing $\mathcal{T}/{\approx^{\mathsf{div}}}$

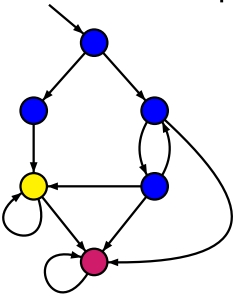



$AP = \{ \textit{blue} \}$

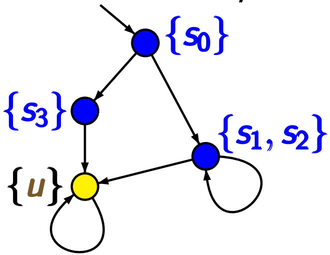
stutter cycles

$u\,u$ and $s_1\,s_2\,s_1$

div.-sensitive exp. $\overline{\mathcal{T}}$



$\mathcal{T}/{\approx^{\mathsf{div}}}$



# Summary: equivalences

# Summary: equivalences



| | bisimulation equivalence | stutter bisimulation with divergence | trace equivalence |
|---|---|---|---|
| temporal logic characteriz. | **CTL*** <br> **CTL** | $\mathbf{CTL^*}_{\setminus \bigcirc}$ <br> $\mathbf{CTL}_{\setminus \bigcirc}$ | **LTL** |
| | | | |

# Summary: equivalences



| | bisimulation equivalence | stutter bisimulation with divergence | trace equivalence |
|---|---|---|---|
| temporal logic characteriz. | CTL*<br>CTL | $\text{CTL}^*_{\setminus \bigcirc}$<br>$\text{CTL}_{\setminus \bigcirc}$ | LTL |
| equivalence checking | PTIME | PTIME | PSPACE-complete |

# Summary: equivalences



| | bisimulation equivalence | stutter bisimulation with divergence | trace equivalence |
|---|---|---|---|
| temporal logic characteriz. | CTL* <br> CTL | $\text{CTL}^*_{\setminus \bigcirc}$ <br> $\text{CTL}_{\setminus \bigcirc}$ | LTL |
| equivalence checking | PTIME <br> $\mathcal{O}(m \cdot \log \lvert S \rvert)$ | PTIME <br> $\mathcal{O}(m \cdot \log \lvert S \rvert)$ | PSPACE-complete |

where $AP$ is fixed and

$m$ = number of edges $\geq$ number of states = $|S|$

# Summary: equivalences



| | bisimulation equivalence | stutter bisimulation with divergence | trace equivalence |
|---|---|---|---|
| temporal logic characteriz. | CTL* <br> CTL | $CTL^*_{\setminus \bigcirc}$ <br> $CTL_{\setminus \bigcirc}$ | LTL |
| equivalence checking | PTIME <br> $\mathcal{O}(m \cdot \log \lvert S \rvert)$ | PTIME <br> $\mathcal{O}(m \cdot \log \lvert S \rvert)$ | PSPACE-complete |
| graph minimization | $\surd$ | $\surd$ | — |

where $AP$ is fixed and

$m$ = number of edges $\geq$ number of states = $|S|$