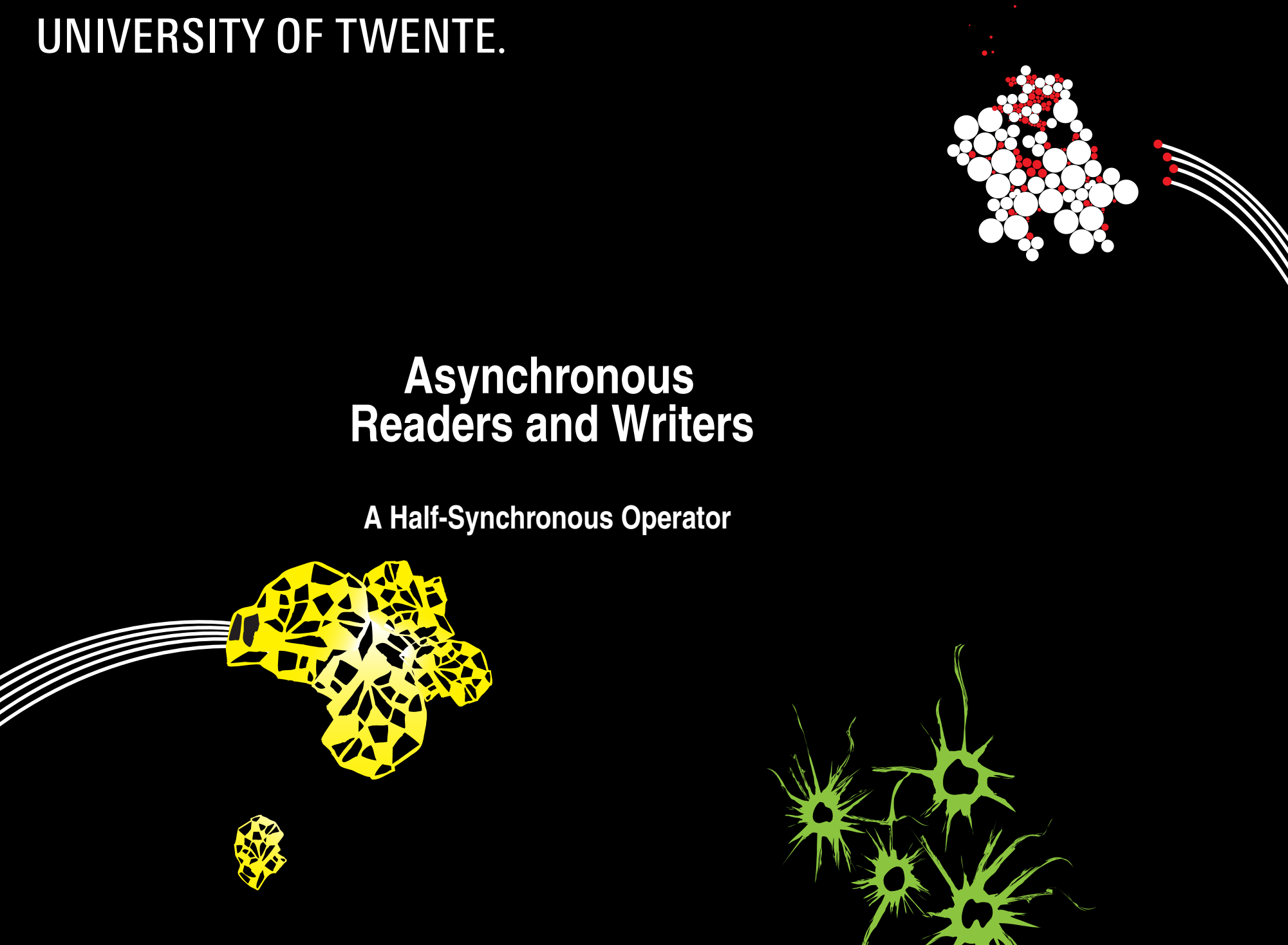
UNIVERSITY OF TWENTE.



Asynchronous Readers and Writers

A Half-Synchronous Operator

Overview

Introduction of the ⇓ together with the ¡ and ¿

! and ? versus ¡ and ¿

¡ and ¿ using $\dot{\square}$

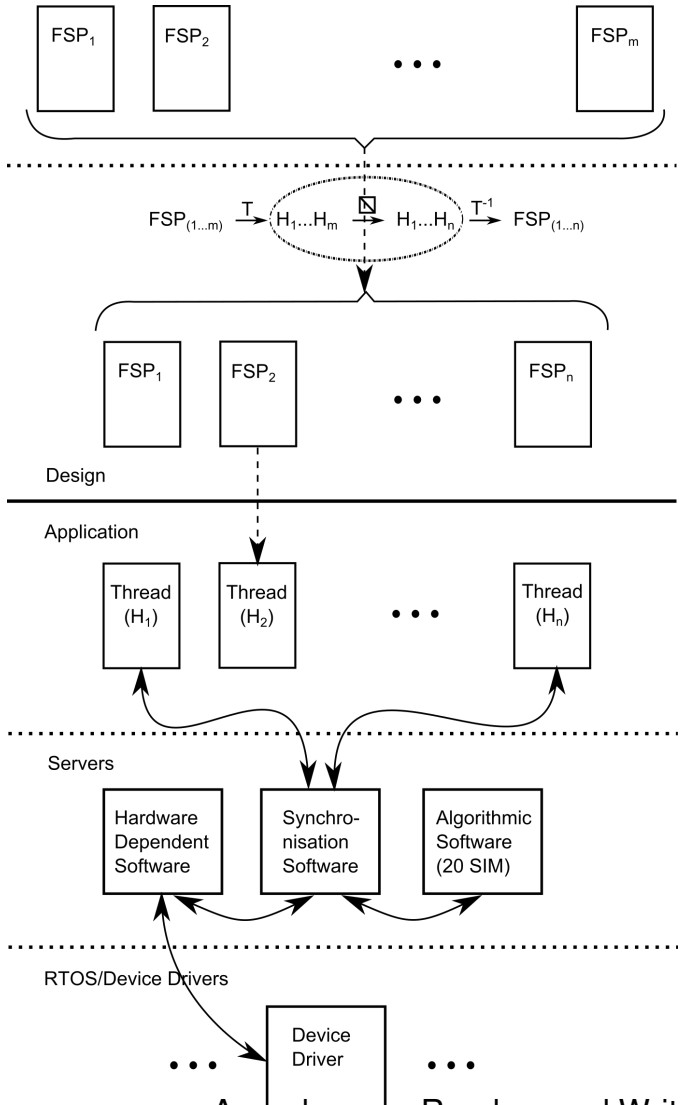
Design Level

Semantics of ⇓, ¡ and ¿

Example Application

The Future

Introduction, system overview



Introduction

- Purpose: Asynchronous Writing and Reading.

Introduction

- Purpose: Asynchronous Writing and Reading.
- In CSP writing over a channel is restricted to two processes interacting synchronously via an action containing the **!** and the **?**.

Introduction

- Purpose: Asynchronous Writing and Reading.
- In CSP writing over a channel is restricted to two processes interacting synchronously via an action containing the **!** and the **?**.
- Proposal: a *half-synchronous action* which allows a process to write a value x over a channel c,

Introduction

- Purpose: Asynchronous Writing and Reading.
- In CSP writing over a channel is restricted to two processes interacting synchronously via an action containing the **!** and the **?**.
- Proposal: a *half-synchronous action* which allows a process to write a value *x* over a channel *c*,
- without the requirement that the reading processes must be in a state where they can read the value *x* over a channel *c*.

Introduction

- Purpose: Asynchronous Writing and Reading.
- In CSP writing over a channel is restricted to two processes interacting synchronously via an action containing the **!** and the **?**.
- Proposal: a *half-synchronous action* which allows a process to write a value x over a channel c,
- without the requirement that the reading processes must be in a state where they can read the value x over a channel c.
- Together with a half-synchronous parallel alphabetised operator.

Introduction

Advantages of the half-synchronous operator $\Downarrow$ with half-synchronous actions containing ¡ or ¿:

Introduction

Advantages of the half-synchronous operator $\Downarrow$ with half-synchronous actions containing ¡ or ¿:

- it eases the complexity of the design eliminating arguably complex process specifications:

Introduction

Advantages of the half-synchronous operator $\Downarrow$ with half-synchronous actions containing ¡ or ¿:

- it eases the complexity of the design eliminating arguably complex process specifications:
  - it is not necessary to use a buffer process in the model to achieve asynchronous writing and reading,

Introduction

Advantages of the half-synchronous operator ⇓ with half-synchronous actions containing **¡** or **¿**:

- it eases the complexity of the design eliminating arguably complex process specifications:
  - it is not necessary to use a buffer process in the model to achieve asynchronous writing and reading,
  - the writes (**¡**) and reads (**¿**) are asynchronous, which makes it possible to have an order of writes and reads that, if synchronous (**!**, **?**), would lead to a deadlock,

Introduction

Advantages of the half-synchronous operator $\Downarrow$ with half-synchronous actions containing ¡ or ¿:

- it eases the complexity of the design eliminating arguably complex process specifications:
  - it is not necessary to use a buffer process in the model to achieve asynchronous writing and reading,
  - the writes (¡) and reads (¿) are asynchronous, which makes it possible to have an order of writes and reads that, if synchronous (**!**, **?**), would lead to a deadlock,
- by reducing the number of actions involved in this asynchronous writing and reading of the processes, improves the performance of the periodic hard real-time application,

Introduction

Advantages of the half-synchronous operator ⇓ with half-synchronous actions containing ¡ or ¿:

- it eases the complexity of the design eliminating arguably complex process specifications:
  - it is not necessary to use a buffer process in the model to achieve asynchronous writing and reading,
  - the writes (¡) and reads (¿) are asynchronous, which makes it possible to have an order of writes and reads that, if synchronous (**!**, **?**), would lead to a deadlock,
- by reducing the number of actions involved in this asynchronous writing and reading of the processes, improves the performance of the periodic hard real-time application,
- in a distributed computing system, for example a processor-coprocessor combination, the waiting time of the processor-coprocessor can be reduced.

Overview

Introduction of the ⇓ together with the ¡ and ¿

! and **?** versus **¡** and **¿**

¡ and ¿ using

Design Level

Semantics of ⇓, ¡ and ¿

Example Application

The Future

! and ? versus ¡ and ¿

Listing 1: Deadlock due to synchronous writing and reading

$A = c!x_1 \rightarrow c!y_1 \rightarrow d?x_2 \rightarrow d?y_2 \rightarrow SKIP$

$B = c?x_1 \rightarrow d!x_2 \rightarrow c?y_1 \rightarrow d!y_2 \rightarrow SKIP$

$AB = A||B$

! and ? versus ¡ and ¿

Listing 1: Deadlock due to synchronous writing and reading

$A = c!x_1 \rightarrow c!y_1 \rightarrow d?x_2 \rightarrow d?y_2 \rightarrow SKIP$

$B = c?x_1 \rightarrow d!x_2 \rightarrow c?y_1 \rightarrow d!y_2 \rightarrow SKIP$

$AB = A||B$

trace: $c.x_1$

! and ? versus ¡ and ¿

Listing 1: Deadlock due to synchronous writing and reading

$A = c!x_1 \rightarrow c!y_1 \rightarrow d?x_2 \rightarrow d?y_2 \rightarrow SKIP$

$B = c?x_1 \rightarrow d!x_2 \rightarrow c?y_1 \rightarrow d!y_2 \rightarrow SKIP$

$AB = A||B$

trace: $c.x_1$

Listing 2: No deadlock due to asynchronous writing and reading

$A = c¡x_1 \rightarrow c¡y_1 \rightarrow d¿x_2 \rightarrow d¿y_2 \rightarrow SKIP$

$B = c¿x_1 \rightarrow d¡x_2 \rightarrow c¿y_1 \rightarrow d¡y_2 \rightarrow SKIP$

$AB = A \Downarrow B$

! and ? versus ¡ and ¿

Listing 1: Deadlock due to synchronous writing and reading

$A = c!x_1 \rightarrow c!y_1 \rightarrow d?x_2 \rightarrow d?y_2 \rightarrow SKIP$

$B = c?x_1 \rightarrow d!x_2 \rightarrow c?y_1 \rightarrow d!y_2 \rightarrow SKIP$

$AB = A||B$

trace: $c.x_1$

Listing 2: No deadlock due to asynchronous writing and reading

$A = c¡x_1 \rightarrow c¡y_1 \rightarrow d¿x_2 \rightarrow d¿y_2 \rightarrow SKIP$

$B = c¿x_1 \rightarrow d¡x_2 \rightarrow c¿y_1 \rightarrow d¡y_2 \rightarrow SKIP$

$AB = A \Downarrow B$

many possible traces, for example:

$c¡x_1 \rightarrow c¡y_1 \rightarrow c¿x_1 \rightarrow d¡x_2 \rightarrow c¿y_1 \rightarrow d¡y_2 \rightarrow d¿x_2 \rightarrow d¿y_2 \rightarrow SKIP$

Overview

Introduction of the ⇓ together with the ¡ and ¿

! and ? versus ¡ and ¿

¡ and ¿ using 

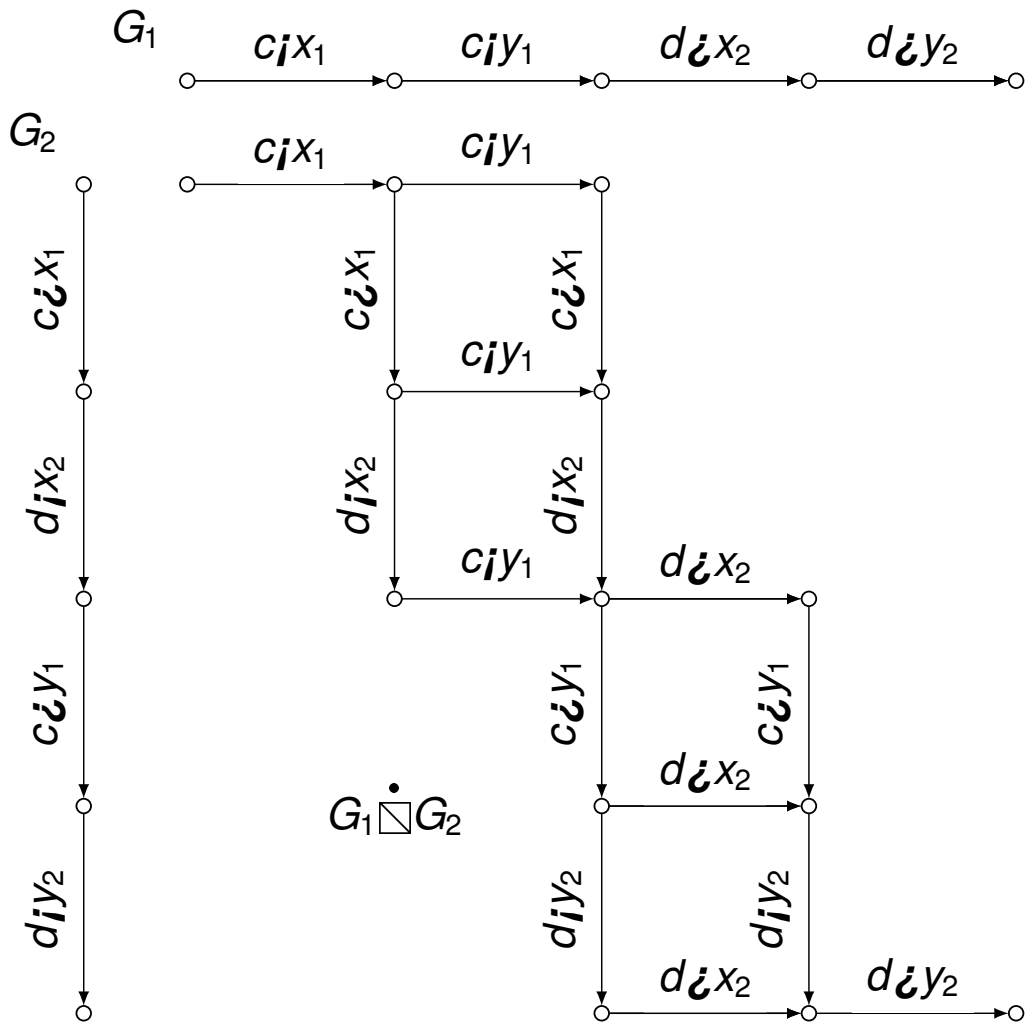
Design Level

Semantics of ⇓, ¡ and ¿

Example Application

The Future

¡ and ¿ using $\dot{\boxtimes}$



Overview

Introduction of the $\Downarrow$ together with the ¡ and ¿

! and ? versus ¡ and ¿

¡ and ¿ using

Design Level

Semantics of $\Downarrow$, ¡ and ¿

Example Application

The Future

Design Level

Separation of write/read actions in time

Design Level

Separation of write/read actions in time
Needs a Buffer

Design Level

Separation of write/read actions in time
Needs a Buffer

$$A = write.x \rightarrow \text{SKIP}$$
$$B = read.x \rightarrow \text{SKIP}$$
$$Buffer = write.x \rightarrow read.x \rightarrow \text{SKIP}$$
$$AB = A||B||Buffer$$

Design Level

Separation of write/read actions in time
Needs a Buffer

$$
\begin{aligned}
A &= write.x \rightarrow \text{SKIP} \\
B &= read.x \rightarrow \text{SKIP} \\
Buffer &= write.x \rightarrow read.x \rightarrow \text{SKIP} \\
AB &= A||B||Buffer
\end{aligned}
$$

$$
\begin{aligned}
A &= c!x \rightarrow \text{SKIP} \\
B &= c?x \rightarrow \text{SKIP} \\
AB &= A \Downarrow B
\end{aligned}
$$

Overview

Introduction of the ⇓ together with the ¡ and ¿

! and ? versus ¡ and ¿

¡ and ¿ using $\dot{\square}$

Design Level

Semantics of ⇓, ¡ and ¿

Example Application

The Future

Semantics of ⇓, ¡ and ¿

$$\frac{P \overset{c\,¡\,x:T}{\rightsquigarrow} P',\ Q_1 \overset{c\,¿\,x:T}{\longrightarrow} Q_1', \cdots, Q_n \overset{c\,¿\,x:T}{\longrightarrow} Q_n'}{P \Downarrow Q_1 \Downarrow \cdots \Downarrow Q_n \overset{c\,¡\,x:T}{\rightsquigarrow} P' \Downarrow Q_1 \Downarrow \cdots \Downarrow Q_n \overset{c\,¿\,x:T}{\longrightarrow} P' \Downarrow Q_1' \Downarrow \cdots \Downarrow Q_n'}\ ,\quad c\,¿\,x : T \notin (X, Z)$$

Semantics of ⇓, ¡ and ¿

$$\frac{P \overset{c\,¡\,x:T}{\rightsquigarrow} P',\ Q_1 \overset{c\,¿\,x:T}{\rightarrow} Q'_1, \cdots, Q_n \overset{c\,¿\,x:T}{\rightarrow} Q'_n}{P \Downarrow Q_1 \Downarrow \cdots \Downarrow Q_n \overset{c\,¡\,x:T}{\rightsquigarrow} P' \Downarrow Q_1 \Downarrow \cdots \Downarrow Q_n \overset{c\,¿\,x:T}{\rightarrow} P' \Downarrow Q'_1 \Downarrow \cdots \Downarrow Q'_n}, \quad c\,¿\,x : T \notin (X, Z)$$

$$\frac{Q_i \overset{c\,¿\,x:T}{\rightarrow} Q'_i,\ Q_j \overset{y}{\rightarrow} Q'_j}{Q_i \Downarrow Q_j \overset{y}{\rightarrow} Q_i \Downarrow Q'_j}, \ y \neq c\,¿\,x : T, c\,¿\,x : T \in (Y_i \cdot Y_j), \ y \notin (X, Y_{k=1,\cdots n, j\neq k}, Z)$$

Semantics of ⇓, ¡ and ¿

$$\frac{P \overset{c\,¡\,x:T}{\rightsquigarrow} P',\ Q_1 \overset{c\,¿\,x:T}{\rightarrow} Q_1', \cdots, Q_n \overset{c\,¿\,x:T}{\rightarrow} Q_n'}{P \Downarrow Q_1 \Downarrow \cdots \Downarrow Q_n \overset{c\,¡\,x:T}{\rightsquigarrow} P' \Downarrow Q_1 \Downarrow \cdots \Downarrow Q_n \overset{c\,¿\,x:T}{\rightarrow} P' \Downarrow Q_1' \Downarrow \cdots \Downarrow Q_n'}, \quad c\,¿\,x : T \notin (X, Z)$$

$$\frac{Q_i \overset{c\,¿\,x:T}{\rightarrow} Q_i',\ Q_j \overset{y}{\rightarrow} Q_j'}{Q_i \Downarrow Q_j \overset{y}{\rightarrow} Q_i \Downarrow Q_j'}, \ y \neq c\,¿\,x : T, c\,¿\,x : T \in (Y_i \cdot Y_j), \ y \notin (X, Y_{k=1,\cdots n, j\neq k}, Z)$$

$$\frac{P \rightsquigarrow P',\ Q_i \overset{c\,¿\,x:T}{\rightarrow} Q_i'}{P \rightsquigarrow P'}, (\alpha(\rightsquigarrow) \cdot (Y_1, \cdots, Y_n, Z)) = \varnothing$$

Semantics of ⇓, ¡ and ¿

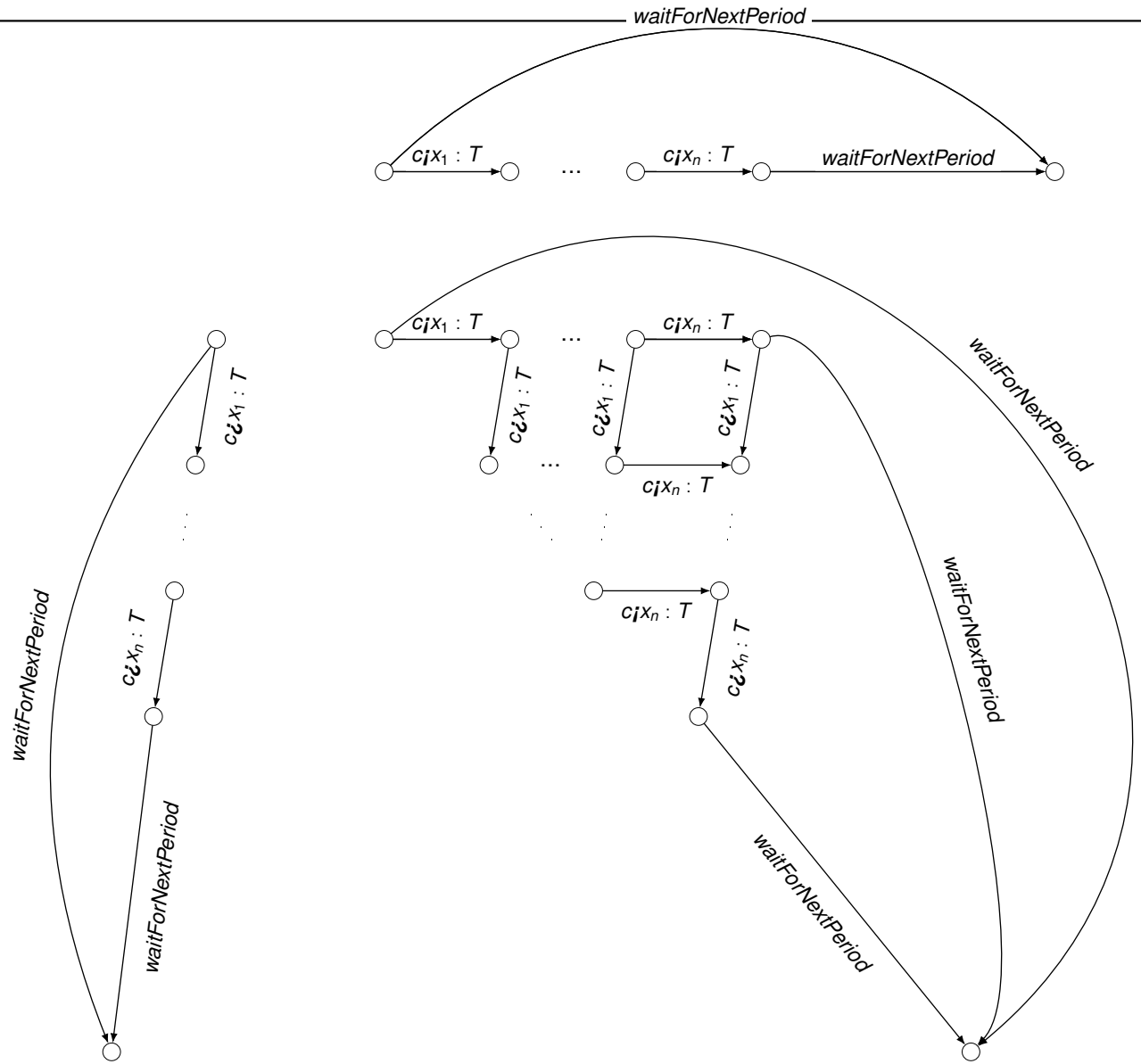
$$\frac{P \overset{c\,¡\,x:T}{\rightsquigarrow} P',\ Q_1 \overset{c\,¿\,x:T}{\rightarrow} Q'_1, \cdots, Q_n \overset{c\,¿\,x:T}{\rightarrow} Q'_n}{P \Downarrow Q_1 \Downarrow \cdots \Downarrow Q_n \overset{c\,¡\,x:T}{\rightsquigarrow} P' \Downarrow Q_1 \Downarrow \cdots \Downarrow Q_n \overset{c\,¿\,x:T}{\rightarrow} P' \Downarrow Q'_1 \Downarrow \cdots \Downarrow Q'_n}, \quad c\,¿\,x : T \notin (X, Z)$$

$$\frac{Q_i \overset{c\,¿\,x:T}{\rightarrow} Q'_i,\ Q_j \overset{y}{\rightarrow} Q'_j}{Q_i \Downarrow Q_j \overset{y}{\rightarrow} Q_i \Downarrow Q'_j}, \ y \neq c\,¿\,x : T, c\,¿\,x : T \in (Y_i \cdot Y_j), \ y \notin (X, Y_{k=1,\cdots n, j\neq k}, Z)$$

$$\frac{P \rightsquigarrow P',\ Q_i \overset{c\,¿\,x:T}{\rightarrow} Q'_i}{P \rightsquigarrow P'}, (\alpha(\rightsquigarrow) \cdot (Y_1, \cdots, Y_n, Z)) = \varnothing$$

$$\frac{Q_i \overset{c\,¡\,x:T}{\rightarrow} Q'_i,\ Q_j \overset{c\,¡\,x:T}{\rightarrow} Q'_j}{SKIP}, i \neq j$$

Example


waitForNextPeriod
$c_i x_1 : T$
$c_i x_n : T$
waitForNextPeriod
$c_i x_1 : T$
$c_i x_n : T$
waitForNextPeriod
$c_i x_1 : T$
$c_i x_n : T$
$c_i x_1 : T$
$c_i x_n : T$
$c_i x_n : T$
$c_i x_n : T$
$c_i x_1 : T$
$c_i x_n : T$
waitForNextPeriod
waitForNextPeriod
waitForNextPeriod
waitForNextPeriod

Overview

Case study

$$
\begin{aligned}
\mathit{Application} \;=\; & c_1\,!\,x_1 : T \rightarrow c_2\,?\,y_1 : T \rightarrow \\
& \ldots \\
& c_1\,!\,x_8 : T \rightarrow c_2\,?\,y_8 : T \rightarrow \\
& \mathit{display\_f}(y_1, \cdots, y_8) \rightarrow \mathrm{SKIP}
\end{aligned}
$$

Case study

$$
\begin{aligned}
\mathit{Application} \;=\; & c_1\,!\,x_1 : T \rightarrow c_2\,?\,y_1 : T \rightarrow \\
& \ldots \\
& c_1\,!\,x_8 : T \rightarrow c_2\,?\,y_8 : T \rightarrow \\
& \mathit{display\_f}(y_1, \cdots, y_8) \rightarrow \mathrm{SKIP}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Controller} \;=\; & c_1\,?\,x_1 : T \rightarrow \mathit{writeCoProc}.x_1 \rightarrow \\
& \mathit{readCoProc}.y_1 \rightarrow c_2\,!\,y_1 : T \rightarrow \\
& \ldots \\
& c_1\,?\,x_8 : T \rightarrow \mathit{writeCoProc}.x_8 \rightarrow \\
& \qquad \mathit{readCoProc}.y_8 \rightarrow c_2\,!\,y_8 : T \rightarrow \mathit{SKIP}
\end{aligned}
$$

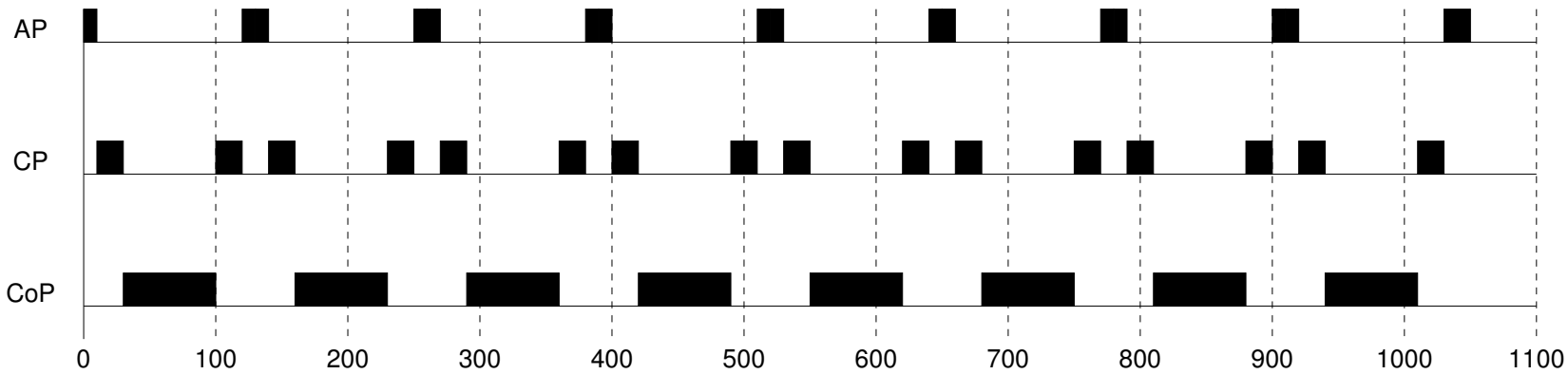
Case study

$$
\begin{aligned}
\mathit{Application} \;=\; & c_1\,!\,x_1 : T \rightarrow c_2\,?\,y_1 : T \rightarrow \\
& \ldots \\
& c_1\,!\,x_8 : T \rightarrow c_2\,?\,y_8 : T \rightarrow \\
& \mathit{display\_f}(y_1, \cdots, y_8) \rightarrow \mathrm{SKIP}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Controller} \;=\; & c_1\,?\,x_1 : T \rightarrow \mathit{writeCoProc}.x_1 \rightarrow \\
& \mathit{readCoProc}.y_1 \rightarrow c_2\,!\,y_1 : T \rightarrow \\
& \ldots \\
& c_1\,?\,x_8 : T \rightarrow \mathit{writeCoProc}.x_8 \rightarrow \\
& \qquad \mathit{readCoProc}.y_8 \rightarrow c_2\,!\,y_8 : T \rightarrow \mathit{SKIP}
\end{aligned}
$$

$$
\mathit{System}_1 \;=\; \mathit{Application}_{A} ||_{C} \mathit{Controller}
$$

Example application



Example application

$$\begin{aligned} \mathit{Application} \;=\; & c_1 \text{¡} x_1 : T \rightarrow \cdots \rightarrow c_1 \text{¡} x_8 : T \rightarrow \\ & c_2 \text{¿} y_1 : T \rightarrow \cdots \rightarrow c_2 \text{¿} y_8 : T \rightarrow \\ & \mathit{display\_f}(y_1, \cdots, y_8) \rightarrow \mathrm{SKIP} \end{aligned}$$

Example application

$$
\begin{aligned}
\mathit{Application} \;=\; & c_1 \,!\, x_1 : T \rightarrow \cdots \rightarrow c_1 \,!\, x_8 : T \rightarrow \\
& c_2 \,?\, y_1 : T \rightarrow \cdots \rightarrow c_2 \,?\, y_8 : T \rightarrow \\
& \mathit{display\_f}(y_1, \cdots, y_8) \rightarrow \mathrm{SKIP}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Controller} \;=\; & c_1 \,?\, x_1 : T \rightarrow \mathit{writeCoProc}.x_1 \rightarrow \\
& \mathit{readCoProc}.y_1 \rightarrow c_2 \,!\, y_1 : T \rightarrow \\
& \cdots \\
& c_1 \,?\, x_8 : T \rightarrow \mathit{writeCoProc}.x_8 \rightarrow \\
& \mathit{readCoProc}.y_8 \rightarrow c_2 \,!\, y_8 : T \rightarrow \mathit{SKIP}
\end{aligned}
$$

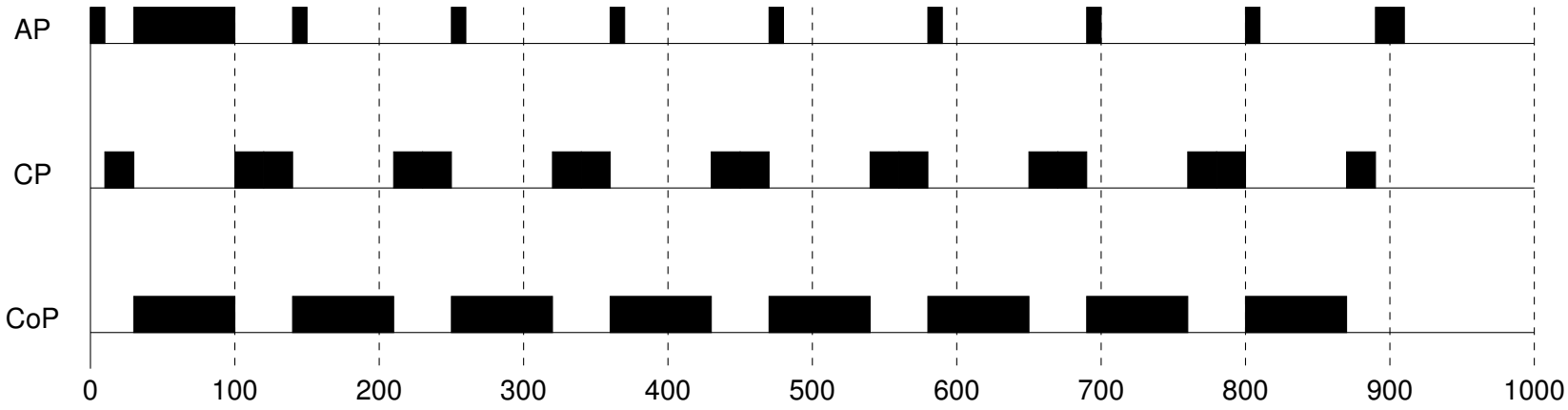
Example application

$$
\begin{aligned}
\mathit{Application} \;=\; & c_1 \,¡\, x_1 : T \rightarrow \cdots \rightarrow c_1 \,¡\, x_8 : T \rightarrow \\
& c_2 \,¿\, y_1 : T \rightarrow \cdots \rightarrow c_2 \,¿\, y_8 : T \rightarrow \\
& \mathit{display\_f}(y_1, \cdots, y_8) \rightarrow \mathrm{SKIP}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{Controller} \;=\; & c_1 \,¿\, x_1 : T \rightarrow \mathit{writeCoProc}.x_1 \rightarrow \\
& \mathit{readCoProc}.y_1 \rightarrow c_2 \,¡\, y_1 : T \rightarrow \\
& \cdots \\
& c_1 \,¿\, x_8 : T \rightarrow \mathit{writeCoProc}.x_8 \rightarrow \\
& \mathit{readCoProc}.y_8 \rightarrow c_2 \,¡\, y_8 : T \rightarrow \mathit{SKIP}
\end{aligned}
$$

$$
\mathit{System}_2 \;=\; \mathit{Application}_{A \Downarrow_C}\; \mathit{Controller}
$$

Example application



Overview

Introduction of the ⇓ together with the ¡ and ¿

! and ? versus ¡ and ¿

¡ and ¿ using

Design Level

Semantics of ⇓, ¡ and ¿

Example Application

The Future

The Future

Future work:

The Future

Future work:

- Index the half-synchronous action such that it is set-wise asynchronous and intra-set-wise synchronous,

The Future

Future work:

- Index the half-synchronous action such that it is set-wise asynchronous and intra-set-wise synchronous,
- elaborate the graph-theoretical characteristics of $\dot{\boxslash}$ (VRSP together with the half-synchronous operator),

The Future

Future work:

- Index the half-synchronous action such that it is set-wise asynchronous and intra-set-wise synchronous,
- elaborate the graph-theoretical characteristics of $\dot{\boxslash}$ (VRSP together with the half-synchronous operator), i.e. it is a commutative monoid of consistent graphs,

The Future

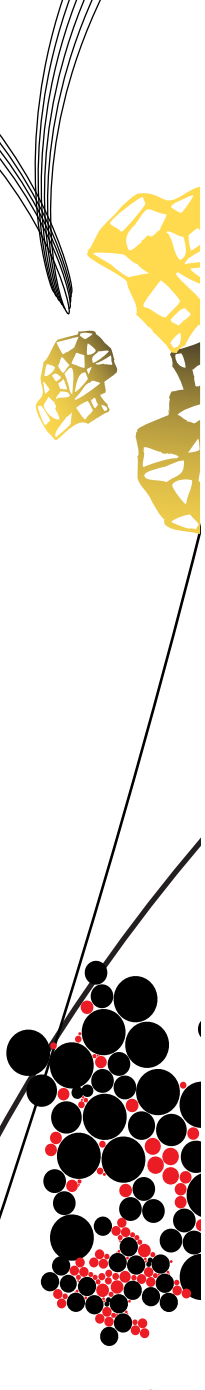
Future work:

- Index the half-synchronous action such that it is set-wise asynchronous and intra-set-wise synchronous,
- elaborate the graph-theoretical characteristics of $\dot{\boxslash}$ (VRSP together with the half-synchronous operator), i.e. it is a commutative monoid of consistent graphs,
- implementation in a tool-chain

The Future

Future work:

- Index the half-synchronous action such that it is set-wise asynchronous and intra-set-wise synchronous,
- elaborate the graph-theoretical characteristics of $\dot{\boxslash}$ (VRSP together with the half-synchronous operator), i.e. it is a commutative monoid of consistent graphs,
- implementation in a tool-chain and
- perform a case-study on a periodic hard real-time system.



Thanks!