Taiming Hierarchical Connectors

José Proença & Alexandre Madeira

CISTER Research Center in Real-Time & Embedded Computing Systems

INESCTEC

universidade de aveiro

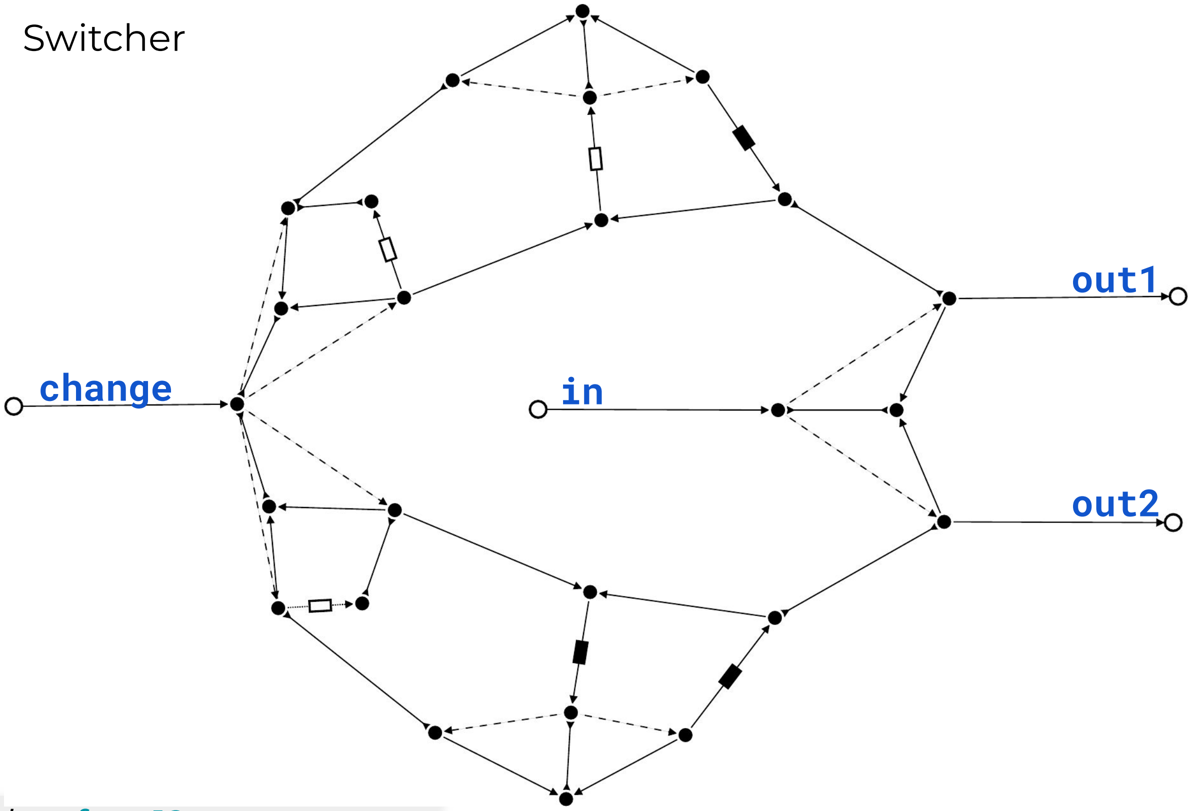
Universidade do Minho

Slides: *fsen19.proenca.org*

Download paper *Download slides*

Motivation

Switcher



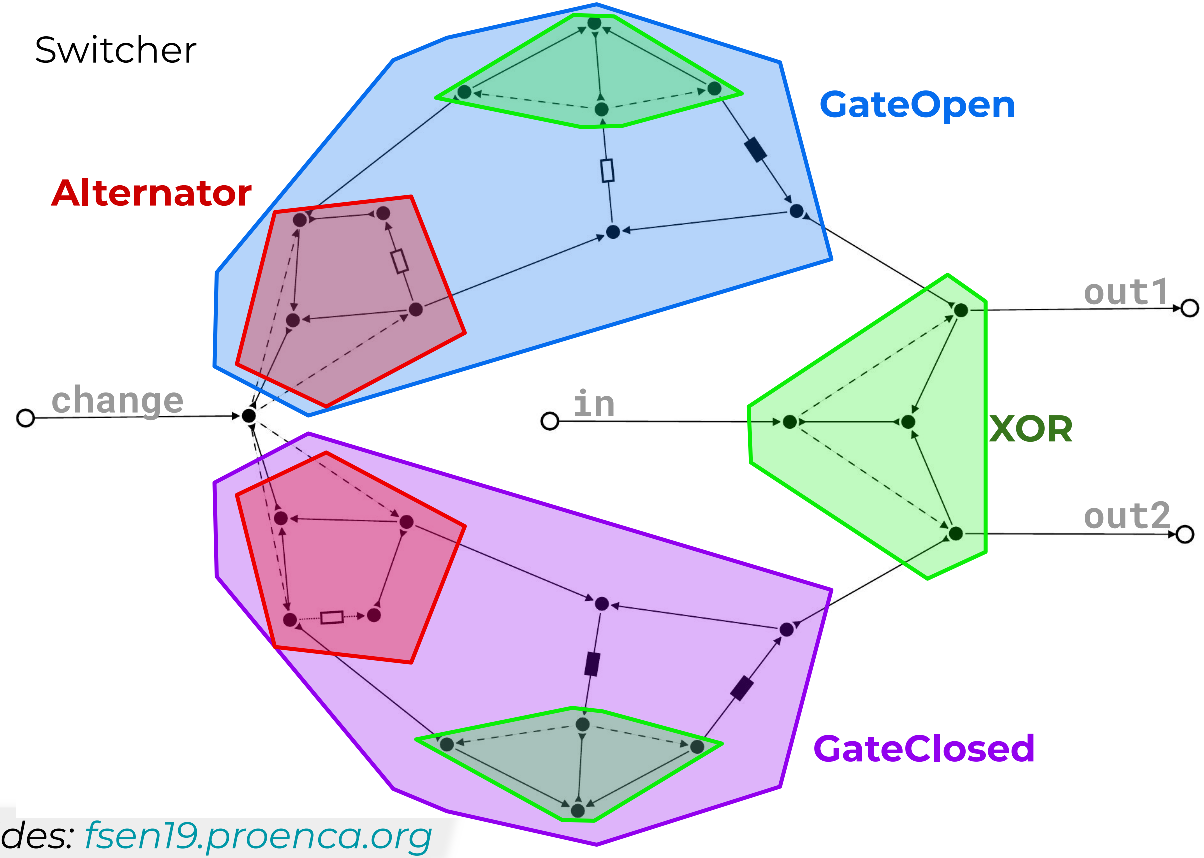
How to:

- Build
- **Verify behaviour**

Look ***inside*** the connector

Slides: *fsen19.proenca.org*

Motivation



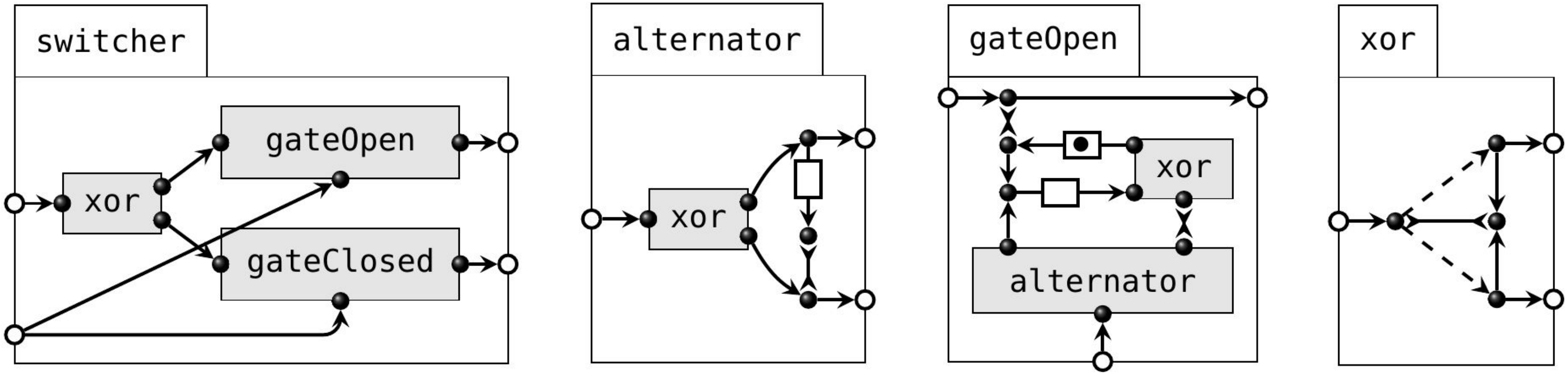
How to:

- Build
- **Verify behaviour**

Look ***inside*** the connector

Slides: *fsen19.proenca.org*

Reason over nested *containers*



Containers: { switcher, alternator, gateOpen, xor, - - - -➤ , ─□→ , ──→ , ...}

Specify and **verify** properties over containers

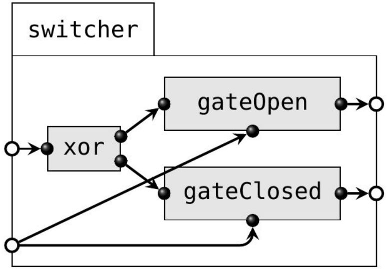
Modal logic

mCRL2
analysing system behaviour

Slides: fsen19.proenca.org

Outline

Hierarchical connectors



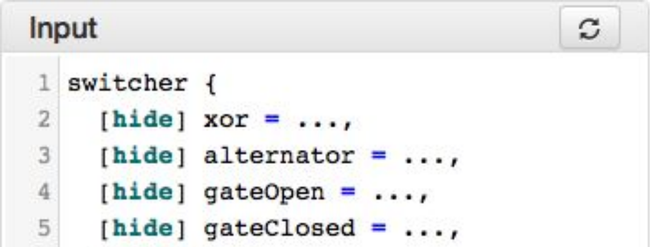
Logic over containers

$$\langle \mathsf{all}^* \ .\ \texttt{gateOpen}\rangle$$

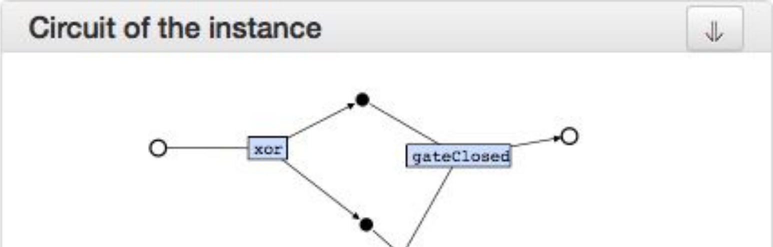
$$@_{\texttt{gateOpen}}\ \langle \overline{\texttt{alternator}}^{*} \rangle$$

$$\partial \langle \texttt{gateOpen}\rangle\ \texttt{true}$$

Online tools http://194.117.30.117

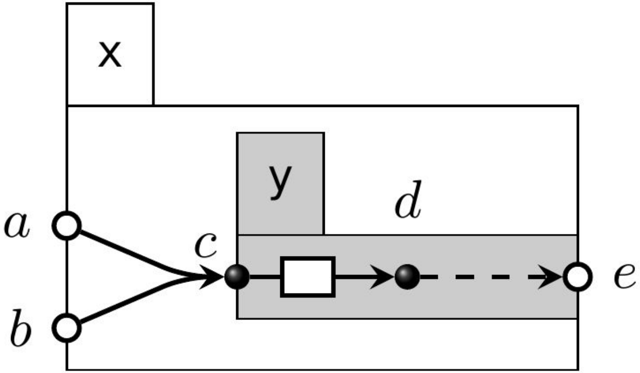


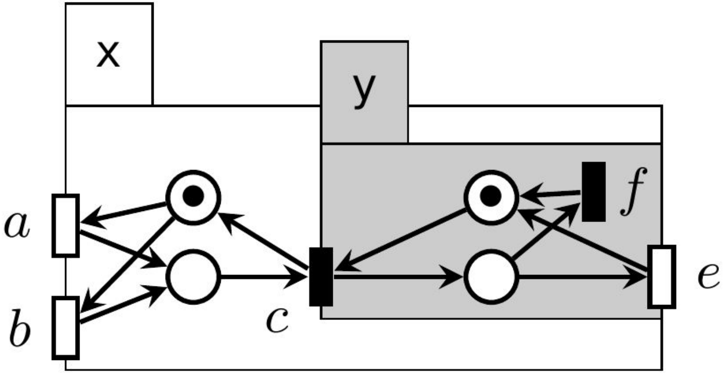
```
switcher {
  [hide] xor = ...,
  [hide] alternator = ...,
  [hide] gateOpen = ...,
  [hide] gateClosed = ...,
```



Slides: fsen19.

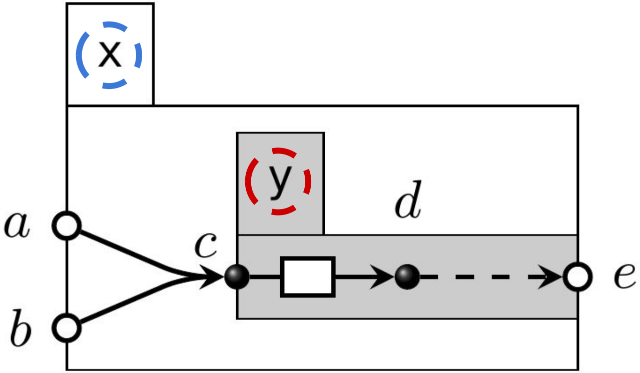
Hierarchical connectors

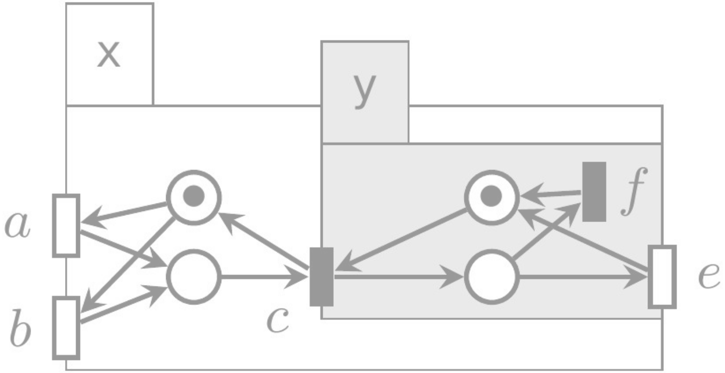


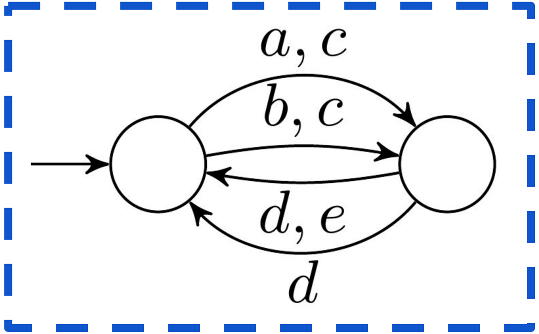


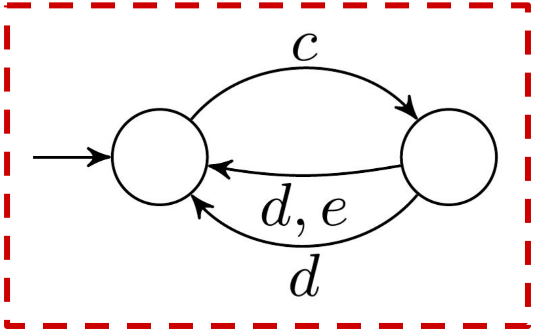
Slides: fsen19.proenca.org

Hierarchical connectors



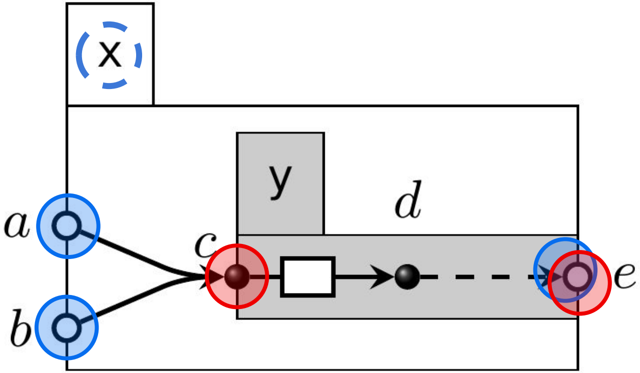


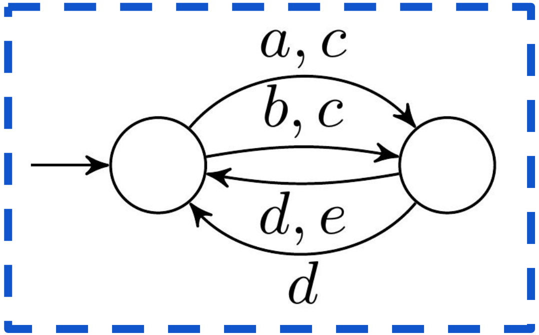


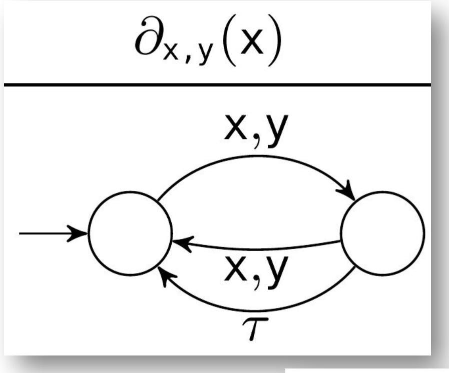


Slides: fsen19.proenca.org

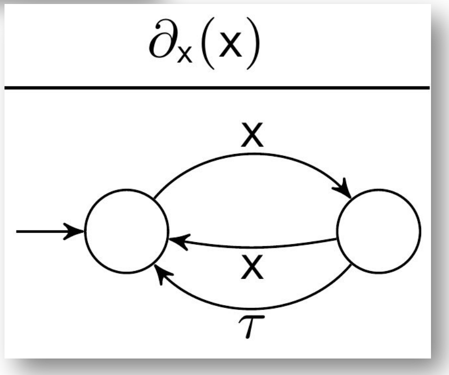
Container Abstraction





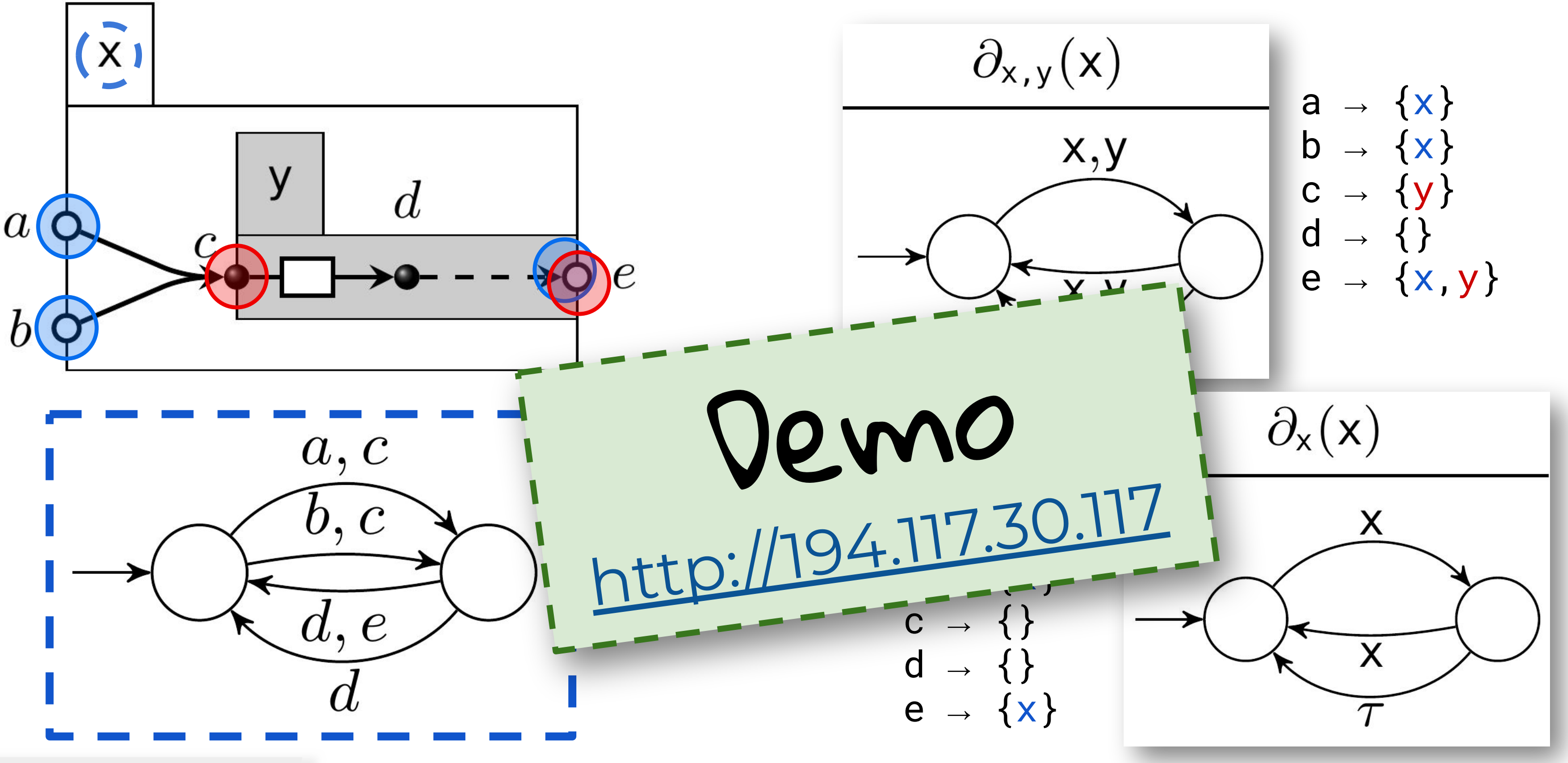


a	→	{x}
b	→	{x}
c	→	{y}
d	→	{}
e	→	{x, y}



a	→	{x}
b	→	{x}
c	→	{}
d	→	{}
e	→	{x}

Container Abstraction



a → {x}
b → {x}
c → {y}
d → {}
e → {x, y}

c → {}
d → {}
e → {x}

Demo

http://194.117.30.117

Slides: fsen19.proenca.org

Logic for Containers

$$\psi := \text{true} \mid \text{false} \mid \langle\phi\rangle\,\psi \mid [\phi]\,\psi \mid @_c\,\psi \mid \partial\,\psi \qquad \text{(state formula)}$$

$$\phi := \varphi \mid \phi^* \mid \phi + \phi \mid \phi\,.\,\phi \qquad \text{(regular formula)}$$

$$\varphi := c \mid \tau \mid \text{all} \mid \text{none} \mid \overline{\varphi} \mid \varphi + \varphi \mid \varphi\,\&\,\varphi \qquad \text{(action formula)}$$

Based on mCRL2 analysing system behaviour

Hennessy-Milner with regular modalities

Logic for Containers

Enter “c”

Go to parent

$$\psi := \text{true} \mid \text{false} \mid \langle\phi\rangle\,\psi \mid [\phi]\,\psi \mid @_c\,\psi \mid \partial\,\psi \qquad \text{(state formula)}$$

$$\phi := \varphi \mid \phi^* \mid \phi+\phi \mid \phi\,.\,\phi \qquad \text{(regular formula)}$$

$$\varphi := c \mid \tau \mid \text{all} \mid \text{none} \mid \overline{\varphi} \mid \varphi+\varphi \mid \varphi\,\&\,\varphi \qquad \text{(action formula)}$$

Based on mCRL2 analysing system behaviour

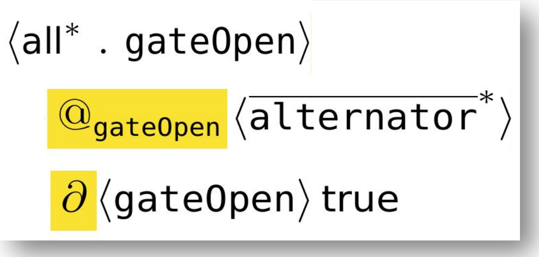
Hennessy-Milner with
regular modalities

Logic for Containers

$\psi := \mathsf{true} \mid \mathsf{false} \mid \langle\phi\rangle\,\psi \mid [\phi]\,\psi \mid @_c\,\psi \mid \partial\,\psi$ (state formula)

$\phi := \varphi \mid \phi^* \mid \phi + \phi \mid \phi \,.\, \phi$ (regular formula)

$\varphi := c \mid \tau \mid \mathsf{all} \mid \mathsf{none} \mid \overline{\varphi} \mid \varphi + \varphi \mid \varphi \,\&\, \varphi$ (action formula)



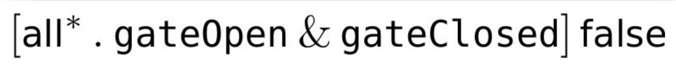
$\langle \mathsf{all}^* \,.\, \mathtt{gateOpen}\rangle$

$@_{\mathtt{gateOpen}}\langle\overline{\mathtt{alternator}}^*\rangle$

$\partial\langle\mathtt{gateOpen}\rangle\,\mathsf{true}$

At some point,

- **gateOpen** can interact twice
- without the **alternator** interacting



$[\mathsf{all}^* \,.\, \mathtt{gateOpen} \,\&\, \mathtt{gateClosed}]\,\mathsf{false}$

gateOpen and **gateClosed** cannot interact in the same step

Logic for Containers

$\psi := \text{true} \mid \text{false} \mid \langle\phi\rangle\,\psi \mid [\phi]\,\psi \mid @_c\,\psi \mid \partial\psi$ (state formula)

$\phi := \varphi \mid \phi^* \mid \phi + \phi \mid \phi$ (regular formula)

$\varphi := c \mid \tau \mid \text{all} \mid \text{r}$ (action formula)

Demo

http://194.117.30.117

$\langle \text{all}^* \,.\, \texttt{gateOpen} \rangle$

$@_{\texttt{gateOpen}} \langle \overline{\texttt{alternator}}^* \rangle$

$\partial \langle \texttt{gateOpen} \rangle\, \text{true}$

At some point,

- gateOpen can interact twice
- without the **alternator** interacting

$[\text{all}^* \,.\, \texttt{gateOpen} \,\&\, \texttt{gateClosed}]\, \text{false}$

gateOpen and **gateClosed** cannot interact in the same step

Gained insights

mCRL2
analysing system behaviour

is **fast!**

Fine control over "***hiding***" is helpful:

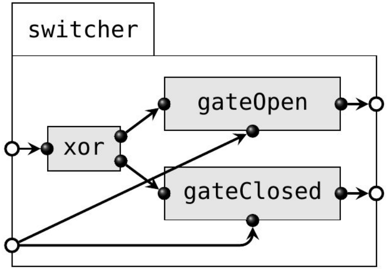
- For more **complex** connectors
- To reason about the **internals**
- To improve **performance** of model checking

What is a "***τ***" (tau)?

- *(there is room for improvements)*
- ***τ*** of `gateOpen`?
- Weak bisimilarity down to `gateOpen`, `gateClosed`

Wrap up

Hierarchical connectors



Container abstraction

Logic over containers

$$\langle \mathsf{all}^* \;.\; \mathtt{gateOpen}\rangle$$

$$@_{\mathtt{gateOpen}}\langle\overline{\mathtt{alternator}}^*\rangle$$

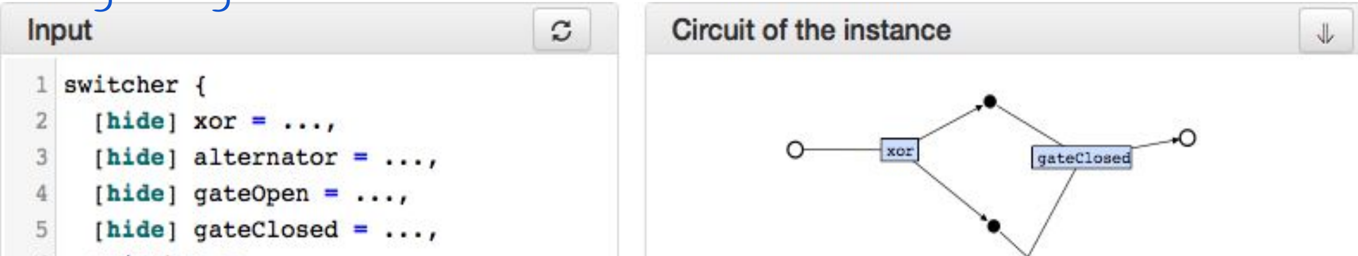
$$\partial\langle\mathtt{gateOpen}\rangle\,\mathsf{true}$$

Hiding mCRL2 generated details

Thank you

Online tools http://194.117.30.117

Still getting better...



```
switcher {
  [hide] xor = ...,
  [hide] alternator = ...,
  [hide] gateOpen = ...,
  [hide] gateClosed = ...,
```

Slides: *fsen19.proenca.org*