Modelling Typed Connectors in mCRL2

Rúben Cruz
Supervisor: José Carlos Espírito Santo
Co-Supervisor: José Proença

Departamento de Matemática e Aplicações
Universidade do Minho

Departamento de Informática
Universidade do Minho

March 16, 2018

Summary

Context

Goals

Web Site

- Improve Layout;
- Improve Graph Design;
- Add mCRL2 Support;
- Add Modal Logic Support

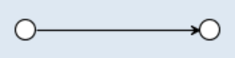
Typed Connectors

- Model typed connectors semantic in mCRL2;
- Extend it to families of typed connectors;
- Adapt a modal logic to verify connector families

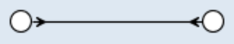
Reo

Reo is a language for coordination between components that compose individual processes.
Composed of nodes and primitive channels which enable the flow of data between the components.

Sync

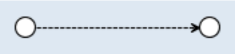


SyncDrain

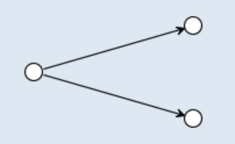


Fifo

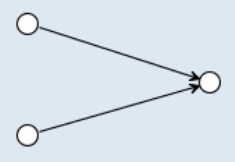
LossySync



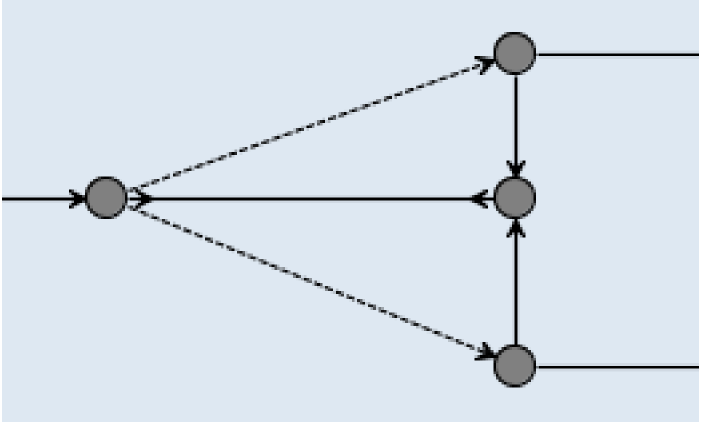
Duplicator



Merger

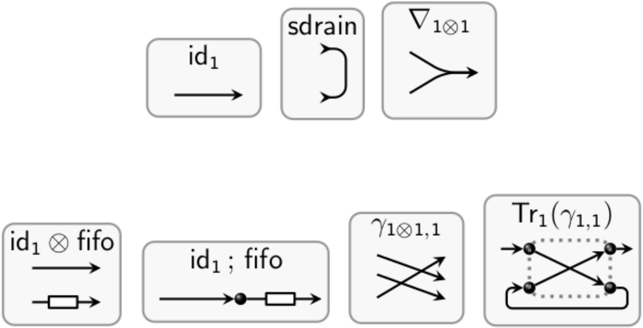


Reo - example



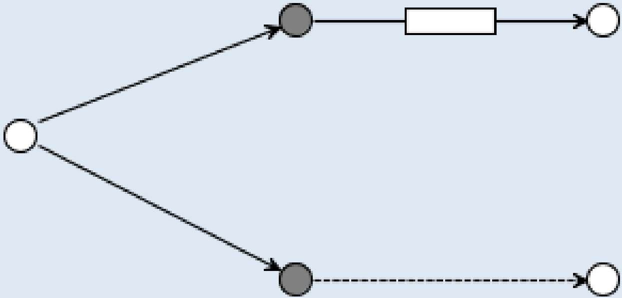
Reo - Typed Connectors

- Pointfree approach to describe a reo connector
- Primitives describe basic channels (e.g. fifo)
- We combine the primitives to form complex connectors



Reo - Typed Connectors

```
dupl ; (fifo * lossy)
```



mCRL2

- Specification language and associated toolset;
- Used to model and analyse communicating processes;
- Uses process algebra to model the behaviour;
- Uses μ-calculus to verify properties over a model;

Anatomy

- Actions - atomic elements of a program;
- Processes - combine and manage comunication between actions;
    - Composition (`P . Q`)
    - Paralellism (`P` $\|$ `Q`)
    - Multiactions (`a|b`)
    - Communication ($\Gamma_R(P)$)
    - Block ($\partial_R(P)$)

mCRL2 - anatomy of a program

$$P1 = b|c$$
$$P2 = \partial_{\{d,b\}}(\Gamma_{\{d|b->e\}}((a.d)\|P1))$$

Web Site

Web Site - Main Changes

- Improved graph layout (using new javascript library)
- Improved page layout
- added mCRL2 output box for concrete instances of connectors
- Responsiveness
- Port automata of the instance (for teaching purposes)

Connector Calculus in mCRL2

Goals

1. Convert Tile Model to Port Automata
2. Prove correctness of 1
3. Adapt mCRL2 model from Port Automata
4. Prove correctness of 3

Primitive Conversion

$$Chan : prim \times 2^{port} \times 2^{port} \times \mathbb{N} \rightarrow mCRL2_{prim}$$
$$Node : port \times port \times \mathbb{N} \rightarrow mCRL2_{node}$$

Chan(fifo, {a}, {b}, n)	$Fifo_n =$ a''.b'' . $Fifo_n$
Chan(id(1), {a}, {b}, n)	$Sync_n =$ a'' | b'' . $Sync_n$
Chan(lossy, {a}, {b}, n)	$Lossy_n =$ (a'' + a''|b'') . $Lossy_n$
Chan(dupl, {a}, {b, c}, n)	$Dupl_n =$ a''|b''|c'' . $Dupl_n$
Chan(merger, {a, b}, {c}, n)	$Merger_n =$ (a''|c'' + b''|c'') . $Merger_n$
Chan(fifofull, {a}, {b}, n)	$FifoFull_n =$ b''.a''. $FifoFull_n$
Node(a, b, n)	$Node_n =$ a'|b' . $Node_n$

Typed Connector Conversion

$$Convert : TC \rightarrow 2^{mCRL2_{prim}} \times 2^{mCRL2_{node}} \times port^* \times port^*$$

Convert(prim) $=$
(Chan(prim, $\{a\}$, $\{b\}$, n), $\emptyset$, [a], [b])

Convert(c1 ; c2) $=$
let (ch1, no1, in1, out1) $=$ Convert(c1);
(ch2, no2, in2, out2) $=$ Convert(c2);
$new\_nodes = \{Node(out1_i, in2_i) | 0 \leq i \leq \#out1\}$;
in
$(ch1 \cup ch2,\ no1 \cup no2 \cup new\_nodes,$ in1, out2$)$

Syncronization

$Init : 2^{mCRL2_{node}} \to 2^{mCRL2_{init}}$

$\mathbf{Init(\emptyset)} = \tau$

$\mathbf{Init(}\{h\} \cup t\mathbf{)} =$

let

$\quad \mathrm{C} = \{\mathrm{Chan(e)} \mid e \in P \wedge lnk(e,h) \wedge e \notin Init(t)\}$

in

$\quad \partial_{H_h}(\Gamma_{C_h}(Init(t) \| h \| \prod_{c \in C} c)$

Example: dupl; fifo*lossy

Channel Conversion

d	Chan(dupl, $\{d1\}$, $\{d2, d3\}$)	Dupl $= d1'' \mid d2'' \mid d3''$. Dupl
f	Chan(fifo, $\{f1\}$, $\{f2\}$)	Fifo $= f1''.f2''$. Fifo
l	Chan(lossy, $\{l1\}$, $\{l2\}$)	Lossy $= l1'' + l1'' \mid l2''$. Lossy

ϕ = fifo*lossy conversion

Conversion(fifo)	({f}, $\emptyset$, $[f1]$, $[f2]$)
Conversion(lossy)	({l}, $\emptyset$, $[l1]$, $[l2]$)
Conversion(fifo*lossy)	({f, l}, $\emptyset$, $[f1, l1]$, $[f2, l2]$)

Example: dupl; fifo*lossy

Nodes

n_1	Node($d2$, $f1$, 1)	$Node_1 = d2' \mid f1'.Node_1$
n_2	Node($d3$, $l1$, 2)	$Node_2 = d3' \mid l1'.Node_2$

ϕ = dupl; fifo*lossy conversion

Conversion(dupl)	$(\{\mathrm{d}\}, \emptyset, [d1], [d2, d3])$
Conversion(fifo*lossy)	$(\{\mathrm{f}, \mathrm{l}\}, \emptyset, [f1, l1], [f2, l2])$
Conversion(ϕ)	$(\{\mathrm{f}, \mathrm{l}, \mathrm{d}\}, \{n_1, n_2\}, [d1], [f2, l2])$

syncronization

Init($\emptyset$)	τ
Init($\{n_1\}$)	$\partial_{d2', d2'', f1', f1''}(\Gamma_{d2' \mid d2'' \rightarrow d2, f1' \mid f1'' \rightarrow f1}(Init(\emptyset) \| Node_1 \| Dupl \| Fifo))$
Init($\{n_2, n_1\}$)	$\partial_{d3', d3'', l1', l1''}(\Gamma_{d3' \mid d3'' \rightarrow d3, l1' \mid l1'' \rightarrow l1}(Init(\{n_1\}) \| Node_2 \| Lossy))$

Current and Future Work

Future Work

Connector Calculus

- Explore mCRL2 semantics of families of connector calculus;
- Adapt a modal logic to verify a given connector;
- (Possibly) Define the semantics of connector calculus in alloy

web site

- Implement a server based web site;
- Show Color Semantics;
- Add a model logic input and output box

Modelling Typed Connectors in mCRL2

Rúben Cruz
Supervisor: José Carlos Espírito Santo
Co-Supervisor: José Proença

Departamento de Matemática e Aplicações
Universidade do Minho

Departamento de Informática
Universidade do Minho

March 16, 2018