

10. Introduction to mCRL2

David Pereira José Proença Eduardo Tovar

RAMDE 2021/2022

Requirements and Model-driven Engineering

CISTER – ISEP

Porto, Portugal

<https://cister-labs.github.io/ramde2122>

<http://mcr12.org>

- Formal [specification language](#) with an associated toolset
- Used for [modelling](#), [validating](#) and [verifying](#) concurrent systems and protocols
- Tool suggestion: use [mcr12ide](#) (not mcr12-gui)

Recall CCS semantics

$$\begin{array}{c} \text{(act)} \\ \hline \alpha.P \xrightarrow{\alpha} P \end{array} \quad \begin{array}{c} \text{(sum-1)} \\ \frac{P_1 \xrightarrow{\alpha} P'_1}{P_1 + P_2 \xrightarrow{\alpha} P'_1} \end{array} \quad \begin{array}{c} \text{(sum-2)} \\ \frac{P_2 \xrightarrow{\alpha} P'_2}{P_1 + P_2 \xrightarrow{\alpha} P'_2} \end{array}$$
$$\begin{array}{c} \text{(res)} \\ \frac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L} \quad \alpha \notin L \end{array} \quad \begin{array}{c} \text{(rel)} \\ \frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]} \end{array}$$
$$\begin{array}{c} \text{(com1)} \\ \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \end{array} \quad \begin{array}{c} \text{(com2)} \\ \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'} \end{array} \quad \begin{array}{c} \text{(com3)} \\ \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau_a} P'|Q'} \end{array}$$

Processes in mCRL2

Syntax (by example)

$$a.0 \rightarrow a$$

$$a.P \rightarrow a.P$$

$$P_1 + P_2 \rightarrow P_1 + P_2$$

$$P \setminus L \rightarrow \mathit{block}(L, P)$$

$$P[f] \rightarrow \mathit{rename}(f, P)$$

$$a.P \mid \bar{a}.Q \rightarrow \mathit{comm}(\{a_1 \mid a_2 \rightarrow a\}, a_1.P \parallel a_2.Q)$$

$$a.P \mid \bar{a}.Q \setminus \{a\} \rightarrow \mathit{block}(\{a_1, a_2\}, \mathit{comm}(\{a_1 \mid a_2 \rightarrow a\}, a_1.P \parallel a_2.Q))$$

Syntax (by example)

$$a.0 \rightarrow a$$
$$a.P \rightarrow a.P$$
$$P_1 + P_2 \rightarrow P_1 + P_2$$
$$P \setminus L \rightarrow \mathit{block}(L, P)$$
$$P[f] \rightarrow \mathit{rename}(f, P)$$
$$a.P \mid \bar{a}.Q \rightarrow \mathit{hide}(\{a\}, \mathit{comm}(\{a_1 \mid a_2 \rightarrow a\}, a_1.P \parallel a_2.Q))$$
$$a.P \mid \bar{a}.Q \setminus \{a\} \rightarrow \mathit{hide}(\{a\}, \mathit{block}(\{a_1, a_2\}, \mathit{comm}(\{a_1 \mid a_2 \rightarrow a\}, a_1.P \parallel a_2.Q)))$$

$$CM = \text{coin}.\overline{\text{coffee}}.CM$$

$$CS = \text{pub}.\overline{\text{coin}}.\text{coffee}.CS$$

$$SmUni = (CM|CS)\{\text{coin}, \text{coffee}\}$$
act

```
coin, coin', coinCom,
coffee, coffee', coffeeCom, pub;
```

proc

```
CM = coin.coffee'.CM;
CS = pub.coin'.coffee.CS;
SmUni = block({coffee, coffee', coin, coin'},
  comm({coffee|coffee' → coffeeCom,
    coin|coin' → coinCom},
  CM || CS ));
```

init

```
SmUni;
```

The screenshot shows the mCRL2 IDE interface with the following components:

- Title Bar:** mCRL2 IDE - CM
- Toolbars:** A toolbar with icons for file operations (mCRL2, CRL2, mCRL2) and four icons for parsing, simulation, state space generation, and verification, all enclosed in a red dashed box. To the right are icons for properties and a question mark.
- Code Editor:** Contains the following code:

```
1 act
2   coin, coin', coinCom,
3   coffee, coffee', coffeeCom, pub;
4 proc
5   CM = coin.coffee'.CM;
6   CS = pub.coin'.coffee.CS;
7   SmUni = block({coffee,coffee', coin, coin'},
8               comm({coffee|coffee' -> coffeeCom,
9                   coin|coin' -> coinCom},
10              CM||CS ));
11 init
12   SmUni;
```
- Properties Panel:** A panel titled "Properties" with the text "No properties have been defined".
- Console Panel:** A panel titled "Console" with tabs for "Parsing" (selected), "Simulation", "State Space Generation", and "Verification".

Parse

Simulate

Visualize

Minimize &

Visualize

Specifications *.mcr12

act

```
action1, action2, ...;  
action3, action4 : Type;
```

proc

```
P1 = ...;  
P2(x: Bool) = ...;  
    % Process expression
```

init

```
SmUni;
```

```
sort List = struct  
    empty | cons(A,List);
```

```
map sum2: Int # Int → Int;
```

```
var x, y: Int;
```

eqn

```
sum2(x,y) = (x+y) * (x+y);  
    % Data patterns & expressions
```

https://mcr12.org/web/user_manual/language_reference/index.html

Process Expressions

$$P = PE ;$$

a *Action*

a|b *Multi-action*

P *Process*

delta *Deadlock*

a(DataExpr) *Parameterized Act.*

P(DataExpr) *Parameterized Proc.*

a.PE *Sequencing*

PE1 + PE2 *Choice*

PE1 || PE2 *Parallel*

block({a,b},PE) *Block*

allow({a,b},PE) *Allow*

rename({a→b},PE) *Rename*

comm({a|b→c},PE) *Communicate*

sum m: Nat . PE *Gen. Choice*

$P(\text{exp})$

true *Boolean*

42 *Pos, Nat, Int, Real*

!exp *Not*

exp && exp *And*

exp || exp *Or*

exp => exp *Implies*

forall n:Nat . exp *For all*

exists n:Nat . exp *Exists*

exp + exp *Sum*

max(exp, exp) *And*

exp mod exp *Remainder of div.*

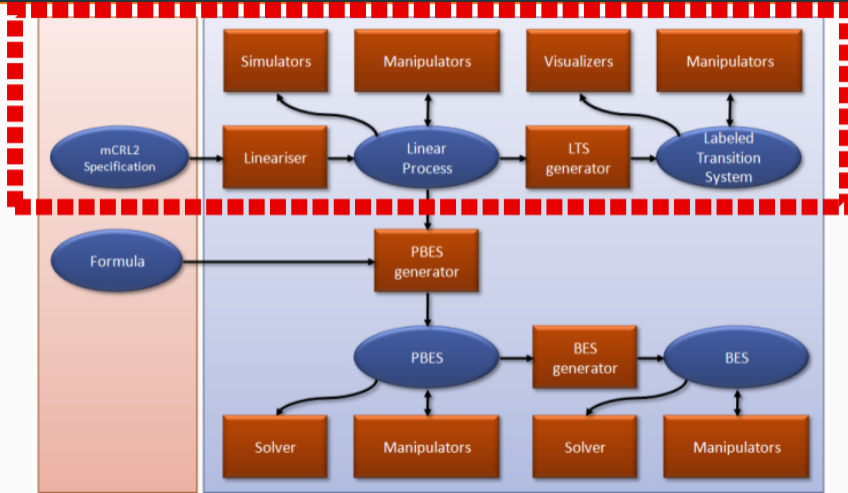
[exp, exp, ...] *List*

{exp, exp, ...} *Set*

{exp:2, exp:1, ...} *Bag*

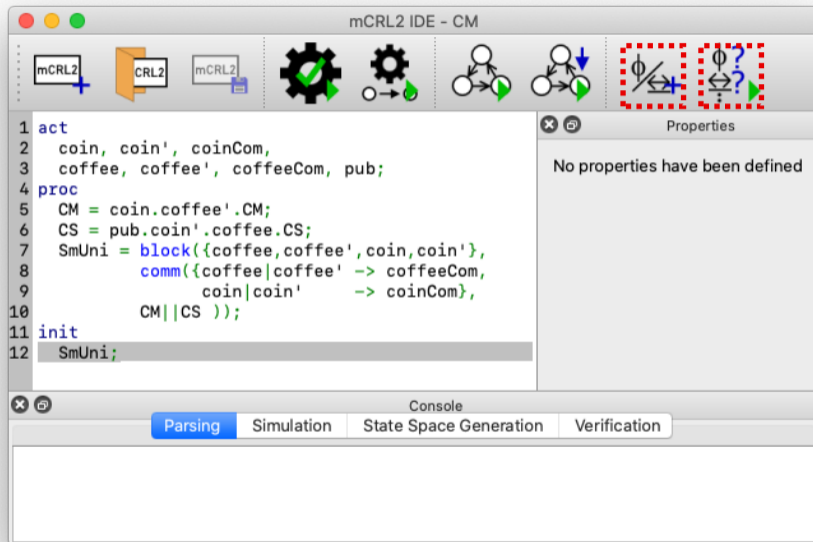
lambda n:Nat . exp *Function*

mCRL2 toolset overview



Assignment 1: <https://cister-labs.github.io/ramde2122/assignments/a1-modelling.pdf>

Logic and Verification



Add
properties

Verify
properties

Syntax (simplified)

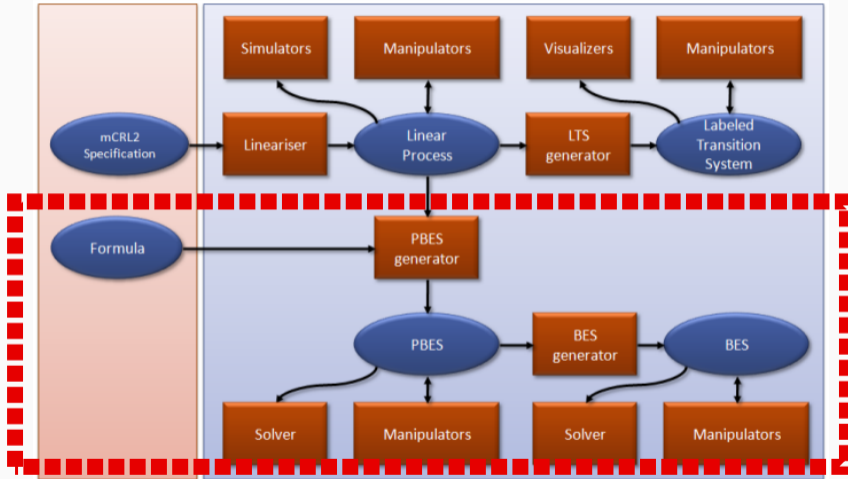
$$\phi = \text{true} \mid \text{false} \mid \text{forall } x:T.\phi \mid \text{exists } x.:T\phi \\ \mid \phi \text{ OP } \phi \mid !\phi \mid [\text{mod}]\phi \mid \langle \text{mod} \rangle \phi \mid \dots$$
$$\text{mod} = \alpha \mid \text{nil} \mid \text{mod}+\text{mod} \mid \text{mod}.\text{mod} \mid \text{mod}^* \mid \text{mod}+$$
$$\alpha = a(d) \mid a|b|c \mid \text{true} \mid \text{false} \mid \alpha \text{ OP } \alpha \mid !\alpha \\ \mid \text{forall } x:T.\alpha \mid \text{exists } x:T.\alpha \mid \dots$$

where $T = \{Bool, Nat, Int, \dots\}$ and $OP = \{=, \&\&, \parallel\}$

Example

“ $[\text{true}^*.a]\langle b \rangle \text{true}$ ” means: *whenever an ‘a’ appears after any number of steps, it must be immediately followed by ‘b’.*

mCRL2 toolset overview



Assignment 2: <https://cister-labs.github.io/ramde2122/assignments/a2-verification.pdf>