

Specification of a type system for mCRL2

Collaborative work with Michel Reniers

Jeroen Keiren

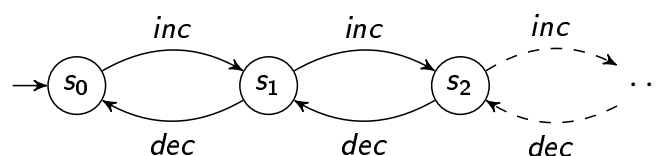
January 20, 2011

mCRL2

3/23

- ▶ Specification language
- ▶ Process algebra
- ▶ Data

proc $Counter(n:\mathbb{N}) =$
 $+ (n > 0) \rightarrow inc \cdot Counter(n + 1)$
 $+ (n > 0) \rightarrow dec \cdot Counter(n - 1)$
init $Counter(0)$



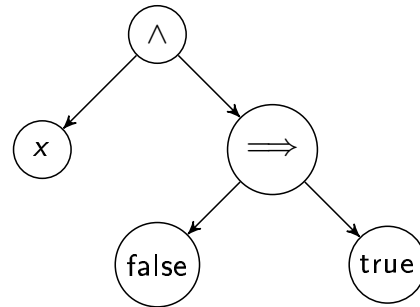
- ▶ Standard data types ($\mathbb{B}, \mathbb{N}^+, \mathbb{N}, \mathbb{Z}, \mathbb{R}$)
- ▶ Basic types ($S, T, U, Colour$) (including standard data types)
- ▶ Function types ($S \times T \times U \rightarrow V$)
- ▶ Container types ($List(S), Bag(T), Set(U)$)

- ▶ Variables (x, y, z)
- ▶ Functions ($f, g, h, 0, succ, \triangleright, +$)
- ▶ Binding ($\lambda x : S.e, \forall x : S.e, \exists x : S.e$)
- ▶ Application ($e(e_1, \dots, e_n), succ(0), 3 \triangleright [5, 481], f(x)$)

Example (Data specification)

```

sort   Tree;
cons   leaf:  $\mathbb{B} \rightarrow Tree$ ;
          node: ( $\mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$ )
               $\times Tree \times Tree \rightarrow Tree$ ;
map   isNode:  $Tree \rightarrow \mathbb{B}$ ;
          left, right :  $Tree \rightarrow Tree$ ;
var   t1, t2:  $Tree$ ;
          b:  $\mathbb{B}$ ;
          f:  $\mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$ ;
eqn   isNode(leaf(b)) = false;
          isNode(node(f, t1, t2)) = true;
          left(node(f, t1, t2)) = t1;
          right(node(f, t1, t2)) = t2;
    
```



```

node( $\wedge$ ,
     leaf(x),
     node( $\implies$ ,
          leaf(false),
          leaf(true)))
    
```

Specifying data, formally

Fix signature $\Sigma = (S_{Basic}, \Omega)$

- ▶ S_{Basic} set of basic sorts
- ▶ Ω set of function declarations

Note: specifying equations not relevant for typing expressions

Assume:

- ▶ Signature Σ
- ▶ Context Γ (stack of variable declarations)

$$\frac{x : s \in \Gamma}{\Gamma \vdash_{\Sigma} x : s} \text{ (Var)} \quad \frac{f : s_1 \times \dots \times s_n \rightarrow s \in \Sigma}{\Gamma \vdash_{\Sigma} f : s_1 \times \dots \times s_n \rightarrow s} \text{ (Func)}$$

$$\frac{\Gamma, x_1 : s_1 \vdash_{\Sigma} e : s}{\Gamma \vdash_{\Sigma} (\lambda x_1 : s_1. e) : s_1 \rightarrow s} \text{ (Abs)}$$

$$\frac{\Gamma \vdash_{\Sigma} e : s_1 \times \dots \times s_n \rightarrow s \quad \Gamma \vdash_{\Sigma} e_1 : s_1 \quad \dots \quad \Gamma \vdash_{\Sigma} e_n : s_n}{\Gamma \vdash_{\Sigma} e(e_1, \dots, e_n) : s} \text{ (Appl)}$$

$$\frac{\Gamma, x_1 : s_1 \vdash_{\Sigma} e : \mathbb{B}}{\Gamma \vdash_{\Sigma} (\forall x_1 : s_1. e) : \mathbb{B}} \text{ (Forall)} \quad \frac{\Gamma, x_1 : s_1 \vdash_{\Sigma} e : \mathbb{B}}{\Gamma \vdash_{\Sigma} (\exists x_1 : s_1. e) : \mathbb{B}} \text{ (Exists)}$$

$$\frac{\Gamma, x : s \vdash_{\Sigma} e : \mathbb{B}}{\Gamma \vdash_{\Sigma} \{x : s \mid e\} : \text{Set}(s)} \text{ (Set)} \quad \frac{\Gamma, x : s \vdash_{\Sigma} e : \mathbb{N}}{\Gamma \vdash_{\Sigma} \{x : s \mid e\} : \text{Bag}(s)} \text{ (Bag)}$$

Let:

- ▶ $n:\mathbb{N}$
- ▶ $f:\mathbb{Z} \rightarrow S$

$\Gamma \vdash_{\Sigma} f(n) : ???$

Two solutions:

1. Require **casts**: $N2I:\mathbb{N} \rightarrow \mathbb{Z}$. User writes $f(N2I(n))$
2. Allow for **subtyping**: $\mathbb{N} \subseteq \mathbb{Z}$

Add following rule to type inference system:

$$\frac{\Gamma \vdash_{\Sigma} e : s' \quad s' \subseteq s}{\Gamma \vdash_{\Sigma} e : s} \text{ (Subtyping)}$$

Axioms for \subseteq :

$$\frac{}{\mathbb{N}^+ \subseteq \mathbb{N}} \text{ (P2N)} \quad \frac{}{\mathbb{N} \subseteq \mathbb{Z}} \text{ (N2I)} \quad \frac{}{\mathbb{Z} \subseteq \mathbb{R}} \text{ (I2R)}$$

$$\frac{s'_1 \subseteq s_1 \quad \dots \quad s'_n \subseteq s_n \quad s \subseteq s'}{s_1 \times \dots \times s_n \rightarrow s \subseteq s'_1 \times \dots \times s'_n \rightarrow s'} \text{ (SubFunc)}$$

Let

- ▶ $map: (\mathbb{N} \rightarrow S) \times List(\mathbb{N}) \rightarrow List(S)$
- ▶ $f: \mathbb{R} \rightarrow S$
- ▶ $x: List(\mathbb{N})$

Can $map(f, x)$ be typed?

- ▶ $(\mathbb{N} \rightarrow S) \times List(\mathbb{N}) \rightarrow List(S) \subseteq (\mathbb{N} \rightarrow S) \times List(\mathbb{N}) \rightarrow List(S)$, if
- ▶ $(\mathbb{R} \rightarrow S) \subseteq (\mathbb{N} \rightarrow S)$, if
- ▶ $\mathbb{N} \subseteq \mathbb{R}$

Intuition: function passed as argument to another function can be **applied to anything given to it by the context**.

Subtyping function domain (wrong alternative)

$$\frac{s_1 \subseteq s'_1 \quad \dots \quad s_n \subseteq s'_n \quad s \subseteq s'}{s_1 \times \dots \times s_n \rightarrow s \subseteq s'_1 \times \dots \times s'_n \rightarrow s'} \text{ (SubFunc)}$$

Let

- ▶ $map: (\mathbb{R} \rightarrow S) \times List(\mathbb{R}) \rightarrow List(S)$
- ▶ $f: \mathbb{N} \rightarrow S$
- ▶ $x: List(\mathbb{R})$

Can $map(f, x)$ be typed?

- ▶ $(\mathbb{N} \rightarrow S) \times List(\mathbb{R}) \rightarrow List(S) \subseteq (\mathbb{R} \rightarrow S) \times List(\mathbb{R}) \rightarrow List(S)$, if
- ▶ $(\mathbb{N} \rightarrow S) \subseteq (\mathbb{R} \rightarrow S)$, if
- ▶ $\mathbb{N} \subseteq \mathbb{R}$.

Problem: how is $f(y)$ defined for $y < 0$?

Standard definition of +:

$$+:\mathbb{N}^+ \times \mathbb{N}^+ \rightarrow \mathbb{N}^+$$

$$+:\mathbb{N} \times \mathbb{N}^+ \rightarrow \mathbb{N}^+$$

$$+:\mathbb{N}^+ \times \mathbb{N} \rightarrow \mathbb{N}^+$$

$$+:\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$+:\mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$$

$$+:\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$$

Need support for **overloading** of operators

Assigning types to subexpressions

mCRL2 uses **rewriting** to simplify expressions.

- ▶ Equations as rewrite rules
- ▶ Does expression **match** with a rule?

Requires efficient ($\mathcal{O}(1)$) matching



Attribute each **subexpression** with its **type**

- ▶ Is expression typable?
- ▶ What is “the type” of an expression?
- ▶ How is the type computed?

Type each subexpression

Goal: assign unique type to each subexpression

Idea: **order** type deductions \rightarrow prevent/postpone typecasts

First: order deduction rules { Var, Func, Abs, Appl, Forall, Exists, Set, Bag }

<

{ Subtyping }

<

{ P2N, N2I, I2R, SubList, SubSet, SubBag }

<

{ SubFunc, SubTrans }

$$\frac{\frac{\overline{P_1}(d'_1) \quad \dots \quad \overline{P_N}(d'_N)}{C_1} \quad (\text{Rule}_1)}{\text{Deduction 1}}$$

$$\frac{\frac{\overline{Q_1}(d''_1) \quad \dots \quad \overline{Q_N}(d''_N)}{C_2} \quad (\text{Rule}_2)}{\text{Deduction 2}}$$

Deduction 1 **better than** ($<$) Deduction 2,

- ▶ if $\text{Rule}_1 < \text{Rule}_2$, or
- ▶ $\text{Rule}_1 = \text{Rule}_2$, and $d'_i \leq d''_i$ for all i , and $d'_i < d''_i$ for some i

Choosing a deduction (example)

Let

- ▶ $x:\mathbb{N}^+ \in \Gamma$
- ▶ $f:\mathbb{N} \rightarrow S$

Type $f(x)$

$$\frac{\frac{f:\mathbb{N} \rightarrow S \in \Sigma}{\Gamma \vdash_{\Sigma} f:\mathbb{N} \rightarrow S} \quad (\text{Func}) \quad \frac{\frac{x:\mathbb{N}^+ \in \Gamma}{\Gamma \vdash_{\Sigma} x:\mathbb{N}^+} \quad (\text{Var}) \quad \frac{}{\mathbb{N}^+ \subseteq \mathbb{N}} \quad (\text{P2N})}{\Gamma \vdash_{\Sigma} x:\mathbb{N}} \quad (\text{Subtyping})}{\Gamma \vdash_{\Sigma} f(x):S} \quad (\text{Appl})$$

$$\frac{\frac{f:\mathbb{N} \rightarrow S \in \Sigma}{\Gamma \vdash_{\Sigma} f:\mathbb{N} \rightarrow S} \quad (\text{Func}) \quad \frac{\frac{}{\mathbb{N}^+ \subseteq \mathbb{N}} \quad (\text{P2N})}{\mathbb{N} \rightarrow S \subseteq \mathbb{N}^+ \rightarrow S} \quad (\text{SubFunc})}{\Gamma \vdash_{\Sigma} f:\mathbb{N}^+ \rightarrow S} \quad (\text{Subtyping}) \quad \frac{x:\mathbb{N}^+ \in \Gamma}{\Gamma \vdash_{\Sigma} x:\mathbb{N}^+} \quad (\text{Var})}{\Gamma \vdash_{\Sigma} f(x):S} \quad (\text{Appl})$$

- ▶ Rules for typing mCRL2
- ▶ Determine “the type” of an expression
- ▶ Assign types to subexpressions

Future work:

- ▶ Make widening explicit (automatically add casts)
- ▶ Give algorithm to type check mCRL2
- ▶ Investigate generalisation to other languages