Guarded Recursion and Mathematical Operational Semantics

Mauro Jaskelioff  Neil Ghani

School of Computer Science

Coalgebraic Methods in Computer Science 2008
When is a collection of rules a well-behaved SOS?
Syntactic Rule Formats

- A theory of SOS?
- Rule formats restrict the syntax of rules.
- Given for a concrete transition relation.

Example (GSOS)

\[
\begin{array}{c}
\{ x_i \xrightarrow{a} y_{ij} \}_{1 \leq i \leq n, a \in A_i} \quad \{ x_i \xrightarrow{b} \}_{1 \leq i \leq n} \quad \{ b \in B_i \} \\
\sigma(x_1, \ldots, x_n) \xrightarrow{c} t
\end{array}
\]

- \( A_i, B_i \subseteq A \).
- \( x_i \) and \( y_{ij}^a \) are all distinct.
- Those are the only variables that occur in the term \( t \).
SOS is a distributive law of a monad over a comonad

\[ TD \rightarrow DT \]
Constructing Distributive Laws

Semantics given by an abstract operational rule

\[ \rho : \Sigma(\text{Id} \times B) \to BT \]

We get a lifting \( \psi \) of \( T \) to the \( B \)-coalgebras.
If Beauty Is Not Enough…

- Benefits of Mathematical Operational Semantics:
  - Language-independent.
  - Bisimulation is a congruence.
  - Adequate denotational model.
  - Derive rule formats.

Recursive programs are expressed via equations:

\[
\begin{align*}
    x_1 &= t_1(x_1, \ldots, x_n) \\
    & \vdots \\
    x_n &= t_n(x_1, \ldots, x_n)
\end{align*}
\]

Processes \( s_1, \ldots, s_n \) are a solution if

\[
\begin{align*}
    s_1 & \sim t_1(s_1, \ldots, s_n) \\
    & \vdots \\
    s_n & \sim t_n(s_1, \ldots, s_n)
\end{align*}
\]
Equations should be guarded to ensure existence and uniqueness of solutions.

**Definition (Guarded Equations for ACP)**

An equation is *guarded* if its RHS can be written as:

\[ a_1 \cdot t'_1(x_1, \ldots, x_n) + \cdots + a_k \cdot t'_k(x_1, \ldots, x_n) + b_1 + \cdots + b_l \]
Guarded Equations, Abstractly

- The guardedness condition is
  \[ x_i = a_1 \cdot t'_1(x_1, \ldots, x_n) + \cdots + a_k \cdot t'_k(x_1, \ldots, x_n) + b_1 + \cdots + b_l \]
- The behaviour for ACP is \( \mathcal{P}_f(A \times - + A) \)
- More abstractly, an equation is guarded if it is a function:
  \[ X \rightarrow BTX \]
- \( B \) is expressing a reflection of behaviour in the syntax.
- Note that \( X \rightarrow \Sigma TX \) is not enough!
  \[ x = x \cdot t \]
Reflecting Behaviour in Syntax

- Given a signature $\Sigma$ with semantics

$$\rho : \Sigma(\text{Id} \times B) \xrightarrow{\rho} BT_\Sigma$$

- we have semantics for signature $B$

$$\beta : B(\text{Id} \times B) \xrightarrow{B\pi_1} B \xrightarrow{B\eta} BT_B$$

- and injections

$$\iota^\Sigma : T_\Sigma \to T_{\Sigma+B}$$
$$\iota^B : T_B \to T_{\Sigma+B}$$

- we obtain

$$\iota^\Sigma \circ \rho + \iota^B \circ \beta : (\Sigma + B)(\text{Id} \times B) \to BT_{\Sigma+B}$$
Turi’s Guarded Equations

- Given \( \rho: \Sigma(\text{id} \times B) \to BT \) and \( e: X \to BTX \)

\[
\begin{array}{cccc}
X & \xrightarrow{\eta_X} & TX & \xleftarrow{\text{inr}_X} & \Sigma TX \\
\downarrow e & & \downarrow \psi(k) & & \downarrow \Sigma \langle \text{id}, \psi(k) \rangle \\
BTX & \xleftarrow{B\mu_X} & BT^2 X & \xleftarrow{\rho_{TX}} & \Sigma(TX \times BTX)
\end{array}
\]

- This is not a lifting of \( T \) to the \( B \)-coalgebras!

- No distributive law is obtained.
Recursive Programs as Operators

- Rather than variables, we have recursive operators $\Omega$.

- Equations are natural transformations

$$\Omega \rightarrow BT_{\Sigma+\Omega}$$

- Parameter-passing recursive operators are now possible.
Construction of an Abstract Operational Rule

- Given $\rho: \Sigma(Id \times B) \rightarrow BT_\Sigma$ and $e: \Omega \rightarrow BT_{\Sigma+\Omega}$

\[
\begin{array}{c}
\Sigma(Id \times B) \xrightarrow{\text{inl}} (\Sigma + \Omega)(Id \times B) \xleftarrow{\text{inr}} \Omega(Id \times B)
\end{array}
\]

- We construct a plain abstract operational rule.

- We can generalize to $\Omega(Id \times B) \rightarrow BT_{\Sigma+\Omega}$
Summary

- Added guarded equations in mathematical operational semantics.

- Guarded equations are not really necessary: operational rules are enough for describing (guarded) recursive programs.

- Recursive programs are a reflection in syntax of operationally-defined infinitary behaviour.
Thank you!