Guarded Recursion and Mathematical Operational Semantics

Mauro Jaskelioff   Neil Ghani

School of Computer Science

The University of Nottingham

Coalgebraic Methods in Computer Science 2008

Structural Operational Semantics

$$\frac{}{a \xrightarrow{a} \surd} \qquad \frac{p \xrightarrow{a} p'}{p \cdot q \xrightarrow{a} p' \cdot q} \qquad \frac{p \xrightarrow{a} \surd}{p \cdot q \xrightarrow{a} q} \qquad \frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p + q \xrightarrow{a} q'}$$

$$\frac{p \xrightarrow{a} \surd}{p + q \xrightarrow{a} \surd} \qquad \frac{q \xrightarrow{a} \surd}{p + q \xrightarrow{a} \surd} \qquad \frac{p \xrightarrow{a} p'}{p \parallel q \xrightarrow{a} p' \parallel q} \qquad \frac{q \xrightarrow{a} q'}{p \parallel q \xrightarrow{a} p \parallel q'}$$

$$\frac{p \xrightarrow{a} \surd}{p \parallel q \xrightarrow{a} q} \qquad \frac{q \xrightarrow{a} \surd}{p \parallel q \xrightarrow{a} p} \qquad \frac{p \xrightarrow{a} p' \quad q \xrightarrow{b} q'}{p \parallel q \xrightarrow{\gamma(a,b)} p' \parallel q'} \qquad \frac{p \xrightarrow{a} \surd \quad q \xrightarrow{b} \surd}{p \parallel q \xrightarrow{\gamma(a,b)} \surd}$$

$$\frac{p \xrightarrow{a} \surd \quad q \xrightarrow{b} q'}{p \parallel q \xrightarrow{\gamma(a,b)} q'} \qquad \frac{p \xrightarrow{a} p' \quad q \xrightarrow{b} \surd}{p \parallel q \xrightarrow{\gamma(a,b)} p'} \qquad \frac{p \xrightarrow{a} \surd \quad a \notin H}{\partial_H(p) \xrightarrow{a} \surd} \qquad \frac{p \xrightarrow{a} p' \quad a \notin H}{\partial_H(p) \xrightarrow{a} \partial_H(p')}$$

- 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)

$$\frac{\{x_i \xrightarrow{a} y_{ij}^a\}_{1\leq j\leq m_i^a}^{1\leq i\leq n, a\in A_i} \quad \{x_i \stackrel{b}{\not\rightarrow}\}_{b\in B_i}^{1\leq i\leq n}}{\sigma(x_1,\ldots,x_n) \xrightarrow{c} t}$$

- $A_i, B_i \subseteq A$.
- x_i and y_{ij}^a are all distinct.
- Those are the only variables that occur in the term t.

Mathematical Operational Semantics

SOS is a distributive law of a monad over a comonad

$$TD \longrightarrow DT$$

Constructing Distributive Laws

Semantics given by an *abstract operational rule*

$$\rho\colon \Sigma(Id \times B) \to BT$$

We get a lifting Ψ of T to the B-coalgebras.

$$\begin{array}{ccccccc}
X & \xrightarrow{\eta_X} & TX & & \xleftarrow{\mathrm{inr}_X} & & \Sigma TX \\
{\scriptstyle k}\downarrow & & \downarrow{\scriptstyle \Psi(k)} & & & & \downarrow{\scriptstyle \Sigma\langle id,\Psi(k)\rangle} \\
BX & \xrightarrow[B\eta_X]{} & BTX & \xleftarrow[B\mu_X]{} & BT^2X & \xleftarrow[\rho_{TX}]{} & \Sigma(TX \times BTX)
\end{array}$$

If Beauty Is Not Enough. . .

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

D. Turi. and G. Plotkin. Towards a mathematical operational semantics. *12th LICS Conf.*, 1997.

Back to “Syntactic” SOS

Recursive programs are expressed via equations:

$$\begin{array}{rcl} x_1 & = & t_1(x_1, \ldots, x_n) \\ & \vdots & \\ x_n & = & t_n(x_1, \ldots, x_n) \end{array}$$

Processes $s_1, \ldots, s_n$ are a solution if

$$\begin{array}{rcl} s_1 & \sim & t_1(s_1, \ldots, s_n) \\ & \vdots & \\ s_n & \sim & t_n(s_1, \ldots, s_n) \end{array}$$

Guarded equations

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:

$$X \to BTX$$

- B is expressing a reflection of behaviour in the syntax.
- Note that $X \to \Sigma TX$ is not enough!

$$x = x \cdot t$$

Reflecting Behaviour in Syntax

- Given a signature Σ with semantics

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

- we have semantics for signature B

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

- and injections

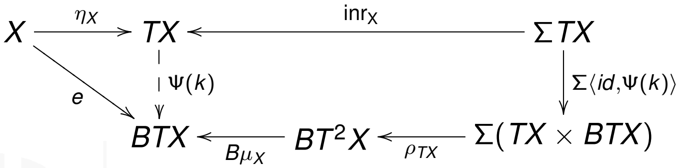
$$\begin{array}{rcl} \iota^\Sigma & : & T_\Sigma \to T_{\Sigma+B} \\ \iota^B & : & T_B \to T_{\Sigma+B} \end{array}$$

- we obtain

$$\iota^\Sigma \circ \rho + \iota^B \circ \beta \;:\; (\Sigma + B)(Id \times B) \to BT_{\Sigma+B}$$

Turi's Guarded Equations

- Given $\rho\colon \Sigma(Id \times B) \to BT$ and $e\colon X \to BTX$



- 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 Ω.
- Equations are natural transformations

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

- Parameter-passing recursive operators are now possible.

Construction of an Abstract Operational Rule

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

$$
\begin{array}{ccccc}
\Sigma(Id \times B) & \xrightarrow{\text{inl}} & (\Sigma+\Omega)(Id \times B) & \xleftarrow{\text{inr}} & \Omega(Id \times B) \\
\downarrow{\scriptstyle \rho} & & \downarrow & & \downarrow{\scriptstyle \pi_1} \\
BT_\Sigma & \hookrightarrow & BT_{\Sigma+\Omega} & \xleftarrow[e]{} & \Omega
\end{array}
$$

- We construct a plain abstract operational rule.

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

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!