Rule formats for bounded nondeterminism in structural operational semantics

Luca Aceto **Álvaro García-Pérez** Anna Ingólfsdóttir

Reykjavík University

Lyngby, January 8th, 2016

Motivation

Structural operational semantics and bounded nondeterminism

A *transition system specification* (TSS) consists of inference rules that induce a *labelled transition system* (LTS) $\{p \xrightarrow{a} p'\}$

Structural operational semantics and bounded nondeterminism

A *transition system specification* (TSS) consists of inference rules that induce a *labelled transition system* (LTS) $\{p \xrightarrow{a} p'\}$

Exercises 3.3 and 3.4 in *Semantics with Applications: An Appetizer* [Nielson and Nielson, 2007]

While language with nondeterminisitc choice and statement `random(x)`.

```
x:=-1; while x<=0 do (x:=x-1 or x:=(-1)*x)
```

An LTS is *finite branching* iff for every p, the set $\{(a, p') \mid p \xrightarrow{a} p'\}$ is finite.

Structural operational semantics and bounded nondeterminism

A *transition system specification* (TSS) consists of inference rules that induce a *labelled transition system* (LTS) $\{p \xrightarrow{a} p'\}$

Exercises 3.3 and 3.4 in *Semantics with Applications: An Appetizer* [Nielson and Nielson, 2007]

While language with nondeterminisitc choice and statement `random(x)`.

```
x:=-1; while x<=0 do (x:=x-1 or x:=(-1)*x)
```

An LTS is *finite branching* iff for every p, the set $\{(a, p') \mid p \xrightarrow{a} p'\}$ is finite.

Rule formats for finite branching: statically checkable (ideally) conditions on TSSs that guarantee continuous Scott-Strachey semantics ([Apt and Plotkin, 1986]).

Existing rule format for finite branching [Fokkink and Vu, 2003]

Theorem (Correctness of rule format)

Let R be a TSS. The LTS associated to R is finite branching if the following conditions hold:

(i) *R has no unguarded recursion (**strict stratification**).*

(ii) *Each rule in R gives rise to finitely many transitions from each process (**bounded nondeterminism format**).*

(iii) *Only finitely many rules in R can give rise to transitions from each process (**uniformity** and **finitely inhabited η-types**).*

Example (Rules for merge in BPA)

$$\cdots \qquad \frac{x_0 \xrightarrow{c} x_0'}{x_0 \| x_1 \xrightarrow{c} x_0' \| x_1} \qquad \frac{x_1 \xrightarrow{c} x_1'}{x_0 \| x_1 \xrightarrow{c} x_0 \| x_1'} \qquad \cdots$$

Example (Rules for merge in BPA)

$$\cdots \qquad \frac{x_0 \xrightarrow{c} x_0'}{x_0 \| x_1 \xrightarrow{c} x_0' \| x_1} \qquad \frac{x_1 \xrightarrow{c} x_1'}{x_0 \| x_1 \xrightarrow{c} x_0 \| x_1'} \qquad \cdots$$

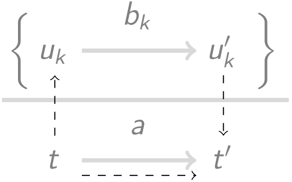
Strict stratification:

$$\begin{array}{rcl} S(c) & = & 0 \\ S(p_0 \| p_1) & = & 1 + S(p_0) + S(p_1) \\ & & \cdots \end{array}$$

Example (Rules for merge in BPA)

$$\cdots \qquad \frac{x_0 \xrightarrow{c} x_0'}{x_0 \| x_1 \xrightarrow{c} x_0' \| x_1} \qquad \frac{x_1 \xrightarrow{c} x_1'}{x_0 \| x_1 \xrightarrow{c} x_0 \| x_1'} \qquad \cdots$$

Bounded nondeterminism format:



Example (Rules for merge in BPA)

$$\cdots \qquad \frac{x_0 \xrightarrow{c} x_0'}{x_0 \| x_1 \xrightarrow{c} x_0' \| x_1} \qquad \frac{x_1 \xrightarrow{c} x_1'}{x_0 \| x_1 \xrightarrow{c} x_0 \| x_1'} \qquad \cdots$$

Uniformity and finitely inhabited η-types:

$$\eta(x_0 \| x_1) = \{x_0, x_1\}$$

Example (Rules for merge in BPA)

$$\cdots \qquad \frac{x_0 \xrightarrow{c} x_0'}{x_0 \| x_1 \xrightarrow{c} x_0' \| x_1} \qquad \frac{x_1 \xrightarrow{c} x_1'}{x_0 \| x_1 \xrightarrow{c} x_0 \| x_1'} \qquad \cdots$$

Uniformity and finitely inhabited η-types:

$$\langle x_0 \| x_1, \{x_0 \mapsto \{c\}, x_1 \mapsto \emptyset\}\rangle$$

$$\eta(x_0 \| x_1) = \{x_0, x_1\}$$

Example (Rules for merge in BPA)

$$\cdots \qquad \frac{x_0 \xrightarrow{c} x_0'}{x_0 \| x_1 \xrightarrow{c} x_0' \| x_1} \qquad \frac{x_1 \xrightarrow{c} x_1'}{x_0 \| x_1 \xrightarrow{c} x_0 \| x_1'} \qquad \cdots$$

Uniformity and finitely inhabited η-types:

$$\langle x_0 \| x_1, \{x_0 \mapsto \{c\}, x_1 \mapsto \emptyset\}\rangle \qquad \langle x_0 \| x_1, \{x_0 \mapsto \emptyset, x_1 \mapsto \{c\}\}\rangle$$

$$\eta(x_0 \| x_1) = \{x_0, x_1\}$$

The problem

- Mechanising the proof of correctness of the rule format?

  ### Claim [Fokkink and Vu, 2003]

  For every term t there are finitely many maps ψ such that there exists a rule r of η-type $\langle t, \psi \rangle$ which gives rise to transitions.

  Proof: by assuming that the set of different maps ψ is infinite and deriving a contradiction. □

  Reasoning by contradiction here is not constructive!

- Bounded-nondeterminism properties other than finite branching?

  An LTS is *image finite* iff for every p and a the set $\{p' \mid p \xrightarrow{a} p'\}$ is finite.
  An LTS is *initials finite* iff for every p the set $\{a \mid \exists p'.p \xrightarrow{a} p'\}$ is finite.

  Rule formats for initials finiteness and for finite branching?

Our contribution

Constructive proof of correcteness of the rule format

For each process $p = \sigma(t)$, the ψ maps such that there exists a rule r of η-type $\langle t, \psi \rangle$ which gives rise to transitions are dependent functions of type $\psi : \Pi_{v \in \eta(t)} \{a \mid \sigma(v) \xrightarrow{a} q\}$.

Constructivity enables the mechanisation of the proof with a state-of-the-art proof assistant (work in progress).

Rule formats for image finiteness and initials finiteness

Definition (Image finiteness and initials finiteness)

An LTS is *image finite* iff for every p and a the set $\{p' \mid p \xrightarrow{a} p'\}$ is finite.
An LTS is *initials finite* iff for every p the set $\{a \mid \exists p'.p \xrightarrow{a} p'\}$ is finite.

The properties require modified η-types that either *ignore the targets* or *keep track of both actions and targets* in transitions.

Rule formats for image finiteness and initials finiteness

Definition (Image finiteness and initials finiteness)

An LTS is *image finite* iff for every p and a the set $\{p' \mid p \xrightarrow{a} p'\}$ is finite.
An LTS is *initials finite* iff for every p the set $\{a \mid \exists p'.p \xrightarrow{a} p'\}$ is finite.

The properties require modified η-types that either *ignore the targets* or *keep track of both actions and targets* in transitions.

Example (Statement `random(x)`)

$$\frac{}{\langle \texttt{random}(x); S, s\rangle \xrightarrow{n} \langle S, s[x \mapsto n]\rangle}\ , \qquad n \in \mathbb{N}.$$

Related and Future work

- Generalise the rule formats to other bounded-nondeterminism properties [Aceto et al., 2016].
- Extend the rule formats to SOS with terms as labels [Aceto et al., 2016].
- Modify the rule formats to cover cases that we are aware are not covered yet.
- Extend the rule formats to many sorted signatures and Nominal SOS.

Summary

- Rule formats for bounded nondeterminism are useful to check whether a language admits a standard continuous semantics a la Scott-Strachey.
- We provide a constructive proof of correctness of the rule format for finite branching in [Fokkink and Vu, 2003].
- We provide rule formats for initials finiteness and image finiteness.

Summary

- Rule formats for bounded nondeterminism are useful to check whether a language admits a standard continuous semantics a la Scott-Strachey.
- We provide a constructive proof of correctness of the rule format for finite branching in [Fokkink and Vu, 2003].
- We provide rule formats for initials finiteness and image finiteness.

Happy Birthday to Hanne and Flemming!

References I

Aceto, L., Fábregas, I., García-Pérez, A., and Ingólfsdóttir, A. (2016).
A unified rule format for bounded nondeterminism in SOS with terms as labels.
Submitted.

Apt, K. R. and Plotkin, G. D. (1986).
Countable nondeterminism and random assignment.
Journal of the ACM, 33(4):724–767.

Fokkink, W. and Vu, T. D. (2003).
Structural operational semantics and bounded nondeterminism.
Acta Informatica, 39(6-7):501–516.

Nielson, H. R. and Nielson, F. (2007).
Semantics with Applications: An Appetizer.
Springer-Verlag New York.