Process Algebra

Bas Luttik
s.p.luttik@tue.nl
HG 6.85

FSA seminar (22 September 2011)

Technische Universiteit Eindhoven University of Technology

Algebra

$$
\begin{aligned}
x + y &= y + x \\
(x + y) + z &= x + (y + z) \\
x + 0 &= x \\
x \cdot y &= y \cdot x \\
x \cdot 1 &= x \\
(x + y) \cdot z &= x \cdot z + y \cdot z
\end{aligned}
$$

Algebra

$$
\begin{aligned}
x + y &= y + x \\
(x + y) + z &= x + (y + z) \\
x + 0 &= x \\
x \cdot y &= y \cdot x \\
x \cdot 1 &= x \\
(x + y) \cdot z &= x \cdot z + y \cdot z
\end{aligned}
$$

This collection of axioms is sound (and complete?) for $(\mathbb{N}, +, \cdot, 0, 1)$.

Algebra

$$\begin{array}{rcl} x + y & = & y + x \\ (x + y) + z & = & x + (y + z) \\ x + 0 & = & x \\ x \cdot y & = & y \cdot x \\ x \cdot 1 & = & x \\ (x + y) \cdot z & = & x \cdot z + y \cdot z \end{array}$$

This collection of axioms is sound (and complete?) for $(\mathbb{N}, +, \cdot, 0, 1)$.

sound: only **valid** equations can be derived;

Algebra

$$
\begin{aligned}
x + y &= y + x \\
(x + y) + z &= x + (y + z) \\
x + 0 &= x \\
x \cdot y &= y \cdot x \\
x \cdot 1 &= x \\
(x + y) \cdot z &= x \cdot z + y \cdot z
\end{aligned}
$$

This collection of axioms is sound (and complete?) for $(\mathbb{N}, +, \cdot, 0, 1)$.

sound: only **valid** equations can be derived;

complete: **all** valid equations can be derived.

Process algebra

$$\begin{aligned} x + y &= y + x \\ (x + y) + z &= x + (y + z) \\ x + 0 &= x \\ x \cdot y &= y \cdot x \\ x \cdot 1 &= x \\ (x + y) \cdot z &= x \cdot z + y \cdot z \end{aligned}$$

Process algebra

$$
\begin{aligned}
x + y &= y + x \\
(x + y) + z &= x + (y + z) \\
x + 0 &= x \\
x \cdot y &= y \cdot x \\
x \cdot 1 &= x \\
(x + y) \cdot z &= x \cdot z + y \cdot z
\end{aligned}
$$

Let $\mathbb{P}$ be a set of processes, $0 \in \mathbb{P}$ is the **deadlocked process**, $1 \in \mathbb{P}$ is the **(successfully) terminated** process, $+$ is **nondeterministic choice** and $\cdot$ is **sequential composition**.

Process algebra

$$
\begin{aligned}
x + y &= y + x \\
(x + y) + z &= x + (y + z) \\
x + 0 &= x \\
x \cdot y &= y \cdot x \\
x \cdot 1 &= x \\
(x + y) \cdot z &= x \cdot z + y \cdot z
\end{aligned}
$$

Let $\mathbb{P}$ be a set of processes, $0 \in \mathbb{P}$ is the **deadlocked process**, $1 \in \mathbb{P}$ is the **(successfully) terminated** process, $+$ is **nondeterministic choice** and $\cdot$ is **sequential composition.**

Question

Is the collection of axioms sound and complete for $(\mathbb{P}, +, \cdot, 0, 1)$?

Process algebra

$$\begin{array}{rcl} x + y & = & y + x \\ (x + y) + z & = & x + (y + z) \\ x + 0 & = & x \\ x \cdot y & = & y \cdot x \\ x \cdot 1 & = & x \\ (x + y) \cdot z & = & x \cdot z + y \cdot z \end{array}$$

Let $\mathbb{P}$ be a set of processes, $0 \in \mathbb{P}$ is the **deadlocked process**, $1 \in \mathbb{P}$ is the **(successfully) terminated** process, $+$ is **nondeterministic choice** and $\cdot$ is **sequential composition**.

Question

Is the collection of axioms sound and complete for $(\mathbb{P}, +, \cdot, 0, 1)$?

Answer

Most likely not, but first tell me a bit more about $(\mathbb{P}, +, \cdot, 0, 1)$!

Recipe for a concrete process algebra (1)

1. **syntax**: $P ::= 0 \mid 1 \mid a.P \mid P \cdot P \mid P + P$.

Recipe for a concrete process algebra (1)

1. **syntax**: $P ::= 0 \mid 1 \mid a.P \mid P \cdot P \mid P + P$.
2. **operational semantics**:

$$\frac{}{1\downarrow} \qquad \frac{}{a.P \xrightarrow{a} P} \qquad \frac{P \xrightarrow{a} P'}{P \cdot Q \xrightarrow{a} P' \cdot Q}$$

$$\frac{P\downarrow \quad Q \xrightarrow{a} Q'}{P \cdot Q \xrightarrow{a} Q'} \qquad \frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \qquad \frac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'}$$

Recipe for a concrete process algebra (1)

1. **syntax**: $P ::= 0 \mid 1 \mid a.P \mid P \cdot P \mid P + P$.
2. **operational semantics**:

$$\frac{}{1\downarrow} \qquad \frac{}{a.P \xrightarrow{a} P} \qquad \frac{P \xrightarrow{a} P'}{P \cdot Q \xrightarrow{a} P' \cdot Q}$$

$$\frac{P\downarrow \quad Q \xrightarrow{a} Q'}{P \cdot Q \xrightarrow{a} Q'} \qquad \frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \qquad \frac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'}$$

3. **behavioural equivalence**: $\leftrightarrows$ is the largest equivalence relation on the syntax such that, whenever $P \leftrightarrows Q$,
   - if $P \xrightarrow{a} P'$, then $Q \xrightarrow{a} Q'$ and $P' \leftrightarrows Q'$;
   - if $P\downarrow$, then $Q\downarrow$.

Recipe for a concrete process algebra (1)

1. **syntax**: $P ::= 0 \mid 1 \mid a.P \mid P \cdot P \mid P + P$.
2. **operational semantics**:

$$\frac{}{1\downarrow} \qquad \frac{}{a.P \xrightarrow{a} P} \qquad \frac{P \xrightarrow{a} P'}{P \cdot Q \xrightarrow{a} P' \cdot Q}$$

$$\frac{P\downarrow \quad Q \xrightarrow{a} Q'}{P \cdot Q \xrightarrow{a} Q'} \qquad \frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'} \qquad \frac{Q \xrightarrow{a} Q'}{P + Q \xrightarrow{a} Q'}$$

3. **behavioural equivalence**: $\leftrightarroweq$ is the largest equivalence relation on the syntax such that, whenever $P \leftrightarroweq Q$,
   - if $P \xrightarrow{a} P'$, then $Q \xrightarrow{a} Q'$ and $P' \leftrightarroweq Q'$;
   - if $P\downarrow$, then $Q\downarrow$.
4. Define $\mathbb{P}$ as the set of $\leftrightarroweq$-classes of expressions.

Recipe for a concrete process algebra (2)

Recall that $\mathbb{P}$ consists of equivalence classes of expressions.

Recipe for a concrete process algebra (2)

Recall that $\mathbb{P}$ consists of equivalence classes of expressions.
Convenient notation: $[P] = \{Q \mid Q \leftrightarroweq P\}$.

Recipe for a concrete process algebra (2)

Recall that $\mathbb{P}$ consists of equivalence classes of expressions.
Convenient notation: $[P] = \{Q \mid Q \leftrightarroweq P\}$.

Suggested definition of $+$, $\cdot$, 0 and 1:

$$[P] + [Q] = [P + Q] \qquad 0 = [0]$$
$$[P] \cdot [Q] = [P \cdot Q] \qquad 1 = [1]$$

Recipe for a concrete process algebra (2)

Recall that $\mathbb{P}$ consists of equivalence classes of expressions.
Convenient notation: $[P] = \{Q \mid Q \leftrightarroweq P\}$.

Suggested definition of $+$, $\cdot$, 0 and 1:

$$[P] + [Q] = [P + Q] \qquad 0 = [0]$$
$$[P] \cdot [Q] = [P \cdot Q] \qquad 1 = [1]$$

5. Check whether the behavioural equivalence is compatible with the syntax, i.e.,

   if $P_1 \leftrightarroweq Q_1$ and $P_2 \leftrightarroweq Q_2$,
   then $P_1 + P_2 \leftrightarroweq Q_1 + Q_2$ and $P_1 \cdot P_2 \leftrightarroweq Q_1 \cdot Q_2$.

   (A syntax-compatible equivalence is called a congruence.)

Recipe for a concrete process algebra (2)

Recall that $\mathbb{P}$ consists of equivalence classes of expressions.
Convenient notation: $[P] = \{Q \mid Q \leftrightarroweq P\}$.

Suggested definition of $+$, $\cdot$, 0 and 1:

$$[P] + [Q] = [P + Q] \qquad 0 = [0]$$
$$[P] \cdot [Q] = [P \cdot Q] \qquad 1 = [1]$$

5. Check whether the behavioural equivalence is compatible with the syntax, i.e.,

   if $P_1 \leftrightarroweq Q_1$ and $P_2 \leftrightarroweq Q_2$,

   then $P_1 + P_2 \leftrightarroweq Q_1 + Q_2$ and $P_1 \cdot P_2 \leftrightarroweq Q_1 \cdot Q_2$.

   (A syntax-compatible equivalence is called a congruence.)
6. Define $+$, $\cdot$, 0 and 1 as suggested above.

Assignment

Prove that *rooted divergence-preserving branching bisimilarity* is a congruence for a syntax with $+$, $\cdot$, $\ldots$.

Possible follow-up graduation projects:

- find a collection of axioms and prove that it is sound and complete;
- find SOS format ensuring that rooted divergence-preserving branching bisimilarity is comptabile with all syntactic constructions (i.e., is a congruence).

Back to the axioms

$$\begin{array}{rcl} x + y & = & y + x \\ (x + y) + z & = & x + (y + z) \\ x + x & = & x \\ (x + y) \cdot z & = & x \cdot z + y \cdot z \\ x + 0 & = & x \\ x \cdot 1 & = & x \\ 1 \cdot x & = & x \end{array}$$

Theorem

This collection of axioms is sound and complete for $(\mathbb{P}, +, \cdot, 0, 1)$.

There are many variations possible with respect to:

- the behavioural equivalence used to define $\mathbb{P}$;
- the operations defined on $\mathbb{P}$.

Assignment: extend one of the results in [CFLN08] with sequential composition.