Transition System Specifications

Aceto, Fokkink, Verhof: Structural Operational Semantics

"The first systematic study of TSSs may be found in [208], while the first study of TSSs with negative premises appeared in [57]." (Aceto et al.)

[208] R. d. Simone, Calculabilité et Expressivité dans l'Algèbre de Processus Parallèles Meije, thèse de 3 e cycle, Univ. Paris 7, 1984.

[57] B. Bloom, S. Istrail, and A. Meyer, Bisimulation can't be traced: preliminary report in Conference Record 15th ACM Symposium on Principles of Programming Languages, San Diego, California, 1988, pp. 229–239.
Preliminary version of Bisimulation can't be traced, J. Assoc. Comput. Mach., 42 (1995), pp. 232–268.

Fix a term algebra $\mathrm{Term}_{\Sigma,\mathcal{V}}$ and a set of labels A

A transition system specification on A is a set of inference rules

TSS for short

$$\frac{H}{\alpha}$$

finite set of literals

positive literal

LITERALS

positive $t \xrightarrow{a} t'$ or $t \in X$

negative $t \not\xrightarrow{a}$ or $t \notin X$

where $a \in A$, $t \in \mathrm{Term}_{\Sigma,\mathcal{V}}$, $X \subseteq \mathrm{Term}_{\Sigma,\mathcal{V}}$

Operational semantics of regular expressions

A TSS

(Act) $\dfrac{a \in A}{a \xrightarrow{a} 1}$ (Tic) $\dfrac{}{1 \xrightarrow{\varepsilon} 1}$

(Seq$_1$) $\dfrac{x \xrightarrow{a} x' \quad x' \neq 1}{x \cdot y \xrightarrow{a} x' \cdot y}$ (Seq$_2$) $\dfrac{x \xrightarrow{a} 1}{x \cdot y \xrightarrow{a} y}$

(Cho$_1$) $\dfrac{x \xrightarrow{a} x' \quad x' \neq 1}{x + y \xrightarrow{a} x'}$ (Cho$_2$) $\dfrac{x \xrightarrow{a} 1}{x + y \xrightarrow{a} 1}$

(Cho$_3$) $\dfrac{y \xrightarrow{a} y' \quad y' \neq 1}{x + y \xrightarrow{a} y'}$ (Cho$_4$) $\dfrac{y \xrightarrow{a} 1}{x + y \xrightarrow{a} 1}$

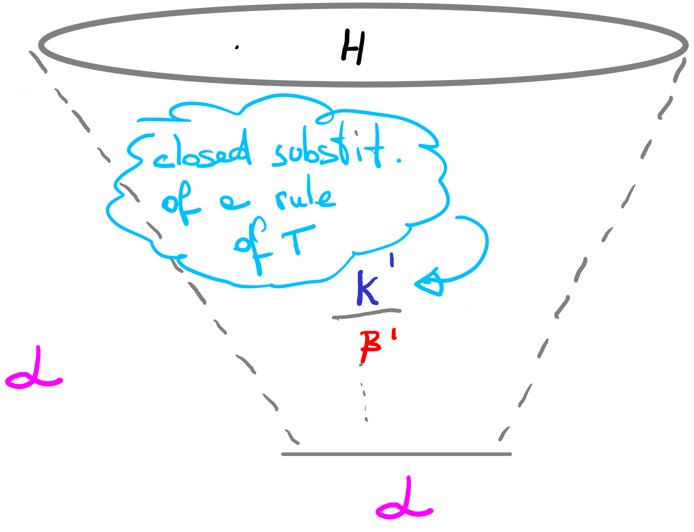
Basic Process Algebras with $a \in A \cup \{\varepsilon\}$

(Star$_1$) $\dfrac{}{x^* \xrightarrow{\varepsilon} 1}$ (Star$_2$) $\dfrac{x \xrightarrow{a} x'}{x^* \xrightarrow{a} x' \cdot x^*}$

Note that
- x & y range over the set of reg exp
- these rules form a TSS
- each operator has a set of rules (including 0, which has $\emptyset$!)

Exercise 9
Simplify the TSS above (Hint: Think about the rules for choice)

LTSs as proofs of TSSs



A proof in a TSS T of a closed transition rule $\frac{H}{\alpha}$ is an upwardly finitely-branching tree whose

- nodes are (labelled by) literals with the root (labelled by) α
- the leaves of the tree are the literals in H
- if K is the set of childrens of β then there is $\frac{H'}{\alpha'} \in T$ and $\sigma : V \to Term_{\Sigma,\emptyset}$ s.t. $K = H'\sigma$ and $\beta = \alpha'\sigma$

$\frac{H}{\alpha}$ provable in T, written $T \vdash \frac{H}{\alpha}$ if there is a proof of $\frac{H}{\alpha}$ in T

Exercise 10
Formally define closed-term substitutions and their application to terms of a term algebra.

An example

Let's fix the alphabet $V = \{e, c, t, d\}$

$$
\dfrac{\dfrac{\dfrac{\dfrac{e \in V}{e \xrightarrow{e} 1}\,(\text{act})}{e.c \xrightarrow{e} c \qquad c \neq 1}\,(\text{seq}_2)}{e.c + e.t \xrightarrow{e} c \qquad c \neq 1}\,(\text{cho}_1)}{(e.c + e.t).d \xrightarrow{e} c.d}\,(\text{seq}_1)
$$

$$
\begin{tikzcd}
(e.c+e.t).d \arrow[r, "e"] \arrow[dr, "e"'] & c.d \arrow[r, "c"] & d \arrow[r, "d"] & 1 \\
& t.d \arrow[ur, "t"'] & &
\end{tikzcd}
$$

Exercise 10
Give the LTS of a*(b+c)

RegExp & their operational semantics

We saw that we can define the language of an FSA $M = (Q, \Sigma, q_0, \delta, F)$ as

$$\mathcal{L}_M = \{ a_1 \ldots a_n \in \Sigma^* \mid \exists q_1, \ldots, q_n \mid q_0 \xrightarrow{a_1} \ldots q_{n-1} \xrightarrow{a_n} q_n \xrightarrow{\checkmark} \bullet \}$$

where $\rightarrow$ is the relation of the LTS corresponding to M

This can be generalised to ANY LTS e.g.

Since the TSS of reg exp induces an LTS, we can use the very same definition to define the language $\mathcal{L}_E$ of a reg exp E; so

$$\mathcal{L}_E = \{ a_1 \ldots a_n \in \Sigma^* \mid \exists E_1, \ldots, E_n : E \xrightarrow{a_1} E_1 \ldots E_{n-1} \xrightarrow{a_n} E_n = 1 \}$$

where now $\xrightarrow{a_i}$ are transition to be proved by applying the rules of our TSS!

Example .

Show that $aab \in \mathcal{L}_E$ where $E = a^*(b + c)$ and $A = \{a, b, c, d\}$

1. find E_1, E_2 s.t. $E_1 \xrightarrow{a} E_2$ & there is E_3 s.t. $E_2 \xrightarrow{a} E_3 \xrightarrow{b} 1$
   - a candidate for E_3 is b since (Act) $\dfrac{b \in A}{b \xrightarrow{b} 1}$
   - likewise a candidate for E_2 is ab WHY?

$$\text{(seq}_2\text{)}\ \dfrac{\text{(Act)}\ \dfrac{a \in A}{a \xrightarrow{a} 1}}{ab \xrightarrow{a} b}$$

$$\text{seq}_2\ \dfrac{\text{act}\ \dfrac{a \in A}{a \xrightarrow{a} 1}}{aab \xrightarrow{a} ab} \qquad \text{seq}_2\ \dfrac{\dfrac{a \in A}{a \xrightarrow{a} 1}\ \text{act}}{\xrightarrow{a} b} \qquad \dfrac{b \in A}{\xrightarrow{b} 1}\ \text{act}$$

A formal model of concurrency [Bergstra et al.]

From regular expressions to process algebras: a model of concurrency

$A_\tau = A \cup \{\tau\} \qquad \tau \notin A$

$E ::= \ \cdots \mid E \parallel E$

without Kleene star and 0

Note the different yet equivalent definition wrt [Bergstra et al.]

$$(\text{Act})\ \frac{a \in A_\tau}{a \xrightarrow{a} 1}$$

$$(\text{Cho}_1)\ \frac{x \xrightarrow{a} x' \quad x' \neq 1}{x + y \xrightarrow{a} x'} \qquad (\text{Cho}_2)\ \frac{x \xrightarrow{a} 1}{x + y \xrightarrow{a} 1}$$

$$(\text{Cho}_3)\ \frac{y \xrightarrow{a} y' \quad y' \neq 1}{x + y \xrightarrow{a} y'} \qquad (\text{Cho}_4)\ \frac{y \xrightarrow{a} 1}{x + y \xrightarrow{a} 1}$$

$$(\text{Seq}_1)\ \frac{x \xrightarrow{a} x' \quad x' \neq 1}{x \cdot y \xrightarrow{a} x' \cdot y} \qquad (\text{Seq}_1)\ \frac{x \xrightarrow{a} 1}{x \cdot y \xrightarrow{a} y}$$

$$(\text{per}_1)\ \frac{x \xrightarrow{a} x' \quad x' \neq 1}{x \parallel y \xrightarrow{a} x' \parallel y}$$

$$(\text{per}_2)\ \frac{y \xrightarrow{a} y' \quad y' \neq 1}{x \parallel y \xrightarrow{a} x \parallel y'}$$

$$(\text{per}_3)\ \frac{x \xrightarrow{a} 1}{x \parallel y \xrightarrow{a} y}$$

$$(\text{per}_4)\ \frac{y \xrightarrow{a} 1}{x \parallel y \xrightarrow{a} x}$$

Interleaving semantics