Full Abstraction in Structural Operational Semantics

(extended abstract)

Rob van Glabbeek*
Computer Science Department, Stanford University
Stanford, CA 94305, USA
`rvg@cs.stanford.edu`

Abstract

This paper explores the connection between semantic equivalences for concrete sequential processes, represented by means of transition systems, and formats of transition system specifications using Plotkin's structural approach. For several equivalences in the linear time – branching time spectrum a format is given, as general as possible, such that this equivalence is a congruence for all operators specifiable in that format. And for several formats it is determined what is the coarsest congruence with respect to all operators in this format that is finer than partial or completed trace equivalence.

1 Preorders and equivalences on labelled transition systems

Definition 1 A *labelled transition system (LTS)* is a pair $(\mathbb{P}, \longrightarrow)$ with $\mathbb{P}$ a set (of *processes*) and $\longrightarrow \subseteq \mathbb{P} \times A \times \mathbb{P}$ for A a set (of *actions*).

Notation: Write $p \xrightarrow{a} q$ for $(p,a,q) \in \longrightarrow$ and $p \xrightarrow{a}$ for $\exists q \in \mathbb{P} : p \xrightarrow{a} q$.

The elements of $\mathbb{P}$ represent the processes we are interested in, and $p \xrightarrow{a} q$ means that process p can evolve into process q while performing the action a. By an action any activity is understood that is considered as a conceptual entity on a chosen level of abstraction. Different activities that are indistinguishable on the chosen level of abstraction are interpreted as occurrences of the same action $a \in A$. Actions may be instantaneous or durational and are not required to terminate, but in a finite time only finitely many actions can be carried out (i.e. only *discrete* systems are considered).

Below several semantic preorders and equivalences will be defined on processes represented by means of labelled transition systems. These preorders can be defined in terms of the *observations* that an experimentator could make during a session with a process.

*This work was supported by ONR under grant number N00014-92-J-1974.

Definition 2 The set $\mathbb{O}_A$ of *potential observations* over an action set A is defined inductively by:

$T \in \mathbb{O}_A$. The trivial observation, obtained by terminating the session.

$a\varphi \in \mathbb{O}_A$ if $\varphi \in \mathbb{O}_A$ and $a \in A$. The observation of an action a, followed by the observation φ.

$\widetilde{X} \in \mathbb{O}_A$ for $X \subseteq A$. The investigated system cannot perform further actions from the set X.

$X \in \mathbb{O}_A$ for $X \subseteq A$. The investigated system can now perform any action from the set X.

$\bigwedge_{i\in I} \varphi_i \in \mathbb{O}_A$ if $\varphi_i \in \mathbb{O}_A$ for all $i \in I$. The systems admits each of the observations φ_i.

$\neg\varphi \in \mathbb{O}_A$ if $\varphi \in \mathbb{O}_A$. (It can be observed that) φ cannot be observed.

Definition 3 Let $(\mathbb{P}, \rightarrow)$ be a LTS, labelled over A. The function $\mathcal{O}_A : \mathbb{P} \rightarrow \mathcal{P}(\mathbb{O}_A)$ of *observations* of a process is inductively defined by the clauses below.

$$\begin{array}{ll}
T & \in \mathcal{O}_A(p) \\
a\varphi & \in \mathcal{O}_A(p) \text{ if } p \xrightarrow{a} q \wedge \varphi \in \mathcal{O}_A(q) \\
\widetilde{X} & \in \mathcal{O}_A(p) \text{ if } p \stackrel{a}{\not\longrightarrow} \text{ for } a \in X \\
X & \in \mathcal{O}_A(p) \text{ if } p \xrightarrow{a} \text{ for } a \in X \\
\bigwedge_{i\in I} \varphi_i & \in \mathcal{O}_A(p) \text{ if } \varphi_i \in \mathcal{O}_A(p) \text{ for all } i \in I \\
\neg\varphi & \in \mathcal{O}_A(p) \text{ if } \varphi \notin \mathcal{O}_A(p)
\end{array}$$

As the structure of the set A of actions will play no rôle of significance in this paper, the corresponding index will from here on be omitted. Below several sublanguages of observations are defined.

$\mathbb{O}_T$ $\varphi ::= T \mid a\psi$ the *(partial) trace* observations

$\mathbb{O}_{CT}$ $\varphi ::= T \mid a\psi \mid \widetilde{A}$ the *completed trace* observations

$\mathbb{O}_F$ $\varphi ::= T \mid a\psi \mid \widetilde{X}$ the *failure* observations

$\mathbb{O}_R$ $\varphi ::= T \mid a\psi \mid \widetilde{X} \wedge Y$ the *readiness* observations

$\mathbb{O}_{FT}$ $\varphi ::= T \mid a\psi \mid \widetilde{X} \wedge \psi$ the *failure trace* observations

$\mathbb{O}_{RT}$ $\varphi ::= T \mid a\psi \mid \widetilde{X} \wedge \psi \mid X \wedge \psi$ the *ready trace* observations

$\mathbb{O}_S$ $\varphi ::= T \mid a\psi \mid \bigwedge_{i\in I} \varphi_i$ the *simulation* observations

$\mathbb{O}_{FS}$ $\varphi ::= T \mid a\psi \mid \widetilde{X} \mid \bigwedge_{i\in I} \varphi_i$ the *failure simulation* observations

$\mathbb{O}_{RS}$ $\varphi ::= T \mid a\psi \mid \widetilde{X} \mid X \mid \bigwedge_{i\in I} \varphi_i$ the *ready simulation* observations

$\mathbb{O}_B$ $\varphi ::= T \mid a\psi \mid \bigwedge_{i\in I} \varphi_i \mid \neg\psi$ the *bisimulation* observations

$\mathbb{O}_{nS}$ $\varphi ::= T \mid a\psi \mid \bigwedge_{i\in I} \varphi_i \mid \neg\psi$ ($\psi \in \mathbb{O}_{mS}$ for some $m < n$) the *n-nested simulation* observations

For each of these notions N, $\mathcal{O}_N(p)$ is defined to be $\mathcal{O}(p) \cap \mathcal{P}(\mathbb{O}_N)$.

Definition 4 Two processes $p, q \in \mathbb{P}$ are *N-equivalent*, denoted $p =_N q$, if $\mathcal{O}_N(p) = \mathcal{O}_N(q)$.
p is *N-prequivalent* to q, denoted $p \sqsubseteq_N q$, if $\mathcal{O}_N(p) \subseteq \mathcal{O}_N(q)$.

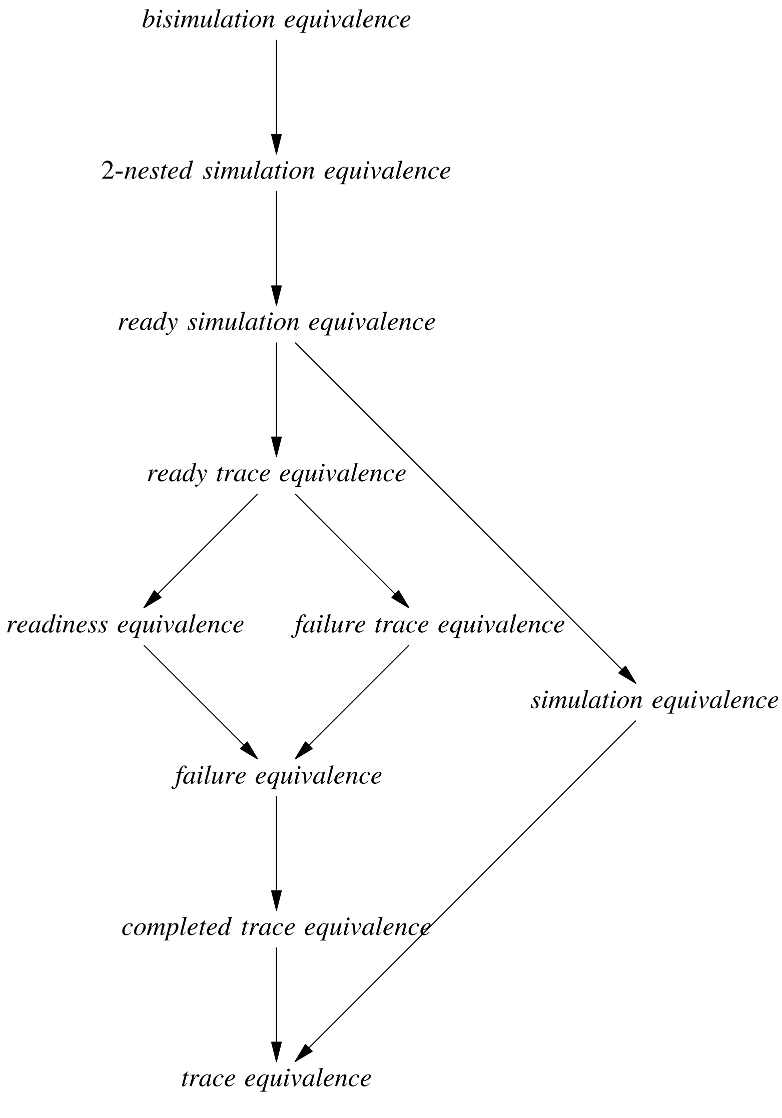

Figure 1: The linear time – branching time spectrum

On the left these equivalences are ordered w.r.t. inclusion. In Van Glabbeek [5] the observations above and the corresponding equivalences are motivated by means of testing scenarios, phrased in terms of 'button pushing experiments' on generative and reactive machines. There it is also observed that restricted to the domain of finitely branching, concrete, sequential processes, most semantic equivalences found in the literature 'that can be defined uniformly in terms of action relations' coincide with one of the equivalences defined above. The same can be said for preorders. Here *concrete* refers to the absence of internal actions (τ-moves) or internal choice. In order to facilitate the connections with other work it is worth remarking that on the mentioned domain readiness equivalence coincides with *acceptance-refusal* equivalence, failure equivalence coincides with Hennessy and De Nicola's *(must) testing equivalence*, failure trace equivalence coincides with Phillips *refusal (testing)*, and ready trace equivalence coincides with *barbed* equivalence and with *exhibited behaviour* equivalence. In order to clarify a few more relations, the following *relational characterizations* of certain equivalences may be helpful.

Definition 5 Let $(\mathbb{P}, \longrightarrow)$ be an LTS. A *ready simulation* is a relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying

- $pRq \wedge p \xrightarrow{a} p' \quad \Rightarrow \quad \exists q' : q \xrightarrow{a} q' \wedge p'Rq'$
- $pRq \wedge p \stackrel{a}{\not\longrightarrow} \quad \Rightarrow \quad q \stackrel{a}{\not\longrightarrow}$

Theorem 1 $p \sqsubseteq_{RS} q$ iff $p \sqsubseteq_{FS} q$ iff there is a ready simulation R with pRq.

Proof: "$p \sqsubseteq_{RS} q \Rightarrow p \sqsubseteq_{FS} q$" is trivial. For "$p \sqsubseteq_{FS} q \Rightarrow$ there is a ready simulation R with pRq" it suffices to establish that $\sqsubseteq_{FS}$ is a ready simulation.

- Suppose $\mathcal{O}_{FS}(p) \subseteq \mathcal{O}_{FS}(q)$ and $p \xrightarrow{a} p'$. I have to show that $\exists q' \in \mathbb{P}$ with $q \xrightarrow{a} q'$ and $\mathcal{O}_{FS}(p') \subseteq \mathcal{O}_{FS}(q')$. Let Q be $\{q' \in \mathbb{P} \mid q \xrightarrow{a} q' \wedge \exists \varphi_{q'} \in \mathcal{O}_{FS}(p') - \mathcal{O}_{FS}(q')\}$. Then $a \bigwedge_{q' \in Q} \varphi_{q'} \in \mathcal{O}_{FS}(p) \subseteq \mathcal{O}_{FS}(q)$, so there must be a $q' \in \mathbb{P}$ with $q \xrightarrow{a} q'$ and $q' \notin Q$.
- Let $\mathcal{O}(p) \subseteq \mathcal{O}(q)$ and $p \stackrel{a}{\not\longrightarrow}$. Then $\widetilde{\{a\}} \in \mathcal{O}_{FS}(p) \subseteq \mathcal{O}_{FS}(q)$ and hence $q \stackrel{a}{\not\longrightarrow}$.

Finally I have to prove that for R a ready simulation one has $pRq \Rightarrow (\varphi \in \mathcal{O}_{RS}(p) \Rightarrow \varphi \in \mathcal{O}_{RS}(q))$. I will do so with induction on φ.

- Suppose pRq and $a\varphi \in \mathcal{O}_{RS}(p)$. Then there is a $p' \in \mathbb{P}$ with $p \xrightarrow{a} p'$ and $\varphi \in \mathcal{O}_{RS}(p')$. As R is a ready simulation, there must be a $q' \in \mathbb{P}$ with $q \xrightarrow{a} q'$ and $p'Rq'$. So by induction $\varphi \in \mathcal{O}_{RS}(q')$, and hence $a\varphi \in \mathcal{O}_{RS}(q)$.

The cases that φ is T, $\widetilde{X}$, X or $\bigwedge_{i \in I} \varphi_i$ are straightforward. □

Definition 6 Let $(\mathbb{P}, \longrightarrow)$ be an LTS. A *simulation* is a relation $R \subseteq \mathbb{P} \times \mathbb{P}$ satisfying

- $pRq \wedge p \xrightarrow{a} p' \quad \Rightarrow \quad \exists q' : q \xrightarrow{a} q' \wedge p'Rq'$

A *bisimulation* is a symmetric simulation.

Theorem 2 $p \sqsubseteq_S q$ iff there is a simulation R with pRq.
$p \sqsubseteq_B q$ iff $p =_B q$ iff there is a bisimulation R with pRq.

2 Structural Operational Semantics

In this paper $\mathcal{V}$ and $\mathcal{N}$ are two disjoint countably infinite sets of *variables* and *names*. Many concepts that will appear are parameterized by the choice of $\mathcal{V}$ and $\mathcal{N}$, but as in this paper this choice is fixed, a corresponding index is suppressed.

Definition 7 *(Signatures).* A *function declaration* is a pair (f, n) of a *function symbol* $f \in \mathcal{N}$ and an *arity* $n \in \mathbb{N}$. A function declaration $(c, 0)$ is also called a *constant declaration.* A *signature* is a set of function declarations. The set $\mathbb{T}(\Sigma)$ of *terms* over a signature Σ is defined inductively by:

- $\mathcal{V} \subseteq \mathbb{T}(\Sigma)$,
- if $(f, n) \in \Sigma$ and $t_1, \ldots, t_n \in \mathbb{T}(\Sigma)$ then $f(t_1, \ldots, t_n) \in \mathbb{T}(\Sigma)$.

A term $c()$ is often abbreviated as c. For $t \in \mathbb{T}(\Sigma)$, $\mathcal{V}(t)$ denotes the set of variables that occur in t. $T(\Sigma)$ is the set of *closed* terms over Σ, i.e. the terms $t \in \mathbb{T}(\Sigma)$ with $\mathcal{V}(t) = \emptyset$. A Σ-*substitution* σ is a partial function from $\mathcal{V}$ to $\mathbb{T}(\Sigma)$. If σ is a substitution and S any syntactic object, then $S[\sigma]$ denotes the object obtained from S by replacing, for x in the domain of σ, every occurrence of x in S by $\sigma(x)$. In that case $S[\sigma]$ is called a *substitution instance* of S.

Definition 8 *(Transition system specifications).* Let Σ be a signature. A *positive* Σ-*literal* is an expression $t \xrightarrow{a} t'$ and a *negative* Σ-*literal* an expression $t \stackrel{a}{\nrightarrow}$ with $t, t' \in \mathbb{T}(\Sigma)$ and $a \in \mathcal{N}$. For $t, t' \in \mathbb{T}(\Sigma)$ the literals $t \xrightarrow{a} t'$ and $t \stackrel{a}{\nrightarrow}$ are said to *deny* each other. A *transition formula* over Σ is an expression of the form $\frac{H}{\alpha}$ with H a set of Σ-literals (the *antecedents* of the the rule) and α a Σ-literal (the *conclusion*). A formula $\frac{H}{\alpha}$ with $H = \emptyset$ is also written α. A literal or transition formula is *closed* if it contains no variables. An *action rule* is a transition formula with a positive conclusion. A *transition system specification (TSS)* is a pair (Σ, R) with Σ a signature and R a set of action rules over Σ. A TSS is *positive* if all literals in the antecedents of its rules are positive.

The concept of a TSS was introduced in GROOTE & VAANDRAGER [7]; the negative premisses were added in GROOTE [6]. The notion constitutes the first formalization of PLOTKIN's *Structural Operational Semantics (SOS)* [8] that is sufficiently general to cover most, if not all, of its applications.

Definition 9 *(Proof).* Let $P = (\Sigma, R)$ be a TSS. A *proof* of a transition formula $\frac{H}{\alpha}$ from P is a well-founded, upwardly branching tree of which the nodes are labelled by Σ-literals, such that:

- the root is labelled by α, and
- if β is the label of a node q and K is the set of labels of the nodes directly above q, then
  - either $K = \emptyset$ and $\beta \in H$,
  - or $\frac{K}{\beta}$ is a substitution instance of a rule from R,

If a proof of $\frac{H}{\alpha}$ from P exists, then $\frac{H}{\alpha}$ is *provable* from P, notation $P \vdash \frac{H}{\alpha}$.

Definition 10 *(Transition relation).* Let Σ be a signature. A *transition relation* over Σ is a relation $\longrightarrow \subseteq T(\Sigma) \times \mathcal{N} \times T(\Sigma)$. Elements (t, a, t') of a transition relation are written as $t \xrightarrow{a} t'$. Thus a transition relation over Σ can be regarded as a set of closed positive Σ-literals (*transitions*). A closed literal α *holds* in a transition relation T, notation $T \models \alpha$, if $\alpha \in T$ or $\alpha = (t \stackrel{a}{\not\longrightarrow})$ and $(t \xrightarrow{a} t') \in T$ for no $t' \in T(\Sigma)$. Write $T \models H$, for H a set of closed literals, if $T \models \alpha$ for all $\alpha \in H$.

A positive TSS specifies a transition relation in a straightforward way as the set of all derivable transitions. But as pointed out in Groote [6], it is much less trivial to associate a transition relation to a TSS with negative premisses. Several solutions are proposed in [6] and [3]. The most general of those is through the notion of *stability*. It is not difficult to show that the concept of stability defined below is the same as that of Bol and Groote.

Definition 11 *(Stable transition relation).* Let $P = (\Sigma, R)$ be a TSS and let $\longrightarrow$ be a transition relation over Σ. $\longrightarrow$ is *stable* for P if:

$$\alpha \in \longrightarrow \;\Leftrightarrow\; \begin{array}{l}\text{there is a closed transition formula } \frac{H}{\alpha} \text{ without} \\ \text{positive antecedents with } P \vdash \frac{H}{\alpha} \text{ and } T \models H.\end{array}$$

According to Bol & Groote [3] the transition relation *associated* to a TSS is its unique stable transition relation if it exists. They argue that there is no satisfying way to accociate a tranition relation to a TSS that has no or multiple stable transition relations.

3 Formats and congruence theorems

Definition 12 *(ntyft/ntyxt-format).* An action rule $\frac{H}{t \xrightarrow{a} t'}$ over a signature Σ is in *ntyft-format* if t has the form $f(x_1, ..., x_n)$ for certain $(f, n) \in \Sigma$ and $x_1, ..., x_n \in \mathcal{V}$, and all its positive antecedents have the form $t \xrightarrow{a} y$ with $y \in \mathcal{V} - \mathcal{V}(t)$ and all y different. It is in *ntxft-format* if t has the form $x \in \mathcal{V}$ and all its positive antecedents have the form $t \xrightarrow{a} y$ with $x \neq y \in \mathcal{V}$ and all y different. A TSS is in *ntyft/ntyxt-format* if all its rules are in *ntyft* or *ntyxt*-format.

Definition 13 The *bound* variables of an action rule $\frac{H}{t \xrightarrow{a} t'}$ over a signature Σ are inductively defined as the ones that occur in t or in the target s' of a positive antecedent $(s \xrightarrow{b} s') \in H$ where s contains bound variables only. The rule is *pure* if all variables that occur in it are bound, and a TSS is *pure* if it consists of pure rules only. A rule has *no lookahead* if all bound variables in the source of its antecedents also occur in the source of its conclusion. *Connectedness* is the smallest equivalence relation between the bound variables that appear in a rule such that x and y are connected if there is an antecedent $x \xrightarrow{a} y$.

Definition 14 A TSS is in *bisimulation format* if it is positive after reduction —as defined in [3]—and in *ntyft/ntyxt*-format. A TSS is in *nested simulation format* or *tyft/tyxt-format* if it is positive and in *ntyft/ntyxt*-format. A TSS is in *ready simulation format* if it is in bisimulation format and its rules have no lookahead. A TSS is in *ready trace format* if it is in ready simulation format and no two occurrences of variables in the target of a rule are connected in that rule. A TSS is in *failure format* if it is positive, in ready simulation format, and all occurrences of variables in the antecedents of a rule are different.

Definition 15 *(nxyft-format).* An action rule $\frac{H}{t \xrightarrow{a} t'}$ over a signature Σ is in *nxyft-format* if it is in *ntyft*-format and its positive antecedents have the form $x \xrightarrow{a} y$ with $x, y \in \mathcal{V}$. A TSS is in *nxyft-format* if all its rules are in *nxyft*-format.

Theorem 3 Every TSS in bisimulation format can be converted into an equivalent TSS in pure *nxyft*-format. Moreover the conversion preserves the formats of Definition 14.

The proof of this theorem will appear in the full version of this paper. Theorem 3 has independently been found by WILLEM JAN FOKKINK [4]. Using Theorem 3, the following theorem follows easily from the slightly less general results published in [3, 7, 2, 9], except for the congruence theorem for ready trace semantics, which is new. In BLOOM [1] a format for readiness congruence is presented, as well as evidence that the ready trace format can be further generalized.

Theorem 4 *(Congruence).* Bisimulation equivalence is a congruence for any TSS in bisimulation format. Similarly, n-nested simulation equivalence (for any $n \in \mathbb{N}$) is a congruence for any TSS in nested simulation format, ready simulation equivalence is a congruence for any TSS in ready simulation simulation format, ready trace equivalence is a congruence for any TSS in ready trace format and failure equivalence as well as trace equivalence are congruences for any TSS in failure format.

4 Full abstraction

Definition 16 An equivalence is said to be *fully abstract* with respect to a set of operators L and another equivalence $\sim_{obs}$ if it is the coarsest congruence with respect to the operators in L that is finer that $\sim_{obs}$. An equivalence on labelled transition systems is *fully abstract* with respect to a TSS-format and an equivalence $\sim_{obs}$ if it is the coarsest congruence with respect to all operators specifiable by a TSS in that format that is finer that $\sim_{obs}$.

The following theorem, stated a bit differently, has in a slightly less general form been proven in [3, 7, 2, 9], except for the case of ready trace semantics. A proof will appear in the full version of this paper.

Theorem 5 Bisimulation equivalence is fully abstract w.r.t. the bisimulation format and trace equivalence. 2-nested simulation equivalence is fully abstract for the n-nested simulation format and completed trace equivalence. Simulation equivalence (=1-nested simulation equivalence) is fully abstract for the n-nested simulation format and trace equivalence. Ready simulation equivalence is fully abstract for the ready simulation simulation format and trace equivalence, as well as for the positive ready simulation format and completed trace equivalence. Ready trace equivalence is fully abstract for the ready trace format and trace equivalence. And failure equivalence is fully abstract for the failure format and completed trace equivalence.

References

[1] B. Bloom. Ready, set, go: Structural operational semantics for linear-time process algebras. Technical Report TR 93-1372, Department of Computer Science, Cornell University, Ithaca, New York, August 1993.

[2] B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced: Preliminary report. In *Conference Record of the* 15^{th} *ACM Symposium on Principles of Programming Languages,* San Diego, California, pages 229–239, 1988. Full version available as Technical Report 90-1150, Department of Computer Science, Cornell University, Ithaca, New York, August 1990. Accepted to appear in *Journal of the ACM.*

[3] R.N. Bol and J.F. Groote. The meaning of negative premises in transition system specifications (extended abstract). In J. Leach Albert, B. Monien, and M. Rodríguez, editors, *Proceedings* 18^{th} *ICALP,* Madrid, volume 510 of LNCS, pages 481–494. Springer-Verlag, 1991. Full version appeared as Report CS-R9054, CWI, Amsterdam, 1990.

[4] W.J. Fokkink. The tyft/tyxt format reduces to tree rules, 1993. To appear as CWI Report, Amsterdam.

[5] R.J. van Glabbeek. The linear time – branching time spectrum. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90,* Amsterdam, volume 458 of LNCS, pages 278–297. Springer-Verlag, 1990.

[6] J.F. Groote. Transition system specifications with negative premises. Report CS-R8950, CWI, Amsterdam, 1989. An extended abstract appeared in J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90*, Amsterdam, LNCS 458, pages 332–341. Springer-Verlag, 1990.

[7] J.F. Groote and F.W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202–260, October 1992.

[8] G.D. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[9] R. de Simone. Higher-level synchronising devices in MEIJE–SCCS. *Theoretical Computer Science*, 37:245–267, 1985.