The Meaning of Ordered SOS

MohammadReza Mousavi[1,2,⋆], Iain Phillips[3],
Michel A. Reniers[1], Irek Ulidowski[4]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
[2] Reykjavík University, Reykjavík, Iceland
[3] Imperial College, London, United Kingdom
[4] University of Leicester, Leicester, United Kingdom

Abstract. Structured Operational Semantics (SOS) is a popular method for defining semantics by means of deduction rules. An important feature of deduction rules, or simply SOS rules, are *negative premises*, which are crucial in the definitions of such phenomena as priority mechanisms and time-outs. Orderings on SOS rules were proposed by Phillips and Ulidowski as an alternative to negative premises. The meaning of general types of SOS rules with orderings has not been studied hitherto. This paper presents satisfactory ways of giving a meaning to general SOS rules with orderings. We also give semantics-preserving transformations between the two paradigms, namely, SOS with negative premises and SOS with orderings.

1 Introduction

It is well-known that negative premises in Structured Operational Semantics (SOS) are useful and non-trivial additions but at the same time they may lead to ambiguities and paradoxical phenomena with respect to the semantics of SOS [4, 5]. As an alternative to negative premises, [9] proposes to furnish SOS deduction rules with an ordering. But to avoid the same difficulties as those with negative premises, [9] restricts itself to the positive subset of GSOS [3] which does not allow for look-ahead or complex terms as sources of the premises.

It is also well-known from the term rewriting literature that the introduction of orderings (called priorities) to term rewrite systems introduces challenges for the well-definedness of the semantics of term rewrite systems [2]. SOS specifications can be seen as conditional term rewrite systems and thus one expects similar or even more difficult challenges when studying the general semantics of SOS with ordering.

A fundamental study of the semantics of ordered SOS (in its full generality) has not been carried out to date and even misconceptions exist. In [8, Theorem 4], it is mentioned (without formal proof) that they can generalize their particular rule format for ordered SOS with look-ahead while preserving the congruence

⋆ The work of the first author has been partially supported by the project "The Equational Logic of Parallel Processes" (nr. 060013021) of The Icelandic Research Fund.

property. However, as we show in the remainder of this paper, the introduction of either look-ahead or complex terms as sources of premises to ordered SOS jeopardizes the well-definedness of the induced transition relation (let alone the congruence result).

In the remainder of this paper, in Section 2, we define the basic concept of Ordered Transition System Specification (OTSS) which is a general framework for ordered SOS. In the same section, we give some examples, both for illustrating the applications of ordered SOS and for showing that the semantics of OTSS's is not always clear. Then, in Section 3, following [5], we define a model-theoretic and a proof-theoretic view to the meaning of ordered SOS and prove them equal. Subsequently, in Sections 4 and 5, we give semantics-preserving transformations from a novel rule format for ordered SOS (called otyft, for order tyft, where tyft is a coding for the terms admitted in the deduction rules) to a rule format for SOS with negative premises (called ntyft [6]) and vice versa. In Section 6 we show that our otyft rule format indeed induces congruence for strong bisimilarity. Section 7 concludes the paper.

2 (Ordered) Transition System Specification

2.1 Basic Concepts

Definition 1 (Signature, Term and Substitution) Assume a countable set of variables V (with typical members x, y, x', y', x_i, $y_i \ldots$). A *signature* Σ is a set of function symbols (operators, with typical members f, g, $\ldots$) with fixed arities $ar : \Sigma \to I\!N$. Functions with zero arity are called constants and are typically denoted by a, b, c and d. Terms $s, t, t_i, \ldots \in \mathcal{T}$ are constructed inductively using variables and function symbols. A list of terms is denoted by $\overrightarrow{t}$. When we write $f(\overrightarrow{t})$, we assume that $\overrightarrow{t}$ has the right size, i.e., $ar(f)$. All terms are considered open terms. Closed terms $p, q, \ldots \in \mathcal{C}$ are terms that do not contain a variable and are typically denoted by $p, q, l, p', p_i, \ldots$. The set of variables appearing in term t is denoted by $vars(t)$.

Definition 2 (Ordered Transition System Specification (OTSS)) Given a signature and a set of variables, a *Transition System Specification (TSS)* is a set R of deduction rules.

A deduction rule $r \in R$, is defined as a tuple (H, c) where H is a set of formulae and c is a positive formula. For all $t, t' \in \mathcal{T}$ and $l \in \mathcal{C}$ we define that $\phi = t \xrightarrow{l} t'$ is a positive formula and $\phi' = t \xrightarrow{l} \not\rightarrow$ is a negative formula. A formula is a positive or a negative formula. We denote the set of formulae by Φ and the set of positive fomulae by Φ_p. Term t is called the source of both ϕ and ϕ', denoted by $src(\phi)$ and $src(\phi')$, and t' is called the target of ϕ. The formula c is called the *conclusion* of r, denoted by $conc(r)$, and the formulae in H are called its *premises* and denoted by $prem(r)$. A positive deduction rule (TSS) is a deduction rule of which all the premises (all the deduction rules) are positive. The notions of source and target generalize to a set of formulae, as expected.

Also, the notion of "variables of" is naturally lifted to sets of terms, formulae, sets of formulae and deduction rules.

An *Ordered Transition System Specification (OTSS)* is a pair $(R, <)$ where R is a *positive TSS* and $< \subseteq R \times R$ is an *arbitrary relation* on the deduction rules. For a rule r, $\mathit{higher}(r)$ is defined as $\{r' \mid r' \in R \wedge r' > r\}$, i.e., the set of rules placed above r by the ordering $<$.

The intuition behind the ordering on rules is that a rule r can only be applied when all rules $r' \in \mathit{higher}(r)$ are disabled since they do not have a "reason" (or "proof") for their premises to hold. As we show in the remainder, this notion of "reason" or "proof" is not trivial to define and involves the same complications as those concerning the semantics of TSS's with negative premises [5].

2.2 Rule Formats

A major line of research in the SOS meta-theory concerns defining syntactic schema for TSS's which guarantee certain properties such as congruence of strong bisimilarity. A distinguished example of such rule formats is the ntyft rule format due to [6] which has powerful and complicating features such as look-ahead and negative premises.

Definition 3 ((N)Tyft) A rule is in the ntyft rule format when it is of the form $\dfrac{\{t_i \xrightarrow{l_i} y_i \mid i \in I\} \cup \{t_j \xrightarrow{l_j} \mid j \in J\}}{f(\overrightarrow{x}) \xrightarrow{l} t}$ where all variables in $\overrightarrow{x}$ and y_i's are pairwise distinct (i.e., for all $i, i' \in I$ and $1 \leq j < j' \leq ar(f)$, $y_i \neq x_j$, $x_j \neq x_{j'}$ and if $i \neq i'$ then $y_i \neq y_{i'}$), f is a function symbol from the signature, I and J are (possibly infinite) sets of indices, t, t_i's and t_j's are arbitrary terms and l, l_i's and l_j's are closed terms.

A TSS is in the ntyft rule format when all its rules are. A rule (TSS) is in the tyft rule format when it is positive and in the ntyft rule format.

Our goal is to show that ordering on rules is at least as expressive (and of course complicated in nature) as negative premises and thus, we introduce the following otyft rule format which will be proved equal in expressiveness to the ntyft rule format (in Sections 4 and 5).

Definition 4 (Otyft) An OTSS $(R, <)$ is in the otyft rule format when (1) for all rules $r \in R$, either r is in the tyft rule format or $\mathit{conc}(r) \in \mathit{prem}(r)$, (2) for all rules $r, r' \in R$ such that $r' \in \mathit{higher}(r)$ (a) if a premise of r has the same target as that of a premise of r', then the two premises are the same (i.e., have the same source and label) and (b) $\mathit{vars}(\mathit{src}(\mathit{prem}(r'))) \subseteq (\mathit{vars}(\mathit{prem}(r)) \cup \mathit{vars}(\mathit{src}(\mathit{conc}(r))))$.

The above rule format generalizes the OSOS rule format of [9] by allowing for look-ahead and arbitrary terms as sources of premises (both conditions 2.a and 2.b are trivially satisfied by OTSS's in the OSOS rule format). In the forthcoming

extended version of this paper, we prove that removing condition 2.b gives rise to a more general rule format called universal otyft, while preserving the congruence result. The expressiveness of this rule format, called universal otyft, goes beyond that of the ntyft rule format in that all transition relations specifiable by a TSS in the ntyft rule format can be specified by an OTSS in the universal otyft rule format but *not* vice versa.

The non-tyft rules allowed by the otyft rule format are mainly for convenience: our definitions of semantics in Section 3 are insensitive to such rules and they are useful in the translation between the ntyft and the otyft rule formats in Section 4.

2.3 Examples

Orderings on positive rules can replace negative premises in rules [9]. In the remainder, we start with a simple example motivating the use of ordering (as an alternative to negative premises). Then we show that our new otyft rule format extends the applicability of the ordered SOS paradigm by specifying an example involving look-ahead. Finally, we show that this extension comes at a price, namely, the semantics of general OTSS's (e.g., those involving look-ahead) is not always clear and should be studied more thoroughly.

Example 5 (Priority) The priority operator θ [1] may be used to represent such phenomena as time-outs and interrupts. For a given partial order $\prec$ on actions (a set of constants, denoted by a, b, c, $\ldots \in Act$), $\theta(p)$ is a restriction on the behavior of p such that action a can happen only if no b with $a \prec b$ is possible. If $B_a = \{b \mid a \prec b\}$, then θ can be defined by this TSS (where the rule is actually a rule schema which should be repeated for each action $a \in Act$): $\dfrac{x \xrightarrow{a} y \quad \{x \overset{b}{\not\rightarrow} \mid b \in B_a\}}{\theta(x) \xrightarrow{a} \theta(y)}$.

Alternatively, θ can be defined by positive rules r_a, equipped with the ordering defined by $r_a < r_b$ whenever $a \prec b$: $(r_a)\dfrac{x \xrightarrow{a} y_a}{\theta(x) \xrightarrow{a} \theta(y_a)}$ where y_a are distinct variables for all $a \in Act$. (Note that the naming of variables in the rules related by ordering is indeed important; if for two different actions a and b such that $a \prec b$, $y_a = y_b$, then the OTSS specified by the above rules is not in the otyft rule format and as shown in Section 6, it may lack intuitive properties such as congruence of bisimilarity.)

Example 6 (Timed Parallel Composition) Consider the following TSS defining the semantics of a subset of Hennessy and Regan's Process Algebra for Timed Systems (TPA) [7]. The signature consists of a constant nil, unary operators $a._-$ (action prefixing, for all $a \in Act$), $\tau._-$ (internal action prefixing) and $\sigma._-$ (time step prefixing) and a binary operator $\|$ (parallel composition). (Constants a, τ σ are also introduced in the signature to model the labels.)

$$(a)\frac{}{a.x \xrightarrow{a} x} \quad (\tau)\frac{}{\tau.x \xrightarrow{\tau} x} \quad (\sigma_0)\frac{}{\sigma.x \xrightarrow{\sigma} x} \quad (\sigma_1)\frac{}{a.x \xrightarrow{\sigma} a.x} \quad (\sigma_2)\frac{}{nil \xrightarrow{\sigma} nil}$$

$$(\|_0)\frac{x_0 \xrightarrow{a} y_0}{x_0 \parallel x_1 \xrightarrow{a} y_0 \parallel x_1} \quad (\|_1)\frac{x_1 \xrightarrow{a} y_1}{x_0 \parallel x_1 \xrightarrow{a} x_0 \parallel y_1}$$

$$(\tau_0)\frac{x_0 \xrightarrow{\tau} y_0}{x_0 \parallel x_1 \xrightarrow{\tau} y_0 \parallel x_1} \quad (\tau_1)\frac{x_1 \xrightarrow{\tau} y_1}{x_0 \parallel x_1 \xrightarrow{\tau} x_0 \parallel y_1}$$

$$\textbf{(comm)}\frac{x_0 \xrightarrow{a} y_0 \quad x_1 \xrightarrow{\overline{a}} y_1}{x_0 \parallel x_1 \xrightarrow{\tau} y_0 \parallel y_1} \quad \textbf{(time)}\frac{x_0 \xrightarrow{\sigma} y_0 \quad x_1 \xrightarrow{\sigma} y_1 \quad x_0 \parallel x_1 \xrightarrow{\tau} \not\rightarrow}{x_0 \,||\, x_1 \xrightarrow{\sigma} y_0 \,||\, y_1}$$

In the semantics of the parallel composition operator, $p \parallel q$ can pass time (denoted by label σ) if both p and q can pass time, and if they are stable and cannot communicate (i.e. $p \parallel q \xrightarrow{\tau} \not\rightarrow$).

The above semantics can be specified in ordered SOS by placing a positive version of the rule **(time)** below the rules (τ_0), (τ_1) and **(comm)** as shown below. All other rules are copied to the following OTSS and are unrelated (in terms of ordering) to the rules below.

$\downarrow$	$(\tau_0)\frac{x_0 \xrightarrow{\tau} y_0}{x_0 \parallel x_1 \xrightarrow{\tau} y_0 \parallel x_1}$	$(\tau_1)\frac{x_1 \xrightarrow{\tau} y_1}{x_0 \parallel x_1 \xrightarrow{\tau} x_0 \parallel y_1}$	$\textbf{(comm)}\frac{x_0 \xrightarrow{a} y_0 \quad x_1 \xrightarrow{\overline{a}} y_1}{x_0 \parallel x_1 \xrightarrow{\tau} y_0 \parallel y_1}$
	$\textbf{(time)}\frac{x_0 \xrightarrow{\sigma} y_0' \quad x_1 \xrightarrow{\sigma} y_1'}{x_0 \parallel x_1 \xrightarrow{\sigma} y_0' \parallel y_1'}$		

We fix the above notation for ordering so that in each column, rules of the upper row have priority over rules of the lower row, i.e., rules of the lower row can only be "applied" when no rule in the upper row (of the same column) can be "applied". Formally, we have the following orderings: $(\tau_0) >$ **(time)**, $(\tau_1) >$ **(time)**, and **(comm)** > **(time)**.

In the following example, we address the idea of extending OSOS [9] with look-ahead as suggested by [8, Theorem 4] and show that it may lead to pathological specifications with an unclear meaning. (The rule format of [8] extends traditional OTSS with probabilities but the problem we address below is orthogonal to the presence or absence of probabilities and hence, we use the plain OTSS setting as defined before.)

Example 7 (OSOS with Look-Ahead) Consider the OTSS with the following rules. Note that according to the notation fixed before, in the following OTSS, it holds that $\frac{x \xrightarrow{b} y \quad y \xrightarrow{d} z}{f(x) \xrightarrow{d} d} > \frac{x \xrightarrow{b} y}{f(x) \xrightarrow{c} c}$ and $\frac{x \xrightarrow{a} y \quad y \xrightarrow{c} z}{g(x) \xrightarrow{c} c} > \frac{x \xrightarrow{a} y}{g(x) \xrightarrow{d} d}$ but it does not hold that $\frac{}{a \xrightarrow{a} f(a)} > \frac{x \xrightarrow{b} y}{f(x) \xrightarrow{c} c}$.

$\downarrow$	$\dfrac{x \xrightarrow{b} y \quad y \xrightarrow{d} z}{f(x) \xrightarrow{d} d}$	$\dfrac{x \xrightarrow{a} y \quad y \xrightarrow{c} z}{g(x) \xrightarrow{c} c}$	$\dfrac{}{a \xrightarrow{a} f(a)}$	$\dfrac{}{a \xrightarrow{b} g(a)}$
	$\dfrac{x \xrightarrow{b} y}{f(x) \xrightarrow{c} c}$	$\dfrac{x \xrightarrow{a} y}{g(x) \xrightarrow{d} d}$		

At first sight, it is not intuitively clear which of the following three transition relations should be considered as the meaning of the above OTSS.

1. $\{a \xrightarrow{a} f(a), a \xrightarrow{b} g(a), f(a) \xrightarrow{c} c, g(a) \xrightarrow{c} c\}$ or
2. $\{a \xrightarrow{a} f(a), a \xrightarrow{b} g(a), f(a) \xrightarrow{d} d, g(a) \xrightarrow{d} d\}$ or
3. $\{a \xrightarrow{a} f(a), a \xrightarrow{b} g(a)\}$.

So, a convincing semantics for OTSS's should either be neutral about different possibly derivable transitions (in items 1 and 2) or reject the above OTSS altogether due to its ambiguous nature. We present solutions that cater for both possibilities in the remainder of this paper.

The situation with the following OTSS is even worse.

$\downarrow$	$\dfrac{x \xrightarrow{a} y \quad y \xrightarrow{b} z}{f(x) \xrightarrow{b} a}$	$\dfrac{}{a \xrightarrow{a} f(a)}$
	$\dfrac{x \xrightarrow{a} y}{f(x) \xrightarrow{b} b}$	

If one initially assumes that from rules in the first row one cannot derive any transition with $f(a)$ as its source (which is a legitimate assumption), then the rule below allows for deriving $f(a) \xrightarrow{b} b$. This transition, in turn, enables the premises of the rule above it (leading to the conclusion that $f(a) \xrightarrow{b} a$ should be derivable) and thus the very same rule below must have been disabled and the chain of contradictory conclusions goes forever. Again, any convincing semantics for OTSS's should either find a way to deal with the contradicting conclusions (e.g., by considering all of them uncertain, yet possibly, derivable transitions) or reject the above OTSS altogether due to its paradoxical nature. The notions of semantics presented in the remainder allow for both interpretations.

The above examples make the case for a more profound study of the meaning of ordered SOS which is the subject of the following section.

3 Semantics of OTSS

An OTSS is supposed to induce a unique transition relation on closed terms but as Example 7 already suggested, for some OTSS's the way to assign such

a transition relation may not be straightforward. This phenomenon has been known in several areas such as logic programming and term rewriting and even inside the SOS meta-theory as the result of introducing negative premises to SOS rules. For TSS's with negative premises, several notions of semantics have been defined and used of which [5] provides an overview and a comparison. In this paper, due to lack of space, we only present two very general model-theoretic and proof-theoretic approaches to giving semantics to OTSS's. To avoid repeating the phrase "an instance of rule r under a closing substitution σ", in this section, we assume that the OTSS's only contain closed terms. To define the semantics of an arbitrary OTSS, one may instantiate the rules and the ordering relation under all closing substitutions and then use the notions of the semantics in the remainder of this section.

We start with the following notion of provability which is the usual way of giving semantics to ordinary positive TSS's (i.e., without ordering or negative premises).

Definition 8 (Proof) Given an OTSS $(R, <)$, a proof p for a formula ϕ is a well-founded upwardly branching tree of which

1. the nodes are formulae,
2. the root is ϕ, and
3. if a node is labelled ϕ' and the nodes above it form the set K; then there is a rule $r = \frac{K}{\phi'} \in R$.

An r-proof for ϕ is a proof in which the last step is due to rule r. We write $\vdash_p \phi$ when p is a proof in $(R, <)$ for ϕ. We denote the set of rules used in a proof p by $\mathit{rules}(p)$.

3.1 Model-Theoretic Solution

For OTSS's the above notion of provability is too lax because it neglects the ordering among rules. Hence, we have to provide an addendum to the above concept which makes sure that the rules placed above those used in the proof are disabled. The first way to specify this addendum is the following (model-theoretic) notion of correctness.

Definition 9 (Correct) Given an OTSS $(R, <)$ and a transition relation T, we say that a rule $r = \frac{H}{\phi} \in R$ is correct w.r.t. T when for all $r' = \frac{H'}{\psi} \in \mathit{higher}(r)$, $H' \not\subseteq T$.

Our first solution is based on the following notion of three-valued stable model. Such three-valued solutions assign three transition relations to each OTSS, namely the set of transitions that are certainly derivable denoted by C, transitions that are possibly derivable denoted by P (thus $C \subseteq P$) and the set of transitions that are impossible denoted by I. Three-valued solutions may be written as pairs of these sets, i.e., (C, P) or (C, I), with the third component

being easily constructed given the other two. Later in this section, we discuss how to adopt the three-valued stable model to define a single transition relation for an OTSS.

Definition 10 (Three-Valued Stable Model) Given an OTSS $(R, <)$, a pair of transition relations (C, P) is a three-valued stable model when $C \subseteq P$ and

1. $\phi \in C \Leftrightarrow \vdash_p \phi$ for some proof p such that $\forall_{r \in rules(p)}\ r$ is correct w.r.t. P and
2. $\phi \in P \Leftrightarrow \vdash_p \phi$ for some proof p such that $\forall_{r \in rules(p)}\ r$ is correct w.r.t. C.

The third value of the stable model I, for impossible, is the set of transitions that are not included in P. Of particular interest, among three-valued stable models, is the least one with respect to the information ordering, i.e., $(C, P) < (C', P')$ when $C \subseteq C'$ and $P' \subseteq P$.

The following reduction technique [4] is a method to calculate the least three-valued stable model (thus such a least model indeed exists).

Definition 11 (Reduction Technique) For an ordinal α, define:

$$
\begin{array}{l}
C_0 \doteq \emptyset \\
U_0 \doteq \Phi_p \\
C_\alpha \doteq \{\phi \mid \ \vdash_p \phi \wedge \exists_{\beta<\alpha} \forall_{r \in rules(p)} r \text{ is correct w.r.t. } C_\beta \cup U_\beta\} \\
U_\alpha \doteq \{\phi \mid \ \vdash_p \phi \wedge \forall_{\beta<\alpha} \forall_{r \in rules(p)} r \text{ is correct w.r.t. } C_\beta\}
\end{array}
$$

Lemma 12 Given an OTSS $(R, <)$, for all ordinals α and β such that $\alpha < \beta$, the following statements hold:

1. $C_\alpha \subseteq C_\beta$;
2. $U_\beta \subseteq U_\alpha$;
3. $C_\beta \subseteq C_\alpha \cup U_\alpha$;
4. $C_\beta \cup U_\beta \subseteq C_\alpha \cup U_\alpha$.

From items 1 and 2 of the above lemma (and Tarski's fixpoint theorem), it follows that both C_α and U_α will reach fixpoints, which we denote by C and U, respectively. From item 1-4 and Definition 11, it follows that $(C, C \cup U)$ is the least three-valued stable model of the OTSS under consideration.

Example 13 Consider the first OTSS of Example 7. Its three-valued stable model consists of a certain component $C = \{a \xrightarrow{a} f(a), a \xrightarrow{b} g(a)\}$ and a possible component $P = \{a \xrightarrow{a} f(a), a \xrightarrow{b} g(a),\ f(a) \xrightarrow{c} c, g(a) \xrightarrow{c} c,\ f(a) \xrightarrow{d} d, g(a) \xrightarrow{d} d\}$.

Similarly, for the second OTSS of Example 7, the certain component of the least three-valued stable model only contains $a \xrightarrow{a} f(a)$ and the possible component contains $a \xrightarrow{a} f(a)$ as well as both $f(a) \xrightarrow{b} a$ and $f(a) \xrightarrow{b} b$.

Now the question is how to reduce the three-valued model to a two-valued one, i.e., to associate a unique transition relation to a (meaningful) OTSS. The following notions provide two plausible answers to this question.

Semantics 1 (Complete) An OTSS is meaningful when for its least three-valued stable model (C, P) it holds that $C = P$ (such an OTSS is called *complete*) and its meaning is the least three-valued stable model.

In order to obtain useful meta-results, e.g., the congruence meta-result (discussed in Section 6), one has to restrict attention to complete OTSS's and for practical applications the OTSS under consideration should be complete or should be rejected. However, one might want to generalize Semantics 1 to the following notion of irrefutability which assigns a transition relation to all OTSS's.

Semantics 2 (Irrefutable) All OTSS are meaningful and their meaning is the P component of their least three-valued stable model.

3.2 Proof-Theoretic Solution

An alternative way of giving semantics to OTSS's is by means of a well-supported proof. In a well-supported proof, in addition to constructing a traditional proof, we provide a "proof" for inapplicability of the higher rules; such a "proof" is called a *well-supported denial*. A well-supported denial makes sure that a formula is not derivable since all proofs leading to the formula contain a rule that is provably disabled (i.e., there is a higher rule that has a well-supported proof for all of its premises).

Definition 14 (Well-Supported Proof) Given an OTSS $(R, <)$ and a rule $r \in R$, a well-supported r-proof (or just a well-supported proof) for ϕ is a well-founded upwardly branching tree of which

1. the nodes are formulae,
2. the root is ϕ,
3. if a node is labelled ϕ' and the nodes above it form the set K, then there is a rule $r' = \frac{K'}{\phi'} \in R$ such that $K' \subseteq K$ (for the root node, $r' = r$) and for all $r'' = \frac{H'}{\psi} \in \mathit{higher}(r')$, there exists a set $D_{\psi'} \subseteq K$ denying some $\psi' \in H'$ by a well-supported proof.

A set D_ϕ denies a formula ϕ when for all proofs p such that $\vdash_p \phi$ (in the sense of Definition 8), there exists a rule $r \in \mathit{rules}(p)$ and there exists a rule $r' = \frac{H'}{\phi'} \in \mathit{higher}(r)$ such that $H' \subseteq D_\phi$. The structure providing a well-supported proof for all $\psi \in D_\phi$ is called a well-supported denial for ϕ.

We write $\vdash_{ws} \phi$ ($\vdash_{ws} \neg\phi$) when there is a well-supported proof (denial) for ϕ.

The following theorem states that the model-theoretic and the proof-theoretic views of the least well-supported semantics indeed match.

Theorem 15 Given an OTSS $(R, <)$, let C' be the set of all formulae that have a well-supported proof and I' the set of all formulae that have a well-supported denial. Let (C, P) be the least three-valued stable model of $(R, <)$. Then, $(C', (\Phi_p \setminus I')) = (C, P)$.

4 From Ntyft to Otyft

We assume in the remainder that the TSS's in the ntyft rule format are pure, i.e., only contain variables among the source of the conclusion and targets of the premises. Impure TSS's can be transformed to pure ones (while keeping the TSS in ntyft rule format and preserving the semantics) by making many copies of rules each instantiating the other variables by a closed term [6]. (Hence, there is no expressiveness gap between the ntyft rule format and its pure subset, i.e., all transition relations that can be specified by the ntyft rule format can also be specified by the pure ntyft rule format and vice versa.) Our translation is correct for impure rules, as well but the outcome will not be in the otyft rule format.

Definition 16 (Pure Ntyft to Otyft: Translation Scheme) Given a TSS R in the pure ntyft rule format, its translation to otyft, denoted by $otyft(R)$, is an OTSS $(R', <)$ where $R' \doteq \{r^+, s_{r,j} \mid r \in R, j \in J_r\}$, $<\doteq \{(r^+, s_{r,j}) \mid r \in R, j \in J_r\}$ and for each $r \in R$ of the form $\frac{\{t_i \xrightarrow{l_i} y_i | i \in I_r\} \cup \{t_j \not\xrightarrow{l_j} | j \in J_r\}}{f(\overrightarrow{x}) \xrightarrow{l} t}$, r^+ and $s_{r,j}$ (for each $j \in J_r$) are defined as follows.

$$(r^+)\frac{\{t_i \xrightarrow{l_i} y_i | i \in I_r\}}{f(\overrightarrow{x}) \xrightarrow{l} t} \qquad (s_{r,j})\frac{\{t_j \xrightarrow{l_j} y_j\}}{t_j \xrightarrow{l_j} y_j}$$

In rule $s_{r,j}$, y_j is a fresh variable not appearing in r.

The following theorem states that the diagram depicted in Figure 1.(a) commutes.

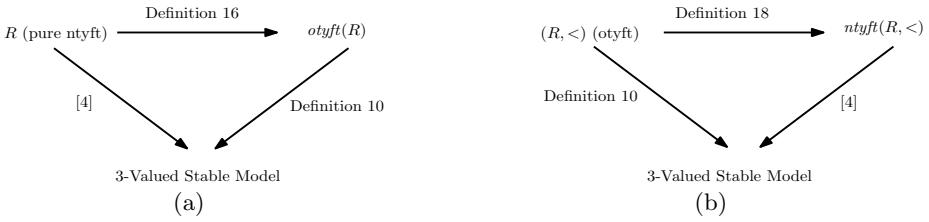

Fig. 1. Correctness of translations: (a) from ntyft to otyft and (b) from otyft to ntyft

Theorem 17 (Pure Ntyft to Otyft: Correctness) The translation from pure ntyft to otyft preserves its three-valued stable model.

5 From Otyft to NTyft

Definition 18 (Otyft to Ntyft: Translation Scheme) Given an OTSS $(R, <)$ in the otyft rule format, partial function $S_r : R \rightharpoonup \mathcal{I}$, where $\mathcal{I} \doteq \bigcup_{i \in I_r} I_r$, is a selection function for $r \in R$ when for all $s \in \mathit{higher}(r)$ of the form $\frac{\{t_i \xrightarrow{l_i} y_i | i \in I_s\}}{t \xrightarrow{l} t'}$, it holds that $S_r(s) \in I_s$. (Thus, if $I_s = \emptyset$ for some $s \in \mathit{higher}(r)$, then the set of selection functions for r is empty.)

Given an OTSS $(R, <)$ in the otyft rule format, its translation to ntyft, denoted by $\mathit{ntyft}(R, <)$, is defined as $\{r_S \mid r \in \mathit{tyft}(R), S \text{ is a selection function for } r\}$ where $\mathit{tyft}(R)$ is the subset of R that conforms to the tyft rule format and for each $r \in \mathit{tyft}(R)$ of the form $\frac{\{t_i \xrightarrow{l_i} y_i | i \in I_r\}}{f(\overrightarrow{x}) \xrightarrow{l} t}$, r_S is defined as follows.

$$(r_S)\ \frac{\{t_i \xrightarrow{l_i} y_i | i \in I_r\} \cup \{t_{S(s)} \overset{l_{S(s)}}{\nrightarrow} | s \in \mathit{higher}(r)\}}{f(\overrightarrow{x}) \xrightarrow{l} t}$$

The idea of the above translation is that for each rule r in R, for all rules s placed above r, an arbitrary premise $S(s)$ is negated and included in the premises of r_S. This way, we make sure that r_S is applicable if and only if r is applicable and all rules above it are disabled. We can safely exclude rules that do not conform to the tyft rule format in our translation since their conclusion is among their premises and thus, they do not contribute to the least three-valued well-supported model.

The following theorem states that the diagram depicted in Figure 1.(b) commutes.

Theorem 19 (Otyft to Ntyft: Correctness) The translation from otyft to ntyft preserves its three-valued stable model.

6 Congruence Meta-Theorem

As it is shown in [4], for a complete TSS in the ntyft rule format, strong bisimilarity is a congruence. Since our translation (in Section 5) provably preserves the three-valued stable model, we can recast this result to the setting with ordered SOS, as follows.

Theorem 20 (Congruence for Otyft) For a complete OTSS in the otyft rule format, bisimilarity is a congruence.

Note that our only essential addition to the constraints of tyft rule format is the constraint 2.a of Definition 4 (as mentioned before, constraint 2.b can be removed and is only needed to obtain compatibility with the ntyft rule format). The following counter-example shows that constraint 2.a is indeed useful for the purpose of congruence and cannot be dropped.

Example 21

$\downarrow$	$\dfrac{b \xrightarrow{a} y}{b \xrightarrow{a} y}$	$\dfrac{}{a \xrightarrow{a} b}$	$\dfrac{}{b \xrightarrow{a} a}$
	$\dfrac{x \xrightarrow{a} y}{f(x) \xrightarrow{a} a}$		

The above OTSS is complete and it meets all the constraints of Definition 4 except for the constraint 2.a. The C (P) component of the least three-valued stable model of the above OTSS is $\{a \xrightarrow{a} b, b \xrightarrow{a} a, f(a) \xrightarrow{a} a\}$ but $f(b) \xrightarrow{a} a$ is not included in it. Hence, for the above OTSS a and b are bisimilar while $f(a)$ and $f(b)$ are not.

7 Conclusions

In this paper, we presented ways of giving a meaning to ordered SOS specifications. Furthermore, we gave semantics-preserving translations (w.r.t. our chosen notion of semantics) between general ordered SOS and SOS rule formats, namely otyft and ntyft, respectively. Finally, thanks to our semantics-preserving translation, we obtained a congruence meta-result for complete OTSS's in the otyft rule format.

References

1. J. C. M. Baeten, J. A. Bergstra, J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae*, XI(2):127–168, 1986.
2. J. C. M. Baeten, J. A. Bergstra, J.W. Klop, and W. P. Weijland. Term-rewriting systems with rule priorities. *TCS*, 67(2&3):283–301, 1989.
3. B. Bloom, S. Istrail and A.R. Meyer. Bisimulation Can't Be Traced. *JACM*, 42(1):232–268, 1995.
4. R. Bol and J.F. Groote. The meaning of negative premises in transition system specifications. *JACM*, 43(5):863–914, 1996.
5. R.J. van Glabbeek. The meaning of negative premises in transition system specifications II. *JLAP*, 60-61:229–258, 2004.
6. J.F. Groote. Transition system specifications with negative premises. *TCS*, 118(2):263–299, 1993.
7. M. Hennessy and T. Regan. A process algebra for timed systems. *I&C*, 117(2):221–239, 1995.
8. R. Lanotte and S. Tini. Probabilistic congruence for semistochastic generative processes. In *Proceedings of FOSSACS'05*, volume 3441 of *LNCS*, pages 63–78. Springer, 2005.
9. I. Ulidowski and I.C.C. Phillips. Ordered SOS Rules and Process Languages for Branching and Eager Bisimulations. *I&C*, 178(1):180–213, 2002.