The Meaning of Negative Premises in Transition System Specifications II

(extended abstract)[†]

R.J. van Glabbeek[*]
Computer Science Department, Stanford University
Stanford, CA 94305, USA.
rvg@cs.stanford.edu

This paper reviews several methods to associate transition relations to transition system specifications with negative premises in Plotkin's structural operational style. Besides a formal comparison on generality and relative consistency, the methods are also evaluated on their taste in determining which specifications are meaningful and which are not.

1 Transition system specifications

In this paper V and A are two sets of *variables* and *actions*. Many concepts that will appear are parameterised by the choice of V and A, but as in this paper this choice is fixed, a corresponding index is suppressed.

Definition 1 (*Signatures*). A *function declaration* is a pair (f,n) of a *function symbol* $f \notin V$ 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 recursively by:

- $V \subseteq \mathbb{T}(\Sigma)$,
- 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. A Σ-*substitution* σ is a partial function from V to $\mathbb{T}(\Sigma)$. If σ is a substitution and S any syntactic object (built from terms), 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. S is said to be *closed* if it contains no variables. The set of closed terms is denoted $\mathsf{T}(\Sigma)$.

Definition 2 (*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 $t \not\xrightarrow{a}$ or $t \not\xrightarrow{a} t'$ with $t,t' \in \mathbb{T}(\Sigma)$ and $a \in A$. For $t,t' \in \mathbb{T}(\Sigma)$ the literals $t \xrightarrow{a} t'$ and $t \not\xrightarrow{a}$, as well as $t \xrightarrow{a} t'$ and $t \not\xrightarrow{a} t'$, are said to *deny* each other. A *transition rule* over Σ is an expression of the form $\frac{H}{\alpha}$ with H a set of Σ-literals (the *premises* or *antecedents* of the the rule) and α a Σ-literal (the *conclusion*). A rule $\frac{H}{\alpha}$ with $H = \emptyset$ is also written α. An *action rule* is a transition rule 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 *standard* if its rules have no antecedents of the form $t \not\xrightarrow{a} t'$, and *positive* if all antecedents of its rules are positive.

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

[†]In Proc. ICALP '96, Paderborn (F. Meyer auf der Heide & B. Monien, eds.), LNCS 1099, pp. 512-513, 1996. Full version available as Report STAN-CS-TN-95-16, 1995, and by ftp from boole.stanford.edu.

The first systematic study of transition system specifications with negative premises appears in Bloom, Istrail & Meyer [2]. The concept of a (positive) TSS presented above was introduced in Groote & Vaandrager [9]; the negative premises $t \stackrel{a}{\not\longrightarrow}$ were added in Groote [8]. The notion generalises the *GSOS rule systems* of [2] and constitutes the first formalisation of Plotkin's *Structural Operational Semantics (SOS)* [10] that is sufficiently general to cover most, if not all, of its applications. The premises $t \stackrel{a}{\not\longrightarrow} t'$ are added here, mainly for technical reasons.

The following definition tells when a transition is provable from a TSS. It generalises the standard definition (see e.g. [9]) by (also) allowing the derivation of transition rules. The derivation of a transition $t \stackrel{a}{\longrightarrow} t'$ corresponds to the derivation of the transition rule $\frac{H}{t \stackrel{a}{\longrightarrow} t'}$ with $H = \emptyset$. The case $H \neq \emptyset$ corresponds to the derivation of $t \stackrel{a}{\longrightarrow} t'$ under the assumptions H.

Definition 3 (*Proof*). Let $P = (\Sigma, R)$ be a TSS. A *proof* of a transition rule $\frac{H}{\alpha}$ 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

1. either $K = \emptyset$ and $\beta \in H$,
2. 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 $P \vdash \frac{H}{\alpha}$.
A closed negative literal α is *refutable* if $P \vdash \beta$ for a literal β denying α.

Definition 4 (*Transition relation*). Let Σ be a signature. A *transition relation* over Σ is a relation $T \subseteq \mathsf{T}(\Sigma) \times A \times \mathsf{T}(\Sigma)$. Elements (t, a, t') of a transition relation are written as $t \stackrel{a}{\longrightarrow} t'$. 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 α is positive and $\alpha \in T$ or $\alpha = (t \stackrel{a}{\not\longrightarrow} t')$ and $(t \stackrel{a}{\longrightarrow} t') \notin T$ or $\alpha = (t \stackrel{a}{\not\longrightarrow})$ and $(t \stackrel{a}{\longrightarrow} t') \in T$ for no $t' \in \mathsf{T}(\Sigma)$. Write $T \models H$, for H a set of closed literals, if $T \models \alpha$ for all $\alpha \in H$. Write $T \models p$, for p a closed proof, if $T \models \alpha$ for all literals α that appear as node-labels in p.

A positive TSS specifies a transition relation in a straightforward way as the set of all provable transitions. But as pointed out in Groote [8], it is much less trivial to associate a transition relation to a TSS with negative premises. Several solutions are proposed in [8] and Bol & Groote [3]. Here I will present these solutions from a somewhat different point of view, and also review a few others.

P_1

$$\frac{c \stackrel{a}{\not\longrightarrow}}{c \stackrel{b}{\longrightarrow} c} \qquad \frac{c \stackrel{b}{\not\longrightarrow}}{c \stackrel{a}{\longrightarrow} c}$$

The TSS P_1 can be regarded as an example of a TSS that does not specify a well-defined transition relation (under any plausible definition of 'specify').[1] So unless a systematic way can be found to associate a meaning to TSSs like P_1, one has to accept that some TSSs are meaningless. Hence there are two questions to answer:

Which TSSs are meaningful, (1)
and which transition relations do they specify? (2)

[1] All my examples P_i consider TSSs (Σ, R) in which Σ consists of the single constant c only.

In this paper I present 8 possible answers to these questions, each consisting of a class of TSSs and a mapping from this class to transition relations. Two such solutions are *consistent* if they agree which transition relation to attach to a TSS in the intersection of their domains. Solution S' *extends* S if the class of meaningful TSSs according to S' extends that of S and the two are consistent, i.e. seen as partial functions $S \subseteq S'$.

Logic programming

The problems analysed in [8] in associating transition relations to TSSs with negative premises had been encountered long before in logic programming, and most of the solutions reviewed in the present paper stem from logic programming as well. However, the proof theoretic approach to Solution 7, as well as Solutions 6 and 8 and some comparative observations, are, as far as I know, new here.

The connection with logic programming may be best understood by introducing *proposition system specifications (PSSs)*. These are obtained by replacing the set A of actions by a set of *predicate declarations* (p, n) with $p \notin V$ a *predicate symbol* (different from any function symbol) and $n \in \mathbb{N}$. A literal is then an expression $p(t_1, \ldots, t_n)$ or $\neg p(t_1, \ldots, t_n)$ with $t_i \in \mathbb{T}(\Sigma)$. A PSS is now defined in terms of literals in a same way as a TSS. A *proposition* is a closed positive literal, and a *proposition relation* or *closed theory* a set of propositions. The problem of associating a proposition relation to a PSS is of a similar nature as associating a transition relation to a TSS, and in fact all concepts and results mentioned in this paper apply equally well to both situations.

If I would not consider TSS involving literals of the form $t \stackrel{a}{\not\longrightarrow}$, a TSS would be a special case of a PSS, and it would make sense to present the paper in terms of PSSs. The main reason for not doing so is to do justice to the rôle of literals $t \stackrel{a}{\not\longrightarrow}$ in denying literals of the form $t \stackrel{a}{\longrightarrow} t'$. However, as elaborated in the full paper, every TSS can be encoded as a PSS and vice versa, in such a way that all concepts of this paper are preserved under the translations.

A logic program is just a PSS obeying some finiteness conditions. Hence everything I say about TSSs applies to logic programming too. Consequently, this paper can in part be regarded as an overview of a topic within logic programming, but avoiding the logic programming jargon. However, I do not touch issues that are relevant in logic programming, but not manifestly so for transition system specifications. For these, and many more references, see Apt & Bol [1].

2 Model theoretic solutions

Solution 1 (*Positive*). A first and rather conservative answer to (1) and (2) is to take the class of positive TSSs as the meaningful ones, and associate with each positive TSS the transition relation consisting of the provable transitions.

Before proposing more general solutions, I will first recall two criteria from Bloom, Istrail & Meyer [2] and Bol & Groote [3] that can be imposed on solutions.

Definition 5 (*Supported model*). A transition relation T *agrees* with a TSS P if:

$T \models t \stackrel{a}{\longrightarrow} t' \Leftrightarrow$ there is a closed substitution instance $\frac{H}{t \stackrel{a}{\longrightarrow} t'}$ of a rule of P with $T \models H$.

T is a *model* of P if "$\Leftarrow$" holds; T is *supported* by P if "$\Rightarrow$" holds.

The first and most indisputable criterion imposed on a transition system T specified by a TSS P is that it is a model of P. This is called being *sound* for P in [2]. This criterion says that the rules of P, interpreted as implications in first-order or conditional logic, should evaluate to true statements about T. The second criterion, of being supported, says that T does not contain any transitions for which it has no plausible justification to contain them. In [2] being supported is called *witnessing*. Note that the universal transition relation on $\mathsf{T}(\Sigma)$ is a model of any TSS. It is however rarely the intended one, and the criterion of being supported is a good tool to rule it out. Next I check that Solution 1 satisfies both criteria.

Proposition 1 Let P be a positive TSS and T the set of transitions provable from P. Then T is a supported model of P. Moreover T is the least model of P.

Starting from Proposition 1 there are at least three ways to generalise Solution 1 to TSSs with negative premises. One can generalise either the concept of a proof, or the least model property, or the least supported model property of positive TSSs. Starting with the last two possibilities, observe that in general no least model and no least supported model exists. A counterexample is given by the TSS P_1 (given earlier), which has two minimal models, $\{c \xrightarrow{a} c\}$ and $\{c \xrightarrow{b} c\}$, both of which are supported.

Solution 2 (*Least*). A TSS is meaningful iff it has a least model (this being its specified transition relation).

Solution 3 (*Least supported*). A TSS is meaningful iff it has a least supported model.

These two solutions turn out to have incomparable domains. The TSS P_2 below has $\{c \xrightarrow{a} c\}$ as its least model, but has no supported models. On the other hand P_3 has two minimal models, namely $\{c \xrightarrow{b} c\}$ and $\{c \xrightarrow{a} c\}$, of which only the latter one is supported. This is its least supported model.

P_2 $\boxed{\dfrac{c \overset{a}{\not\longrightarrow}}{c \xrightarrow{a} c}}$ $\qquad$ P_3 $\boxed{\dfrac{c \overset{b}{\not\longrightarrow}}{c \xrightarrow{a} c}}$

Obviously Solution 1 is extended by both solutions above. However, Solutions 2 and 3 turn out to be inconsistent with each other. P_4 has both a least model and a least supported model, but they are not the same.

P_4 $\boxed{\dfrac{c \overset{a}{\not\longrightarrow}}{c \xrightarrow{a} c} \quad \dfrac{c \xrightarrow{b} c}{c \xrightarrow{a} c} \quad \dfrac{c \xrightarrow{b} c}{c \xrightarrow{b} c}}$ $\qquad$ P_5 $\boxed{\dfrac{c \xrightarrow{a} c}{c \xrightarrow{a} c}}$

Solution 2 is not very productive, because if fails to assign a meaning to the perfectly reasonable TSS P_3. Moreover, it can be criticised for yielding unsupported transition systems, as in the case of P_2. However, in P_4 the least model $\{c \xrightarrow{a} c\}$ appears to be a better choice than the least supported model $\{c \xrightarrow{a} c,\ c \xrightarrow{b} c\}$, as the 'support' for transition $c \xrightarrow{b} c$ is not overwhelming. Thus, to my taste, Solution 3 is somewhat unnatural.

In Bloom, Istrail & Meyer [2] the following solution is applied.

Solution 4 (*Unique supported*). A TSS is meaningful iff it has a unique supported model.

The positive TSS P_5 above has two supported models, $\emptyset$ and $\{c \xrightarrow{a} c\}$, and hence shows that Solution 4 does not extend Solution 1.

Although for the kind of TSSs considered in [2] (the *GSOS rule systems*) this solution coincides with all acceptable solutions mentioned in this paper, in general it suffers from the same drawback as Solution 3. The least supported model of P_4 is even the unique supported model of this TSS. My conclusion is that the criterion of being supported is too weak to be of any use in this context.

This conclusion was also reached by FAGES [5] in the setting of logic programming, who proposes to strengthen this criterion. Being supported can be rephrased as saying that a transition may only be present if there is a nonempty proof of its presence, starting from transitions that are also present. However, these premises in the proof may include the transition under derivation, thereby allowing for loops, as in the case of P_4. Now the idea behind a *well-supported model* is that the *absence* of a transition may be assumed a priory, as long as this assumption is consistent, but the *presence* of a transition needs to be proven without assuming the presence of (other) transitions. Thus a transition may only be present if it admits a valid proof, starting from negative literals only.

Definition 6 (*Well-supported*).[2] A transition relation T is *well-supported* by a TSS P if:

$$T \models t \xrightarrow{a} t' \quad \Leftrightarrow \quad \begin{array}{l}\text{there is a closed proof } p, \text{ with } T \models p, \text{ of a} \\ \text{transition rule } \frac{N}{t \xrightarrow{a} t'} \text{ without positive antecedents.}\end{array}$$

Note that "$\Leftarrow$" is trivial, and a well-supported transition relation is surely supported.

My concept of well-supportedness can easily be seen to coincide with the one of FAGES [5]. It is closely related to the earlier concept of *stability*, developed by GELFOND & LIFSCHITZ [7] in logic programming, and adapted for TSSs by BOL & GROOTE [3].

Definition 7 (*Stable transition relation*). A transition relation T is *stable* for a TSS P if:

$$T \models t \xrightarrow{a} t' \quad \Leftrightarrow \quad \begin{array}{l}\text{there is a closed transition rule } \frac{N}{t \xrightarrow{a} t'} \text{ without positive antecedents} \\ \text{with } P \vdash \frac{N}{t \xrightarrow{a} t'} \text{ and } T \models N.\end{array}$$

Proposition 2 The concept of stability of Definition 7 coincides with that from [3]. Moreover, T is stable for P iff it is a well-supported model of P.

The following two solutions are adaptations of Solutions 3 and 4, were the requirement of being supported has been replaced by that of being well-supported. The second is taken from [3].

Solution 5 (*Stable*). A TSS is meaningful iff it has a least stable transition relation.

Solution 5 (*Stable*). A TSS is meaningful iff it has a unique stable transition relation.

The particular numbering of these two solutions is justified by the following.

Proposition 3 A TSS has a least stable transition relation iff it has a unique stable transition relation. Moreover, Solution 5 (*stable*) extends Solution 1 (*positive*) and is consistent with Solution 2 (*least*) and 3 (*least supported*).

[2] The full version of this paper, which appeared as Stanford report STAN-CS-TN-95-16, contained an incorrect definition of well-supportedness (but leading to the same notion of a well-supported model). As observed by Jan Rutten, Proposition 3 in that version, stating that well-supported transition relations are supported, was false. The mistake had no other bad consequences.

Solution 5 improves Solutions 3 and 4 by rejecting the TSS P_4 as meaningless. It also improves Solution 2 by rejecting the TSS P_2 (whose least model was not supported). Surprisingly however, Solution 5 not only differs from the earlier solutions by being more fastidious; it also provides meaning to perfectly acceptable TSSs that were left meaningless by Solutions 2, 3 and 4.

$$P_6 \quad \boxed{\frac{c \overset{a}{\not\longrightarrow}}{c \overset{b}{\longrightarrow} c} \qquad \frac{c \overset{a}{\longrightarrow} c}{c \overset{a}{\longrightarrow} c}}$$

An example is the TSS P_6. There is clearly no satisfying way to obtain $c \overset{a}{\longrightarrow} c$. Hence $c \overset{a}{\not\longrightarrow}$ and consequently $c \overset{b}{\longrightarrow} c$. $\{c \overset{b}{\longrightarrow} c\}$ is indeed the unique stable transition relation of this TSS. However, P_6 has two minimal models, both of which are supported, namely $\{c \overset{b}{\longrightarrow} c\}$ and $\{c \overset{a}{\longrightarrow} c\}$.

It is interesting to see how the various solutions deal with *circular* rules, such as $\frac{c \overset{a}{\longrightarrow} c}{c \overset{a}{\longrightarrow} c}$, and rules like $\frac{c \overset{a}{\not\longrightarrow}}{c \overset{a}{\longrightarrow} c}$. The support-based solutions (3 and 4) may use a circular rule to obtain a transition that would be unsupported otherwise (Example P_4). This is my main argument to reject these solutions. In addition they may (or may not) reject TSSs as meaningless because of the presence of such a rule (Example P_6). On the other hand, Solutions 2 and 5 politely ignore these rules. To my taste, there are two acceptable attitudes towards circular rules: to ignore them completely (as done by Solutions 1, 2 and 5), or to reject any TSS with such a rule for being ambiguous, unless there is independent evidence for a transition $c \overset{a}{\longrightarrow} c$. A strong argument in favor of the first approach is the existence of useful rules of which only certain substitution instances are circular (cf. [3]). A solution that caters to the second option will be proposed in the next section.

Solution 2 can treat a rule $\frac{c \overset{a}{\not\longrightarrow}}{c \overset{a}{\longrightarrow} c}$ as equivalent to $c \overset{a}{\longrightarrow} c$ (namely if there are no other closed terms than c, cf. P_2), which gives rise to unsupported transition relations. Solutions 3, 4 and 5 do not go so far, but use such a rule to choose between two otherwise equally attractive transition relations. This is illustrated by the TSS P_7, which determines the transition system $\{c \overset{a}{\longrightarrow} c\}$ according to each of the solutions 2–5.

$$P_7 \quad \boxed{\frac{c \overset{a}{\not\longrightarrow}}{c \overset{b}{\longrightarrow} c} \qquad \frac{c \overset{b}{\not\longrightarrow}}{c \overset{a}{\longrightarrow} c} \qquad \frac{c \overset{a}{\not\longrightarrow}}{c \overset{a}{\longrightarrow} c}} \qquad\qquad P_8 \quad \boxed{c \overset{a}{\longrightarrow} c \qquad \frac{c \overset{a}{\not\longrightarrow}}{c \overset{a}{\longrightarrow} c}}$$

Ignoring rules like $\frac{c \overset{a}{\not\longrightarrow}}{c \overset{a}{\longrightarrow} c}$ is unacceptable, as this would yield unsound transition relations (non-models). But it could be argued that any TSS with such a rule should be rejected as meaningless, unless there is independent evidence for a transition $c \overset{a}{\longrightarrow} t$, as in P_8. This would rule out P_7. Solutions that cater to this taste will be proposed next.

3 Proof theoretic solutions

In this section I will propose solutions based on a generalisation of the concept of a proof. Note that in a proof two kinds of steps are allowed, numbered 1 and 2 in Definition 3. Step 1 just allows hypotheses to enter, in case one wants to prove a transition rule. This step can not be used when merely proving transitions. The essence of the notion is step 2. This step reflects the postulate that the desired transition relation must be a model of the given TSS. As a consequence those and only those transitions are provable that appear in any model. When generalising the notion of a proof to derive negative literals

it makes sense to import more postulates about the desired transition relation. Note that a model T of a TSS P is supported iff

$T \not\models t \xrightarrow{a} t' \;\Leftarrow\;$ for each closed substitution instance $\frac{H}{t \xrightarrow{a} t'}$ of a rule of P one has $T \not\models H$.

and well-supported iff

$T \not\models t \xrightarrow{a} t' \;\Leftarrow\;$ for each set of negative closed literals N with $P \vdash \frac{N}{t \xrightarrow{a} t'}$, one has $T \not\models N$.

Therefore I propose the following two concepts of provability.

Definition 8 (*Supported proof*). A *supported proof* of a closed literal α from a TSS P is like a closed (positive) proof (see Definition 3), but admitting steps of the form

3. β is negative and for each closed substitution instance of a rule of P whose conclusion denies β, a literal in K denies one of its antecedents.

α is *s-provable*, notation $P \vdash_s \alpha$, if a supported proof of α from P exists.
A literal is *s-refutable* if a denying literal is s-provable.

Definition 9 (*Well-supported proof*). A *well-supported proof* of a closed literal α from a TSS P is like a closed (positive) proof (Definition 3), but admitting steps of the form

3. β is negative and for every set N of negative closed literals such that $P \vdash \frac{N}{\gamma}$ for γ a closed literal denying β, a literal in K denies one in N.

α is *ws-provable*, notation $P \vdash_{ws} \alpha$, if a well-supported proof of α from P exists.
A literal is *ws-refutable* if a denying literal is ws-provable.

Note that these proof-steps establish the validity of β when K is the set of literals established earlier. In case K and N are sets of closed literals and a literal in K denies one in N, one has $T \not\models N$ for any transition relation T with $T \models K$. Thus step 3 from Definition 9 allows one to infer $t \xrightarrow{a}\!\!\!\!\!/\;\; t'$ whenever it is manifestly impossible to infer $t \xrightarrow{a} t'$, or $t \xrightarrow{a}\!\!\!\!\!/\;\;$ whenever for any term t' it is manifestly impossible to infer $t \xrightarrow{a} t'$. This practice is sometimes referred to as *negation as failure* [4]. Definition 8 allows such an inference only if the impossibility to derive $t \xrightarrow{a} t'$ can be detected by examining all possible proofs that consist of one step only. This corresponds with the notion of *negation as finite failure* of CLARK [4]. The extension of these notions (especially $\vdash_{ws}$) from closed to open literals α, or to transition rules $\frac{H}{\alpha}$, is somewhat problematic, and not needed in this paper. The following may shed more light on $\vdash_s$ and $\vdash_{ws}$. From here onwards, statements hold with or without the text enclosed in square brackets.

Proposition 4 Let P be a TSS. Then $P \vdash_s t \xrightarrow{a}\!\!\!\!\!/\;\; [t']$ iff every closed substitution instance $\frac{H}{t \xrightarrow{a} t'}$ of a rule of P has an s-refutable antecedent. Moreover $P \vdash_{ws} t \xrightarrow{a}\!\!\!\!\!/\;\; [t']$ iff every set N of closed negative literals with $P \vdash \frac{N}{t \xrightarrow{a} t'}$ contains an ws-refutable literal.

Proposition 5 For P a TSS and α a closed literal $\quad P \vdash \alpha \;\Rightarrow\; P \vdash_s \alpha \;\Rightarrow\; P \vdash_{ws} \alpha$.

Definition 10 For P a TSS and α a closed literal, write $P \models_s \alpha$ if $T \models \alpha$ for any supported model T of P and $P \models_{ws} \alpha$ if $T \models \alpha$ for any well-supported model T of P.
A notion $\vdash_x$ is called

- *consistent* if there is no TSS deriving two literals that deny each other.
- *sound* w.r.t. $\models_x$ if for any TSS P and closed literal α, $P \vdash_x \alpha \Rightarrow P \models_x \alpha$.

- *complete* w.r.t. $\models_x$ if for any TSS P and closed literal α, $P \vdash_x \alpha \Leftarrow P \models_x \alpha$.

Proposition 6 $\vdash_{[w]s}$ is consistent, $\vdash_{ws}$ is sound w.r.t. $\models_{ws}$ and $\vdash_s$ is sound w.r.t. $\models_s$.

However, $\vdash_s$ and $\vdash_{ws}$ are not complete w.r.t. $\models_{[w]s}$. A trivial counterexample concerns TSSs like P_2 that have no [well-]supported models. $P_2 \models_{[w]s} \alpha$ for any α, which by Proposition 6 (consistency) is not the case for $\vdash_{[w]s}$. A more interesting counterexample concerns the TSS P_7, which has only one [well-]supported model, namely $\{c \xrightarrow{a} c\}$. In spite of this, $P_7 \not\vdash_{[w]s} c \xrightarrow{a} c$ and $P_7 \not\vdash_{[w]s} c \not\xrightarrow{b}$.

As argued in the previous section, there is a point in excluding P_7 from the meaningful TSSs, since there is insufficient evidence for the transition $c \xrightarrow{a} c$. Here the incompleteness of $\vdash_{[w]s}$ w.r.t. $\models_{[w]s}$ comes as a blessing rather than a shortcoming.

3.1 Solutions based on completeness

I will now introduce the concept of a *complete* TSS: one in which any transition is either provable or refutable. Just as in the theory of logic there is a distinction between the completeness of a logic (e.g. first-order) and the completeness of a particular theory (e.g. arithmetic), here the completeness of a TSS is something different from the completeness of a proof-method $\vdash_x$. Let x be s or ws.

Definition 11 (*Completeness of a TSS*). A TSS P is *x-complete* if for any transition $t \xrightarrow{a} t'$ either $P \vdash_x t \xrightarrow{a} t'$ or $P \vdash_x t \not\xrightarrow{a} t'$. By 'complete' I will mean 'ws-complete'.

Solution 6 (*Complete with support*). A TSS is meaningful iff it is s-complete. The associated transition relation consists of the s-provable transitions.

Solution 7 (*Complete*). A TSS is meaningful iff it is (ws-)complete. The associated transition relation consists of the ws-provable transitions.

The TSS P_6 is complete, but not complete with support. P_3 is even complete with support. In Bol & Groote [3] a method called *reduction* for associating a transition relation with a TSS was proposed, inspired by the *well-founded models* of Van Gelder, Ross & Schlipf [6] in logic programming. In the full version of this paper I show that this solution coincides with Solution 7. Solution 7 can therefore be regarded as a proof theoretical characterisation of the ideas from [6, 3]. Solution 6 may be new.

Proposition 7 The set of $[w]s$-provable transitions of any TSS is well-supported. Moreover, the set of $[w]s$-provable transitions of a $[w]s$-complete TSS P is a model of P.

Proposition 8 Solution 6 [7] is strictly extended by Solution 4 [5].

Proof: Suppose P is $[w]s$-complete. By Proposition 7 the $[w]s$-provable transitions constitute a [well-]supported model of P, and by Proposition 6 (soundness) this is the only such model. Strictness follows from the TSS P_7, which has an unique [well-]supported model, but is left meaningless by Solutions 6 and 7. □

At the end of Section 2 I recommended two acceptable attitudes towards rules like $\frac{c \xrightarrow{a} c}{c \xrightarrow{a} c}$. In the full paper I show that Solution 7 ignores such rules completely (which is one option), whereas Solution 6 rejects a TSS with such a rule, unless there is independent evidence for a transition $c \xrightarrow{a} c$ (the other option). Moreover, Solutions 6 and 7 reject any TSS containing rules like $\frac{c \not\xrightarrow{a}}{c \xrightarrow{a} c}$, unless there is independent evidence for a transition $c \xrightarrow{a} t$. As shown by counterexample P_7 all model theoretic solutions fail this test.

3.2 Attaching meaning to *all* transition system specifications

In this section I will associate a transition relation to arbitrary TSSs. As illustruted by P_1 and P_2, such a transition relation can not always be a supported model. I will insist on soundness (being a model), and thus have to give up support.

Let me first decide what to do with P_1. Since the associated transition relation should be a model, it must contain either $c \xrightarrow{a} c$ or $c \xrightarrow{b} c$. For reasons of symmetry I cannot choose between these transitions, so the only way out is to include both. There is no reason to include any more transitions. Hence the transition system associated to P_1 should be $\{c \xrightarrow{a} c,\ c \xrightarrow{b} c\}$.

In the full paper I reject a model theoretic solution that gives this result. Among the proof theoretic solutions the best I could find was

Solution 8 (*Irrefutable*). Any TSS is meaningful. The associated transition relation consists of the ws-irrefutable transitions.

This solution is inspired by the following proposition.

Proposition 9 The set of x-irrefutable transitions of any TSS consititutes a model.

In the case of P_1 Solution 8 yields the desired result $\{c \xrightarrow{a} c,\ c \xrightarrow{b} c\}$ and likewise P_2, P_3 and P_4 yield $\{c \xrightarrow{a} c\}$. The transition relation of P_7 is the same as the one of P_1. This indicates that Solution 8 is inconsistent with Solutions 2–5. I don't consider this to be a problem, as the model theoretic allocation of a transition relation to P_7 was not very convincing.

4 Compositionality

In concurrency theory it is common practice to group together representations of concurrent systems in equivalence classes. As system representations often closed terms over some signature are considered. The equivalence relation employed is then formulated in terms of the transition relation between closed terms obtained from a given TSS over that signature. All equivalence relations employed in concurrency have the properties that systems for which the reachable parts of the transition relation are isomorphic are equivalent, and that a system without outgoing transitions (a *deadlock*) cannot be equivalent to a system with an outgoing a-transition.

In order to allow modular reasoning it is important to use an equivalence relation that is a *congruence*. This means that the meaning (the associated equivalence class) of a closed term $f(t_1, ..., t_n)$ is completely determined by the meaning of the subterms $t_1, ..., t_n$. The most popular equivalence relation is *bisimulation equivalence*. In Bol & Groote [3] it was established that for complete TSSs whose rules satisfy a syntactic criterion (the *well-founded ntyft/ntyxt format*, developed earlier in [9, 8]), bisimulation equivalence is guaranteed to be a congruence, and so are many other equivalence relations. Moreover, a counterexample was given against the extension of this result to TSSs that are meaningful according to Solution 5 (stable). Of course the example concerned an incomplete TSS in well-founded ntyft/ntyxt format with a unique stable transition relation for which bisimulation is not a congruence. This TSS also has a unique supported model, and thus shows that the congruence theorem does not generalise to Solution 4 either. Here I show that also Solution 8—or any other proof theoretic solution giving

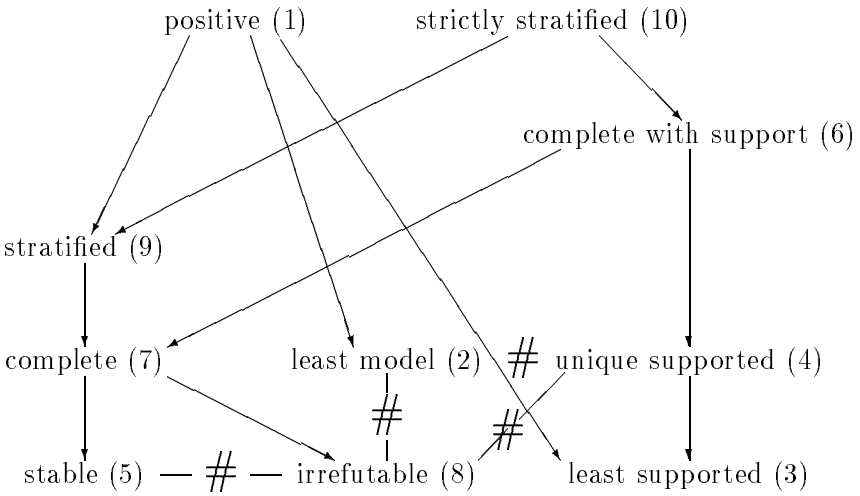
meaning to all TSSs for that matter—does not lend itself to such a generalisation, indicating that Solution 7 (complete) is the most general one for which this nice result holds. My counterexample concerns the following TSS S over a signature with constants c, d and e and a unary function f.

$$S \quad \boxed{c \xrightarrow{a} f(c) \quad \frac{x \xrightarrow{a} y \quad x \not\xrightarrow{a}}{f(x) \xrightarrow{a} c} \quad d \xrightarrow{a} e}$$

This TSS is surely in the well-founded ntyft/ntyxt format. The transitions $c \xrightarrow{a} f(c)$, $d \xrightarrow{a} e$ and $f(d) \xrightarrow{a} c$ are $[w]s$-provable, and with the exception of $f(c) \xrightarrow{a} c$, all other transitions are $[w]s$-refutable. As the validity of $f(c) \xrightarrow{a} c$ is left undetermined, the TSS is incomplete (has no meaning according to Solution 7). It also has no meaning under Solution 5 (stable). The proof theorectic approach offers only one choice, namely whether or not to include the transition $f(c) \xrightarrow{a} c$. Each of these possibilities yields a transition relation for which no equivalence relation used in concurrency theory is a congruence. Solution 8 (irrefutable) includes the transition $f(c) \xrightarrow{a} c$. Now c and $f(c)$ are equivalent (the reachable part of the transition relation from each of them is an a-loop), but $f(c)$ and $f(f(c))$ are inequivalent ($f(f(c))$ deadlocks). Taking only the provable transitions (instead of the irrefutable ones) would exclude the transition $f(c) \xrightarrow{a} c$. In that case c and d are equivalent, but $f(c)$ and $f(d)$ are not.

5 Conclusion

This paper dealt with the problem of associating a transition relation to a given TSS. The related problem of finding a good TSS to specify a given transition relation is left for future research. I presented 8 answers to the question of which transition system specifications are meaningful and which transition relations they specify. The relations between these 8 solutions, as well as the two solutions (9 and 10) proposed in [8], are indicated below. There $S_1 \longrightarrow S_2$ indicates that solution S_2 extends S_1, as defined in Section 1,



and $S_1 \# S_2$ indicates that S_1 and S_2 are inconsistent. By the definition of extension and consistency, $S_1 \longrightarrow S_2 \longrightarrow S_3$ implies $S_1 \longrightarrow S_3$ (*transitivity*) and $S_1 \# S_2 \longrightarrow S_3$ implies $S_1 \# S_3$ (*conflict heredity*). All extensions are strict and there are no more extensions

or inconsistencies than indicated in the figure (or derivable by transitivity and conflict heredity). Strictness, the absence of further extensions and the inconsistencies follow from the information collected in the table below, which indicates which of the TSSs P_1–P_8 given in this paper are meaningful according to each of the solutions. A '$-$' indicates that the TSS is meaningless, a '$+$' that it has the same meaning as given by Solution 8, and a '$*$' that is has a meaning different from the one given by Solution 8.

Solution		P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
1	positive	$-$	$-$	$-$	$-$	$+$	$-$	$-$	$-$
2	least	$-$	$+$	$-$	$+$	$+$	$-$	$*$	$+$
3	least supported	$-$	$-$	$+$	$*$	$+$	$-$	$*$	$+$
4	unique supported	$-$	$-$	$+$	$*$	$-$	$-$	$*$	$+$
5	stable	$-$	$-$	$+$	$-$	$+$	$+$	$*$	$+$
6	complete with support	$-$	$-$	$+$	$-$	$-$	$-$	$-$	$+$
7	complete	$-$	$-$	$+$	$-$	$+$	$+$	$-$	$+$
8	irrefutable	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
9	stratified	$-$	$-$	$+$	$-$	$+$	$+$	$-$	$-$
10	strictly stratified	$-$	$-$	$+$	$-$	$-$	$-$	$-$	$-$

Evaluation of the solutions

Solution 9 (stratified) stems from Przymusinski [11] and is the perhaps the best known solution in logic programming. A variant that only allows TSSs with a unique supported model is Solution 10 (strictly stratified), proposed by Groote [8].

Solution 1 is the classical interpretation of TSSs without negative premises, and Solutions 2 (least model) and 3 (least supported model) are two straightforward generalisations. Solution 4 (unique supported model) stems from Bloom, Istrail & Meyer [2], where it was used to ascertain that TSSs in their so-called GSOS format are meaningful (such TSSs have unique supported models). My counterexample P_4 shows that Solution 4 yields contraintuitive results and is therefore not suited to base such a conclusion on. Fortunately, TSSs in the GSOS format are even strictly stratified, which is one of the most restrictive criteria for meaningful TSSs considered. Solution 3 can be rejected on the same grounds as Solution 4 and Solution 2 is not very useful because it leaves most TSSs with negative premises meaningless (cf. P_3).

Solution 5 (unique stable transition relation) stems from Gelfond & Lifschitz [7] and is generally considered to be the most general acceptable solution available. Counterexample P_7 however suggests that this solution may yield debatable results, although to a lesser extent than Solutions 3 and 4.

Solution 7 (well-founded, positive after reduction, complete) is essentially due to Van Gelder, Ross & Schlipf [6]. It is the most general solution without undesirable properties. In Bol & Groote [3], where this solution has been adapted to TSSs, an example in the area of concurrency is given (the modelling of a priority operator in basic process algebra with abstraction) that can be handled with Solution 7, but not with Solution 9. This example can neither be handled by Solution 6, showing that the full generality of Solution 7 can be useful in applications.

My presentation of Solution 7 differs so much from the original one [6, 3] that I gave it a new name. It is based on a concept of provability incorporating the notion of *negation as failure* of Clark [4]. In the full paper I establish the correspondence between my version and the one from [6, 3]. There I also illustrate how my proof-theoretic characterisation can be useful in applications.

Solutions 6 (complete with support) and 8 (irrefutable) may be new. The first is based on a notion of provability that is somewhat simpler to apply, and only incorporates the notion of *negation as finite failure* [4]. Moreover, it only yields unique supported models, like Solution 10 (and 4). Solution 8 appears to be the best way to associate a transition relation to arbitrary TSSs. However, it has the disadvantage that it sometimes yields unstable transition relations, and even unsupported models. A good example from concurrency theory of an incomplete TSS is Basic Process Algebra with a priority operator, unguarded recursion and renaming, as defined in GROOTE [8]. This TSS has no supported models. Solution 8 does give a meaning to this TSS, but it appears rather arbitrary and not very useful. In particularly, recursively defined processes do no longer satisfy their defining equation, which makes algebraic reasoning virtually impossible. Also the absence of a congruence theorem as demonstrated in Section 4 is a bad property of this Solution. Hence, Solution 7 (complete) remains the most general completely acceptable answer to (1) and (2).

Acknowledgments This paper benefited greatly from the insightful comments of Roland Bol. Also my thanks to the audience of the PAM seminar for useful feedback. Finally, Jan Rutten is gratefully acknowledged for spotting a mistake in a previous version of this paper (see Footnote 2).

References

[1] K.R. APT & R. BOL (1994): *Logic programming and negation: A survey. Journal of Logic Programming* 19–20, pp. 9–71.

[2] B. BLOOM, S. ISTRAIL & A.R. MEYER (1995): *Bisimulation can't be traced.* JACM 42(1), pp. 232–268.

[3] R.N. BOL & J.F. GROOTE (1991): *The meaning of negative premises in transition system specifications (extended abstract).* In J. Leach Albert, B. Monien & M. Rodríguez, editors: *Proceedings* 18^{th} *ICALP,* Madrid, LNCS 510, Springer-Verlag, pp. 481–494. Full version to appear in JACM.

[4] K.L. CLARK (1978): *Negation as failure.* In H. Gallaire & J. Minker, editors: *Logic and Databases*, Plenum Press, New York, pp. 293–322.

[5] F. FAGES (1991): *A new fixpoint semantics for general logic programs compared with the well-founded and the stable model semantics. New Generation Computing* 9(4), pp. 425-443.

[6] A. VAN GELDER, K. ROSS & J.S. SCHLIPF (1991): *The well-founded semantics for general logic programs*, JACM 38(3), pp. 620–650.

[7] M. GELFOND & V. LIFSCHITZ (1988): *The stable model semantics for logic programming.* In R. Kowalski & K. Bowen, editors: *Proceedings* 5^{th} *International Conference on Logic Programming*, MIT Press, Cambridge, USA, pp. 1070–1080.

[8] J.F. GROOTE (1993): *Transition system specifications with negative premises. Theoretical Computer Science* 118(2), pp. 263–299.

[9] J.F. GROOTE & F.W. VAANDRAGER (1992): *Structured operational semantics and bisimulation as a congruence. Information and Computation* 100(2), pp. 202–260.

[10] G.D. PLOTKIN (1981): *A structural approach to operational semantics.* Report DAIMI FN-19, Computer Science Department, Aarhus University.

[11] T.C. PRZYMUSINSKI (1988): *On the declarative semantics of deductive databases and logic programs.* In Jack Minker, editor: *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann Publishers, Inc., pp. 193–216.