Termination of Simply Typed Term Rewriting by Translation and Labelling*

Takahito Aoto[1] and Toshiyuki Yamada[2]

[1] Research Institute of Electrical Communication, Tohoku University, Japan
aoto@nue.riec.tohoku.ac.jp

[2] Faculty of Engineering, Mie University, Japan
toshi@cs.info.mie-u.ac.jp

Abstract. Simply typed term rewriting proposed by Yamada (RTA 2001) is a framework of term rewriting allowing higher-order functions. In contrast to the usual higher-order term rewriting frameworks, simply typed term rewriting dispenses with bound variables. This paper presents a method for proving termination of simply typed term rewriting systems (STTRSs, for short). We first give a translation of STTRSs into many-sorted first-order TRSs and show that termination problem of STTRSs is reduced to that of many-sorted first-order TRSs. Next, we introduce a labelling method which is applied to first-order TRSs obtained by the translation to facilitate termination proof of them; our labelling employs an extension of semantic labelling where terms are interpreted on a many-sorted algebra.

1 Introduction

Simply typed term rewriting proposed by Yamada [10] is a simple extension of first-order term rewriting. It can deal with higher-order functions. Equational specification using higher-order functions, like functional programs, are naturally expressed in this framework. In contrast to the usual higher-order term rewriting frameworks [5, 7, 9], simply typed term rewriting dispenses with bound variables. In this respect, simply typed term rewriting reflects limited higher-order features. On the other hand, simply typed term rewriting framework is succinct and theoretically much more easier to deal with.

This paper presents a method for proving termination of simply typed term rewriting systems (STTRSs, for short). Termination of STTRSs based on monotone interpretation has been investigated in [10]. We are concerned here with syntactic methods and propose techniques which are more suitable for automated termination proof.

We first give a translation of STTRSs into many-sorted first-order TRSs such that the termination of an STTRS is induced from that of the many-sorted first-order TRS obtained by this translation. Since a many-sorted first-order TRS is

* This work was partially supported by a grant from Japan Society for the Promotion of Science, No. 14780187.

terminating whenever the underlying first-order TRS obtained by omitting the sort information is terminating, one can apply any known methods for proving termination of ordinary TRSs to infer the termination of STTRSs based on our translation. Next, we introduce a labelling method which is applied to first-order TRSs obtained by the translation to facilitate termination proof of them. Our labelling method is based on semantic labelling proposed by Zantema [11]—in fact, it is a particular kind of the semantic labelling over fixed models and labels (determined by the signature of STTRSs). However, the original semantic labelling method for unsorted TRSs is insufficient for our purpose and thus our labelling employs an extension of semantic labelling where terms are interpreted on a many-sorted algebra.

The remainder of the paper is organized as follows: In Section 2, we review the basic notions and terminology of simply typed term rewriting. In Section 3, we give a translation of STTRSs into many-sorted first-order TRSs such that the termination problem of an STTRS is reduced to that of the corresponding many-sorted TRS. In Sections 4 and 5, we present a labelling method that facilitates termination proof of many-sorted TRSs obtained by our translation. We first introduce our labelling method in unsorted setting using the original semantic labelling, and examine why the labelling in this setting is not often useful (in Section 4). Then the labelling method is improved based on the many-sorted version of semantic labelling (in Section 5).

2 Preliminaries

In this section, we recall the basic notions and terminology of simply typed term rewriting, which were introduced in [10]. We assume the reader to be familiar with (first-order) term rewriting [3, 4, 8].

For a set B of *basic types*, the set of *simple types* is the smallest set $\mathrm{ST}(B)$ such that (1) $B \subseteq \mathrm{ST}(B)$, and (2) $\tau_1 \times \cdots \times \tau_n \to \tau_0 \in \mathrm{ST}(B)$ whenever $\tau_0, \tau_1, \ldots, \tau_n \in \mathrm{ST}(B)$. Note that our definition allows multiple basic types whereas the original one in [10] is based on a single basic type. When clear, simple type is abbreviated as *type*, and $\mathrm{ST}(B)$ is written as ST.

Each *constant* or *variable* is associated with its type; the sets of constants and variables of type τ are denoted by C^τ and V^τ, respectively. C and V stand for the sets of all constants and variables, respectively: $C = \bigcup_{\tau \in \mathrm{ST}} C^\tau$ and $V = \bigcup_{\tau \in \mathrm{ST}} V^\tau$. We assume that V^τ is countably infinite for any $\tau \in \mathrm{ST}$. The set $\mathrm{T_{ST}}(C, V)^\tau$ of *simply typed terms* of type τ over C, V is defined as follows: (1) $C^\tau \cup V^\tau \subseteq \mathrm{T_{ST}}(C, V)^\tau$, and (2) if $s \in \mathrm{T_{ST}}(C, V)^{\tau_1 \times \cdots \times \tau_n \to \tau}$ and $t_i \in \mathrm{T_{ST}}(C, V)^{\tau_i}$ for all $i \in \{1, \ldots, n\}$ then $(s\, t_1 \cdots t_n) \in \mathrm{T_{ST}}(C, V)^\tau$. The outermost parenthesis of simply typed terms may be omitted when no confusion arises. A simply typed term s has type τ (denoted by s^τ) when $s \in \mathrm{T_{ST}}(C, V)^\tau$. It is clear that each simply typed term has a unique type; thus τ is also referred to as the type of s. The set $\mathrm{T_{ST}}(C, V)$ of all simply typed terms is $\bigcup_{\tau \in \mathrm{ST}} \mathrm{T_{ST}}(C, V)^\tau$; when clear, $\mathrm{T_{ST}}(C, V)$ is abbreviated as $\mathrm{T_{ST}}$. The *head symbol* of a simply typed term is defined as follows: (1) $\mathrm{head}(t) = t$ if $t \in C \cup V$, and (2) $\mathrm{head}((s\, t_1 \cdots t_n)) = \mathrm{head}(s)$. The set of variables occurring in a term t is written as $\mathrm{V}(t)$.

A pair $\langle l, r\rangle$ of simply typed terms is a *simply typed rewrite rule* when (1) l and r have the same type, (2) $\mathrm{head}(l) \in C$, and (3) $\mathrm{V}(r) \subseteq \mathrm{V}(l)$. A simply typed rewrite rule $\langle l, r\rangle$ will be often written as $l \to r$. A triple $\langle B, C, R\rangle$ consisting of a set B of basic types, a set C of constants, and a set R of simply typed rewrite rules is called a simply typed term rewriting system (STTRS, for short).

A *substitution* is a mapping $\sigma : V \to \mathrm{T}_{\mathrm{ST}}(C, V)$ that satisfies the following conditions: (1) $\mathrm{Dom}(\sigma) = \{x \mid \sigma(x) \neq x\}$ is finite; and (2) for each $x \in \mathrm{Dom}(\sigma)$, x and $\sigma(x)$ have the same type. The homomorphic extension of σ to $\mathrm{T}_{\mathrm{ST}}(C, V)$ is also denoted by σ. As usual, $\sigma(t)$ is written as $t\sigma$. The *rewrite relation* $\to_{\mathcal{R}}$ induced by a simply typed term rewriting system $\mathcal{R} = \langle B, C, R\rangle$ is the smallest relation over $\mathrm{T}_{\mathrm{ST}}(C, V)$ satisfying the following conditions: (1) $l\sigma \to_{\mathcal{R}} r\sigma$ for all $l \to r \in R$ and for all substitutions σ, and (2) if $s \to_{\mathcal{R}} t$ then $(s_0 \cdots s \cdots s_n) \to_{\mathcal{R}} (s_0 \cdots t \cdots s_n)$ for all $s_0, \ldots, s_n$. An STTRS $\mathcal{R}$ is *terminating* if there is no infinite reduction sequences $s_0 \to_{\mathcal{R}} s_1 \to_{\mathcal{R}} \cdots$.

Example 1 (simply typed term rewriting). Let $\mathcal{R} = \langle B, C, R\,\rangle$ be an STTRS where $B = \{\, \mathit{Nat}, \mathit{List}\,\}$, $C = \{\, \mathsf{0}^{\mathit{Nat}}, \mathsf{s}^{\mathit{Nat}\to\mathit{Nat}}, [\,]^{\mathit{List}}, :^{\,\mathit{Nat}\times\mathit{List}\to\mathit{List}}, \mathsf{map}^{(\mathit{Nat}\to\mathit{Nat})\times\mathit{List}\to\mathit{List}}, \circ^{(\mathit{Nat}\to\mathit{Nat})\times(\mathit{Nat}\to\mathit{Nat})\to(\mathit{Nat}\to\mathit{Nat})}, \mathsf{twice}^{(\mathit{Nat}\to\mathit{Nat})\to(\mathit{Nat}\to\mathit{Nat})}\,\}$, and

$$R = \begin{cases} \mathsf{map}\ F\ [\,] & \to\ [\,] \\ \mathsf{map}\ F\ (x : xs) & \to\ (F\ x) : (\mathsf{map}\ F\ xs) \\ (F \circ G)\ x & \to\ F\ (G\ x) \\ \mathsf{twice}\ F & \to\ F \circ F \end{cases}$$

Here is a rewrite sequence of $\mathcal{R}$:

$$\begin{array}{rll} \mathsf{map}\ (\mathsf{twice}\ \mathsf{s})\ (\mathsf{0} : [\,]) & \to_{\mathcal{R}} & \mathsf{map}\ (\mathsf{s} \circ \mathsf{s})\ (\mathsf{0} : [\,]) \\ & \to_{\mathcal{R}} & ((\mathsf{s} \circ \mathsf{s})\ \mathsf{0}) : (\mathsf{map}\ (\mathsf{s} \circ \mathsf{s})\ [\,]) \\ & \to_{\mathcal{R}} & (\mathsf{s}\ (\mathsf{s}\ \mathsf{0})) : (\mathsf{map}\ (\mathsf{s} \circ \mathsf{s})\ [\,]) \\ & \to_{\mathcal{R}} & (\mathsf{s}\ (\mathsf{s}\ \mathsf{0})) : [\,]. \end{array}$$

3 Translating STTRSs to Many-Sorted TRSs

The framework of STTRSs can deal with higher-order variables and thus it is not within the framework of (usual) first-order term rewriting. However, this does not necessarily imply that STTRSs can not be seen as TRSs via suitable encoding. In this section, we will see that STTRSs can be encoded into many-sorted TRSs.

First, we define the notion of arity, which will be used to reflect the type information of the simply typed terms in our encoding.

Definition 1 (arity). *The set of* arities *is the smallest set* $\mathrm{Ar}(B)$ *such that (1)* $B \subseteq \mathrm{Ar}(B)$, *and (2) if* $n \geq 1$ *and* $a_0, a_1, \ldots, a_n \in \mathrm{Ar}(B)$ *then* $\langle a_1 \cdots a_n\, a_0\rangle \in \mathrm{Ar}(B)$. *The mapping* $\mathrm{ar} : \mathrm{ST}(B) \to \mathrm{Ar}(B)$ *is defined as follows:*

$$\mathrm{ar}(\tau) = \begin{cases} \tau & \text{if } \tau \in B, \\ \langle \mathrm{ar}(\tau_1) \cdots \mathrm{ar}(\tau_n)\ \mathrm{ar}(\tau_0)\rangle & \text{if } \tau = \tau_1 \times \cdots \times \tau_n \to \tau_0. \end{cases}$$

Arities can denote simple types concisely. For example, $\mathrm{ar}((Nat \to Nat) \times List \to List) = \langle\langle Nat\ Nat\rangle List\ List\rangle$. We also employ the following conventions for simplicity:

1. $Nat, List$, etc. are abbreviated as N, L, etc. when no confusion arises.
2. $\langle a_1 \cdots a_n\rangle$ is abbreviated as a_1^n when $a_1 = \cdots = a_n$. (Note that this implies that $(N^2)^2$ differs from N^4; the former denotes $\langle\langle NN\rangle\langle NN\rangle\rangle$ while the latter is $\langle NNNN\rangle$.)
3. Outer most brackets of arities are omitted. (Thus, for example, $\langle\langle NL\rangle LL\rangle$ is written as $\langle NL\rangle LL$.)

These conventions are illustrated in the following table, which contains some examples of arities.

symbol $c \in C$	simple type τ	arity $\mathrm{ar}(\tau)$
$\mathsf{0}$	Nat	N
s	$Nat \to Nat$	N^2
$:$	$Nat \times List \to List$	NLL
map	$(Nat \to Nat) \times List \to List$	N^2LL
twice	$(Nat \to Nat) \to (Nat \to Nat)$	$(N^2)^2$

We next give a translation from STTRSs to many-sorted TRSs. A many-sorted TRS is specified by a triple $\langle S, F, R\rangle$ where S is a set of sorts, F a set of S-sorted function symbols, and R a set of rewrite rules.

Definition 2 (translation). *We define a translation Θ from simply typed signature (terms, STTRSs) to many-sorted signature (terms, many-sorted TRSs, respectively).*

1. $\Theta(C) = C \cup \{\ @_a \mid a \in \mathrm{Ar}(B) \setminus B\ \}$
   where each function symbol in $\Theta(C)$ is associated with its sort as follows:
   $\mathrm{sort}(f) = \begin{cases} \mathrm{ar}(\tau) & \textit{if } f \in C^\tau \\ \langle a_1 \cdots a_n a_0\rangle \times a_1 \times \cdots \times a_n \to a_0 & \textit{if } f = @_{\langle a_1 \cdots a_n a_0\rangle} \end{cases}$
2. $\Theta(t) = \begin{cases} t & \textit{if } t \in C \cup V \\ @_{\mathrm{ar}(\tau)}(\Theta(s), \Theta(t_1), \ldots, \Theta(t_n)) & \textit{if } t = (s^\tau\ t_1 \cdots t_n) \end{cases}$
3. $\Theta(R) = \{\ \Theta(l) \to \Theta(r) \mid l \to r \in R\ \}$
4. $\Theta(\langle B, C, R\rangle) = \langle \mathrm{Ar}(B), \Theta(C), \Theta(R)\rangle$

The following table shows examples of symbols and their sorts in a translated signature.

symbol $f \in \Theta(C)$	sort $\mathrm{sort}(f)$
$\mathsf{0}$	N
s	N^2
map	N^2LL
$@_{N^2}$	$N^2 \times N \to N$
$@_{NLL}$	$NLL \times N \times L \to L$

Example 2 (translation). Let $\mathcal{R}$ be an STTRS given in Example 1. Given below is the rewrite rules of $\Theta(\mathcal{R})$.

$$\begin{cases} @_{N^2LL}(\mathsf{map}, F, [\,]) & \rightarrow \ [\,] \\ @_{N^2LL}(\mathsf{map}, F, @_{NLL}(:, x, xs)) & \rightarrow \ @_{NLL}(:, @_{N^2}(F, x), @_{N^2LL}(\mathsf{map}, F, xs)) \\ @_{N^2}(@_{(N^2)^3}(\circ, F, G), x) & \rightarrow \ @_{N^2}(F, @_{N^2}(G, x)) \\ @_{(N^2)^2}(\mathsf{twice}, F) & \rightarrow \ @_{(N^2)^3}(\circ, F, F) \end{cases}$$

Theorem 1 (completeness of translation). *Let $\mathcal{R}$ be an STTRS. Then $\Theta(\mathcal{R})$ is terminating if and only if $\mathcal{R}$ is terminating.*

Proof. Easy. □

Thus, the termination problem of STTRSs is reduced to the termination problem of many-sorted TRSs. Clearly, many-sorted TRS is terminating when its underlying unsorted TRS obtained by eliminating its sort information is terminating. Therefore, for proving termination of an STTRS, any existing proof techniques can be applied to the TRS obtained by the translation Θ.

Example 3 (termination of STTRS by translation). Let $\mathcal{R}$ be the STTRS in Example 1. Termination of $\Theta(\mathcal{R})$ is shown, for example, by the lexicographic path order (LPO) [6] and thus, $\mathcal{R}$ is terminating.

4 Primary Symbol Labelling on Single Sort

In simply typed term rewriting, the usual expression $f(s_1, \ldots, s_n)$ of terms is changed to $(f\ s_1\ \cdots\ s_n)$ so that function variables does not appear as a node of tree structure of terms like $F(s_1, \ldots, s_n)$ but as a leaf like $(F\ s_1\ \cdots\ s_n)$. Similarly, for TRSs of the form $\Theta(\mathcal{R})$, the terms are of the form $@_a(f, s_1, \ldots, s_n)$. Then, contrasted to the usual first-order term formulation, when one tries to compare terms using path-orders like RPO, LPO, etc., each leading symbol f is no longer compared before its arguments $s_1, \ldots, s_n$, and only application symbols with type information are compared before their arguments. Since the type information of a term is more common information compared to the function symbol, path-orders are not effectively applied to TRSs obtained by the translation.

Example 4 (motivation for labelling). Let $B = \{\ Nat\ \}$, $C = \{\ \mathsf{0}^{Nat}, \mathsf{s}^{Nat \rightarrow Nat}, +^{Nat \rightarrow (Nat \rightarrow Nat)}, *^{Nat \rightarrow (Nat\ \rightarrow Nat)}\ \}$, and

$$R = \begin{cases} (+\ \mathsf{0})\ y & \rightarrow \ y \\ (+\ (\mathsf{s}\ x))\ y & \rightarrow \ \mathsf{s}\ ((+\ x)\ y) \\ (*\ \mathsf{0})\ y & \rightarrow \ \mathsf{0} \\ (*\ (\mathsf{s}\ x))\ y & \rightarrow \ +\ ((*\ x)\ y)\ y \end{cases}$$

Then $\Theta(\mathcal{R})$ has the following rewrite rules:

$$\begin{cases} @_{N^2}(@_{NN^2}(+,\mathsf{0}),y) & \to \quad y \\ @_{N^2}(@_{NN^2}(+,@_{N^2}(\mathsf{s},x)),y) & \to \quad @_{N^2}(\mathsf{s},@_{N^2}(@_{NN^2}(+,x),y)) \\ @_{N^2}(@_{NN^2}(*,\mathsf{0}),y) & \to \quad \mathsf{0} \\ @_{N^2}(@_{NN^2}(*,@_{N^2}(\mathsf{s},x)),y) & \to \quad @_{N^2}(@_{NN^2}(+,@_{N^2}(@_{NN^2}(*,x),y)),y) \end{cases}$$

The fourth rule can not be made decreasing by LPO at any precedence—this is because $+$ and $*$ have the same type (i.e. both lhs and rhs of the rule have the same root symbol $@_{N^2}$).

This observation leads us to the idea of labelling application symbols not only by types but also head symbols. That is, we label each application symbol by the leftmost non-application symbol in its first argument. For example, the fourth rule of the TRS above becomes

$$\begin{aligned} &@_{N^2*}(@_{NN^2*}(*,@_{N^2\mathsf{s}}(\mathsf{s},x)),y) \\ &\qquad\qquad \to @_{N^2+}(@_{NN^2+}(+,@_{N^2*}(@_{NN^2*}(*,x),y)),y) \end{aligned}$$

and the root symbols of lhs and rhs of the rule get distinguished. For this purpose, semantic labelling [11] seems to be helpful.

In this section, we define this particular labelling, called primary symbol labelling, for first-order TRSs. Our labelling method is obtained from semantic labelling by fixing ordered algebra and a labelling specification. It turns out that the labelling method obtained from the original semantic labelling is not useful in many cases. In the end of the section, we will exhibit by examples why this naive approach is unsatisfactory. The labelling method in this section will be modified in the next section.

In semantic labelling, algebra operations are used to determine a label of a function symbol. Since our labelling is based on the quasi-model version of semantic labelling, the carrier of an algebra for labelling need to be partially ordered. Let Σ be a first-order signature. A *weakly monotone Σ-algebra* $\mathcal{A} = \langle A, \geq_{\mathcal{A}}, \{f_{\mathcal{A}}\}_{f\in\Sigma}\rangle$ consists of a non-empty carrier set A, a partial order $\geq_{\mathcal{A}}$ on A, and a family $\{f_{\mathcal{A}}\}_{f\in\Sigma}$ of interpretations. Each $f \in \Sigma$ takes a fixed number of arguments and the set of all function symbols in Σ taking n arguments is denoted by Σ_n. Each constant $c \in \Sigma_0$ is interpreted as an element $c_{\mathcal{A}} \in A$ and each $f \in \Sigma_n$ $(n \geq 1)$ is interpreted as an algebra operation $f_{\mathcal{A}} : A^n \to A$. All operations need to be weakly monotone, i.e., if $a_i \geq_{\mathcal{A}} b_i$ for $i = 1, \ldots, n$ then $f_{\mathcal{A}}(a_1, \ldots, a_n) \geq_{\mathcal{A}} f_{\mathcal{A}}(b_1, \ldots, b_n)$. The result of evaluating a term t in $\mathcal{A}$ is denoted by $[\![t]\!]_{\mathcal{A}}$. The subscript $\mathcal{A}$ will be omitted when no confusion occurs.

We define a particular ordered algebra suitable for our labelling.

Definition 3 (algebra for primary symbol labelling). *Let Σ be a first-order signature and $\geq$ a partial order on the set Σ_0 of all constant symbols. The Σ-algebra* $\mathbf{Pri}$ *is defined as $\langle \Sigma_0, \geq, \{f_{\mathbf{Pri}}\}_{f\in\Sigma}\rangle$ where the interpretation of a function symbol $f \in \Sigma_n$ is defined as follows:*

$$f_{\mathbf{Pri}}(a_1, \ldots, a_n) = \begin{cases} f & \textit{if } n = 0, \\ a_1 & \textit{if } n \geq 1. \end{cases}$$

Observe that $f_{\mathbf{Pri}}$ is weakly monotone for any partial order $\geq$.

Example 5 (Σ-algebra **Pri***).* Let Σ be the signature of $\Theta(\mathcal{R})$ in Example 4, namely, $\Sigma = \{\, \mathsf{0}, \mathsf{s}, +, *, @_{N^2}, @_{NN^2}, \dots \}$ and $\Sigma_0 = \{\, \mathsf{0}, \mathsf{s}, +, * \,\}$. Let $\geq$ be any partial order on Σ_0. Terms are evaluated in **Pri** as follows: $[\![\mathsf{0}]\!] = \mathsf{0}$, $[\![\mathsf{s}]\!] = \mathsf{s}$, $[\![@_{N^2}(\mathsf{s}, \mathsf{0})]\!] = \mathsf{s}$, $[\![@_{NN^2}(*, @_{N^2}(\mathsf{s}, \mathsf{0}))]\!] = *$, and $[\![@_{N^2}(@_{NN^2}(*, @_{N^2}(\mathsf{s}, \mathsf{0})), \mathsf{0})]\!] = *$. Note that $\mathrm{head}(t) = [\![\Theta(t)]\!]$ holds for all simply typed term t without variables.

Now let us consider labelling for terms and rewrite rules. The set of first-order variables is denoted by X. An *assignment* on A is a function from X to A. The result of evaluating a term t in a Σ-algebra $\mathcal{A}$ under an assignment α is denoted by $[\![t]\!]_{\mathcal{A},\alpha}$, or simply $[\![t]\!]_\alpha$. Let $\mathcal{R} = \langle \Sigma, R \rangle$ be a first-order TRS. A weakly monotone Σ-algebra $\mathcal{A}$ is a *quasi-model* of $\mathcal{R}$ if $[\![l]\!]_\alpha \geq [\![r]\!]_\alpha$ for all rewrite rule $l \to r \in R$ and assignments α on the carrier of $\mathcal{A}$.

Definition 4 (primary symbol). *Let Σ be a first-order signature, $\geq$ a partial order on the set Σ_0 of constant symbols. For any assignment α on Σ_0,* the primary symbol *of a first-order term t is defined as $[\![t]\!]_{\mathbf{Pri},\alpha} \in \Sigma_0$.*

Example 6 (primary symbol). Let Σ be the signature of $\Theta(\mathcal{R})$ in Example 4 and $\geq$ be any partial order on Σ_0. Suppose an assignment α satisfies $\alpha(F) = *$. The primary symbol of a term is computed as follows: $[\![@_{N^2}(\mathsf{s}, \mathsf{0})]\!]_\alpha = \mathsf{s}$, $[\![@_{N^2}(@_{NN^2}(F, @_{N^2}(\mathsf{s}, x)), \mathsf{0})]\!]_\alpha = *$. Note that, if the head symbol of a simply typed term t is a constant, then it is identical to the primary symbol of $\Theta(t)$; otherwise, the assignment determines the primary symbol of $\Theta(t)$.

In our labelling method, every function symbol in a term is labelled by the primary symbol of its first argument.

Definition 5 (primary symbol labelling for terms). *Let Σ be a first-order signature and $\geq$ a partial order on Σ_0. The set $\mathrm{Lab}(\Sigma)$ of labelled function symbols is defined as*

$$\mathrm{Lab}(\Sigma) = \Sigma_0 \cup \{ f_\ell \mid f \in \Sigma_n,\ n \geq 1,\ \ell \in \Sigma_0 \}$$

where f_ℓ has the same arity as f. For each assignment α on Σ_0, we define the labelling function $\mathrm{lab}_\alpha : \mathrm{T}(\Sigma, X) \to \mathrm{T}(\mathrm{Lab}(\Sigma), X)$ as follows:

$$\mathrm{lab}_\alpha(t) = \begin{cases} t & \text{if } t \in \Sigma_0 \cup X, \\ f_{[\![t_1]\!]_\alpha}(\mathrm{lab}_\alpha(t_1), \dots, \mathrm{lab}_\alpha(t_n)) & \text{if } t = f(t_1, \dots, t_n), n \geq 1. \end{cases}$$

Example 7 (primary symbol labelling for terms). Let Σ be the signature of $\Theta(\mathcal{R})$ in Example 4. Then, for any assignment α on Σ_0, we have $\mathrm{lab}_\alpha(@_{N^2}(\mathsf{s}, \mathsf{0})) = @_{N^2\mathsf{s}}(\mathsf{s}, \mathsf{0})$, and $\mathrm{lab}_\alpha(@_{N^2}(@_{NN^2}(*, \mathsf{0}), x)) = @_{N^2*}(@_{NN^2*}(*, \mathsf{0}), x)$.

Definition 6 (primary symbol labelling for rewrite systems). *For any TRS $\langle \Sigma, R \rangle$ and any partial order $\geq$ on Σ_0, we define*

$$\mathrm{LabDec}(\langle \Sigma, R \rangle) = \langle \mathrm{Lab}(\Sigma), \mathrm{Lab}(R) \cup \mathrm{Dec}(R) \rangle$$

where

$$
\begin{aligned}
\mathrm{Lab}(R) &= \{\, \mathrm{lab}_\alpha(l) \to \mathrm{lab}_\alpha(r) \mid l \to r \in R,\ \alpha : X \to \Sigma_0 \,\}, \\
\mathrm{Dec}(R) &= \{\, f_\ell\,(x_1, \ldots, x_n) \to f_{\ell'}\,(x_1, \ldots, x_n) \mid \\
&\qquad f \in \Sigma_n,\ n \geq 1,\ \ell, \ell' \in \Sigma_0,\ \ell > \ell' \,\}.
\end{aligned}
$$

Theorem 2 (soundness of primary symbol labelling). *Let $\mathcal{R} = \langle \Sigma, R \rangle$ be a first-order TRS and $\geq$ a partial order on Σ_0. If the Σ-algebra* **Pri** *is a quasi-model of $\mathcal{R}$ and* $\mathrm{LabDec}(\mathcal{R})$ *is terminating, then $\mathcal{R}$ is terminating.*

Proof. We use the quasi-order version of semantic labelling [11]. For all function symbols $f \in \Sigma \setminus \Sigma_0$, take $\langle \Sigma_0, \geq \rangle$ as the partially ordered set of labels. For the mapping $\pi_f : {\Sigma_0}^n \to \Sigma_0$ $(n \geq 1)$, which determines a label for f, we choose the projection to the first argument: $\pi_f(\ell_1, \ldots, \ell_n) = \ell_1$. □

Corollary 1 (termination of STTRS by labelling). *Let $\mathcal{R}$ be an STTRS, $\mathcal{R}' = \langle \Sigma, R' \rangle$ the unsorted version of $\Theta(\mathcal{R})$, and $\geq$ a partial order on Σ_0. If* **Pri** *is a quasi-model of $\mathcal{R}'$ and* $\mathrm{LabDec}(\mathcal{R}')$ *is terminating, then $\mathcal{R}$ is terminating.*

As mentioned in the beginning of this section, the naive labelling presented in this section is not successfully applied. Let us illustrate why the primary symbol labelling defined in this section is ineffective.

Example 8 (drawback of naive labelling). Consider the STTRS $\mathcal{R}$ in Example 4. The set of constant symbols of $\Theta(\mathcal{R})$ is $\Sigma_0 = \{\, \mathsf{0}, \mathsf{s}, +, * \,\}$. To guarantee that **Pri** is a quasi-model of the unsorted version of $\Theta(\mathcal{R})$, the partial order $\geq$ on Σ_0 should satisfy the following constraints.

1. $+ \geq \mathsf{0}, \mathsf{s}, +, *$ from the first rule, and
2. $* \geq +$ from the fourth rule.

This is impossible because $+ \geq *$ and $* \geq +$ can not be satisfied at the same time. Since every rule should be decreasing w.r.t. the interpretation, only very restricted partial orders are possible (especially, when collapsing rules are contained as above), and often the labelling is useless.

Example 9 (termination of STTRS by labelling). Consider the STTRS $\mathcal{R}$ in Example 1. Then $\mathrm{LabDec}(\Theta(\mathcal{R}))$ has the following rewrite rules:

$$
\left\{
\begin{array}{lll}
@_{N^2LL\mathsf{map}}(\mathsf{map}, F, [\,]) & \to & [\,] \\
\multicolumn{3}{l}{@_{N^2LL\mathsf{map}}(\mathsf{map}, F, @_{NLL:}(\,:\,, x, xs))} \\
\multicolumn{3}{l}{\qquad\qquad \to\ @_{NLL:}(\,:\,, @_{N^2\mathsf{s}}(F, x), @_{N^2LL\mathsf{map}}(\mathsf{map}, F, xs))} \\
\multicolumn{3}{l}{@_{N^2LL\mathsf{map}}(\mathsf{map}, F, @_{NLL:}(\,:\,, x, xs))} \\
\multicolumn{3}{l}{\qquad\qquad \to\ @_{NLL:}(\,:\,, @_{N^2\mathsf{map}}(F, x), @_{N^2LL\mathsf{map}}(\mathsf{map}, F, xs))} \\
 & \vdots & \\
@_{N^2\circ}(@_{(N^2)^3\circ}(\circ, F, G), x) & \to & @_{N^2\mathsf{s}}(F, @_{N^2\mathsf{s}}(G, x)) \\
@_{N^2\circ}(@_{(N^2)^3\circ}(\circ, F, G), x) & \to & @_{N^2\mathsf{s}}(F, @_{N^2\mathsf{map}}(G, x)) \\
@_{N^2\circ}(@_{(N^2)^3\circ}(\circ, F, G), x) & \to & @_{N^2\mathsf{s}}(F, @_{N^2:}(G, x)) \\
 & \vdots & \\
@_{(N^2)^2\mathsf{twice}}(\mathsf{twice}, F) & \to & @_{(N^2)^3\circ}(\circ, F, F)
\end{array}
\right.
$$

When labelling rewrite rules, we have to consider all assignments for variables. This gives rise to making a large number of labelled rules from a rewrite rule containing function variables.

However, these drawbacks can be avoided by restricting possibility of labelling by using sort information. To do so, we use the many-sorted version of semantic labelling.

5 Primary Symbol Labelling on Multiple Sorts

In [11], semantic labelling is developed for unsorted TRSs. The many-sorted version of semantic labelling is verified exactly in the same way as in the unsorted case. In this section, we apply the many-sorted version of semantic labelling to improve the labelling method introduced in the previous section.

Let S be a set of sorts, Σ be an S-sorted first-order signature. A *weakly monotone S-sorted Σ-algebra* $\mathcal{A} = \langle \{A^a\}_{a \in S}, \geq_{\mathcal{A}}, \{f_{\mathcal{A}}\}_{f \in \Sigma} \rangle$ consists of a family $\{A^a\}_{a \in S}$ of non-empty carrier sets, a partial order $\geq_{\mathcal{A}}$ on $A = \bigcup \{A^a \mid a \in S\}$, and a family $\{f_{\mathcal{A}}\}_{f \in \Sigma}$ of interpretations. Each constant $c \in \Sigma$ of sort a is interpreted as an element $c_{\mathcal{A}} \in A^a$, and each function symbol $f \in \Sigma$ of sort $a_1 \times \cdots \times a_n \to a_0$ is interpreted as an algebra operation $f_{\mathcal{A}} : A^{a_1} \times \cdots \times A^{a_n} \to A^{a_0}$. Similarly to the unsorted case, the operations need to be weakly monotone. Although many-sorted algebras allow pairwise disjoint domains, below we will choose carrier sets that overlap each other. In order to deal with such a case easily, we use a single partial order $\geq_{\mathcal{A}}$ over A.

Since a simply typed term, with B a set of basic types, is transformed into an $\mathrm{Ar}(B)$-sorted term by the translation Θ, we take $\mathrm{Ar}(B)$ as the set of sorts. We now verify how to fix carrier sets of many-sorted algebra, motivated by the following two observations.

Observation 1. *Constants that may occur as the primary symbol of t is restricted by the sort of t.*

Example 10 (restricting terms by sort information). Let $B = \{Nat\}$ and $C = \{\ \mathsf{0}^{Nat},\ +^{Nat \times Nat \to Nat},\ \mathsf{curry}^{(Nat \times Nat \to Nat) \to (Nat \to (Nat \to Nat))}\ \}$. Terms with the constant curry appearing as the primary symbol are limited to those having sorts $N^3\langle NN^2 \rangle$, NN^2, N^2, or N. For example,

term	sort
curry	$N^3\langle NN^2 \rangle$
$@_{N^3\langle NN^2 \rangle}(\mathsf{curry}, +)$	$\langle NN^2 \rangle$
$@_{\langle NN^2 \rangle}(@_{N^3\langle NN^2 \rangle}(\mathsf{curry}, +), \mathsf{0})$	N^2
$@_{N^2}(@_{\langle NN^2 \rangle}(@_{N^3\langle NN^2 \rangle}(\mathsf{curry}, +), \mathsf{0}), \mathsf{0})$	N

This observation leads to the following definition.

Definition 7 (range-order). *The* range-order $\trianglerighteq$ *on the set* $\mathrm{Ar}(B)$ *is the smallest partial order such that* $\langle a_1 \cdots a_n a_0 \rangle \trianglerighteq a_0$.

Example 11 (range-order). Let $B = \{ \mathit{Nat}, \mathit{List} \}$. Then $N^3 \langle NN^2 \rangle \trianglerighteq NN^2 \trianglerighteq N^2 \trianglerighteq N$. Also we have $\langle NN \rangle LL \trianglerighteq L$.

Thus, in contrast with the unsorted case, we can restrict carrier sets based on the sort information. We tentatively fix the carrier sets as $A^a = \{ c \in C^\tau \mid \mathrm{ar}(\tau) \trianglerighteq a \}$ for all sorts $a \in \mathrm{Ar}(B)$.

The first observation gives rise to unnecessarily large carrier sets $A^{\mathrm{ar}(\tau)} (= A^\tau)$ for basic types τ. Because usually most of rewrite rules are of basic types, this will impose many restrictions to our choice of the partial order. However, the following observation reveals that elements in a carrier set of a basic type need not to be distinguished.

Observation 2. *No constant of basic type is used as a label.*

Any constant of basic type, like $\mathsf{0}$ and $[\,]$, does not occur as a primary symbol in a translated term except for a constant itself, because it does not appear at the first argument of an application symbol $@_a$. This means that elements of A^τ are not used as labels whenever τ is a basic type, and thus there is no need to distinguish them. Therefore, we can take the singleton set $A^\tau = \{\bullet\}$ for every basic type τ. This in particular implies that the rewrite rules of basic types do not impose any restriction on the selection of the partial order of the many-sorted algebra.

Now, based on the observations so far, we define a many-sorted algebra.

Definition 8 (many-sorted algebra for primary symbol labelling). *Let B be a set of basic types and C a set of constants for simply typed terms. Then a many-sorted algebra $\mathbf{Pri}(B, C)$ is a weakly monotone S-sorted Σ-algebra $\langle \{A^a\}_{a \in S}, \geq, \{f_{\mathbf{Pri}(B,C)}\}_{f \in \Sigma} \rangle$ where*

- *$S = \mathrm{Ar}(B)$ and $\Sigma = \Theta(C)$,*
- *$\bullet$ is an element not contained in C and $\geq$ is a partial order on $A = \{\bullet\} \cup C$,*
- *The carrier set A^a of sort a is defined as:*

$$A^a = \begin{cases} \{\bullet\} & \text{if } a \in B \text{ or } C^{\trianglerighteq a} = \emptyset, \\ C^{\trianglerighteq a} & \text{otherwise,} \end{cases}$$

  where $C^{\trianglerighteq a} = \{ c \in C^\tau \mid \mathrm{ar}(\tau) \trianglerighteq a \}$,
- *The interpretation of a function symbol $f \in \Sigma$ of sort $a_1 \times \cdots \times a_n \to a_0$ is defined as follows:*

$$f_{\mathbf{Pri}(B,C)}(a_1, \ldots, a_n) = \begin{cases} f & \text{if } n = 0, \\ \bullet & \text{if } n \geq 1 \text{ and } A^{a_0} = \{\bullet\}, \\ d & \text{if } n \geq 1 \text{ and } A^{a_0} \neq \{\bullet\} \text{ and } A^{a_1} = \{\bullet\}, \\ a_1 & \text{otherwise,} \end{cases}$$

  where d is an arbitrarily fixed element in A^{a_0}. It is easy to see that every $f_{\mathbf{Pri}(B,C)}$ is weakly monotone.

Example 12 (many-sorted Σ-algebra $\mathbf{Pri}(B,C)$*).* Let $B = \{ \mathit{Nat} \}$, and $C = \{ \mathsf{0}^{\mathit{Nat}}, \mathsf{s}^{\mathit{Nat}\to\mathit{Nat}}, +^{\mathit{Nat}\times\mathit{Nat}\to\mathit{Nat}}, *^{\mathit{Nat}\times\mathit{Nat}\to\mathit{Nat}} \}$. Then $A = \{ \bullet, \mathsf{0}, \mathsf{s}, +, * \}$. $A^{N^2} = \{ \mathsf{s} \}$, $A^{N^3} = \{ +, * \}$, and $A^a = \{ \bullet \}$ for $a \notin \{ N^2, N^3 \}$. The order $\geq$ can be any partial order on A.

The set of many-sorted variables is denoted by X. An *assignment* on A is a function $\alpha : X \to A$ such that $x \in X^a$ implies $\alpha(x) \in A^a$. The notions of interpretation, quasi-model, and the primary symbol of a many-sorted term are defined similarly to the unsorted case.

Example 13 (interpretation). Let B and C be the ones in Example 10, and α be any assignment. Then $[\![@_{N^2}(\mathsf{s},\mathsf{0})]\!]_\alpha = [\![@_{N^3}(+,\mathsf{0},\mathsf{0})]\!]_\alpha = \bullet$. Also, $[\![+]\!]_\alpha = +$, and $[\![@_{N^3\langle NN^2\rangle}(\mathsf{curry},+)]\!]_\alpha = [\![@_{\langle NN^2\rangle}(@_{N^3\langle NN^2\rangle}(\mathsf{curry},+),\mathsf{0}))]\!]_\alpha = \mathsf{curry}$. Note that by definition $[\![\Theta(t)]\!]_\alpha = \bullet$ for all simply typed term t of basic type.

The set of labels and labelling functions are chosen similarly to the unsorted case.

Definition 9 (primary symbol labelling for terms). *Let B be a set of basic types and C a set of constants for simply typed terms, and suppose that $S = \mathrm{Ar}(B)$ and $\Sigma = \Theta(C)$. The set $\mathrm{Lab}(\Sigma)$ of S-sorted labelled function symbols is defined as*

$$\mathrm{Lab}(\Sigma) \;=\; \Sigma_0 \cup \{ f_\ell \mid f \in \Sigma,\ \mathrm{sort}(f) = a_1 \times \cdots \times a_n \to a_0,\ n \geq 1,\ \ell \in A^{a_1} \}$$

where each f_ℓ has the same sort as f. The labelling function is defined in the same way as the unsorted case.

Definition 10 (primary symbol labelling for rewrite systems). *Let B be a set of basic types and C a set of constants for simply typed terms, and suppose that $S = \mathrm{Ar}(B)$ and $\Sigma = \Theta(C)$. For any many-sorted TRS $\langle S, \Sigma, R\rangle$, define*

$$\mathrm{LabDec}(\langle S, \Sigma, R\rangle) = \langle S, \mathrm{Lab}(\Sigma), \mathrm{Lab}(R) \cup \mathrm{Dec}(R)\rangle$$

where $\mathrm{Lab}(R)$ and $\mathrm{Dec}(R)$ are defined in the same way as unsorted case.

Theorem 3 (soundness of primary symbol labelling). *Let $\mathcal{R} = \langle B, C, R\rangle$ be an STTRS, and $\geq$ be a partial order on $C \cup \{\bullet\}$. If the many-sorted algebra $\mathbf{Pri}(B,C)$ is a quasi-model of $\Theta(\mathcal{R})$ and $\mathrm{LabDec}(\Theta(\mathcal{R}))$ is terminating then $\Theta(\mathcal{R})$ is terminating.*

Proof. We use the many-sorted version of semantic labelling. The partially ordered set of labels $\mathcal{L}_f = \langle L_f, \geq\rangle$ of labels for each $f \in \Sigma$ of sort $a_1 \times \cdots \times a_n \to a_0$ ($n \geq 1$) consists of $L_f = A^{a_1}$ and the partial order $\geq$ given by the restriction of $\geq_{\mathcal{A}}$ on A^{a_1}. Take the labelling function $\pi_f : A^{a_1} \times \cdots \times A^{a_n} \to L_f$ for each $f \in \Sigma$ of sort $a_1 \times \cdots \times a_n \to a_0$ ($n \geq 1$) is defined as: $\pi_f(\ell_1, \ldots, \ell_n) = \ell_1$. $\square$

Example 14 (termination of STTRS by labelling). Let $\mathcal{R} = \langle B, C, R \rangle$ be an STTRS where $B = \{Nat\}$ and $C = \{\ \mathsf{0}^{Nat}, \mathsf{s}^{Nat \to Nat}, +^{Nat \times Nat \to Nat}, *^{Nat \times Nat \to Nat}, \mathsf{curry}^{(Nat \times Nat \to Nat) \to (Nat \to (Nat \to Nat))}, \mathsf{add}^{Nat \to (Nat \to Nat)}, \mathsf{mult}^{Nat \to (Nat \to Nat)}\ \}$, and

$$R = \begin{cases} + \ \mathsf{0}\ y & \to \ y \\ + \ (\mathsf{s}\ x)\ y & \to \ \mathsf{s}\ (+\ x\ y) \\ * \ \mathsf{0}\ y & \to \ \mathsf{0} \\ * \ (\mathsf{s}\ x)\ y & \to \ +\ y\ (*\ x\ y) \\ ((\mathsf{curry}\ F)\ x)\ y & \to \ F\ x\ y \\ \mathsf{add} & \to \ \mathsf{curry}\ + \\ \mathsf{mult} & \to \ \mathsf{curry}\ * \end{cases}$$

Note that one can not prove termination of $\Theta(\mathcal{R})$ by LPO.

Since the first five rules are of basic type, both sides evaluate to $\bullet$ in the algebra. Hence, in order to fix the partial order, we only need to focus on the last two rules, which are of non-basic types. Based on the interpretations of the both hand sides, we take the smallest partial order satisfying $\mathsf{add} > \mathsf{curry}$ and $\mathsf{mult} > \mathsf{curry}$, by which the many-sorted algebra $\mathbf{Pri}(B, C)$ is a quasi-model of $\Theta(\mathcal{R})$. Then $\mathrm{LabDec}(\Theta(\mathcal{R}))$ has the following rewrite rules:

$$\begin{cases} @_{N^3+}(+, \mathsf{0}, y) & \to \ y \\ @_{N^3+}(+, @_{N^2\mathsf{s}}(\mathsf{s}, x), y) & \to \ @_{N^2\mathsf{s}}(\mathsf{s}, @_{N^3+}(+, x, y)) \\ @_{N^3*}(*, \mathsf{0}, y) & \to \ \mathsf{0} \\ @_{N^3*}(*, @_{N^2\mathsf{s}}(\mathsf{s}, x), y) & \to \ @_{N^3+}(+, y, @_{N^3*}(*, x, y)) \\ @_{N^2\mathsf{curry}}(@_{NN^2\mathsf{curry}}(@_{N^3\langle NN^2\rangle\mathsf{curry}}(\mathsf{curry}, F), x), y) \ \to \ @_{N^3+}(F, x, y) \\ @_{N^2\mathsf{curry}}(@_{NN^2\mathsf{curry}}(@_{N^3\langle NN^2\rangle\mathsf{curry}}(\mathsf{curry}, F), x), y) \ \to \ @_{N^3*}(F, x, y) \\ \mathsf{add} & \to \ @_{N^3\langle NN^2\rangle\mathsf{curry}}(\mathsf{curry}, +) \\ \mathsf{mult} & \to \ @_{N^3\langle NN^2\rangle\mathsf{curry}}(\mathsf{curry}, *) \\ @_{N^2\mathsf{add}}(F, x) & \to \ @_{N^2\mathsf{curry}}(F, x) \\ @_{N^2\mathsf{mult}}(F, x) & \to \ @_{N^2\mathsf{curry}}(F, x) \\ @_{NN^2\mathsf{add}}(F, x) & \to \ @_{NN^2\mathsf{curry}}(F, x) \\ @_{NN^2\mathsf{mult}}(F, x) & \to \ @_{NN^2\mathsf{curry}}(F, x) \end{cases}$$

The last four rules of $\mathrm{LabDec}(\Theta(\mathcal{R}))$ are required, since the last two rules of $\mathcal{R}$ have non-basic type. In contrast to $\Theta(\mathcal{R})$, termination of $\mathrm{LabDec}(\Theta(\mathcal{R}))$ can be shown by LPO. Thus, by Theorem 3, $\mathcal{R}$ is terminating. Note that $A^{N^3\langle NN^2\rangle} = \{\mathsf{curry}\}$, $A^{NN^2} = \{\mathsf{curry}, \mathsf{add}, \mathsf{mult}\}$, $A^{N^2} = \{\mathsf{curry}, \mathsf{add}, \mathsf{mult}, \mathsf{s}\}$, $A^{N^3} = \{+, *\}$, and $A^a = \{\bullet\}$ for any $a \notin \{\ N^3\langle NN^2\rangle, NN^2, N^2, N^3\ \}$.

Corollary 2 (soundness of primary symbol labelling). *Let $\mathcal{R}$ be an STTRS such that all rules are of basic types. Then, if $\mathrm{Lab}(\Theta(\mathcal{R}))$ is terminating then $\mathcal{R}$ is terminating.*

Proof. Take the discrete order as the partial order $\geq$ on the many-sorted algebra $\mathbf{Pri}(B, C)$. $\square$

Example 15 (termination of STTRS by labelling). Let STTRS $\mathcal{R}$ be the one in Example 4. Then $\mathrm{Lab}(\Theta(\mathcal{R}))$ has the following rewrite rules:

$$\begin{cases} @_{N^2+}(@_{NN^2+}(+,\mathsf{0}),y) & \to \quad y \\ @_{N^2+}(@_{NN^2+}(+,@_{N^2\mathsf{s}}(\mathsf{s},x)),y) & \to \quad @_{N^2\mathsf{s}}(\mathsf{s},@_{N^2+}(@_{NN^2+}(+,x),y)) \\ @_{N^2*}(@_{NN^2*}(*,\mathsf{0}),y) & \to \quad \mathsf{0} \\ @_{N^2*}(@_{NN^2*}(*,@_{N^2\mathsf{s}}(\mathsf{s},x)),y) & \\ \qquad\qquad \to \quad @_{N^2+}(@_{NN^2+}(+,@_{N^2*}(@_{NN^2*}(*,x),y)),y) & \end{cases}$$

Termination of $\mathrm{Lab}(\Theta(\mathcal{R}))$ can be shown by LPO and therefore $\mathcal{R}$ is terminating by Theorem 2.

Example 16 (termination of STTRS by labelling). Let $\mathcal{R} = \langle B, C, R\rangle$ be an STTRS where $B = \{Nat, Bool, List\}$, $C = \{\ \mathsf{0}^{Nat}, \mathsf{s}^{Nat\to Nat}, \mathsf{true}^{Bool}, \mathsf{false}^{Bool}, \mathsf{gt}^{Nat\times Nat\to Bool}, \mathsf{le}^{Nat\times Nat\to Bool}, [\,]^{List}, :^{\,Nat\times List\to List}, \mathsf{filter}^{(Nat\times Nat\to Bool)\times Nat \times List\to List}, \mathsf{filtersub}^{Bool\times(Nat\times Nat\to Bool)\times Nat\times List\ \to List}, \mathsf{high}^{Nat\times List\to List}, \mathsf{low}^{Nat \times List\to List}, \mathsf{append}^{List\times List\to List}, \mathsf{qsort}^{List\to List}\ \}$, and

$$R = \begin{cases} \mathsf{gt}\ \mathsf{0}\ x & \to \quad \mathsf{false} \\ \mathsf{gt}\ (\mathsf{s}\ x)\ \mathsf{0} & \to \quad \mathsf{true} \\ \mathsf{gt}\ (\mathsf{s}\ x)\ (\mathsf{s}\ y) & \to \quad \mathsf{gt}\ x\ y \\ \mathsf{le}\ \mathsf{0}\ x & \to \quad \mathsf{true} \\ \mathsf{le}\ (\mathsf{s}\ x)\ \mathsf{0} & \to \quad \mathsf{false} \\ \mathsf{le}\ (\mathsf{s}\ x)\ (\mathsf{s}\ y) & \to \quad \mathsf{le}\ x\ y \\ \mathsf{filter}\ F\ x\ \mathsf{nil} & \to \quad \mathsf{nil} \\ \mathsf{filter}\ F\ x\ (y : ys) & \to \quad \mathsf{filtersub}\ (F\ x\ y)\ F\ x\ (y : ys) \\ \mathsf{filtersub}\ \mathsf{true}\ F\ x\ (y : ys) & \to \quad y : (\mathsf{filter}\ F\ x\ ys) \\ \mathsf{filtersub}\ \mathsf{false}\ F\ x\ (y : ys) & \to \quad \mathsf{filter}\ F\ x\ ys \\ \mathsf{high}\ x\ xs & \to \quad \mathsf{filter}\ \mathsf{gt}\ x\ xs \\ \mathsf{low}\ x\ xs & \to \quad \mathsf{filter}\ \mathsf{le}\ x\ xs \\ \mathsf{append}\ [\,]\ xs & \to \quad xs \\ \mathsf{append}\ (x : xs)\ ys & \to \quad x : (\mathsf{append}\ xs\ ys) \\ \mathsf{qsort}\ [\,] & \to \quad [\,] \\ \mathsf{qsort}\ (x : xs) & \\ \qquad \to \ \mathsf{append}\ (\mathsf{qsort}\ (\mathsf{low}\ x\ xs))\ (x : (\mathsf{qsort}\ (\mathsf{high}\ x\ xs))) & \end{cases}$$

Using the dependency pair method [2], the inequality required to show termination of $\mathrm{Lab}(\Theta(\mathcal{R}))$ is reduced to the following:

$$
\begin{array}{lcl}
@_{NNB\,\mathsf{gt}}(\mathsf{gt}, \mathsf{0}, x) & \geq & \mathsf{false} \\
@_{NNB\,\mathsf{gt}}(\mathsf{gt}, @_{N^2\mathsf{s}}(\mathsf{s}, x), \mathsf{0}) & \geq & \mathsf{true} \\
@_{NNB\,\mathsf{gt}}(\mathsf{gt}, @_{N^2\mathsf{s}}(\mathsf{s}, x), @_{N^2\mathsf{s}}(\mathsf{s}, y)) & \geq & @_{NNB\mathsf{gt}}(\mathsf{gt}, x, y) \\
@_{NNB\,\mathsf{le}}(\mathsf{le}, \mathsf{0}, x) & \geq & \mathsf{true} \\
@_{NNB\,\mathsf{le}}(\mathsf{le}, @_{N^2\mathsf{s}}(\mathsf{s}, x), \mathsf{0}) & \geq & \mathsf{false} \\
@_{NNB\,\mathsf{le}}(\mathsf{le}, @_{N^2\mathsf{s}}(\mathsf{s}, x), @_{N^2\mathsf{s}}(\mathsf{s}, y)) & \geq & @_{NNB\,\mathsf{le}}(\mathsf{le}, x, y) \\
[\,] & \geq & [\,] \\
@_{NLL\,:}(\,:\,, y, ys) & \geq & @_{NLL\,:}(\,:\,, y, ys) \\
@_{NLL\,:}(\,:\,, y, ys) & \geq & ys \\
@_{NLL\,\mathsf{high}}(\mathsf{high}, x, xs) & \geq & xs \\
@_{NLL\,\mathsf{low}}(\mathsf{low}, x, xs) & \geq & xs \\
@_{L^3\mathsf{append}}(\mathsf{append}, [\,], xs) & \geq & xs \\
@_{L^3\mathsf{append}}(\mathsf{append}, @_{NLL\,:}(\,:\,, x, xs), ys) & \geq & @_{NLL\,:}(\,:\,, x, @_{L^3\mathsf{append}}(\mathsf{append}, xs, ys)) \\
@_{L^2\mathsf{qsort}}(\mathsf{qsort}, [\,]) & \geq & [\,] \\
@_{L^2\mathsf{qsort}}(\mathsf{qsort}, @_{NLL\,:}(\,:\,, x, xs)) & & \\
\quad \geq \; @_{L^3\mathsf{append}}(\mathsf{append}, @_{L^2\mathsf{qsort}}(\mathsf{qsort}, @_{NLL\,\mathsf{low}}(\mathsf{low}, x, xs)), & & \\
\qquad\qquad @_{NLL\,\mathsf{cons}}(\mathsf{cons}, x, @_{L^2\mathsf{qsort}}(\mathsf{qsort}, @_{NLL\,\mathsf{high}}(\mathsf{high}, x, xs)))) & & \\
\mathsf{A}_{NNB\,\mathsf{gt}}(\mathsf{gt}, @_{N^2\mathsf{s}}(\mathsf{s}, x), @_{N^2\mathsf{s}}(\mathsf{s}, y)) & > & \mathsf{A}_{NNB\,\mathsf{gt}}(\mathsf{gt}, x, y) \\
\mathsf{A}_{NNB\,\mathsf{le}}(\mathsf{le}, @_{N^2\mathsf{s}}(\mathsf{s}, x), @_{N^2\mathsf{s}}(\mathsf{s}, y)) & > & \mathsf{A}_{NNB\,\mathsf{le}}(\mathsf{le}, x, y) \\
@_{NLL\,:}(:, y, ys) & > & ys \\
\mathsf{A}_{L^3\mathsf{append}}(\mathsf{append}, @_{NLL\,:}(\,:\,, x, xs), ys) & > & \mathsf{A}_{L^3\mathsf{append}}(\mathsf{append}, xs, ys) \\
@_{NLL\,:}(\,:\,, x, xs) & > & @_{NLL\,\mathsf{low}}(\mathsf{low}, x, xs) \\
@_{NLL\,:}(\,:\,, x, xs) & > & @_{NLL\,\mathsf{high}}(\mathsf{high}, x, xs)
\end{array}
$$

where $\mathsf{A}_{a\ell}$ denotes the tuple symbol for $@_{a\ell}$. The following argument filtering is used: $\pi(@_{\langle NNB\rangle NLL}) = \pi(\mathsf{A}_{\langle NNB\rangle NLL}) = 4$, $\pi(@_{B\langle NNB\rangle NLL}) = \pi(\mathsf{A}_{B\langle NNB\rangle NLL}) = 5$, $\pi(\mathsf{A}_{L^2\mathsf{qsort}}) = 2$. These inequalities are satisfied by LPO based on the following precedence: $\mathsf{gt} > \mathsf{false}$, $\mathsf{gt} > \mathsf{true}$, $\mathsf{le} > \mathsf{false}$, $\mathsf{le} > \mathsf{true}$, $@_{L^2\mathsf{qsort}} > @_{L^3\mathsf{append}} > @_{NLL:} > @_{NLL\mathsf{low}}$, $@_{L^2\mathsf{qsort}} > @_{L^3\mathsf{append}} > @_{NLL\,:}$, $\mathsf{qsort} > \mathsf{append} > \,:\, > \mathsf{low}$, $\mathsf{qsort} > \mathsf{append} > \,:$, $@_{NLL\,:} > @_{NLL\mathsf{high}}$, $: > \mathsf{high}$. Thus $\Theta(\mathcal{R})$ is terminating and therefore $\mathcal{R}$ is terminating by Theorem 2.

6 Concluding Remarks

In this paper, we have presented a method to prove termination of simply typed term rewriting systems.

We have defined a translation Θ from STTRSs $\mathcal{R}$ to many-sorted first-order TRSs $\Theta(\mathcal{R})$, and showed that $\Theta(\mathcal{R})$ is terminating if and only if $\mathcal{R}$ is terminating. The translation Θ is easily automated and thus termination of STTRSs can be proved automatically using any known automated methods for proving termination of first-order term rewriting systems.

We further presented a labelling method that facilitates termination proof of many-sorted TRSs obtained by our translation. For this, we use the many-sorted extension of semantic labelling method which is originally proved for unsorted term rewriting. We defined the labelling LabDec and Lab of many-sorted TRSs such that $\mathrm{Lab}(\Theta(\mathcal{R})) \subseteq \mathrm{LabDec}(\Theta(\mathcal{R}))$, and showed that any STTRS $\mathcal{R}$ is terminating whenever $\mathrm{LabDec}(\Theta(\mathcal{R}))$ is terminating and that any STTRS $\mathcal{R}$ whose all rules are of basic type is terminating whenever $\mathrm{Lab}(\Theta(\mathcal{R}))$ is terminating.

We have demonstrated that useful labelling by primary symbol is not implemented using unsorted semantic labelling and proposed the one using many-sorted semantic labelling. However, there is another direction to overcome this defect—that is to restrict labels so that there are just two kinds of labels p and n that distinguish defined primary symbols and constructor primary symbols. Then the soundness theorem like Theorem 3 is proved using the original semantic labelling method and that like Theorem 2 is proved by slightly modifying the original semantic labelling proof [1].

Finally we also note that another termination proof technique for STTRSs, called monotone interpretation, has been proposed in [10]. Unlike our method, the one in [10] is a semantical method and we think our techniques are more suitable for automated termination proof.

References

1. T. Aoto and T. Yamada. Proving termination of simply typed term rewriting systems automatically. *IPSJ Transactions on Programming*, 44(SIG 4 PRO 17):67–77, 2003. In Japanese.
2. T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.
3. F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, Cambridge, 1998.
4. N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–320. Elsevier Science Publishers, 1990.
5. J.-P. Jouannaud and M. Okada. Executable higher-order algebraic specification languages. In *Proceedings of the 6th IEEE Symposium on Logic in Computer Science (LICS'91)*, pages 350–361, 1991.
6. S. Kamin and J.-J. Lévy. Two generalizations of the recursive path ordering. Unpublished manuscript, University of Illinois, 1980.
7. J. W. Klop. *Combinatory Reduction Systems*. PhD thesis, Rijksuniversiteit, Utrecht, 1980.
8. J. W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 1–116. Oxford University Press, 1992.
9. R. Mayr and T. Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192:3–29, 1998.
10. T. Yamada. Confluence and termination of simply typed term rewriting systems. In *Proceedings of the 12th International Conference on Rewriting Techniques and Applications (RTA'01)*, volume 2051 of *Lecture Notes in Computer Science*, pages 338–352. Springer-Verlag, 2001.
11. H. Zantema. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae*, 24:89–105, 1995.