Predictive Labeling

Nao Hirokawa and Aart Middeldorp

Institute of Computer Science
University of Innsbruck
6020 Innsbruck, Austria
{nao.hirokawa,aart.middeldorp}@uibk.ac.at

Abstract. Semantic labeling is a transformation technique for proving the termination of rewrite systems. The semantic part is given by a quasi-model of the rewrite rules. In this paper we present a variant of semantic labeling in which the quasi-model condition is only demanded for the usable rules induced by the labeling. Our variant is less powerful in theory but maybe more useful in practice.

1 Introduction

Numerous methods are available for proving the termination of term rewrite systems, ranging from simplification orders like the Knuth-Bendix order [10], polynomial interpretations [12, 3], and path orders [4, 9], via transformation methods like semantic labeling [18] and the dependency pair method [1], to recent methods based on results from automata theory [5, 6].

In this paper we revisit the semantic labeling method of Zantema [18]. Invented back in 1995, only recently the method has become available in tools that aim to prove termination automatically. Zantema implemented a version with a binary (quasi-)model in his termination prover TORPA [19] for string rewrite systems. The termination prover TPA [11] developed by Koprowski for term rewrite systems, additionally employs natural numbers as semantics and labels. As shown by the performance of TPA in the TRS category of the 2005 termination competition,[1] this is surprisingly powerful.

We present a variant of semantic labeling which comes with less constraints on the part of the semantics. More precisely, our variant does not require that all rewrite rules of the rewrite system that we want to prove terminating need to be considered when checking the quasi-model condition. To make the discussion more concrete, let us consider the following example.

Example 1. Consider the TRS $\mathcal{R}$ consisting of the following rewrite rules:

$$\begin{aligned}
\mathsf{fact}(\mathsf{0}) &\to \mathsf{s}(\mathsf{0}) & \mathsf{0} + y &\to y \\
\mathsf{fact}(\mathsf{s}(x)) &\to \mathsf{fact}(\mathsf{p}(\mathsf{s}(x))) \times \mathsf{s}(x) & \mathsf{s}(x) + y &\to \mathsf{s}(x + y) \\
\mathsf{p}(\mathsf{s}(\mathsf{0})) &\to \mathsf{0} & \mathsf{0} \times y &\to \mathsf{0} \\
\mathsf{p}(\mathsf{s}(\mathsf{s}(x))) &\to \mathsf{s}(\mathsf{p}(\mathsf{s}(x))) & \mathsf{s}(x) \times y &\to (x \times y) + y
\end{aligned}$$

[1] http://www.lri.fr/~marche/termination-competition/2005

This is the leading example from [18] extended with the rule $\mathsf{fact}(\mathsf{0}) \to \mathsf{s}(\mathsf{0})$ and recursive rules for addition and multiplication. These additional rules cause no problems for the "standard" semantic labeling proof, which employs natural numbers as semantics and as labels for the function symbol fact, using the natural interpretations $\mathsf{0}_{\mathbb{N}} = 0$, $\mathsf{s}_{\mathbb{N}}(x) = x + 1$, $\mathsf{p}_{\mathbb{N}}(x) = \max\{x - 1, 0\}$, $x +_{\mathbb{N}} y = x + y$, $x \times_{\mathbb{N}} y = x \times y$, $\mathsf{fact}_{\mathbb{N}}(x) = x!$ and the labeling function $\mathsf{fact}_{\ell}(x) = x$. Note that the resulting algebra is a model of the rewrite rules of $\mathcal{R}$. By replacing the two rules

$$\mathsf{fact}(\mathsf{0}) \to \mathsf{s}(\mathsf{0}) \qquad\qquad \mathsf{fact}(\mathsf{s}(x)) \to \mathsf{fact}(\mathsf{p}(\mathsf{s}(x))) \times \mathsf{s}(x)$$

with the infinitely many rules

$$\mathsf{fact}_0(\mathsf{0}) \to \mathsf{s}(\mathsf{0}) \qquad \mathsf{fact}_{i+1}(\mathsf{s}(x)) \to \mathsf{fact}_i(\mathsf{p}(\mathsf{s}(x))) \times \mathsf{s}(x) \quad (\forall\, i \geqslant 0)$$

the labeled TRS $\mathcal{R}_{\mathsf{lab}}$ is obtained. The rules of this TRS are oriented from left to right by the lexicographic path order induced by the well-founded precedence

$$\mathsf{fact}_{i+1} > \mathsf{fact}_i > \cdots > \mathsf{fact}_0 > \times > + > \mathsf{p} > \mathsf{s}$$

and hence $\mathcal{R}_{\mathsf{lab}}$ is terminating. The soundness of semantic labeling guarantees that $\mathcal{R}$ is terminating, too.

Semantic labeling requires that the algebra defining the semantics is a (quasi-)model of all rewrite rules of the TRS that we want to prove terminating. This entails that we need to define semantics for all function symbols occurring in the TRS. In the variant we present in this paper, we need to define the semantics of the function symbols that appear below a function symbol that we want to label as well as the function symbols that depend on them, and the (quasi-)model condition is required only for the rules that define these function symbols. In our example, the interpretations of the function symbols $+$, $\times$, and fact may be ignored. Furthermore, the (quasi-)model condition needs to be checked for the two rules

$$\mathsf{p}(\mathsf{s}(\mathsf{0})) \to \mathsf{0} \qquad\qquad \mathsf{p}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{p}(\mathsf{s}(x)))$$

only. We prove that this is sound provided an additional condition on the algebras that may be used in connection with our variant of semantic labeling is imposed. This condition makes our variant less powerful in theory but maybe more useful in practice. Our variant is certainly more difficult to prove correct since the standard proof of transforming a presupposed infinite rewrite sequence into an infinite labeled rewrite sequence will not work without further ado due to a lack of semantic information. In the correctness proof we predict this missing information, which is why we call our variant *predictive* labeling.

The remainder of the paper is organized as follows. In the next section we recapitulate the formal definition of semantic labeling. In Section 3 we present our main result. Some more examples are presented in Section 4 and we conclude with mentioning some open issues in Section 5.

2 Preliminaries

We assume that the reader is familiar with term rewriting [2, 14]. Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and let $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$ be an $\mathcal{F}$-algebra. A *labeling* ℓ for $\mathcal{A}$ consists of sets of labels $L_f \subseteq A$ for every $f \in \mathcal{F}$ together with mappings $\ell_f \colon A^n \to L_f$ for every n-ary function symbol $f \in \mathcal{F}$ with $L_f \neq \varnothing$. The labeled signature $\mathcal{F}_{\mathsf{lab}}$ consists of n-ary function symbols f_a for every n-ary function symbol $f \in \mathcal{F}$ and label $a \in L_f$ together with all function symbols $f \in \mathcal{F}$ such that $L_f = \varnothing$. The mapping ℓ_f determines the label of the root symbol f of a term $f(t_1, \ldots, t_n)$ based on the values of the arguments $t_1, \ldots, t_n$. Let $\mathcal{V}$ be the set of variables. For every assignment $\alpha \colon \mathcal{V} \to A$ the mapping $\mathsf{lab}_\alpha \colon \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}_{\mathsf{lab}}, \mathcal{V})$ is inductively defined as follows:

$$\mathsf{lab}_\alpha(t) = \begin{cases} t & \text{if } t \text{ is a variable,} \\ f(\mathsf{lab}_\alpha(t_1), \ldots, \mathsf{lab}_\alpha(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } L_f = \varnothing, \\ f_a(\mathsf{lab}_\alpha(t_1), \ldots, \mathsf{lab}_\alpha(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } L_f \neq \varnothing \end{cases}$$

where a denotes the label $\ell_f([\alpha]_{\mathcal{A}}(t_1), \ldots, [\alpha]_{\mathcal{A}}(t_n))$. The *labeled* TRS $\mathcal{R}_{\mathsf{lab}}$ over the signature $\mathcal{F}_{\mathsf{lab}}$ consists of the rewrite rules

$$\mathsf{lab}_\alpha(l) \to \mathsf{lab}_\alpha(r)$$

for all rules $l \to r \in \mathcal{R}$ and assignments $\alpha \colon \mathcal{V} \to A$.

Theorem 2 (Zantema [18]). *Let $\mathcal{R}$ be a TRS. Let the algebra $\mathcal{A}$ be a non-empty model of $\mathcal{R}$ and let ℓ be a labeling for $\mathcal{A}$. The TRS $\mathcal{R}$ is terminating if and only if the TRS $\mathcal{R}_{\mathsf{lab}}$ is terminating.* $\square$

The condition that $\mathcal{A}$ is a model is somewhat restrictive. A stronger (in the sense that more terminating TRSs can be transformed into TRSs that can be proved terminating by simple methods) result is obtained by equipping $\mathcal{A}$ with a well-founded order such that all algebra operations and all labeling functions are weakly monotone in all coordinates.

A well-founded weakly monotone $\mathcal{F}$-algebra $(\mathcal{A}, >)$ consists of a non-empty $\mathcal{F}$-algebra $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$ and a well-founded order $>$ on the carrier A of $\mathcal{A}$ such that every algebra operation is weakly monotone in all coordinates, i.e., if $f \in \mathcal{F}$ has arity $n \geqslant 1$ then

$$f_{\mathcal{A}}(a_1, \ldots, a_i, \ldots, a_n) \geqslant f_{\mathcal{A}}(a_1, \ldots, b, \ldots, a_n)$$

for all $a_1, \ldots, a_n, b \in A$ and $i \in \{1, \ldots, n\}$ with $a_i > b$. The relation $\geqslant_{\mathcal{A}}$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ is defined as follows: $s \geqslant_{\mathcal{A}} t$ if $[\alpha]_{\mathcal{A}}(s) \geqslant [\alpha]_{\mathcal{A}}(t)$ for all assignments α. We say that $(\mathcal{A}, >)$ is a quasi-model of a TRS $\mathcal{R}$ if $\mathcal{R} \subseteq {\geqslant_{\mathcal{A}}}$.

A labeling ℓ for $\mathcal{A}$ is called weakly monotone if all labeling functions ℓ_f are weakly monotone in all coordinates. The TRS $\mathcal{D}\mathrm{ec}$ consists of all rewrite rules

$$f_a(x_1, \ldots, x_n) \to f_b(x_1, \ldots, x_n)$$

with f an n-ary function symbol, $a, b \in L_f$ such that $a > b$, and $x_1, \ldots, x_n$ pairwise different variables.

Theorem 3 (Zantema [18]). *Let $\mathcal{R}$ be a TRS, $(\mathcal{A}, >)$ a well-founded weakly monotone quasi-model for $\mathcal{R}$, and ℓ a weakly monotone labeling for $(\mathcal{A}, >)$. The TRS $\mathcal{R}$ is terminating if and only if the TRS $\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}$ is terminating.* $\square$

In [13] it is shown how Theorem 3 can be used to transform any terminating TRS into a so-called *precedence terminating* TRS, which are defined as having the property that there exists a well-founded precedence $\sqsupset$ such that $\mathrm{root}(l) \sqsupset f$ for every rewrite rule $l \to r$ and every function symbol $f \in \mathcal{F}\mathsf{un}(r)$. This condition ensures that the rewrite rules can be oriented from left to right by the lexicographic path order induced by the precedence. Needless to say, this particular transformation is not effective.

We conclude this preliminary section with a simple but useful fact that underlies the dependency pair method [1]. This fact is used to obtain the main result presented in the next section. The easy proof can be found in [8]. Here $\mathcal{T}_\infty$ denotes the set of minimal non-terminating terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$, minimal in the sense that all arguments are terminating.

Lemma 4. *For every term $t \in \mathcal{T}_\infty$ there exists a rewrite rule $l \to r$, a substitution σ, and a non-variable subterm u of r such that $t \xrightarrow{>\epsilon}{}^* l\sigma \xrightarrow{\epsilon} r\sigma \trianglerighteq u\sigma$ and $l\sigma, u\sigma \in \mathcal{T}_\infty$.* $\square$

In the following we do not use the fact that all steps in the rewrite sequence from t to $l\sigma$ take place below the root.

3 Predictive Labeling

Our aim is to weaken the quasi-model condition $\mathcal{R} \subseteq \geqslant_{\mathcal{A}}$ in Theorem 3 by replacing $\mathcal{R}$ with the *usable rules* of the labeling ℓ. The concept of usable rules originates from [1]. We extend the definition to labelings.

Definition 5. *For function symbols f and g we write $f \rhd_{\mathrm{d}} g$ if there exists a rewrite rule $l \to r \in \mathcal{R}$ such that $f = \mathrm{root}(l)$ and g is a defined function symbol in $\mathcal{F}\mathsf{un}(r)$. Let ℓ be a labeling and t a term. We define*

$$\mathcal{G}_\ell(t) = \begin{cases} \varnothing & \text{if } t \text{ is a variable,} \\ \mathcal{F}\mathsf{un}(t_1)^* \cup \cdots \cup \mathcal{F}\mathsf{un}(t_n)^* & \text{if } t = f(t_1, \ldots, t_n) \text{ and } L_f \neq \varnothing, \\ \mathcal{G}_\ell(t_1) \cup \cdots \cup \mathcal{G}_\ell(t_n) & \text{if } t = f(t_1, \ldots, t_n) \text{ and } L_f = \varnothing \end{cases}$$

where F^* denotes the set $\{g \mid f \rhd_{\mathrm{d}}^* g \text{ for some } f \in F\}$. Furthermore we define

$$\mathcal{G}_\ell(\mathcal{R}) = \bigcup_{l \to r \in \mathcal{R}} \mathcal{G}_\ell(l) \cup \mathcal{G}_\ell(r).$$

The set of usable rules *for ℓ is defined as $\mathcal{U}(\ell) = \{l \to r \in \mathcal{R} \mid \mathrm{root}(l) \in \mathcal{G}_\ell(\mathcal{R})\}$.*

In the following we simply write $\mathcal{G}_\ell$ for $\mathcal{G}_\ell(\mathcal{R})$.

Example 6. With respect to the TRS $\mathcal{R}$ and the labeling ℓ restricted to fact in Example 1 we have $\mathcal{G}_\ell = \{\mathsf{0}, \mathsf{p}, \mathsf{s}\}$. Since $\mathsf{0}$ and s are constructors, $\mathcal{U}(\ell)$ consists of the two rules $\mathsf{p}(\mathsf{s}(\mathsf{0})) \to \mathsf{0}$ and $\mathsf{p}(\mathsf{s}(\mathsf{s}(x))) \to \mathsf{s}(\mathsf{p}(\mathsf{s}(x)))$ that define the function symbol p.

In our version of semantic labeling we require $\mathcal{U}(\ell) \subseteq \geqslant_{\mathcal{A}}$ instead of $\mathcal{R} \subseteq \geqslant_{\mathcal{A}}$. Moreover, we only need to define semantics for the function symbols in $\mathcal{G}_\ell$. Without further ado, this would be unsound, as can be seen from the following example.

Example 7. Consider the non-terminating TRS $\mathcal{R}$ (from [16])

$$\mathsf{f}(\mathsf{a}, \mathsf{b}, x) \to \mathsf{f}(x, x, x) \qquad \mathsf{g}(x, y) \to x \qquad \mathsf{g}(x, y) \to y$$

We want to distinguish the two occurrences of the function symbol f. This can be achieved by an algebra $\mathcal{A}$ consisting of the carrier $\{0, 1, a, b\}$ equipped with the well-founded order $> = \{(1, 0)\}$ and the interpretations $\mathsf{a}_{\mathcal{A}} = a$ and $\mathsf{b}_{\mathcal{A}} = b$, together with the weakly monotone labeling function

$$\ell_{\mathsf{f}}(x, y, z) = \begin{cases} 1 & \text{if } x = a \text{ and } y = b \\ 0 & \text{otherwise} \end{cases}$$

We have $\mathcal{G}_\ell = \{\mathsf{a}, \mathsf{b}\}$ and $\mathcal{U}(\ell) = \varnothing$. Obviously $\mathcal{U}(\ell) \subseteq \geqslant_{\mathcal{A}}$. The transformed TRS $\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}$

$$\begin{array}{lll} \mathsf{f}_1(\mathsf{a}, \mathsf{b}, x) \to \mathsf{f}_0(x, x, x) & \qquad \mathsf{g}(x, y) \to x & \qquad \mathsf{g}(x, y) \to y \\ \mathsf{f}_1(x, y, z) \to \mathsf{f}_0(x, y, z) & & \end{array}$$

is terminating.

Definition 8. *Let $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$ be an algebra equipped with a proper order $>$ on its carrier A. We say that $(\mathcal{A}, >)$ is a $\sqcup$-algebra if for all finite subsets $X \subseteq A$ there exists a least upper bound $\bigsqcup X$ of X in A. We denote $\bigsqcup \varnothing$ by $\bot$.*

Since every element of A is an upper bound of $\varnothing$, it follows that $\bot$ is the minimum element of A. This is used in the proof of Lemma 13 below. Note that the algebra in Example 7 is not a $\sqcup$-algebra as the set $\{a, b\}$ has no upper bound.

In the remainder of this section we assume that $\mathcal{R}$ is a *finitely branching* TRS over a signature $\mathcal{F}$, $(\mathcal{A}, >)$ with $\mathcal{A} = (A, \{f_{\mathcal{A}}\}_{f \in \mathcal{F}})$ a well-founded weakly monotone $\sqcup$-algebra, and ℓ a weakly monotone labeling for $(\mathcal{A}, >)$ such that $\mathcal{U}(\ell) \subseteq \geqslant_{\mathcal{A}}$ and $f_{\mathcal{A}}(a_1, \ldots, a_n) = \bot$ for all $f \notin \mathcal{G}_\ell$ and $a_1, \ldots, a_n \in A$. The latter assumption is harmless because function symbols in $\mathcal{F} \setminus \mathcal{G}_\ell$ are not involved when computing $\mathcal{R}_{\mathsf{lab}}$ or verifying $\mathcal{U}(\ell) \subseteq \geqslant_{\mathcal{A}}$.

The if-direction of Theorem 3 is proved in [18] by transforming a presupposed infinite rewrite sequence in $\mathcal{R}$ into an infinite rewrite sequence in $\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}$. This transformation is achieved by applying the labeling function $\mathsf{lab}_\alpha(\cdot)$ (for an arbitrary assignment α) to all terms in the infinite rewrite sequence of $\mathcal{R}$. The key property is that

$$\mathsf{lab}_\alpha(s) \to^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} \mathsf{lab}_\alpha(t)$$

whenever $s \to_{\mathcal{R}} t$. In our setting this approach does not work since we lack sufficient semantic information to label arbitrary terms.

In the following definition an interpretation function $[\alpha]^*_{\mathcal{A}}(\cdot)$ is given for all terminating terms in $\mathcal{T}(\mathcal{F},\mathcal{V})$ which provides more information than the standard interpretation function $[\alpha]_{\mathcal{A}}(\cdot)$. We write $\mathcal{SN}$ for the subset of $\mathcal{T}(\mathcal{F},\mathcal{V})$ consisting of all terminating terms.

Definition 9. *Let $t \in \mathcal{SN}$ and α an assignment. We define the interpretation $[\alpha]^*_{\mathcal{A}}(t)$ inductively as follows:*

$$[\alpha]^*_{\mathcal{A}}(t) = \begin{cases} \alpha(x) & \text{if } t \text{ is a variable,} \\ f_{\mathcal{A}}([\alpha]^*_{\mathcal{A}}(t_1), \dots, [\alpha]^*_{\mathcal{A}}(t_n)) & \text{if } t = f(t_1,\dots,t_n) \text{ and } f \in \mathcal{G}_\ell, \\ \bigsqcup\{[\alpha]^*_{\mathcal{A}}(u) \mid t \to^+_{\mathcal{R}} u\} & \text{if } t = f(t_1,\dots,t_n) \text{ and } f \notin \mathcal{G}_\ell. \end{cases}$$

Note that the recursion in the definition of $[\alpha]^*_{\mathcal{A}}(\cdot)$ terminates because the union of $\to^+_{\mathcal{R}}$ and the proper superterm relation $\rhd$ is a well-founded relation on $\mathcal{SN}$. Further note that the operation $\bigsqcup$ is applied only to finite sets as $\mathcal{R}$ is assumed to be finitely branching. The definition of $[\alpha]^*_{\mathcal{A}}(t)$ can be viewed as a semantic version of a transformation of Gramlich [7, Definition 3], which is used for proving the modularity of collapsing extending ($\mathcal{C}_{\mathcal{E}}$) termination of finite branching TRSs. Here $\mathcal{R}$ is *$\mathcal{C}_{\mathcal{E}}$-terminating* if $\mathcal{R} \cup \{\mathsf{g}(x,y) \to x, \mathsf{g}(x,y) \to y\}$ with g a fresh function symbol is terminating. We remark that every $\sqcup$-algebra $(\mathcal{A}, >)$ satisfies $\mathsf{g}(x,y) \geqslant_{\mathcal{A}} x$ and $\mathsf{g}(x,y) \geqslant_{\mathcal{A}} y$ by taking the interpretation $\mathsf{g}_{\mathcal{A}}(x,y) = \bigsqcup\{x,y\}$. Variations of Gramlich's definition have been more recently used in [17, 8, 15] to reduce the constraints originating from the dependency pair method.

The induced labeling function can be defined for terminating and for minimal non-terminating terms but not for arbitrary terms in $\mathcal{T}(\mathcal{F},\mathcal{V})$.

Definition 10. *Let $t \in \mathcal{SN} \cup \mathcal{T}_\infty$ and α an assignment. We define the labeled term $\mathsf{lab}^*_\alpha(t)$ inductively as follows:*

$$\mathsf{lab}^*_\alpha(t) = \begin{cases} t & \text{if } t \text{ is a variable,} \\ f(\mathsf{lab}^*_\alpha(t_1), \dots, \mathsf{lab}^*_\alpha(t_n)) & \text{if } L_f = \varnothing, \\ f_a(\mathsf{lab}^*_\alpha(t_1), \dots, \mathsf{lab}^*_\alpha(t_n)) & \text{if } L_f \neq \varnothing \end{cases}$$

where $a = \ell_f([\alpha]^*_{\mathcal{A}}(t_1), \dots, [\alpha]^*_{\mathcal{A}}(t_n))$.

We illustrate the above definitions on a concrete rewrite sequence with respect to the factorial example of the introduction.

Example 11. Consider the TRS $\mathcal{R}$ and the labeling ℓ restricted to fact of Example 1. We assume that $f_{\mathbb{N}}(x_1,\dots,x_n) = 0$ for all function symbols $f \in \{\mathsf{fact}, +, \times\}$ and all $x_1,\dots,x_n \in \mathbb{N}$. Consider the rewrite sequence

$$\mathsf{fact}(\mathsf{s}(0) + \mathsf{fact}(0)) \to \mathsf{fact}(\mathsf{s}(0 + \mathsf{fact}(0))) \to \mathsf{fact}(\mathsf{s}(\mathsf{fact}(0))) \to \mathsf{fact}(\mathsf{s}(\mathsf{s}(0)))$$

and let α be an arbitrary assignment. (Since we deal with ground terms, the assignment does not matter.) We have

$$\begin{aligned}
[\alpha]^*_{\mathbb{N}}(\mathsf{s}(\mathsf{0})) &= 1 \\
[\alpha]^*_{\mathbb{N}}(\mathsf{s}(\mathsf{s}(\mathsf{0}))) &= 2 \\
[\alpha]^*_{\mathbb{N}}(\mathsf{fact}(\mathsf{0})) &= [\alpha]^*_{\mathbb{N}}(\mathsf{0} + \mathsf{s}(\mathsf{0})) = \bigsqcup\{[\alpha]^*_{\mathbb{N}}(\mathsf{s}(\mathsf{0}))\} = \bigsqcup\{1\} = 1 \\
[\alpha]^*_{\mathbb{N}}(\mathsf{s}(\mathsf{fact}(\mathsf{0}))) &= \mathsf{s}_{\mathbb{N}}([\alpha]^*_{\mathbb{N}}(\mathsf{fact}(\mathsf{0}))) = 1 + 1 = 2 \\
[\alpha]^*_{\mathbb{N}}(\mathsf{0} + \mathsf{fact}(\mathsf{0})) &= \bigsqcup\{[\alpha]^*_{\mathbb{N}}(\mathsf{fact}(\mathsf{0})), [\alpha]^*_{\mathbb{N}}(\mathsf{0} + \mathsf{s}(\mathsf{0})), [\alpha]^*_{\mathbb{N}}(\mathsf{s}(\mathsf{0}))\} = \bigsqcup\{1\} = 1 \\
[\alpha]^*_{\mathbb{N}}(\mathsf{s}(\mathsf{0} + \mathsf{fact}(\mathsf{0}))) &= \mathsf{s}_{\mathbb{N}}([\alpha]^*_{\mathbb{N}}(\mathsf{0} + \mathsf{fact}(\mathsf{0}))) = 1 + 1 = 2 \\
[\alpha]^*_{\mathbb{N}}(\mathsf{s}(\mathsf{0}) + \mathsf{fact}(\mathsf{0})) &= \bigsqcup\{\cdots\} = 2
\end{aligned}$$

and hence by applying $\mathsf{lab}^*_\alpha(\cdot)$ to all terms in the above rewrite sequence we obtain the sequence

$$\begin{aligned}
\mathsf{fact}_2(\mathsf{s}(\mathsf{0}) + \mathsf{fact}_0(\mathsf{0})) &\to \mathsf{fact}_2(\mathsf{s}(\mathsf{0} + \mathsf{fact}_0(\mathsf{0}))) \to \mathsf{fact}_2(\mathsf{s}(\mathsf{fact}_0(\mathsf{0}))) \\
&\to \mathsf{fact}_2(\mathsf{s}(\mathsf{s}(\mathsf{0})))
\end{aligned}$$

in $\mathcal{R}_{\mathsf{lab}}$.

The following lemma compares the predicted semantics of an instantiated terminating term to the original semantics of the uninstantiated term, in which the substitution becomes part of the assignment.

Definition 12. *Given an assignment α and a substitution σ such that $\sigma(x) \in \mathcal{SN}$ for all variables x, the assignment α^*_σ is defined as $[\alpha]^*_{\mathcal{A}} \circ \sigma$ and the substitution $\sigma_{\mathsf{lab}^*_\alpha}$ as $\mathsf{lab}^*_\alpha \circ \sigma$.*

Lemma 13. *If $t\sigma \in \mathcal{SN}$ then $[\alpha]^*_{\mathcal{A}}(t\sigma) \geqslant [\alpha^*_\sigma]_{\mathcal{A}}(t)$. If in addition $\mathcal{F}\mathsf{un}(t) \subseteq \mathcal{G}_\ell$ then $[\alpha]^*_{\mathcal{A}}(t\sigma) = [\alpha^*_\sigma]_{\mathcal{A}}(t)$.*

Proof. We use structural induction on t. If t is a variable then

$$[\alpha]^*_{\mathcal{A}}(t\sigma) = ([\alpha]^*_{\mathcal{A}} \circ \sigma)(t) = [\alpha^*_\sigma]_{\mathcal{A}}(t).$$

Suppose $t = f(t_1, \ldots, t_n)$. We distinguish two cases.

1. If $f \in \mathcal{G}_\ell$ then

$$\begin{aligned}
[\alpha]^*_{\mathcal{A}}(t\sigma) &= f_{\mathcal{A}}([\alpha]^*_{\mathcal{A}}(t_1\sigma), \ldots, [\alpha]^*_{\mathcal{A}}(t_n\sigma)) \\
&\geqslant f_{\mathcal{A}}([\alpha^*_\sigma]_{\mathcal{A}}(t_1), \ldots, [\alpha^*_\sigma]_{\mathcal{A}}(t_n)) \\
&= [\alpha^*_\sigma]_{\mathcal{A}}(t)
\end{aligned}$$

   where the inequality follows from the induction hypothesis (note that $t_i\sigma \in \mathcal{SN}$ for all $i = 1, \ldots, n$) and the weak monotonicity of $f_{\mathcal{A}}$. If $\mathcal{F}\mathsf{un}(t) \subseteq \mathcal{G}_\ell$ then $\mathcal{F}\mathsf{un}(t_i) \subseteq \mathcal{G}_\ell$ and the inequality is turned into an equality.

2. If $f \notin \mathcal{G}_\ell$ then $f_\mathcal{A}(a_1, \ldots, a_n) = \bot$ for all $a_1, \ldots, a_n \in A$ and thus

$$[\alpha]^*_\mathcal{A}(t\sigma) \geqslant \bot = [\alpha^*_\sigma]_\mathcal{A}(t)$$

   In this case $\mathcal{F}\mathsf{un}(t) \subseteq \mathcal{G}_\ell$ does not hold, so the second part of the lemma holds vacuously. □

The next two lemmata do the same for labeled terms. Since the label of a function symbol depends on the semantics of its arguments, we can deal with minimal non-terminating terms.

Lemma 14. *Let $t\sigma \in \mathcal{SN} \cup \mathcal{T}_\infty$. If $\mathcal{F}\mathsf{un}(t_1) \cup \cdots \cup \mathcal{F}\mathsf{un}(t_n) \subseteq \mathcal{G}_\ell$ when $t = f(t_1, \ldots, t_n)$ then $\mathsf{lab}^*_\alpha(t\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$.*

Proof. We use structural induction on t. If t is a variable then

$$\mathsf{lab}^*_\alpha(t\sigma) = t\sigma_{\mathsf{lab}^*_\alpha} = \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}.$$

Suppose $t = f(t_1, \ldots, t_n)$. The induction hypothesis yields

$$\mathsf{lab}^*_\alpha(t_i\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t_i)\sigma_{\mathsf{lab}^*_\alpha}$$

for $i = 1, \ldots, n$. We distinguish two cases.

1. If $L_f = \varnothing$ then

$$\begin{aligned}\mathsf{lab}^*_\alpha(t\sigma) &= f(\mathsf{lab}^*_\alpha(t_1\sigma), \ldots, \mathsf{lab}^*_\alpha(t_n\sigma))\\ &= f(\mathsf{lab}_{\alpha^*_\sigma}(t_1)\sigma_{\mathsf{lab}^*_\alpha}, \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n)\sigma_{\mathsf{lab}^*_\alpha})\\ &= f(\mathsf{lab}_{\alpha^*_\sigma}(t_1), \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n))\sigma_{\mathsf{lab}^*_\alpha}\\ &= \mathsf{lab}_{\alpha^*_\sigma}(f(t_1, \ldots, t_n))\sigma_{\mathsf{lab}^*_\alpha}.\end{aligned}$$

2. If $L_f \neq \varnothing$ then

$$\begin{aligned}\mathsf{lab}^*_\alpha(t\sigma) &= f_a(\mathsf{lab}^*_\alpha(t_1\sigma), \ldots, \mathsf{lab}^*_\alpha(t_n\sigma))\\ &= f_a(\mathsf{lab}_{\alpha^*_\sigma}(t_1)\sigma_{\mathsf{lab}^*_\alpha}, \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n)\sigma_{\mathsf{lab}^*_\alpha})\\ &= f_a(\mathsf{lab}_{\alpha^*_\sigma}(t_1), \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n))\sigma_{\mathsf{lab}^*_\alpha}\end{aligned}$$

   and

$$\mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha} = f_b(\mathsf{lab}_{\alpha^*_\sigma}(t_1), \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n))\sigma_{\mathsf{lab}^*_\alpha}$$

   with $a = \ell_f([\alpha]^*_\mathcal{A}(t_1\sigma), \ldots, [\alpha]^*_\mathcal{A}(t_n\sigma))$ and $b = \ell_f([\alpha^*_\sigma]_\mathcal{A}(t_1), \ldots, [\alpha^*_\sigma]_\mathcal{A}(t_n))$. Because $\mathcal{F}\mathsf{un}(t_i) \subseteq \mathcal{G}_\ell$, Lemma 13 yields $[\alpha]^*_\mathcal{A}(t_i\sigma) = [\alpha^*_\sigma]_\mathcal{A}(t_i)$, for all $i = 1, \ldots, n$. Hence $a = b$ and therefore $\mathsf{lab}^*_\alpha(t\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$ as desired. □

Lemma 15. *If $t\sigma \in \mathcal{SN} \cup \mathcal{T}_\infty$ then $\mathsf{lab}^*_\alpha(t\sigma) \to^*_{\mathcal{D}\mathrm{ec}} \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$.*

Proof. We use structural induction on t. If t is a variable then we obtain $\mathsf{lab}^*_\alpha(t\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$ from Lemma 14. Suppose $t = f(t_1, \ldots, t_n)$. Note that $t_1, \ldots, t_n \in \mathcal{SN}$. The induction hypothesis yields $\mathsf{lab}^*_\alpha(t_i\sigma) \to^*_{\mathcal{D}\mathrm{ec}} \mathsf{lab}_{\alpha^*_\sigma}(t_i)\sigma_{\mathsf{lab}^*_\alpha}$ for all $i = 1, \ldots, n$. We distinguish two cases.

1. If $L_f = \varnothing$ then

$$\begin{aligned}\mathsf{lab}^*_\alpha(t\sigma) = {}& f(\mathsf{lab}^*_\alpha(t_1\sigma), \ldots, \mathsf{lab}^*_\alpha(t_n\sigma)) \\ \to^*_{\mathcal{D}\mathrm{ec}} {}& f(\mathsf{lab}_{\alpha^*_\sigma}(t_1)\sigma_{\mathsf{lab}^*_\alpha}, \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n)\sigma_{\mathsf{lab}^*_\alpha}) \\ = {}& f(\mathsf{lab}_{\alpha^*_\sigma}(t_1), \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n))\sigma_{\mathsf{lab}^*_\alpha} \\ = {}& \mathsf{lab}_{\alpha^*_\sigma}(f(t_1, \ldots, t_n))\sigma_{\mathsf{lab}^*_\alpha}.\end{aligned}$$

2. If $L_f \neq \varnothing$ then

$$\begin{aligned}\mathsf{lab}^*_\alpha(t\sigma) = {}& f_a(\mathsf{lab}^*_\alpha(t_1\sigma), \ldots, \mathsf{lab}^*_\alpha(t_n\sigma)) \\ \to^*_{\mathcal{D}\mathrm{ec}} {}& f_a(\mathsf{lab}_{\alpha^*_\sigma}(t_1)\sigma_{\mathsf{lab}^*_\alpha}, \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n)\sigma_{\mathsf{lab}^*_\alpha})\end{aligned}$$

   and

$$\begin{aligned}\mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha} &= f_b(\mathsf{lab}_{\alpha^*_\sigma}(t_1), \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n))\sigma_{\mathsf{lab}^*_\alpha} \\ &= f_b(\mathsf{lab}_{\alpha^*_\sigma}(t_1)\sigma_{\mathsf{lab}^*_\alpha}, \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n)\sigma_{\mathsf{lab}^*_\alpha})\end{aligned}$$

   with $a = \ell_f([\alpha]^*_{\mathcal{A}}(t_1\sigma), \ldots, [\alpha]^*_{\mathcal{A}}(t_n\sigma))$ and $b = \ell_f([\alpha^*_\sigma]_{\mathcal{A}}(t_1), \ldots, [\alpha^*_\sigma]_{\mathcal{A}}(t_n))$. Lemma 13 yields $[\alpha]^*_{\mathcal{A}}(t_i\sigma) \geqslant [\alpha^*_\sigma]_{\mathcal{A}}(t_i)$ for all $i = 1, \ldots, n$. Because the labeling function ℓ_f is weakly monotone in all its coordinates, $a \geqslant b$. If $a > b$ then $\mathcal{D}\mathrm{ec}$ contains the rewrite rule $f_a(x_1, \ldots, x_n) \to f_b(x_1, \ldots, x_n)$ and thus (also if $a = b$)

$$f_a(\mathsf{lab}_{\alpha^*_\sigma}(t_1)\sigma_{\mathsf{lab}^*_\alpha}, \ldots, \mathsf{lab}_{\alpha^*_\sigma}(t_n)\sigma_{\mathsf{lab}^*_\alpha}) \to^=_{\mathcal{D}\mathrm{ec}} \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}.$$

   We conclude that $\mathsf{lab}^*_\alpha(t\sigma) \to^*_{\mathcal{D}\mathrm{ec}} \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$. □

The next lemma states that the rewrite sequence in Lemma 15 is empty when t is a subterm of the right-hand side of a rule.

Lemma 16. *If $l \to r \in \mathcal{R}$ and $t \trianglelefteq r$ such that $t\sigma \in \mathcal{SN} \cup \mathcal{T}_\infty$ then $\mathsf{lab}^*_\alpha(t\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$.*

Proof. We use structural induction on t. If t is a variable then we obtain

$$\mathsf{lab}^*_\alpha(t\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$$

from Lemma 14. Suppose $t = f(t_1, \ldots, t_n)$. The induction hypothesis yields $\mathsf{lab}^*_\alpha(t_i\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t_i)\sigma_{\mathsf{lab}^*_\alpha}$ for all $i = 1, \ldots, n$. If $L_f = \varnothing$ then we obtain $\mathsf{lab}^*_\alpha(t\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$ as in the proof of Lemma 14. If $L_f \neq \varnothing$ then $\mathcal{F}\mathsf{un}(t_1) \cup \cdots \cup \mathcal{F}\mathsf{un}(t_n) \subseteq \mathcal{G}_\ell$ by the definition of $\mathcal{G}_\ell$. Since $t_i\sigma \in \mathcal{SN}$, we have $t\sigma \in \mathcal{T}_\infty$ and therefore Lemma 14 yields $\mathsf{lab}^*_\alpha(t\sigma) = \mathsf{lab}_{\alpha^*_\sigma}(t)\sigma_{\mathsf{lab}^*_\alpha}$. □

We are now ready for the key lemma, which states that rewrite steps between terminating and minimal non-terminating terms can be labeled.

Lemma 17. *Let $s, t \in \mathcal{SN} \cup \mathcal{T}_\infty$. If $s \to_{\mathcal{R}} t$ then $\mathsf{lab}^*_\alpha(s) \to^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} \mathsf{lab}^*_\alpha(t)$.*

Proof. Write $s = C[l\sigma]$ and $t = C[r\sigma]$. We use structural induction on the context C. If $C = \Box$ then

$$\begin{aligned} \mathsf{lab}^*_\alpha(s) = \mathsf{lab}^*_\alpha(l\sigma) &\rightarrow^*_{\mathcal{D}\mathrm{ec}} \mathsf{lab}_{\alpha^*_\sigma}(l)\sigma_{\mathsf{lab}^*_\alpha} \\ &\rightarrow_{\mathcal{R}_{\mathsf{lab}}} \mathsf{lab}_{\alpha^*_\sigma}(r)\sigma_{\mathsf{lab}^*_\alpha} = \mathsf{lab}^*_\alpha(r\sigma) \end{aligned}$$

using Lemmata 15 and 16. Let $C = f(s_1, \ldots, C', \ldots, s_n)$. The induction hypothesis yields $\mathsf{lab}^*_\alpha(C'[l\sigma]) \rightarrow^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} \mathsf{lab}^*_\alpha(C'[r\sigma])$. We distinguish two cases.

1. If $L_f = \varnothing$ then

$$\begin{aligned} \mathsf{lab}^*_\alpha(s) = {} & f(\mathsf{lab}^*_\alpha(s_1), \ldots, \mathsf{lab}^*_\alpha(C'[l\sigma]), \ldots, \mathsf{lab}^*_\alpha(s_n)) \\ \rightarrow^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} {} & f(\mathsf{lab}^*_\alpha(s_1), \ldots, \mathsf{lab}^*_\alpha(C'[r\sigma]), \ldots, \mathsf{lab}^*_\alpha(s_n)) \\ = {} & \mathsf{lab}^*_\alpha(t). \end{aligned}$$

2. If $L_f \neq \varnothing$ then

$$\begin{aligned} \mathsf{lab}^*_\alpha(s) = {} & f_a(\mathsf{lab}^*_\alpha(s_1), \ldots, \mathsf{lab}^*_\alpha(C'[l\sigma]), \ldots, \mathsf{lab}^*_\alpha(s_n)) \\ \rightarrow^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} {} & f_a(\mathsf{lab}^*_\alpha(s_1), \ldots, \mathsf{lab}^*_\alpha(C'[r\sigma]), \ldots, \mathsf{lab}^*_\alpha(s_n)) \end{aligned}$$

with

$$a = \ell_f([\alpha]^*_{\mathcal{A}}(s_1), \ldots, [\alpha]^*_{\mathcal{A}}(C'[l\sigma]), \ldots, [\alpha]^*_{\mathcal{A}}(s_n))$$

and

$$\mathsf{lab}^*_\alpha(t) = f_b(\mathsf{lab}^*_\alpha(s_1), \ldots, \mathsf{lab}^*_\alpha(C'[r\sigma]), \ldots, \mathsf{lab}^*_\alpha(s_n))$$

with

$$b = \ell_f([\alpha]^*_{\mathcal{A}}(s_1), \ldots, [\alpha]^*_{\mathcal{A}}(C'[r\sigma]), \ldots, [\alpha]^*_{\mathcal{A}}(s_n))$$

If we can show that

$$[\alpha]^*_{\mathcal{A}}(C'[l\sigma]) \geqslant [\alpha]^*_{\mathcal{A}}(C'[r\sigma]) \tag{1}$$

then $a \geqslant b$ by weak monotonicity of ℓ_f and thus

$$f_a(\mathsf{lab}^*_\alpha(s_1), \ldots, \mathsf{lab}^*_\alpha(C'[r\sigma]), \ldots, \mathsf{lab}^*_\alpha(s_n)) \rightarrow^=_{\mathcal{D}\mathrm{ec}} \mathsf{lab}^*_\alpha(t).$$

We prove (1) by structural induction on C'.

(a) First assume that $C' = \Box$. We distinguish two cases. If $\mathrm{root}(l\sigma) = \mathrm{root}(l) \in \mathcal{G}_\ell$ then $l \rightarrow r \in \mathcal{U}(\ell)$ and $\mathcal{F}\mathsf{un}(r) \subseteq \mathcal{G}_\ell$ according to the definition of $\mathcal{G}_\ell$. Hence

$$[\alpha]^*_{\mathcal{A}}(l\sigma) \geqslant [\alpha^*_\sigma]_{\mathcal{A}}(l)$$

by Lemma 13,

$$[\alpha^*_\sigma]_{\mathcal{A}}(l) \geqslant [\alpha^*_\sigma]_{\mathcal{A}}(r)$$

since $l \geqslant_{\mathcal{A}} r$ due to the assumption $\mathcal{U}(\ell) \subseteq {\geqslant_{\mathcal{A}}}$, and

$$[\alpha^*_\sigma]_{\mathcal{A}}(r) = [\alpha]^*_{\mathcal{A}}(r\sigma)$$

by another application of Lemma 13. The combination yields the desired $[\alpha]^*_{\mathcal{A}}(l\sigma) \geqslant [\alpha]^*_{\mathcal{A}}(r\sigma)$. If $\mathrm{root}(l\sigma) = \mathrm{root}(l) \notin \mathcal{G}_\ell$ then

$$[\alpha]^*_{\mathcal{A}}(l\sigma) = \bigsqcup \{[\alpha]^*_{\mathcal{A}}(u) \mid l\sigma \to^+_{\mathcal{R}} u\}$$

Because $l\sigma \to_{\mathcal{R}} r\sigma$, $[\alpha]^*_{\mathcal{A}}(r\sigma) \in \{[\alpha]^*_{\mathcal{A}}(u) \mid l\sigma \to^+_{\mathcal{R}} u\}$ and thus also in this case $[\alpha]^*_{\mathcal{A}}(l\sigma) \geqslant [\alpha]^*_{\mathcal{A}}(r\sigma)$.

(b) Next assume that $C' = g(u_1, \ldots, C'', \ldots, u_m)$. The induction hypothesis yields $[\alpha]^*_{\mathcal{A}}(C''[l\sigma]) \geqslant [\alpha]^*_{\mathcal{A}}(C''[r\sigma])$. If $g \in \mathcal{G}_\ell$ then

$$[\alpha]^*_{\mathcal{A}}(C'[l\sigma]) = g_{\mathcal{A}}([\alpha]^*_{\mathcal{A}}(u_1), \ldots, [\alpha]^*_{\mathcal{A}}(C''[l\sigma]), \ldots, [\alpha]^*_{\mathcal{A}}(u_m))$$

and

$$[\alpha]^*_{\mathcal{A}}(C'[r\sigma]) = g_{\mathcal{A}}([\alpha]^*_{\mathcal{A}}(u_1), \ldots, [\alpha]^*_{\mathcal{A}}(C''[r\sigma]), \ldots, [\alpha]^*_{\mathcal{A}}(u_m))$$

and thus $[\alpha]^*_{\mathcal{A}}(C'[l\sigma]) \geqslant [\alpha]^*_{\mathcal{A}}(C'[r\sigma])$ by the weak monotonicity of $g_{\mathcal{A}}$. If $g \notin \mathcal{G}_\ell$ then

$$[\alpha]^*_{\mathcal{A}}(C'[l\sigma]) = \bigsqcup \{[\alpha]^*_{\mathcal{A}}(u) \mid C'[l\sigma] \to^+_{\mathcal{R}} u\}$$

Because $C'[l\sigma] \to_{\mathcal{R}} C'[r\sigma]$, $[\alpha]^*_{\mathcal{A}}(C'[r\sigma]) \in \{[\alpha]^*_{\mathcal{A}}(u) \mid C'[l\sigma] \to^+_{\mathcal{R}} u\}$ and thus $[\alpha]^*_{\mathcal{A}}(C'[l\sigma]) \geqslant [\alpha]^*_{\mathcal{A}}(C'[r\sigma])$. □

We now have all the ingredients to prove the soundness of predictive labeling.

Theorem 18. *Let $\mathcal{R}$ be a TRS, $(\mathcal{A}, >)$ a well-founded weakly monotone $\sqcup$-algebra, and ℓ a weakly monotone labeling for $(\mathcal{A}, >)$ such that $\mathcal{U}(\ell) \subseteq \geqslant_{\mathcal{A}}$. If $\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}$ is terminating then $\mathcal{R}$ is terminating.*

Proof. According to Lemma 4 for every term $t \in \mathcal{T}_\infty$ there exist a rewrite rule $l \to r \in \mathcal{R}$, a substitution σ, and a subterm u of r such that

$$t \xrightarrow{>\epsilon}{}^* l\sigma \xrightarrow{\epsilon} r\sigma \trianglerighteq u\sigma$$

and $l\sigma, u\sigma \in \mathcal{T}_\infty$. Let α be an arbitrary assignment. We will apply lab^*_α to the terms in the above sequence. From Lemma 17 we obtain

$$\mathsf{lab}^*_\alpha(t) \to^*_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} \mathsf{lab}^*_\alpha(l\sigma).$$

Since $r\sigma$ need not be an element of $\mathcal{T}_\infty$, we cannot apply Lemma 17 to the step $l\sigma \xrightarrow{\epsilon} r\sigma$. Instead we use Lemma 15 to obtain

$$\mathsf{lab}^*_\alpha(l\sigma) \to^*_{\mathcal{D}\mathrm{ec}} \mathsf{lab}_{\alpha^*_\sigma}(l)\sigma_{\mathsf{lab}^*_\alpha}.$$

Since $\mathsf{lab}_{\alpha^*_\sigma}(l) \to \mathsf{lab}_{\alpha^*_\sigma}(r) \in \mathcal{R}_{\mathsf{lab}}$,

$$\mathsf{lab}_{\alpha^*_\sigma}(l)\sigma_{\mathsf{lab}^*_\alpha} \to_{\mathcal{R}_{\mathsf{lab}}} \mathsf{lab}_{\alpha^*_\sigma}(r)\sigma_{\mathsf{lab}^*_\alpha}.$$

Because u is a subterm of r, $\mathsf{lab}_{\alpha^*_\sigma}(r)\sigma_{\mathsf{lab}^*_\alpha} \trianglerighteq \mathsf{lab}_{\alpha^*_\sigma}(u)\sigma_{\mathsf{lab}^*_\alpha}$. Lemma 16 yields $\mathsf{lab}_{\alpha^*_\sigma}(u)\sigma_{\mathsf{lab}^*_\alpha} = \mathsf{lab}^*_\alpha(u\sigma)$. Putting everything together, we obtain

$$\mathsf{lab}^*_\alpha(t) \to^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} \cdot \trianglerighteq \mathsf{lab}^*_\alpha(u\sigma).$$

Now suppose that $\mathcal{R}$ is non-terminating. Then $\mathcal{T}_\infty$ is non-empty and thus there is an infinite sequence

$$t_1 \xrightarrow{>\epsilon}{}^* \cdot \xrightarrow{\epsilon} \cdot \trianglerighteq t_2 \xrightarrow{>\epsilon}{}^* \cdot \xrightarrow{\epsilon} \cdot \trianglerighteq t_3 \xrightarrow{>\epsilon}{}^* \cdot \xrightarrow{\epsilon} \cdot \trianglerighteq \cdots$$

By the above argument, this sequence is transformed into

$$\mathsf{lab}^*_\alpha(t_1) \to^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} \cdot \trianglerighteq \mathsf{lab}^*_\alpha(t_2) \to^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} \cdot \trianglerighteq \mathsf{lab}^*_\alpha(t_3) \to^+_{\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}} \cdot \trianglerighteq \cdots$$

By introducing appropriate contexts, the latter sequence gives rise to an infinite rewrite sequence in $\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}$, contradicting the assumption that the latter system is terminating. □

We conclude this section by showing that, due to the least upper bound condition on the algebras that may be used in connection with Theorem 18, predictive labeling does not succeed in transforming every terminating TRS into a precedence terminating TRS.

Example 19. The one-rule TRS $\mathcal{R} = \{\mathsf{f}(\mathsf{a},\mathsf{b},x) \to \mathsf{f}(x,x,x)\}$ is terminating but not precedence terminating. Suppose $\mathcal{R}$ can be transformed by predictive labeling into a precedence terminating TRS. This is only possible if the two occurrences of f get a different label. Let $\mathcal{A} = (A, \{\mathsf{f}_\mathcal{A}, \mathsf{a}_\mathcal{A}, \mathsf{b}_\mathcal{A}\})$ be a well-founded weakly monotone $\sqcup$-algebra and ℓ a weakly monotone labeling such that $\mathcal{U}(\ell) \subseteq \geqslant_\mathcal{A}$ and $\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}$ is precedence terminating. Since $L_f \neq \varnothing$, the labeling function ℓ_f exists and we must have $\ell_\mathsf{f}(\mathsf{a}_\mathcal{A}, \mathsf{b}_\mathcal{A}, x) \neq \ell_\mathsf{f}(x,x,x)$ for all $x \in A$. Take $x = \bigsqcup\{\mathsf{a}_\mathcal{A}, \mathsf{b}_\mathcal{A}\}$. Since $x \geqslant \mathsf{a}_\mathcal{A}$ and $x \geqslant \mathsf{b}_\mathcal{A}$, we obtain

$$a = \ell_\mathsf{f}(x,x,x) \geqslant \ell_\mathsf{f}(\mathsf{a}_\mathcal{A}, \mathsf{b}_\mathcal{A}, x) = b$$

from the weak monotonicity of ℓ_f. Since we cannot have $a = b$, $a > b$ must hold. Hence $\mathcal{D}\mathrm{ec}$ contains the rule $\mathsf{f}_a(x,y,z) \to \mathsf{f}_b(x,y,z)$ whereas $\mathcal{R}_{\mathsf{lab}}$ contains the rule $\mathsf{f}_b(\mathsf{a},\mathsf{b},x) \to \mathsf{f}_a(x,x,x)$ It follows that $\mathcal{R}_{\mathsf{lab}} \cup \mathcal{D}\mathrm{ec}$ cannot be precedence terminating, contradicting our assumption.

4 Examples

In this section we present two more examples.

Example 20. Consider the TRS $\mathcal{R}$ consisting of the following rewrite rules:

1:	$x - \mathsf{0}$	$\to$	x	6:	$\mathsf{gcd}(\mathsf{0}, y)$	$\to$	y
2:	$\mathsf{s}(x) - \mathsf{s}(y)$	$\to$	$x - y$	7:	$\mathsf{gcd}(\mathsf{s}(x), \mathsf{0})$	$\to$	$\mathsf{s}(x)$
3:	$\mathsf{0} \leq y$	$\to$	true	8:	$\mathsf{gcd}(\mathsf{s}(x), \mathsf{s}(y))$	$\to$	$\mathsf{ifgcd}(y \leq x, \mathsf{s}(x), \mathsf{s}(y))$
4:	$\mathsf{s}(x) \leq \mathsf{0}$	$\to$	false	9:	$\mathsf{ifgcd}(\mathsf{true}, \mathsf{s}(x), \mathsf{s}(y))$	$\to$	$\mathsf{gcd}(x - y, \mathsf{s}(y))$
5:	$\mathsf{s}(x) \leq \mathsf{s}(y)$	$\to$	$x \leq y$	10:	$\mathsf{ifgcd}(\mathsf{false}, \mathsf{s}(x), \mathsf{s}(y))$	$\to$	$\mathsf{gcd}(y - x, \mathsf{s}(x))$

We use the interpretations

$$0_{\mathbb{N}} = \mathsf{true}_{\mathbb{N}} = \mathsf{false}_{\mathbb{N}} = \leq_{\mathbb{N}}(x,y) = 0 \qquad \mathsf{s}_{\mathbb{N}}(x) = x+1 \qquad -_{\mathbb{N}}(x,y) = x$$

and the labeling

$$\ell_{\mathsf{gcd}}(x,y) = x+y \qquad \ell_{\mathsf{ifgcd}}(x,y,z) = y+z$$

We have $\mathcal{G}_\ell = \{\mathsf{0}, \mathsf{s}, \mathsf{true}, \mathsf{false}, \leq, -\}$ and thus $\mathcal{U}(\ell) = \{1,2,\ldots,5\}$. One easily checks that $\mathcal{U}(\ell) \subseteq \geqslant_{\mathbb{N}}$. The TRS $\mathcal{R}_{\mathsf{lab}}$ consists of the rewrite rules

$$\begin{array}{rrcl}
1\colon & x - \mathsf{0} & \to & x \\
2\colon & \mathsf{s}(x) - \mathsf{s}(y) & \to & x - y \\
3\colon & \mathsf{0} \leq y & \to & \mathsf{true} \\
4\colon & \mathsf{s}(x) \leq \mathsf{0} & \to & \mathsf{false} \\
5\colon & \mathsf{s}(x) \leq \mathsf{s}(y) & \to & x \leq y \\
6'\colon & \mathsf{gcd}_j(\mathsf{0}, y) & \to & y \\
7'\colon & \mathsf{gcd}_{i+1}(\mathsf{s}(x), \mathsf{0}) & \to & \mathsf{s}(x) \\
8'\colon & \mathsf{gcd}_{i+j+2}(\mathsf{s}(x), \mathsf{s}(y)) & \to & \mathsf{ifgcd}_{i+j+2}(y \leq x, \mathsf{s}(x), \mathsf{s}(y)) \\
9'\colon & \mathsf{ifgcd}_{i+j+2}(\mathsf{true}, \mathsf{s}(x), \mathsf{s}(y)) & \to & \mathsf{gcd}_{i+j+1}(x - y, \mathsf{s}(y)) \\
10'\colon & \mathsf{ifgcd}_{i+j+2}(\mathsf{false}, \mathsf{s}(x), \mathsf{s}(y)) & \to & \mathsf{gcd}_{i+j+1}(y - x, \mathsf{s}(x))
\end{array}$$

for all $i, j \geqslant 0$ and the TRS $\mathcal{D}\mathrm{ec}$ consists of the rules $\mathsf{gcd}_i(x,y) \to \mathsf{gcd}_j(x,y)$ and $\mathsf{ifgcd}_i(x,y,z) \to \mathsf{ifgcd}_j(x,y,z)$ for all $i > j \geqslant 0$. Their union is oriented from left to right by the lexicographic path order induced by the well-founded precedence

$$\mathsf{ifgcd}_{i+1} > \mathsf{gcd}_{i+1} > \mathsf{ifgcd}_i > \mathsf{gcd}_i > \cdots > \mathsf{gcd}_0 > - > \leq > \mathsf{true} > \mathsf{false}$$

and hence terminating. Theorem 18 yields the termination of $\mathcal{R}$.

Example 21. Consider the TRS $\mathcal{R}$ consisting of the following rewrite rules:

$$\begin{array}{rrcl@{\qquad}rrcl}
1\colon & \mathsf{half}(\mathsf{0}) & \to & \mathsf{0} & 4\colon & \mathsf{bits}(\mathsf{0}) & \to & \mathsf{0} \\
2\colon & \mathsf{half}(\mathsf{s}(\mathsf{0})) & \to & \mathsf{0} & 5\colon & \mathsf{bits}(\mathsf{s}(x)) & \to & \mathsf{s}(\mathsf{bits}(\mathsf{half}(\mathsf{s}(x)))) \\
3\colon & \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) & \to & \mathsf{s}(\mathsf{half}(x)) & & & &
\end{array}$$

We use the interpretations

$$0_{\mathbb{N}} = 0 \qquad \mathsf{s}_{\mathbb{N}}(x) = x+1 \qquad \mathsf{half}_{\mathbb{N}}(x) = \max\{x-1, 0\}$$

and the labeling $\ell_{\mathsf{bit}}(x) = x$. We have $\mathcal{G}_\ell = \{\mathsf{0}, \mathsf{s}, \mathsf{half}\}$ and $\mathcal{U}(\ell) = \{1,2,3\}$. Clearly $\mathcal{U}(\ell) \subseteq \geqslant_{\mathbb{N}}$. The TRS $\mathcal{R}_{\mathsf{lab}}$ consists of the rewrite rules

$$\begin{array}{rrcl@{\qquad}rrcl}
1\colon & \mathsf{half}(\mathsf{0}) & \to & \mathsf{0} & 4'\colon & \mathsf{bits}_0(\mathsf{0}) & \to & \mathsf{0} \\
2\colon & \mathsf{half}(\mathsf{s}(\mathsf{0})) & \to & \mathsf{0} & 5'\colon & \mathsf{bits}_{i+1}(\mathsf{s}(x)) & \to & \mathsf{s}(\mathsf{bits}_i(\mathsf{half}(\mathsf{s}(x)))) \\
3\colon & \mathsf{half}(\mathsf{s}(\mathsf{s}(x))) & \to & \mathsf{s}(\mathsf{half}(x)) & & & &
\end{array}$$

for all $i \geqslant 0$ and the TRS $\mathcal{D}\mathrm{ec}$ consists of the rules $\mathsf{bits}_i(x) \to \mathsf{bits}_j(x)$ for all $i > j \geqslant 0$. Their union is oriented from left to right by the lexicographic path order induced by the well-founded precedence

$$\mathsf{bits}_{i+1} > \mathsf{bits}_i > \cdots > \mathsf{bits}_0 > \mathsf{half} > \mathsf{s} > \mathsf{0}$$

and thus we conclude that $\mathcal{R}$ is terminating.

5 Conclusion

Predictive labeling (Theorem 18) is a variant of the quasi-model version of semantic labeling (Theorem 3). A natural question is whether the usable rules refinement is also applicable to the model version of semantic labeling (Theorem 2). This would be interesting not so much because we would get rid of the rewrite rules in $\mathcal{D}\mathrm{ec}$ but especially because without the weak monotonicity condition more labeling functions are possible. The termination prover TORPA mentioned in the introduction implements both versions of semantic labeling for that reason. Unfortunately, it is not immediately clear how the definitions and proofs in Section 3 have to be modified in order to obtain a model version of predictive labeling.

Although the example at the end of Section 3 shows that predictive labeling is less powerful than semantic labeling, we believe that predictive labeling may be more useful when it comes to automation. First note that the algebras used in the implementations of semantic labeling mentioned in the introduction, $\{0, 1\}$ and $\mathbb{N}$ equipped with the standard order, are $\sqcup$-algebras. Second, the usable rules induced by the labeling are often a small subset of the set of all rewrite rules. Hence the possibility of finding a suitable interpretation increases while at the same time the search space decreases. The overhead of computing $\mathcal{U}(\ell)$ is negligible. Therefore we believe that termination provers that incorporate semantic labeling may benefit from our result. In the version of semantic labeling implemented in TPA [11] all function symbols are labeled. This entails that most rewrite rules are usable. As a consequence the termination proving power of TPA is only modestly increased by predictive labeling (Adam Koprowski, personal communication).

Acknowledgments

We thank Adam Koprowski, Georg Moser, and the anonymous referees for several suggestions to improve the presentation.

References

1. T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.
2. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

3. A. Ben Cherifa and P. Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computer Programming*, 9:137–159, 1987.
4. N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17:279–301, 1982.
5. A. Geser, D. Hofbauer, and J. Waldmann. Match-bounded string rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 15:149–171, 2004.
6. A. Geser, D. Hofbauer, J. Waldmann, and H. Zantema. On tree automata that certify termination of left-linear term rewriting systems. In *Proceedings of the 16th International Conference on Rewriting Techniques and Applications*, volume 3467 of *Lecture Notes Computer Science*, pages 353–367, 2005.
7. B. Gramlich. Generalized sufficient conditions for modular termination of rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 5:131–158, 1994.
8. N. Hirokawa and A. Middeldorp. Dependency pairs revisited. In *Proceedings of the 16th International Conference on Rewriting Techniques and Applications*, volume 3091 of *Lecture Notes Computer Science*, pages 249–268, 2004.
9. S. Kamin and J.J. Lévy. Two generalizations of the recursive path ordering. Unpublished manuscript, University of Illinois, 1980.
10. D.E. Knuth and P. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.
11. A. Koprowski. TPA: Termination proved automatically. In *Proceedings of the 17th International Conference on Rewriting Techniques and Applications*, Lecture Notes Computer Science, 2006. This volume.
12. D. Lankford. On proving term rewriting systems are Noetherian. Technical Report MTP-3, Louisiana Technical University, 1979.
13. A. Middeldorp, H. Ohsaki, and H. Zantema. Transforming termination by self-labelling. In *Proceedings of the 13th International Conference on Automated Deduction*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 373–386, 1996.
14. Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.
15. R. Thiemann, J. Giesl, and P. Schneider-Kamp. Improved modular termination proofs using dependency pairs. In *Proceedings of the 2nd International Joint Conference on Automated Reasoning*, volume 3097 of *Lecture Notes in Artificial Intelligence*, pages 75–90, 2004.
16. Y. Toyama. Counterexamples to the termination for the direct sum of term rewriting systems. *Information Processing Letters*, 25:141–143, 1987.
17. X. Urbain. Modular & incremental automated termination proofs. *Journal of Automated Reasoning*, 32:315–355, 2004.
18. H. Zantema. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae*, 24:89–105, 1995.
19. H. Zantema. TORPA: Termination of rewriting proved automatically. In *Proceedings of the 15th International Conference on Rewriting Techniques and Applications*, Lecture Notes Computer Science, pages 95–104, 2004.