International Journal of Foundations of Computer Science

PROPERTIES OF QUASI-RELABELING TREE BIMORPHISMS

Andreas Maletti* and Cătălin Ionuţ Tîrnăucă
Universitat Rovira i Virgili
Departament de Filologies Romàniques
Research Group on Mathematical Linguistics
Avinguda de Catalunya 35, 43002 Tarragona, Spain
email: andreas.maletti@urv.cat
catalinionut.tirnauca@estudiants.urv.cat

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

The fundamental properties of the class QUASI of quasi-relabeling relations are investigated. A quasi-relabeling relation is a tree relation that is defined by a tree bimorphism (φ, L, ψ), where φ and ψ are quasi-relabeling tree homomorphisms and L is a regular tree language. Such relations admit a canonical representation, which immediately also yields that QUASI is closed under finite union. However, QUASI is not closed under intersection and complement. In addition, many standard relations on trees (e.g., branches, subtrees, v-product, v-quotient, and f-top-catenation) are not quasi-relabeling relations. If quasi-relabeling relations are considered as string relations (by taking the yields of the trees), then every Cartesian product of two context-free string languages is a quasi-relabeling relation. Finally, the connections between quasi-relabeling relations, alphabetic relations, and classes of tree relations defined by several types of top-down tree transducers are presented. These connections yield that quasi-relabeling relations preserve the regular and algebraic tree languages.

Keywords: regular tree language, tree homomorphism, tree bimorphism, tree transducer

1. Introduction

Tree relations were extensively study in the past four decades from the algebraic point of view offered by tree bimorphisms [3, 6, 25, 27, 28] or from the dynamic point of view provided by tree transducers [4, 9, 11, 19, 21]. Recently, new types of tree transducers were used with considerable success in modeling translations between natural languages especially because of their ability to capture syntax-sensitive transformations and complex reorderings of the syntax trees of sentences. Those tree transducers are now an essential device in the new field of syntax-based machine translation (see [12, 14, 15, 17] and the references therein). Unfortunately,

*This author was financially supported by the *Ministerio de Educación y Ciencia* (MEC) grant JDCI-2007-760.

properties that are essential for the translation process (e.g., closure under composition and preservation of recognizable and algebraic tree languages [14, 15]) do not hold in general for most of the main tree transducer types [5, 9, 11, 17], which shows that the added power comes with severe drawbacks.

Synchronous grammars [1, 22, 23, 24], which were first proposed as models of compilers [13], describe translations between natural languages in a very natural way. Such devices consist of two formal grammars, of which the productions are linked by some criteria. This link extends to the derivations and in this way related sentences are generated simultaneously. They can thus easily model syntax-sensitive transformations because their one-sided (say input) derivation can essentially be seen as a syntax tree of the sentence it generates. The links can communicate information about the shape of the input parse tree to the output side. However, this mechanism also limits their power because the shapes of both derivations should be similar. However, they can describe local rotations commonly used in phrase-based machine translation (a phrase is any part of the input sentence). Unfortunately, the mathematical framework offered by such formalisms is quite poor since, for example, no results for closure under composition were known until [23].

Tree bimorphisms offer an elegant algebraic way to define tree relations. A tree bimorphism is formed by two tree homomorphisms defined on the same common tree language. Tree bimorphisms were used with considerable success in proving properties like closure under composition and preservation of recognizability by imposing suitable restrictions on its constituents [3, 6, 25, 27, 28]. Moreover, by taking the yields of the input and output trees, they can be seen as devices that generate string relations. A survey on the main classes of tree bimorphisms and their characteristics is [20].

Using the tree bimorphism formalism, STUART SHIEBER was the first one who linked tree transducers and synchronous grammars in an attempt to improve the mathematical framework of the latter devices [23, p.95: "...the *bimorphism characterization* of tree transducers has led to a series of composition closure results. Similar techniques may now be *applicable* to synchronous formalisms, *where no composition results are known...*"]. Following this lead, the class of quasi-alphabetic tree bimorphisms (we call them quasi-relabeling in this paper) that define the same translations as syntax-directed translation schemata of [1] was introduced in [26].

It was already shown in [26] that the tree relations defined by quasi-relabeling (or quasi-alphabetic in [26]) tree bimorphisms, which are called *quasi-relabeling relations*, are closed under composition and inverses and preserve the recognizability of tree languages. In the present work we further investigate the properties of this class from a theoretical point of view. We are interested in its other closure properties, common operations that are preserved, canonical representations, and their place in the tree transducer hierarchy. In addition, we outline some properties of the string relations computed by them.

Our results can be summarized as follows. We show in Section 3 that there is a canonical representation of quasi-relabeling relations that allows us to prove that quasi-relabeling relations are closed under union. Unfortunately, but not surprisingly, we can also show that they are not closed under intersection and complement. We end this section with a result on string relations: every Cartesian product of two context-free languages is a quasi-relabeling string relation. This strengthens a result of [6], where it was shown for a more general class of bimorphisms. In Section 4 we investigate the connection of quasi-relabeling relations with other well-known classes of tree relations such as alphabetic relations [6], finite-state relabelings [9], tree relations defined by several types of top-down tree transducers [21, 29, 9] and top-down tree transducers with look-ahead [10]. All the results are depicted in the HASSE diagram in Figure 5, which also shows the relation between the corresponding classes of string relations. Moreover, as an immediate consequence of the fact that the class of quasi-relabeling relations is contained in the class of alphabetic relations, we obtain that quasi-relabeling relations preserves regular and algebraic tree languages.

2. Preliminaries

Let R, S, and T be sets, and consider a relation $\tau \subseteq S \times T$. The fact $(s,t) \in \tau$ can also be expressed by writing $s \;\tau\; t$. For every $s \in S$, let $s\tau = \{t \mid s \;\tau\; t\}$. More generally, for every $A \subseteq S$, we let $A\tau = \bigcup_{a \in A} a\tau$. The *inverse* of τ is the relation $\tau^{-1} = \{(t,s) \mid s \;\tau\; t\}$. The *composition* of two relations $\rho \subseteq R \times S$ and $\tau \subseteq S \times T$ is the relation $\rho \circ \tau = \{(r,t) \mid \exists s \in S \colon r \;\rho\; s \;\tau\; t\}$. The *identity relation* id_S is $\{(s,s) \mid s \in S\}$. For (total) mappings $\varphi \colon S \to T$ we generally identify $s\varphi$ and $\varphi(s)$ for every $s \in S$.

The nonnegative integers are denoted by $\mathbb{N}$. For every $k \in \mathbb{N}$, we write $[k]$ for the set $\{i \in \mathbb{N} \mid 1 \leq i \leq k\}$. For a set V, V^* is the set of *strings* over V with $\varepsilon \in V^*$ denoting the *empty string*. By an *alphabet* we mean a finite set of symbols. A *ranked alphabet* (Σ, rk) consists of an alphabet Σ and a mapping $\mathrm{rk} \colon \Sigma \to \mathbb{N}$. Often we leave the mapping rk implicit. For every $k \geq 0$, let $\Sigma_k = \{f \in \Sigma \mid \mathrm{rk}(f) = k\}$. We will write $\Sigma = \{f_1/k_1, \ldots, f_n/k_n\}$ to indicate that Σ consists of the symbols $f_1, \ldots, f_n$ with the respective ranks $k_1, \ldots, k_n$.

Let Σ be a ranked alphabet and T a set. Then

$$\Sigma(T) = \{f(t_1, \ldots, t_k) \mid f \in \Sigma_k \text{ and } t_1, \ldots, t_k \in T\} \ .$$

For every (leaf) alphabet V, the set $T_\Sigma(V)$ of all Σ-*trees indexed by* V is the smallest set T such that $V \subseteq T$ and $\Sigma(T) \subseteq T$. Subsets of $T_\Sigma(V)$ are called (*tree*) *languages*. Generally, for all considered trees we assume that the ranked alphabet is disjoint with the leaf alphabet. For every tree $t \in T_\Sigma(V)$, the set $\mathrm{pos}(t) \subseteq \mathbb{N}^*$ of *positions* of t is inductively given by $\mathrm{pos}(v) = \{\varepsilon\}$ for every $v \in V$, and

$$\mathrm{pos}(f(t_1, \ldots, t_k)) = \{\varepsilon\} \cup \{iw \mid i \in [k] \text{ and } w \in \mathrm{pos}(t_i)\}$$

for every $f \in \Sigma_k$ and $t_1, \ldots, t_k \in T_\Sigma(V)$. The *label* of t at position $w \in \mathrm{pos}(t)$ is denoted by $t(w)$, the *subtree* of t at w is denoted by $t|_w$, and the replacement of that subtree in t by the tree $u \in T_\Sigma(V)$ is denoted by $t[u]_w$. For every $\Omega \subseteq \Sigma \cup V$, let $\mathrm{pos}_\Omega(t) = \{w \in \mathrm{pos}(t) \mid t(w) \in \Omega\}$ and $\mathrm{pos}_f(t) = \mathrm{pos}_{\{f\}}(t)$ for every $f \in \Sigma \cup V$. For example, consider the tree $u \in T_\Sigma(V)$ of Figure 1 with $\Sigma = \{f/2, g/1, a/0\}$ and $V = \{v\}$. Then $\mathrm{pos}(u) = \{\varepsilon, 1, 2, 11, 21, 22\}$, the label at position 22 is $u(22) = v$, the subtree at 1 is $u|_1 = g(v)$, and $u[a]_2 = f(g(v), a)$. Finally, $\mathrm{pos}_f(u) = \{\varepsilon, 2\}$. The set of *branches* of t is $\mathrm{br}(t) = \mathrm{pos}_{\Sigma_0 \cup V}(t)$, and the set of subtrees of t is $\mathrm{sub}(t) = \{t|_w \mid w \in \mathrm{pos}(t)\}$. Finally, $|t|_f = \mathrm{card}(\mathrm{pos}_f(t))$, and the *height* of t is $\mathrm{hg}(t) = \max\{|w| \mid w \in \mathrm{br}(t)\}$. In other words, $|t|_f$ is the number of f-symbols in t, and $\mathrm{hg}(t)$ is the length of a branch of maximal length (among all branches). For the tree u of Figure 1 we have

$$\mathrm{br}(u) = \{11, 21, 22\}, \quad \mathrm{sub}(u) = \{v, a, g(v), f(a, v), u\}, \quad |u|_f = 2, \quad \mathrm{hg}(u) = 2 \enspace .$$

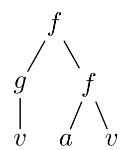

Figure 1: Example tree.

A tree $t \in T_\Sigma(V)$ is *linear* (respectively, *nondeleting*) in $Y \subseteq V$ if $|t|_y \leq 1$ (respectively, $|t|_y \geq 1$) for every $y \in Y$. The Y-*yield* of a tree $t \in T_\Sigma(V)$ is defined inductively by $\mathrm{yd}_Y(y) = y$ for every $y \in Y$, $\mathrm{yd}_Y(v) = \varepsilon$ for every $v \in V \setminus Y$, and $\mathrm{yd}_Y(f(t_1, \dots, t_k)) = \mathrm{yd}_Y(t_1) \cdots \mathrm{yd}_Y(t_k)$ for every $f \in \Sigma_k$ and $t_1, \dots, t_k \in T_\Sigma(V)$. Let $V = \{v\}$. Then the tree u of Figure 1 is not linear in V, but nondeleting in V. The V-yield of u is $\mathrm{yd}_V(u) = vv$.

We fix a set $X = \{x_i \mid i \geq 1\}$ of *formal variables* (disjoint to all other ranked alphabets and leaf alphabets). Let $n \geq 0$. We let $X_n = \{x_i \mid i \in [n]\}$ and

$$C^n_\Sigma(V) = \{t \in T_\Sigma(V \cup X_n) \mid \forall i \in [n] \colon |t|_{x_i} = 1\} \enspace .$$

In other words, $C^n_\Sigma(V)$ contains all those trees of $T_\Sigma(V \cup X_n)$ in which each variable $x_1, \dots, x_n$ occurs exactly once. For every $t \in T_\Sigma(V \cup X_n)$ and $f \in \Sigma_1$, we let $f^0(t) = t$ and $f^{k+1}(t) = f(f^k(t))$ for all $k \geq 0$.

For all $t, t_1, \dots, t_n \in T_\Sigma(V \cup X_n)$, we denote by $t[t_1, \dots, t_n]$ the result obtained by replacing, for every $i \in [n]$, every occurrence of x_i in t by t_i. For all $L, L_1, \dots, L_n \subseteq T_\Sigma(V \cup X_n)$, $L[L_1, \dots, L_n]$ denotes

$$\{t[t_1, \dots, t_n] \mid t \in L, t_1 \in L_1, \dots, t_n \in L_n\} \enspace .$$

Let $n = |t|_v$. More generally, for every $v \in V$, the result of replacing, for every $i \in [n]$, the i-th (with respect to the usual lexicographic order on the positions) occurrence of v by t_i is denoted by $t[v \leftarrow (t_1, \dots, t_n)]$. For every $f \in \Sigma_k$, the f-*top-catenation* of $L_1, \dots, L_k \subseteq T_\Sigma(V \cup X_n)$ is

$$f(L_1, \dots, L_k) = \{f(t_1, \dots, t_k) \mid t_1 \in L_1, \dots, t_k \in L_k\} \enspace .$$

Moreover for every $v \in V$, the v-*product* $L \bullet_v L'$ of two languages $L, L' \subseteq T_\Sigma(V)$ is

$$L \bullet_v L' = \{t[v \leftarrow (t_1, \dots, t_n)] \mid t \in L, n = |t|_v, \text{ and } t_1, \dots, t_n \in L'\} \enspace .$$

Then, the *v-quotient* of L by L' is $L /_v L' = \{t \in T_\Sigma(V) \mid (\{t\} \bullet_v L') \cap L \neq \emptyset\}$. For a more detailed description of these operations on tree languages, we refer the reader to [6] or [11].

Let us illustrate the previous notions on an example. Let $\Sigma = \{f/3, g/2, e/0\}$ and $V = \{v\}$, and consider the tree $t = f(g(x_2), v, f(e, v, x_1)) \in T_\Sigma(V \cup X_2)$ and two arbitrary trees $t_1, t_2 \in T_\Sigma(V)$. Then

$$\begin{aligned} t[t_1, t_2] &= f(g(t_2), v, f(e, v, t_1)) \\ t[v \leftarrow (t_1, t_2)] &= f(g(x_2), t_1, f(e, t_2, x_1)) \\ f(\{e, t_1\}, \{x_2\}, \{t_2\}) &= \{f(e, x_2, t_2), f(t_1, x_2, t_2)\} \ . \end{aligned}$$

Now let $L = \{v, f(v, e, f(e, v, e))\}$ and $L' = \{t_1, t_2\}$. Obviously, $L \bullet_v L'$ equals

$$L' \cup \{f(t_1, e, f(e, t_1, e)), f(t_1, e, f(e, t_2, e)), f(t_2, e, f(e, t_1, v)), f(t_2, e, f(e, t_2, e))\} \ ,$$

and if $t_1 = v$ and $t_2 = e$, then

$$\begin{aligned} L /_v L' = \{&f(v, v, f(v, v, v)), f(v, v, f(v, v, e)), f(v, v, f(e, v, v)), f(v, v, f(e, v, e))\} \\ &\cup \{f(v, e, f(v, v, v)), f(v, e, f(v, v, e)), f(v, e, f(e, v, v))\} \cup L \ . \end{aligned}$$

A (*tree*) *homomorphism* $\varphi\colon T_\Sigma(V) \to T_\Delta(Y)$ can be presented by a mapping $\varphi_V\colon V \to T_\Delta(Y)$ and mappings $\varphi_k\colon \Sigma_k \to T_\Delta(Y \cup X_k)$ for every $k \geq 0$ as follows:

(i) $v\varphi = \varphi_V(v)$ for every $v \in V$, and

(ii) $f(t_1, \ldots, t_k)\varphi = \varphi_k(f)[t_1\varphi, \ldots, t_k\varphi]$ for every $f \in \Sigma_k$ and $t_1, \ldots, t_k \in T_\Sigma(V)$.

We say that it is *normalized* if $\mathrm{yd}_X(\varphi_k(f)) = x_1 \cdots x_k$ for every $f \in \Sigma_k$. Moreover, such a homomorphism φ is

- *linear* [11, 6, 7] (respectively, *complete* [7]) if $\varphi_k(f)$ is linear (respectively, nondeleting) in X_k for every $f \in \Sigma_k$,
- *symbol-to-symbol* [7] if $\varphi_V(v) \in Y$ for every $v \in V$ and $\varphi_k(f) \in \Delta(X_k)$ for every $f \in \Sigma_k$,
- *alphabetic* [6, 2] (démarquage linéaire in [2]) if it is linear, $\varphi_V(v) \in Y$ for every $v \in V$, and $\varphi_k(f) \in X_k \cup \Delta(X_k)$ for every $f \in \Sigma_k$, and
- *strictly alphabetic* [6] if it is complete, alphabetic and symbol-to-symbol.

We denote by lH, cH, ssH, aH, and saH the classes of all linear, complete, symbol-to-symbol, alphabetic, and strictly alphabetic tree homomorphisms, respectively. Further subclasses of tree homomorphisms can be obtained by combining any of these restrictions. For example, lcH is the class of all linear complete tree homomorphisms.

Example 1 *Let* $\Sigma = \{f/3, g/2, e/0\}$*,* $V = \{v\}$*, and* $Y = \{0, 1\}$*. Consider the tree homomorphisms* $\varphi, \psi, \eta \colon T_\Sigma(V) \to T_\Sigma(Y)$ *defined by*

$$
\begin{array}{llll}
\varphi_3(f) = g(x_2, x_1) & \varphi_2(g) = g(x_1, x_1) & \varphi_0(e) = e & \varphi_V(v) = 0 \\
\psi_3(f) = x_2 & \psi_2(g) = g(x_2, x_1) & \psi_0(e) = e & \psi_V(v) = 1 \\
\eta_3(f) = f(x_3, x_1, x_2) & \eta_2(g) = g(x_1, x_2) & \eta_0(e) = e & \eta_V(v) = 1 \ .
\end{array}
$$

Then, φ *is symbol-to-symbol,* ψ *is alphabetic and* η *is strictly alphabetic. Moreover, note that* φ *is neither linear nor complete and* ψ *is not complete. For the tree* $t = f(e, g(v, e), v)$ *in* $T_\Sigma(V)$*, we get* $t\varphi = h(h(0, 0), d)$*,* $t\psi = h(d, 1)$ *and* $t\eta = f(1, d, h(1, d))$*. The input tree and the obtained trees are displayed in Figure 2.*

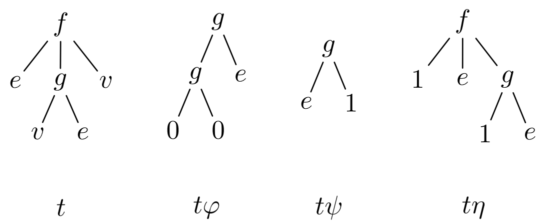

Figure 2: Application of tree homomorphisms (see Example 1).

A (*tree*) *bimorphism* is a triple $B = (\varphi, L, \psi)$, where $L \subseteq T_\Gamma(Z)$ is a tree language and $\varphi \colon T_\Gamma(Z) \to T_\Sigma(V)$ and $\psi \colon T_\Gamma(Z) \to T_\Delta(Y)$ are homomorphisms. The *tree relation defined by* B is $\tau_B = \varphi^{-1} \circ \mathrm{id}_L \circ \psi = \{(t\varphi, t\psi) \mid t \in L\}$, and the *translation defined by* B is

$$\mathrm{yd}(\tau_B) = \{(\mathrm{yd}_V(t\varphi), \mathrm{yd}_Y(t\psi)) \mid t \in L\} = \{(\mathrm{yd}_V(t), \mathrm{yd}_Y(u)) \mid (t, u) \in \tau_B\} \ .$$

For all classes $\mathcal{H}_1$ and $\mathcal{H}_2$ of homomorphisms and every class $\mathcal{L}$ of tree languages, we denote by $\mathcal{B}(\mathcal{H}_1, \mathcal{L}, \mathcal{H}_2)$ the class of tree relations τ_B, where $B = (\varphi, L, \psi)$ with $\varphi \in \mathcal{H}_1$, $L \in \mathcal{L}$, and $\psi \in \mathcal{H}_2$.

Example 2 *Let* $\Sigma = \{f/3, g/1\}$*,* $\Delta = \{h/1\}$*,* $V = \{v\}$*, and* $Y = \{1\}$*. Moreover, let*

$$L = \{f(g^l(v), g^m(v), g^n(v)) \mid l, m, n \in \mathbb{N}\} \subseteq T_\Sigma(V) \ ,$$

and let $\varphi \colon T_\Sigma(V) \to T_\Sigma(V)$ *and* $\psi \colon T_\Sigma(V) \to T_\Delta(Y)$ *be the homomorphisms given by*

$$
\begin{array}{lll}
\varphi_3(f) = f(x_3, x_2, x_1) & \varphi_1(g) = g(x_1) & \varphi_V(v) = v \\
\psi_3(f) = x_3 & \psi_1(g) = h(x_1) & \psi_V(v) = 1 \ .
\end{array}
$$

Then the tree relation defined by $B = (\varphi, L, \psi)$ *is*

$$\tau_B = \{(f(g^n(v), g^m(v), g^l(v)),\ h^n(1)) \mid l, m, n \in \mathbb{N}\}$$

with $\tau_B \in \mathcal{B}(\mathrm{saH}, \mathrm{Rec}, \mathrm{aH})$*, where* Rec *is the class of regular tree languages. The translation defined by* B *is* $\mathrm{yd}(\tau_B) = \{(vvv, 1)\}$*.*

A *top-down tree transducer* [21, 29] is a system $(Q, \Sigma, \Delta, I, R)$, where

- $Q = Q_1$ is a unary ranked alphabet of *states* disjoint with $\Sigma \cup \Delta$,
- Σ and Δ are an *input* and an *output alphabet*, respectively,
- $I \subseteq Q$ is a set of *initial states*, and
- R is a finite set of *rules* of the form $q(f(x_1, \ldots, x_k)) \to r$, where $q \in Q$, $f \in \Sigma_k$, and $r \in T_\Delta(Q(X_k))$.

The top-down tree transducer $M = (Q, \Sigma, \Delta, I, R)$ is *linear* (respectively, *nondeleting*) if r is linear (respectively, nondeleting) in X_k for every rule $q(f(x_1, \ldots, x_k)) \to r$ in R. The one-step *derivation relation* $\Rightarrow_M$ is defined as follows. For all sentential forms $\xi, \zeta \in T_\Delta(Q(T_\Sigma))$ we have $\xi \Rightarrow_M \zeta$ if and only if there exists a rule $q(f(x_1, \ldots, x_k)) \to r \in R$ and a position $w \in \mathrm{pos}(\xi)$ such that $\xi(w) = q$, $\xi(w1) = f$, and $\zeta = \xi[u]_w$, where $u = r[\xi|_{w11}, \ldots, \xi|_{w1k}]$. We illustrate a derivation step in Figure 3. Let $\Rightarrow_M^*$ be the reflexive and transitive closure of $\Rightarrow_M$. The *tree relation* computed by M is

$$\tau_M = \{(s, t) \in T_\Sigma \times T_\Delta \mid \exists q \in I \colon q(s) \Rightarrow_M^* t\} \ .$$

The class of all tree relations computable by linear (respectively, linear and nondeleting) top-down tree transducers is denoted by l-TOP (respectively, ln-TOP).

Example 3 *Consider the ranked alphabet $\Sigma = \{f/3, g/2, h/1, e/0\}$ and $Q = \{p, q\}$. Then $M = (Q, \Sigma, \Sigma, \{q\}, R)$ is a nondeleting, linear top-down tree transducer with the following rules*

$$\begin{aligned} q(f(x_1, x_2, x_3)) &\to g(g(q(x_1), p(x_2)), q(x_3)) & & \\ q(h(x_1)) &\to q(x_1) & \qquad q(e) &\to e \\ p(h(x_1)) &\to h(p(x_1)) & \qquad p(e) &\to e \ . \end{aligned}$$

Then $q(f(h^l(e), h^m(e), h^n(e))) \Rightarrow_M^* g(g(e, h^m(e)), e)$ for every $l, m, n \in \mathbb{N}$. The first rule and a derivation step involving that rule are illustrated in Figure 3.

Let $M = (Q, \Sigma, \Delta, I, R)$ be a top-down tree transducer. It is a *finite-state relabeling* [9], if for every rule $q(f(x_1, \ldots, x_k)) \to r \in R$ there exist $q_1, \ldots, q_k \in Q$ and $g \in \Delta_k$ such that $r = g(q_1(x_1), \ldots, q_k(x_k))$. If additionally, $f = g$ for all rules as in the previous sentence (i.e., $r(\varepsilon) = f$ for every $q(f(x_1, \ldots, x_k)) \to r \in R$), then M is a *finite-state tree automaton* (fta) [9]. We generally write rules of an fta in the form $q \to f(q_1, \ldots, q_k)$ instead of $q(f(x_1, \ldots, x_k)) \to f(q_1(x_1), \ldots, q_k(x_k))$. Note that τ_M coincides with id_L for some $L \subseteq T_\Sigma$ if M is an fta. This L is also denoted by $L(M)$, and additionally, for every $q \in Q$, the notation $L(M)_q$ stands for $L(N)$ where $N = (Q, \Sigma, \Delta, \{q\}, R)$. In other words, $L(M)_q$ is the language accepted by M if q were the only initial state. A language L is *regular* if there exists an fta M such that $L(M) = L$. The class of regular tree languages [11, Chapter II] is denoted by Rec, and $\mathrm{Rec}(\Sigma, V) = \{L \in \mathrm{Rec} \mid L \subseteq T_\Sigma(V)\}$. Finally, M is a *relabeling* [9] if it a finite-state relabeling and $\mathrm{card}(Q) = 1$. We denote the classes of tree relations

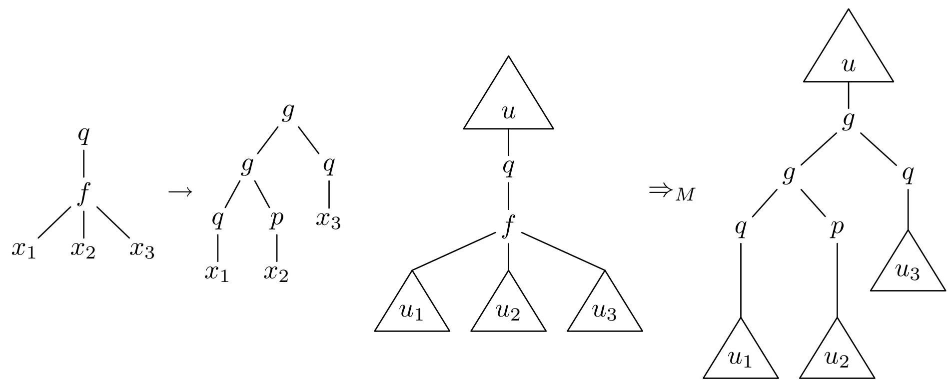

Figure 3: A sample rule and an illustration of a derivation step using that rule.

computed by finite-state relabelings, relabelings, and fta by QREL, REL, and FTA, respectively.

The top-down tree transducer M can be equipped with a look-ahead facility [10, 17]. The pair $\langle M, c\rangle$, where $M = (Q, \Sigma, \Delta, I, R)$ is a top-down tree transducer and $c \colon R \to \mathrm{Rec}(\Sigma)$, is called a *top-down tree transducer with regular look-ahead*. The look-ahead c is *finite* if $c(l \to r) \in L[T_\Sigma, \ldots, T_\Sigma]$ for a finite tree language $L \subseteq T_\Sigma(X)$. In the latter case, we often write $c(l \to r) = L$. Note that all finite tree languages are regular. The transducer $\langle M, c\rangle$ inherits the properties 'linear' and 'nondeleting' from M. The semantics of a top-down tree transducer $\langle M, c\rangle$ with look-ahead is defined as for the top-down tree transducer M with the additional condition that $\xi|_{w1} \in c(q(f(x_1, \ldots, x_k)) \to r)$ in the definition of $\Rightarrow_M$. The class of tree relations computed by linear top-down tree transducers with finite (respectively, regular) look-ahead is denoted by $\text{l-TOP}^{\mathrm{F}}$ (respectively, $\text{l-TOP}^{\mathrm{R}}$).

3. Properties of Quasi-Relabelings

Let us start by recalling the main notion of this contribution. A quasi-relabeling homomorphism is linear, complete, and basically symbol-to-symbol, but allows variables as successors of an output symbol. The precise definition from [26, Section 3] (where they are called 'quasi-alphabetic') follows.

Definition 1 *A tree homomorphism* $\varphi \colon T_\Sigma(V) \to T_\Delta(Y)$ *is a* quasi-relabeling *if*

(i) it is linear and complete,

(ii) $\varphi_V(v) \in Y$ *for every* $v \in V$*, and*

(iii) $\varphi_k(f) \in \Delta(Y \cup X_k)$ *for every* $f \in \Sigma_k$*.*

By qH *we denote the class of all quasi-relabelings. A* quasi-relabeling bimorphism *is a bimorphism* (φ, L, ψ) *such that* φ *and* ψ *are quasi-relabelings and* L *is regular.*

We introduce the new name 'quasi-relabeling' because their original name 'quasi-alphabetic' used in [26] clashes with the name 'alphabetic' used here and in [6]. In fact, using our terminology, quasi-relabelings are alphabetic (see Theorem 5), which is why we chose to rename them. Note that in [26] 'alphabetic' is used in the sense of [11] (a relabeling in our terminology).

Let $\mathrm{QUASI} = \mathcal{B}(\mathrm{qH}, \mathrm{Rec}, \mathrm{qH})$ and $\mathrm{ALPH} = \mathcal{B}(\mathrm{aH}, \mathrm{Rec}, \mathrm{aH})$ be the classes of all tree relations defined by quasi-relabeling bimorphisms and alphabetic bimorphisms, respectively. The elements of QUASI and ALPH are also called *quasi-relabeling* and alphabetic relations, respectively.

Example 4 *Let $\Gamma = \{f/3, g/1, e/0\}$, $\Sigma = \{h/6, g/1\}$, $\Delta = \{f/3, g/1\}$, $V = \{v\}$, and $Y = \{0, 1\}$. Consider the regular language*

$$L = \{f(g^m(v), e, g^n(v)) \mid m, n \in \mathbb{N}\} \subseteq T_\Gamma(V)$$

and the quasi-relabelings $\varphi\colon T_\Gamma(V) \to T_\Sigma(V)$ and $\psi\colon T_\Gamma(V) \to T_\Delta(Y)$ given by

$$\begin{array}{llll} \varphi_3(f) = h(v, x_2, x_3, v, v, x_1) & \varphi_1(g) = g(x_1) & \varphi_0(e) = v & \varphi_V(v) = v \\ \psi_3(f) = f(x_1, x_2, x_3) & \psi_1(g) = g(x_1) & \psi_0(e) = f(1,1,0) & \psi_V(v) = 1 \ . \end{array}$$

Then, $B = (\varphi, L, \psi)$ is a quasi-relabeling tree bimorphism, and

$$\tau_B = \{(h(v, v, g^n(v), v, v, g^m(v)),\ f(g^m(1), f(1,1,0), g^n(1))) \mid m, n \in \mathbb{N}\} \ .$$

Every quasi-relabeling maps each input symbol to an output symbol possibly with some output leaf variables as direct subtrees. However, the variables of X have to occur as direct subtrees of the root output symbol. This immediately yields the following proposition.

Proposition 1 *Let $\varphi\colon T_\Sigma(V) \to T_\Delta(Y)$ be a homomorphism and $t \in T_\Sigma(V)$.*

- *If φ is a quasi-relabeling, then* $\mathrm{hg}(t) \le \mathrm{hg}(t\varphi) \le \mathrm{hg}(t) + 1$.
- *If φ is symbol-to-symbol, then* $\mathrm{hg}(t\varphi) \le \mathrm{hg}(t)$.
- *If φ is strictly alphabetic, then* $\mathrm{hg}(t\varphi) = \mathrm{hg}(t)$.

Let us quickly recall some relevant existing results of [11].

Theorem 1 *Let $L_1, L_2 \in \mathrm{Rec}(\Sigma, V)$ and $\varphi\colon T_\Sigma(V) \to T_\Delta(Y)$ be a homomorphism.*

(i) Then $L_1 \cap L_2$, $L_1 \cup L_2$, and $L_1 \setminus L_2$ are regular.

(ii) If φ is linear, then $L_1\varphi$ is regular.

(iii) $L\varphi^{-1}$ is regular for every $L \in \mathrm{Rec}(\Delta, Y)$.

Proof. Items (i), (ii), and (iii) are proved in [11, Theorem II.4.2], [11, Theorem II.4.16], and [11, Theorem II.4.18], respectively. □

Now we investigate the fundamental properties of quasi-relabeling relations. We start our investigation with a canonical representation of quasi-relabeling relations in the spirit of [6, Proposition 3.1]. Note that our product data structure is simpler than the corresponding one of [6] (see Definition 2). The canonical representation will allow us to conclude that quasi-relabeling relations are closed under union.

For the rest of this section, let $B = (\varphi, L, \psi)$ with $\varphi\colon T_\Gamma(Z) \to T_\Sigma(V)$ and $\psi\colon T_\Gamma(Z) \to T_\Delta(Y)$ be a quasi-relabeling bimorphism. Let $[\Sigma \times \Delta]$ be the ranked alphabet such that for every $k \geq 0$

$$[\Sigma \times \Delta]_k = \{\langle t, u\rangle \mid t \in \Sigma(V \cup X_k) \cap C^k_\Sigma(V) \text{ and } u \in \Delta(Y \cup X_k) \cap C^k_\Delta(Y)\} \ .$$

In other words, our product data structure allows us to store $\langle \varphi_k(f), \psi_k(f)\rangle$ for some $f \in \Gamma_k$. Clearly, there are canonical quasi-relabelings

$$\pi^1\colon T_{[\Sigma\times\Delta]}(V \times Y) \to T_\Sigma(V) \qquad \text{and} \qquad \pi^2\colon T_{[\Sigma\times\Delta]}(V \times Y) \to T_\Delta(Y)$$

given by

$$\pi^1_{V\times Y}(\langle v, y\rangle) = v \qquad\qquad \pi^1_k(\langle t, u\rangle) = t$$
$$\pi^2_{V\times Y}(\langle v, y\rangle) = y \qquad\qquad \pi^2_k(\langle t, u\rangle) = u$$

for every $\langle v, y\rangle \in V \times Y$ and $\langle t, u\rangle \in [\Sigma \times \Delta]_k$. Henceforth, we will use these projections also for other product ranked alphabets. The next two statements were proved in [6, Proposition 3.1] for alphabetic relations.

Proposition 2 *There exists a quasi-relabeling $\eta\colon T_\Gamma(Z) \to T_{[\Sigma\times\Delta]}(V \times Y)$ such that $t\varphi = (t\eta)\pi^1$ and $t\psi = (t\eta)\pi^2$ for every $t \in T_\Gamma(Z)$.*

Proof. Let $\eta\colon T_\Gamma(Z) \to T_{[\Sigma\times\Delta]}(V \times Y)$ be the tree homomorphism such that $\eta_Z(z) = \langle \varphi_Z(z), \psi_Z(z)\rangle$ for every $z \in Z$ and $\eta_k(f) = \langle \varphi_k(f), \psi_k(f)\rangle$ for every $f \in \Gamma_k$. Clearly, η is a quasi-relabeling, and it is easy to check that $t\varphi = (t\eta)\pi^1$ and $t\psi = (t\eta)\pi^2$ for every $t \in T_\Gamma(Z)$. □

Using the previous proposition, we can now eliminate the ranked alphabet Γ, the index set Z, and the particular tree homomorphisms φ and ψ from the bimorphism B. Essentially, every quasi-relabeling relation $\tau \subseteq T_\Sigma(V) \times T_\Delta(Y)$ is determined by a regular language $L \in \mathrm{Rec}([\Sigma \times \Delta], V \times Y)$. The construction is illustrated in Figure 4.

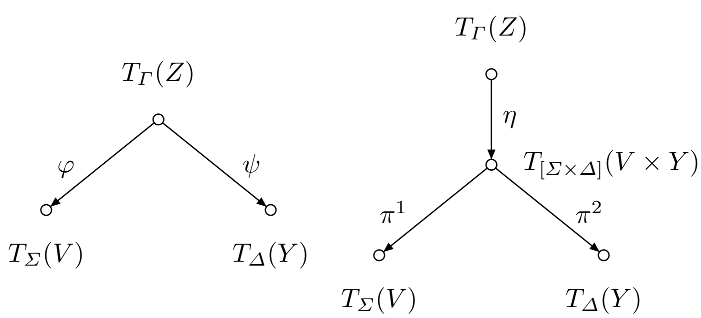

Figure 4: Illustration of the construction in Theorem 2.

Theorem 2 *A relation $\tau \subseteq T_\Sigma(V) \times T_\Delta(Y)$ is a quasi-relabeling relation if and only if there exists a regular $L \subseteq T_{[\Sigma\times\Delta]}(V \times Y)$ such that $\tau = \{(t\pi^1, t\pi^2 \mid t \in L\}$.*

Proof. The if-direction is trivial because (π^1, L, π^2) is a quasi-relabeling bimorphism defining τ. For the converse, let $B = (\varphi, L', \psi)$ be a quasi-relabeling bimorphism such that $\tau_B = \tau$. By Proposition 2 there exists a quasi-relabeling $\eta\colon T_\Gamma(Z) \to T_{[\Sigma\times\Delta]}(V \times Y)$ such that $\tau = \{(t\eta\pi^1, t\eta\pi^2) \mid t \in L'\}$. Consequently, the language $L = \eta(L')$ has the desired properties because it is regular by Theorem 1(ii). $\square$

We immediately note that quasi-relabeling relations are trivially closed under inverses [26, Theorem 4]; i.e., if $\tau \in$ QUASI, then so is τ^{-1}. As promised, let us use the previous theorem to prove that quasi-relabeling relations are closed under union. The corresponding property for alphabetic relations was proved in [6, Proposition 3.2].

Corollary 1 QUASI *is closed under union.*

Proof. Let $\tau_1, \tau_2 \subseteq T_\Sigma(V) \times T_\Delta(Y)$ be quasi-relabeling relations. By Theorem 2, there exist regular $L_1, L_2 \subseteq T_{[\Sigma\times\Omega]}(V \times Y)$ such that

$$\tau_1 = \{(t\pi^1, t\pi^2) \mid t \in L_1\} \quad \text{and} \quad \tau_2 = \{(t\pi^1, t\pi^2) \mid t \in L_2\} \ .$$

Then

$$\tau_1 \cup \tau_2 = \{(t\pi^1, t\pi^2) \mid t \in L_1\} \cup \{(t\pi^1, t\pi^2) \mid t \in L_2\} = \{(t\pi^1, t\pi^2) \mid t \in L_1 \cup L_2\} \ ,$$

which proves that $\tau_1 \cup \tau_2$ is a quasi-relabeling relation by Theorem 2 (because $L_1 \cup L_2$ is regular by item (i) of Theorem 1). $\square$

Let us move on to closure under intersection. For it we would need to align the two input homomorphisms and the two output homomorphisms at the same time and enforce equality both-sided. The next theorem shows that we are not able to achieve this and hence quasi-relabeling relations are not closed under intersection.

Theorem 3 *Any class $\mathcal{C}$ of tree relations such that*

$$\text{lcssH} \subseteq \mathcal{C} \subseteq \mathcal{B}(\text{H}, \text{Rec}, \text{lH})$$

is not closed under intersection.

Proof. Let $\Sigma = \{f/2, g/1, e/0\}$. We consider the linear complete symbol-to-symbol homomorphisms $\psi_1, \psi_2 \colon T_\Sigma \to T_\Sigma$ that are defined by

$$\begin{aligned} &\psi_1(f) = f(x_1, x_2) \qquad &&\psi_1(g) = g(x_1) \qquad &&\psi_1(e) = e \\ &\psi_2(f) = f(x_2, x_1) \qquad &&\psi_2(g) = g(x_1) \qquad &&\psi_2(e) = e \ . \end{aligned}$$

Clearly, ψ_1 and ψ_2 belong to $\mathcal{C}$. Note that $\psi_1 = \mathrm{id}_{T_\Sigma}$. We observe that for every $m, n \in \mathbb{N}$

$$\begin{aligned} f(g^m(e), g^n(e))\psi_1 &= f(g^m(e), g^n(e)) \\ f(g^m(e), g^n(e))\psi_2 &= f(g^n(e), g^m(e)) \ . \end{aligned}$$

Let $L = \{f(g^m(e), g^n(e)) \mid m, n \in \mathbb{N}\}$. Clearly, L is a regular language. Assume that there exists $\tau \in \mathcal{B}(\text{H}, \text{Rec}, \text{lH})$ such that $\tau = \psi_1 \cap \psi_2$. Since τ preserves regular languages by Theorem 1, the image $L\tau$ should be regular, but

$$L\tau = \{f(g^n(e), g^n(e)) \mid n \in \mathbb{N}\}$$

is not regular. Hence no τ with the given properties exists, which proves the statement. $\square$

Corollary 2 (of Theorem 3) QUASI *is not closed under intersection.*

Finally, we note that QUASI is trivially not closed under complement by Proposition 1. Now let us consider common relations on trees. We immediately observe that the intersection of a quasi-relabeling relation with id_L, where L is a regular language, is again a quasi-relabeling relation. Also the union with id_L is a quasi-relabeling relation because id_L is a quasi-relabeling relation for every regular language L and quasi-relabeling relations are closed under union by Corollary 1.

In general, the tree relations 'sub' and 'br' (if we consider the branches as trees over a ranked alphabet of symbols of rank 0 and 1) are not quasi-relabeling. Moreover, for regular $L \subseteq T_\Sigma(V)$ and $v \in V$, also the following relations τ and ρ, which are defined for every $t \in T_\Sigma(V)$ by $t\tau = t \bullet_v L$ and $t\rho = t /_v L$, are not quasi-relabeling relations, in general (in contrast to [6, Proposition 4.2 & p. 191–200] where it is shown that all those relations are alphabetic). All these can easily be proved using Proposition 1. Moreover, in general, quasi-relabeling relations are not closed under f-top-concatenation (again in contrast to alphabetic relations; see [6, Proposition 3.6]).

Now, let us turn our attention to the translations computed by quasi-relabeling bimorphisms. We call them *quasi-relabeling translations*. In [26] it was shown that they define the syntax-directed translations [1]. Here we add to this result that every Cartesian product of context-free string languages (for definitions and details about context-free string languages the reader is referred to [11, Section I.6]) is a quasi-relabeling translation. This sharpens [6, Proposition 3.6], where the same property was proved for alphabetic bimorphisms.

Theorem 4 *For all context-free string languages K_1 and K_2 over the same alphabet V, there exists a quasi-relabeling bimorphism B such that* $\mathrm{yd}(\tau_B) = K_1 \times K_2$.

Proof. By [11, Corollary 2.4], there exist regular tree languages $L_1 \subseteq T_\Sigma(V)$ and $L_2 \subseteq T_\Delta(V)$ such that $K_1 = \{\mathrm{yd}_V(t_1) \mid t_1 \in L_1\}$ and $K_2 = \{\mathrm{yd}_V(t_2) \mid t_2 \in L_2\}$. Let $\phi\colon Y \to V$ be a bijection, and Y be disjoint with $\Sigma \cup \Delta$. Then extend ϕ to $\phi_\Sigma\colon \Sigma \cup Y \to \Sigma \cup V$ and $\phi_\Delta\colon \Delta \cup Y \to \Delta \cup V$ such that $\phi_\Sigma(f) = f$ and $\phi_\Delta(g) = g$ for every $f \in \Sigma$ and $g \in \Delta$. We denote the ranked alphabets $\Sigma \cup Y$ and $\Delta \cup Y$, in which all symbols of Y are nullary, by $\bar{\Sigma}$ and $\bar{\Delta}$, respectively. Next, we define the ranked alphabet

$$\bar{\Sigma} \vee \bar{\Delta} = \{\langle f, g\rangle \mid f \in \bar{\Sigma}, g \in \bar{\Delta}\}$$

such that $\mathrm{rk}(\langle f, g\rangle) = \max(\mathrm{rk}(f), \mathrm{rk}(g))$. In a similar way the ranked alphabets $\Sigma \vee \bar{\Delta}$ and $\bar{\Sigma} \vee \Delta$ are defined. Without loss of generality, we can assume that $\bar{\Sigma}_0 \neq Y \neq \bar{\Delta}_0$ and $\Sigma_1 \neq \emptyset \neq \Delta_1$.

Next we show how to embed a tree of $T_\Sigma(V)$ into $T_{\bar{\Sigma} \vee \bar{\Delta}}$. Roughly speaking, we read off the first components of the symbols of $\bar{\Sigma} \vee \bar{\Delta}$ while neglecting the additional subtrees. However, we need to make sure that the neglected subtrees do not contain symbols of Y because a quasi-relabeling cannot ignore the additional subtrees, but should clearly not produce a piece of output string for them. To this end, we define the linear top-down tree transducer M_Σ with regular look-ahead c such that $M_\Sigma = (\{\star\}, \bar{\Sigma} \vee \bar{\Delta}, \Sigma \cup V, \{\star\}, R)$, and for every $\langle f, g\rangle \in (\bar{\Sigma} \vee \bar{\Delta})_k$ we have the rule

$$r = \star(\langle f, g\rangle(x_1, \dots, x_k)) \to \phi_\Sigma(f)(\star(x_1), \dots, \star(x_{\mathrm{rk}(f)}))$$

with look-ahead $c(r) = \langle f, g\rangle(T_1, \dots, T_k)$ in R, where $T_1 = \cdots = T_{\mathrm{rk}(f)} = T_{\bar{\Sigma}\vee\bar{\Delta}}$ and $T_{\mathrm{rk}(f)+1} = \cdots = T_k = T_{\Sigma\vee\bar{\Delta}}$. In an analogous way the top-down tree transducer M_Δ with regular look-ahead is defined. Let $L = \tau^{-1}_{M_\Sigma}(L_1) \cap \tau^{-1}_{M_\Delta}(L_2)$, which is regular by [11, Corollary IV.3.17] and Theorem 1. Next, we take the quasi-relabeling $\varphi\colon T_{\bar{\Sigma}\vee\bar{\Delta}} \to T_{\Sigma\vee\bar{\Delta}}(V)$, which is defined for every $\langle f, g\rangle \in (\bar{\Sigma} \vee \bar{\Delta})_k$ by

$$\varphi_k(\langle f, g\rangle) = \begin{cases} \langle f, g\rangle(x_1, \dots, x_m) & \text{if } f \in \Sigma_m \\ \langle h_1, h_2\rangle(\phi(f)) & \text{otherwise} \end{cases}$$

where $\langle h_1, h_2\rangle \in \Sigma_1 \times \Delta_1$ is arbitrary. In an analogous fashion, the quasi-relabeling $\psi\colon T_{\bar{\Sigma}\vee\bar{\Delta}} \to T_{\bar{\Sigma}\vee\Delta}(V)$ is defined. Now if we take the quasi-relabeling bimorphism $B = (\varphi, L, \psi)$, then it should be clear that $\mathrm{yd}_V(t\varphi) = \mathrm{yd}_V(t\tau_{M_\Sigma})$ and $\mathrm{yd}_V(t\psi) = \mathrm{yd}_V(t\tau_{M_\Delta})$ for every $t \in T_{\bar{\Sigma}\vee\bar{\Delta}}$. Consequently, $\mathrm{yd}(\tau_B) = K_1 \times K_2$, which concludes our proof. □

4. Relation to Other Classes

In this section, we relate the class of quasi-relabeling relations to other known classes of tree relations. We focus on classes of relations defined by bimorphisms [3, 8, 7] and classes of relations computed by various top-down tree transducers [21, 29, 11]. Recall that $\mathrm{QUASI} = \mathcal{B}(\mathrm{qH}, \mathrm{Rec}, \mathrm{qH})$ and $\mathrm{ALPH} = \mathcal{B}(\mathrm{aH}, \mathrm{Rec}, \mathrm{aH})$. Moreover, we let $\mathrm{SALPH} = \mathcal{B}(\mathrm{saH}, \mathrm{Rec}, \mathrm{saH})$. Clearly, every strictly alphabetic homomorphism is a quasi-relabeling and thus $\mathrm{SALPH} \subseteq \mathrm{QUASI}$. We start by showing that the class QREL of tree relations computed by finite-state relabellings [9] is included in the class SALPH.

Proposition 3 $\mathrm{QREL} \subseteq \mathrm{SALPH}$.

Proof. Let $\tau \in \mathrm{QREL}$. Since $\mathrm{QREL} \subseteq \text{ln-TOP} = \mathrm{REL} \circ \mathrm{FTA} \circ \mathrm{lcH}$ [9, Theorem 3.5], there exists a relabeling M such that $\tau_M \subseteq T_\Sigma(V) \times T_\Gamma(Z)$, a regular tree language $L \subseteq T_\Gamma(Z)$, and a linear and complete homomorphism $\psi\colon T_\Gamma(Z) \to T_\Delta(Y)$ such that $\tau = \{(t\tau_M^{-1}, t\psi) \mid t \in L\}$. Moreover, by the constructions of [9], ψ is symbol-to-symbol and $\tau_M^{-1}\colon T_\Gamma(Z) \to T_\Sigma(V)$ [i.e., τ_M^{-1} is computed by a deterministic relabeling]. Consequently, τ_M^{-1} and ψ are strictly alphabetic because every deterministic relabeling is strictly alphabetic. Thus, the strictly alphabetic bimorphism (τ_M^{-1}, L, ψ) defines τ. □

The next proposition shows that every quasi-relabeling relation can be computed by a linear top-down tree transducer with finite look-ahead [17]. With that we establish a rough upper bound to the power of quasi-relabeling bimorphisms.

Proposition 4 $\mathrm{QUASI} \subseteq \text{l-TOP}^{\mathrm{F}}$.

Proof. Let us consider a quasi-relabeling bimorphism $B = (\varphi, L, \psi)$, where $\varphi\colon T_\Gamma(Z) \to T_\Sigma(V)$ and $\psi\colon T_\Gamma(Z) \to T_\Delta(Y)$. Without loss of generality, let φ be normalized. Moreover, let $N = (Q, \Gamma \cup Z, \Gamma \cup Z, I, R)$ be an fta such that $L(N) = L$. We construct the linear top-down tree transducer M with finite look-ahead c such that $M = (Q, \Sigma \cup V, \Delta \cup Y, I, R')$ and

- for every transition $q \to z \in R$ with $z \in Z$, we have the rule $r = q(z\varphi) \to z\psi$ with look-ahead $c(r) = \{x_1\}$ in R', and
- for every transition $q \to f(q_1, \ldots, q_k) \in R$ with $f \in \Gamma_k$ and $q_1, \ldots, q_k \in Q$ we have the rule

  $$r = q(\varphi_k(f)(\varepsilon)(x_1, \ldots, x_n)) \to \psi_k(f)[q_1(x_{j_1}), \ldots, q_k(x_{j_k})]$$

  with look-ahead $c(r) = \{\varphi_k(f)\}$ in R', where $j_i = \mathrm{pos}_{x_i}(\varphi_k(f))$ for every $i \in [k]$.

First, let us prove $\tau_B \subseteq \tau_M$ by showing $q(t\varphi) \Rightarrow^*_M t\psi$ for every $q \in Q$ and $t \in L(N)_q$. Let $t \in Z$. Then $q(t\varphi) \Rightarrow_M t\psi$ using a rule constructed in the first item. Now let $t = f(t_1, \ldots, t_k)$ for some $f \in \Gamma_k$ and $t_1, \ldots, t_k \in T_\Gamma(Z)$. Moreover, let $q_1, \ldots, q_k \in Q$ be such that $t_i \in L(N)_{q_i}$ for every $i \in [k]$ and $q \to f(q_1, \ldots, q_k) \in R$. Then

$$\begin{aligned} q(f(t_1, \ldots, t_k)\varphi) &= q(\varphi_k(f)[t_1\varphi, \ldots, t_k\varphi]) \\ &= q(g(u_1[t_1\varphi, \ldots, t_k\varphi], \ldots, u_n[t_1\varphi, \ldots, t_k\varphi])) \end{aligned}$$

where $\varphi_k(f) = g(u_1, \ldots, u_n)$ for some $g \in \Sigma_n$ and $u_1, \ldots, u_n \in T_\Sigma(V)$. Let $j_i = \mathrm{pos}_{x_i}(\varphi_k(f))$ for every $i \in [k]$. Then

$$q(f(t_1, \ldots, t_k)\varphi) \Rightarrow_M \psi_k(f)[q_1(u_{j_1}[t_1\varphi, \ldots, t_k\varphi]), \ldots, q_k(u_{j_k}[t_1\varphi, \ldots, t_k\varphi])]$$

using a rule constructed in the second item. Note that the look-ahead restriction is trivially fulfilled. Clearly, $u_{j_i} = x_i$ for every $i \in [k]$ and thus we have

$$q(f(t_1, \ldots, t_k)\varphi) \Rightarrow_M \psi_k(f)[q_1(t_1\varphi), \ldots, q_k(t_k\varphi)] \ .$$

By the induction hypothesis, we have $q_i(t_i\varphi) \Rightarrow^*_M t_i\psi$ for every $i \in [k]$. Consequently, we obtain

$$q(t\varphi) \Rightarrow_M \psi_k(f)[q_1(t_1\varphi), \ldots, q_k(t_k\varphi)] \Rightarrow^*_M \psi_k(f)[t_1\psi, \ldots, t_k\psi] = t\psi \ .$$

This proves the auxiliary statement and $\tau_B \subseteq \tau_M$ if we consider states of I.

The converse inclusion can be proved using the statement: For every $q \in Q$, $t \in T_\Sigma(V)$, and $u \in T_\Delta(Y)$, if $q(t) \Rightarrow^*_M u$, then there exists $s \in L(N)_q$ such that $t = s\varphi$ and $u = s\psi$. This can be proved by induction on the length of the derivation in M. We omit the details here. □

Next let us show that the class of alphabetic relations is essentially different from the classes of tree relations computed by top-down tree transducers. For the specific class TOP this was already remarked in [6], and here we only refine their argument to the statements necessary for our purposes.

Proposition 5 ALPH $\not\subseteq$ l-TOP$^{\mathrm{R}}$ *and* ln-TOP $\not\subseteq$ ALPH.

Proof. It is known that l-TOP$^{\mathrm{R}} \subseteq$ BOT, where BOT is the class of all tree relations computable by bottom-up tree transducers [30, 9]. As claimed in [6, page 188], the class ALPH is incomparable to BOT. Consequently, ALPH $\not\subseteq$ l-TOP$^{\mathrm{R}}$. Moreover, it is known that lcH $\subseteq$ ln-TOP. Suppose that lcH $\subseteq$ ALPH. Then also every linear and complete inverse homomorphism is an alphabetic relation because alphabetic relations are trivially closed under inverses. However, the proof of the main theorem in [3, Section 3.4] would then show that alphabetic relations are not closed under composition. This contradicts [6, Theorem 5.2], thus lcH $\not\subseteq$ ALPH. This yields ln-TOP $\not\subseteq$ ALPH. $\square$

Next we consider the relation of quasi-relabeling and alphabetic relations. We show that every quasi-relabeling relation is also alphabetic. The strictness of this inclusion can be obtained using Proposition 5. For this result we need a product ranked alphabet of [6, Section 2].

Definition 2 *Let Σ and Δ be ranked alphabets, V and Y leaf alphabets, and $n \in \mathbb{N}$ be the minimal integer such that $\Sigma = \bigcup_{i=1}^{n} \Sigma_i$ and $\Delta = \bigcup_{i=1}^{n} \Delta_i$. We define the ranked alphabets $\Sigma^{[n]}$ and $\Delta^{[n]}$ such that for every $k \geq 1$*

$$\begin{aligned} \Sigma_0^{[n]} &= \Sigma_0 \qquad & \Sigma_k^{[n]} &= \{u \in \Sigma(X_k) \mid u \text{ linear in } X_k \text{ and } |u|_{x_k} = 1\} \cup \{k\} \\ \Delta_0^{[n]} &= \Delta_0 \qquad & \Delta_k^{[n]} &= \{u \in \Delta(X_k) \mid u \text{ linear in } X_k \text{ and } |u|_{x_k} = 1\} \cup \{k\} \ . \end{aligned}$$

The supremum *of Σ and Δ, denoted by $\Sigma \vee \Delta$, is defined for every $k \in \mathbb{N}$ by*

$$(\Sigma \vee \Delta)_k = \bigcup_{\max(l,m)=k} \Sigma_l^{[n]} \times \Delta_m^{[n]} \ .$$

Moreover, the two canonical alphabetic homomorphisms $\varphi^{\Sigma} \colon T_{\Sigma\vee\Delta}(V \times Y) \to T_{\Sigma}(V)$ and $\varphi^{\Delta} \colon T_{\Sigma\vee\Delta}(V \times Y) \to T_{\Delta}(Y)$ are defined by

$$\varphi^{\Sigma}_{V\times Y}(\langle v, y\rangle) = v \qquad \varphi^{\Sigma}_k(\langle t, u\rangle) = \begin{cases} x_k & \text{if } t = k \\ t & \text{otherwise} \end{cases}$$

$$\varphi^{\Delta}_{V\times Y}(\langle v, y\rangle) = y \qquad \varphi^{\Delta}_k(\langle t, u\rangle) = \begin{cases} x_k & \text{if } u = k \\ u & \text{otherwise} \end{cases}$$

for every $\langle v, y\rangle \in V \times Y$ and $\langle t, u\rangle \in (\Sigma \vee \Delta)_k$ with $k \in \mathbb{N}$.

Theorem 5 QUASI $\subseteq$ ALPH.

Proof. Let us take a quasi-relabeling tree bimorphism $B = (\varphi, L, \psi)$, where $\varphi \colon T_{\Gamma}(Z) \to T_{\Sigma}(V)$ and $\psi \colon T_{\Gamma}(Z) \to T_{\Delta}(Y)$. Without loss of generality, let $v \in V$ and $y \in Y$. We construct the linear homomorphism $\rho \colon T_{\Gamma}(Z) \to T_{\Sigma\vee\Delta}(V \times Y)$ such that $\rho_Z(z) = \langle z\varphi, z\psi\rangle$ for every $z \in Z$ and

$$\rho_k(f) = \langle t(\varepsilon)_w, u(\varepsilon)_{w'}\rangle(x_1, \dots, x_k, t_1, \dots, t_l)$$

for every $f \in \Gamma_k$ where

- $t = \varphi_k(f)$ and $u = \psi_k(f)$,

- $\{i_1, \ldots, i_m\} = \mathrm{pos}_V(t)$ and $\{j_1, \ldots, j_n\} = \mathrm{pos}_Y(u)$,
- $l = \max(m, n)$ and

$$t_a = \begin{cases} \langle t(i_a), u(j_a)\rangle & \text{if } a \leq \min(m, n) \\ \langle t(i_a), y\rangle & \text{if } n < a \leq m \\ \langle v, u(j_a)\rangle & \text{if } m < a \leq n \end{cases}$$

  for every $a \in [l]$, and

- $w = w_1 \cdots w_{k+m}$ and $w' = w'_1 \cdots w'_n$ are such that $t(w_a) = \rho_k(f)(a)\pi^1$ for every $a \in [k+m]$ and $t(w'_b) = \rho_k(f)(b)\pi^2$ for every $b \in [k+n]$ where π^1 and π^2 are the usual projections to the first and second components, respectively, with $x\pi^1 = x = x\pi^2$ for every $x \in X$.

By Theorem 1(ii), $L\rho$ is regular. An easy proof shows that

$$\tau_B = \{t\varphi^\Sigma, t\varphi^\Delta) \mid t \in L\rho\}$$

where φ^Σ and φ^Δ are the canonical alphabetic homomorphisms of Definition 2. Hence, τ_B is an alphabetic relation by [6, Proposition 3.1], which is the analogue of our Theorem 2 for alphabetic relations. □

As an immediate consequence of Theorem 5 and [6, Proposition 3.7], which proves that alphabetic relations preserve regular and algebraic tree languages, we get the following result.

Corollary 3 *Quasi-relabeling relations preserve the regular and the algebraic tree languages.*

Finally, we need to show that linear top-down tree transducers are not sufficiently powerful to implement all quasi-relabeling relations.

Proposition 6 QUASI $\not\subseteq$ l-TOP.

Proof. Let $\Sigma = \{f/2, e/0\}$ and $V = \{v_1, v_2\}$. Moreover, let $\varphi : T_\Sigma \to T_\Sigma(V)$ be a quasi-relabeling with $\varphi_0(e) = f(v_1, v_2)$. Then $B = (\varphi, \{e\}, \mathrm{id}_{T_\Sigma})$ is a quasi-relabeling tree bimorphism that defines $\{(f(v_1, v_2), e)\}$. It is known [9, Example 2.6] that τ_B is not in l-TOP, and hence QUASI $\not\subseteq$ l-TOP. □

Let us collect our results in a Hasse diagram (see Figure 5). Note that in such a diagram every edge is oriented upwards and denotes strict inclusion. We also add the corresponding classes of translations, which we denote by $\mathrm{yd}(\mathcal{C})$ if $\mathcal{C}$ is the class of tree relations.

Theorem 6 *Figure 5 is a* Hasse *diagram.*

Proof. The following six statements are sufficient to prove the claims of the left

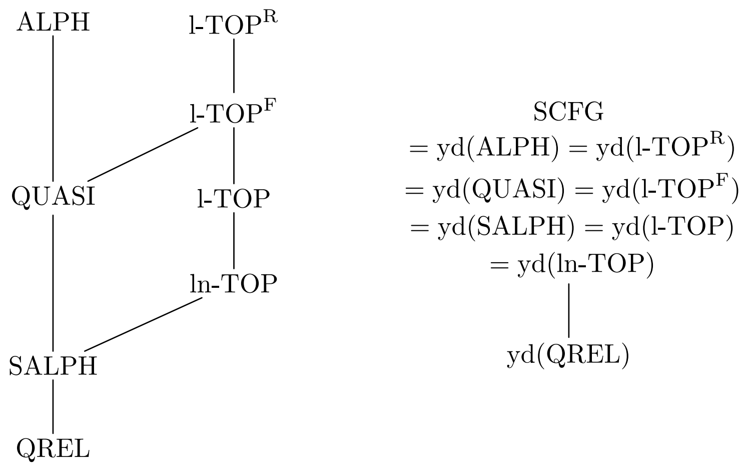

Figure 5: Hasse diagram of classes of tree relations (left) and corresponding string relations (right).

diagram.

$$\mathrm{QREL} \subset \mathrm{SALPH} \subseteq \mathrm{QUASI} \subseteq \mathrm{ALPH} \tag{1}$$
$$\mathrm{SALPH} \subseteq \text{ln-TOP} \subset \text{l-TOP} \subseteq \text{l-TOP}^{\mathrm{F}} \subseteq \text{l-TOP}^{\mathrm{R}} \tag{2}$$
$$\mathrm{QUASI} \subseteq \text{l-TOP}^{\mathrm{F}} \tag{3}$$
$$\mathrm{QUASI} \not\subseteq \text{l-TOP} \tag{4}$$
$$\text{ln-TOP} \not\subseteq \mathrm{ALPH} \tag{5}$$
$$\mathrm{ALPH} \not\subseteq \text{l-TOP}^{\mathrm{R}} \tag{6}$$

Statement 1 is clear using Proposition 3. The strictness is due to the fact that QREL is closed under intersection whereas this is not true for SALPH by Theorem 3. The final inclusion of (1) is proved in Theorem 5. The inclusions of (2) are all obvious and (3) is shown in Proposition 4. Finally, the inequality (4) is proved in Proposition 6 and inequalities (5) and (6) are proved in Proposition 5.

It is proved in [18, Theorems 3 and 7] that $\mathrm{yd}(\mathrm{QREL}) \subsetneq \mathrm{yd}(\mathrm{SALPH}) = \mathrm{SCFG}$, where SCFG denotes the class of string translations computed by synchronous context-free grammars (or equivalently, syntax-directed translation schemas) [1]. Moreover, [26, Theorem 1] proves that $\mathrm{SCFG} = \mathrm{yd}(\mathrm{QUASI})$. To prove that the remaining classes also collapse to SCFG, we prove that for every $\tau \in \mathcal{B}(\mathrm{lH}, \mathrm{Rec}, \mathrm{lH})$ we can construct a quasi-relabeling bimorphism B such that $\mathrm{yd}(\tau_B) = \mathrm{yd}(\tau)$. It is clear that alphabetic bimorphisms are linear and $\text{l-TOP}^{\mathrm{R}} \subseteq \mathcal{B}(\mathrm{lH}, \mathrm{Rec}, \mathrm{lH})$ by [16, Theorem 4]. To this end, we first prove that $\mathrm{yd}(\tau) \in \mathrm{yd}(\mathcal{B}(\mathrm{lcH}, \mathrm{Rec}, \mathrm{lcH}))$ using a construction that is similar to the one in the proof of Theorem 4 and [18, Lemma 9] (eliminating variables in the center tree language and turning the homomorphisms into complete ones such that no variables are output for the subtrees that were deleted by the original homomorphisms). Next we flatten the output trees. Let $B' = (\varphi, L, \psi)$ be a linear, complete bimorphism such that $L \subseteq T_\Gamma$

and $\varphi\colon T_\Gamma \to T_\Sigma(V)$ and $\psi\colon T_\Gamma \to T_\Delta(Y)$. Then we construct quasi-relabelings φ' and ψ' for every $f \in \Gamma_k$ by

$$\varphi'_k(f) = g(t_1, \dots, t_n) \quad \text{and} \quad \psi'_k(f) = g'(u_1, \dots, u_{n'}) \enspace ,$$

where g and g' are new output symbols. In addition, $t_1, \dots, t_n \in V \cup X$ and $u_1, \dots, u_{n'} \in Y \cup X$ are such that

$$\mathrm{yd}_{V \cup X}(\varphi_k(f)) = t_1 \cdots t_n \quad \text{and} \quad \mathrm{yd}_{Y \cup X}(\psi_k(f)) = u_1 \cdots u_{n'} \enspace .$$

Now let $B'' = (\varphi', L, \psi')$. It should be clear that $\mathrm{yd}(\tau_{B''}) = \mathrm{yd}(\tau_{B'})$, which proves the statement because $\tau_{B''} \in \mathrm{QUASI}$. □

Acknowledgements

The authors are grateful to the reviewer for pointing out some shortcomings of the draft. The authors also acknowledge the financial support by the ESF program "AutoMathA: from Mathematics to Applications."

References

1. A. V. Aho and J. D. Ullman, *Parsing*, vol. 1 of *The Theory of Parsing, Translation, and Compiling* (Prentice Hall, 1972).
2. A. Arnold and M. Dauchet, "Forêts algébriques et homomorphismes inverses," *Inform. Control* **37** (1978) 182–196.
3. A. Arnold and M. Dauchet, "Morphismes et bimorphismes d'arbres," *Theor. Comput. Sci.* **20** (1982) 33–93.
4. B. S. Baker, "Tree transducers and tree languages," *Inform. Control* **37** (1978) 241–266.
5. B. S. Baker, "Composition of top-down and bottom-up tree transductions," *Inform. Control* **41** (1979) 186–213.
6. S. Bozapalidis, "Alphabetic tree relations," *Theor. Comput. Sci.* **99** (1992) 177–211.
7. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, C. Löding, S. Tison and M. Tommasi, "Tree automata—techniques and applications," Available at `http://tata.gforge.inria.fr/`, 2007.
8. M. Dauchet and S. Tison, "Structural complexity of classes of tree languages," in *Tree Automata and Languages* (North-Holland, 1992), pp. 327–354.
9. J. Engelfriet, "Bottom-up and top-down tree transformations: A comparison," *Math. Syst. Theory* **9** (1975) 198–231.
10. J. Engelfriet, "Top-down tree transducers with regular look-ahead," *Math. Syst. Theory* **10** (1977) 289–303.
11. F. Gécseg and M. Steinby, *Tree Automata* (Akadémiai Kiadó, Budapest, 1984).
12. J. Graehl, K. Knight and J. May, "Training tree transducers," *Comput. Ling.* **34** (2008) 391–427.
13. E. T. Irons, "A syntax directed compiler for ALGOL 60," *Comm. ACM* **4** (1961) 51–55.
14. K. Knight, "Capturing practical natural language transformations," *Machine Learning* **21** (2007) 121–133.

15. K. Knight and J. Graehl, "An overview of probabilistic tree transducers for natural language processing," in *Proc. CICLing* (Springer, 2005), vol. 3406 of LNCS, pp. 1–24.
16. A. Maletti, "Compositions of extended top-down tree transducers," *Inf. Comput.* **206** (2008) 1187–1196.
17. A. Maletti, J. Graehl, M. Hopkins and K. Knight, "The power of extended top-down tree transducers," *SIAM J. Comput.* **39** (2008) 410–430.
18. A. Maletti and C. I. Tîrnăucă, "Syntax-directed translations and quasi-alphabetic tree bimorphisms — revisited," in *Proc. 3rd Int. Conf. Algebraic Informatics* (Springer, 2009), vol. 5725 of LNCS.
19. G. Rahonis, "Alphabetic and synchronized tree transducers," *Theor. Comput. Sci.* **255** (2001) 377–399.
20. J.-C. Raoult, "A survey of tree transductions," in *Tree Automata and Languages* (North-Holland, 1992), pp. 311–326.
21. W. C. Rounds, "Mappings and grammars on trees," *Math. Syst. Theory* **4** (1970) 257–287.
22. G. Satta and E. Peserico, "Some computational complexity results for synchronous context-free grammars," in *Proc. HLT/EMNLP* (Association for Computational Linguistics, 2005), pp. 803–810.
23. S. M. Shieber, "Synchronous grammars as tree transducers," in *Proc. TAG+7* (2004), pp. 88–95.
24. S. M. Shieber and Y. Schabes, "Synchronous tree-adjoining grammars," in *Proc. COLING* (Association for Computational Linguistics, 1990), pp. 253–258.
25. M. Steinby, "On certain algebraically defined tree transformations," in *Algebra, Combinatorics and Logic in Computer Science* (North-Holland, 1986), vol. 42 of Colloquia Mathematica Societatis János Bolyai, pp. 745–764.
26. M. Steinby and C. I. Tîrnăucă, "Syntax-directed translations and quasi-alphabetic tree bimorphisms," in *Proc. CIAA* (Springer, 2007), vol. 4783 of LNCS, pp. 265–276.
27. M. Takahashi, "Primitive transformations of regular sets and recognizable sets," in *Proc. ICALP* (North-Holland, 1972), pp. 475–480.
28. M. Takahashi, "Rational relations of binary trees," in *Proc. ICALP* (Springer, 1977), vol. 52 of LNCS, pp. 524–538.
29. J. W. Thatcher, "Generalized^2 sequential machine maps," *J. Comput. Syst. Sci.* **4** (1970) 339–367.
30. J. W. Thatcher, "Tree automata: An informal survey," in *Currents in the Theory of Computing* (Prentice Hall, 1973), pp. 143–172.