To appear: *Theoretical Computer Science*, 1996.

Structural Equivalence and ET0L Grammars[§]

Kai Salomaa[*], Derick Wood[†], and Sheng Yu[‡]

Technical Report HKUST-CS95-44
September 1995

[*]Department of Mathematics
University of Turku
Turku
Finland

[†]Department of Computer Science
Hong Kong University of Science & Technology
Clear Water Bay, Kowloon
Hong Kong

[‡]Department of Computer Science
University of Western Ontario
London, Ontario
Canada

Abstract

For a given context-sensitive grammar G we construct ET0L grammars G_1 and G_2 that are structurally equivalent if and only if the language generated by G is empty, which implies that structural equivalence is undecidable for ET0L grammars. In contrast, structural equivalence is decidable for E0L grammars and for extended E0L grammars. In fact, we show that structural equivalence is undecidable for propagating ET0L grammars in which the number of tables is restricted to be at most two. A stronger notion of equivalence that requires the sets of syntax trees to be isomorphic is shown to be decidable for ET0L grammars.

[§]This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada and from the Information Technology Research Centre of Ontario.

Structural Equivalence and ET0L Grammars[1]

Kai Salomaa[2] Derick Wood[3] Sheng Yu[4]

September 12, 1995

Abstract

For a given context-sensitive grammar G we construct ET0L grammars G_1 and G_2 that are structurally equivalent if and only if the language generated by G is empty, which implies that structural equivalence is undecidable for ET0L grammars. In contrast, structural equivalence is decidable for E0L grammars and for extended E0L grammars. In fact, we show that structural equivalence is undecidable for propagating ET0L grammars in which the number of tables is restricted to be at most two. A stronger notion of equivalence that requires the sets of syntax trees to be isomorphic is shown to be decidable for ET0L grammars.

1 Introduction

When considering various devices such as grammars and automata for defining languages, a central question is to determine whether two such devices are equivalent; that is, whether they generate (or recognize) the same language. It is well known that language equivalence is undecidable for context-free and E0L grammars, since two grammars may, in general, be language equivalent although the derivations of a given sentence are completely different. When dealing with sequential or parallel context-free grammars, we may consider the notion of structural equivalence, also known as strong equivalence, instead of language equivalence. Two grammars are structurally equivalent if the structures of the syntax trees that correspond to each sentence are the same. We define the structure of a syntax tree as the tree that is obtained by deleting the nonterminals that label internal nodes. An even stronger notion of equivalence, which we call syntax equivalence, requires that the sets of syntax trees are identical modulo a renaming of the nonterminal symbols.

Paull and Unger [9], and McNaughton [5] showed that structural equivalence of context-free grammars is decidable. Thatcher [15, 16] gave a considerably simpler proof of decidability by reducing it to the emptiness problem of finite-state tree automata. Ginsburg and Harrison [2] established the decidability of a more restricted problem, namely, they encoded the syntax trees of a

[1]This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada and by a grant from the Information Technology Research Centre of Ontario.

[2]Department of Mathematics, University of Turku, Turku, Finland.

[3]Contact author: Department of Computer Science, Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong.

[4]Department of Computer Science, University of Western Ontario, London, Ontario, Canada N6A 5B7.

context-free grammar as a bracketed context-free language and showed that equivalence of bracketed languages is decidable. Thatcher [15] also established decidability of structural equivalence for extended context-free grammars ("context-free grammars" that have productions with regular right-hand sides), and Cameron and Wood [1, 18] give a grammatical proof of decidability that is similar to McNaughton's proof for context-free grammars as expounded by Salomaa [12].

The question of structural equivalence for E0L grammars was first raised by Ottmann and Wood [7, 8], where they also obtained partial decidability results for certain restricted types of grammars. E0L structural equivalence was shown to be decidable by Salomaa and Yu [14] using the automata-theoretic approach of Thatcher [15, 16]. The same proof can be used to see that syntax equivalence of E0L grammars is also decidable. An alternative grammatical proof for the decidability of E0L structural equivalence was given by Niemi [6] based on the approach of Ottmann and Wood [8]. The grammatical proof is more complicated (as in the case of context-free grammars), but it has the advantage that it produces for a given E0L grammar a structurally equivalent normal form such that two E0L grammars in the normal form are structurally equivalent if and only if they are isomorphic. This decidability result has been extended by Cameron and Wood [1, 18] to extended E0L grammars ("E0L grammars" that have productions with regular right-hand sides). The complexity of the E0L structural equivalence problem has been studied by Salomaa *et al.* [13]. Recently, Istrate [3] has shown that structural equivalence of ET0L grammars is decidable when we require that corresponding syntax trees use isomorphic sequences of tables. The decidability of structural equivalence for T0L and EDT0L grammars remains open. We conjecture that T0L structural equivalence is decidable since every level in two structurally equivalent syntax trees must give identical terminal strings. We cannot relabel the internal nodes of a syntax tree as we do for ET0L syntax trees. It was noted by Salomaa and Yu [14] that structural equivalence is undecidable for indexed grammars.

Here we show that structural equivalence is undecidable for ET0L grammars. More specifically, structural equivalence is already undecidable for a propagating E0L grammar and a propagating ET0L grammar. Furthermore, the number of tables in the ET0L grammar can be restricted to two. In contrast we show that syntax equivalence is decidable for ET0L grammars. These results demonstrate that the notions of structural and syntax equivalence are essentially different.

The proof of undecidability uses a reduction from the emptiness problem for context-sensitive languages, which is a well known undecidable problem [12, 17]. For a given context-sensitive grammar G, we construct ET0L grammars G_1 and G_2 that are structurally equivalent if and only if $L(G) = \emptyset$. The construction is considerably simplified by the use of a normal form for context-sensitive grammars in which the productions have only one-sided context that was established by Penttonen [10].

Intuitively, the grammar G_1 simulates the context-sensitive grammar G by ignoring the context conditions. For technical reasons we add new branches to the derivations in G_1 that we use to guarantee that the syntax trees are in one-to-one correspondence with their structures. The grammar G_2 simulates G in a similar way but, in addition, it nondeterministically verifies that the context conditions of G are violated at least once. Intuitively, G_2 uses a context-sensitive production once, which can be accomplished by sending, nondeterministically, messages down the

syntax tree. The choice of table that is used to delete the messages gives the necessary context information for the context-sensitive derivation step.

2 Preliminaries

We assume that the readers are familiar with the basics of formal language theory and with ET0L grammars in particular [11, 12, 17]. In the following, we review the definitions of the syntax trees of ET0L grammars.

Let A be a finite set. The cardinality of A is denoted $\#A$ and the power set of A is $\mathcal{P}(A)$. The family of finite multisets over A is $\mathcal{M}(A)$. A multiset is denoted by listing its elements in double braces. Let $a \in A$ and $B = \{\{b_1, \ldots, b_m\}\} \in \mathcal{M}(A)$; then, $\#_a[B]$ denotes the number of occurrences of a in the sequence $b_1, \ldots, b_m$. Also,

$$\mathrm{base}(B) = \{a \in A \ : \ \#_a[B] \geq 1\}.$$

The set of finite strings (resp. nonempty finite strings) over A is A^* (resp. A^+). The empty string is denoted by λ. For $a \in A$ and $w \in A^*$, $\#_a(w)$ denotes the number of occurrences of a in the string w. Also we define

$$\mathrm{alph}(w) = \{a \in A \ : \ \#_a(w) \geq 1\}.$$

An *ET0L grammar* G is specified by a tuple

$$G = (V, \Sigma, S, H), \tag{1}$$

where V is a finite alphabet of nonterminals, Σ is a finite alphabet of terminals, $S \in V$ is the initial nonterminal, and H is a finite set of tables of productions from V to finite subsets of $(V \cup \Sigma)^*$. We define a table $h \in H$ as a finite set of productions $a \to w$, where $a \in V$ and $w \in (V \cup \Sigma)^*$. A grammar G is an E0L grammar if it has only one table, that is, $\#H = 1$. We say that a grammar is propagating if the right side of every production is nonempty, that is, for all $h \in H$, $a \in V$: $(a \to \lambda) \notin h$. Propagating ET0L and E0L grammars are called EPT0L and EP0L grammars, respectively. Although we can restrict our attention to EPT0L grammars for the undecidability of ET0L structural equivalence, for the *decidability* of syntax equivalence we need to deal with ET0L grammars to obtain the strongest result.

In the following, G denotes an ET0L grammar as in (1). Let F_G be the set of all rooted ordered trees where the nodes are labeled by elements of $V \cup \Sigma \cup \{\hat{\lambda}\}$. Here $\hat{\lambda}$ is a new symbol corresponding to the empty string λ. The set of nodes of $T \in F_G$ is denoted as $\mathrm{dom}(T)$, the domain of T. The label function associating an element of $V \cup \Sigma \cup \{\hat{\lambda}\}$ to each node of T is denoted by $\mathrm{lab}_T : \mathrm{dom}(T) \to V \cup \Sigma \cup \{\hat{\lambda}\}$.

We define the parallel derivation relation $\to_G^{\mathrm{par}} \subseteq F_G \times F_G$ as follows. Let $T, T' \in F_G$. Then $T \to_G^{\mathrm{par}} T'$ if and only if T' is obtained from T as follows. Assume that T has n external nodes $u_1, \ldots, u_n$ where $\mathrm{lab}_T(u_i) = a_i \ \in V \cup \{\hat{\lambda}\}$, $i = 1, \ldots, n$. Consider a table $h \in H$ and for every $i \in \{1, \ldots, n\}$ such that $a_i \neq \hat{\lambda}$ choose a production

$$a_i \to b_1^i \cdots b_{k_i}^i \in h,$$

$b^i_j \in V \cup \Sigma$, $j = 1, \ldots k_i$, $k_i \geq 0$. If $a_i \neq \hat{\lambda}$ and $k_i \geq 1$, then in T' the node u_i has k_i successors labeled respectively by the symbols $b^i_1, \ldots, b^i_{k_i}$. If $a_i \neq \hat{\lambda}$ and $k_i = 0$, then the node u_i has exactly one successor labeled by the symbol $\hat{\lambda}$. If $a_i = \hat{\lambda}$, then u_i has no successors in T'.

The set of syntax trees $S(G)$ of an ET0L grammar G is defined by

$$S(G) = \{T \in F_G \ : \ S' \ (\rightarrow^{\text{par}}_G)^* \ T\},$$

where S' is the tree with a single node labeled by S.

In a syntax tree T all paths from the root to an external node labeled by an element of $V \cup \Sigma$ have the same length. Note that a path from the root to a node labeled with $\hat{\lambda}$ need not be the same length as the paths from the root to nodes labeled with elements of $V \cup \Sigma$. In an EPT0L grammar, however, all root-to-external-node paths as the same length. According to our definition, if a tree $T \in S(G)$ has an external node labeled by a terminal symbol, then the derivation cannot be continued from T, that is, G is synchronized, see the text of Rozenberg and Salomaa [11]. We observe that the assumption of syncronization does not affect our results. Given a nonsynchronized ET0L grammar, we can convert it into an equivalent synchronized ET0L grammar by introducing a new nonterminal symbol, a nonterminal partner, for each terminal symbol in the grammar. Next, we replace every appearance of a terminal symbol in the productions of the grammar with its partner and, finally, add to each table a production that rewrites each nonterminal partner as its corresponding terminal symbol. Clearly, every terminal syntax tree in the nonsynchronized grammar corresponds to a terminal syntax tree in the synchronized grammar that has an extra last level that uses the added productions and conversely. Thus, based on this construction, two nonsynchronized ET0L grammars are structurally equivalent if and only if their synchronized versions are structurally equivalent. Alternatively, since nonsynchronized ET0L grammars are a wider class than synchronized grammars, the undecidablity carries over directly.

A syntax tree $T \in S(G)$ is *terminal* if all external nodes of T are labeled by elements of $\Sigma \cup \{\hat{\lambda}\}$. The set of terminal syntax trees of G is denoted by $TS(G)$.

Let $e_\lambda : (V \cup \Sigma \cup \{\hat{\lambda}\})^* \rightarrow (V \cup \Sigma)^*$ be the morphism defined by the conditions $e_\lambda(a) = a$ if $a \in V \cup \Sigma$ and $e_\lambda(\hat{\lambda}) = \lambda$. For $T \in F_G$ denote by w_T $(\in (V \cup \Sigma \cup \{\hat{\lambda}\})^+)$ the string obtained by catenating from left to right the symbols labeling the external nodes of T. The *yield* of T is defined as

$$\text{yield}(T) = e_\lambda(w_T).$$

The set of *sentential forms* of an ET0L grammar G is

$$\text{sf}(G) = \{\text{yield}(T) \ : \ T \in S(G)\}.$$

The relation $\rightarrow^{\text{par}}_G$ determines a parallel derivation relation $\Rightarrow^{\text{par}}_G$ on $(V \cup \Sigma)^*$ as follows. Let $w_1, w_2 \in (V \cup \Sigma)^*$. Then $w_1 \Rightarrow^{\text{par}}_G w_2$ if and only if there exist $T_i \in F_G$, $i = 1, 2$, with $\text{yield}(T_i) = w_i$ such that $T_1 \rightarrow^{\text{par}}_G T_2$. The language generated by G is

$$L(G) = \text{sf}(G) \cap \Sigma^* = \{w \in \Sigma^* \ : \ S \ (\Rightarrow^{\text{par}}_G)^* \ w\}.$$

Clearly the preceding definition of $L(G)$ is equivalent to the standard definition of the language generated by an ET0L grammar given by Rozenberg and Salomaa [11].

Let $T \in TS(G)$. The *structure of the terminal syntax tree* T, $\mathrm{str}_G(T)$, is the external-node-labeled tree obtained from T by removing the labels of all internal nodes of T (that is, nodes that are not external). Formally, $\mathrm{str}_G(T) = T'$ can be defined as follows. Let c be a new symbol not in $V \cup \Sigma$. Then $\mathrm{dom}(T') = \mathrm{dom}(T)$, $\mathrm{lab}_{T'}(u) = \mathrm{lab}_T(u)$ if u is an external node of T and $\mathrm{lab}_{T'}(u) = c$ if u is an internal node of T. We denote

$$STS(G) = \{\mathrm{str}_G(T) \ : \ T \in TS(G)\}.$$

Now we can define the various notions of equivalence of grammars considered here. Let G_1 and G_2 be ET0L grammars. The grammars G_1 and G_2 are said to be

- *language equivalent* if $L(G_1) = L(G_2)$,
- *structurally equivalent* if $STS(G_1) = STS(G_2)$, and
- *syntax equivalent* if $TS(G_1)$ and $TS(G_2)$ are equal modulo a renaming of the nonterminals.

Note that syntax equivalent grammars are always structurally equivalent and structural equivalence in turn implies language equivalence. It is well known that language equivalence is undecidable already for context-free grammars. Structural equivalence of context-free and E0L grammars is decidable [5, 6, 8, 9, 14, 15, 16]. Syntax equivalence of context-free grammars is considered in Ginsburg and Harrison [2].

To conclude this section we recall the definition of a normal form for context-sensitive grammars established by Penttonen [10]. A PNF (Penttonen-normal-form) context-sensitive grammar is specified by a tuple $G_{PNF} = (U_N, U_T, I, P)$, where U_N is a finite set of nonterminals, U_T is a finite set of terminals, $I \in U_N$ is the initial nonterminal, and P is a set of productions of the following three types:

- Right-context productions: $BD \to CD$, where $B, C, D \in U_N$
- Context-free productions: $B \to CD$, where $B, C, D \in U_N$
- Terminating productions: $B \to b$, where $B \in U_N$, $b \in U_T$

Thus, we allow only one-sided context in the productions. The productions of P define, in a natural way, the (sequential) rewrite-relation $\Rightarrow_{G_{PNF}} \subseteq (U_N \cup U_T)^+ \times (U_N \cup U_T)^+$ and the language generated by G_{PNF} is

$$L(G_{PNF}) = \{w \in U_T^+ \ : \ I \Rightarrow_{G_{PNF}}^+ w\}.$$

Strictly speaking, instead of the preceding productions with a right context condition Penttonen normal form [10] allows only left context in the productions of the grammar, (that is, productions of the form $DB \to DC$.) The definitions are, however, completely symmetric. Penttonen [10] proved the following result.

Theorem 2.1 (Penttonen [10]) *For an arbitrary context-sensitive grammar G_{CS} (with no length reducing productions) we can effectively construct a PNF grammar G_{PNF} such that $L(G_{PNF}) = L(G_{CS})$.*

3 Syntax equivalence

For context-free and E0L grammars both syntax equivalence and structural equivalence are decidable [2, 5, 6, 14]. Before proving our main undecidability result, we show that syntax equivalence is decidable for ET0L grammars.

Lemma 3.1 *Given ET0L grammars* $G_i = (V_i, \Sigma_i, S_i, H_i)$, $i = 1, 2$, *we can effectively decide whether*

$$TS(G_1) = TS(G_2).$$

Proof. We say that an ET0L grammar G is *reduced* if all nonterminal and terminal symbols of G appear in some terminal syntax tree of G. Using standard methods we can effectively find the subsets $V_i' \subseteq V_i$, $\Sigma_i' \subseteq \Sigma_i$, $1 \leq i \leq 2$, that consist of all symbols appearing in some tree $T \in TS(G_i)$. Thus, we can also effectively construct a reduced grammar G_i' that is syntax equivalent to G_i, for $i = 1, 2$, simply by removing the unnecessary symbols of $V_i \cup \Sigma_i$ and the productions that contain some of these symbols. Hence, without loss of generality, we can assume that the grammars G_1 and G_2 are reduced and that $V_1 = V_2 = V$, $\Sigma_1 = \Sigma_2 = \Sigma$, $S_1 = S_2 = S$, because if, say, $V_1 \neq V_2$ and G_1 and G_2 are reduced, then $TS(G_1) \neq TS(G_2)$.

The proof is based on the straightforward observation that $TS(G_1) \subseteq TS(G_2)$ if and only if, for every set of productions $p_1, \ldots, p_m \in h$, $h \in H_1$, that can be used in one parallel step of a successful derivation of G_1, there exists a table $h' \in H_2$ such that $p_1, \ldots, p_m \in h'$. We define a family γ_1 of multisets over $V \cup \Sigma$ that determines which sets of productions of H_1 are simultaneously applicable in a derivation starting from the initial nonterminal. Also, we define a collection η_1 of sets over $V \cup \Sigma$ that determines which sets of productions of H_1 yield a sentential form that can eventually be rewritten to a terminal string or sentence. (Note that, although G_1 is reduced, it is still possible that, for productions $a_i \to w_i$, $i = 1, 2$, belonging to a table of H_1, the string $w_1 w_2$ cannot yield a sentence.) Then, to complete the proof it is sufficient to show that the sets γ_1, η_1 (and the corresponding sets γ_2, η_2 constructed for the grammar G_2) are recursive. We now give the details of the proof.

For $h \in H_i$, $1 \leq i \leq 2$, we denote by M_h, the maximal number of productions of h that have the same left-hand side $a \in V$. Then, we define

$$M = \max\{M_h \ : \ h \in H_i, \ 1 \leq i \leq 2\}.$$

We say that $w \in (V \cup \Sigma)^*$ *covers* a multiset $B \in \mathcal{M}(V \cup \Sigma)$ if

- $\mathrm{alph}(w) = \mathrm{base}(B)$, and
- $(\forall a \in V \cup \Sigma) \ \ \#_a(w) \geq \#_a[B]$.

Intuitively, if w covers B, then w consists of exactly those symbols that belong to B and the multiplicity of each symbol a in B is at most the number of occurrences of a in the string w.

Let Ω_M consist of all multisets $B \in \mathcal{M}(V \cup \Sigma)$ such that

$$(\forall a \in V \cup \Sigma) \ \ \#_a[B] \leq M.$$

For $i = 1, 2$, we define a family $\gamma_i \subseteq \mathcal{M}(V \cup \Sigma)$ of multisets as:

$$\gamma_i = \{B \ : \ B \in \Omega_M \text{ and } (\exists w \in \mathrm{sf}(G_i)) \text{ such that } w \text{ covers } B\}.$$

The collection γ_i of multisets can be effectively constructed. The family Ω_M is finite and, for a given multiset $B \in \Omega_M$, we can determine whether $B \in \gamma_i$ as follows. Denote by L_B the set $\{w \in (V \cup \Sigma)^* \ : \ w \text{ covers } B\}$. Clearly, L_B is a regular language; thus, we can construct an ET0L grammar G_i^B such that

$$L(G_i^B) = \mathrm{sf}(G_i) \cap L_B.$$

To decide whether $B \in \gamma_i$, we merely check whether $L(G_i^B)$ is nonempty. Recall that the emptiness problem for ET0L grammars is decidable [4].

Next, for $i = 1, 2$, we define $\eta_i \subseteq \mathcal{P}(V \cup \Sigma)$ by

$$\eta_i = \{\{a_1, \ldots, a_m\} : (\exists w \in \Sigma^*) \ a_1 \cdots a_m \ (\Rightarrow_{G_i}^{\mathrm{par}})^* \ w \text{ and } a_1, \ldots, a_m \in V \cup \Sigma\}. \tag{2}$$

Note that, for $A \in \mathcal{P}(V \cup \Sigma)$, the relation $A \in \eta_i$ depends only on the set A although condition (2) contains a sequence of elements of A. Similar to the preceding argument, using the decidability of emptiness for ET0L grammars, we verify that η_i can be effectively constructed.

Now, for $i = 1, 2$, define the set $\gamma_i \diamond \eta_i \subseteq \Omega_M$ as:

$$\gamma_i \diamond \eta_i = \{B \ : \ B \in \gamma_i \text{ and } \mathrm{base}(B) \in \eta_i\}.$$

Thus, $B \in \gamma_i \diamond \eta_i$ if and only if there exist $w_1 \in (V \cup \Sigma)^*$ and $w_2 \in \Sigma^*$ such that w_1 covers B and

$$S \ (\Rightarrow_{G_i}^{\mathrm{par}})^* \ w_1 \ (\Rightarrow_{G_i}^{\mathrm{par}})^* \ w_2.$$

It should be clear that, if $\gamma_1 \diamond \eta_1 \neq \gamma_2 \diamond \eta_2$, then $TS(G_1) \neq TS(G_2)$. Therefore, we need consider only the case

$$\gamma_1 \diamond \eta_1 = \gamma_2 \diamond \eta_2 = \omega.$$

Let $B = \{\{b_1, \ldots, b_m\}\} \in \Omega_M$ be such that $\mathrm{base}(B) \subseteq V$. We say that B is *(1,2)-consistent* if the following condition holds:

For every $h \in H_1$ and sequence

$$p_1 : b_1 \to w_1, \ \ \ldots, \ \ p_m : b_m \to w_m, \tag{3}$$

of productions of h, where $w_1, \ldots, w_m \in (V \cup \Sigma)^*$, such that

$$\bigcup_{i=1}^{m} \mathrm{alph}(w_i) \in \eta_1, \tag{4}$$

there exists $h' \in H_2$ such that $p_1, \ldots, p_m \in h'$. Note that B is a multiset and the elements $b_1, \ldots, b_m$ are not necessarily distinct.

We claim that

$$TS(G_1) \subseteq TS(G_2) \tag{5}$$

if and only if

(6) $$(\forall B \in \omega) \quad B \text{ is (1,2)-consistent.}$$

First assume that (6) holds and let

$$S = T_0 \rightarrow_{G_1}^{\text{par}} T_1 \rightarrow_{G_1}^{\text{par}} \cdots \rightarrow_{G_1}^{\text{par}} T_n$$

be the derivation of an arbitrary syntax tree $T_n \in TS(G_1)$. Let $j \in \{0, \ldots, n-1\}$ and assume that the derivation step

$$D_j : T_j \rightarrow_{G_1}^{\text{par}} T_{j+1}$$

uses a table $h \in H_1$. For $a \in V$ denote by $D_j(a)$ the number of different productions of h with left side a used in D_j. Let B_j be the multiset containing $D_j(a)$ copies of the element $a \in V$. Then yield(T_j) covers B_j and it follows that $B_j \in \gamma_1 \diamond \eta_1 = \omega$. (Note that $B_j \in \Omega_M$ since $D_j(a) \leq M$ for every $a \in V$.) Hence B_j is (1,2)-consistent by the assumption (6). Since yield(T_{j+1}) $(\Rightarrow_{G_1}^{\text{par}})^*$ yield$(T_n) \in \Sigma^*$ it follows that the set of productions of h used in the derivation step D_j satisfies the condition (4). Now by the (1,2)-consistency condition there exists $h' \in H_2$ that can be used to exactly simulate the derivation step D_j, $0 \leq j \leq n-1$. Thus $T_n \in TS(G_2)$.

Conversely, assume that $B = \{\{b_1, \ldots, b_m\}\} \in \omega$ is not (1,2)-consistent. Then there exists $h \in H_1$ and $p_1, \ldots, p_m \in h$ as in (3) and (4) such that

(7) $$p_1, \ldots, p_m \text{ are not contained in any table of } H_2.$$

Since $B \in \omega$, there exists $T \in S(G_1)$ such that yield(T) covers B. Let $\{u_1, \ldots, u_n\}$ be the set of external nodes of T that are labeled with elements of V. Since yield(T) covers B, there exists a surjective mapping f from $\{u_1, \ldots, u_n\}$ to the multiset B such that $f(u_i) = \text{lab}_T(u_i)$, $1 \leq i \leq n$. Consider the derivation step

$$D : \; T \rightarrow_{G_1}^{\text{par}} T_1,$$

where in an external node u_i, $1 \leq i \leq n$, one applies the production $p_j : b_j \rightarrow w_j$ where $f(u_i) = b_j$, $j \in \{1, \ldots, m\}$. Since f is surjective, the derivation step D uses all productions $p_1, \ldots, p_m$. Also, since $\bigcup_{i=1}^{m} \text{alph}(w_i) \in \eta_1$, there exists $T_2 \in TS(G_1)$ such that T_1 $(\rightarrow_{G_1}^{\text{par}})^*$ T_2. On the other hand, it is clear that $T_2 \notin TS(G_2)$. Note that by (7), $T \not\rightarrow_{G_2}^{\text{par}} T_1$.

For a given multiset $B \in \omega$, the (1,2)-consistency condition is decidable, since η_1 can be computed. Since ω is finite and recursive, we can decide whether (5) holds. Finally, by symmetry, we can determine whether $TS(G_2) \subseteq TS(G_1)$. □

Since the number of nonterminals is finite the following result follows immediately from Lemma 3.1.

Theorem 3.1 *Syntax equivalence is decidable for ET0L grammars.*

4 Structural equivalence

Here we prove our main result: structural equivalence of ET0L grammars is undecidable. In the proof we need to consider only propagating grammars and, furthermore, one of the grammars can

be restricted to have only one table. The proof uses a reduction from the emptiness problem for context-sensitive languages, which is well known to be undecidable [12, 17]. For a given PNF context-sensitive grammar G_{PNF} we construct ET0L grammars G_1 and G_2 that are structurally equivalent if and only if the language generated by G_{PNF} is empty. The grammar G_1 essentially simulates the derivation of G_{PNF} but ignores the context conditions. The grammar G_2 is as G_1 but, in addition, it sends messengers down the syntax tree that nondeterministically verify that the context condition of G_{PNF} is violated somewhere in the syntax tree.

The construction that we use requires that the productions of the given context-sensitive grammar have only one-sided context, more specifically, right-sided context. (The proof could easily be modified to use left-sided context which is the original normal form of Penttonen.) Because of this restriction, we need only two messengers in the syntax tree. More important, the right-sided context allows us to interpret an arbitrary parallel derivation step of the corresponding ET0L grammar G_1, $T_1 \rightarrow_{G_1}^{\text{par}} T_2$, as a sequence of rewrite steps of G_{PNF}, *performed from left to right* by checking the context conditions only with respect to the initial sentential form, yield(T_1). If we had productions with two-sided context, the construction would be considerably more involved.

Theorem 4.1 *Given an EP0L grammar G_1 and an EPT0L grammar G_2, it is undecidable whether*

$$STS(G_1) = STS(G_2). \tag{8}$$

Proof. Let $G_{PNF} = (U_N, U_T, I, P)$ be an arbitrary PNF context-sensitive grammar. Let

$$p_1, \ldots, p_{k_1}, p_{k_1+1} \ldots, p_k, \quad 0 \leq k_1 \leq k \tag{9}$$

be an enumeration of the nonterminating productions of P, where $p_1, \ldots, p_{k_1}$ are the right-context productions and $p_{k_1+1}, \ldots, p_k$ are the context-free productions of P. We construct an EP0L grammar G_1 and EPT0L grammar G_2 such that (8) holds if and only if $L(G_{PNF}) = \emptyset$.

Choose

$$G_1 = (U_N \cup \{X, Y, S_1\}, U_T \cup \{\$, @\}, S_1, \{h\}),$$

where X, Y, S_1 are new nonterminal symbols, $\$, @$ are new terminal symbols ($X, Y, S_1, \$, @ \notin U_N \cup U_T$), and the table h contains exactly the following productions:

(G_1.0) $S_1 \rightarrow IY$.

(G_1.1) $B \rightarrow CX^i$ if $p_i : BD \rightarrow CD$, $1 \leq i \leq k_1$, is a right-context production of P.

(G_1.2) $B \rightarrow CDX^i$ if $p_i : B \rightarrow CD$, $k_1 < i \leq k$, is a context-free production of P.

(G_1.3) $B \rightarrow B$ for every $B \in U_N$.

(G_1.4) $B \rightarrow b$ if $B \rightarrow b$ is a terminating production of P.

(G_1.5) $X \rightarrow X$, $X \rightarrow \$$, $Y \rightarrow Y$, $Y \rightarrow @$.

Intuitively, the grammar G_1 simulates the derivations of G_{PNF} by ignoring the context conditions: the productions defined in $(G_1.1)$ can be applied independently of the right context. In addition, the grammar G_1 adds, for each nonterminating production p_i, i copies of the nonterminal X to the derivation tree. This technical modification ensures that a terminal syntax tree $T \in TS(G_1)$ is always uniquely determined by $\mathrm{str}_{G_1}(T)$. Also, for technical reasons, the production $(G_1.0)$ introduces a "right endmarker" Y in the derivation. Note that the initial nonterminal S_1 appears only at the root of each syntax tree of G_1.

Claim 1. The function $\mathrm{str}_{G_1} : TS(G_1) \to STS(G_1)$ is injective.

Proof of Claim 1. Let $T_1, T_2 \in TS(G_1)$ be such that $\mathrm{str}_{G_1}(T_1) = \mathrm{str}_{G_1}(T_2) = t$. The tree T_i, $i = 1, 2$, is determined completely by the domain $\mathrm{dom}(T_i)$ and the label function lab_{T_i}. Since $\mathrm{str}_{G_1}(T_1) = \mathrm{str}_{G_1}(T_2)$, it follows that $\mathrm{dom}(T_1) = \mathrm{dom}(T_2) = \mathrm{dom}(t)$ and $\mathrm{lab}_{T_1}(u) = \mathrm{lab}_{T_2}(u)$, for every external node u. We show that, for every internal node $u \in \mathrm{dom}(t)$, $\mathrm{lab}_{T_1}(u) = \mathrm{lab}_{T_2}(u)$ by induction on the distance $d(u)$ of u from the root of t.

(i) If $d(u) = 0$ or $d(u) = 1$, then $\mathrm{lab}_{T_i}(u)$ is uniquely determined by the production $(G_1.0)$, $1 \leq i \leq 2$.

(ii) Let u be an internal node of t that is not the root and let $d(u) = e \geq 2$. Let v be the parent of u and let $\tilde{y} = (u_1, \ldots, u_m)$, $m \geq 1$, be the sequence of children of v, where $u_r = u$, for some $r \in \{1, \ldots, m\}$. Let $\tilde{z} = (u_{j+1}, \ldots, u_m)$, $0 \leq j \leq m$, be the subsequence of $\tilde{y}$ that consists of all nodes u_i that are the roots of unary trees that have external nodes labeled with \$. (From the definition of the productions of G_1 it follows that $\tilde{z}$ is necessarily a suffix of $\tilde{y}$.)

(a) If $\tilde{z}$ is the empty sequence, then necessarily $m = 1$ and the production that is applied at the node v in the syntax tree T_i, $1 \leq i \leq 2$, has to be of the forms $B \to B$, $B \in U_N$, or $Y \to Y$. By the inductive assumption $\mathrm{lab}_{T_1}(v) = \mathrm{lab}_{T_2}(v)$ and it follows that $\mathrm{lab}_{T_1}(u) = \mathrm{lab}_{T_2}(u)$.

(b) If $\tilde{y} = \tilde{z}$, then necessarily $m = 1$ and the production that is applied at v in T_i, $1 \leq i \leq 2$, has to be $X \to X$.

(c) If $\tilde{y} \neq \tilde{z}$ and $\tilde{z}$ is nonempty, then $1 \leq j < m$. Clearly, for $i \in \{1, 2\}$ and $n \in \{1, \ldots, m\}$, $\mathrm{lab}_{T_i}(u_n) = X$ if and only if $n > j$. (The external nodes of the subtrees that correspond to the nodes $u_{j+1}, \ldots, u_m$ are labeled by the terminal symbol \$.) Thus, the production applied at the node v in T_i, $1 \leq i \leq 2$, has to be the production that corresponds (by $(G_1.1)$ or $(G_1.2)$) to the production p_{m-j} of the grammar G_{PNF}. It follows that $\mathrm{lab}_{T_1}(u_n) = \mathrm{lab}_{T_2}(u_n)$, for all $n \in \{1, \ldots, m\}$.

We have completed the proof of the claim.

We say that a terminal syntax tree $T \in TS(G_1)$ is *context-sensitive* if the rewrite steps that are indicated in T do not violate the context conditions of the grammar G_{PNF}.

More formally, we define the *context-sensitive derivation relation* (of G_1 with respect to G_{PNF}) on the set of syntax trees, $\to^{\mathrm{par}}_{G_1[CS]} \subseteq \to^{\mathrm{par}}_{G_1}$, as follows. Let $T_1, T_2 \in F_{G_1}$ and $T_1 \to^{\mathrm{par}}_{G_1} T_2$. Assume

that the sequence of external nodes of T_1 from left to right is $(u_1, \ldots, u_m)$ where u_i is labeled by $A_i \in U_N \cup \{X, Y, S_1\}$, $i = 1, \ldots, m$. (Note that G_1 is propagating.) Furthermore, assume that T_2 is obtained from T_1 by attaching $r_i \geq 1$ successors labeled by the symbols $B_1^i, \ldots, B_{r_i}^i$ to the node u_i. Then,

$$T_1 \rightarrow^{\mathrm{par}}_{G_1[CS]} T_2$$

if and only if the following condition holds.

(CS) Let $i \in \{1, \ldots, m\}$ be such that

$$B_1^i \cdots B_{r_i}^i = CX^{r_i - 1},\ C \in U_N,\ 2 \leq r_i \leq k_1 + 1.$$

This means that the production $A_i \rightarrow B_1^i \cdots B_{r_i}^i$ of h corresponds to the right-context production $p_{r_i-1} : A_i D \rightarrow CD$ of P for some $D \in U_N$. Then, there exists $j \in \{i+1, \ldots, m\}$ such that $A_{i+1} = A_{i+2} = \cdots = A_{j-1} = X$ and $A_j = D$; that is, the next nonterminal symbol different from X in the yield of T_1 is D as required by the context condition of the production p_{r_i-1}.

The set of *context-sensitive syntax trees* of G_1 is defined as

$$S_{CS}(G_1) = \{T \in F_{G_1} \ :\ S_1' \ (\rightarrow^{\mathrm{par}}_{G_1[CS]})^* \ T\},$$

where S_1' denotes the tree with one node labeled with the initial nonterminal S_1. Also define

$$TS_{CS}(G_1) = S_{CS}(G_1) \cap TS(G_1).$$

Claim 2. $TS_{CS}(G_1) \neq \emptyset$ if and only if $L(G_{PNF}) \neq \emptyset$.

Proof of Claim 2. Let $f_\lambda : (U_N \cup U_T \cup \{X, Y, \$, @\})^* \rightarrow (U_N \cup U_T)^*$ be the morphism defined by $f_\lambda(a) = a$, for $a \in U_N \cup U_T$, and $f_\lambda(X) = f_\lambda(Y) = f_\lambda(\$) = f_\lambda(@) = \lambda$.

First assume that $TS_{CS}(G_1) \neq \emptyset$ and let

$$T_0 \rightarrow^{\mathrm{par}}_{G_1[CS]} T_1 \rightarrow^{\mathrm{par}}_{G_1[CS]} \cdots \rightarrow^{\mathrm{par}}_{G_1[CS]} T_m$$

be a parallel context-sensitive derivation of $T_m \in TS_{CS}(G_1)$, where T_0 is the tree with one node labeled by S_1. From condition (CS) in the definition of the relation $\rightarrow^{\mathrm{par}}_{G_1[CS]}$ it follows that

$$f_\lambda(\mathrm{yield}(T_i)) \Rightarrow^*_{G_{PNF}} f_\lambda(\mathrm{yield}(T_{i+1})), \tag{10}$$

$1 \leq i \leq m-1$. Note that the productions of G_{PNF} involve only right context conditions. Hence if we ignore the external nodes labeled by the nonterminal X, then a parallel derivation step of G_1 that satisfies (CS) correctly simulates a sequence of derivation steps of G_{PNF} *performed from left-to-right.* By (10) it follows that

$$I = f_\lambda(\mathrm{yield}(T_1)) \Rightarrow^*_{G_{PNF}} f_\lambda(\mathrm{yield}(T_m)) \in U_T^+;$$

thus, $L(G_{PNF}) \neq \emptyset$.

For the proof in the "if"-direction, assume that

$$I = w_0 \Rightarrow_{G_{PNF}} w_1 \Rightarrow_{G_{PNF}} \cdots \Rightarrow_{G_{PNF}} w_m \Rightarrow^*_{G_{PNF}} w_{m+1},$$

where $w_0, \ldots, w_m \in U_N^+$, $w_{m+1} \in U_T^+$ and the derivation $w_m \Rightarrow^*_{G_{PNF}} w_{m+1}$ uses only the terminating productions of G_{PNF}. Since the context conditions of G_{PNF} involve only nonterminals, every string $w_{m+1} \in L(G_{PNF})$ has a derivation of this form. We show that, for every $i \in \{0, \ldots, m\}$, there exists $T_i \in S_{CS}(G_1)$ such that

$$f_\lambda(\text{yield}(T_i)) = w_i \text{ and } \text{yield}(T_i) \in (U_N \cup \{X, Y\})^*. \tag{11}$$

(i) For $i = 0$, we choose T_0 to be the tree obtained from S_1 with the production $(G_1.0)$.

(ii) Assume that there is a $T_i \in S_{CS}(G_1)$ that satisfies (11), $i < m$. Assume that w_{i+1} is obtained from w_i using a production $p_j : BD \to CD$, $(B, C, D \in U_N)$, $1 \leq j \leq k_1$. The case where the production is context-free is similar and simpler. Let u be the external node of T_i that is labeled with the corresponding occurrence of the nonterminal B. We construct T_{i+1} by applying, to the external node u, the production

$$B \to CX^j$$

of h and to all other external nodes appropriate productions $E \to E$, $E \in U_N \cup \{X, Y\}$. Since $f_\lambda(\text{yield}(T_i)) = w_i$, it is clear that this derivation step satisfies the condition (CS).

Since w_m can be rewritten to give w_{m+1} using only terminating productions, there exists $T_{m+1} \in TS_{CS}(G_1)$ such that $T_m \to^{\text{par}}_{G_1[CS]} T_{m+1}$ using the productions $(G_1.4)$ and $X \to \$$, $Y \to @$. This concludes the proof of the claim.

Next, we define the EPT0L grammar G_2. Intuitively, the grammar G_2 generates exactly all syntax trees of G_1 that are not context-sensitive. We augment the nonterminals of G_1 with additional components that nondeterministically verify that the context condition is violated somewhere in the syntax tree. For this purpose, G_2 needs more than one table. Let $Z = \{z, z_1, z_2\}$ and define the EPT0L grammar $G_2 = (V, \Sigma, S, H)$, where

- $V = U_N \cup \{X, Y\} \cup ((U_N \cup \{Y, S_1\}) \times Z)$,
- $\Sigma = U_T \cup \{\$, @\}$,
- $S = (S_1, z) \in V$, and
- $H = \{g, g_1, \ldots, g_{k_1}\}$, where k_1 is from (9).

The tables $g, g_1, \ldots, g_{k_1}$ are defined as follows. The table g contains productions $(G_1.1) - (G_1.5)$ of h and additionally the following productions:

$(G_2.0)$ **(i)** $(S_1, z) \to (I, z)Y$,

(ii) $(S_1, z) \to (I, z_1)(Y, z_2)$.

(G_2**.1**) If $B \to CX^i \in h$, $B, C \in U_N$, $1 \leq i \leq k_1$, then

(i) $(B, z) \to (C, z)X^i$,

(ii) $(B, z_j) \to (C, z_j)X^i$, $j = 1, 2$.

(G_2**.2**) If $B \to CDX^i \in h$, $B, C, D \in U_N$, $k_1 < i \leq k$, then

(i) $(B, z) \to (C, z)DX^i$, $(B, z) \to C(D, z)X^i$,

(ii) $(B, z) \to (C, z_1)(D, z_2)X^i$,

(iii) $(B, z_1) \to C(D, z_1)X^i$,

(iv) $(B, z_2) \to (C, z_2)DX^i$.

(G_2**.3**) For every $B \in U_N$, the productions $(B, z) \to (B, z)$, $(B, z_j) \to (B, z_j)$, $j = 1, 2$, and, $(Y, z_2) \to (Y, z_2)$.

Let $r \in \{1, \ldots, k_1\}$ and assume that the right-context production p_r of G_{PNF} is of the form

$$p_r : \ BD \to CD, \quad (B, C, D \in U_N). \tag{12}$$

The table g_r contains the productions $(G_1.1) - (G_1.5)$ of h and the productions

(M_1^r) $(B, z_1) \to CX^r$.

(M_2^r) $(E, z_2) \to w$, where $E \in (U_N - \{D\}) \cup \{Y\}$, $w \in (U_N \cup \{X, Y\})^+$ and $E \to w \in h$.

Let $f_Z : (V \cup \Sigma)^* \to (U_N \cup U_T \cup \{X, Y, S_1, \$, @\})^*$ be the morphism determined by the conditions: $f_Z((x, y)) = x$, $x \in U_N \cup \{Y, S_1\}$, $y \in Z$, and $f_Z(x) = x$ when $x \in U_N \cup U_T \cup \{X, Y, \$, @\}$. The function f_Z simply erases the second components belonging to Z from the nonterminals. Then every production $L \to R$ belonging to the tables $g, g_1, \ldots, g_{k_1}$ has the property that

$$f_Z(L) \to f_Z(R) \ \in h. \tag{13}$$

If $T \in S(G_2)$, we denote by $f_Z(T)$ the tree defined by the conditions $\mathrm{dom}(f_Z(T)) = \mathrm{dom}(T)$, and $\mathrm{lab}_{f_Z(T)}(u) = f_Z(\mathrm{lab}_T(u))$, $u \in \mathrm{dom}(T)$. It follows by (13) that

$$(\forall T \in S(G_2))\ f_Z(T) \in S(G_1). \tag{14}$$

Hence it follows also that

$$STS(G_2) \subseteq STS(G_1).$$

Intuitively, the symbols z, z_1, z_2 can be seen as messengers that travel nondeterministically down in a syntax tree of G_1 and find a position where the syntax tree violates the context condition (CS). In a sentential form of G_2 the messengers z_1 and z_2 will always be forced to be located in nonterminals $N_1, N_2 \in U_N \cup \{Y\}$ that are separated only by a sequence of nonterminals X. Thus N_1 and N_2 represent consecutive nonterminals in the derivation of G_{PNF} that is simulated or N_1 is the rightmost nonterminal in the derivation of G_{PNF} and $N_2 = Y$ is the "right endmarker." The

tables of G_2 are defined so that the only possibility to delete the symbols z_1 and z_2 is to apply productions of a table g_r, $1 \leq r \leq k_1$, that force the context condition to be violated.

In the following we show that G_2 generates exactly the structures of syntax trees of G_1 that do not correspond to a context-sensitive syntax tree.

Claim 3. $STS(G_2) = STS(G_1) - \mathrm{str}_{G_1}(TS_{CS}(G_1))$.

Proof of Claim 3. Let $t \in STS(G_1) - \mathrm{str}_{G_1}(TS_{CS}(G_1))$. By Claim 1, there exists a unique $T \in TS(G_1) - TS_{CS}(G_1)$ such that $\mathrm{str}_{G_1}(T) = t$. Denote the parallel derivation sequence of T by

$$T_0 \rightarrow_{G_1}^{\mathrm{par}} T_1 \rightarrow_{G_1}^{\mathrm{par}} \cdots \rightarrow_{G_1}^{\mathrm{par}} T_m = T, \tag{15}$$

where T_0 is the tree with one node labeled by S_1. (Note that given T the derivation sequence (15) is uniquely determined.) Since $T \notin TS_{CS}(G_1)$, there exists $i \in \{1, \ldots, m-1\}$ such that the derivation step

$$T_i \rightarrow_{G_1}^{\mathrm{par}} T_{i+1} \tag{16}$$

does not satisfy the condition (CS). Thus in T_i there exists an external node u labeled with $A \in U_N$ such that in the derivation step (16) at u we apply a production

$$A \rightarrow CX^j \quad \in h, \tag{17}$$

$1 \leq j \leq k_1$, the next external node u' of T_i to the right from u that is labeled by an element different from X is labeled with $E \in U_N \cup \{Y\}$, and the production p_j of G_{PNF} has the form $AD \rightarrow CD$, where $D \neq E$. Denote by u_0 the least common predecessor of u and u'; that is, u_0 is the common predecessor of u and u' furthest from the root of T_i. Let the distance of u_0 from the root be e, that is, u_0 is an external node of T_e, $0 \leq e < i$. We construct a derivation sequence of G_2

$$(S_1, z) = T'_0 \rightarrow_{G_2}^{\mathrm{par}} T'_1 \rightarrow_{G_2}^{\mathrm{par}} \ldots \rightarrow_{G_2}^{\mathrm{par}} T'_m \tag{18}$$

as follows. The first components of the nonterminals in the derivation (18) simulate directly the derivation (15), that is, $f_Z(T'_c) = T_c$, $c = 0, \ldots, m$. The first i steps of (18) use only the table g. In the first e derivation steps the messenger symbol z travels nondeterministically to the external node u_0 of T'_e using productions $(G_2.0)$(i), $(G_2.1)$(i), $(G_2.2)$(i) and $(G_2.3)$. The external node u_0 is in the natural way viewed also as a node of T'_e. In the following we always identify the corresponding nodes of T_c and T'_c, $c \in \{0, \ldots, m\}$. Since u_0 is the least common predecessor of u and u', it follows that necessarily the production applied at u_0 in (15) is either $(G_1.0)$ or of the type $(G_1.2)$. (The productions $(G_1.1)$ and $(G_1.3)$ have only one successor labeled by an element of $U_N \cup \{Y\}$.) In the derivation step $T'_e \rightarrow_{G_2}^{\mathrm{par}} T'_{e+1}$ at the node u_0 we use the corresponding production $(G_2.0)$(ii) or $(G_2.2)$(ii) that branches the z-messenger into the messengers z_1 and z_2. By the definition of the productions $(G_2.1)$(ii), $(G_2.2)$(iii),(iv) and $(G_2.3)$ it is clear that in the tree T'_i the z_1-messenger has reached the node u and the z_2-messenger is in the node u'. Note that in productions $(G_2.2)$(iii) and (iv) the z_1-messenger always follows the rightmost branch not consisting of X-nonterminals and the symbol z_2 always follows the leftmost branch. These paths are just the paths from u_0 to the nodes u and u', since u and u' are consecutive external nodes of T_i when we disregard nodes labeled by the X-nonterminals.

Thus, the node u in T'_i is labeled by (A, z_1) and the node u' is labeled by (E, z_2); see (17). The derivation step $T'_i \rightarrow^{\mathrm{par}}_{G_2} T'_{i+1}$ uses the table g_j to eliminate the messengers z_1 and z_2 by productions (M_1^j) and (M_2^j). (Here j is from (17).) At external nodes of T'_i other than u and u', we apply the same productions as in (16), which is possible, since the table g_j contains the productions $(G_1.1)$–$(G_1.5)$.

Now, $\mathrm{yield}(T'_{i+1}) = \mathrm{yield}(T_{i+1})$ and the derivation (18) can be completed as in (15); hence, $t = \mathrm{str}_{G_2}(T'_m)$, $T'_m \in TS(G_2)$ implies that $t \in STS(G_2)$.

For the converse, let $t \in STS(G_2)$ and assume that $t = \mathrm{str}_{G_2}(T)$, $T \in TS(G_2)$. By (14), $f_Z(T) \in TS(G_1)$; thus, $t = \mathrm{str}_{G_1}(f_Z(T)) \in STS(G_1)$. Let

$$(S_1, z) = T_0 \rightarrow^{\mathrm{par}}_{G_2} T_1 \rightarrow^{\mathrm{par}}_{G_2} \ldots \rightarrow^{\mathrm{par}}_{G_2} T_m = T \tag{19}$$

be a derivation of T. Since T is a terminal syntax tree, in some step $T_i \rightarrow^{\mathrm{par}}_{G_2} T_{i+1}$ of (19), $0 \leq i < m-1$, we have to divide the messenger z to the pair of messengers z_1 and z_2 that are then finally destroyed by productions (M_1^r) and (M_2^r) of a suitable table g_r, $1 \leq r \leq k_1$, in a derivation step $T_j \rightarrow^{\mathrm{par}}_{G_2} T_{j+1}$, $i < j \leq m-1$. (There is no other way to delete the messenger symbols.) It is easy to see inductively that, for all $n \in \{i+1, \ldots, j\}$, we can write

$$\mathrm{yield}(T_n) = w_1(A_1, z_1)X^s(A_2, z_2)w_2,$$

where $A_1 \in U_N$, $A_2 \in U_N \cup \{Y\}$, $w_1 \in (U_N \cup \{X\})^*$, $w_2 \in (U_N \cup \{X, Y\})^*$, and $s \geq 0$; that is, the messenger symbols z_1 and z_2 label consecutive nonterminals in the yield when we disregard nodes labeled by X. From the form of the productions (M_1^r) and (M_2^r), the derivation step $f_Z(T_j) \rightarrow^{\mathrm{par}}_{G_1} f_Z(T_{j+1})$ does not satisfy the condition (CS); thus, $f_Z(T) \notin TS_{CS}(G_1)$. Since str_{G_1} is injective, we deduce that $t \notin \mathrm{str}_{G_1}(TS_{CS}(G_1))$, which completes the proof of Claim 3.

Combining Claims 2 and 3, we obtain $STS(G_1) = STS(G_2)$ if and only if $L(G_{PNF}) = \emptyset$. By Theorem 2.1, this implies that (8) is undecidable in general. □

The following simple example illustrates the construction of the proof of Theorem 4.1.

Example 4.1 Consider the PNF grammar $G_{PNF} = (U_N, U_T, I, P)$ where $U_N = \{I, A, B, C\}$, $U_T = \{a, b\}$ and P consists of the right-context production $p_1 : CB \rightarrow AB$, the context-free productions $p_2 : I \rightarrow CA$, $p_3 : I \rightarrow CB$ and the terminating productions $A \rightarrow a$, $B \rightarrow b$. (p_1, p_2, p_3 are the names for the nonterminating productions as used in the proof of Theorem 4.1.)

Let G_1 and G_2 be the EP(T)0L grammars constructed from G_{PNF} as in the proof of Theorem 4.1. The grammar G_1 has for instance the following parallel derivation of a sentence:

$$S_1 \Rightarrow^{\mathrm{par}}_G IY \Rightarrow^{\mathrm{par}}_G CAXXY \Rightarrow^{\mathrm{par}}_G AXAXXY \Rightarrow^{\mathrm{par}}_G a\$a\$\$@. \tag{20}$$

(In the nonterminating parallel steps we always apply to the nonterminals X and Y the productions $X \rightarrow X$, $Y \rightarrow Y$.) In the third parallel derivation step the grammar G_1 rewrites the leftmost nonterminal C by a production simulating p_1 but ignoring the right-context condition. Thus G_2 can simulate the derivation (20) as follows:

$$(S_1, z) \Rightarrow^{\mathrm{par}}_G (I, z)Y \Rightarrow^{\mathrm{par}}_G (C, z_1)(A, z_2)XXY \Rightarrow^{\mathrm{par}}_G AXAXXY \Rightarrow^{\mathrm{par}}_G a\$a\$\$@. \tag{21}$$

In the third parallel step of (21) G_2 uses the table g_1 that verifies that the context condition is violated in the consecutive nonterminals (C, z_1) and (A, z_2). It is clear that the structures of the syntax trees corresponding to the derivations (20) and (21) are identical.

However, $L(G_{PNF})$ is nonempty and the EP0L grammar G_1 has the following parallel derivation simulating a correct context-sensitive derivation of G_{PNF}:

$$S_1 \Rightarrow_G^{\text{par}} IY \Rightarrow_G^{\text{par}} CBXXXY \Rightarrow_G^{\text{par}} AXBXXXY \Rightarrow_G^{\text{par}} a\$b\$\$\$@ \tag{22}$$

The EPT0L grammar G_2 does not have any derivation with the same structure as the preceding derivation. (If G_2 attempts to "simulate" (22) it cannot get rid of the z-symbols.) Thus G_1 and G_2 are not structurally equivalent as required since $L(G_{PNF}) \neq \emptyset$.

The contrasting results of Theorems 3.1 and 4.1 can be interpreted by saying that, at least in the ET0L case, one loses essential information about a derivation when going from syntax trees to the corresponding structure trees.

In the proof of Theorem 4.1, the number of tables of the EPT0L grammar G_2 depends on the PNF context-sensitive grammar G_{PNF}. Every ET0L grammar is language equivalent to an ET0L grammar that has only two tables [11], but the corresponding transformation clearly does not preserve structural equivalence of the grammars. We can, however, strengthen Theorem 4.1 somewhat.

Theorem 4.2 *Given an EP0L grammar G_1 and an EPT0L grammar G_2 that has two tables it is undecidable whether*

$$STS(G_1) = STS(G_2).$$

Proof. Given a PNF context-sensitive grammar G_{PNF} we construct the grammar G_1 exactly as in the proof of Theorem 4.1 and transform the grammar G_2 into a grammar G_2' that has two tables as follows. In G_2', we merge the tables $g_1, \ldots, g_{k_1}$ into one table by coding, in the messenger symbols, the information about the production of G_{PNF} whose context condition the derivation is going to violate. When the messenger z branches into two messengers using the production $(G_2.2)$(ii) or $(G_2.0)$(ii), the grammar chooses, nondeterministically, a pair of messengers z_1^r, z_2^r, $1 \leq r \leq k_1$. The first table of G_2' is essentially the table g augmented with the preceding nondeterministic choice. The second table g' contains the productions $(G_1.1)$–$(G_1.5)$ of h and, for every $r \in \{1, \ldots, k_1\}$ and p_r of G_{PNF} of the form given in (12), g' contains the productions

- $(B, z_1^r) \to CX^r$.
- $(E, z_2^r) \to w$, where $E \in (U_N - \{D\}) \cup \{Y\}$, $w \in (U_N \cup \{X, Y\})^+$ and $(E \to w) \in h$.

Intuitively, in the syntax trees of G_2', we determine which of the tables $g_1, \ldots, g_{k_1}$ will be used to delete the messengers when we choose the symbols z_1^r and z_2^r, $1 \leq r \leq k_1$. It is clear that $STS(G_2') = STS(G_2)$; therefore, we cannot decide whether G_1 and G_2' are structurally equivalent. □

Theorem 4.2 is optimal with respect to the number of tables, since structural equivalence is decidable for E0L grammars. On the other hand, it is clear that the proof method of Theorem 4.1 does not work if the tables of a ET0L grammar are homomorphisms; that is, we have EDT0L grammars [11]. It is an open question whether structural equivalence is decidable for EDT0L grammars.

References

[1] H. A. Cameron and D. Wood, Structural equivalence of extended context-free and E0L grammars, unpublished manuscript, 1995.

[2] S. Ginsburg and M. Harrison, Bracketed context-free languages, *Journal of Computer and System Sciences* **1** (1967), 1–23.

[3] G. Istrate, The strong structural equivalence of ET0L grammars. In: "Developments in Language Theory," G. Rozenberg and A. Salomaa (eds.), World Scientific Publishing, Singapore, 1994, pp. 81–89.

[4] N. D. Jones and S. Skyum, Complexity of some problems concerning L systems, *Mathematical Systems Theory* **13** (1979), 29–43.

[5] R. McNaughton, Parenthesis grammars, *Journal of the Association of Computing Machinery* **14** (1967), 490–500.

[6] V. Niemi, A normal form for structurally equivalent E0L grammars. In: "Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology", G. Rozenberg and A. Salomaa (eds.), Springer-Verlag, 1992, pp. 133–148.

[7] Th. Ottmann and D. Wood, Defining families of trees with E0L grammars, *Discrete Applied Mathemtaics* **32** (1991), 195–209.

[8] Th. Ottmann and D. Wood, Simplifications of E0L grammars. In: "Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology", G. Rozenberg and A. Salomaa (eds.), Springer-Verlag, 1992, pp. 149–166.

[9] M. Paull and S. Unger, Structural equivalence of context-free grammars, *Journal of Computer and System Sciences* **2** (1968), 427–463.

[10] M. Penttonen, One-sided and two-sided context in formal grammars, *Information and Control* **25** (1974), 371–392.

[11] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems.* Academic Press, New York, 1980.

[12] A. Salomaa, *Formal Languages.* Academic Press, New York, 1973.

[13] K. Salomaa, D. Wood, and S. Yu, Complexity of E0L structural equivalence, *RAIRO Informatique théorique et Applications* (1995), to appear.

[14] K. Salomaa and S. Yu, Decidability of structural equivalence of E0L grammars, *Theoretical Computer Science* **82** (1991), 131–139.

[15] J. W. Thatcher, Characterizing derivation trees of a context-free grammar through a generalization of finite-automata theory, *Journal of Computer and System Sciences* **1** (1967), 317–322.

[16] J. W. Thatcher, Tree automata: an informal survey. In: "Currents in the Theory of Computing", A. V. Aho (ed.), Prentice Hall, Englewood Cliffs, NJ, 1973, pp. 143–172.

[17] D. Wood, *Theory of Computation.* John Wiley & Sons, New York, 1987.

[18] D. Wood, Standard Generalized Markup Language: Mathematical and philosophical issues. In: "Computer Science Today," J. van Leeuwen (ed.), Springer-Verlag Lecture Notes in Computer Science 1000 (1995), to appear.