Properties of Elementary Formal Systems as Translation Grammars

Sugimoto, Noriko
Department of Artificial Intelligence Kyushu Institute of Technology

出版情報：RIFIS Technical Report. 107, 1995-04-04. Research Institute of Fundamental Information Science, Kyushu University

RIFIS-TR-CS-107

RIFIS Technical Report

Properties of Elementary Formal Systems as Translation Grammars

Noriko Sugimoto

April 4, 1995
Revised : October 1995

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan
E-mail:sugimoto@ai.kyutech.ac.jp Phone:0948-29-7630

Properties of elementary formal systems as translation grammars

Noriko Sugimoto
Department of Artificial Intelligence
Kyushu Institute of Technology
Kawazu 680-4, Iizuka 820, Japan

Abstract

We show some properties of elementary formal systems (EFS's, for short) as translation grammars. A translation is a binary relation over strings. Since the EFS's are logic programs over strings, they can easily define translations.

In this paper, we consider the problem for deciding whether the number of target sentences, which correspond to output strings, is finite or not for any source sentence, which corresponds to an input string. When we need all target sentences, the finiteness of the given translation is important. However, the finiteness problem is unsolvable in general. Therefore, we give a class of translations in which the problem is solvable. On the other hand, for some integer k, a k-bounded translation is defined as the translation in which the number of target sentences is less than or equal to k for any source sentence. Furthermore, we give a class of EFS's which define 1-bounded translations.

KEY WORDS: elementary formal system, translation, formal language

C.R. CATEGORIES: F.4, I.6

1 Introduction

An elementary formal system (EFS, for short) was first introduced by Smullyan [9] to develop his recursive function theory. Arikawa *et al.* [3, 5] showed that various classes of languages in Chomsky hierarchy can be defined by the EFS's. Furthermore, since an EFS is a kind of logic program [11], we can easily define various classes of relations over strings. Since a translation can be considered as a binary relation over strings, we can define translations by EFS's. Irons [6] introduced a Syntax Directed Translation (SDT, for short), which is an extension of CFG, as translation grammar. Many researcher discussed properties of translations by SDT's [1, 2, 6, 7]. We can obtain restricted EFS's which are equivalent to the SDT's. Furthermore, EFS's can define larger classes of translations.

Essentially, a SDT defines a translation over context-free languages, because a SDT is an extension of a CFG . On the other hand, since EFS's can define recursively enumerable languages, we are possible to obtain more rich classes of translations.

In this paper, we investigate some properties of translations defined by EFS's. Especially, we focus on the finiteness of translations. That is, we consider the problem for deciding whether the number of target sentences, which correspond to output strings, is finite or not for any source sentence, which corresponds to input strings, in the translation. When we need all target sentences, it is important whether the given translation is finite or not. On the other hand, we consider translations by compilers, we wish the number of target sentences is at most one. Hence, we formalize the finiteness of translations. In general, for given translation T, the problem for deciding whether T is finite or not is unsolvable. However, we show that we can easily obtain the class of translations in which the problem is solvable by EFS's. Furthermore, in this paper, for some non-negative integer k, a k-bounded translation is defined as the translation in which the number of target sentences is less than or equal to k for any source sentence. In general, for given translation T, the problem for deciding whether T is 1-bounded or not is unsolvable. In this paper, we give conditions under which the number of target sentences is at most one.

2 Elementary formal systems

In this section, we present basic definitions for EFS's according to [3, 4, 5].

Let Σ, X and Π be mutually disjoint sets. We assume that Σ and Π are finite. We refer to each element of Σ as a *constant symbol*, to each element of X as a *variable*, and to each element of Π as a *predicate symbol*. In particular, Σ is called *alphabet*. Each predicate symbol is associated with a non-negative integer called its *arity*. For a set A, we denote the set of all finite strings of symbols from A by A^*, and the set $A^* - \{\varepsilon\}$ by A^+, where ε is a string whose length is 0. A *term* is an element of $(\Sigma \cup X)^+$. A term is said to be *ground* if it is an element of Σ^+. An *atomic formula* (*atom*, for short) is of the form $p(\pi_1, \pi_2, \ldots, \pi_n)$, where p is a predicate symbol with arity n and each π_i is a term $(1 \leq i \leq n)$. An atom $p(\pi_1, \pi_2, \ldots, \pi_n)$ is said to be *ground* if all $\pi_1, \pi_2, \ldots, \pi_n$ are ground. A *definite clause* (*clause*, for short) is of the form $A \leftarrow B_1, \ldots, B_n$ $(n \geq 0)$, where $A, B_1, \ldots, B_n$ are atoms. The atom A is called the *head* and the sequence $B_1, \ldots, B_n$ of atoms is called the *body* of the clause. A *goal clause* (*goal*, for short) is of the form $\leftarrow B_1, \ldots, B_n$ $(n \geq 0)$ and the goal with $n = 0$ is called the *empty goal*. A *substitution* is a finite set of the form $\{x_1/\pi_1, \ldots, x_n/\pi_n\}$, where $x_1, \ldots, x_n$ are distinct variables and each π_i is a term distinct from x_i $(0 \leq i \leq n)$. We refer to either a term, an atom or a clause as an *expression*. Let E be an expression. Then, for a substitution $\theta = \{x_1/\pi_1, \ldots, x_n/\pi_n\}$, $E\theta$ is the expression obtained from E by simultaneously replacing each occurrence of the variable x_i in E by the term π_i $(1 \leq i \leq n)$. We say that $E\theta$ is an *instance* of E. If there is no variable occurring in $E\theta$, the instance $E\theta$ of E is said to be ground. Let E_1 and E_2 be a pair of expressions and θ be a substitution. If $E_1\theta = E_2\theta$ then we say that θ is a *unifier* of E_1 and E_2. For an expression E, the set of variables occurring in E is denoted by $v(E)$. The length of a term π is denoted by $|\pi|$.

An *elementary formal system* (EFS, for short) S is a triplet (Σ, Π, Γ), where Γ is a finite set of clauses [9]. Each clause in Γ is called an *axiom* of S. For an EFS S and a ground atom α, a *derivation tree* of α on S is a finite tree that satisfies the following conditions.

1. Each node of the tree is a ground atom.
2. The root node is α.
3. For each internal node β and its children $\beta_1, \ldots, \beta_n$ $(n \geq 1)$, $\beta \leftarrow \beta_1, \ldots, \beta_n$ is a ground instance of an axiom of S.

A *proof tree* of α on S is a derivation tree of α on S such that each leaf of the tree is a ground instance of an axiom of S with empty body.

Let I be a set of ground atoms. We define the function T_S as follows:

$$T_S(I) = \{\alpha \mid \text{there exists a ground instance } \alpha \leftarrow \beta_1, \ldots, \beta_n \text{ of an axiom of } S \text{ such that } \beta_i \in I \text{ for any } i (1 \leq i \leq n)\}.$$

The set $T_S{\uparrow}\omega$ is defined as follows:

1. $T_S{\uparrow}0 = \emptyset$,
2. $T_S{\uparrow}n = T_S(T_S{\uparrow}(n-1))$ for $n \geq 1$,
3. $T_S{\uparrow}\omega = \bigcup_{n \geq 0} T_S{\uparrow}n$.

We define $SS(S)$ as the set of all ground atoms α such that there exists a proof tree of α on S. Yamamoto [11] showed that $T_S{\uparrow}\omega = SS(S)$ for every EFS S.

3 The finiteness problem for translations

In this section, we formalize the finiteness problem for translations. A *translation over* an alphabet Σ is a subset of $\Sigma^+ \times \Sigma^+$. For example, $\{(0,\ 0),\ (1,\ 1),\ (10,\ 2),\ (11,\ 3),\ (100,\ 4),\ (101,\ 5),\ \ldots\}$ is a translation over $\{0, 1, \ldots, 9\}$. For each element (w_1, w_2) of the translation T, w_1 is called the *source sentence* and w_2 is called the *target sentence.* In what follows, we will often omit indicating the alphabet Σ over which the translations defined.

Let T be a translation, and $w \in \Sigma^+$. Then, we define

$$dom(T) = \{w_1 \mid\ (w_1, w_2) \in T\}, \qquad range(T) = \{w_2 \mid\ (w_1, w_2) \in T\},$$
$$T(w) = \{u \mid (w, u) \in T\}, \qquad T^R(w) = \{v \mid (v, w) \in T\}.$$

That is, $T(w)$ denotes the set of all target sentences into which the source sentence w is translated by T. Conversely, $T^R(w)$ denotes the set of all source sentences which should be translated into w by T.

A translation T is said to be *finite* if the set $T(w)$ is finite for every $w \in dom(T)$. Furthermore, T is said to be *bidirectionally finite* if both $T(w_1)$ and $T^R(w_2)$ are finite for every $w_1 \in dom(T)$ and every $w_2 \in range(T)$. For some non-negative integer k, T is said to be k*-bounded* if $|T(w)| \leq k$ for every $w \in dom(T)$.

Example 1 Let T be a translation by which a binary number translated into the corresponding decimal number. Then, for any binary number i, $|T(i)| = 1$. Therefore, T is finite and 1-bounded.

Example 2 Let $T = \{(\neg^n A, \neg^{n+2m} A) \mid n, m \geq 0\} \cup \{(\neg^n A, \neg^{n-2m} A) \mid n \geq 2m \geq 0\}$ be a translation. We can regard T as the logically equivalent transformation with respect to negation. When $\neg\neg\neg A$ is given, it translates into $\neg A, \neg\neg\neg A, \neg\neg\neg\neg\neg A, \ldots$. Thus, clearly, T is not finite.

The finiteness problem for translations is, for given translation T, the problem for deciding whether T is finite or not. Next, we show that, in general, the finiteness problem is unsolvable. For any non-negative integer i, we define

$$T_i = \{(w_1, w_2) \in \Sigma^+ \times \Sigma^+ \mid \varphi_i(\overline{(w_1, w_2)}) = 1\},$$

where $\overline{(w_1, w_2)}$ is a code of the string (w_1, w_2), and φ_i is a partial recursive function computed by the Turing machine M whose Gödel number is i. It is obvious that T_i defines a translation. Then, we get the following theorem.

Theorem 3 *For arbitrary non-negative integer i, the problem for deciding whether T_i is* *finite or not is unsolvable.*

Proof: Let P be the set of all partial recursive functions with arity 1. We define

$$G = \{x \mid T_x \text{ is a finite translation}\}, \qquad C = \{\varphi_x \mid x \in G\}.$$

Since there exists a finite translation as shown in Example 1, $C \neq \emptyset$. On the other hand, since there exists a translation which is not finite as shown in Example 2, $C \neq P$. Thus, G is not a recursive set by Rice's theorem [8]. For any non-negative integer x, T_x is finite if and only if $x \in G$. Therefore, the problem for deciding whether T_x is finite or not is unsolvable. □

Similarly, for an arbitrary non-negative integer i, the problem for deciding whether T_i is k-bounded for some fixed k or not is unsolvable.

4 Classes of translations defined by EFS's

In this section, we introduce some classes of translations defined by EFS's and give some conditions under which the translation is finite and 1-bounded.

We define a *translation EFS* (*TEFS*, for short) as a EFS with at least one predicate symbol with arity 2. Let $S = (\Sigma, \Pi, \Gamma)$ be a TEFS, and $p \in \Pi$ be a predicate symbol with arity 2. Then, we define

$$T(S,p) = \{(w_1, w_2) \in \Sigma^+ \times \Sigma^+ \mid \text{ there exists a proof tree of } p(w_1, w_2) \text{ on } S\}.$$

A translation T is said to be *defined* by a TEFS S and a predicate symbol p if $T = T(S,p)$. For a translation T, if there exists a TEFS S such that $T = T(S,p)$ for some predicate symbol p, then T is said to be *definable* by TEFS's.

Arikawa *et al.* [5] introduced some classes of restricted EFS's: variable-bounded EFS's, length-bounded EFS's, regular EFS's and one-sided linear EFS's, and show that recursively enumerable, context-sensitive, context-free, regular languages in Chomsky hierarchy are definable by them, respectively. Furthermore, they introduced an important class which is called simple EFS. We can define various subclasses of TEFS's which correspond to these subclasses of EFS's. However, in this paper, we focus on the subclasses of TEFS's which correspond to simple EFS's and one-sided linear EFS's.

A TEFS $S = (\Sigma, \Pi, \Gamma)$ is *simple* if the arity of each predicate symbol in Π is 2 and each axiom of S is of the form $p(\pi_1, \pi_2) \leftarrow q_1(x_1, y_1), \dots, q_n(x_n, y_n)$, where $x_1, \dots, x_n, y_1, \dots, y_n$ are mutually distinct variables, and $v(\pi_1) = \{x_1, \dots, x_n\}$ and $v(\pi_2) = \{y_1, \dots, y_n\}$. If each axiom satisfies that for any variable x, x occurs in the head of the axiom at most once, then the restricted simple TEFS's are equivalent to SDT's.

Proposition 4 *Let T be a translation defined by a simple TEFS $S = (\Sigma, \Pi, \Gamma)$ and $p \in \Pi$. If S has no axiom of the form $q(x, \pi) \leftarrow r(x, y)$, then T is finite, where x, y are variables and π is a term. Furthermore, S also has no axiom of the form $q(\pi, y) \leftarrow r(x, y)$, then T is bidirectionally finite.*

Proof: First, we show that for any integer $l \geq 1$, if $q(w, w') \in T_s\!\uparrow\! l - T_s\!\uparrow\!(l-1)$ then $|w| \geq l$, by the induction on l. If $q(w, w') \in T_s\!\uparrow\! 1 - T_s\!\uparrow\! 0$ then $q(w, w') \leftarrow \in \Gamma$ from the definition of $T_s\!\uparrow\! n$. From the definition of simple TEFS's, the length of w is more then or equal to 1. If $q(w, w') \in T_s\!\uparrow\!(k+1) - T_s\!\uparrow\! k$ then there exists a ground instance $q(w, w') \leftarrow r_1(u_1, v_1), \ldots, r_m(u_m, v_m)$ of an axiom of S and $\{r_1(u_1, v_1), \ldots, r_m(u_m, v_m)\} \subseteq T_s\!\uparrow\! k$. On the other hand, there exists i $(1 \leq i \leq m)$ such that $r_i(u_i, v_i) \in T_s\!\uparrow\! k - T_s\!\uparrow\!(k-1)$. If all $r_j(u_j, v_j)$ $(1 \leq j \leq m)$ are in $T_s\!\uparrow\!(k-1)$ then $q(w, w') \in T_s\!\uparrow\! k$. This contradicts with the assumption. Then, there exists i such that. The length of u_i is more than or equal to k by the inductions assumption. Since Γ has no axiom of the form $q'(x, \pi) \leftarrow r(x, y)$, the length of w is more than or equal to $k+1$.

Next, we prove that for any non-negative integer l, the set $T_s\!\uparrow\! l$ is finite by the induction on l. From the definition, $T_s\!\uparrow\! 0$ is finite. For any $k \geq 1$, $T_s\!\uparrow\! k$ is the set of all ground atoms $q(w, w')$ such that $q(w, w') \leftarrow r_1(u_1, v_1), \ldots, r_m(u_m, v_m)$ is a ground instance of an axiom and $\{r_1(u_1, v_1), \ldots, r_m(u_m, v_m)\} \subseteq T_S\!\uparrow\!(k-1)$. Note that, for any axiom $C = q(\tau, \tau') \leftarrow r_1(\pi_1, \pi'_1), \ldots, r_m(\pi_m, \pi'_m)$ of a simple TEFS, if ground atoms $r_1(u_1, v_1), \ldots, r_m(u_m, v_m)$ are given, then the instance of the head of C is ground and uniquely determined, because each variable which occurs in the head of C must be occur in the body of C. Since $T_s\!\uparrow\!(k-1)$ is finite by the inductions assumption, the number of all ground instances of axioms such that all ground atoms in their bodies are in $T_s\!\uparrow\!(k-1)$ is finite. Thus, $T_s\!\uparrow\! k$ is finite.

Since, for any $w \in \Sigma^+$, if there exists a proof tree of $p(w, w')$ on S then $p(w, w')$ is an element of $T_s\!\uparrow\!|w|$ and $T_s\!\uparrow\!|w|$ is finite, $T(w)$ is finite.

The latter part of the statement can be proved similarly. □

A simple TEFS is *right linear* if each axiom of the TEFS is of one of the following forms:

1. $p(u, v) \leftarrow$,
2. $p(ux, vy) \leftarrow q(x, y)$,

where $u, v \in \Sigma^+$.

Similarly, we can define *left linear TEFS's* by replacing the second condition with $p(xu, yv) \leftarrow q(x, y)$. We refer to a right linear TEFS or a left linear TEFS as a *one-sided linear TEFS*.

If a TEFS is one-sided linear then the TEFS is simple. Hence, we obtain the following proposition.

Proposition 5 *A translation which is definable by one-sided linear TEFS's is bidirectionally finite.*

Proof: A one-sided linear TEFS satisfies the both conditions in Proposition 4. □

A one-sided linear TEFS $S = (\Sigma, \Pi, \Gamma)$ is *deterministic* if, for each $p \in \Pi$ and each $a \in \Sigma$, Γ includes at most one clause whose head is of the form $p(a\pi, \tau)$ ($p(\pi a, \tau)$), where π and τ are terms. The following proposition is directly obtained from the definition.

Proposition 6 *A translation which is definable by deterministic one-sided linear TEFS's is* 1*-bounded.*

5 The finiteness problem for simple TEFS's

In previous section, we showed that, for an arbitrary translation, the finiteness of the translation is undecidable. In contrast, in this section, we show that the problem is solvable in the class of translations which is definable by simple TEFS's. First, we introduce a reduced form of a simple TEFS w.r.t. a translation which is defined by the TEFS. We show that the finiteness problem for translations which are definable by reduced simple TEFS's is solvable. Furthermore, any simple TEFS can be shown to be transformed into a reduced form w.r.t. a translation defined by the original TEFS. In consequence, we can obtain the result that the finiteness of a translation defined by a simple TEFS is decidable.

Let T be a translation defined by a TEFS $S = (\Sigma, \Pi, \Gamma)$ and a predicate symbol $p \in \Pi$. We say that a predicate symbol $q \in \Pi$ is *useless w.r.t.* T if, there exists no element (w_1, w_2) of T such that q occurs in the proof tree of $p(w_1, w_2)$ on S. We say that q is *useful w.r.t.* T if q is not useless.

Example 7 Let $T = \{(a^n, b^n) \mid n \geq 1\}$ be a translation and $S = (\{a, b\}, \{p, r_1, r_2\}, \Gamma)$ be a TEFS, where

$$\Gamma = \left\{ \begin{array}{l} p(ax, by) \leftarrow p(x, y) \\ p(ax, by) \leftarrow r_1(x, y) \\ r_2(ax, cy) \leftarrow p(x, y) \\ p(a, b) \leftarrow \end{array} \right\}.$$

Then, r_1 and r_2 are useless w.r.t. T.

Since useless predicates are not necessary to define the translation, we can remove the useless predicates from the simple TEFS.

Let T be a translation defined by a simple TEFS $S = (\Sigma, \Pi, \Gamma)$. We say that S is *reduced w.r.t.* T if S satisfies the following conditions:

1. Π has no useless predicate symbol w.r.t. T.
2. There is no axiom whose head is of the form $q(x, y)$, where x and y are variables.

We can prove the following proposition [10].

Proposition 8 *Any simple TEFS* $S = (\Sigma, \Pi, \Gamma)$ *can be transformed into the reduced TEFS* S' *w.r.t.* $T(S, p)$ *such that* $T(S', p) = T(S, p)$. *and* $p \in \Pi$. *We can*

We show that the finiteness problem for translations which are definable by reduced simple TEFS's is solvable. For a simple TEFS $S = (\Sigma, \Pi, \Gamma)$, a *level mapping* of S is a total function from Π to the set of integers.

Lemma 9 *Let* T *be a translation defined by a reduced simple TEFS* $S = (\Sigma, \Pi, \Gamma)$ *w.r.t.* T. *Let* Γ' *be the set of all the clauses* $C \in \Gamma$ *such that the head of* C *is of the form* $q(x, \pi)$. *If* T *is finite then there exists a level mapping* f *such that* $f(q) > f(r)$ *for any clause* $q(x, \pi) \leftarrow r(x, y) \in \Gamma'$.

Proof: We prove the contraposition of the statement in this lemma. We assume that there exists no level mapping f such that $f(q) > f(r)$ for any $q(x, \pi) \leftarrow r(x, y) \in \Gamma'$. Then, there exist clauses

$$\begin{array}{rcl} p_1(x_1, \alpha_1 y_1 \beta_1) & \leftarrow & p_2(x_1, y_1), \\ p_2(x_2, \alpha_2 y_2 \beta_2) & \leftarrow & p_3(x_2, y_2), \\ & \vdots & \\ p_l(x_l, \alpha_l y_l \beta_l) & \leftarrow & p_1(x_l, y_l) \end{array}$$

in Γ' such that either α_i or β_i is not ϵ for each i $(1 \leq i \leq l)$. By the definition of reduced TEFS's, there exist ground terms u and v such that $p_1(u, v)$ occurs in the proof tree of $p(w_1, w_2)$ for some $w_1, w_2 \in \Sigma^+$. Note that, from the definition of simple TEFS's, for any derivation tree of $p(w_1, w_2)$ on S, if a ground atom $q(u, v)$ occurs in the derivation tree, then u, v are substrings of w_1, w_2, respectively. Since $p_1(u, v)$ is in $T_S\!\uparrow\!\omega$, all $p_l(u, \alpha_l v \beta_l), \ldots, p_2(u, \alpha_2 \ldots \alpha_l v \beta_l \ldots \beta_2)$ are in $T_S\!\uparrow\!\omega$, from the definition of $T_S\!\uparrow\!\omega$. Furthermore, for any non-negative integer k, all $p_1(u, \alpha^k v \beta^k)$ is in $T_S\!\uparrow\!\omega$, where $\alpha = \alpha_1 \ldots \alpha_l$ and $\beta = \beta_l \ldots \beta_1$.

Let P be a proof tree of $p(w_1, w_2)$ on S in which $p_1(u, v)$ occurs. In above discussion, we showed that such a proof tree exists. Now we construct a proof tree of $p(w_1, w_2^k)$ on S in which $p_1(u, \alpha^k v \beta^k)$ occurs, for each k. Let $r_0(u_0, v_0), \ldots, r_m(u_m, v_m)$ be ancestors of $p_1(u, v)$ such that $r_i(u_i, v_i)$ is a parent of $r_{i+1}(u_{i+1}, v_{i+1})$ for each $i \geq 0$. Let $\hat{v}_{m+1}$ be $\alpha^k v \beta^k$, and v_{m+1} be v.

For each i $(i = m, \ldots, 0)$, we construct $\hat{v}_i$ from $\hat{v}_{i+1}$ as follows. Let θ_i be a substitution and $r_i(\pi_1, \pi_2) \leftarrow s_1(x_1, y_1), \ldots, s_n(x_n, y_n)$ be an axiom of S which satisfy following conditions.

1. θ_i is an unifier of $r_i(\pi_1, \pi_2)$ and $r_i(u_i, v_i)$.
2. In the proof tree P, the children of $r_i(u_i, v_i)$ are $s_1(x_1, y_1)\theta_i, \ldots, s_n(x_n, y_n)\theta_i$.

Since there exists j $(1 \leq j \leq n)$ such that $s_j(x_j, y_j)\theta_i = r_{i+1}(u_{i+1}, v_{i+1})$, let $\hat{\theta}_i = \theta_i - \{y_j / v_{i+1}\} \cup \{y_j / \hat{v}_{i+1}\}$ and $\hat{v}_i = \pi_2 \hat{\theta}_i$.

Let P_k be the tree such that

1. The subtree of P whose root node is $p_1(u, v)$ is replaced by a proof tree of $p_1(u, \alpha^k v \beta^k)$ on S, and
2. Nodes $r_0(u_0, v_0), \ldots, r_m(u_m, v_m)$ in P replaced by $r_1(u_1, \hat{v}_1), \ldots, r_m(u_m, \hat{v}_m)$.

Then, P_k is a proof tree of $p(w_1, w_2^k)$. For any non-negative integers j_1 and j_2, if $j_1 \neq j_2$ then $w_2^{j_1} \neq w_2^{j_2}$, because either α_i or β_i is not ε for each i. Since all w_2^k $(k \geq 0)$ are target sentences of w_1, T is not finite. $\square$

Lemma 10 *Let T be a translation defined by a simple TEFS $S = (\Sigma, \Pi, \Gamma)$ and Γ' be the set of all the clauses $C \in \Gamma$ such that the head of C is of the form $q(x, \pi)$. If there exists a level mapping f such that $f(q) > f(r)$ for any clause $q(x, \pi) \leftarrow r(x, y) \in \Gamma'$ then T is finite.*

Proof: Without loss of generality, we can assume that f is a mapping from Π to $\{1, 2, \ldots, m\}$, where $m = |\Pi|$. For any integer $l \geq 1$, if a ground atom $p(w, w')$ is an element of $T_S\!\uparrow\!(l \cdot m + 1) - T_S\!\uparrow\!(l \cdot m)$ then $|w| \geq l + 1$. Hence, for any $w, w' \in \Sigma^+$, if there exists a proof tree of $p(w, w')$ on S then $p(w, w')$ is an element of $T_S\!\uparrow\!|w| \cdot m$. Moreover, for any non-negative integer l, $T_S\!\uparrow\! l$ is finite. Therefore, T is finite. $\square$

Let T be a translation defined by a simple TEFS $S = (\Sigma, \Pi, \Gamma)$ which is reduced w.r.t. T. Let Γ' be the set of all the clauses $C \in \Gamma$ such that the head of C is of the form $q(x, \pi)$. From above two lemmas, T is finite if and only if there exists a level mapping f such that $f(q) > f(r)$ for any clause $q(x, \pi) \leftarrow r(x, y) \in \Gamma'$. Then, we can obtain following theorem.

Theorem 11 *The finiteness problem for the translations which is definable by simple TEFS's is solvable.*

Proof: Let T be a translation defined by a simple TEFS $S = (\Sigma, \Pi, \Gamma)$ and $p \in \Pi$. Suppose that $S' = (\Sigma, \Pi', \Gamma')$ is a reduced TEFS of S w.r.t. T. Let Γ'' be the set of all clauses $C \in \Gamma'$ such that the head of C is of the form $q(x, \pi)$, and the number of all predicate symbols occurring in Γ'' be m. The number of patterns of assignment $1, \ldots, m$ to each predicate symbol is finite. If, there exists a level mapping f such that $f(q) > f(r)$ for any axiom $q(x, \pi) \leftarrow r(x, y)$, then T is finite, and if there is not such a level mapping then T is not finite. The time for constructing S' and Γ'' is finite. Therefore, the finiteness problem is solvable. $\square$

6 Conclusion

We showed that an EFS has good properties as a translation grammar. Especially, we focused on the finiteness of translations. We formalized the finiteness problem for translations, and show that the problem is solvable in the class of translations defined by restricted EFS's called simple TEFS's. A simple TEFS is so rich that it can define over languages in larger class than that of context free languages. Furthermore, we give the class of translations in which the number of target sentence is at most one.

We can determine whether a pair of strings is an element of a translation defined by a simple TEFS, by the derivation procedure [11]. It is future work that we formalize a procedure to produce the target sentence from a source sentence. On the other hand, Arikawa *et al.* [3, 5] showed that EFS is a good framework for language learning. We will discuss learning translations in various classes by EFS's.

References

[1] A.V. Aho and J.D. Ullman, Properties of syntax directed translations, *J. Comput. System Sci.*, Vol. 3, 1969, pp. 319–334.

[2] A.V. Aho and J.D. Ullman, Syntax directed translation and the pushdown assembler, *J. Comput. System Sci.*, Vol. 3, 1969, pp. 37–56.

[3] S. Arikawa, Elementary formal systems and formal language - simple formal systems, *Mem. Fac. Sci. kyushu Univ. Ser.A 24*, 1970, pp. 47–75.

[4] S. Arikawa, T. Shinohara and A. Yamamoto, Elementary formal systems as a unifying framework for language learning, In *Proc. COLT'89*, 1989, pp. 312–327.

[5] S. Arikawa, T. Shinohara and A. Yamamoto, Learning elementary formal systems, *Theoret. Comput. Sci.*, 1992, pp. 97–113.

[6] E.T. Irons, A syntax directed compiler for ALGOL-60, *CACM*, Vol. 4, 1961, pp. 51–55.

[7] P.M. Lewis and R.E. Stearns, Syntax-directed transduction, *J. ACM*, Vol. 15, 1986, pp. 465–488.

[8] H.G. Rice, Classes of recursively enumerable sets and their decision problem, *Trans.AMS89*, 1953.

[9] R.M. Smullyan, Theory of formal systems, *Princeton Univ. Press*, 1961.

[10] N. Sugimoto, Translations based on elementary forml systems and its learning, Master's thesis, Kyushu University, 1994, in Japanese.

[11] A. Yamamoto, Elementary formal system as a logic programming language. In *Logic Programming Conf.'89*, 1989, pp. 123–132.