TSUKUBA J. MATH.
Vol. 2 (1978). 11—26

A SYSTEM OF GRAPH GRAMMARS WHICH GENERATES ALL RECURSIVELY ENUMERABLE SETS OF LABELLED GRAPHS

By

Tadahiro UESU

Introduction

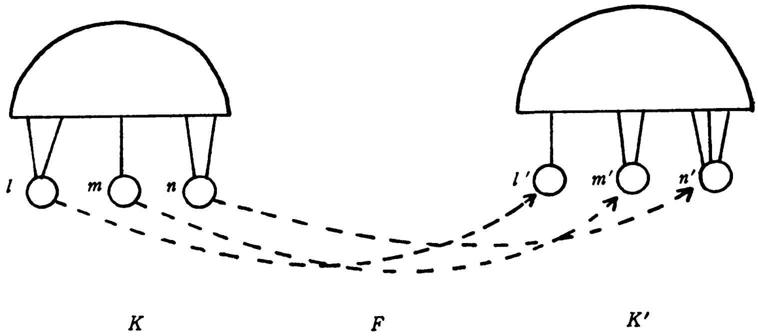
The aim of this paper is to prove that the system of simple graph grammars (Einfache Graph-Grammatiken) in [1] is complete. A graph grammar is a generalization of a Chomsky-grammar which defines a graph language, i.e. a language composed of a set of directed graphs with labelled nodes and arcs. A system of graph grammars is said to be complete if it satisfies the following property: For each finite set T of labels, the class of all graph languages over T defined by graph grammars of the system is identical with the class of all recussively enumerable sets of labelled graphs over T. Where a set of labelled graphs over T is recursively enumerable if the associated set of integers by a Gödel numbering from the labelled graphs over T into the integers is recursively enumerable.

Several authors have introduced their system of graph grammars (e.g. [1], [2], [3], [4], [5], [6], [7]). However they did not refer to the completeness of their systems.

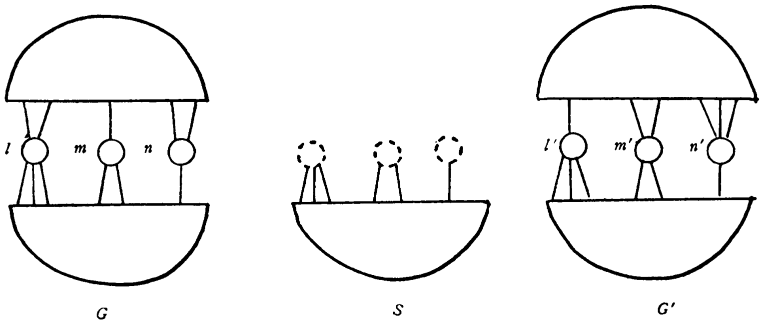
The system of simple graph grammars was introduced in [1] as a subsystem of the system in [2]. In this paper we modify the definition of the system in the following manner: A production of our system is an ordered triple (K, F, K') which consists of two labelled graphs K, K' and a one-to-one function F from a set of nodes of K to a set of nodes of K'. If (K, F, K') is a production, and if K occurs in a labelled graph G, then the production is applicable to G, and the effect of the application is to replace an occurrence of K in G by K'. The function F specifies the embedding of K' into the remainder of removing K from G; if a node n' of K' is the image of a node n of K, then n' is put upon the trace of n in the remainder. This is illustrated by Figure 0.

The formal definitions of labelled graphs and several concepts with labelled graphs are given in the first section of this paper. The definition of simple graph grammars is given in the second section. In the third section it is proved that the system of simple graph grammars is complete. The final section gives a generalization of the system of simple graph grammars.

Received November 29, 1977.



Production $(K,\ F,\ K')$. Straight lines represent arcs. Circles $l,\ m,\ n,\ l',\ m'$ and n' represent nodes. Dotted arrows represent the function F from $\{l,\ m,\ n\}$ to $\{l',\ m',\ n'\}$.



S is the remainder of removing K from G. Dotted circles represent the traces of the nodes $l,\ m,\ n$ of K. G' is the result of applying the production (K, F, K') to G.

Fig. 0.

In the following we list the notations of set theory and theory of formal languages which we will use.

If $\pi(x)$ is a proposition concerning x, then the symbol

$$\{x|\pi(y)\}$$

denotes the set of those elements x for which the proposition $\pi(x)$ is true. The *union* and the *difference* of sets A and B are denoted respectively by $A \cup B$ and $A-B$. The symbol 0 is used to denote the *empty set*. An *ordered n-tuple* is denoted by the symbol $(x_1,\ x_2, \cdots,\ x_n)$. The *membership relation* is denoted by ϵ. A *relation* is a set of ordered pairs. A *function* is a relation f such that for every x and for every pair $y,\ y'$ $(x,y) \epsilon f$ and $(x,y') \epsilon f$ implies $y=y'$. If f is a function from A onto B, then A is called the *domain* of f, written $\mathrm{dom}(f)$, and then B is called the *range* of f, written $\mathrm{rng}(f)$. If f is a function and x is an element of $\mathrm{dom}(f)$, then the *value* of f for the argument x is denoted by $f(x)$. If f is a function from A to B and g is a function from B to C, then the function h from A to C such that for every x in A $h(x)=g(f(x))$ is denoted by $g\circ f$.

An *alphabet* is a finite set of symbols. A *string* over an alphabet is a finite string composed of symbols from the alphabet. The symbol ε denotes the empty string. A *Chomsky-grammar* is an ordered triple (T, S, P) in which T is an alphabet, S an symbol not in T and P a finite set of expressions of the form $s\to t$, where s is a string with at most one symbol not in T and t is any string. If t, u and v are strings, and if s is a nonempty string, then the string utv is said to be *directly derived* from the string usv according to $s\to t$. The *language defined* by the Chomsky-grammar (T, S, P) is the set of all strings w over T satisfying that there exists a finite sequence $s_0, s_1, \cdots, s_n$ of strings such that s_0 is S, s_n is w and s_{i+1} is directly derived from s_i according to some element in P for $i=0, 1, \cdots, n-1$. The fact that *each recursively enumerable set of strings over an alphabet is a language defined by some Chomsky-grammar* is well known.

1. Labelled Graphs

DEFINITION 1.1. A *partially labelled graph* is an ordered sextuple $(N, A, \partial_s, \partial_t, \lambda_n, \lambda_a)$ where N and A are finite sets, ∂_s and ∂_t are functions from A to N, and λ_n and λ_a are functions such that $\mathrm{dom}(\lambda_n)$ is a subset of N and $\mathrm{dom}(\lambda_a)$ is a subset of A. A partially labelled graph $(N, A, \partial_s, \partial_t, \lambda_n, \lambda_a)$ is said to be *over* an alphabet T, if $\mathrm{rng}(\lambda_n)$ and $\mathrm{rng}(\lambda_a)$ are subsets of T.

Let G be a partially labelled graph $(N, A, \partial_s, \partial_t, \lambda_n, \lambda_a)$. The elements of N are called the *nodes* of G. The elements of A are called the *arcs* of G. If a is an *arc* of G, then the *node* $\partial_s(a)$ is called the *source* of a and the node $\partial_t(a)$ is called the *target* of a. If n is the source or the traget of an arc a, then n is said to be *incident* with a and a is said to be *incident* with n. The nodes in $\mathrm{dom}(\lambda_n)$ are said to be *labelled*, and the nodes in $N-\mathrm{dom}(\lambda_n)$ are said to be *unlabelled*. Similarly, the arcs in $\mathrm{dom}(\lambda_a)$ are said to be *labelled*, and the arcs in $A-\mathrm{dom}(\lambda_a)$ are said to be *unlalelled*. For each labelled node n, the value $\lambda_n(n)$ of λ_n is called the *label* of n. Similarly, for each labelled arc a, the value $\lambda_a(a)$ of λ_a is called the *label* of a.

DEFINITION 1.2. A partially labelled graph

$$(N, A, \partial_s, \partial_t, \lambda_n, \lambda_a)$$

is a *labelled graph* if and only if $\mathrm{dom}(\lambda_n)=N$ and $\mathrm{dom}(\lambda_a)=A$. The labelled graph

$$(0, 0, 0, 0, 0, 0)$$

is called the empty graph, and donoted by ε.

DEFINITION 1.3. Let G and G' be two labelled graphs

$$(N, A, \partial_s, \partial_t, \lambda_n, \lambda_a) \quad \text{and} \quad (N', A', \partial_s', \partial_t', \lambda_n', \lambda_a')$$

respectively. If there is a pair f_n, f_a of functions such that f_n is a one-to-one function from N onto N', f_a is a one-to-one function from A onto A', $\partial_s' \circ f_a = f_n \circ \partial_s$, $\partial_t' \circ f_a = f_n \circ \partial_t$, $\lambda_n = \lambda_n' \circ f_n$ and $\lambda_a = \lambda_a' \circ f_a$, then G is said to be *isomorphic* to G', and also, G and G' are said to be *isomophic*.

We define the concept of partitions of labelled graphs. For example, consider the labelled graph G as shown in Figure 1.1. If we remove the subgraph K from G, then the partially labelled graph S remains, where unlabelled nodes of S represent the traces of labelled nodes of K. Conversely, G is obtained by connecting suitably the unlabelled nodes of S to the labelled nodes of K.

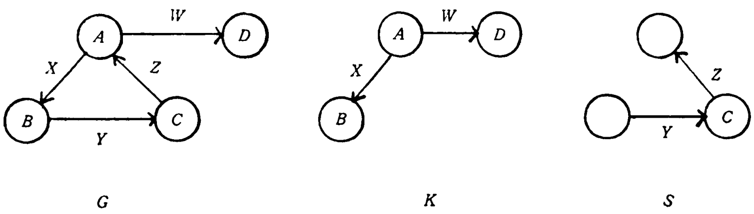


Fig. 1.1

Formally we define the following way:

DEFINITION 1.4. Let K be a labelled graph

$$(N^K, A^K, \partial_s^K, \partial_t^K, \lambda_n^K, \lambda_a^K)$$

and S a partially labelled graph

$$(N^S, A^S, \partial_s^S, \partial_t^S, \lambda_n^S, \lambda_a^S)$$

without unlabelled arc such that $N^K \cap N^S = 0$ and $A^K \cap A^s = 0$. A *coupling* between K and S is a one-to-one function from the set of all unlabelled nodes of S to the set N^K of labelled nodes of K. If f is a coupling between K and S, and if a labelled graph H is isomorphic to the labelled graph

$$(N, A, \partial_s, \partial_t, \lambda_n, \lambda_a)$$

in which

$$\mathrm{N} = N^K \cup (N^S - \mathrm{dom}(f))$$

$$A = A^K \cup A^S$$

$$\partial_s = \partial_s^S - \{(a, n) \mid (a, n) \in \partial_s^S \text{ and } n \text{ is an unlabelled node of } S\}$$

$\cup\{(a, f(n)) \mid (a, n)\in\partial_s{}^S$ and n is an unlabelled node of $S\}$
$\cup\,\partial_s{}^K$,

$\partial_t=\partial_t{}^S-\{(a, n) \mid (a, n)\in\partial_t{}^S$ and n is an unlabelled node of $S\}$
$\cup\{(n, f(n)) \mid (a, n)\in\partial_t{}^S$ and n is an unlabelled node of $S\}$
$\cup\,\partial_t{}^K$,

$$\lambda_n=\lambda_n{}^K\cup\lambda_n{}^S \quad \text{and} \quad \lambda_a=\lambda_a{}^K\cup\lambda_a{}^S,$$

then the ordered triple (K, S, f) is called a *simple partition* of H, and K and S are called respectively the *kernel* and the *shell* of the simple partition (K, S, f).

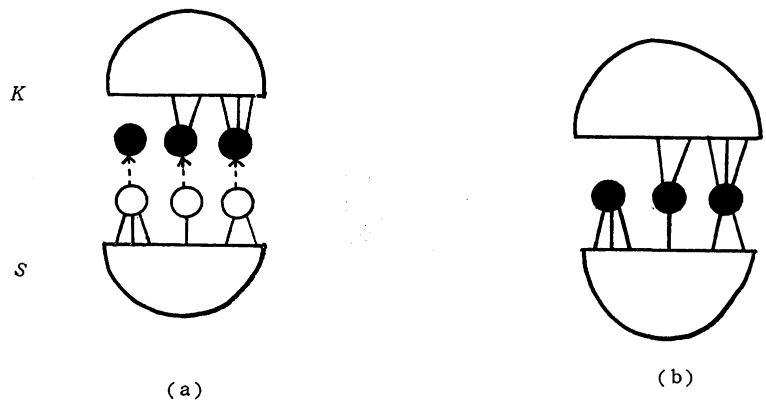


Fig. 1.2 (a) denotes a simple partition of (b). In (a) dotted arrows represent the coupling between K and S.

2. Simple Graph Grammars

DEFINITION 2.1. A *simple production* is an ordered triple (K_1, F, K_2) where K_1 and K_2 are labelled graphs, K_1 is not the empty graph, and F is a one-to-one function from a set of nodes of K_1 to the set of nodes of K_2. K_1, K_2 and F are called respectively the *left kernel*, the *right kernel* and the *embedding* of the simple production (K_1, F, K_2),

EXAMPLE 2.1. Let K_1 and K_2 be two labelled graphs

$$(\{0, 1\}, \{0\}, \{(0, 0)\}, \{(0, 1)\}, \{(0, A), (1, B)\}, \{(0, X)\})$$

and

$$(\{0, 1, 2\}, \{0, 1\}, \{(0, 0), (1, 1)\}, \{(0, 1), (1, 2)\}, \{(0, C), (1, D), (2, E)\}, \{(0, Y), (1, Z)\})$$

respectively and let K_3 be the empty graph ε. Consider the simple productions (K_1, F_1, K_2), (K_1, F_2, K_2), (K_1, F_3, K_2) and (K_1, F_3, K_3) where F_1 and F_2 are the function $\{(0, 0), (1, 2)\}$, $\{(0, 0)\}$ respectively and F_3 is the empty function 0. The

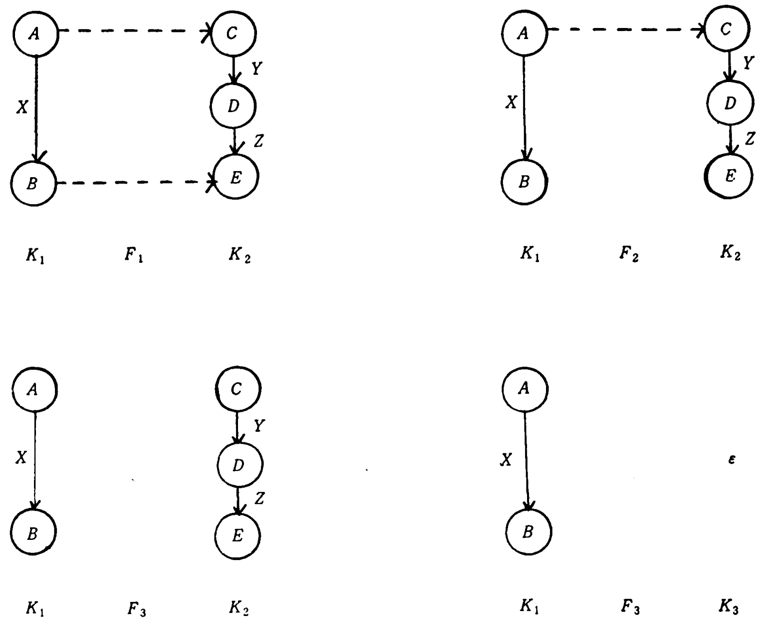


Fig. 2.1

corresponding diagrams are as shown in Figure 2.1.

DEFINITION 2.2. A labelled graph G_2 is *directly derived* from a labelled graph G_1 according to a simple production (K_1, F, K_2) if there exist simple partitions (K_1, S, f_1) of G_1 and (K_2, S, f_2) of G_2 such that $\mathrm{rng}(f_1)=\mathrm{dom}(F)$ and $F\circ f_1=f_2$. A labelled graph G is *derived* from a labelled graph H according to a set P of simple productions if there exists a finite sequence $H_0, H_1, \cdots, H_n$ of labelled graphs such that H_0 is H, H_n is G, and H_{i+1} is directly derived from H_i according to some simple production in P for $i=0, 1, \cdots, n-1$.

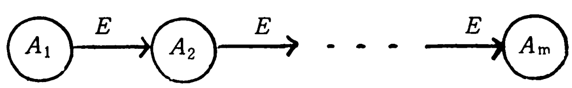
DEFINITION 2.3. A *simple graph grammar* over an alphabet T is an ordered triple (T, S, P) in which S is a symbol not in T and P is a finite set of simple productions whose left kernels are not over T.

In the following, we identify isomorphic labelled graphs.

DEFINITION 2.4. If $\mathfrak{S}$ is a simple graph grammar (T, S, P), then the set of all the labelled graphs over the alphabet T that are derived from the labelled graph, which consists only of one node with the label S, according to the set P of simple productions is called the *graph language defined* by $\mathfrak{S}$; it is denoted by $L(\mathfrak{S})$.

If $\mathfrak{S}$ is a simple graph grammar, then the graph language defined by $\mathfrak{S}$ is a countable set, since isomorphic labelled graphs are identified.

For each string $A_1, A_2, \cdots, A_m$, the labelled graph



is called the *graph-expression* of the string.

The following example explains the translation of Chomsky-grammars to simple graph grammars.

EXAMPLE 2.2. Consider the Chomsky-grammar

$$(\{A, B\}, S, \{S \to \varepsilon, S \to ASB\}),$$

and simple productions p_1, p_2, p_3 and p_4 as shown in Figure 2.2. The pair p_1, p_2 corresponds to $S \to \varepsilon$, and the pair p_3, p_4 corresponds to $S \to ASB$. Then the graph language defined by the simple graph grammar

$$(\{A, B, E\}, S, \{p_1, p_2, p_3, p_4\})$$

consists of all the graph-expressions of strings in the language defined by the Chomsky-grammar.

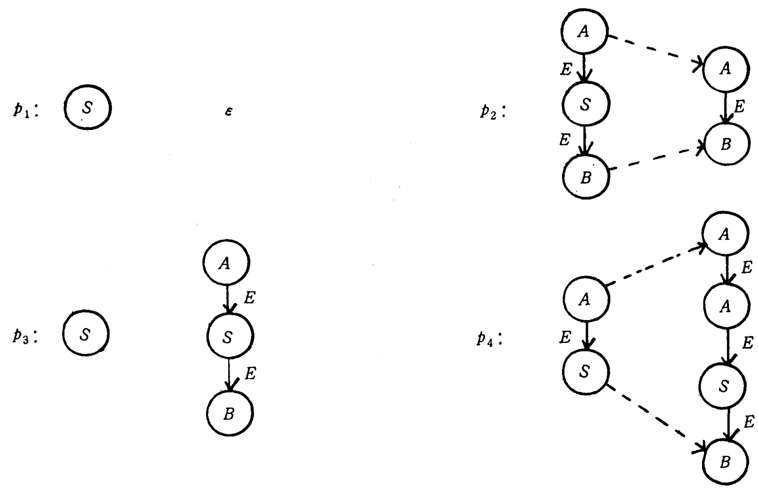


Fig. 2.2

It is generalized as the following proposition:

PROPOSITION 2.1. *For each Chomsky-grammar* $\mathfrak{C}$. *There exists a simple graph grammar* $\mathfrak{S}$ *such that the graph language* $L(\mathfrak{S})$ *defined by* $\mathfrak{S}$ *consists of all graph-expressions of strings in the language* $L(\mathfrak{C})$ *defined by* $\mathfrak{C}$.

3. The Completeness of the System of Simple Graph Grammars

Throughout this section we identify isomorphic labelled graphs, and we assume that each set of nodes is a set of positive integers.

DEFINITION 3.1. The *representing pair* of a node n of a labelled graph is the ordered pair (n, A), in which A is the label of n. The *representing sextuple* of an arc a of a labelled graph is the ordered sextuple (m, A, a, B, n, C) in which m

and n are the source and the traget of a respectively, A and C are the labels of m and n respectively, and B is the label of a.

If α is a representing sextuple of the form (m, A, a, B, n, C), then $\bar{\alpha}$ denotes the string

$$A^\circ \underbrace{11\cdots1}_{m+1} B^\circ \underbrace{11\cdots1}_{n+1} C^\circ.$$

If β is a representing pair of the form (n, A), then $\bar{\beta}$ denotes the string

$$\underbrace{22\cdots2}_{n+1} A^\circ.$$

DEFINITION 3.2. If $\alpha_1, \alpha_2, \cdots, \alpha_r$ is a sequence without repetition of all the representing sextuples of arcs of a labelled graph G and $\beta_1, \beta_2, \cdots, \beta_s$ is a sequence without repetition of all the representing pairs of nodes of G, then the string

$$\bar{\alpha}_1\bar{\alpha}_2\cdots\bar{\alpha}_r\bar{\beta}_1\bar{\beta}_2\cdots\bar{\beta}_s$$

is called a *string-expression* of G. The empty string is the string-expression of the empty graph.

PROPOSITION 3.1. *If $\mathfrak{C}$ is a Chomsky-grammar such that the lauguage $L(\mathfrak{C})$ defined by $\mathfrak{C}$ is a set of string-expressions of labelled graphs, then there exists a simple graph grammar $\mathfrak{S}$ such that*

(a) *euery string in $L(\mathfrak{C})$ is a string-expression of some labelled graph in $L(\mathfrak{S})$, and*

(b) *every labelled graph in $L(\mathfrak{S})$ has a string-expression in $L(\mathfrak{C})$.*

PROOF. Let $\mathfrak{C}$ be a Chomsky-grammar such that the language $L(\mathfrak{C})$ defined by $\mathfrak{C}$ is a set of string-expressions of labelled graphs over an alphabet T and let $\mathfrak{S}_0$ be a simple graph grammar (T_0, S, P_0), in which

$$T_0=\{A^\circ|A\in T\}\cup\{1, 2, E\},$$

such that the graph language $L(\mathfrak{S}_0)$ defined by $\mathfrak{S}_0$ consists of all the graph-expressions of strings in $L(\mathfrak{C})$. The existence of such $\mathfrak{S}_0$ is assured by proposition 2.1. Let P_1, P_2 and P_3 be the sets of simple productions as shown in Figure 3.1, Figure 3.2 and Figure 3.3 respectively. We may assume that the symbols $1'$, $2'$ and A', for A in T, are not contained in any kernels of simple productions in P_0, and

$$(\{A^\circ|A\in T\}\cup\{A'|A\in T\}\cup\{1, 2, 1', 2', E, S\})\cap T=0.$$

Set

$$\mathfrak{S}=(T, S, P_0\cup P_1\cup P_2\cup P_3).$$

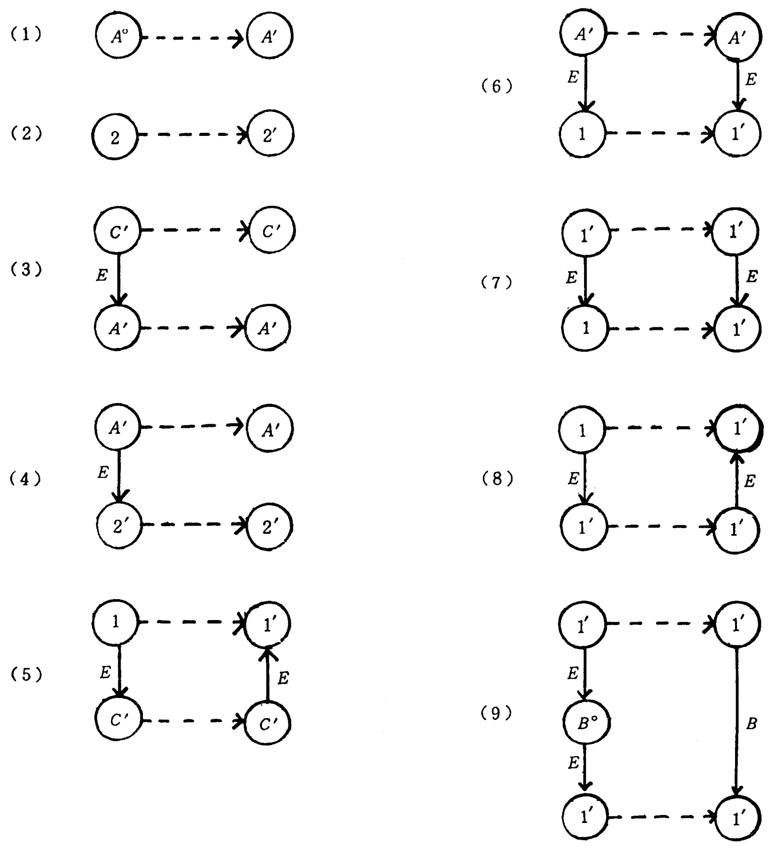


Fig. 3.1 The set P_1 of simple productions. A, B and C are symbols from T.

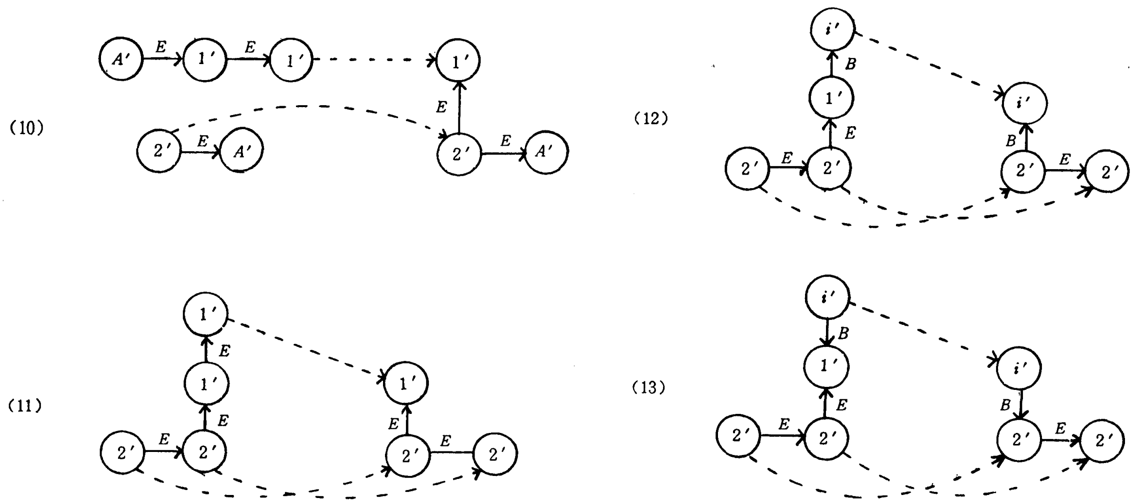
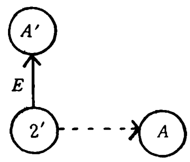


Fig. 3.2 The set P_2 of simple productions. A and B are symbols from T and $i=1,2$.

(14)



(15)

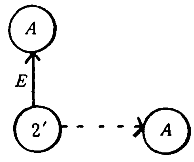


Fig. 3.3 The set P_3 of simple productions. A is a symbol from T.

We shall prove (a) and (b). Let G be a labelled graph over T as shown in Figure 3.4 and let the set $\{1, 2, \cdots, t\}$ be the set of nodes of G.

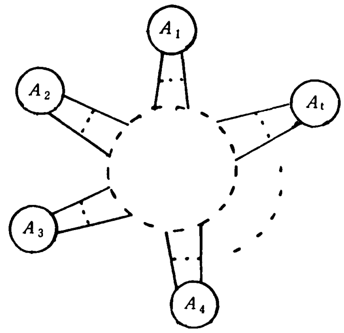
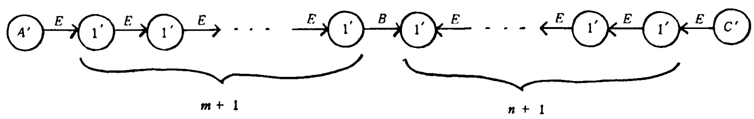
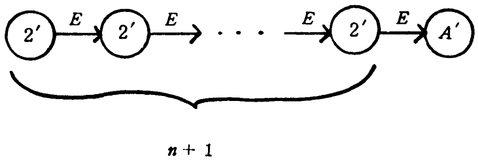


Fig. 3.4 Labelled graph G. A_i is the label of the node i of G for $i=1, 2, \cdots, t$. Straight lines represent arcs of G.

For each representing sextuple α of an arc a of G of the form (m, A, a, B, n, C), let $\hat{\alpha}$ denote the following labelled graph:



For each representing pair β of a node n of G of the form (n, A), let $\hat{\beta}$ denote the following labelled graph:



Let

$$\bar{\alpha}_1\bar{\alpha}_2\cdots\bar{\alpha}_r\bar{\beta}_1\bar{\beta}_2\cdots\bar{\beta}_s$$

be a string expression of G where α_i is a representing sextuple for $i=1, 2, \cdots, r$ and β_j is a representing pair for $j=1, 2, \cdots, s$. Let H be the graph-expression of the string-expression

$$\bar{\alpha}_1\bar{\alpha}_2\cdots\bar{\alpha}_r\bar{\beta}_1\bar{\beta}_2\cdots\bar{\beta}_s,$$

and let H_1 be labelled graph of the form

$$\hat{\alpha}_1\ \hat{\alpha}_2\ \cdots\ \hat{\alpha}_r\ \hat{\beta}_1\ \hat{\beta}_2\ \cdots\ \hat{\beta}_s.$$

Then H_1 is derived from H according to P_1. If $(1, A_1)$ is the representing pair of the node 1 of G, then H_1 is of the form as shown in Figure 3.5. Hence, the labelled graph $H_1{}^1$ in Figure 3.5 is derived from H_1 according to P_2.

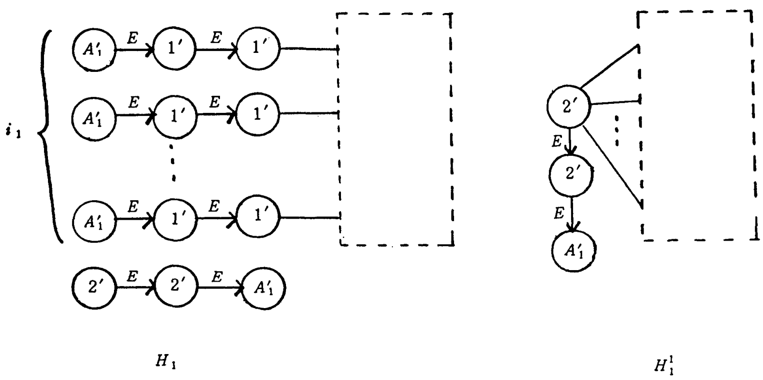


Fig. 3.5 Straight lines represent arcs with labels in T and each dotted square represents the remainder of H_1. i_1 is the sum of the number of arcs whose source is 1 and the number of arcs whose target is 1.

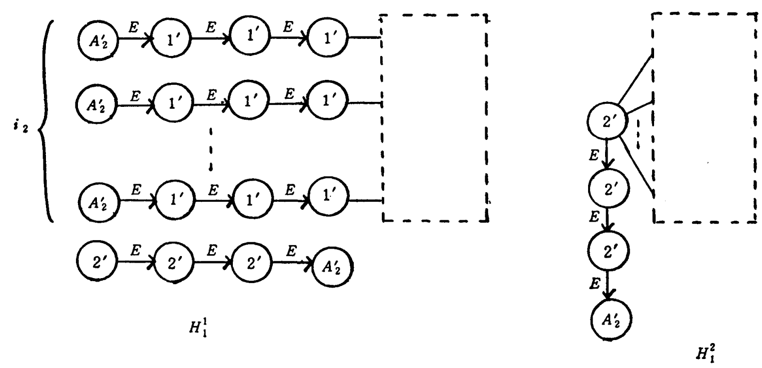


Fig. 3.6 Straight lines represent arcs with labels in T and each dotted square represents the remainder of $H_1{}^1$. i_2 is the sum of the number of arcs whose source is 2 and the number of arcs whose target is 2.

If $(2, A_2)$ is the representing pair of the node 2 of G, then $H_1{}^1$ is of the form as shown in Figure 3.6, so that the labelled graph $H_1{}^2$ in Figure 3.6 is derived from $H_1{}^2$ according to P_2. In this way we get the labelled graph H_2 as shown in Figure 3.7 which is derived from H_1 according to P_2.

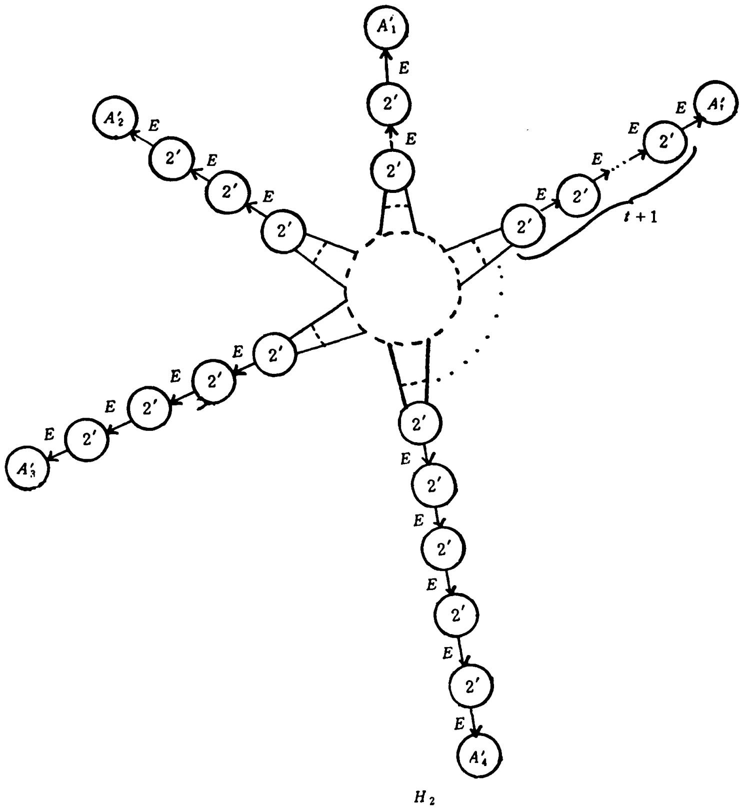


Fig. 3.7

Clearly the labelled graph G is derived from H_2 according to P_3. Therefore G is derived from H according to $P_1 \cup P_2 \cup P_3$. Thus the condition (a) holds. It is easily seen that for each labelled graph H in $L(\mathfrak{S}_0)$, there exists uniquely a labelled graph G in $L(\mathfrak{S})$ such that G is derived from H according to $P_1 \cup P_2 \cup P_3$. Conversely, for each labelled graph G in $L(\mathfrak{S})$, there exists a labelled graph H in $L(\mathfrak{S}_0)$ such that G is derived from H according to $P_1 \cup P_2 \cup P_3$. Therefore the condition (b) holds. This completes the proof.

From the above proposition we are able to deduce the main result.

THEOREM. *For each alphabet the class of all recursively enumerable sets of labelled graphs over the alphabet and the class of all graph language defined by simple graph grammars over the alphabet are identical.*

PROOF. It is clear that a graph language defined by a simple graph grammar is recursively enumerable set of labelled graphs over an alphabet T. Then the set E' of all the string-expressions of labelled graphs in E is recursively enumerable, and so there is a Chomsky-grammar $\mathfrak{C}$ such that the language defined by $\mathfrak{C}$ is the set E'. Therefore, by virtue of proposition 3.1, there exists a simple graph grammar $\mathfrak{S}$ such that

$$L(\mathfrak{S}) = E.$$

4. General Productions

In our system of simple graph grammars, a simple production consists of two kernels which are labelled graphs and an embedding which is a partial fuction from the set of nodes of the left kernel to the set of nodes of the right kernel. This can be generalized in the following way; we allow the embedding of a production to be a *relation* between the set of nodes of the left kernel and the set of nodes of the right kernel. If (K, R, K') is such a production and K occurs in a labelled graph G, then the application of the production is to replace an occurrence of K in G by K' such that if a is an arc of the remainder of removing the occurrence of K from G and has been incident with a node n of K, then a is connected with some node n' of K' which satisfies $(n, n') \in R$. Moreover, we allow the kernel of a production to be partially labelled graphs without unlabelled arc. Then such a production may be regarded as a version of a Schneider-Ehrig's production in [7], since a labelled partial graph in [7] may be interpreted as a partially labelled graph without unlabelled arc.

We shall not give the precise definition of the generalized system here. However, we illustrate the direct derivation for the generalized system with two examples (cf. pp. 303–306 in [7]). Consider the the general production in Figure 4.1. and the labelled graph G_1 in Figure 4.2. Then the labelled graph G_2 is directly derived from G_1 as shown in Figure 4.2.

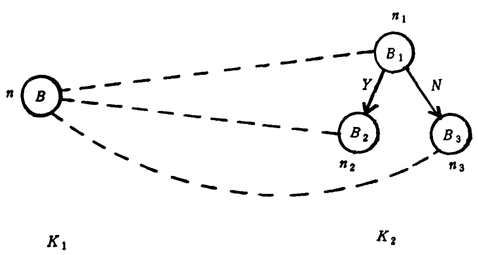


Fig. 4.1 Dotted lines represent the relation between the set $\{n\}$ of the node of K_1 and the set $\{n_1, n_2, n_3\}$ of nodes K_2.

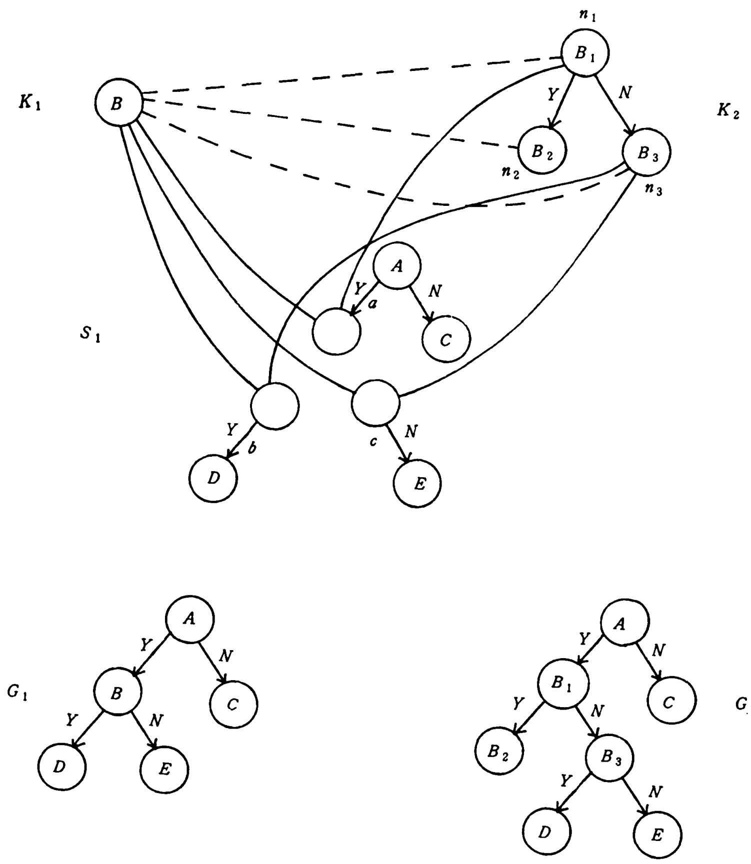


Fig. 4.2 The pair K_i, S_1 is a partition of G_i for $i=1,2$. Curved lines represent the couplings. The arcs a, b, c are distributed such that a is to the node n_1, and b and c are to the node n_3.

However, if we consider the partition of G_2' as shown in Figure 4.3, then G_2' is derived from G_1 according to the production also.

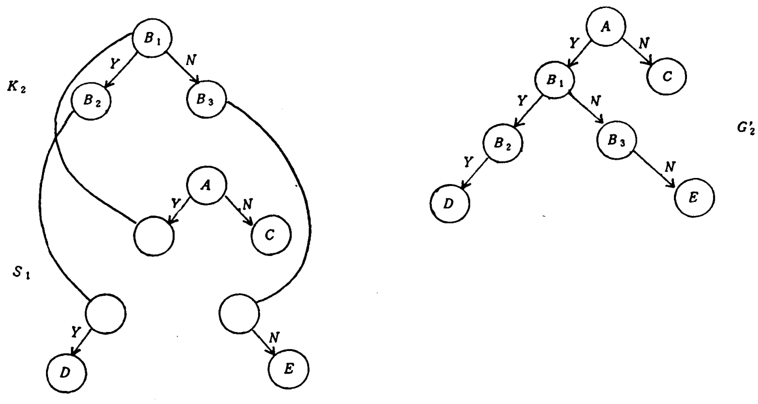


Fig. 4.3

Thus, 27 labelled graphs are directly derived from the labelled graph G_1 according to the production.

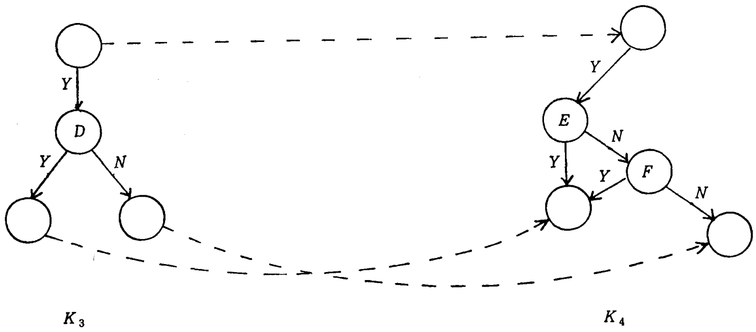


Fig. 4.4 Dotted arrows represent the one-to-one function from the set of unlabelled nodes of K_3 onto the set of unlabelled nodes of K_4.

Concider the general production in Figure 4.4. Then the labelled graph G_4 is directly derived from the labelled graph G_3 as shown in Figure 4.5.

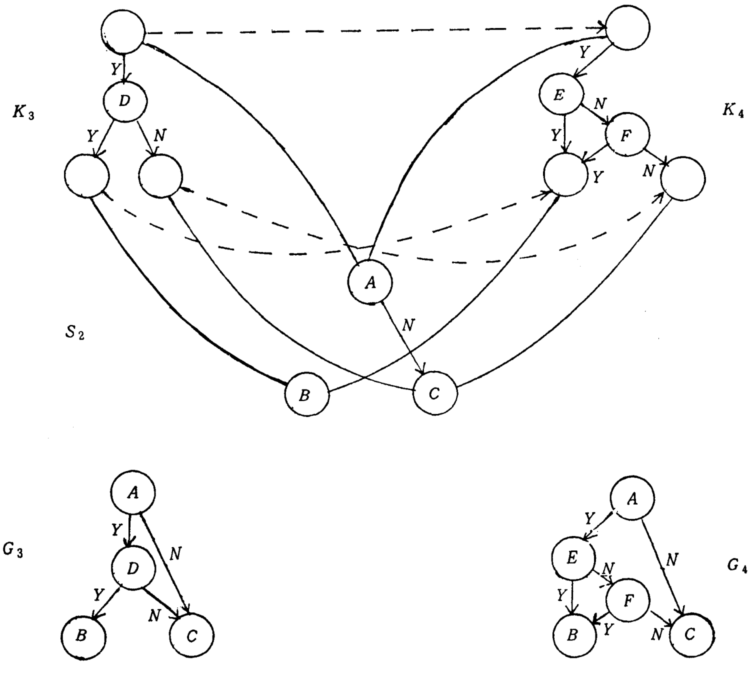


Fig. 4.5 The pair K_i, S_2 is a partition of G_i for $i=3, 4$. Curved lines represent the couplings.

References

[1] E. Denert & H. Ehrig, Mehrdimensionale Sprachen, Skript WS 1975/76, mimeographed notes.

[2] H. Ehrig, M. Pfender & H.J. Schneider, Graph-Grammars: an algebraic approach, Proc. 14th Ann. IEEE Symp. Switch. Automat. Theory, Iowa City, 1973, 167-180.

[3] U.G. Montanari, Separable Graphs, Planar Graphs and Web Grammars, Information and Control 16 (1970) 243-267.

[4] M. Nagl, Formal Languages of Labelled Graphs, Computing 16 (1976) 113-137.
[5] J.L. Pfaltz & A. Rosenfeld, Web grammars. Proc. 1st Inter. Joint Conf. on Artificial Intelligence, Washington, D.C., May 1969, 609-619.
[6] A. Rosenfeld & D.L. Milgram, Web Automata and Web Grammars, Machine Intellingence 7 (1972) 307-324.
[7] H.J. Schneider & H. Ehrig, Grammars on Partial Graphs, Acta Informatica 6 (1976) 297-316.

Department of Mathematics
Fuculty of Science
Kyushu University
Fukuoka, 812, Japan