HRNCE 문법의 언어 생성력

정 태 의† · 박 동 선††

요 약

스트링 문법은 그래프 언어의 부분집합인 스트링 언어를 생성하는 반면 그래프 문법은 그래프 언어를 생성한다. 그래프 문법 모델중 가장 성공적인 것중의 하나인 NLC 문법은 노드 레이블을 이용하여 하나의 노드를 하나의 그래프로 치환하므로서 그래프를 생성한다. 그래프를 포함하는 하이퍼그래프를 생성하는 하이퍼그래프 문법 모델로는 미리 정해놓은 접착점들의 순서를 이용하여 하나의 하이퍼에지를 하나의 하이퍼그래프로 치환하는 CFHG 문법, CFHG 문법의 확장형으로서 복제 및 삭제의 방법을 통해 하이퍼에지와 그것이 포함하고 있는 노드들, 즉 하나의 핸들을 하나의 하이퍼그래프로 치환하는 HH 문법, 그리고 eNCE 방식을 이용하여 하나의 핸들을 하나의 하이퍼그래프로 치환하는 HRNCE 문법 등이 소개되었다. 본 논문에서는 HRNCE 문법이 생성하는 그래프 언어와 위에서 언급된 기타의 그래프 문법들이 생성하는 그래프 언어들을 비교하므로서 HRNCE 문법과 타 그래프 문법들의 언어 생성력을 비교 검토하고자 한다.

Language-generating Power of HRNCE Grammars

Tae Eui Jeong† · Dong Sun Park††

ABSTRACT

Graph grammars generate graph languages while string grammars generate string languages which are the subset of graph languages. One of the most successful graph grammars models is the NLC grammars, which generate graphs by replacing a node by a graph through node labels. For grammars generating hypergraphs which are the superset of graphs, there are CFHG grammars, which replace a hyperedge by a hypergraph through their preidentified gluing points, an extension of CFHG grammars called HH grammars, which replace a handle by a hypergraph through the rewriting mechanism that can also duplicate or delete the hyperedges surrounding the replaced handle, and finally HRNCE grammars, which replace a handle by a hypergraph through an eNCE way of rewriting. In this paper, we compare the language-generating power of HRNCE grammars with that of graph grammars mentioned above by comparing graph languages generated by them, respectively.

1. Introduction

Graph grammars extend the traditional string grammars by replacing the left-handside of a production by graphs instead of strings and thus generate sets of graphs. They were originally introduced to describe picture patterns but soon found applications in many other subareas of computer science such as computer aided design, programming languages and compilers, databases, software specifications, concurrency, pat-

† 정 회 원:서경대학교 컴퓨터과학과 전임강사

†† 정 회 원:전북대학교 정보통신공학과 조교수

논문접수:1996년 7월 3일, 심사완료:1996년 11월 8일

tern recognition and computer vision. Various applications and related research efforts of graph grammars can be found in the proceedings of the international workshops on "Graph Grammars and Their Application to Computer Science"[2, 4-6].

One of the most successful graph grammars models is the node-label-controlled(NLC) grammars of Janssens and Rozenberg[14], which replaces a single node by a graph in a derivation step and the embedding of the newly introduced graph into the existing graph is based on node labels only. They can describe PSPACE-complete languages[15]. Many subclasses of NLC grammars such as B-NLC grammars[20] with improved properties have appeared in the literature, mostly obtained by imposing certain structural restrictions or context-free conditions on NLC grammars. Extensions of NLC grammars are NCE grammars [16], eNCE grammars[7-9], and edNCE grammars[10]. Restrictions defined for NLC grammars can be easily defined for these extensions of NLC grammars. The eNCE grammars and their restrictions have been particularly successful since they are as simple as NLC grammars yet preserve or improve many nice features of NLC grammars.

For grammars generating hypergraphs, there are context-free hypergraph(CFHG) grammars introduced by Bauderon and Courcelle[1] and Habel and Kreowski[11, 12], which replace a hyperedge by a hypergraph through their preidentified gluing points; an extension of CFHG grammars called the handle-rewriting hypergraph(HH) grammars introduced by Courcelle, Engelfriet and Rozenberg[3], which replace a handle (i.e., a hyperedge together with its incident nodes) by a hypergraph through an extension of the CFHG rewriting mechanism that can also duplicate or delete the hyperedges surrounding the replaced handle; and hypergraph grammars with an eNCE way of rewriting(HRNCE grammars) introduced by Kim and Jeong[18, 19], which replace a handle by a hypergraph through the eNCE rewriting mechanism. Both CFHG and HH grammars generate directed hypergraphs. CFHG grammars generate NP languages[17]. HRNCE grammars can generate all recursively enumerable languages. Separated HRNCE(S-HRNCE) languages are in NP and there is an NP-complete linear HRNCE(Lin-HRNCE) language[19].

We compare (hyper)graph-generating power of HRNCE grammars to that of NLC, CFHG, and HH grammars. It is proved that S-HRNCE grammars have the same graph-generating power as B-eNCE-bntd grammars as well as CFHG grammars. It is also proved that S-HRNCE grammars are more powerful than CFHG grammars for generating hypergraphs, and HRNCE(S-HRNCE) grammars are not comparable to HH(S-HH) grammars.

2. Preliminaries

The reader is assumed to be familiar with formal language theory and complexity theory in the context of, e.g., [13]. This section contains definitions related to hypergraphs and their languages needed in this paper. In the sequel, the *empty set* is denoted by ϕ and, for a finite set A, its *cardinality* is denoted by #A. The empty word is denoted by Λ.

Let Σ, Γ be alphabets. A (hyper)graph over Σ and Γ is a system $H=(V, E, \phi, \Psi)$, where V is a finite set of *nodes*, E is a finite set of *(hyper)edges*, $\phi: V \rightarrow \Sigma$ is a node-labeling function, and $\Psi: E \rightarrow \Gamma$ is an edge-labeling function. Each edge $e \in E$ consists of a nonempty subset of V, denoted by $V(e)$. Note that H is a graph if each edge consists of one or two nodes. For any (hyper)graph H, its four components are denoted by V_H, E_H, ϕ_H, Ψ_H.

A node v and an edge e in H are *incident* to each other if $v \in V_H(e)$. Two edges e and e' in H are *adjacent* (or *a-adjacent*) if there is a node $v \in V_H(e) \cap V_H(e')$ (with $\phi_H(v)=a$). The *degree* of a node v in H is the number of its incident edges; the degree of H is the maximum degree of its nodes. The *rank* of an edge e in H is the number of its incident nodes; the rank of H is the maximum rank of its edges.

Two (hyper)graphs H and K are *isomorphic* if there are bijections $\alpha: V_H \rightarrow V_K$ and $\beta: E_H \rightarrow E_K$ such that for all $v \in V_H$, $\phi_K(\beta(e)) = \Psi_H(e)$ and $V_K(\beta(e)) = \{\alpha(v) \mid v \in V_H(e)\}$. H is a dual of K, denoted by *dual*(K), if $V_H = E_K$, $E_H = V_K$, $\phi_H = \Psi_K$, $\Psi_H = \phi_K$, and for all $e \in E_H$, $V_H(e) = \{e' = E_K \mid e \in V_K(e')\}$. (When a graph, considered as a hypergraph, is converted to its dual, an isolated node of degree zero turns into an edge of rank one, and vice versa.) Note that $dual(dual(H)) = H$ for every hypergraph H.

The set of all graphs (hypergraphs) over Σ and Γ is denoted by $GR_{\Sigma,\Gamma}$ ($HGR_{\Sigma,\Gamma}$). A graph (hypergraph) language is any subset of $GR_{\Sigma,\Gamma}$ ($HGR_{\Sigma,\Gamma}$). A (hyper) graph language is connected if it contains connected (hyper)graphs only and is degree-bounded (rank-bounded) if the degree (rank) of each of its members is at most k, for some fixed $k \geq 0$.

3. HRNCE Grammars

HRNCE grammars generate node-and hyperedge-labeled undirected hypergraphs. An HRNCE grammar replaces a hyperedge (i.e., a set of nodes constituting the edge) by a hypergraph. Its embedding mechanism fully utilizes all available local information as in an eNCE grammar. In a more general setting, any hypergraph may be replaced by another by using the same embedding mechanism. We restrict the left-hand side of a production to a single edge.

A hypergraph grammar with an eNCE way of rewriting (HRNCE grammar) is a system $G = (\Sigma, \Delta, \Gamma, \Omega, P, Z)$, where Σ is an alphabet of node labels, Δ ($\subseteq \Sigma$) is the set of *terminal node labels* (the elements in Σ-Δ are *nonterminal node labels*), Γ is an alphabet of edge labels, Ω ($\subseteq \Gamma$) is the set of *terminal edge labels* (the elements in Γ-Ω are *nonterminal edge labels*), P is a finite set of productions of the form $\pi = (A, X, C)$, where $A \in \Gamma$-Ω (the *left-hand side*), $X \in HGR_{\Sigma,\Gamma}$ (the *right-hand side*), and $C \subseteq V_X \times \Sigma \times \Gamma$ (the *embedding relation*), and $Z (\in HGR_{\Sigma,\Gamma})$ is the *axiom hypergraph*.

We discuss informally how a production $\pi = (A, X, C)$ is applied to a nonterminal edge e with $\Psi(e) = A$ in a hypergraph $H \in HGR_{\Sigma,\Gamma}$. It is similar to the eNCE derivation step. First, remove e and all its incident nodes from H. Second, add an isomorphic copy of X to the resulting hypergraph. Now, for each $v \in V_X$ and each $(v, a, B) \in C$, add v to each edge labeled by B that is a-adjacent to e in H. If any edge adjacent to e in H contains no node after applying π to e, then it is removed immediately. For the formal definition, refer to [19].

A sequence of direct derivation steps is called a *derivation*. The transitive reflexive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$. A (hyper)graph $H \in GR_{\Sigma,\Gamma}(HGR_{\Sigma,\Gamma})$ such that $Z \Rightarrow^* H$ is called a *sentential form* of G. The language generated by G, denoted by L(G), is the set $\{H \in GR_{\Delta,\Omega} \mid Z \Rightarrow^* H\}$.

An HRNCE grammar is a *separated HRNCE (S-HRNCE) grammar* if no two nonterminal edges are adjacent in the axiom and in the right-hand side of any production. (This implies that no two nonterminal edges are adjacent in any of its sentential forms, and vice versa; similar to other separated classes of grammars.)

The *context* of a nonterminal edge e in a hyperedge H, denoted by $context_H(e)$, is the set $\{(a, B) \mid$ an edge e' with $\Psi_H(e') = B$ is a-adjacent to $e\}$. An S-HRNCE grammar $G = (\Sigma, \Delta, \Gamma, \Omega, P, Z)$ is *context-consistent* if there is a function $\eta: \Gamma - \Omega \rightarrow 2^{\Sigma \times \Gamma}$ such that, for each sentential form H in G and for each nonterminal edge e in H, $context_H(e) = \eta(\Psi_H(e'))$. The function η satisfying this property is called the *context-describing function* of G.

An S-HRNCE grammar $G = (\Sigma, \Delta, \Gamma, \Omega, P, Z)$ is neighbourhood preserving if, for all $H, K \in HGR_{\Sigma,\Gamma}$ such that $Z \Rightarrow^* H \Rightarrow_{(e,\pi)} K$, each edge adjacent to e in H is again adjacent to at least one edge from the right-hand side of π in K.

Lemma 3.1. *Every S-HRNCE language can be generated by a context-consistent S-HRNCE grammar*[19].

Lemma 3.2. *Every S-HRNCE language without* Λ *can be generated by a neighbourhood preserving S-HRNCE grammar*[19].

4. Graph-generating Power of HRNCE Grammars

NLC grammars replace a single node by a graph in a derivation step and the embedding of the newly introduced graph into the existing graph is based on node labels only. One of the extension models of NLC grammars is the graph grammars with neighbourhood-controlled embedding (NCE grammars)[16] in which the embedding mechanism makes use of the identity as well as the label of the nodes in the right-hand sides of productions. NCE grammars are further extended to eNCE grammars[7-9] by adding edge labels.

An eNCE grammar is a 6-tuple $G=(\Sigma, \Delta, \Gamma, \Omega, P, Z)$, where Σ is an alphabet of *node labels*, Δ is an alphabet of *terminal node labels* (elements of $\Sigma-\Delta$ are *nonterminal node labels*), Γ is an alphabet of *edge labels*, Ω is an alphabet of *terminal edge labels* ($\Gamma-\Omega$ are *nonterminal edge labels*), P is a finite set of *productions* of the form $\pi=(A, X, C)$, with $A\in\Sigma-\Delta$, and $X\in GR_{\Sigma,\Gamma}$, and $C\subseteq V_X\times\Gamma\times\Gamma\times\Sigma$, and $Z\in GR_{\Sigma,\Gamma}$ is the *axiom graph*.

The derivation of eNCE grammar is as follows. Let $G=(\Sigma, \Delta, \Gamma, \Omega, P, Z)$ be an eNCE grammar, and let $K, K'\in GR_{\Sigma,\Gamma}$. Then, K directly derives K' by the application of a production $(A, X, C)\in P$ of an eNCE grammar G such that the following tasks are performed on K in one step: (1) locate a nonterminal node v labeled by A; (2) remove v and all its incident edges, and (3) add X′ isomorphic to X to the resulting graph such that, if $(v', \alpha, \beta, b)\in C$ with $v'\in V_X$, $b\in\Sigma$, and $\alpha, \beta\in\Gamma$ such that there was an edge with a label α between a node v with a label A and a node u with a label b, then an edge with a label β is created between the node v' and the node u.

A *boundary eNCE (B-eNCE) grammar* G is an eNCE grammar such that no two nonterminals are adjacent in any sentential form. If there is at most one nonterminal in every sentential form, then G is called a *Lin-eNCE grammar. A bounded nonterminal degree B-eNCE (B-eNCE$_{bnd}$) grammar* G′ is a B-eNCE grammar such that, in every sentential form of G′, the degree of every nonterminal node is at most $d\geq 0$. The family of all eNCE (B-eNCE, Lin-eNCE, B-eNCE$_{bnd}$) languages is denoted by *eNCE (B-eNCE, Lin-eNCE, B-eNCE$_{bnd}$)*.

We cannot directly compare the generating power between HRNCE and eNCE grammars since eNCE grammars generate graphs only and HRNCE grammars can generate hypergraphs which are the superset of graphs such that we consider the graph-generating power only between them. It can be easily recognized that there is a Lin-eNCE grammar which generates the set of all complete graphs which cannot be directly generated by HRNCE grammars. Hence, we consider the case of bounded nonterminal degree for B-eNCE grammars.

Definition 4.1. An S_2-HRNCE grammar G is an S-HRNCE grammar such that, in every sentential form of G, the degree of every node is at most 2.

Lemma 4.2. *S_2-HRNCE grammars have the same graph-generating power as S-HRNCE.*

Proof. Let $G=(\Sigma, \Delta, \Gamma, \Omega, P, Z)$ be an S-HRNCE grammar such that $dual(L(G))\in GR_{\Sigma,\Gamma}$. Then, the degree of every node in $H\in L(G)$ is at most 2. Observe that, in every sentential form of G, the degree of a node incident to terminal edges only is at most 2, and that of a node incident to a nonterminal edge is at most $\#\Delta+1$; i.e., there is at most one (a, α)-adjacent edge of the nonterminal edge for each $a\in\Sigma$ and $\alpha\in\Delta$ since an incident node of a terminal edge generated by the nonterminal edge is connected to all or none of (a, α)-adjacent edges such that the degree of the node can be more than 2 if there are more than one (a, α)-adjacent edges. Hence, the basic idea is such that we let each nonterminal edge have at most $\#\Sigma \cdot \#\Delta$ number of incident nodes each of which is incident to at most one terminal edge, and show that it does not affect the language generated.

Formally, we construct an S_2-HRNCE grammar $G' = (\Sigma', \Delta, \Gamma, \Omega, P', Z)$ such that $L(G') = L(G)$. Let $\Sigma' = \Sigma \times \Delta \cup \Sigma$. We may assume that Z has a single nonterminal edge. A production $\pi = (A, X', C') \in P'$ is obtained from each production $\pi = (A, X, C) \in P$; every nonterminal edge in X′ has at most $\#\Sigma \cdot \#\Delta$ number of incident nodes which are pairwise-distinctively labeled by a_α for each $a \in \Sigma$ and $\alpha \in \Delta$, such that the node labeled by a_α is incident to a terminal edge with a label α if any; for each embedding relation $(v, a, \{\alpha_1, \alpha_2, ..., \alpha_k\}) \in C$, where $v \in V_X$, $a \in \Sigma$, and $\alpha_1, \alpha_2, ..., \alpha_k \in \Delta$, $(v_1, a_{\alpha 1}, \alpha_1), (v_2, a_{\alpha 2}, \alpha_2), ..., (v_k, a_{\alpha k}, \alpha_k) \in C'$ when $v_1', v_2', ..., v_k' \in v_{X'}$, $\psi_{X'}(v_i') = (\psi_X(v), \alpha_i)$ for $1 \le i \le k$, and $a_{\alpha 1}, a_{\alpha 2}, ..., a_{\alpha k} \in \Sigma'$.

To see $L(G') = L(G)$, observe the following. when a nonterminal edge in a sentential form is replaced by the right-hand side of the production applied, connections are established by the embedding relation depending on the context of the nonterminal edge, where the context is expressed by the labels of nodes and edges, not by nodes and edges themselves. Hence, there is no difference in the context of the same nonterminal edges in G and G′. Therefore, it will not be difficult to see that $L(G') = L(G)$. □

Lemma 4.3. *Let L be a hypergraph language such that* $dual(L) \in GR_{\Sigma,\Gamma}$. *If* $L \in S_2\text{-}HRNCE$, *then* $dual(L) \in B\text{-}eNCE_{bnd}$.

Proof. Let $G = (\Sigma, \Delta, \Gamma, \Omega, P, Z)$ be an S_2-HRNCE grammar such that $L(G) = L$. We construct a B-eNCE grammar $G' = (\Sigma', \Delta', \Gamma', \Omega', P', Z')$ such that $dual(L(G)) \subseteq L(G')$.

Let $\Sigma' = \Gamma$, $\Delta' = \Delta$, and $\Gamma' = \Omega' = \Sigma$. Assume without loss of generality that Z has a single nonterminal edge such that $Z' = dual(Z)$. Each production $\pi' = (A, X', C') \in P'$ is obtained from a production $\pi = (A, X, C) \in P$ such that $X' = dual(X)$, and $C' = \{(dual(e), a, \psi_X(v), \alpha) \mid v \in e, e \in E_X, a \in \Sigma, \text{ and } \alpha \in \Delta \text{ such that } (v, a, \alpha) \in C\}$.

Consider a derivation $D: H_0(=Z) \Rightarrow H_1 \Rightarrow \cdots \Rightarrow H_n (=H) \in HGR_{\Sigma,\Gamma}$ of G. If we use a function *dual* for each H_i, $0 \le i \le n$, such that $H_i' = dual(H_i)$, then it will not be difficult to see that it is a derivation D′: $H_0'(=Z') \Rightarrow H_1' \Rightarrow \cdots \Rightarrow H_n' \ (=H') \in HGR_{\Sigma,\Gamma}$ of G′ such that $H' = dual(H)$. Hence, $dual(L(G)) \subseteq L(G')$.

Note that the degree of every nonterminal node in each sentential form of G′ is at most $\#\Sigma \cdot \#\Delta$. □

A ***nonterminal neighbour deterministic*** ($B\text{-}eNCE_{nd}$) ***grammar*** $G = (\Sigma, \Delta, \Gamma, \Omega, P, Z)$ is a B-eNCE grammar such that, in each sentential form of G, a nonterminal node v has at most one (α, a)-neighbour for each (α, a), where $\alpha \in \Gamma$ and $a \in \Delta$, and a (α, a)-adjacent node of v is not again (β, a)-adjacent to v, where $\beta \ne \alpha$ [10]. The family of all $B\text{-}eNCE_{nd}$ languages is denoted by $B\text{-}eNCE_{nd}$.

Lemma 4.4. *Let L be a hypergraph language. If dual* $(L) \in B\text{-}eNCE_{nd}$, *then* $L \in S_2\text{-}HRNCE$.

Proof. Let $G = (\Sigma, \Delta, \Gamma, \Omega, P, Z)$ be a $B\text{-}eNCE_{nd}$ grammar such that $L(G) = dual(L)$. We construct an S_2-HRNCE grammar $G' = (\Sigma', \Gamma', \Delta', P', Z')$ such that $L(G) \subseteq dual(L(G'))$.

Let $\Sigma' = \Gamma$, $\Gamma' = \Sigma$, and $\Delta' = \Delta$. Assume without loss of generality that Z has a single nonterminal node such that $Z' = dual(Z)$. From a production $\pi = (A, X, C) \in P$, each production $\pi' = (A, X', C') \in P'$ is obtained such that $X' = dual(X) + \{v' \mid$ for each node v in X with $(v, \alpha, \beta, a) \in C$, a node v' is created and incident to $dual(v)$ in $dual(X)$ such that $\psi_{dual(X)}(v') = \beta$, where $\alpha, \beta \in \Gamma\}$, and $C' = \{(v', \alpha, a) \mid (v, \alpha, \beta, a) \in C$ such that a node v' is created and incident to $dual(v)$ in $dual(X)$ with $\psi_{dual(X)}(v') = \beta$, where $a \in \Delta$ and $\alpha, \beta \in \Gamma\}$.

Consider a derivation $D: H_0(=Z) \Rightarrow H_1 \Rightarrow \cdots \Rightarrow H_n (=H) \in HGR_{\Sigma,\Gamma}$ of G. We use a function *dual* for each H_i, $0 \le i \le n$, such that $H_i' = dual(H_i)$. Note that a nonterminal node v in a derivation of G has at most one (α, a)-neighbour for each (α, a) while a nonterminal edge e in a derivation of G′ also has at most one (α, a)-neighbour for each (α, a). Hence, it will not be difficult to see that a derivation D′: $H_0{}'(=Z') \Rightarrow H_1{}' \Rightarrow \cdots \Rightarrow H_n{}'$ $(=H') \in HGR_{\Sigma';\Delta'}$ is a derivation of G′ such that $H' = dual(H)$. Therefore, $L(G) \subseteq dual(L(G'))$. □

Theorem 4.5. *Let L be a hypergraph language such that $dual(L) \in GR_{\Sigma,\Delta}$. Then, $L \in S\text{-}HRNCE$ if and only if $dual(L) \in B\text{-}eNCE_{bnd}$.*

Proof. From Lemma 4.2, S_2-HRNCE grammars have the same graph-generating power as S-HRNCE grammars. If $L \in S_2\text{-}HRNCE$, then $dual(L) \in B\text{-}eNCE_{bnd}$ from Lemma 4.3. If $dual(L) \in B\text{-}eNCE_{nd}$, then $L \in S_2\text{-}HRNCE$ from Lemma 4.4. Then, $B\text{-}edNCE_{bnd} = B\text{-}edNCE_{nd}$ from Lemma 8 in [10]; this also holds for $B\text{-}eNCE_{bnd}$ and $B\text{-}eNCE_{nd}$. Hence, $L \in S\text{-}HRNCE$ if and only if $dual(L) \in B\text{-}eNCE_{bnd}$. □

From Theorem 4.5, S-HRNCE grammars have the same graph-generating power as B-eNCE$_{bnd}$ grammars while B-eNCE$_{bnd}$ grammars do as CFHG grammars[10] such that S-HRNCE grammars have the same graph-generating power as CFHG grammars.

5. Hypergraph-generating Power of HRNCE Grammars

An *edge-labeled, directed hypergraph* is a system $H = (V, E, \Gamma, \Psi, nod)$, where V is a finite set of *nodes*, E is a finite set of *edges*, Γ is an alphabet of *edge labels*, $\Psi: E \to \Gamma$ is an *edge-labeling function*, and $nod: E \to V$ is the *incidence function* such that $nod(e) = (nod(e, 1), nod(e, 2), \ldots, nod(e, k))$ for some k. A hypergraph H is a hypergraph over Γ if $\Gamma_H \subseteq \Gamma$. The set of all edge-labeled, directed hypergraphs over Γ is denoted by HGR_Γ.

A context-free hypergraph grammar (CFHG grammar)[1, 11, 12] is a 4-tuple $G = (\Gamma, \Delta, P, Z)$, where Γ is an alphabet of *edge labels*, Δ is an alphabet of *terminal edge labels* (elements of $\Gamma - \Delta$ are *nonterminal edge labels*), P is a finite set of *productions* of the form $\pi = (f, Q)$, where $f \in E_Q$, $\Psi_Q(f) \in \Gamma - \Delta$, $Q \in HGR_\Gamma$, and $Z \in HGR_\Gamma$ is the *axiom hypergraph*.

A derivation of a CFHG grammar is as follows. Let $G = (\Gamma, \Delta, P, Z)$ be a CFHG grammar, and let H, $H' \in HGR_\Gamma$. Then, H directly derives H' by the application of a production $\pi = (f, Q) \in P$ of a CFHG grammar G such that the following tasks are performed on H in one step; (1) locate an edge e with a label $\Psi_Q(f) \in \Gamma - \Delta$; (2) remove the edge e only from H such that H-e is the resulting hypergraph; and (3) glue the isomorphic copy of Q-f to the resulting hypergraph H-e such that $nod(e) = nod(f)$.

The family of all CFHG languages is denoted by *CFHG*. Note that CFHG languages are degree-unbounded but rank-bounded [10]; there is a CFHG grammar which generates the set of all graphs of star form [17].

A CFHG grammar G is *identification-free* if, for every production $\pi = (f, Q)$ of G, the nodes of $nod(f)$ are all distinct. Note that, for every CFHG grammar, there is an equivalent one that is identification-free [10]. Assume that every CFHG grammar is identification-free from now on.

Definition 5.1. A directed HRNCE (dHRNCE) grammar $G = (\Sigma, \Gamma, \Delta, P, Z)$ with $\pi = (A, X, C) \in P$ is an HRNCE grammar such that Z and X are directed hypergraphs and $C \subseteq v \times \Sigma \times \Gamma \times d_1 \times d_2$ with $v \in V_X$ and $d_1, d_2 \in \{in, out\}$.

For a hypergraph language L, we denote by $nu(L)$ the node-unlabeled language of L, i.e., the set $\{nu(H) \mid H \in L\}$.

Theorem 5.2. *$CFHG \subseteq nu(dHRNCE)$*

Proof. We will show the inclusion first. The basic idea is such that CFHG grammars embed hypergraphs by using gluing nodes, and such gluing can be resembled by dHRNCE grammars by keeping node labels same for embedding relations.

Let $G = (\Gamma, \Delta, P, Z)$ be an arbitrary CFHG grammar with a production $\pi = (f, Q) \in P$ such that each node for gluing is identified by a function $nod(f, i)$, $1 \le i \le \tau(f)$, in Q. We construct a dHRNCE grammar $G' = (\Sigma, \Gamma', \Delta, P', Z')$ such that $nu(L(G')) = L(G)$.

Let $\Sigma = \{a_1, a_2, ..., a_n\}$, where n is the maximal number of nodes in the right-hand side of a production of G, and $\Gamma' = (\Gamma - \Delta) \times P(\Sigma) \cup \Delta$. We may assume that there is a single nonterminal edge e in Z such that Z' is obtained from Z as follows; $V_{Z'} = V_Z$; $E_{Z'} = E_Z$; each node which is $nod(e, i)$ in Z is labeled by a_i in Z'. $1 \le i \le \tau(e)$; and the nonterminal edge e with a label $A \in \Gamma - \Delta$ in Z is labeled by $A_{a_1 a_2 \dots a_{\tau(e)}} \in \Gamma' - \Delta$ in Z'. Now, starting from a production $\pi = (f, Q) \in P$ with $\Psi_Q(f) = \Psi_Z(e)$, each production $\pi' = (A_{a_{i_1} a_{i_2} \dots a_{i_k}}, X, C) \in P'$ is obtained as follows, where $\Psi_Q(f) = A \in \Gamma - \Delta$ and $a_{i_j} \in \Sigma$ with $1 \le i_j \le n$ and $1 \le j \le k = \tau(f)$; $V_X = V_Q$; $E_X = E_Q - f$; each node of $nod(f)$ in Q is labeled by a_{i_j} in X, where $a_{i_j} \in \{a_{i_1}, a_{i_2}, ..., a_{i_k}\}$, and the rest nodes in X are pairwise-distinctively labeled by $a_l \in \Sigma$, $1 \le l \le n$, such that $a_l \notin \{a_{i_1}, a_{i_2}, ..., a_{i_k}\}$; each nonterminal edge e' with a label $B \in Q - f$ is labeled by $B_{a'_{i_1} a'_{i_2} \dots a'_{i_k}}$ in X, where a'_{i_j} is the label of an incident node of e' and $\Psi_X(e') = \Psi_Q(e')$ if $\Psi_Q(e') \in \Delta$; finally, $(v, \phi_X(v), l, d, d) \in C$ for each node v in X which is corresponding to $nod(f, i)$ in Q and for every $l \in \Gamma'$, where $d \in \{in, out\}$. Note that the direction d exactly follows the one in Q. We repeat the process until no new productions are generated in P'.

Note that there are at most C^n_k number of productions in P' for each $\pi = (f, Q) \in P$, where $k = \tau(f)$ and n is the maximal number of nodes in Q such that $k \le n$; given n colors, there are at most C^n_k number of possibilities to color k nodes pairwise-distinctively. In the HRNCE grammar G', every incident node of a nonterminal edge keeps the same label such that the embedding relation can keep the same connection relation to each adjacent edge of the nonterminal edge in a derivation. In such a way, G' exactly follows each derivation step of G. Hence, clearly, $L(G) \subseteq nu(L(G'))$. Conversely, consider a derivation $D': H_0' (= Z') \Rightarrow_{\pi'1} H_1' \Rightarrow_{\pi'2} \cdots \Rightarrow_{\pi'n} H_n' (= H') \in HGR_{\Sigma,\Delta}$ of G'. Let ρ be an edge relabeling function such that $\rho(A_{a_1 a_2 \dots a_k}) = A$ if $A_{a_1 a_2 \dots a_k} \in \Gamma' - \Delta$ and $\rho(\alpha)$ if $\alpha \in \Delta$. If we use the edge relabeling function ρ and drop the node labels in D', then this yields a derivation $D: H_0 (= Z) \Rightarrow_{\pi 1} H_1 \Rightarrow_{\pi 2} \cdots \Rightarrow_{\pi n} H_n (= H) \in HGR_\Delta$ of G such that $nu(L(G')) \subseteq L(G)$. Therefore, $nu(L(G')) = L(G)$.

For inequality, note that CFHG languages are rank-bounded while HRNCE language are not. Therefore, the theorem follows. □

A CFHG grammar is a *separated CFHG (S-CFHG) grammar* if all right-hand sides of its productions are separated [3]. The class of languages generated by separated CFHG grammars is denoted by *S-CFHG*.

Theorem 5.3. *S-CFHG* $\subseteq$ *S-dHRNCE*

Proof. We will show the inclusion first. The proof is similar to one in Theorem 5.2. Let $G = (\Gamma, \Delta, P, Z)$ be an arbitrary S-CFHG grammar. We construct an S-dHRNCE grammar $G' = (\Sigma, \Gamma, \Delta, P', Z')$ such that $L(G') = L(G)$.

Let $\Sigma = \{a_1, a_2, ..., a_n\}$, where n is the maximal number of nodes in the right-hand side of a production in G. We may assume that there is a single nonterminal edge e in Z such that Z' is obtained from Z as follows; $V_{Z'} = V_Z$; $E_{Z'} = E_Z$; each node which is $nod(e, i)$ in Z is labeled by a_i in Z', $1 \le i \le \tau(e)$; and $\Psi_{Z'}(e) = \Psi_Z(e)$. Each production $\pi' = (A, X, C) \in P'$ is obtained from $\pi = (f, Q) \in P$ with $\Psi_Q(f) = A \in \Gamma - \Delta$; $V_X = V_Q$; $E_X = E_Q - f$; for every nonterminal edge e' in $Q - f$, each node of $nod(e')$ is pairwise-distinctively labeled by $a_i \in \Sigma$ in X and the rest nodes which are incident to terminal edges only are labeled by $a_1 \in \Sigma$; $\Psi_X(e) = \Psi_{Q-f}(e)$ for every edge e in X; finally, $(v, a, l, d, d) \in C$ for every $l \in \Delta$ and for every node v in X which is corresponding to $nod(f, i)$ in Q, where $1 \le i \le \tau(f)$ and $d \in \{in, out\}$.

In the S-HRNCE grammar G', the embedding relation keeps the same connection relation to each adjacent edge of the nonterminal edge in a derivation such that G' exactly follows each derivation step of G. Hence, clearly, $L(G) \subseteq L(G')$. Conversely, consider a derivation $D': H_0' (=Z') \Rightarrow_{\pi'1} H_1' \Rightarrow_{\pi'2} \cdots \Rightarrow_{\pi'n} H_n'$ $(=H') \in HGR_{\Sigma,\Delta}$ of G′. If we drop node labels in D′, then it will not be difficult to see that this yields a derivation $D: H_0 (=Z) \Rightarrow_{\pi 1} H_1 \Rightarrow_{\pi 2} \cdots \Rightarrow_{\pi n} H_n (=H)$ $\in HGR_\Delta$ of G such that $L(G') \subseteq L(G)$. Therefore, $L(G') = L(G)$.

For inequality, note that S-CFHG languages are rank-bounded while S-HRNCE languages are not. Therefore, the theorem follows. □

Definition 5.4. An S-HRNCE grammar $G = (\Sigma, \Gamma, \Delta, P, Z)$ is a neighbourhood-consistent S-HRNCE (S_{NC}-HRNCE) grammar if, for each incident node v' of a nonterminal edge with a label A, there exists exactly one node v in X with $(v, \psi(v'), l) \in C$ for every $l \in \Delta$, where $\pi = (A, X, C) \in P$.

Theorem 5.5. *S_{NC}-dHRNCE = S-CFHG*

Proof. The S-dHRNCE grammar constructed in the proof of Theorem 5.3 is an S_{NC}-dHRNCE grammar. Hence, it follows that *S_{NC}-dHRNCE = S-CFHG*. □

A hypergraph with ports over Γ is a pair (H, *port*) such that $H \in HGR_\Gamma$ and port is a finite subset of $N \times V_H$, where $N = \{0, 1, 2, \ldots\}$. The set of all directed hypergraphs with ports over Γ is denoted by HGR_Γ^p. If $(i, v) \in port$, then the node v is an i-port and i is the port number of v. The set of all i-ports is denoted by $port(i)$ such that $port(i) = \{v \in V_H | (i, v) \in port\}$, and the set of all port numbers of a given node v is denoted by $port^{-1}(v) = \{i \in N^+ | (i, v) \in port\}$, where $N^+ = \{1, 2, \ldots\}$.

A handle-rewriting hypergraph grammar (HH grammar)[3] is a generalization of a CFHG grammar, in which a handle (a structure similar to a hyperedge in a CFHG grammar) together with its incident edges are replaced by a hypergraph in a derivation step. The left-hand side of a production is a nonterminal handle and the right-hand side is a hypergraph with ports in which every node is assigned by a port number.

A *handle-rewriting hypergraph grammar* (*HH grammar*)[3] is a 4-tuple $G = (\Gamma, \Delta, P, Z)$, where Γ is an alphabet of *edge labels*, Δ is an alphabet of *terminal edge labels* (elements of $\Gamma - \Delta$ are *nonterminal edge labels*), P is a finite set of *productions* of the form $\pi = (A, (Q, port))$, where $A \in \Gamma - \Delta$ and $(Q, port) \in HGR_\Gamma^p$, and $Z \in HGR_\Gamma^p$ is the *axiom hypergraph*.

A derivation of an HH grammar is as follows. Let $G = (\Gamma, \Delta, P, Z)$ be an HH grammar, and let $H, H' \in HGR_\Gamma^p$. Then, H directly derives H' by the application of a production $\pi = (A, (Q, port)) \in P$ such that the following tasks are performed on H in one step: (1) locate an edge e labeled by A; (2) remove the edge e and all its incident nodes from H; and (3) add (Q, *port*) to the resulting hypergraph such that the embedding is controlled by the ports of (Q, *port*) as follows; with an edge $e' \neq e$ which is incident to the i-th node of e in H, (i) if $port(i)$ contains exactly one node v, then a copy of e' is created and incident to v; (ii) if $port(i)$ contains more than one node, then the edge e' is duplicated and both edges are incident to each $port(i)$; and (iii) if $port(i)$ contains none, then the edge e' is deleted.

The family of all HH languages is denoted by *HH*. From a derivation of an HH grammar mentioned above, it can be easily recognized that HH languages are degree-unbounded but rank-bounded; there is an HH grammar which generates the set of all complete graphs [3].

Theorem 5.6. *dHRNCE\HH ≠ ∅*

Proof. HH languages are rank-bounded while dHRNCE languages are not. □

Theorem 5.7. *HH\dHRNCE ≠ ∅*

Proof. Note that the connecting edges can be duplicated in HH grammars while the connecting nodes are fixed by the right-hand side of a production in dHRNCE grammars. Hence, there is an HH grammar which generates the set of all complete graphs while there is no such a dHRNCE grammar. □

An HH grammar is a *separated HH (S-HH) grammar* if all right-hand sides of its productions are separated [3]. The class of languages generated by separated HH grammars is denoted by *S-HH*.

Corollary 5.8. $S\text{-}dHRNCE \backslash S\text{-}HH \neq \phi$

Proof. It follows from the proof in Theorem 5.6.

Corollary 5.9. $S\text{-}HH \backslash S\text{-}dHRNCE \neq \phi$

Proof. It follows from the proof in Theorem 5.7.

6. Discussion

Considering graph-generating power, it is proved that S-HRNCE grammars have the same graph-generating power as B-eNCE$_{bnd}$ grammars such that CFHG grammars. Comparing hypergraph-generating power of HRNCE grammars to that of CFHG and HH, it is proved that S-HRNCE grammars are more powerful than CFHG grammars while HRNCE (S-HRNCE) grammars are not comparable to HH (S-HH) grammars.

Since HRNCE grammars are a hypergraph-generating system while NCE grammars a graph-generating system, there is some difficulty to directly compare them. In [10] they treated a hyperedge as a node to compare graph-generating power between boundary graph grammars with CFHG grammars, but it cannot be accepted for comparing graphs generated by graph grammars and HH grammars since replacing several nodes at one time makes the grammar too much powerful. There is not a general mechanism known for comparing (hyper)graph-generating systems. Hence, it can be a good research subject to find a more general mechanism for comparing hypergraphs with graphs.

HRNCE grammars are not comparable to HH grammars. It means that there exists at least one hypergraph language that cannot be generated by each other. Hence, we need to find restricted language classes of HH grammars. However, subclasses of HH grammars are not much known such that restricted versions of HH grammars can be an interesting research subject.

References

[1] M. Bauderon and B. Courcelle, "Graph expressions and graph rewritings," *Math. Systems Theory* **20** (1987), 83-127.

[2] V. Claus, H. Ehrig and G. Rozenberg (Eds.), "Proc. 1st Intern. Workshop on Graph Grammars and Their Application to Computer Science and Biology," *Lecture Notes in Computer Science* **73** (Springer-Verlag, Berlin, 1979).

[3] B. Courcelle, J. Engelfriet and G. Rozenberg, "Handle-rewriting hypergraph grammars," *J. Comput. System Sci.* **46** (1993), 218-270.

[4] H. Ehrig, H.-J. Kreowski, G. Rozenberg and A. Rosenfeld (Eds.), "Proc. 4th Intern, Workshop on Graph Grammars and Their Application to Computer Science," *Lecture Notes in Computer Science* **532** (Springer-Verlag, Berlin, 1991).

[5] H. Ehrig, M. Nagl and G. Rozenberg (Eds.), "Proc. 2nd Intern. Workshop on Graph Grammars and Their Application to Computer Science," *Lecture Notes in Computer Science* **153**(Springer-Verlag, Berlin, 1983).

[6] H. Ehrig, M. Nagl, G. Rozenberg and A. Rosenfeld(Eds.), "Proc. 3rd Intern. Workshop on Graph Grammars and Their Application to Computer Science," *Lecture Notes in Computer Science* **291** (Springer-Verlag, Berlin, 1987).

[7] J. Engelfriet and G. Leih, "Linear graph grammars:power and complexity." *Inform. and Comput.* **81** (1989), 88-121.

[8] J. Engelfriet and G. Leih, "Complexity of boundary graph languages," *Theoretical Informatics and Applications* **24** (1990), 267-274.

[9] J. Engelfriet, G. Leih and E. Welzl, "Boundary graph grammars with dynamic edge relabeling," *J. Comput. System Sci.* **40** (1990), 307-345.

[10] J. Engelfriet and G. Rozenberg, "A comparison

of boundary graph grammars and context-free hypergraph grammars," *Inform. and Comput.* **84** (1990), 163-206.

[11] A. Habel and H.-J. Kreowski, "Some structural aspects of hypergraph languages generated by hyperedge replacement," Proc. STACS 87, *Lecture Notes in Computer Science* **247** (1987), 207-219.

[12] A. Habel, H.-J. Kreowski and W. Vogler, "Metatheorems for decision problems on hyperedge replacement graph languages," *Acta Inform.* **26** (1989), 657-677.

[13] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Reading, Mass., 1979).

[14] D. Janssens and G. Rozenberg, "On the structure of node-label-controlled graph languages," *Inform. Sci.* **20** (1980), 191-216.

[15] D. Janssens and G. Rozenberg, "Restrictions, extensions, and variations of NLC grammars," *Inform. Sci.* **20** (1980), 217-244.

[16] D. Janssens and G. Rozenberg, "Graph grammars with neighbourhood-controlled embedding," *Theoretical Computer Science* **21** (1982), 55-74.

[17] C. Lautemann, "The complexity of graph languages generated by hyperedge replacement," *Acta Inform.* **27** (1990), 399-421.

[18] C. Kim and T.E. Jeong, "HRNCE Grammars-A Hypergraph Generating System with an eNCE Way of Rewriting," *Lecture Notes in Computer Science* **1073** (1996), 383-396.

[19] C. Kim and T.E. Jeong, "HRNCE Grammars-A Hypergraph Generating System with an eNCE Way of Rewriting," To appear in *Theoretical Computer Science*.

[20] G. Rozenberg and E. Welzl, "Boundary NLC graph grammars-Basic definitions, normal forms, and complexity," *Inform. and Control* **69** (1986), 136-167.

정 태 의

1979년 고려대학교 전자공학과(학사)
1982년 오하이오주립대 전기공학과(석사)
1989년 오클라호마대학 전산학과(석사)
1994년 오클라호마대학 전산학과(박사)
1983년 10월~1986년 6월 금성반도체 연구소 컴퓨터부문 선임연구원
1986년 7월~1987년 8월 United Microtek, Inc. (San Jose, CA) Engineering Manager
1995년 3월~현재 서경대학교 컴퓨터과학과 전임강사
관심분야: 형식언어, 컴파일러, 알고리즘, 패턴인식, 객체지향데이타베이스

박 동 선

1979년 고려대학교 전자공학과(학사)
1984년 미주리주립대 전기 및 컴퓨터공학과(석사)
1990년 미주리주립대 전기 및 컴퓨터공학과(박사)
1991년 3월~현재 전북대학교 정보통신공학과 조교수
관심분야: 패턴인식, 인공지능, 멀티미디어 통신, 알고리즘