

# 文脈依存 NCE グラフ文法とその構文解析可能性

安達 由洋<sup>†</sup> 小林 卓<sup>††</sup>

グラフ文法は、ビジュアルオブジェクトに対する生成、解析、変換などの操作を数学的に厳密にモデル化するメカニズムを提供する。本論文では、まず、文脈依存 NCE グラフ文法と呼ぶ非常に柔軟なグラフ書き換えメカニズムを持つグラフ文法を定義する。この文法はよく知られた(文脈自由)NCE グラフ文法の文脈依存形への拡張となっており、広いクラスのビジュアル言語のグラフ構文規則を柔軟に定式化することができる。そして、文脈依存 NCE グラフ文法に基づいたグラフ構文規則の定式化についてシグナルフローグラフを例に用いて説明する。次に、文脈依存 NCE グラフ文法の構文解析可能性について議論する。構文解析可能性とは、文法により生成されるグラフに対して、文法のプロダクションの逆適用によりそのグラフを生成する導出を見つけることができるという性質である。そして、文脈依存 NCE グラフ文法が構文解析可能であるための十分条件を与える。NCE タイプの文脈依存グラフ文法の定式化とその構文解析可能性に関する研究は本論文が初めてのものである。

## The Context-Sensitive NCE Graph Grammar and Its Parsability

YOSHIHIRO ADACHI<sup>†</sup> and SUGURU KOBAYASHI<sup>††</sup>

Graph grammars provide a mechanism in which generation and transformation of visual objects can be modeled in a mathematically precise way. In this paper we describe a context-sensitive NCE graph grammar. This grammar is an extension of the well-known (context-free) NCE graph grammar, and is flexible enough for use in formalizing the graph-syntax rules of a wide range of visual languages. We then discuss the parsability of the context-sensitive NCE graph grammar, that is, whether or not a derivation can be found for a given graph by the reverse application of the grammar's productions. Our work described in this paper is the first attempt to formalize an NCE type of context-sensitive graph grammar, and to investigate its parsability.

### 1. はじめに

グラフ文法は文字列に対する形式言語理論のグラフへの自然な一般化であり、ビジュアルオブジェクトに対する生成、解析、編集などの操作を数学的に厳密にモデル化するメカニズムを提供する。グラフ文法あるいはグラフ変換に関しては多くの理論的研究がなされており<sup>1)~4)</sup>、ビジュアル言語をはじめとするさまざまな分野へのグラフ文法の実用的応用研究も数多く報告されている<sup>5)~8)</sup>。グラフ文法は大きく文脈自由形と文脈依存形に分類できるが、なかでも文脈依存グラフ文法は表現力が高く、より柔軟にビジュアル言語のグラフ構文規則を定式化することができる。Marriottらは、ビジュアル言語の表現力に関する研究の結果をふまえて、ビジュアル言語に対する文脈依存記述の研究

の重要性を指摘している<sup>3)</sup>。

Rekersらは“layered graph grammar”と呼ぶ文脈依存グラフ文法を定義し、この文法により文脈自由文法では記述することが難しいプロセス流れ図のグラフ構文規則が簡潔に定式化できることを報告した<sup>9),10)</sup>。彼らは同じ文献で“layered graph grammar”の構文解析法についても報告している。Zhangらは“layered graph grammar”をもとにして“reserved graph grammar”を開発した<sup>11)</sup>。この“reserved graph grammar”も文脈依存グラフ文法である。文献11)では、“reserved graph grammar”が多項式時間の計算的複雑さで構文解析できる条件について議論している。一方、安達らはグラフ書き換え過程で、隣接ノードと埋め込んだノードの間だけに結合命令に従って新しいエッジが張られるNCEタイプの文脈依存グラフ文法を考案し、この文法を用いてブロック線図文法<sup>12)</sup>を定式化した。文脈依存グラフ文法に関する研究は、これらのものを除いてほとんど報告されていない。

本論文では、文脈依存 NCE グラフ文法 ( context-

<sup>†</sup> 東洋大学工学部

Faculty of Engineering, Toyo University

<sup>††</sup> NEC ソフト

NEC Soft. Ltd.

sensitive graph grammar with neighborhood controlled embedding) と呼ぶ非常に柔軟なグラフ書き換えメカニズムを持つグラフ文法を定義する．この文法はよく知られた(文脈自由)NCEグラフ文法<sup>1)</sup>の文脈依存形への拡張となっており, 広いクラスのビジュアル言語のグラフ構文規則を柔軟に定式化することができる．そして, この文脈依存 NCE グラフ文法によるビジュアル言語のグラフ構文規則の定式化をシグナルフローグラフを例に用いて説明する．

次に, 文脈依存 NCE グラフ文法の構文解析可能性について議論する．構文解析可能性とは, 文法により生成されるグラフに対して文法のプロダクションの逆適用によりそのグラフを生成する導出を見つけることができるという性質である．まず, ボトムアップ構文解析アルゴリズムの停止性を保証するレイヤードの性質について述べる．そして, 文脈依存 NCE グラフ文法が構文解析可能であるための十分条件を与える．本論文で述べるボトムアップ構文解析アルゴリズムは, レイヤードの性質を満たす文脈依存 NCE グラフ文法に基づいて任意のグラフを構文解析したとき, 文法に対して正しいグラフには ‘valid’ とそのグラフを導出するプロダクションの系列を出力して停止する．また, 文法に対して正しくないグラフには ‘invalid’ を出力して停止する．

NCE タイプの文脈依存グラフ文法の定式化と構文解析可能性に関する研究は本論文が初めてのものである．

なお, 本論文の内容の一部は, 1999 IEEE Symposium on Visual Languages (VL'99)<sup>3)</sup>, および International Conference on Software: Theory and Practice of the 16th IFIP World Computer Congress (ICS2000)<sup>4)</sup> で報告している．

## 2. 文脈依存 NCE グラフ文法

文脈依存 NCE グラフ文法と, この文法に基づくグラフ書き換えを定義する．文脈依存 NCE グラフ文法に基づくグラフ書き換えでは, 埋め込み過程で埋め込んだノードと隣接ノードの間のみ接続命令に従って新しいエッジが作られる．

なお, 本論文で用いる用語や記法は, その多くを Engelfriet らの論文<sup>1)</sup>を参考にして定義している．

### 2.1 文脈依存 NCE グラフ文法の定義

$\Sigma$  をノードラベルのアルファベット,  $\Gamma$  をエッジラベルのアルファベットとする． $\Sigma$  と  $\Gamma$  上のグラフとは, 組  $H = (V, E, \lambda)$  である．ここで,  $V$  はノードの空でない有限集合,  $E \subseteq \{(v, \gamma, w) \mid v, w \in V, v \neq w, \gamma \in \Gamma\}$

はエッジの有限集合, そして  $\lambda: V \rightarrow \Sigma$  はノードラベル関数である．エッジ  $e = (v, \gamma, w)$  は, ノード  $v$  からノード  $w$  に向かって方向付けられている．したがって,  $e$  は  $v$  の出力エッジであり, かつ  $w$  の入力エッジである．グラフ  $H$  の構成要素をそれぞれ  $V_H, E_H, \lambda_H$  と書くことにする．あるノードの入力エッジの数をそのノードの入力次数, 出力エッジの数を出力次数という．グラフはすべての  $(v, \gamma, w) \in E$  に対して  $(w, \gamma, v) \in E$  であるとき無向グラフである．ノードラベルを持たないグラフは  $\Sigma = \{\#\}$  とし, またエッジラベルを持たないグラフは  $\Gamma = \{*\}$  としてモデル化する．

2つのグラフ  $H$  と  $K$  に対して全単射  $\theta: V_H \rightarrow V_K$  が存在して,  $E_K = \{(\theta(v), \gamma, \theta(w)) \mid (v, \gamma, w) \in E_H\}$ , かつ, すべての  $v \in V_H$  に対して  $\lambda_K(\theta(v)) = \lambda_H(v)$  が成り立つとき,  $H$  と  $K$  は同形であるといい,  $\theta$  を  $H$  から  $K$  への同形写像という．また,  $V_H \cap V_K = \phi$  のとき,  $H$  と  $K$  は互いに素であるという． $\Sigma$  と  $\Gamma$  上のすべてのグラフの集合を  $GR_{\Sigma, \Gamma}$  で表し,  $GR_{\Sigma, \Gamma}$  の部分集合をグラフ言語という．

本論文では, 有向グラフを生成し, かつ生成されたグラフがノードラベルとともにエッジラベルも持つ文脈依存 NCE グラフ文法を定義する．無向グラフを生成する文法や, エッジラベルを持たないグラフを生成する文法などを文脈依存 NCE グラフ文法の特別な場合として議論できる．

定義 1 (文脈依存 NCE グラフ文法)

文脈依存 NCE グラフ文法とは 6 項組  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  である．ただし,

- (1)  $\Sigma_N$  は非終端ノードラベルのアルファベットである．
- (2)  $\Sigma_T$  は終端ノードラベルのアルファベットである． $\Sigma_N$  と  $\Sigma_T$  はそれぞれ空でない有限集合であり, 互いに素である． $\Sigma_{N \cup T} = \Sigma_N \cup \Sigma_T$  はノードラベルのアルファベットである．
- (3)  $\Gamma_N$  は非終端エッジラベルのアルファベットである．
- (4)  $\Gamma_T$  は終端エッジラベルのアルファベットである． $\Gamma_N$  と  $\Gamma_T$  はそれぞれ空でない有限集合であり, 互いに素である． $\Gamma_{N \cup T} = \Gamma_N \cup \Gamma_T$  はエッジラベルのアルファベットである．
- (5)  $S \in \Sigma_N$  はスタートラベルである．スタートグラフ  $H_S$  はスタートラベル  $S$  を持つエッジのないノード 1 点からなるグラフである．
- (6)  $P$  はプロダクションの空でない有限集合である． $P$  の各要素  $p$  は 3 項組  $(A, B, C)$  で表さ

れる．ただし，

- (a)  $A \in GR_{\Sigma_N, \Gamma_{NUT}}$  ,  
 $B \in GR_{\Sigma_{NUT}, \Gamma_{NUT}}$  .
- (b)  $C \subseteq \Sigma_{NUT} \times V_A \times \Gamma_{NUT} \times \Gamma_{NUT} \times V_B$   
 $\times \{\text{in, out}\}$  .

プロダクション  $p = (A, B, C)$  において， $A$  を  $p$  の左辺といい， $lhs(p)$  と書く．また， $B$  を  $p$  の右辺といい， $rhs(p)$  と書く． $C$  を結合関係といい， $C$  の各要素  $(\sigma, v_A, \beta, \gamma, v_B, d)$  を  $p$  の結合命令という．■

本論文では，プロダクション  $p = (A, B, C)$  を具体的に与えるときには， $'p: A ::= B, C'$  の形式で記すことにする．

## 2.2 グラフの導出

2つのプロダクション  $p_1 = (A_1, B_1, C_1)$  と  $p_2 = (A_2, B_2, C_2)$  に対して， $A_1$  から  $A_2$  への(左辺の)同形写像  $\theta_l$  と  $B_1$  から  $B_2$  への(右辺の)同形写像  $\theta_r$  が存在して， $C_2 = \{(\sigma, \theta_l(v_A), \beta, \gamma, \theta_r(v_B), d) \mid (\sigma, v_A, \beta, \gamma, v_B, d) \in C_1\}$  が成り立つとき， $p_1$  と  $p_2$  は同形であるという．また，写像の対  $(\theta_l, \theta_r)$  を  $p_1$  から  $p_2$  への同形写像という．プロダクション集合  $P$  には，同形なプロダクションを含まないと仮定する． $p \in P$  に同形なプロダクションの集合を  $copy(p)$  で表す． $copy(p)$  の要素を  $p$  のプロダクションコピーという． $copy(P) = \bigcup_{p \in P} copy(p)$  とする．

文脈依存 NCE グラフ文法に基づくグラフ書き換えは以下の手順で行われる：書き換える対象となるグラフ(ホストグラフ)を  $H$  とし， $H$  の(ノード)誘導部分グラフを  $A'$  とする． $p' = (A', B', C') \in copy(P)$  とし， $B'$  と  $H$  が互いに素であるとする．(1)  $H$  から  $A'$  (マザーグラフ)と  $A'$  のノードに接続している  $H$  のエッジをすべて取り除き得られたグラフ(レストグラフ)を  $H^-$  とする．(2)  $H^-$  に  $B'$  (ドーターグラフ)を置く．(3)  $C'$  の結合命令に従って， $B'$  のノードと  $H^-$  のノード間のエッジを新しく作る．これを  $B'$  の  $H^-$  への埋め込みという．

埋め込み過程において，結合命令  $(\sigma, x, \beta, \gamma, y, \text{in}) \in C'$  は次の意味を持つ．ホストグラフ  $H$  においてラベル  $\sigma$  を持つノード  $v \in V_H - V_{A'}$  から  $x \in V_{A'}$  へ向かうエッジ  $(v, \beta, x)$  があれば， $v$  から  $y \in V_{B'}$  へ向かうエッジ  $(v, \gamma, y)$  を新しく作る．このとき，マザーグラフのノードとの間にエッジを持つレストグラフのノード，たとえば  $v$  を隣接ノードという．結合命令  $(\sigma, x, \beta, \gamma, y, \text{out}) \in C'$  の場合もエッジの向きが異なるだけであとは同様である．埋め込み過程でドーターグラフとレストグラフの間に新しく作られるエッジはブリッジと呼ばれることがある．なお，このグラフ書

き換えにより，エッジの両端もしくは片端のノードがグラフに含まれない懸垂エッジ(dangling edge)が出現することはない．

文脈依存 NCE グラフ文法に基づくグラフ書き換え過程を形式的に定義すると以下ようになる．

### 定義 2 (導出)

$G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  を文脈依存 NCE グラフ文法とする． $H$  と  $H'$  を  $\Sigma_{NUT}$  と  $\Gamma_{NUT}$  上のグラフ， $A'$  を  $H$  の誘導部分グラフとする．そして  $p' = (A', B', C') \in copy(P)$  であり， $H$  と  $B'$  は互いに素であるとする．このとき，プロダクションコピー  $p'$  のグラフ  $H$  への左適用とは，次の要素  $V_{H'}$ ， $E_{H'}$ ， $\lambda_{H'}$  を持つグラフ  $H'$  を構成することである．

$$\begin{aligned} V_{H'} &= (V_H - V_{A'}) \cup V_{B'}, \\ E_{H'} &= \{(v, \gamma, w) \in E_H \mid v, w \in V_H - V_{A'}\} \cup E_{B'} \cup \{(v, \gamma, y) \mid v \in V_H - V_{A'}, (v, \beta, x) \in E_H, (\sigma, x, \beta, \gamma, y, \text{in}) \in C', \lambda_H(v) = \sigma\} \cup \{(y, \gamma, v) \mid v \in V_H - V_{A'}, (x, \beta, v) \in E_H, (\sigma, x, \beta, \gamma, y, \text{out}) \in C', \lambda_H(v) = \sigma\}, \\ \lambda_{H'}(x) &= \begin{cases} \lambda_H(x), & x \in V_H - V_{A'} \\ \lambda_{B'}(x), & x \in V_{B'}. \end{cases} \end{aligned}$$

グラフ  $H$  にプロダクションコピー  $p'$  を左適用して  $H'$  が得られるとき  $H \Rightarrow_{p'} H'$  と書き， $H$  から  $H'$  への導出ステップという．また，導出ステップの系列  $H_0 \Rightarrow_{p'_1} H_1 \Rightarrow_{p'_2} \dots \Rightarrow_{p'_n} H_n$  を導出といい，系列の長さ  $n$  を導出の長さという．■

例を用いて導出ステップを説明する． $\Sigma_N = \{[X1], [X2], [Y1]\}$ ， $\Sigma_T = \{\textcircled{a}, \textcircled{b}, \textcircled{c}, \textcircled{d}, \textcircled{e}, \textcircled{f}\}$ ， $\Gamma_N = \{\alpha, \beta, \gamma\}$ ，そして  $\Gamma_T = \{\delta, \varepsilon\}$  とする．図 1(a) のホストグラフ  $H$  に，図 1(b) に示すプロダクション  $p$  のプロダクションコピー  $p'$  を左適用する導出過程を図 2 に示す．図 2(a) はレストグラフ  $H^-$  であり，図 2(b) は，レストグラフにドーターグラフを追加したグラフを示している．結合命令に従って追加したドーターグラフとレストグラフのノード間にエッジを生成した図 2(c) が導出ステップにより得られた結果のグラフである．なお，プロダクションの定義(図 1(b))で各ノードの右下の添字  $x1, x2, y1, y2$ ，および  $y3$  はノードを区別するための識別子である．

文脈依存 NCE グラフ文法のプロダクションを，結合命令も含めて簡潔に分かりやすく図的表現する方法を説明する．図 3 は図 1(b) のプロダクションを図的表現したものである．プロダクションの左辺と右辺は，

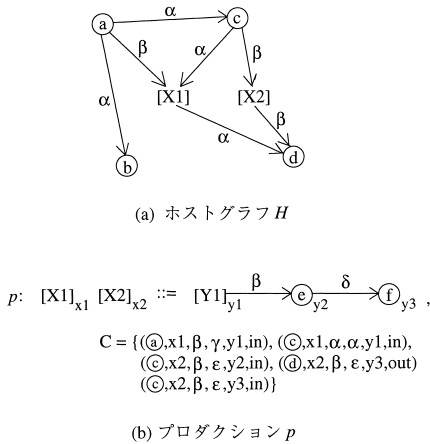


図 1 ホストグラフとプロダクション  
Fig.1 A host graph and production.

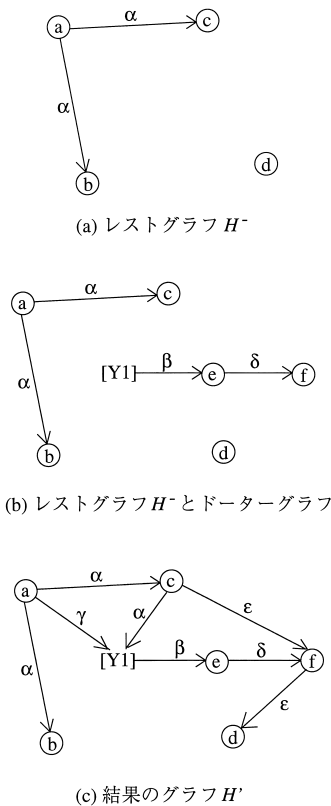


図 2 左適用によるグラフ書き換え過程 (導出ステップ)  
Fig.2 Graph rewriting by left application of a production copy (derivation step).

右辺のグラフを大きな四角 (箱) の中に書くことを除いてもとのプロダクションの定義と同じ表現である。箱の外側は右辺と同形なグラフ (ドーターグラフ) を埋め込む環境を表している。そして、箱の外側に書かれた対、たとえば  $(\textcircled{a}, x1)$  は 'x1' の隣接ノードで、か

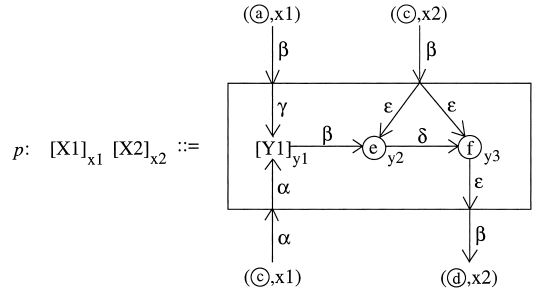


図 3 文脈依存 NCE グラフ文法のプロダクションの図形表現  
Fig.3 Graphical specification of a production of the context-sensitive NCE graph grammar.

つ 'a' でラベル付けされたノードを表している。結合命令は箱の内側のノードと外側の隣接ノードを有向線分でつなぐことにより表す。この有向線分はエッジの向きを表す (同じ方向の) 2 つの矢印と 2 つのラベルを持っている。四角の外側のラベルは書き換えをする前の '古い' エッジラベルであり、四角の内側のラベルは左適用してグラフを書き換えたあとの '新しい' エッジラベルを表す。すなわち、箱の外側に書かれた対  $(\textcircled{a}, x1)$  とそこから箱の内側のノード 'y1' へ向けて出ているラベルの付いた有向線分により結合命令  $(\textcircled{a}, x1, \beta, \gamma, y1, \text{in})$  を表している。なお、このプロダクション全体の図的表現法は (文脈自由) NCE グラフ文法に対して文献 1) で提案された方法を文脈依存プロダクションを表現できるように拡張したものである。

文脈依存 NCE グラフ文法  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  の導出  $H_0 \Rightarrow_{p'_1} H_1 \Rightarrow_{p'_2} \dots \Rightarrow_{p'_n} H_n, n \geq 0,$  は  $H_0$  と  $\text{rhs}(p'_i) = B'_i, 1 \leq i \leq n,$  が互いに素であるとき生成的であるという。今後、生成的な導出のみを扱うことにする。生成的な導出  $H = H_0 \Rightarrow_{p'_1} H_1 \Rightarrow_{p'_2} \dots \Rightarrow_{p'_n} H_n = H'$  が存在するとき、 $H \Rightarrow^* H'$  と書く。G の文形式とはスタートグラフ  $H_S$  から (生成的に) 導出されるグラフのことであり、文形式の集合を  $S(G)$  と書く。  $\mathcal{L}(G) = \{H \in GR_{\Sigma_T, \Gamma_T} \mid H_S \Rightarrow^* H\}$  を G によって生成されるグラフ言語という。

文脈依存 NCE グラフ文法は、"layered graph grammar" あるいは "reserved graph grammar" では記述することが困難であるグラフ言語、たとえば、完全グラフや完全 2 分グラフなどの構文規則を容易に記述することができる。ノードラベルおよびエッジラベルを持たない (無向) 完全 2 分グラフからなる言語を生成する文脈依存 NCE グラフ文法  $G_B = (\Sigma_{NB}, \Sigma_{TB}, \Gamma_{NB}, \Gamma_{TB}, [B], P_B)$  を図 4 (a) に示す。結合命令の中の '# ' は、任意のノードラベルとマッチす

る記号である．このグラフ文法による導出例を図 4 (b) に示す．図 4 (a) および図 4 (b) では，無向エッジを単にノード間の線分で表示している．

2.3 プロダクションの右適用

文脈依存 NCE グラフ文法に基づいてグラフを構文解析するには，導出とは逆向きにプロダクションコピーを適用してグラフを書き換える操作が必要となる．

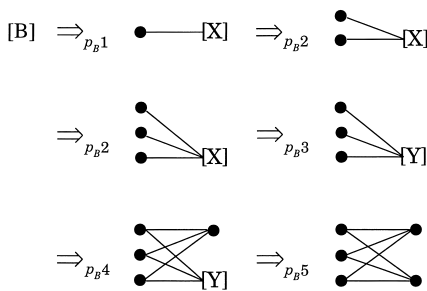
定義 3 (還元)

$G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  を文脈依存 NCE グラフ文法とする． $H$  と  $H'$  を  $\Sigma_{N \cup T}$  と  $\Gamma_{N \cup T}$  上のグラフ， $B'$  を  $H$  の誘導部分グラフとする．そして  $p' = (A', B', C')$  は  $p \in P$  に対するプロダクションコピーで， $H$  と  $A'$  は互いに素であるとする．このとき，プロダクションコピー  $p'$  のグラフ  $H$  への右適用とは，次の要素  $V_{H'}$  ,  $E_{H'}$  ,  $\lambda_{H'}$  を持つグラフ  $H'$  を構成することである．

$$\Sigma_{N_B} = \{[B], [X], [Y]\}, \Sigma_{T_B} = \{\#\}, \Gamma_{N_B} = \{*\}, \Gamma_{T_B} = \{*\},$$

$$P_B = \{ (p_{B1}: [B]_{x1} ::= \bullet_{y1} \text{---} [X]_{y2}, C = \phi), (p_{B2}: [X]_{x1} ::= \bullet_{y1} \text{---} [X]_{y2}, C = \{(\#,x1,*,*,y2,in), (\#,x1,*,*,y2,out)\}), (p_{B3}: [X]_{x1} ::= [Y]_{y1}, C = \{(\#,x1,*,*,y1,in), (\#,x1,*,*,y1,out)\}), (p_{B4}: [Y]_{x1} ::= \bullet_{y1} \text{---} [Y]_{y2}, C = \{(\#,x1,*,*,y1,in), (\#,x1,*,*,y1,out), (\#,x1,*,*,y2,in), (\#,x1,*,*,y2,out)\}), (p_{B5}: [Y]_{x1} ::= \bullet_{y1}, C = \{(\#,x1,*,*,y1,in), (\#,x1,*,*,y1,out)\}) \}$$

(a) 完全 2 部グラフを生成する文脈依存 NCE グラフ文法



(b) 完全 2 部グラフの導出

図 4 完全 2 分グラフを生成する文脈依存 NCE グラフ文法  
Fig. 4 A context-sensitive NCE graph grammar for generating complete bipartite graphs.

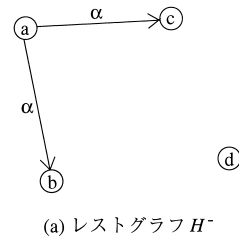
$$V_{H'} = (V_H - V_{B'}) \cup V_{A'},$$

$$E_{H'} = \{(v, \gamma, w) \in E_H \mid v, w \in V_H - V_{B'}\} \cup E_{A'} \cup \{(v, \beta, x) \mid v \in V_H - V_{B'}, (v, \gamma, y) \in E_H, (\sigma, x, \beta, \gamma, y, in) \in C', \lambda_H(v) = \sigma\} \cup \{(x, \beta, v) \mid v \in V_H - V_{B'}, (y, \gamma, v) \in E_H, (\sigma, x, \beta, \gamma, y, out) \in C', \lambda_H(v) = \sigma\},$$

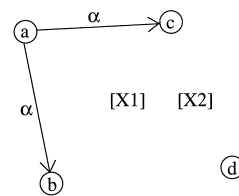
$$\lambda_{H'}(x) = \begin{cases} \lambda_H(x), & x \in V_H - V_{B'} \\ \lambda_{A'}(x), & x \in V_{A'}. \end{cases}$$

プロダクションコピー  $p'$  の右適用によって，グラフ  $H$  が  $H'$  に変換されるとき， $H \Rightarrow_{p'} H'$  と書く． $H \Rightarrow_{p'} H'$  を還元ステップという．また，還元ステップの系列  $H_0 \Rightarrow_{p_{1-1}} H_1 \Rightarrow_{p_{2-1}} H_2 \cdots \Rightarrow_{p_{n-1}} H_n$  を還元といい，系列の長さ  $n$  を還元の長さという．特に， $H_0 \Rightarrow_{p_{1-1}} H_1 \Rightarrow_{p_{2-1}} H_2 \cdots \Rightarrow_{p_{n-1}} H_n = H_S$  をスタートグラフ  $H_S$  への還元という．

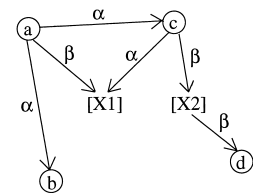
あるプロダクションコピー  $p'$  に対して， $H \Rightarrow_{p'} H'$



(a) レストグラフ  $H$



(b) レストグラフ  $H$  とドーターグラフ



(c) 結果のグラフ  $H'$

図 5 右適用によるグラフ書き換え過程 (還元ステップ)  
Fig. 5 Graph rewriting by right application of a production copy (reduction step).

かつ  $H' \Rightarrow_{p',-1} H''$  が成立しても  $H = H''$  が成立するとは限らない．たとえば，図 2(c) のグラフを，このグラフを導出するために用いたプロダクションコピー  $p'$  を右適用して変換するとその還元ステップは図 5 のようになり，図 5 (c) のグラフが得られる．このとき，図 5 (c) のグラフと図 1 (a) のグラフは異なっている．

### 3. シグナルフローグラフの生成文法

シグナルフローグラフは，システム内の信号間の因果関係と演算関係を明示的に表現できる図的モデルで電気回路，制御工学あるいはフィルタ設計など広く工学分野で利用されている．シグナルフローグラフを用いたシステムの構造解析や設計をコンピュータ上で高度に支援するには，シグナルフローグラフを生成，解析および変換するための形式的モデルの開発が必要である．そこで，1 入力 1 出力シグナルフローグラフのグラフ構文規則を定式化した文脈依存 NCE グラフ文法を定義する．

1 入力 1 出力シグナルフローグラフは以下の性質を持つ弱連結（無向グラフとして見たとき連結）な有向グラフ  $D$  として定義される．

- (1)  $D$  は  $d_D^-(v_0) = 0 \wedge d_D^+(v_0) = 1$  なるノード  $v_0$  と  $d_D^-(v_1) = 1 \wedge d_D^+(v_1) = 0$  なるノード  $v_1$  をそれぞれ 1 個ずつ持ち， $v_0$  は‘入力’， $v_1$  は‘出力’とラベル付けされている．
- (2) 上記 (1) 以外の  $D$  の任意のノード  $v$  に対して
  - (i)  $1 \leq d_D^-(v) < \infty \wedge 1 \leq d_D^+(v) < \infty$   
または
  - (ii)  $d_D^-(v) = 1 \wedge d_D^+(v) = 1$
 であり，(i) のタイプのノードには‘節’，(ii) には‘枝’とラベル付けされている．
- (3) ‘入力’（とラベル付けされた）ノードと‘出力’ノード，‘入力’ノードと‘節’ノード，‘出力’ノードと‘節’ノード，あるいは‘節’ノードと‘節’ノードの間が直接エッジでつながれることはない．また，‘枝’ノードと‘枝’ノードも直接エッジでつながれることはない．

ここで  $d_D^-(v)$ ， $d_D^+(v)$  はそれぞれ有向グラフ  $D$  におけるノード  $v$  の入力次数と出力次数を表すものとする．

定義 4（シグナルフローグラフの生成文法）

文脈依存 NCE グラフ文法  $G_{Signal} = (\Sigma_{N_{Signal}}, \Sigma_{T_{Signal}}, \{*\}, \{*\}, [SFG], P_{Signal})$  はシグナルフローグラフの集合を生成する．ここで，非終端ノードラベルのアルファベット  $\Sigma_{N_{Signal}}$  および終端ノードラベルのアルファベット  $\Sigma_{T_{Signal}}$  は図 6 に示すものである．シグナルフローグラフはエッジラベルのないグラフとし

$\Sigma_{N_{Signal}}$	$\Sigma_{T_{Signal}}$
[SFG] : スタートラベル	⊙ : 入力
[E] : 非終端要素	⊛ : 出力
[N] : 非終端節	○ : 節
	→ : 枝

図 6  $G_{Signal}$  のノードラベルのアルファベット

Fig. 6 The node alphabets of  $G_{Signal}$ .

$$p_{S1}: [SFG]_{x1} ::= \odot_{y1} \rightarrow [E]_{y2} \rightarrow \odot_{y3}, \quad C = \phi$$

$$p_{S2}: [E]_{x1} ::= [E]_{y1} \rightarrow [N]_{y2} \rightarrow [E]_{y3}, \\ C = \{(\#,x1,*,*,y1,in),(\#,x1,*,*,y3,out)\}$$

$$p_{S3}: [N]_{x1} [N]_{x2} ::= [N]_{y1} \rightarrow [E]_{y2} \rightarrow [N]_{y3}, \\ C = \{(\#,x1,*,*,y1,in),(\#,x1,*,*,y1,out), \\ (\#,x2,*,*,y3,in),(\#,x2,*,*,y3,out)\}$$

$$p_{S4}: [E]_{x1} ::= \rightarrow_{y1}, \quad C = \{(\#,x1,*,*,y1,in),(\#,x1,*,*,y1,out)\}$$

$$p_{S5}: [N]_{x1} ::= \circ_{y1}, \quad C = \{(\#,x1,*,*,y1,in),(\#,x1,*,*,y1,out)\}$$

図 7  $G_{Signal}$  のプロダクションの集合

Fig. 7 The production set of  $G_{Signal}$ .

てモデル化しているので非終端エッジラベルおよび終端エッジラベルのアルファベットはともに  $\{*\}$  とする．プロダクションの集合  $P_{Signal}$  は図 7 に示すものである．

図 8 のシグナルフロー線図例に対する導出を図 9 に示す．

$G_{Signal}$  によって生成されるグラフがシグナルフローグラフとして正しいこと，つまり十分性を示す．

定理 1  $G_{Signal}$  によって生成されるグラフは 1 入力 1 出力シグナルフローグラフである．

[証明]  $G_{Signal}$  によるグラフの生成は，スタートグラフに対するプロダクション  $p_{S1}$  の左適用から始まる．このグラフ書き換えで，‘入力’と‘出力’が必ず 1 個ずつ出現し， $p_{S1}$  以外には‘入力’と‘出力’が現れるプロダクションはない．よって，‘入力’と‘出力’がちょうど 1 個ずつ存在することがいえる． $p_{S1}$  の左適用後は，適当にプロダクションを左適用してグラフを生成してゆく．このとき，各プロダクションの定義より，最終的に生成される終端グラフ  $H \in \mathcal{L}(G_{Signal})$  はシグナルフローグラフとしての入力次数と出力次数の条件を満たし，またノード間の接続の制約も満たしていることが分かる．以上より， $G_{Signal}$  が生成する任意のグラフはシグナルフローグラフであることがいえる．



図 8 シグナルフローグラフの例  
Fig. 8 A signal flow graph example.

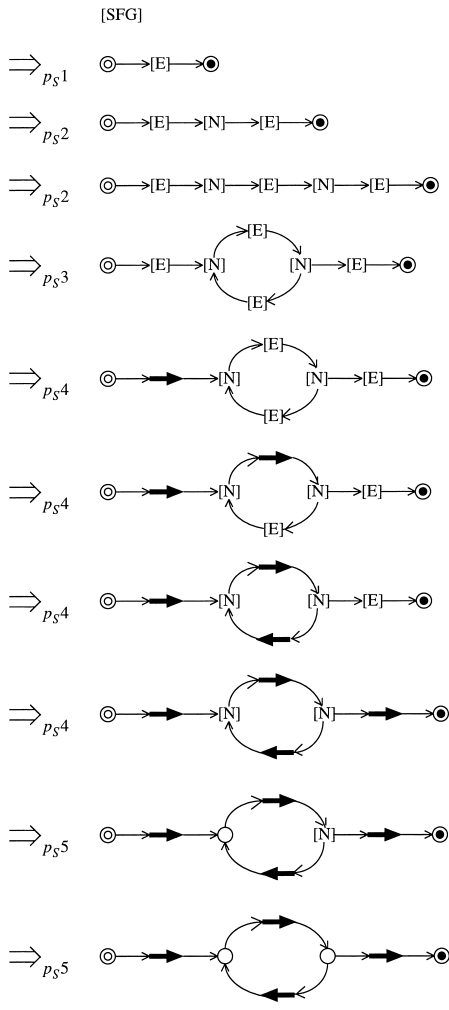


図 9 図 8 のシグナルフローグラフの導出  
Fig. 9 A derivation generating the signal flow graph shown in Fig. 8.

$G_{Signal}$  は制御工学でよく用いられる構造のシグナルフローグラフ、たとえば、直列結合、並列結合、フィードバック結合、あるいはそれらを合成した構造などを生成することができる。また、FIR フィルタ、IIR フィルタの構造も生成できる。しかし、図 10 に示すようなグラフは定義に基づけばシグナルフローグラフと見なせるが  $G_{Signal}$  では生成できない。

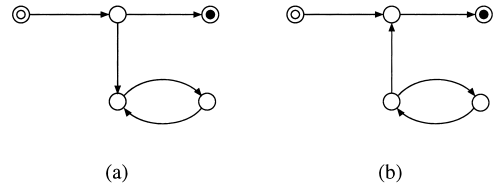


図 10  $G_{Signal}$  で生成できないシグナルフローグラフ  
Fig. 10 A signal flow graph that can not be derived by  $G_{Signal}$ .

#### 4. 文脈依存 NCE グラフ文法の構文解析可能性

グラフ文法に基づいて構文解析するとは、与えられたグラフに対してその文法のプロダクションを用いた導出を見つけだすことを意味する。グラフ文法の構文解析は、グラフ文法に基づく図形言語の性質の解析や処理系の実現のために重要である。

##### 4.1 レイヤード文法

プロダクションの左辺および右辺が任意のグラフであるグラフ文法は 0 型言語を生成することができる。しかし、一般の 0 型言語のメンバーシップ問題は決定不能であることがよく知られている<sup>15)</sup>。そこで、文献 9) および 10) で提案されたレイヤードラベル (layered label) のアイデアを導入して、構文解析アルゴリズムが構成できるように文脈依存 NCE グラフ文法に制限を加える。

ノードラベルのアルファベットの直和分解  $\Sigma_{N \cup T} = L_0 \oplus L_1 \oplus \dots \oplus L_n$  をレイヤードラベル集合という。また、 $L_0, L_1, \dots, L_n$  をレイヤー割当てという。関数  $layer$  は、与えられたグラフ  $H$  の任意のノードに対して、そのラベルが属するレイヤーのインデックスを返すものとする。すなわち

$$\forall x \in V_H, layer(x) = i \Leftrightarrow \lambda(x) \in L_i.$$

定義 5  $L_0 \oplus L_1 \oplus \dots \oplus L_n$  をあるレイヤー割当てとする。このレイヤードラベル集合上のグラフ間の順序を次のように定義する： $H < H' \Leftrightarrow \exists i : |H|_i < |H'|_i \wedge \forall j < i : |H|_j = |H'|_j$ 。ここで  $|H|_k$  は  $\{x \in V_H \mid layer(x) = k\}$  と定義する。すなわち、 $H$  のレイヤー  $L_k$  のラベルを持つノードの数である。■

還元ステップをノードの変化のみに着目すれば、プロダクションコピー  $p = (A, B, C)$  の右辺グラフ  $B$  に属するノードをホストグラフからすべて取り除き、その結果のグラフに左辺グラフ  $A$  に属するノードをすべて加える操作と見ることができる。したがって、次の命題が明らかに成り立つ。

命題 1  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  を文脈依存

NCE グラフ文法,  $\Sigma_{NUT} = L_0 \oplus L_1 \oplus \dots \oplus L_n$  をレイヤードラベル集合とする. このレイヤー割当てに対して, プロダクションコピー  $p = (A, B, C) \in \text{copy}(P)$  が  $A < B$  であるとする. このとき,  $H \Rightarrow_{p-1} H'$  であれば, 同じレイヤー割当てに対して  $H > H'$  となる. ■

**定理 2**  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  を文脈依存 NCE グラフ文法,  $\Sigma_{NUT} = L_0 \oplus L_1 \oplus \dots \oplus L_n$  をレイヤードラベル集合とする. このレイヤー割当てに対して, すべてのプロダクション  $p = (A, B, C) \in P$  が  $A < B$  であるとする. このとき, 任意の (有限) グラフ  $H \in GR_{\Sigma_{NUT}, \Gamma_{NUT}}$  と還元  $H_0 \Rightarrow_{p_1-1} H_1 \Rightarrow_{p_2-1} H_2 \dots \Rightarrow_{p_m-1} H_m$  に対して, 長さ  $m$  は有限である. ■

[証明] レイヤー割当て  $L_0, L_1, \dots, L_n$  に対して, すべてのプロダクション  $p = (A, B, C) \in P$  が  $A < B$  であると仮定する. このとき同じレイヤー割当てに対して, 任意のプロダクションコピー  $p' = (A', B', C') \in \text{copy}(P)$  は  $A' < B'$  を満足するので, 命題 1 より  $H_0 > H_1 > \dots > H_m$  が成り立つ. したがって, 長さ  $m$  は有限である. ■

文脈依存 NCE グラフ文法  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  は, そのすべてのプロダクション  $p = (A, B, C) \in P$  が  $A < B$  となるノードラベルアルファベットのレイヤー割当てが存在するとき, レイヤードの性質を満たすという. また, このとき  $G$  をレイヤード文脈依存 NCE グラフ文法という. 次節で述べるように, レイヤードの性質は文脈依存 NCE グラフ文法の構文解析アルゴリズムが停止することを保証するための条件である. なお, 前述のシグナルフローグラフを生成する文脈依存 NCE グラフ文法  $G_{\text{Signal}}$  は,  $\Sigma_{N_{\text{Signal}}} \cup \Sigma_{T_{\text{Signal}}} = L_0 \oplus L_1$ ,  $L_0 = \{\text{入力, 出力, 節, 枝, スタートラベル}\}$ , そして  $L_1 = \{\text{非終端要素, 非終端節}\}$  のようにレイヤー割当てするとすべてのプロダクション  $p_{\text{Signal}} = (A_{\text{Signal}}, B_{\text{Signal}}, C_{\text{Signal}}) \in P_{\text{Signal}}$  について  $A_{\text{Signal}} < B_{\text{Signal}}$  となる. したがって,  $G_{\text{Signal}}$  はレイヤードの性質を満たす.

#### 4.2 レイヤード文脈依存 NCE グラフ文法の構文解析

構文解析は, 通常, 導出と逆向きにプロダクションを適用, すなわち, 右適用してスタートグラフ  $H_S$  への還元を探索することにより行われる. そこで, 以下のようにレイヤード文脈依存 NCE グラフ文法の構文解析可能性を定義する.

**定義 6**  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  をレイヤード文脈依存 NCE グラフ文法とする.  $G$  が構文

解析可能であるとは, 任意のグラフ  $H \in \mathcal{L}(G)$  について, スタートグラフ  $H_S$  への還元  $H = H_n \Rightarrow_{p_n-1} \dots \Rightarrow_{p_2-1} H_1 \Rightarrow_{p_1-1} H_0 = H_S$  が存在し, 対応する導出  $H_S = H_0 \Rightarrow_{p_1} H_1 \Rightarrow_{p_2} H_2 \Rightarrow \dots \Rightarrow_{p_n} H_n = H$  が成立することである. このとき, 還元  $H = H_n \Rightarrow_{p_n-1} \dots \Rightarrow_{p_2-1} H_1 \Rightarrow_{p_1-1} H_0 = H_S$  を構文解析列という. ■

レイヤード文脈依存 NCE グラフ文法により生成されるグラフで, プロダクションの右適用を用いて構文解析列を求めることができないグラフが存在することを例により示す. 図 11 のグラフ文法  $G_a$  では, 図 11 (b) のグラフ  $H_a \in \mathcal{L}(G_a)$  に対して還元が存在しない. また, 図 12 に示すグラフ文法  $G_b$  ではグラフ  $H_b \in \mathcal{L}(G_b)$  に対して唯一の還元  $H_b \Rightarrow_{p_b4-1} H_1 \Rightarrow_{p_b2-1} H_S$  が存在するが,  $H_S \Rightarrow_{p_b2} H_1 \Rightarrow_{p_b4} H_b$  は成立しない. これらの考察に基づいて, 定義 6 に示す文脈依存 NCE グラフ文法の構文解析可能性を定義した.

ここで, 文脈依存 NCE グラフ文法が構文解析可能であるための条件についてさらに考察する.

**定義 7**  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  をレイヤード文脈依存 NCE グラフ文法とする. 任意の文形式  $H \in S(G)$  と  $p \in \text{copy}(P)$  に対して,  $H \Rightarrow_p H'$  ならば  $H' \Rightarrow_{p-1} H$  であるとき  $G$  は可逆であるという. ■

**定理 3**  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  をレイヤード文脈依存 NCE グラフ文法とする.  $G$  が可逆であれば, 構文解析可能である. ■

$$\Sigma_{N_a} = \{[S], [X]\}, \Sigma_{T_a} = \{\textcircled{a}, \textcircled{b}\},$$

$$\Gamma_{N_a} = \{*\}, \Gamma_{T_a} = \{*\},$$

$$P_a = \{ (p_{a1}: [S]_{x1} ::= \boxed{\textcircled{a}}_{y1} \rightarrow [X]_{y2} ),$$

$$(p_{a2}: [X]_{x1} ::= \boxed{\textcircled{b}}_{y1} ) \}$$

(a) 文脈依存 NCE グラフ文法  $G_a$

$$H_S: [S] \quad H_a: \textcircled{a} \quad \textcircled{b}$$

$$p_{a1} \downarrow \quad p_{a2}^{-1} \downarrow$$

$$\textcircled{a} \rightarrow [X] \quad \textcircled{a} \quad [X]$$

$$p_{a2} \downarrow \quad \times$$

$$H_a: \textcircled{a} \quad \textcircled{b}$$

(b) 導出

(c) (b) のプロダクション列の逆向きの適用

図 11 文脈依存 NCE グラフ文法  $G_a$

Fig. 11 Context-sensitive NCE graph grammar  $G_a$ .



$$\Sigma_{N_b} = \{[S], [A], [B]\}, \quad \Sigma_{T_b} = \{(\textcircled{a}), (\textcircled{b})\},$$

$$\Gamma_{N_b} = \{\alpha, \beta\}, \quad \Gamma_{T_b} = \{\gamma, \delta\},$$

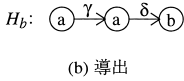
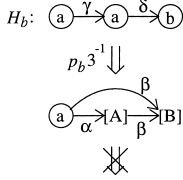
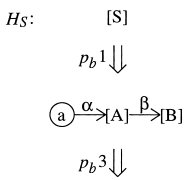
$$P_b = \{ (p_b1: [S]_{x1} ::= \boxed{\textcircled{a} \xrightarrow{\alpha}_{y1} [A]_{y2} \xrightarrow{\beta} [B]_{y3}}),$$

$$(p_b2: [S]_{x1} ::= \boxed{\textcircled{a} \xrightarrow{\alpha}_{y1} [A]_{y2} [B]_{y3}}),$$

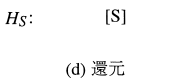
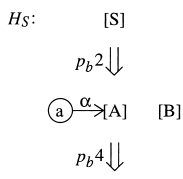
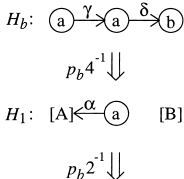
$$(p_b3: [A]_{x1} \xrightarrow{\beta} [B]_{x2} ::= \begin{array}{c} (\textcircled{a}, x1) \xrightarrow{\alpha} \\ (\textcircled{a}, x2) \xrightarrow{\beta} \end{array} \boxed{\textcircled{a} \xrightarrow{\gamma}_{y1} \textcircled{b}_{y2}}),$$

$$(p_b4: [A]_{x1} [B]_{x2} ::= \boxed{\textcircled{a}_{y1} \textcircled{b}_{y2} \xrightarrow{\delta} (\textcircled{a}, x1)} \xleftarrow{\alpha} (\textcircled{a}, x1))$$

(a) 文脈依存 NCE グラフ文法  $G_b$



(c) (b) のプロダクション列の逆向きの適用



(e) (d) のプロダクション列の逆向きの適用

(d) 還元

図 12 文脈依存 NCE グラフ文法  $G_b$

Fig. 12 Context-sensitive NCE graph grammar  $G_b$ .

[ 証明 ]  $H \in S(G)$  かつ  $H_S = H_0 \Rightarrow_{p_1} H_1 \Rightarrow_{p_2} \dots \Rightarrow_{p_n} H_n = H$  を導出とする .  $G$  が可逆であると仮定すると,  $H_i \Rightarrow_{p_{i-1}} H_{i-1} \quad 1 \leq i \leq n$  が成立する . したがって, 還元  $H = H_n \Rightarrow_{p_{n-1}} H_{n-1} \Rightarrow_{p_{n-2}} \dots \Rightarrow_{p_1} H_0 = H_S$  が存在する . この還元に対して, 明らかに  $H_S = H_0 \Rightarrow_{p_1} H_1 \Rightarrow_{p_2} H_2 \Rightarrow \dots \Rightarrow_{p_n} H_n = H$  が成立する . ■

定理 3 で与えたレイヤード文脈依存 NCE グラフ文法  $G$  が構文解析可能であるための十分条件, すなわち, 可逆である条件はすべての文形式に対してチェ

クする必要があり, 一般的問題に対して適用することは難しい . そこで, プロダクションの結合命令を調べて構文解析可能性をチェックできる実用的な十分条件を次に与える .

定理 4  $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  をレイヤード文脈依存 NCE グラフ文法とする . スタートグラフを左辺に持つプロダクションを除くすべてのプロダクション  $p = (A, B, C) \in P$  に対して次の条件 (1) と (2) が成り立つとき,  $G$  は構文解析可能である :

- (1)  $\forall \sigma \in \Sigma_{NUT}, \forall v_A \in V_A, \forall \beta \in \Gamma_{NUT}, \forall d \in \{\text{in}, \text{out}\}, (\sigma, v_A, \beta, \gamma, v_B, d) \in C$ .
- (2)  $(\sigma, v_{A1}, \beta_1, \gamma, v_B, d) \in C \wedge (\sigma, v_{A2}, \beta_2, \gamma, v_B, d) \in C$  ならば,  $v_{A1} = v_{A2} \wedge \beta_1 = \beta_2$  . ■

[ 証明 ] 条件 (1) と (2) が成り立つと仮定する . このとき, 任意のグラフ  $H \in GR_{\Sigma_{NUT}, \Gamma_{NUT}}$  とプロダクションコピー  $p' \in \text{copy}(P)$  に対して,  $H \Rightarrow_{p'} H'$  ならば  $H' \Rightarrow_{p'^{-1}} H$  である . なぜならば, 導出ステップ  $H \Rightarrow_{p'} H'$  においてマザーグラフとレストグラフのノード間にあった取り除かれたエッジが, 還元ステップ  $H' \Rightarrow_{p'^{-1}} H$  で条件 (1) によりすべて還元され, かつ条件 (2) により余分なエッジが生成されないからである . したがって, 条件 (1) と (2) が成り立つとき  $G$  は可逆であり, 定理 3 より構文解析可能である . ■

前節で定式化したシグナルフローグラフの生成文法  $G_{\text{Signal}}$  は定理 4 の条件 (1) および (2) を満足することが容易に分かる . したがって,  $G_{\text{Signal}}$  は構文解析可能である .

文脈依存 NCE グラフ文法に対するボトムアップ幅優先探索による構文解析アルゴリズム PARSE を図 13 に示す . このアルゴリズムで, 任意のグラフ  $H \in S[d]$  は解析するグラフ  $H_{\text{in}}$  から長さ  $d$  の還元で導かれるグラフである . したがって, 定理 2 より, 任意のレイヤード文脈依存 NCE グラフ文法に対してこのアルゴリズムは停止する . また, グラフ書き換えの各レベル  $d$  で, すべてのグラフ  $H \in S[d]$  について,  $H$  に右適用可能なプロダクションコピー  $p' \in \text{copy}(P)$  で  $H \Rightarrow_{p'^{-1}} H'$  に対して  $H' \Rightarrow_{p'} H$  となるすべてのプロダクションコピーを右適用して, 次のレベルのグラフの集合  $S[d+1]$  を網羅的に構成している . したがって, 構文解析可能な文脈依存 NCE グラフ文法のもとでこの文法に対して正しいグラフを解析した場合, ある  $d$  で  $H_S \in S[d]$  となる . さらにこのとき, 右適用の各ステップ  $H \Rightarrow_{p'^{-1}} H'$  に対して, 同じプロダクションコピーを左適用  $H' \Rightarrow_{p'} H$  してもとのグラフ  $H$  に戻ることをチェックしているので, このアルゴリズムで求めたプロダクションコピーの系列をスタートグラフ

```

PARSE{
  input
   $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$  : 文脈依存 NCE グラフ文法
   $H_{in}$ : 解析するグラフ
  output
  "valid" or "invalid" : プロダクションコピーの系列
  variable
   $S[]$  : グラフの集合の配列
   $d$  : グラフ書き換えのレベル
   $S_{map}$  : 組  $(p_B, rhs(p'_B))$  の集合,  $p'_B \in copy(p_B)$ 
  method
  グラフ  $H_{in}$  に記号  $\perp$  を付随させる ;
   $d \leftarrow 0$  ;
   $S[d] \leftarrow \{H_{in}\}$  ;
  while {
    if  $S[d] = \phi$  then
      output("invalid") and exit ;
     $S[d+1] \leftarrow \phi$  ;
    for  $S[d]$  にあるすべてのグラフ  $H$  do {
       $S_{map} \leftarrow \phi$  ;
      while  $\left( \begin{array}{l} \exists p' = (A', B', C') \in copy(P), \\ B' \text{ は } H \text{ の誘導部分グラフ,} \\ (p, rhs(p')) \notin S_{map} \end{array} \right)$  do {
         $H \Rightarrow_{p'-1} H'$  ;
         $H' \Rightarrow H''$  ;
        if  $H, p' = H$  then {
           $H'$  に  $p'$  を付随させる ;
           $H$  に付随している記号から  $H'$  に付随している
          記号へリンクを張る ;
           $S[d+1] \leftarrow S[d+1] \cup \{H'\}$  ;
        }
         $S_{map} \leftarrow S_{map} \cup \{(p, rhs(p'))\}$  ;
      }
    }
     $d \leftarrow d+1$  ;
    if (スタートグラフ  $H_S$  が  $S[d]$  に存在する) then {
      output("valid");
       $H_S$  に付随したプロダクション・コピーから  $\perp$  に到達するまで
      リンクをたどりながらプロダクション・コピーを出力する ;
      exit ;
    }
  }
}

```

図 13 ボトムアップ幅優先探索による構文解析アルゴリズム PARSE

Fig. 13 Parsing algorithm PARSE in a bottom-up and breadth-first manner.

に左適用するともとのグラフ  $H_{in}$  を導出する。したがって、次の性質 1 が成立する。

**性質 1** アルゴリズム PARSE はレイヤードの性質を満たし、かつ構文解析可能である文脈依存 NCE グラフ文法に基づいて任意のグラフを構文解析したとき、文法に対して正しいグラフには 'valid' とそのグラフを導出するプロダクションの系列を出力して停止する。また、文法に対して正しくないグラフには 'invalid' を出力して停止する。 ■

## 5. おわりに

広いクラスのビジュアル言語のグラフ構文規則を柔軟に定式化することができる文脈依存 NCE グラフ文法を定義した。そして、この文法に対する構文解析可能性について議論した。シグナルフローグラフのグラフ構文規則を定式化した文脈依存 NCE グラフ文法に

対して、本論文で述べたボトムアップ幅優先探索に基づくパーサを実現してその理論的および実用的有効性を確認している<sup>16)</sup>。

現在、文脈依存 NCE グラフ文法に基づいたビジュアル言語に対して、図の編集や再利用などの操作をプロダクションを用いて形式的に記述する研究を行っている<sup>13)</sup>。前節で言及した文脈依存 NCE グラフ文法に対するボトムアップ幅優先探索による構文解析アルゴリズム PARSE は指数時間の計算的複雑さを持つ。文献 11) で考察されたような多項式時間複雑さの構文解析アルゴリズムと、そのアルゴリズムを適用できるための条件についての研究も今後の課題である。

謝辞 本研究を進めるにあたって、有益なご討論とご助言をいただきました日本大学の夜久竹夫教授ならびに東洋大学の土田賢省助教授に深く感謝いたします。

本研究の一部は、平成 12 年度東洋大学特別研究の援助により行いました。ここに謝意を表します。

## 参考文献

- Engelfriet, J. and Rozenberg, G.: Node Replacement Graph Grammar, *Handbook of Graph Grammars and Computing by Graph Transformation*, Rozenberg, G. (Ed.), pp.1-94, chapter 2, World Scientific (1997).
- Costagliola, G., De Lucia, A., Orefice, S. and Tortora, G.: A Parsing Methodology for the Implementation of Visual Systems, *IEEE Trans. Softw. Eng.*, Vol.23, No.12, pp.777-799 (1997).
- Marriott, K. and Meyer, B.: On the Classification of Visual Languages by Grammar Hierarchies, *Journal of Visual Languages and Computing*, Vol.8, pp.375-402 (1998).
- Erwig, M.: Visual Graphs, *Proc. IEEE Symp. Visual Languages*, pp.122-129 (1999).
- Bardohl, R.: GENGED A Generic Graphical Editor for Visual Languages based on Algebraic Graph Grammars, *Proc. IEEE Symp. Visual Languages*, pp.48-55 (1998).
- Zhang, D. and Zhang, K.: VisPro: A Visual Language Generation Toolset, *Proc. IEEE Symp. Visual Languages*, pp.195-202 (1998).
- Münch, M., Schürr, A. and Winter, A.: Integrity Constraints in the Multi-paradigm Language PROGRES, *Proc. IEEE Symp. Visual Languages*, pp.84-85 (1998).
- Adachi, Y., Imaki, T., Tsuchida, K. and Yaku, T.: Program Visualization by Attribute Graph Grammars, *Proc. CDROM The Fundamental Conference of the 15th IFIP World Computer Congress* (1998).

- 9) Rekers, J. and Schürr, A.: A Parsing Algorithm for Context-Sensitive Graph Grammars, Technical Report 95-05, Leiden University, the Netherlands.  
Available via ftp from ftp.wi.leidenuniv.nl, file/pub/CS/TechnicalReports/1995/tr95-95.ps.gz (1995).
- 10) Rekers, J. and Schürr, A.: Defining and Parsing Visual Languages with Layered Graph Grammars, *Journal of Visual Languages and Computing*, Vol.8, pp.27-55 (1997).
- 11) Zhang, D. and Zhang, K.: Reversed Graph Grammar: A Specification Tool for Diagrammatic VPLs, *Proc. IEEE Symp. Visual Languages*, pp.284-291 (1997).
- 12) 安達由洋, 小林 卓, 土田賢省, 夜久竹夫: ブロック線図文法, 信学論 ( D-I ), Vol.J83-D-I, No.1, pp.45-54 (2000).
- 13) Adachi, Y., Kobayashi, S., Tsuchida, K. and Yaku, T.: An NCE Context-sensitive Graph Grammar for Visual Design Languages, *Proc. IEEE Symp. Visual Languages*, pp.228-235 (1999).
- 14) Adachi, Y., Nakajima, Y. and Kobayashi, S.: An NCE Context-sensitive Graph Grammar and its Application to Visual Languages, *Proc. Conference on Software: Theory and Practice of the 16th IFIP World Computer Congress*, pp.108-116 (2000).
- 15) Salomaa, A.: *Formal Languages*, ACM Monograph Series, Academic Press, New York (1973).
- 16) Adachi, Y., Kobayashi, S., Suda, Y. and Tsuchida, K.: Block Diagram Grammar and Signal Flow Graph Grammar with Their Applications, *Proc. IEEE System, Man, Cybernetics*, Vol.IV, pp.141-146 (1999).

(平成 12 年 8 月 29 日受付)

(平成 12 年 9 月 21 日採録)

安達 由洋 (正会員)



1980 年名古屋大学大学院工学研究科情報工学専攻博士後期課程修了。同年東洋大学工学部情報工学科講師。2001 年同大学工学部情報工学科教授。工学博士。グラフ文法, グラフ描画, 制約プログラミング等の研究に従事。ACM, IEEE, 電子情報通信学会, 人工知能学会各会員。

小林 卓



1999 年東洋大学大学院工学研究科電気工学専攻博士前期課程修了。グラフ文法とその応用に関する研究に従事。現在, NEC ソフト (株) 勤務。