

文脈依存 NCE グラフ文法とその構文解析可能性

安達 由洋

小林 卓

東洋大学

NCE ソフト株式会社

概要

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。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。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

Suguru KOBAYASHI

Toyo University

NEC SOFT Ltd.

Abstract

Graph grammars provide a mechanism in which generation and transformation of visual objects can be modeled in a mathematically precise way. The context-sensitive NCE graph grammar defined in this paper 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 discuss the parsability of the context-sensitive NCE graph grammar, and present a sufficient condition for determining whether or not a context-sensitive NCE graph grammar is parsable. 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 [1, 2], 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 [3, 4]. 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、中

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 等は、ビジュアル言語の表現力に関する研究の結果を踏まえて、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 [2].

Rekers 等は “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 [5]. 文献 [5] では “layered graph grammar” の

構文解析法についても報告している。Zhang 等は “layered graph grammar” をもとにして “reserved graph grammar” を開発した [6]。そして, “reserved graph grammar” を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, 安達等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 [7] 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-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 [1] 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。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。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。

本論文の内容の一部は, 1999 IEEE Symposium on Visual Languages (VL'99) [8], 及び International Conference on Software: Theory and Practice of the 16th IFIP World Computer Congress (ICS2000) [9] で報告している。

2 文脈依存 NCE グラフ文法

本論文で用いる用語や記法は, その多くを Engelfriet 等の論文 [1] 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

Σ をノードラベルのアルファベット, Γ をエッジラベルのアルファベットとする。 Σ と Γ 上のグラフとは, 三つ組 $H = (V, E, \lambda)$ 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 $\lambda: V \rightarrow \Sigma$ はノードラベリング関数である。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, 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, λ_H と書くことにする。

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) Σ_N は非終端ノードラベルのアルファベットである。
- (2) Σ_T は終端ノードラベルのアルファベットである。 Σ_N と Σ_T はそれぞれ空でない有限集合であり, 互いに素である。 $\Sigma_{NUT} = \Sigma_N \cup \Sigma_T$ はノードラベルのアルファベットである。
- (3) Γ_N は非終端エッジラベルのアルファベットである。
- (4) Γ_T は終端エッジラベルのアルファベットである。 Γ_N と Γ_T はそれぞれ空でない有限集合であり, 互いに素である。 $\Gamma_{NUT} = \Gamma_N \cup \Gamma_T$ 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。 \square

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 θ_l と B_1 から B_2 への (右辺の) 同形写像 θ_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。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.

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 Σ_{NUT} と Γ_{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.

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

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 $H \xrightarrow{p'} H'$ 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 $H_0 \xrightarrow{p'_1} H_1 \xrightarrow{p'_2} \dots \xrightarrow{p'_n} H_n$ を導出といい, 系列の長さ n を導出の長さという. \square

導出 $H = H_0 \xrightarrow{p'_1} H_1 \xrightarrow{p'_2} \dots \xrightarrow{p'_n} H_n = H'$ が存在するとき, $H \xrightarrow{*} H'$ 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 $\mathcal{S}(G)$ と書く. $\mathcal{L}(G) = \{H \in GR_{\Sigma_T, \Gamma_T} \mid H_S \xrightarrow{*} H\}$ を G によって生成されるグラフ言語という.

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 Σ_{NUT} と Γ_{NUT} 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.

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

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' に変換される時, $H \xrightarrow{p'} H'$ と書く. $H \xrightarrow{p'} H'$ を還元ステップという. また, 還元ステップの系列 $H_0 \xrightarrow{p'_{n-1}} H_1 \xrightarrow{p'_{n-2}} H_2 \dots \xrightarrow{p'_1} H_n$ を還元といい, 系列の長さ n を還元の長さという. 特に, $H_0 \xrightarrow{p'_{n-1}} H_1 \xrightarrow{p'_{n-2}} H_2 \dots \xrightarrow{p'_1} H_n = H_S$ をスタートグラフ H_S への還元という. \square

3 構文解析可能性

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.

3.1 レイヤード文法

ノードラベルのアルファベットの直和分解 $\Sigma_{NUT} = L_0 \oplus L_1 \oplus \dots \oplus L_n$ をレイヤードラベル集合という. また, L_0, L_1, \dots, L_n をレイヤー割り当てという. 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 4 $L_0 \oplus L_1 \oplus \dots \oplus L_n$ をあるレイヤー割り当てとする. 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。 □

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り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。レイヤードの性質は文脈依存 NCE グラフ文法に対するボトムアップ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。

3.2 構文解析可能条件

レイヤード文脈依存 NCE グラフ文法の構文解析可能性を次のように定義する。

定義 5 $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$ 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 $H = H_n \xRightarrow{p_{n-1}} \cdots \xRightarrow{p_2} H_2 \xRightarrow{p_1} H_1 \xRightarrow{p_0} H_0 = H_S$ が存在し、対応する導出 $H_S = H_0 \xRightarrow{p_1} H_1 \xRightarrow{p_2} H_2 \xRightarrow{p_3} \cdots \xRightarrow{p_n} H_n = H$ が成立することである。このとき、還元 $H = H_n \xRightarrow{p_{n-1}} \cdots \xRightarrow{p_2} H_2 \xRightarrow{p_1} H_1 \xRightarrow{p_0} H_0 = H_S$ を構文解析列という。 □

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更に考察する。

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。任意の文形式 $H \in S(G)$ と $p \in \text{copy}(P)$ に対して、 $H \xRightarrow{p} H'$ ならば $H' \xRightarrow{p^{-1}} H$ であるとき G は可逆であるという。 □

定理 1 $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$ 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。 □

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 2 $G = (\Sigma_N, \Sigma_T, \Gamma_N, \Gamma_T, S, P)$ 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) $\forall \sigma \in \Sigma, \forall v_A \in V_A, \forall \beta \in \Gamma, \forall d \in \{\text{in}, \text{out}\}, (\sigma, v_A, \beta, \gamma, v_B, d) \in C.$

- (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、 $v_{A1} = v_{A2} \wedge \beta_1 = \beta_2.$ □

次の性質を持つボトムアップ構文解析アルゴリズム PARSE を構成できる。

性質 1 アルゴリズム PARSE は、構文解析可能でかつ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。 □

4 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。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 [8]。

参考文献

- [1] J. Engelfriet and G. Rozenberg, *Handbook of Graph Grammars and Computing by Graph Transformation* (Rozenberg, G., eds), World Scientific (1997), 1-94.
- [2] Marriott, K., and Meyer, B., *Journal of Visual Languages and Computing*, 8 (1998), 375-402.
- [3] Costagliola, G., De Lucia, A., Orefice, S., and Tortora, G., *IEEE Trans. Software Engineering*, Vol.23, No.12 (1997), 777-799.
- [4] Adachi, Y., Imaki, T., Tsuchida, K. and Yaku, T., *Proc. CDROM The Fundamental Conference of the 15th IFIP World Computer Congress*, (1998).
- [5] Rekers, J., and Schürr, A., *Journal of Visual Languages and Computing*, 8 (1997), 27-55.
- [6] Zhang, D., and Zhang, K., *Proc. IEEE Symp. Visual Languages* (1997), 284-291.
- [7] 安達由洋, 小林卓, 土田賢省, 夜久竹夫, 信学論 (D - I), Vol.J83-D-I, No.1, pp.45-54, Jan. 2000.
- [8] Adachi, Y., Kobayashi, S., Tsuchida, K., and Yaku T., *Proc. IEEE Symp. Visual Languages* (1999), 228-235.
- [9] Adachi, Y., Nakajima, Y., and Kobayashi, S., *Proc. Conference on Software: Theory and Practice of the 16th IFIP World Computer Congress* (2000), 108-116.