

シグナルフローグラフのためのグラフ文法*

1 S-1

小林 卓† 伊藤 崇† 安達 由洋†

† 東洋大学工学部

1 はじめに

シグナルフローグラフの構文規則をグラフ文法を用いて定式化する。このシグナルフローグラフ文法 (SFGG) は、文脈依存のプロダクションを一部に含む6個のプロダクションからなっており、非常に簡潔なものである。さらに、シグナルフローグラフをSFGGに基づいて構文解析するパーザについても報告する。

2 グラフ文法

SFGG を定式化するための基礎となる文脈依存グラフ文法を定義する。

グラフ文法とは4つ組 $G = (\Sigma_n, \Sigma_t, S, P)$ 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, 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。 $\Sigma = \Sigma_n \cup \Sigma_t$ はノードラベルのアルファベットである。
- (3) $S \in \Sigma_n$ はエッジのないノード1点からなる スタートグラフ G_s のノードのラベルである。
- (4) P は プロダクション p の空でない有限集合である。プロダクション p は組 $p = (A ::= B, C)$ で表される。ただし、
 - (a) $A = (V_A, E_A, \lambda_A)$ は Σ 上のグラフ。
 - (b) $B = (V_B, E_B, \lambda_B)$ は Σ 上のグラフ。ここで、 $|V_A| \leq |V_B|$ 。
 - (c) $C \subseteq V_X \times V_Y \times \{\text{in, out}\}$ は 結合関係 (connection relation) である。 C の各要素 (v_x, v_y, d) を p の 結合命令 (connection instruction) 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重複して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 に対する プロダクション・コピー あるいは単に プロダクション・コピー という。 $copy(P) = \cup_{p \in P} copy(p)$ とする。

$G = (\Sigma_n, \Sigma_t, S, P)$ をグラフ文法とする。 $H = (V_H, E_H, \lambda_H)$ を Σ 上のグラフ、 M を H の誘導部

分グラフ、 $p \in P$ を G のプロダクション、そして $p' = (A' ::= B', C')$ を p に対するプロダクション・コピーとする。そして H と B' は互いに素であるとする。 $A' = M$ かつ、 Σ 上のグラフ H' が次のように構成されるるとき、 $H \Rightarrow H'$ あるいは単に $H \Rightarrow H'$ と書く。

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

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

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

$H \Rightarrow_{p'} H'$ を H から H' への 導出ステップ (derivation step) と呼ぶ。また、導出ステップの系列を 導出 (derivation) と呼ぶ。

\Rightarrow の反射的推移的閉包を \Rightarrow^* と表す。

スタートグラフ G_s から導出される終端グラフの集合、すなわち $\mathcal{L}(G) = \{H \in \mathcal{G}_\Sigma \mid G_s \Rightarrow^* H\}$ を グラフ文法 G によって生成されるグラフ言語 という。

3 SFGG

前節で定義したグラフ文法に基づいて(1入力1出力)SFGG を定式化する。

定義 1 (SFGG)

$G_{\text{Signal}} = (\Sigma_{n, \text{Signal}}, \Sigma_{t, \text{Signal}}, [SFG], P_{\text{Signal}})$ をグラフ文法とする。ただし、終端ノードアルファベット $\Sigma_{n, \text{Signal}}$ と非終端ノードアルファベット $\Sigma_{t, \text{Signal}}$ を Fig. 1 に示し、プロダクション集合 P_{Signal} を Fig. 2 を示す。

Nonterminal node alphabet $\Sigma_{n, \text{Signal}}$	Terminal node alphabet $\Sigma_{t, \text{Signal}}$
[SFG] : start label	⊙ : input
[Circle] : nonterminal signal	● : output
[Arrow] : nonterminal element	○ : signal
[Arrows] : nonterminal element list	→ : element

Fig. 1. The node alphabet of the SFGG

G_{Signal} によって生成されるグラフ言語 $\mathcal{L}(G_{\text{Signal}})$ を シグナルフローグラフ言語 と呼び、 $\mathcal{L}(G_{\text{Signal}})$ の要素

*A Graph Grammar for Signal Flow Graph
†Suguru Kobayashi, Takashi Itoh, Yoshihiro Adachi, Department of Information and Computer Sciences, Toyo University

- P1. $[SFG]_{x_1} ::= \text{[Circle]}_{y_1} \dots \text{[Arrows]}_{y_2} \dots \text{[Circle]}_{y_3} \quad C = \phi$
- P2. $[\text{Arrows}]_{x_1} ::= [\text{Arrow}]_{y_1} \dots \text{[Circle]}_{y_2} \dots [\text{Arrows}]_{y_3}$
 $C = ((x_1, y_1, in), (x_1, y_3, out))$
- P3. $[\text{Circle}]_{x_1} [\text{Circle}]_{x_2} ::= [\text{Circle}]_{y_1} \dots [\text{Arrows}]_{y_2} \dots [\text{Circle}]_{y_3}$
 $C = ((x_1, y_1, in), (x_1, y_1, out), (x_2, y_3, in), (x_2, y_3, out))$
- P4. $[\text{Arrows}]_{x_1} ::= [\text{Arrow}]_{y_1} \quad C = ((x_1, y_1, in), (x_1, y_1, out))$
- P5. $[\text{Circle}]_{x_1} ::= \text{[Circle]}_{y_1} \quad C = ((x_1, y_1, in), (x_1, y_1, out))$
- P6. $[\text{Arrow}]_{x_1} ::= \text{[Arrow]}_{y_1} \quad C = ((x_1, y_1, in), (x_1, y_1, out))$

Fig. 2. The production set P_{signal} of the SFGG

をシグナルフローグラフと定義する。 G_{signal} をシグナルフローグラフ文法と呼ぶ。 □

4 シグナルフローグラフの生成例

Fig. 3のシグナルフローグラフ例に対するSFGGに基づいた導出の一つとしてFig. 4がある。



Fig. 3. Signal flow graph example

5 シグナルフローグラフ・パーザ

SFGGに基づいて、図を構文解析するシグナルフローグラフ・パーザを Prolog を用いてを実現した。このパーザは、与えられたシグナルフローグラフを構文解析して、シグナルフローグラフとして適格か否かを判断する。そして、適格なシグナルフローグラフに対してはそれを生成するプロダクション・コピー列を出力する。このパーザは、以下のような特徴を持つ；パーザは、必ず停止する（停止性）、パーザから出力されたプロダクション・コピー列をスタートグラフに適用すると解析したシグナルフローグラフを導出する（健全性）、正しいシグナルフローグラフに対して、それを生成するプロダクション・コピー列をパーザは必ず出力する（完全性）。

パーザで構文解析した例を Fig. 5に示す。

6 まとめ

シグナルフローグラフを生成するSFGGをグラフ文法を用いて定式化し、これに基づいたパーザを実現した。シグナルフローグラフのノードと矢（信号）の両方をSFGGのノードとして定義している。これはブロック線図文法と異なっている [1]。

参考文献

[1] K.Anzai, Y.Adachi et al., IEEE ISCAS'97 pp.1970-1973(1997).

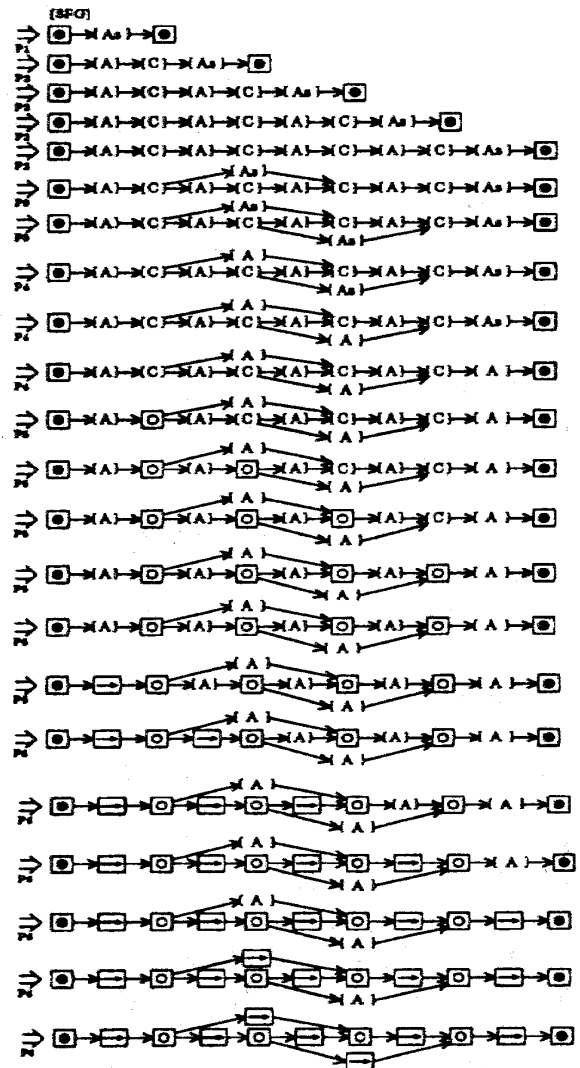


Fig. 4. Derivation to the signal flow graph

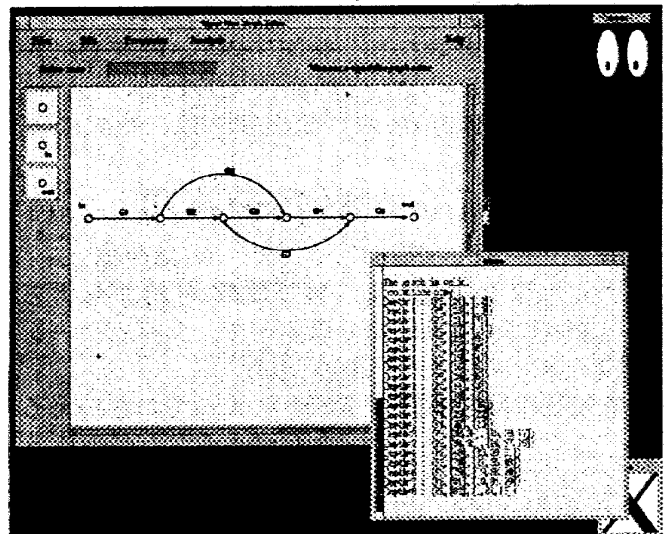


Fig. 5. A snapshot of the signal flow graph editor and parser