図形型プログラム言語のグラフ文法による定義

7D－2

深瀬　宰史　東京電機大学
田辺　文雄　日本電気（株）
山崎　浩一　東京電機大学
夜久　竹夫　日本大学
米田　信夫　東京電機大学

１．はじめに

効果的なプログラムの設計・開発・保守を目的とした，プログラム流れ図を中心とした図形型プログラム言語は盛んに研究され[2][5][6]，ＰＡＤ，ＳＰＤ，階層化流れ図言語Ｈｉｃｈａｒｔ[2][6]等が知られている．しかし，図形型プログラム言語に対しては，厳密な文法が定められていない．そのため処理系の体系的な設計が困難である等の問題点がある．

そこで，我々は，図形型プログラム言語へのグラフ言語理論の応用を考える．すなわち，図形型プログラム言語を記述するグラフ文法を定義する．本論文では，Ｃ言語を例として，Ｈｉｃｈａｒｔプログラム図式を記述するグラフ文法を提案する．本論文の結果は，処理系の体系的な設計に応用される．

２．準備

グラフ文法とは，グラフの集合（グラフ言語）を生成する規則のことで，文字列文法の拡張として知られている．図形型プログラム言語をグラフの集合（グラフ言語）と見なし，そのグラフの集合，すなわち図形型プログラム言語を生成するグラフ文法を定める．

定義２．１　（グラフ）

Ω_M，Ω_Aはそれぞれ異なった有限アルファベット，Ω_Mは節の記号の有限集合，Ω_Aは辺のラベルの有限集合とする．

Ω_M，Ω_A上のグラフＧは３個組$G=(N_G, V_G, E_G)$で表される，ここで，N_Gは節の有限集合，$V_G: N_G \to \Omega_M$（節とアルファベットの対応付け），$E \subseteq N_G \times \Omega_A \times N_G$（辺の集合）．□

グラフＧとグラフＨが同型（Ｇ≡Ｈと書く）であるとは，ある全単射$f: N_G \to N_H$が存在して，

（１）$\forall n \in N_H,\ V_G(n) = V_H(f(n))$，

（２）$(n, a, m) \in E_G \overset{iff}{\Leftrightarrow} (f(n), a, f(m)) \in E_H$

を満たすことである．

集合｛Ｇ｜ＧはΩ_M，Ω_A上のグラフ｝を$g^*(\Omega_M, \Omega_A)$で表す．□

定義２．２　（グラフ文法）

グラフ文法は５個組$Q=(\Omega_N, \Omega_T, \Omega_A, S, R)$で表される，ここで，$\Omega_N$は非終端記号の有限アルファベット，$\Omega_T$は終端記号の有限アルファベット，$\Omega_A$は辺のラベルの有限アルファベット，$S \in \Omega_N$は開始非終端記号，Ｒは生成規則の有限集合で，生成規則ｒは４個組$r=(G, H, I, O)$で表される．ただし，$G \in g^*(\Omega_N, \Omega_A)$はただ一つの節からなるグラフ，$H \in g^*(\Omega_N \cup \Omega_T, \Omega_A)$かつ$N_H \neq \phi$，$I \in N_H$は入力節，$O \in N_H$は出力節．

生成規則$r=(G, H, I, O)$，ここでＧは非終端記号Ａをもつ節からなるグラフ，は”$A \to H^I_O$”の形で表現される．□

定義２．３　（グラフ文法の導出）

$Q=(\Omega_N, \Omega_T, \Omega_A, S, R)$をグラフ文法，$G, H \in g^*(\Omega_N \cup \Omega_T, \Omega_A)$とする．このとき，ＨがＱによってＧから直接導出される（$G \overset{p}{\Rightarrow} H$，又は単に$G \Rightarrow H$と書く）とは，$(L, K, I, O) \in R$が存在して，

（ａ）ＧがＧ’とＧ”に分けられ，Ｇ’≡Ｌ（このとき$N_{G'}$はただ一つの節からなり，その節をｎで表す），

（ｂ）ＨがＨ’とＨ”に分けられ，

（１）Ｈ”≡Ｇ”，

（２）Ｈ’≡Ｋ，

（３）$E_H = E_{H'} \cup E_{H''} \cup E_1 \cup E_2 \cup E_3$，

ここで，

$E_1=\{(m,a,I) \mid (m,a,n) \in E_G \text{ and } m \neq n\}$，

$E_2=\{(O,a,m) \mid (n,a,m) \in E_G \text{ and } m \neq n\}$，

$E_3=\{(O,a,I) \mid (n,a,n) \in E_G\}$．□

次にペア文法を定義する[1]．ペア文法は２つのグラフ文法G_1，G_2の組から成る．ペア文法はG_1とG_2を対応付けを行う．文字列文法はグラフ文法の制限されたものと見なせるので，G_1としてグラフ文法の代わりに文字列文法を選択することができる．ここであるプログラム言語のＢＮＦをG_1として選び，ペア文法G_pを構成する．これによりG_1（ＢＮＦ）と対応するグラフ文法G_2が構築される．図形型プログラム言語の処理系作成上のG_2の重要性は，プログラム言語の処理系作成上のＢＮＦ（G_1）の重要性と等しい．

定義２．４　（非終端ペアリング，グラフペア）

Ω_N，Ω_T，Ω_Aを各アルファベット，$G, H \in g^*(\Omega_N, \Omega_T, \Omega_A)$，$N_G^{NT} = \{n \in N_G \mid V_G(n) \in \Omega_N\}$，$N_H^{NT} = \{n \in N_H \mid V_H(n) \in \Omega_N\}$とする，このとき，ある関数ｈがＧとＨの非終端ペアリングとは，ｈは

A Graph Grammar for A Program Diagram Language

Suzuhito Fukase , Tokyo Denki University
Fumio Tanabe , NEC Corp.
Koichi Yamazaki , Tokyo Denki University
Takeo Yaku , Nihon University
Nobuo Yoneda , Tokyo Denki University

$N_G^T \to N_H^T$への全単射で，全ての節$n \in N_G^T$に対して
$V_G(n)=V_H(h(n))$を満たすときをいう．

Ω_N，Ω_T，Ω_A上のグラフペアXは3個組$X=(G, H, h)$で表す，ここで，

（1）$G, H \in g^*(\Omega_N \cup \Omega_T, \Omega_A)$，

（2）hはGとHの非終端節ペアリング． □

定義2．5　（ペア文法）

ペア文法Qは5個組$Q=(\Omega_N, \Omega_T, \Omega_A, S, P)$と表される，ここで，$\Omega_N$は非終端のアルファベット，$\Omega_T$は終端のアルファベット，$\Omega_A$は辺のラベルのアルファベット，$S \in \Omega_N$は開始非終端，Pは$(l, r, h)$の形の3個組の有限集合，ただし，

（1）l，r：定義2．2で定義した生成規則，

（2）生成規則lとrの”左辺”はともに等しい，

（3）h：GとHの非終端ペアリング． □

定義2．6　（ペア文法の導出）

$Q=(\Omega_N, \Omega_T, \Omega_A, S, P)$をペア文法

$X=(G_x, H_x, h_x)$，$Y=(G_y, H_y, h_y)$をΩ_N，Ω_T，Ω_A上のグラフペアとする．YがQによってXから直接導出される（$X \Rightarrow Y$と書く）とは，$(l, r, h) \in P$と$n \in G_x$，$m \in H_x$（n，m：非終端節）が存在して，

（1）$h_x(n)=m$，

（2）$G_x \Rightarrow G_y$（規則lを節nに適用），

（3）$H_x \Rightarrow H_y$（規則rを節mに適用），

（4）$h_y=h \cup h_x-\{(n, m)\}$

を満たすことである．

YがQによってXから導出される（$X \overset{+}{\Rightarrow} Y$と書く）とは，$Z_1, Z_2, \cdots, Z_n$（$\Omega_N, \Omega_T, \Omega_A$上のグラフペア）が存在して，$X=Z_1$，$Y=Z_n$，$Z_i \Rightarrow Z_{i+1}$(i=1,2,…,n-1)を満たすことである． □

定義2．7　（ペア言語）

ペア文法Qによって定義されるペア言語L_Qは

$L_Q=\{X \mid X: \Omega_T, \Omega_A$上のグラフペア，$S_Q \overset{+}{\Rightarrow} X\}$

である，ただし，S_Qは開始グラフペア$S=(N, M, h)$，ここで，Nは記号Sをもつ節nからなるグラフ，Mは記号Sをもつ節mからなるグラフ，$h(n)=m$． □

3．結果

この節では，C言語のBNF（G_1），Hichart言語を生成するグラフ文法（G_2）からなるペア文法G_Pを考え，そのHichart言語を生成するグラフ文法（G_2）を示す．ここで，ⓘは入力節であることを意味し，ⓞは出力節であることを意味する．又，G::=Hは生成規則（G，H，I，O）を意味する（I，OはH内の↓↑）．

例3．1　C言語に対応するHichart言語を生成するグラフ文法（一部）

$Q_H=(\Omega_{H,N}, \Omega_{H,T}, \Omega_{H,A}, S_H, P_H)$

ここで，

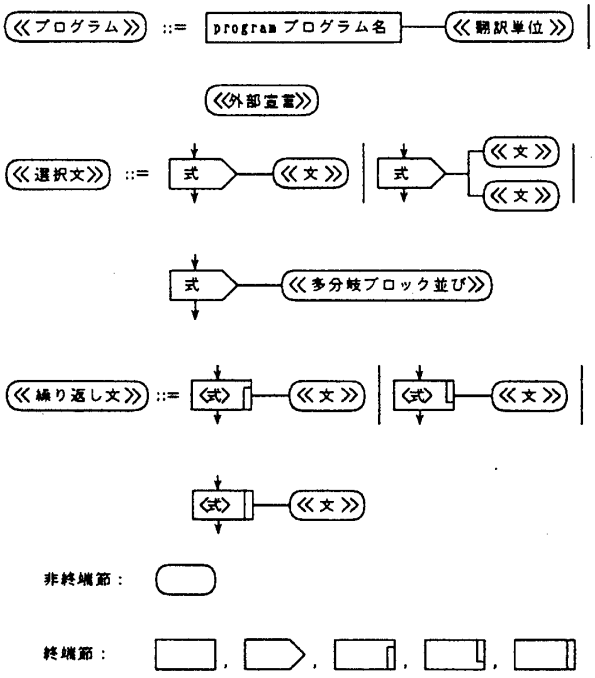
$\Omega_{H,N}=\{$<<ﾌﾟﾛｸﾞﾗﾑ>>,<<翻訳単位>>,<<文>>,…$\}$

$\Omega_{H,T}=\{a,b,\cdots,0,1,\cdots 9,*,\cdots\}$

$\Omega_{H,A}=\{I,O,T,F,\cdots\}$

S_H　=<<ﾌﾟﾛｸﾞﾗﾑ>>

P_H：（一部）



主張3．1

Hichart言語の基本部分は例3．1のグラフ文法により定義される． □

参考文献

[1] Reihold Franck, A class of linealy parsable graph grammars, Acta Informatica10 (1978), 175-201.

[2] N.Go, T.Yaku et al., Generation of the Hichart program diagrams, J.Inform.Proc, 15 (1992), 293-300

[3] Terrence W. Patt, Pair grammars, graph languages and string-to-graph translations, JCSS 5 (1971), 560-595.

[4] 田辺文夫，’階層型流れ図言語Hichart／Cに対するグラフ文法の研究’早稲田大学卒業論文(1990).

[5] 鵜林尚靖，木下恂，Cから木構造プログラム図式への変換について，情報処理学会第36回全国大会論文集 (1988), 921-922.

[6] 夜久竹夫，二木厚吉，足立暁生，守屋悦朗，番場浩，階層的流れ図言語Hichartの情報処理記号，早稲田大学情報科学教育センター紀要 3 (1986), 92-107.