

# 属性グラフ文法に基づくHichartからPascalへの トランスレータ

1 L-6

今木 孝哲<sup>†</sup>, 大井 裕一<sup>†</sup>, 安達 由洋<sup>††</sup>, 夜久 竹夫<sup>††</sup>  
<sup>†</sup>東洋大学, <sup>††</sup>日本大学

## 1. はじめに

視覚的プログラミング環境として、我々は階層的プログラム図式言語Hichartを用いてプログラム図を編集し、それに対応するPascalプログラムを生成する図的プログラミングツールを研究している。

本発表では、この図的プログラミングツールに不可欠であるHichartの内部表現からPascalプログラムを生成するトランスレータに初めて属性グラフ文法[2]を導入し、そのトランスレータについて報告する。このトランスレータは、Prologで記述され属性グラフ文法に基づいて構文エラーや宣言をチェックする機能も備えている。さらに、この手法を応用してPascalの構文チェック機能を持つHichartエディタを実現する。

## 2. Hichart/Pascal対応 属性グラフ文法[2]

図2. 1に、if文に対する属性グラフ文法の例を示す。属性パターン以下は、属性の意味を表している。また、図2. 2に実際の内部表現の例を示す。

図2. 2の内部表現は、以下に示すデータから構成されている。

hichart\_code: セルの情報  
 connect\_code: セル同士のつながり、階層構造の制御線  
 id: セルの識別番号  
 cell: セルの種類  
 location: セルの(左上の)座標  
 size: セルの大きさ  
 [Parent, Children, Previous, Next]: 他のセルとの接続関係  
 cellList: 文字リスト

Hichart to Pascal Translator Based on An Attribute Graph Grammar

Takanori Imaki<sup>†</sup>, Yuichi Oi<sup>†</sup>, Yoshihiro Adachi<sup>†</sup>,  
 Takeo Yaku<sup>††</sup>

<sup>†</sup> Faculty of Engineering, Toyo University

<sup>††</sup> College of Humanities and Sciences, Nihon University

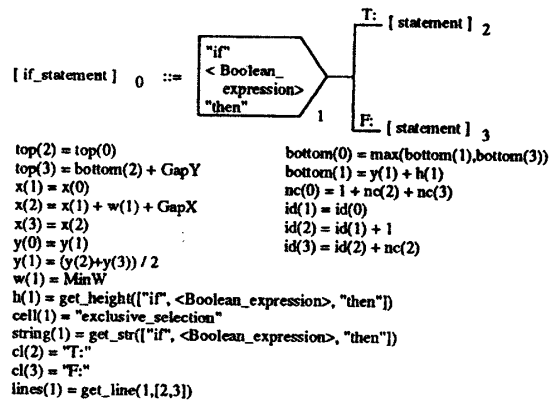


図2. 1 属性グラフ文法の例

```

hichart_code(id(hanoi, 0), cell(2), location(50, 182),
size(100, 50), [id(hanoi, (/)), [id(hanoi, 1), id(hanoi, 3),
id(hanoi, 15)], id(hanoi, (/)), id(hanoi, (/))],
cellList([string([program, hanoi]))])
    
```

```

connect_code(id(hanoi, 15), id(hanoi, 16), [[330, 527],
[370, 527], [370, 527], [410, 527]]).
    
```

図2. 2 内部表現の例

## 3. HichartからPascalへのトランスレータ

本トランスレータは、Pascal対応Hichart図表示の属性グラフ文法に基づいた内部表現から、Pascalのソースプログラムを生成する。

このトランスレータは、はじめに識別番号0（ルート）のセルの文字リストの解析と変換を行い、そのセルの親子関係や上位関係をたどって階層的に解析と変換を行う。図2. 1のif文は図3. 1で解析、変換される。

このトランスレータは、hichart\_codeに含まれるセルの識別番号、他のセルとの接続関係や文字リストによって構文解析を行う。図2. 1の属性グラフ文法に基づいた右辺のパターンをプログラム図全体のグラフの中からパターンマッチを行い、一致すれ

```

h2pas_conditional(Indent, ID, [indent(Indent), Label, 'if',
Boolean, ' then', 'Yn', Then, 'Yn', indent(Indent), else, 'Yn',
Else]):-
    hichart_code(ID, _ _ _ [_ [True, False|_]. _ _ _],
    cellList(CellList)),
    member(string(String), CellList),
/* if文の抽出 */
    get_boolean_of_ifThen(String, Boolean),
    get_label_from_cellList(CellList, Label),
    Indent1 is Indent + 1,
/* then文の抽出 */
    h2pas_conditional_then(Indent1, True, Then),
/* else文の抽出 */
    h2pas_conditional_else(Indent1, False, Else).

```

図 3. 1 トランスレータの変換部分 (一部)

ば対応するPascalプログラムの文字列を生成する。もし、パターンに一致するものがなければ、構文上のエラーとなり、そのセルの識別番号を返す。

図 3. 2 で、プログラム図を作成する画面と変換されたソースプログラムの表示画面を示す。

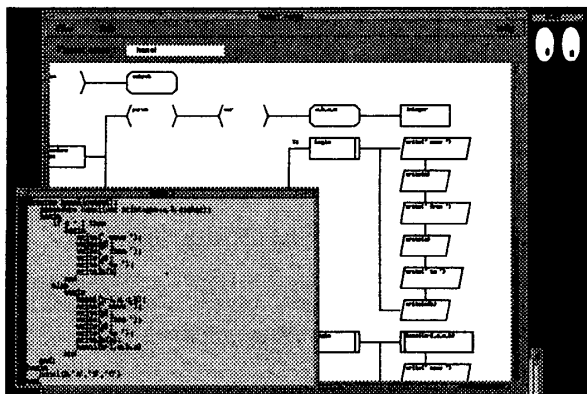


図 3. 2 実行例：入力と出力

トランスレータは、同時に変数や関数が二重に宣言されていたり、未宣言の変数や関数を用いているかをチェックしている。構文上や宣言上でエラーが生じた場合、そのセルをハイライトする事によって、ユーザに注意を与える。図 3. 3 は、エラーが生じたときの例である。

## 5. おわりに

我々は、属性グラフ文法に基づいたプログラム図の内部表現からPascalのソースプログラムへのトランスレータをはじめて実現した。このトランスレー

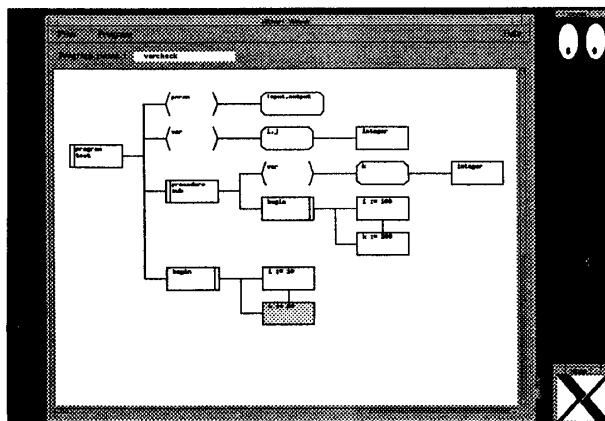


図 3. 3 エラー画面

タは、プログラム図から汎用プログラム言語へ変換するとき不可欠である。なお、トランスレータはIF/Prologで記述されSUNOS 4.1.3及びSolaris 2.4上で稼働している。

今後は、C, Fortranやデータ交換言語DXL[3]に対応するトランスレータを開発する。

## 参考文献

- [1]夜久, 他: 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研究教育センター紀要, Vol. 3, pp. 92-107(1986)
- [2]大井, 安達, 夜久: PascalからHichartへのトランスレータの属性グラフ文法による記述とPrologによる実現, 電子情報通信学会技術研究報告, Vol. 94, No. 525, pp. 89-96(1995)
- [3]長野, 他: 木構造図用CASEツール間のデータ交換言語: DXL, 情報処理, Vol. 35, No. 4, pp. 341-346(Apr. 1994).
- [4]久保, 他: 属性グラフ文法に基づいたHichartエディタ, 情報処理学会第51回全国大会, 1L-2, (1995)
- [5]Teitelbaum, T. and Reps, T.: The Cornell Program Synthesizer: A Syntax-Directed Programming Environment, Comm. ACM, Vol. 24, pp. 563-573(1981)

## 付録. エディタ

一般に、構文指向型エディタ(e.g. [4])は、incrementalに、構文を解析しプログラム図を作成する。一方、我々は、はじめにエディタ画面上に自由にセルを配置し、制御線をつなげて、プログラム図を作成する形式のエディタを実現した。このエディタは、プログラム図を一括して構文解析し属性グラフ文法に組み込まれている描画条件[2]に基づき再描画する機能を持っている。