

Prolog プログラムの属性グラフ文法に基づいた可視化

2D-1

今木 孝哲[†], 安達 由洋[†]
[†] 東洋大学

1 はじめに

Prolog は、ユニフィケーションやバックトラックなどのメカニズムによりプログラムを簡潔にかつ柔軟に記述できる強力なプログラミング言語である。しかしながら、C や PASCAL などの手続き型言語にはないこれらの機能が、初心者にとって Prolog を学習することを難しくしている。また、現在の Prolog 処理系が備えているデバッグツールは一般にボックスモデルに基づくテキストベースのトレース機能が主で、プログラムの全体構造の把握や実行過程の解析を支援する道具として機能が不十分である。

そこで我々は、Prolog プログラムを木構造図として表示し、このプログラム図上で実行過程を可視化するための研究を行っている。

ところで、プログラム図表示システムを実現するとき、そのプログラム言語を定義した文法に基づいて任意の正しいプログラムが表示できること（完全性）と表示したプログラム図が正しいこと（健全性）を保証できることが望ましい。また、プログラム図をディスプレイ上に表示するためには図のレイアウトが不可欠である [1]。これらの観点から、まず Prolog の構文規則をもとにプログラム図の生成を形式的に定義するプロダクションとレイアウト情報を属性として計算する意味規則からなる Prolog プログラム図の属性グラフ文法を定義した。

次に、この文法に基づいてユーザが指定したゴールに対する Prolog プログラム図を X window 上に描画する Prolog 可視化システムを実現した。このシステムはプログラムの実行過程をリアルタイムにプログラム図上に表示する機能も持っており、プログラムの動作の解析やデバッグを容易にする。

2 Prolog プログラム図の属性グラフ文法

Prolog プログラム図を生成する属性グラフ文法の継承属性および合成属性の一部を Fig.1 に、終端アルファベットおよび非終端アルファベットの一部を Fig.2 に示す。また、属性グラフ文法のプロダクションと意味規則の抜粋を Fig.3, Fig4 に示す。ただし、プロダクション

継承属性

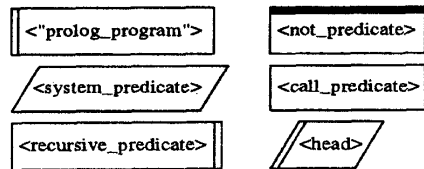
x,y : 配置されるセルの座標
 RootX : ルート(一番左上の)セルのX座標
 RootY : ルート(一番左上の)セルのY座標
 MinW,MinH : セルの最小幅, 最小の高さ
 GapX,GapY : セルとセルの間隔
 id : セルの識別番号

合成属性

w,h : セルの幅, 高さ
 string : セル内部の文字列
 subwidth : 指定されたセルを根とする部分木の幅
 nc : セルの数
 line : セルとセルを結ぶ制御線

Fig. 1. 属性一覧

終端アルファベット



非終端アルファベット

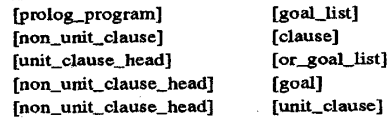


Fig. 2. 生成規則のアルファベット

および意味規則中の記号、関数は以下のようである。

- 開始記号 = [prolog-program]
- "●" : 辺の形を整えるためのダミー-接点
- get_str(X) : セル内の文字列を求める関数
- get_width(X) : 文字列のリストからセルの幅を求める関数
- get_line(S,E) : 始点セル S, 終点セル E を結ぶ線の始点と終点の座標を求める関数

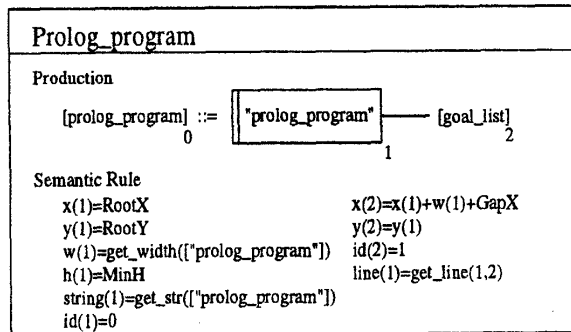


Fig. 3. 開始記号の書き換え規則

Prolog Visualization Based on Attribute Graph Grammar
 Takanori Imaki[†], Yoshihiro Adachi[†]
[†] Faculty of Engineering, Toyo University

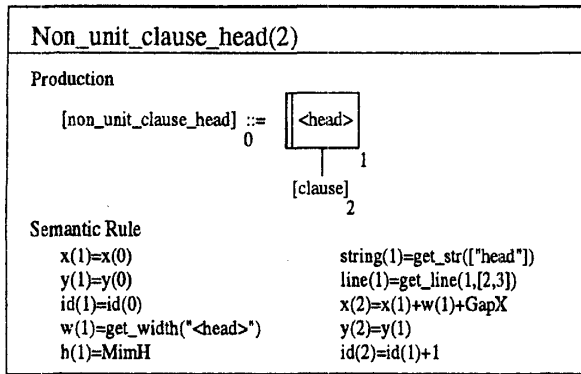


Fig. 4. 頭部の書き換え規則

```

prolog_cell(id(File,N),text(L),
  link(Parent,Upper,Children,Lower),Type,Graphic).
connect_orLine(Cell1,Cell2,X1,Y1,X2,Y2).
connect_andLine(Cell1,Cell2,X1,Y1,X2,Y2).
connect_neckLine(Cell1,Cell2,X1,Y1,X2,Y2).
connect_callLine(Cell1,Cell2,X1,Y1,X2,Y2).
id : セルの識別番号
text : ソースプログラムの行番号
link : 他のセルとの接続関係
Type : セルの種類
Graphic : セル内の文字列やセルの座標
Cell1,Cell2 : 制御線の始点セルと終点セル
X1,X2,Y1,Y2 : 制御線の始点と終点の座標
    
```

Fig. 5. 内部表現の一部

3 属性グラフ文法に基づく可視化システム

文法に基づいて Prolog からプログラム図の内部表現に変換するトランスレータと、内部表現から対応するプログラム図を描画するビューアで Prolog 可視化システムは構成されている。

3.1 トランスレータ

トランスレータは、Prolog プログラムを入力とし指定したゴールに対応する内部表現を生成する。その処理は以下の手順で行われている。

1. 入力されたソースプログラムを一行ごとに抽出して、行番号を付加する。これにより、プログラム図のセルとソースプログラムを対応させる。
2. Prolog インタープリタで字句解析と構文解析をする。ここで、コメント文は、削除される。
3. 指定したゴールに対して文法に基づき属性評価を行い内部表現を生成する。

上記の手順で生成される内部表現を Fig.5 に示す。

トランスレータで内部表現を生成すると同時に、トレースに必要な Prolog プログラムをメタプログラミング技法を用いて生成している。生成されるメタプログラムの構成を Fig.6 に示す。

3.2 ビューア

トランスレータで生成した内部表現を入力し、プログラム図を描画する。Prolog プログラムの AND/OR の

```

[prolog_program] :- prolog2D_startClause(Head,ID1,Parent),
  [goal_list],
  prolog2D_endClause(Head,ID1).
[goal_list] :- prolog2D_callGoal(Goal,ID2),
  [goal],
  prolog2D_exitGoal(Goal,ID2).
[goal_list] :- prolog2D_callGoal(Goal,ID2),
  [goal_list],
  prolog2D_exitGoal(Goal,ID2).
    
```

Fig. 6. メタプログラム

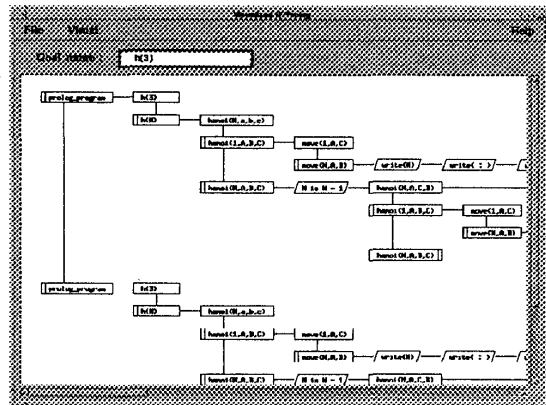


Fig. 7. Prolog の図表示

木構造をプログラム図では、X 軸方向を AND, Y 軸方向を OR に対応させて表示している。表示画面を Fig.7 に示す。

● トレース機能

メタプログラムにより生成したトレース用プログラムを実行して、ゴールの実行に対応するセルの色を変化させる。これによりユーザはリアルタイムに実行過程を認識することができる。

● プログラム図とソースプログラムとのリンク

ソースプログラムを Popup Window 上に表示し、セルあるいはソースプログラムの 1 行をマウスでクリックすることで、セルと対応する行がハイライトされる。

4 おわりに

Prolog プログラムを図表示するための属性グラフ文法を定式化し、それに基づいた Prolog 視覚化システムを実現した。本システムは、IF/Prolog Ver5.0A と OSF/Motif で記述されており、SUN OS 4.1.3 及び Solaris 2.4 上で稼働している。

今後は、プログラムの動的な変化や変数への代入過程を表示する機能などを付加してより高度な Prolog プログラミング支援環境へと発展させる。

参考文献

[1] 安齊, 他: 属性グラフ文法に基づいた構文指向型 Hichart エディタ, 第 37 回プログラミング・シンポジウム報告書, pp.97-108(1996).