

# プログラム作成支援システムπの構造エディタ

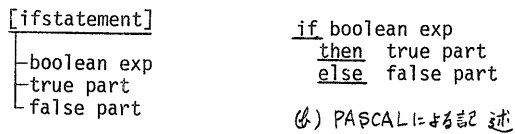
車谷博之 中本幸一 荻原兼一 荒木俊郎 都倉信樹  
( 大 阪 大 学 基 礎 工 学 部 )

## 1. まえがき

πシステムは、以下のような特徴を持つ、対話型のプログラム作成支援システムである。

- (1) アルゴリズムを構成する種々の概念にできるだけ近い形(これをフレームという)を用いてのプログラミングが可能である。
- (2) プログラムテキストばかりでなく、文書情報も統一的に扱う。
- (3) プログラムの「部分品」を利用したプログラム作りを促進する。
- (4) (2)の情報を一つの木構造(プログラム木、略して、P-木という)で表現して、データベース化している。この木構造を対象とした各種ユーティリティプログラム(文書化プログラム、代書プログラムなど)がある。
- (5) (1)の各種概念を記述するフレームは、追加可能であり、どのようなフレームを用いるとアルゴリズムが記述しやすいかを更迭することができる。

本論文は、πシステムにおいて、ユーザとの接点となるエディタ(πエディタ)について述べる。



(a) if文のフレーム

図1. フレームの例

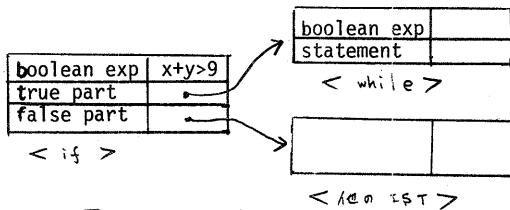


図2 部分P-木

## 2. プログラム木

フレームの例として、if文のフレームを図1に挙げる。このフレームはboolean exp, true part, false part から構成されている。

フレームはプログラムの記述を行なうための鋳型で、それ自身木構造をしており、各々の節点をフィールド(field)という。フィールドには名前(フィールド名、図1では boolean exp など)が、付けられている。このフレームに  $x+y>9$  といった文字列データを記入する。このように、フレームにデータを記入したものをインスタンス(instance, 略してIST)という。また、フィールドに対応するデータを項目(item)という。ISTもまた項目を節点とする木構造である。ここで、if文のIST(if文のフレームのISTのこと、以下このようにいう)のtrue partに他のIST(while loop)を結びつける。(図2) if文のtrue part から見るとwhile loopのISTをより詳細化したものである。πシステムでは、このように、ISTを節点とする木構造でプログラムを表現する。この木がP-木である。

## 3. πエディタ

πエディタは、P-木の作成・編集を担当し、木構造を扱う構造エディタである。πエディタは、ユーザとの接点であり、快適に、P-木が作成できなければならぬ。そのために、果すべき役割を次に挙げる。

- (1) P-木の内容をユーザに分りやすくCRTに表示すること。
- (2) プログラムの各部分の入力もれなどをチェックするなど、各種のエラーチェックを行なうこと。
- (3) ライブラリ中のプログラムの「部分品」

の検索を助け、ユーザプログラムとして使えること。

(4)エディット中に、クロスリファレンス表示など、プログラム作成に有用な情報を表示すること。

現時点では $\pi$ システムのライブラリは未設計であり、本論文では、(3)については述べない。

### 3.1 ISTの作成・編集の使用感

フレームは、フィールド名など、項目の説明のための情報を持つ。そこでISTを作成し、編集する使用感は、人間がエンピツと消しゴムを使って、事務書類や届けなどの書類を書くときの使用感と同様にする。この書類の書式の空欄へ順次記入するような感覚で、プログラムをIST単位で入力する。この使用感にするため、次の要求が生じる。

①ユーザがある項目をアクセスしているときに他の関連ある項目もCRT上に写っていること。

②いわゆるスクリーン・エディタの使用感にし、自由にデータを追加、消去、書き込みができること。

### 3.2 P-木の表示について

P-木全体を表示しようとするると巨大な画面が必要になる。そこで、CRTのように比較的限られた画面で、P-木の情報を充分に知るためには、必要に応じて、

(1)P-木全体の骨組

(2)P-木の部分木

(3)この部分木の周りの環境

という情報を得ることができればよい。

上述の見え方を直観的に示した図が、

図3である。

## 3.3 $\pi$ エディタの外部仕様

### 3.3.1 アクセス機能

項目を指すポインタを用意する。ユーザはポインタが指す項目をアクセスすることができる。このポインタを動かすことによってP-木のすべてをたどることができる。以下、ポインタの指す項目を対象項目、対象項目の属すISTを対象ISTという。

### 3.3.2 表示について

現在、使用しているCRTは、80文字×24行しか出力できない。そのためwindow(図3)の投影対象の範囲を拡大すると出力される情報は、より粗くなる。また、プログラムの入力は、IST単位である。

以上の点を考慮して、windowの投影対象の範囲のレベル(表示レベル)を次の4つの段階に設定した。レベル3から、レベル0に移るに従って投影対象の範囲が拡大されていく。以下の4つのレベルについて、各々、表示対象がCRT画面に入りきらないときは、ある部分を基点として、画面枠に入るだけ、表示する。

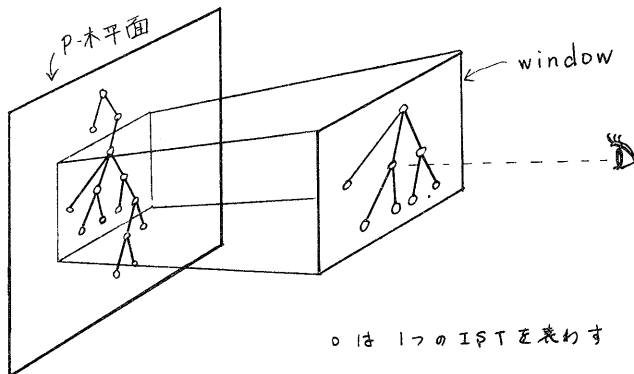


図3 ユーザからのP-木の見え方

<説明>

- (1)ユーザはwindowを通して、P-木を見る。
- (2)windowはP-木の部分木を投影するための画面である。
- (3)投影対象の範囲は移動、拡大、縮小できる。

〈レベル3〉 対象ISTの情報を対象項目を基点として、表示する。表示内容はフィールド名、項目、フィールドの説明などである。(図4)

〈レベル2〉 表示内容はフィールド名と項目であり、レベル3から説明文を省いたのが、レベル2である。(図5)

〈レベル1〉 対象ISTとその子孫のIST(どの子孫が表示されるかは、個々のフレームによって決まる。)を、対象項目を基点にして、表示する。1つのISTについて表示する内容は、レベル2と同じであるが、子孫のISTの情報を表示する分だけ、対象ISTを表示するスペースがなくなる。これは、対象ISTの子孫がすでに入力されている場合、その子孫を見ながら、対象ISTに入力する際使用する。(図6)

〈レベル0〉 ISTのフレーム名とこのIST中から、πシステムで指定した情報(例えば、手続きのISTであれば、手続き名など、これをISTガイドという)を節点とし、対象ISTを中心として、ISTの階層関係を木構造(レベル0木という)で表示する。階層の範囲は対象ISTの2~3世代上の先祖から、2~3世代下の子孫までである。これは対象ISTの周辺のP-木の骨組を知るために用いる。(図7)

以上の内、レベル0を除く他の表示レベルでは、ポインタを動かし、対象項目の修正などができる。

```

program {PASCALCOMP}
  module {Input}
    procedure {Insymbol}
  module {IDtable}
    procedure {EnterID}
    procedure {SearchID}
  
```

図7 <表示レベル0>の例。

program, module, procedure はフレーム名, {...}はISTガイド。programのISTはmoduleのISTを2つ子供に持ち、moduleのISTはそれぞれ、1,2のprocedureのISTを子供に持つ。

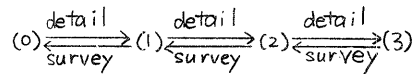
### 3.3.3 πエディタのコマンド

コマンドの多くは、操作性をよくするために、CRT端末のファンクションキーの1ストロークで実行される。他のコマンド(searchなど)はπエディタと会話しながら、コマンドを入力する。

#### 表示レベルの変更のコマンド

表示レベルを変更すると自動的にそのレベルでの表示が行なわれるので、表示せよ、というコマンドはない。

表示レベルを変更するには次の2種のコマンド、surveyとdetailによって、次のように表示レベルは遷移する。



[eachelement]

```

- element (id) t
- set (id) S
- process (ptr[statement])--operationcall
  
```

図4 <表示レベル3>の例

(id)は識別子が書き込めることを表わす。(ptr[statement])は、if文などの文に当るISTをつなぐことが可能であることを表わす。

[eachelement]

```

- element t
- set S
- process --operationcall
  
```

図5 <表示レベル2>の例

[eachelement]

```

- element t
- set S
- process --[operationcall]
  - variable t
  - operation initialize
  - parameter
  
```

図6 <表示レベル1>の例

この例は、集合Sの各要素オについて、オに属すオペレーションinitializeを実行するプログラムである。

## P-木の作成・編集のコマンド

### (1) ポインタの移動

表示レベル1,2,3については, IST内を移動するコマンド(木構造に従ったコマンド, 親や子供への移動)及びIST間を移動するコマンドがある。表示レベル0については, IST間を移動するコマンドがある。

### (2) ISTを生成するコマンド

これは, ユーザによって指定されたフレーム(フレーム名を指定)のISTを生成し, 対象項目につなぐコマンドである。

### (3) repeatコマンド

ある種のフィールドには, 可変個の項目を書き込むことができる。このコマンドが実行されると, 対象項目と同じ項目が, 1つ新たに生成される。

### (4) 検索コマンド(search)

検索対象はフレーム名または項目中に現われる文字列であり, 検索範囲は対象ISTを根とする部分P-木, または対象ISTの先祖である。

### (5) 項目の入力・編集

CRTのカーソルを用いたスクリーンエディタ方式である。

#### (コマンド例)

- ① 文字キー: カーソルの指す文字の前にこの文字キーの文字を挿入。
- ② del: カーソルの指す文字を消去。
- ③ カーソルを動かすコマンド(→↓←↑)

## 3.3.4 P-木情報サービス機能

エディット中に, クロスリファレンス表など, プログラム作成上有用な情報が得られれば便利である。以下にコマンドを挙げる。

- (1) 対象項目につなぐことのできるフレームとその説明文を表示。
- (2) 指定した識別子(identifier)が定義された環境を表示。
- (3) 対象ISTで参照可能な識別子を表示。(πシステムでは, ISTを単位としたALGOL流のブロック構造に従ったスコープルールがある。)

(4) 指定した変数(抽象データ型の変数<sup>(1)</sup>)のオペレーションを表示。

## 3.3.5 エラーチェック機能

πエディタのプログラム入力時のチェックは, IST単位のローカルなエラーチェックであり, ユーザのコマンド指令に関係なく, 自動的に行なわれる。チェックには式(expression)の構文のチェックなどがある。

なお, グローバルなチェックはチェックプログラム<sup>(1)</sup>が行ない, このユーティリティを呼び出すコマンドが, πエディタに用意されている。

## 3.4 インプリメント

CRTに表示すべき主な情報は, P-木とフレームセッド<sup>(1)</sup>にあり, ディスクにファイルとして格納されている。

各表示レベルに応じて, 主記憶上に表示すべき情報のデータ構造(IDSという)を用意した。表示レベル1,2,3のIDSはP-木及びISTと同じ構造であり, 表示レベル0のIDSはレベル0木(3.3.2参照)である。

IDSを持った理由を挙げる。

- (1) 表示すべき情報だけを持つことにより, πエディタが作成しやすくなる。
- (2) 表示する情報は人間が見るものであり, どのように出力されるかは比較的大きな自由度が要求される。また, CRTが変更される可能性があり, IDSとCRTとの対応の仕方を変更可能なようにする必要がある。IDSは表示すべき情報の論理的な構造である。

πエディタは当研究室のミニコンピュータNOVA3(主記憶96K語)上で, 現在インプリメント中である。

なお, 本研究は一部文部省科研費の援助を受けた。

## 参考文献

- (1) 岡根, 住田, 萩原, 荒木, 都倉: “あるプログラム作成支援システム(πシステム)の設計”, 信学特報, AL80-39(1980-10)。