数理解析研究所講究録
第593巻 1986年 168-182

ワークステーション上でのマイクロプログラムＰｒｏｌｏｇインタプリタの構成について

名大工学部　堀切　和典(Kazunori Horikiri)

名大工学部　吉田　雄二(Yuuji Yoshida)

名大工学部　福村　晃夫(Teruo Fukumura)

１．はじめに

我々はＰｒｏｌｏｇ処理系の効率を高めるために処理系で基本となるユニフィケーション，バックトラキングなどの操作をマイクロプログラムで記述したＰｒｏｌｏｇインタプリタを高性能ワークステーションＰＥＲＱ上に実現し、その評価を行った。Ｐｒｏｌｏｇ処理系をビットマップディスプレイを有するワークステーション上に実現したことにより、マルチウィンドウ等による高品質のマンマシン・インターフェイスを提供できるので本処理系は個人向きの知識情報処理システムを構成する上で極めて強力なツールとなりうる。

２．ワークステーションＰＥＲＱ

ここではワークステーションＰＥＲＱ（ＰＥＲＱ１Ａ）についてその概要を説明する。

２．１ＣＰＵ

マイクロプログラム制御によるＣＰＵのデータバス幅は１６bit，アドレスバス幅は２０bitである。マイクロ命令は１語４８bitの水平型で１マイクロサイクルは１７０ナノ秒で１６Ｋ語の書き替え可能な制御記憶（Ｗ．Ｃ．Ｓ．）を備えている。基本的にはＰＥＲＱはＵＣＳＤ－Ｐａｓｃａｌの Ｐ－Ｃｏｄｅに類似したＰａｓｃａｌ向きに設計された中間コード（Ｑ－Ｃｏｄｅ）をマイクロプログラムで直接解釈実行するＰａｓｃａｌマシンである。また、このＱ－Ｃｏｄｅの実行速度は約１ＭＩＰＳとなっている。

２．２主記憶と補助記憶

主記憶はアクセスタイムが６８０ナノ秒（４マイクロサイクル）のＲＡＭを用いて構成され１MByteの容量を有する。補助記憶としては２４MByteのウインチェスターディスクが用意されている。

２．３オペレーティングシステム

オペレーティングシステムはＰＯＳ（Ｐｅｒｑ Ｏｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍ）と呼ばれ，Ｐａｓｃａｌで記述されている。ＰＯＳはシングルユーザ，マルチプロセスのシステムであり仮想記憶（仮想記憶空間は４GByte）、マルチウインドウ等をサポートしており、ファイルシステムは階層化ディレクトリを持っている。またＯＳの機能はＰａｓｃａｌ

プログラムから呼び出すことが可能である。

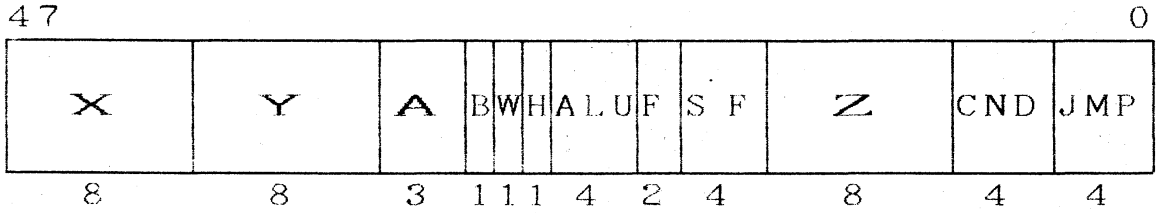
２．４ディスプレイとポインティングデバイス

７６８×１０２４ドットの高解像度縦型モノクロディスプレイとポインティングデバイスとしてのデジタイザが用意されており、質の高いユーザインタフェースを与えている。

２．５マイクロ命令

ＰＥＲＱのＣＰＵはデータバス幅が１６bitであることから１６bitマシンであるといえるが、アドレスバス幅は２０bitであり１MByteの記憶空間を扱うことができる。この２０bitのアドレスを計算するためにＣＰＵ内部のレジスタファイル，ＡＬＵなどのエレメントは２０bitの幅を持っている。

ＰＥＲＱのマイクロ命令は１語４８bitの水平型であり各語は図１に示すような１２個のフィールドより構成されている。それぞれのフィールドは以下に示すような意味を持っている。



（数字はビット幅を表す。）

図１．マイクロ命令の構成

X：ＸＹレジスタのＸポートのアドレスまたはＸＹレジスタに書き込みを行う場合のアドレスを示す。

Ｙ：ＸＹレジスタのＹポートのアドレスまたは定数の下位バイトを指定する。

Ａ：ＡＬＵのＡポートの入力を選択する。

Ｂ：ＡＬＵのＢポートの入力を選択する。

Ｗ：ＸＹレジスタに書き込みを行うか否かを指定する。

Ｈ：ＩＯデバイスがメモリーをアクセスするのを禁止する。

ＡＬＵ：ＡＬＵが実行するファンクションを指定する。

Ｆ：ＳＦ及びＺフィールドの機能を指定する。

ＳＦ：特殊機能，メモリー制御，ロングジャンプを指定する。

Ｚ：定数の上位バイト，ショートジャンプ，シフタの制御を指定する。

ＣＮＤ：条件付きジャンプの条件を指定する。

ＪＭＰ：ＡＭＤ２９１０に与える命令を指定する。

３．Ｐｒｏｌｏｇ処理系の設計

ここでは実現したＰｒｏｌｏｇ処理系の、言語仕様，処理系の構成及びＰｒｏｌｏｇマシンのアーキテクチャについて述べる。なお、以下ではここで構成するＰｒｏｌｏｇ処理系及び言語をＰＲＱＬＯＧ（Ｐｅｒｑ　Ｐｒｏｌｏｇ）と呼ぶ。

３．１言語仕様

ＰＲＱＬＯＧの文法は、現在のＰｒｏｌｏｇの標準となっているＤｅｃ１０－Ｐｒｏｌｏｇに準拠している。すなわち

ホーン節は以下のように記述される。

（Ｆａｃｔ）　　ｈ．

（Ａｘｉｏｍ）　ｈ：－ｂ$_1$，．．．．，ｂ$_n$．

（Ｄｉｒｅｃｔｉｖｅ）　？－ｑ$_1$，．．．．，ｑ$_n$．

例．

```
append([],X,X).
append([A|D],X,[A|Y]):-
  append(D,X,Y).
?-append([a,b,c],[d,e],X).
```

３．２処理系の構成

ＰＥＲＱはＰａｓｃａｌマシンであり、そのＯＳもＰａｓｃａｌで記述されている。従って処理系がＯＳの機能（メモリーアロケーション，Ｉ／Ｏハンドリング等）を利用するためにはＰａｓｃａｌマシンとＰｒｏｌｏｇマシンを共存させる必要がある。このため、システムはＰａｓｃａｌで記述されている部分とマイクロプログラムで記述されている部分から構成されている。この両者はお互いにコールし合いながらＰｒｏｌｏｇプログラムを実行する。

Ｐａｓｃａｌで記述されている部分はシステム・スーパバイザ及びマイクロプログラム・レベルでは記述が極めて煩雑となる入出力を扱う部分である。システム・スーパバイザは

マイクロプログラムのロード，呼び出し、メモリーの割り当て等を行う。入力ルーチンは項（Ｐｒｏｌｏｇプログラム等）を読み込み、構文解析を行いマイクロプログラムルーチンのための内部表現に変換し、出力ルーチンはこれと逆に内部表現を文字列に変換し出力する。

マイクロプログラムで記述されている部分はＰｒｏｌｏｇインタプリタで主要となる部分であり、ユニフィケーション，バックトラッキングなどのアルゴリズムが主体となる特に高速性が要求される部分である。図２にこの考え方に基づいたＰＲＱＬＯＧの処理系の構成を示す。この図における上下関係は依存関係を示す。またＰＥＲＱのレジスタおよびＷ．Ｃ．Ｓ．の割り当てを図３に示す。

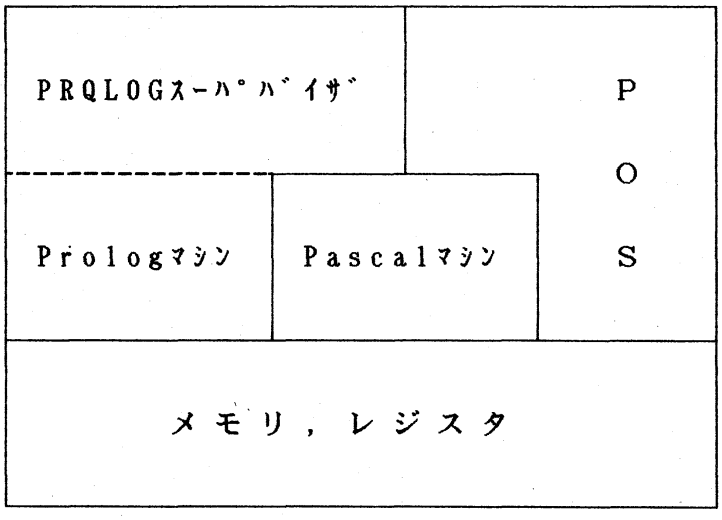


図２．ＰＲＱＬＯＧ処理系の構成

図３に示されているようにＰＲＱＬＯＧ処理系（Ｐｒｏｌ

ｏｇマシン）のための領域はＰａｓｃａｌマシンの領域とは独立に確保されている。

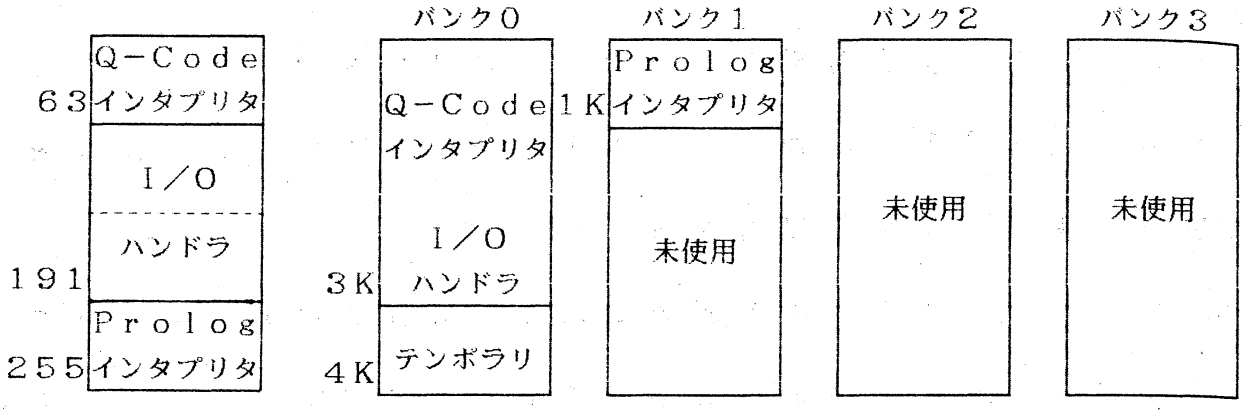


図３．レジスタおよびＷ．Ｃ．Ｓ．の割り当て

３．３Ｐｒｏｌｏｇマシンのアーキテクチャ

Ｐｒｏｌｏｇマシンは以下の７つの要素により構成されている。

(i)項領域：節の内部表現の基本単位である項を格納する。

(ii)シンボルテーブル：シンボルを管理する。

(iii)述語テーブル：述語の登録を管理する。

(iv)エンバイロメントスタック：証明手続き中の環境を保存する。

(v)ユニフィケーションスタック：構造体の再帰的なユニファイを実現する。

(vi)トレールスタック：バックトラックの際に変数の束縛を解放するのに用いる。

(vii)レジスタ群：テーブル，スタック等へのポインタを格納

する。

(i)～(vii)を実現するために様々なデータ構造が用いられる。次にこれらのデータ構造について述べる。

ＰＥＲＱは１６bitマシンであるのでデータの内部表現も１６bitを基本単位としている。またＰＥＲＱの仮想記憶はセグメンテーション方式であり１つのデータセグメントの大きさは最大でも１６bitでアドレス可能な大きさ（１２８KByte）である。このため内部表現で用いるアドレス（ポインタ）はセグメント内でのオフセットアドレスを用い１６bitで表現することとする。

（１）項の内部表現

節の内部表現における基本単位は項と呼ばれるものである。項は図４に示すような３２bitの項記述子によって表現される。

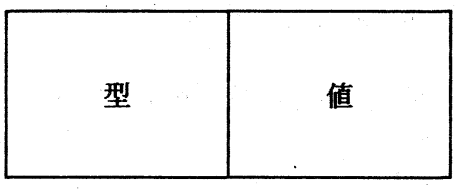


図４．項記述子

項記述子は１６bit×２語で構成され、第１語は型フィールドでその項の種類を示し、整数，シンボル，変数，構造体に分類する。第２語は値フィールドで項の値を示す。

各々の型に対する値フィールドの内容は以下のようである。

整数　　：値は整数値の２の補数表現。

シンボル：シンボルを表すシンボル記述子のアドレス。この

アドレスはシンボルの識別子として用いられる。すなわち、シンボルは識別子が等しい時またその時に限って同一であると見なされる。シンボル記述子については後述する。

変数　：節中でその変数が何番目に現れた変数であるかを示す番号。例として、

ａ（Ｘ）：－ｂ（Ｚ，Ｘ），ｃ（Ｙ）．　という節を考えると、変数Ｘには１，Ｙには３，Ｚには２という番号が付けられる。

構造体　：構造体を表す構造体記述子のアドレス。構造体記述子については次に述べる。

（２）構造体記述子

構造体記述子は構造体を表現するのに用いられる。構造体は構造体名と１つ以上の引数を持つ。構造体記述子は引数の数及び構造体名を表すヘッダと引数を表す項記述子からなる。引数の数は１６bitで表され、構造体名はシンボル記述子のアドレスとして表される。また各引数を表す項記述子がヘッダの次に格納される。

（３）シンボルテーブル

システム内に存在する全てのシンボルは全てそれに対応するシンボル記述子により識別される。シンボル記述子はオー

プンハッシュ法によりハッシュテーブルとして構成されているシンボルテーブルにより管理される。

シンボル記述子は次の５つのフィールドを持つ。

(i) シンボルを表す文字列の長さ：８bitの整数。

(ii) シンボルの持つ属性：属性を８bit表現したものであり、属性にはPrefix, Infix, Suffix, Othersの４つがある。

(iii)述語記述子のアドレス：もしそのシンボルを用いて述語が構成されていれば述語記述子のアドレスを１６bitで格納する。

(iv) ハッシュリンクポインタ：同じハッシュ値を持つシンボル記述子に対するリンクポインタ。

(v) シンボルを表す文字列：シンボルを表す文字列を８bitのＡＳＣＩＩコード列で表したもの。

（４）述語テーブル

述語テーブルは、節のヘッドリテラルに用いられているシンボルをキーとして構成された述語の表である。シンボルが与えられるとそれに対応するシンボル記述子を用いて述語記述子のアドレスを知ることができる。図５に述語テーブル及び述語記述子の構成を示す。

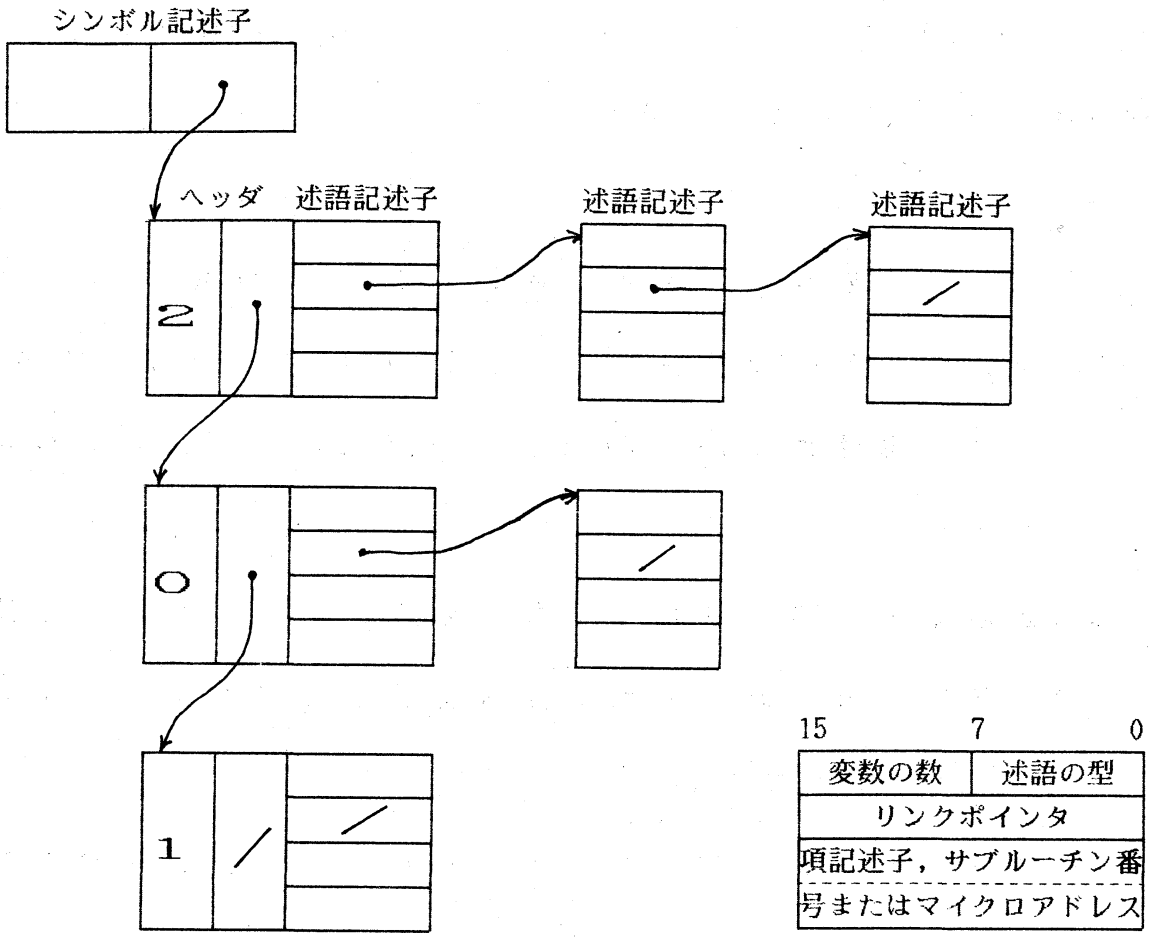


図５．述語テーブル及び述語記述子の構成

述語テーブルは引数の数による分類を示すヘッダと述語記述子により構成されており、述語記述子は次の５つのフィールドから構成されている。

(i)変数の数：節中に変数がいくつあるかを示す。

(ii)述語の型：その述語がＰｒｏｌｏｇ，Ｐａｓｃａｌ，マイクロプログラムのうちどれで記述されているかを示す。

(iii)リンクポインタ：同一のシンボル，引数を持つ述語記述

子へのポインタ。

(iv)項記述子，サブルーチン番号またはマイクロアドレス：その述語がＰｒｏｌｏｇで実現されている場合には項記述子が、Ｐａｓｃａｌの場合にはサブルーチン番号が、マイクロプログラムの場合にはマイクロアドレスが格納される。

図５の述語テーブルの状態はあるシンボルを用いて２引数の述語が３つ，０引数の述語が２つ，１引数の述語が１つ登録されている状態を示している。

（５）エンバイロメントスタック

述語テーブルに登録された節（Ｆａｃｔ，Ｄirective）が証明のために用いられると環境がエンバイロメントスタック上に作られる。環境には次の６つの情報が含まれる。

(i)ゴール：その節を呼び出した項記述子のポインタ。

(ii)親の環境：その節を呼び出した節の持つ環境。

(iii)トレールポインタ：その節の呼び出しの時点のトレールスタックポインタの値。

(iv)バックトラックポイント：ゴールが失敗した場合にバックトラックする環境のポインタ。

(v)次候補節集合：その節の適用に失敗した場合に用いる節のリスト。

(vi)バインドリスト：変数のバインドを表すセル。

以上のように原始項と変数の束縛からなるインスタンスの表現にはストラクチャシェアリング方式を用いた。

４．処理系の動作

処理系に入力された文字列は初めにトークンと呼ばれる単位に切り分けられ先読みのためのキューに置かれる。パーザはこのキューを参照してトップダウンパージングを行い、項記述子を項領域上に作る。入力された節がDirectiveの場合にはＰａｓｃａｌマシンがＰｒｏｌｏｇマシンを呼び出す。Ｐｒｏｌｏｇマシンは項領域，シンボルテーブル，述語テーブルを参照し、適用した節に対して環境をエンバイロメントスタックにプッシュしていく。構造体のユニィファイの際には再帰的ユニファイの実現のためにユニフィケーションスタックが用いられる。束縛の起こった変数セルのアドレスはトレールスタックにプッシュされバックトラックが起こった場合にはこのスタックを参照して束縛を解放する。Ｐａｓｃａｌで実現された述語の呼び出しに対してはＰｒｏｌｏｇマシンが逆にＰａｓｃａｌマシンを呼び出す。ゴールが全て成功した場合には制御がＰｒｏｌｏｇマシンからＰａｓｃａｌマシンへ戻りトップレベルの変数の値を出力して入力待ちになる。

5．処理系の評価

Ｐｒｏｌｏｇ処理系は一般に次のような点から評価される。

(i)性能に関するもの

処理速度，ユーザーが使用可能なメモリー領域

処理速度の評価についてはＷａｒｒｅｎらの用いた３０要素のリストのナイーブリバースを用いた。この実行には約０．１秒を要し、これは約５ＫＬＩＰＳに相当している。また、ユーザーが使用可能なメモリー領域は項領域，述語テーブル等のヒープ領域が約１２８KByte、エンバイロメントスタックのスッタク領域が１２８KByteの合計２５６KByteである。

(ii)言語機能に関するもの

シンタックス，グラフィック機能，コンパイラ

本処理系のシンタックスはＤｅｃ１０－Ｐｒｏｌｏｇに準拠しており、グラフィック機能については将来ビットマップディスプレイを活かした述語を取り入れる予定である。またコンパイラについてはＰｒｏｌｏｇ向きに設計された中間言語をオブジェクトとするコンパイラをＰｒｏｌｏｇで記述しその中間言語をマイクロプログラムで直接解釈実行する方式を計画している。

(iii)プログラム開発環境に関するもの

エディタ，デバッガ

現在のところエディタはＰＥＲＱのＯＳのユーティリティーであるスクリーンエディタを用いている。このエディタはＰｒｏｌｏｇの外から呼び出すかたちになっている。またデバッガはない。

(iv)応用

他言語とのインターフェイス

他言語とのインターフェイスはＰａｓｃａｌとのインターフェイスを取ることが可能である。

６．あとがき

今後の課題としては、組み込み述語の整備，効率的なバックトラックを可能にするなどの処理系の最適化，５．で述べた方式によるコンパイラの作成が考えられる。