6D-3

制限付きＧＨＣの逐次型処理系

佐藤 健
(株) 富士通研究所

1. はじめに

筆者は並列推論マシンの研究を行っている。その一環として第５世代コンピュータ計画の中で検討されている並列論理型言語ＧＨＣ〔1〕のサブセットについて，逐次型処理系を作成した。

本文では，試作した逐次型処理系ならびにその内部構成について述べる。

2. ＧＨＣに対する制限

ＧＨＣを逐次マシン上で効率よくインプリメントするために以下のような制限をした。これはFlatＧＨＣとよばれているものとほぼ同じである。

(1) ガード内には，組み込み述語しか書けない。

(2) 同じ手続き（述語名が同じ節の集まり）内の各節のPASSIVE 部は並列に試されず，上から下へ順に試される。

(3) 節内のPASSIVE 部では，まずヘッドのUNIFICATIONが実行され，次にガード内が左から右へ順に実行される。

(4) ガード部での変数どうしのEQUALITY CHECKは，両方の値が決まるまで，実行されない。

3. プロセスの生成方式

ＧＨＣのプロセスの生成に関して以下の方式を採用した。

(1) プロセス領域確保の最適化〔2〕

ＧＨＣの実行はプロセスが新たなプロセスにおきかわって行われる。ACTIVE部において新たなプロセスが生成されたときに，もしそれがそのACTIVE部で生成される最初のプロセスであれば，元のプロセスの領域を使うようにした。これにより，最初のプロセスを生成するときには，新たな領域をとる必要がなくなった。

(2) BREADTH-FIRST SCHEDULING

実行可能リストの先頭のプロセスから順に実行を行いプロセスが新しく生成された場合に，実行可能リストの最後尾に加えられる方式である。新たに生成されたプロセスは少なくとも一回はかならず実行される。

4. プロセスの再起動方式

ＧＨＣではプロセスがSUSPEND することがありうる。このSUSPEND したプロセスを再起動するために以下の方式を採用した。

(1) NON-BUSY WAITING方式〔2〕

SUSPEND したプロセスを SUSPENDの原因になった変数のSUSPEND リストにつなげ，変数がINSTANTIATE されたときにプロセスを再起動するようにした。

(2) SUSPEND した節の最初から再開する方式

あるプロセスがある節でSUSPEND した場合，その節に関するSUSPEND プロセスを作り，変数のSUSPEND リストにつなげる。変数がINSTANTIATE されたときにそのプロセスを再起動することにより対応する節の先頭から実行を再開するようにした。

これにより変数がINSTANTIATE されたときにその変数が原因でSUSPEND した節に対応するプロセスのみが実行されるようになり，手続きの先頭から再開する方式に比べ，重複する実行部分が少なくなった。

5. 処理系の構造

処理系は以下の領域からなっている。

(1) データベース

USER定義の述語を格納しておく領域である。
処理の高速化のために同じ述語名を持つ節を手続きとみなし，節どうしをポインタで結んでいる。

(2) ヒープ領域

実行中に作られる構造体を蓄える領域である。
ACTIVE部のUNIFICATION で構造体のコピー方式を採用しているので，新たに構造体が作られたときや新たに変数が生成したときにこの領域に蓄えられる。

(3) 引数ベクトル領域

プロセスの環境を蓄える領域である。プロセスを手続き呼び出しをしているものとみなし，その引数を環境として蓄えることにした。この領域の生成，消滅を容易にするため，引数の最大数をきめて各プロセスの環境の大きさを固定し，それらをポインタでつないで管理するようにした。

A Sequential Implementation of GHC subset
Ken SATOH
Fujitsu Labs. Ltd.

(4) ローカル領域

PASSIVE 部で起こるUNIFICATION の結果を一時的に蓄える領域である。PASSIVE 部ではUNIFICATION はすべて参照となるのでPASSIVE 部の実行の間ここに参照結果を蓄えるようにした。

(5) プロセス領域

処理の基本単位であるプロセスの制御情報（プロセスディスクリプタ）を蓄える領域である。図1に構造を示す。図で，兄弟プロセスとは，同じ手続き呼び出しによってSUSPEND したプロセスのことである。

この領域もディスクリプタごとにポインタでつなげて管理し，生成，消滅を容易にしてある。

6. UNIFICATION の方式

本処理系では，〔3〕と同様にＧＨＣの複雑なUNIFICATIONの機構は不必要で，以下のようになる。

(1) PASSIVE 部のUNIFICATION

メモリ参照比較の操作にプロセスのSUSPEND の操作が加わったもの。

(2) ACTIVE部のUNIFICATION

通常のUNIFICATION にプロセスの再起動の操作が加わったもの。

7. 実行例

以下のＧＨＣプログラムを実行したときの，ｐのプロセス実行直後の状態を図2に示す。

```
①  p(a,Y) :- true | true.
②  p(X,b) :- true | true.
③  q(X,Y) :- true | Y=b.
?- p(X,Y),q(X,Y).
```

図においてｑのプロセスのＰフィールドのPRCD③は上の③で始まる手続きを実行することを表す。

ｐのプロセスのＰフィールドのCLS ①，②は，それぞれ①，②の節を実行することを表す。

ｐのプロセスは実行されると①，②の節に対してそれぞれSUSPEND するので，それぞれの節に対してSUSPEND プロセスを生成し，SUSPEND の原因となった変数とともにＶフィールドにより循環リストを形成し，また兄弟プロセスどうしでＢフィールドにより循環リストを形成する。（図2参照）

次にｑのプロセスが実行され，Ｙにｂが INSTANTIATE されるので，Ｙで待つｐのプロセスが再起動され，②の節のみが再実行され，TRUST が起きたときにＸで待つｐのプロセスが取り除かれる。

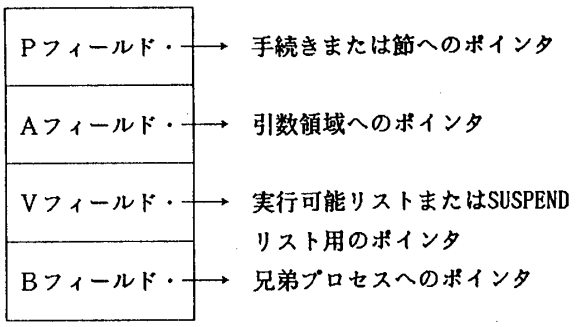

図1　プロセスディスクリプタの構造

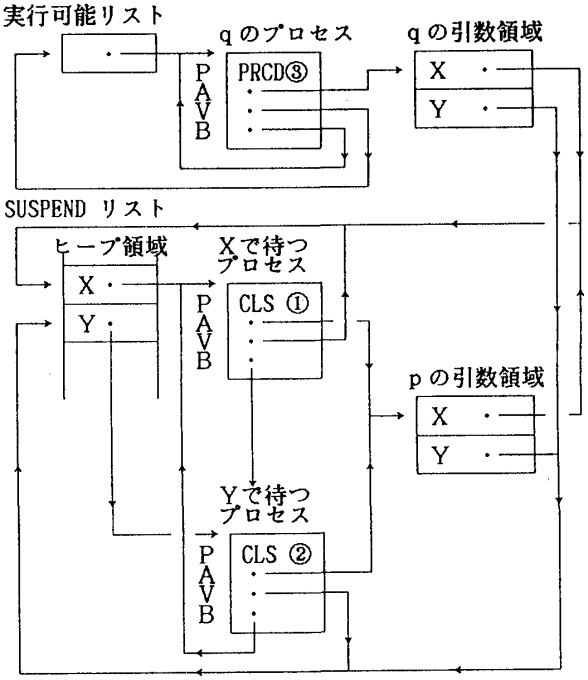

図2　ｐのプロセス実行直後の状態

8. おわりに

以上逐次型処理系の内部構成について述べた。本処理系はSUN-WORKSTATION 上のＣ言語で書かれた３６００行のプログラムとなった。

最後に，本文を書くにあたり御討論いただいた当研究部関係各位に感謝致します。

〔参考文献〕

〔1〕上田，Guarded Horn Clauses
ICOT Technical Report TR-103

〔2〕Foster,I., Gregory,S., Ringwood,G., Satoh,K.,
A Sequential Implementation of PARLOG
3rd Internatinal Conference on Logic
Programming (1986)

〔3〕宮崎，瀧　Multi-PSI におけるFlat GHCの実現方式
The Logic Programming Conference'86