TR-405

無限プロセスを含む
並列論理型プログラムの宣言的意味論

村上　昌己

June, 1988

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

無限プロセスを含む並列論理型プログラムの宣言的意味論

村上 昌己

(財) 新世代コンピュータ技術開発機構研究所

June 10, 1988

Abstract

flat GHC のような Horn 論理に基づく並列言語の宣言的意味論について述べる．本論文では通常の純 Horn 論理型プログラムの宣言的セマンティクスにおける Herbrand 基底に代わって、入出力履歴 (I/O history:I/O 履歴) の領域を導入し，プログラムの表示 (denotation) は、I/O 履歴の集合により与えられる．ゴール節及びコミット・オペレーターを含む Horn 節の集合について真／偽の概念を導入し、プログラムの意味はその節の集合を真とする I/O 履歴の集合として定義される．さらにこのように定義されたセマンティクスが、プログラムの構文から定まる関数の最大不動点によって特徴づけられることを示す．このアプローチによりガード／コミット機構によって制御されながら無限に計算を続けるプロセスを含むプログラムの性質の議論が可能となる．

1 はじめに

近年、Horn 論理に基づく並列プログラミング言語として PARLOG [Clark 86]、 Concurrent Prolog [Shapiro 86], GHC [Ueda 88] などが提案されている．この言語では、入出力用のストリームを用いて通信を行いながら無限に計算を続けるプロセスを自然に表現することができる．このようなプログラム言語のセマンティクスについての形式的な議論の方法については、いくつかの結果が報告されている．[Takeuchi 86, Saraswat 85, 87, Ueda 86, Shibayama 87] これらの結果は、いずれも操作的アプローチに基づいたものである．ゆえにこれらの結果はどちらかといえば、処理系の形式的な仕様といった趣のものである．実際プログラムの検証、変換などの手法の論理的な基礎としては何らかの宣言的な手法によるセマンティクスが与えられることが求められる．

純 Horn 論理型プログラミング言語の場合、[Apt 82, Lloyd 84] らによる宣言的なセマンティクスが既に報告されている．このアプローチではプログラムのセマンティクスはそのプログラムを記述する節集合の最小モデル，すなわちプログラム自身と”等価”な単位節の集合によって与えられる．また，このような単位節の集合はプログラム節の集合から定まる関数の最小不動点によって与えられる．このアプローチによってプログラムの実行メカニズムとは独立に，プログラムの論理的帰結として導かれるものを解の集合として特徴づけることができる．したがって，このようなアプローチは論理プログラムの明解さを生かすにはもっとも適したアプローチといえる．また領域を完備 Herbrand 空間に拡張することにより，有限時間で停止しないような計算の定式化への拡張なども試みられてる．[Lloyd 84, Sakakibara 85] しかしながらこれら結果は純 Horn 論理型プログラムについてのものであり，このままではガード／コミット機構をもつような並列プログラミング言語に適用することはできない．すなわち，ガード／コミット機構を用いてのゴールの反駁において，（コミット記号を論理積とみなした場合）解としてえられるものは節集合の論理的帰結になってはいるが，論理的帰結となるものがすべて得られる可能性があるわけではない．すなわちコミット記号を含む節集合のセマンティクスとしては，最小モデルはその手続き的な解集合に比べて大き過ぎる．また [Falaschi 87] で指摘された通り，全て基底原子式で特徴づけるという方法も適切ではない．したがって，プログラムの計算結果を論理的帰結として議論するためには，flat GHC の計算規則のようなコミット記号を含む節を用いたゴールの反駁が証明手続きとしてある意味で完全かつ無矛盾とするようなモデル理論を再構築することが必要となる．そこでこのアプローチの並列言語への拡張が [Levi 87 88, Falaschi 88] 等で試みられた．[Levi 88] では flat GHC プログラムのセマンティクスを，GHC における単位節にあたるガード付原子式の集合によって議論する方法が報告されている．ガード付原子式とは例えば次のような形をしたガード付節である．

$$p(X_1,\ldots,X_n,Y_1,\ldots,Y_m) : -X_1 = \tau_1,\ldots,X_n = \tau_n | Y_1 = \theta_1,\ldots,Y_m = \theta_m.$$

この意味は直観的には，GHC プログラムのガード付節としての意味に等しい．すなわち，

$p(\tau_1, \ldots, \tau_n, Y_1, \ldots, Y_m)$という形をしたゴールが実行されると，$Y_1 = \theta_1, \ldots, Y_m = \theta_m$という解代入が得られる事を意味する．これによって従来の純 Horn 論理型プログラムで用いられた宣言的意味論がコミット機構を含むプログラムの変数を含むゴールについての動作を議論することが可能になった．例えば次のようなプログラムについて：

```
p(X, Y) :- X = [X1|Y1] | q(X1, Y1, Y).
q(X, Y, Z) :- true | Z = [X|X1], r(Y, X1).
r(X, Y) :- X = [X1|Y1] | Y = [X1].
s(X, Y) :- X = [X1|Y1] | Y = [Yb|Y2], h(Yb).
h(Y) :- true | Y = b.
t(X, Y) :- true | p([A|X], Y), h1(A), s(Y, X)
h1(A) :- true | A = a.
```

このプログラムの denotation のうち，q, p, tに関するものは次のようなものとなる．

```
q(X, Y, Z) :- Y = [X1|Y1] | Z = [X|X1].
p(X, Y) :- X = [X1, X2|Y1] | Y = [X1|X2].
t(X, Y) :- true | Y = [a|b], X = [b|Y1]
```

しかしながら，ここではガード付原子式はゴールが成功した時に得られる出力代入と入力代入の関係を記述しているのみである．[Takeuchi 86] に報告されているように，このような関係のみを用いて入出力関係だけでは動作を記述することが困難であり，計算の中間結果をも用いることが必要となる例が存在する．例えば，この方法では依然として次のような二つのプログラムp1, p2を区別することはできない．

```
p1 :
p(X, Y) :- X = [A|X] | Y = [A|Z1], pp(X1, Z1)
pp(X, Y) :- X = [A|X1] | Y = [A|N], nil(N).

p2 :
p(X, Y) :- X = [A|X1] | p2(X1, Y, A).
p2(X, Y, A) :- X = [B|X1] | Y = [A|Y1], p22(B, Y1).
p22(B1, Y1) :- true | Y1 = [B1|N], nil(N).
nil(N) :- true | N = nil.
```

この二つのプログラムは，pについてガード付原子式を用いて denotation を与える限り同じものである．しかし，両者は出力を具体化するタイミングが異なる．このようなプログラムが非決定的なプログラムと並列に実行された場合に，どちらのプログラムを用いるかによって全体の動作が異なるものになる可能性があることは，[Brock, 81, Takeuchi 86] 等によって指摘されている．このように flat GHC プログラムの動作を十分に記述するためにはプログラムの実行途中の振舞について必要にして十分な情報を記述しうるような記法を用いたセマンティクスが必要となる．さらに，無限に計算が続くプロセスの動作を議論するのは，ゴールが成功したときの得られる代入によって議論するのは不可能である．無限に計算が続くプロセスを含むようなプログラム（例えば素数をつぎつぎとあるストリームに出力するようなプログラム）の場合には，実行によって外部からなんらかの計算結果が観測できるのは，正に計算の途中の振舞からであり，この意味においてはプログラムの実行途中の（外部から観測できる）振舞を記述できるセマンティクスこそが望むべきものである．

本論文では，flat GHC 風の言語で記述された無限プロセスを含むようなプログラムに対して宣言的な，すなわち純 Horn 節におけるモデル論的セマンティクスの拡張としてのセマンティクスを与える．すなわち通常の単位節の概念に代わって、入出力履歴 (input /output history: I/O 履歴) の概念を導入する．I/O 履歴の集合は Herbrand 基底に対応する．またI/O 履歴はガード付原子式の拡張である．すなわち，I/O 履歴はあるプログラムがデッドロックも失敗もせずに計算が進んだときにたどる path （または木）を外側から観測したものになっている．ここではさらにゴールまたはプログラムが (宣言的に) 真であるということをI/O 履歴の概念をもとに再定義する．プログラムのセマンティクスはそのプログラムを真とするI/O 履歴の最大の集合として与えられる．ここで導入するゴール節の

真／偽の概念は，即反駁の成功／不成功には対応しない．例えば，無限に実行を続けるプロセスを起動するゴール節は正常な動作を続けるものであるかぎり真とされる．また始めからゴールが十分に具体化されて起動されるのではなく，実行途中に外部から（例えば他のプロセス）から入力を受け取って実行を継続するようなプロセスを表わすゴールは，途中でサスペンドするものであっても，その原因が入力待ちであるような場合には真とみなされる． さらにここではこのように定義されたプログラムのdenotationが，プログラムの構文から定まる関数の最大不動点によって特徴付けられることがしめされる．

2 ガード付ストリーム

本節では，ガード付ストリームについて述べる．まず準備として幾つかの基本的な定義を示す．本稿では議論を簡単にするために，{a, b} という領域の上のリストを扱うプログラムに対象を限定する．

[定義 1]

*Var*を可算無限個の変数の集合，*Fun*= {a, b, nil, cons} を関数記号の集合とする．*Fun*の各要素に対して arity をa, b, nil/0, cons/2 のように定める．

[定義 2]

Terms を次のように定まる項の集合とする。

(1) τ が変数または a, b, nil の場合、$\tau \in Terms$.

(2) $\tau_1, \tau_2 \in Terms$ のとき、$\mathtt{cons}(\tau_1, \tau_2) \in Terms$.

[定義 3]

項τが単純であるとは，τが変数項であるか、a, b, nilのいずれかであるか，$\mathtt{cons}(X, Y)$ の形をしていてX, Yが異なる変数である場合をいう．

[定義 4]

写像$\sigma : Var \rightarrow Term$ が次の条件を充たすとき代入と呼ばれる．

$$\Sigma = \{X | \sigma X \neq X, X \in \mathrm{Var}\}$$

とするとき，Σは有限集合．

ここで，代入の定義域をVarからTermへ拡張する．

[定義 5]

$\tau \in Terms$について、$\sigma\tau$ は以下のように定義される．

$$\sigma\tau = \begin{cases} \sigma X & \text{if } \tau \text{ is } X \in \mathrm{Var} \\ \tau & \text{if } \tau \in \{\mathtt{a},\ \mathtt{b},\ \mathtt{nil}\} \\ \mathtt{cons}(\sigma\tau_1, \sigma\tau_2) & \text{if } \tau = \mathtt{cons}(\tau_1, \tau_2), \tau_1, \tau_2 \in Terms \end{cases}$$

[定義 6]

$X \in Var$, τを単純な項とするとき，次のような式を単純な代入式 (又は、単に代入式) という．

$$X = \tau$$

特に$X = X$という式は*true*と表記する．単純な代入式の有限集合を用いて代入を表現することができる．しかしながら一般に単純な代入式の有限集合が常に代入を定めるわけではない．

以下，代入式の集合とそれが定める代入とを同一視する．また以下では慣習にしたがい$\mathtt{cons}(X, Y)$ は$[X|Y]$ で，nil は [ ]で表わす．

[定義 7]

σ を単純な代入式の集合とする．σ が代入であるか又は、ある代入 θ に対して以下のように定義される$\bigcup_{k=\infty} \theta_k$ に等しいとき、σ をω−代入とよぶ．

$$\begin{aligned} &\theta_0 = \theta \\ &\theta_{k+1} = \theta_k \cup \{X = \tau | \text{ある} (Y = \tau') \in \theta_k \text{ が存在し、} X \text{は} \tau' \text{に出現し、} (X = \tau'') \notin \theta_k \text{かつ、} \\ &\qquad\qquad \tau \text{に出現するすべての変数について} \theta_k \text{のいかなる元にも出現しない．} \} \end{aligned}$$

ω−代入は項から無限項への写像を定義する．

[定義 8]
Vを変数の集合，Σを写像σから [定義 4] の通りに定まる集合とする．$\Sigma \subset V$ であるとき、σはVに制限されているという．また、$\Sigma \cap V = \phi$ のとき、σはVについて不変であるという．

本稿で提案する I/O 履歴は，純 Horn 論理型プログラムの単位節の概念にあたるものであり，[Levi 88] のガード付き原子式の概念の拡張である．また，[Takeuchi 86] の derivation tree からプロセスの内部変数についての情報をとりのぞいたものとも考えられる．すなわち I/O 履歴はプロセスの呼ばれたときの形を表わすヘッドHとプロセスのある実行における入出力の関係を示すボディ部GUを用いて次のように表わされる．

$$H :- GU$$

ここでHは互いに異なる変数に述語記号を適用したもの，GUは単純な代入式の集合で表現された代入σとボディ部での単一化の実行を表す式U_bの対$< \sigma|U_b >$ の集合である．直観的にはHという形のゴールの引数がσによって具体化されるとU_b という単一化が行われることを意味する．たとえば次のプログラムについて，

```
p1(X,Y) :- X = [A|X1], A = a | Y = [B|Y1], B = b, p1(X1,Y1).
p1(X,Y) :- X = [B|X1], B = b | Y = [A|Y1], A = a, p1(X1,Y1).
```

次の I/O 履歴はp1の入力ストリームXにaが入力され (すなわち Xの第要素がaに具体化され)、出力ストリームYにbが出力され続いてbが入力されるとaが出力される様子を記述している．

$$\begin{aligned} \mathtt{p1(X,Y)} :- \{ & < \{\mathtt{X} = [\mathtt{A}|\mathtt{X1}], \mathtt{A} = \mathtt{a}\} | \mathtt{Y} = [\mathtt{B}|\mathtt{Y1}] >, < \{\mathtt{X} = [\mathtt{A}|\mathtt{X1}], \mathtt{A} = \mathtt{a}\} | \mathtt{B} = \mathtt{b} >, \\ & < \{\mathtt{X} = [\mathtt{A}|\mathtt{X1}], \mathtt{A} = \mathtt{a}, \mathtt{X1} = [\mathtt{B1}|\mathtt{X2}], \mathtt{B1} = \mathtt{b}\} | \mathtt{Y1} = [\mathtt{A1}|\mathtt{Y2}] >, \\ & < \{\mathtt{X} = [\mathtt{A}|\mathtt{X1}], \mathtt{A} = \mathtt{a}, \mathtt{X1} = [\mathtt{B1}|\mathtt{X2}], \mathtt{B1} = \mathtt{b}\} | \mathtt{A1} = \mathtt{a} >, \ldots \} \end{aligned}$$

Hの一つの I/O 履歴はプロセスHの可能な実行の一つを表わす．したがって，実行中に異なる節にコミットする可能性がある場合は，異なる I/O 履歴が存在する．また，同じ節にコミットした実行でも，並列に走っているプロセスと異なるスケジューリングの違いによって異なる I/O 履歴が存在する可能性がある．

I/O 履歴のボディ部分は flat GHC プログラムの正常な実行を表現している．したがって、GUは実行順序に関して半整列集合となっている．すなわち、$< \sigma_1|U_{b1} >, < \sigma_2|U_{b2} > \in GU$について、$\sigma_1 \subset \sigma_2$ ならば、U_{b1}はU_{b2}より先に実行可能となる．

さらにGUは GHC プログラムの正常な実行の性質に対応する特徴を幾つか備えている．本節の残りの部分ではこのような特徴を反映した集合，ガード付ストリームについて述べる．

[定義 9]
$X \in Var$, τを単純な項とするとき，次のような式を単純な判定式 (または単に判定式) という．

$$X? = \tau$$

[定義 10]
代入σ，変数X，単純な項τについて，$uni(X,\tau)$を代入式$X = \tau$ または半定式$X? = \tau$とするとき、$< \sigma|uni(X,\tau) >$をガード付代入とよぶ．このときσを$< \sigma|uni(X,\tau) >$のガード部，$uni(X,\tau)$を能動部とよぶ．

直観的には$uni(X,\tau)$が代入式であったときは実際にXを具体化する単一化を，判定式であった場合はテスト・ユニフィケーションをあらわす．

[定義 11]
$< \sigma|uni(X,\tau) >$をガード付代入とする．$|< \sigma|uni(X,\tau) >|$ は次のように定義される判定式又は代入式の集合である．

$$|< \sigma|U >| = \{U\} \cup \sigma$$

[定義 12]

与えられたガード付代入の集合GUの上に次のような関係$\prec$を定義する．$< \sigma_1|u_1 >, < \sigma_2|u_2 > \in GU$とする．$\sigma_1 \subset \sigma_2$ かつ $\sigma_1 \neq \sigma_2$ のとき、

$$< \sigma_1|u_1 > \prec < \sigma_2|u_2 >$$

とする．このように定義された関係$\prec$が半整列集合になることは容易に示せる．

[定義 13]

ガード付代入の集合GUが次の条件をみたすとき，GUをガード付ストリームとよぶ．

1) $< \sigma_1|U_1 >, < \sigma_2|U_2 > \in GU, < \sigma_1|U_1 > \neq < \sigma_2|U_2 >$ かつU_1とU_2が同じ左辺をもつならば、
少なくとも一方は判定式であり，右辺は互いに等しい．またU_1が代入式でU_2 が判定式であったとき

$$< \sigma_2|U_2 > \prec < \sigma_1|U_1 >$$

でない．

2) 任意の $< \theta|X = \tau > \in GU$ について、$< \sigma|U > \in GU$ ならば、$(X = \tau) \notin \sigma$．

3) 任意の $< \theta|X? = \tau >, < \sigma|U > \in GU$について、$\tau \neq \tau'$ ならば、$(X = \tau') \notin \sigma$．

4) 任意の $< \sigma_1|u_1 >, < \sigma_2|u_2 > \in GU$について、$(X = \tau) \in \sigma$ かつ $(X = \tau') \in \sigma'$ ならば、$\tau = \tau'$．

上の条件 1),;4) は、いずれも GHC における変数が論理的な変数であり、一度具体化された値が別の値になることがないことによる．

[例 1]

次のような集合GU_1, GU_2について，GU_1はZにaの列を次々と読みこんでb の列をYに出力するプロセスの，GU_2はXにa の列を，Yにbの列を読みこんで両者をマージし，Zに出力するプロセスの動作の 1 例をそれぞれあらわしている．

$$GU_1 = \{gu_{1i}(1 \leq i)\},$$

$gu_{11} =< \{\mathtt{Z} = [\mathtt{A}|\mathtt{Z1}], \mathtt{A} = \mathtt{a}\}|\mathtt{Y} = [\mathtt{B}|\mathtt{Y1}] >$
$gu_{12} =< \{\mathtt{Z} = [\mathtt{A}|\mathtt{Z1}], \mathtt{A} = \mathtt{a}\}|\mathtt{B} = \mathtt{b} >$
$gu_{13} =< \{\mathtt{Z} = [\mathtt{A}|\mathtt{Z1}], \mathtt{A} = \mathtt{a}, \mathtt{Z1} = [\mathtt{A1}|\mathtt{Z2}], \mathtt{A1} = \mathtt{a}\}|\mathtt{Y1} = [\mathtt{B1}|\mathtt{Y2}] >$
$gu_{14} =< \{\mathtt{Z} = [\mathtt{A}|\mathtt{Z1}], \mathtt{A} = \mathtt{a}, \mathtt{Z1} = [\mathtt{A1}|\mathtt{Z2}], \mathtt{A1} = \mathtt{a}\}|\mathtt{B1} = \mathtt{b} >$
.....

$$GU_2 = \{gu_{2j}(1 \leq j)\}$$

$gu_{21} =< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}\}|\mathtt{Z} = [\mathtt{A}|\mathtt{Z1}] >$
$gu_{22} =< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}\}|\mathtt{A} = \mathtt{a} >$
$gu_{23} =< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}, \mathtt{Y} = [\mathtt{B}|\mathtt{Y1}], \mathtt{B} = \mathtt{b}\}|\mathtt{Z1} = [\mathtt{A1}|\mathtt{Z2}] >$
$gu_{24} =< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}, \mathtt{Y} = [\mathtt{B}|\mathtt{Y1}], \mathtt{B} = \mathtt{b}\}|\mathtt{A1} = \mathtt{b} >$
$gu_{25} =< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}, \mathtt{Y} = [\mathtt{B}|\mathtt{Y1}], \mathtt{B} = \mathtt{b}, \mathtt{Y1} = [\mathtt{B1}|\mathtt{Y2}], \mathtt{B1} = \mathtt{b}\}|\mathtt{Z2} = [\mathtt{B2}|\mathtt{Z3}] >$
$gu_{26} =< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}, \mathtt{Y} = [\mathtt{B}|\mathtt{Y1}], \mathtt{B} = \mathtt{b}, \mathtt{Y1} = [\mathtt{B1}|\mathtt{Y2}], \mathtt{B1} = \mathtt{b}\}|\mathtt{B2} = \mathtt{b} >$
.....

□

次の概念は、並列に走る複数のゴールを含むようなゴール節について、各ゴールの動きを記述するガード付ストリームから、全体の動作を記述するガード付ストリームを得る操作を定義している．

[定義 14]

$GU_1, \ldots, GU_n$ をガード付きストリームとする．$Gu_k(1 \leq k)$ を次のように定義する．

$$Gu_1 = \{< \sigma|U > | \exists i, \exists < \sigma|U > \in GU_i, \forall (X = \tau) \in \sigma, \forall j, < \sigma'|X = \tau > \notin GU_j\}$$

$$Gu_{k+1} = Gu_k \cup \{< \sigma|U > | \exists i, \exists < \sigma'|U > \in GU_i, \forall (X = \tau) \in \sigma',$$

$$((\forall j, < \sigma''|X = \tau > \notin GU_j) \vee < \sigma''|X = \tau > \in Gu_k) \wedge$$
$$\sigma = (\sigma' - \{X = \tau | < \sigma''|X = \tau > \in Gu_k\}) \cup \{U|U \in \sigma'' < \sigma''|X = \tau > \in Gu_k\}$$

このとき、GU を以下のように定める.

$$GU = \bigcup_{k \to \infty} Gu_k$$

このGUがガード付ストリームになり、かつ

$$\{U| < \sigma|U > \in GU\} = \{U|\exists i < \sigma|U > \in GU_i\}$$

のとき、GUを $GU_1, \ldots, GU_n$の同期付マージとよび、

$$GU_1 \| \ldots \| GU_n.$$

であらわす.

$n = 1$ の場合同期付マージは常に定義でき結果はGU_1自身と等しくなる.

[定義 14] において、ある$< \sigma|U > \in GU_i$について $< \sigma|U > \in Gu_1$ であった場合、U は $GU_1 \| \ldots \| GU_n$ の外部から σ という入力を待ち、$GU_j (j \neq i)$ からは何も待たないことを意味する. また、$< \sigma|U > \in Gu_{k+1}$ であった場合、U は$GU_1 \| \ldots \| GU_n$ の外側からの入力に加えて、ある外側からσ''が与えられるのを待っていることが既に判っているU'が (すなわち$< \sigma''|U' > \in Gu_k) GU_j (j \neq i)$で実行されるのを待っていることを意味する。 この場合、U'が$GU_1 \| \ldots \| GU_n$の外側から待っている$U'' \in \sigma''$ を、Uも待っていることになる.

[例 2]

先の [例 1] に述べたガード付ストリームGU_1, GU_2 は同期付マージできない. すなわち,

$Gu_1 = \{gu_{21}, gu_{22}\}$

$Gu_2 = Gu_1 \cup \{< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}\}|\mathtt{Y} = [\mathtt{B}|\mathtt{Y1}] >, < \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}\}|\mathtt{B} = \mathtt{b} >\}$

$Gu_3 = Gu_2 \cup \{< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}\}|\mathtt{Z1} = [\mathtt{A1}|\mathtt{Z2}] >, < \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}\}|\mathtt{A1} = \mathtt{b} >\}$

$Gu_4 = Gu_3 \cup \{< \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}, \mathtt{A1} = \mathtt{a}\}|\mathtt{Y1} = [\mathtt{B1}|\mathtt{Y2}] >, < \{\mathtt{X} = [\mathtt{A0}|\mathtt{X1}], \mathtt{A0} = \mathtt{a}, \mathtt{A1} = \mathtt{a}\}|\mathtt{B1} = \mathtt{b} >\}$

.....

ここでA1 はGu_4のガード部でaに具体化されているにもかかわらず、Gu_3のボディ部でbに具体化されている. したがって、$\bigcup_{k \to \infty} Gu_k$ はガード付ストリームとはならない. したがって、GU_1とGU_2の同期付きマージは定義できない.

□

[例 3] Brock Ackermann の例題 次の例は [Takeuchi 86] の Brock Ackermann の例題を表わしている.

$GU_1 = \{gu_{11}, gu_{12}, gu_{13}, gu_{14}\}$

$gu_{11} =< \{\mathtt{In} = [\mathtt{A}|\mathtt{In1}], \mathtt{A} = \mathtt{a}, \mathtt{Mid} = [\mathtt{B}|\mathtt{M1}], \mathtt{B} = \mathtt{b}\}|\mathtt{Out} = [\mathtt{A1}|\mathtt{Out1}] >$
$gu_{12} =< \{\mathtt{In} = [\mathtt{A}|\mathtt{In1}], \mathtt{A} = \mathtt{a}, \mathtt{Mid} = [\mathtt{B}|\mathtt{M1}], \mathtt{B} = \mathtt{b}\}|\mathtt{A1} = \mathtt{a} >$
$gu_{13} =< \{\mathtt{In} = [\mathtt{A}|\mathtt{In1}], \mathtt{A} = \mathtt{a}, \mathtt{Mid} = [\mathtt{B}|\mathtt{M1}], \mathtt{B} = \mathtt{b}\}|\mathtt{Out1} = [\mathtt{B1}|\mathtt{Out2}] >$
$gu_{14} =< \{\mathtt{In} = [\mathtt{A}|\mathtt{In1}], \mathtt{A} = \mathtt{a}, \mathtt{Mid} = [\mathtt{B}|\mathtt{M1}], \mathtt{B} = \mathtt{b}\}|\mathtt{B1} = \mathtt{b} >$

$GU_2 = \{gu_{21}, gu_{22}\}$

$gu_{21} =< \{\mathtt{Out} = [\mathtt{A1}|\mathtt{Out1}], \mathtt{A1} = \mathtt{a}\}|\mathtt{Mid} = [\mathtt{B}|\mathtt{M1}] >$
$gu_{22} =< \{\mathtt{Out} = [\mathtt{A1}|\mathtt{Out1}], \mathtt{A1} = \mathtt{a}\}|\mathtt{B} = \mathtt{b} >$

この場合，$Gu_1 = \phi$であり、したがって$GU = \phi$. 一方，

$$U_2 = \{\texttt{Out = [A1|Out1], A1 = a, Out1 = [B1|Out2], B1 = b, Mid = [B|M1], B = b}\}$$

であり，ゆえにGU_1とGU_2の同期付マージは定義できない．しかし，GU_2は次のようなGU_1'とは同期付マージが定義できる．

$GU_1' = \{gu_{11}', gu_{12}', gu_{13}', gu_{14}'\}$

$gu_{11}' =< \{\texttt{In = [A|In1], A = a}\}|\texttt{Out = [A1|Out1]} >$
$gu_{12}' =< \{\texttt{In = [A|In1], A = a}\}|\texttt{A1 = a} >$
$gu_{13}' =< \{\texttt{In = [A|In1], A = a, Mid = [B|M1], B = b}\}|\texttt{Out1 = [B1|Out2]} >$
$gu_{14}' =< \{\texttt{In = [A|In1], A = a, Mid = [B|M1], B = b}\}|\texttt{B1 = b} >$

結果は次のようになる．$\sigma_{BA} = \{\texttt{In = [A|In1], A = a}\}$ とおくと:

$$GU_1'\|GU_2 = \{< \sigma_{BA}|\texttt{Out = [A1|Out1]} >, < \sigma_{BA}|\texttt{A1 = a} >, < \sigma_B A|\texttt{Mid = [B|M1]} >,$$
$$< \sigma_{BA}|\texttt{B = b} >, < \sigma_{BA}|\texttt{Out1 = [B1|Out2]} >, < \sigma_{BA}|\texttt{B1 = b} >\}$$

□

3 モデル論的セマンティクス

本節では先に述べたガード付ストリームの概念をもとに、純 Horn 論理型プログラムにおける Herbrand 基底，単位節等の概念に相当するものを並列論理型言語に対して導入する． この節では GHC の計算規則にそって実行される簡単な並列プログラミング言語を示す．この言語は議論を単純にするために flat GHC の簡単なサブセットを採用している．しかし，この制限が一般性を失うものではないことは容易に示される．本質的には [Levi 88] の strong normal form への変換アルゴリズムの拡張によって示される．

述語記号の集合を$Pred$で表わす．$Pred$の元にはそれぞれ arity が定められているものとする．$Pred$の元 (n-ary) を n 個の項に適用したものを原子式という．

[定義 15]

$H, B_1, B_2, \ldots, B_n$を原子式，かつHの引数部に出現するのは全て異なる変数，$U_{g1}, \ldots, U_{gm}, U_{b1}, \ldots, U_{bh}$を単純な代入式とする．このとき次の節：

$$H :- U_{g1}, \ldots, U_{gm} | U_{b1}, \ldots, U_{bh}, B_1, B_2, \ldots, B_n$$

をガード付節とよぶ．ガード付節の有限集合をプログラムとよぶ．

以下では，Hが$p(X_1, X_2, \ldots, X_k)$のとき $Var(H) = \{X_1, X_2, \ldots, X_k\}$とする．

[定義 16]

GUをガード付きストリームとする．擬 I/O 履歴tとは次のようなものである．

$$p(X_1, X_2, \ldots, X_k) :- GU$$

ここで$p \in Pred$ でアリティ k、かつ任意の$gu \in GU$について、$U \in |gu|$ならば、Uの左辺の変数はあるiについて以下のように定まる変数の集合 $V_i(GU)$に含まれる．

$V_0(GU) = Var(p(X_1, X_2, \ldots, X_k))$
$V_{i+1}(GU) = V_i(GU) \cup \{X | \exists gu \in GU, \exists uni(Y, \tau) \in |gu|, X$ は τ に出現し、かつ $Y \in V_i(GU)$ であり、
$\forall gu' \in GU, gu \prec\!< \sigma|U >$ ならば X は gu に出現しない。$\}$

ここで $p(X_1, X_2, \ldots, X_k)$ を t のヘッド部分，GUをボディ部分とよぶ．直観的にはGUにはヘッド部分から”見える”変数しか出現しない．したがって，プロセスの内部変数についての情報は棄てられている．すなわち，ボディ部のガード付ストリームが表わしているのは「このプロセスが何をするか」であり「このプロセスはどのように計算を

行うか」ではない．この意味では計算木 [Takeuchi 86] によるアプローチより，抽象化されたセマンティクスを定義している．

擬 I/O 履歴は並列言語における単位節の概念に相当する．しかしながら擬 I/O 履歴では本質的に同じ計算に対して複数の記法が可能となる．すなわちヘッドに出現する変数以外の変数については，いかなる変数が用いられようと外から観測する限りにおいては同じ計算を表現しているものとみなすことができる．そこで擬 I/O 履歴の領域に適当な同値関係を導入し、その同値類で Herbrand 基底にあたるものを定義する．

[定義 17]

写像$\sigma : Var \rightarrow Var$について、$\sigma'$が存在し、

$$\sigma\sigma' = \sigma'\sigma.$$

となるとき、σを呼び換え写像という．

GUをガード付ストリーム、σを呼び換え写像とする．σGUは次のように定まるガード付ストリームである．

$$\sigma GU = \{\sigma gu | gu \in GU\}$$

ここで、$gu =< \theta | uni(Y, \tau') >$のとき、

$$\sigma gu =< \sigma * \theta | uni(\sigma Y, \sigma\tau') >,$$

とする．$\sigma * \theta$は以下のように定まる代入である．

$$\sigma * \theta = \{\sigma X = \sigma\tau | X = \tau \in \theta\}$$

GUがガード付ストリームであるとき、σGUもガード付ストリームとなることは容易に示せる．

[定義 18]

擬 I/O 履歴 t_1, t_2 について、$t_1 : H :- GU_1, t_2 : H :- GU_2$とする．すなわち t_1, t_2 はおなじ原子式をヘッドにもち，かつ GU_1 に出現する変数の集合からGU_2 に出現する変数の集合への$Var(H)$について不変な呼び変え写像σが存在し$\sigma GU_1 = GU_2$となるとき

$$t_1 \approx t_2$$

とする．

関係≈が同値関係となることは容易に示せる．以下では、FUN, VAR, PRED から定まるすべての擬 I/O 履歴の≈による商集合を$I/O - hist$, その元を I/O 履歴とよぶ．以下では$I/O - hist$は Herbrand 基底に代わるものとしての役割を果たす．

[定義 19]

$I/O - hist$の任意の部分集合を解釈と呼ぶ．

[定義 20]

t を I/O 履歴、g をゴールとするとき、t がg のトレースであるとは、次の (1) (3) が成り立つことをいう．

(1) t のインスタンスは$H :- GU$という形をしており、あるω−代入σ が存在して、$\sigma H = g$となる．

(2) 任意の$< \theta | U >\in GU$について、代入式の集合σ'が存在し、$\sigma' \cup \theta = \sigma' \cup \sigma$ かつ $\sigma' \cup \theta (= \sigma' \cup \sigma)$ が ω−代入となる．

(3) 任意の$< \theta | U >\in GU$について、Uが代入式$X = \tau$の場合 σはXを具体化せず、U が判定式$X? = \tau$の場合 σXは $\sigma\tau$に等しい．

ここで写像σが変数Xを具体化しないとは、$\sigma X = Y (\in Var)$ かつ$\sigma Z = Y$ となるZがX以外に存在しないことをいう．

[定義 21]

Iを解釈、gをゴールとするとき、gがIで真であるとはgのトレースがI に含まれることをいう．

ゴール$g_1, \ldots, g_n$について、$t_1, \ldots, t_n$をそれぞれのトレースとする．もしこれらが、$g_1, \ldots, g_n$というゴール節を実行したときの各ゴールの動作をあらわしているならば、各t_i, t_jに共通に出現する変数は、i, jで同じ値をもつはずであり、また同じ変数の値の部分項として出現するはずである．次に定義する概念はこの条件を定式化している．

[定義 22]

$t_1,\ldots,t_n$を I/O 履歴とする. $t_1,\ldots,t_n$が変数互換であるとは、いかなるi,j $(1 \leq i,j \leq n)$についても、 t_iとt_jに共通に出現する変数の集合は、次に定義される変数の変数の集合 $Common(t_i,t_j)$ と等しいことをいう.

$$
\begin{aligned}
&COM_0(t_i,t_j) = Var(H_i) \cap Var(H_j)\\
&COM_{k+1}(t_i,t_j) = COM_k(t_i,t_j) \cup\\
&\qquad \{X \mid \exists Y \in COM_k(t_i,t_j), \exists \tau : \tau \text{ は } X,[X|Z], \text{ または } [Z|X] \text{ の形であり},\\
&\qquad\quad \exists gu_i \in GU_i, ((Y=\tau) \in |gu_i| \vee (Y?=\tau) \in |gu_i|) \wedge\\
&\qquad\quad \forall gu_i' \in GU_i, gu_i' \prec gu_i \text{ ならば} X \text{ は、いかなる} U \in |gu_i'| \text{にも現れない.} \wedge\\
&\qquad\quad \exists gu_j \in GU_j, ((Y=\tau) \in |gu_j| \vee (Y?=\tau) \in |gu_j|) \wedge\\
&\qquad\quad \forall gu_j' \in GU_j, gu_j' \prec gu_j \text{ ならば} X \text{ は、いかなる} U \in |gu_j'| \text{にも現れない.} \}
\end{aligned}
$$

$$Common(t_i,t_j) = \bigcup_{k-\infty} COM_k$$

ここでH_mはt_mのヘッド部、GU_mはt_mのボディ部.

$n=1$の場合、明らかにt_1は変数互換である.

[定義 23]

Iを解釈、$g_1,\ldots,g_n$をゴール節とするとき、$g_1,\ldots,g_n$ が Iで真であるとは、各$i(1 \leq i \leq n)$のトレースt_iがIに含まれ、$t_1,\ldots,t_n$の変数互換なインスタンスのボディ部分の同期付マージが定義できることである. また空(すなわち$n=0$ の場合) なゴール節は常に真であるとする.

この定義が well defined であるためには、$g_1,\ldots,g_n$のそれぞれのトレースのインスタンスのボディ部分の同期付マージが定義できるか否かが、インスタンスの選び方に依存しないことである. これは次の命題より容易に示すことができる.

[命題 1]

ガード付ストリーム$GU_1,\ldots,GU_n$と呼び換え写像σについて、 $GU_1,\ldots,GU_n$の同期付マージが定義できるならば $\sigma GU_1,\ldots,\sigma GU_n$についても同期付マージが定義できる.

証明は同期付マージの定義より straight forward である.

[定義 24]

GUをガード付ストリーム、Vを変数の有限集合とする. ここでGUのVによる制限$GU \downarrow V$とは次のような集合である.

$$GU \downarrow V = \{< \sigma|uni(X,\tau) > \mid \exists k < \sigma|uni(X,\tau) > \in GU, X \in V_k\}$$

ここで、

$$
\begin{aligned}
&V_0 = V\\
&V_{i+1} = V_i \cup \{X \mid \exists gu \in GU, \exists uni(Y,\tau) \in gu, X \text{は} \tau \text{に含まれ} Y \in V_i, \text{かつ}\\
&\qquad\qquad \forall gu' \in GU, gu' \prec\!< \sigma|U > \text{ならば} X \text{は} gu' \text{に現れない.} \}
\end{aligned}
$$

GUがガード付ストリームのとき、$GU \downarrow V$もガード付ストリームとなる.

[定義 25]

GUをガード付ストリーム, θを単純な代入式の集合とするとき、 θとGUから定まる次の集合がやはりガード付ストリームとなるとき、

$$\{< \sigma|U_k > \mid < \sigma'|U_k > \in GU, \sigma = \theta \cup \sigma'\}$$

これを $GU \bowtie \theta$ で表わす.

[定義 26]

ガード付節の集合Dについて、解釈IがDのモデルであるとは、 Iの任意の元tについてある節:

$$H :- U_{g1},\ldots,U_{gm} | X_1 = \tau_1,\ldots,X_h = \tau_h, B_1,\ldots,B_k \in D$$

が存在して、tのインスタンスが次のような形をしていることである.

$$H : -\{< \{U_{g1}, \ldots, U_{gm_1}\} | uni(X_1, \tau_1) >, \ldots, < \{U_{g1}, \ldots, U_{gm_1}\} | uni(X_h, \tau_h) >\} \cup$$
$$((GU_1 || \ldots || GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}) \downarrow Var(H)$$

ここでGU_iはσB_iというゴールのトレース$(\in I)$のインスタンスのボディ部、
σは

$$\{U_{g1}, \ldots, U_{gm}\} \cup \{X_1 = \tau_1, \ldots, X_h = \tau_h\} \cup \sigma'$$

という形の$Var(H) \cup \{X_1, \ldots, X_n\}$に制限された$\omega$−代入であり、かつ

$$\forall < \theta | U > \in (GU_1 || \ldots || GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}, \theta \subset \sigma.$$

となる.

次の命題はモデルの定義より明らかになりたつ.

[命題]

添え字の集合Indについて、節集合Dのモデルの族 $M_i(i \in Ind)$を考える. このとき

$$\bigcup_{i \in Ind} M_i$$

はやはりDのモデルである.

上の命題よりDの全てのモデルの集合和をとったものはやはりDのモデルであり、これはDの最大のモデルとなる. 節集合DのセマンティクスとはDの最大モデルによって定義される. 直観的には最大モデルとはDの上で実行したすべての正常な計算のパスを集めたものと考えることができる.

[例 4]

次のプログラムDについて、

```
shuffle(X, Y, Z) :- X = [A1|X1] | Z = [A1|Z1], shuffle(X1, Y, Z1).
shuffle(X, Y, Z) :- Y = [B1|Y1] | Z = [B1|Z1], shuffle(X, Y1, Z1).

inva(Z, Y) :- Z = [A|Z1], A = a |Y = [B|Y1], B = b, inva(Z1, Y1).
inva(Z, Y) :- Z = [B|Z1], B = b | inva(Z1, Y).
```

GU_1, GU_2を [例 1] の通りとするとき、Dの最大モデルは次のような元を含む.

$$\mathtt{inva(Z,Y)} : -GU_1,$$
$$\mathtt{shuffle(X,Y,Z)} : -GU_2$$

という形の元を含む. すなわちこれらのゴールは単独に実行した場合, GU_1, GU_2 に示されるような動作をする可能性がある. しかしながら次のようなゴール節として同時に実行した場合, それぞれのゴールが上のような動作をすることはない.

$$\mathtt{inva(Z,Y)}, \mathtt{shuffle(X,Y,Z)}$$

一般に上の様なゴール節においてXにaの列を無限に与え実行を続けた場合、Zの具体化された部分ではbの数はaの数を決して上回ることはない. このことは, このセマンティクスにおいてそれぞれのゴールの任意のトレースの同期付マージの結果についてbがaより先行するような結果がありえないことを示すことによって議論できる. このような議論は [Levi 88] での方法や、[Sakakibara 85] のアプローチでは不可能なものである.

□

与えられたプログラム節の集合Dのモデルで真となるゴール節は, [Ueda 85] に述べられた GHC の計算規則のようなガード / コミットを考慮した並列入力反駁の規則もとで " 正常に動作する " ゴール節を定めることを目的としている. しかしながらここでいう " 正常な動作 " は必ずしも有限時間で成功するゴールを意味しない. すなわち先

に述べたように、無限に結果を出力しながら計算を続けるプロセスはここでは正常な動作をするものと見し、そのようなプロセスを表わすゴール節は真となる．　また、サスペンドするようなゴール (またはゴール節) についても場合によっては真となる．すなわち次のようなプログラムについて、

```
p(X, Y) :- X = a | Y = b.
q(X, Y) :- Y = b | X = a.
t(X, Y) :- X = a | true, p(X, Y), q(X, Y).
```

このプログラムのもとではt(X, Y)というゴールはサスペンドするが、 X がaに具体化されることにより実行がさらに進む．この場合、Dの最大モデルには

$$\mathtt{t(X,Y)} : -\{< \{\mathtt{X} = \mathtt{a}\}|\mathtt{true} >, < \{\mathtt{X} = \mathtt{a}\}|\mathtt{Y} = \mathtt{b} >, < \{\mathtt{X} = \mathtt{a}\}|\mathtt{X?} = \mathtt{a} >\}$$

という元が含まれ、したがってこのようなゴールは真となる．しかしながら、プログラムの行目のtの定義を次のように変更した場合を考える．

この場合、ゴールtを実行すると、p(X, Y), q(X, Y)というゴール節が呼び出された後サスペンドし、以後実行が進む可能性はない．このようなゴールは先の定義では偽となる．

4　不動点的セマンティクス

本節では先に定義したモデル論的セマンティクスがDから定義される関数の最大不動点によって特徴づけられることをしめす．

$I/O - hist$が与えられたとき、全ての解釈の集合IPは集合の包含関係のもとで完備束となり、最大元は$I/O - hist$、最小限は空集合ϕである．

[定義 27]

ガード付節の有限集合から定まる関数$\Phi_D : IP \to IP$ を次のように定義する．

$\Phi_D(S) = S \cap \{t|t$の各インスタンスは、ある節

$$H : -U_{g1}, \ldots, U_{gm}|X_1 = \tau_1, \ldots, X_h = \tau_h, B_1, \ldots, B_k \in D$$

が存在し、次のような形をしている．

$$H : -\{< \{U_{g1}, \ldots, U_{gm}\}|uni(X_1, \tau_1) >, \ldots, < \{U_{g1}, \ldots, U_{gm}\}|uni(X_h, \tau_h) >\} \cup ((GU_1 \| \ldots \| GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}) \downarrow Var(H)$$

ここで GU_iはσB_iのトレースの元のボディ部．σは

$$\sigma = \{U_{g1}, \ldots, U_{gm}, X_1 = \tau_1, \ldots, X_h = \tau_h\} \cup \sigma'$$

という形で$Var(H) \cup \{X_1, \ldots, X_h\}$に制限された$\omega -$代入で、
$\forall < \theta|U > \in (GU_1 \| \ldots \| GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}$について、$\theta \subset \sigma\}$

鎖$S_i : S_1 \supset S_2 \supset \ldots$について、 S_iの最大下界を $\bigcap S_i$ で表わすことにする．

[定義 28]

Lを完備束とする．関数$f : L \to L$が，任意の鎖$S_i : S_1 \supset S_2 \supset \ldots$について

$$\bigcap\{f(S_i)|0 \leq i\} = f(\bigcap\{S_i|0 \leq i\}).$$

となるとき、fは下向きに$\omega -$連続であるという．

よく知られているように，関数fが$\omega -$連続ならばfは単調である．すなわち$S_1 \supset S_2$ ならば $f(S_1) \supset f(S_2)$ である．また次の 2 つの命題はよく知られたものである．

[命題 3]

完備束Lの最大元を$\top$とするとき、fが下向きに連続であるならばfの最大不動点$\mathrm{gfp}f$は次のような式で特徴づけられる.

$$\mathrm{gfp}f = \bigcap \{f^n(\top) | n \geq 0\}$$

ここで、$f^0(X) = X, f^{n+1}(X) = f(f^n(X))$とする.

[命題 4]

fが単調な関数のとき、fの最大不動点は次の集合の上限に等しい.

以下に、Φ_Dの性質を幾つか示す.

[命題 5]

Φ_D は下向きにω−連続である.

証明:

任意の鎖 $S_i : S_1 \supset S_2 \supset \ldots$ について、$\Phi_D(\bigcap S_j) = \bigcap\{\Phi_D(S_j)\}$であることを示せばよい。

(1) $\Phi_D(\bigcap S_j) \subset \bigcap\{\Phi_D(S_j)\}$:

任意の$t \in \Phi_D(\bigcap S_j)$について定義より$t$のインスタンスは、ある

$$H : -U_{g1}, \ldots, U_{gm} | X_1 = \tau_1, \ldots, X_h = \tau_h, B_1, \ldots, B_k \in D$$

および [定義 27] の条件を充たすσについて次のような形をしている。

$$H : -\{< \{U_{g1}, \ldots, U_{gm}\} | uni(X_1, \tau_1) >, \ldots, < \{U_{g1}, \ldots, U_{gm}\} | uni(X_h, \tau_h) >\} \cup \\ ((GU_1 \| \ldots \| GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}) \downarrow Var(H)$$

ここで GU_i は $\bigcap\{S_j\}$ に含まれる σB_i のトレース t_iの元のボディ部. $\bigcap\{S_j\}$ は S_j の下限であることより、すべてのi, jについて $t_i \in S_j$ となる. したがって、すべての j について $t \in \Phi_D(S_j)$.

$\bigcap\{\Phi_D(S_j)\}$ は $\{\Phi_D(S_j)\}$ の下界のうち最大のものであることより、

$$t \in \bigcap\{\Phi_D(S_j)\}.$$

(2) $\Phi_D(\bigcap S_j) \supset \bigcap\{\Phi_D(S_j)\}$:

$t \in \bigcap\{\Phi_D(S_j)\}$ とする. 任意のjについて $t \in \Phi_D(S_j)$. 定義より、t のインスタンスは、σB_i のあるトレース$t_i \in S_j$についてそのインスタンスのボディ部 GU_i に対して、

$$H : -\{< \{U_{g1}, \ldots, U_{gm}\} | uni(X_1, \tau_1) >, \ldots, < \{U_{g1}, \ldots, U_{gm}\} | uni(X_h, \tau_h) >\} \cup \\ ((GU_1 \| \ldots \| GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}) \downarrow Var(H)$$

という形で書き表わせる. 各t_i は任意のjについて $t_i \in S_j$ より $t_i \in \bigcap\{S_j\}$. したがって、$t \in \Phi_D(\bigcap\{S_j\})$.

□

[命題 6]

$\Phi_D(I) \supset I$であるときかつそのときのみ、IはDのモデル.

証明:

if part:

I が D のモデルであるとする. $t \in I$ とすると、モデルの定義よりある節

$$H : -U_{g1}, \ldots, U_{gm} | X_1 = \tau_1, \ldots, X_h = \tau_h, B_1, \ldots, B_k \in D$$

が存在して、t は [定義 27] の条件を充たすσについて次のような形をしている.

$$H : -\{< \{U_{g1}, \ldots, U_{gm}\} | uni(X_1, \tau_1) >, \ldots, < \{U_{g1}, \ldots, U_{gm}\} | uni(X_h, \tau_h) >\} \cup \\ ((GU_1 \| \ldots \| GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}) \downarrow Var(H)$$

ここで、GU_i は σB_i というゴールのトレース($\in I$) のインスタンスのボディ部である。Φ_D の定義より、明らかに$t \in \Phi_D(I)$ となる.

only if part:

$\Phi_D(I) \supset I$であるとする. 条件より任意の $t \in I$ について、$t \in \Phi_D(I)$ となる. すなわち$\Phi_D(I)$ の定義より、

$t \in \{t|t$の各インスタンスは、ある節

$$H : -U_{g1}, \ldots, U_{gm} | X_1 = \tau_1, \ldots, X_h = \tau_h, B_1, \ldots, B_k \in D$$

が存在し、次のような形をしている.

$$H : -\{< \{U_{g1}, \ldots, U_{gm}\} | uni(X_1, \tau_1) >, \ldots, < \{U_{g1}, \ldots, U_{gm}\} | uni(X_h, \tau_h) >\} \cup ((GU_1 \| \ldots \| GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}) \downarrow Var(H)$$

ここで GU_iはσB_iのトレースの元のボディ部. σは

$$\sigma = \{U_{g1}, \ldots, U_{gm}, X_1 = \tau_1, \ldots, X_h = \tau_h\} \cup \sigma'$$

という形で$Var(H) \cup \{X_1, \ldots, X_h\}$に制限された$\omega$ − 代入で、

$$\forall < \theta | U > \in (GU_1 \| \ldots \| GU_k) \bowtie \{U_{g1}, \ldots, U_{gm}\}$$について、$\theta \subset \sigma\}$

これはI がモデルであることの定義に他ならない。

□

以上より、次の定理が得られる.

[定理] 節集合Dの最大モデルM_DはΦ_D の最大不動点であり、次の式で表わされる.

$$M_D = \bigcap \{\Phi_D^n(I/O - hist) | n \geq 0\}$$

5 結び

本論文では、Flat GHC のサブセットについてその最大モデルによる宣言的セマンティクスを与え、さらに最大不動点による特徴付けを行った. ここで導入されたセマンティクスによって、従来の方法では不可能であったガード / コミット機構によって制御される無限計算の途中結果として得られる解をプログラム節の集合の論理的帰結としての特徴付けることが可能となった.

ここで提案したセマンティクスは、一種の成功集合セマンティクスである. したがって、プログラムが正常に実行されたときに得られる結果についての議論はこのセマンティクスで十分である. しかしながら、このセマンティクスで真となるゴールは、正常に実行される可能性のあるゴールであり、いかなる場合でも正常に実行されるものではない. GHC は "don't care" な非決定性言語である. したがって、いかなる場合も正常に実行されるゴールを特徴付ける方法も必要である. このようなゴールは、有限時間で失敗したりデッドロックに陥ったりする可能性のあるゴールの集合に含まれないゴールとして特徴づけることができる. このような、"失敗集合" セマンティクスについては、逐字的な論理型言語については [Falaschi 88] で報告されている. 本稿で述べたガード付ストリームや同期付マージの概念は並列言語の失敗集合セマンティクスを議論するにも有効なものであると考えられる.

6 謝辞

本研究をすすめるにあたり、有益な議論をして下さった ICOT 古川次長、第 1 研究室の皆様、および Pisa 大学 Levi 教授に感謝します.

References

[Apt 82] K. Apt, and M. H. Van Emden, Contributions to the theory of logic programming, J. Assoc. Comput. Mach. 29, (1982)

[Brock 81] J. D. Brock, W. B. Ackermann, Scenarios: A Model of Non-determinate Computation, Lecture Notes in Computer Science, No. 107, Springer, 1981

[Clark 86] K. L. Clark and S. Gregory, PARLOG: Parallel programming in logic, ACM Trans. on Programming Language and Systems 86, 1986

[Falaschi 88a] M. Falaschi, G. Levi, M. Martelli, and C. Palamidessi, A more general Declarative Semantics for Logic Programming Languages, Dipartimento di Informatica, Università di Pisa, Italy, Techn. Report, January 1988

[Falaschi 88b] M. Falaschi, and G. Levi, Operational and fixpoint semantics of a class of commited-choice logic languages, Dipartimento di Informatica, Università di Pisa, Italy, Techn. Report, January 1988

[Levi 87] G. Levi and C. Palamidessi, An Approach to the Declarative Semantics of Synchronization in Logic Language, Proc. of International Conf. on Logic Programming 87, 1987

[Levi 88] G. Levi, A new declarative semantics of Flat Guarded Horn Clauses, Tech. Rep. of ICOT, to appear, 1988

[Lloyd 84] J. W. Lloyd, Foundations of logic programming, Springer-Verlag, 1984

[Maher 87] M. J. Maher, Logic Semantics for a Class of Commited-Choice Programs, Proc. of International Conf. on Logic Programming 87, 1987

[Park 69] D. Park, Fixpoint induction and proofs of program properties, Machine Intelligence 5, Edinburgh University Press, Edinburgh, 1969

[Sakakibara 85] Y. Sakakibara, A Fixpoint Characterization of Stream Parallelism in Logic Programs, Proc. of 2nd Cof. JSSST, in Japanese, 1985

[Saraswat 85] V. A. Saraswat, Partial Correctness Semantics for CP[$\downarrow$, |, &], Lecture Notes in Comp. Sci., No. 206, 1985

[Saraswat 87] V. A. Saraswat, The Concurrent logic programming CP: definition and operational semantics, Proc. of ACM Symp. on Principles of Programming Languages, 1987

[Shapiro 86] E. Y. Shapiro, Concurrent Prolog: A progress report, Lecture Notes in Comp. Sci. No. 232, 1986

[Shibayama 87] E. Shibayama, A Compositional Semantics of GHC, Proc. of 4th Cof. JSSST, 1987

[Takeuchi 86] A. Takeuchi, Towards a Semantic Model of GHC, Tech. Rep. of IECE, COMP86-59, 1986

[Ueda 88] K. Ueda, Guarded Horn Clauses, MIT Press, 1988

[Ueda 86] K. Ueda, On Operational Semantics of Guarded Horn Clauses, Tech. Memo of ICOT, TM-0160, 1986