ICOT Technical Memorandum: TM-0408

TM-0408

GHCプログラムの公理的意味論について

村上昌己

November, 1987

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

ＧＨＣプログラムの公理的意味論について

On an Axiomatic Semantics for GHC Programs

村上　昌己
(財)新世代コンピュータ技術開発機構
Masaki Murakami
Institute for New Generation Computer Technology

ABSTRACT: Guarded Horn Clauses (GHC) is a parallel programming language based on Horn logic. It is impossible to prove the correctness of programs controlled by the guard/commit mechanism for enough strong output conditions, by using the verification method for pure Horn logic programs. This paper proposes an axiomatic verification method for partial correctness of GHC program as an extension or a modification of Hoare logic.

１．はじめに

筆者は先にＦＧＣＨプログラムの部分的正当性の検証のための公理体系について報告した〔文献 3〕。しかしこの体系では，あるプロセスがプロセス間の同期によって決定的に動作するときには，十分に強い出力条件について部分的正当性を証明することができない場合があった。その結果，Brock Ackermannの例題〔文献 1,4〕の様に，互いに動作の異なる２つのプログラムを区別することが出来ないなどの不都合が生じた。本稿では，〔文献 3〕の体系の一部を修正し，プロセス間の同期によって成立する出力条件についての部分的正当性を証明する方法についてのべる。これによって，〔文献 3〕の方法では区別することができなかったBrock Ackermannの二つのプログラムについて，それぞれ十分に強い出力条件について部分的正当性を示すことにより，二つのプログラムを区別することが可能となる。

２．ゴール節形式

GHCプログラムはガード付Horn節の集りである。ここでは簡単化のためガード部分には＝，＜のみ出現し，またボディ部分に出現するシステム述語は＝のみであるものとする。プログラム変数の集合を VAR，関数記号の集合を FUN，それらから作られる項の集合を TERM，述語名の集合をPREDとする。プログラムの部分的正当性は通常Φ｛ｐ｝Ψのようなの記法を用いて表現する。ここでΦ，Ψはそれぞれ入出力条件，ｐはプログラムである。GHCプログラムの正当性についてもこの記法を用いる。ここでΦ，Ψはプログラムが実行される領域のうえの述語であるものとする。ここでは FUN の元から作られる Herbrand領域のうえの関係としてそのセマンティクスが定まるものに制限する。ｐの部分には，ゴール節を表わす表現を用いる。

〔定義 1〕 ゴール形式

Ｄをガード付Horn節の集合とする。ｐをPREDに含まれる述語記号，τ1,…,τn を FUN，VAR及びVARと共通部分をもたない集合Var の元から作られる項とするとき，（このような項の集合をTermで表わす）ｐ(τ1，…,τn) を<u>ゴール形式</u> と呼ぶ。ここでVarの元はTERM の上をはしるメタ変数と考えられる。本稿では'変数'とはVarの元を表わし，ｘ，ｙ,…のような小文字であらわすことによって通常の変数項（すなわちVARの元)と区別する。プログラム節に出現する変数は便宜上 Varの元と考える。また本稿では'項'とは一般にTERMの元をあらわし，Term の元は'項形式'と呼んで区別する。ゴール形式ｇに対し，ある代入 Σ：Var → TERM を施すことにより，具体的なゴールＧが得られる，すなわちゴール形式ｇは次のように定まるゴールの集合|ｇ| を定めるものと考えることができる。

$$|g| = \{G \mid \exists \Sigma : Var \rightarrow TERM,\ G = \Sigma g\}$$

ゴール形式の並びを<u>ゴール節形式</u>と呼ぶ。ゴール節形式はゴール節の集りを表現している。GHCプログラムの正当性は｛｝の内側にトップレベルのゴール節形式を持つ式のよって表現される。トップレベルのゴール節の形を与えると，その実行中に呼ばれるサブゴールの形式が定まる。次にトップレベルの性質を証明するのに必要なサブゴールの性質を表現する式について述べる。

〔定義 2〕 プロセス形式

Ｄをガード付Horn節の集合，ｇ1,…,ｇn をトップレベルのゴール節形式とするとき：

i) ｇ1,…,ｇn はプロセス形式である。

ii) プロセス形式ｇがＤに含まれるある節のヘッドＨとｇｕ θによって単一化可能であるとき，その節のボディ部Ｂ1，…，Ｂkにθを適用したθＢ1，…，θＢk はプロセス形式である。

iii) ｇ をプロセス形式，代入Σ :Var → Term をあるｇ'についてΣｇ' = ｇとするとき ｇ' はプロセス形式である。

通常GHCプログラムは実行の方向を暗黙の内に定めているが，ここでは，議論を簡単にするため，すべてのゴール形式について，それ自身が具体化する変数は高々一つであるものと仮定する。このような変数を出力変数と呼ぶ。ゴール形式ｇが表現しているゴールとしては，出力変数に変数項を代入するようなΣによって得られるもののみを考えれば充分である。またトップレベルのゴール節形式ｇ1，…，ｇn について，あるｇiの出力以外の変数ｘがあるｇj(i ≠ j) によって実行中に具体化されるとき，ｇjをｘのプロデューサとよぶ。ここでは議論を簡単にするためすべての非出力変数についてそのプロデューサはΣに係わらず定まっているものとする。またｘを，Ｄに含まれるある節：

Ｈ:- Ａ1，…，Ａk ｜ Ｂ1，…，Ｂh

のＢ1，…，Ｂhに出現し，かつＨ，Ａ1，…，Ａk には現われない変数とするとき，ｘを具体化できるＢi(１≦ i≦ h) が高々一つ定まるものとする。これをｘのプロデューサとよぶ。ｇの出力変数をｙ，ｇとＨのｙを具体化しない

mguをθとするとき，yはθB1，…，θBhの出力変数とよぶ。

3.1 記法

あるゴール形式gについて|g|に含まれるゴールのサブゴールの集合を考える際，gをDに含まれるある節を用いて展開したg1,…,gnについて|g1,…gn|の元でgに出現しない変数xが変数項であるものだけを考えれば十分である。このように，変数項の場合のみをかんがえればよい変数には，本稿では↓をつけることにより区別する。すなわちgが↓のついた変数を含む場合，

|g| = {Σg | ∃Σ:Var → TERM ここで ↓のついた変数xについては，Σx ∈ VAR}

本稿では議論を簡単にするため，各プロセス形式について出力変数以外には↓のついた変数は高々一つしか出現しないものとする。トップレベルのゴール形式g1,…,gnについて，↓のついた変数が含まれれば，g1,…,gnから定義されるプロセス形式に出現する同じ変数についても↓がつけられる。さらに，H :- A1，…，Am | B1，…，BkをDに含まれる節，gをプロセス形式としある代入θについてθH = θgとする。yがθB1，…，θBkに出現し，かつθHにもθB1，…，θBkにも出現しない場合，yは↓をつけることができる。ガード付節の集合D，トップレベルのゴール節形式 g1,…,gn またはg1，…，gn から定義されるプロセス形式g1',…,gm' およびアサーション言語しについて，Φ，Ψ ∈ Lならば Φ {g1，…，gn}ₚΨという形の式をもちいる。} の後のDは特に混乱がない限り省略する。

4．計算木集合

ここではGHCの操作的意味論について簡単に述べる。ここで述べるセマンティクスは，計算木の概念をゴール(節)形式に対して拡張したものとなっている。本稿では，操作的意味論はGHCプログラムに対する部分的正当性の概念を定式化することを目的としている。従って，成功する計算のみが問題となる。すなわち GHCプログラムのセマンティクスはガード付節の集合とゴール節形式から定まる成功する計算の集合として与えられる。

具体的なゴール節に対する計算の概念は，〔文献 4〕のtrace treeの概念と基本的に同様である。すなわち直観的には，GHCプログラムの個々の計算とは，それぞれ根として各ゴールをもつ有限木の組である。それぞれの木は，各ゴールをゴール節のなかで実行したときにつくられるAND木である。これをそのゴールの計算木とよぶ。計算木の各節点は計算が成功したときに結果として得られる代入によって具体化されたゴールである。各内部節点の子供はその親のゴールがある節にコミットしたときに得られるサブゴールに対応する。 GHCプログラムは一般的には非決定性を含むので，各ゴール節について成功する計算木は複数存在する。並列に実行される複数のゴールを含むゴール節について，計算木の組〈t1,…,tn〉の集合が同様に定義される。ガード付節の集合Dとゴール節G1，…，Gn から定まる計算木の組の集合をCOMP(G1,…，Gn，D)で表わす。

〔定義 3〕計算木集合

トップレベルのゴール節形式g1，…,gnについての計算木の組の集合 Comp(g1，…,gn，D)は次のように定義される。

Comp(g1，…,gn，D) =
{〈t1,…,tn〉 | G1，…，Gn ∈ |g1，…，gn|,
〈t1,…,tn〉 ∈ COMP(G1,…，Gn，D)}

Comp(g1，…,gn，D) の元〈t1,…,tn〉からi番目の成分をとりだしたとき，その集合は Comp(g，D)とは一般には等しくはない。すなわち前節で述べたように↓のついた変数xが最初具体化されずに呼ばれ，実行中に外から具体化されるプロセスの動作を考えなければならない。この場合は↓は，xは最初具体化されずに呼ばれ，別のプロデューサによって具体化されることを意味する。このとき，gの計算木としては，gの出力変数がxのプロデューサを起動させるに充分な具体化をさせる単一化の先祖に，xが具体化されていなければできないようなコミットをふくむようなものは考慮する必要はないことを意味する。すなわちgの計算木の集合は，xのプロデューサを起動させるにgが出力をどの程度具体化させる必要があるかを定めることにより与えられる。ここではプロデューサを起動させるに必要な出力の具体化の程度は一つの項形式により表現できるものと仮定する。すなわち，一般にトップレベルでないプロセス形式の g1,…,gn のセマンティクスは，ある項形式τから計算木の集合を与える関数 Comp(g1,…,gn，D)(τ) として与えられる。

ゴール p(T) の節Cへのコミットが，T を変数項に置き換えて得られるゴールp'が Cにコミットできないとき，Tについてnon-trivialであるという。

〔定義 4〕計算木集合(その2)

g1,…,gnの出力変数をxとするとき，

Comp(g1,…,gn,D)(τ) =
{t | t ∈ COMP(Σg1,…,Σgn,D) かつxはtの↓のついた変数についてnon-trivialなコミットをするゴールの子孫以外に出現する単一化を実行してτより具体化される。}

〔定義 5〕

Γを Θ {g} Tという形の式の集りとする。ここでgはトップレベルでないゴール節形式g1，…，gnから定まるプロセス形式である。g1,…,gn と仮定の集合Γ及び項形式τについて

|= Φ {g1,…,gn} Ψ

であるとは，任意の〈t1，…,tn〉 ∈ Comp(g1,…,gn，D)(τ) について各tiの根をσΣgiであらわすことにすると，Γのすべての元がトップレベルで真ならばΣΦ ⇒ σΣΨであることを表わす。ここで，Θ {g} T がトップレベルで真であるとは，すべてのt ∈ Comp(g，D)について，tの根がσΣgであるならば ΣΘ ⇒ σΣTであることを表わす。またΣはgから実際に呼ばれたゴールを得る代入，σはΣgが成功した結果得られた代入である。トップレベルのゴール節形式 g1,…，gn が入出力条件Φ，Ψについて部分的正当であるとは，空なΓ及びg1,…gnの出力変数に決して代入されるようなことがないようなある項τについて|= Φ {g1,…,gn} Ψ であることを

いう，

5．公理系

ここで述べる公理系は基本的には次のような考え方に基づいている。並列に走るプロセスを含むゴール節形式g1，…，gnの性質は各gi(1≦i≦n)の性質から導かれる。また各giの性質はそのサブゴールの性質から導かれる。再帰的に定義された述語についてはある種の帰納法を用いる。

推論規則 ：

代入:

$$\frac{\Phi\ \{g\}\ \Psi}{\sigma\Phi\ \{\sigma g\}\ \sigma\Psi}$$

ここでσは↓のついた変数を具体化しない。

緩結1:

$$\frac{\Phi\ \{g1, \cdots, gn\}\ \Psi \qquad \Psi \Rightarrow \Psi'}{\Phi\ \{g1, \cdots, gn\}\ \Psi'}$$

緩結2:

$$\frac{\Phi' \Rightarrow \Phi \qquad \Phi\ \{g1, \cdots, gn\}\ \Psi}{\Phi'\ \{g1, \cdots, gn\}\ \Psi}$$

導出1:

$$\frac{\Phi \wedge T=S \Rightarrow \Psi}{\Phi\ (T=S)\ \Psi}$$

導出2:

$$\frac{P1, \cdots, Ps}{\Phi\ \{g\}\ \Psi}$$

ここでP1，…，Psは次のようなPj (1 ≦ j ≦ s)を全てならべたものである。節Cj (j = 1, s) ∈ D:

$$Hj :- Bj1, \cdots, Bj\,h_j \mid Aj1, \cdots, Aj\,m_j$$

が存在して，↓のついた変数を具体化せずσ$_j$ g = σ$_j$ Hjとなるσ$_j$について，Pjはつぎのような形をしている。

$$Pj \equiv \bigwedge_{k=1,\,h_j} \sigma_j Bjk \wedge \sigma_j \Phi \quad \{\sigma_j Aj1, \cdots, \sigma_j Ajm\}\ \sigma_j \Psi$$

ここで各 Bjk (k=1, h_j)，についてBjkに出現する↓の付いた変数を具体化しない代入λjk が存在し，λjkBjk が真となる。

この規則を用いる推論で，結論に↓の付いた変数が出現し，かつその↓を取り除いた式がこの推論の前提から↓を取り除いた式から導くことが出来ないとき，この推論を縮退した推論と呼ぶ。

次に並列化についての規則を導入する。この規則は，各プロセスgiについての性質を記述する式を前提とし，g1，…，gn についての性質を記述する式を導く。この規則は，この推論の結論を根とする部分証明図Pに後に示すような条件が成立したときのみ有効である。（部分証明図は後に定義される証明図の部分木として定義される。）xを変数，τをTermの元，g1をxを出力以外の変数としてもつゴール形式，g2をxを出力変数としてもつゴール形式，form（g）はPに出現するΦ｛g｝Ψという形の式，ｆｒを縮退した推論の結論とするとき，R（x，τ，form(g1)，ｆｒ，P）及びO(x，τ，form(g2)，ｆｒ，P）を定義する。

```
R(x,τ,form(g1), f r, P) =
   if 「form(g1) がΘ {g1} T のとき，
        x = τ∧ Θ は常にfalse.」
   then true
   else if 「xのプロデューサ pがPに出現する」
        then O(x, τ, form(p), f r, P)
        else true
```

直観的には，R(x，τ，form(g1)，ｆｒ，P）がtrueであるとは，xはg1が呼び出された時点ではτの形までは具体化されてないことを意味する。

```
O(x, τ, form(g2), f r, P) =
if 「f r が form(g2)」
then true
else if 「f g2 はxと T の単一化を含む」
     then if 「T とτが単一化可能」
          then if 「∃ σ: T = στ」
               then false
               else
        「＼／O(xi, σxi, form(pi), f r,P)
        i = 1, h
         ここでσはTとτのmguでTに出現する変数
         x1,…,xhを具体化する。またpiはxiのプ
         ロデューサ」
           else true
        else
         「／＼( ∨ R(yu, σk yu, form(g2),
        1≦k≦n 1≦u≦w                f r,  P)
    ∨  O(x, τ, form(qk(…,x)), f r, P))
         ここで次のような節が存在し：

            Hk(…,y) :- Bk |
               …,qk(…,y),…. (1≦k≦m)

         ある代入 σk について σkg2 = σk Hk，
         σy = x となりσkはg2の変数 y1, …，
         yw を具体化する.」
```

直観的には，O(x，τ，form(g2)，form(gr)，P）がtrueであるとは，g2はgrを実行することなしに，xをτの形に具体化することができないことを意味する。

並列化：プロセス形式 g1,…,gn について，giが↓のつけられた変数xをもつとき，かつxのプロデューサgj(1≦j≦n)が存在するならばgi'をgiからxの↓を取り除いて得られるゴール形式とする。もしxのプロデューサがg1,…,gnに現われなければ，gi' = giとする。このとき，Φi｛gi｝Ψi (1≦i≦n)の部分証明図に出現するすべての縮退した推論について，次の式が成立するとき：

$$\wedge R(x_j, \tau_j, \Theta_j\{h_j\}T_j, \quad 1 \leqq j \leqq n \quad \Theta_j\{h_j\}T_j, P) = \text{true}$$

このとき

$$\frac{\Phi_1\{g_1\}\Psi_1, \cdots, \Phi_n\{g_n\}\Psi_n}{\wedge_{i=1,n}\Phi_i\{g_1', \cdots, g_n'\} \wedge_{i=1,n}\Psi_i}$$

ここで$\Theta_j\{h_j\}T_j$ ($1 \leqq j \leqq n$: nは縮退した推論の数) は各縮退した推論の結論である。またx_jは縮退の原因になった↓のついた変数，同じくτ_jはヘッド側に出現する縮退の原因となった項形式。

トップレベルでないプロセス形式$p(x\downarrow, \cdots)$について，$\Phi\{p(x\downarrow, \cdots)\}\Psi$を根として縮退した推論を含む部分証明図が存在することは，ある仮定のもとでxがあるタイミングで遅れて具体化されれば，Φを充たす入力に対する計算の結果は，Ψを充たすことを意味する。さらに，$\Phi\{p(x\downarrow, \cdots)\}\Psi$の部分証明図と$x$のプロデューサについての部分証明図とのあいだに，並列化の規則が適用できる条件が成立することは，プロデューサがxを具体化するタイミングと，$\Phi\{p(x\downarrow, \cdots)\}\Psi$の証明で想定した$x$の具体化の遅れが整合がとれていることを意味する。

通信:
$$\frac{\Phi\{g(\cdots, x, \cdots)\}\Psi}{\Phi\{g(\cdots, x\downarrow, \cdots)\}\Psi}$$

ここで結論の{}の内側でのすべてのxの出現について，↓がつけられる。

この公理系では，通常のHoare論理の公理系と同様に，プログラムの走る領域の上の定理は，全て公理とみなす。さらに次の式を公理として導入する。

公理

1) $\text{false}\{g_1, \cdots, g_n\}\Psi$
2) $\Phi\{g_1, \cdots, g_n\}\text{true}$

通常Hoare論理の体系では，証明図は葉が公理に対応し，最下結論に対応する木として定義される。この体系では，公理に加えて'帰納法の仮定'が証明図の葉の部分に出現することを許す。

〔定義．6〕証明図

トップレベルのゴール節形式$g_1, \cdots, g_n$に対して，$\Phi\{g_1, \cdots, g_n\}\Psi$の証明図とは，次のように定義される木である。

1) 木の根は$\Phi\{g_1, \cdots, g_n\}\Psi$に対応する。
2) すべての節点 nについて，次のa) または b) が成り立つ。

a) nはnの子供から先に示した推論規則を用いて導かれる。
b) nは葉であり，次のいずれかが成立する。

(i) nは公理である。
(ii) nはある先祖の節点n'と同一の式であり，nからn'に至る道の上で，少なくとも1回導出2の規則が使われ，かつnは↓のついた変数を出力変数として含まない。

↓のついた変数は次のようにして証明図に出現する。

1) 前提に出現する↓のついた変数xは，結論でも↓がつけられる。
2) $\Phi\{g\}\Psi$が導出2の結論で，すべてのj ($1 \leqq j \leqq s$) σ_jがxを具体化しないならば，結論のgに出現するxは↓をつけることができる。
3) $\Phi\{S=T\}\Psi$が導出1の結論でxがSまたはtに出現する変数ならば，↓をつけることができる。
4) xが帰納法の仮定に出現するとき，xは↓をつけることができる。

プロセス形式$g_1, \cdots, g_n$および入出力条件Φ, Ψについて，$\Phi\{g_1, \cdots, g_n\}\Psi$を根にもち，公理でない式$\Gamma_1, \Gamma_2, \cdots, \Gamma_k$が葉に出現するような部分証明図が存在するとは，$\Gamma = \{\Gamma_1, \Gamma_2, \cdots, \Gamma_k\}$と縮退した推論の子孫でない部分に出現する出力変数への単一化を合成した結果τについて

$$\models \Phi\{g_1, \cdots, g_n\}\Psi.$$

特に，$g_1, \cdots, g_n$がトップレベルの場合，$\Phi\{g_1, \cdots, g_n\}\Psi$の証明図が存在するとき，$g_1, \cdots, g_n$は$\Phi$と$\Psi$について部分的に正当である。

6. 結び

本稿では，GHCプログラムの部分的正当性を検証する公理系について述べた。この公理系を用いて，最初にのべたBrock Ackermannの例題の二つのプログラムをそれぞれ異なる出力条件について部分的正当なプログラムとして区別することができた。本稿では議論を簡単にするために対象とするGHCプログラムに対して様々な制限を加えた。しかしこの制限はガードがフラットなものであることを除いて本質的なものではなく，本稿で述べた方法を拡張することにより，一般的なフラットなGHCプログラムの検証が可能であると考えられる。

〈謝辞〉日頃有益な御討論をして下さるICOT古川次長，及び第1研究室の皆様に感謝します。

参考文献:

〔1〕 J. D. Brock, W. B. Ackermann, Scenarios: A Model of Non-determinate Computation, Lecture Notes in Computer Science, No. 107 Springer, 1981
〔2〕 Y. Kameyama, Axiomatic System for Concurrent Logic Programming Languages, Master's Thesis of the University of Tokyo, 1987
〔3〕 M. Murakami, Proving Partial Correctness of Guarded Horn Clauses, The Logic Programming Conference '87 1987
〔4〕 A. Takeuchi, Towards a Semantic Model of GHC, Tech. Rep. of IECE, COMP86-59, 1986