GHCプログラムの検証について
7B-6 ー同期による決定的動作の検証ー

村上 昌己
(財)新世代コンピュータ技術開発機構

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［村上 87］．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ことによって成り立つ出力条件については，検証できなかった．その結果，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場合があった．本稿では，先にのべた体系の一部を修正し，従来は扱えなかった性質の検証方法について述べる．

2．相互通信による決定的動作

GHCプログラムにおいて，プロセスがその同期によって決定的に動作する典型的な場合は，Brock-Ackermanの例題［竹内86］にみられる．すなわちあるプロセスpがある変数xが具体化されずに呼ばれ，この影響でpがコミットできる節が決定的に定まってしまう場合である．一般にxは後になって別のプロセスによって具体化されうる．

［村上87］で述べた体系では，基本的には次のような考え方でプログラムの性質を証明するように推論規則が構成されている．すなわち，幾つかのゴールが並列に走っているとき，その性質は独立に証明された各プロセスの性質の論理積から証明される．また各プロセスの性質は，そのプロセスがコミットできる節のボディ部分に出現するサブゴールの性質から導かれる．しかし，あるプロセスが上に述べたような状況で決定的に動作することによって成立する性質については，独立に証明された各プロセス単独の性質の単純な論理積からは証明できない．そこで本稿では，次のような考え方で公理系を構成する．すなわち各プロセスについては，ある変数の具体化についてあるタイミングを想定した上で，そのプロセスの動作についての性質を証明する．さらにそのプロセスを他のプロセスと並列に走らせたとき，想定されたタイミングがプロセス間の同期と整合がとれていることが示されたとき，想定されたタイミングのもとで成り立つ性質の論理積から，全体の性質を導くという形にする．ある具体化のタイミングのもとでのプロセスの性質は，そのプロセスの非決定性を制御されたときの各動作について一般的に成り立つことである．プロセスpが他のプロセスからの入力タイミングによって決定的に動作するような場合，pの動作としては，pを単独で引数を最初から具体化された状態で実行した場合の動作のうち，出力変数が具体化される以前には，入力タイミングが遅れる変数が具体化されていなくても可能なコミットのみを行うものだけを考えればよい．ここではプログラムの動作を計算木によって表現している．

3．GHCプログラムの計算木集合

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．さらにすべての述語は高々1個の出力用引数しか持たないものとする．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．PREDとTERMからつくられる定義節の集合D，ゴール列G1,…,Gnに対して計算木の集合を COMP(G1,…,Gn, D)であらわす．COMP（G1,…,Gn, D）の各元はG1,…,GnをDの定義に従って実行したときに得られるand-treeの組である．このようなtreeによる定義は［竹内 86］に見られる．

ここでは，プログラムの計算木を考える目的は，公理的意味論で用いられるformulのセマンティクスを定義することである．公理的意味論では個々の具体的なゴールについてではなく，ある述語の呼出しについての一般的な議論について関心がある．したがって以下では，ある形をしたゴールの集合についての計算木の集合を考える．

ここではゴールの各引数にVARに含まれない変数を用いることにより，不特定多数のゴールの集合を表現する．このような表現をゴール形式と呼ぶ．具体的なゴールはゴール形式に含まれる変数にTERMの元を割り当てる代入Σを施すことにより得られる．ゴール形式の並び(ゴール節形式と呼ぶ．) g1, …, gn の計算木の集合を Comp(g1,…, gn,D)で表わし，次のように定義する．

$$Comp(g1,\cdots,gn,D) = \{t \mid t \in COMP(G1,\cdots,Gn,D),\ \exists\Sigma : Gi = \Sigma gi\ (1 \leqq i \leqq n)\}$$

ゴール節形式から一つのゴール形式（すなわちプロセス）p(x)をとりだしたとき，p(x)についての計算木の集合は，Comp(p(x), 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ある変数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．このようなプロセスをp(x↓)のようなゴール形式で表わす．ゴール節形式中でこのプロセスにxを出力するプロセスをxのプロデューサと呼ぶ．ここでは与えられたゴール節形式に対してどの変数のプロデューサも一意に定まっているものとする．↓のついた変数を含むゴール形式gの動作の集合は↓を取り除いたゴール形式g'の計算木の次のような集合として定まる．

Comp(g, D) =
{t | t∈COMP(G, D), ∃Σ:G = Σg', かつ t では出力変数を具体化する単一化の先頭に出現するコミットは，↓のついた変数が具体化されていなくても可能なもののみである．}

A Verification Method for GHC Programs

Masaki Murakami
Institute for New Generation Computer Technology

定義節の集合D，ゴール節形式g1,…,gnが入力条件Φ及び出力条件Ψについて部分的正当であるとは，Comp(g1,…,gn, D)の全ての元tについて，successする計算木でかつ入力がΦを真にするならば，計算の結果得られた出力についてΨが真であることをいう．このとき次のような記法を用いる．

$$\Phi\ \{g1,\cdots,gn\}\ \Psi$$

↓のついた変数を含むゴール形式gについてもΦ{g}Ψの意味は同様に定まる．ここでは入出力条件として現われる述語は，FUNとVARから作られる規定項の集合としてそのセマンティクスが定まるものに制限する．

4．推論規則

ここで提案する公理系は［村上87］で提案したものを基本とし，公理，推論規則を次のように追加，修正する．これ以外の推論規則，公理および証明図の定義などは［村上87］と同様である．

〈導出〉〈修正〉［村上87］における〈導出2'〉の規則の↑を↓におきかえる．この規則による推論の結論が↓のついた変数xを含む形gのとき，この推論の前提の各formulaのxから↓を取り去ったものから，gのxから↓を取り去ったものがこの導出規則によって導けないとき，gを導いたこの推論は縮退しているという．

〈代入〉〈修正〉 $\Phi\ \{g\}\ \Psi \vdash \sigma\Phi\ \{\sigma g\}\ \sigma\Psi$

ただしgはゴール形式，またσは↓のついた変数および出力変数を具体化しない．

次に，並列化規則を修正する．ここで導入される規則は，前提としてformulaだけでなく，そのformulaを導くまでに構成された部分証明図をも参照する．準備としてR(x, τ, form(g1), fr, P)及びO(x, τ, form(g2), fr, P)を次のように定義する．ここで，xを変数，τは具体的な項，g1はxを入力として含むゴール形式，g2はxを出力として含むゴール形式，form(g)はゴール形式gを{}の内側にもつformulaの出現，frを縮退した推論の結論，証明図全体をPをとする．

```
R(x, τ, form(g1), fr, P) =
 if 「form(g1) がΦ(…のとき x = τ ∧Φが常にfales」
  then true
  else    ／＼O(x, τ, form(p), fr, P)
        p∈Proc
```

ここでProcはxのプロデューサの集合である．直観的にはR(x, τ, form(g1), fr, P)は「g1が呼ばれたときに，xはτの形であることはない」ことを意味する．

```
O(x, τ, form(g2), fr, P) =
  if 「fr がform(g2)」 then true
    else if 「g2はx = tの形をしている」then
          if 「tとτはユニファイアブル」 then
            if 「t = στとなる代入σが存在する」
              then false
            else
    「σt = στのとき，σによって具体化される変数を
     x1,…,xh, xiのプロデューサをpi とするとき，
     ＼／ O(xi, σxi,form(pi), fr, P)」
    i = 1,h
          else true
        else
「g2 にコミットしうる節を
Hk(…,y) :- Bk|…,qk(…,y), …, (1≦k≦a)
σkg1 = σkHk, σy = xであったとき，g2で出現する変数でσkで具体化されるものをy1, …,yu とするとき，
 ／＼ （ ＼／ R(yu, σkyu, form(g2), fr, P) ∨
1≦k≦a  1≦u≦u
          O(x, τ, form(qk(…,x)), fr, P))」
```

直観的にはO(x, τ, form(g2), form(gr), P)は「g2はgrを実行しない限り，xをτの形にはできない」ことを表わす．

(並列化)(修正)giが↓のついた変数を含み，xのプロデューサ gj' (i≠j, 1≦j≦n) が存在するならばgi'はgiのxから↓を除いたもの，存在しなければgi' = giとするとき

$$\Phi 1\ \{g1\}\ \Psi 1,\ \cdots,\Phi n\ \{gn\}\ \Psi n \vdash \bigwedge_{i=1,n} \Phi i\ \{gj',\cdots,gn'\} \wedge \bigwedge_{i=1,n} \Psi i$$

ただし，Φi{gi}Ψi (1≦i≦n)の各部分証明図に含まれる縮退した推論の結論をΘj{hj}Tj (1≦j≦a)，xjを↓のついた(すなわち縮退の原因となった)変数，τj を縮退によって無視された節のヘッドでxjと同じ引数にあらわれた項とするとき，

$$\bigwedge_{1\leqq j\leqq a} R(xj,\ \tau j,\ \Theta j\{hj\}Tj,\ P) = true$$

が成立する．

gがある部分的正当性を充たすことが，縮退した推論を含む証明図Pgによって示されたということは，「gがあるxをあるタイミングで受け取ったとき，その性質が成立する」ことを意味する．またgの証明図とxのプロデューサの証明図との間に並列化規則が適用できる条件が成立したことは，「xのプロデューサの出力のタイミングが，Pgで想定したxの入力タイミングと整合が取れている」ことを意味する．

〈通信〉(追加)$\Phi\ \{g(\cdots,x,\cdots)\}\ \Psi \vdash \Phi\ \{g(\cdots,x\downarrow,\cdots)\}\ \Psi$

〈公理〉(追加) i) false {g1,…,gn} Ψ

ii) Φ {g1,…,gn} true

5．結び

同期により決定的に動作するＧＨＣプログラムの検証方法についてのべた．この公理系により，Brock-Ackermannの例題［竹内86］で２つのプログラムの検証例を試み，これらを区別することができた．

謝辞：有益な討論をして下さった，ＩＣＯＴ古川次長，および第１研究室の皆様に感謝します．

〈文献〉

［村上87］N.Murakami, Proving Partial Correctness of Guarded Horn Clauses Progras, Proc. of The LPC 87

［竹内86］A.Takeuchi, Towards a Semantic Model of GHC, 信学技法, COMP86-59, 1986