失敗集合に基づく並列論理型プログラムの宣言的意味論

A Declarative Semantics of Parallel Logic Programs based on Failure/Deadlock Set

村上 昌己
Masaki Murakami

(財) 新世代コンピュータ技術開発機構研究所
Institute for New Generation Computer Technology

Abstract

本稿では、flat GHC プログラムの失敗集合すなわち実行中にデッドロックしたり、失敗する可能性のあるゴールの集合を、入出力履歴の概念をもとに定義する。このアプローチによって、ガード/コミット機構によって制御されながら計算を行うプログラムのデッドロックフリー性等の性質の議論が可能となる。

1 はじめに

筆者は先に、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報告した [Murakami 88]。そこで提案したセマンティクスは、純 Horn 論理型プログラムについての宣言的なセマンティクス [Lloyd 84] の拡張 / 変形であり、プログラムのセマンティクスは、そのプログラムを記述している論理式の集合のあるモデルとして定義された。ゴール節がこのモデルで真であるという概念は、そのゴール節が失敗もデッドロックもしないで、実行を続けることが可能であること (すなわち ω 成功する可能性があること) に対応していた。またこのモデルでは、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) すなわちプログラムの ω成功集合 (ω success set) を I/O 履歴の集合により定義した。

しかしながら、GHC のようなコミット機構を持つような論理型言語の場合、成功集合とデッドロック又は失敗する可能性のあるゴールの集合は空でない共通部分を持つため、あるゴール節がデッドロックしないこと又は単一化が実行中に失敗しないこと等は、成功集合だけでは議論することはできない [Falaschi 88]。

そこでここでは、flat GHC プログラムの失敗集合すなわち実行中にデッドロック / 失敗する可能性のあるゴールの集合を、入出力履歴の概念をもとに定義する。すなわち、ここではゴール節が真であることを、そのゴール節が異常動作をする可能性があることに対応させ、与えられたプログラムの失敗集合を、そのプログラムの最小モデルとして定義する。

2 ガード付ストリーム

本節では、[Murakami 88] で定義したガード付ストリームの概念を拡張し、プログラムの正常及び異常な動作をそれぞれ表現できるようにする。

Varを可算無限個の変数の集合，Fun を関数記号の集合とする．Funの各要素に対しては arity がそれぞれ定まっているものとする。 VarとFunの元から通常のように定まる項の集合を$Terms$ であらわす。 項 τ が単純であるとは、τ が変数項であるか、 arity が 0 の関数記号であるか，$f(X_1,\ldots,X_n)$ の形で$f\in Fun$ かつ $X_1,\ldots,X_n$ が異なる変数である場合をいう．

$Terms$ の上の代入は通常通り定義される。

[定義 1]

$X\in Var$, 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。

[定義 2]

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$\bigcup_{k\to\infty}\theta_k$ に等しいとき、σ をω代入とよぶ．

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

ω代入は無限項の上の写像を定義する．

[Murakami 88] で提案したガード付ストリームは、flat GHC プログラムにおける単位節の概念である I/O 履歴のボディ部分を構成するものである。すなわち、プログラムの正常な動作を表現するものである。本稿では、flat GHC の失敗する動作に着目するために、異常動作を表現するガード付ストリームをも同時に定義する。

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を示す 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 はある節にコミットするまでに通過するガードを解くのに必要な代入 σ とコミットした節の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．

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のスケジューリングによって異なるI/O履歴が存在する可能性がある．

ここでは失敗 / デッドロックする計算は U_b のかわりに $\perp$ を含むガード付ストリームによって表現される。

[定義 3]

$X \in Var$, 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 4]

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 判定式$X? = \tau$とするとき、$< \sigma|U >$をガード付代入とよぶ．ここでUは$uni(X,\tau)$または記号$\perp$であるとする。このときσを$< \sigma|U >$のガード部，U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に，判定式であった場合はテスト・ユニフィケーションに対応する。

また能動部に$\perp$が出現したときは、ここでもはやいかなる選択をしても、失敗 / デッドロックするゴールが呼びだされたことを表わす。

$< \sigma|U >$をガード付代入とする．$| < \sigma|U > |$ 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．

U が判定式又は代入式のとき、

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

$U = \perp$ のとき、

$$| < \sigma|U > | = \sigma$$

プログラムのある動作を表現するガード付代入の集合 GU は、実行順序に関して半整列集合となってい る
[定義 5]

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 6]

ガード付代入の集合 GU が次の条件をみたすとき、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，右辺は単一化可能．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、$\tau$ と τ' が単一化不能ならば、$(X = \tau') \notin \sigma$.

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、τ と τ' は単一化可能。

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．

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．

以下ではUは代入式、判定式、または$\perp$であるとする。

[定義 7]

$GU_1, \ldots, GU_n$ 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．

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

$$\begin{aligned} Gu_{k+1} = Gu_k \cup \\ & \{< \sigma|U > | \exists i, \exists < \sigma'|U > \in GU_i, \forall (X = \tau) \in \sigma', \\ & ((\forall j, < \sigma''|X = \tau > \notin GU_j) \vee \\ & < \sigma''|X = \tau > \in Gu_k) \wedge \\ & \sigma = (\sigma' - \{X = \tau| < \sigma''|X = \tau > \in Gu_k\}) \cup \\ & \{U|U \in \sigma'' < \sigma''|X = \tau > \in Gu_k\}\} \end{aligned}$$

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 8]

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.

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

ここで、

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

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 9]

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、

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

これを $GU \bowtie \theta$ で表わす.

3 モデル論的意味論

次に、先に述べたガード付ストリームの概念を基に、純 Horn 論理型プログラムにおける Herbrand 基底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.

まず、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.

[定義 10]

述語記号の集合を$Pred$ とする。 $H, B_1, B_2, \ldots, B_n$ を $Pred$, $Terms$ から作られる原子式, 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 11]

p を arity k の $Pred$ の元、$X_1, X_2, \ldots, X_k$ を互いに異なる変数、σ を ω 代入とするとき、$\sigma p(X_1, X_2, \ldots, X_k)$ はゴールである。

単一化はいきなりは現われないことに注意されたい。

[定義 12]

ゴール節とは、ゴール $g_i (1 \le i \le n)$ の並び $g_1, \ldots, g_n$ である。

[定義 13]

GUをガード付きストリームとする。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.

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

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。

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 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。そこで厳密には I/O 履歴の領域に変数名のつけかえによる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のが適切である。

以下では、Fun, Var, $Pred$ から定まるすべての I/O 履歴の集合の変数名のつけかえによる同値類から、適当な代表元を選びだした集合を $I/Ohist$ で表し、その元を I/O 履歴とよぶ。

[定義 14]

$H :- GU$ を入出力履歴とする。GU の任意の元 $< \sigma | U >$ について U が判定式又は代入式であるとき、$H :- GU$を正常な履歴とよぶ。一方、 U が $\perp$ であるような元を含むとき、失敗する履歴と呼ぶ。

$I/Ohist$は、すべての正常な履歴の集合$I/Ohist_\infty$と失敗する履歴の集合$I/Ohist_\perp$に分割される。

[定義 15]

$I/Ohist_\infty$ の任意の部分集合を∞解釈、 $I/Ohist_\perp$の任意の部分集合を$\perp$解釈と呼ぶ。

[定義 16]

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 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、$\theta\subset\sigma$ 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.

ゴールgのトレースtについて、tが正常な履歴であるならば正常なトレースとよぶ。また失敗する履歴であるとき、失敗するトレースとよぶ。

[定義 17]

$I_\perp$を$\perp$解釈、gをゴールとするとき、gが $I_\perp$で真であるとはあるω代入σが存在し、σgの失敗するトレースが$I_\perp$に含まれることをいう. また∞解釈: I_∞でgが真であるとは、gの成功するトレースがI_∞に含まれることをいう。

[定義 18]

$\perp$解釈:$I_\perp$および、∞解釈:I_∞が互いに矛盾しないとは、次の条件を充たすことをいう。

(1) 任意の $t\in I_\perp$ について、$t = H:-<\sigma|\perp>$ ならば、任意の $\theta\supset\sigma$ である θ について θH のトレースはI_∞に含まれない。

(2) $\sigma\subset\theta$ となる任意の θ について、θH のトレースがI_∞に含まれなければ、

$$H:-<\sigma|\perp>\in I_\perp$$

である。

以下では$(I_\perp, I_\infty)$ を互いに矛盾しない ∞解釈: I_∞及び$\perp$解釈:$I_\perp$ の対とする。

[定義 19]

ゴール節 $g_1,\ldots,g_n$ が $(I_\perp, I_\infty)$ で真であるとは、ある ω 代入 σ が存在し、各 $g_i(1\le i\le n)$ について σg_i のトレース t_i が $I_\perp\cup I_\infty$ に含まれ、少なくともひとつの j について $t_j\in I_\perp$ であり、かつ $t_1,\ldots,t_n$ 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.

ここで与えられた定義は、手続き的にはたとえ無事に走り続けるプロセスがあったとしても、どれかひとつのプロセスが失敗 / デッドロックすれば、全体としては異常動作とみなす立場をとっている。

[定義 20]

ガード付節:

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

が$(I_\perp, I_\infty)$で真であるとは、 Hから見えない変数を具体化せずかつ $B_1,B_2,\ldots,B_n$ を真とする ω 代入 σ が存在するならば、次のような入出力履歴が$I_\perp$に含まれることである。

$$\begin{aligned}H:-\{&<\{U_{g1},\ldots,U_{gm},\}|U_{b1}>,\ldots,\\&<\{U_{g1},\ldots,U_{gm},\}|U_{bh}>\}\cup\\&((GU_1\|\ldots\|GU_k)\ltimes\{U_{g1},\ldots,U_{gm}\})\downarrow Var(H)\end{aligned}$$

ここでGU_iはσB_iというゴールのトレース$(\in I)$のボディ部分。

[定義 21]

Dを GHC プログラム、M^D_∞ を D の ω成功集合とする。$\perp$解釈:$I_\perp$ が D の $\perp$モデルであるとは、$I_\perp$ がM^D_∞と矛盾せず、かつD に含まれる全ての節が、$(I_\perp, M^D_\infty)$ で真であることをいう。

定義より、D の$\perp$モデルは複数存在する。これらの複数の $\perp$モデルのうち、最小なものが存在することが示される。これを D の失敗集合とよび、$M^D_\perp$ で表わす。D のセマンティクスは $(M^D_\perp, M^D_\infty)$ によって定まる。

与えられた flat GHC プログラム D について、上で定義されたセマンティクスで真となるゴール節は、D で実行した際に失敗 / デッドロックする可能性のあるゴール節となっている。

4 おわりに

本稿では、flat GHC プログラムのデッドロック / 失敗する動作を、宣言的に特徴付けるセマンティクスすなわち、プログラムの失敗集合について述べた。本稿では、与えられたプログラムについてその$\perp$モデルを定める手法に、例えば不動点による特徴付けについては触れなかった。これらの結果については、稿を改めて報告する。

5 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.

6 参考文献

[Falaschi 88] 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

[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

[Murakami 88] M. Murakami, A Declarative Semantics of Parallel Logic Programs with Perpetual Processes, to appear in Proc. of FGCS '88, 1988