TM-1112

Flat GHC プログラムの抽象解釈のための意味論

堀内 謙二

September, 1991

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

Flat GHC プログラムの抽象解釈のための意味論

The Semantics for Abstract Interpretation of Flat GHC Programs

堀内 謙二

Kenji HORIUCHI

(財) 新世代コンピュータ技術開発機構

概要 本稿では，Flat GHC プログラムの不動点意味論を定義する．この意味論は Flat GHC プログラムの抽象解釈のベースとなる意味論として使われる．そのため，ここで提案される意味論は通常の意味論より抽象度の低い意味論である．また，操作的意味論との関係も考察する。

1. はじめに

本稿では，Flat GHC プログラムの不動点意味論を定義する．この意味論は Flat GHC プログラムの抽象解釈のベースとなる意味論として使われる．また，操作的意味論との対応も議論される．

まず並行論理型言語 Flat GHC とその操作的意味論について述べた後，ここで提案する不動点意味論を定義し，操作的意味論との関係を論ずる．最後に抽象解釈にどのように利用されるかについて述べる．

2. 準備

ここでは，本稿で持ちいられる基本的な概念について定義する．それらの多くは文献 [9][10] で定義されているものである．

Var を変数の集合，$Func$ を関数記号の集合，$Pred$ を述語記号の集合，$Term$ を Var と $Func$ 上で定義されるすべての項の集合，$Atom$ を $Term$ と $Pred$ 上で定義されるすべてのアトムの集合とする．表現は項，アトム，表現の並びまたは表現の集合や multi 集合であるとする．表現 E に現われる全ての変数の集合を $var(E)$ で表わす．代入 θ は Var から $Term$ への mapping であり，$\{V \in Var \mid \theta(V) \neq V\}$ で定義されるその domain は有限であるとする．また，θ の domain を $dom(\theta)$ で表わす．また，代入 θ は $\{V \leftarrow t \mid V \in dom(\theta) \wedge \theta(V) \equiv t\}$ で表されるような assignment の集合と見なすこともできる．その時，$\bigcup_{V \in dom(\theta)} var(\theta(V))$ を θ の range と呼び，$ran(\theta)$ で表す．

E を表現とすると，$E\theta$(または $(E)\theta$) は E に現われる変数 V を同時に項 $\theta(V)$ で置き換えることによって得られる表現を表わしている．2 つの代入 θ と σ の composition を $\theta\sigma$ で表し，[6][12][10] などの定義と同様であるものとする．代入はいつも idempotent(すなはち，$dom(\theta) \cap ran(\theta) = \emptyset$．ここで，$\emptyset$ は空集合を表している．) であるものとし [6]，すべての idempotent な代入の集合を $Subst$ で表す．Var から Var への bijection であるような代入を特に renaming と呼ぶ．すべての renaming の集合を $Renam$ で表す．また $\sigma = \{V \leftarrow t \mid V \in var(E)\} \subseteq \theta$ なる代入 σ を θ の表現 E への制限と呼び，$\theta_{|E}$ と表す．

$\theta_1, \theta_2, \sigma \in Subst$ である時，$Subst$ 上の前順序 $\preceq$ を $\theta_1 \preceq \theta_2$ iff $\exists\sigma(\theta_1\sigma = \theta_2)$ で定義し，instantiation ordering と呼ぶ．また，instantiation ordering $\preceq$ に関する同値関係を $\sim$ で表し，$\theta_1 \sim \theta_2$ iff $\exists\sigma \in Renam(\theta_1\sigma \equiv \theta_2)$ と定義する．また，$\theta_1 \sim \theta_2$ であるとき，$Subst$ の同値関係 $\sim$ による modulo を $Subst_{/\sim}$ で表す．このことにより，$Subst_{/\sim}$ 上の半順序を $Subst$ 上の前順序 $\preceq$ から自然に定義できる．この半順序もまた instantiate ordering と呼び，$\preceq$ と記述することにする．また，θ と同値関係 $\sim$ にある代入の集合 (θ の equivalent class と呼ぶが) を $\theta_\sim$ で表す．特に区別の必要がない時は，単に θ で表す場合もある．

$\preceq$ 上の最小要素は空集合 $\emptyset$ なので (特に空代入と呼ぶこともある)，$\preceq$ 上の最大要素として $\top$ を導入し，$Subst \cup \{\top\}$ を $Subst^\top$ で表すことにしよう．そして，$Subst_{/\sim}$ を $Subst^\top_{/\sim}$ に拡張すると，$(Subst^\top_{/\sim}, \preceq)$ は完備束を形成する．[4]

表現 E_1, E_2 の most general unifier(mgu) $mgu(E_1, E_2)$ で表し，$mgu(E_1, E_2) \equiv \theta$ iff $E_1\theta \equiv E_2\theta$ and $\forall\theta'$ ($E_1\theta' \equiv E_2\theta' \supset \theta \preceq \theta'$ w.r.t. $var(E_1) \cup var(E_2)$) と定義する．代入 θ に対応する unification atom の集合を $Eq(\theta)$ と記述し，$Eq(\theta) = \{X = t \mid (X \leftarrow t) \in \theta\}$ と定義する．unification atom の集合 U を $\{s_1 = t_1, \ldots, s_n = t_n\}$ とすると，$mgu([s_1, \ldots, s_n], [t_1, \ldots, t_n])$ を $mgu(U)$ で表すことにする．

代入 θ_1, θ_2 に対して，$var(\theta_1) \cap var(\theta_2) = dom(\theta_1) \cap dom(\theta_2)$ である時，θ_1 と θ_2 は直交していると呼ぶ．

$(Subst^\top_{/\sim}, \preceq)$ は完備束を形成しているので，$Subst^\top_{/\sim}$ の部分集合は least upper bound(lub) と greatest lower bound(glb) を持つ．これらの lub や glb を求めるアルゴリズムは [4][7][10] 達によってすでに提案されている．[10] においては parallel composition($\uparrow$ で表される) と parallel factorization($\downarrow$ で表される) と呼ばれる $Subst_{/\sim} \times Subst_{/\sim} \rightarrow Subst_{/\sim}$ な操作が提案されている．またそこでは，それらの操作が $\theta_1 \uparrow \theta_2 = lub(\theta_1, \theta_2)$ や $\theta_1 \downarrow \theta_2 = glb(\theta_1, \theta_2)$ の特性があることが示されている．

次に 2 つの代入がお互いに矛盾を引き起こす場合を考えよう．$lub(\theta_1, \theta_2) \equiv \top$ であるとき θ_1 と θ_2 は両立しないと言い，$\theta_1 \not\asymp \theta_2$ で表す．

代入 $\theta_\sim \in Subst_{/\sim}$ と両立しないすべての代入の集合を θ の補代入と呼び，$\overline{\theta}_\sim$ または $\overline{\theta}$ と記述する．$\overline{\theta}_\sim$ は代入 (の equivalent class の) 集合で $\{\theta'_\sim \in Subst_{/\sim} \mid \theta_\sim \not\asymp \theta'_\sim\}$ と定義される．

ここで，[10] で導入された 2 つの操作 parallel composition と parallel factorization を簡単に説明しよう．θ_1 と

θ_2 を代入の equivalent class とする．$\theta_1 \uparrow \theta_2$ は $mgu(Eq(\theta_1) \cup Eq(\theta_2))$ と定義される．また，$\theta_1 \downarrow \theta_2$ は factorization algorithm と呼ばれるアルゴリズムを用いて定義されている．このアルゴリズムは 2 つの代入 θ_1 と θ_2 の異なる binding の部分を繰り返し取り除いていくことによって，最終的には，結果となる代入 η と $theta_1$ と θ_2 とに対応して付随する代入 σ_1 と σ_2 を生成する．これらの代入の間には $\eta\sigma_1 = \theta_1$ and $\eta\sigma_2 = \theta_2$ の関係があることが示されている．ここで，この付随する代入 σ_1, σ_2 を side-代入と呼んでいる．本稿では σ_1 (または σ_2) を特に θ_2 (または θ_1) の θ_1 (or θ_2) からの most general difference(mgd) と呼び，$mgd(\theta_1, \theta_2)$ (または $mgd(\theta_1, \theta_2)$) で表す．

3. Flat Guarded Horn Clauses

3.1 FGHC プログラム

FGHC プログラムは flat guarded clause の集合で，flat guarded clause (または単に clause) は，以下の形をしている．

$p(t_1, \ldots, t_k) :- G_1, \ldots, G_m \mid B_1, \ldots, B_n.(k, m, n \geq 0)$

ここで，$p \in Pred$ で p は k 引数の述語記号，$t_1, \ldots, t_m \in Term$ で $G_1, \ldots, G_m, B_1, \ldots, B_n \in Atom$ である．また，$G_1, \ldots, G_m$ は guard goal と，$B_1, \ldots, B_n$ は body goal と呼ばれる．アトム $p(t_1, \ldots, t_k)$ は head と呼ばれ，head と guard goal とを合わせた部分を guard と，body goal の部分は body と呼ばれる．また，2 つの項を unify するための 2 引数の述語記号 "=" のみが built-in として準備されており，"=" による goal を unification goal と呼ぶ．各 guard goal G_i は unification goal である．

3.2 Flat GHC の操作的意味論

[14] や [11] にしたがって，FGHC の操作的意味論を transition system で定義してみよう．

FGHC プログラム P の transition system を configuration と transition relation を用いて定義する．configuration は $var(B) \cup C \subseteq V$ を満たすような goal の multi 集合 B と (environment と呼ばれる) 変数集合 V 付きの unification アトムの multi 集合 C との対 $\langle B, C : V \rangle$ で表わされる．ここで，C は goal B に関する具体化情報を V はその具体化情報に関わった変数集合を表現している．プログラム P における B_0 が初期ゴールとするとプログラムの実行は以下で定義される configuration 上の 2 項関係 ($\cdot \rightarrow \cdot$ で表すが) で定義される．

(1) もし $\langle B_1, C_1 : V_1 \rangle \rightarrow \langle B_1', C_1' : V_1' \rangle$ ならば，$\langle B_1 \cup B_2, C_1 : V_1 \rangle \rightarrow \langle B_1' \cup B_2, C' : V_1' \rangle$ である．

(2) もし $\exists c \in P((H :- G \mid B) \equiv c\eta)$ で η は $V \cap var(c\eta) \equiv \emptyset$ となるような renaming で $\{A = H\} \cup G \equiv C_g$ でかつ $\models \forall.(C \supset \exists(var(C_g) \setminus var(A))C_g)$ であるならば，$\langle \{A\}, C : V \rangle \rightarrow \langle B, C \cup C_g : V \cup var(c\eta) \rangle$ である．

(3) $\langle \{t_1 = t_2\}, C : V \rangle \rightarrow \langle \emptyset, C \cup \{t_1 = t_2\} : V \cup var(t_1 = t_2) \rangle$

$c_1 \rightarrow c_2$ が成り立つ時 c_1 はプログラム P の元で c_2 に導出されると言う．また，上記 (1)-(3) のどの規則を用いて得られた transition relation かを特に区別したい時にはそれぞれに対応して $\xrightarrow{\cup}$, $\xrightarrow{c}$ や $\xrightarrow{t_1=t_2}$ などの記法を用いる．また，$\rightarrow$ の reflexive で transitive な閉包 (closure) を $\xrightarrow{*}$ で表し，$\xrightarrow{c}$(または $\xrightarrow{t_1=t_2}$) を一回だけ適用し，それ以外は (0 回かも知れないが) $\xrightarrow{\cup}$ を適用することによって得られた閉包 $\xrightarrow{*}$ を $\stackrel{S}{\Rightarrow}$ (または $\stackrel{t_1=t_2}{\Rightarrow}$) と記述する．

goal b と clause c に対して，$\langle \{b\}, C : V \rangle \xrightarrow{c} c_1$ なる environment $C : V$ と configuration c_1 が存在する時，$C : V$ は c による b の導出に対して十分な environment であるという．また，$C : V$ が c による b の導出に対して十分な environment でありどの他の (c による b の導出に対して) 十分な environment $C' : V'$ について $C \preceq C'$ が成り立つ時，C を clause c に対する goal b の most general environment (mge) と呼ぶ．

初期ゴール B_0 のプログラム P の下での実行は初期 configuration $\langle B_0, \emptyset : var(B_0) \rangle$ で始まる (無限に続くかも知れない)transition の連鎖として表現できる．すなはち，$c_1 \rightarrow c_2 \rightarrow \ldots \rightarrow c_i \rightarrow \ldots$ で，$c_1 = \langle B_0, \emptyset : var(B_0) \rangle$ である．この各 configuration $c_i (i \geq 1)$ を (プログラム P の下で) goal B_0 から 到達可能な configuration と呼ばれる．

4 不動点意味論

ここでは，それを構成する各 goal は並行に実行される得るような goal の multi 集合 G の実行を各 goal それぞれの実行の interleaving としてモデル化する．また，そのようなモデルがここで定義される意味関数の最小不動点として特徴付けられることを示す．

4.1 Atom Reaction

θ が代入である時，その代入に '+' または '−' の annotation を付加したものを unit reaction と呼び，θ^+ または θ^- と記述する．また特に，θ^+ を input unit reaction と，また，θ^- を output unit reaction と呼ぶ．また，unit reaction $\delta \equiv \theta^a$ から annotation を取り除いた代入 θ を $|\delta|$ と記述する．すべての input unit reaction の集合 $\{\theta^+ \mid \theta \in Subst\}$ を $UReact^+$ と，すべての output unit reaction の集合 $\{\theta^- \mid \theta \in Subst\}$ を $UReact^-$ と，すべての unit reaction の集合 $UReact^+ \cup UReact^-$ を $UReact$ と記述する．

次に，reactive な動作の特別な状態を表す記号を導入し，停止記号と呼ぶ．$\bot_{suc}, \bot_{rf}, \bot_{uf}, \bot_{dl}$ でこれらはそれぞれ成功，reduction 失敗，unification 失敗，dead lock を表現している．また，$|\bot_{suc}| = |\bot_{rf}| = |\bot_{dl}| = \emptyset$ で，$|\bot_{uf}| = \top$ である．$\{\bot_a \mid a \in \{suc, rf, uf, dl\}\}$ を $\{\bot\}$ と，$UReact \cup \{\bot\}$ を $UReact_\bot$ と記述する．

unit reaction 上の種々の操作や定義を既に定義した代入上の操作を用いて定義する．δ を unit reaction，θ を代入とする．unit reaction の domain や range は，$dom(\delta) \equiv dom(|\delta|)$ や $ran(\delta) \equiv ran(|\delta|)$ と定義される．また，mgu や mgd も，$mgu(\delta, \sigma) \equiv mgu(|\delta|, \sigma)$ や $mgd(\delta, \sigma) \equiv mgd(|\delta|, \sigma)$ や $mgd(\sigma, \delta) \equiv mgd(\sigma, |\delta|)$ と定義される．

$\delta_i \in UReact$ と unit reaction の並び Δ に対して，$\exists \delta_i (1 \preceq i \preceq n \wedge \Delta = \delta_1 \delta_2 \ldots \delta_n)$ の時，$\delta_i \in \Delta$ と記述し，δ_i は Δ にあると呼ぶ．unit reaction の並び Δ にある任意の unit reaction δ_i, δ_j に対して，$\exists \delta_i, \delta_j \in \Delta (\Delta = \delta_1 \ldots \delta_n \wedge 1 \preceq i < j \preceq n)$ の時，$\delta_i \prec \delta_j \in \Delta$ と記述し，Δ において δ_i は δ_j

の前にあると呼ぶ．unit reaction の空の並びを $\Box$ と記述する．

任意の $\delta_i \prec \delta_j \in \Delta$ に対して，$dom(\delta_i) \cap dom(\delta_j) \equiv \emptyset$ でかつ $dom(\delta_i) \cap ran(\delta_j) \equiv \emptyset$ である時，そのような unit reaction の並びを特に reaction sequence と呼ぶ．すべての reaction sequence の集合を $RSeq$ と記述する．$\Delta \in RSeq$ の domain は，$\{V \mid \exists \delta \in \Delta(V \in dom(\delta)) \wedge \forall \delta' \in \Delta(V \notin ran(\delta'))\}$ なる変数の集合で，$dom(\Delta)$ と記述する．また，2 章で定義した代入と代入が直交するという概念もこの domain の定義を用いて代入と reaction sequence との直交，reaction sequence と reaction sequence の直交へと自然に拡張できる．また，$\delta_1\delta_2\ldots\delta_n \in RSeq$ で $\Delta \equiv \delta_2\ldots\delta_n$ とすると，$\delta\cdot\Delta$ もまた $\delta_1\delta_2\ldots\delta_n$ である．

$\Delta \equiv \delta_1\delta_2\ldots\delta_n \in RSeq$ の時，composition $|\delta_1||\delta_2|\ldots|\delta_n|$ を Δ の composition と呼び，$comp(\Delta)$ と記述する．

$A \in Atom$ でかつ $\Delta \in RSeq$ でかつ $dom(\Delta) \subseteq var(A)$ である時，アトム A と reaction sequenceΔ との対を atom reaction と呼び，(A,Δ) と記述する．また，atom reaction (A,Δ) の reaction sequenceΔ の最後に停止記号を付加したものもまた atom reaction と呼ばれる．ここで，すべての atom reaction の集合を $AReact$ であらわす．すなはち，$AReact = \{(A,\Delta\delta) \mid A \in Atom \wedge \Delta \in RSeq \wedge \delta \in UReact_\perp\}$ である．

代入 θ と atom reaction Δ が直交している時，θ と (A,Δ) もまた直交していると言う．また，reaction sequence Δ_1 と Δ_2 が直交している時，atom reaction (A_1,Δ_1) と (A_2,Δ_2) もまた直交していると言う．

atom reaction $(A,\delta_1\delta_2\ldots\delta_n)$ が以下の条件を満たす時，プログラム P に対して correct であると呼ぶ．

(0) $n=0$，すなはち，$\delta_1\delta_2\ldots\delta_n = \Box$ である．

$B_1 = \{A\}$ で $C_1' = \emptyset$ である時，任意の $i(1 \preceq i \preceq n)$ に対して，$mgu(C_{\delta_i}) \sim |\delta_i|$ とすると，

(1) $\delta_i \in UReact^+$ ならば，$C_i' = C_i \cup C_{\delta_i}$ で $B_{i+1} = (B_i \setminus \{H\}) \cup B$ なる transition $\langle B_i, C_i : V_i\rangle \Rightarrow \langle B_{i+1}, C_{i+1}' : V_i \cup var(c\eta)\rangle$ が存在する．ここで，$(H \text{ :- } G \mid B) \equiv c\eta$ で η は renaming である．

(2) $\delta_i \in UReact^-$ ならば，$C_i = C_i'$ で $C_{i+1} = C_i \cup C_{\delta_i}$ で $B_{i+1} = B_i \setminus \{t=s\}$ なる transition $\langle B_i, C_i : V_i\rangle \overset{t=s}{\Rightarrow} \langle B_{i+1}, C_{i+1}' : V_i \cup var(s=t)\rangle$ が存在する．あるいは，

(3) $\delta_n \in \{\perp\}$ ならば，任意の $i(1 \preceq i \preceq n-1)$ に対して，上記 (0)–(1) のいづれかの条件を満たし，かつ，$\delta_n = \perp_{suc}$ の時，$B_i = \emptyset$ で，$\delta_n = \perp_{rf}$ の時，任意の $b \in B_{i-1}$, $(H \text{ :- } G \mid B) \in P$ に対して $mgu(C_{i-1}') \not\approx mgu(\{b = H\} \cup G)$ で，$\delta_n = \perp_{uf}$ の時，ある $t = s \in B_{i-1}$ に対して，$mgu(C_{i-1}') \not\approx mgu(\{t=s\})$ で，$\delta_n = \perp_{dl}$ の時，$\langle B_{n-1}, C_{n-1}' : V_{n-1}\rangle \overset{\delta}{\not\Rightarrow}$ である．

4.2 Topdown 不動点意味論

まず，atom reaction への代入の適用を定義する．θ は代入，$\Delta = \delta_1\delta_2\ldots\delta_n$ は reaction sequence で，θ と Δ は直交しているとする．その時，θ の Δ への適用は，$\sigma_0 = \theta$ で $\sigma_i = mgd(\sigma_{i-1}, \delta_i)$ とした時に，各 $i(1 \preceq i \preceq n)$ に対して $\delta_i' = mgd(\delta_i, \sigma_{i-1})$ なる unit reaction の並び $\delta_1'\delta_2'\ldots\delta_n'$ であり，$(\Delta)\theta$ と記述する．ここで，σ_i と δ_i は互いに直交しているとする．(A,Δ) を atom reaction とすると，代入 θ の (A,Δ) への適用もまた定義できる．それは $(A,\Delta)\theta$ と記述され，$(A\theta,(\Delta)\theta)$ と定義される．

Lemma 4.1 *P を FGHC プログラムとし，(A,Δ) は atom reaction で θ は (A,Δ) と直交する代入であるとする．その時，(A,Δ) は P に関して correct である iff $(A,\Delta)\theta$ は P に関して correct である*

次に，複数の reaction sequence があった時，それらの interleaving を定義する．reaction sequence の集合 $\{\Delta_1, \ldots, \Delta_n\}$ の変数集合 V 上での interleaving は，$int(\Delta_1, \ldots, \Delta_n)_{|V}$ と記述され，以下で定義される unit reaction の並びの集合である．

$\{\delta\cdot\Delta \mid \exists i \in n(\Delta_i = \delta_i\cdot\Delta_i' \wedge$
$(\Delta = \Box \wedge \delta = \perp_{dl}$
if $\delta_i \in UReact^+ \wedge V \subset dom(\delta_i) \vee$
$\Delta = \Box \wedge \delta = \delta_i$ if $\Delta_i \in \{\perp_{rf}, \perp_{uf}, \perp_{dl}\} \vee$
$\Delta = int(\Delta_1''\delta_i, \ldots, \Delta_n''\delta_i)_{|V\delta_i} \wedge \delta = \delta_i$
if $\Delta_i' = \Delta_i''\cdot\perp_{suc} \wedge \forall j \in n(\Delta_j = \Delta_j''\cdot\perp_{suc}) \vee$
$\Delta = int(\Delta_1\delta_i, \ldots, \Delta_i'\delta_i, \ldots, \Delta_n\delta_i)_{|V\delta_i} \wedge$
$\delta = \delta_i$ otherwise)$\}$.

Lemma 4.2 *任意の変数集合 V に対して，もし $\Delta_1, \Delta_2, \ldots, \Delta_n$ がお互いに直交し合う reaction sequence ならば，$\Delta \in int(\Delta_1, \ldots, \Delta_n)_{|V}$ なる unit reaction の並びは reaction sequence である．*

すべての atom reaction の集合 $AReact$ のベキ集合を Den で表わし，これを意味関数の領域とする．プログラム P とゴール G が与えられた時意味関数 $T_{P,G} : Den \to Den$ を以下のように与える．

$T_{P,G}(I) \quad =$
$\{(G,\Box)\} \cup$
$\{(s=t,\theta^-) \mid (s=t,\Box) \in I \wedge \theta = mgu(s,t)\} \cup$
$\{(B_i\theta_g,\Box) \mid \exists (A,\Box) \in I \wedge \exists H \text{ :- } G \mid B \in P$
$\theta_g = mgu(\{A=H\} \cup G) \wedge \exists B_i \in B\} \cup$
$\{(A,\theta_g^+\Delta) \mid \exists (A,\Box) \in I \wedge \exists H \text{ :- } G \mid B \in P$
$\theta_g = mgu(\{A=H\} \cup G) \wedge$
$\forall B_i \in B \exists (B_i\theta_g, \Delta_i) \in I \wedge$
$(B_i\theta_g, \Delta_i)$ はお互いに直交
$\Delta = int(\Delta_1, \Delta_2, \ldots, \Delta_n)_{|var(A)}\}$
$\{(A,\theta_g^+\perp_{rf}) \mid \exists (A,\Box) \in I \wedge$
$\forall H \text{ :- } G \mid B \in P \exists \theta_g \in \overline{\theta}($
$\theta = mgu(\{A=H\} \cup G))\}$

Den は bottom element $\emptyset$ と top element $AReact$ を持つ集合の包含関係 $\subseteq$ において完備束を形成する．

意味関数 $T_{P,G}(I)$ は以下の状況を表す 4 つの atom reaction の集合の union として定義されている．

(1) unification atom $s=t$ は環境に binding $mgu(s,t)$ を加える．

(2) ゴール A はそのサブゴールとして $B_i\sigma\theta_g$ を起動する．

(3) また，ゴール A は $\theta_g^+\Delta$ なる reactive な動作を行なう．すなはち，ゴール A が binding θ_g を得ることによりその各サブゴールの reaction sequence の可能な inter-

leaving の一つである Δ が表現する動作を行なうことを意味している.

(4) ゴール A がそのガード $\{A=H\}\cup G$ を解くことができる binding と incompatible な binding θ_g に遭遇した時には, A は suspend する.

Lemma 4.3 関数 $T_{P,G}$ は連続である.

Lemma4.3 より明らかに $T_{P,G}$ は単調であるので, $T_{P,G}$ は最小不動点 $lfp(T_{P,G})$ を持ち, かつ, $lfp(T_{P,G}) = T_{P,G}\uparrow\omega$ [9] である.

P を FGHC プログラムで G をゴールとすると, $lfp(T_{P,G})$ をプログラム P のゴール G に関する topdown 不動点意味論と呼び, $[\![P]\!]_G$ と記述する.

ここで定義した topdown 不動点意味論と 3.2 章で紹介した操作的意味論の関係について述べる. ここで示したいのは以下の定理である.

Theorem 4.4 *P を FGHC プログラムで G をゴールとすると, atom reaction(G,Δ) がプログラム P の下で correct である iff $lfp(T_{P,G})$ に (G,Δ) がある*

5 簡単な例

ここでは, 以下のプログラムを用いて topdown 不動点意味論について説明する. プログラム Q と P_1 と P_2 があったとしよう.

Q :
```
q(X,Y) :- | X=a, Y=b.
```
P_1 :
```
p(a,Y) :- | q(Y).
q(b).
q(c).
```
P_2 :
```
p(a,Y) :- | q1(Y).
p(a,Y) :- | q2(Y).
q1(b).
q2(c).
```
ここで, $\theta_a = \{X \leftarrow a\}$, $\theta_b = \{X \leftarrow b\}$ $\overline{\theta_{bc}} = \overline{\theta_a} \cup \overline{\theta_b}$ とし, $(A, \overline{\theta}\perp_{rf}) = \{(A, \sigma\perp_{rf}) \mid \sigma \in \overline{\theta}\}$ する.

$[\![Q]\!]_{q(X,Y)}$ は

$\{(q(X,Y), \theta_a^-\theta_b^-\perp_{suc}), (q(X,Y), \theta_b^-\theta_a^-\perp_{suc})\}$

である. (ここで, $(q(X,Y), \Box)$ や $(q(X,Y), \theta_a^-)$ などは省いている.)

また, $[\![P_1]\!]_{p(X,Y)}$ は

$\{(p(X,Y), \theta_a^+\theta_b^+\perp_{suc}), (p(X,Y), \theta_a^+\theta_c^+\perp_{suc}),$
$(p(X,Y), \theta_a^+\overline{\theta_{bc}}^+\perp_{rf}), (p(X,Y), \overline{\theta_a}^+\perp_{rf}),$

である.

一方, $[\![P_2]\!]_{p(X,Y)}$ は

$\{(p(X,Y), \theta_a^+\theta_b^+\perp_{suc}), (p(X,Y), \theta_a^+\overline{\theta_b}^+\perp_{rf}),$
$(p(X,Y), \theta_a^+\theta_c^+\perp_{suc}), (p(X,Y), \theta_a^+\overline{\theta_c}^+\perp_{rf}),$
$(p(X,Y), \overline{\theta_a}^+\perp_{rf})\}$

である.

ここに, `PQ : pq :- p(X,Y), q(X,Y).` を加えると, $[\![PQ\cup P_2\cup Q]\!]_{pq}$ は $[\![Q]\!]_{q(X,Y)}$ の 1 つ目と $[\![P_2]\!]_{p(X,Y)}$ の 4 つ目の atom reaction から reduction 失敗 (すなはち, deadlock) の可能性が検出される. しかし, $[\![PQ\cup P_1\cup Q]\!]_{pq}$ にはその可能性はない.

6 おわりに

Flat GHC プログラムの不動点意味論を定義し, 操作的意味論との関係を議論した. この意味論には「プログラム解析に用いられる」という性質上、通常の意味論より抽象度の低い意味論が要求される.

References

[2] Codish, M., J. Gallagher, "A Semantic Basis for the Abstract Interpretation of Concurrent Logic Programs", Technical Report CS89-26, November, 1989.

[5] Falaschi, M., G. Levi, M. Martelli, C. Palamidessi, "A Model-theoretic Reconstruction of the Operational Semantics of Logic Programs", Università di Pisa, Technical Report TR-32/89, 1989.

[7] Lassez, J. L., M. J. Maher, and K. Marriott, "Unification Revised", Proc. of the Workshop on Logic and Data Bases, Minker, J. (ed.), 1987.

[8] Levi, G., "A New Declarative Semantics of Flat Guarded Horn Clauses", Technical Report, ICOT, 1988.

[9] Lloyd, J.W., "Foundation of Logic Programming", Second, Extended Edition, Springer-Verlag, 1087.

[11] Saraswat, A. V., "Concurrent Constraint Programming", Proc. of the 17th POPL, pp 232-245, 1990.

[10] Palamidessi, C., "Algebraic Properties of Idempotent Substitutions", Proc. of the 17th ICALP, pp 386-399, 1990.

[12] Ueda, K. and K. Furukawa, "Transformation Rules for GHC Programs", Proc. of the International Conference on FGCS'88, pp.582-591, Tokyo, 1988.