

相互通信逐次型プロセス系の検証

村上 昌己 稲垣 康善
名古屋大学 工学部

1. まえがき

Hoare⁽⁷⁾ が提案した相互通信逐次型プロセス系 (CSP) は, 共有変数を含まず, プロセス間の同期及び情報の交換を通信命令によって行う並列計算のモデルで, 最近 INMOS グループによって開発された並列処理言語 occam⁽¹⁰⁾ のモデルになる等, 注目を集めている.

本稿では, まず CSP の状態集合上の二項関係及び計算履歴集合による形式的意味論を与え, CSP の部分的正当性, デッドロックフリー性の検証体系を提案する.

2. CSP のシンタックス

CSP のプログラムは空文 Λ , 割当文 $x := \tau$, 防護, 通信命令, 及び skip 文からなる. 通信命令とは, 入力命令 $P?x$ 及び出力命令 $Q!\tau$ のいずれかである. ここで, P は発信元, Q は送付先でいずれもプロセス名である. 通信命令の対 C_k, C_h の実行を $C_k \leftrightarrow C_h$ で表す.

定義 1 ステートメント (以下「文」と呼ぶ) とは, 防護の直後に通信命令または skip 文を含む被防護命令 (guarded command) の並びである.

文 $A_1, \dots, A_k$ が次の条件 i), ii), iii) を満たすとき, 各文をプロセスとよぶ.

- i) 各 A_i は P_i という名前をもつ. 以下ではプロセスの名前とそれを表す記号を同一視して用いる.
- ii) プロセス間の共有変数はない. すなわち $i \neq j$ ならば P_i, P_j の両方に現れる変数はない.
- iii) 各 P_i に現れる通信命令はいずれも通信相手として他のプロセスの名前を指定している.

$P_1, \dots, P_n$ がプロセスであるとき, $P_1 \parallel \dots \parallel P_n$ を CSP という.

3. CSPのセマンティクス(1)

まず, CSPの入出力関係のみに着目したセマンティクスを導入する.
定義2 $CSP, P \equiv P_1 \parallel \dots \parallel P_n$ の状態とは各プロセスのすべての変数の値の組である. CSP, P のとりうるすべての状態の集合を U_P で表す.

以下では, 状態 i で述語 B が成り立つことを $i \models B$, 成り立たないことを $i \not\models B$ で表す.

定義3 プログラム P の各割当文 $x := \tau$ および通信命令の対 C_k, C_h について U_P 上の二項関係 $m(x := \tau), m(C_k \leftrightarrow C_h)$ を次のように定義する.

$m(x := \tau) = \{(i, j) \mid j \text{ は, } x \text{ の値を状態 } i \text{ での}$

τ の値に更新して得られる状態}\}.

$m(C_k \leftrightarrow C_h)$ は, あるプロセス P, Q に対して, C_k がプロセス Q の命令 $P?x$, かつ C_h がプロセス P の命令 $Q! \tau$ の場合のみ定義され,
 $m(C_k \leftrightarrow C_h) = m(x := \tau)$.

定義4 $CSP, P \equiv P_1 \parallel \dots \parallel P_n$ に対して, U_P 上の二項関係 $m(P_1 \parallel \dots \parallel P_n)$ を次のように定義する.

- i) $m(A \parallel \dots \parallel A) = \{(i, i) \mid i \in U_P\}$.
- ii) $x := \tau$ が P_i の命令で, $(i, i') \in m(x := \tau)$,
 かつ, $(i', j) \in m(P_1 \parallel \dots \parallel P_i \parallel \dots \parallel P_n)$ のとき,
 $(i, j) \in m(P_1 \parallel \dots \parallel x := \tau; P_i \parallel \dots \parallel A_n)$.
- iii) $\text{if } B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \text{ fi}$ の $C_1, \dots, C_k$ が, いずれも通信命令で, その通信相手のプロセスが全て Λ のとき,
 $m(A_1 \parallel \dots \parallel \text{if } B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \text{ fi}; A_k \parallel \dots \parallel A_n) = \emptyset$.
- iv) $\text{do } B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \text{ od}$ のすべての $h (1 \leq h \leq k)$ で,
 $i \not\models B_h$ または C_h の相手が Λ であるとき (以下ではこの条件を LE_m であらわす),

$(i, j) \in m(A_1 \parallel \dots \parallel A_m \parallel \dots \parallel A_n)$ ならば

$(i, j) \in m(A_1 \parallel \dots \parallel \text{do } B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \text{ od}; A_m \parallel \dots \parallel A_n)$.

v) $(i', j) \in m(A_1 \parallel \dots \parallel S_k; A_k \parallel \dots \parallel S_l; A_l \parallel \dots \parallel A_n)$,

$$[B_1; C_1 \rightarrow S_1 \square \cdots \square B_k; C_k \rightarrow S_k] ; P_i,$$

と書く。

また, P_i/k を次のように定義する.

$$(\text{do } B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \square \dots \square B_l; C_l \rightarrow S_l \text{ od}; P_i')/k$$
$$\equiv S_k ; \text{ do } B_1 ; C_1 \rightarrow S_1 \square \cdots \square B_k ; C_k \rightarrow S_k$$
$$\square \dots \square B_l; C_l \rightarrow S_l \text{ od}; P_i'.$$
$$(i f \ B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \square \dots \square B_l; C_l \rightarrow S_l \ f i; P_i')/k$$
$$\equiv S_k; P_i'.$$

以上の記法のもとで

$$P \equiv P_1 \parallel \dots \parallel [\dots \square B_k; C_k \rightarrow S_k \square \dots] ; P_m', \parallel$$
$$\dots \parallel [\dots \square B_L; C_L \rightarrow S_L \square \dots] ; P_{s'} \parallel \dots \parallel P_n$$

に対して、

$i \in I$, $i \models B_k \wedge B_l$ かつ $\mu(C_k, C_l)$ ならば,

$$(i, i') \in m(C_k \leftrightarrow C_l) \dots \dots \dots \times$$

のとき、

$$\langle i', e \rangle w \in H(P_1 \parallel \dots \parallel [\dots \square B_k; C_k \rightarrow S_k \square \dots]) ; P_m' / k$$
$$\parallel \dots \parallel \quad [\dots \square B_L ; C_L \rightarrow S_L \square \dots] \quad ; P_s' / L$$
$$\| \dots \| P_n, \{i'\})$$

ならば、

$$\langle i, C_k \leftrightarrow C_l \rangle \langle i', e \rangle \quad w \in H (P_1 \parallel \dots \parallel [\dots \Box B_k ; C_k \rightarrow S_k \Box \dots] ;$$
$$P_m' \parallel \dots \parallel [\dots \square B_i; C_i \rightarrow S_i \square \dots] ; P_s' \parallel \dots \parallel P_n, \quad I)$$

である。

(vii) $P \equiv \Lambda \parallel \cdots \parallel \Lambda$ でなく, かつ, $i \in I$ について (ii) ~ (vi) の

どの※の条件もあてはまらないとき, $\langle i, j \rangle \in H(P, I)$.

5. CSPの性質の検証

以下では前節で定義したセマンティクスをもとにして，CSPの性質の検証するための公理系について述べる．本稿で扱うCSPの性質は次のようなものである．

定義 7

(1) 部分的正当性 …… プログラム P が入力条件 Φ と出力条件 Ψ について部

分的に正当であるとは、 Φ の成り立つ状態から P を実行して、停止すれば常に Ψ が成り立つことを言う。すなわち、任意の

$(i, j) \in m(P)$ について、 $i \models \Phi$ ならば $j \models \Psi$ である。

(2) デッドロックフリー性 ……あるCSPプログラムが状態 i でデッドロックであるとは、少なくともひとつ A でないプロセスが存在し、任意の A でないプロセス P_k について、ステートメントの先頭がif文かdo文で $i \models LE_k$ かつ i で真となる防護に続く命令がすべて通信命令で、かつこれらの通信命令のどの対 C_l, C_k についても $\mu(C_l, C_k)$ がなりたたないことをいう。

あるCSPプログラム $P \equiv P_1 \parallel P_2 \parallel \dots \parallel P_n$ が入力条件 Φ のもとでデッドロックフリーであるとは、 Φ の成り立つ任意の状態 i から P を実行したとき、途中で通るいかなる状態でもプログラムの残りの部分がデッドロックにならないことをいう。すなわち、任意の $w \in H(P, \{i \mid i \models \Phi\})$ について w の最後が $\langle j, \perp \rangle$ でないことをいう。

本稿では P の Φ, Ψ に関する部分的正当性を $\Phi \{P\} \Psi$ とあらわす。また P が Φ, Ψ について部分的正当かつ Φ についてデッドロックフリーであることを、 $\Phi [P] \Psi$ とあらわす。

並列プログラムの部分的正当性を検証する体系の多くは、個々のプロセスについてそれぞれ独立にある性質を証明し、かつ各プロセス間にある条件が成立するとき、それらの性質から全体の正当性がいえる、といった推論の手順をふむ。文献(1)のCSPに関する証明体系でも同様である。しかしこのような手順では推論の過程が、実際の実行と結びつきにくく、証明がわかりづらいものとなる。またこのような証明の方法ではデッドロックフリー性を同時に証明することは困難である。

本稿でのべる推論方法は前節で状態集合上の二項関係あるいは計算履歴集合を定義した手順と同様に、文献(3)のcentralized approachを応用したものである。この方法では、部分的正当性とデッドロックフリー性を、同時に扱うことができる。以下でCSPの部分的正当性の検証体系を示し、その無矛盾性を証明する。続いて推論規則の一部を補足してデッドロックフリー性を同時に検証する体系を示す。

5.1 CSPの部分的正当性

CSPの部分的正当性を証明する体系は次のようなものである．まず推論規則をしめす．

I 1)

$$\frac{\Gamma}{\Phi \{P\} \Psi}$$

ここで、 Γ は論理式の並びで次の i) ~ iv) のあてはまるものを全てふくむ．

i) P が $P_1 \dots \parallel x := \tau; P_1' \parallel \dots \parallel P_n$ のとき、

$$\Phi \{x := \tau\} X, X \{P_1 \dots \parallel P_1' \parallel \dots \parallel P_n\} \Psi$$

ii) P が $P_1 \parallel \dots \parallel \underline{\text{if}} \dots \square B_k; C_k \rightarrow S_k \square \dots \underline{\text{fi}}; P_m' \parallel \dots \parallel \underline{\text{if}} \dots \square B_l; C_l \rightarrow S_l \square \dots \underline{\text{fi}}; P_s' \parallel \dots \parallel P_n$

のとき、

$$\Phi \wedge B_k \wedge B_l \{C_k \leftrightarrow C_l\} X,$$

$$X \{P_1 \parallel \dots \parallel S_k; P_m' \parallel \dots \parallel S_l; P_s' \parallel \dots \parallel P_n\} \Psi$$

iii) P が

$$P_1 \parallel \dots \parallel \underline{\text{if}} \dots \square B_k; \text{skip} \rightarrow S_k \square \dots \underline{\text{fi}}; P_m' \parallel \dots \parallel P_n$$

のとき、

$$\Phi \wedge B_k \{P_1 \parallel \dots \parallel S_k; P_m' \parallel \dots \parallel \dots \parallel P_n\} \Psi$$

iv) P が

$$P_1 \parallel \dots \parallel \underline{\text{do}} B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \underline{\text{od}}; P_m' \parallel \dots \parallel P_n$$

のとき、

$$\Phi \wedge L E_m \{P_1 \parallel \dots \parallel P_m' \parallel \dots \parallel P_n\} \Psi,$$

$$\Phi \wedge \neg L E_m \{P_1 \parallel \dots \parallel$$

$$\underline{\text{if}} B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \underline{\text{fi}};$$

$$\underline{\text{do}} B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \underline{\text{od}};$$

$$P_m' \parallel \dots \parallel P_n\} \Psi$$

I 2)

$$\frac{\Phi \supset \Psi (x/\tau)}{\Phi \{x := \tau\} \Psi}$$

I 3)

$$\frac{\Phi \supset \Psi (x/\tau)}{\Phi \{C_k \leftrightarrow C_l\} \Psi}$$

ここで $\Psi(x/\tau)$ は Ψ での x のすべての自由な出現に τ を代入したもの.

I 4)

$$\frac{\Phi \supset \Phi' \quad \Phi' \quad \{P\} \Psi}{\Phi \quad \{P\} \Psi}$$

I 5)

$$\frac{\Phi \quad \{P\} \Phi' \quad \Phi' \supset \Psi}{\Phi \quad \{P\} \Psi}$$

I 6)

$$\frac{\Phi \supset \Psi}{\Phi \quad \{\Lambda \parallel \dots \parallel \Lambda\} \Psi}$$

公理として次のようなものがある.

A 1) 任意の P , 任意の Ψ について,

$$\text{false} \quad \{P\} \Psi.$$

A 2) P が推論規則 1) の i) ... iv) のいずれもあてはまらないとき, 任意の Φ, Ψ について,

$$\Phi \quad \{P\} \Psi.$$

A 3) 自然数論の全ての定理.

定義 8 論理式 $\Phi \quad \{P\} \Psi$ について, 次のような木をその証明図という.

(1) 根のラベルは $\Phi \quad \{P\} \Psi$ である.

(2) すべての内部節点のラベルは, その子孫のラベルから直接推論される.

(3) すべての葉は, ある内部節点で使われた推論規則 I1) vi) の前提

$\Phi \wedge \neg L E_m \quad \{P_1 \parallel \dots \parallel \text{if } B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \text{ fi}; \text{do } B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \text{ od}; P_m' \parallel \dots \parallel P_n\} \Psi$ の子孫であり, かつそのラベルは $\Phi \quad \{P_1 \parallel \dots \parallel \text{do } B_1; C_1 \rightarrow S_1 \square \dots \square B_k; C_k \rightarrow S_k \text{ od}; P_m' \parallel \dots \parallel P_n\} \Psi$ であるか, または公理をラベルとしてもつ.

以上のようにして, 定められた体系について次のことが成り立つ.

定理 1

CSP プログラム P について, $\Phi \quad \{P\} \Psi$ の証明図が存在するとき, 任意の $(i, j) \in m(P)$ について, $i \models \Phi$ ならば

$$j \models \Psi.$$

5. 2 デッドロックフリー性

前節で述べた部分的正当性の証明体系を基にデッドロックフリー性を

も検証できる体系を与える。

推論規則を次のように与える。

推論規則 J 1)

$$\frac{\Gamma}{\Phi \text{ 【 P 】 } \Psi}$$

ここで、 Γ は論理式の並びで、次のあてはまるものをすべてふくむ。

i) $\sim$ v) 部分的正当性の推論規則 I 1) の i) から v) の {} を全て【】に置き換えて得られる式の並び。

vi) P の Δ でないすべてのプロセスについて、その先頭が if 文であるとき、

$$\Phi \supset (\bigvee_{h,l} (B_h \wedge B_l)) \vee \bigvee_l B_l$$

ここで $\bigvee_{h,l}$ は $\mu(C_l, C_h)$ なる l, h について、また $\bigvee_l$ は C_l が skip 文である l についての論理和でを表す。

推論規則 J 2) $\sim$ J 6) として推論規則 I 1) $\sim$ I 6) ですべての {} を【】に置き換えたものを採用する。またこの体系の公理 A' 1) $\sim$ A' 3) は部分的正当性の体系の公理 A 1) $\sim$ A 3) の {} を【】に置き換えて得られるものとする。又、証明図の定義は部分的正当性の体系と同じである。

この体系について、つぎの定理が成立する。

定理 2

CSP プログラム P について、 $\Phi \text{ 【 P 】 } \Psi$ の証明図が存在するとき、任意の $(i, j) \in m(P)$ について、 $i \models \Phi$ ならば、

$$j \models \Psi$$

かつ、任意の $w \in H(P, \{i \mid i \models \Phi\})$ について、

w の最後は $\langle j, \perp \rangle$ でない。

6. あとがき

本稿では CSP のセマンティクスとして状態集合の上の二項関係、及び計算履歴集合の概念を定義し、それらを基に CSP の部分的正当性及びデッドロックフリー性の検証体系を与えた。

謝辞 日頃御指導下さる豊橋技術科学大学本多波雄学長、名古屋大学福

村晃夫教授，並びに御討論下さる研究室の皆様に感謝致します。

文献

- 1) Apt, K.R., N. Francez and W.P. de Roever, "A Proof System for Communicating Sequential Processes" ACM Trans. on Programming Language and Systems vol.2, no.3 (1980)
- 2) Brookes, S.D., "A Model for Communicating Sequential Processes" D.Phil. Dissertation, Oxford Univ., Oxford, England, (1983)
- 3) Elrad, T. and N. Francez, "A Weakest Precondition Semantics for Communicating Sequential Processes", IBM Reserch Report (1982)
- 4) Francez, N., C.A.R. Hoare, N. D. Lehmann and W. P. de Roever, "Semantics of Nondeterminism, Concurrency and Communication" JCSS vol.19 (1979).
- 5) Francez, N., D. Lehmann and A. Pnueli, "A Linear History Semantics for Distributed Language: Extended Abstract", Proc. of FOCS, ACM (1980)
- 6) Harel, D., "First-Order Dynamic Logic", Lecture Notes in Computer Sci. no.68 (1979)
- 7) Hoare, C.A.R. "Communicating Sequential Processes", CACM. vol.21, no.8 (1978) 666-676
- 8) Levin, G.M., D. Gries, "A Proof Technique for Communicating Sequential Processes" Acta Informatica, vol.15, 281-302 (1981)
- 9) 村上，稲垣，"相互通信プロセス系のKripke的セマンティクス"，信学技報，AL84-53，(1985-01)
- 10) 尾内，"並行プロセス記述言語 occum"，bit, vol.16, no.11, 共立出版 (1984)
- 11) Soundararajan, N., "Axiomatic Semantics of Communicating Sequential Processes" ACM Trans. on Programming Languages and Systems, vol.6, no.4 (1984)