数理解析研究所講究録
第556巻 1985年 187-196



# 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Ａ，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!τ 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 $A_1, \cdots, A_k$ 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．

$P_1, \cdots, P_n$ がプロセスであるとき，$P_1 \| \cdots \| P_n$ 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，$P \equiv P_1 \| \cdots \| P_n$ 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は，ｘの値を状態ｉでの
$\tau$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，$P \equiv P_1 \| \cdots \| P_n$に対して，$U_P$上の二項関係$m(P_1 \| \cdots \| P_n)$を次のように定義する．

i) $m(\Lambda \| \cdots \| \Lambda) = \{(i,i) \mid i \in U_P\}$．

ii) $x := \tau$が$P_t$の命令で，$(i,i') \in m(x := \tau)$，
かつ，$(i',j) \in m(P_1 \| \cdots \| P_t \| \cdots \| P_n)$のとき，
$(i,j) \in m(P_1 \| \cdots \| x := \tau ; P_t \| \cdots \| A_n)$．

iii) $\underline{\text{if}} B_1 ; C_1 \rightarrow S_1 \square \cdots \square B_k ; C_k \rightarrow S_k \underline{\text{fi}}$の$C_1, \cdots, C_k$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，
$m(A_1 \| \cdots \| \underline{\text{if}} B_1 ; C_1 \rightarrow S_1 \square \cdots \square B_k ; C_k \rightarrow S_k \underline{\text{fi}} ; A_k \| \cdots \| A_n) = \phi$．

iv) $\underline{\text{do}} B_1 ; C_1 \rightarrow S_1 \square \cdots \square B_k ; C_k \rightarrow S_k \underline{\text{od}}$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），

$(i,j) \in m(A_1 \| \cdots \| A_m \| \cdots \| A_n)$ならば
$(i,j) \in m(A_1 \| \cdots \| \underline{\text{do}} B_1 ; C_1 \rightarrow S_1 \square \cdots \square B_k ; C_k \rightarrow S_k \underline{\text{od}} ; A_m \| \cdots \| A_n)$．

v) $(i',j) \in m(A_1 \| \cdots \| S_k ; A_k \| \cdots \| S_t ; A_t \| \cdots \| A_n)$，

かつ，

あるiで$i \models B_k \wedge B_l$，かつ入力命令$C_k$と出力命令$C_l$（あるいはその逆）が互いのプロセスを通信相手として指定しているとき（この条件を$\mu(C_k, C_l)$と書く），

$(i,i') \in m(C_k \leftrightarrow C_l)$ ならば，

$(i,j) \in m(A_1 \parallel \cdots \parallel$ if$\cdots \square B_k;C_k \rightarrow S_k \square \cdots$fi; $A_k \parallel \cdots \parallel$ if$\cdots \square B_l;C_l \rightarrow S_l \square \cdots$fi; $A_l \parallel \cdots \parallel A_n)$.

vi) $(i,j) \in m(A_1 \parallel \cdots \parallel S_k;A_k \parallel \cdots \parallel A_n)$ かつ $i \models B_k$ のとき，

$(i,j) \in m(A_1 \parallel \cdots \parallel$ if$\cdots \square B_k;$skip$\rightarrow S_k \square \cdots$fi$;A_k \parallel \cdots \parallel A_n)$.

vii) $(i,j) \in m(A_1 \parallel \cdots \parallel$ if$B_1;C_1 \rightarrow S_1 \square \cdots \square B_k;C_k \rightarrow S_k$ fi;

do$B_1;C_1 \rightarrow S_1 \square \cdots \square B_k;C_k \rightarrow S_k$ od;

$A_l \parallel \cdots \parallel A_n)$ であるとき，

〔すなわち，if文とそれに続くdo文の内部が全く同じであるとき〕

$(i,j) \in m(A_1 \parallel \cdots \parallel$ do$B_1;C_1 \rightarrow S_1 \square \cdots \square B_k;C_k \rightarrow S_k$od;

$A_l \parallel \cdots \parallel A_n)$.

## 4．ＣＳＰのセマンティクス(2)

次にプログラムの実行中に通る状態も記録したセマンティクスとして，ＣＳＰの計算履歴集合を定義する．

定義5　次の，

$\langle i_1, e_1\rangle\langle i_2,e_2\rangle\cdots\langle i_{n-1},e_{n-1}\rangle\langle i_n, \varepsilon\rangle$，または，

$\langle i_1, e_1\rangle\langle i_2,e_2\rangle\cdots\langle i_{n-1},e_{n-1}\rangle\langle i_n, \perp\rangle$

のような形の系列を，計算履歴という．ただし，$e_k$は割り当て文または$\mu(C_k, C_l)$をみたす通信命令の対$C_k \leftrightarrow C_l$で，$0 < h \leq n$について$(i_{h-1}, i_h) \in m(e_h)$.

定義6　ＣＳＰプログラム$P \equiv P_1 \parallel P_2 \parallel \cdots \parallel P_n$，と状態集合$I \subseteq U_P$に対して，計算履歴の集合Ｈ（Ｐ，Ｉ）を次のように定義する．

（ｉ）$H(\Lambda \parallel \cdots \parallel \Lambda, I) = \{\langle i, \varepsilon\rangle \mid i \in I\}$.

（ii）$P \equiv P_1 \parallel \cdots \parallel x := \tau; P_l' \parallel \cdots \parallel P_n$ のとき，

$(i,i') \in m(x := \tau)$，$i \in I$ ……※

かつ，$\langle i', e\rangle$で始まる計算履歴$\langle i', e\rangle w$が，

$H(P_1 \| \cdots \| P_l' \| \cdots \| P_n, \{i'\})$ の元であるとき，

$\langle i, x := \tau \rangle \langle i', e \rangle$ wは，

$H(P_1 \| \cdots \| x := \tau; P_l' \| \cdots \| P_n, I)$

の元である．

(iii) $P \equiv P_1 \| \cdots \| \underline{if} \cdots \square B_k; skip \rightarrow S_k \square \cdots \underline{fi}; P_m' \| \cdots \| P$

のとき，　$i \in I$ で $i \models B_k$ ……※

であり，かつ，

$w \in H(P_1 \| \cdots \| S_m; P_m' \| \cdots \| P_n, \{i\})$

ならば，wは

$H(P_1 \| \cdots \| \underline{if} \cdots \square B_k; skip \rightarrow S_k \square \cdots \underline{fi}; P_m' \| \cdots \| P_n, I)$

の元である．

(iv) $P \equiv P_1 \| \cdots \| \underline{do} \cdots \square B_k; skip \rightarrow S_k \square \cdots \underline{od}; P_m' \| \cdots \| P_n$

のとき，　$i \in I$ で $i \models B_k$ ……※

であり，かつ，

$w \in H(P_1 \| \cdots \| S_k; \underline{do} \cdots \square B_k; skip \rightarrow S_k \square \cdots \underline{od}; P_m' \| \cdots \| P_n, \{i\})$ ならば，

$w \in H(P_1 \| \cdots \| \underline{do} \cdots \square B_k; skip \rightarrow S_k \square \cdots \underline{od}; P_m' \| \cdots \| P_n, I)$

である．

(v) $P \equiv P \| \cdots \| \underline{do} B_1; C_1 \rightarrow S_1 \square \cdots \square B_k; C_k \rightarrow S_k \underline{od}; P_m' \| \cdots \| P_n$ の場合，

$i \in I$　かつ　$i \models LE_m$ ……※

のとき，

$w \in H(P_1 \| \cdots \| P_m' \| \cdots \| P_n, \{i\})$ ならば

$w \in H(P_1 \| \cdots \| \underline{do} B_1; C_1 \rightarrow S_1 \square \cdots \square B_k; C_k \rightarrow S_k \underline{od}; P_m' \| \cdots \| P_n, I)$.

(vi) プロセス $P_i$ が，

$\underline{do}\ B_1; C_1 \rightarrow S_1 \square \cdots \square B_k; C_k \rightarrow S_k\ \underline{od}; P_i'$

または

$\underline{if}\ B_1; C_1 \rightarrow S_1 \square \cdots \square B_k; C_k \rightarrow S_k\ \underline{fi}; P_i'$

のとき，

$[B_1;C_1 \to S_1 \square \cdots \square B_k;C_k \to S_k\ ]\ ;P_i'$

と書く．

また，$P_i/k$を次のように定義する．

$(\underline{do} B_1;C_1 \to S_1 \square \cdots \square B_k;C_k \to S_k \square \cdots \square B_l;C_l \to S_l \underline{od};P_i')/k$

$\equiv S_k\ ;\ \underline{do}\ B_1;C_1 \to S_1 \square \cdots \square B_k;C_k \to S_k$

$\square \cdots \square B_l;C_l \to S_l\ \underline{od};P_i'$.

$(\underline{if}\ B_1;C_1 \to S_1 \square \cdots \square B_k;C_k \to S_k \square \cdots \square B_l;C_l \to S_l\ \underline{fi};P_i')/k$

$\equiv S_k;P_i'$.

以上の記法のもとで

$P \equiv P_1 \parallel \cdots \parallel [\cdots \square B_k;C_k \to S_k \square \cdots]\ ;P_m' \parallel$

$\cdots \parallel [\cdots \square B_l;C_l \to S_l \square \cdots]\ ;P_s' \parallel \cdots \parallel P_n$

に対して，

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

$(i,i') \in m(C_k \leftrightarrow C_l)$ ……※

のとき，

$\langle i',e\rangle w \in H(P_1 \parallel \cdots \parallel [\cdots \square B_k;C_k \to S_k \square \cdots]\ ;P_m'/k$

$\parallel \cdots \parallel [\cdots \square B_l;C_l \to S_l \square \cdots]\ ;P_s'/l$

$\parallel \cdots \parallel P_n\ ,\ \{i'\})$

ならば，

$\langle i,C_k \leftrightarrow C_l\rangle\langle i',e\rangle\ w \in H(\ P_1 \parallel \cdots \parallel [\cdots \square B_k;C_k \to S_k \square \cdots]\ ;$

$P_m' \parallel \cdots \parallel [\cdots \square B_l;C_l \to S_l \square \cdots]\ ;P_s' \parallel \cdots \parallel P_n,\ I)$

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，$\langle i,\perp\rangle \in H(P, I)$．

## 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）∈m（P）について，i⊨Φならばj⊨Ψ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$i \not\models LE_k$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$C_l$，$C_h$についても
$\mu$（$C_l$，$C_h$）がなりたたないことをいう．

あるCSPプログラム$P \equiv P_1 \parallel P_2 \parallel \cdots \parallel P_n$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∈H（P，{i | i⊨Φ}）についてwの最後が<j，⊥>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Φ｛P｝Ψ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，Φ【P】Ψ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など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．

Ｉ１）

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

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$P_1 \cdots \| \ x := \tau\ ;\ P_\iota' \ \| \cdots \| \ P_n$のとき，

$\Phi\ \{x := \tau\}\ X,\quad X\ \{P_1 \cdots \| \ P_\iota' \ \| \cdots \| \ P_n\}\ \Psi$

ｉｉ）Ｐが

$P_1 \| \cdots \| \underline{\text{if}} \cdots \square\ B_k ; C_k \rightarrow S_k \square \cdots \underline{\text{fi}} ; P_m' \|$

$\cdots \| \underline{\text{if}} \cdots \square\ B_\iota ; C_\iota \rightarrow S_\iota \square \cdots \underline{\text{fi}} ; P_s' \| \cdots \| P_n$

のとき，

$\Phi \wedge B_k \wedge B_\iota\ \{C_k \leftrightarrow C_\iota\}\ X,$

$X\ \{P_1 \| \cdots \| S_k\ ;\ P_m' \| \cdots \| S_\iota\ ;\ P_s' \| \cdots \| P_n\}\ \Psi$

ｉｉｉ）Ｐが

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

のとき，

$\Phi \wedge B_k\ \{P_1 \| \cdots \| S_k\ ;\ P_m' \| \cdots \| \cdots \| P_n\}\ \Psi$

ｉｖ）Ｐが

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

のとき，

$\Phi \wedge \mathrm{LE}_m\ \{P_1 \| \cdots \| P_m' \| \cdots \| P_n\}\ \Psi,$

$\Phi \wedge \neg \mathrm{LE}_m\ \{P_1 \| \cdots \|$

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

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

$P_m' \| \cdots \| P_n\}\ \Psi$

Ｉ２）

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

Ｉ３）

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

ここでΨ〔x/τ〕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）

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

I 5）

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

I 6）

$$\frac{\Phi \supset \Psi}{\Phi \ \{A \parallel \cdots \parallel A\}\ \Psi}$$

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，

false {P} Ψ．

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，

Φ {P} Ψ．

A 3）自然数論の全ての定理．

定義8　論理式Φ {P} Ψ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Φ {P} Ψ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

$\Phi \wedge \neg \mathrm{LE}_m \{P_1 \parallel \cdots \parallel \underline{\mathrm{if}}\ B_1; C_1 \to S_1 \ \square \cdots \square\ B_k; C_k \to S_k\ \underline{\mathrm{fi}};\ \underline{\mathrm{do}}\ B_1; C_1 \to S_1 \ \square \cdots \square\ B_k; C_k \to S_k\ \underline{\mathrm{od}};\ P_m' \parallel \cdots \parallel P_n\}\ \Psi$

の子孫であり，かつそのラベルは$\Phi \{P_1 \parallel \cdots \parallel \underline{\mathrm{do}}\ B_1; C_1 \to S_1 \ \square \cdots \square\ B_k; C_k \to S_k\ \underline{\mathrm{od}};\ P_m' \parallel \cdots \parallel P_n\}\ \Psi$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，Φ {P} Ψの証明図が存在するとき，任意の（i，j）∈m（P）について，i ⊨ Φならば

j ⊨ Ψ．

## 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Ｊ１）

$$\frac{\Gamma}{\Phi\ 【P】\ \Psi}$$

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）～ｖ）部分的正当性の推論規則Ｉ１）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Ａ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Ｊ２）～Ｊ６）として推論規則Ｉ１）～Ｉ６）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）～Ａ’３）は部分的正当性の体系の公理Ａ１）～Ａ３）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，Φ【Ｐ】Ψの証明図が存在するとき，任意の（ｉ，ｊ）∈ｍ（Ｐ）について，ｉ⊨Φならば，

$$j \models \Psi$$

かつ，任意のｗ∈Ｈ（Ｐ，｛ｉ｜ｉ⊨Φ｝）について，

ｗの最後は<ｊ，⊥>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 Semmantics 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)