

CCS に基づく擬似非同期システムモデル†

高 原 厚^{††} 南 谷 崇^{††}

ハードウェア・アルゴリズムがハードウェアとして詳細化される以前に関数的に正しく動作することを確認する場合、事象生起の因果関係に基づいた非同期モデルによる表現が自然である。しかし、実際のシステムは同期システムとして実現されることが多く、同期システムを非同期モデルで解析する場合、同期システムでは考える必要のない状態までも生成し、解析を一層複雑にすることがある。非同期モデルで表現されたハードウェアを解析する場合、同期システムとして実現される点を考慮すると、同期システムでは考える必要のない状態を取り除くことができ、システムの動作解析を簡潔に行える。そこで、本論文では、Milner⁵⁾によって提案された CCS (A Calculus of Communicating Systems) の表記法を基本とし、グローバルなクロックの概念を取り入れた新しい非同期モデルを提案する。このモデルでは、計算機システムを非同期的に表現するが、互いに独立な事象はグローバル・クロックに同期して全く同時に起きるものと解釈される。非同期的表現を同期的に解釈するという意味で、本論文で提案するモデルを擬似非同期システムモデル (Pseudo Asynchronous System model: PAS-model) と呼ぶ。本論文では、PAS-model の表記法、解析規則、解析例を示し、擬似非同期モデルの特徴を明らかにした。

1. ま え が き

シストリック・アルゴリズムに代表される、いわゆるハードウェア・アルゴリズム¹⁾を設計解析する場合、実現されるシステムの表現レベルは、次の二つの段階に大別して考えることができる。

1. 設計されたアルゴリズムが意図した動作をすることを調べることを目的としてシステムを表現し解析する段階。
2. アルゴリズムを実現するハードウェアの詳細な構造を明らかにし、その動作の解析が行われる段階。

段階 1 では、アルゴリズムが、ハードウェアとして詳細化される以前に関数的に正しく動作することを確認することを目的として、事象の生起の順序 (因果関係) に基づいてシステムがモデル化される。

このようなレベルではハードウェアの動作がクロックごとの動作として詳細化されていないため、事象生起の因果関係を基本とした非同期モデル²⁾⁻⁶⁾で記述するのが自然である。しかし、これらの非同期モデルを解析する場合に問題となる点は、解析時に導出される状態数が膨大なものとなり、大規模なシステムの動作の解析が難しいことである。

一方、段階 2 のハードウェア・レベルでは実際の動作は普通クロックに同期して行われる。したがって、

段階 1 で表された非同期モデルを解析する場合に、段階 2 の同期式動作を考慮すると、同期システムでは考える必要のない状態をあらかじめ除くことができる。例えば、二つの互いに独立な事象の並列動作は図 1 の petri net において、二つの transition が同時に発火可能となることによって表現される。この動作を解析する場合、petri net ではこの二つの transition に発火順序をつけ、図 2 に示した S1, S2 のような中間的な状態を生成する。しかし、この二つの動作が同期している場合には S1, S2 という中間的な状態を経由することなく、直接 S0 から最終状態の S3 に遷移すると考えると解析の手間が減る。

このように、上位レベルの動作は非同期的だが下位レベルの動作は同期式であるシステムは完全な非同期モデルとしてモデル化するよりも、同期的な解析を可能とした非同期モデルを用いる方が、システムの動作を簡潔に解析することができる。そこで、本論文では、Milner⁵⁾によって提案された CCS (A Calculus of Communicating Systems) の表記法を基本とし、グローバルなクロックの概念を取り入れた新しい非同期モデルを提案する。このモデルでは、計算機システムを非同期的に表現するが、互いに独立な事象はグローバル・クロックに同期して全く同時に起きるものと解釈される。非同期的表現を同期的に解釈するという意味で、本論文で提案するモデルを擬似非同期システムモデル (Pseudo Asynchronous System model: PAS-model) と呼ぶ。以下では、第 2 章で計算機システムの PAS-model による表記法、第 3 章で擬似非同期の概念、第 4 章で PAS-model の解析手法、第 5 章

† A Pseudo Asynchronous System Model Based on CCS by
ATSUSHI TAKAHARA and TAKASHI NANYA (Department of
Computer Science, Faculty of Engineering, Tokyo Institute
of Technology).

†† 東京工業大学工学部情報工学科

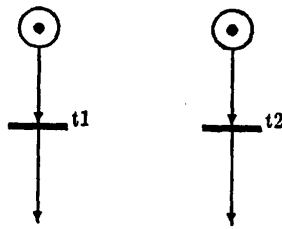


図 1 petri net
Fig. 1 Petri net.

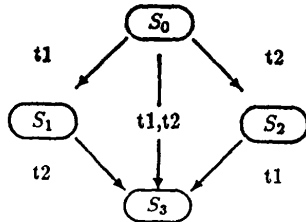


図 2 transition の発火順序
Fig. 2 Firing Sequences of transitions.

で PAS-model による計算機システムの解析例について述べる。

2. PAS-model の表記法

PAS-model では、システムの動作は構成要素間でデータ転送を行う通信動作として表現される。各構成要素は、ポート* と呼ばれる入出力端子を持つ。システムはこのポートを持つ機能素子の集合として表現される。

2.1 ポート動作

ポートは名前と属性（入力、出力の別）により示される。入力を“>”，出力を“<”の記号で表現し、属性と名前を合わせた表現をポート名と呼ぶ。ポート動作はポート名とポートの通信動作で受け渡される値の組で示される。ポート動作を“.”で接続した接続を動作体と呼ぶ。動作体の名前前でポートの逐次動作を参照することができる。

図 3** に示される乗算器を PAS-model で表現すると式 (1) となる。

$$MUL = (>in\ 1\ x). (>in\ 2\ y). (<out\ 1\ x * y).$$

$$MUL \quad (1)$$

ここで、>in 1, >in 2, <out 1 はポート名、x, y, $x * y$ は当該ポートの通信動作で受け渡される値、MUL は動作体の名前を表す。MUL は、三つのポート動作を接続の順に行うことを示している。ポート動

* CCS では sort と呼ばれているものに相当する。

** 以下、図では□は入力ポートを、■は出力ポートを示す。

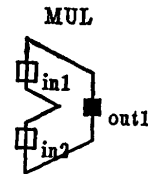


図 3 乗算器
Fig. 3 Multiplier.

作の接続の最後に現れる MUL はポート out 1 の動作が終了した後、MUL の動作が再び行われることを示している。

式 (1) の表現では乗算器の入力ポートが in 1, in 2 の順で動作する。乗算器の場合にはどちらの入力が先に与えられるかは不明であるので、このように順序をあらかじめ定めることは正しくない。そこで、二つのポートのいずれか一方が非決定的に選択されて動作することを非決定的動作として“+”を用いて表現する。この“+”の表現を用いると乗算器を式 (2)～(3) として表現できる。

$$MUL = (>in\ 1\ x). (>in\ 2\ y). MUL\ 1(x, y) \\ + (>in\ 2\ y). (>in\ 1\ x). MUL\ 1(x, y) \quad (2)$$

$$MUL1(p\ 1, p\ 2) = (<out\ 1\ p\ 1 * p\ 2). MUL \quad (3)$$

MUL の表現では、二つの入力ポートの動作の順序を二通り表し、これらを+でつないで表現している。これは、二通りのポート動作のいずれか一方が非決定的に選択され動作することを示す。

2.2 同時動作

非決定的動作の表現においてもポートの動作については順序が定められている。しかし、同期式システムをモデル化する場合には、動作が同時であることと、動作が順番に起こることを区別することが必要である。そこで、“二つの事象が同時に起こる”ということ表現する必要がある。CCS においては、同時動作の表現を、可能なすべての順序を尽くすことにより表現しているが、PAS-model では同時動作の表現を導入する。

定義 2.1 (同時動作) {} で囲まれたポート動作の表現は、ポートが同時に通信を行うことを示す。ただし、同じ変数を用いる異なる属性を持つポート動作が含まれる場合、入力の属性を持つポート動作が先行されるとする。

同時動作表現を用いた乗算器の表現は式 (4) となる。

$$MUL = \{(>in\ 1\ x), (>in\ 2\ y)\}. (<out\ 1\ x * y).$$

$$MUL \quad (4)$$

これは、ポート $in\ 1$, $in\ 2$ が同時に動作することを示している。そして、この二つのポート動作が終了した時点で、ポート $out\ 1$ の動作が行われる。また、式(5)の表現は、変数 x が同時動作表現に含まれる入力ポートと出力ポートの両方で用いられている例である。

$$THROUGH = \{(>in\ 1\ x), (<out\ 1\ x)\}.$$

$$THROUGH \quad (5)$$

この場合、定義 2.1 より、入力ポート $in\ 1$ が先に通信を行い、その後、出力ポート $out\ 1$ が通信を行うことになる。

PAS-model では、グローバル・クロックの存在を仮定し、時刻を次のように定義する。

定義 2.2 (時間単位) ポート動作、または、同時動作が行われる時間を PAS-model の 1 時間単位とする。

この定義より、式(5)における同時動作は、一つのポート動作と同じ時間で行われることになる。そのため、式(5)の動作体は受け取った値を遅延なく他の動作体に渡すことになる。なお、ここで述べた同時動作の表現は、一つの動作体の複数のポート動作が必ず同時に動作することを示している。2.4 通信において、「複数の動作体の通信可能なポートが同期的に通信を行うと解釈する」場合の同時の意味とは異なることに注意を要する。

次に、条件を表現する記述を導入する。PAS-model では、条件を [] で表現する (式(6))。

$$[条件]. ポート動作_1, \dots, ポート動作_n. 動作体名 \quad (6)$$

条件が成立する場合にはそれに続くポート動作が行われ、不成立の場合にはそれ以後の動作は行われない。また、条件が非決定的動作と合わせて用いられた場合には、条件が成立する動作が選ばれる。条件が成立する動作体が複数ある場合には、それらの非決定的動作となる。なお、以下ではポート動作、同時動作、条件から成る接続、および、それらの非決定的動作を含めて動作体と呼ぶ。

2.3 動作体の合成

PAS-model では、システムを名構成要素に対応する動作体の集合として表現する。ポートの通信動作とは、異なる動作体間でデータを受け渡すことである。この時、動作体間の接続関係を示す表現が必要となる。

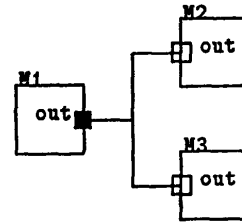


図4 動作体の合成として許されない例
Fig. 4 Inhibited ports.

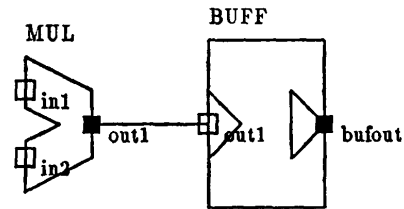


図5 動作体の合成
Fig. 5 Composition of behavior.

定義 2.3 (動作体の合成) 動作体の合成は以下のよう表現される。

$$(動作体_1 | 動作体_2 | \dots | 動作体_m)$$

動作体の合成において、ポート名が同じで互いに逆の属性を持つ二つのポートが現れるならば、これらのポートの対を“内部ポート対”，また、各ポートを“内部ポート”と呼ぶ。どちらか一方の属性しか伴わないポート名を持つポートを“外部ポート”と呼ぶ^{**}。さらに、同じ名前と同じ属性をもつポート動作は動作体の合成中の異なる動作体に存在することはない^{***}。

動作体の合成の例として、式(4)で示された乗算器とバッファを接続することを考える(図5)。バッファは PAS-model では式(7)としてモデル化できる。

$$BUFF = (>out\ 1\ res). (<bufout\ res). BUFF \quad (7)$$

次に、バッファ BUFF と乗算器 MUL を合成する表現は式(8)となる。

$$(MUL \setminus BUFF) \quad (8)$$

式(8)に、式(4)、(7)を併せた表現において、内部ポート対は $<out\ 1$, $>out\ 1$ であり、これらの二つのポート間が接続されていることを表す。

2.4 通 信

通信はポートが動作することにより行われる。動作

* CCS では composition と呼ばれたものである。

** CCS では composition に含まれる sort の中で restriction により示される sort のみが接続される。本モデルでは、restriction 操作を定義せず、動作体の合成に含まれる同じ名前のポートはすべて接続されるものとする。

*** これは、図4のようなポートを許さないことを示している。

体では、ポートの動作順序が決めている。そのため、動作体の合成において互いに接続された内部ポート対は常に通信可能というわけではなく、動作体により示されるポート動作の順序に従い通信可能となるポートが異なる。

定義 2.4 (通信可能) ポート動作、同時動作が通信可能か否かは、動作体の合成に含まれる動作体の先頭の表現により次のように定められる。

1. 先頭表現がポート動作の場合

内部ポートならば、その相手となるポートも他の動作体の先頭にある場合のみ通信可能となる。外部ポートの場合には、常に通信可能である。

2. 先頭表現が同時動作の場合

同時動作に含まれるすべてのポートが条件 1 を満たす時に、通信可能である。

3. 先頭表現が条件の場合

条件が成立する場合には、その条件の後に続く動作体 (ポート動作、同時動作、非決定的動作) について通信可能かどうかを判断する。条件が不成立の場合には、その表現に続く動作体は通信不可能である。

4. 先頭表現が非決定的動作の場合

非決定的に選択される動作体それぞれについて 1~3 が成立するかどうかを調べる。

例えば、乗算器 (式 (4)) とバッファ (式 (7)) を合成した式 (9) について、通信可能なポート動作を考える。

$$\{(\langle in 1 x \rangle, \langle in 2 y \rangle) . (\langle out 1 x * y \rangle . MUL | \langle out 1 res \rangle) . (\langle bufout res \rangle . BUFF) \} \quad (9)$$

この記述では、 $\{(\langle in 1 x \rangle, \langle in 2 y \rangle)\}$ と $\langle out 1 res \rangle$ が先頭の表現である。 $\{(\langle in 1 x \rangle, \langle in 2 y \rangle)\}$ は二つのポートとも外部ポートなのでただちに通信可能となる。一方、 $\langle out 1 \rangle$ は内部ポートであるため、その対応するポート $\langle out 1 \rangle$ が他の動作体の先頭になければならない。しかし、 $\langle out 1 \rangle$ は式 (9) の表現のどの動作体の先頭の表現にも現れていないことからポート $\langle out 1 \rangle$ は通信不可能となる。

次に通信可能なポートが実際に通信を行う規則を示す。PAS-model ではポート動作を同期的に解釈する、すなわち、通信可能な複数個のポート動作はすべて同時刻に起こると考える。(これと対照的に、CCS では通信可能な複数個のポートが同時刻に通信を行うとは考えない。これは、CCS がポート動作を非同期的に解釈することによる。) PAS-model では時間単位を刻むグローバル・クロックを仮定し、このクロックのある

時刻で通信可能な内部ポートはすべてその時刻で必ず通信動作を行うと考える。一方、外部ポートが通信可能な場合は CCS と同様にそのポートが通信可能となった時刻において通信を行う場合と行わない場合の二つの場合があるとする。これは、外部ポートは与えられた動作体の合成の表現のみから、その外部ポートが通信を行う相手が明らかではないことから、通信相手の状態が通信可能な場合と不可能な場合の二つの場合を考えなければならないためである。

ただし、同期的解釈を非決定的動作に含まれる複数の通信可能なポートに適用する場合には矛盾が生ずる。例えば、式 (10)~(13) の表現を考えてみる。

$$B1 = (\langle ax 1 \rangle) . (\langle bx 1 \rangle) . B1 \\ + (\langle cx 2 \rangle) . (\langle dx 2 \rangle) . B1 \quad (10)$$

$$B2 = (\langle a 0 \rangle) . B2 \quad (11)$$

$$B3 = (\langle c 1 \rangle) . B3 \quad (12)$$

$$S0 = (B1 | B2 | B3) \quad (13)$$

式 (13) の動作体 $B1, B2, B3$ の合成 $S0$ では、動作体 $B1$ のポート a, c が同時に通信可能であることから動作体 $B1$ では、非決定的な二者択一関係にある二つのポートが同時に動作を行うことになり非決定的動作の意味に矛盾する。このような状態を PAS-model では *conflict* と呼ぶ。

定義 2.5 (*conflict*) 動作体の合成に含まれる動作体のうち非決定的動作に含まれるポート動作の接続の先頭にあるポート動作、または、同時動作が二つ以上通信可能である場合、その動作体は *conflict* 状態にあるという。

以上を、ポートの通信規則としてまとめる。

規則 2.1 (通信規則)

- *conflict* 状態にある動作体に含まれない内部ポート

通信可能となった時刻に通信を行う。

- *conflict* 状態にある動作体に含まれない外部ポート

その時刻に通信を行うかどうかは非決定的である。通信を行う場合には他のポートと同時に動作する。

- *conflict* 状態にある動作体に含まれるポート
conflict 状態にある動作体に含まれる通信可能なポートから、一つのポートが選択されて動作する。

2.5 非決定的動作に関する分配律

非決定的動作におけるポート動作を考える。非決定

的動作の意味から、非決定的動作に含まれる複数のポートが同時に通信可能となっても、そのうちから一つが選ばれて動作することになる。一方、動作体の合成に非決定的動作が含まれる場合には、分配律を用いて複数の動作体の合成の非決定的動作に分解することができる。

定義 2.6 (動作体の合成の分解) 左辺で表される動作体の合成の意味は右辺の非決定的動作であるものとする。

$$\begin{aligned} & (B_1 | B_2 | \dots | B_i' + \dots + B_j' + \dots + B_m' | \dots | B_n) \\ &= (B_1 | B_2 | \dots | B_i' | \dots | B_n) \\ &+ (B_1 | B_2 | \dots | B_j' | \dots | B_n) \\ &+ (B_1 | B_2 | \dots | B_m' | \dots | B_n) \end{aligned}$$

ここで、 B_i は動作体とする。 B_j' は通信可能な動作体を先頭に持つ動作体とする。また、左辺の動作体を右辺の表現に置き換える操作を動作体の分解と呼ぶ。 ■

定義 2.6 を式 (14) ~ (16) に適用してみる。

$$\begin{aligned} M3 &= (<c\ 0). M3 \\ &+ (<d\ 1). M3 \end{aligned} \quad (14)$$

$$\begin{aligned} M3' &= (>c\ x). M3' \\ &+ (>d\ y). M3' \end{aligned} \quad (15)$$

$$S\ 0 = (M3 | M3') \quad (16)$$

式 (14) ~ (16) に示される動作体の合成に含まれる非決定的動作を、定義 2.6 を繰り返し適用し、分解した結果が式 (20) ~ (23) である。

$$S\ 0 = (M3 | M3') \quad (17)$$

$$= ((<c\ 0). M3 | M3') \quad (18)$$

$$+ ((<d\ 1). M3 | M3') \quad (19)$$

$$= ((<c\ 0). M3 | (>c\ x). M3') \quad (20)$$

$$+ ((<d\ 1). M3 | (>c\ x). M3') \quad (21)$$

$$+ ((<c\ 0). M3 | (>d\ y). M3') \quad (22)$$

$$+ ((<d\ 1). M3 | (>d\ y). M3') \quad (23)$$

上の $S\ 0$ において、式 (20) で表される動作体の合成が選択されるとポート対 $(<c, >c)$ が、また、式 (23) ではポート対 $(<d, >d)$ がそれぞれ通信可能である。しかし、式 (21), (22) においては通信可能なポートが存在しない。これは、動作体の合成、式 (21), または、式 (22) では、各動作体の先頭にあるポートが異なっているからである。一つの独立な動作体においては、非決定的動作は“+”で結ばれた動作体の一つが非決定的に選択されて動作することを表現するが、動作体の合成では通信可能条件によって束縛されていることから、選択枝のうちいくつかは実際に起こり得ない事象となる。したがって、式 (21), (22) の動作は実際には

起こり得ないものであり、これらの動作体の合成を削除しても表現されるシステムの動作は変わらない。このように非決定的動作が動作体の合成に含まれる場合には、それらを分解して得られた動作体の合成のうち通信可能となるものだけが選択される。4章で実際の解析手続きが述べられる。

3. 擬似非同期モデルの意味

PAS-model ではシステムの動作がポートにおける通信の待ち合わせにより表現される。この待ち合わせが成立した場合にかぎりポートの通信動作が可能となる。この意味で、*PAS-model* はシステム動作を非同期的に表現すると考えられる。一方、*PAS-model* では通信可能なポートが複数ある場合には、それらが *conflict* 状態にない限りグローバル・クロックに同期して同時に通信すると解釈される。例えば、

$$M1 = (<a\ 0). M1 \quad (24)$$

$$M1' = (>a\ x). M1' \quad (25)$$

$$M2 = (<b\ 1). M2 \quad (26)$$

$$M2' = (>b\ y). M2' \quad (27)$$

$$S = (M1 | M1' | M2 | M2') \quad (28)$$

で表現される動作体の合成 S では、二つの内部ポート対 a, b は同時に通信を行うと解釈される。

一方、式 (29) ~ (33) では、動作体 $B1, B2$ 間では二つの内部ポート対が通信可能であり $B3, B4$ では一つの内部ポート対が通信可能である。

$$\begin{aligned} B1 &= (>a\ x\ 1). (<a\ 1\ x\ 3). B1 \\ &+ (>b\ x\ 1). (<b\ 1\ x\ 3). B1 \end{aligned} \quad (29)$$

$$\begin{aligned} B2 &= (<a\ x\ 2). (>a\ 1\ x\ 4). B2 \\ &+ (<b\ x\ 2). (>b\ 1\ x\ 4). B2 \end{aligned} \quad (30)$$

$$B3 = (<k\ y\ 1). (>k\ 1\ y\ 3). B3 \quad (31)$$

$$B4 = (>k\ y\ 2). (<k\ 1\ y\ 4). B4 \quad (32)$$

$$S\ 0 = (B1 | B2 | B3 | B4) \quad (33)$$

$B1, B2$ 間で動作が可能なる二つの内部ポート対 a, b は *conflict* の状態にありそのどちらか一方の動作のみが可能である。また、これとは独立に内部ポート対 k は通信可能となっている。この場合にも、独立な内部ポート対は同時に動作することになり、内部ポート対 a と内部ポート対 k , または、内部ポート対 b と内部ポート対 k のそれぞれのポートが同時に動作すると解釈される。

同期式計算機システムの動作を表現することを考えた場合、システムの動作はクロックに同期した動作であるが並列に動作するモジュールが多数存在する。こ

の並列動作を同期的に表現しようとする、システム全体の動作を各クロックごとにまとめて表現しなければならない。これに対し、*PAS-model* ではシステムは各モジュールの動作を非同期的に表現し、それらの並列動作は合成として容易に表現することができる。一方で、システムの動作を解析する場合には、グローバル・クロックを仮定していることから、すべてのモジュールが同期した動作として解釈される。*PAS-model* を用いてシステムを解析すると、非同期的に示されたモジュールから合成されたシステムが同期式システムとして実現された場合の各クロックごとの動作を得ることができる。

このように、*PAS-model* ではシステム動作を非同期的に表現する一方で、それを解析する場合には同期式ハードウェアの具体的な動作を反映して、同期的解釈を与えることに特徴がある。

4. 解析手法

PAS-model の解析は、グローバル・クロックの各時刻におけるポートの動作を追跡し、各時刻における動作体の合成の表現を導出することである。この解析は二つの規則に従って実行される。一つは、*conflict* の状態にある動作体を含む動作体の合成を *PAS-model* の非決定的動作の解釈に従い分解する規則であり、もう一つは、動作体の合成の中でポートが動作した後に得られる動作体の合成を求める規則である。

まず、*conflict* の状態にある動作体の合成の分解を行う解析規則を示す。

規則 4.1 (解析規則 I) 動作体の合成 BC が与えられた時、次の規則で動作体を分解する。

$$BC = (B_0 | \dots | B_i | \dots | B_n)$$

ここで、 B_i は動作体である。

conflict が存在しない場合、通信可能なポートはすべて同時に通信を行うものとする。*conflict* が生じた場合には次の規則で動作体の合成を分解する。

1. 動作体の合成 BC に含まれる非決定的動作を定義 2.6 を繰り返し適用し分解する。

$$BC = \sum_{i=0}^k BC_i'$$

ここで、 BC_i' は分解された動作体の合成

2. BC を分解することにより得られた各動作体の合成 (BC_i') について、*chklist* に含まれ、*alist* に含まれないポート p を含む *conflictset* の要素 C について、 C に含まれる p 以外のポート

q が次の二つの条件をともに満たす動作体の合成は削除される。

条件 1 すべての q が *alist* に含まれない。

条件 2 q の対応するポート q を含む *conflictset* の要素 C' に含まれるすべてのポートが *alist* に含まれない。

conflictset 分解される以前の動作体の合成 (BC) に含まれる各動作体 (B_i) の中で *conflict* を起こしているポート群の集合

chklist 展開される以前の動作体の合成 (BC) において通信可能なポートの集合

alist 展開された動作体の合成 (BC_i') において通信可能なポートの集合

条件 1, 条件 2 を共に満たす時、分解する以前に通信可能であった内部ポート対が分解以後通信不可能となったことを示している。これは、この分解により通信可能な内部ポート対を含む二つの動作体で、通信するポートが存在しないことになり、通信規則に反する。そこで、条件 1, 条件 2 を満たす分解された動作体の合成は削除される。

例として、式 (33) の動作体の合成について解析規則 I を適用してみる。まず、式 (33) の動作体の合成を分解する。

$$S_0 = (B_1 | B_2 | B_3 | B_4) \quad (34)$$

$$= ((>ax\ 1). (<a\ 1\ x\ 3). B_1 | B_2 | B_3 | B_4) \quad (35)$$

$$+ ((>bx\ 1). (<b\ 1\ x\ 3). B_1 | B_2 | B_3 | B_4) \quad (36)$$

$$= ((>ax\ 1). (<a\ 1\ x\ 3). B_1 | (<ax\ 2). (>a\ 1\ x\ 4). B_2 | B_3 | B_4) \quad (37)$$

$$+ ((>ax\ 1). (<a\ 1\ x\ 3). B_1 | (<bx\ 2). (>b\ 1\ x\ 4). B_2 | B_3 | B_4) \quad (38)$$

$$+ ((>bx\ 1). (<b\ 1\ x\ 3). B_1 | (<ax\ 2). (>a\ 1\ x\ 4). B_2 | B_3 | B_4) \quad (39)$$

$$+ ((>bx\ 1). (<b\ 1\ x\ 3). B_1 | (<bx\ 2). (>b\ 1\ x\ 4). B_2 | B_3 | B_4) \quad (40)$$

次に、分解する以前の *conflictset*, *chklist*, および、分解された動作体 (式 (37)) に対する *alist* を求める。

• *conflictset* ($(>a>b) (<a<b)$)

• *chklist* ($(>a>b<a<b<k>k)$)

• *alist* ($(>a<a<k>k)$)

この場合、分解される以前に通信可能で、かつ、分解された以後通信不可能となるポートは b である。 b の含まれる *conflictset* の b 以外の要素 a は *alist* に含

まれ通信可能であるから、規則 4.1 の条件に当てはまらずこの動作体の合成は削除されない。

一方、式 (39) に対する alist は次のようになる。

• alist ($\langle k \rangle$)

この場合、分解される以前に通信可能でかつ、分解された以後通信不可能となるポートは a, b である。しかし、 a, b は同じ B , conflictset の要素に含まれることから規則 4.1 の二つの条件を満たす。このため、分解により得られた式 (39) の動作体の合成は削除される。

次に、動作体の合成の中でポートが動作した後に得られる動作体の合成を求める規則を示す。これは、規則 4.1 で得られた動作体の合成の各要素に対して、通信可能なポートの動作を導出する規則であり、次の手順で行われる。いま、三つの動作体 B_1, B_2, B_3 の合成^{*}、式 (44) を解析することを考える。

$$B_1 = C_1. I_1. B'_1 \quad (41)$$

$$B_2 = C_2. I_2. B'_2 \quad (42)$$

$$B_3 = C_3. E_3. B'_3 \quad (43)$$

$$(B_1 | B_2 | B_3) \quad (44)$$

ここで式 (44) におけるポート動作を解析する。今、内部ポート I_1, I_2 が双対なポートとし通信可能であるとする。この時、式 (44) では、内部ポート I_1, I_2 は通信をかならず行うが、外部ポート E_3 は通信を行う場合と行わない場合の二通り考えられる。よって、式 (44) からは次の二つの動作体の合成が導出される。

$$[C_1 \text{ and } C_2 \text{ and } C_3] (B'_1 | B'_2 | B'_3) \quad (45)$$

$$+ [C_1 \text{ and } C_2] (B'_1 | B'_2 | B_3) \quad (46)$$

これは、内部ポートと外部ポートの両者が共に動作する場合 (式 (45))、内部ポートのみが動作する場合 (式 (46)) の二つを示し、これらのどちらかが可能であることを非決定動作により示している。この解析規則を一般化した規則が規則 4.2 である。

規則 4.2 (解析規則 II) 規則 4.1 により得られた動作体の合成 BC_i' は、

$$BC_i' = (B_1 | \dots | B_i | \dots | B_n)$$

ここで、 B_i は動作体を表す。

以下のように分解できる。

$$BC_i' = [C_1 \text{ and } \dots \text{ and } C_j \text{ and } \dots \text{ and } C_m].$$

$$(B_1 | \dots | B'_{r_1} | \dots | B'_{r_j} | \dots | B'_{r_m} | \dots | B_n)$$

$$+ \sum_{p=1}^s \{C_1, \dots, C_j, \dots, C_m, \prod_{k=1}^p C_k^{nr_p}, E_k\}.$$

$$(B_1 | \dots | B'_{nr_1} | \dots | B'_{r_1} | \dots | B'_{nr_p} | \dots$$

* ここで、 C_1, C_2, C_3 は条件を、 I_1, I_2 は内部ポートを、 E_3 は外部ポートを、 B_j, B'_j ($1 \leq j \leq 3$) は動作体を示す。

$$|B'_{r_j} | \dots | B'_{nr_p} | \dots | B'_{r_m} | \dots | B_n)$$

ここで、

$$B_{r_j} = C_j. I_j. B'_{r_j} \quad (1 \leq j \leq m)$$

C_j : 条件, I_j : 内部ポート

B'_{r_j} : 入力ポート I_j に現れる変数をそのポートが通信動作で受け取った値で置き換えた動作体

$$B_{nr_q} = C_q^{nr_p}. E_q. B'_{nr_q} \quad (1 \leq q \leq s)$$

E_q : 外部ポート

$\prod_{i=1}^p p_i$: p_i の集合から、 y 個の要素からなるすべての部分集合を作り出す演算

5. 解析例

解析規則 4.1, 4.2 を適用することにより動作体の合成がどのように通信動作を行うかを解析することができる。例として、arbiter に制御されて動作する回路 (図 6) を考えてみよう。

$$ST1 = (\langle st1 \rangle START). ST1 \quad (47)$$

$$P1 = (\langle st1 \rangle t). (\langle req1 \rangle 1). (\langle gr1 \rangle u). \quad (48)$$

$$(\langle do1 \rangle ACT). (\langle end1 \rangle 1). P1$$

$$ST2 = (\langle st2 \rangle START). ST2 \quad (49)$$

$$P2 = (\langle st2 \rangle q). (\langle rep2 \rangle 1). (\langle gr2 \rangle v). \quad (50)$$

$$(\langle do2 \rangle ACT). (\langle end2 \rangle 1). P2$$

$$AR = (\langle req1 \rangle w). (\langle gr1 \rangle OK). (\langle end1 \rangle x). AR \\ + (\langle req2 \rangle y). (\langle gr2 \rangle OK). (\langle end2 \rangle z). AR \quad (51)$$

$$S0 = (ST1 | ST2 | P1 | P2 | AR) \quad (52)$$

解析は $S0$ から出発する。 $S0$ では内部ポート対 $st1, st2$ が通信可能である。 $S0$ に解析規則 I を適用すると、 conflict が存在しないことから動作体の分解は行われない。次に、解析規則 II を適用すると、内部ポート対 $st1, st2$ が同時に通信した後の動作体の合成が得られる。

$$S0 = (ST1 | ST2 |$$

$$(\langle req1 \rangle 1). (\langle gr1 \rangle u). (\langle do1 \rangle ACT).$$

$$(\langle end1 \rangle 1). P1 | (\langle req2 \rangle 1). (\langle gr2 \rangle v).$$

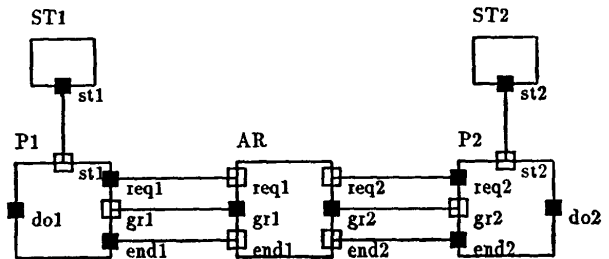


図 6 arbiter 回路
Fig. 6 Arbiter circuit.

$$\begin{aligned}
& (<do\ 2\ ACT). (<end\ 2\ 2). P2|(>req\ 1\ w). \\
& (<gr\ 1\ OK). (>end\ 1\ x). AR \\
& +(>req\ 2\ y). (<gr\ 2\ OK). (>end\ 2\ z). AR) \\
& \quad (53)
\end{aligned}$$

次に, $S0$ から導出された動作体の合成 (式(53)) に解析規則を適用する. 式(53)では, 内部ポート対 $req\ 1$, $req\ 2$ が通信可能である. しかし, この二つのポートは動作体 AR の非決定的動作で結ばれた動作体の先頭のポートであることから, 動作体 AR は *conflict* 状態にある. そこで, 解析規則 I を適用すると次の分解が得られる.

$$\begin{aligned}
& (ST1|ST2| \\
& \quad (<req\ 1\ 1). (>gr\ 1\ u). (<do\ 1\ ACT). \\
& \quad (<end\ 1\ 1). P1|(<req\ 2\ 1). (>gr\ 2\ v). \\
& \quad (<do\ 2\ ACT). (<end\ 2\ 2). P2|(>req\ 1\ w). \\
& \quad (<gr\ 1\ OK). (>end\ 1\ x). AR) \quad (54) \\
& + (ST1|ST2| \\
& \quad (<req\ 1\ 1). (>gr\ 1\ u). (<do\ 1\ ACT). \\
& \quad (<end\ 1\ 1). P1|(<req\ 2\ 1). (>gr\ 2\ v). \\
& \quad (<do\ 2\ ACT). (<end\ 2\ 2). P2|(>req\ 2\ y). \\
& \quad (<gr\ 2\ OK). (>end\ 2\ z). AR) \quad (55)
\end{aligned}$$

さらに, 解析規則 II を適用することにより, 内部ポート対 $req\ 1$, $req\ 2$ のそれぞれが通信した後の動作体の合成が得られる.

$$\begin{aligned}
& (ST1|ST2| \\
& \quad (>gr\ 1\ u). (<do\ 1\ ACT). (<end\ 1\ 1). P1| \\
& \quad (<req\ 2\ 1). (>gr\ 2\ v). (<do\ 2\ ACT). \\
& \quad (<end\ 2\ 2). P2|(<gr\ 1\ OK). (>end\ 1\ x). AR) \\
& \quad (56) \\
& + (ST1|ST2| \\
& \quad (<req\ 1\ 1). (>gr\ 1\ u). (<do\ 1\ ACT). \\
& \quad (<end\ 1\ 1). P1|(>gr\ 2\ v). (<do\ 2\ ACT). \\
& \quad (<end\ 2\ 2). P2|(<gr\ 2\ OK). (>end\ 2\ z). AR) \\
& \quad (57)
\end{aligned}$$

これは, AR により, 動作体 $P1$, または, $P2$ のどちらか一方の動作が選択されたことを示している. 式(56)の解析を続けると, 外部ポート $<do\ 1$ が動作可能となる (式(58)).

$$\begin{aligned}
& (ST1|ST2|(<do\ 1\ ACT). (<end\ 1\ 1). P1| \\
& \quad (<req\ 2\ 1). (>gr\ 2\ v). (<do\ 2\ ACT). \\
& \quad (<end\ 2\ 2). P2|(>end\ 1\ x). AR) \quad (58)
\end{aligned}$$

式(58)では, 通信可能なポートは $<do\ 1$ のみであり, このポートが動作した後の動作体の合成は式(59)となる.

$$\begin{aligned}
& (<do\ 1\ ACT). (ST1|ST2|(<end\ 1\ 1). P1| \\
& \quad (<req\ 2\ 1). (>gr\ 2\ v). (<do\ 2\ ACT). \\
& \quad (<end\ 2\ 2). P2|(>end\ 1\ x). AR) \quad (59)
\end{aligned}$$

6. むすび

Milner によって提案された非同期モデル CCS を土台として, 同期式ハードウェアの解析に適した擬似非同期モデル *PAS-model* を提案した. *PAS-model* では, 表現形式として通信の待ち合わせによる非同期的表現を用いる. 一方, 解析は *conflict* がない限り同期的解釈に基づいて行うことを基本としている. ただし, *conflict* が起きた場合には, 従来の非同期モデルと同様に, 各ポートの動作を起こり得るすべての場合に分解して解析する. 従来の非同期モデルによる解析でも, 同期的な意味を表現することが可能であるが, その表現が複雑であり, 解析に膨大な記憶容量を必要とする. 一方 *PAS-model* では解析を同期的に解釈して行うため, 同期式計算機ハードウェアの解析が効率的に行える.

PAS-model の解析システム*は Unix 上にインプリメントされている. CCS に比べた場合, 解析時に導出される状態数が少なく効率的に解析を行うことができるが, 動作体の合成に含まれるポート数が多くなった場合, 通信可能となるポートを検索するのに時間を要する. 大規模なシステムに対応するためにはまだ改善する必要がある. また, *PAS-model* では, ポートの通信動作で受け渡される値の表現が変数, 式という形式に限定されている. 解析を行う場合, 受け渡される値について代数的な扱いが可能なモデルが望まれる. 抽象データ型などが扱えるように *PAS-model* を拡張すること, および, CCS との表現能力の差を明らかにすることが, 今後の課題として残されている. また, *PAS-model* の一つの応用として比較的抽象度の高いレベルにおける計算機ハードウェアの設計検証への利用がある⁷⁾.

参 考 文 献

- 1) 矢島, 都倉, 富田編: 特集; VLSI 向きハードウェア, 情報処理, Vol. 26, No. 6 (1985).
- 2) Chen, B. and Yeh, R. T.: Formal Specification and Verification of Distributed Systems, *IEEE Trans. Softw. Eng.*, Vol. SE-9, No. 9, pp. 710-722 (1983).
- 3) Hoare, C. A. R.: Communicating Sequential

* Franz Lisp を用いて記述され, ソースで約 3,000 行である.

- Processes, *Comm. ACM*, Vol. 21, No. 8, pp. 666-677 (1978).
- 4) Milne, G. J.: Simulation and Verification; Related Techniques for Hardware Analysis, *Proc. of CHDL 85*, pp. 404-417 (1985).
 - 5) Milner, R.: A Calculus of Communicating Systems, Goos, G. and Hartmanis, J. eds., *Lecture Notes in Computer Science*, Springer-Verlag (1980).
 - 6) Peterson, J. L.: *Petri Net Theory and The Modeling of Systems*, Prentice-Hall (1981).
 - 7) Takahara, A. and Nanya, T.: A Higher Level Design Verification, *Proc. of ICCD 88*, pp. 596-599 (1988).

(昭和63年1月5日受付)

(昭和63年11月14日採録)



高原 厚

昭和34年生。昭和58年東京工業大学工学部情報工学科卒業。昭和60年同大学院修士課程修了。昭和63年同大学院博士後期課程修了。同年日本電信電話(株)入社。現在、NTT LSI研究所勤務。ハードウェア記述言語、ハードウェア設計検証などに興味を持つ。工学博士。ACM、電子情報通信学会各会員。



南谷 崇 (正会員)

昭和21年生。昭和44年東京大学工学部計数工学科卒業。昭和46年同大学院修士課程修了。同年日本電気(株)入社。中央研究所勤務。昭和56年東京工業大学工学部情報工学科助教授。現在に至る。主として論理設計方法論、非同期システム論、計算機の耐故障設計に関する研究に従事。工学博士。著書「PLAの使い方」、「順序機械」(共著)。IEEE、電子通信学会、情報通信学会各会員。