プログラミング 18－5
(1998. 3. 23)

準弱双模倣性をもとにした仕様の段階的合成方法

磯部 祥尚, 佐藤 豊, 大蒔 和仁

電子技術総合研究所 情報アーキテクチャ部
〒305-8568 茨城県つくば市梅園1-1-4

あらまし 大規模なシステム全体を完全に一度で設計することは容易ではない。そこで、設計者の負担を減らすために、複数の部分的な仕様を記述して、それらに矛盾しないシステムを徐々に合成する方法が有効である。この方法により、複数の設計者が同時に複数の部分仕様を記述することができ、各設計者はシステム全体を把握する必要がなくる。本稿では、このような部分仕様の無矛盾性を判定する方法と、それら部分仕様を満たすために必要な最も弱い要求を表す仕様を合成する方法を提案する。我々の方法の利点は、着目したアクション以外を無視して、部分仕様を記述できることである。このような部分仕様は分散システムの局所的な仕様等に適している。

キーワード 仕様合成、部分仕様、多重仕様、プロセス代数、双模倣性

A Stepwise Synthesis Method for Specifications based on Quasi-Weak Bisimilarity

Yoshinao ISOBE, Yutaka SATO, Kazuhito OHMAKI

Computer Science Division, Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Ibaraki 305-8568, Japan
E-mail: isobe@etl.go.jp, ysato@etl.go.jp, ohmaki@etl.go.jp

Abstract It is difficult for designers to completely design a large system in one step. In order to decrease responsibility of each designer, it is useful to incrementally synthesize a system from *partial specifications* such that the final system satisfies the all partial specifications. In this case, many designers describe partial specifications in parallel and each designer does not have to know the whole system. In this paper, we present a method for checking consistency between given partial specifications and a method for synthesizing a flexible specification which shows the weakest requirement for satisfying all the partial specifications. The important advantage of our method is that designers can describe partial specifications by ignoring unknown actions. Such partial specifications are applicable to local specifications in distributed systems.

key words specification synthesis, partial specification, multi-specification, process algebra, bisimilarity

1 はじめに

計算の高速化や情報の分散化に伴い、並行プロセスは広く利用されているが、プロセス間通信による相互作用を考慮するため、その設計は容易ではない。一方、逐次的なプロセスには全ての動作を明確に記述できる利点がある。そこで、並行プロセスと観測的に等価な逐次プロセスを示すことによって、その並行プロセスの動作を明確にする方法が有効である。プロセス代数[1][2]は、並行プロセスと逐次プロセスの等価性を判定できる数学的な枠組として知られている。実装される並行プロセスに対し、それに等価な逐次プロセスをその仕様と呼ぶ。

並行プロセスの仕様を記述してその等価性を示すことにより、並行プロセスの信頼性は保証される。しかし、並行プロセスの仕様はしばしば複雑になり、一度に完全な仕様を記述することは容易ではない。また、同じシステムに対して、複数の設計者がさまざまな角度から部分的な仕様を記述することもある。このような場合、複数の部分仕様を満たす仕様を徐々に生成することによって、最終的に全ての部分仕様を満たす仕様を得る方法が提案されている。例えば、木村ら[3]は、μ計算で記述された複数の特性を満たすようにCCSのプロセスを生成する方法を提案し、Steenら[4]は、LOTOSで記述された複数の部分仕様から一つの仕様を合成する方法を提案している。

我々は、いくつかのアクションに着目して部分仕様を記述し、そのような複数の部分仕様を満たすように仕様を合成する方法を検討している。具体的には、各部分仕様は**有効アクション集合**と呼ばれるアクションの部分集合をもっており、その有効アクション集合に含まれるアクションの関係だけを示しているとする。つまり、有効アクション集合に含まれないアクションを観測できない内部アクションとみなし無視することができる。これにより、部分仕様を記述する設計者の負担を減少することができる。

仕様は基本的にCCSの逐次プロセスである。部分仕様は仕様Pと有効アクション集合Ωの組であり、$P::\Omega$と書かれる。ただし、文脈から有効アクション集合Ωが明かなときは、部分仕様$P::\Omega$を単にPと略記する。

例として、図1の通信機*TRANS*の部分仕様を記述する。この通信機は2人のメッセージの送受信を行なう。ここで、snd_iは送信、rec_iは受信、ack_iは返信、tm_iは時間切れを表す。まず、各サイト$i \in \{1,2\}$において、有効アクション集合として$\Omega_{STi} = \{snd_i, rec_i, ack_i, tm_i\}$をもつ部分仕様$ST_i$を次のように記述する。

$$ST_i \stackrel{\text{def}}{=} snd_i.(\tau.ack_i.ST_i + \tau.tm_i.ST_i) + rec_i.ST_i$$

ここで、ピリオド'.'は逐次演算子、'+'は選択演算子である。$\stackrel{\text{def}}{=}$は左辺の**仕様定数**ST_iを右辺の仕様として定義することを意味し、再帰定義を表す。また、τは観測できない内部アクションであり、自動的に実行される。つまり、送信snd_iした後、返信ack_iが戻るか、時間切れtm_iになるかは、サイトiの利用者にとっては非決定的に知らされる。次にサイトiからサイトjへの通信について、有効アクション集合として$\Omega_{TRij} = \{snd_i, rec_j, ack_i\}$をもつ部分仕様$TR_{ij}$ $(i \neq j)$を記述する。

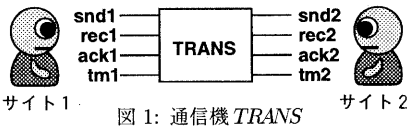

図1: 通信機*TRANS*

$$TR_{ij} \stackrel{\text{def}}{=} snd_i.(rec_j.ack_i.TR_{ij} + \tau.TR_{ij})$$

この部分仕様には、送信snd_1に対して受信rec_2が起こった場合は返信ack_1を受けとれることが明記されている。ただし、時間切れについては、その可能性のみがτによって記述されている。我々の目的は、これら部分仕様$ST_1, ST_2, TR_{12}, TR_{21}$を満たす仕様が存在するかを判定する方法と、その仕様を合成する方法を確立することである。

すでに我々は、強双模倣性[1]を等価性の基準として、仕様を合成する方法を与えた[9]。本稿では、より実用的な**準弱双模倣性**を等価性の基準として、複数の部分仕様の無矛盾性を判定する方法と、それら全ての部分仕様を満たすために必要な最も弱い要求を表す柔軟な仕様を合成する方法を与える。ここで、準弱双模倣性とは観測的に状態を変えない内部アクションを無視する双模倣性であり、3.2小節で定義される。我々は弱双模倣性[1]をもとにした合成方法を開発することを最終的な目的にしているが、弱双模倣性をもとにした合成方法は非常に複雑になる。そこで、今回はその準備としてより扱い易い準弱双模倣性をもとにする。

仕様の合成や詳細化に関する研究としては、すでに[3, 4, 5, 6, 7]等が提案されている。従来の方法と比較した我々の合成方法の特徴は、次のようにまとめられる。

- 有効アクション集合に含まれないアクションを隠す。
- 全ての部分仕様を満たす最も弱い柔軟な仕様を合成する。

つまり、有効アクション集合を指定して、未知のアクションを考慮せずに部分仕様を記述できる。また、柔軟な仕様を合成することにより、次々に加えられる部分仕様を満たすように、段階的に仕様を合成することができる。

2節で、本稿で用いる**多重仕様**の構文と意味を定義する。この多重仕様には複数の仕様を一つに畳み込んで記述できる。これにより、全ての部分仕様を満たす最も弱い仕様を効果的に表現できる。3節では、二つの部分仕様$P_1::\Omega_1$と$P_2::\Omega_2$を満たす仕様が存在するかを判定するために、準弱(Ω_1, Ω_2)**双模倣性**を定義する。4節で、全ての部分仕様を満たす最も弱い仕様の合成方法として**PQ**法を与える。5節で、1節で記述した通信機の部分仕様を用いて**PQ**法による合成例を示す。尚、紙面の都合により証明は省略する。

2 多重仕様の定義

本節では、**多重仕様**を記述するための言語*MSPEC*を定義する。多重仕様は複数の異なる仕様を一つに畳み込んで表現することができる。我々はすでに多重仕様を記述する

ための言語として、CCS[1] の逐次プロセスの演算子と多重演算子 $\vee$ から構成される言語 $SPEC^{\vee}$[9] を提案している。この多重演算子 $\vee$ は複数の仕様を畳み込むために使われ、$\vee$ は「かつ / または」と読むことができる。例えば、多重仕様 $(a.\mathbf{0} \vee b.\mathbf{0})$ は三つの仕様 $(a.\mathbf{0})$, $(b.\mathbf{0})$, $(a.\mathbf{0}+b.\mathbf{0})$ を表している。選択演算子 $+$ との直観的な違いは、$(a.\mathbf{0}+b.\mathbf{0})$ は実行時に $(a.\mathbf{0})$ のように振舞うか、$(b.\mathbf{0})$ のように振舞うかを選択する**動的選択**である。これに対し、$(a.\mathbf{0} \vee b.\mathbf{0})$ は設計時に $(a.\mathbf{0})$ として実装するか、$(b.\mathbf{0})$ として実装するか、$(a.\mathbf{0}+b.\mathbf{0})$ として実装するかを選択する**静的選択**である。

今回は多重仕様から一つの仕様を選択するときにある条件 (マーク) を付加できるように、$SPEC^{\vee}$ を拡張している。2.1小節で *MSPEC* の構文、2.2小節でその意味を与える。

2.1 *MSPEC* の構文

まず、名前の集合 $\mathcal{N}$ が与えられていると仮定する。このとき、アクションの集合 Act を $\mathcal{N} \cup \{\tau\}$ とし、その要素を $\alpha, \beta, \cdots$ で表す。τ は**内部アクション**と呼ばれる観測されないアクションを表しており、$\mathcal{N}$ には含まれないとする。さらに**仕様定数**の集合 $\mathcal{K}$ が与えられていると仮定する。このとき多重仕様の構文を次のように定義する。

定義 2.1 多重仕様 P の構文は *BNF* 記法を用いて

$$P \quad ::= \quad A \mid \mathbf{0} \mid \alpha.[P,P] \mid P+P \mid P \vee P$$

により与えられる。ここで、$\alpha \in Act$、$A \in \mathcal{K}$ である。 ∎

多重仕様の集合を $\mathcal{P}$ で表し、その要素を $P, Q, \cdots$ で表す。仕様定数は**定義式**によって意味を与えられる多重仕様であり、実際に全ての仕様定数 A について、$A \stackrel{\text{def}}{=} P$ $(P \in \mathcal{P})$ の形の定義式があると仮定する。演算子の結合の優先順位は、{ 逐次演算子 '.' > 多重演算子 '$\vee$' > 選択演算子 '+' } である。便宜上、次の記法も用いる。

$$\sum S \equiv \begin{cases} \mathbf{0} & (S = \emptyset) \\ P_1 + P_2 + \cdots + P_n & (S = \{P_1, \cdots, P_n\}) \end{cases}$$

$$\bigvee S \equiv \begin{cases} \mathbf{0} & (S = \emptyset) \\ P_1 \vee P_2 \vee \cdots \vee P_n & (S = \{P_1, \cdots, P_n\}) \end{cases}$$

$$\alpha.P \equiv \alpha.[P, \mathbf{0}]$$

ここで、関係 $\equiv$ は構文的に等しいことを表す。

多重演算子 $\vee$ を含まない多重仕様を特に**基本仕様**または**逐次プロセス**と呼び、$P_0, Q_0, \cdots$ で表す。基本仕様の集合 $\mathcal{P}_0$ は多重仕様の集合 $\mathcal{P}$ の部分集合であり、その構文は次の BNF 記法により与えられる。

$$P_0 \quad ::= \quad A_0 \mid \mathbf{0} \mid \alpha.[P_0, \mathbf{0}] \mid P_0 + P_0$$

ここで、$A_0 \in \mathcal{K}_0 \subseteq \mathcal{K}$ かつ $\alpha \in Act$ である。また、全ての仕様定数 A_0 について、$A_0 \stackrel{\text{def}}{=} P_0$ $(P_0 \in \mathcal{P}_0)$ の形の定義式があると仮定する。

部分仕様とは**有効アクション集合**をもつ多重仕様のことである。有効アクション集合の集合 (要素を Ω で表す) は名前の集合 $\mathcal{N}$ の部分集合の集合 $2^{\mathcal{N}}$ である。有効アクション集合 Ω をもつ多重仕様 P は、Ω に含まれるアクションの関係のみ表しており、$P :\Omega$ と書かれる。

MSPEC の多重仕様は $SPEC^{\vee}$ と同様に多重演算子 $\vee$ によって複数の基本仕様を表現する。$SPEC^{\vee}$ との違いは、逐次演算子 $\alpha.[P,Q]$ の Q の存在である。これは、アクション α による遷移の後は多重仕様 P のように振舞うが、その振舞いは多重仕様 Q による制約を受けることを表している。この Q のことを**マーカ**と呼び、次のように P の遷移にマークを付ける：もし P と Q が同じ遷移をもつならば、P のその遷移はマークされる。このとき、マーカ Q は P に対して有効であるという。もたないならば P は全くマークされない。例えば、$a.[(b.P + c.Q), (b.P)]$ では、a による遷移の後、b による遷移はマークされる。マークされない典型的な例は $\alpha.[P, \mathbf{0}]$ である。

マークされた状態遷移によって、また新たに別の状態遷移がマークされることがある。ここで重要なことは、無限に連続してマークが付くような遷移は禁止されていることである。つまり、いつかはマークの付いていない遷移が実行されるなければならない。これによって、「いつかはアクション a が実行可能な基本仕様」だけを含むような多重仕様の記述が可能となる。多重仕様と基本仕様の形式的な関係は 3.2小節で与えられる。

2.2 *MSPEC* の意味

言語 *MSPEC* の意味は**マーク付多重ラベル付遷移システム** (Marked Multi-Labelled Transition System : MMLTS と略す) により与えられる。MMLTS は多重ラベル付遷移システム [9] を拡張して本小節で定義される。まず、任意の集合 S について関数 $\langle\,\rangle$ と $[\,]$ を次のように定義する。

$$\langle S \rangle = \{\langle e_1, e_2, \cdots, e_n \rangle : e_i \in S, n \geq 1\}$$
$$[S] = \{[e_1, e_2] : e_1, e_2 \in S\}.$$

さらに集合 $\langle S \rangle$ 上に次の三つの関数を定義する。

定義 2.2 $s, s' \in \langle S \rangle$、$i \in \{1, 2, \cdots\}$ とする。

- $(\#s)$: s の長さ。(例 $\#\langle a, b, c \rangle = 3$)
- $(s; s')$: s と s' の結合。(例 $\langle a, b \rangle; \langle c, d \rangle = \langle a, b, c, d \rangle$)
- $(s \triangleleft i)$: s の i 番目の要素。(例 $\langle a, b, c, d \rangle \triangleleft 3 = c$) ∎

このとき、MMLTS を次のように定義する。

定義 2.3 MMLTS は三つ組 $(S, L, \rightarrow)$ である。ここで、

1. S は状態の集合、
2. L はラベルの集合、
3. $\rightarrow$ は次のような遷移関係である。
   $\rightarrow \subseteq \{(e, u, s) : e \in S, u \in \langle L \rangle, s \in \langle [S] \rangle, \#u = \#s\}$

慣習に従い $(e, u, s) \in \rightarrow$ のとき $e \xrightarrow{u} s$ と記述する。 ∎

多重仕様の意味は、MMLTS $(\mathcal{P}, Act, \rightarrow)$ により与えられる。ここで、$\rightarrow$ は定義 2.4によって与えられる。以下、M, N は $\langle [\mathcal{P}] \rangle$ の要素を表し、μ, ν は $\langle Act \rangle$ の要素を表す。

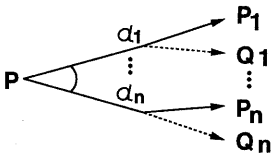

図 2: 多重仕様の遷移

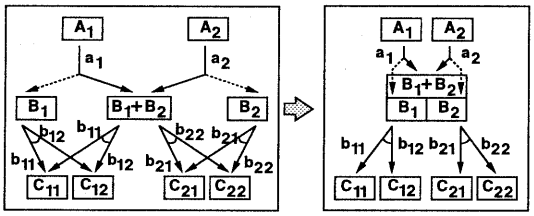

図 3: マーカの略記法

定義 2.4 遷移関係 $\rightarrow$ は次の推論規則を満たす最小の関係である。推論規則は、横棒の上が0個以上の仮定、右横が条件、下が結果を表している。

$$\mathbf{Act}\ \frac{}{\alpha.[P,Q] \xrightarrow{\langle\alpha\rangle} \langle[P,Q]\rangle} \qquad \mathbf{Choice}_1\ \frac{P \xrightarrow{\mu} M}{P+Q \xrightarrow{\mu} M}$$

$$\mathbf{Con}\ \frac{P \xrightarrow{\mu} M}{A \xrightarrow{\mu} M}(A \stackrel{\text{def}}{=} P) \qquad \mathbf{Choice}_2\ \frac{Q \xrightarrow{\nu} N}{P+Q \xrightarrow{\nu} N}$$

$$\mathbf{Mul}\ \frac{P \xrightarrow{\mu} M \quad Q \xrightarrow{\nu} N}{P \vee Q \xrightarrow{\mu;\nu} M;N}$$

ここで、**Mul** の ';' は定義 2.2の結合関数である。 ∎

多重仕様の状態遷移図を記述するために、遷移関係

$$P \xrightarrow{\langle\alpha_1,\alpha_2,\cdots,\alpha_n\rangle} \langle[P_1,Q_1],[P_2,Q_2],\cdots,[P_n,Q_n]\rangle$$

を図2に示すように弧と点線矢印を用いて表す。点線矢印の先にマーカを記述するが、マーカが $\mathbf{0}$ のときはその点線矢印を省略する。また、多重仕様 P の有効なマーカ Q の多くは P の部分式である。例えば、次のように有効なマーカ B_1 は (B_1+B_2) の部分式である。

$$A_i \stackrel{\text{def}}{=} a_i.[(B_1+B_2),B_i],$$
$$B_i \stackrel{\text{def}}{=} b_{i1}.[C_{i1},\mathbf{0}] \vee b_{i2}.[C_{i2},\mathbf{0}]$$

この状態遷移図は図3の左図のようになる。本稿では、これを図3の右図のように略記する。

3 無矛盾性の判定法

本節では、二つの部分仕様 $P_1::\Omega_1$ と $P_2::\Omega_2$ の無矛盾性を判定するために、準弱 (Ω_1,Ω_2) 双模倣性を定義する。まず、3.1小節でアクションの無矛盾性を定義する。これはすでに [9] で与えられた定義と同じであるが、要点をここでも述べる。その後、3.2小節で部分仕様の無矛盾性を定義する。3.3小節では、例をもとに多重仕様のマーカを説明する。

3.1 アクションの無矛盾性

まず、有効アクション集合 Ω に含まれないアクションを内部アクションに変換するためにフィルタ関数を用意する。アクション α が有効アクション集合 Ω に含まれるとき、α は Ω に対して有効であるという。

定義 3.1 フィルタ関数 ($/$: $Act \times 2^{\mathcal{N}} \rightarrow Act$) を次のように定義する。

$$\alpha/_\Omega = \begin{cases} \alpha & (\alpha \in \Omega) \\ \tau & (\alpha \notin \Omega) \end{cases}$$

∎

次に、二つの部分仕様 $P_1::\Omega_1$ と $P_2::\Omega_2$ の各アクション α_1 と α_2 から有効なアクションを選択するために、選択関数 $\bullet$ を用意する。もし二つのアクションが矛盾するならば $\perp(\notin Act)$ を出力する。

定義 3.2 選択関数 $\bullet$: $Act \times 2^{\mathcal{N}} \times Act \times 2^{\mathcal{N}} \rightarrow Act \cup \{\perp\}$ を、次のように定義する。

$$(\alpha_1,\Omega_1)\bullet(\alpha_2,\Omega_2) = \begin{cases} \alpha_1/_{\Omega_1}(\alpha_1/_{\Omega_1} = \alpha_2/_{\Omega_2}) & \text{(C1)} \\ \alpha_1 \quad (\alpha_1 \in \Omega_1 - \Omega_2, \alpha_2 \notin \Omega_2) & \text{(C2)} \\ \alpha_2 \quad (\alpha_2 \in \Omega_2 - \Omega_1, \alpha_1 \notin \Omega_1) & \text{(C3)} \\ \perp \quad (\text{上記以外}) & \text{(C4)} \end{cases}$$

∎

条件 (C4) は $(\alpha_1 \neq \alpha_2,\ \alpha_1 \in \Omega_1 \cap \Omega_2)$ または $(\alpha_1 \neq \alpha_2,\ \alpha_2 \in \Omega_1 \cap \Omega_2)$ または $(\alpha_1 \in \Omega_1 - \Omega_2,\ \alpha_2 \in \Omega_2 - \Omega_1)$ の場合である。(C1) により、二つのアクションが共に有効でかつ等しければ、結果はそのアクションとなり、共に無効であれば、結果は内部アクション τ となる。また、(C2) と (C3) により、一方が有効で他方が無効である場合、その有効なアクションが選択される。次に選択関数の適用例を示す。

$$(a,ab)\bullet(a,ac) = a \qquad (a,ab)\bullet(\tau,ac) = \perp$$
$$(b,ab)\bullet(\tau,ac) = b \qquad (b,ab)\bullet(c,ac) = \perp$$

二つの部分仕様 $P_1::\Omega_1$ と $P_2::\Omega_2$ の各 α_1 と α_2 が選択関数により矛盾しないとき、α_1 と α_2 は (Ω_1,Ω_2)-無矛盾といい、$(\alpha_1\ {}_{\Omega_1}\doteq_{\Omega_2}\ \alpha_2)$ と書く。つまり、${}_{\Omega_1}\doteq_{\Omega_2}$ は

$${}_{\Omega_1}\doteq_{\Omega_2} \ = \ \{(\alpha_1,\alpha_2) : (\alpha_1,\Omega_1)\bullet(\alpha_2,\Omega_2) \neq \perp\}$$

によって与えられる。

次の命題 3.1の (1) は両方のアクション α_1,α_2 に無矛盾なアクション α_{12} が存在するときは、α_1 と α_2 も互いに無矛盾であることを示している。これは、二つの部分仕様に矛盾しない仕様が存在するときは、それを合成できることを導く。また、命題 3.1の (2) は α_1 と α_2 から選択された α_{12} は、その両方に無矛盾であることを示している。これは合成された仕様がもとの部分仕様に矛盾しないことを導く。

命題 3.1 $\Omega_i \subseteq \mathcal{N}$、$\Omega_{12} = \Omega_1 \cup \Omega_2$ とする。任意の $\alpha_i \in Act$ について次の関係が成り立つ。

(1) もし $\alpha_1\ {}_{\Omega_1}\doteq_{\Omega_{12}}\ \alpha_{12}$ かつ $\alpha_2\ {}_{\Omega_2}\doteq_{\Omega_{12}}\ \alpha_{12}$ ならば、
$(\alpha_1,\Omega_1)\bullet(\alpha_2,\Omega_2) = \alpha_{12}/_{\Omega_{12}} \neq \perp$ である。

(2) もし $(\alpha_1,\Omega_1)\bullet(\alpha_2,\Omega_2) = \alpha_{12}$ ならば、
$\alpha_1\ {}_{\Omega_1}\doteq_{\Omega_{12}}\ \alpha_{12}$ かつ $\alpha_2\ {}_{\Omega_2}\doteq_{\Omega_{12}}\ \alpha_{12}$ である。 ∎

$$\textbf{Base}\ \frac{P \xrightarrow{\alpha}_{\tau} P' \quad [Q,R] \xrightarrow{\beta}_{\exists\exists} [Q',R'] \quad [Q,R] \not\xrightarrow{}_{\exists\exists}}{(P,[Q,R]) \overset{(\alpha,\beta)}{\Longrightarrow}_{(\Omega_1,C,\Omega_2)} (P',[Q',R'])} \left(\begin{array}{l} \alpha\ \Omega_1 \doteq \Omega_2\ \beta, \\ (P',Q') \in C \end{array} \right)$$

$$\textbf{Seq}\ \frac{P \xrightarrow{\alpha_1}_{\exists} P_1 \quad (P_1,[Q,R]) \overset{(\alpha,\beta)}{\Longrightarrow}_{(\Omega_1,C,\Omega_2)} (P',[Q',R'])}{(P,[Q,R]) \overset{(\alpha,\beta)}{\Longrightarrow}_{(\Omega_1,C,\Omega_2)} (P',[Q',R'])} \left(\begin{array}{l} \alpha_1\ \Omega_1 \doteq \Omega_2\ \tau, \\ (P_1,Q) \in C \end{array} \right)$$

$$\textbf{Conc}\ \frac{P \xrightarrow{\alpha}_{\tau} P' \quad [Q,R] \xrightarrow{\beta}_{\exists\exists} [Q',R'] \quad (P',[Q',R']) \overset{(\alpha_1,\beta_1)}{\Longrightarrow}_{(\Omega_1,C,\Omega_2)} (P_1,[Q_1,R_1])}{(P,[Q,R]) \overset{(\alpha,\beta)}{\Longrightarrow}_{(\Omega_1,C,\Omega_2)} (P',[Q',R'])} \left(\begin{array}{l} \alpha\ \Omega_1 \doteq \Omega_2\ \beta, \\ (P',Q') \in C \end{array} \right)$$

図 4: 無矛盾状態遷移関係 $\overset{(\alpha,\beta)}{\Longrightarrow}_{(\Omega_1,C,\Omega_2)}$ の推論規則

3.2 部分仕様の無矛盾性

本小節では、二つの部分仕様の無矛盾性を定義する。まず、準備として次の遷移関係を定義する。

定義 3.3 マーク付遷移関係 $\longrightarrow_{\exists\exists} \subseteq [\mathcal{P}] \times Act \times [\mathcal{P}]$ は次の推論規則を満たす最小の関係である。

$$\textbf{Mark}\ \frac{P \xrightarrow{\mu} M \quad Q \xrightarrow{\mu} M}{[P,Q] \xrightarrow{\alpha}_{\exists\exists} [P',Q']} \left(\begin{array}{l} \exists i, \mu \triangleleft i = \alpha, \\ M \triangleleft i \equiv [P',Q'] \end{array} \right)$$

また、$P \xrightarrow{\mu} M$ かつ $Q \xrightarrow{\mu} M$ となるような μ と M が存在しない場合、$[P,Q] \not\xrightarrow{}_{\exists\exists}$ と書く。 ∎

定義 3.4 二つの遷移関係 $\longrightarrow_{\exists}, \longrightarrow_{\tau} \subseteq \mathcal{P} \times Act \times \mathcal{P}$ は次の推論規則を満たす最小の関係である。

$$\textbf{Exist}\ \frac{P \xrightarrow{\mu} M}{P \xrightarrow{\alpha}_{\exists} P'} \left(\begin{array}{l} \exists (i,Q'), \mu \triangleleft i = \alpha, \\ M \triangleleft i \equiv [P',Q'] \end{array} \right)$$

$$\textbf{Tau}\ \frac{P \xrightarrow{\alpha}_{\exists} P'}{P \xrightarrow{\alpha}_{\tau} P'} \qquad \textbf{Id}\ \frac{}{P \xrightarrow{\tau}_{\tau} P}$$ ∎

次に、無矛盾な二つの遷移の関係を定義する。

定義 3.5 $\Omega_1, \Omega_2 \subseteq \mathcal{N}$、$C \subseteq \mathcal{P} \times \mathcal{P}$ とする。無矛盾状態遷移関係 $\Longrightarrow_{(\Omega_1,C,\Omega_2)} \subseteq \mathcal{P} \times [\mathcal{P}] \times \langle Act \rangle \times \langle Act \rangle \times \mathcal{P} \times [\mathcal{P}]$ とは、図 4の推論規則を満たす最小の関係である。 ∎

規則 **Base** は、多重仕様とマーカの組 $[Q,R]$ がマーク付遷移 β を実行できるとき、P は β と矛盾しない α を実行でき、実行後は P' と Q' が関係 C を満たすことを表している。ここで、重要なことは $[Q',R']$ がマーク付遷移をもたないことである。もし $[Q',R']$ がマーク付遷移をもつならば、規則 **Base** ではなく、規則 **Conc** を用いなければならない。これが、いつかは $[Q,R]$ がマーク付遷移をもたない状態に到達することを導く。規則 **Seq** は、P が実行すべき α を、内部アクションに相当する α_1 によって延期できることを表している。ただし、P_1 と Q は関係 C を満たさなければならない。これが、弱双模倣性と準弱双模倣性の違いとなる。

二つの部分仕様 $P_1 :: \Omega_1$ と $P_2 :: \Omega_2$ の無矛盾性を判定するために、準弱 (Ω_1,Ω_2) 双模倣性を定義する。準弱 (Ω_1,Ω_2) 双模倣性は次の準弱 (Ω_1,Ω_2) 双模倣の最大の関係である。

定義 3.6 $\Omega_1, \Omega_2 \subseteq \mathcal{N}$ とする。多重仕様の二項関係 $\mathcal{S} \subseteq \mathcal{P} \times \mathcal{P}$ が準弱 (Ω_1,Ω_2) 双模倣 であるとは、$(P,Q) \in \mathcal{S}$ ならば、次の二つの条件が成り立つことである。

(*i*) もし $P \xrightarrow{\mu} M$ ならば、ある i, α, β, P', Q', R' で、
$(Q,[P,P]) \overset{(\beta,\alpha)}{\Longrightarrow}_{(\Omega_2,\mathcal{S}^{-1},\Omega_1)} (Q',[P',R'])$ かつ
$\mu \triangleleft i = \alpha$ かつ $M \triangleleft i \equiv [P',R']$ である。

(*ii*) もし $Q \xrightarrow{\nu} N$ ならば、ある i, α, β, P', Q', R' で、
$(P,[Q,Q]) \overset{(\alpha,\beta)}{\Longrightarrow}_{(\Omega_1,\mathcal{S},\Omega_2)} (P',[Q',R'])$ かつ
$\nu \triangleleft i = \beta$ かつ $N \triangleleft i \equiv [Q',R']$ である。

ここで、$\mathcal{S}^{-1} = \{(Q,P) : (P,Q) \in \mathcal{S}\}$ である。 ∎

定義 3.7 $\Omega_1, \Omega_2 \subseteq \mathcal{N}$ とする。ある準弱 (Ω_1,Ω_2) 双模倣 $\mathcal{S}$ において $(P,Q) \in \mathcal{S}$ ならば、二つの多重仕様 P と Q は準弱 (Ω_1,Ω_2) 双模倣的であるといい、$P\ {}_{\Omega_1}\!\sim_{\Omega_2} Q$ と書く。 ∎

定義 3.6で重要なことは、$\mu \triangleleft i = \alpha$ かつ $M \triangleleft i \equiv [P',R']$ となる i が一つ存在すれば良いことである。これは、多重仕様 P に含まれる複数の基本仕様のなかに、矛盾しない基本仕様が一つあれば十分であることを表している。

本稿では、部分仕様 $P :: \Omega$ の等価性の基準に準弱 (Ω,Ω) 双模倣性を用いる。特に有効アクション集合が名前の集合である準弱 $(\mathcal{N},\mathcal{N})$ 双模倣性を**準弱双模倣性**と呼び、$\simeq$ で表す。基本仕様 P_0, Q_0 に対する準弱 (Ω,Ω) 双模倣性は同値関係であり、強双模倣性 $\sim$ と弱双模倣性 $\approx$ に対して次の関係が成り立つ (基本仕様を CCS の逐次プロセスとみなす)。

$$P_0/\Omega \sim Q_0/\Omega \text{ ならば } P_0\ {}_{\Omega}\!\simeq_{\Omega} Q_0 \text{ ならば } P_0/\Omega \approx Q_0/\Omega$$

ここで、P/Ω は P から有効アクション集合 Ω に含まれない全てのアクションを内部アクション τ に置き換えて得られる仕様であり、CSP の隠蔽演算子 [2] に似ている。例えば、$(a.b.c.d.\mathbf{0})/\{a,c\}$ は $(a.\tau.c.\tau.\mathbf{0})$ である。このように、準弱双模倣性は弱双模倣性よりも強い関係となるが、

$$P \simeq \tau.P\,, \quad P \simeq P + \tau.P\,, \quad P + Q \simeq \tau.(P+Q) + P$$

のように弱双模倣性の興味深い特性は残されている。弱双模倣的であるが、準弱双模倣的でない典型的な例を示す。

$$\alpha.(P + \tau.Q) \not\simeq \alpha.(P + \tau.Q) + \alpha.Q$$

準弱 (Ω,Ω) 双模倣性を基に、部分仕様 $P::\Omega$ を満たす基本仕様の集合を次のように定義する。

$$\mathbf{G}(P,\Omega)=\{P_0 : P_0 \ {}_{\Omega}\!\simeq_{\Omega} P, P_0 \in \mathcal{P}_0\}$$

このとき、共通仕様を次のように定義する。

定義 3.8 次の条件を満たす基本仕様 P_0 を n 個の部分仕様 $Q_1::\Omega_1,\cdots,Q_n::\Omega_n$ の共通仕様という。

$$P_0 \in \bigcap_{1\le i\le n} \mathbf{G}(Q_i,\Omega_i) \qquad \blacksquare$$

次の命題より、準弱 (Ω_1,Ω_2) 双模倣性を用いて二つの部分仕様 $P_1::\Omega_1$ と $P_2::\Omega_2$ の共通仕様の存在を判定できる。

命題 3.2 任意の $P_1,P_2\in\mathcal{P}$、$\Omega_1,\Omega_2\subseteq\mathcal{N}$ について、

$$P_1 \ {}_{\Omega_1}\!\simeq_{\Omega_2} P_2 \iff \mathbf{G}(P_1,\Omega_1)\cap\mathbf{G}(P_2,\Omega_2)\neq\emptyset \qquad \blacksquare$$

3.3 多重仕様の例

本小節で、*MSPEC* の特徴であるマーカについて、次の多重仕様 $ABC::\{a,b\}$ の例を用いて説明する。

$$ABC \stackrel{\text{def}}{=} AC + BC \qquad AC \stackrel{\text{def}}{=} a.[ABC,AC]\vee b.[ABC,\mathbf{0}]$$
$$BC \stackrel{\text{def}}{=} a.[ABC,\mathbf{0}]\vee b.[ABC,BC]$$

この ABC は「常にいつかはアクション a を起こすことができ、かつ常にいつかはアクション b を起こすことができる全ての基本仕様」を表している。このような基本仕様は、例えば次のように無限に存在する。

$$AB_1 \stackrel{\text{def}}{=} a.a.b.AB_1, \quad AB_2 \stackrel{\text{def}}{=} a.AB_2 + b.AB_2$$

重要なことは、ABC のマーカ AC が、a だけを実行する基本仕様 $(A \stackrel{\text{def}}{=} a.A)$ を排除していることである。これは次のように説明できる:ABC から a が選択された場合、ABC の遷移 a には AC によってマークが付けられる。ここで、遷移 b が選択されるとマークは消えるが、遷移 a が選択されると再度 ABC の遷移 a にマークが付けられる。この繰り返しにより、ABC からはいつか b が選択されることになる。

このようにして、「いつかは起こせるアクション」を記述することができる。Manna と Wolper は要求を命題時相論理で記述し、その要求を満たす仕様を生成する方法を提案した [7]。例えば、上記の「常にいつかは a を起こすことができ、かつ常にいつかは b を起こすことができる」要求は命題時相論理では $(\Box\Diamond a)\wedge(\Box\Diamond b)$ と記述できる。ここで、$\Box$ 演算子は「常に」を意味し、$\Diamond$ 演算子は「いつかは」を意味している。[7] でも、要求を満たす全ての仕様を生成することを目的にしているが、上記のように $\Diamond$ 演算子を含む場合は必ずしも全ての仕様を生成しない問題がある。

4 主仕様の合成法

本節では、複数の部分仕様の全ての共通仕様を表す多重仕様を合成する方法を与える。これは、全ての部分仕様を満たすための最も弱い要求を表している。我々はそのような多重仕様を主仕様と呼び、次のように定義する。

定義 4.1 次の条件を満たす多重仕様 P を n 個の部分仕様 $Q_1::\Omega_1,\cdots,Q_n::\Omega_n$ の主仕様という。

$$\mathbf{G}(P,\Omega)=\bigcap_{1\le i\le n}\mathbf{G}(Q_i,\Omega_i) \neq\emptyset$$

ここで、$\Omega=\bigcup_{i\in[1,n]}\Omega_i$ である。 $\blacksquare$

次に、二つの部分仕様 $Q_1::\Omega_1$ と $Q_2::\Omega_2$ から多重仕様 $\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1,Q_2)$ を生成する **PQ** 法を与える。

定義 4.2 $\Omega_1,\Omega_2\subseteq\mathcal{N}$、$Q_1,Q_2\in\mathcal{P}$ とする。このとき、多重仕様 $\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1,Q_2)$ を次のように定義する。

$$\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1,Q_2) \stackrel{\text{def}}{=} \mathbf{PR}^{\Omega_1}_{\Omega_2}(Q_1,Q_2)+\mathbf{PR}^{\Omega_2}_{\Omega_1}(Q_2,Q_1)$$
$$\mathbf{PR}^{\Omega_1}_{\Omega_2}(Q_1,Q_2) \stackrel{\text{def}}{=} \sum\{\mathbf{PA}^{\Omega_1}_{\Omega_2}(Q_1,\nu_1,N_1,Q_2) : Q_1 \xrightarrow{\nu_1} N_1\}$$

$$\mathbf{PA}^{\Omega_1}_{\Omega_2}(Q_1,\nu_1,N_1,Q_2) \stackrel{\text{def}}{=}$$
$$\bigvee\{\alpha.[\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1',Q_2'),\mathbf{0}] : (\alpha,Q_1',Q_2')\in E^{\Omega_1}_{\Omega_2}(\nu_1,N_1,Q_2)\}$$
$$\cup\{\alpha.[\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1',Q_2'),\mathbf{PA}^{\Omega_1}_{\Omega_2}(Q_1',\nu_1',N_1',Q_2')] : (\alpha,Q_1',\nu_1',N_1',Q_2')\in F^{\Omega_1}_{\Omega_2}(\nu_1,N_1,Q_2)\}$$
$$\cup\{\alpha.[\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1,Q_2'),\mathbf{PA}^{\Omega_1}_{\Omega_2}(Q_1,\nu_1,N_1,Q_2')] : (\alpha,Q_2')\in G^{\Omega_1}_{\Omega_2}(Q_1,Q_2)\}$$

ここで、$E^{\Omega_1}_{\Omega_2}(\nu_1,N_1,Q_2)$, $F^{\Omega_1}_{\Omega_2}(\nu_1,N_1,Q_2)$, $G^{\Omega_1}_{\Omega_2}(Q_1,Q_2)$ は次のように定義される集合である。

$$E^{\Omega_1}_{\Omega_2}(\nu_1,N_1,Q_2)=\{(\alpha,Q_1',Q_2') : \exists(\beta_1,\beta_2,j_1,R_1'),$$
$$\nu_1\lhd j_1=\beta_1, N_1\lhd j_1\equiv[Q_1',R_1'], Q_2\xrightarrow{\beta_2}_{\tau} Q_2',$$
$$(\beta_1,\Omega_1)\bullet(\beta_2,\Omega_2)=\alpha, Q_1' \ {}_{\Omega_1}\!\simeq_{\Omega_2} Q_2', [Q_1',R_1']\not\longrightarrow_{\exists\exists}\}$$

$$F^{\Omega_1}_{\Omega_2}(\nu_1,N_1,Q_2)=\{(\alpha,Q_1',\nu_1',N_1',Q_2') : \exists(\beta_1,\beta_2,j_1,R_1'),$$
$$\nu_1\lhd j_1=\beta_1, N_1\lhd j_1\equiv[Q_1',R_1'], Q_2\xrightarrow{\beta_2}_{\tau} Q_2',$$
$$(\beta_1,\Omega_1)\bullet(\beta_2,\Omega_2)=\alpha, Q_1' \ {}_{\Omega_1}\!\simeq_{\Omega_2} Q_2',$$
$$Q_1'\xrightarrow{\nu_1'} N_1', R_1'\xrightarrow{\nu_1'} N_1'\}$$

$$G^{\Omega_1}_{\Omega_2}(Q_1,Q_2)=\{(\alpha,Q_2') : \exists\beta_2,$$
$$Q_2\xrightarrow{\beta_2}_{\exists} Q_2', (\tau,\Omega_1)\bullet(\beta_2,\Omega_2)=\alpha, Q_1 \ {}_{\Omega_1}\!\simeq_{\Omega_2} Q_2'\} \qquad \blacksquare$$

以下、定義 4.2の合成過程を各多重仕様ごとに簡単に説明する。$\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1,Q_2)$ は、Q_1 の各遷移に対応する Q_2 の全ての遷移を探索する $\mathbf{PR}^{\Omega_1}_{\Omega_2}(Q_1,Q_2)$ と Q_2 の各遷移に対応する Q_1 の全ての遷移を探索する $\mathbf{PR}^{\Omega_2}_{\Omega_1}(Q_2,Q_1)$ から構成される。$\mathbf{PA}^{\Omega_1}_{\Omega_2}(Q_1,\nu_1,N_1,Q_2)$ は Q_1 の遷移 $Q_1\xrightarrow{\nu_1}N_1$ に対応する Q_2 の遷移を探索し、選択関数を用いて有効なアクションによる遷移を生成する。**PA** の定義式右辺の第 1 項と第 2 項は、Q_1 の遷移に対応する Q_2 の遷移がある場合である。ただし、Q_1' が有効なマーカ R_1' をもつときは、修正されたマーカ $\mathbf{PA}^{\Omega_1}_{\Omega_2}(Q_1',\nu_1',N_1',Q_2')$ が付けられる。第 3 項は遷移 $Q_1\xrightarrow{\nu_1}N_1$ が Q_2 の遷移 $Q_2\xrightarrow{\beta_2}_{\exists}Q_2'$ によって延期される場合であり、延期された遷移がいつかは実行されるように新たにマーカ $\mathbf{PA}^{\Omega_1}_{\Omega_2}(Q_1,\nu_1,N_1,Q_2')$ が付けられる。

集合 $E^{\Omega_1}_{\Omega_2}(\nu_1,N_1,Q_2)$ と $F^{\Omega_1}_{\Omega_2}(\nu_1,N_1,Q_2)$ の和集合は、遷移 $Q_1\xrightarrow{\nu_1}N_1$ に矛盾しない全ての Q_2 の遷移を表している。特に、前者は遷移後にマーカが無効な場合、後者は有効な場合である。また、集合 $G^{\Omega_1}_{\Omega_2}(Q_1,Q_2)$ は、Q_1 の全ての遷移を延期できる全ての Q_2 の遷移を表している。

次に $\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1,Q_2)$ に期待される重要な命題を示す。

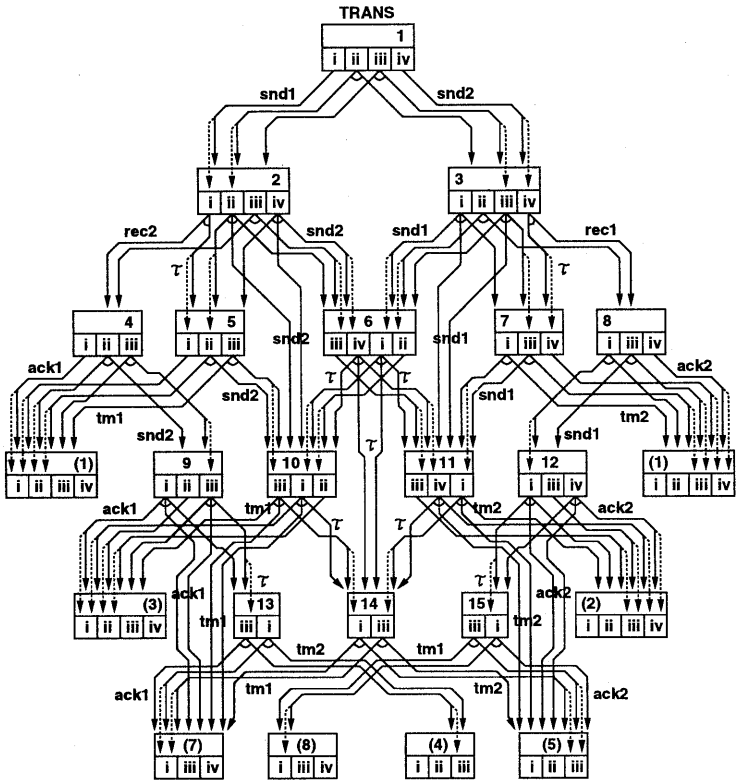

図 5: 主仕様 *TRANS* の状態遷移図

命題 4.1 もし $Q_1 \ {}_{\Omega_1}\!\simeq_{\Omega_2} \ Q_2$ ならば、$\mathbf{PQ}^{\Omega_1}_{\Omega_2}(Q_1, Q_2)$ は部分仕様 $Q_1 :: \Omega_1$ と $Q_2 :: \Omega_2$ の主仕様である。 ∎

最後に、三つ以上の部分仕様の主仕様については、定義 4.1と命題 4.1より、次の定理が得られる。また、この定理は共通仕様が存在するかを判定する方法も示している。

定理 4.2 P_i $(i \in \{1,2\})$ を $Q_{i1} :: \Omega_{i1}, \cdots, Q_{in_i} :: \Omega_{in_i}$ の主仕様とする。このとき、次の関係が成り立つ。

(1) $P_1 \ {}_{\Omega_1}\!\simeq_{\Omega_2} \ P_2$ ならば、$\mathbf{PQ}^{\Omega_1}_{\Omega_2}(P_1, P_2)$ は
$Q_{i1} :: \Omega_{i1}, \cdots, Q_{in_i} :: \Omega_{in_i}$ $(i \in \{1,2\})$ の主仕様である。

(2) $P_1 \ {}_{\Omega_1}\!\not\simeq_{\Omega_2} \ P_2$ ならば、$Q_{i1} :: \Omega_{i1}, \cdots, Q_{in_i} :: \Omega_{in_i}$
$(i \in \{1,2\})$ の共通仕様は存在しない。

ここで、$\Omega_i = \bigcup_{1 \le j \le n_i} \Omega_{ij}$ である。 ∎

5 仕様合成の例

本節では、1節で記述した通信機の部分仕様を用いて **PQ** 法による合成例を示す。まず、$ST_i \ {}_{\Omega_{STi}}\!\simeq_{\Omega_{TRij}} TR_{ij}$ を証明できるので、命題 3.2より $ST_i :: \Omega_{STi}$ と $TR_{ij} :: \Omega_{TRij}$ の共通仕様は存在する。よって、**PQ** 法により合成された次の仕様は、$ST_i :: \Omega_{STi}$ と $TR_{ij} :: \Omega_{TRij}$ の主仕様である。

$$STR_i \equiv \mathbf{PQ}^{\Omega_{STi}}_{\Omega_{TRij}}(ST_i, TR_{ij})$$
$$\stackrel{\text{def}}{=} snd_i.(rec_j.ack_i.STR_i + \tau.tm_i.STR_i) + rec_i.STR_i$$

さらに、$STR_1 \ {}_{\Omega_{STR1}}\!\simeq_{\Omega_{STR2}} \ STR_2$ の関係が得られる。ここで、$\Omega_{STRi} = \Omega_{STi} \cup \Omega_{TRij}$ である。これは 4 つの部分仕様 $ST_1, ST_2, TR_{12}, TR_{21}$ の共通仕様が存在することを意味している。そこで、**PQ** 法により合成されたその 4 つの部分仕様の主仕様 $TRANS \equiv \mathbf{PQ}^{\Omega_{STR1}}_{\Omega_{STR2}}(STR_1, STR_2)$ の状態遷移図を図 5に示す。ただし、状態 1 から状態 3 への三つの遷移は同じラベル snd_1 をもつので、同じラベルは省略している。他の遷移についても同様である。また、下方の括弧付きの番号をもつ状態は上方のその番号の状態と同じであり、繰り返し動作を表している。

図 5には、二つまたは三つの遷移が弧で結ばれている分岐がある。多重演算子の特徴から、そのような分岐では、遷移を一つ残して他の遷移を消すことができる。このようにして複数 (この例では無限大) の基本仕様を得ることができる。ただし、いつかは無効なマーカがあるように遷移を消す必要がある。例えば、状態 2 から状態 4 への rec_2 による二つの遷移を消した場合、マーカ (点線矢印) に注目すると

$$[1,1(i)] \xrightarrow{snd_i}_{\exists\exists} [2,2(i)] \xrightarrow{\tau}_{\exists\exists} [5,5(i)] \xrightarrow{ack_1}_{\exists\exists} [1,1(i)]$$

となり、無効なマーカに到達できない。遷移 rec_2 によって、$[2,2(i)] \xrightarrow{rec_2}_{\exists\exists} [4,\mathbf{0}]$ のように無効なマーカに到達できる。

主仕様 *TRANS* に含まれる基本仕様は全て 4 つの部分仕様 $ST_1, ST_2, TR_{12}, TR_{21}$ の共通仕様であり、その全ての共通仕様は *TRANS* に含まれている。主仕様 *TRANS* に含まれる

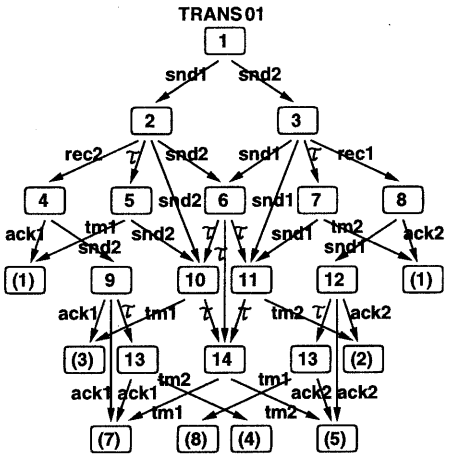

図 6: 基本仕様 $TRANS_{01}$ の状態遷移図

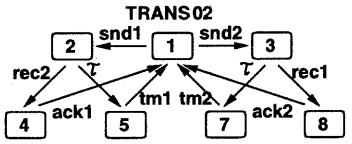

図 7: 基本仕様 $TRANS_{02}$ の状態遷移図

一つの基本仕様は、主仕様の全ての多重演算子 $\vee$ を選択演算子 $+$ に置き換え、全てのマーカを $\mathbf{0}$ に置き換えて得られる。そうして得られた基本仕様 $TRANS_{01}$ を図6に示す。弱双模倣性は準弱双模倣性を含むので、次の関係が成り立つ。

$$TRANS_{01}/\Omega_{STi} \approx ST_i, \quad TRANS_{01}/\Omega_{TRij} \approx TR_{ij}$$

この基本仕様 $TRANS_{01}$ はそのまま逐次プロセスとして実装することもできる。これをさらに並行プロセスとして実装する方法は [8] 等に与えられている。

次に、$\Omega_{EX} = \{snd_1, snd_2, ack_1, ack_2, tm_1, tm_2\}$ を有効アクション集合にもつ次の部分仕様を用意する。

$$EX \stackrel{\text{def}}{=} snd_1.(\tau.ack_1.EX + \tau.tm_1.EX) + snd_2.(\tau.ack_2.EX + \tau.tm_2.EX)$$

この部分仕様は、一方が送信 snd_i した場合、返信 ack_i か時間切れ tm_i を受けるまでは、他方が送信 snd_j できないことを表している。この部分仕様 EX がすでにある4つの部分仕様と共通仕様をもつことは、$TRANS\ _{\Omega_{TRANS}}\simeq_{\Omega_{EX}} EX$ により示される。ここで、$\Omega_{TRANS} = \Omega_{STR1} \cup \Omega_{STR2}$ である。よって、$\mathbf{PQ}_{\Omega_{EX}}^{\Omega_{TRANS}}(TRANS, EX)$ は5つの部分仕様の主仕様となる。図7にその主仕様に含まれる一つの基本仕様 $TRANS_{02}$ を示す。

ここで重要なことは、$TRANS_{01}\ _{\Omega_{TRANS}}\not\simeq_{\Omega_{EX}} EX$ となることである。基本仕様 $TRANS_{01}$ のような、4つの部分仕様 $ST_1, ST_2, TR_{12}, TR_{21}$ の共通仕様を一つ合成することは、主仕様 $TRANS$ を合成するよりも非常に容易である。しかし、この $TRANS_{01}$ からでは、EX がすでにある4つの部分仕様と共通仕様をもつことは的確に判定できない。そこで、与えられた部分仕様を満たす最も弱い要求を表すような主仕様 $TRANS$ が必要となる。定理4.2に示されるように、主仕様 $TRANS$ と共通仕様をもたないことは、すでに与えられた部分仕様と共通仕様をもたないことと同じである。**PQ** 法により、設計者は簡単な4つの部分仕様を記述するだけで、このような主仕様を得ることができる。

6 おわりに

本稿では、複数の部分仕様の共通仕様が存在するかを判定する方法と、そのような全ての共通仕様を表す多重仕様(主仕様)を合成する方法を提案した。本稿では多重仕様を記述するために言語 *MSPEC* を与えている。

多重仕様は多重演算子とマーカをもつため、設計者が多重仕様を理解し記述することは容易ではない。しかし、**PQ** 法をもとに設計支援システムを実装し、多重仕様から基本仕様を得る方法を自動化すれば、一般に設計者は基本仕様(ラベル付遷移システム)の知識だけで仕様を合成することができる。主仕様はその後に追加される部分仕様に対応するために、設計支援システム内に保存される。

弱双模倣性は準弱双模倣性に比べて定義が複雑であるため、今回は準弱双模倣性をもとしにた合成方法を検討した。ここで得られた成果は次の課題である弱双模倣性を基にした主仕様について検討するために重要な役割を果たす。

参考文献

[1] Milner, R.: *Communication and Concurrency*, Prentice-Hall, 1989.

[2] Hoare, C.A.R.:*Communicating Sequential Processes*, Prentice-Hall, 1985.

[3] Kimura, S., Togashi, A., and Shiratori, N.: Synthesis Algorithm for Recursive Processes by μ-calculus, Algorithmic Learning Theory, LNCS 872,Springer-Verlag, pp.379-394, 1994.

[4] Steen, M.W.A., Bowman, H., and Derrick, J.: Composition of LOTOS specification, Protocol Specification, Testing and Verification, XV, pp.73-88, 1995

[5] Brinksma, E.: Constraint-oriented specification in a constructive formal description technique, LNCS 430, Springer-Verlag, pp.130-152, 1989.

[6] Kleuker, S.: Incremental Development fo Deadlock-Free Communicating Systems, LNCS 1217, Springer-Verlag, pp.306-320, 1997.

[7] Manna, Z., and Wolper, P.: Synthesis of Communicating Processes from Temporal Logic Specifications, *ACM Trans. on Programming Languages and Systems*, Vol.6, No.1, pp.67-93, 1984.

[8] Langerak, R.: Decomposition of functionality : a correctness preserving LOTOS transformation, Protocol Specification, Testing and Verification, X, pp.229-242, 1990.

[9] 磯部祥尚, 中田秀基, 佐藤豊, 大蒔和仁 : 強フィルタ双模倣を基にした部分仕様の段階的合成, 第10回 回路とシステム軽井沢ワークショップ論文集, pp.327-332, 1997.