自動組立手順生成のためのペトリネットの段階的詳細化*

川瀬 俊夫^{*1}, 包原 孝英^{*1}, 鈴木 達也^{*1}
大熊 繁^{*2}, 稲葉 昭夫^{*3}Stepwise Refinement of Petri Net for generating
Automatic Assembly SequenceToshio KAWASE, Takahide KANEHARA, Tatsuya SUZUKI,
Shigeru OKUMA and Akio INABA

There is a growing need for top-down design methodology for a discrete event system. In top-down design of the discrete event system, we begin with a high-level model of the system and refine it step-wisely until a sufficiently low-level model is obtained. In the previous researches, most of the proposed refinement technique has been a "transition-based technique", which replaces each transition with subnet. However, in case of assembly Petri net, each place has useful informations for the refinement rather than each transition. This means that a "place-based refinement technique", is needed for assembly Petri net. In this paper, we propose a new place-based refinement technique of Petri net for automatic assembly procedure. In the proposed method, we assign some attributes to places and set the input-output relationships of transitions which connect the place-attribute combinations. The places can be divided into groups according to their attributes and we should take into account only transitions between these groups. This leads to that we can refine the Petri net easily comparing with the usual transition-based technique.

Key Words: Assembly, Production System, Modelling, Stepwise Refinement, Petri Net, Discrete Event System

1. 緒 言

近年、大規模な離散事象システムに対する体系的な設計手法の確立が望まれている。一般に、大規模なシステムの設計においては、システムを段階的に詳細化していくトップダウン手法が用いられる。システムのモデル化に段階的な詳細化手法を用いれば、一度に詳細なモデルを生成するよりも効率の良いモデル化が可能と考えられる。一方、離散事象システムのモデル化ツールとして、ペトリネット⁽¹⁾⁽²⁾の有効性が認識されつつある。筆者らは、これまで製品の組立手順をモデル化したAND/ORグラフ⁽³⁾⁽⁴⁾⁽⁵⁾をベースとするペトリネットモデル(以下、組立ペトリネットと呼ぶ)を提案してきた⁽⁶⁾。組立ペトリネットにおいても、より詳細なレベルまで作業手順を計画するためには、組立作業を行う半製品の場所や、半製品内の固定部品、及

びその固定方向などの自動組立システムの作業環境を含めた詳細なモデル化が要求される⁽⁷⁾が、そのような詳細なモデルを一度に生成することは、ネットを構成するプレース及びトランジションの数が、製品の部品数の増加に伴い膨大になるため非常に困難である。従って、組立ペトリネットに関しても、段階的な詳細化によるモデル生成が必要となる。

これまで提案されてきた一般のペトリネットモデルに対する段階的な詳細化手法としては、ネットを構成する各トランジションを、そのトランジションが表すタスクに対応するより詳細なモデル(サブネット)で置換するというトランジションベースの手法がほとんどであった⁽⁸⁾⁽⁹⁾⁽¹⁰⁾。これは、これまでのペトリネットを用いた離散事象システムの表現においては、主として、プレースは、条件あるいはシステムの状態を表していたのに対し、トランジションは何かの事象(タスク)を表していたため、トランジションの方が詳細化に関する情報をより多く含んでいたためであ

* 原稿受付 平成5年10月29日。

^{*1} 名古屋大学工学部 (〒464-01 名古屋市千種区不老町)。^{*2} 正員, 名古屋大学工学部。^{*3} 岐阜県金属試験場 (〒501-32 関市小瀬 1288)。

る。しかし、筆者らの提案した組立ペトリネットにおいては、各プレースは組立過程における半製品を表しており、その場合、各プレースは半製品の置き場や、固定部品といった様々な属性を有する状態とみなすことができる。このように詳細化すべき情報が工学的な意味を持ったプレースの属性として定義できる場合には、簡単な操作によりプレースの集合が属性付きプレースの集合に拡張され、後はトランジションの入出力関係を規定するだけでその属性を考慮に入れたプレースベースの詳細化が可能となる。これに対し、トランジションベースの詳細化手法を用いる場合、サブネット内のプレースとトランジションの両方を考慮しなければならず膨大な労力を必要とする恐れがある。

以上の点をふまえて、本論文では、組立ペトリネットに対するプレースベースの段階的詳細化手法を提案する。本論文で提案する手法は、ペトリネットのプレースに属性を与えることによって、半製品のより詳細な状態とその状態遷移を表すタスクを表現するペトリネットモデルを生成するというプレースベースの詳細化手法である。この手法を用いてペトリネットモデルを詳細化するためには、詳細化後のトランジションの入出力関係に相当する属性付きプレース間の状態遷移に関する情報が必要であるが、半製品の状態のように詳細化すべき情報が、プレースの属性として与えられる場合には、詳細化後のプレースを工学的に共通の意味を持つプレースごとにグループ化することが可能となる。従って、提案する手法においては、詳細化後のプレースのグループ化及びそのグループ間の状態遷移を考慮するだけで、ネットを容易に詳細化することができる。また、本手法は組立ペトリネットが有する様々なグラフ論的性質を保存するため、詳細化後の組立ペトリネットにおける最適作業手順の探索にそのまま線形計画法⁽¹¹⁾が適用できるという利点がある。

以下、2章において、本論文で用いる記号と用語の定義を行い、3章において、組立ペトリネットの構成について述べる。4章では、まず詳細化の際に用いる詳細化情報の定義および提案する詳細化手法を示し、次に詳細化の前後におけるネットの諸性質の保存について考察する。そして5章で、実際の組立ペトリネットに本手法を適用した例を示し、本手法の有効性を示す。6章はまとめである。

2. 諸 定 義

本論文で用いる記号と用語の定義を行う。

ペトリネットは、プレース、トランジション(それぞれ、丸印○、棒印—で表される)と呼ばれる2種類のノードの集合と、それらを接続する有向アークで構成される2部グラフである。プレース上に置かれるトークン(黒丸●で表される)の配置をマーキングといい、マーキングを用いて

図1 製品のモデル

ペトリネットの状態を表現する。

(定義1) ペトリネット

ペトリネット N は $N = (P, T, IN, OUT)$ の4項組により構成される。ここで、 P をプレースの集合、 T をトランジションの集合 ($P \cap T = \emptyset$)、 $IN: T \times P \rightarrow Z^+$ を入力関数、 $OUT: T \times P \rightarrow Z^+$ を出力関数とする。また、マーキングを $M: P \rightarrow Z^+$ で表す。ただし、 Z^+ を非負整数の集合とする。

(定義2) 発火条件

ペトリネット N において、 $\forall p \in P$ に対して、

$$IN(t, p) \leq M(p) \dots\dots\dots (1)$$

である時、トランジション t はマーキング M で発火可能であるという。

(定義3) マーキング遷移

マーキング M_1 は、 M_1 で発火可能なトランジション t が発火すると、

$$\forall p \in P; M_2(p) = M_1(p) - IN(t, p) + OUT(t, p) \dots\dots\dots (2)$$

で定まるマーキング M_2 に遷移する。

(定義4) 状態方程式

ペトリネットの状態方程式は以下のように定義されている。

$$Dx = M - M_0 \dots\dots\dots (3)$$

ここで、 D は状態遷移行列、 x はトランジションの発火回数ベクトル、 M は現在のマーキングベクトル、 M_0 は初期マーキングベクトルである。

状態遷移行列は、行がプレース、列がトランジションに対応している。状態遷移行列の各要素は、 $IN(t_j, p_i) = a$ ならば $D_{ij} = -a$ 、 $OUT(t_j, p_i) = a$ ならば $D_{ij} = a$ ($a \in Z^+$)、他の要素は0である。

マーキングベクトル、初期マーキングベクトルは以下のように定義される。

$$M = [M(p_1), \dots, M(p_n)]^T \dots\dots\dots (4)$$

$$M_0 = [M_0(p_1), \dots, M_0(p_n)]^T \dots\dots\dots (5)$$

ただし、 n はプレースの総数である。

3. 組立ペトリネットとその性質

組立ペトリネットは製品の組立手順をペトリネットによりモデル化したもので、複数の単一部品から完成された製品へのすべての可能な組立手順を表現している⁽⁶⁾。図1の製品に対する組立ペトリネットを図2に示す。図2の各プレースが表す状態を以下に示す。 $p_1 = \{\text{CSRH}\}$, $p_2 = \{\text{CSR}\}$, $p_3 = \{\text{SRH}\}$, $p_4 = \{\text{SR}\}$, $p_5 = \{\text{CS}\}$, $p_6 = \{\text{CR}\}$, $p_7 = \{\text{RH}\}$, $p_8 = \{\text{SH}\}$, $p_9 = \{\text{C}\}$, $p_{10} = \{\text{S}\}$, $p_{11} = \{\text{R}\}$, $p_{12} = \{\text{H}\}$

図2に示される組立ペトリネットは、半製品の部品構成のみを状態として区別するAND/ORグラフ⁽³⁾⁽⁴⁾⁽⁵⁾から、ノードをプレースに、向きを逆にしたアークを2入力1出力のトランジションに置換することによってペトリネットに変換したモデルである。なお、組立手順を表すAND/ORグラフの生成法については、文献(3)(4)(5)で議論されており、ここでは省略する。

組立ペトリネットでは、各トランジションにより2つの半製品を1つの半製品に組み立てるタスクを表現し、各プレースにより対応する半製品を表現している。また、製品の組立状態はマーキングに対応づけることによって表現される。任意の組立手順は、単一部品に対応するプレースにのみ1個のトークンがあるマーキングを初期マーキング、完成された製品に対応するプレースにのみ1個のトークンがあるマーキングを最終マーキングとして、マーキングを遷移させるトランジションの発火系列によって与えられる。

次に、組立ペトリネットが有するグラフ論的かつ代数学的特徴について述べておく。図2に示されるように、組立ペトリネットは本質的に複数のツリーが重ね合わされた構造をとるため、グラフ論的には、以下の性質を持つ。

(性質1) 組立ペトリネットは弱連結ネットであり、製品及び単一部品に対応するプレースは、それぞれ逆ペトリ

ネットにおけるルート及びリーフである。

(性質2) すべてのトランジションは1入力1出力または2入力1出力である。

(性質3) 各トランジションの異なる2つの入力アークは同一のプレースからのパスを持たない。

文献(6)によれば、ペトリネットが性質1~3を満たす場合、(3)式において、実際の組立手順を表す解の集合 X_{asm} と基底解から成る集合 X_{bas} は完全に一致することが明らかにされている。このことは各トランジションに重みを割当て、その重みの総和が最小となるような最適組立手順を探索する問題を、整数解制約のない線形計画法を用いて解くことができることを意味しており、このことが代数学的に見た時の組立ペトリネットの大きな長所である。また、性質1~3を満たすペトリネットがセーフで、目標マーキング以外においてデッドロックとならないことが文献(12)において明らかにされている。

4. プレースの属性による詳細化

4.1 詳細化手法 本節では、ペトリネットのプレースに属性を与えることによって、半製品のより詳細な状態を表現し、属性付きプレース間の状態遷移に関する情報を用いて詳細化後のトランジションの入出力関係を規定し、その結果、より詳細なペトリネットモデルを生成するというプレースベースの詳細化手法について述べる。

初めに、プレース集合 P を詳細化する属性に関して拡張した属性付きプレースの集合 P' を定義する。

(定義5) 属性付きプレース

$$P' = \{[p_i, u_q]; i = 1, \dots, n, q = 1, \dots, s\} \dots (6)$$

ただし、 p_i はプレース集合 P に含まれるプレース、 u_q はプレースに付加される属性、 n はプレースの総数、 s は属性の総数とする。

ペトリネットモデルをプレースベースで詳細化するためには、詳細化後のトランジションの入出力関係に相当する属性付きプレース間の状態遷移に関する情報が必要であるが、この情報をすべての属性付きプレース間に対して個別に与えようとすると、ネットのノード数の増加とともに、詳細化の労力が著しく増大するという問題がある。しかしながら実際には、 P' の各要素を工学的に共通な意味を持つプレースごとにグループ化することが可能で、そのグループ間の状態遷移のみを考慮することによって、詳細化後のトランジションを容易に生成することができる。

図3は2つの部品A,Bからなる製品に対する組立ペトリネットを半製品の場所に関して詳細化する例を示す。図3の左図のネットに対して、プレース $p_i (i = 1, 2, 3)$ を属性付きプレース $[p_i, u_q] (i = 1, 2, 3, q = 1, 2)$ に拡張した後、トランジション t_1 に対応する属性付きプレース間の状態遷移を

図2 ペトリネット1

図3 プレースペースの詳細化

考慮すると、 $[p_2, u_1], [p_3, u_2] \rightarrow [p_1, u_2], [p_3, u_1], [p_2, u_2] \rightarrow [p_1, u_2]$, という2つの状態遷移が挙げられる。ただし、 u_1 は部品置き場 (Parts)、 u_2 は作業所 (Work) を表す。そして、部品を表すプレースの集合 $X_1 = \{p_2, p_3\}$ 、製品を表すプレースの集合 $X_2 = \{p_1\}$ のようにプレースをグループ化することによって、 $[X_1, u_1], [X_1, u_2] \rightarrow [X_2, u_2]$ のように、より簡潔に表現することが可能となる。このように工学的に共通な意味を持つプレースごとにグループ化された情報を詳細化情報として以下のように定義する。

(定義6) 詳細化情報

$$R = \{\delta_k, X_i, u_q; k, i, q \in Z^+\} \dots\dots\dots (7)$$

$$\delta_k : [X_{i'}, u_{q'}], [X_{i''}, u_{q''}] \rightarrow [X_i, u_q] \dots\dots\dots (8)$$

または、

$$\delta_k : [X_{i'}, u_{q'}] \rightarrow [X_i, u_q] \dots\dots\dots (9)$$

ただし、 R は詳細化情報、 $\delta_k (k \in Z^+)$ は属性付きプレース間の状態遷移規則 (以後、遷移則と呼ぶ) である。 $X_i (i \in Z^+)$ は、プレースの集合 P の部分集合で、ある共通の工学的意味を持つプレースの集合、 $u_q (q \in Z^+)$ は詳細化する属性である。

ただし、この詳細化情報は以下の詳細化条件を満たすものとする。

(定義7) 詳細化条件

$$(1) \forall [p_i, u_q] \in P'_0; \exists k [p_i, u_q] \in \text{src}(\delta_k) \dots\dots (10)$$

$$(2) \forall [p_i, u_q] \in P'_F; \exists k [p_i, u_q] \in \text{dest}(\delta_k) \dots\dots (11)$$

$$(3) \forall k_1 [p_i, u_q] \in \text{dest}(\delta_{k_1}) \wedge [p_i, u_q] \notin P'_F \\ \Rightarrow \exists k_2 [p_i, u_q] \in \text{src}(\delta_{k_2}) \dots\dots\dots (12)$$

ただし、 P'_0 は、初期マーキングにおいてトークンのある属性付きプレースの集合で、単一部品を表す属性付きプレースの集合である。 P'_F は、最終マーキングにおいてトークンのある属性付きプレースの集合で、製品を表す属性付きプレースの集合である。 $\text{src}()$ は、遷移則の遷移元に含まれる属性付きプレースの集合、 $\text{dest}()$ は、遷移則の遷移先に含まれる属性付きプレースの集合である。

同種の属性に関して詳細化する場合は、組立ペトリネットの構造が異なる場合でも、遷移則はそのまま、プレースの集合 $X_i (i \in Z^+)$ の要素のみを変更すれば良い。

以下に、詳細化情報を参照して組立ペトリネットの状態遷移行列 D から詳細化モデルの状態遷移行列 D' を生成するアルゴリズムを示す。図4はアルゴリズムの概略を示す。このアルゴリズムを適用し、状態遷移行列と入力及び出力関数とを変換することによって、組立ペトリネット $N = \langle P, T, IN, OUT \rangle$ は、 $N' = \langle P', T', IN', OUT' \rangle$ に詳細化される。

アルゴリズム

I. 詳細化情報 R を読み込む。

II. プレースの集合 P を属性付きプレースの集合 P' に拡張する (定義5 参照)。トランジションの集合 T' 、状態遷移行列 D' を、 $T' = \phi, D' = O$ とする。

III. トランジションの集合 T' 及び状態遷移行列 D' を以下のように生成する。

[1] トランジションの集合 T に含まれるすべてのトランジションについて以下の操作を行う。

(1) $t_j \in T$ について入力及び出力プレースを検索する。すなわち、 $D_{ij} = -1$ ならば、 p_i は入力プレース、 $D_{ij} = 1$ ならば、 p_i は出力プレースであるとする。

(2) 入力プレース $p_{i_1} (, p_{i_2})$ が遷移則の遷移元のプレースの集合に含まれ、かつ、出力プレース p_{i_3} が遷移則の遷移先のプレースの集合に含まれるかどうか調べる。含まれないならば、次の遷移則へ。

(3) 対応する遷移則を $\delta_k : [X_{*}, u_{q_1}], [X_{*}, u_{q_2}] \rightarrow [X_{*}, u_{q_3}]$ とすると (ただし、 X_{*} は詳細化情報に含まれる任意のプレース集合)、入力プレースが $[p_{i_1}, u_{q_1}], [p_{i_2}, u_{q_2}]$ 、出力プレースが $[p_{i_3}, u_{q_3}]$ であるトランジション $t_{j'}$ を T' に追加する。そして、状態遷移行列 D' の第 j' 列を以下のように規定する。

$$D'_{s \times (i_1-1)+q_1, j'} (= D'_{s \times (i_2-1)+q_2, j'}) = -1 \dots (13)$$

図4 アルゴリズムの概略

$$D'_{s \times (i_3-1)+q_3, j'} = 1 \dots\dots\dots (14)$$

ただし、他の要素は0とする。

[2] プレースの集合 P に含まれるすべてのプレースについて以下の操作を行う。

(1) $p_i \in P$ を選び、 p_i が遷移則の遷移元のプレースの集合に含まれ、かつ、遷移則の遷移先のプレースの集合に含まれるかどうか調べる。満たさなければ次の遷移則へ。

(2) 対応する遷移則を $\delta_k : [X_s, u_{q_1}] \rightarrow [X_s, u_{q_2}]$ とすると(ただし、 X_s は詳細化情報に含まれる任意のプレース集合)、入力プレースが $[p_i, u_{q_1}]$ 、出力プレースが $[p_i, u_{q_2}]$ であるトランジション t'_j を T' に追加する。そして、状態遷移行列 D' の第 j' 列を以下のように規定する。

$$D'_{s \times (i-1)+q_1, j'} = -1 \dots\dots\dots (15)$$

$$D'_{s \times (i-1)+q_2, j'} = 1 \dots\dots\dots (16)$$

ただし、他の要素は0とする。

IV. 孤立プレースを削除する。

状態遷移行列 D' のある行のすべての要素が0ならば、状態遷移行列 D' からその行を消去し、プレース集合 P' から対応するプレースを削除する。

4.2 性質の保存 段階的詳細化において問題になるのは、ネットの持つ諸性質が保存されるかどうかである。本節では、4.1 節で示したアルゴリズムを用いた場合、3章で示した組立ペトリネットが持つグラフ論的性質が保存されるかどうかについて検討する。

詳細化における性質1～3の保存に関して、詳細化情報に対する拘束条件が問題となるが^{(8) (9) (10) (13)}、これについて以下の定理が成り立つ。

(定理1)

詳細化情報が詳細化条件を満足するならば、性質1～3を保存したままペトリネットモデルを詳細化することができる。

(証明)

● 性質1に関して

詳細化条件(1)より、詳細化後のネットの初期マーキングにおいてトークンのある属性付きプレース(単一部品を表す属性付きプレース)は、いずれかの遷移則の遷移元に必ず含まれる。詳細化条件(2)より、詳細化後のネットの最終マーキングにおいてトークンのある属性付きプレース(製品を表す属性付きプレース)は、いずれかの遷移則の遷移先に必ず含まれる。詳細化条件(3)より、遷移則の遷移先に含まれるすべての属性付きプレースは、他の遷移則の遷移元に含まれる。ただし、それが詳細化後のネットの最終マーキングにおいてトークンのある属性付きプレースである場合を除く。

従って、詳細化条件を満たす詳細化情報を与えた時、性質1は保存される。

図5 性質3を満たさない例

● 性質2に関して

遷移則の定義より、1入力1出力または2入力1出力のトランジションのみ生成することは明らかである。

● 性質3に関して

詳細化前のペトリネットが性質3を満たしていて、詳細化後のペトリネットが性質3を満たしていないと仮定し、矛盾を導く。

このとき、詳細化後のネットは図5のようなサブネットを含んでいる。入力プレースが $[p_{i2}, u_{i2}]$ 、 $[p_{i3}, u_{i3}]$ 、出力プレースが $[p_i, u_i]$ であるトランジション t'_j の異なる2つの入力アークがプレース $[p_i, u_i]$ からのパスをそれぞれ持っている。

従って、詳細化前のネットでは、トランジション t'_j の代わりに、入力プレースが p_{i2} 、 p_{i3} 、出力プレースが p_i であるトランジション t_j の異なる2つの入力アークがプレース p_i からのパスをそれぞれ持っていることになる。よって、矛盾。

故に、詳細化において組立ペトリネットの持つ性質1～3が保存されている。 ■

定理1によって、詳細化後の組立ペトリネットがセーフで、目標マーキング以外でデッドロックとならないこと、及び、最適手順の探索が整数解制約のない線形計画法を用いて行えることが明らかになった。なお、詳細化条件を満たさない詳細化情報を与えた場合、詳細化後のネットが性質1を保存しない時がある。例えば、図6において、左の

図6 性質1を満たさない例

ネットに以下のような詳細化条件 (3) を満たさない詳細化情報 R を与えると右のネットのように弱連結性を保存しないネットを生成する。

$$R = \{\delta_k, X_i, u_q; k = 1, i = 1, \dots, 3, q = 1, 2\}$$

$$\delta_1: [X_1, u_2], [X_2, u_1] \rightarrow [X_3, u_1]$$

$$X_1 = \{p_2, p_4\}, X_2 = \{p_3, p_5\}, X_3 = \{p_1, p_2\}$$

$$u_1: \text{属性 } 1, u_2: \text{属性 } 2$$

しかし、このような詳細化条件を満たさない詳細化情報は、あくまで記号論上でのみ構成することが可能で、工学的に意味を持つものではなく、実際の自動組立システムで実行可能なタスクを規定する詳細化情報は、必ず詳細化条件を満たしている。

5. 具体例

半製品の場所に関する以下の詳細化情報 R_1 を与え、本手法を適用することによって、図 2 の組立ペトリネットを図 7 の組立ペトリネットに詳細化する例を示す。

詳細化情報 R_1 : 半製品の場所

$$R_1 = \{\delta_k, X_i, u_q; k = 1, \dots, 5, i = 1, \dots, 5,$$

$$q = 1, \dots, 4\}$$

$$\delta_1: [X_1, u_1] \rightarrow [X_1, u_2]$$

$$\delta_2: [X_2, u_2] \rightarrow [X_2, u_3]$$

$$\delta_3: [X_2, u_2], [X_2, u_3] \rightarrow [X_3, u_2]$$

$$\delta_4: [X_4, u_2], [X_1, u_2] \rightarrow [X_3, u_2]$$

$$\delta_5: [X_5, u_2] \rightarrow [X_5, u_4]$$

$X_1 = \{p_9, p_{10}, p_{11}, p_{12}\}$: 単一部品を表すプレースの集合,

$X_2 = \{p_5, p_6, p_7, p_8\}$: 退避所に運搬されうる半製品を表すプレースの集合,

$X_3 = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$: 組立後の半製品を表すプレースの集合,

$X_4 = \{p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}\}$: 組立前の半製品を表すプレースの集合,

$X_5 = \{p_1\}$: 製品を表すプレースの集合,

u_1 : 部品置き場 (Parts)、 u_2 : 作業所 (Work)、 u_3 : 退避所 (Stack)、 u_4 : 製品置き場 (Product)

詳細化情報 R_1 の解釈は以下の通りである。

δ_1 は、単一部品を部品置き場から作業所へ運搬する。 δ_2 は、半製品を作業所から退避所へ運搬する。 δ_3 は、退避所にある半製品と作業所にある半製品を組み立てる。 δ_4 は、作業所で半製品と単一部品を組み立てる。 δ_5 は、製品を作業所から製品置き場へ運搬する。

図 7 の各プレースが表す状態を以下に示す。 $p_1 = \{\text{CSRH, Product}\}$, $p_2 = \{\text{CSRH, Work}\}$, $p_3 = \{\text{CSR, Work}\}$, $p_4 = \{\text{SRH, Work}\}$, $p_5 = \{\text{CS, Stack}\}$, $p_6 = \{\text{CR, Stack}\}$, $p_7 = \{\text{RH, Stack}\}$, $p_8 = \{\text{SH, Stack}\}$, $p_9 = \{\text{SR, Stack}\}$, $p_{10} =$

$\{\text{CS, Work}\}$, $p_{11} = \{\text{CR, Work}\}$, $p_{12} = \{\text{RH, Work}\}$, $p_{13} = \{\text{SH, Work}\}$, $p_{14} = \{\text{C, Work}\}$, $p_{15} = \{\text{S, Work}\}$, $p_{16} = \{\text{R, Work}\}$, $p_{17} = \{\text{H, Work}\}$, $p_{18} = \{\text{C, Parts}\}$, $p_{19} = \{\text{S, Parts}\}$, $p_{20} = \{\text{R, Parts}\}$, $p_{21} = \{\text{H, Parts}\}$

例えば、図 2 のトランジション t_2 に関して、入力プレースが p_5, p_7 、出力プレースが p_1 であるので、これには遷移則 δ_3 が対応する。従って、 $[p_5, u_2], [p_7, u_3] \rightarrow [p_1, u_2]$, $[p_7, u_2], [p_5, u_3] \rightarrow [p_1, u_2]$ のような状態遷移を示す 2 つのトランジションと状態遷移行列の対応する列ベクトルが生成される。

また、図 2 のプレース p_9 に関して、入力及び出力プレースがともに p_9 であるトランジションを想定すると、遷移則 δ_1 が対応する。従って、 $[p_9, u_1] \rightarrow [p_9, u_2]$ のような状態遷移を示すトランジションと状態遷移行列の対応する列ベクトルが生成される。

同様に、詳細化された図 7 の組立ペトリネットに、さらに半製品の固定部品に関する以下の詳細化情報 R_2 を与え、本手法を適用することによって、図 8 の組立ペトリネットが生成される。

詳細化情報 R_2 : 半製品の固定部品

$$R_2 = \{\delta_k, X_i, u_q; k = 1, \dots, 21, i = 1, \dots, 9,$$

$$q = 1, \dots, 5\}$$

$$\delta_1: [X_1, u_5] \rightarrow [X_2, u_5], \delta_2: [X_1, u_5] \rightarrow [X_3, u_1]$$

$$\delta_3: [X_1, u_5] \rightarrow [X_4, u_2], \delta_4: [X_1, u_5] \rightarrow [X_5, u_3]$$

$$\delta_5: [X_1, u_5] \rightarrow [X_6, u_4]$$

$$\delta_6: [X_3, u_1] \rightarrow [X_7, u_5], \delta_7: [X_4, u_2] \rightarrow [X_7, u_5],$$

$$\delta_8: [X_5, u_3] \rightarrow [X_7, u_5], \delta_9: [X_6, u_4] \rightarrow [X_7, u_5]$$

$$\delta_{10}: [X_7, u_5], [X_3, u_1] \rightarrow [X_8, u_1]$$

$$\delta_{11}: [X_7, u_5], [X_4, u_2] \rightarrow [X_8, u_2]$$

$$\delta_{12}: [X_7, u_5], [X_5, u_3] \rightarrow [X_8, u_3]$$

図 7 ペトリネット 2

$$\delta_{13} : [X_7, u_5], [X_6, u_4] \rightarrow [X_8, u_4]$$

$$\delta_{14} : [X_2, u_5], [X_3, u_1] \rightarrow [X_8, u_1]$$

$$\delta_{15} : [X_2, u_5], [X_4, u_2] \rightarrow [X_8, u_2]$$

$$\delta_{16} : [X_2, u_5], [X_5, u_3] \rightarrow [X_8, u_3]$$

$$\delta_{17} : [X_2, u_5], [X_6, u_4] \rightarrow [X_8, u_4]$$

$$\delta_{18} : [X_8, u_1] \rightarrow [X_9, u_5], \delta_{19} : [X_8, u_2] \rightarrow [X_9, u_5]$$

$$\delta_{20} : [X_8, u_3] \rightarrow [X_9, u_5], \delta_{21} : [X_8, u_4] \rightarrow [X_9, u_5]$$

$X_1 = \{p_{18}, p_{19}, p_{20}, p_{21}\}$: 部品置き場にある単一部品を表すプレースの集合,

$X_2 = \{p_{14}, p_{15}, p_{16}, p_{17}\}$: 作業所にある単一部品を表すプレースの集合,

$X_3 = \{p_3, p_{10}, p_{11}, p_{14}\}$: 作業所にある部品 C を含む半製品及び部品自身を表すプレースの集合,

$X_4 = \{p_3, p_4, p_9, p_{10}, p_{13}, p_{15}\}$: 作業所にある部品 S を含む半製品及び部品自身を表すプレースの集合,

$X_5 = \{p_3, p_4, p_9, p_{11}, p_{12}, p_{16}\}$: 作業所にある部品 R を含む半製品及び部品自身を表すプレースの集合,

$X_6 = \{p_4, p_{12}, p_{13}, p_{17}\}$: 作業所にある部品 H を含む半製品及び部品自身を表すプレースの集合,

$X_7 = \{p_5, p_6, p_7, p_8\}$: 退避所にある半製品を表すプレースの集合,

$X_8 = \{p_2, p_3, p_4, p_9, p_{10}, p_{11}, p_{12}, p_{13}\}$: 作業所にある製品及び半製品を表すプレースの集合,

$X_9 = \{p_1\}$: 製品置き場にある製品を表すプレースの集合,

u_1 : 部品 C を固定 (C)、 u_2 : 部品 S を固定 (S)、 u_3 : 部品 R を固定 (R)、 u_4 : 部品 H を固定 (H)、 u_5 : 固

定されていない (*)

詳細化情報 R_2 の解釈は以下の通りである。

$\delta_1, \dots, \delta_5$ は、単一部品を部品置き場から作業所へ運搬し、ある部品を固定する。 $\delta_6, \dots, \delta_9$ は、半製品を作業所から退避所へ運搬する。 $\delta_{10}, \dots, \delta_{13}$ は、作業所にある固定された半製品と退避所にある半製品を組み立てる。 $\delta_{14}, \dots, \delta_{17}$ は、作業所にある固定された半製品と作業所にある単一部品を組み立てる。 $\delta_{18}, \dots, \delta_{21}$ は、製品を作業所から製品置き場へ運搬する。

図 8 の各プレースが表す状態を以下に示す。 $p_1 = \{\text{CSRH, Product, *}\}$, $p_2 = \{\text{CSRH, Work, C}\}$, $p_3 = \{\text{CSRH, Work, S}\}$, $p_4 = \{\text{CSRH, Work, R}\}$, $p_5 = \{\text{CSRH, Work, H}\}$, $p_6 = \{\text{CSR, Work, C}\}$, $p_7 = \{\text{CSR, Work, S}\}$, $p_8 = \{\text{CSR, Work, R}\}$, $p_9 = \{\text{SRH, Work, S}\}$, $p_{10} = \{\text{SRH, Work, R}\}$, $p_{11} = \{\text{SRH, Work, H}\}$, $p_{12} = \{\text{CS, Stack, *}\}$, $p_{13} = \{\text{CR, Stack, *}\}$, $p_{14} = \{\text{RH, Stack, *}\}$, $p_{15} = \{\text{SH, Stack, *}\}$, $p_{16} = \{\text{SR, Work, S}\}$, $p_{17} = \{\text{SR, Work, R}\}$, $p_{18} = \{\text{CS, Work, C}\}$, $p_{19} = \{\text{CS, Work, S}\}$, $p_{20} = \{\text{CR, Work, C}\}$, $p_{21} = \{\text{CR, Work, R}\}$, $p_{22} = \{\text{RH, Work, R}\}$, $p_{23} = \{\text{RH, Work, H}\}$, $p_{24} = \{\text{SH, Work, S}\}$, $p_{25} = \{\text{SH, Work, H}\}$, $p_{26} = \{\text{C, Work, C}\}$, $p_{27} = \{\text{S, Work, S}\}$, $p_{28} = \{\text{R, Work, R}\}$, $p_{29} = \{\text{H, Work, H}\}$, $p_{30} = \{\text{C, Work, *}\}$, $p_{31} = \{\text{S, Work, *}\}$, $p_{32} = \{\text{R, Work, *}\}$, $p_{33} = \{\text{H, Work, *}\}$, $p_{34} = \{\text{C, Parts, *}\}$, $p_{35} = \{\text{S, Parts, *}\}$, $p_{36} = \{\text{R, Parts, *}\}$, $p_{37} = \{\text{H, Parts, *}\}$

6. 結 論

自動組立手順生成のためのペトリネットの段階的詳細化手法を提案した。詳細化するべき情報が工学的な意味を持ったプレースの属性として定義できる場合には、従来より提案されてきたトランジションベースの詳細化手法に対して、プレースベースの詳細化手法を適用することが望ましいと考えられる。

提案した手法においてある共通の意味を持つプレースをグループ化し、そのグループ間の状態遷移に関する情報のみを与えることによって、ペトリネットモデルを容易に詳細化することができる。

また、本論文で示した詳細化手法は組立ペトリネットが有する様々なグラフ論的性質を保存するため、詳細化後の組立ペトリネットにおける最適な作業手順の探索にそのまま線形計画法を適用できるという利点がある。

文 献

- (1) J.L.Peterson(著), 市川, 小林(訳), ペトリネット入門～情報システムのモデル化, (1984), 共立出版(株)
- (2) W.Reisig(著), 長谷川, 高橋(訳), ペトリネット理論入門～並列同時進行の表現と解析, (1988), シュプリンガー・フェ

図 8 ペトリネット 3

アラーク東京(株)

- (3) L.S.Homen de Mello, A.C.Sanderson, AND/OR Graph Representation of Assembly Plans, IEEE Trans. Robo. Autom., 6-2 (1990), 188-199
- (4) L.S.Homen de Mello, A.C.Sanderson, Representations of Mechanical Assembly Sequences, IEEE Trans. Robo. Autom., 7-2 (1991), 211-227
- (5) L.S.Homen de Mello, A.C.Sanderson, A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences, IEEE Trans. Robo. Autom., 7-2 (1991), 228-240
- (6) 包原, 稲葉, 川瀬, 鈴木, 大熊, ペトリネットを用いた組立作業計画-線形計画法による探索-, 計測自動制御学会論文集, 29-4 (1993), 461-469
- (7) 稲葉, 包原, 鈴木, 大熊, 可操作性を用いた組立作業計画の評価, 日本ロボット学会学術講演会予稿集, 10-2 (1992), 737-740
- (8) I.Suzuki, T.Murata, A method for stepwise refinement and abstraction of Petri nets, J. Comput. Syst. Sci., 27-1 (1983), 51-76
- (9) 中村, 辻, 熊谷, 児玉, ペトリネットによる階層表現におけるライブ性の保存性について, 電子情報通信学会論文誌 A, J71-A-4 (1988), 989-998
- (10) 潮, 松本, 芳野, ペトリネットの逐次詳細化法における有界性とライブ性の保存のための十分条件, 電子情報通信学会論文誌 A, J71-A-8 (1988), 1563-1568
- (11) 相良, 数理計画入門, (1976), 森北出版(株)
- (12) T.Cao, A.C.Sanderson, Task Sequence Planning in a Robot Workcell using AND/OR Nets, IEEE Proc. Inter. Sympo. Intelligent Control, (1991-8), 239-244
- (13) 胡, 兼重, 小野木, 西村, 離散事象システムの段階的詳細化について, 計測自動制御学会論文集, 28-6 (1992), 733-741
- (14) K.Jensen, Colored Petri Nets, Vol.1, (1992), Springer Verlag
- (15) 林, 市川, 色付きペトリネットとそれによる論理回路・シーケンス制御系の記述, 計測自動制御学会論文集, 23-10 (1987), 66-73