

## 可重构制造系统监督控制器的自动重构

李俊<sup>1</sup> 戴先中<sup>1</sup> 孟正大<sup>1</sup>

**摘要** 提出了基于改进的网重写系统 (Improved net rewriting system, INRS) 的可重构制造系统 (Reconfigurable manufacturing systems, RMS) Petri 网监督控制器的自动重构方法, 以快速适应由市场需求变化所引起的制造系统构形的频繁变化. INRS 解决了网重写系统存在的问题, 可动态调整给定 Petri 网模型的结构而不改变其行为属性. 以集合和图的组合形式定义了 RMS 的构形, 并提出了基于 INRS 的一类模块化、可重构的 Petri 网控制器的设计方法. 针对这类 Petri 网控制器, 提出了基于 INRS 的自动重构方法. 方法可将 RMS 构形的变化转变为 INRS 的图重写规则, 并作用于当前 Petri 网控制器, 使其快速、自动地重构为所求的新控制器. 所提出的 Petri 网控制器的设计与重构方法, 均从理论上保证了结果的正确性, 免校验. 仿真研究验证了方法的有效性.

**关键词** 可重构制造系统, Petri 网, 监督控制器, 重构, 设计  
**中图分类号** TP273

### Automatic Reconfiguration of Supervisory Controllers for Reconfigurable Manufacturing Systems

LI Jun<sup>1</sup> DAI Xian-Zhong<sup>1</sup> MENG Zheng-Da<sup>1</sup>

**Abstract** This paper presents an improved net rewriting system (INRS) – based method for automatic reconfiguration of Petri net (PN) supervisory controllers for reconfigurable manufacturing systems (RMS) to adapt rapidly to frequent changes in the manufacturing system configurations driven by market changes. INRS addresses the problems of net rewriting systems and can dynamically change the structure of a PN model with preserving its original behavioral properties. An RMS configuration is formally defined by incorporating the formalisms of set and graph. A design method for a class of modular and reconfigurable PN controllers is presented with an approach of INRS. Subsequently, an INRS-based method for automatic reconfiguration of this class of PN controllers is proposed. Using the reconfiguration method, changes in an RMS configuration can be automatically transformed into graph rewriting rules of INRS; by applying the rules to the existing controller, it will be reconfigured into the needed one. The resultant PN controllers obtained by both the design and reconfiguration methods are guaranteed to be valid in theory, without the need of verification. Simulation results show the validity of the presented methods.

**Key words** Reconfigurable manufacturing systems (RMS), Petri nets (PN), supervisory controller, reconfiguration, design

离散事件系统 (Discrete event systems, DES) 的监督控制是自动控制领域的一个重要的分支<sup>[1]</sup>. 制造系统作为典型的 DES, 其监督控制器的重构性已成为研究热点<sup>[2-4]</sup>. 尤其是随着可重构制造系统 (Reconfigurable manufacturing systems, RMS)<sup>[5]</sup> 的提出, 企业急切地要求制造系统的监督控制器能通过重构来快速响应由市场需求变化引起的制造系统构形的频繁变化.

部分学者提出了可重构的监督控制系统的设计框架、体系<sup>[6-8]</sup>, 以确保监督控制器的模块化, 但未给出控制器重构的具体方法. 另一部分工作则侧

重于研究监督控制器的具体重构机制与方法<sup>[9-11]</sup>, 但不能保证结果控制器的正确性和有效性. 针对这一问题, 本文提出 RMS 的一类模块化的 Petri 网形式的监督控制器的重构方法, 旨在不损害现有 Petri 网控制器控制属性的前提下, 实现该控制器的快速、自动重构, 以响应由产品需求变化所引起的制造系统构形的变化.

#### 1 问题陈述与解决思路

RMS 借助于构形的变化, 能方便、快速地转为同一产品族内另一产品的生产<sup>[5]</sup>. 若从监督控制角度考虑, RMS 的构形可定义如下.

**定义 1.** 可重构制造系统  $S$  的构形定义为三元式  $C = (W, O, Z)$ , 其中,

- 1)  $W = \{w_1, w_2, \dots, w_m\}$  为  $S$  的有限机器集;
- 2)  $O = \{o_1, o_2, \dots, o_n\}$  为机器的有限操作序列集.  $\gamma: O \rightarrow W$  将每个操作序列映射到一个机器;

收稿日期 2007-06-06 收修改稿日期 2008-01-12  
Received June 6, 2007; in revised form January 12, 2008  
国家重点基础研究发展计划 (973 计划) (2002CB312204) 资助  
Supported by National Basic Research Program of China (973 Program) (2002CB312204)  
1. 东南大学自动化学院 南京 210096  
1. School of Automation, Southeast University, Nanjing 210096  
DOI: 10.3724/SP.J.1004.2008.01337

3)  $Z = (V, E)$  为  $S$  的构形图 (Configuration graph), 其中  $V = \{v_1, v_2, \dots, v_m\}$  为  $Z$  的有限顶点集.  $\eta : V \rightarrow W$  将每个顶点映射到一个机器,  $E = \{e_1, e_2, \dots, e_k\}$  为  $Z$  的有限边集, 每条边反映两个机器之间的逻辑关系.

例如, 图 1 为一可重构加工单元, 具体说明见文献 [12]. 该加工单元的当前构形图  $Z1$ , 如图 2(a)所示, 可批量生产  $A$  零件, 其中  $w_1 \sim w_4$  指代 4 个工作站  $W1 \sim W4$ ,  $a_{11} \sim a_{15}$  指代  $AGV1$  的搬运任务,  $l$  指代零件装载/卸载站.

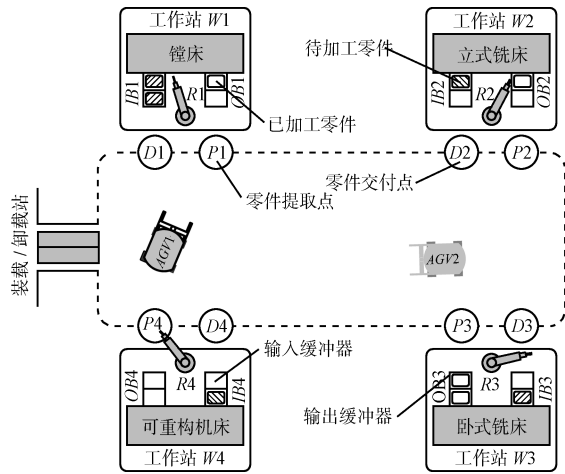
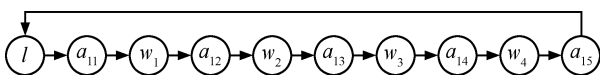


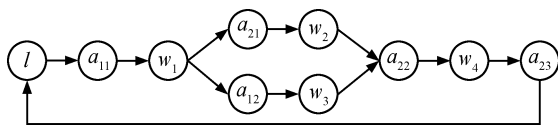
图 1 可重构加工单元布局

Fig. 1 Layout of a reconfigurable manufacturing cell



(a) 构形图  $Z1$

(a) Configuration graph  $Z1$



(b) 构形图  $Z2$

(b) Configuration graph  $Z2$

图 2 可重构加工单元构形

Fig. 2 Two configuration graphs of the cell

假定客户需要生产另一批量的  $B$  零件, 相应的构形图如图 2(b) 所示, 其中零件先由  $AGV1$  运到  $W1$  加工后, 可选择由  $AGV2$  运往  $W2$  或由  $AGV1$  运往  $W3$  加工. 之后, 由  $AGV2$  运往  $W4$  加工, 并输出.  $a_{2i} (i = 1, 2, 3)$  代表  $AGV2$  的 3 个搬运任务. 这就需要加工单元的重构. 随着重构的发生, 现有的 Petri 网控制器便不再满足新生产过程的监督控制

需要. 尽管可以重新设计控制器, 但这一思路对于对响应性、自适应性要求较高的 RMS 来说, 代价过于昂贵.

对此, 本文的思路是: 为 Petri 网控制器设计一种自动重构的机制, 以鉴别制造系统的当前构形与目标构形之间的差异, 并将这种差异作用于当前构形所对应的 Petri 网控制器上, 使其快速地重构为目标构形所对应的新 Petri 网控制器, 同时保留原控制器的行为属性 (如活性、有界性和可逆性). 相对于重设计, 重构突出了控制器的自适应性和响应性. 按这一思路, 可引入网重写系统 (Net rewriting system, NRS)<sup>[13]</sup>, 作为处理 Petri 网控制器模型动态重构的工具. 但进一步研究发现, NRS 不能保留被处理的 Petri 网模型的原有活性、有界 (或安全性) 和可逆性的行为属性. 这些行为属性反映了被建模系统的无死锁、稳定以及行为循环的特性, 对于正确地操作和控制被建模系统至关重要. 经分析, 发现可对 NRS 进行改进, 以解决上述问题.

本文首先提出了改进的网重写系统 (Improved net rewriting system, INRS). 此外, 考虑到 Petri 网控制器的结构也决定着它能否重构, 提出了基于 INRS 的一类模块化的、可重构的 Petri 网控制器的设计方法. 此后, 同样基于 INRS, 着重研究了这类 Petri 网控制器的自动重构方法.

## 2 改进的网重写系统

网重写系统可以动态地变化 Petri 网模型的结构, 但同时会损害模型的行为属性<sup>[12]</sup>, 对此, 本节提出改进的网重写系统.

### 2.1 改进的网重写系统的定义

定义 2. 改进的网重写系统 (INRS) 定义为三元式  $\mathcal{N} = (\mathcal{L}, \mathcal{R}, G)$ , 其中,

1)  $\mathcal{L} = (\mathcal{L}_t, \mathcal{L}_p)$  为通用的网模块类型库.  $\mathcal{L}_t$  为变迁型接口的网模块类型的集合,  $\mathcal{L}_p$  为库所型接口的网模块类型的集合. 所有网模块类型均在第 2.2 节介绍.

2)  $G = (N, M_0)$  为  $\mathcal{N}$  的初始基础 Petri 网, 其中  $N = (P, T, F)$  为  $G$  的结构,  $M_0$  为  $G$  的初始标识.

3)  $\mathcal{R} = \{r_1, r_2, \dots, r_k\}$  为有限的重写规则集. 任意重写规则  $r$  表示为  $r = (L, R, f, \bullet\tau, \tau\bullet)$ .  $r$  的左、右手侧  $L, R$  均属于  $\mathcal{L}$  中的类型.  $f : L \rightarrow \bar{G}$  为全嵌入映射, 将  $L$  映射为  $G$  中的全子网  $\bar{G}$ .  $r$  的传递关系  $\tau = \bullet\tau \cup \tau\bullet$ , 其中输入、输出接口关系分别为  $\bullet\tau = \{(A, B)\}$ 、 $\tau\bullet = \{(C, D)\}$ , 其中  $A = \{x_{in}^1, x_{in}^2, \dots, x_{in}^n\}$  和  $B = \{y_{in}^1, y_{in}^2, \dots, y_{in}^m\}$  分别为  $L$  和  $R$  的输入节点集,  $C = \{x_{out}^1, x_{out}^2, \dots, x_{out}^n\}$  和  $D =$

$\{y_{out}^1, y_{out}^2, \dots, y_{out}^{m'}\}$  分别为  $L$  和  $R$  的输出节点集. 所有的节点同为变迁或库所, 其中  $n, m, n', m'$  为自然数.

4) 假设对  $G$  应用  $r$  得到  $G'$ , 则  $G'$  的流关系  $F'$  由下式给出

$$F'(x, y) = \begin{cases} F(x, y), & x, y \notin R \\ F_R(x, y), & x, y \in R \\ \frac{\sum_{y_i \in \bullet \tau y} F(x, f(y_i))}{|\bullet \tau y|}, & x \notin R, y \in R \\ \frac{\sum_{x_i \in \tau \bullet x} F(f(x_i), y)}{|\tau \bullet x|}, & x \in R, y \notin R \end{cases} \quad (1)$$

而库所  $p \in P'$  的标识  $M'(p)$  由下式给出

$$M'(p) = \begin{cases} M(p), & p \notin R \\ M_R(p), & p \in R \end{cases} \quad (2)$$

**注 1.** 对重写规则  $r \in \mathcal{R}$ , 取其左手侧的相异输入节点对  $\langle x_{in}^i, x_{in}^j \rangle$  和相异输出节点对  $\langle x_{out}^i, x_{out}^j \rangle$ , 则节点对  $\langle f(x_{in}^i), f(x_{in}^j) \rangle$  以及  $\langle f(x_{out}^i), f(x_{out}^j) \rangle$  分别具有共同的前集和后集.

**注 2.** 基础 Petri 网变迁的触发和网重写过程禁止同时发生.  $\forall r \in \mathcal{R}, p \in (G - f(L))$ , 若  $M_0(p) = 0$ , 则禁止  $r$  的应用.

**注 3.** INRS 的网重写过程具有 3 个重要属性, 即局部性、传递性和可逆性, 这些属性对于维持被重写 Petri 网的重要行为属性起着重要的作用, 详细介绍见文献 [12].

### 2.2 通用的网模块类型

库所可分为两类, 以区分操作与资源. 给定标记 Petri 网  $G = (P, T, F, M_0)$ , 对库所  $p \in P$ , 若  $M_0(p) = 0$ , 则  $p$  称为操作库所; 否则,  $p$  称为资源库所, 指代物理资源. 对应地, 库所集  $P$  可划分为资源库所子集  $P^r$  和操作库所子集  $P^o$ . 以下网模块均为平凡 Petri 网.

**定义 3.** 给定 Petri 网  $G = (P^o, T, F, M_0)$ , 若  $P^o = \{p\}$ ,  $T = \emptyset$  且  $F = \emptyset$ , 则称  $G$  为单库所 (Single place, SP) 网, 记为  $G = (p)$ .  $G$  的输入、输出节点同为  $p$ ; 若  $T = \{t\}$ ,  $P^o = \emptyset$ , 且  $F = \emptyset$ , 则称  $G$  为单变迁 (Single transition, ST) 网, 记为  $G = (t)$ , 其输入和输出节点  $t_{in}, t_{out}$  同为  $t$ .

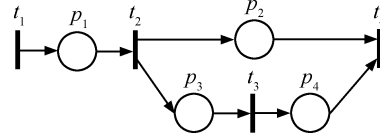
**定义 4.** 给定 Petri 网  $G = (P^o, T, F, M_0)$ , 其中  $|P^o| \geq 1, |T| \geq 1$ , 且  $G$  中无有向回路.

1)  $G$  称为开式标记图 (Open marked graph, OMG), 若  $\forall p \in P^o$ , 有  $|\bullet p| = |p \bullet| = 1$ , 且  $\exists t_i, t_j \in T$  ( $i \neq j$ ), 使得  $\bullet t_i = t_j \bullet = \emptyset$ .  $G$  的输入节点  $t_{in}$  和

输出节点  $t_{out}$  分别为  $t_i$  和  $t_j$ .

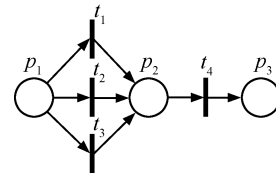
2)  $G$  称为开式状态机 (Open state machine, OSM), 若  $\forall t \in T$ , 有  $|\bullet t| = |t \bullet| = 1$ , 且  $\exists p_i, p_j \in P^o$  ( $i \neq j$ ), 使得  $\bullet p_i = p_j \bullet = \emptyset$ .  $G$  的输入节点  $p_{in}$  和输出节点  $p_{out}$  分别为  $p_i$  和  $p_j$ .

OMG 或 OSM 均可描述动作或者行为序列, 但 OMG 表示并发而非决策 (冲突), OSM 允许表示决策而非并行行为的同步. OMG 和 OSM 的示例分别如图 3 (a) 和 3 (b) 所示.



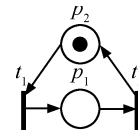
(a) OMG 示例

(a) An example of OMG



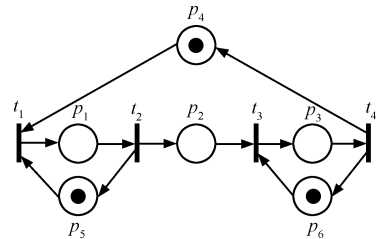
(b) OSM 示例

(b) An example of OSM



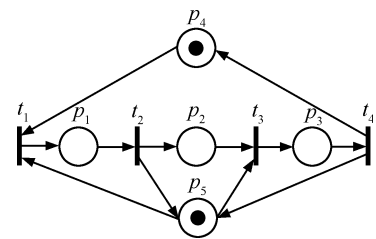
(c) CMG 示例

(c) An example of CMG



(d) SCMG 示例 1

(d) An example of SCMG



(e) SCMG 示例 2

(e) Another example of SCMG

图 3 定义 4~6 的示例

Fig. 3 Examples for Definitions 4~6

**定义 5.** 给定 Petri 网  $G = (P, T, F, M_0)$ , 其中  $P = P^r \cup P^o$ ,  $P^r = \{\tilde{p}\}$  且  $|P^o| \geq 1$ ,  $G$  称为闭式标记图 (Closed marked graph, CMG), 若同时满足以下条件:

- 1)  $\exists t_i, t_j \in T (i \neq j)$ , 使得  $\bullet t_i = \{\tilde{p}\}$  且  $t_j \bullet = \{\tilde{p}\}$ ;
- 2)  $\tilde{p} \bullet = \{t_i\}$  且  $\bullet \tilde{p} = \{t_j\}$ ;
- 3)  $G$  中存在 OMG 类子网  $G' = (P^o, T, F', M'_0)$ , 其中  $F' = F - F(\tilde{p}, t_i) - F(t_j, \tilde{p})$  且  $\forall p \in P^o, M'_0(p) = M_0(p)$ .  $G$  的输入和输出节点记为  $t_{in} = t_i$  和  $t_{out} = t_j$ .

实际上, CMG 是由 OMG 和资源库所组成, 表示一旦所需的资源可用, 则一个过程或者行为序列便开始执行. 图 3(c) 为一个典型的 CMG 类模型, 可视为设备的简单功能描述.

**定义 6.** 给定 Petri 网  $G = (P, T, F, M_0)$ , 其中  $P = P^r \cup P^o$ ,  $|P^o| \geq 1$ , 以及  $|T| \geq 1$ ,  $G$  称为顺序闭式标记图 (Sequential closed marked graph, SCMGM), 若同时满足以下条件:

- 1)  $G$  中存在 OMG 类子网  $G' = (P^o, T, F', M'_0)$ , 其中  $\forall p \in P^o, M'_0(p) = M_0(p)$  且  $F' = F - \bigcup_{p \in P^r} (\bigcup_{t \in p \bullet} F(p, t) + \bigcup_{t \in \bullet p} F(t, p))$ . 令  $G'$  的输入和输出节点分别为  $\dot{t}$  和  $\dot{t}$ .
- 2)  $\forall p \in P^r, p$  与子网  $G'$  组成  $m (\geq 1)$  个 CMG, 其中  $M_0(p) = k (\geq 1)$ ,  $p$  的输入和输出弧沿  $G'$  中令牌流动的方向交替出现, 且  $|p \bullet| = |p \bullet|$ .
- 3)  $G$  中无自回路 (Self-loop).  $G$  的输入和输出节点同  $G'$ , 即  $t_{in} = \dot{t}$  和  $t_{out} = \dot{t}$ .

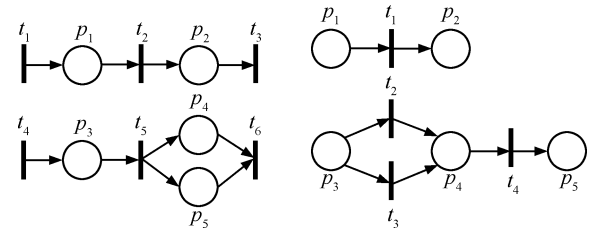
SCMGM 类模块为 CMG 的扩展, 可对设备或过程建模. 图 3(d) 和 3(e) 为两个典型的 SCMGM 的示例.

**定义 7.** 设 Petri 网  $G = (P, T, F, M_0)$  由  $k (\geq 2)$  个独立的子网组成, 即  $G = \bigcup_1^k G^i$  且  $G^i \cap G^j = \emptyset$ , 其中  $i \neq j$ . 对于任意子网  $G^i$ ,

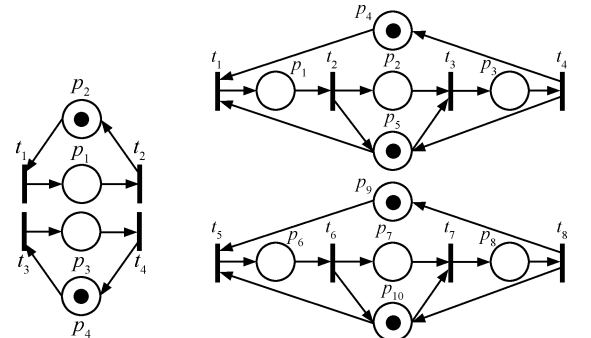
- 1) 若为 OMG, 则  $G$  称为  $k$ -平行开式标记图, 记为 OMG( $k$ );
- 2) 若为 OSM, 则  $G$  称为  $k$ -平行开式状态机, 记为 OSM( $k$ );
- 3) 若为 CMG, 则  $G$  称为  $k$ -平行闭式标记图, 记为 CMG( $k$ );
- 4) 若为 SCMGM, 则  $G$  称为  $k$ -平行串行闭式标记图, 记为 SCMGM( $k$ ).

对 1)、3)、4), 设  $G^i$  的输入和输出节点对为  $\langle t_{in}^i, t_{out}^i \rangle$ , 则  $G$  的输入和输出节点集分别为  $T_{in} = \{t_{in}^1, t_{in}^2, \dots, t_{in}^k\}$  和  $T_{out} = \{t_{out}^1, t_{out}^2, \dots, t_{out}^k\}$ ; 对 2), 设  $G^i$  的输入和输出节点对为  $\langle p_{in}^i, p_{out}^i \rangle$ , 则  $G$  的输入和输出节点集分别为  $P_{in} = \{p_{in}^1, p_{in}^2, \dots, p_{in}^k\}$  和  $P_{out} = \{p_{out}^1, p_{out}^2, \dots, p_{out}^k\}$ .

图 4 (a) ~ (d) 依次为 OMG (2), OSM (2), CMG (2) 和 SCMGM (2) 的示例. OMG ( $k$ ) 或 OSM ( $k$ ) 可表示  $k$  个相互独立的过程或者行为序列, 而 CMG ( $k$ ) 或 SCMGM ( $k$ ) 则可表示  $k$  个平行工作的独立设备或过程.



(a) OMG (2) 示例 (b) OSM (2) 示例  
(a) An example of OMG (2) (b) An example of OSM (2)



(c) CMG (2) 示例 (d) SCMGM (2) 示例  
(c) An example of CMG (2) (d) An example of SCMGM (2)

图 4 定义 7 的示例  
Fig. 4 Examples for Definition 7

**定义 8.** 给定 Petri 网  $G = (P, T, F, M_0)$ , 其中  $P = P^r \cup P^o$ ,  $T = \bigcup_1^3 T_i$  且  $T_i \cap T_j = \emptyset (i \neq j)$ .  $\forall t \in T_1$ , 有  $\bullet t = \emptyset$  且  $t \bullet = \{\tilde{p}\}$ , 以及  $\forall t \in T_3$ , 有  $t \bullet = \emptyset$  且  $\bullet t = \{\tilde{p}\}$ , 其中  $\tilde{p}, \tilde{p} \in P^o, \tilde{p} \neq \tilde{p}$ . 令  $T_1 = \{t_{in}^1, t_{in}^2, \dots, t_{in}^k\}$  和  $T_3 = \{t_{out}^1, t_{out}^2, \dots, t_{out}^l\}$ . 对于  $t_{in}^i \in T_1$  以及  $t_{out}^j \in T_3$ ,

- 1) 若  $P^r = \emptyset$  且  $G$  中存在着 OMG 子网  $G' = (P^o, T', F', M'_0)$ , 其中  $T' = \{t_{in}^i\} \cup T_2 \cup \{t_{out}^j\}$  且  $F' = F - \bigcup_{t \in T_1, t \neq t_{in}^i} F(t, \tilde{p}) - \bigcup_{t \in T_3, t \neq t_{out}^j} F(\tilde{p}, t)$ , 则  $G$  称为具有  $k$  输入和  $l$  输出的变体开式标记图, 记为 OMG( $k, l$ );
- 2) 若  $P^r = \tilde{p}$  且  $G$  中存在着 CMG 子网  $G' = (P, T', F', M'_0)$ , 其中  $T' = \{t_{in}^i\} \cup T_2 \cup \{t_{out}^j\}$ ,  $F' = F - \bigcup_{t \in T_1, t \neq t_{in}^i} F(t, \tilde{p}) - \bigcup_{t \in \tilde{p} \bullet, t \neq t_{in}^i} F(\tilde{p}, t) - \bigcup_{t \in \bullet \tilde{p}, t \neq t_{out}^j} F(t, \tilde{p})$ , 且  $\forall p \in P, M'_0(p) = M_0(p)$ , 则  $G$  称为具有  $k$  输入和  $l$  输出的变体闭式标记图, 记为 CMG( $k, l$ ).

上述 2 种情况中,  $G$  的输入和输出节点集均分别为  $T_1$  和  $T_3$ .

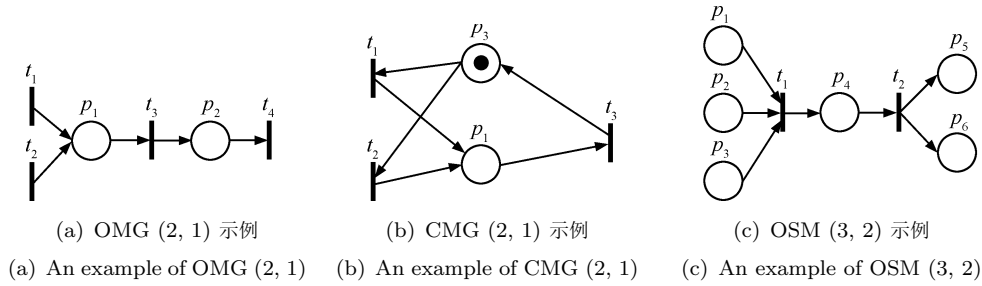


图5 定义8、9的示例

Fig. 5 Examples for Definitions 8 and 9

图5 (a) 和 5 (b) 依次为 OMG (2, 1) 和 CMG (2, 1) 类模型示例. OMG ( $k, l$ ) 或 CMG ( $k, l$ ) 表示先前的动作或者过程有  $k \times l$  种后续的选择. 特别地, CMG ( $k, l$ ) 可建模具有  $k$  个输入和  $l$  个输出的缓冲器.

**定义 9.** 给定 Petri 网  $G = (P, T, F, M_0)$ , 其中  $|T| \geq 2$ ,  $P = \bigcup_1^3 P_i$  且  $P_i \cap P_j = \emptyset$  ( $i \neq j$ ).  $\forall p \in P_1$ , 有  $\bullet p = \emptyset$  且  $p^\bullet = \{\dot{t}\}$ , 以及  $\forall p \in P_3$ , 有  $p^\bullet = \emptyset$  且  $\bullet p = \{\dot{t}\}$ , 其中  $\dot{t}, \ddot{t} \in T, \dot{t} \neq \ddot{t}$ . 令  $P_1 = \{p_{in}^1, p_{in}^2, \dots, p_{in}^k\}$  且  $P_3 = \{p_{out}^1, p_{out}^2, \dots, p_{out}^l\}$ .  $\forall p_{in}^i \in P_1, p_{out}^j \in P_3$ , 若  $G$  的子网  $G' = (P', T, F', M_0)$  为 OSM 类型, 其中  $P' = \{p_{in}^i\} \cup P_2 \cup \{p_{out}^j\}$ ,  $F' = F - \bigcup_{p \in P_1, p \neq p_{in}^i} F(p, \dot{t}) - \bigcup_{p \in P_3, p \neq p_{out}^j} F(\ddot{t}, p)$ , 则  $G$  称为具有  $k$  输入和  $l$  输出的变体开式状态机, 记为 OSM ( $k, l$ ).  $G$  的输入和输出节点集分别为  $P_1$  和  $P_2$ .

图5 (c) 为 OSM (3, 2) 类模型的示例. 通常 OSM ( $k, l$ ) 用于连接其他类型网模块. 此外, OSM ( $k, l$ ) 可用于建模具有  $k$  个组合动作和  $l$  个分解动作的过程, 如制造系统中的装配和拆卸的过程.

### 2.3 INRS 的属性保持理论

不失一般性, 规定下文中涉及的 INRS 仅包含一条重写规则, 且其全嵌入映射存在, 并已确定.

**定理 1.** 给定 INRS  $\mathcal{N} = (\mathcal{L}, \mathcal{R}, G)$ , 其中  $G = (\Gamma, M_0)$  且  $\mathcal{R} = \{r\}$ , 这里  $r = (L, R, f, \bullet\tau, \tau^\bullet)$ , 其左侧  $L$  和右侧  $R$  均为  $\mathcal{L}_p$  中类型的网模块, 若  $G$  为活的、有界 (安全) 和可逆的 (Live, bounded (safe), and reversibility, LBR), 则经重写得到的结果 Petri 网  $G' = (\Gamma', M'_0)$  也为 LBR.

**证明.** 考虑以下两种情况:

1)  $L$  和  $R$  之一为 SP. 证明见文献 [12] 中的定理 3.3.1 和推论 3.3.1.

2)  $L$  和  $R$  均非 SP. 假定在  $G$  与  $G'$  之间存在过渡结果 Petri 网  $\bar{G} = (\bar{\Gamma}, \bar{M}_0)$ . 构建两个 INRS  $\mathcal{N}_1 = (\mathcal{L}, \mathcal{R}_1, G)$  和  $\mathcal{N}_2 = (\mathcal{L}, \mathcal{R}_2, \bar{G})$ , 其中  $\mathcal{R}_1 = \{r_1\}$ ,  $r_1 = (L_1, R_1, f_1, \bullet\tau_1, \tau_1^\bullet)$  以及  $\mathcal{R}_2 = \{r_2\}$ ,  $r_2 = (L_2, R_2, f_2, \bullet\tau_2, \tau_2^\bullet)$ . 令  $L_1 = L, R_1$  为 SP  $\bar{p}, L_2 = R_1,$

$R_2 = R$ , 且  $G - f_1(L_1) = \bar{G} - f_2(L_2)$ . 有  $G \xrightarrow{\mathcal{R}_1} \bar{G}$  且  $\bar{G} \xrightarrow{\mathcal{R}_2} G'$ . 由文献 [12] 的推论 3.3.1 可知,  $\bar{G}$  为 LBR. 进一步, 根据 1) 可知,  $G'$  为 LBR. 考虑到 INRS 的传递性, 则有  $G \xrightarrow{\mathcal{R}_1} \bar{G}, \bar{G} \xrightarrow{\mathcal{R}_2} G'$  等价于  $G \xrightarrow{\mathcal{R}} G'$ . 也就是说, 当  $G$  为 LBR,  $G'$  为 LBR.  $\square$

定理 1 允许使用库所型接口的网模块对 LBR Petri 网中的全子网 (不只是单一的库所) 进行属性保持的重写. 该定理的实例已在文献 [12] 中给出.

**定理 2.** 给定 INRS  $\mathcal{N} = (\mathcal{L}, \mathcal{R}, G)$ , 其中  $G = (\Gamma, M_0), \mathcal{R} = \{r\}$ . 这里  $r = (L, R, f, \bullet\tau, \tau^\bullet)$ ,  $L$  和  $R$  均为  $\mathcal{L}_t$  中类型的网模块, 若  $G$  为 LBR, 则经重写得到的结果 Petri 网仍为 LBR.

定理 2 的证明类似于定理 1 的证明, 见文献 [12] 中的定理 3.3.4. 定理 2 支持用网模块对 LBR Petri 网中的全子网 (不只是变迁) 进行属性保持的重写. 文献 [12] 中给出了该定理的实例.

### 2.4 INRS 与 NRS 的比较

文献 [12] 用实例从多个角度比较了 INRS 和 NRS, 并得出以下结论.

1) INRS 中的图重写规则的左、右侧的类型受限于通用的网模块类型. 实际的系统和过程均可以抽象或具体地由这些类型模块来描述. 此外, 通过组合这些类型的模块可构建更复杂的系统或过程的模型.

2) INRS 仍然保留了 NRS 所具有的动态改变 Petri 网模型结构的能力, 它的特点在于重写过程是局部的、传递的和可逆的, 只要初始的基础 Petri 网为 LBR, 则结果 Petri 网仍将是 LBR.

3) NRS 的子类 — 可重构 Petri 网已出现在文献 [13] 中. 它的结构变化表现为固定库所集上对有向弧的调整, 因此, 它并不适合于对具有可扩展结构的实际系统 (如 RMS) 进行建模, 而 INRS 则胜任.

综上, INRS 优于 NRS 及其子类可重构 Petri 网.

### 3 基于 INRS 的 Petri 网控制器设计

由定义 1 可见, RMS 的构形具有层次化的结构, 层次由高到低依次是构形图、机器集、操作序列集. 机器层通过双射  $\eta^-$  (即  $\eta$  的逆映射) 与顶层联系起来, 如对于机器  $w_1$ 、构形图的顶点  $v_1$ , 存在  $\eta(v_1) = w_1$  及  $\eta^-(w_1) = v_1$ , 而底层与机器层通过单射  $\gamma$  联系起来, 另外, 底层与顶层可通过  $\eta^- \circ \gamma$  (其中  $\circ$  表示两个映射 (函数) 的联合) 联系起来. 据此, 本节提出基于 INRS 的可重构制造系统的一类模块化的 Petri 网控制器的结构化设计方法.

#### 3.1 子模型的构建

按由低到高的次序, 依次对构形的操作序列、机器、构形图, 进行 Petri 网建模.

为了描述构形的各元素与 Petri 网子模型的对应关系, 定义以下 4 种映射.

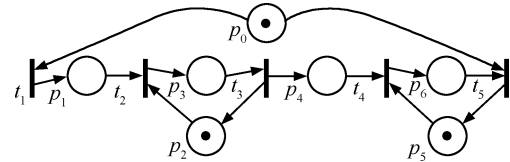
**定义 10.**

- 1) 操作-网映射  $\xi$ : 将操作序列映射到网块;
- 2) 机器-网映射  $\psi$ : 将机器映射到网块;
- 3) 图-网映射  $\zeta$ : 将构形图映射到网模型;
- 4) 对象映射  $\varphi$ : 一对一地映射操作序列或机器与 Petri 网控制器中的对象模块.

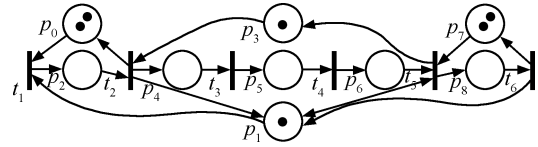
为了正确利用 INRS 技术, 规定构建的操作序列的网模块必须属于子库  $\mathcal{L}_p$  中的类型, 机器的网模块必须属于子库  $\mathcal{L}_t$  中的类型. 以第 1 节介绍的可重构加工单元为例, 可构建各工作站的操作序列所对应的 Petri 网模型. 再如, 对移动小车 1 和工作站 4 进行建模, 结果分别如图 6(a) 和 6(b) 所示. 对于构形图, 其顶点必须建模为子库  $\mathcal{L}_t$  所含类型的网模块, 并且其本身的 Petri 网模型必须为 LBR, 以反映制造系统的无死锁、稳定和循环的生产特点. 对图 2(a) 所示的构形图 Z1 进行建模, 得到如图 6(c) 所示的 Petri 网模型  $G_{z_1}$ , 其为活的、有界和可逆的. Z1 的每个顶点均对应着该 Petri 网中的一个简单 OMG 类模块, 如图 6(d) 所示, 例如 Z1 的顶点  $a_{11}$  对应着  $G_{z_1}$  中的模块  $(t_1, p_1, t_2)$ .

#### 3.2 子模型的集成

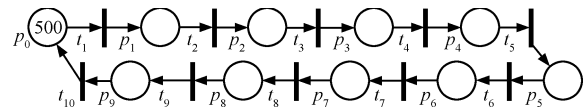
首先, 将各机器的模型集成到构形图的 Petri 网子模型中. 先构建 INRS  $\mathcal{N} = (\mathcal{L}, \mathcal{R}, \zeta(Z))$ , 其中  $\zeta(Z)$  为给定构形图  $Z$  所对应的 Petri 网模型; 而  $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$  为重写规则集.  $\forall r \in \mathcal{R}$ ,  $r = (L, R, f, \bullet\tau, \tau\bullet)$ , 其中  $L$  为  $Z$  中的顶点  $v$  所对应的 OMG 类模块; 假定顶点  $v$  对应机器  $m$ , 即  $\eta(v) = m$ , 而  $R$  为机器  $m$  的 Petri 网类模型  $\psi(m)$ ;



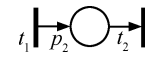
(a) AGV1 的 Petri 网模型  
(a) Petri net for AGV1



(b) W4 的 Petri 网模型  
(b) Petri net for W4



(c) Z1 的 Petri 网模型  
(c) Petri net for Z1



(d) 简单 OMG  
(d) Simple OMG

图 6 设计过程中的几个子模型

Fig. 6 Several subnets involved in the design process

$f$  为全嵌入, 将  $L$  映射为  $\zeta(Z)$  的一个全子网, 即对象模块  $\varphi(m)$ ;  $\bullet\tau$  和  $\tau\bullet$  分别为  $L$  和  $R$  的输入和输出接口关系. 举例来说, 假定图 2(a) 所示的构形图 Z1 中的顶点  $w_4$  对应的 W4 的对象模型为图 6(c) 所示的 Petri 网中的全子网  $(t_8, p_8, t_9)$ , 则令  $L$  为图 6(d) 所示的 OMG 类模块, 而  $R$  为如图 6(b) 所示的 W4 的网类模型  $\psi(W4)$ ,  $f(L) = \varphi(W4)$ . 应用  $r$ ,  $\psi(W4)$  集成到  $\zeta(Z)$  中.

其次, 将各个机器的操作序列的 Petri 网子模型集成到获得的 Petri 网中, 可获得更为细化的 Petri 网控制器模型.

最后, 对该 Petri 网进行控制解释, 即将模型中的库所关联控制动作, 以操控执行机构, 而将变迁关联被控制系统的输出传感信号<sup>[12]</sup>, 便可获得具有实际监督控制意义的 Petri 网监督控制器.

#### 3.3 资源库所融合

有时, 由第 3.2 节所得到的 Petri 网控制器不一定是所求的最终 Petri 网控制器, 因为其中可能存在着多个指代同一物理资源的资源库所, 表示对物理资源的共享. 为一致性考虑, 这些资源库所必须合而为一. 这种合并称为资源库所的融合, 可划分为串行和并行两种模式.

### 3.3.1 资源库所的串行融合

图 3(d) 所示的类模块可描述机床的功能, 其中库所  $p_1$ 、 $p_2$ 、 $p_3$  分别代表装载、加工和卸载, 而  $p_5$ 、 $p_6$  分别指代装载和卸载过程中的搬运工具. 设  $p_5$ 、 $p_6$  指代同一机械臂, 为了达到物理意义的一致性, 必须将它们融合, 可得如图 3(e) 所示的资源库所  $p_5$ , 这一过程称为资源库所的串行融合.

多个机器或过程占有资源的场景通常可以建模为 SCMG 类型模块, 且资源库所串行融合后的场景仍可用 SCMG 来表示. 根据定理 2, 可断言资源库所的串行融合, 并不会损害原 Petri 网控制器的活性、有界(或安全)性、可逆性.

### 3.3.2 资源库所的并行融合

已知类型库中存在着两种网类型 CMG ( $k$ )、SCMG ( $k$ ), 可以表示  $k$  个机器或过程并行运转. 例如图 4(c) 和 4(d) 所示的网模块, 但它们仅表示机器或过程分别独占各自的资源. 对此, 可补充两个网模块类型来反映资源的并行共享.

1) 模块类  $\overline{\text{CMG}}(k)$  定义为模块类 CMG ( $k$ ) 的补, 它通过将 CMG ( $k$ ) 模块中的  $k$  个并行独立 CMG 子网的资源库所融合为单一的资源库所而获得. 例如可以通过将图 4(c) 所示的 CMG (2) 的资源库所  $p_2$  与  $p_4$  融合为一个库所而获得该网块的补  $\overline{\text{CMG}}(2)$ .

2) 模块类  $\overline{\text{SCMG}}(k)$  定义为模块类 SCMG ( $k$ ) 的补. 将 SCMG ( $k$ ) 模块中的  $k$  个并行独立 SCMG 子网的资源库所融合为单一的资源库所可获得其对应的  $\overline{\text{SCMG}}(k)$ . 例如可以通过将图 4(d) 所示的 SCMG (2) 的资源库所  $p_5$  与  $p_{10}$  融合而获得其对应的补  $\overline{\text{SCMG}}(2)$ .

由 CMG ( $k$ ) (SCMG ( $k$ )) 到其对应的  $\overline{\text{CMG}}(k)$  ( $\overline{\text{SCMG}}(k)$ ) 的转变, 即为资源库所的并行融合.

令集合  $\bar{\mathcal{L}}_t = \{\overline{\text{CMG}}(k), \overline{\text{SCMG}}(k)\}$ , 作为  $\mathcal{L}_t$

的补充. 资源库所的并行融合仍会保持原有 Petri 网的行为属性, 相关命题及其证明已在文献[12] 中给出. 因此, 资源库所并行融合的场景可方便地通过使用  $\overline{\text{CMG}}(\cdot)/\overline{\text{SCMG}}(\cdot)$  模块重写对应的 CMG ( $\cdot$ )/SCMG ( $\cdot$ ) 模块来表示, 并且这种融合并不影响原来 Petri 网控制器的活性、有界(安全)性和可逆性的行为属性.

融合第 3.2 节所获得的中间 Petri 网中具有相同物理含义的资源库所, 可得到如图 7 所示的 Petri 网控制器, 仍为活的、有界和可逆的, 其中各库所和变迁的物理意义参见表 1 (见下页).

使用基于 INRS 的设计方法可获得 LBR Petri 网控制器. 这类控制器具有模块化结构, 各模块之间的交互仅通过明确的接口完成, 是可重构的. 下文将展开对这类 Petri 网控制器重构问题的研究.

## 4 基于 INRS 的 Petri 网控制器自动重构

### 4.1 自动重构机制

**定义 11.** 重构器定义为 INRS  $\mathcal{N} = (\mathcal{L}, \mathcal{R}, G)$ , 其中  $G$  为当前构形下的 RMS 的 LBR Petri 网控制器, 由第 3 节的设计方法获得,  $\mathcal{L}$  为通用的网模块类型库,  $\mathcal{R}$  是反映 RMS 构形变化的有限重写规则集. 自动应用这些重写规则, 当前控制器  $G$  将转变为目标构形所对应的控制器  $G'$ , 该过程称为  $G$  重构为  $G'$ .

重构器在实现 Petri 网控制器自动重构的过程中扮演着重要的角色, 具体的工作机理如下.

如图 8 (见下页) 所示, 给定的 RMS 的当前构形为  $C_1 = (W_1, O_1, Z_1)$ , 物理构形  $\Omega_1$ , Petri 网控制器为  $G = (\Gamma, M_0)$ . 设新生产任务所对应的新制造系统构形已给出, 记为  $C_2 = (W_2, O_2, Z_2)$ . 相应地,  $S$  由  $\Omega_1$  重构为  $\Omega_2$ .

以当前控制器  $G$  作为基础 Petri 网构建重构器

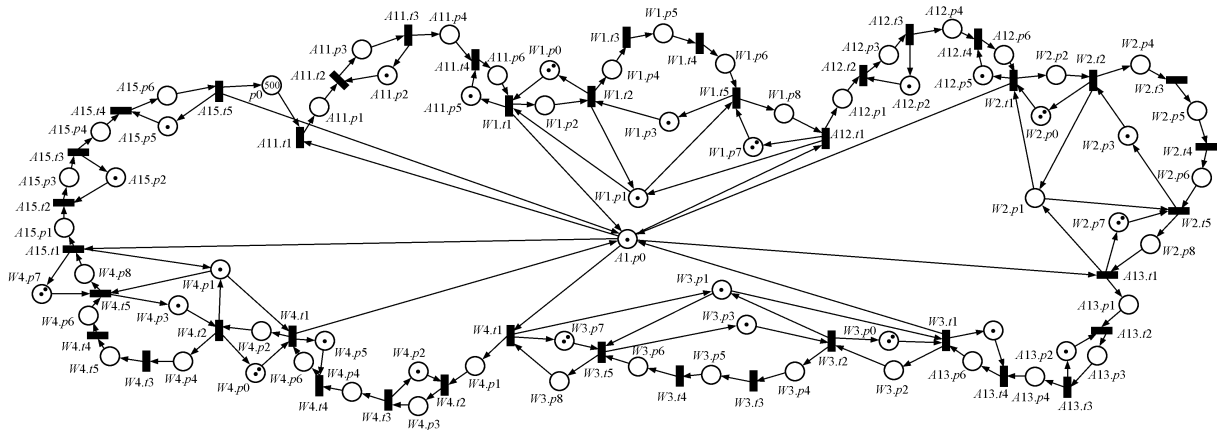


图 7 设计的结果 Petri 网控制器

Fig. 7 The resultant PN controller obtained by the design method

表 1 Petri 网控制器  $G_1$  中库所和变迁的意义  
Table 1 Meanings of places and transitions of PN supervisor  $G_1$

| 库所/变迁     | 意义                           | 库所/变迁     | 意义                 | 库所/变迁    | 意义              |
|-----------|------------------------------|-----------|--------------------|----------|-----------------|
| $p_0$     | 制造单元和待加工零件可用                 | $A1.p_0$  | AGV1 可用            | $Wi.t_2$ | $Ri$ 装载任务结束     |
| $A11.t_1$ | A11 任务启动                     | $A1i.t_1$ | $W_j$ 停止、 $A1i$ 启动 | $Wi.p_3$ | $Wi$ 的机床可用      |
| $A11.p_1$ | 移动到装/卸站 $L$                  | $A1i.p_1$ | 移动到零件提取点 $P_j$     | $Wi.p_4$ | $Wi$ 加工前的动作     |
| $A1i.p_2$ | $L (i = 1)$ 或提取点 $P_j$ 上零件可用 | $A1i.t_2$ | AGV1 停止移动          | $Wi.t_3$ | $Wi$ 加工前动作结束    |
| $A1i.p_3$ | AGV1 装载零件                    | $A1i.t_3$ | 结束装载               | $Wi.p_5$ | $Wi$ 加工进行中      |
| $A1i.p_4$ | 移动到交付点 $Di$                  | $A15.p_4$ | 移动到装/卸站 $L$        | $Wi.t_4$ | $Wi$ 加工结束       |
| $A1i.p_5$ | 交货点 $Di$ 可用                  | $A1i.t_4$ | 停止移动               | $Wi.p_6$ | $Wi$ 加工后的动作     |
| $A1i.p_6$ | AGV1 卸载零件                    | $Wi.t_1$  | $A1i$ 结束、 $Wi$ 启动  | $Wi.t_5$ | $Wi$ 加工后动作结束    |
| $A15.t_5$ | 任务 A15 结束                    | $Wi.p_0$  | 输入缓冲器 $IBi$ 可用     | $Wi.p_7$ | $OBi$ 上零件可用     |
| $Wi.p_1$  | 机器人 $Ri$ 可用                  | $Wi.p_2$  | $Ri$ 转移零件至 $IBi$   | $Wi.p_8$ | $Ri$ 转移零件至 $Pi$ |

注:  $i = 1, \dots, 5$  且  $j = i - 1, j \neq 0$

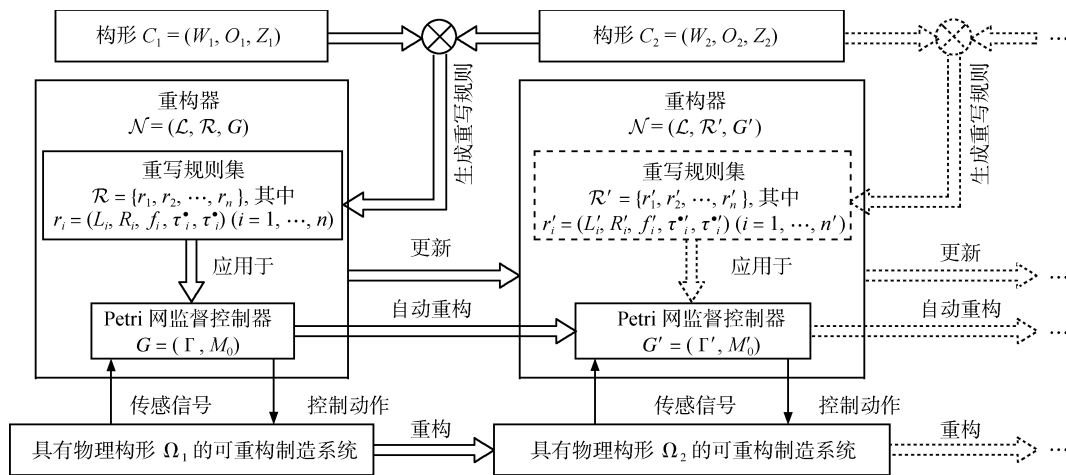


图 8 Petri 网控制器自动重构机制

Fig. 8 The mechanism of PN supervisor reconfiguration

$\mathcal{N} = (\mathcal{L}, \mathcal{R}, G)$ , 其中重写规则集  $\mathcal{R}$  中的规则是通过比较构形  $C_1$  和  $C_2$ , 辨识差异而生成. 重写规则的自动生成技术将在第 4.2 节介绍. 重构器将重写规则应用于  $G$ ,  $G$  会快速、自动地重构为 RMS 的新构形  $C_2$  所对应的新 Petri 网控制器  $G'$ .

更进一步, 随着生产需求驱动的制造系统构形的再次变化, 通过再次比较最近两次构形的差异, 可生成新的重写规则集. 以其更新重构器的当前规则并作用于当前控制器, 使得当前控制器再次重构为目标构形所对应的控制器. 以此类推, 重构器可以随着 RMS 构形的变化而不断演进, 实现 Petri 网控制器连续重构, 从而满足新的监督控制要求. 任意的重构过程中, 当前控制器所具有的活性、有界 (或安全性) 和可逆性的重要的行为属性将保持不变, 这一点为 INRS 所保证.

### 4.2 构形变化与重写规则生成

构形图上两个相异顶点的 3 种可能的连接方式

定义如下.

**定义 12.** 若构形图中任意两个相异的节点  $v_i, v_j$  满足以下任一条件: 1)  $v_i \bullet = \{v_j\}$  且  $\bullet v_j = \{v_i\}$ ; 2)  $\bullet v_i = \{v_j\}$  且  $v_j \bullet = \{v_i\}$ ; 3)  $v_i \bullet = v_j \bullet$  且  $\bullet v_i = \bullet v_j$ , 则称  $v_i$  和  $v_j$  直接连接, 而这些条件则称为直接连接条件, 对应的  $v_i$  和  $v_j$  组合结构可分别记为  $v_i|v_j, v_j|v_i$  和  $v_j \parallel v_i$ , 或统一表示为  $v_j \oplus v_i$ .

假定可重构制造系统  $S$  由构形  $C_1 = (W_1, O_1, Z_1)$  转变为  $C_2 = (W_2, O_2, Z_2)$ . 令添加到系统  $S$  中的机器的集合为  $M_a$ , 从  $S$  中移除的机器集合为  $M_r$ , 以及保留的机器的集合为  $M_s$ , 分别对应构形图的顶点集  $V_a, V_r$  与  $V_s$ .

#### 4.2.1 移除机器

假定  $m$  为移除的机器,  $m \in M_r$ , 并且构形图  $Z_1$  中存在两个直接连接的顶点  $v$  和  $v'$ , 其中  $v \in V_r$  和  $v' \in V_s$ , 分别对应着机器  $m$  和  $m'$ . 从  $S$  中移除  $m$  的构形变化可表示为一条重写规则, 即以  $Z_1$  中



的组合结构  $v' \oplus v$  所对应的组合网模块  $\psi \circ \eta(v' \oplus v)$  为该图重写规则的左手侧, 它对应着当前控制器中的全子网 (对象模块)  $\varphi \circ \eta(v' \oplus v)$ ; 而以  $m'$  对应的网模块  $\psi(m') (= \psi \circ (v'))$  为重写规则的右手侧, 图重写规则的输入和输出接口关系由左、右手侧的网模块类型本身决定. 若  $v$  与  $v'$  的组合为  $v|v'$  或  $v'|v$ , 即呈串行, 则重写规则可以简化为: 以  $m$  对应的 Petri 网模块  $\psi(m)$  为该重写规则的左手侧, 它对应着当前 Petri 网控制器中的全子网  $\varphi \circ \eta(m)$ , 而以 ST 为重写规则的右手侧.

此后, 删除  $Z_1$  中的顶点  $v$  并更新  $Z_1$ . 若  $S$  中有多个机器被移除, 则重复上述过程.

#### 4.2.2 添加机器

设机器  $m$  为添加到  $S$  中的机器,  $m \in M_a$ . 找到构形图  $Z_2$  中两个直接连接的顶点  $v$  和  $v'$ , 其中  $v \in V_a$ ,  $v' \in V_s$ ,  $\eta(v) = m$ ,  $\eta(v') = m'$ . 该构形变化可转变为 INRS 的图重写规则, 即以机器  $m'$  对应的网模块  $\psi(m') (= \psi \circ \eta(m'))$  为重写规则的左手侧, 它对应着当前控制器中的全子网  $\varphi \circ \eta(v')$ ; 而以组合结构  $v' \oplus v$  对应的网模块  $\varphi \circ \eta(v' \oplus v)$  为该重写规则的右手侧; 重写规则的输入、输出接口关系由左、右手侧的网模块类型本身决定.

此后, 将构形图  $Z_2$  中的结构  $v \oplus v'$  替换为顶点  $v'$ . 若多个机器被添加到系统中, 则重复上述过程.

#### 4.2.3 机器之间的逻辑关系的调整

若  $Z_1 \neq Z_2$  且  $V_1 = V_2$ , 则意味着系统  $S$  重构过程中机器间的逻辑关系发生改变. 对任意相异的顶点  $v$  和  $v'$ , 考虑下列三种情况:

1)  $v$  和  $v'$  在  $Z_1$  与  $Z_2$  中分别满足不同的直接连接条件.

这种构形变化可由一条图重写规则来表示, 该规则的左手侧为构形  $Z_1$  中的组合  $v \oplus v'$  所对应的网模块  $\psi_1 \circ \eta(v \oplus v')$ , 它对应着当前 Petri 网控制器中的对象模块  $\varphi \circ \eta(v \oplus v')$ . 重写规则的右手侧为  $Z_2$  中的组合  $v \oplus v'$  对应的网模块  $\psi_2 \circ \eta(v \oplus v')$ . 重写规则的输入和输出接口关系由左、右手侧的网模块类型决定. 规则生成后, 以顶点  $v$  或  $v'$  分别替换  $Z_1$  与  $Z_2$  中的组合结构  $v \oplus v'$ .

2)  $Z_1$  中  $v$  和  $v'$  是直接连接的; 而  $Z_2$  中  $v$  和  $v'$  未直接连接且无共同的前集或后集.

对于该构形变化, 可将  $v$  或  $v'$  视为被移除的顶点, 因此可以等效地按“移除机器”的情况来处理, 规则的具体生成过程见第 4.2.1 节. 相应地, 应以  $v$  或  $v'$  替换  $Z_1$  中的组合结构  $v \oplus v'$ .

3)  $Z_2$  中  $v$  和  $v'$  直接连接; 而  $Z_1$  中  $v$  和  $v'$  未直接连接且无共同的前集或后集.

可将  $v$  或  $v'$  视为添加的顶点, 于是可按照“添

加机器”的情况来处理, 具体过程见第 4.2.2 节. 相应地, 需以  $v$  或  $v'$  替换  $Z_2$  中的组合结构  $v \oplus v'$ . 此后, 若  $|V_1| > |V_2|$ , 重复第 4.2.1 节的过程; 若  $|V_1| < |V_2|$ , 则执行第 4.2.2 节的过程; 其他情况, 即  $|V_1| = |V_2|$  且  $Z_1 \neq Z_2$ , 则执行本节的 1)、2) 或者 3) 的过程, 直到  $Z_1 = Z_2$ .

#### 4.2.4 资源占有方式的转变

不同机器独占与共享资源场景的相互转变也是 RMS 的重要重构情况, 可表示为资源库所的融合与分裂. 资源库所的融合代表多个机器由分别独占不同的物理资源转变为共享同一物理资源的过程, 而资源库所的分裂反映了多个机器由共享物理资源转变为各自独占资源的过程. 与第 3.3 节所讨论的资源库所融合略有不同的是, 这里的资源库所允许代表不同的物理资源. 但它们在本质上是一致的, 即通过 INRS 的途径, 资源库所的串、并融合的过程仍然维持原 Petri 网控制器的活性、有界 (或安全) 性和可逆性.

资源库所分裂可划分为串、并行分裂两类. 举例来说, 以图 3(e) 所示的网模块作为机床的模型, 其中资源库所  $p_5$  表示装载和卸载零件用的机械臂 1. 若添加机械臂 2 来完成零件装载任务, 此时原来指代机械臂 1 的资源库所  $p_5$  应分解为两个独立的资源库所来表示这种变化, 可转变为图 3(d) 所示的网模块, 其中  $p_5$  和  $p_6$  分别指代机械臂 1 和机械臂 2. 由 SCMG 的定义可知, 串行资源库所分裂前后的场景均可表示为 SCMG 型网模块. 由定理 2 可知, 资源库所的串行分裂不损害原 Petri 网的重要行为属性.

资源库所的并行分裂问题可描述为, 将表示各机器共享同一物理资源的网模块  $\overline{\text{CMG}}(\cdot)$  ( $\overline{\text{SCMG}}(\cdot)$ ) 转换为表示各机器分别独享各自物理资源的对应网模块  $\text{CMG}(\cdot)$  ( $\text{SCMG}(\cdot)$ ), 即以后者重写前者, 并且这一过程不影响原 Petri 网控制器的活性、有界 (安全) 性和可逆性.

#### 4.2.5 其他情况

此外, 机器的具体操作序列的变化反映了机器的功能调整. 操作序列应建模为通用的网模块类型库中所定义的类型模块. 而操作序列的变化, 很容易由 INRS 的重写规则来刻画, 即以新的操作序列的网模块为重写规则的右手侧, 而以当前的动作序列的网模块为重写规则的左手侧. RMS 产能变化可最终表现为产品的批量的改变, 以及机器的输入、输出缓冲器容量的变化. 根据 Petri 网的原则, 这些变化可以表现为对应的资源库所中令牌数目的变化.

5 仿真研究

设第 1 节给出的可重构加工单元的当前构形为  $C_1 = (W_1, O_1, Z_1)$ , 目标构形  $C_2 = (W_2, O_2, Z_2)$ , 其中构形图  $Z_1 = (V_1, E_1)$  和  $Z_2 = (V_2, E_2)$  分别如图 2(a) 和 2(b) 所示. 构形  $C_1$  对应 Petri 网控制器  $G_1$ , 如图 7 所示, 它是可重构的, 具有活性、有界性和可逆性. 比较  $Z_1$  和  $Z_2$ , 可确定  $V_a = \{a_{21}, a_{22}, a_{23}\}$ ;  $V_r = \{a_{13}, a_{14}, a_{15}\}$ . Petri 网控制器  $G_1$  重构的步骤如下.

**步骤 1.** 构建重构器  $\mathcal{N} = (\mathcal{L}, \mathcal{R}, G)$ , 令  $\mathcal{R} = \emptyset$ ,  $G = G_1$ .

**步骤 2.** 考虑移除  $A13$ 、 $A14$  和  $A15$  的构形变化. 从  $V_r$  中分别取出顶点  $a_{13}$ 、 $a_{14}$ 、 $a_{15}$ , 找出  $Z_1$  中与它们直接连接的顶点, 依次为  $w_2$ 、 $w_3$ 、 $w_4$ . 可生成 3 条重写规则, 应用后获得中间 Petri 网控制器  $G_I$ . 此时,  $G = G_I$ .

**步骤 3.** 考虑添加  $A21$ 、 $A22$  和  $A23$  的构形变化. 依次从  $V_a = (a_{21}, a_{22}, a_{23})$  中取出各顶点, 找出  $Z_2$  中与之直接连接的顶点, 分别为  $w_2$ 、 $w_4$  和  $w_4$ ,

其中  $a_{22}$  与  $w_4$ , 以及  $a_{23}$  与  $w_4$  满足不同的直接连接条件. 同样可生成 3 条重写规则, 应用后获得中间 Petri 网控制器  $G_{II}$ . 此时,  $G = G_{II}$ ,  $Z_1 \neq Z_2$  且  $V_1 = V_2$ .

**步骤 4.** 考虑机器之间的逻辑关系调整.

1) 顶点  $w_1$  和  $a_{12}$  在  $Z_1$  中直接连接, 而在  $Z_2$  中不满足直接连接条件且无共同的前集或后集. 据此, 可以生成 1 条重写规则, 应用后获得 Petri 网控制器  $G_{III}$ . 此时,  $G = G_{III}$ ,  $|V_1| < |V_2|$ , 即  $V_2$  中多了顶点  $a_{12}$ .

2) 检查  $Z_2$ , 找到与  $a_{12}$  直接连接的顶点, 为  $w_3$ . 由此, 可生成图重写规则并应用, 得到 Petri 网控制器  $G_{IV}$ . 此时,  $G = G_{IV}$ ,  $V_1 = V_2$  但  $Z_1 \neq Z_2$ .

3) 顶点  $w_2$  和  $w_3$  在  $Z_1$  与  $Z_2$  中满足不同的直接连接条件, 可生成重写规则并应用, 得到控制器  $G_V$ .  $Z_1$  与  $Z_2$  的变化如图 9 所示.

**步骤 5.** 对  $G_V$  中相同物理意义的资源库所进行融合, 获得如图 10 所示的 Petri 网控制器  $G_2$ , 即为可重构加工单元构形  $C_2$  所对应的 Petri 网控制器. 可见,  $G_2$  仍为活的、有界和可逆的.

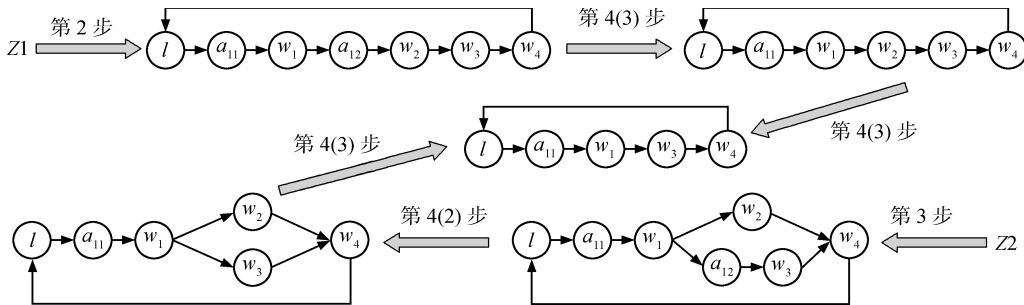


图 9 构形图  $Z_1$ 、 $Z_2$  的演变

Fig. 9 Evolution of configuration graphs  $Z_1$  and  $Z_2$

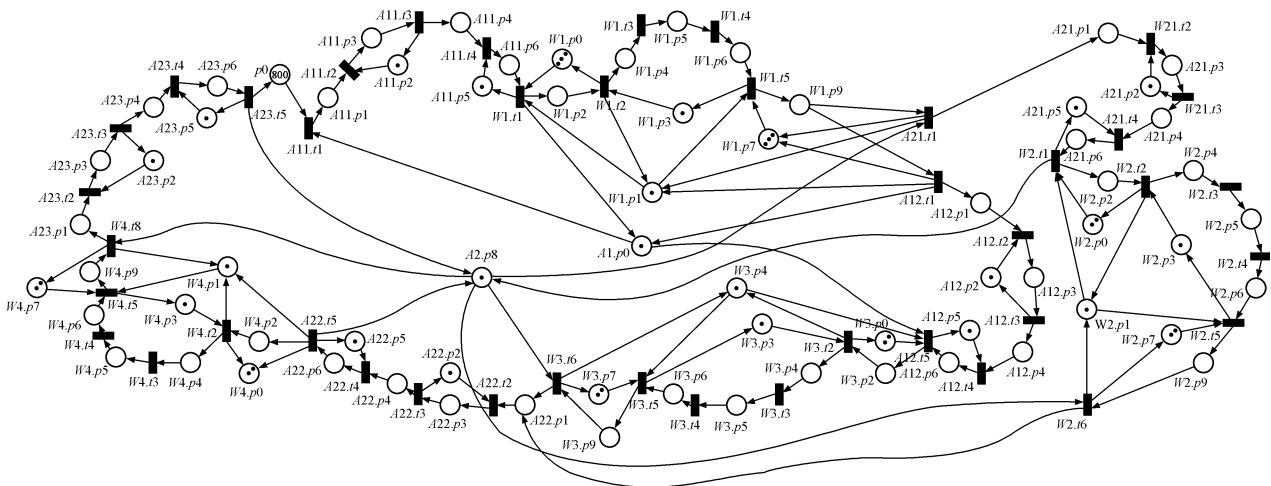


图 10 重构的结果 PN 控制器

Fig. 10 The resultant PN supervisor obtained by reconfiguration

仿真结果表明, 本文提出的自动重构方法, 可实现 Petri 网控制器自动、有效的重构, 从而快速地响应制造系统构形的变化.

## 6 结论

1) 提出了改进网重写系统, 它既可以动态改变 Petri 网模型结构, 又不损害其原有的行为属性.

2) 给出了 RMS 构形的形式化表示, 介绍了基于 INRS 的模块化、可重构的 Petri 网监督控制器的设计方法. 针对这类 Petri 网控制器, 提出了基于 INRS 的自动重构方法. Petri 网控制器设计与重构方法均可保证结果的正确性, 避免了校验.

3) 仿真结果表明, Petri 网控制器自动重构方法是有效的.

此外, 随着 RMS 规模的变大, 构形变化将复杂化, 因此控制器重构的复杂性和代价问题应作为下一步的研究目标.

## References

- Ramadge P J, Wonham W M. Supervisory control of a class of discrete-event processes. *SIAM Journal on Control and Optimization*, 1987, **25**(1): 206–230
- Zhang J, Chan F T S, Li P G, Lau H C W, Ip R W L, Samaranyak P. Investigation of the reconfigurable control system for an agile manufacturing cell. *International Journal of Production Research*, 2002, **40**(15): 3709–3723
- Brennan R W, Fletcher M, Norrie D H. An agent-based approach to reconfiguration of real-time distributed control systems. *IEEE Transactions on Robotics and Automation*, 2002, **18**(4): 444–451
- Monfared R P, Weston R H. The re-engineering and reconfiguration of manufacturing cell control systems and reuse of their components. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 1997, **211**(7): 495–508
- Koren Y, Heisel U, Jovane F, Moriwaki T, Pritchow G, Uisoy G. Reconfigurable manufacturing systems. *CIRP Annals-Manufacturing Technology*, 1999, **48**(2): 527–540
- Park J, Kim J. A generic event control framework for modular flexible manufacturing systems. *Computers and Industrial Engineering*, 2000, **38**(1): 107–123
- Kalita D, Khargonekar P P. Formal verification for analysis and design of logic controllers for reconfigurable machining systems. *IEEE Transactions on Robotics and Automation*, 2002, **18**(4): 463–474
- Park E, Tilbury D M, Khargonekar P P. A modeling and analysis methodology for modular logic controllers of machining systems using Petri nets formalism. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Application and Reviews*, 2001, **31**(2): 168–188
- Jiang Z, Zou M J, Tu P Y, Fung R Y K. Object-oriented Petri nets with changeable structure (OPNs-CS) for production systems modeling. *The International Journal of Advanced Manufacturing Technology*, 1999, **15**(6): 445–459
- Liu J, Darabi H. Control reconfiguration of discrete event systems controllers with partial observation. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 2004, **34**(6): 2262–2272
- Chandra V, Bhattacharyya S. Adaptive supervisory control of hybrid systems having discrete event abstractions. In: *Proceedings of the ISA EXPO 2005 Technical Conference*. Chicago, USA: ISA, 2005. 1018–1029
- Li Jun. Research on Automatic Reconfiguration of Petri Net Controllers for Reconfigurable Manufacturing Systems [Ph. D. dissertation], Southeast University, 2007 (李俊. 可重构制造系统 Petri 网控制器自动重构的研究 [博士学位论文], 东南大学, 2007)
- Llorens M, Oliver J. Structural and dynamic changes in concurrent systems: reconfigurable Petri nets. *IEEE Transactions on Computers*, 2004, **53**(9): 1147–1158



李俊 东南大学博士后. 主要研究方向为 Petri 网, 离散事件仿真与控制, 可重构制造系统. 本文通信作者.

E-mail: j.li@seu.edu.cn

(LI Jun Postdoctoral at Southeast University. His research interest covers Petri nets, discrete-event simulation and control, and reconfigurable manufacturing systems. Corresponding author of this paper.)



戴先中 东南大学教授. 主要研究方向为人工神经网络测量与控制, 运动控制, 过程控制与电力系统控制.

E-mail: xzdai@seu.edu.cn

(DAI Xian-Zhong Professor at Southeast University. His research interest covers artificial neural network measure and control, motion control, process control, and power system control.)



孟正大 东南大学教授. 主要研究方向为机器人运动和力控制.

(MENG Zheng-Da Professor at Southeast University. His research interest covers motion and force control of robotics.)