

DOI: 10.7652/xjtuxb201611007

对称离散事件系统状态树结构模型的控制函数不变性研究

焦亭, 甘永梅, 肖国春
(西安交通大学电气工程学院, 710049, 西安)

摘要: 针对对称离散事件系统中使用监督控制理论计算得到的自动机形式的监督控制器状态数较多且无法清晰反映控制逻辑等问题,提出了基于状态树结构的抽象控制函数计算方法。该方法通过充分利用系统的对称性,对避免缓冲区出现上溢或者下溢的性能指标,采用事件重标记映射提取各组中处于加工完成状态的组件数目,从而省去了各组件复杂的运行细节;然后利用状态树结构计算得到基于抽象状态信息的控制函数;最后,结合各组组件并行工作与串行工作的实例,分析所得控制函数的不变性,即在缓冲区容量固定的前提下,控制函数所需的状态信息与组件总数量无关。实验结果表明:在实际应用中借助控制函数的不变性可在结构相同的组件增减时免于重复计算控制函数;各组中重标记为同一事件的各可控事件可由同一控制函数进行使能,有效减少了控制函数的个数;相比于自动机形式的监督控制器,控制函数状态数更少且能更清晰地描述控制逻辑。

关键词: 离散事件系统;监督控制理论;状态树结构;控制函数;谓词逻辑

中图分类号: TP273 **文献标志码:** A **文章编号:** 0253-987X(2016)11-0043-06

Invariance Property of the Control Functions in Symmetric Discrete-Event Systems Modeled by State Tree Structures

JIAO Ting, GAN Yongmei, XIAO Guochun
(School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: The symmetric discrete-event systems (DES) consist of groups of identical components. For such DES, the supervisor in the form of automata synthesized by the supervisory control theory often has large number of state and is nontransparent in control logic. Thus a computational approach of abstract control function is proposed. This approach fully exploits the symmetry of the system. For the performance index to avoid the underflow or overflow of buffers, the number of components in the status of processing finish is extracted with event relabeling map, thereby the processing details of each component are omitted. Then the abstract control functions are computed with the state tree structures (STS) based on the abstract status information. Finally, by illustrating examples of components working in parallel and serial, the invariance property of control functions are analyzed. Namely, with fixed buffer sizes, the status information required by the control function is irrelevant to the total number of components. The experimental results showed that by utilizing the invariance property of control functions, the repeated computation of control functions is avoided when identical components are

收稿日期: 2016-05-12。 作者简介: 焦亭(1988—),男,博士生;甘永梅(通信作者),女,副教授。 基金项目: 国家建设高水平大学公派留学生资助项目(留金发[2014]3026)。

网络出版时间: 2016-09-14 网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20160914.1805.006.html>

added or deleted. Meanwhile, all controllable events relabeled by the same symbol can be enabled by one abstract control function; thus the number of control functions is reduced. Moreover, compared with the controller represented by automata, the control functions have fewer states and are more transparent in control logic.

Keywords: discrete-event systems; supervisory control theory; state tree structures; control functions; predicate logic

在由 Ramadge 与 Wonham 建立的离散事件系统监督控制理论^[1-2]中,由多组具有相同结构组件组成的系统具有对称性。对于不关注各组件之间性能指标差异的系统,可利用离散事件系统的对称性对系统进行化简。文献[3]利用群理论描述具有相同结构组件串联所构成系统的对称性,以达到简化控制器的目的。文献[4]对由具有相同结构组件组成的系统执行商自动机构建运算,以简化集中监督控制器规模。对于由同一模板生成的系统,文献[5]分析了系统的死锁与阻塞问题,文献[6]提出了基于广播方式且满足交换律、结合律的系统合成运算算法。这些化简方法均基于商自动机构建算法,计算过程复杂,本文提出通过事件重标记映射提取系统的抽象状态信息以简化系统。事件重标记映射不但能在很大程度上缩减被控对象与监督控制器规模,而且化简的实现过程更加直观简便,且所得的重标记后的监督控制器能直接用于控制。

在利用事件重标记映射提取系统的抽象状态信息后,基于所得的抽象状态信息建立状态树结构模型^[7-10],计算得到系统中各可控事件的控制函数。此外,文献[11]通过证明使用 supreduce 算法计算所得控制同余(control congruence)的不变性以证明控制器的不变性,但此证明过程较为繁琐,而状态树结构中控制函数清晰的控制逻辑使得控制函数的不变性证明变得直观简便。

文献[12-14]中的性能指标主要是参数化模板或谓词表达式,本文主要研究在缓冲区容量固定的前提下防止缓冲区上溢与下溢的性能指标。在缓冲区容量固定的情况下,借助状态树结构计算系统中可控事件的控制函数,可发现控制函数具有不变性,即控制函数不随组件数目的增减而变化。

1 基础知识

1.1 DES 自动机模型

离散事件系统利用自动机对系统进行建模,自动机可表示为 5 元组 $G = (Q, \Sigma, \delta, q_0, Q_m)$, 其中 Q 为状态集合, Σ 为事件集合, $\delta: Q \times \Sigma \rightarrow Q$ 用于描述

系统中状态的变迁关系, q_0 为初始状态, $Q_m \subseteq Q$ 为标识状态集合。一般地, 转移函数可扩展为 $\delta: Q \times \Sigma^* \rightarrow Q$, 其中 Σ^* 表示空字符串 ϵ 与 Σ 上的所有有限长度字符串的并集, 并用 $\delta(q_0, s)!$ 表示 $\delta(q_0, s)$ 有定义。定义自动机 G 所表示语言的闭特性为 $L(G) = \{s \in \Sigma^* \mid \delta(q_0, s)!\}$, 标识特性为 $L_m(G) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}$ 。

对任意语言 $K \subseteq \Sigma^*$, 如果满足 $(\forall s, \sigma) s \in \bar{K} \ \& \ \sigma \in \Sigma_u \ \& \ s\sigma \in L(G) \Rightarrow s\sigma \in \bar{K}$, 则称 K 关于 G 可控。

对于 G 的监督控制可定义为映射 $V: L(G) \rightarrow \Gamma$, 其中所有控制模式组成的集合 Γ 记作 $\Gamma = \{\gamma \in Pwr(\Sigma) \mid \gamma \supseteq \Sigma_u\}$ 。 G 在 V 的监控下可记作 V/G , 并用 $L(V/G)$ 表示 V/G 的闭特性^[1]。令 $M \subseteq L_m(G)$, 定义 (M, G) 的标识非阻塞监督控制为映射 $V: L(G) \rightarrow \Gamma$, V/G 的标识特性定义为 $L_m(V/G) = L(V/G) \cap M$ 。假设用 $E \subseteq \Sigma^*$ 表示性能指标, E 中的所有可控语言表示为 $C(E) = \{K \subseteq E \mid K \text{ 关于 } G \text{ 可控}\}$ 。因 $C(E)$ 中的元素形成上晶格结构, 存在最大元素 $\sup C(E) = \bigcup \{K \mid K \in C(E)\}$, 并可由 TCT 软件中的 supcon 算法计算得到对应自动机形式的监督控制器。

1.2 状态树结构

对 DES 进行建模的状态树结构^[8-9]可表示为 6 元组 $G_{st} = (S_t, \mathcal{H}, \Sigma, \Delta, S_{t_0}, S_{tm})$, 其中状态树 S_t 将系统状态集合表示为分层结构, H 为被控系统 G_{st} 中描述系统局部行为的各组件实时子整体, Σ 为事件集合, 其被划分为可控事件集合 Σ_c 与不可控事件集合 Σ_u 。令 $S(S_t)$ 表示 S_t 上所有的子状态树集合, $\Delta: S(S_t) \times \Sigma \rightarrow S(S_t)$ 为全局转移函数, $S_{t_0} \in S(S_t)$ 为初始状态树, $S_{tm} \subseteq S(S_t)$ 为标识状态树集合。

$\mathcal{B}(S_t) \subseteq S(S_t)$ 表示所有基本状态树集合, 定义 $\mathcal{B}(S_t)$ 上的谓词 $P: \mathcal{B}(S_t) \rightarrow \{0, 1\}$, 其中 0、1 分别表示逻辑假、逻辑真。谓词 P 可由基本状态树集合 $B_P := \{b \in \mathcal{B}(S_t) \mid P(b) = 1\}$ 判定, 其中 $P(b) = 1$ 常被记作 $b \models P$ 。对于子状态树 $T \in S(S_t)$ 定义 $T \models P$ 当且仅当 $(\forall b \in \mathcal{B}(T)) b \models P$, 状态树 G_{st} 也可记作 $G_{st} = (S_t, \mathcal{H}, \Sigma, \Delta, P_0, P_m)$, 其中 P_0 、 P_m 为初始谓

词、标识谓词。

定义 $P(S_t)$ 为 $\mathcal{B}(S_t)$ 上的谓词集合。引入偏序关系 $<$, 使得 $P < P'$, 当且仅当 $(\neg P) \vee P'$, 对任意 $b \in \mathcal{B}(S_t)$, 如果 $b \models P \Rightarrow b \models P'$, 则 $P < P'$ 。可达谓词、协同可达谓词可分别记作 $R(G_{st}, P)$ 、 $C_r(G_{st}, P)$, 如果 $R(G_{st}, P) < C_r(G_{st}, P)$, 则称谓词 P 关于 G_{st} 非阻塞。

对任意 $\sigma \in \Sigma$ 定义映射 $M_\sigma(P): P(S_t) \rightarrow P(S_t)$ 来表示 $b \models M_\sigma(P)$, 当且仅当 $\Delta(b, \sigma) \models P$ 。如果 P 满足 $(\forall \sigma \in \Sigma_u) P < M_\sigma(P)$, 则称谓词 P 弱可控。将谓词 P 所有非阻塞且弱可控的子谓词记作 $\mathcal{N}_C(P)$, 由于 $\mathcal{N}_C(P)$ 非空且在任意并运算 $\vee$ 作用下封闭, 故存在最大元素 $\sup \mathcal{N}_C(P) := \vee \{K \mid K \in \mathcal{N}_C(P)\}$ 。定义状态反馈控制 $f: \mathcal{B}(S_t) \rightarrow \Pi$, $\Pi := \{\Sigma' \subseteq \Sigma \mid \Sigma_u \subseteq \Sigma'\}$, 对任意事件 $\sigma \in \Sigma$ 定义控制函数 $f_\sigma: \mathcal{B}(S_t) \rightarrow \{0, 1\}$, $f_\sigma(b) = 1$ 当且仅当 $\sigma \in f(b)$, 控制函数 f 可由集合 $\{f_\sigma \mid \sigma \in \Sigma\}$ 表示。由定义可知, 对于所有不可控事件 σ , $f_\sigma(b) = 1$ 。状态反馈控制 f 可由控制函数 $f_\sigma := M_\sigma(\sup \mathcal{N}_C(P))$ 表示, 对任意 $b \in \mathcal{B}(S_t)$, $f_\sigma(b) = 1$, 当且仅当 $\Delta(b, \sigma) \models \sup \mathcal{N}_C(P)$ 。控制函数可由 STSLib 软件计算得到^[8]。

1.3 事件重标记映射

假设被控对象 $G = (Q, \Sigma, \delta, q_0, Q_m)$ 协同可达, 其中 $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, $\Sigma_r = \Sigma_o$ 。为将被进行重标记操作的事件集合, Σ_{uo} 为不可观测事件集合。令 $T = T_c \dot{\cup} T_u$ 表示重标记后的事件集合, 定义事件重标记映射 $R: \Sigma^* \rightarrow T^*$ 使其满足条件: $R(\epsilon) = \epsilon$; $R(\sigma) = \epsilon, \sigma \in \Sigma_{uo}$; $R(\sigma) \in T_u, \sigma \in \Sigma_u$; $R(\sigma) \in T_c, \sigma \in \Sigma_c$; $R(s\sigma) = R(s)R(\sigma), s \in \Sigma^*, \sigma \in \Sigma$ 。事件重标记映射 R 将不可观测事件重标记为 ϵ , 且要求可控事件经重标记后仍为可控事件, 不可控事件经重标记后仍为不可控事件。

事件重标记映射 R 的示意图如图 1 所示, ϕ 的结果代表自动机对应的语言。使用 TCT 中改进后的 relabel 算法, 用 λ 表示, 计算进行事件重标记后所得的自动机 $G_R = \lambda(G)$, 使其满足 $L_m(G_R) = R(L_m(G))$, $L(G_R) = R(L(G))$ 。

事件重标记算法 λ 主要分 2 步实现: 将 G 中所有不可观测事件重标记为空字符 ϵ , 其余事件按事先制定的重标记规则进行标记; 将事件重标记后得到的不确定有限状态自动机通过子集生成算法^[15] 转换为确定有限状态自动机, 即为所求的结果 G_R 。子集生成算法的功能为将不确定有限状态自动机转

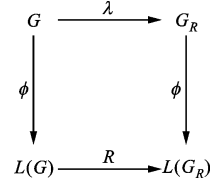


图 1 重标记映射及实现过程示意图

换为与其等价的确有限状态自动机。对于任意字符串 $t \in L_m(G_R)$, 均存在 $s \in L_m(G)$, 使得 $R(s) = t$; 反之, 对于任意字符串 $s \in L_m(G)$, 均有 $R(s) \in L_m(G_R)$, 所以 $L_m(G_R) = R(L_m(G))$, 同理可得 $L(G_R) = R(L(G))$ 。在具有对称性的离散事件系统 G 中, 被事件重标记映射标记为同一字符串的所有字符串进入的状态将生成一个状态子集。由于系统的对称性, 所生成状态子集为对 G 中状态的一个等价划分^[16], 因此生成的状态子集总数量少于原系统的状态总数, 即进行事件重标记后所得结果 G_R 的状态数比 G 的状态数少。

2 抽象状态信息提取

本文研究的系统由 $n \in N$ 组具有相同结构的组件组成, 对于各组 $\mathcal{G}_i, i \in \{1, \dots, n\}$ 中的任意组件 $G_{ij}, G_{ij'}, j, j' \in \{1, \dots, |\mathcal{G}_i|\}$, $\lambda(G_{ij}) = \lambda(G_{ij'})$, 即通过事件重标记映射忽略各组中组件个体差异。

本文制造系统如图 2 所示, 系统中的组件划分为 $\mathcal{G}_1 = \{I_i\}, i \in \{1, \dots, m\}$; $\mathcal{G}_2 = \{O_j\}, j \in \{1, \dots, n\}$, 对所有 $i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$, 将事件 $i_{11}, i_{12}, j_{21}, j_{22}$ 分别重标记为 11、12、21、22。本文所有的状态转移图中, 用无源的单向箭头进入状态表示初始状态, 用无目标状态的箭头出来状态表示标识状态, 如果一个状态既是初始状态又是标识状态, 则用双向箭头表示。

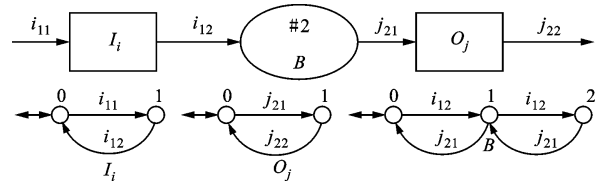


图 2 本文制造系统示意图

在忽略组件个体差异后, 针对防止缓冲区出现上溢或下溢的性能指标, 需要获知各组中当前状态处于即将往缓冲区中放入工件的组件数量, 即 I_i 中处于状态 1 的组件数量。为获得该数据, 给出定义 $I_i = (Z_i, \Sigma_i, \delta_i, z_{io}, Z_{im})$, 将其中取工件和往缓冲区中放入工件的事件分别记作 σ_{il}, σ_{iu} 。假定 $\delta_i(q,$

$\sigma_{liu}) = q'$, 拟放入工件事件集合为 $\Sigma_{liq} = \{\sigma \in \Sigma_{li} \mid \sigma \neq \sigma_{liu}, \delta_{li}(q, \sigma)!\}$ 。对于图 2 中的组件 $I_i, q=1, \sigma_{li} = i_{11}, \sigma_{liu} = i_{12}, \Sigma_{liq} = \emptyset$; 对于图 3 中的组件 $I_i, q=1, \sigma_{li} = i_{11}, \sigma_{liu} = i_{12}, \Sigma_{liq} = \{i_{14}\}$ 。 Σ_{li} 中其他事件因与缓冲区上溢或下溢无关可设为不可观测事件, 在进行 λ 运算时这些事件被重标记为 ϵ 。

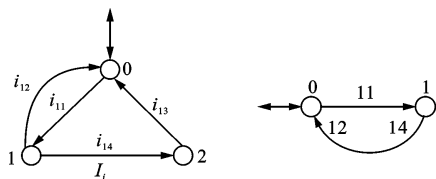


图3 复杂组件重标记前后的结构示意图

提取 I_i 所在组控制信息的实现过程为: 计算 $I = ||_{i=1}^m I_i$, ‘||’ 代表同步积运算; $I_R := \lambda(I) = (X, \Sigma_r, \xi, x_0, X_m), \Sigma_r = \{R(\sigma_{li}), R(\sigma_{liu})\}$; 对于 $\forall \sigma \in \Sigma_{liq}, x, x' \in X$, 如果 $\xi(x, R(\sigma_{liu})) = x'$, 则须往 I_R 中添加转移函数 $\xi(x, R(\sigma)) = x'$ 。

以图 2 中所示系统为例, 取 $m=3$, 所得 I_R 有 4 个状态, 即 $q \in \{0, \dots, 3\}$, 各状态表示目前 I_i 所在组中有 q 个组件处于即将往缓冲区中放入工件的状态; 一旦有组件往缓冲区中放入工件, 则将 I_R 的当前状态更新为 $q-1$ 。根据同步积的性质可知 I 具有 2^m 个状态, 由于输入侧最多有 m 个组件同时工作, I_R 仅有 $m+1$ 个状态, 由此可见事件重标记运算能化简对称离散事件系统的状态数。类似地, 可得 $O_R = \lambda(||_{j=1}^n O_j)$ 。

3 控制函数不变性研究

3.1 问题描述

经事件重标记映射计算得到的各组抽象状态信息可建立状态树结构模型, 将自动机转换为状态树结构的具体过程可参见文献[8], 然后通过 STSLib 软件计算得到各可控事件对应的控制函数。

对于基于抽象状态信息建立的状态树结构 G_{st} 与防止缓存区上溢与下溢的性能指标对应的谓词表达式 P , 假设利用 STSLib 软件计算得到各可控事件 τ_i , 对应的控制函数为 $f_i, i \in \{1, \dots, |T_c|\}$, 其中 $T_c = R(\Sigma_c) := \{\tau_i\}$ 表示重标记后系统的可控事件集合。如果对任意数目的组件, 控制函数 f_i 保持不变, 则称控制函数具有不变性。

3.2 并行工作系统控制函数不变性分析

对于图 2 中所示系统, 各组中的组件均处于并行工作状态, 利用事件重标记映射分别计算输入侧与输出侧的控制信息模型 I_R, O_R 。令 $m=3, n=1$,

经 STSLib 软件计算可得事件 11、21 的控制函数分别为 f_{11}, f_{21} , 如图 4 所示。其中, 实线箭头代表逻辑真, 虚线箭头代表逻辑假。

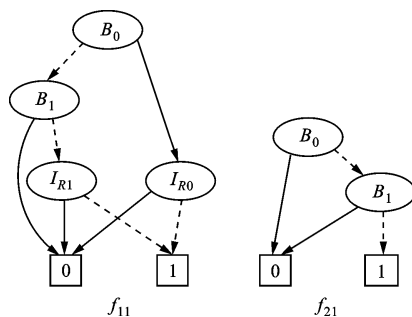


图4 控制函数 f_{11}, f_{21} 示意图

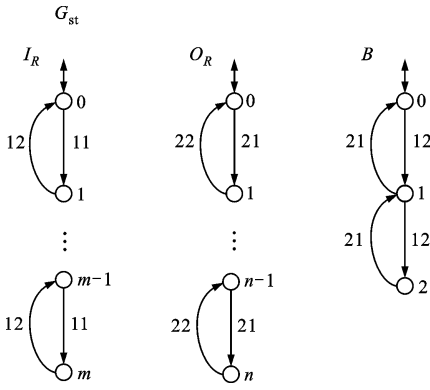
以 f_{11} 为例, I_R 有 4 个状态, 用 $I_{R0}=0, I_{R1}=0$ 表示 I_R 的当前状态为状态 0, 用 $I_{R0}=1, I_{R1}=0$ 表示 I_R 的当前状态为状态 1, 用 $I_{R0}=0, I_{R1}=1$ 表示 I_R 的当前状态为状态 2, 用 $I_{R0}=1, I_{R1}=1$ 表示 I_R 的当前状态为状态 3。由于缓冲区 B 有 3 个状态, 因此用 $B_0=0, B_1=0$ 表示 B 的当前状态为状态 0; 用 $B_0=1, B_1=0$ 表示 B 的当前状态为状态 1; 用 $B_0=0, B_1=1$ 表示 B 的当前状态为状态 2。箭头所指向的方框如果为 0, 则表示系统在当前状态组合下事件 11 (即各事件 i_{11}) 不能发生; 箭头所指向的方框如果为 1, 则表示系统在当前状态组合下事件 11 (即各事件 i_{11}) 可以发生, 此时缓冲区中有 1 个工件, 输入侧无组件工作, 可再允许输入侧 1 个组件取工件。

命题 1 对任意 $m \geq 3, n \geq 1, f_{11}, f_{21}$ 具有不变性。

系统对应的状态树结构模型 $G_{st} := (S_t, \mathcal{H}, T, \Delta, b_0, \{b_0\})$, 如图 5 所示, 其中重标记后事件集合 $T = \{11, 12, 21, 22\}$ 。 b_0 表示初始基本状态树, 其中 $v_{R1}=0, v_{R0}=0, v_B=0$, 即 I_R, O_R, B 均处于初始状态。由 f_{11} 可知使能事件 11 的谓词为 $i+k < 2$, 其中 $i \in \{0, \dots, m\}, j \in \{0, \dots, n\}, k \in \{0, \dots, 2\}, i, j$ 分别表示输入侧与输出侧处于状态 1 的组件数量, k 表示缓冲区中现有组件数量。

因此, 仅有 3 种合法基本状态树, 即 $i=0, k=0, i=1, k=0, i=0, k=1$, 分别记作 $B_{11}^1, B_{11}^2, B_{11}^3$ 。将事件 11 被使能的合法基本状态树记作 $B_{11} := \{B_{11}^1, B_{11}^2, B_{11}^3\}$ 。

由于 I_R 中仅有状态 0、1、2 可在系统运行过程中被访问到, 因此 B_{11} 不变, 并且只有状态 0、1 允许事件 11 发生, 因此 B_{11} 可等价地表示成 f_{11} 。 f_{11} 可解读为: 如果 $B_0=0, B_1=0$ 缓冲区为空, 当输入侧无组件处于状态 1 ($I_{R0}=0, I_{R1}=0$) 或仅有 1 个组件

图 5 状态树结构 G_{st}

处于状态 1 ($I_{R0}=1, I_{R1}=0$) 时事件 11 被使能; 如果 $B_0=1, B_1=0$, 即缓冲区已有 1 个工件, 则输入侧无组件处于状态 1 ($I_{R0}=0, I_{R1}=0$) 时事件 11 才能被使能; 如果 $B_0=0, B_1=1$, 即缓冲区已满, 则事件 11 不被使能, f_{11} 具有不变性。使能事件 21 的谓词为 $k=1$ 或 2, 在缓冲区容量固定为 2 的前提下, 对任意 $m \geq 3, n \geq 1$ 该谓词保持不变, f_{21} 具有不变性。

由命题 1 可知, 对任意数量的输入侧与输出侧组件 $m \geq 3, n \geq 1$, 输入侧组件开始工作 (事件 i_{11} 被使能) 当且仅当 f_{11} 使能事件 11; 输出侧组件开始工作 (事件 j_{21} 被使能) 当且仅当 f_{21} 使能事件 21。由于 f_{11}, f_{21} 与 m, n 取值无关, 因此控制函数具有不变性。

当 $m=1$ 时, I_R 仅有 2 个状态, 输入侧至多有 1 个组件处于状态 1, 因此只要缓冲区未滿 ($B_1=0$), f_{11} 就可使能事件 11。当 $m=2$ 时, I_R 有 3 个状态, 此时输入侧至多有 2 个组件处于状态 1, 当缓冲区为空时, 不论输入侧组件处于何种状态, f_{11} 均可使能事件 11。因此, $m=1$ 或 2 只是 $m \geq 3$ 时的特殊情况。

针对防止缓存区上溢与下溢的性能指标, 以 $m=9, n=9$ 为例, 文献[11]中计算得到的集中监督控制器有 29 184 个状态, 其对应的重标记后的自动机有 60 个状态; 相比之下, 满足相同的性能指标, f_{11}, f_{21} 比集中监督控制器更为简化。此外, 集中监督控制器的规模将随着输入侧与输出侧组件数量的增多而增大, 而 f_{11}, f_{21} 保持不变。

3.3 串行工作系统控制函数不变性分析

对于图 6 所示串行工作系统, 文献[3]选用构建商自动机的方式对其进行化简, 实现过程较为复杂, 本文从控制函数不变性的角度直观地获得系统的控制逻辑, 并结合系统对称性证明 M_0, M_1, M_2 可由同一谓词进行控制。

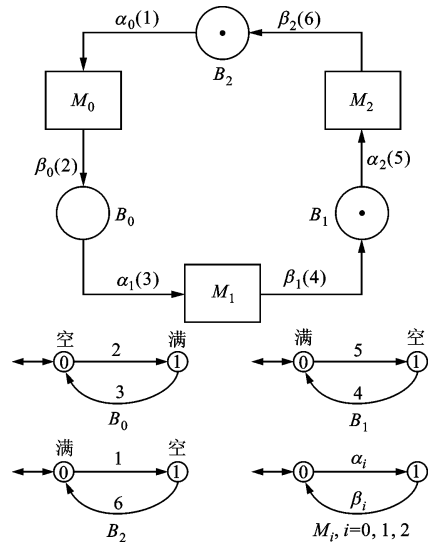
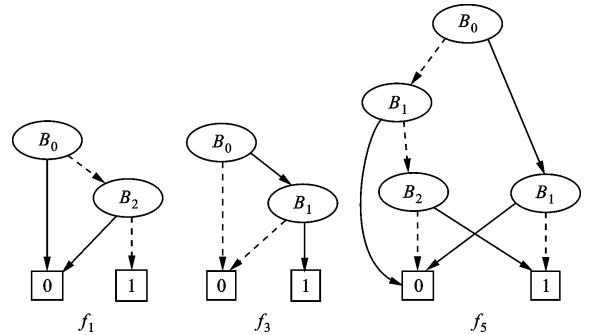


图 6 具有对称性的串行工作系统及其自动机示意图

该系统中, 缓冲区 B_0, B_1, B_2 的容量均为 1, 初始值分别为 0, 1, 1; M_0, M_1, M_2 具有对称性。建立该系统的状态树结构模型可得图 7 所示的可控事件 1, 3, 5 的控制函数 f_1, f_3, f_5 。由于 f_1, f_3, f_5 只与缓冲区 B_0, B_1, B_2 的状态 (空或者满) 有关, 因此系统中可控事件的使能只取决于缓冲区状态 (空或者满), 而与各组件状态无关; 由观察可知 f_1, f_3, f_5 满足谓词 $P: \alpha_i, i \in \{0, 1, 2\}$ 被使能当且仅当缓冲区 B_i 空、 $B_{i \oplus 2}$ 满, 其中 $i \oplus 2 = (i+2) \bmod 3$ 。

图 7 控制函数 f_1, f_3, f_5 示意图

命题 2 f_1, f_3, f_5 满足谓词 P 。

由命题 2 可知, 可利用谓词 P 对各组件的运行进行监督控制, 并且对于不同的缓冲区初始状态, 谓词 P 均能防止各缓冲区出现上溢与下溢, 因此具有不变性。

4 结 论

本文利用离散事件系统的对称性选用事件重标记映射提取各组组件的状态信息, 然后基于状态信息构建状态树结构从而计算系统中可控事件对应的

控制函数,并结合组件并行与串行工作的实例证明控制函数的不变性。由于控制函数具有不变性,在实际应用中可在组件增减时免于重复计算控制函数,且各组中重标记为同一事件的各可控事件可由同一控制函数进行使能,有效减小了控制函数的个数。相比于自动机形式的监督控制器,控制函数所需状态数更少。

本文所得的控制函数不变性建立在防止缓冲区出现上溢与下溢的性能指标之上,由于组件个体的细节与该性能指标无关,因此各组组件之间的交互不会影响到相应控制函数的选择,只要组件当前处于可控事件有定义的状态且控制函数允许该可控事件对应的重标记事件发生,就允许组件执行该可控事件。

参考文献:

- [1] WONHAM W M. Supervisory control of discrete-event systems [EB/OL]. [2016-03-01]. <http://www.control.utoronto.ca/~wonham/>.
- [2] CASSANDRAS C G, LAFORTUNE S. Introduction to discrete event systems [M]. Berlin, Germany: Springer, 2008: 268-369.
- [3] EYZELL J M, CURY J E R. Exploiting symmetry in the synthesis of supervisors for discrete event systems [J]. IEEE Transactions on Automatic Control, 2001, 46(9): 1500-1505.
- [4] ROHLOFF K, LAFORTUNE S. The verification and control of interacting similar discrete event systems [J]. SIAM Journal on Control and Optimization, 2006, 45(2): 634-667.
- [5] WANG W, SU R, LIN L. On analysis of deadlock and blocking freeness in isomorphic module systems [C]// American Control Conference. Piscataway, NJ, USA: IEEE, 2013: 923-928.
- [6] SU R. Discrete-event modeling of multi-agent systems with broadcasting-based parallel composition [J]. Automatica, 2013, 49(11): 3502-3506.
- [7] BRYANT R. Graph-based algorithms for Boolean function manipulation [J]. IEEE Transactions on

Computers, 1986, 35(8): 677-691.

- [8] MA C, WONHAM W M. Nonblocking supervisory control of state tree structures [M]. Berlin, Germany: Springer, 2005: 11-125.
- [9] MA C, WONHAM W M. Nonblocking supervisory control of state tree structures [J]. IEEE Transactions on Automatic Control, 2006, 51(5): 782-793.
- [10] 晁武杰, 甘永梅, 王兆安, 等. 实时状态树结构模型的最优非阻塞模块化监督控制研究 [J]. 西安交通大学学报, 2013, 47(4): 86-91.
CHAO Wujie, GAN Yongmei, WANG Zhao'an, et al. An optimal nonblocking modular supervisory control to real-time state tree structures [J]. Journal of Xi'an Jiaotong University, 2013, 47(4): 86-91.
- [11] JIAO T, GAN Y, YANG X, et al. Exploiting symmetry of discrete-event systems with parallel components by relabeling [C]// TENCON 2015-2015 IEEE Region 10 Conference. Piscataway, NJ, USA: IEEE, 2015: 1-4.
- [12] BHERER H, DESHARNAIS J, ST-DENIS R. Control of parameterized discrete event systems [J]. Discrete Event Dynamic Systems, 2009, 19(2): 213-265.
- [13] GRIGOROV L, BUTLER B E, CURY J E R, et al. Conceptual design of discrete-event systems using templates [J]. Discrete Event Dynamic Systems, 2011, 21(2): 257-303.
- [14] GRIGOROV L, CURY J E R, RUDIE K. Design of discrete-event systems using templates [C]// Proceedings of the American Control Conference. Piscataway, NJ, USA: IEEE, 2008: 499-504.
- [15] RABIN M O, SCOTT D. Finite automata and their decision problems [J]. IBM Journal of Research and Development, 1959, 3(2): 114-125.
- [16] JIAO T, GAN Y, XIAO G, et al. Exploiting symmetry of state tree structures for discrete-event systems with parallel components [C]// 13th International Workshop on Discrete Event Systems. Piscataway, NJ, USA: IEEE, 2016: 97-102.

(编辑 赵炜)