

基于服务组合模型结构特征的性能瓶颈定位方案

沈 华¹ 何炎祥² 张明武¹

(湖北工业大学计算机学院 武汉 430068)¹ (武汉大学计算机学院 武汉 430072)²

摘 要 在满足功能需求的前提下,组合服务的性能是赢得用户的关键,如何发现和消除性能瓶颈则是服务组合面临的重大挑战。面对这个挑战,提出一种基于服务组合模型结构特征的性能瓶颈分析方案。为了确保方案的可行性和有效性,除了对方案技术路线的可行性进行论证,还指出了实现方案需要解决的关键基本问题,即找出服务组合模型的最小结构完备集。提出一种基于随机 Petri 网的 Web 服务组合性能分析模型,并给出此模型最小结构完备集的求解和证明过程。最后,通过一个应用实例说明了方案的有效性。

关键词 服务组合,性能评价,性能瓶颈,随机 Petri 网

中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.022

Performance Bottlenecks Location Scheme Based on Structural Features of Service Composition Model

SHEN Hua¹ HE Yan-xiang² ZHANG Ming-wu¹

(School of Computer Science, Hubei University of Technology, Wuhan 430068, China)¹

(School of Computer, Wuhan University, Wuhan 430072, China)²

Abstract The key of achieving the preference of customers is the performance of Web services composition in terms of satisfying capability requirements. How to identify and eliminate the performance bottlenecks of Web services composition is still a challenging research issue. Faced with this challenge, this paper proposed a performance bottleneck analysis scheme based on structural features of the services composition model. In order to ensure the viability and effectiveness of the scheme, the viability of the scheme's technical route was proved. And the key fundamental problem needed to be solved is to find the smallest structural complete set of services composition model. Based on stochastic Petri net, this paper proposed a performance analysis model of Web services composition and introduced the methods for solving and proving the smallest structural complete set. Finally, the effectiveness of the scheme was illustrated by an application example of the scheme.

Keywords Services composition, Performance evaluation, Performance bottlenecks, Stochastic Petri net

20 世纪 90 年代末出现的 Web 服务是由全球资源定位器 (Uniform Resource Locator, URL) 识别的软件系统,其公共接口和绑定信息用可扩展标记语言 (Extensive Markup Language, XML) 进行定义和描述^[1]。Web 服务通过 Internet 部署和使用^[2],Web 服务之间通过交换消息进行交互,Web 服务的价值在于服务重用。自 2007 年底 IBM 公司宣布云计算^[3-7]计划以来,云计算就成为 IT 产业界和学术界研究的热点。文献[8]在研究云制造^[9,10]的过程中提出通过对云制造服务平台中已有的适合的云服务进行重用、组合、验证,构建能随大众需求而即时应变的松耦合制造云服务应用。可见,不论是对传统的 Web 服务还是对正兴起的云服务而言,服务组合都是提高服务重用度及减少计算成本实现服务增值的主要途径。能够提供保证服务质量的组合服务是服务组合走向实用的必要条件。所以,为了确保服务组合的正确性并确保它们能正常执行以满足预期目标,需要对服务组合进行验证和性能分析。

进行性能分析的一类形式化方法是基于建模的性能评价。根据分析性能模型的不同方法,性能分析又细分为基于数学方法的性能分析和基于模拟仿真的性能分析^[11],其中自动机理论^[12]、Pi 演算^[13]、进程代数^[14,15]和 Petri 网^[15-18]等是使用较多的几种形式化方法。由于 Petri 网具有异步性和并发性,适用于分布式软件系统,因此主要讨论基于 Petri 网的性能建模和分析。文献[19]利用着色 Petri 网 (Colored Petri Net, CPN) 对 Web 服务组合进行建模,给出了基于 CPN 的 Web 服务及服务运算定义,分析了运算的代数性质及运行性质,并给出了服务组合的建立与执行算法。文献[20]应用一阶逻辑语言得到 Web 服务的基于 Petri 网的形式化描述,并为 Web 服务的模拟、验证和组合提供决策过程。文献[21]提出了一种面向性能分析的 Web 服务组合平台 sPAC (web services Performance Analysis Center)。用户利用这个平台可以图形化设计一个新的 Web 服务组合或者对 Web 服务组合的性质进行分析。sPAC 是采用测试和模拟相结合的方法

到稿日期:2014-05-18 返修日期:2014-07-11 本文受国家自然科学基金项目(61272404, 61370224),湖北省自然科学基金重点项目(2013CFA046)资助。

沈 华(1978—),女,博士,讲师,CCF 会员,主要研究方向为服务计算、信息安全, E-mail: nancy78733@126.com; 何炎祥(1952—),男,博士,博士生导师,CCF 高级会员,主要研究方向为服务计算、可信计算; 张明武(1971—),男,博士,CCF 会员,主要研究方向为信息安全。

来进行性能分析的。基于测试的分析方法主要是在低负载的条件下进行,而基于模拟的分析方法主要是在高负载的条件下进行。为了提高分析的精度,文献[22]将模拟的分析建立在测试分析的结果之上。基于测试的分析通过响应时间和吞吐量记录了在低负载条件下 Web 服务组合的行为特征,sPAC 利用这些表示特征的数据来设置模拟的输入参数。

与上述基于 Petri 网的服务组合性能分析方法不同的是,本文从基于随机 Petri 网的服务组合性能分析模型的结构入手,找出结构和性能瓶颈这个整体涌现性(整体涌现性是指结构按照某种方式整合为系统时所呈现出来的特性)之间的关系,从而实现对服务组合性能瓶颈的预测。第 1 节在给出基于随机 Petri 网的服务组合性能模型的基础上,提出一种性能瓶颈定位方案;第 2 节主要围绕方案的可行性展开论证;第 3 节关注的是实施方案需要解决的基础性问题;第 4 节给出方案针对 Web 服务组合性能瓶颈定位问题的应用;最后总结本文工作及展望未来工作。

1 方案描述

1.1 建模工具及模型简述

本文采用连续时间随机 Petri 网(Stochastic Petri Net, SPN)作为建模工具。

定义 1(连续时间 SPN)^[23] 一个四元组 $N=(S, T; F; M_0)$, 其中 S 为库所集, T 为变迁集, N 是一个连续时间 SPN 当且仅当满足: (1) $S \cup T \neq \emptyset$; (2) $S \cap T = \emptyset$; (3) $F \subseteq (S \times T) \cup (T \times S)$; (4) M_0 为初始标识; (5) $\text{dom}(F) \cup \text{cod}(F) = S \cup T$; (6) $\forall t \in T$ 的可实施时刻到实施时刻之间的时间间隔被看成是一个连续随机变量 x_t (取正实数值), 且服从分布函数 $F_t(x) = P(x_t \leq x) = 1 - e^{-\lambda_t x}$, 其中实参数 $\lambda_t > 0$ 是变迁 t 的平均实施速率, 变量 $x \geq 0$ 。

定义 2(变迁的前置集和后置集)^[22] 对于 $\forall t \in T, t$ 的前置集记为 $\cdot t$, 其定义为: $\cdot t = \{s | (s, t) \in F\}$; t 的后置集记为 $t \cdot$, 其定义为: $t \cdot = \{s | (t, s) \in F\}$ 。

定义 3(库所的前置集和后置集)^[22] 对于 $\forall s \in S, s$ 的前置集记为 $\cdot s$, 其定义为: $\cdot s = \{t | (t, s) \in F\}$; s 的后置集记为 $s \cdot$, 其定义为: $s \cdot = \{t | (s, t) \in F\}$ 。

定义 4(起始库所和终止库所) 如果 $\exists s \in S(|\cdot s| = 0)$, 则库所 s 是一个起始库所; 如果 $\exists s \in S(|s \cdot| = 0)$, 则库所 s 是一个终止库所。

基于 SPN 提出一种 Web 服务组合性能分析模型(Web Services Composition Performance Analysis Model, WSCPAM)^[23]。在 WSCPAM 中, Web 服务或子 Web 服务组合用变迁表示, 变迁的前置库所(也称为输入库所)存放对该变迁的请求队列, 变迁的后置库所(也称为输出库所)存放的是该变迁对某个后置变迁的服务请求, 且请求队列的长度不限。因此, 一个库所对应一个变迁序偶对。此外, 在 Web 服务组合逻辑中不存在服务的自我请求, 故在 WSCPAM 中任一库所和任一变迁之间最多只存在一条弧。上述两条约束条件的形式化描述在定义 5 中给出。

定义 5(WSCPAM 模型)^[23] WSCPAM 是一个同时满足以下两个约束条件的 SPN。

约束条件 1: $\forall s \in S(|\cdot s| \leq 1 \wedge |s \cdot| \leq 1)$; 约束条件 2: $(\forall s \in S(\cdot s \cap s \cdot = \emptyset)) \wedge (\forall t \in T(\cdot t \cap t \cdot = \emptyset))$ 。

1.2 一些相关概念

在阐述方案之前先给出一些相关概念的定义。

定义 6(最小结构完备集) 假设结构集合 S 是构成网 Σ 所有可能结构类型的集合, 符号 $\odot$ 表示某种结构类型, $\odot \angle S$ 表示结构类型 $\odot$ 可以由集合 S 中的结构类型组合得到。如果命题 $\exists S_1, S_2 \subseteq S \ \&\& \ S_1 \cap S_2 = \emptyset \ \&\& \ S_1 \cup S_2 = S: \forall \odot \in S_2 (\odot \angle S_1) \ \&\& \rightarrow \exists \odot \in S_1 (\odot \angle S_1)$ 成立, 那么 S_1 称为 Σ 的最小结构完备集。

上述定义的另一表述方式是: 将 S 中的所有可能的结构类型划分为两个集合, 假设分别为 S_1 和 S_2 , 如果 S_1 和 S_2 满足 S_2 中的所有结构类型均可以由 S_1 中的结构类型组合得到, 并且 S_1 中不存在可以由 S_1 中的其他结构类型组合得到的结构, 那么 S_1 即为 S 的最小结构完备集。

定义 7(原子结构) 将最小结构完备集中的结构称为原子结构。

1.3 方案描述

根据上述最小完备集的概念, 提出如下瓶颈定位方案: 从 WSCPAM 的两类顶点(库所和变迁)之间的二元关系入手, 分析出所有可能的二元关系, 即得到所有可能的结构类型。对得到的结构类型进行结构转换, 当结构转换停止时, 各结构类型中的公共结构类型构成的集合就是方案所需要的最小结构完备集。将最小结构完备集作为结构分析集, 对结构分析集中的结构类型进行性能瓶颈定位分析, 得到一个性能瓶颈定位分析算法库, 然后通过调用算法库中的算法来实现对复杂结构类型的性能瓶颈定位分析。

2 方案可行性论证

通过下面两个结论的证明可知, 1.3 节中所阐述的技术方案是可行的。

结论 1 WSCPAM 存在最小结构完备集。

证明: 根据 WSCPAM 模型的定义可知, WSCPAM 中包含两类顶点: 库所和变迁。WSCPAM 中的弧是两类顶点间的二元关系, 二元关系共有 4 种: 一对一、一对多、多对一、多对多, 由此可知 WSCPAM 中的结构类型是有限的。因此, WSCPAM 存在最小结构完备集。

结论 2 通过结构分析集中的各结构的瓶颈定位方法可以分析复杂结构中可能存在的性能瓶颈。

证明: 运用分治的思想进行证明。因为结构分析集中的结构类型足够简单, 所以直接对它们进行性能分析是可行的。其他结构均可以通过完备集中的原子结构经过一次或多次复合后得到。换句话说, 任何复杂的结构都可以看作是具有“大顶点”的原子结构, “大顶点”又可以看成是“粒度小一些的顶点”构成的原子结构, …… , 持续这样的“划分过程”, 最终会得到粒度最小的顶点构成的原子结构(即最小完备集中的某种原子结构)。通过对粒度最小顶点构成的原子结构的瓶颈定位预测可以预测粒度大一些的顶点构成的原子结构可能存在的瓶颈, …… , 持续这个“合并过程”, 最终会得到“大顶点”原子结构的瓶颈预测结果。

3 需要解决的关键基础性问题

对 WSCPAM 实施上述方案需要解决的关键基础性问题: 找出 WSCPAM 的最小结构集, 并证明它是完备的。

3.1 WSCPAM 的结构类型分析

3.1.1 WSCPAM 的基本结构集

根据 1.1 节中的定义 1 可知:在 SPN 中存在两类顶点(库所,变迁),两类弧(库所→变迁,变迁→库所)。因此,从图论的角度可以把 SPN 重新定义为:

定义 8(基于图论的 SPN) $SPN=(V, E), V=S \cup T, E=E_S \cup E_T$, 其中 S 为库所集, T 为变迁集。

定义 9(库所流集和变迁流集) 用符号 E_S 表示库所流集, 用符号 E_T 表示变迁流集, 它们的定义分别为:

$$E_S = \{ \langle s_i, t_j \rangle \mid s_i \in S, t_j \in T \}$$

$$E_T = \{ \langle t_j, s_k \rangle \mid t_j \in T, s_k \in S \}$$

定义 10(基本结构) 基本结构是指库所与变迁之间的直接关系, 即库所与变迁之间的一步关系。换句话说, 基本结构描述了库所和变迁之间的一种原子关系。用符号 $SPN|BS$ 表示 SPN 的基本结构集, 用符号 $WSCPAM|BS$ 表示 WSCPAM 的基本结构集。

用符号 s^* 表示库所集, $s^* \subseteq S, |s^*| \geq 1$; t^* 表示变迁集, 且 $t^* \subseteq T, |t^*| \geq 1$ 。由 SPN 的定义可知, 在 SPN 中库所与变迁之间的直接关系只可能有以下 8 种:

$$\begin{aligned} \rho_1 &= E_S : R_{1,1} \\ &= \{ \langle s_i, t_j \rangle \mid s_i \in S, t_j \in T, \langle s_i, t_j \rangle \in E_S \} \\ \rho_2 &= E_T : R_{1,1} \\ &= \{ \langle t_j, s_i \rangle \mid s_i \in S, t_j \in T, \langle t_j, s_i \rangle \in E_T \} \\ \rho_3 &= E_S : R_{n,1} \\ &= \{ \langle s^*, t_j \rangle \mid s^* \subseteq S, t_j \in T, \forall s_i \in s^* (\langle s_i, t_j \rangle \in E_S) \} \\ \rho_4 &= E_T : R_{1,n} \\ &= \{ \langle t_j, s^* \rangle \mid s^* \subseteq S, t_j \in T, \forall s_i \in s^* (\langle t_j, s_i \rangle \in E_T) \} \\ \rho_5 &= E_T : R_{n,1} \\ &= \{ \langle t^*, s_i \rangle \mid t^* \subseteq T, s_i \in S, \forall t_j \in t^* (\langle t_j, s_i \rangle \in E_T) \} \\ \rho_6 &= E_T : R_{m,n} \\ &= \{ \langle t^*, s^* \rangle \mid t^* \subseteq T, s^* \subseteq S, s_i \in S, \forall t_j \in t^*, \forall s_i \in s^* (\langle t_j, s_i \rangle \in E_T) \} \\ \rho_7 &= E_S : R_{1,n} = \{ \langle s_i, t^* \rangle \mid t^* \subseteq T, s_i \in S, \forall t_j \in t^* (\langle s_i, t_j \rangle \in E_S) \} \\ \rho_8 &= E_S : R_{m,n} = \{ \langle s^*, t^* \rangle \mid s^* \subseteq S, t^* \subseteq T, \forall s_i \in s^*, \forall t_j \in t^* (\langle s_i, t_j \rangle \in E_S) \} \end{aligned}$$

以上 8 种直接关系如图 1 所示。

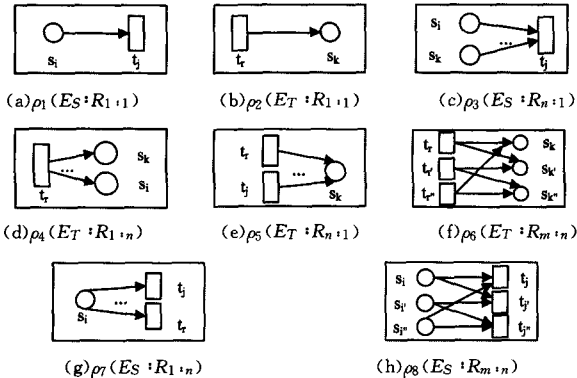


图 1 顶点集 E_S 与顶点集 E_T 之间的 8 种直接关系

根据定义 10 可知:

$$\begin{aligned} SPN|BS &= \{ \rho_1(E_S : R_{1,1}), \rho_2(E_T : R_{1,1}), \rho_3(E_S : R_{n,1}), \\ &\quad \rho_4(E_T : R_{1,n}), \rho_5(E_T : R_{n,1}), \rho_6(E_T : R_{m,n}), \\ &\quad \rho_7(E_S : R_{1,n}), \rho_8(E_S : R_{m,n}) \} \end{aligned}$$

由定义 1 和定义 5 可知, $WSCPAM \subseteq SPN \Rightarrow WSCPAM|BS \subseteq SPN|BS$, 因此可先令 $WSCPAM|BS = SPN|BS$ 。根据定义 5, WSCPAM 必须满足结构约束 1, 所以在 $WSCPAM|BS$ 中不可能存在基本结构 ρ_7 和 ρ_8 。故

$$\begin{aligned} WSCPAM|BS &= \{ \rho_1(E_S : R_{1,1}), \rho_2(E_T : R_{1,1}), \rho_3(E_S : \\ &\quad R_{n,1}), \rho_4(E_T : R_{1,n}), \rho_5(E_T : R_{n,1}), \rho_6 \\ &\quad (E_T : R_{m,n}) \} \end{aligned}$$

根据 $WSCPAM|BS$ 中的 6 种基本关系的外部结构表现特征, 抽象得到由入口线型和出口线型共同表征的 6 种结点型。

定义 11(线型) 用二元组 $(type, num)$ 表示线型, 其中 $type$ 表示线的类型, 其取值要么是 s 要么是 t ; num 表示线的数量, 它是一个非负整数。

定义 12(6 种抽象结点型 Abstract Node Type, ANT)

抽象结点型 I(ANT I): 具有一个 s 入口线和一个 t 出口线的结点类型; 抽象结点型 II(ANT II): 具有一个 t 入口线和一个 s 出口线的结点类型; 抽象结点型 III(ANT III): 具有多个 s 入口线和一个 t 出口线的结点类型; 抽象结点型 IV(ANT IV): 具有一个 t 入口线和多个 s 出口线的结点类型; 抽象结点型 V(ANT V): 具有多个 t 入口线和一个 s 出口线的结点类型; 抽象结点型 VI(ANT VI): 具有多个 t 入口线和多个 s 出口线的结点类型。

显然, 基本关系 ρ_1 属于 ANT I, 基本关系 ρ_2 属于 ANT II, 基本关系 ρ_3 属于 ANT III, 基本关系 ρ_4 属于 ANT IV, 基本关系 ρ_5 属于 ANT V, 基本关系 ρ_6 属于 ANT VI。由定义 12 可知, ANT 是通过结点的入口线型和出口线型进行区分的, 因此, ANT 也可以表示成元素为线型的二元组: $(in(type, num), out(type, num))$ 。定义 12 中的 6 种抽象结点型可以分别表示为: ANT I($in(s, 1), out(t, 1)$), ANT II($in(t, 1), out(s, 1)$), ANT III($in(s, m), out(t, 1)$), ANT IV($in(t, 1), out(s, n)$), ANT V($in(t, m), out(s, 1)$), ANT VI($in(t, m), out(s, n)$), 也可以分别简记为: $(1s, 1t)$ 、 $(1t, 1s)$ 、 $(ms, 1t)$ 、 $(1t, ns)$ 、 $(mt, 1s)$ 、 (mt, ns) 。

下面说明上述 6 种抽象结点型分别代表了 6 种(结构)等价关系。假设用符号 $\rho_i R_k \rho_j$ 表示关系 ρ_i 和关系 ρ_j 同属于抽象结点型 k ($k=I, II, III, IV, V, VI$), 显然有: (1) 对所有的 ρ_i 有 $\rho_i R_k \rho_i$, 因此关系 R_k 是自反的; (2) 对所有的 ρ_i 和 ρ_j , 如果有 $\rho_i R_k \rho_j$, 那么 $\rho_j R_k \rho_i$ 一定成立, 因此关系 R_k 是对称的; (3) 对所有的 ρ_i, ρ_j 和 ρ_r , 如果有 $\rho_i R_k \rho_j$ 和 $\rho_j R_k \rho_r$, 那么 $\rho_i R_k \rho_r$ 一定成立, 因此关系 R_k 是传递的。故关系 R_k 是等价关系, 因此可以分别用符号 $[\rho_1]_{RI}, [\rho_2]_{RII}, [\rho_3]_{RIII}, [\rho_4]_{RIV}, [\rho_5]_{RV}$ 和 $[\rho_6]_{RVI}$ 表示基本关系 $\rho_1, \rho_2, \rho_3, \rho_4, \rho_5$ 和 ρ_6 的等价类。

定义 13(S 类结点型和 T 类结点型) 根据结点入口线的类型将上述 6 种抽象结点型又分为两类: S 类结点型和 T 类结点型。S 类结点型是指入口线的类型为 s 的抽象结点型; T 类结点型是指入口线的类型为 t 的抽象结点型。

显然, ANT I 和 ANT III 属于 S 类结点型; ANT II, ANT IV, ANT V 和 ANT VI 属于 T 类结点型。

3.1.2 WSCPAM 的复合结构

定义 14(复合结构或复合关系) 设 ρ_i 是一个从库所集 S 到变迁集 T 的关系, ρ_j 是一个从库所集 T 到变迁集 S 的关系, 则 ρ_i 和 ρ_j 的复合关系是一个由库所集 S 到库所集 S 的关

系,用 $\rho_i \cdot \rho_j$ 表示(或简记为 $\rho_i \rho_j$); ρ_j 和 ρ_i 的复合关系是一个由变迁集 T 到变迁集 T 的关系,用 $\rho_j \cdot \rho_i$ 表示(或简记作 $\rho_j \rho_i$)。这种从 ρ_i 和 ρ_j 得到 $\rho_i \cdot \rho_j$ 或 $\rho_j \cdot \rho_i$ 的运算称为关系的复合运算或结构的复合运算。

为了保证复合结构的入口线的类型和出口线的类型相异,即复合结点入口线和出口线不能同时为 s 类型或同时为 t 类型,在结构复合时必须将定义 14 中定义的复合运算执行偶数次。

WSCPAM|BS 中的基本关系可以用以下关系矩阵来表示:

$$A = \begin{matrix} & s^* & t^* \\ \begin{matrix} s^* \\ t^* \end{matrix} & \begin{bmatrix} 0 & MS_1 \\ MS_2 & 0 \end{bmatrix} \end{matrix}$$

其中 $MS_1 = B_1 \cup B_3$, 且 B_1 和 B_3 不能同时为空; $MS_2 = B_2 \cup B_4 \cup B_5 \cup B_6$, 且 B_2, B_4, B_5 和 B_6 不能同时为空。令

$$\begin{aligned} B_1 &= \{\underbrace{\rho_1, \rho_1, \dots, \rho_1}_{n(\geq 0)}\}, B_2 = \{\underbrace{\rho_2, \rho_2, \dots, \rho_2}_{n(\geq 0)}\} \\ B_3 &= \{\underbrace{\rho_3, \rho_3, \dots, \rho_3}_{n(\geq 0)}\}, B_4 = \{\underbrace{\rho_4, \rho_4, \dots, \rho_4}_{n(\geq 0)}\} \\ B_5 &= \{\underbrace{\rho_5, \rho_5, \dots, \rho_5}_{n(\geq 0)}\}, B_6 = \{\underbrace{\rho_6, \rho_6, \dots, \rho_6}_{n(\geq 0)}\} \\ B_1' &= \{\underbrace{\rho_1, \rho_1, \dots, \rho_1}_{n(\geq 2)}\}, B_2' = \{\underbrace{\rho_2, \rho_2, \dots, \rho_2}_{n(\geq 2)}\} \\ B_3' &= \{\underbrace{\rho_3, \rho_3, \dots, \rho_3}_{n(\geq 2)}\}, B_4' = \{\underbrace{\rho_4, \rho_4, \dots, \rho_4}_{n(\geq 2)}\} \\ B_5' &= \{\underbrace{\rho_5, \rho_5, \dots, \rho_5}_{n(\geq 2)}\}, B_6' = \{\underbrace{\rho_6, \rho_6, \dots, \rho_6}_{n(\geq 2)}\} \end{aligned}$$

符号 $\{\rho\}$ 表示包含一个关系 ρ 的集合; 符号 $\{\rho\}^*$ 表示包含两个及两个以上关系 ρ 的集合, 因此可以用 $\{\rho_1\}^*, \{\rho_2\}^*, \{\rho_3\}^*, \{\rho_4\}^*, \{\rho_5\}^*$ 和 $\{\rho_6\}^*$ 分别表示 $B_1', B_2', B_3', B_4', B_5'$ 和 B_6' ; $\{\rho_i, \rho_j\}$ 表示同时包含一个关系 ρ_i 和一个关系 ρ_j 的集合; $\{\rho_i, \rho_j\}^*$ 表示同时包含关系 ρ_i 和关系 ρ_j 的集合, 且集合中至少有一种关系的个数大于 1; $\{\rho_{i1}, \rho_{i2}, \dots, \rho_{in}\}$ 表示同时包含 n 种关系 ρ_{ij} ($j=1, 2, \dots, n, n \geq 2$) 且每种关系只有一个的集合; $\{\rho_{i1}, \rho_{i2}, \dots, \rho_{in}\}^*$ 表示同时包含 n 种关系 ρ_{ij} ($j=1, 2, \dots, n, n \geq 2$) 且其中至少有一种关系的个数大于 1 的集合。上述表达中的关系 ρ, ρ_i, ρ_j 和 ρ_{ij} 表示任意一种基本关系或任意一种复合关系。根据上述符号说明, 可以得到如下表达式:

$$\begin{aligned} B_1 &= \emptyset \cup \{\rho_1\} \cup B_1', B_2 = \emptyset \cup \{\rho_2\} \cup B_2' \\ B_3 &= \emptyset \cup \{\rho_3\} \cup B_3', B_4 = \emptyset \cup \{\rho_4\} \cup B_4' \\ B_5 &= \emptyset \cup \{\rho_5\} \cup B_5', B_6 = \emptyset \cup \{\rho_6\} \cup B_6' \end{aligned}$$

符号 $\{\rho\}^+$ 表示如下集合: $\{\rho\}^+ = \{\{\rho\}, \{\rho\}^*\}$, 其中 ρ 表示任意一种基本关系或任意一种复合关系。类似地, 符号 $\{\rho_i, \rho_j\}^+$ 表示如下集合: $\{\rho_i, \rho_j\}^+ = \{\{\rho_i, \rho_j\}, \{\rho_i, \rho_j\}^*\}$, 其中 ρ_i, ρ_j 分别表示任意一种基本关系或任意一种复合关系。

WSCPAM 的结构集用符号 $Struct(WSCPAM)$ 表示, 其定义如下:

定义 15(WSCPAM 的结构集) $\rho \in Struct(WSCPAM)$, 当且仅当 ρ 能通过有限次使用以下的(1)–(3)生成:

(1) $WSCPAM|BS \subseteq Struct(WSCPAM)$; (2) 如果 $\rho_i, \rho_j, \rho_k \in Struct(WSCPAM)$, 其中 ρ_i 和 ρ_k 对应的抽象结点型是 S 类结点型, ρ_j 对应的抽象结点型是 T 类结点型, 且 $\rho_i' \in \{\rho_i\}^+, \rho_j' \in \{\rho_j\}^+, \rho_k' \in \{\rho_k\}^+$, 那么 $\rho_i' \rho_j' \rho_k' \in Struct(WSCPAM)$, 且 $\rho_i' \rho_j' \rho_k'$ 对应的抽象结点型属于 S 类结点型; (3) 如

果 $\rho_i, \rho_j, \rho_k \in Struct(WSCPAM)$, 其中 ρ_i 和 ρ_k 对应的抽象结点型是 T 类结点型, ρ_j 对应的抽象结点型是 S 类结点型, 且 $\rho_i' \in \{\rho_i\}^+, \rho_j' \in \{\rho_j\}^+, \rho_k' \in \{\rho_k\}^+$, 那么 $\rho_i' \rho_j' \rho_k' \in Struct(WSCPAM)$, 且 $\rho_i' \rho_j' \rho_k'$ 对应的抽象结点型属于 T 类结点型。

定义 16(WSCPAM 的 k 级复合结构) (1)规定 WSCPAM 的基本结构集 $WSCPAM|BS$ 是 0 级复合结构, 用 $Struct^0(WSCPAM)$ 表示; (2) k 级复合结构是指通过下述构造过程得到的复合结构: 假设

$$Struct^{k-1} = \bigcup_{i=0}^{k-1} Struct^i(WSCPAM)'$$

并假设 $\rho_i, \rho_j, \rho_k \in Struct^{k-1}$, 且 ρ_i 和 ρ_k 对应的抽象结点型是 T 类结点型(或者是 S 类结点型), ρ_j 对应的抽象结点型是 S 类结点型(或者是 T 类结点型)。如果 $\rho_i' \in \{\rho_i\}^+, \rho_j' \in \{\rho_j\}^+, \rho_k' \in \{\rho_k\}^+$, 并且 ρ_i' 的出口线型与 ρ_j' 的入口线型匹配, 同时 ρ_j' 的出口线型与 ρ_k' 的入口线型匹配, 那么 $\rho_i' \rho_j' \rho_k' \in Struct^k(WSCPAM)$ 。

WSCPAM 的 1 级复合结构可以通过上述关系矩阵相乘得到:

$$\begin{aligned} A \cdot A \cdot A &= \begin{matrix} & s^* & t^* \\ \begin{matrix} s^* \\ t^* \end{matrix} & \begin{bmatrix} 0 & MS_1 \\ MS_2 & 0 \end{bmatrix} \end{matrix} \cdot \begin{matrix} & s^* & t^* \\ \begin{matrix} s^* \\ t^* \end{matrix} & \begin{bmatrix} 0 & MS_1 \\ MS_2 & 0 \end{bmatrix} \end{matrix} \cdot \begin{matrix} & s^* & t^* \\ \begin{matrix} s^* \\ t^* \end{matrix} & \begin{bmatrix} 0 & MS_1 \\ MS_2 & 0 \end{bmatrix} \end{matrix} \\ &= \begin{matrix} & s^* & t^* \\ \begin{matrix} s^* \\ t^* \end{matrix} & \begin{bmatrix} MS_1 MS_2 & 0 \\ 0 & MS_2 MS_1 \end{bmatrix} \end{matrix} \cdot \begin{matrix} & s^* & t^* \\ \begin{matrix} s^* \\ t^* \end{matrix} & \begin{bmatrix} 0 & MS_1 \\ MS_2 & 0 \end{bmatrix} \end{matrix} \\ &= \begin{matrix} & s^* & t^* \\ \begin{matrix} s^* \\ t^* \end{matrix} & \begin{bmatrix} 0 & MS_1 MS_2 MS_1 \\ MS_2 MS_1 MS_2 & 0 \end{bmatrix} \end{matrix} \end{aligned}$$

1. 首先讨论 MS_1 和 MS_2 的所有可能取值。

(1) 列出 MS_1 的 6 种可能取值, 并标识出每种取值的入口出口线型。

(i) $B_1 \neq \emptyset, B_3 = \emptyset$

① $MS_1 = \{\rho_1\} (1s, 1t)$, ② $MS_1 = \{\rho_1\}^* (ms, nt)$;

(ii) $B_1 = \emptyset, B_3 \neq \emptyset$

③ $MS_1 = \{\rho_3\} (ms, 1t)$, ④ $MS_1 = \{\rho_3\}^* (ms, nt)$;

(iii) $B_1 \neq \emptyset, B_3 \neq \emptyset$

⑤ $MS_1 = \{\rho_1, \rho_3\} (ms, nt)$, ⑥ $MS_1 = \{\rho_1, \rho_3\}^* (ms, nt)$ 。

按值的入口出口线型可以得到 MS_1 值域的一个划分, 用 γ 表示“同入口出口线型的关系”, 可用符号 $\pi_\gamma^{MS_1}$ 表示在 MS_1 值域上由 γ 所导出的等价划分。显然有: $\pi_\gamma^{MS_1} = [\{\rho_1\}]_\gamma, [\{\rho_3\}]_\gamma, [\{\rho_1\}^*]_\gamma$ 。其中, $[\{\rho_1\}]_\gamma^{MS_1} = \{\textcircled{1}\}_{(1s, 1t)}, [\{\rho_3\}]_\gamma^{MS_1} = \{\textcircled{3}\}_{(ms, 1t)}, [\{\rho_1\}^*]_\gamma^{MS_1} = \{\textcircled{2}, \textcircled{4}, \textcircled{5}, \textcircled{6}\}_{(ms, nt)}$ 。

(2) 列出 MS_2 的 30 种可能取值。

(i) $B_1 \neq \emptyset (i=2, 4, 5, 6); \forall j \in \{2, 4, 5, 6\} - \{i\}, B_j = \emptyset$

① $MS_2 = \{\rho_2\} (1t, 1s)$, ② $MS_2 = \{\rho_2\}^* (mt, ns)$,

③ $MS_2 = \{\rho_4\} (1t, ns)$, ④ $MS_2 = \{\rho_4\}^* (mt, ns)$,

⑤ $MS_2 = \{\rho_5\} (mt, 1s)$, ⑥ $MS_2 = \{\rho_5\}^* (mt, ns)$,

⑦ $MS_2 = \{\rho_6\} (mt, ns)$, ⑧ $MS_2 = \{\rho_6\}^* (mt, ns)$ 。

(ii) $B_1 \neq \emptyset \wedge B_j \neq \emptyset (j=2, 4, 5, 6 \ \& \ i \neq j); \forall k \in \{2,$

4,5,6}-\{i,j\}, B_k=\emptyset

$MS_2=\{\rho_i, \rho_j\}(mt, ns), MS_2=\{\rho_i, \rho_j\}^*(mt, ns)$

共 $C_3^4 \times 2 = 6 \times 2 = 12$ 种取值, 对应取值编号为 ⑨-⑯。

(iii) $B_i \neq \emptyset (i=2, 4, 5, 6); \forall k \in \{2, 4, 5, 6\} - \{i\}, B_k = \emptyset$

$MS_2=\{\rho_i, \rho_j, \rho_k\}(mt, ns), MS_2=\{\rho_i, \rho_j, \rho_k\}^*(mt, ns)$

共 $C_3^4 \times 2 = 4 \times 2 = 8$ 种取值, 对应取值编号为 ⑰-⑳。

(iv) $\forall i \in \{2, 4, 5, 6\}, B_i \neq \emptyset$

$MS_2=\{\rho_2, \rho_4, \rho_5, \rho_6\}(mt, ns), MS_2=\{\rho_2, \rho_4, \rho_5, \rho_6\}^*(mt, ns)$

(mt, ns)

共 $C_1^4 \times 2 = 1 \times 2 = 2$ 种取值, 对应取值编号为 ㉑-㉒。

用符号 $\pi_\gamma^{MS_2}$ 表示在 MS_2 值域上由 γ 所导出的等价类分, 则: $\pi_\gamma^{MS_2} = \{[\{\rho_2\}]_\gamma, [\{\rho_4\}]_\gamma, [\{\rho_5\}]_\gamma, [\{\rho_6\}]_\gamma^*\}$ 。其中 $[\{\rho_2\}]_\gamma^{MS_2} = \{①\}_{(1t, 1s)}, [\{\rho_4\}]_\gamma^{MS_2} = \{③\}_{(1t, ns)}, [\{\rho_5\}]_\gamma^{MS_2} = \{⑤\}_{(ms, 1s)}, [\{\rho_6\}]_\gamma^{MS_2} = \{②, ④, ⑥-⑳\}_{(mt, ns)}$ 。

2. 下面讨论 WSCPAM 的 1 级复合结构的构造过程:

(1) $MS_1 MS_2 MS_1$

(i) 第 1 步执行复合运算 $MS_1 \cdot MS_2$:

(1-1) $\forall \rho \in [\{\rho_1\}]_\gamma^{MS_1} \xrightarrow{(1s, 1t)} \forall \rho' \in [\{\rho_2\}]_\gamma^{MS_2} \xrightarrow{(1t, 1s)} (1s, 1s)$ I 型

(1-2) $\forall \rho \in [\{\rho_1\}]_\gamma^{MS_1} \xrightarrow{(1s, 1t)} \forall \rho' \in [\{\rho_4\}]_\gamma^{MS_2} \xrightarrow{(1t, ns)} (1s, ns)$ II 型

(1-3) $\forall \rho \in [\{\rho_3\}]_\gamma^{MS_1} \xrightarrow{(ms, 1t)} \forall \rho' \in [\{\rho_2\}]_\gamma^{MS_2} \xrightarrow{(1t, 1s)} (ms, 1s)$ III 型

(1-4) $\forall \rho \in [\{\rho_3\}]_\gamma^{MS_1} \xrightarrow{(ms, 1t)} \forall \rho' \in [\{\rho_4\}]_\gamma^{MS_2} \xrightarrow{(1t, ns)} (ms, ns)$ IV 型

(1-5) $\forall \rho \in [\{\rho_1\}]_\gamma^{MS_1} \xrightarrow{(ms, nt)} \forall \rho' \in [\{\rho_5\}]_\gamma^{MS_2} \xrightarrow{(mt, 1s)} (ms, 1s)$ III 型

(1-6) $\forall \rho \in [\{\rho_1\}]_\gamma^{MS_1} \xrightarrow{(ms, nt)} \forall \rho' \in [\{\rho_2\}]_\gamma^{MS_2} \xrightarrow{(mt, ns)} (ms, ns)$ IV 型

(ii) 第 2 步执行复合运算 $MS_1 MS_2 \cdot MS_1$:

(2-1) $\rho \in \text{I 型} \xrightarrow{(1s, 1s)} \forall \rho' \in [\{\rho_1\}]_\gamma^{MS_1} \xrightarrow{(1s, 1t)} (1s, 1t)$ ANT I

(2-2) $\rho \in \text{II 型} \xrightarrow{(1s, ns)} \forall \rho' \in [\{\rho_3\}]_\gamma^{MS_1} \xrightarrow{(ms, 1t)} (1s, 1t)$ ANT I

(2-3) $\rho \in \text{II 型} \xrightarrow{(1s, ns)} \forall \rho' \in [\{\rho_1\}]_\gamma^{MS_1} \xrightarrow{(ms, nt)} (1s, nt)$

(2-4) $\rho \in \text{III 型} \xrightarrow{(ms, 1s)} \forall \rho' \in [\{\rho_1\}]_\gamma^{MS_1} \xrightarrow{(1s, 1t)} (ms, 1t)$ ANT III

(2-5) $\rho \in \text{IV 型} \xrightarrow{(ms, ns)} \forall \rho' \in [\{\rho_3\}]_\gamma^{MS_1} \xrightarrow{(ms, 1t)} (ms, 1t)$ ANT III

(2-6) $\rho \in \text{IV 型} \xrightarrow{(ms, ns)} \forall \rho' \in [\{\rho_1\}]_\gamma^{MS_1} \xrightarrow{(ms, nt)} (ms, nt)$

因为 $\{\rho_1\}$ 是属于 $[\{\rho_1\}]_\gamma^{MS_1}$ 的唯一元素, $\{\rho_2\}$ 是属于 $[\{\rho_2\}]_\gamma^{MS_2}$ 的唯一元素, $\{\rho_3\}$ 是属于 $[\{\rho_3\}]_\gamma^{MS_1}$ 的唯一元素, $\{\rho_4\}$ 是属于 $[\{\rho_4\}]_\gamma^{MS_2}$ 的唯一元素, 所以 (2-1) 对应的 1 级复合结构为 $\rho_1 \rho_2 \rho_1$; (2-2) 对应的 1 级复合结构为 $\rho_1 \rho_4 \rho_3$; (2-4) 对应的一个 1 级复合结构为 $\rho_3 \rho_2 \rho_1$; 分别取 $[\{\rho_1\}]_\gamma^{MS_1}$ 中的元素 $\{\rho_1\}^*, \{\rho_3\}^*, \{\rho_1, \rho_3\}, \{\rho_1, \rho_3\}^*$ 进行 (1-5) 的复合运算,

可以得到 (2-4) 对应的 1 级复合结构 $\{\rho_1\}^* \rho_5 \rho_1, \{\rho_3\}^* \rho_5 \rho_1, \{\rho_1, \rho_3\} \rho_5 \rho_1, \{\rho_1, \rho_3\}^* \rho_5 \rho_1$ 。

$\rho_1 \rho_2 \rho_1, \rho_1 \rho_4 \rho_3, \rho_3 \rho_2 \rho_1, \{\rho_1\}^* \rho_5 \rho_1, \{\rho_3\}^* \rho_5 \rho_1, \{\rho_1, \rho_3\} \rho_5 \rho_1, \{\rho_1, \rho_3\}^* \rho_5 \rho_1$ 分别如图 2(a) 一图 2(g) 所示。

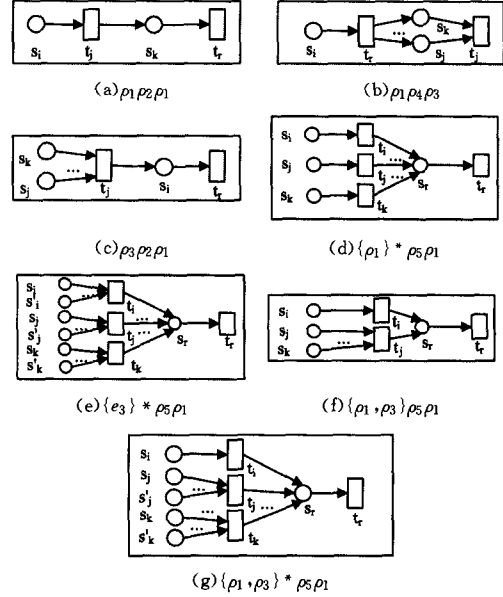


图 2 1 级复合结构(I)

易知 $\rho_1 \rho_2 \rho_1 \in [\rho_1]_{RI}$ 和 $\rho_1 \rho_4 \rho_3 \in [\rho_1]_{RI}$, 因此可以把结构 $\rho_1 \rho_2 \rho_1, \rho_1 \rho_4 \rho_3$ 看作是粗粒度的基本关系 ρ_1 ; $\rho_3 \rho_2 \rho_1 \in [\rho_3]_{RIII}$, $\{\rho_1\}^* \rho_5 \rho_1 \in [\rho_3]_{RIII}, \{\rho_3\}^* \rho_5 \rho_1 \in [\rho_3]_{RIII}, \{\rho_1, \rho_3\} \rho_5 \rho_1 \in [\rho_3]_{RIII}, \{\rho_1, \rho_3\}^* \rho_5 \rho_1 \in [\rho_3]_{RIII}$, 因此可以把 $\rho_3 \rho_2 \rho_1, \{\rho_1\}^* \rho_5 \rho_1, \{\rho_3\}^* \rho_5 \rho_1, \{\rho_1, \rho_3\} \rho_5 \rho_1$ 和 $\{\rho_1, \rho_3\}^* \rho_5 \rho_1$ 看作是粗粒度的基本关系 ρ_3 。利用它们按照 ρ_3 在构造 1 级复合结构过程中的构造规则就可以构造出更为复杂的结构。

(2-3) 对应的 1 级复合结构具有不同于定义 12 给出的 6 种抽象结点型的外部特征, 如分别取 $[\{\rho_1\}]_\gamma^{MS_1}$ 中的元素 $\{\rho_1\}^*, \{\rho_3\}^*, \{\rho_1, \rho_3\}, \{\rho_1, \rho_3\}^*$ 进行 (2-3) 对应的复合运算, 得到的 1 级复合结构有: $\rho_1 \rho_4 \{\rho_1\}^*, \rho_1 \rho_4 \{\rho_3\}^*, \rho_1 \rho_4 \{\rho_1, \rho_3\}, \rho_1 \rho_4 \{\rho_1, \rho_3\}^*$, 如图 3(a) 一图 3(d) 所示。

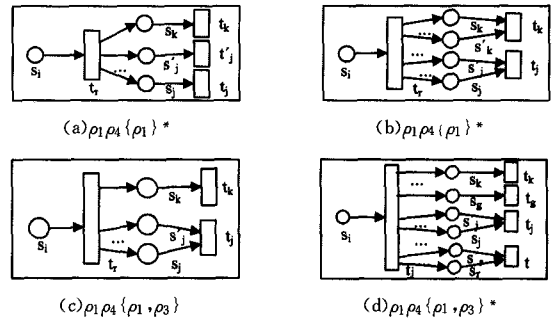


图 3 1 级复合结构(II)

将这个新的外部结构特征也抽象为一种结点型。

定义 17 (第 7 种抽象结点型) 抽象结点型 VII 是具有一 s 入口线和多个 t 出口线的结点类型, 可以表示为 ANT VII(in(s, 1), out(t, n))。

定义 17 作为定义 12 的补充。类似地, 可以用 $[\rho_1 \rho_4 \{\rho_3\}^*]_{RVII}$ 表示 1 级复合关系 $\rho_1 \rho_4 \{\rho_3\}^*$ 的等价类。为了说明 ANT VII 与定义 12 给出的 6 种抽象结点型的不同, 本文引入了“抽象结点型尺度”这一个概念。

定义 18(抽象结点型尺度) 用符号 $ANT(k)$ 表示抽象结点型 k , 用符号 $Scale(ANT(k))$ 表示抽象结点型 k 的尺度, 则: $Scale(ANT(k)) = \min\{i | \forall \rho \in [*]_{Rk} \wedge \rho \in Struct^i(WSC-PAM)\}$. 其中 $\forall \rho \in [*]_{Rk}$ 表示属于抽象结构型 k 的任意一个结构。

根据上述定义可知, 抽象结点型的尺度即为属于该抽象结点型的结构的最小级数。因此, $Scale(ANT(I)) = Scale(ANT(II)) = Scale(ANT(III)) = Scale(ANT(IV)) = Scale(ANT(V)) = Scale(ANT(VI)) = 0$; $Scale(ANT(VII)) = 1$ 。

因为 $\{\rho_3\}$ 是属于 $[\{\rho_3\}]_y^{MS_1}$ 的唯一元素, $\{\rho_4\}$ 是属于 $[\{\rho_4\}]_y^{MS_2}$ 的唯一元素, 所以 (2-5) 对应的一个 1 级复合结构为 $\rho_3\rho_4\rho_3$; 分别取 $[\{\rho_1\}]_y^{MS_1}$ 中的元素 $\{\rho_1\}^*$ 、 $\{\rho_3\}^*$ 、 $\{\rho_1, \rho_3\}^*$ 以及 $[\{\rho_2\}]_y^{MS_2}$ 中的元素 $\{\rho_2\}^*$ 、 $\{\rho_4\}^*$ 、 $\{\rho_5\}^*$ 、 $\{\rho_6\}^*$ 、 $\{\rho_2, \rho_4, \rho_5, \rho_6\}^*$ 先进行 (1-6) 的复合运算, 再与基本关系 ρ_2 进行复合运算, 可以得到 (2-5) 对应的 1 级复合结构 $\{\rho_1\rho_2\}^*\rho_3$ 、 $\{\rho_1\rho_4\}^*\rho_3$ 、 $\{\rho_1\}^*\rho_5$ 、 $\{\rho_3\}^*\rho_5$ 、 $\{\rho_1\}^*\rho_6$ 、 $\{\rho_3\}^*\rho_6$ 、 $\{\rho_1, \rho_3\}^*\rho_5$ 、 $\{\rho_1, \rho_3\}^*\rho_6$ 、 $\{\rho_3\rho_2\}^*\rho_3$ 、 $\{\rho_3\rho_4\}^*\rho_3$ 、 $\{\{\rho_3\}^*\rho_5\}^*\rho_3$ 、 $\{\{\rho_3\}^*\rho_6\}^*\rho_3$ 、 $\{\rho_3\}^*\{\rho_2, \rho_4, \rho_5, \rho_6\}^*\rho_3$ 、 $\{\rho_1\rho_2, \rho_3\rho_2\}^*\rho_3$ 、 $\{\rho_1\rho_4, \rho_3\rho_4\}^*\rho_3$ 、 $\{\{\rho_1, \rho_3\}^*\rho_5\}^*\rho_3$ 、 $\{\{\rho_1, \rho_3\}^*\rho_6\}^*\rho_3$ 、 $\{\rho_1\rho_2, \rho_3\rho_2\}^*\rho_3$ 、 $\{\rho_1\rho_4, \rho_3\rho_4\}^*\rho_3$ 、 $\{\{\rho_1, \rho_3\}^*\rho_5\}^*\rho_3$ 、 $\{\{\rho_1, \rho_3\}^*\rho_6\}^*\rho_3$ 、 $\{\rho_2, \rho_4, \rho_5, \rho_6\}^*\rho_3$, 图示省略。 $[\{\rho_2\}]_y^{MS_2}$ 其它取值参与的 (1-6) 复合运算与上述过程类似, 在此省略。

取 (1-4) 产生的 IV 型。因为 $\{\rho_3\}$ 是属于 $[\{\rho_3\}]_y^{MS_1}$ 的唯一元素, $\{\rho_4\}$ 是属于 $[\{\rho_4\}]_y^{MS_2}$ 的唯一元素, 此时 (2-6) 对应的部分 1 级复合结构以 $\rho_3\rho_4$ 开始, 分别取 $[\{\rho_1\}]_y^{MS_1}$ 中的元素 $\{\rho_1\}^*$ 、 $\{\rho_3\}^*$ 、 $\{\rho_1, \rho_3\}^*$ 进行 (2-6) 的复合运算, 可以得到以下 1 级复合结构: $\rho_3\rho_4\{\rho_1\}^*$ 、 $\rho_3\rho_4\{\rho_3\}^*$ 、 $\rho_3\rho_4\{\rho_1, \rho_3\}^*$, 如图 4 所示。

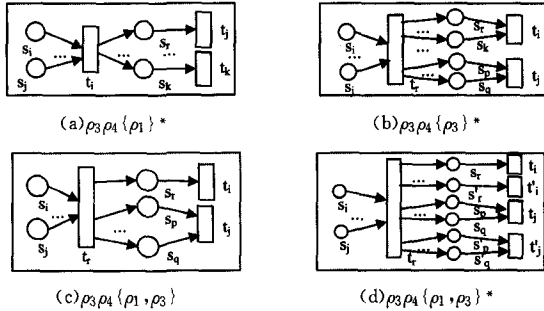


图 4 1 级复合结构(III)

观察图 4 可以发现, 图中 4 个结构的外部特征相同且均不同于前面已定义的 7 种抽象结点型, 因此将这个新的外部结构特征抽象为结点型 VIII。

定义 19(第 8 种抽象结点型) 抽象结点型 VIII 是具有多个 s 入口线和多个 t 出口线的结点类型, 可以表示为 $ANT(VIII)(in(s, n), out(t, n))$ 。

定义 19 也作为定义 12 的补充, 可以用 $[\rho_3\rho_4\{\rho_1\}^*]_{RVIII}$ 表示 1 级复合关系 $\rho_3\rho_4\{\rho_1\}^*$ 的等价类。显然, $Scale(ANT(VIII)) = 1$ 。

取 (1-6) 产生的 IV 型参与 (2-6) 的复合运算, 因为 $[\{\rho_1\}]_y^{MS_1}$ 有 4 种取值, $[\{\rho_2\}]_y^{MS_2}$ 有 25 种取值, 因此可以产生大量的 1 级复合结构, 这些复合结构均属于 ANT VIII。它们构造过程的分析与前面的 (2-1) 至 (2-5) 的分析类似, 在此省略。

(2) $MS_2MS_1MS_2$

(i) 第 1 步执行复合运算 $MS_2 \cdot MS_1$:

$$\begin{aligned}
 (3-1) \quad & \forall \rho \in [\{\rho_2\}]_y^{MS_2} \xrightarrow{(1t, 1s)} \forall \rho' \in [\{\rho_1\}]_y^{MS_1} \xrightarrow{(1s, 1t)} (1t, 1t) \quad \text{V 型} \\
 (3-2) \quad & \forall \rho \in [\{\rho_4\}]_y^{MS_2} \xrightarrow{(1t, ns)} \forall \rho' \in [\{\rho_3\}]_y^{MS_1} \xrightarrow{(ms, 1t)} (1t, 1t) \quad \text{V 型} \\
 (3-3) \quad & \forall \rho \in [\{\rho_4\}]_y^{MS_2} \xrightarrow{(1t, ns)} \forall \rho' \in [\{\rho_1\}^*]_y^{MS_1} \xrightarrow{(ms, nt)} (1t, nt) \quad \text{VI 型} \\
 (3-4) \quad & \forall \rho \in [\{\rho_5\}]_y^{MS_2} \xrightarrow{(mt, ns)} \forall \rho' \in [\{\rho_1\}]_y^{MS_1} \xrightarrow{(1s, 1t)} (mt, 1t) \quad \text{VII 型} \\
 (3-5) \quad & \forall \rho \in [\{\rho_5\}]_y^{MS_2} \xrightarrow{(mt, ns)} \forall \rho' \in [\{\rho_3\}]_y^{MS_1} \xrightarrow{(ms, 1t)} (mt, 1t) \quad \text{VII 型} \\
 (3-6) \quad & \forall \rho \in [\{\rho_2\}^*]_y^{MS_2} \xrightarrow{(mt, ns)} \forall \rho' \in [\{\rho_1\}^*]_y^{MS_1} \xrightarrow{(ms, nt)} (mt, nt) \quad \text{VIII 型}
 \end{aligned}$$

(ii) 第 2 步执行复合运算 $MS_2MS_1 \cdot MS_2$:

$$\begin{aligned}
 (4-1) \quad & \rho \in \text{V 型} \xrightarrow{(1t, 1t)} \forall \rho' \in [\{\rho_2\}]_y^{MS_2} \xrightarrow{(1t, 1s)} (1t, 1s) \quad \text{ANT II} \\
 (4-2) \quad & \rho \in \text{V 型} \xrightarrow{(1t, 1t)} \forall \rho' \in [\{\rho_4\}]_y^{MS_2} \xrightarrow{(1t, ns)} (1t, ns) \quad \text{ANT IV} \\
 (4-3) \quad & \rho \in \text{VI 型} \xrightarrow{(1t, nt)} \forall \rho' \in [\{\rho_5\}]_y^{MS_2} \xrightarrow{(mt, 1s)} (1t, 1s) \quad \text{ANT I} \\
 (4-4) \quad & \rho \in \text{VI 型} \xrightarrow{(1t, nt)} \forall \rho' \in [\{\rho_2\}^*]_y^{MS_2} \xrightarrow{(mt, ns)} (1t, ns) \quad \text{ANT IV} \\
 (4-5) \quad & \rho \in \text{VII 型} \xrightarrow{(mt, 1t)} \forall \rho' \in [\{\rho_2\}]_y^{MS_2} \xrightarrow{(1t, 1s)} (mt, 1s) \quad \text{ANT V} \\
 (4-6) \quad & \rho \in \text{VII 型} \xrightarrow{(mt, 1t)} \forall \rho' \in [\{\rho_4\}]_y^{MS_2} \xrightarrow{(1t, ns)} (mt, ns) \quad \text{ANT VI} \\
 (4-7) \quad & \rho \in \text{VIII 型} \xrightarrow{(mt, nt)} \forall \rho' \in [\{\rho_5\}]_y^{MS_2} \xrightarrow{(mt, 1s)} (mt, 1s) \quad \text{ANT V} \\
 (4-8) \quad & \rho \in \text{VIII 型} \xrightarrow{(mt, nt)} \forall \rho' \in [\{\rho_2\}^*]_y^{MS_2} \xrightarrow{(mt, ns)} (mt, ns) \quad \text{ANT VI}
 \end{aligned}$$

因为 $\{\rho_1\}$ 是属于 $[\{\rho_1\}]_y^{MS_1}$ 的唯一元素, $\{\rho_2\}$ 是属于 $[\{\rho_2\}]_y^{MS_2}$ 的唯一元素, $\{\rho_3\}$ 是属于 $[\{\rho_3\}]_y^{MS_1}$ 的唯一元素, $\{\rho_4\}$ 是属于 $[\{\rho_4\}]_y^{MS_2}$ 的唯一元素, 由 (3-1) 和 (3-2) 可知, V 型对应的复合运算只有以下两种: $\rho_2\rho_1$ 和 $\rho_4\rho_3$ 。故 (4-1) 和 (4-2) 对应的两个 1 级复合结构为: $\rho_2\rho_1\rho_2$ 和 $\rho_4\rho_3\rho_4$, 如图 5 所示。

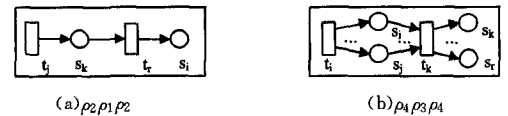


图 5 1 级复合结构(IV)

从图 5 可以直观看到, $\rho_2\rho_1\rho_2 \in [\rho_2]_{RIV}$, $\rho_4\rho_3\rho_4 \in [\rho_4]_{RIV}$, 因此 $\rho_2\rho_1\rho_2$ 可以被看作是粗粒度的基本关系 ρ_2 , $\rho_4\rho_3\rho_4$ 可以被看作是粗粒度的基本关系 ρ_4 。

$[\{\rho_1\}]_y^{MS_1}$ 中共有 4 个元素 $\{\rho_1\}^*$ 、 $\{\rho_3\}^*$ 、 $\{\rho_1, \rho_3\}^*$ 、 $\{\rho_1, \rho_3\}^*$ 。由 (3-3) 可知, VI 型对应的复合运算有: $\rho_4\{\rho_1\}^*$ 、

$\rho_4 \{ \rho_3 \}^* \sim \rho_4 \{ \rho_1, \rho_3 \} \sim \rho_4 \{ \rho_1, \rho_3 \}^*$ 。故(4-3)对应的1级复合结构为: $\rho_4 \{ \rho_1 \}^* \sim \rho_5 \sim \rho_4 \{ \rho_3 \}^* \sim \rho_5 \sim \rho_4 \{ \rho_1, \rho_3 \} \sim \rho_5 \sim \rho_4 \{ \rho_1, \rho_3 \}^* \sim \rho_5$, 如图6所示。

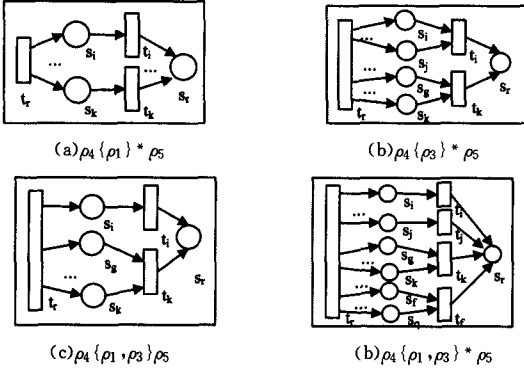


图6 1级复合结构(V)

观察图6可以发现, 图中所示的4种1级复合结构均属于 ρ_2 的等价类 $[\rho_2]_{RII}$, 因此它们都可以被看作是粗粒度的基本关系 ρ_2 。

在VI型的基础上进行(4-4)的复合运算可以得到许多1级复合结构, 这是因为 $[\{\rho_2\}^*]_Y^{MS_2}$ 的可能取值多达25个。这里仅取 $[\{\rho_2\}^*]_Y^{MS_2}$ 中的几个值进行讨论。

a) 取 $\{\rho_2\}^*$ 。VI型结构与 $\{\rho_2\}^*$ 进行复合运算, 可以得到如下1级复合结构: $\rho_4 \{ \rho_1 \rho_2 \}^* \sim \rho_4 \{ \rho_3 \rho_2 \}^* \sim \rho_4 \{ \rho_1 \rho_2, \rho_3 \rho_2 \}^*$ 和 $\rho_4 \{ \rho_1 \rho_2, \rho_3 \rho_2 \}^*$, 如图7所示。

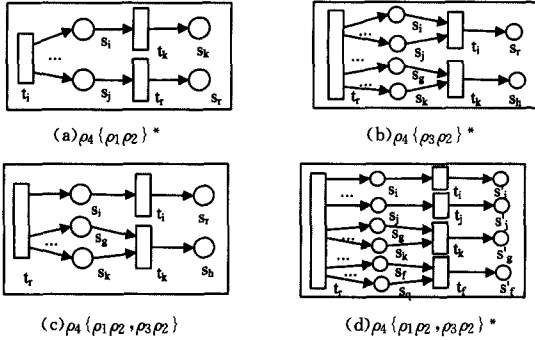


图7 1级复合结构(VI)

b) 取 $\{\rho_6\}$ 。VI型结构与 $\{\rho_6\}$ 进行复合运算, 可以得到如下1级复合结构: $\rho_4 \{ \rho_1 \}^* \sim \rho_6 \sim \rho_4 \{ \rho_3 \}^* \sim \rho_6 \sim \rho_4 \{ \rho_1, \rho_3 \} \rho_6$ 和 $\rho_4 \{ \rho_1, \rho_3 \}^* \rho_6$, 如图8所示。

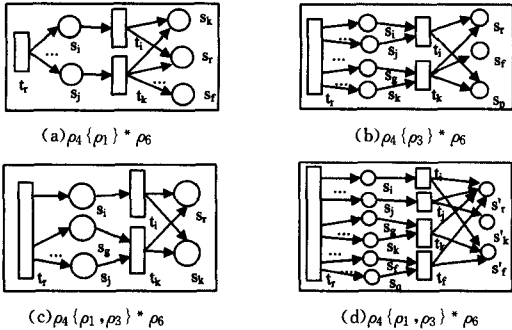


图8 1级复合结构(VII)

c) 取 $\{\rho_2, \rho_4\}$ 。VI型结构与 $\{\rho_2, \rho_4\}$ 进行复合运算, 根据本节前面对符号 $\{\rho\}$ 的说明可知, 此时VI型结构 $\rho_4 \{ \rho_1 \}^*$ 、 $\rho_4 \{ \rho_3 \}^*$ 中的 $\{\rho_1\}^*$ 和 $\{\rho_3\}^*$ 分别只能包含两个基本关系 ρ_1 和 ρ_3 , 而VI型结构 $\rho_4 \{ \rho_1, \rho_3 \}^*$ 中的 $\{\rho_1, \rho_3\}^*$ 至少包含3个基本

关系 ρ_1 或 ρ_3 , 因此不参与此次复合运算, 所以可以得到如下1级复合结构: $\rho_4 \{ \rho_1 \rho_2, \rho_1 \rho_4 \}^*$ 、 $\rho_4 \{ \rho_3 \rho_2, \rho_3 \rho_4 \}^*$ 、 $\rho_4 \{ \rho_1 \rho_2, \rho_3 \rho_4 \}^*$ 和 $\rho_4 \{ \rho_1 \rho_4, \rho_3 \rho_2 \}^*$, 如图9所示。

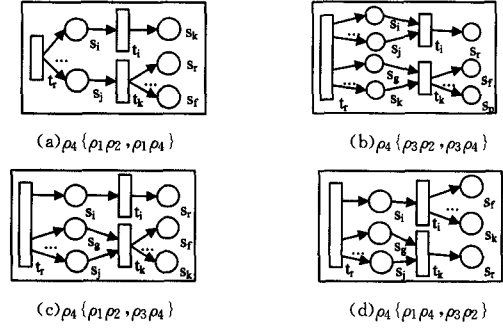


图9 1级复合结构(VIII)

d) 取 $\{\rho_2, \rho_4\}^*$ 。VI型结构与 $\{\rho_2, \rho_4\}^*$ 进行复合运算, 可以得到如下1级复合结构: $\rho_4 \{ \rho_1 \rho_2, \rho_1 \rho_4 \}^* \sim \rho_4 \{ \rho_3 \rho_2, \rho_3 \rho_4 \}^*$ 、 $\rho_4 \{ \rho_1 \rho_2, \rho_3 \rho_4 \}^*$ 和 $\rho_4 \{ \rho_1 \rho_4, \rho_3 \rho_2 \}^*$, 图示省略。

e) 取 $\{\rho_2, \rho_5\}$ 。因为VI型结构为

$$\forall \rho \in [\rho_4]_Y^{MS_2} \xrightarrow{(lt, ms)} \forall \rho' \in [\rho_1]_Y^{MS_1} \xrightarrow{(ms, nt)} (lt, nt)$$

属于等价类 $[\{\rho_1\}^*]_Y^{MS_1}$ 的取值有 $\{\rho_1\}^*$ 、 $\{\rho_3\}^*$ 、 $\{\rho_1, \rho_3\}$ 、 $\{\rho_1, \rho_3\}^*$ 。由 ρ_2 和 ρ_5 所代表的结构(参见图1(b)和图1(e))可知, $\{\rho_2, \rho_5\}$ 需要3个及3个以上的 t 入口线, 而 $\{\rho_1, \rho_3\}$ 只具有2个 t 出口线(ρ_1 和 ρ_3 所代表的结构参见图1(a)和图1(c)), 因此 $\{\rho_1, \rho_3\}$ 不能参与此结构复合。属于等价类 $[\{\rho_1\}^*]_Y^{MS_2}$ 的取值只有 ρ_4 , 因此得到中间结构有: $\rho_4 \{ \rho_1 \}^* \sim \rho_4 \{ \rho_3 \}^*$ 和 $\rho_4 \{ \rho_1, \rho_3 \}^*$ 。它们与 $\{\rho_2, \rho_5\}$ 进行复合, 可以得到的1级复合结构有: $\rho_4 \{ \rho_1 \rho_2, \{\rho_1\}^* \rho_5 \}^*$ 、 $\rho_4 \{ \rho_3 \rho_2, \{\rho_3\}^* \rho_5 \}^*$ 、 $\rho_4 \{ \rho_1 \rho_2, \{\rho_3\}^* \rho_5 \}^*$ 、 $\rho_4 \{ \{\rho_1\}^* \rho_5, \rho_3 \rho_2 \}^*$ 、 $\rho_4 \{ \rho_1 \rho_2, \{\rho_1, \rho_3\}^+ \rho_5 \}^*$ 、 $\rho_4 \{ \rho_3 \rho_2, \{\rho_1, \rho_3\}^+ \rho_5 \}^*$, 如图10所示。

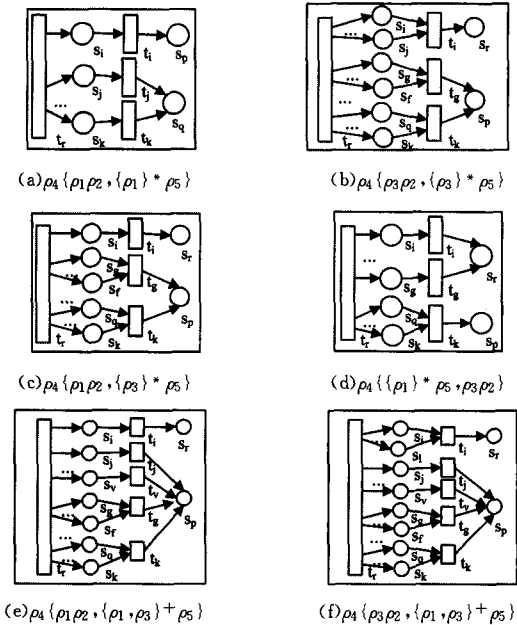


图10 1级复合结构(IX)

由(3-4)可知, VII型中间结构为 $\rho_5 \rho_1$, 在此基础上进行(4-5)的复合得到的1级复合结构为 $\rho_5 \rho_1 \rho_2$; 进行(4-6)的复合得到的1级复合结构为 $\rho_5 \rho_1 \rho_4$, 如图11所示。

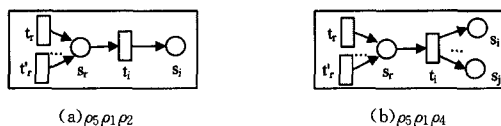


图 11 1 级复合结构(X)

因为 $[\rho_e]^{-1}]_7^{MS_2}$ 有25种取值,所以由(3-5)确定的VII型中间结构和由(3-6)确定的VIII型中间结构有很多,这里不一一列举,仅举几个它们参与(4-5)、(4-6)、(4-7)和(4-8)复合运算得到的1级复合结构,如图12所示。

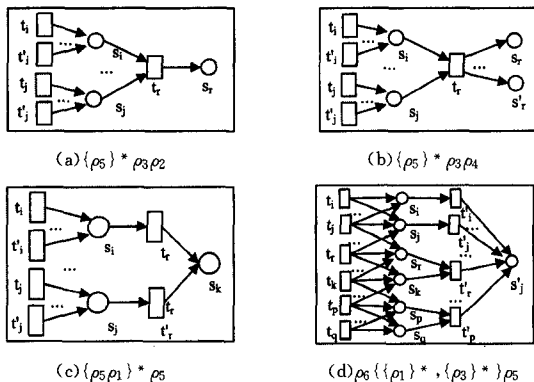


图 12 1 级复合结构(XI)

3.1.3 WSCPAM 基本结构集的完备性

定理 1(WSCPAM 结构完备性定理) WSCPAM|BS 是完备的,即由 WSCPAM|BS 中的基本结构可以构造出 WSCAPM 中的所有结构。

证明:通过 3.1.2 节对 1 级复合结构构造过程的分析可以发现,整个构造过程可以看成是一个函数 f_1 ,而 f_1 表示由基本结构构造 1 级复合结构的过程,其定义域为基本结构集,其值域为抽象结构类型。根据 WSCPAM 的定义可知, f_1 的定义域是完备的。根据定义 11 对线型的定义可知,线型第一属性“线类型”的取值范围为集合 $\{s, t\}$,第二属性“线数量”的取值类型有两种 $\{1, n\}$ (其中 $n \geq 2$),因此共有 $C_2 \times C_2 = 4$ 种线型。再根据定义 12 对抽象结点类型的定义可知,共有 $C_2 \times 4 = 8$ 种抽象结点型。WSCPAM 的结构约束条件使得 WSCPAM 的基本结构集只涵盖了 6 种抽象结点型,但在利用基本结构构造 1 级复合结构的过程中产生了另外两种抽象结点型(见定义 17 和定义 19),从而使得 1 级复合结构涵盖了所有抽象结点型,因此 f_1 的值域是完备的。

通过前面的 1 级复合结构的构造过程可以看到, 结构构造是根据结构入口线型和出口线型进行的, 能够进行复合的结构其出口线型和入口线型必须是匹配的。每构造一个新的复合结构都需要进行两次复合运算, 因为共有 4 种线型, 所以匹配的组合数是有限的、可以穷尽的。按照这种构造方式可以得到所有的 1 级复合结构, 故 f_2 的定义域是完备的且定义域覆盖了所有抽象结点型。因为抽象结点型的入口线型与出口线型的构造过程是完备的, 所以可以得到所有的 2 级复合结构, 故 f_3 的定义域是完备的。显然抽象结点型集合上的上述结构复合运算是封闭的, 所以得到的所有 2 级复合结构覆盖所有抽象结点型, 即 f_2 的值域是完备的, 且 f_3 的定义域覆盖了所有抽象结点型, 从而保证了后续结构复合过程的完备性。以此类推可知, 由最初的 WSCPAM 的基本结构可以构造得到 WSCPAM 的所有可能结构, 因此 WSCPAM 的基本

结构集是完备的。定理 1 得证。

3.2 WSCPAM 的最小结构完备集

定义 20(最小基本结构集) 假设结构集合 BSS 是网 Σ 的基本结构集, 符号 $\odot$ 表示某种基本结构类型, 符号 $\odot \angle BSS$ 表示基本结构类型 $\odot$ 可以由集合 BSS 中的其他基本结构类型组合得到。如果命题 $\exists BSS_1, BSS_2 \subseteq BSS \ \&\& \ BSS_1 \cap BSS_2 = \emptyset \ \&\& \ BSS_1 \cup BSS_2 = BSS; \forall \odot \in BSS_2 (\odot \angle BSS_1) \ \&\& \rightarrow \exists \odot \in BSS_1 (\odot \angle BSS_1)$ 成立, 那么 BSS_1 称为 Σ 的最小基本结构集(简称为最小结构集), 用符号 $\nabla(\Sigma)$ 表示。

定理 2 WSCPAM 的最小结构集 (Smallest Structure Set of WSCPAM) 为:

$$\nabla(\text{WSCPAM}) = \{\rho_1(E_S:R_{1:1}), \rho_2(E_T:R_{1:1}), \rho_3(E_S:R_{n:1}), \rho_4(E_T:R_{1:n})\}$$

证明:先对 BSS_1 和 BSS_2 做初始化。

$$BSS_1(WSCPAM) = WSCPAM \mid BS, BSS_2(WSCPAM) =$$

(1)对 $\rho_6(E_T : R_{m:n})$ 经过1次复合运算(参见(3-5)、(3-6))可能得到的中间结构做如图13所示的结构转换。

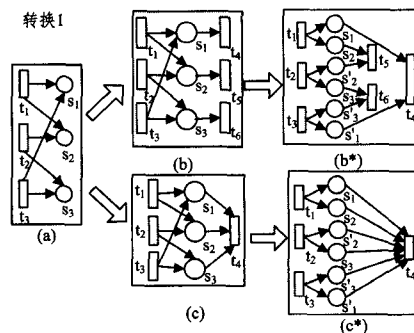


图 13 结构转换(I)

基本结构 $\rho_6(E_T; R_{m:n})$ 经过 1 次复合运算得到的结构还有一种, 因为它是图 13 中(b)和(c)的综合, 因此在图 13 中并未画出。从图 13 所示的转换 1 中可以看到, ρ_6 对应的两个中间结构 $\rho_6\{\rho_1\}^*$ 和 $\rho_6\rho_3$ 可以转换为由基本结构 ρ_4 和 ρ_3 经过复合运算得到的中间结构, 分别为 $\{\rho_4\}^* \cdot \{\rho_3\}^*$ 和 $\{\rho_4\}^* \rho_3$ 。由上述分析可知, ρ_6 对应的所有中间结构均可以通过 ρ_4 和 ρ_3 进行表示, 因此可以认为 ρ_6 可以通过 ρ_4 和 ρ_3 进行表示。即

$$\begin{aligned} & \because \rho_6(E_T : R_{m:n}), \rho_4(E_T : R_{l:n}), \rho_3(E_S : R_{m:1}) \in BSS_1 \\ & \text{(WSCPAM)} \end{aligned}$$

$$\text{又} \because \rho_6(E_T:R_{m:n}) \angle (\rho_4(E_T:R_{1:n}), \rho_3(E_S:R_{m:1}))$$

$$\therefore BSS_1(WSCPAM) = BSS_1(WSCPAM) - \{\rho_6(E_T: R_{m:n})\}$$

$$BSS_2(WSCPAM) = BSS_2(WSCPAM) \cup \{\rho_6(E_T : R_{m:n})\}$$

(2) 为了进一步明确地表示 Web 服务或子 Web 服务组合之间的前后执行的制约关系,用 Web 服务或子 Web 服务组合之间的一个库所来表示它们之间的直接制约关系;直接前驱变迁向库所中发送服务请求,后继变迁从库所中取出服务请求进行消费。换句话说,本文把库所看作是具有直接制约关系的每对 Web 服务或子 Web 服务组合所拥有的一个私有管道。前驱服务向管道里发消息,后继服务从管道里取消息,从而协调它们之间的执行次序。根据上述思想,可以进行如图 14 和图 15 所示的结构转换。

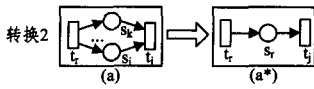


图 14 结构转换(II)

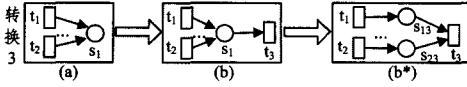


图 15 结构转换(III)

因为 $\rho_5(E_T:R_{m,1})$ 的出口线型为 $(s, 1)$, 入口线型与之相匹配的只有基本结构 $\rho_1(E_S:R_{1,1})$, 因此 $\rho_5(E_T:R_{m,1})$ 经过 1 次复合运算得到的中间结构只有一种, 如图 15(b) 所示。从图 15 所示转换 3 中可以看到, ρ_5 对应的唯一中间结构 $\rho_5\rho_1$ 可以转换为由基本结构 ρ_2 和 ρ_3 经过复合运算得到的中间结构 $\{\rho_2\} * \rho_3$, 所以可以认为 ρ_5 能够通过 ρ_2 和 ρ_3 进行表示。

$\therefore \rho_5(E_T:R_{m,1}), \rho_2(E_T:R_{1,1}), \rho_3(E_S:R_{m,1}) \in BSS_1(WSCPAM)$

又 $\therefore \rho_5(E_T:R_{m,1}) \angle (\rho_2(E_T:R_{1,1}), \rho_3(E_S:R_{m,1}))$

$\therefore BSS_1(WSCPAM) = BSS_1(WSCPAM) - \{\rho_5(E_T:R_{m,1})\}$

$BSS_2(WSCPAM) = BSS_2(WSCPAM) \cup \{\rho_5(E_T:R_{m,1})\}$

此时, $BSS_1(WSCPAM) = \{\rho_1(E_S:R_{1,1}), \rho_2(E_T:R_{1,1}), \rho_3(E_S:R_{m,1}), \rho_4(E_T:R_{m,1})\}$ 。由图 14 所示的转换 2 可知, 中间结构 $\rho_3\rho_1$ 可以看成是由 $\rho_2\rho_1$ 构成的顺序结构。除此之外, 容易发现这 4 个基本结构已相当基本, 它们中的任何一个的所有中间结构都不可能通过其余的结构进行表示。因此 $BSS_1(WSCPAM)$ 即为所求最小结构集 $\nabla(WSCPAM)$ 。

推论 1 $WSCPAM$ 的最小结构集 $\nabla(WSCPAM)$ 是完备的。

证明: 因为 $\nabla(WSCPAM) \subset WSCPAM \mid BS$, 且由定理 1 可知, $WSCPAM \mid BS$ 是完备的, 所以 $\nabla(WSCPAM)$ 是完备的, $\nabla(WSCPAM)$ 即为 $WSCPAM$ 的最小结构完备集。

4 方案实施

文献[24]提出的瓶颈定位策略 BNL-WSCPAM(Bottle-Neck Location in WSCPAM)就是本文提出的瓶颈定位方案的具体实施。

根据本文 1.3 节给出的方案路线, BNL-WSCPAM 的具体实施过程简述如下: (1) 以 $WSCPAM$ 最小结构完备集为理论基础, 构建 BNL-WSCPAM 的实现基础。首先以 $WSCPAM$ 最小结构完备集为基础, 提出瓶颈定位分析结构集{顺序结构、分支结构、汇聚结构}, 然后根据库所中托肯消耗的速度将库所分成主动受阻库所、被动受阻库所和流通库所 3 类, 接着分别基于 3 类库所在分析结构集上的不同分布情况, 通过顺向结构分析和逆向结构分析相结合的方法, 讨论瓶颈可能存在的位置, 形成一个性能瓶颈定位分析算法库。(2) BNL-WSCPAM 的瓶颈定位实现。筛选出目标库所集, 从目标库所集中选择一个库所作为瓶颈定位开始库所。根据结构发现定理判断该库所所处的结构类型, 然后调用相应的分析算法从该库所所在分析结构开始对该库所所能影响的区域进行性能瓶颈定位分析。分析结束后, 判断目标库所集是否还有尚未判断的库所, 如果还有, 那么任选一个库所重复上面的分析过程, 这个过程持续到目标库所集中的所有库所均已判

断为止。BNL-WSCPAM 的技术路线如图 16 所示。

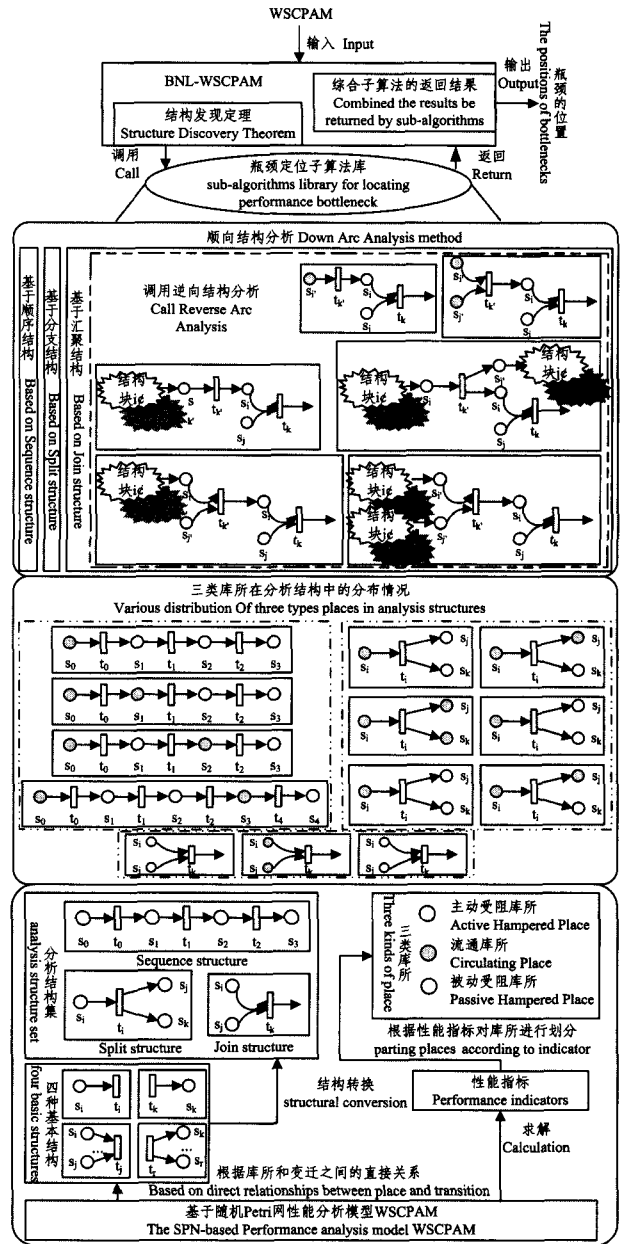


图 16 方案的具体实施 BNL-WSCPAM

某计算机公司计算机销售过程的工作流如图 17 所示[22]。用户通过 Internet 或其他方式向公司发出订单, 给出所需要的计算机基本配置和数量。公司接收用户订单, 对订单进行检查、计算价格、检查零部件的库存情况、进行配置检查、确认用户的订单技术上是否可行, 根据检查结果做出决策, 如果检查通过, 则通知用户付款并发生生产通知, 在用户付款成功和装配完成后进行发货; 如果检查不通过, 则提出订单修改意见, 并向用户反馈意见。

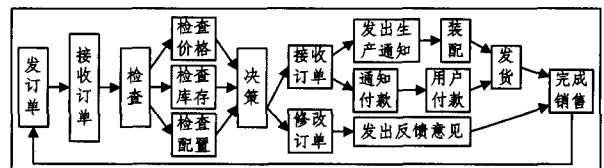


图 17 计算机销售过程的工作流程

文献[24]对上述实例应用了 BNL-WSCPAM 策略, 实验

结果证明 BNL-WSCPAM 的预测结果是有效的,从而说明了本方案是可行的和有效的。

结束语 本文提出一种基于服务组合模型结构特征的性能瓶颈分析方案。方案的主要思想是,首先分析反映服务动态特征的数据在“小结构”中的各种分布情况,每种分布情况可能预示的性能瓶颈信息是什么。在此基础上通过“小结构”之间的级联分析讨论复杂结构中可能蕴藏的性能瓶颈信息。

为了能够更加准确和清晰地描述该方案,首先给出一种基于 SPN 的 Web 服务组合性能分析模型 WSCPAM,然后给出方案的具体描述,并论证其中技术路线的可行性。接着指出实现此方案需要解决的基本问题——找出模型的最小结构完备集,并给出 WSCPAM 最小结构完备集的求解和证明过程。最后,将此方案的一个具体实现 BNL-WSCPAM 应用于一个实例,实验结果验证了方案的有效性。

WSCPAM 实质上是结构部分受限的 SPN,因此如何将方案扩展到一般的 Petri 网模型并给出方案在云服务组合性能评价的应用实例将是我们今后的努力方向。

参 考 文 献

- [1] Dustdar S, Schreiner W. A survey on web services composition [J]. International Journal of Web and Grid Services, 2005, 1(1): 1-30
- [2] Ferrara A. Web services; a process algebra approach [C]// Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC04). New York, USA, 2004
- [3] Armbrust M, Fox A, Griffith R, et al. Above the clouds: a Berkeley view of cloud computing; UCB/EECS-2009-28 [R]. EECS Department, University of California, Berkeley UCB/EECS, 2009
- [4] Boss G, Malladi P, Quan D, et al. Cloud computing. IBM White Paper [EB/OL]. http://download.boulder.ibm.com/imdl/pub/software/dw/wes/hipods/Cloud_compcompu_wp_final_8Oct.pdf, 2007
- [5] 陈康, 郑伟民. 云计算: 系统实例与研究现状 [J]. 软件学报, 2009, 20(5): 1337-1348
Chen Kang, Zheng Wei-Min. Cloud Computing: System Instances and Current Research [J]. Journal of Software, 2009, 20(5): 1337-1348
- [6] Hwang K, Fox G C, Dongarra J J. Distributed and Cloud Computing [M]. Morgan Kaufmann, 2012
- [7] 林闯, 苏文博, 孟坤, 等. 云计算安全: 架构、机制与模型评价 [J]. 计算机学报, 2013, 36(9): 1765-1784
Lin Chuang, Su Wen-bo, Meng Kun, et al. Cloud Computing Security: Architecture, Mechanism and Modeling [J]. Chinese Journal of Computer, 2013, 36(9): 1765-1784
- [8] 陶飞, 张霖, 郭华, 等. 云制造特征及云服务组合关键问题研究 [J]. 计算机集成制造系统, 2011, 17(3): 477-486
Tao Fei, Zhang Lin, Guo Hua, et al. Typical Characteristics of Cloud Manufacturing and Several Key Issues of Cloud Service Composition [J]. Computer Integrated Manufacturing Systems, 2011, 17(3): 477-486
- [9] 李伯虎, 张霖, 王时龙, 等. 云制造一面向服务的网络化制造新模式 [J]. 计算机集成制造系统, 2010, 16(1): 1-7, 16
Li Bo-hu, Zhang Lin, Wang Shi-long, et al. Cloud Manufacturing: a New Service-oriented Networked Manufacturing Model [J]. Computer Integrated Manufacturing Systems, 2010, 16(1):

- 1-7, 16
- [10] 李伯虎, 张霖, 任磊, 等. 再论云制造 [J]. 计算机集成制造系统, 2011, 17(3): 449-457
Li Bo-hu, Zhang Lin, Ren Lei, et al. Further Discussion on Cloud Manufacturing [J]. Computer Integrated Manufacturing Systems, 2011, 17(3): 449-457
- [11] Herzog U. Formal Methods for Performance Evaluation [C]// European Educational Forum; School on Formal Methods and Performance Analysis 2000, 2001: 1-37
- [12] 雷丽晖, 段振华. 一种基于扩展有限自动机验证组合 Web 服务的方法 [J]. 软件学报, 2007, 18(12): 2980-2990
Lei Li-hui, Duan Zhen-hua. An Extended Deterministic Finite Automata Based Method for the Verification of Composite Web Services [J]. Journal of Software, 2007, 18(12): 2980-2990
- [13] 廖军, 谭浩, 刘锦德. 基于 Pi-演算的 Web 服务组合的描述和验证 [J]. 计算机学报, 2005, 28(4): 635-643
Liao Jun, Tan Hao, Liu Jin-de. Describing and Verifying Web Service Using Pi-calculus [J]. Chinese Journal of Computers, 2005, 28(4): 635-643
- [14] 肖芳雄, 黄志球, 曹子宁, 等. Web 服务组合功能与 QoS 的形式化统一建模和分析 [J]. 软件学报, 2011, 22(11): 2698-2715
Xiao Fang-xiong, Huang Zhi-qiu, Cao Zi-ning, et al. Unified Formal Modeling and Analyzing Both Functionality and QoS of Web Services Composition [J]. Journal of Software, 2011, 22(11): 2698-2715
- [15] 林闯, 魏丫丫. 随机进程代数与随机 Petri 网 [J]. 软件学报, 2002, 13(2): 203-213
Lin Chuang, Wei Ya-ya. Stochastic Process Algebras and Stochastic Petri Nets [J]. Journal of Software, 2002, 13(2): 203-213
- [16] Lin Chuang, Li Ya-juan, Wang Zhong-min. Status and Development of Formal Methods for Performance Evaluation [J]. Acta Electronica Sinica, 2002, 30(12A): 1918-1922
- [17] 顾军, 罗军舟, 曹欢新, 等. 基于排队 Petri 网的服务系统性能建模与分析方法 [J]. 计算机学报, 2011, 34(12): 2435-2455
Gu Jun, Luo Jun-zhou, Cao Jiu-xin, et al. Performance Modeling and Analysis of Service Systems Using Queuing Petri Nets [J]. Chinese Journal of Computers, 2011, 34(12): 2435-2455
- [18] 许家俊, 姚淑珍. 基于 Petri 网的服务可靠性评价方法研究 [J]. 计算机科学, 2014, 41(7): 52-57
Xu Jia-jun, Yao Shu-zhen. Petri Nets Based Reliability Evaluation of Service [J]. Computer Science, 2014, 41(7): 52-57
- [19] 郭玉彬, 杜玉越, 奚建清. Web 服务组合的有色网模型及运算性质 [J]. 计算机学报, 2006, 29(7): 1067-1075
Guo Yu-bin, Du Yu-yue, Xi Jian-qing. A CP-Net Model and Operation Properties for Web Service Composition [J]. Chinese Journal of Computers, 2006, 29(7): 1067-1075
- [20] Narayanan S, McIlraith S A. Simulation, Verification and Automated Composition of Web Services [C]// Proceedings of the eleventh international conference on World Wide Web (WWW 2002). New York: ACM Press, 2002: 77-88
- [21] 钱柱中, 陆桑璐, 谢立. 基于 Petri 网的 Web 服务自动组合研究 [J]. 计算机学报, 2006, 29(7): 1057-1066
Qian Zhu-zhong, Lu Sang-lu, Xie Li. Automatic Composition of Petri Net Based Web Services [J]. Chinese Journal of Computers, 2006, 29(7): 1057-1066
- [22] Song H G, Lee K. sPAC (Web Services Performance Analysis Center): Performance Analysis and Estimation Tool of Web

Services [C]//Proc. of the 3rd International Conf. on Business Process Management (BPM 2005). LNCS 3649, Berlin, Heidelberg; Springer-Verlag, 2005; 109-119

[23] 林闯. 随机 Petri 网和系统性能评价(第 2 版)[M]. 北京:清华大学出版社, 2005

Lin Chuang. Stochastic petri net and performance evaluation of system(2nd Edit)[M]. Beijing: Tsinghua University Press,

2005

[24] 何炎祥, 沈华. 一种基于随机 Petri 网的 Web 服务组合性能瓶颈定位策略[J]. 计算机学报, 2013, 36(10): 1953-1966
He Yan-xiang, Shen Hua. A Stochastic Petri Net-Based Performance Bottleneck Location Strategy for Web Services Composition [J]. Chinese Journal of Computers, 2013, 36(10): 1953-1966

(上接第 85 页)

从图 6 对比可看出,在真实的嘈杂环境中,自适应倒谱距离算法的端点检测效果要优于改进的倒谱距离向量算法。

表 1 给出了两种算法在不同信噪比环境下通过多次实验后的实验结果。

表 1 两种方法下带噪声语音端点检测的检测正确率

SNR	白噪声		粉色噪声	
	Dcep	Dcep1	Dcep	Dcep1
-5dB	82.5%	91.0%	81.7%	90.2%
0dB	85.1%	92.4%	84.6%	91.7%
5dB	90.9%	95.3%	90.4%	94.9%
15dB	94.2%	98.9%	93.5%	98.1%
多信噪比	40.2%	87.4%	38.3%	86.7%

表 1 中, Dcep 表示传统倒谱距离端点检测方法, Dcep1 表示改进的自适应倒谱距离端点检测方法。从表中可以看出,改进的倒谱距离在单信噪比环境下的语音端点检测正确率要高于传统倒谱距离检测算法;在多信噪比环境下的端点检测效果明显高于传统倒谱距离检测算法。总的来说,改进的自适应倒谱距离端点检测算法明显优于传统的倒谱距离检测算法。

结束语 本文在传统倒谱距离端点检测的基础上引入倒谱距离乘数和门限增量系数,提出了自适应的倒谱距离语音端点检测方法,针对不同信噪比采用不同的倒谱距离乘数,并采用自适应判决门限的方法进行语音端点检测。用 MATLAB 软件进行仿真,在纯净语音中添加白噪声和粉噪声以及在同一段语音中添加多信噪比噪声进行实验,同时还采用了大量嘈杂环境下的实际语音信号进行实验仿真。实验结果表明,在不同背景噪声和多变信噪比环境下,与传统倒谱距离检测方法相比较,该方法检测正确率更高且运算量不大,更适用于复杂多变的强背景噪声环境下的语音端点检测。

在信噪比低于 -20dB 时,采用改进的倒谱距离波形用于端点检测,强度较弱的语音信号或清音信号仍然会出现丢失或是误检现象,并且清音检测的平滑度不够好,此问题将是下一步研究的重点。

参 考 文 献

- [1] 李晋, 刘甫, 王玲, 等. 改进的语音端点检测技术[J]. 计算机工程与应用, 2009, 45(24): 133-135, 149
Li Jing, Liu Fu, Wang Ling, et al. Improved technology for speech endpoint detection[J]. Computer Engineering and Applications, 2009, 25(24): 133-135, 149
- [2] 吕钊. 噪声环境下的语音识别算法研究[D]. 合肥: 安徽大学, 2011
Lv Zhao. Research on speech recognition algorithm in noisy environment[D]. Hefei: Anhui University, 2009
- [3] 张婷, 何凌, 黄华, 等. 基于临界频带及能量熵的语音端点检测

[J]. 计算机应用, 2013, 33(1): 175-178

Zhang Ting, He Ling, Huang Hua, et al. Speech endpoint detection based on critical band and energy entropy[J]. Journal of Computer Applications, 2013, 33(1): 175-178

[4] Li Jie, Zhou Ping, Jing Xin-xing, et al. Speech endpoint detection method based on TEO in noisy environment[J]. Procedia Engineering, 2012, 29(1): 2655-2660

[5] 朱恒军, 于泓博, 王发智. 小波分析和支持向量机相融合的语音端点检测算法[J]. 计算机科学, 2012, 39(6): 244-246, 265
Zhu Heng-jun, Yu Hong-bo, Wang Fa-zhi. Speech Endpoints Detection Algorithm Based on Support Vector Machine and Wavelet Analysis[J]. Computer Science, 2012, 39(6): 244-246, 265

[6] Wei Xiao-dong, Hu Guang-rui, Ren Xiao-lin. Endpoint detection of noisy speech by the use of cepstrum [J]. Engineering Village, 2014, 34(2): 185-188

[7] 胡光锐, 韦晓东. 基于倒谱特征的带噪声语音端点检测[J]. 电子学报, 2000, 28(10): 95-97

Hu Guang-rui, Wei Xiao-dong. Endpoint Detection of Noisy Speech Based on Cepstrum[J]. Acta Electronica Sinica, 2000, 28(10): 95-97

[8] 沈红丽, 曾毓敏, 王鹏. 一种改进的基于倒谱特征的带噪声语音端点检测方法[J]. 通信技术, 2009, 42(2): 156-158

Shen Hong-li, Zeng Yu-min, Wang Peng. An Improved Speech Endpoint Detection Method based on Cepstrum Distance[J]. Communications Technology, 2009, 42(2): 156-158

[9] 王博, 郭英, 李宏伟, 等. 基于倒谱距离的语音端点检测改进算法[J]. 空军工程大学学报(自然科学版), 2006, 7(1): 59-63

Wang Bo, Guo Ying, Li Hong-wei, et al. An Improved Voice Activity Detection Method Based on Cepstrum Distance[J]. Journal of Air Force Engineering University (Natural Science Edition), 2006, 7(1): 59-63

[10] 缪涛, 侯丽敏. 基于倒谱距离窗移最小失真分割的语种辨识[J]. 上海大学学报(自然科学版), 2007, 13(2): 116-120

Miu Wei, Hou Li-min. Language Identification Based on Minimum Distortion of Cepstrum Distance Segmentation[J]. Journal of Shanghai University(Natural Science Edition), 2007, 13(2): 116-120

[11] 于迎霞, 史家茂. 一种改进的基于倒谱特征的带噪声端点检测方法[J]. 计算机工程, 2004, 30(19): 85-87

Yu Ying-xia, Shi Jia-mao. A Modified Endpoint Detection Method of Noisy Speech Based on Cepstral[J]. Computer Engineering, 2004, 30(19): 85-87

[12] 蒋建中, 张东方, 张连海. 一种新的强噪声环境下的语音增强算法[J]. 计算机工程与应用, 2013, 49(20): 222-225

Jiang Jian-zhong, Zhang Dong-fang, Zhang Lian-hai. Speech enhancement algorithm for high noise environment[J]. Computer Engineering and Applications, 2013, 49(20): 222-225