

基于扩展投影时序逻辑的组合 Web 服务描述与验证

雷丽晖, 段振华

(西安电子科技大学计算理论与技术研究所, 710071, 西安)

摘要: 针对编制方式生成的组合 Web 服务需要用一个工作流引擎执行这个特征, 对投影时序逻辑进行了扩展, 并用扩展投影时序逻辑描述和验证组合 Web 服务. 将组合 Web 服务视为一个基于过程的工作流, 将工作流引擎与组合 Web 服务组成部分的一次不可分割的消息交互抽象为一个原子过程. 用扩展投影时序逻辑公式描述原子过程, 用扩展投影时序逻辑操作符定义过程组合规则、连接过程描述公式, 得到组合 Web 服务描述公式. 通过模拟组合 Web 执行, 可验证组合 Web 服务满足系统的需求和性质, 为提高组合 Web 服务设计的可靠性提供了依据.

关键词: 组合 Web 服务; 投影时序逻辑; 形式化验证

中图分类号: TP393 **文献标识码:** A **文章编号:** 0253-987X(2007)10-1155-05

Specification and Verification of Composite Web Services Based on Extended Projection Temporal Logic

Lei Lihui, Duan Zhenhua

(Institute of Computing Theory and Technology, Xidian University, Xi'an 710071, China)

Abstract: According to the feature that the execution of web services composed by orchestration is completed by a workflow execution engine, the projection temporal logic is extended. An approach based on this extended projection temporal logic (EPTL) for the specification and verification of composite web services is proposed. A composite web service is regarded as a process-based workflow. An indivisible message exchange between the workflow execution engine and a component of the composite web service is extracted as an atomic process represented by an EPTL formula. The connection rules of the processes are defined by the EPTL operators which connect the formulae representing processes to obtain new formulae representing composite web services. With simulating the execution of composite web services, whether services satisfy system requirements and properties can be verified. Therefore, this approach provides the basis for improving the reliability of the design of composite web services.

Keywords: composite web services; projection temporal logic; formal verification

Web 服务组合是整合 Internet 上现有多数异构 Web 服务(基本 Web 服务)的过程, 产生跨组织的新 Web 服务(组合 Web 服务). 目前, 越来越多的企业正在尝试使用单个 Web 服务来封装简单、独立的业务功能, 使用组合 Web 服务实现复杂的应用集

成. 应用集成首要考虑的问题是, 组合 Web 服务能否满足企业提出的系统需求和性质, 解决该问题的一种较好方法是对组合 Web 服务进行形式化验证. 在现有的验证技术中, 时序逻辑能方便、直观地描述系统的需求和性质, 且相关验证技术也比较成

熟^[1-2],然而现有的时序逻辑并不适于描述组合 Web 服务.为此,本文首先扩展了投影时序逻辑,然后用扩展投影时序逻辑描述组合 Web 服务,最后用形式化方法对组合 Web 服务进行验证.

1 组合 Web 服务

现有的 Web 服务组合方式有 2 种,即编制和编排^[3-4],本文就如何描述和验证用编制方式生成的组合 Web 服务展开了研究.

下面给出一个实例作为后面讨论的基础.设 BuyGoods 由基本 Web 服务 BookSeller、CDSeller 和 Bank 以编制方式组合,其结构如图 1 所示. Agent 是执行 BuyGoods 的工作流引擎,其业务流程如图 2 所示.购置图书和 CD 的交易过程并发执行.

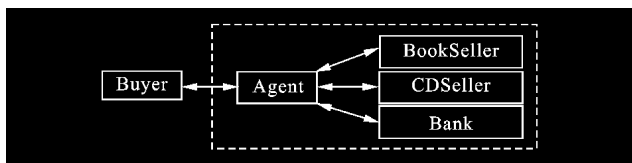


图 1 采用编制方式生成的 BuyGoods

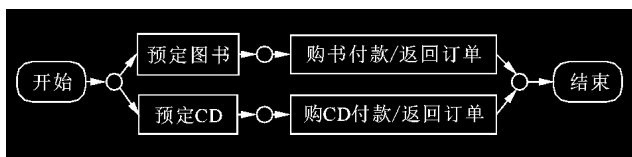


图 2 组合 Web 服务 BuyGoods 的业务流程

预定图书、CD. 根据 Buyer 发送的书名、CD 名,向 BookSeller 查询要订购的书、CD,如果有库存图书、CD,则向 Buyer 返回图书、CD 编号,否则返回无库存的通知,购置交易过程结束.

购书、CD 付款/返回订单. 接收 Buyer 发来的有效付款信息和邮寄信息,在与 Bank 交互完成付款后,根据图书、CD 编号向 Buyer 返回订单,购置交易过程结束.

2 扩展投影时序逻辑

2.1 投影时序逻辑的语法和语义

投影时序逻辑^[5](PTL)的语法定义为

表达式 $e ::= \mu | a | A | g(e_1, \dots, e_n) | \bigcirc e$

公式 $f ::= p | P(e_1, \dots, e_n) | e_1 = e_2 |$

$\neg f | f_1 \wedge f_2 | \forall v. f | \bigcirc f | \square f | f_1 ; f_2 |$

$f * | \text{skip} | (f_1, \dots, f_n) \text{prj} f_k$

式中: $\wedge$ 和 $\neg$ 为命题逻辑操作符; $\forall$ 和 $=$ 为一阶逻辑操作符; $\bigcirc$ 和 $\square$ 为基本时序操作符; skip、prj 以及

“;”和“*”为时序操作符; p 为命题; μ 为常量; a 为静态变量(区间上的值不变); A 为状态变量(区间上的值可变); v 为静态或状态变量; g 为函数符号; P 为谓词符号.

定义 1 状态 $s = (I_v, I_p)$ 是一个二元组, $I_v: V \rightarrow D$, $I_p: P_{\text{prop}} \rightarrow B$, 其中 P_{prop} 为命题的有穷集合, V 是变量的有穷集合, B 是布尔值集合, D 是值域.

定义 2 区间 $\sigma = \langle s_0, s_1, \dots, s_n \rangle$ 是一个非空有穷状态序列, 其长度为 $|\sigma|$, 等于区间包含的状态的个数减 1. 只有 1 个状态的区间长度为 0.

定义 3 解释 $I = (\sigma, i, k, j)$ 是一个四元组, σ 是一个区间, i, k, j 都是整数且 $0 \leq i \leq k \leq j \leq |\sigma|$. 解释 I 的直观语义为: 在 σ 子区间 $\langle s_i, \dots, s_j \rangle$ 上, 当前状态为 s_k , 给出表达式的值或公式的真假.

定义 4 若 $(\sigma, 0, 0, |\sigma|) \models f$, 则称区间 σ 满足公式 f , 记作 $\sigma \models f$. 如果存在 σ , 使得 $\sigma \models f$, 则称 f 是可满足的, 且 σ 是 f 的模型.

本文使用的导出操作符定义如下

$$\text{af}(x) \stackrel{\text{def}}{=} p_x,$$

若 x 被赋值则 p_x 为 true, 否则为 false

$$\text{lbf}(x) \stackrel{\text{def}}{=} \neg \text{af}(x) \rightarrow \exists b: (\exists x = b \wedge x = b)$$

$$\text{frame}(x) \stackrel{\text{def}}{=} \square(\bigcirc \text{true} \rightarrow \bigcirc \text{lbf}(x))$$

2.2 交错和限制性交错操作符

由 PTL 基本操作符导出的操作符 $\otimes(f_1, \dots, f_n)$

$$\stackrel{\text{def}}{=} (\neg f_1 \wedge \otimes(f_2, \dots, f_n)) \vee (f_1 \wedge \neg \otimes(f_2, \dots, f_n)) \text{ 且}$$

$$\otimes(f_1) \stackrel{\text{def}}{=} f_1, \text{ 以及 } \text{stable}(e) \stackrel{\text{def}}{=} \exists a. \square(e = a).$$

定义 5 设 W 为一个状态变量集合且 $W \subseteq V$, $\text{frame}(W)$ 表示在某区间上 W 中的状态变量若被赋值则变化, 否则保持不变, 而 W 以外的变量值不变. 语义定义如下: $I \models \text{frame}(W)$, 当且仅当对所有 $v \in V - W$ 和 $w \in W$, $I \models \text{stable}(v) \wedge w. \text{frame}(w)$.

设 W 是公式 f 中的状态变量的集合, 那么 $\text{Frm}(f) \stackrel{\text{def}}{=} \text{frame}(W) \wedge f$, 若 W 为空则 $\text{Frm}(f) \stackrel{\text{def}}{=} f$.

定义 6 设 $f_1 \stackrel{\text{def}}{=} f_1^1; \dots; f_m^1$, 并有 $\langle s_0^1, \dots, s_m^1 \rangle \models f_1$ 和 $\langle s_k^1, s_{k+1}^1 \rangle \models f_{k+1}^1$ ($0 \leq k \leq m-1$), $f_2 \stackrel{\text{def}}{=} f_2^1; \dots; f_n^2$, 且有 $\langle s_0^2, \dots, s_n^2 \rangle \models f_2$ 和 $\langle s_h^2, s_{h+1}^2 \rangle \models f_{h+1}^2$ ($0 \leq h \leq n-1$), 那么 $(f_1, f_2) \stackrel{\text{def}}{=} \otimes(\text{Frm}(f_1^1); \dots; \text{Frm}(f_n^2), \dots, \text{Frm}(f_1^2); \dots; \text{Frm}(f_m^1))$, 该公式构造方法如下.

(1) 根据 f_1 构造 $M_1 = (\Sigma_1, Q_1, \delta_1, q_{10}, F_1)$, 其中 $\Sigma_1 = \{f_1^1, \dots, f_m^1\}$, $Q_1 = \{s_0^1, \dots, s_m^1\}$, $\delta_1(s_i^1, f_{i+1}^1) =$

$s_{i+1}^1 (0 \leq i \leq m-1), q_{10} = s_0^1, F_1 = \{s_m^1\}$.

根据 f_2 构造 $M_2 = (\Sigma_2, Q_2, \delta_2, q_{20}, F_2)$, 其中 $\Sigma_2 = \{f_1^2, \dots, f_n^2\}, Q_2 = \{s_0^2, \dots, s_n^2\}, \delta_2(s_i^2, f_{i+1}^2) = s_{i+1}^2 (0 \leq i \leq n-1), q_{20} = s_0^2, F_2 = \{s_n^2\}$.

构造 $M = (\Sigma, Q, \delta, q_0, F)$ 如图 3 所示, 其中 $\Sigma = \Sigma_1 \cup \Sigma_2, Q = Q_1 \times Q_2, q_0 = (s_0^1, s_0^2), F_1 = \{(s_m^1, s_n^2)\}, \delta = \{((s_k^1, s^2), f_{k+1}^1, (s_{k+1}^1, s^2)) \in (Q_1 \times Q_2) \times \Sigma_1 \times (Q_1 \times Q_2) \mid (s_k^1, f_{k+1}^1, s_{k+1}^1) \in \delta_1\} \cup \{((s^1, s_h^2), f_{h+1}^2, (s^1, s_{h+1}^2)) \in (Q_1 \times Q_2) \times \Sigma_2 \times (Q_1 \times Q_2) \mid (s_h^2, f_{h+1}^2, s_{h+1}^2) \in \delta_2\}$.

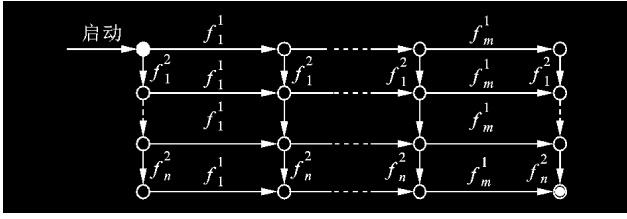


图3 定义6中构造的确定有限自动机 M

(2) M 从初始状态至终止状态得到的所有状态转移标注序列集合为 $\{(f_1^1 \dots f_n^2), \dots, (f_1^2 \dots f_m^1)\}$.

(3) 由状态转移标注序列转换, 可得到 $\otimes(\text{Frm}(f_1^1); \dots; \text{Frm}(f_n^2), \dots, \text{Frm}(f_1^2); \dots; \text{Frm}(f_m^1))$.

定义7 设 $f_1 \stackrel{\text{def}}{=} f_1^1; \dots; f_m^1$, 并有 $\langle s_0^1, \dots, s_m^1 \rangle \models f_1$ 和 $\langle s_k^1, s_{k+1}^1 \rangle \models f_{k+1}^1 (0 \leq k \leq m-1), f_2 \stackrel{\text{def}}{=} f_1^2; \dots; f_n^2$, 且有 $\langle s_0^2, \dots, s_n^2 \rangle \models f_2$ 和 $\langle s_h^2, s_{h+1}^2 \rangle \models f_{h+1}^2 (0 \leq h \leq n-1)$, 那么 $[\&](f_1, f_2) \stackrel{\text{def}}{=} \otimes(\text{Frm}(f_1^1); \dots; \text{Frm}(f_m^1), \dots, \text{Frm}(f_1^2); \dots; \text{Frm}(f_n^2))$, 其构造方法如下.

(1) 根据 f_1 构造 $M_1 = (\Sigma_1, Q_1, Q'_1, \delta_1, q_{10}, F_1)$, 其中 $\Sigma_1 = \{f_1^1, \dots, f_m^1\}, Q_1 = \{s_0^1, \dots, s_m^1\}, Q'_1 = \{s_0^1, s_m^1\}, \delta_1(s_i^1, f_{i+1}^1) = s_{i+1}^1 (0 \leq i \leq m-1), q_{10} = s_0^1, F_1 = \{s_m^1\}$.

根据 f_2 构造 $M_2 = (\Sigma_2, Q_2, Q'_2, \delta_2, q_{20}, F_2)$, 其中 $\Sigma_2 = \{f_1^2, \dots, f_n^2\}, Q_2 = \{s_0^2, \dots, s_n^2\}, Q'_2 = \{s_0^2, s_n^2\}, \delta_2(s_i^2, f_{i+1}^2) = s_{i+1}^2 (0 \leq i \leq n-1), q_{20} = s_0^2, F_2 = \{s_n^2\}$.

构造 $M = (\Sigma, Q, Q', \delta, q_0, F)$ 如图 4 所示, 其中 $\Sigma = \Sigma_1 \cup \Sigma_2, q_0 = (s_0^1, s_0^2), F = \{(s_m^1, s_n^2)\}, Q = (Q'_1 \times Q_2) \cup (Q_1 \times Q'_2), Q' = Q'_1 \times Q'_2, \delta = \{((s_k^1, s^2), f_{k+1}^1, (s_{k+1}^1, s^2)) \in (Q_1 \times Q'_2) \times \Sigma_1 \times (Q_1 \times Q'_2) \mid (s_k^1, f_{k+1}^1, s_{k+1}^1) \in \delta_1\} \cup \{((s^1, s_h^2), f_{h+1}^2, (s^1, s_{h+1}^2)) \in (Q'_1 \times Q_2) \times \Sigma_2 \times (Q'_1 \times Q_2) \mid (s_h^2, f_{h+1}^2, s_{h+1}^2) \in \delta_2\}$.

(2) M 从初始状态至终止状态得到的所有状态

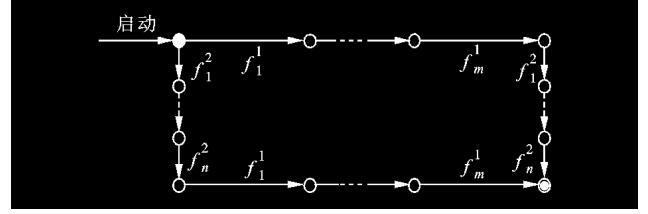


图4 定义7中构造的确定有限自动机 M

转移标注序列集合为 $\{(f_1^1 \dots f_m^1), \dots, (f_1^2 \dots f_n^2)\}$.

(3) 由状态转移标注序列转换, 可得到 $\otimes(\text{Frm}(f_1^1); \dots; \text{Frm}(f_m^1), \dots, \text{Frm}(f_1^2); \dots; \text{Frm}(f_n^2))$.

此外, $\&(f_1, \dots, f_n) \stackrel{\text{def}}{=} \&(f_1, \&(f_2, \dots, f_n))$, $[\&](f_1, \dots, f_n) \stackrel{\text{def}}{=} [\&](f_1, [\&](f_2, \dots, f_n))$. 若满足 f_1 和 f_2 的区间为 σ_1 和 σ_2 , 满足 $\&(f_1, f_2)$ (或 $[\&](f_1, f_2)$) 的区间为 σ , 那么 $|\sigma| = |\sigma_1| + |\sigma_2|$, 且在 σ 上存在 2 个投影 σ'_1 和 σ'_2 , 使得 $\sigma'_1 = \sigma_1, \sigma'_2 = \sigma_2$.

3 组合 Web 服务描述

本文将执行组合 Web 服务的工作流引擎与基本 Web 服务之间的一次不可分割的消息传递, 抽象为一个原子过程. 组合过程是由原子过程分层组合而成, 组合 Web 服务可以用一个组合过程来表示.

3.1 原子过程描述

定义8 参数用公式 $V_t(p_r, D_{\text{type}}) \wedge V_s(p_r, D_{\text{class}})$ 来描述, $V_t(p_r, D_{\text{type}})$ 是二元谓词, 表明 p_r 值的数据类型是 D_{type} 中的值, $V_s(p_r, D_{\text{class}})$ 是二元谓词, 表明 p_r 值的语义是 D_{class} 中的值.

定义9 原子过程可用公式 $P \wedge I \rightarrow \bigcirc \otimes ((E_1 \wedge O_1), \dots, (E_n \wedge O_n))$ 来描述, 其中:

$I \stackrel{\text{def}}{=} i_1 \wedge \dots \wedge i_m$ 是所有输入参数描述公式的并, 若原子过程没有输入, I 为 true;

$O_i \stackrel{\text{def}}{=} o_1 \wedge \dots \wedge o_n$ 是输出参数描述公式的并, 若原子过程没有输出, O_i 为 true;

$$P \stackrel{\text{def}}{=} \begin{cases} p_1 \wedge \dots \wedge p_h & p_i (1 \leq i \leq h) \text{ 过程执行条件} \\ \text{true} & \text{如果过程无需说明执行条件} \end{cases}$$

$$E_i \stackrel{\text{def}}{=} \begin{cases} q_1 \wedge \dots \wedge q_k & q_j (1 \leq j \leq k) \text{ 过程执行效果} \\ \text{true} & \text{如果过程无需说明执行效果} \end{cases}$$

3.2 组合过程描述

扩展投影时序逻辑 (EPTL) 的操作符定义了 4 类过程组合规则: 顺序、选择、循环及并发^[6]. 设过程 $w_1, w_2, \dots, w_n$ 的描述公式为 $f_{w1}, f_{w2}, \dots, f_{wn}$.

顺序组合规则 $w_1, w_2, \dots, w_n$ 顺序执行, 组合过程

描述公式为 $\text{Frm}(f_{w1}); \text{Frm}(f_{w2}); \dots; \text{Frm}(f_{wn})$.

选择组合规则 从 $w_1, w_2, \dots, w_n$ 中选择一个执行. 如果定义了选择条件 $b_1, \dots, b_n$, 组合过程描述公式为 $(b_1 \rightarrow \text{Frm}(f_{w1})) \wedge \neg(b_1 \rightarrow (b_2 \dots (b_n \rightarrow \text{Frm}(f_{wn})) \wedge (\neg b_n \rightarrow \neg \bigcirc \text{true}) \dots))$; 如果没有定义选择条件, 组合过程描述公式为 $\bigotimes(\text{Frm}(f_{w1}), \text{Frm}(f_{w2}), \dots, \text{Frm}(f_{wn}))$.

循环组合规则 循环条件 b 可满足时重复执行 w_k , 组合过程描述公式为 $((b \wedge \text{Frm}(f_{wk}))^+ \vee \neg \bigcirc \text{true}) \wedge \square(\neg \bigcirc \text{true} \rightarrow \neg b)$; 重复执行 w_k 直到循环条件 b 可满足, 组合过程描述公式为 $\text{Frm}(f_{wk}), ((b \wedge \text{Frm}(f_{wk}))^+ \vee \neg \bigcirc \text{true}) \wedge \square(\neg \bigcirc \text{true} \rightarrow b)$.

并发组合规则 $w_1, w_2, \dots, w_n$ 交错执行, 组合过程描述公式为 $\&(\text{Frm}(f_{w1}), \text{Frm}(f_{w2}), \dots, \text{Frm}(f_{wn}))$; $w_1, w_2, \dots, w_n$ 以任意顺序执行, 组合过程描述公式为 $[\&](\text{Frm}(f_{w1}), \text{Frm}(f_{w2}), \dots, \text{Frm}(f_{wn}))$.

3.3 组合 Web 服务实例

组合 Web 服务 BuyGoods 的参数及原子过程

描述公式如表 1、表 2 所示. 如果购置图书和 CD 的过程交错执行, BuyGoods 描述公式为 $\&(\text{Frm}(f_1); ((O_1 \rightarrow \text{Frm}(f_3)) \wedge (\neg O_1 \rightarrow \neg \bigcirc \text{true})), \text{Frm}(f_2); ((O_2 \rightarrow \text{Frm}(f_4)) \wedge (\neg O_2 \rightarrow \neg \bigcirc \text{true})))$; 如果购置图书和 CD 的过程以任意顺序执行, BuyGoods 描述公式为 $[\&](\text{Frm}(f_1); ((O_1 \rightarrow \text{Frm}(f_3)) \wedge (\neg O_1 \rightarrow \neg \bigcirc \text{true})), \text{Frm}(f_2); ((O_2 \rightarrow \text{Frm}(f_4)) \wedge (\neg O_2 \rightarrow \neg \bigcirc \text{true}))))$.

4 组合 Web 服务验证

组合 Web 服务验证的基本思想如下: 先将组合 Web 服务视为一个原子过程组建的工作流, 并用 EPTL 公式描述, 再用 Tempura^[2] 模拟组合 Web 服务执行过程, 获得组合 Web 服务运行过程的每个状态及每个状态下成立的公式, 最后根据状态序列以及每个状态下成立的公式, 证明组合 Web 服务可满足企业提出的系统需求和性质.

如果购置图书和 CD 的过程以任意顺序执行,

Tempura 模拟 BuyGoods 执行过程如图 5a 所示, 其

表 1 以 BuyGoods 为例的组合 Web 服务参数及其描述公式

参数名	参数描述公式
BookName	$I_1 \stackrel{\text{def}}{=} V_t(\text{BookName}, \text{String}) \wedge V_s(\text{BookName}, \text{Name_of_Book})$
BookISBN	$O_1 \stackrel{\text{def}}{=} V_t(\text{BookISBN}, \text{String}) \wedge V_s(\text{BookISBN}, \text{ISBN})$
BOutStock	$O_3 \stackrel{\text{def}}{=} V_t(\text{BOutStock}, \text{String}) \wedge V_s(\text{BOutStock}, \text{Notification})$
CDTitle	$I_2 \stackrel{\text{def}}{=} V_t(\text{CDTitle}, \text{String}) \wedge V_s(\text{CDTitle}, \text{Title_of_CD})$
CDID	$O_2 \stackrel{\text{def}}{=} V_t(\text{CDID}, \text{String}) \wedge V_s(\text{CDID}, \text{ID_of_CD})$
COutStock	$O_4 \stackrel{\text{def}}{=} V_t(\text{COutStock}, \text{String}) \wedge V_s(\text{COutStock}, \text{Notification})$
PayInfo	$I_3 \stackrel{\text{def}}{=} V_t(\text{PayInfo}, \text{String}) \wedge V_s(\text{PayInfo}, \text{CreditCard})$
Delivery	$I_4 \stackrel{\text{def}}{=} V_t(\text{Delivery}, \text{String}) \wedge V_s(\text{Delivery}, \text{Address})$
BookOrder	$O_5 \stackrel{\text{def}}{=} V_t(\text{BookOrder}, \text{String}) \wedge V_s(\text{BookOrder}, \text{Order})$
CDOOrder	$O_6 \stackrel{\text{def}}{=} V_t(\text{CDOOrder}, \text{String}) \wedge V_s(\text{CDOOrder}, \text{Order})$

表 2 以 BuyGoods 为例的组合 Web 服务原子过程描述公式

原子过程名	原子过程描述公式
预定书籍	$f_1 \stackrel{\text{def}}{=} I_1 \rightarrow \bigcirc \bigotimes (O_1, O_3)$
预定 CD	$f_2 \stackrel{\text{def}}{=} I_2 \rightarrow \bigcirc \bigotimes (O_2, O_4)$
购书付款/返回订单	$f_3 \stackrel{\text{def}}{=} I_3 \wedge I_4 \wedge O_1 \wedge \text{valid}(\text{PayInfo}) \rightarrow \bigcirc \bigotimes (O_5)$
购 CD 付款/返回订单	$f_4 \stackrel{\text{def}}{=} I_3 \wedge I_4 \wedge O_2 \wedge \text{valid}(\text{PayInfo}) \rightarrow \bigcirc \bigotimes (O_6)$

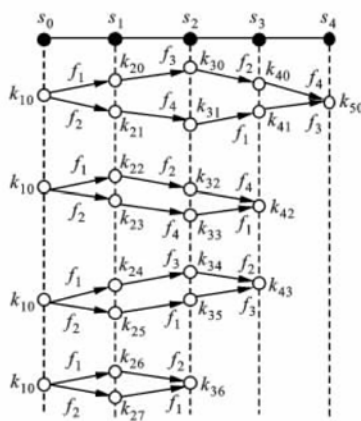
中每个状态下成立的公式放在集合 k_i 中. 根据 k_i 和 k_j 生成顺序, 用有向弧将其连接起来, 并在有向弧上标注被执行的原子过程描述公式, 所得有向图如图 5b 所示. 设企业需要服务: 输入 $\{I_1, I_2, I_3, I_4\}$, 产生输出 $\{O_5, O_6\}$, 或 $\{O_5, O_4\}$, 或 $\{O_3, O_6\}$, 或 $\{O_3, O_4\}$. 由图 5b 可知, k_{10} 包含 I_1, I_2, I_3, I_4 , k_{50} 包含 O_5 和 O_6 , k_{42} 和 k_{43} 分别包含 O_3 和 O_6 , O_5 和 O_4 , k_{36} 包含 O_3 和 O_4 , 所以 BuyGoods 可满足企业需求.

对图 5b 中任意一个有向图, 每条从初始状态到终止状态的路径描述组合 Web 服务的一个执行过程, 也是验证 BuyGoods 性质的依据. 系统性质可分为二类: 一类说明坏事永远不会发生(安全性); 另一

类说明好事最终会发生(活性). 在验证组合 Web 服务可满足企业提出的系统性质之前, 首先要确定企业提出的系统性质. 以 BuyGoods 为例, 一旦准备为预订的图书付款, 就不能再有预订图书的活动发生, 否则返回给用户的订单会包含未付款书籍, 这是企业不允许发生的事情. 如果将预订图书的过程用 f_1 表示, 付款并返回订单的过程用 f_3 表示, 验证 BuyGoods 是否满足上述性质, 只需检查 BuyGoods 的每一组原子过程执行序列是否满足: 序列中只包含 f_1 , 不包含 f_3 , 或序列中包含 f_3 且 f_3 一定在 f_1 之后. 检查结果若满足上述条件, 说明 BuyGoods 可满足这个性质. 判定算法可利用现有图论的基本知识, 这里不再讨论.



(a) 执行过程



(b) 有向图

图 5 模拟 BuyGoods 运行过程

5 结论

本文扩展了投影时序逻辑, 并用扩展投影时序逻辑描述以编制方式生成的组合 Web 服务, 在此基础上对组合 Web 服务进行了验证. 验证工作应在组合 Web 服务设计阶段完成, 可有效提高组合 Web 服务设计的可靠性.

参考文献:

- [1] Zedan H, Cau A, Moszkowski B C. Compositional modeling: the formal perspective [C]// The Proceedings of Workshop on Systems Modeling for Business Process Improvement. Norwood, USA: Artech House, 2000:333-354.
- [2] Solanki M, Cau A, Zedan H. Augmenting semantic web service description with compositional specification [C]// The Proceedings of WWW. New York:

ACM, 2004:544-552.

- [3] Peltz C. Web services orchestration and choreography [J]. Computer, 2003,36(10):46-52.
- [4] Lei Lihui, Duan Zhenhua. Automating Web service composition for collaborative business processes [C]// The Proceedings of Computer Supported Cooperative Work Design. Piscataway, USA: IEEE, 2007: 151-156.
- [5] Duan Zhenhua. An extended interval temporal logic and a framing technique for temporal logic programming [D]. Newcastle, UK: Department of Computing Science, University of Newcastle upon Tyne, 1996.
- [6] Lei Lihui, Duan Zhenhua. Transforming OWL-S process model into EDFA for service discovery [C]// The Proceedings of International Conference Web Service. Piscataway, USA: IEEE, 2006:108-116.

(编辑 苗凌)