

DOI: 10.7652/xjtuxb201409006

扩展 π 演算对时间相关移动并发系统的建模与推演

罗玲^{1,2}, 段振华^{1,2}

(1. 西安电子科技大学计算理论与技术研究所, 710071, 西安;
2. 西安电子科技大学 ISN 国家重点实验室, 710071, 西安)

摘要: 针对 π 演算难于对时间相关移动并发系统进行建模和推演,提出了一种采用扩展 π 演算 $p\text{-}\pi$ 对时间相关移动并发系统进行形式化建模与推演的方法。该方法首先采用区间动作前缀和瞬时动作前缀分别描述系统的时间相关行为和交互行为,并通过操作算子将子进程进行复合,然后利用操作规则构造出系统的时间相关标记迁移系统和可接受的执行路径,最后基于上述迁移系统和执行路径完成对系统性质的推演。对移动车辆控制系统的分析表明,所提方法可对时间相关移动并发系统进行有效建模和推演,保证时间相关移动并发系统的可靠性。

关键词: π 演算;时间相关移动并发系统;形式化建模;推演

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2014)09-0030-07

Modeling and Reasoning of Time-Dependent Concurrent Mobile Systems Based on an Extended π -Calculus

LUO Ling^{1,2}, DUAN Zhenhua^{1,2}

(1. Institute of Computing Theory & Technology, Xidian University, Xi'an 710071, China;
2. State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China)

Abstract: A method based on an extended π -Calculus, called $p\text{-}\pi$, is presented to formally model and reason time-dependent concurrent mobile systems by considering the fact that π -calculus has difficulty in modeling and reasoning time-dependent concurrent mobile systems. Both the interval action prefixes and the instantaneous action prefixes are employed to respectively describe the time-dependent behaviors and communications of systems. Operators are used to composite sub-processes, and then operational rules are utilized to construct a time-dependent labeled transition system and acceptable executive paths of the system. The properties of the system are then deduced by means of the transition system and the acceptable executive paths. Experimental analyses on a mobile vehicle control system (MVCS) show that the approach effectively models and reasons time-dependent concurrent mobile systems, and improves the reliability of time-dependent concurrent mobile systems.

Keywords: π -calculus; time-dependent concurrent mobile system; formal modeling; reasoning

近年来,随着 Software-as-a-Service(SaaS)^[1]和 Cloud Computing^[2]等新型网络计算的发展,以并发性、移动性、时间相关性和异构性等为主要特征的时

间相关移动并发系统得到人们的普遍关注,人们越来越认识到,在动态网络环境中构建时间相关移动并发系统不仅难度大、效率低,且难以发现其中隐含

收稿日期: 2014-01-20。 作者简介: 罗玲(1986—),女,博士生;段振华(通信作者),男,教授,博士生导师。 基金项目: 国家“973”重点基础研究发展规划资助项目(2010CB328102);国家自然科学基金资助项目(61003078);综合业务网理论及关键技术国家重点实验室基金资助项目(ISN1102001)。

的缺陷和错误^[3]。对于安全攸关的时间相关移动并发系统,如移动支付系统、移动通信系统、交通控制系统等,其失误和崩溃可能造成重大损失,因此时间相关移动并发系统的正确性、安全性和时间相关性等属性受到人们的重视。形式化方法以严格的数学理论为支撑,被认为是行之有效的减少设计错误、保证软件质量的重要方法。目前,该领域的研究主要集中在形式化建模和推演上。基于不同理论,各种形式化方法被相继提出,如进程代数、时序逻辑^[4]以及 Petri 网等。 π 演算^[5-6]作为进程代数的一个重要分支,是由 Milner 等人在 CCS 的基础上提出的,由于可以描述拓扑结构动态变化的移动通信系统,因而被广泛应用于移动并发系统的建模和推演中。然而,由于它的动作前缀未考虑时间因素,因而不适合于描述时间相关系统。近年来,已有一些研究对 π 演算进行相应的扩展^[3,7],使其便于对时间相关系统进行建模,但这些方法基本都是直接加入时间变量,导致时间和动作前缀分开,不能反映时间相关系统的本质特点。因此,本文通过加入区间动作前缀^[8]将 π 演算扩展为 $p\text{-}\pi$,由于区间动作前缀隐含了动作消耗时间,使得 $p\text{-}\pi$ 更适用于时间相关移动并发系统的建模和推演。

综上所述,鉴于 $p\text{-}\pi$ 和时间相关移动并发系统的特点,本文提出采用 $p\text{-}\pi$ 为时间相关移动并发系统建模和推演的方法。为了说明 $p\text{-}\pi$ 能够对时间相关系统进行有效建模和推演,文中结合实例进行了展示。

1 扩展 π 演算 $p\text{-}\pi$

1.1 语法

设 N 是名字集合, P_{rop} 是原子命题集合, $\mathbb{N}$ 是非负整数集合。 $p\text{-}\pi$ 的语法定义如下

$$\begin{aligned} \pi::&=x(y)|\bar{x}(y)|\tau|s_{\text{kip}}|\mathbf{I}_p|\epsilon \\ P::&=0|\pi.P|P_1+P_2|P_1|P_2|[a_1=a_2]P|\nu a P| \\ &A\langle a_1, \dots, a_n \rangle \end{aligned}$$

式中: $x, a, a_i (i \in \{1, \dots, n\}) \in N$; y 表示向量 $y_1, \dots, y_n$, 其中 $y_i (i \in \{1, \dots, n\}) \in N$; $x(y), \bar{x}(y), \tau$ 和 ϵ 为瞬时动作前缀, 其中 $\bar{x}(y)$ 和 $x(y)$ 分别表示沿通道 x 发送和接收消息, τ 为内部动作, ϵ 为空动作; s_{kip} 和 $\mathbf{I}_p$ 为区间动作前缀, 其中 $\mathbf{I}_p = \{\bar{p}_1, \dots, \bar{p}_n\}$, $\bar{p}_i$ 表示 $p_i \wedge s_{\text{kip}}$ 且 $p_i \in P_{\text{rop}} (i \in \{1, \dots, n\} (n \in \mathbb{N}))$; 0 为空进程; $\pi.P$ 为前缀动作监视进程; P_1+P_2 表示 P_1 和 P_2 不确定地选择执行; $P_1|P_2$ 表示进程 P_1 和 P_2 并行执行; $[a_1=a_2]P$ 表示若 $a_1=a_2$, 则该进程按 P

执行, 否则为空进程; $\nu a P$ 表示将名字 a 限制在 P 中使用, ν 是限名操作符; $A\langle a_1, \dots, a_n \rangle$ 是进程实例, A 是进程标识符。

进程表达式 P 中出现的名字记作 $n(P)$, 分为自由名和受限名, 分别记作 $f_n(P)$ 和 $b_n(P)$ 。其中受限名又包括两种: $\nu a P$ 中的 a 和 $x(y)$ 中的 y 。除此之外的都是自由名。建模时: 0 常被省略; $\nu a \nu b P$ 简记为 $\nu a, b P$ 。操作符 $., []$ 和 ν 的优先级高于 $+$ 和 $|$ 。下面给出一些派生定义

$$\begin{aligned} s_{\text{kip}}^0 &\triangleq \epsilon \\ s_{\text{kip}}^n &\triangleq s_{\text{kip}} \cdot s_{\text{kip}}^{n-1}, \quad n \geq 1 \\ \text{await}(P) &\triangleq \epsilon.P + s_{\text{kip}}.P + \dots + s_{\text{kip}}^n.P, \quad n \geq 0 \end{aligned}$$

式中: $n \in \mathbb{N}$; $\text{await}(P)$ 中的 P 为发送动作前缀监视进程或接收动作前缀监视进程。导出进程主要用于实现引入区间动作后的进程同步通信。

1.2 语义

定义 1 进程同余。设 $\cong$ 为 $p\text{-}\pi$ 进程集合 $\mathcal{P}$ 上的等价关系, $\cong$ 被称为进程同余当且仅当, 若 $P \cong Q$, 则 $\pi.P \cong \pi.Q, P+R \cong Q+R, R+P \cong R+Q, P|R \cong Q|R, R|P \cong R|Q, [a_1=a_2]P \cong [a_1=a_2]Q$, 且 $\nu a P \cong \nu a Q$ 。

定义 2 结构同余。设 $\equiv$ 为 $\mathcal{P}$ 上的等价关系, 它由以下 16 个等式定义, 它被称为结构同余当且仅当它是进程同余。

$$\begin{aligned} S1: &\nu x P \equiv \nu y \{y/x\}P, \text{ if } y \notin f_n(P) \\ S2: &x(y).P \equiv x(z). \{z/y\}P, \text{ if } z \notin f_n((y)P) \\ S3: &\nu x (P|Q) \equiv P|\nu x Q, \text{ if } x \notin f_n(P) \\ S4: &A\langle a_1, \dots, a_n \rangle \equiv \{a/x\}P_A, \text{ if } A(x_1, \dots, x_n) \triangleq P_A \\ S5: &P|0 \equiv P & S6: &P+Q \equiv Q+P \\ S7: &P|Q \equiv Q|P & S8: &P|(Q|R) \equiv (P|Q)|R \\ S9: &\nu x 0 \equiv 0 & S10: &\nu x \bar{x}(y).P \equiv 0 \\ S11: &\epsilon.P \equiv P & S12: &P+(Q+R) \equiv (P+Q)+R \\ S13: &P+0 \equiv P & S14: &[x=y]P \equiv 0, \text{ if } x \neq y \\ S15: &\nu x y P \equiv \nu y x P & S16: &s_{\text{kip}}.P + s_{\text{kip}}.Q \equiv s_{\text{kip}}.(P+Q) \end{aligned}$$

易证得 $\equiv$ 是进程同余, 故 $\equiv$ 是 $\mathcal{P}$ 上的结构同余。对于任意的 $P, Q \in \mathcal{P}$, 若能通过多次应用以上等式将 P 转换为 Q , 就说 P 和 Q 是结构同余的, 记作 $P \equiv Q$ 。

2 时间相关移动并发系统建模实例

时间相关移动并发系统是一类以移动性、并发性为主要特征并对系统行为的时序性要求严格的系统。移动车辆控制系统 (MVCS) 最早出现于文献 [5], 常被作为表现系统移动性的实例, 但以往的分

析中,都未对系统的时间相关行为建模和推演,本节以 MVCS 为例,展示 $p-\pi$ 对时间相关移动并发系统的建模。

2.1 MVCS 的系统描述

MVCS 的移动通信网络通常包括控制中心、站点和车辆,控制中心负责与站点通信,站点负责与控制中心和与其处于连接状态的车辆通信。通常,系统结构的移动性^[5]包括很多方式,如建立或销毁通信连接、建立或销毁通信进程等。在移动性方面,本例关注通信连接的建立和销毁,即随着车辆的移动,它将销毁与初始站点的连接,继而建立与新站点的连接的过程。

图 1 给出了 MVCS 移动通信的网络拓扑结构图。该系统只考虑控制中心 C,3 个站点 T^1 、 T^2 和 T^3 ,2 辆车 B^1 和 B^2 。 $T^i(i=1,2,3)$ 表现为两种状态:当与 $B^j(j=1,2)$ 连接时表现为 T^i ;当与 B^j 断开时表现为 IDT^i 。图 1 和系统建模所需名字的定义和用途如表 1 所示,其中: $i=1,2,3;j=1,2;1\leq k\leq 15$ 。

表 1 系统名字定义及用途

名字	用途
t_c^i, t_b^i	T^i 与 C 和 B^j 间传递消息的通道
l_s^i, g_t^i	C 向 T^i 发送断开和建立连接的通道
s^i	B^j 接收来自 T^i 切换信息的通道
i_n	C 接收控制信号的自由通道
o_{ut}	车辆发送控制信号的自由通道
b_t^i	表示车辆与 T^i 处于连接的自由通道
l_{tm}^i	T^i 监测的局部拥塞信息
g_m	C 处理过的全局拥塞信息
c_{md}	控制信号
t_{bb}	车辆间进行通信的通道
j_m^i	站点进行拥塞监测的自由通道
m_s	C 处理全局拥塞信息的自由通道
c_n	表示车辆间成功建立连接的自由通道
d^i	C 监测 T^i 完成局部拥塞信息接收的通道
m_k, l_k	等待被替换的局部通道
r_{eq}, a_{ck}	建立车辆间连接的请求与应答

系统功能包括 5 部分:①控制信号的接收和发送,C 接到控制信号后,将该信号传送到相应站点 T^i ,继而由该站点传给 B^j , B^j 在接收到该信号后执行该信号;②连接站点的切换,假设 B^1 先与 T^1 连接,则 C 会先将用于完成 B^1 与新站点 T^2 通信的通道通过 T^1 传给 B^1 ,再将该通道传给 IDT^2 ,这样就建立了 T^2 和 B^1 的连接,即完成了站点切换,如图 1 所示;③路况信息的监测,站点 T^i 可监测路况信息,当出现拥塞时, T^i 会将监测到的局部拥塞信息发送

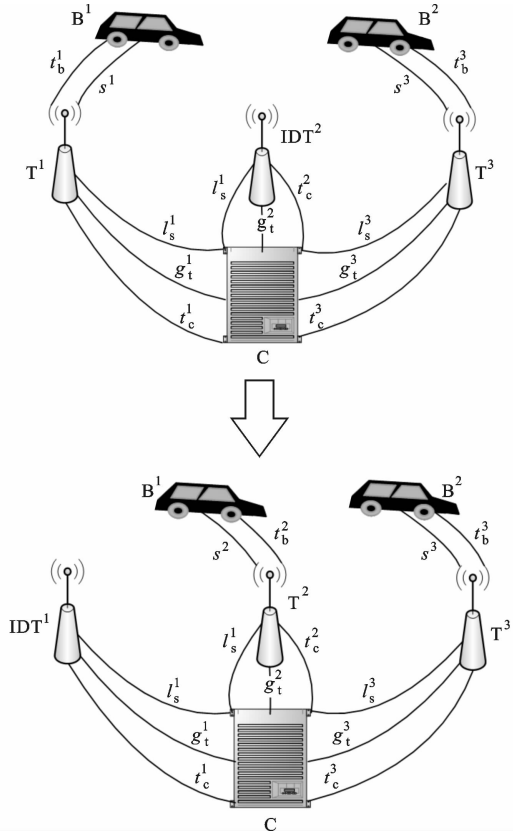


图 1 MVCS 的移动通信过程

给 C;④拥塞信息的广播,C 接收来自各个站点的局部拥塞信息,然后对信息处理后将全局拥塞信息通过站点广播给与各站点相连接的车辆;⑤车辆间通信连接的建立,假设 B^1 需要与 B^2 进行通信连接,那么 B^1 先发出建立连接的申请,该申请会通过 T^1 、C 和 T^3 传递给 B^2 , B^2 接收申请后,若同意建立连接,则将应答信息反馈回 B^1 ,即完成通信连接的建立。该系统涉及的时间相关约束为:切换信息的传输占用 1 到 2 个时间单元;断开和建立连接的过程占用 2 个时间单元;车辆在接收建立连接的应答时最多等待 5 个时间单元,超过 5 个时间单元将执行超时处理;站点处理局部拥塞信息占用 1 个时间单元,控制中心处理全局拥塞信息占用 3 个时间单元;通过自由名的发送和接收动作是异步交互,故其执行不需要等待时间;通过受限名的发送和接收动作是同步交互,故要等待与之同步的接收和发送动作。

2.2 MVCS 的建模

依据系统描述,该系统由 6 个子进程并行构成

$$\nu n(C \langle n_c \rangle \mid T^1 \langle n_{t_1} \rangle \mid IDT^2 \langle n_{t_2} \rangle \mid T^3 \langle n_{t_3} \rangle \mid B^1 \langle n_{b_1} \rangle \mid B^2 \langle n_{b_2} \rangle)$$

式中: $n = t_b^1, s^1, l_s^1, g_t^1, t_c^1, t_b^2, s^2, l_s^2, g_t^2, t_c^2, t_b^3, s^3, l_s^3, g_t^3, t_c^3, d^1, d^2, d^3; n_c = t_c^1, l_s^1, t_b^2, s^2, g_t^2, t_c^2, l_s^2, t_b^3, s^3, g_t^3, t_c^3$

$t_c^3; n_{t_1} = t_c^1, t_b^1, s^1, g_t^1, l_s^1; n_{t_2} = t_c^2, g_t^2, l_s^2; n_{t_3} = t_c^3, t_b^3, s^3, g_t^3, l_s^3; n_{b_1} = t_b^1, s^1$ 且 $n_{b_2} = t_b^3, s^3$, 其中各子进程的定义将逐步给出。

鉴于表1名字的定义, C的进程定义为

$$C(n_1) \triangleq i_n(m_1). C_1 + \text{await}(\overline{l_s^1} \langle t_b^2, s^2 \rangle. C_2) + \text{await}(t_c^1(m_2). [m_2 = r_{eq}] C_3) + (C_4 | C_5 | C_6)$$

$$C_1 \triangleq \text{await}(\overline{t_c^1} \langle m_1 \rangle. C \langle n_1 \rangle)$$

$$C_2 \triangleq s_{kip}^2. C_7$$

$$C_3 \triangleq \text{await}(\overline{t_c^3} \langle m_2 \rangle. C_8)$$

$$C_4 \triangleq \text{await}(t_c^1(m_4). [m_4 = l_m^1] C_9)$$

$$C_5 \triangleq \text{await}(t_c^3(m_5). [m_5 = l_m^3] C_{10})$$

$$C_6 \triangleq \text{await}(d^1. C_{11})$$

$$C_7 \triangleq \text{await}(\overline{g_t^2} \langle t_b^2, s^2 \rangle. C_{12})$$

$$C_8 \triangleq \text{await}(t_c^3(m_3, l_1). [m_3 = a_{ck}] C_{13})$$

$$C_9 \triangleq \text{await}(\overline{d^1})$$

$$C_{10} \triangleq \text{await}(\overline{d^3})$$

$$C_{11} \triangleq \text{await}(d^3. C_{14})$$

$$C_{12} \triangleq s_{kip}^2. C \langle n_2 \rangle$$

$$C_{13} \triangleq \text{await}(\overline{t_c^1} \langle m_3, l_1 \rangle. C \langle n_1 \rangle)$$

$$C_{14} \triangleq \overline{m_s}. C_{15}$$

$$C_{15} \triangleq s_{kip}^3. (C_{16} | C_{17})$$

$$C_{16} \triangleq \text{await}(\overline{t_c^1} \langle g_m \rangle)$$

$$C_{17} \triangleq \text{await}(\overline{t_c^3} \langle g_m \rangle. C \langle n_1 \rangle)$$

式中: $n_1 = t_c^1, l_s^1, t_b^2, s^2, g_t^2, t_c^2, l_s^2, t_b^1, s^1, g_t^1, t_c^3; n_2 = t_c^2, l_s^2, t_b^1, s^1, g_t^1, t_c^1, l_s^1, t_b^2, s^2, g_t^2, t_c^3$ 。

在上述C的定义中, 当系统进行控制信号的接收和发送时, C先通过 $i_n(m_1)$ 接收来自外部的控制信号, 然后将接收的消息由 t_c^1 发送给 T^1 ; 当进行连接站点的切换时, C先通过 $\overline{l_s^1} \langle t_b^2, s^2 \rangle$ 向 T^1 发送断开连接的信息, 再向 IDT^2 发送建立连接的信息 $\overline{g_t^2} \langle t_b^2, s^2 \rangle$, 即完成了站点切换; 在路况监测和拥塞广播方面, C先通过 t_c^1 和 t_c^3 接收来自各活动站点的局部信息并检测接收的信息, 即 $[m_4 = l_m^1]$ 和 $[m_5 = l_m^3]$, 继而采用 m_s 对信息进行处理, 再将全局拥塞信息 g_m 由 t_c^1 和 t_c^3 通过各站点广播到相应车辆; 在建立车辆间的连接方面, C先等待来自 T^1 的申请信息 $t_c^1(m_2)$, 继而将申请发送给 T^3 , 再接收来自 T^3 的应答, 在确认后将其反馈给 T^1 , 相应过程消耗的时间和同步等待由 s_{kip} 和 await 建模。

T^1 与 T^2 的定义类似, 差别在于所涉及的具体通道名的标号不同, 故在此只给出 T^1 和 T^3 的具体定义

$$\begin{aligned} T^1(n_3) &\triangleq \text{await}(t_c^1(m_6). [m_6 = c_{md}] T_1^1) + \\ &\overline{b_t^1}. T_2^1 + \text{await}(t_b^1(m_7). [m_7 = r_{eq}] T_3^1) + j_m^1. T_4^1 \\ T_1^1 &\triangleq \text{await}(\overline{t_b^1} \langle m_6 \rangle. T^1 \langle n_3 \rangle) \\ T_2^1 &\triangleq \text{await}(l_s^1(t, s). T_5^1) \\ T_3^1 &\triangleq \text{await}(\overline{t_c^1} \langle m_7 \rangle. T_6^1) \\ T_4^1 &\triangleq s_{kip}^1. T_7^1 \\ T_5^1 &\triangleq s_{kip}^2. T_8^1 \\ T_6^1 &\triangleq \text{await}(t_c^1(m_8, l_2). [m_8 = a_{ck}] T_9^1) \\ T_7^1 &\triangleq \text{await}(\overline{t_c^1} \langle l_m^1 \rangle. T_{10}^1) \\ T_8^1 &\triangleq \text{await}(s^1 \langle t, s \rangle. T_{11}^1) \\ T_9^1 &\triangleq \text{await}(\overline{t_b^1} \langle m_8, l_2 \rangle. T^1 \langle n_3 \rangle) \\ T_{10}^1 &\triangleq \text{await}(t_c^1(m_9). [m_9 = g_m] T_{12}^1) \\ T_{11}^1 &\triangleq s_{kip}^1. IDT^1 \langle n_4 \rangle + s_{kip}^2. IDT^1 \langle n_4 \rangle \\ T_{12}^1 &\triangleq \text{await}(\overline{t_b^1} \langle m_9 \rangle. T^1 \langle n_3 \rangle) \\ T^3(n_5) &\triangleq \text{await}(t_c^3(m_{10}). ([m_{10} = c_{md}] T_1^3 + \\ &[m_{10} = r_{eq}] T_2^3)) + \text{await}(l_s^3(t, s). T_3^3) + j_m^3. T_4^3 \\ T_1^3 &\triangleq \text{await}(\overline{t_b^3} \langle m_{10} \rangle. T^3 \langle n_5 \rangle) \\ T_2^3 &\triangleq \text{await}(\overline{t_b^3} \langle m_{10} \rangle. T_5^3) \\ T_3^3 &\triangleq s_{kip}^2. T_6^3 \\ T_4^3 &\triangleq s_{kip}^1. T_7^3 \\ T_5^3 &\triangleq \text{await}(t_b^3(m_{11}, l_3). [m_{11} = a_{ck}] T_8^3) \\ T_6^3 &\triangleq \text{await}(s^3 \langle t, s \rangle. T_9^3) \\ T_7^3 &\triangleq \text{await}(\overline{t_c^3} \langle l_m^3 \rangle. T_{10}^3) \\ T_8^3 &\triangleq \text{await}(\overline{t_c^3} \langle m_{11}, l_3 \rangle. T^3 \langle n_5 \rangle) \\ T_9^3 &\triangleq s_{kip}^1. IDT^3 \langle n_6 \rangle + s_{kip}^2. IDT^3 \langle n_6 \rangle \\ T_{10}^3 &\triangleq \text{await}(t_c^3(m_{12}). [m_{12} = g_m] T_{11}^3) \\ T_{11}^3 &\triangleq \text{await}(\overline{t_b^3} \langle m_{12} \rangle. T^3 \langle n_5 \rangle) \end{aligned}$$

式中: $n_3 = t_c^1, t_b^1, s^1, g_t^1, l_s^1; n_4 = t_c^1, g_t^1, l_s^1; n_5 = t_c^3, t_b^3, s^3, g_t^3, l_s^3; n_6 = t_c^3, g_t^3, l_s^3$ 。

在上述 T^i 的定义中, 当 T^i 通过 t_c^i 接收来自C的控制信息后将其由 t_b^i 转发给与之相连的 B^i ; 当通过 l_s^i 接收到C的断开信息后将其通过 s^i 发送给 B^i , 至此 T^i 断开与 B^i 的连接, 切换到 IDT^i ; 在监测拥塞信息时, T^i 通过 j_m^i 检测到拥塞后将其由 t_c^i 发送给C并接收来自C的全局信息继而发送给 B^i ; 在建立车辆间连接时, T^1 先接收 B^1 的申请, 继而将其发送给C, 再接收C的应答并反馈回 B^1 , T^3 则先通过 t_c^3 接收C的申请, 再发送给 B^2 , 继而接收 B^2 的应答并反馈回C。

IDT^i 的进程定义如下

$$IDT^i(n_7) \triangleq \text{await}(g_t^i(t, s). IDT_1^i)$$

$$IDT_1^i \triangleq s_{kip}^2. T^i \langle n_8 \rangle$$

式中: $\mathbf{n}_7 = t_c^i, g_t^i, l_s^i; \mathbf{n}_8 = t_c^i, t, s, g_t^i, l_s^i$ 。

在上述 IDT^i 的定义中, IDT^i 通过 $g_t^i(t, s)$ 接收到来自 C 的建立连接的信息后切换到 T^i 。

B^j 的进程定义如下

$$\begin{aligned} B^1(\mathbf{n}_9) &\triangleq \text{await}(t_b^1(m_{13})). ([m_{13} = c_{\text{md}}] B_1^1 + \\ &[m_{13} = g_m] B^1\langle \mathbf{n}_9 \rangle) + \text{await}(s^1(t, s), B_2^1) + \\ &\text{await}(\overline{t_b^1}\langle r_{\text{eq}} \rangle, B_3^1) \\ B_1^1 &\triangleq \overline{o_{\text{ut}}}\langle m_{13} \rangle, B^1\langle \mathbf{n}_9 \rangle \\ B_2^1 &\triangleq s_{\text{kip}}, B^1\langle t, s \rangle + s_{\text{kip}}^2, B^1\langle t, s \rangle \\ B_3^1 &\triangleq \text{await}(t_b^1(m_{14}), l_4)[m_{14} = a_{\text{ck}}] B_4^1 + s_{\text{kip}}^5, \tau, B^1\langle \mathbf{n}_9 \rangle \\ B_4^1 &\triangleq \overline{c_n}\langle l_4 \rangle, B^1\langle \mathbf{n}_9 \rangle \\ B^2(\mathbf{n}_{10}) &\triangleq \text{await}(t_b^2(m_{15})). ([m_{15} = c_{\text{md}}] B_1^2 + \\ &[m_{15} = g_m] B^2\langle \mathbf{n}_{10} \rangle + [m_{15} = r_{\text{eq}}] B_2^2) + \\ &\text{await}(s^3(t, s), B_3^2) \\ B_1^2 &\triangleq \overline{o_{\text{ut}}}\langle m_{15} \rangle, B^2\langle \mathbf{n}_{10} \rangle \\ B_2^2 &\triangleq \overline{t_b^2}\langle a_{\text{ck}}, t_{\text{bb}} \rangle, B^2\langle \mathbf{n}_{10} \rangle \\ B_3^2 &\triangleq s_{\text{kip}}, B^2\langle t, s \rangle + s_{\text{kip}}^2, B^2\langle t, s \rangle \end{aligned}$$

式中: $\mathbf{n}_9 = t_b^1, s^1; \mathbf{n}_{10} = t_b^3, s^3$ 。

在上述 B^j 的定义中, 当 B^j 通过 t_b^i 接收到 T^i 的消息后, 若是控制信号, 即通过 o_{ut} 执行该信号, 若是全局拥塞信息, 则继续监听; 当通过 s^i 接收到来自 T^i 的切换信息时, B^j 将变成 $B^j\langle t, s \rangle$, 其中 t, s 为与新站点通信的通道; 在建立车辆间连接时, B^1 先发出申请 r_{eq} 并等待应答, 继而通过 c_n 执行成功建立连接的操作, 等待超过 5 个时间单元就做超时处理, B^2 则是等待申请, 继而确认信息后发送应答 a_{ck} 。

3 时间相关移动并发系统推演实例

为便于推演进程, 首先介绍可观察动作 π_o 和操作规则。

$$\pi_o ::= x(\mathbf{y}) \mid \overline{x}\langle \mathbf{y} \rangle \mid \overline{x}(\mathbf{z}) \mid \tau \mid I_P \mid s_{\text{kip}}$$

$$\pi_c ::= x(\mathbf{y}) \mid \overline{x}\langle \mathbf{y} \rangle \mid \overline{x}(\mathbf{z}) \mid \tau$$

$$\pi_t ::= I_P \mid s_{\text{kip}}$$

式中: x 为自由名; $\mathbf{y}$ 和 $\mathbf{z}$ 为受限名; $\overline{x}(\mathbf{z})$ 即 $\nu \mathbf{z} x(\mathbf{z})$ 。

p - π 的操作规则由形如 $P \xrightarrow{\pi_o} Q$ 的迁移构成, 表示 P 可通过执行动作 π_o 演化为 Q , 规则如下。

$$\begin{aligned} \text{Tau}: & \frac{}{\tau, P \xrightarrow{\tau} P} & \text{Out}: & \frac{}{\overline{x}\langle \mathbf{y} \rangle, P \xrightarrow{x(\mathbf{y})} P} \\ \text{Act}_t: & \frac{}{\pi_t, P \xrightarrow{\pi_t} P} & \text{Act}_\epsilon: & \frac{P \xrightarrow{\pi_o} P'}{\epsilon, P \xrightarrow{\pi_o} P'} \\ \text{Sum}: & \frac{P_1 \xrightarrow{\pi_c} P'_1}{P_1 + P_2 \xrightarrow{\pi_c} P'_1} & \text{Close}: & \frac{P_1 \xrightarrow{x(\mathbf{y})} P'_1 P_2 \xrightarrow{x(\mathbf{y})} P'_2}{P_1 \mid P_2 \xrightarrow{\tau} \nu \mathbf{y} (P'_1 \mid P'_2)} \end{aligned}$$

$$\begin{aligned} \text{Com}_{t1}: & \frac{P_1 \xrightarrow{\pi_t} P'_1 P_2 \equiv 0}{P_1 \mid P_2 \xrightarrow{\pi_t} P'_1} \\ \text{Sum}_{t1}: & \frac{P_1 \xrightarrow{s_{\text{kip}}} P'_1 P_2 \xrightarrow{s_{\text{kip}}} P'_2}{P_1 + P_2 \xrightarrow{s_{\text{kip}}} P'_1 + P'_2} \\ \text{Com}: & \frac{P_1 \xrightarrow{x(\mathbf{y})} P'_1 P_2 \xrightarrow{x(\mathbf{z})} P'_2}{P_1 \mid P_2 \xrightarrow{\tau} P'_1 \mid \{ \mathbf{y} / \mathbf{z} \} P'_2} \\ \text{Res}: & \frac{P \xrightarrow{\pi_o} P'}{\nu x P \xrightarrow{\pi_o} \nu x P'} (x \notin n(\pi_o)) \\ \text{In}: & \frac{}{x(\mathbf{z}), P \xrightarrow{x(\mathbf{y})} \{ \mathbf{y} / \mathbf{z} \} P} (\mathbf{y} \notin f_n((\mathbf{z}) P)) \\ \text{Par}: & \frac{P_1 \xrightarrow{\pi_c} P'_1}{P_1 \mid P_2 \xrightarrow{\pi_c} P'_1 \mid P_2} (b_n(\pi_c) \cap f_n(P_2) = \emptyset) \\ \text{Mat}: & \frac{P \xrightarrow{\pi_o} P'}{[a_1 = a_2] P \xrightarrow{\pi_o} P'} (a_1 = a_2) \\ \text{Open}: & \frac{P \xrightarrow{x(\mathbf{y})} P'}{\nu \mathbf{y} P \xrightarrow{x(\mathbf{z})} \{ \mathbf{z} / \mathbf{y} \} P'} (x \notin \mathbf{y}, \mathbf{z} \notin f_n((\mathbf{y}) P')) \\ \text{Ide}: & \frac{\{ a / x \} P_A \xrightarrow{\pi_o} P'}{A \langle a \rangle \xrightarrow{\pi_o} P'} (A(x) \triangleq P_A) \\ \text{Await}: & \frac{\epsilon, P + s_{\text{kip}}, \text{await}(P) \xrightarrow{\pi_o} P'}{\text{await}(P) \xrightarrow{\pi_o} P'} \\ \text{Sum}_{t2}: & \frac{P_1 \xrightarrow{\pi_t} P'_1}{P_1 + P_2 \xrightarrow{\pi_t} P'_1} (P_1 + P_2 \not\xrightarrow{s_{\text{kip}}}) \\ \text{Com}_{t2}: & \frac{P_1 \xrightarrow{\pi_{t1}} P'_1 P_2 \xrightarrow{\pi_{t2}} P'_2}{P_1 \mid P_2 \xrightarrow{\pi_{t1} \cup \pi_{t2}} P'_1 \mid P'_2} \end{aligned}$$

$$(\pi_{t1} \cap \pi_{t2} - s_{\text{kip}} = \emptyset, P_1 \mid P_2 \not\xrightarrow{\tau})$$

部分规则与文献[6]相同, 此处主要关注扩展的 7 条规则: Act_t , Sum_{t1} , Act_ϵ , await , Com_{t1} , Sum_{t2} 和 Com_{t2} 。规则 Act_t 是处理区间动作前缀的公理。规则 Sum_{t1} 用于确定消耗单位时间后进程的执行结果。在 π 演算中, 操作符 $+$ 的执行是非确定的, s_{kip} 的引入导致了这种执行的变化, 故引入 Sum_{t1} 。规则 Act_ϵ 和 Await 分别用于处理 ϵ 和 await 。规则 Com_{t1} 用于处在不同时间点结束的并发进程的执行。规则 Sum_{t2} 与 Sum 类似, 条件 $P_1 + P_2 \not\xrightarrow{s_{\text{kip}}}$ 用来区别 Sum_{t1} 。规则 Com_{t2} 用于带区间动作前缀的并行进程的执行, 条件 $\pi_{t1} \cap \pi_{t2} - s_{\text{kip}} = \emptyset$ 用于避免冲突。

3.1 基于操作规则的系统迁移

依据操作规则, 可得到系统的时间相关标记迁移系统 TDLTS 的定义。

定义 3 对于任意的 p - π 进程 P , P 的 TDLTS 由四

元组 $(\mathcal{P}, P_0, \mathcal{L}, \mathcal{T})$ 表示,其中 $\mathcal{P}$ 是 $p-\pi$ 进程集合, P_0 是初始进程, $\mathcal{L}$ 是可观察动作集合, $\mathcal{T}$ 是迁移集合。元组构造如下:

(1) $P_0 = P$ 且 $P_0 \in \mathcal{P}$;

(2) 对于任意的 $Q \in \mathcal{P}$, 若 $Q \xrightarrow{\pi_0} Q'$, 则 $Q' \in \mathcal{P}$, $\pi_0 \in \mathcal{L}$ 且 $(Q, \pi_0, Q') \in \mathcal{T}$;

(3) 有限次地应用(1)和(2),生成 P 的TDLTS。

MVCS的TDLTS如图2所示,该图部分描述了系统的时间相关迁移过程,其中 $P_i (0 \leq i \leq 57)$ 是形如 $\nu n \langle C \langle n_c \rangle \mid T^1 \langle n_{t_1} \rangle \mid \text{IDT}^2 \langle n_{t_2} \rangle \mid T^3 \langle n_{t_3} \rangle \mid B^1 \langle n_{b_1} \rangle \mid B^2 \langle n_{b_2} \rangle \rangle$ 的进程。

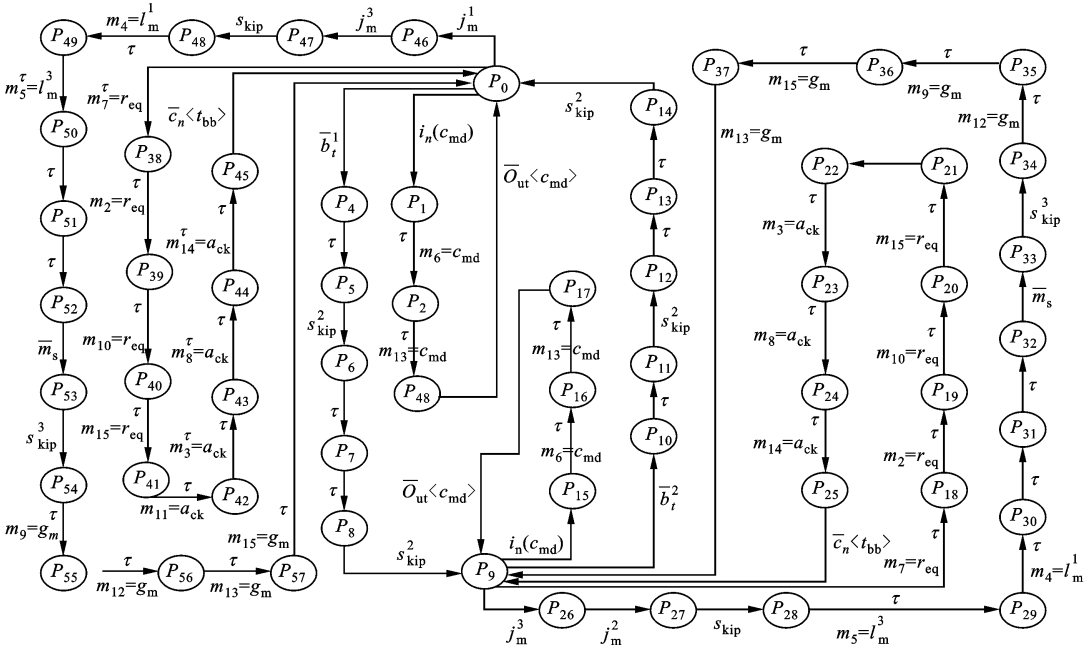


图2 MVCS的部分TDLTS

表示取该动作序列的第 i 个时间单元中的动作, $L\sigma^{-1}(\pi_0)$ 表示动作 π_0 对应的区间时间,即该动作是在第几个时间单元发生的。

由于TDLTS的四元组定义中不包括可接受进程,即终止进程,考虑到系统推演的可终止性,本文确定终止进程集合为 $\{0, P_0\}$,即可接受的执行路径或到达0,或经过循环过程到达初始进程 P_0 ,考察的可接受的有穷和无穷路径定义如下。

定义4 若路径 $\omega: P_0 \xrightarrow{\pi_{01}} P_1 \xrightarrow{\pi_{02}} \dots \xrightarrow{\pi_{0i}} P_i \xrightarrow{\pi_{0i+1}} 0$ 且 $P_0 \neq P_1 \neq \dots \neq P_i$,则称该路径 ω 为可接受的有穷执行路径。

定义5 若路径 $\omega: P_0 \xrightarrow{\pi_{01}} P_1 \xrightarrow{\pi_{02}} \dots \xrightarrow{\pi_{0i}} P_i \xrightarrow{\pi_{0i+1}} 0$ 且 $\exists P_j, P_k$,使得 $P_j = P_k, 0 \leq j < k \leq i$ 或者若 $\omega: P_0 \xrightarrow{\pi_{01}} P_1 \xrightarrow{\pi_{02}} \dots \xrightarrow{\pi_{0i}} P_i$ 且 $P_0 = P_i$,则称路径 ω 为可

系统的执行路径 ω 是形如 $P_0 \xrightarrow{\pi_{01}} P_1 \xrightarrow{\pi_{02}} \dots \xrightarrow{\pi_{0i}} P_i$ 的链,其中 $\pi_{0j} (1 \leq j \leq i)$ 为可观察动作, $P_j (0 \leq j \leq i)$ 为进程,该链执行的迁移动作序列 σ 为 $\pi_{01}, \pi_{02}, \dots, \pi_{0i}, \sigma(i)$ 表示取该动作序列的第 i 个元素, $|\sigma|$ 表示该序列中迁移动作的个数。对于有穷路径,该链中动作的变迁次数是有限的,即 $|\sigma| \in \mathbb{N}$;对于无穷路径,该链中动作的变迁次数是无穷的,即 $|\sigma| = \infty$ 。该迁移动作序列的区间长度 $L(\sigma)$ 为该动作序列中区间动作的变迁次数,即占用的总的时间单元。对于区间有穷路径, $L(\sigma) \in \mathbb{N}$;对于区间无穷路径,该路径的区间长度 $L(\sigma) = \infty$ 。在此令 $L\sigma(i)$

接受的无穷执行路径。

3.2 基于系统迁移的系统推演

本小节将通过系统的TDLTS和可接受执行路径来分析系统的行为和期望性质。本文关注活性、安全性、时间约束性、及时性以及移动性,其中:活性指在一定条件下系统期望的行为终究会发生;安全性指在任何条件下系统避免的行为都不会发生^[9];时间约束性指系统行为的发生要满足一定时间约束;及时性指进程能发生瞬时动作时,优先选择瞬时动作,而丢弃区间动作^[7];移动性指系统拓扑结构的变化性^[5]。

对于该系统,进行以下推演。

(1)活性:系统在接收到控制信号后一定会执行该信号,具体描述如下,对于TDLTS中的可接受执行路径,若该路径包含 $i_n(c_{md})$,则它一定包含

$\overline{o_{ut}}\langle c_{md} \rangle$ 。图2中对于包含 $i_n(c_{md})$ 的路径 $\omega_0: P_0$
 $\xrightarrow{i_n(c_{md})} P_1 \xrightarrow{\tau} P_2 \xrightarrow{\tau} P_3 \xrightarrow{\overline{o_{ut}}\langle c_{md} \rangle} P_0$ 和路径 $\omega_1: P_0$
 $\xrightarrow{\overline{b_t^1}} P_4 \xrightarrow{\tau} P_5 \xrightarrow{s_{kip}^2} P_6 \xrightarrow{\tau} P_7 \xrightarrow{\tau} P_8 \xrightarrow{s_{kip}^2} P_9$
 $\xrightarrow{i_n(c_{md})} P_{15} \xrightarrow{\tau} P_{16} \xrightarrow{\tau} P_{17} \xrightarrow{\overline{o_{ut}}\langle c_{md} \rangle} P_9 \xrightarrow{\overline{b_t^2}} P_{10} \xrightarrow{\tau} P_{11}$
 $\xrightarrow{s_{kip}^2} P_{12} \xrightarrow{\tau} P_{13} \xrightarrow{\tau} P_{14} \xrightarrow{s_{kip}^2} P_0$, 这两条路径都包含
 $\overline{o_{ut}}\langle c_{md} \rangle$, 故系统满足该性质。

(2) 安全性: 对于 TDLTS 中的所有进程, 除了 0 进程外不存在不能进行迁移的进程, 即死进程。遍历 TDLTS 的所有进程, 可得到系统中的进程都存在可迁移的执行动作, 故系统满足该性质。

(3) 时间约束性: 从站点监测到局部拥塞信息到控制中心开始进行全局拥塞信息的处理不超过 2 个时间单元, 对于 TDLTS 的任意可接受执行路径, 该路径的 $L\sigma^{-1}(\overline{m_s}) - L\sigma^{-1}(j_m^i) \leq 2$ 。通过对图2中包含 $\overline{m_s}$ 和 j_m^i 的路径的计算得 $L\sigma^{-1}(\overline{m_s}) - L\sigma^{-1}(j_m^i) = 1$, 故系统满足该性质。

(4) 及时性: 对于 TDLTS 的所有进程, 若该进程能够发生瞬时动作, 将优先执行瞬时动作, 而不会发生区间动作, 直到所有可执行的瞬时动作都执行完才执行区间动作; 遍历 TDLTS 的所有进程可得, 不存在进程使得该进程可执行的迁移既包括区间动作, 又包括瞬时动作, 故系统满足该性质。

(5) 移动性: 由于 MVCS 中存在两个站点动态与 B^1 连接, 移动性表现为 B^1 先处于与 T^1 连接然后切换到与 T^2 连接, 或者 B^1 先与 T^2 连接继而切换到 T^1 ; 建模时采用 b_t^i 表示 B^1 与 T^i 处于连接, 则该性质可描述如下, 对于 TDLTS 的可接受执行路径, 若该路径包括两个 b_t^i , 则这两个 b_t^i 之间一定包含一个 b_t^2 , 即不存在可接收执行路径使得 b_t^1 重复发生而 b_t^2 不出现, 通过对图3中可接受执行路径的考察, 可得系统满足该性质。

4 结 论

针对时间相关移动并发系统需要同时对系统交互行为和与时间相关行为进行建模和分析的需求, 本文提出了一种采用扩展 π 演算 $p\text{-}\pi$ 对时间相关移动并发系统建模和推演的方法, 并结合移动车辆控制系统实例进行了说明。结果表明: 采用区间动作前缀将时间与动作前缀绑定的方法来对时间相关行为建模, 从理论上来说, 更符合进程代数的演算范畴, 故使得部分适用于瞬时动作的规则可在区间动作上实现复用, 因而便于系统推演; 在对时间相关行为建

模时, 并未引入额外操作符, 而只扩展了动作前缀, 使得基于 π 演算的丰富的代数理论经过细微的更改便可适用于时间相关移动并发系统的推演; 在对系统行为推演时, 本文不仅对常用的安全性和活性等性质进行了推演, 还对时间约束性、及时性以及移动性等重要但一般不易于推演的性质进行了推演。因此, 该方法在对时间相关移动并发系统建模和推演方面是可行、有效的, 保证了时间相关移动并发系统的可靠性。

参考文献:

- [1] KIM W, LEE J H, HONG C, et al. An innovative method for data and software integration in SaaS [J]. Computers & Mathematics with Applications, 2012, 64(5): 1252-1258.
- [2] SUBASHINI S, KAVITHA V. A survey on security issues in service delivery models of cloud computing [J]. Journal of Network and Computer Applications, 2011, 34(1): 1-11.
- [3] BERGER M F. Towards abstractions for distributed systems [D]. London, UK: University of Imperial College, 2002.
- [4] 王小兵, 段振华. 面向扩展投影时序逻辑的 Web 服务模型检测 [J]. 西安交通大学学报, 2009, 43(4): 39-44. WANG Xiaobing, DUAN Zhenhua. Projection temporal logic oriented model checking for web services [J]. Journal of Xi'an Jiaotong University, 2009, 43(4): 39-44.
- [5] MILNER R. Communicating and mobile systems: the π -calculus [M]. Cambridge, UK: Cambridge University Press, 1999: 87-112.
- [6] MILNER R, PARROW J, WALKER D. A calculus of mobile processes [J]. Journal of Information and Computation, 1992, 100(9): 41-77.
- [7] LEE J, ZIC J. On modeling real-time mobile processes [C] // Proceedings of 25th Australian Computer Science Communications. Piscataway, NJ, USA: IEEE, 2002: 139-147.
- [8] LUO Ling, DUAN Zhenhua. An extended π -calculus [C] // Proceedings of 2th International Conference on Computers, Networks, Systems and Industrial Applications. Sandy Bay, Tasmania, Australia: SERSC, 2012: 632-637.
- [9] ALPERN B, SCHNEIDER F B. Recognizing safety and liveness [J]. Distributed Computing, 1987, 2(3): 117-126.

(编辑 赵炜)