

移动网可达树

张瑞华 杨 茹 丁志军

(同济大学嵌入式系统与服务计算教育部重点实验室 上海 201804)

摘 要 随着计算机技术和网络通信技术的高速发展,对于并发分布式系统,已经提出了进程代数以及 Petri 网等形式化分析方法。近年来由于移动互联网的出现和快速发展,通过在进程代数中增加移动性得到了 π 演算,与此同时, Petri 网领域也采用谓词/变迁网、颜色网等构建移动系统模型。但它们仍存在一些不足之处。在此基础上, A. Asperti 和 N. Busi 提出了移动网这一系统模型。移动网是在 Petri 网的基础上增加了移动性并结合了进程代数的优势得到的,适用于描述和刻画移动计算系统。然而,目前并没有对于移动网相应分析方法的研究。为此开展了移动网模型分析方法的研究,给出了移动网可达树的构造算法,提供了移动网模型可达性分析方法,并对移动车辆电话通信系统实例进行了分析。

关键词 移动网,移动系统,可达树,算法

中图法分类号 TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.11.010

Reachability Tree of Mobile Net

ZHANG Rui-hua YANG Ru DING Zhi-jun

(The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China)

Abstract With the fast development of computer technology and network communication technology, process algebra, Petri net and other formal analysis methods were proposed for concurrent distributed systems. Due to the emergence and rapid development of the mobile internet in recent years, π calculus is developed by adding mobility to process algebra, and at the same time, Petri nets field also uses predicate/transition net and color net to model the mobile system. But there still exist some weakness of them. Based on all previous works, A. Asperti and N. Busi proposed mobile net. Mobile net is obtained by adding mobility to Petri net and combining with the strengths of process algebra. It is suitable for the description of mobile computing system. However, no corresponding analysis method of mobile net has been developed now. Hence, this paper studied the analysis method for mobile net. The construct algorithm of the reachability tree of mobile net was proposed, and the reachability analysis method based on its reachability tree was provided. In addition, an example of the mobile vehicle telephone communication system was illustrated to demonstrate the effectiveness of this method.

Keywords Mobile net, Mobile system, Reachability tree, Algorithm

1 引言

近年来,随着计算机技术和网络通信技术的高速发展,以并发性、移动性、分布性、实时性、异构性和互操作性等为主要特征的并发分布式系统得到人们的普遍关注。由于并发分布式系统本身非常复杂,在动态网络环境中构建相关并发系统不仅难度大、周期长、效率低,且难以发现其中隐含的缺陷和错误。

为了解决这些问题,进程代数、Petri 网等形式化方法应运而生。形式化方法以严格的数学理论为支撑,被公认为是一种行之有效的减少设计错误、提高系统可信度的重要途径。

以通信系统演算 CCS^[3] 为代表的进程代数由于概念简

洁,可用的数学工具丰富并且可以描述拓扑结构动态变化的移动通信系统,因此在并发系统的建模和推演中获得了广泛的应用。

在进程代数中,并发系统的各个并发分量都是进程,相互间可以通过输入/输出动作实现通信。使用通信,而不是共享存储,作为进程间相互作用的基本手段,表现出面向分布式系统的特征。在语法上,用一组算子作进程构件。算子语义用结构化操作语义方法定义。进程可看成标号迁移系统。把并发性归结为非确定性,将并发执行的进程的行为看成是各个进程的行为的所有可能的交错合成,即所谓的交错式语义。

Petri 网^[12]是另外一种并发系统描述和分析的工具。它特别便于描述系统中进程或部件的顺序、并发以及冲突等关

到稿日期:2015-08-12 返修日期:2016-03-10 本文受国家自然科学基金项目(61173042,61472004),上海市产学研合作项目(沪 CXY-2013-40)资助。

张瑞华(1994—),女,硕士,主要研究方向为 Petri 网理论与应用;杨 茹(1992—),女,博士,主要研究方向为 Petri 网;丁志军(1974—),男,博士,教授,主要研究方向为 Petri 网、服务计算。

系。作为一种系统模型, Petri 网不仅可以刻画系统的结构, 而且可以描述系统的动态行为, 其既有直观的图形表示, 又可以引入许多数学方法对其性质进行分析。作为一种图形化的工具, 它能可视化地模拟通信, 类似于流程图、结构图和网络等。其中的托肯(token)可用于模拟系统的动态和并发的活动。作为一种数学化的工具, 可以对它建立状态方程、代数方程和其他数学模型, 来仿真和控制系统的行为。

在实际系统的应用中, Petri 网通常用库所表示条件, 用变迁表示事件。变迁有若干个输入和输出库所, 分别表示相应事件的前提条件和结果。条件的成立与否由用相应位置中的 token 存在与否表示。

但它们在利用自身优势解决相应问题时, 仍存在一些不足之处: 进程代数和 Petri 网模型均无层次结构, 不能进行抽象和折叠。因此当它们所要描述的进程状态数过大时将面临“状态爆炸”问题, 其描述十分繁琐、复杂、难以理解。

近几年, 随着移动互联网的迅速发展和成熟, 移动互联网应用正深刻影响着人们的各个方面, 移动系统也引起了人们的普遍关注。为了描述移动系统, 进程代数进一步发展了 pi 演算^[10]。pi 演算是在 CCS 的基础上发展起来的, 它具有更强大的表达能力, 而其本身的实体却比 CCS 更少。其中 pi 演算通过进程之间的连接来表示移动性。与此同时, Petri 网领域也采用谓词/变迁网、颜色网^[12]等构建移动系统模型。

在此基础上, A. Asperti 和 N. Busi 提出了移动网^[1]这一系统模型。移动网是在 Petri 网的基础上增加移动性^[7], 并结合进程代数的优势得到的。移动网功能强大且结构相对简单, 同时能够很好地刻画和描述系统的移动性, 但是目前并没有发展相应的分析方法。为此, 本文开展了移动网模型分析方法的研究, 给出了移动网的可达树的构造算法, 提供了移动网模型可达性分析方法, 并以实例进行了解释。

本文第 2 节介绍了 Petri 网和移动网的基本概念和相关术语, 同时给出了一个移动车辆电话通信系统的移动网模型; 第 3 节研究了移动网可达性的分析方法, 给出了可达树的构造算法; 第 4 节根据算法给出了移动车辆电话通信系统模型的可达树; 最后进行总结并对未来的工作做了展望。

2 移动网

本节首先介绍了 Petri 网的基本概念, 进而给出移动网的定义, 最后给出了一个移动车辆电话通信系统的移动网模型。

2.1 Petri 网

首先给出库所无容量限制的库所/变迁网(P/T 网)的定义^[9]。

定义 1 给定一个集合 X , X 上的一个多重集是一个函数 $m: X \rightarrow (N \cup \{\omega\})$ 。 X 上所有多重集的集合用 M_X 来表示。令 $dom(m) = \{x \in X \mid m(x) > 0\}$ 。对于一个多重集 m , 如果对于任意 $x \in X$ 都有 $m(x) = 0$, 则称 m 为空。令:

$$M_X^{post} = \{m \in M_X \mid dom(m) \text{ 是有限的}\}$$

$$M_X^{pre} = \{m \in M_X^{post} \mid m \text{ 是非空的} \wedge \forall x \in X, m(x) \in N\}$$

令 $i < \omega$, 并且对于任意 $i \in N, i + \omega = \omega + i = \omega + \omega = \omega$ 。若对于所有的 $x \in X$ 都有 $m(x) \leq m'(x)$, 则写作 $m \leq m'$ 。

操作符 $\oplus$ 表示多重集的并集: $(m \oplus m')(x) = m(x) + m'(x)$ 。

操作符 $\setminus$ 表示多重集的差: $(m \setminus m')(x) = m(x) \ominus m'(x)$, 其

中 $\ominus$ 是自然数上的一个局部操作符, 定义为: 如果 $i > j, i \ominus j = i - j$; 否则 $i \ominus j = 0$ 。同时 $\omega \ominus i = \omega$ 。

令 X 是名字的一个可数集合, 它将会用来表示网中的库所, 它的元素包括 x, y 等。

定义 2 令 $N = (\nu Y)(T, m_0)$, 其中 $Y \subseteq X$ 是库所的集合, $T \subseteq M_X^{pre} \times M_X^{post}$ 是变迁的集合, $m_0 \in M_X^{pre}$ 是初始标识。如果所有出现在初始标识和变迁的名字都被包含在库所的集合中, 则称 N 是一个 P/T 网, 即: $dom(m_0) \cup \bigcup_{(c,p) \in T} (dom(c) \cup dom(p)) = Y$ 。

M_X 的一个元素称为一个标识。给定一个标识 m 和一个库所 x , 我们说库所 x 包含 $m(x)$ 个托肯。一个变迁 $t = (c, p)$ 可以记作 $c \triangleright p$ 的形式, 其中 c 称作 t 的前集, 代表变迁发生所需的托肯; p 称作 t 的后集, 代表变迁发生得到的托肯。

令 $t = c \triangleright p$ 是一个变迁: 如果 $c \leq m$, 则变迁 t 在标识 m 下使能; 从标识 m 引发变迁 t 得到新标识 m' , 其中 $m' = (m \setminus c) \oplus p$, 可以写作 $m[t > m']$ 。

2.2 移动网

从 Petri 网的观点来看, 可以把托肯当作库所中包含的名字(准确地说是名字的元组)。那么移动网中变迁的后集不是静态的, 而是取决于变迁发生所消耗的托肯的名字。

移动网具有以下特点: 1) 允许名字统一; 2) 移动网允许输入库所托肯的名字是形参, 等到变迁发生的时候再进行实例化; 3) 移动网中变迁的后集不是静态的, 而是取决于变迁发生所消耗的托肯的名字^[1]。

移动网中用名字来表示库所和名字的占位符(变量)。通常来说, 给定两个名字 a 和 b , 符号 $a(b)$ 发生在一个标识或在一个变迁的前集时, 它有不同的含义。在前面一种情况中, b 是一个实参, 而在后一种情况中, b 是一个形参。所以变迁的前集只有与当前标识的一个子集相统一时, 这个变迁才能使用。

定义 3 给定两个集合 X 和 Y , 使 $M_{X,Y} = X \rightarrow (Y \rightarrow (N \cup \omega))$ 。令:

$$dom(m) = \{(x, y) \mid m(x)(y) > 0\}$$

$$M_{X,Y}^{post} = \{m \in M_{X,Y} \mid dom(m) \text{ 是有限的}\}$$

$$M_{X,Y}^{pre} = \{m \in M_{X,Y}^{post} \mid m \text{ 是非空的} \wedge \forall x \in X, y \in Y, m(x)(y) \in N\}$$

操作符 $\oplus$ 表示多重集的并集: $(m \oplus m')(x)(y) = m(x)(y) + m'(x)(y)$ 。

操作符 $\setminus$ 表示多重集的差: $(m \setminus m')(x)(y) = m(x)(y) \ominus m'(x)(y)$ 。

托肯名字的集合定义为 X 上有限序列的集合: $C = \{(x_1, \dots, x_n) \mid n > 0 \wedge x_i \in X, i = 1, \dots, n\}$ 。

下面将使用 $\vec{x}, \vec{y}, \dots$ 来表示有限元组的名字。一个元组的长度定义为: $|(x_1, \dots, x_n)| = n$; 对第 i 个元素的选择定义为: $\pi_i(x_1, \dots, x_n) = x_i, i = 1, \dots, n$ 。当没有混淆时, 可用符号 x 替代 (x) 。给出 X 上的一个替换函数 ρ , 则名字和名字元组的替换定义如下:

$$x\rho = \begin{cases} y, & (x, y) \in \rho \\ x, & (x, y) \notin \rho \end{cases}$$

$$(x_1, \dots, x_n)\rho = (x_1\rho, \dots, x_n\rho)$$

对于任一标识 m , 其中 $m \in M_{X,C}$, 对 m 上所有名字的替换定义为:

$$(m\rho)(x)(\vec{y}) = \sum_{\substack{\rho=v\wedge\vec{y}\rho=\vec{z}}} m(v)(\vec{z}).$$

而只对托肯的名字的替换(用来对接下来的变迁的前集进行实例化)定义为:

$$(m\star_{\rho})(x)(\vec{y}) = \sum_{\vec{y}\rho=\vec{z}} m(x)(\vec{z})$$

类似地,对自由名和绑定名的定义在变迁的前集(前集)和标识(后集)下是不同的。

(1)当将 m 视为一个变迁的前集时,自由名和绑定名分别定义为:

$$fn_P(m) = \{x \mid \exists \vec{y}(m(x)(\vec{y}) > 0)\}$$

$$bn_P(m) = \{x \mid \exists z, \vec{y}, i, \pi_i(\vec{y}) = x \wedge m(z)(\vec{y}) > 0\}$$

(2)当将 m 视为一个标识时,自由名和绑定名分别定义为:

$$fn_M(m) = fn_P(m) \cup bn_P(m)$$

$$bn_M(m) = \emptyset$$

(3)对于一个变迁,前集的绑定名是和后集绑定的。变迁 t 和变迁集合 T 的自由名分别定义为:

$$fn_M(c \triangleright p) = fn_P(c) \cup (fn_M(p) \setminus bn_P(c))$$

$$fn(T) = \bigcup_{t \in T} fn(t)$$

基于上面的定义,给出移动网的定义^[1]。

定义 4 令 $N = (\nu Y)(T, m_0)$, 其中 $Y \subseteq X$ 是库所的集合, $T \subseteq M_{X,C}^{re} \times M_{X,C}^{out}$ 是变迁的集合, $m_0 \in M_{X,C}^{re}$ 是初始标识。如果所有 $fn(T) \cup fn_M(m_0) \subseteq Y$, 则 N 是一个移动网。

下面给出变迁发生的规则。

$m \in M_{X,C}$ 的一个元素称为一个标识。给出一个标识 m 、一个库所 x 和一个托肯的名字 $\vec{y}$, 则库所 x 包含 $m(x)(\vec{y})$ 个 $\vec{y}$ 名字的托肯。

令 $t = c \triangleright p$ 是一个变迁: 如果存在 $\rho \subseteq bn_P(c) \times X$ 使得 $c \star_{\rho} \subseteq m$, 则变迁 t 在 m 使能; 对标识 m 进行 ρ 替换之后引发变迁 t 产生标识 m' , 其中 $m' = (m \setminus c \star_{\rho}) \oplus p\rho$ 。可以写作 $m[t >_{\rho} m']$ 。

2.3 移动车辆电话通信系统的移动网模型

Milner 等人^[10]给出了一个移动车辆电话通信系统的例子。系统中每部车辆通过固定频率与一个发射站连接, 所有发射站都与中央控制台有固定连接。当某些事件(如车辆移动或信号衰竭)发生时, 有的车辆可能会被切换到另一个发射站。

为简单起见, 假设只有一部车辆和两个发射站。在本例中, 连接到汽车的发射站可以和汽车进行通话; 中央控制台也可以在任何时候通知发射站和汽车切断联系, 之后中央控制台会通知另一个发射站和汽车建立新的连接。连接到汽车的发射站在接收了中央控制台切断联系的消息之后会发出消息让汽车切换到新的发射站。

假定中央控制台用 *Control* (下面都简记为 *C*) 来表示, 汽车用 *Car* 来表示, 与汽车连接的发射站用 *Trans* 表示, 未与汽车连接的发射站用 *ldtrans* 表示。初始条件下, 发射站 1 和汽车连接, 则表示为 *Trans*₁; 发射站 2 和汽车没有连接, 用 *ldtrans*₂ 表示。并用 *talk*, *switch*, *lose* 和 *gain* 来表示系统主体之间存在的消息动作, 其中:

(1) *talk* 表示通话消息, *Car* 发送该消息, 与 *Trans* 进行通话。

(2) *switch* 表示切换消息, *Trans* 发送该消息, 通知 *Car* 切换发射站。

(3) *lose* 表示断开连接消息, *C* 发送该消息, 通知 *Trans* 断开和 *Car* 的连接。

(4) *gain* 表示建立连接消息, *C* 发送该消息, 通知 *ldtrans* 和 *Car* 建立新的连接。

综上, 整个系统的示意图如图 1 所示。

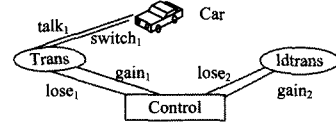


图 1 移动车辆电话通信系统

具体地, 系统中每个主体的执行过程描述如下:

(1) 中央控制台 *C* 可以发送 *lose* 消息给 *Trans* 通知 *Trans* 和 *Car* 切断联系; 之后它发送 *gain* 消息给 *ldtrans* 让 *ldtrans* 和 *Car* 建立新的连接。

(2) 发射站 *Trans* 可以接收 *Car* 发来的 *talk* 消息和 *Car* 进行通信; 或者接收 *C* 发来的 *lose* 消息与 *Car* 切断联系, 之后发送 *switch* 消息给 *Car* 让 *Car* 切换到新的发射站。

(3) 汽车 *Car* 可以发送 *talk* 消息给 *Trans* 和 *Trans* 进行通信; 或者接收 *Trans* 发送的 *switch* 消息切换到新的发射站。

(4) 发射站 *ldtrans* 可以接收 *C* 发来的 *gain* 消息和 *Car* 建立新的连接。

下面给出该系统的移动网模型, 如图 2 所示。

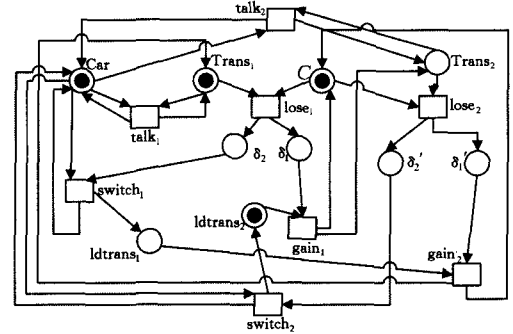


图 2 移动车辆电话通信系统的移动网模型

图 2 所示的移动网模型中共包含了 10 个库所, 分别表示中央控制台、发射站、汽车以及转化的中间状态, 其中:

(1) 库所 *C*(*talk*, *switch*) 表示中央控制台, 其中 (*talk*, *switch*) 为托肯的名字, 是形参; 可以实例化为 (*talk*₁, *switch*₁) 或者 (*talk*₂, *switch*₂)。当 *C* 告诉 *Trans* 切断与汽车的联系时, 它同时发出一对新的名字让 *Trans* 发送给汽车用以替换旧的名字。同时它还将这对新的名字发给另一个(闲置的)发射站, 使其能与汽车建立连接。

(2) 库所 *Car*(*talk*, *switch*) 表示汽车, 其中 (*talk*, *switch*) 为托肯的名字, 是形参; 可以实例化为 (*talk*₁, *switch*₁) 或者 (*talk*₂, *switch*₂), 下标表示当前和 *Car* 连接的发射站。Car 的托肯表示汽车和该发射站相连。

(3) 库所 *Trans*₁(*talk*₁, *switch*₁, *gain*₁, *lose*₁) 表示发射站 1 和汽车 *Car* 处于连接状态, 其中 (*talk*₁, *switch*₁, *gain*₁, *lose*₁) 为托肯的名字。发射站的托肯 *talk*₁, *switch*₁ 表示该发射站 1 和 *Car* 相连, 而 *gain*₁, *lose*₁ 表示发射站 1 和 *C* 相连。

(4) 库所 *Trans*₂(*talk*₂, *switch*₂, *gain*₂, *lose*₂) 表示发射站

2 和汽车 Car 处于连接状态,其中 $(talk_2, switch_2, gain_2, lose_2)$ 为托肯的名字。发射站的托肯 $talk_2, switch_2$ 表示该发射站 2 和 Car 相连,而 $gain_2, lose_2$ 表示发射站 2 和 C 相连。

(5)库所 $ldtrans_1 (gain_1, lose_1)$ 表示发射站 1 未和汽车 Car 连接这一状态,其中 $(gain_1, lose_1)$ 为库所 $ldtrans_1$ 托肯的名字。 $ldtrans_1$ 只和中央控制台连接。

(6)库所 $ldtrans_2 (gain_2, lose_2)$ 表示发射站 2 未和汽车 Car 连接这一状态,其中 $(gain_2, lose_2)$ 为库所 $ldtrans_2$ 托肯的名字。 $ldtrans_2$ 只和中央控制台连接。

(7)库所 $\delta_2 (talk_2, switch_2, gain_1, lose_1)$ 表示中央控制台 C 让 $Trans_1$ 断开和汽车的连接之后发射站 1 的状态。其中 C 通过该库所将参数 $(talk_2, switch_2)$ 传递给 Car,告诉 Car 发生 $switch_1$ 变迁之后和哪个发射站连接。 $Trans_1$ 将参数 $(gain_1, lose_1)$ 传递给 $ldtrans_1$,表示此时发射站 1 由 $Trans_1$ 变为 $ldtrans_1$ 状态,且托肯名字为 $(gain_1, lose_1)$ 。

(8)库所 $\delta_1 (talk_1, switch_1, talk_2, switch_2)$ 表示中央控制台 C 让 $Trans_1$ 断开和汽车的连接之后中央控制台的状态。其中 C 通过该库所将 $(talk_2, switch_2)$ 传递给 $Trans_2$,表明

$ldtrans_2$ 发生 $gain_1$ 变迁后变为 $Trans_2$ 状态;将 $(talk_1, switch_1)$ 传递给中央控制台 C,告诉 C 下一轮转化需要将 Car 连接到哪一个发射站。

(9)库所 $\delta_2' (talk_1, switch_1, gain_2, lose_2)$ 表示中央控制台让 $Trans_2$ 断开和汽车的连接之后发射站 2 的状态。其中 C 通过该库所将 $(talk_1, switch_1)$ 传递给 Car,告诉 Car 发生 $switch_2$ 变迁之后和哪个发射站连接。 $Trans_2$ 将 $(gain_2, lose_2)$ 传递给 $ldtrans_2$,表示此时 $Trans_2$ 变为 $ldtrans_2$ 状态,且托肯名字为 $(gain_2, lose_2)$ 。

(10)库所 $\delta_1' (talk_2, switch_2, talk_1, switch_1)$ 表示中央控制台让 $Trans_2$ 断开和汽车的连接之后中央控制台的状态。其中 C 通过该库所将 $(talk_1, switch_1)$ 传递给 $Trans_1$,表明 $ldtrans_1$ 发生 $gain_2$ 变迁后变为 $Trans_1$ 状态;将 $(talk_2, switch_2)$ 传递给中央控制台 C,告诉 C 下一轮转化需要将 Car 连接到哪一个发射站。

图 2 所示的移动网模型中共包含了 8 个变迁: $talk_1, switch_1, gain_1, lose_1, talk_2, switch_2, gain_2, lose_2$,表 1 对这些变迁作详细说明。

表 1 图 2 中变迁的详细说明

变迁	前集	后集	说明
$talk_1$	$Trans_1 (talk_1, switch_1, gain_1, lose_1), Car (talk_1, switch_1)$	$Trans_1 (talk_1, switch_1, gain_1, lose_1), Car (talk_1, switch_1)$	当库所 Car 和 $Trans_1$ 分别有名字为 $(talk_1, switch_1)$ 和 $(talk_1, switch_1, gain_1, lose_1)$ 的托肯时,这个变迁可以发生。变迁的发生实际表示 $Trans_1$ 和 Car 之间进行通信。
$lose_1$	$Trans_1 (talk_1, switch_1, gain_1, lose_1), C (talk_2, switch_2)$	$\delta_2 (talk_2, switch_2, gain_1, lose_1), \delta_1 (talk_1, switch_1, talk_2, switch_2)$	当库所 C 和 $Trans_1$ 分别有名字为 $(talk_2, switch_2)$ 和 $(talk_1, switch_1, gain_1, lose_1)$ 的托肯时,这个变迁可以发生。变迁的发生实际表示 C 通知 $Trans_1$ 和 Car 断开联系。
$gain_1$	$ldtrans_2 (gain_2, lose_2), \delta_1 (talk_1, switch_1, talk_2, switch_2)$	$C (talk_1, switch_1), Trans_2 (talk_2, switch_2, gain_2, lose_2)$	当库所 $ldtrans_2$ 和 δ_1 分别有名字为 $(gain_2, lose_2)$ 和 $(talk_1, switch_1, talk_2, switch_2)$ 的托肯时,该变迁可以发生。变迁的发生表示 C 让 $Trans_1$ 和 Car 断开联系之后,通知 $ldtrans_2$ 和 Car 连接,此时 $ldtrans_2$ 变为连接状态 $Trans_2$ 。
$switch_1$	$\delta_2 (talk_2, switch_2, gain_1, lose_1), Car (talk_1, switch_1)$	$Car (talk_2, switch_2), ldtrans_1 (gain_1, lose_1)$	当库所 Car 和 δ_2 分别有名字为 $(talk_1, switch_1)$ 和 $(talk_2, switch_2, gain_1, lose_1)$ 的托肯时,这个变迁可以发生。变迁的发生实际表示 C 让 $Trans_1$ 和 Car 断开联系之后, $Trans_1$ 通知 Car 连接到发射站 2,同时 $Trans_1$ 变为未连接状态 $ldtrans_1$ 。
$talk_2$	$Trans_2 (talk_2, switch_2, gain_2, lose_2), Car (talk_2, switch_2)$	$Trans_2 (talk_2, switch_2, gain_2, lose_2), Car (talk_2, switch_2)$	当库所 Car 和 $Trans_2$ 分别有名字为 $(talk_2, switch_2)$ 和 $(talk_2, switch_2, gain_2, lose_2)$ 的托肯时,这个变迁可以发生。变迁的发生实际表示 $Trans_2$ 和 Car 之间进行通信。
$lose_2$	$C (talk_1, switch_1), Trans_2 (talk_2, switch_2, gain_2, lose_2)$	$\delta_2' (talk_1, switch_1, gain_2, lose_2), \delta_1' (talk_2, switch_2, talk_1, switch_1)$	当库所 C 和 $Trans_2$ 分别有名字为 $(talk_1, switch_1)$ 和 $(talk_2, switch_2, gain_2, lose_2)$ 的托肯时,这个变迁可以发生。变迁的发生实际表示 C 让 $Trans_2$ 和 Car 断开联系。
$gain_2$	$\delta_1' (talk_2, switch_2, talk_1, switch_1), ldtrans_1 (gain_1, lose_1)$	$C (talk_2, switch_2), Trans_1 (talk_1, switch_1, gain_1, lose_1)$	当库所 $ldtrans_1$ 和 δ_1' 分别有名字为 $(gain_1, lose_1)$ 和 $(talk_2, switch_2, talk_1, switch_1)$ 的托肯时,该变迁可以发生。变迁的发生表示 C 让 $Trans_2$ 和 Car 断开联系之后,通知 $ldtrans_1$ 和 Car 连接,此时 $ldtrans_1$ 变为 $Trans_1$ 。
$switch_2$	$Car (talk_2, switch_2), \delta_2' (talk_1, switch_1, gain_2, lose_2)$	$Car (talk_1, switch_1), ldtrans_2 (gain_2, lose_2)$	当库所 Car 和 δ_2' 分别有名字为 $(talk_2, switch_2)$ 和 $(talk_1, switch_1, gain_2, lose_2)$ 的托肯时,这个变迁可以发生。变迁的发生实际表示 C 让 $Trans_2$ 和 Car 断开联系之后, $Trans_2$ 让 Car 连接到发射站 1,同时 $Trans_2$ 变为未连接状态 $ldtrans_1$ 。

在该例子中,初始状态下本文假设发射站 1 和汽车连接,发射站 2 和汽车没有连接。所以该移动网模型的初始标识 m_0 为:

$$m_0 = C (talk_2, switch_2), Trans_1 (talk_1, switch_1, gain_1, lose_1), Car (talk_1, switch_1), ldtrans_2 (gain_2, lose_2)$$

综上所述,将该网表示成移动网的正规形式为:

$$N = (\nu (C, \delta_1, \delta_2, \delta_2', \delta_1', ldtrans_1, Trans_1, ldtrans_2, Trans_2, Car)) ((\{ talk_1, switch_1, gain_1, lose_1, talk_2, switch_2, gain_2, lose_2 \}, m_0 = C (talk_2, switch_2), Trans_1 (talk_1, switch_1, gain_1, lose_1), Car (talk_1, switch_1), ldtrans_2 (gain_2, lose_2)))$$

3 移动网可达性分析方法

为了给出移动网的可达树,参照 Petri 网生成可达树的算法,利用 ω 符号来表示库所中名字数目无穷的情况。因此对于一个移动网,将其所有可达标识作为顶点集,以标识之间的直接可达关系为弧集可以构成一棵树。将这个有限树称为移动网 N 的可达树。

定义 5 设 $N = (\nu Y) (T, m_0)$ 为一个移动网。如果存在 $t \in T$, 使 $M \xrightarrow{t} M'$, 则称 M' 为从 M 直接可达的。如果存在变迁序列 $t_1, t_2, \dots, t_k$, 替换函数 $\rho_1, \rho_2, \dots, \rho_k$ 和标识序列 $M_1,$

$M_2, \dots, M_k$ 使 $M[t_1 >_{\rho_1} M_1[t_2 >_{\rho_2} M_2 \dots M_{k-1}[t_k >_{\rho_k} M_k]$, 则称 M_k 为从 M 可达的。从 M 可达的一切标识的集合记为 $R(M)$ 。

定义 6 设 $N=(\nu(Y)(T, m_0))$ 为一个移动网, N 的可达树定义为一个三元组 $RT(N)=(R(m_0), E, P)$, 其中:

$$E = \{(M_i, M_j) \mid M_i, M_j \in R(m_0), \exists t_k \in T: M_i[t_k >_{\rho} M_j]\}$$

$$P: E \rightarrow T, P(M_i, M_j) = t_k \text{ 仅当 } M_i[t_k >_{\rho} M_j]$$

称 $R(m_0)$ 为 $RT(N)$ 的顶点集, E 为 $RT(N)$ 的弧集; 若 $P(M_i, M_j) = t_k$, 则称 t_k 为弧 (M_i, M_j) 的旁标。

定义 7 对于两个集合 A, B , 如果集合 A 中任意一个元素都是集合 B 中的元素, 则就说这两个集合有包含关系, 称集合 A 为集合 B 的子集(subset), 记作 $A \subseteq B$ 。

定义 8 如果集合 $A \subseteq B$, 存在元素 $x \in B$, 且元素 x 不属于集合 A , 则集合 A 是集合 B 的真子集。

下面给出移动网可达树的构造算法。

算法 1 移动网的可达树构造算法

- Step1 将 m_0 作为 $RT(N)$ 的根结点, 标记为“新”。
- Step2 while 存在标记为“新”的结点, Do
任选一个标记为“新”的结点, 设为 M
- Step3 if 从 m_0 到 M 的有向路上存在一个结点的标识为 M , Then 将 M 标注改为“旧”, 返回 Step2。
- Step4 if $\forall t \in T: \neg M[t >_{\rho} \cdot]$ 的 $t \in T$, 即: $\neg \exists \rho \subseteq \text{bnp}(c) \times X$, 使得 $c \star_{\rho} \rho \subseteq M$, Then
将 M 的标注改为“端点”, 返回 Step2。
- Step5 for 每个满足 $M[t >_{\rho} \cdot]$ 的 $t \in T$, 即若 $\exists \rho \subseteq \text{bnp}(c) \times X$, 使得 $c \star_{\rho} \rho \subseteq M$

- 5.1 计算 $M[t >_{\rho} M']$ 中的 M' , 其中 $M' = (M \setminus c \star_{\rho} \rho) \oplus p\rho$ 。
- 5.2 If 从 m_0 到 M' 的有向路上存在 M'' , 使得 M'' 是 M' 的真子集, Then 找出 $M' - M''$ 的那个库所的名字, 将其后面标记为 ω 。
- 5.3 在 $RT(N)$ 中引入一个“新”结点 M' , 从 M 到 M' 画一条有向弧, 并把此弧标为 $t=c \triangleright p$, 擦去 M 的“新”标注, 返回 Step2。

下面对该算法进行解释说明:

在 Step3 中, 如果从初始标识到当前标识 M 的有向路上存在一个结点与 M 相同, 即已经存在当前标识 M , 则不继续向下, 该分支终止。

在 Step4 中, 如果在当前标识 M 下不存在可以发生的变迁, 则将该标识标记为端点, 不继续向下。其中判定是否存在可发生变迁的方法即是否存在一个替换函数, 使得对变迁的前集库所托肯的名字实例化之后是当前标识 M 的子集。

在 Step5 中, 对所有 M 下可发生的变迁(判定变迁是否可发生和 Step4 相同), 分别计算出新得到的标识 M' 。并判断从初始标识 m_0 到 M' 的有向路上是否存在另一个标识 M'' , 其中 M'' 是 M' 的真子集。若存在, 将 M' 改变形式(例如由标识 $A(Y), Y(B)$ 发生一个变迁得到标识 $A(Y), Y(B), Y(B)$ 则将后一个标识表示为 $A(Y), Y(B, \omega)$); 否则, 继续引发其余变迁。

定理 1 由算法 1 得到的移动网的可达树是有限的。

证明: 算法 1 与文献[12]的 Petri 网的可达树的算法类似, 树中的结点是有有限个的, 所以这个可达树是有限的。

引理 1 对于有界 Petri 网, 对任意 $M_i, M_j \in R(m_0)$, M_j 是从 M_i 可达的当且仅当在 $RT(N)$ 中, 从 M_i 到 M_j 存在一条有向路。

推论 1 在 $RT(N)$ 中, 从 m_0 到每个结点都有一条有向路。

例 1 下面给出一个移动网的例子。

$$N=(\nu\{A, B, Y\})((\{A(X) \triangleright A(Y), X(B)\}, A(A), A(B))$$

下图中将变迁 $A(X) \triangleright A(Y), X(B)$ 简记为 t_1 , 且变迁/之后表示的是发生该变迁消耗的库所托肯名字。则由算法 1 可以得到 N 的可达树, 如图 3 所示。(为了方便表示, 将 N 的可达树分成了 3 个部分, 用了 3 棵树表示, 其中图 4、图 5 分别是图 3 的两棵子树。

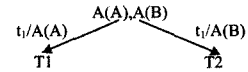


图 3 例 1 的可达树

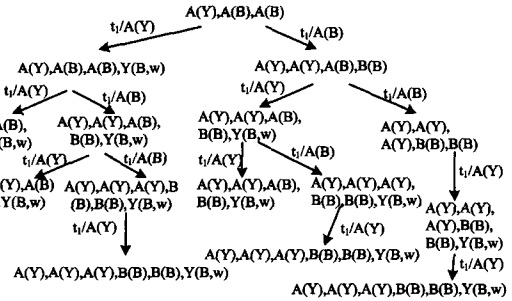


图 4 图 3 的子树 T1

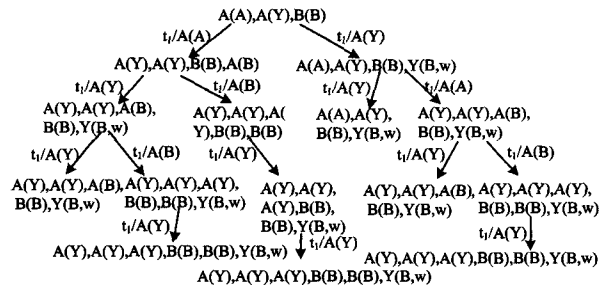


图 5 图 3 的子树 T2

由上图可知, 变迁 t_1 发生时, 当消耗的是名为 A 的托肯时, 该变迁的后集库所为 A ; 而当消耗的是名为 B 的托肯时, 该变迁的后集库所为 A 和 B 。即变迁的后集不是静态的, 而是取决于变迁发生所消耗的托肯。

4 移动车辆电话通信系统模型的可达树

根据算法 1, 可得出例 1 中移动车辆电话通信系统的可达树。由算法 1 得到的例 1 的可达树如图 6 所示。

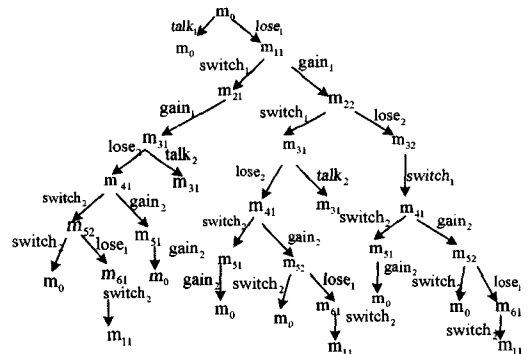


图 6 移动车辆电话通信系统的可达树

图 6 中结点的标识如表 2 所列。

表 2 图 6 中结点的标识

标识	标识代表的实际标识
m_0	$C(talk_2, switch_2), Trans_1(talk_1, switch_1, gain_1, lose_1), Car(talk_1, switch_1), ldtrans_2(gain_2, lose_2)$
m_{11}	$Car(talk_1, switch_1), ldtrans_2(gain_2, lose_2), \delta_2(talk_2, switch_2, gain_1, lose_1), \delta_1(talk_1, switch_1, talk_2, switch_2)$
m_{21}	$Car(talk_2, switch_2), ldtrans_2(gain_2, lose_2), ldtrans_1(gain_1, lose_1), \delta_1(talk_1, switch_1, talk_2, switch_2)$
m_{22}	$C(talk_1, switch_1), Car(talk_1, switch_1), \delta_2(talk_2, switch_2, gain_1, lose_1), Trans_2(talk_2, switch_2, gain_2, lose_2)$
m_{31}	$Car(talk_2, switch_2), ldtrans_1(gain_1, lose_1), C(talk_1, switch_1), Trans_2(talk_2, switch_2, gain_2, lose_2)$
m_{32}	$Car(talk_1, switch_1), \delta_2(talk_2, switch_2, gain_1, lose_1), \delta_2'(talk_1, switch_1, gain_2, lose_2), \delta_1'(talk_2, switch_2, talk_1, switch_1)$
m_{41}	$Car(talk_2, switch_2), ldtrans_1(gain_1, lose_1), \delta_2'(talk_1, switch_1, gain_2, lose_2), \delta_1'(talk_2, switch_2, talk_1, switch_1)$
m_{51}	$Car(talk_1, switch_1), ldtrans_1(gain_1, lose_1), ldtrans_2(gain_2, lose_2), \delta_1'(talk_2, switch_2, talk_1, switch_1)$
m_{52}	$Car(talk_2, switch_2), C(talk_2, switch_2), \delta_2'(talk_1, switch_1, gain_2, lose_2), Trans_1(talk_1, switch_1, gain_1, lose_1)$
m_{61}	$Car(talk_2, switch_2), \delta_2'(talk_1, switch_1, gain_2, lose_2), \delta_2(talk_2, switch_2, gain_1, lose_1), \delta_1(talk_1, switch_1, talk_2, switch_2)$

根据图 6 可知,移动车辆电话通信系统的可达树是有限的。

结束语 移动网是在 Petri 网的基础上增加了移动性,并结合了进程代数的优势得到的。移动网改进了 Petri 网结构静态并且无法反映结构变化的弱点,适于模拟和分析移动计算系统。本文给出了移动网的可达树构造算法,为移动网模型提供了基本的可达性分析手段。我们未来将基于移动网的可达性分析方法研究模型的各种动态性质。

参考文献

[1] Asperti A, Busi N. Mobile Petri Nets[J]. Mathematical Struc-

tures in Computer Science, 2009; 19(6): 1265-1278

[2] Berry G, Boudol G. The chemical abstract machine[J]. Theoretical Computer Science, 1992; 96(1): 217-248

[3] Du X T. Concurrent computing paradigm: CCS and calculus[J]. Computer Science, 2002; 29(10): 1-4 (in Chinese)
杜旭涛. 计算范型: CCS 和 π 演算[J]. 计算机科学, 2002; 29(10): 1-4

[4] Fournet C, Gonthier G. The reflexive cham and the join-calculus [C]// Proceedings of the 23rd Symposium on Principles of Programming Languages, 1996; 372-385

[5] Jensen K. Coloured Petri Nets [M]. EATCS Monographs in Computer Science, SpringerVerlag, 1992; 10-110

[6] Peterson J L. Petri Net Theory and the Modeling of Systems [M]. Englewood Cliffs, NJ: Prentice-Hall, 1981

[7] Milner R, Parrow J, Walker D. A calculus of mobile processes [J]. Information and Computation, 1992; 100(1): 1-77

[8] Montanari U, Rossi F. Contextual nets[J]. Acta Inf., 1995; 32(6): 545-596

[9] Reisig W. An Introduction to Petri Nets [M]. EATCS Monographs in Computer Science, SpringerVerlag, 1985; 1-164

[10] Robin Milner. Communicating and mobile systems; the pi-calculus[M]. Cambridge, the University of Cambridge, 1999; 82-85

[11] Sangiorgi D. Expressing Mobility in Process Algebra[D]. University of Edinburgh, 1993

[12] Aggarwal S, Pathak H. Modelling of hierarchical location management schemes to locate mobile multi agents using colored Petri Net[M]. Computer Engineering and Applications, 2015; 821-825

[13] Wolfgang Reisig. Understanding Petri Nets[M]. Springer Verlag, 2013; 13-23

[14] 吴哲辉. Petri 网导论[M]. 北京, 2006; 47-193

(上接第 23 页)

[6] Elhamifar, Ehsan, Sapiro G, Vidal R. See all by looking at a few: Sparse modeling for finding representative objects[C]// Computer Vision and Pattern Recognition IEEE Conference on (CVPR). 2012; 1600-1607

[7] Cheng B, Yang J, Yan S, et al. Learning with ℓ_1 -graph for image analysis [J]. IEEE Transactions on Image Processing, 2010; 19(4): 858-866

[8] Dang C T, Kumar M, Radha H. Key frame extraction from consumer videos using epitome[C]// Image Processing, International Conference on (ICIP). IEEE, 2012; 93-96

[9] He Ran, Zheng Wei-shi, Hu Bao-gang, et al. Nonnegative sparse coding for discriminative semi-supervised learning[C]// Computer Vision and Pattern Recognition (CVPR). CVPR, IEEE, Providence, USA, 2011; 792-801

[10] Wong W K. Discover latent discriminant information for dimensionality reduction: non-negative sparseness preserving embedding [J]. Pattern Recognition, 2012; 45(4): 1511-1523

[11] Lu Gui-fu, Jin Zhong, Zou Jian. Face recognition using discriminant sparsity neighborhood preserving embedding [J]. Knowledge based Systems, 2012; 31(2): 119-127

[12] Shi Jian-bo, Malik J. Normalized Cuts and Image Segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intel-

ligence, 2000; 22(8): 888-905

[13] Raducanu B, Dornaika F. A supervised non-linear dimensionality reduction approach for manifold learning [J]. Pattern Recognition, 2012; 45(6): 2432-2444

[14] Ilyas M U, Radha H. Identifying Influential Nodes in Online Social Networks Using Principal Component Centrality[C]// 2011 IEEE International Conference on Kyoto. IEEE, Japan, 2011

[15] Almeida, Jurandy, Leite N J, Ricardo da S Torres. Online video summarization on compressed domain [J]. Journal of Visual Communication and Image Representation, 2013; 24(6): 729-738

[16] Almeida, Jurandy, Leite N J, et al. Vision: Video summarization for online applications[J]. Pattern Recognition Letters, 2012; 33(4): 397-409

[17] The Open Video Project[OL]. <http://www.open-video.org>

[18] Kim Y J, Oh Y T, Yoon S H, et al. Efficient Hausdorff Distance computation for freeform geometric models in close proximity [J]. Computer-Aided Design, 2013; 45(2): 270-276

[19] Schuetz O, Equivel X, Lara A, et al. Some comments on GD and IGD and relations to the Hausdorff distance[C]// Proceedings of the 12th Annual Conference Comp on Genetic and Evolutionary Computation (GECCO's 10:). New York, NY, USA, ACM, 2010; 1971-1974