标记迁移系统的组合可达性分析*)

文艳军　王　戟　齐治昌
(国防科技大学计算机学院　长沙 410073)

摘　要　标记迁移系统是一种在计算机辅助设计和验证中得到广泛使用的形式模型。当系统中的模块比较多时，系统的整体模型有可能出现状态空间的指数级爆炸，组合可达性分析是缓解这一问题的一种有效方法。已有的工作缺乏对该方法基本原理的清晰描述和精确表达。本文对其基本原理进行了分析和概括，并作了形式化陈述，证明了相关结论。本文的工作有助于深入理解和澄清组合可达性分析的内部工作机制。

关键词　标记迁移系统，计算机辅助设计和验证，组合可达性分析，状态空间爆炸问题，观察等价

Compositional Reachability Analysis of Labeled Transition Systems

WEN Yan-Jun　WANG Ji　QI Zhi-Chang
(School of Computer, National University of Defense Technology, Changsha 410073)

Abstract　Labeled transition system is a formalism that is widely used in the literature of computer-assistant design and verification. As the module number of the system increases, the state space of the whole system may exponentially explode. Compositional reachability analysis (CRA) is an effective approach to alleviate the problem. The existing work lacks clear and precise descriptions to the principle of CRA. This paper summarizes and formalizes the essential principle of CRA, and proves the main results, which may benefit the deep understanding and clarification of CRA.

Keywords　Labeled transition system, Computer-assistant design and verification, Compositional reachability analysis (CRA), State-explosion problem, Observation equivalence

1　引言

标记迁移系统(Labeled Transition System，简称 LTS)是一种用于系统行为建模的基本的形式模型，在计算机辅助设计和验证领域被广泛使用。许多形式模型(比如自动机的许多变种以及 CCS[1]，CSP[2]等进程代数模型)的操作语义以它作为解释基础。在 LTS 上可以定义并发组合、动作隐藏和重命名等操作，从而可以组合地构造出复杂的系统。

因为多个 LTSs 的并发组合模型的大小与 LTS 的个数成指数相关，所以当系统中的模块比较多时，会造成状态空间的急剧膨胀，带来所谓的状态爆炸问题[3,4]。为此，人们提出了各种缓解状态爆炸的方法，组合可达性分析(Compositional Reachability Analysis)就是其中的一种有效手段。其基本思想是：将系统按照某种层次结构组织起来，然后自底向上逐层计算各个子系统的模型。尤其重要的是，在每一个中间环节都要尽量将外界不可见的动作隐藏起来，并且对得到的模型进行化简，使得其在保持足够多的性质的前提下尽可能地小。这样，由于动作隐藏和化简的影响，一般来说会大幅减小最终得到的系统整体模型的尺寸。比如，在 Sahnani 等人[5]对 Q. 931 协议的测试中，发现中间模型从来没有超过 1000 个状态；而如果直接计算的话，状态数将超过 60000。

关于组合可达性分析，已经有了许多的工作[3~9]，但就作者所知，它们大多采用一种非形式化或半形式化的方式进行叙述，缺乏对其基本原理的清晰描述和精确表达。本文在分析已有工作的基础上，对组合可达性分析的原理做了归纳和形式化陈述，澄清了其内部工作机制。

本文第 2 节介绍了标记迁移系统的基本概念；第 3 节对组合可达性分析做了形式化描述，从前提假设和基本原理两个方面进行介绍，并证明了相关结论；最后是对本文的简单小结。

2　基本概念

本节给出标记迁移系统的定义，同时介绍其上的并发组合与隐藏操作。

定义 1(标记迁移系统(LTS))　一个标记迁移系统是一个四元组 $P=\langle V_P, A_P, \Delta_P, q_P\rangle$，其中：

- V_P 是 P 的状态集合；
- $A_P=\alpha P\cup\{\tau\}$，其中 αP 表示 P 的通信事件集(不包含内部事件 τ)；
- $\Delta_P\subseteq V_P\times A_P\times V_P$，它是一个迁移关系，描述一组迁移步；
- $q_P\in V_P$，它是 P 的初始状态。

设 P 是一个 LTS，$s_1, s_2\in V_P$，$a\in A_P$，设 $\alpha=a_1a_2\cdots a_n\in(A_P)^n$，$\beta=b_1b_2\cdots b_n\in(\alpha_P)^n$，我们定义 P 上的一些关系如下：

- $s_1\xrightarrow{a}_P s_2$ 当且仅当 $(s_1,a,s_2)\in\Delta_P$。
- $s_1\xRightarrow{a}_P s_2$ 当且仅当 $s_1\xrightarrow{a_1}_P\xrightarrow{a_2}_P\cdots\xrightarrow{a_n}_P s_2$。特别地，$s_1\xrightarrow{\epsilon}_P s_1$。

*)本文受国家自然科学基金(重点项目 No. 60233020，重大研究计划项目 No. 90104007)、863 项目(No. 2001AA113202，No. 2001AA113190)资助。文艳军　博士生，主要研究方向为软件的组合模型检验；王　戟　博士，教授，博士生导师，主要研究方向为形式化方法、软件测试、软件可靠性、语义 Web 等；齐治昌　教授，博士生导师，主要研究方向为软件工程等。

其中关系操作符的并置指关系的组合。

两个 LTSs 的并发组合的定义类似于 CSP 中两个进程的并发组合的定义。

定义 2(LTSs 的并发组合(Parallel Composition)) 设 P 和 Q 为两个 LTSs,则其并发组合 $P \| Q$ 定义为 LTS $R=\langle V_R, A_R, \Delta_R, q_R\rangle$,其中:

- $V_R = V_R \times V_Q, A_R = A_P \cup A_Q, q_R = (q_P, q_Q)$。
- Δ_R 由如下三条迁移规则定义:

$$\frac{s_P \xrightarrow{a}_P s'_P}{(s_P, s_Q) \xrightarrow{a}_R (s'_P, s_Q)} (a \notin \alpha Q),$$

$$\frac{s_Q \xrightarrow{a}_Q s'_Q}{(s_P, s_Q) \xrightarrow{a}_R (s_P, s'_Q)} (a \notin \alpha P)$$

$$\frac{s_P \xrightarrow{a}_P s'_P \quad s_Q \xrightarrow{a}_Q s'_Q}{(s_P, s_Q) \xrightarrow{a}_R (s'_P, s'_Q)} (a \in \alpha P \cap \alpha Q)$$

通过一个隐藏操作可以将一个 LTS 的外部动作隐藏起来,使其变成内部动作。

定义 3(LTSs 的隐藏) 一个 LTS P 在动作集合 $L(\tau \notin L)$ 上的隐藏定义为 LTS $P\backslash L = \langle V_P, A_{P\backslash L}, \Delta_{P\backslash L}, q_P\rangle$,其中 $\Delta_{P\backslash L}$ 由如下两条迁移规则定义:

$$\frac{s \xrightarrow{a}_P s'}{s \xrightarrow{\tau}_{P\backslash L} s'} (a \in L), \quad \frac{s \xrightarrow{a}_P s'}{s \xrightarrow{a}_{P\backslash L} s'} (a \notin L)$$

3 组合可达性分析

组合可达性分析(Compositional Reachability Analysis,简称 CRA)是一种层次化渐增式计算系统的整体模型的方法[4, 6, 8, 9]。为了分析一组模型的并发组合,显然最简单直接的办法就是一步计算到位,也就是所谓的"all-at-once"方法[7]。这种方法容易导致状态空间的爆炸,因此对于大系统的分析不适合。CRA 方法试图缓解这一问题:在此方法指导下,系统中的各个模型被按照层次结构组织起来,然后自底向上逐层计算各子系统的并发组合,并按照某种等价语义进行化简,接着使用简化了的模型来计算更高层次的系统模型;如此反复,直至最高层,这时也就计算出了系统的整体模型。因为在中间步骤中可以进行动作隐藏和化简操作,所以一般来说这一方法可以较大幅度地减小系统的状态空间[4, 6],降低计算的空间复杂度。在文[9]中这一方法被称为组合 LTS 构造(Compositional LTS Construction),而在文[5]中称为 ICR 方法(Incremental Composition and Reduction Method)。

本节我们对 CRA 使用的前提条件和基本原理进行形式化的分析和归纳总结。

3.1 前提条件

组合可达性分析的关键在于化简操作,该操作应该遵循某种等价语义,即化简前后的模型在该语义下是等价的。等价语义有很多种,如强互模拟(Strong Bisimilarity)、弱互模拟(Weak Bisimilarity,或称为观察等价)、分枝互模拟(Branching Bisimilarity)、路径(Trace)等价和 CSP 的 Failures-Divergences 模型等[9]。其中,以观察等价最为常用。但是,无论采用那种等价语义,其相应的组合模型检验的基本原理都是相同的。归纳而言,LTSs 上的一个等价语义$\approx$只要满足下列两个定理,那么就可以基于该语义进行组合可达性分析。

首先,对于 LTS 的并发组合操作和隐藏操作来说,$\approx$应该是一个同余(congruence)关系。

定理 1 任给 LTSs P, P', Q 和 Q',以及不包含内部动作 τ 的动作集合 L,下列命题成立:

1. $P \approx P' \Rightarrow P\backslash L \approx P'\backslash L$.
2. $P \approx P' \wedge Q \approx Q' \Rightarrow P \| Q \approx P' \| Q'$。

其次,在等价语义$\approx$下,多个隐藏操作应该可以合并。

定理 2 任给 LTSs P、Q 和不包含内部动作 τ 的动作集合 L_1, L_2,下列命题成立:

1. $P\backslash L_1\backslash L_2 \approx P\backslash(L_1 \cup L_2)$。
2. 如果 $(\alpha P \cap L_2) = (\alpha P \cap L_1) = (L_1 \cap L_2) = \emptyset$,那么 $(P\backslash L_1) \| (Q\backslash L_2) \approx (P \| Q)\backslash(L_1 \cup L_2)$。[1)]

3.2 基本原理

正如前述,在组合可达性分析时,系统被组织成一个层次结构,然后自底向上逐层进行并发组合的计算、动作的隐藏和模型的缩减。本小节对这一过程进行形式化阐述。

一个 LTS 层次结构 H 可以形式地递归定义为:$H ::= (H_1 \| H_2 \| \cdots \| H_n)\backslash L | LTS\backslash L$。其中,$LTS$ 表示一个 LTS,L 表示一个不包含内部动作 τ 的动作集合。由定义可知:H 要么是一组子层次结构的并发组合,要么就是一个 LTS。无论哪种情况,最后都要将 L 中的动作隐藏起来。

设 P 为一个 LTS,我们用 $red(P)$ 表示在保持等价语义$\approx$的前提下对 P 的一个缩减操作(即 $red(P) \approx P$),用 $cra(H)$ 表示按 CRA 方法计算得到的 H 的最终 LTS。那么 CRA 的算法如下所示:

$$cra(H) = \begin{cases} red((\prod_{i=1}^{n} cra(H_i))\backslash L) & \text{if } H = (H_1 \| H_2 \| \cdots \| H_n)\backslash L \text{[2)]} \\ red(P\backslash L) & \text{if } H = P\backslash L \end{cases}$$

为了说明基于逆观察等价的 CRA 的原理,下面我们递归定义层次结构上的几个操作:$lst(H)$、$act(H)$ 和 $hid(H)$,分别表示层次结构 H 中所有的 LTSs、所有的动作和所有被隐藏的动作。

$$lst(H) = \begin{cases} \bigcup_{i=1}^{n} lst(H_i) & \text{if } H = (H_1 \| H_2 \| \cdots \| H_n)\backslash L \\ \{P\} & \text{if } H = P\backslash L \end{cases}$$

$$act(H) = \begin{cases} \bigcup_{i=1}^{n} act(H_i) & \text{if } H = (H_1 \| H_2 \| \cdots \| H_n)\backslash L \\ \alpha P & \text{if } H = P\backslash L \end{cases}$$

$$hid(H) = \begin{cases} (\bigcup_{i=1}^{n} hid(H_i)) \cup L & \text{if } H = (H_1 \| H_2 \| \cdots \| H_n)\backslash L \\ L & \text{if } H = P\backslash L \end{cases}$$

在接下来的讨论里,我们假设对于任意的层次结构 $H = (H_1 \| H_2 \| \cdots \| H_n)\backslash L$,下式成立:

$$\forall i, j \in [1 \cdots n](i \neq j) \Rightarrow (hid(H_i) \cap act(H_j)) = (hid(H_i) \cap hid(H_j)) = \emptyset \quad (A)$$

其目的是使得将隐藏操作推迟到更高层次处理时不会产生动作的混淆。很明显,对于任何不满足上述条件的层次结构,我们都可以通过重命名使得该条件得以满足而不改变各 LTS 的行为(在同构意义下)。因此,上述假设不失一般性。

(下转第 145 页)

1)其中 αP 表示 LTS P 的动作集合[6]。

2)其中 $\prod_{i=1}^{n} cra(H_i)$ 表示 $cra(H_1) \| cra(H_2) \| \cdots \| cra(H_n)$。

趋势。

参 考 文 献

1 Anderson T E, Culler D E, Patterson D. A case for NOW (Networks of Workstations). Micro IEEE, 1995, 15(1): 54～64

2 Alexandrov A, Ionescu M, Schauser K E, et al. LogGP: Incorporating Long Messages into the LogP Model-One Step Closer Towards a Realistic Model of Parallel Computation. In: Proc. Seventh Ann ACM Symp Parallel Algorithms and Architectures, 1995. 95～105

3 Moritz C A, Frank M I. LoGPG: Modeling network contention in message-passing programs. Parallel and Distributed Systems, IEEE Transactions on, 2001, 12(4): 404～415

4 Krizanc R, Saarimaki A. Bulk Synchronous Parallel: Practical Experiencewith a Model for Parallel Comuting, Parallel Architectures and Compilation Techniques. In: Proc. of the 1996 Conf., 1996. 208～217

5 Gibbons P B, Matias Y, Ramachandran V. The QRQW PRAM: Accounting for Contention in Parallel Algorithms. In: Proc. of the fifth annual ACM-SIAM symposium on Discrete algorithms, 1994. 638～648

6 Bar-Noy A, Kipnis S. Designing Broadcasting Algorithms in the Postal Model for Message_Passing Systems. In: Proc. of the fourthannual ACM symposium on Parallel algorithms and architectures, 1992. 13～22

7 Culler D E, Karp R M, Patterson D, et al. LogP: Towards a Realistic Model of Parallel Computation. Communications of the ACM, 1996, 39(11): 78～85

8 Ino F, Fujimoto N, Hagihara K. LogGPS: A Paallel Computational Model for Synchronization Analysis. ACM SIGPLAN Notices, 2001(7): 133～142

9 Touyama T, Horiguchi S. Performance Evaluation of Practical Parallel Computation Model LogPQ. In: Proc. of the 1999 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '99), 1999. 216～222

10 Moritz C A, Frank M I. LoGPC: Modeling Network Contention in Message-Passing Programs. In: Proc. of the 1998 ACM SIGMETRICS Joint Intl. Conf. on Measurement and modeling of computer systems, 1998. 254～263

11 Gibbons P B. What good are shared-memory models? In: Proc of the 1996 ICPP Workshop on Challenges for, 1996. 103～114

(上接第 111 页)

任给层次结构 H，设 $lts(H)=\{T_1, T_2, \cdots, T_n\}$。我们将采用"all-at-once"方式得到的 LTS 记为 $once(H)$，即：

$once(H)=_{df}(\prod_{i=1}^{n} T_i)\backslash hid(H)$

下一引理说明了 $once(H)$ 也可以通过组合方式计算得到。

引理 1 任给一个 LTS 层次结构 $H=(H_1 \parallel H_2 \parallel \cdots \parallel H_n)\backslash L$，则 $once(H)=(\prod_{i=1}^{n} once(H_i))\backslash L$ 成立。

证明：设 $lts(H_i)=\{T_i^1, T_i^2, \cdots, T_i^{n_i}\}$，记 $\prod lts(H_i)=_{df}\prod_{j=1}^{n_i} T_i^j$，我们推理如下：

(1) $once(H)=(\prod_{i=1}^{n}\prod lts(H_i))\backslash((\bigcup_{i=1}^{n} hid(H_i))\cup L)$ 由定义

(2) $once(H_i)=(\prod lts(H_i))\backslash hid(H_i)$ 由定义

(3) $(\prod_{i=1}^{n} once(H_i))\backslash L=(\prod_{i=1}^{n}((\prod lts(H_i))\backslash hid(H_i)))\backslash L$ 由(2)

(4) $(\prod_{i=1}^{n}((\prod lts(H_i))\backslash hid(H_i)))\backslash L=(\prod_{i=1}^{n}\prod lts(H_i))\backslash(\bigcup_{i=1}^{n} hid(H_i))\backslash L=(\prod_{i=1}^{n}\prod lts(H_i))\backslash((\bigcup_{i=1}^{n} hid(H_i))\backslash L)$ 由(3)、假设(A)和定理 2

(5) $once(H)=(\prod_{i=1}^{n} once(H_i))\backslash L$ 由(1)和(4)

得证。

下一定理反映了组合可达性分析的基本原理：对于等价语义$\approx$而言，使用 CRA 方法得到的系统模型与采用"all-at-once"方法得到的系统模型是等价的。

定理 3 任给一个 LTS 层次结构 H，公式 $cra(H)\approx once(H)$ 成立。

证明：根据层次结构的定义递归证明如下：

1. 如果 $H=P\backslash L$，其中 P 是一个 LTS，那么 $cra(H)=red(P\backslash L)$、$once(H)=P\backslash L$。因为 $red(P\backslash L)\approx P\backslash L$，所以 $cra(H)\approx once(H)$。

2. 如果 $H=(H_1 \parallel H_2 \parallel \cdots \parallel H_n)\backslash L$，其中 $H_i(i\in[1\cdots n])$ 为 H 的子层次结构且 $cra(H_i)\approx once(H_i)$，令 $H'=_{df}(cra(H_1) \parallel cra(H_2) \parallel \cdots \parallel cra(H_n))\backslash L$，我们推理证明如下：

(1) $cra(H)=red((\prod_{i=1}^{n} cra(H_i))\backslash L)$ 由定义

(2) $cra(H)=red(H'), red(H')\approx H'$ 由定义

(3) $cra(H)=H'$ 由(2)

(4) $once(H)=(\prod_{i=1}^{n} once(H_i))\backslash L$ 由引理 1

(5) $cra(H_i)\approx once(H_i)(i\in[1\cdots n])$ 前提假设

(6) $\prod_{i=1}^{n} cra(H_i)\approx\prod_{i=1}^{n} once(H_i)$ 由(5)和定理 1

(7) $(\prod_{i=1}^{n} cra(H_i))\backslash L\approx(\prod_{i=1}^{n} once(H_i))\backslash L$ 由(6)和定理 1

(8) $H'\approx once(H)$

(9) $cra(H)\approx once(H)$ 由(3)和(8)

综上所述，$cra(H)\approx once(H)$，得证。

结论 本文对标记迁移系统的组合可达性分析的基本原理做了详细分析和形式化描述，证明了相关结论。本文的工作有助于深入理解和阐明组合可达性分析的内部工作原理，从而为基于标记迁移系统的计算机辅助设计和验证提供支持。

参 考 文 献

1 Milner R. A Calculus of Communicating Systems. New York: Springer-Verlag, Inc, 1982

2 Brookes S D, Hoare C A R, Roscoe A W. A theory of communicationg sequential processes. Journal of the ACM (JACM), 1984, 31(3): 560～599

3 Graf S, Steffen B, Lütgen G. Compositional minimization of finite state systems using interface specifications. Formal Aspects of Computing, 1996, 8(5): 607～616

4 Cheung S C, Kramer J. Enhancing compositional reachability analysis with context constraints. In: Proc. of the 1st ACM SIGSOFT Symposium on Foundations of Software Engineering, Los Angeles, California, United States, 1993

5 Sabnani K K, Lapone A M, Uyar M ü. An algorithmic procedure for checking safety properties of protocols. IEEE Transactions on Communications, 1989, 37: 940～948

6 Cheung S C, Kramer J. Checking subsystem safety properties in compositional reachability analysis. In: Proc. of the 18th Intl. Conf. on Software Engineering, Berlin, Germany, 1996

7 Tai K C, Koppol P V. Hierarchy-based incremental analysis of communication protocols. In: Proc. of First Intl. Conf. on Network Protocols, 1993. 318～325

8 Yeh W J, Young M. Compositional reachability analysis using process algebra. In: Proc. of the Symposium on Testing, Analysis, and Verification, Victoria, British Columbia, Canada, 1991

9 Valmari A. Compositionality in state space verification methods. In: Proc. of 17th Intl. Conf. in Application and Theory of Petri Nets (ICATPN'96), Osaka, Japan, 1996