

基于左右语言的状态迁移系统的优化

刘建国^{1,3} 袁志斌²

(华中科技大学计算机科学与技术学院 武汉 430074)¹ (广东商学院信息学院 广州 510320)²
(武汉理工大学汽车学院 武汉 430070)³

摘要 基于左右语言的优化是完全基于自动机理论的优化方法。所谓左语言是指从初始状态到指定状态的语言，而右语言是指从指定状态到接受状态的语言。K-模拟为左右语言的计算提供了一个高效的算法，研究了基于K-模拟的左右语言的Kripke结构、Büchi自动机的优化方法。

关键词 状态迁移系统, K-模拟, 左右语言

中图分类号 TP311 **文献标识码** A

Left and Right Language-based Optimizing State Transition Systems

LIU Jian-guo^{1,3} YUAN Zhi-bin²

(Department of Computer Science and Technology, Huazhong University of Science & Technology, Wuhan 430074, China)¹
(School of Information, Guangdong Commercial College, Guangdong 510320, China)²
(School of Automobile, Wuhan University & Technology, Wuhan 430070, China)³

Abstract The left and right languages belong to the automata theoretic part of the reduction problem. The left (resp. right) language is the set of calculations leading to (resp. starting from) a state. K-simulations enable a fast detection of these trace inclusions. We suggested an approximate minimization algorithm for Kripke structure, Büchi automata based on left and right languages.

Keywords State transition systems, K-simulation, Left and right languages

1 引言

自动机是计算机理论与科学的一个重要的抽象模型，其应用领域极为广泛。自动机的优化是自动机理论研究的一个重要内容，该问题不仅具有很高的理论价值而且在实践中也有极高的应用价值。T. Jiang 和 B. Ravikumar 证明了非确定自动机的最小化问题是 NP 难问题^[1]，因此我们不可能找到通用的最优化方法，而只能探求启发式算法。文献[2]的方法理论上可行，但实现复杂，因此其实用价值不高。文献[3]采用了基于等价关系的渐进式算法，但是由于语言等价关系计算的困难，该方法在实现上依然困难。

本文研究了基于左右语言的化简方法。其基本思想是利用有穷状态自动机的左、右语言的概念来研究自动机的化简。结点的左语言就是从初始状态到该结点的计算；结点的右语言就是从该结点到接受状态的计算。利用左右语言对自动机优化的核心思想是语言的等价，而语言的等价是 NP-难问题，因此只难寻求有效的近似算法，模拟则是极佳的选择。本文深入研究了基于左语言相同的 Kripke 结构的优化方法，并同基于模拟等价关系的优化方法进行比较，可知基于左语言相同的优化方法在理论上和计算的开销上都有很大的优势。左右语言的概念是建立在有穷语言自动机上的概念，但是根据

无穷语言自动机(包括 Büchi 自动机和广义 Büchi 自动机)语言的特征，可以将无穷语言自动机的优化转化为多个有穷语言自动机的优化，并可以用左右语言来实现。本文还研究了基于 K-直接模拟的 Büchi 自动机的优化方法，该方法可以获得比文献[4]更进一步的优化。

2 优化的理论基础

文献[4]讨论了基于左右语言的自动机优化的基础理论，下面简要介绍本文所需要的相关理论。

定义 1 设有穷迁移系统 $M=(S, S_0, R, L, F)$ $p \in S$, 则状态 p 的左语言为 $\overleftarrow{L}(p) = \{w \in \Sigma^* \mid p \in R(S_0, w)\}$, 状态 p 的右语言为

$$\overrightarrow{L}(p) = \{w \in \Sigma^* \mid R(S_0, w) \cap F \neq \emptyset\}.$$

设有穷迁移系统 $M=(S, S_0, R, L, F)$, $p, q \in S$, 将状态 p 合并到状态 q 后得到的迁移系统为 $merge(M, q, p) = (S', S'_0, R', L', F')$, 则

- 1) $S' = S - p$, 即从状态集合中去掉状态 p .
- 2) 如果 $p \in S_0$ 则 $S'_0 = (S_0 \cup q) - \{p\}$, 否则, $S'_0 = S_0$.
- 3) $R' = (R \cup \{(q, s) \mid (p, s) \in R\} \cup \{(s, q) \mid (s, p) \in R\}) - (R \cap (p) \times S \cup S \times p)$, 即用 q 替代所有出现在迁移中的 p .
- 4) 对于所有的 F_i , 如果 $p \in F_i$, 则 $F' = (F \cup \{q\}) - \{p\}$,

到稿日期:2008-06-25 本文工作得到广东自然科学基金(No. 06023961)及广东省教育厅自然科学基金(No. 052013)的支持。

刘建国(1970-),男,博士生,主要研究方向为人工智能、形式化方法, E-mail: LJJ424@163.com; 袁志斌(1971-),男,博士生,主要研究方向为程序正确性、形式化方法。

否则, $F' = F$ 。

定理 1 如图 1 所示, 设 p, q 为状态迁移系统 M 的两个状态, 将状态 p 合并到状态 q 得到的迁移系统为 $merge(M, q, p)$, 显然以下等式成立:

$$1) \vec{L}_{merge(M, q, p)}(q) = \vec{L}_M(q) \cup \vec{L}_M(p);$$

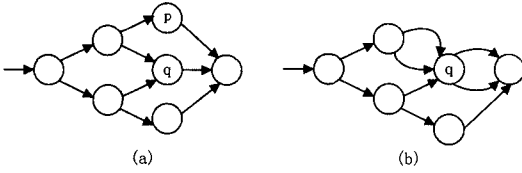


图 1 状态合并示意图

$$2) \vec{L}_{merge(M, q, p)}(p) = \vec{L}_M(q) \cup \vec{L}_M(p)$$

定义 2 设 p, q 为状态迁移系统 M 的两个状态, 迁移系统 M 和 $merge(M, q, p)$ 等价, 即 $L(M') = L(M)$, 则称状态 q 和 p 等价, 记作 $p \sim q$ 。

根据以上定义可知, 如果两个状态等价, 就可以将这两个状态合并, 得到的迁移系统与原系统等价。因此, 要通过状态的合并来简化迁移系统, 就需要研究两个状态等价的充分必要条件。

定理 2 设 p, q 为状态迁移系统 M 的两个状态, 状态 q 和 p 等价的充分必要条件是 $\vec{L}_M(q) \cdot \vec{L}_M(p) \subseteq L(A)$,

$$\text{而且 } \vec{L}_M(p) \cdot \vec{L}_M(q) \subseteq L(A)。$$

证明: 迁移结构 M 和 N 等价的充分条件是对任意的 $u, v \in \Sigma^*$, 如果 $u \cdot v \in L(M)$ 当且仅当 $\exists q' \in N$ 满足 $u \in \vec{L}(q') \wedge v \in \vec{L}(q')$ 。

根据状态合并的定义, 对于任意 $u, v \in L(M)$, 在 $merge(M, q, p)$ 中必有一个与之相对应的结点 $q'u \in \vec{L}(q') \wedge v \in \vec{L}(q')$ 。

设任意 $u \cdot v \in L(merge(M, q, p))$, 当 $u \notin \vec{L}(q)$ 或 $v \notin \vec{L}(q)$ 时, 取对应的结点即可。

只有 $u \in \vec{L}(q)$ 而且 $v \in \vec{L}(q)$ 和 $u \notin \vec{L}(q)$ 而且 $v \in \vec{L}(q)$ 是没有对应的状态, 但是根据条件可知 $u \cdot v \in L(A)$ 。

综上所述,

$$L(M) = L(merge(M, q, p)), \text{ 即 } p \sim q$$

定理 3 根据定理 2, 可以将状态等价划分为以下 3 种情况:

- 1) $\vec{L}(q) = \vec{L}(p)$;
- 2) $\vec{L}(q) \subseteq \vec{L}(p)$;
- 3) $\vec{L}(q) \subseteq \vec{L}(p) \wedge \vec{L}(q) \subseteq \vec{L}(p)$ 。

根据定理 2 和定理 3 可知, 判定两个状态等价的问题就转化为状态之间左右语言的等价或包含关系。而计算语言的等价或包含关系是 NP-难问题。一个可行的方法就是借助模拟的概念来计算语言的等价或包含关系。

定理 4 根据定理 3, 可以将状态等价问题转换为状态间的前序关系:

- 1) $q \overset{\leftarrow}{\subseteq} p$ 而且 $p \overset{\leftarrow}{\subseteq} q$, 即状态 p 和状态 q 的右语言等价, 记作 $p \overset{\leftarrow}{\sim} q$;
- 2) $q \overset{\leftarrow}{\subseteq} p$ 而且 $p \overset{\leftarrow}{\subseteq} q$, 即状态 p 和状态 q 的左语言等价, 记作 $p \overset{\leftarrow}{\sim} q$;
- 3) $q \overset{\leftarrow}{\subseteq} p$ 而且 $p \overset{\leftarrow}{\subseteq} q$;

$$4) p \overset{\leftarrow}{\subseteq} q \text{ 而且 } p \overset{\leftarrow}{\subseteq} q。$$

3 基于左语言的 Kripke 结构的优化

在模型检测中, Kripke 结构是实际系统的抽象模拟。由于实际系统的复杂性, 因此一般 Kripke 结构的规模都很大, 所以对 Kripke 结构进行优化是一个十分重要的工作。

设 AP 为原子命题集合, AP 上的 Kripke 结构 M 为四元组 $M = (S, S_0, R, L)$, 其中: S 为有穷状态集合, $S_0 \subseteq S$ 为初始状态集合, $R \subseteq S \times S$ 为迁移关系, $L: S \rightarrow 2^{AP}$ 为标准函数。

根据定理 3, 当两个状态的左语言相同时就可以对它进行合并。由于 Kripke 结构的特点, 从初始状态出发, 可以在不构造全局状态迁移图的情况下对 Kripke 结构进行优化。这样, 最大所需的存储器只是存储优化后的 Kripke 结构所需要的存储空间。

基于左语言的 Kripke 结构的优化算法如下:

1) 如果 $s_01, s_02 \in S_0$ 而且 $L(s_01) = L(s_02)$, 则合并 S_01 和 S_02 。

2) 任取状态 $o \in S$, 如果 $p, q \in R(o)$, 而且 $L(p) = L(q)$, 则合并 p 和 q 。

根据以上算法, 设 Kripke 结构的状态数为 m 迁移的数为 n , 则算法的时间复杂度为 $O(mn)$ 。

3.1 算法示例

例 1 图 2 给出了基于左语言的 Kripke 结构优化的例子。其优化的方法分为 3 步:

- 1) 根据左语言的定义可知, 状态 2, 3, 4 的左语言相同, 因此将其合并, 得到图中的 (b)。
- 2) 根据图中 (b), 可知状态 5, 6, 7 的左语言相同, 因此可以将其合并, 得到图中的 (c)。
- 3) 根据图中 (d), 可知状态 8, 9 和 10, 11 的左语言分别相同, 因此可以将 8, 9 以及 10, 11 分别合并, 得到图中的 (d)。

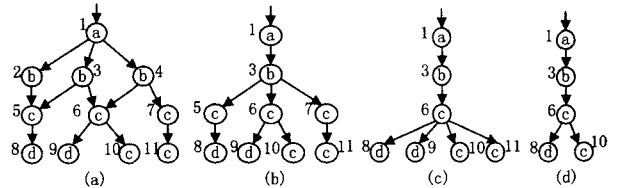


图 2 基于左语言的 Kripke 结构的优化

本例取自文献[5], 与该文献中基于直接模拟前序和小兄弟失连的方法相比较, 基于左语言的优化方法更加直接而且开销要小得多。

4 基于左右语言的 Büchi 自动机优化

有穷字符集 Σ 上的 Büchi 自动机 $A = (S, R, S_0, L, F)^{[6]}$, 其中 S 为有穷状态集合, $S_0 \subseteq S$ 为初始状态集合, $R \subseteq S \times S$ 为迁移关系, $L: S \rightarrow AP$ 为标注函数, $F \subseteq S$ 为接受条件。

定义 3 (自动机的逆) 设自动机 $A = (S, R, S_0, L, F)$, 则其逆系统为 $\tilde{A} = (S, \tilde{R}, F, L, S_0)$, 其中如果 $(p, q) \in R$, 则 $(q, p) \in \tilde{R}$ 。

定理 5 设自动机 $A = (S, R, S_0, L, F)$, 则其逆系统为 $\tilde{M} = (S, \tilde{R}, F, L, S_0)$, 如果在 A 中 $\rho \leq q$, 则 $\vec{L}(\rho) \subseteq \vec{L}(q)$; 如果在 \tilde{A} 中 $\rho \leq q$, 则 $\vec{L}(\rho) \subseteq \vec{L}(q)$ 。

由于 Büchi 自动机所接受的语言与有穷语言的自动机所

识别的语言有着本质的区别,而左右语言定义是基于有穷集的,因此即使是 Büchi 自动机和有穷语言自动机的结构完全相同,在满足定理 4 时,状态合并并不能保证合并前后的 Büchi 自动机的等价^[4]。

定理 6 设 Σ 上的 Büchi 自动机 $A=(S,R,S_0,L,F)$, 则该自动机识别的语言可以表示为

$$L(A) = \bigcup_{i \in S_0, f \in F} L(A(i, f))L(A(f, f))^\omega \quad (1)$$

根据定理 6, 要保证优化后的 Büchi 自动机与原自动机等价, 只要保证所有的 $A(i, f)$ 以及 $A(f, f)$ 的接受语言不变即可。设状态 p 和状态 q 在有穷自动机 $A(i, f)$ 和 $A(f, f)$ 中是等价的, 则 p 和 q 合并后 $L(A(i, f))$ 和 $L(A(f, f))$ 都不会发生改变, 因此, Büchi 自动机的语言就不会发生改变。因此, 对于 Büchi 自动机, 其左右语言的等价可以表达为以下定义^[4]:

定义 4 Büchi 自动机的左右语言的等价可以转换为相应的有穷输入自动机左右语言的等价。

1) $q \xrightarrow{\sim} AP$, 当且仅当对于任意的 $i \in S_0, f \in F$, 都有 $q \xrightarrow{\sim} A(i, f)p$ 而且 $q \xrightarrow{\sim} A(f, f)p$;

2) $q \xrightarrow{\sim} AP$, 当且仅当对于任意的 $i \in S_0, f \in F$, 都有 $q \xrightarrow{\sim} A(i, f)p$ 而且 $q \xrightarrow{\sim} A(f, f)p$ 。

通过以上分析, 将 Büchi 自动机的语言等价转化为了自动机的左右语言等价, 但是左右语言等价的计算是困难的。为此, 一般借助模拟的概念来完成。而普通模拟关系是语言包含的充分非必要条件, 而通过 K -模拟可以得到比普通模拟更大的模拟关系, 而且当 $k=n$ 时 K -模拟关系就是语言包含的充分必要条件。在实际应用 K -模拟进行计算的过程中, 可以根据实际情况逐步选择 K 的值。下面我们给出利用 K -模拟计算的基于左右语言的 Büchi 自动机优化算法。

根据以上分析, 可以得出以下算法:

- 1) 构建 Büchi 自动机 A 的模拟博弈图 G 。
- 2) 对任意的 $A(i, f)$ 和 $A(f, f)$ 计算相应的模拟关系。
- 3) 求以上的模拟关系集合的交, 等到相应的右语言相等的集合, 并合并相应的状态, 得到优化后的自动机 A_1' 。
- 4) 对任意的 $A(i, f)$ 和 $A(f, f)$ 的逆自动机计算相应的模拟关系。
- 5) 求以上的模拟关系集合的交, 等到相应的右语言相等的集合, 并合并相应的状态, 得到优化后的自动机 A_2' 。
- 6) 如果得到的自动机的规模适合模型检测的需求, 则退出优化, 否则 $K++$, 并转到第一步, 继续优化。

在以上算法中由于是利用直接 K -模拟实现左右语言相等的计算, 而计算直接 K -模拟的时间复杂度为 $O(n^{2k-1}m)$, 所以上优化算法的时间复杂度为 $O(n^{2k-1}m)$ 。

4.1 算法示例

例 2 为对图 3(a) 中 Büchi 自动机应用左右语言进行优化, 需要考虑以下输入为有穷字的有穷状态自动机 $A(1, 4)$ 、 $A(1, 5)$ 、 $A(1, 6)$ 、 $A(4, 4)$ 、 $A(5, 5)$ 和自动机 $A(6, 6)$ 。

图 3(b) 为 $A(1, 4)$, 而 $A(1, 5)$ 和 $A(1, 6)$ 只需将 $A(1, 4)$ 中的接受状态分别改为状态 5 和 6 即可。根据以上自动机的定义可知, 自动机 $A(1, 4)$ 、 $A(1, 5)$ 、 $A(1, 6)$ 状态 4, 5, 6 的左语言均为 $(e+f)((ab)^* + (ac)^*)a$; 图 3(c) 为自动机 $A(4, 4)$, 根据其定义可知, 状态 4, 5, 6 的左语言均为 $(b+c)((ab)^*$

$+ (ac)^*)a$; 图 3(d) 为自动机 $A(5, 5)$, 根据自动机 $A(6, 6)$ 的定义可知, 状态 4, 5, 6 的左语言均为 $b((ab)^* + (ac)^*)a$; 自动机 $A(6, 6)$ 只需要将 $A(5, 5)$ 中的初始状态和接受状态改为状态 6 即可, 根据自动机 $A(6, 6)$ 的定义可知, 状态 4, 5, 6 的左语言均为 $c((ab)^* + (ac)^*)a$ 。

根据定义 4, 可知 Büchi 自动机中状态 4, 5, 6 的左语言等价, 因此根据以上算法, 对状态 4, 5, 6 予以合并, 得到图 3 所示的自动机 (e)。

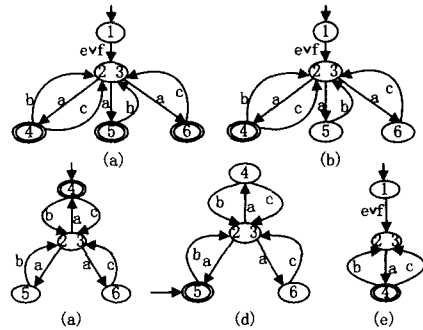


图 3 基于 K -模拟的自动机优化

对图 3(a) Büchi 自动机所识别的语言为 $(e+f)(a(b+c))^\omega$, 经过左右语言理论的优化得到识别语言 $(e+f)(a(b+c))^\omega$ 的最优 Büchi 自动机。

结束语 本文研究了利用 K -模拟计算的基于左右语言的 Kripke 结构、Büchi 自动机优化算法。由于利用左右语言的优化是基于自动机理论的优化, 这种方法可以应用在其他优化方法之后对结构进行进一步的优化。对于 Büchi 自动机优化是根据其语言特征将优化转化为对有穷语言的非确定自动机的优化, 这种方法使得优化中不必考虑无穷语言的情况, 而且计算的开销也大为降低。对于 Kripke 结构利用 K -普通模拟来实现对左右语言的计算, 应用 K -模拟可以更大程度对优化对象进行简化, 而且还可以根据实际需要选择 K 的值, 使得灵活性大大增强。

参考文献

- [1] Jiang Tao, Ravikumar B. Minimal nfa problems are hard[J]. SIAM J. Comput., 1993, 22(6): 1117-1141
- [2] Ilie L, Yu Sheng. Algorithms for computing small nfacs[C]// MFCS'02: Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science. London, UK: Springer-Verlag, 2002: 328-340
- [3] Watson B W, Daciuk J. An efficient incremental dfa minimization algorithm. Nat. Lang. Eng., 2003, 9(1): 49-64
- [4] Champarnaud J-M, Coulon F. Buchi automata reduction by means of left and right race inclusion preorders // in JM' 2004 (Conférence Internationale Journ ées Montoise d' Informatique Thérique). 2004: 1-7
- [5] Bustan D, Grumberg O. Simulation-based minimization. ACM Trans. Comput. Logic, 2003, 4(2): 181-206
- [6] Farwer B. Omega-Automata // Grädel E, Thomas W, Wilke Th, eds. Automata, Logics, and Infinite Games. volume 2500 of Lecture Notes in Computer Science. Springer-Verlag, 2002: 3-21