基于子集构造法的优化的NFA确定化算法

任平红,陈 蠢,曹宝香,禹继国
(山东曲阜师范大学 计算机科学学院,山东 日照 276826)

摘 要:使用子集构造法对非确定有限自动机进行确定化的过程中存在大量重复计算的问题。为解决此问题,基于非确定有限自动机的特点并针对子集构造法的不足,提出了一种优化的非确定有限自动机确定化算法。首先定义了识别符的有效引出状态集概念并证明了 ε -closure 的并定理以保证算法的正确性,其次给出了用于避免重复计算的识别符的有效引出状态集的构造子算法和单状态集的 ε -closure 的求算子算法,基于这两个子算法给出了优化的非确定有限自动机确定化算法,最后将算法应用于实例,实验结果表明计算量远小于子集构造法的计算量。相比子集构造法,算法能更有效地对非确定有限自动机进行确定化。

关键词:子集构造法;非确定有限自动机;优化的;确定化算法
中图分类号:TP301.1 **文献标识码**:A **文章编号**:1673-629X(2011)01-0070-04

An Optimized Algorithm for Transition from NFA to DFA Based on Subset Construction Method

REN Ping-hong, CHEN Chu, CAO Bao-xiang, YU Ji-guo
(College of Computer Science, Qufu Normal University, Rizhao 276826, China)

Abstract: The problem of repetitive computing exits in the process of transition from non-deterministic finite automata to deterministic finite automata using the subset construction method. To solve this problem, an optimized algorithm for transition from NFA to DFA is put forward on the basis of characters of NFA and according to shortcomings of the subset construction method. First, the concept that effective source state set for a mark symbol is defined and the theorem that combination of ε -closure is proved. The theorem guarantees the rightness of the following algorithms. Second, two sub algorithms that construction of effective source state set for a mark symbol and that computing ε -closure of a set with only one state are given. Based on these sub algorithms, the optimized algorithm for transition from NFA to DFA is given. In the end, an experiment is carried out and the result shows that computing amount of this algorithm is far less than that of subset construction method. Comparing to subset construction method, this algorithm can convert NFA to DFA more efficiently.

Key words: subset construction method; non-deterministic finite automata; optimized; algorithm for transition from NFA to DFA

0 引 言

作为计算机科学的基础,有限自动机(Finite Automata, FA)可以抽象绝大多数计算机领域的有限系统,广泛应用于自然语言处理[1]和串匹配[2]等诸多方面。非确定有限自动机(Non-deterministic Finite Automata, NFA)是FA理论的重要组成部分,NFA的化简[3,4]和确定化[5~8]具有重要的理论和实际意义。目前广为采用的子集构造法[9,10]虽然能够正确地完成NFA的确定化,但是在确定化过程中对于NFA状态集存在 ε -closure 重复计算和由于对非 ε 转换的判断而引起的重复计算等问题[5,8,9]。当NFA状态集元素数量较多时重复计算的问题非常严重。为解决此问题,提出一种基于子集构造法的优化的NFA确定化算法,可以有效地避免重复计算的问题。

收稿日期:2010-05-31;**修回日期**:2010-09-03
基金项目:山东省优秀中青年科学家奖励基金(2005BS01016)
作者简介:任平红(1980-),女,山东德州人,讲师,硕士,研究方向为计算机图形图像处理、算法;曹宝香,教授,研究方向为图形图像处理、算法;禹继国,教授,博士,研究方向为算法。

1 识别符的有效引出状态集概念和 ε - closure 并定理

为规整起见,首先给出NFA的一般定义并在全文中统一使用其定义中的符号。

定义1[11]:NFA定义为一个四元组:$M=(Q,\Sigma,\delta,q_0,F)$。其中:$Q$ 是状态集,Σ 是基本字母表,δ 是状态转换函数($Q\times(\Sigma\cup\{\varepsilon\})\rightarrow 2^Q$),$q_0$ 是开始状态,F 是终态集。

基于NFA的定义,定义2给出识别符的有效引出

状态集的定义。

定义2:对于识别符 a ($a \in \Sigma$) 有效的引出状态集 $E[a]$ 定义为:$E[a] = \{q \mid q \in Q \wedge \delta(q,a) \neq \varnothing\}$。

识别符的有效引出状态集 $E[a]$ 指出了 NFA 中哪些状态具有标识为 a 的转移,对应于 NFA 状态图中具有标识为 a 的引出弧的状态集合。在 FA 理论中,由于需要对确定有限自动机(Deterministic Finite Automata, DFA)进行最小化,而使用分割算法对非完全 DFA 直接进行最小化存在一些问题[12],所以非完全 DFA 需要转换为完全 DFA[13]。NFA 的确定化不存在同样的问题,所以不需要将非完全的 NFA 转换为完全的 NFA。对于一个非完全的 NFA, $E[a] \subseteq Q$。在子集构造法中,扫描状态转换矩阵会造成对 $(Q - E[a])$ 这一部分状态集的无效判断和计算,如果能计算出 $E[a]$ 就可以避免这些无效判断和计算。在第二节中将给出计算 $E[a]$ 的子算法。

定义 3[14]:NFA 单状态集 $\{s\}$ ($s \in Q$) 的 ε - closure 定义为:$\varepsilon - \text{closure}(\{s\}) = \{p \mid p \in Q \wedge p \in \delta(s, \varepsilon*)\}$;NFA 状态集 T ($T \subseteq Q$) 的 ε - closure 定义为:$\varepsilon - \text{closure}(T) = \{s \mid s \in \varepsilon - \text{closure}(t) \wedge t \in T\}$。

定义 3 中 NFA 单状态集的 ε - closure 和状态集的 ε - closure 都是无序集合,基于此性质和定义 3 可以得如下定理和推论:

定理 1:$\varepsilon - \text{closure}(T) = \bigcup_{t \in T} \varepsilon - \text{closure}(\{t\})$。

证明:根据定义 3,$\varepsilon - \text{closure}(T) = \{s \mid s \in \varepsilon - \text{closure}(t) \wedge t \in T\} = \bigcup_{t \in T} \{p \mid p \in Q \wedge p \in \delta(t, \varepsilon*)\} = \bigcup_{t \in T} \varepsilon - \text{closure}(\{t\})$。

定理 1 说明求解状态集的 ε - closure 可以转变为求解单状态集的 ε - closure,子集构造法中 NFA 每个状态集的 ε - closure 不必完全重新计算,可以依托于 NFA 每个单状态集 $\{t\}$ ($t \in Q$) 的计算来完成。

由定理 1 可得如下推论:

推论 1:$\varepsilon - \text{closure}(T) = \bigcup_{T' \subseteq T} \varepsilon - \text{closure}(T')$。

证明:$\varepsilon - \text{closure}(T) = \{s \mid s \in \varepsilon - \text{closure}(t) \wedge t \in T\} = \bigcup_{T' \subseteq T} \{s \mid s \in \varepsilon - \text{closure}(t) \wedge t \in T'\} = \bigcup_{T' \subseteq T} \varepsilon - \text{closure}(T')$。

推论 1 说明求解状态集 T 的 ε - closure 可以转变为求解其子集 T' 的 ε - closure,当 $|T'| = 1$ 时就转变为定理 1 中求解的情况。

子集构造法中不同状态集的 ε - closure 计算过程中可能存在重复计算的部分。当状态集元素数量较多时,重复计算现象非常严重。在 NFA 确定化过程中应用定理 1 或推论 1 的结论可以避免重复计算以提高计算的效率。为应用定理 1 或推论 1,需要计算单状态集的 ε - closure,单状态集的 ε - closure 的求算子算法将在第二节中给出。

2　识别符的有效引出状态集的构造子算法和单状态集的 ε -closure 求算子算法

基于定义 2、定理 1 和推论 1,给出识别符的有效引出状态集 $E[a]$ 的构造子算法和单状态集的 ε - closure 的求算子算法,并在第三节中用于基于子集构造法的优化的 NFA 确定化算法中。

2.1　识别符的有效引出状态集的构造子算法

算法思想:按列扫描 NFA 的状态转换矩阵一遍,当到达状态在 Q 中时将出发状态加入对应识别符的有效引出状态集 $E[a]$ 中,一遍扫描后对于 Σ 中任意识别符 a 的 $E[a]$ 都计算完毕。

识别符的有效引出状态集 $E[a]$ 的构造子算法的类 C 伪代码具体描述如下:

```
输入:NFA M = (Q, Σ, δ, q0, F)。
输出:E[a] ( ∀ a ∈ Σ)。
方法:
Σ' = Σ;
while (Σ' ≠ ∅){
∀ a ∈ Σ';
for ( ∀ q ∈ Q)
if (δ(q,a) ≠ ∅)E[a] += {q};
Σ' -= {a};
}
```

通过识别符的有效引出状态集 $E[a]$ 可避免 NFA 状态集(对应 DFA 的某个状态)中没有标识为 a 的引出弧的状态的无效判断和计算。

2.2　单状态集的 ε-closure 的求算子算法

算法思想:对于 NFA 的任意一个状态 q($q \in Q$) 构成的单状态集 $\{q\}$,如果 q 有一个或一个以上的 ε - 转换到达 Q 中的状态,则 $\varepsilon - \text{closure}(\{q\})$ 只需引用这些状态分别对应的单状态集的 ε - closure 的计算结果及 $\{q\}$ 的并集即可。

单状态集 $\{q\}$ 的 $\varepsilon - \text{closure}(\{q\})$ 求算子算法的类 C 伪代码具体描述如下:

```
输入:NFA M = (Q, Σ, δ, q0, F)。
输出:ε - closure({q}) ( ∀q ∈ Q)。
方法:
Q' = Q;
while (Q' ≠ ∅){
for ( ∀q ∈ Q')
if (δ(q, ε) == ∅) ε - closure({q}) = ∅
else if (δ(q, ε) == T) ε - closure({q}) = {q} +
∑ ε - closure({t});
```

Q' -= {q} ;

}

单状态集 ε - closure({q}) 的计算采用了递归计算，ε - closure({q}) 只依赖于 $\delta(q, \varepsilon)$ 集合中的元素的单状态集的 ε - closure。当算法结束时，任意单状态集的 ε - closure 都可以被计算出来。根据定理 1，任意 NFA 状态集的 ε - closure 可通过其元素状态的单状态集的 ε - closure 的合并得到。这样就避免子集构造法中通过入栈出栈[10] 把每一个 NFA 状态集的每一个状态都计算 ε - closure 而带来的大量的重复计算。

3 优化的 NFA 确定化算法

以子集构造法的基本方法为基础，应用识别符的有效引出状态集 $E[a]$ 的构造子算法和单状态集的 ε - closure 的求算子算法，得到优化的 NFA 确定化算法，其类 C 伪代码具体描述如下：

输入：NFA $M = (Q, \Sigma, \delta, q_0, F)$。

输出：DFA $M' = (Q', \Sigma, \delta', q_0', F')$。

方法：

调用识别符的有效引出状态集的构造子算法；

调用单状态集的 ε - closure 的求算子算法；

Q' = {ε - closure({q_0})} ;

unmark = {ε - closure({q_0})} ;

mark = $\varnothing$;while (unmark $\neq \varnothing$) {

$\forall\ U \in$ unmark;

for ($\forall\ a \in \Sigma$) {

$ES = U \cap E[a]$;

$T = \delta(ES, a)$;

$N = \sum_{t \in T} \varepsilon$ - closure({t}) ;

if ! ($N \in Q'$) {

Q' += {N} ;

$\delta'[U, a] = N$;

}

}

unmark -= {U} ;

mark += {U} ;

}

重命名 Q' 和 δ' 对应元素，其中 ε - closure({q_0}) 重命名为 q_0' ;

$F' = \{f \mid \forall M \in$ mark, $(M \cap F \neq \varnothing) \wedge (M$ 重命名为 $f)\}$;

得到 DFA M' 并且 $M' \equiv M$。

算法开始时通过调用识别符的有效引出状态集的构造子算法求得基本字母表 Σ 中每一个基本字母的有效引出状态集，通过调用单状态集的 ε - closure 的计算子算法得到 NFA 状态集 Q 中每一个状态作为唯一元素的状态集的 ε - closure 供算法的后续步骤使用。算法中 unmark 是一个存放未被处理的 NFA 状态集的集合，mark 是一个存放已经处理的 NFA 的状态集的集合。算法通过一个 while 循环完成从 NFA 状态集及其转换表向对应 DFA 状态及其转换表的转变。此循环中 ES 是 NFA 状态集中具有标识为 a 的引出弧的状态子集，T 是状态子集中的状态识别 a 后转换到的状态集，N 是 T 中状态的单状态集的 ε - closure 的并集。当 N 与已知状态集不同时将其加入 DFA 的状态集 Q' 中（对应于 DFA 的一个状态）并把相应的转换函数 $\delta'[U, a] = N$ 加入 DFA 的 δ' 中。每一次循环的最后把该次循环处理的 NFA 的状态集从 unmark 集中移除并移入 mark 集中。

4 实例分析

通过下面的实例说明算法的应用及对避免重复计算的有效性。设 NFA $M = (Q, \Sigma, \delta, q_0, F)$，其中：$Q = \{0,1,2,3,4,5,6,7,8,9,10\}$，$\Sigma = \{a,b\}$，$\delta = \{\delta(0,a) = \varnothing, \delta(0,b) = \varnothing, \delta(0,\varepsilon) = \{1,7\}, \delta(1,a) = \varnothing, \delta(1,b) = \varnothing, \delta(1,\varepsilon) = \{2,4\}, \delta(2,a) = \{3\}, \delta(2,b) = \varnothing, \delta(2,\varepsilon) = \varnothing, \delta(3,a) = \varnothing, \delta(3,b) = \varnothing, \delta(3,\varepsilon) = \{6\}, \delta(4,a) = \varnothing, \delta(4,b) = \{5\}, \delta(4,\varepsilon) = \varnothing, \delta(5,a) = \varnothing, \delta(5,b) = \varnothing, \delta(5,\varepsilon) = \{6\}, \delta(6,a) = \varnothing, \delta(6,b) = \varnothing, \delta(6,\varepsilon) = \{1,7\}, \delta(7,a) = \{8\}, \delta(7,b) = \varnothing, \delta(7,\varepsilon) = \varnothing, \delta(8,a) = \varnothing, \delta(8,b) = \{9\}, \delta(8,\varepsilon) = \varnothing, \delta(9,a) = \varnothing, \delta(9,b) = \{10\}, \delta(9,\varepsilon) = \varnothing, \delta(10,a) = \varnothing, \delta(10,b) = \varnothing, \delta(10,\varepsilon) = \varnothing\}$，$q_0 = 0$，$F = \{10\}$。

根据识别符的有效引出状态集的构造子算法可得：$E[a] = \{2,7\}$，$E[b] = \{4,8,9\}$。根据单状态集的 ε - closure 的求算子算法可得：ε - closure({0}) = {0} $\cup\ \varepsilon$ - closure({1}) $\cup\ \varepsilon$ - closure({7}), ε - closure({1}) = {1} $\cup\ \varepsilon$ - closure({2}) $\cup\ \varepsilon$ - closure({4}), ε - closure({2}) = {2}, ε - closure({3}) = {3} $\cup\ \varepsilon$ - closure({6}), ε - closure({4}) = {4}, ε - closure({5}) = {5} $\cup\ \varepsilon$ - closure({6}), ε - closure({6}) = {6} $\cup\ \varepsilon$ - closure({1}) $\cup\ \varepsilon$ - closure({7}), ε - closure({7}) = {7}, ε - closure({8}) = {8}, ε - closure({9}) = {9}, ε - closure({10}) = {10}。代入可得：ε - closure({0}) = {0,1,2,4,7}, ε - closure({1}) = {1,2,4}, ε - closure({2}) = {2}, ε - closure({3}) = {1,2,3,4,6,7}, ε - closure({4}) = {4}, ε - closure({5}) = {1,2,4,5,6,7}, ε - closure({6}) = {1,2,4,6,7}, ε - closure({7}) = {7}, ε -

closure({8}) = {8},ε - closure({9}) = {9},ε - closure({10}) = {10}。

算法初始时,Q' = {ε - closure({q_0})} = {ε - closure({0})} = {{0,1,2,4,7}},unmark = {ε - closure({q_0})} = {ε - closure({0})} = {{0,1,2,4,7}},mark = ∅。由于 unmark ≠ ∅,所以进入 while 循环。第一次循环:U = {0,1,2,4,7},对于 a,ES = {0,1,2,4,7} ∩ $E[a]$ = {0,1,2,4,7} ∩ {2,7} = {2,7},T = $\delta(ES, a)$ = δ({2,7},a) = {3,8},N = ε - closure({3}) + ε - closure({8}) = {1,2,3,4,6,7,8},Q' = {{0,1,2,4,7}, {1,2,3,4,6,7,8}},δ'[{0,1,2,4,7},a] = {1,2,3,4,6,7,8};对于 b,ES = {0,1,2,4,7} ∩ $E[b]$ = {0,1,2,4,7} ∩ {4,8,9} = {4},T = $\delta(ES, a)$ = δ({4},b) = {5},N = ε - closure({5}) = {1,2,4,5,6,7},Q' = {{0,1,2,4,7},{1,2,3,4,6,7,8},{1,2,4,5,6,7}},δ'[{0,1,2,4,7},b] = {1,2,4,5,6,7}。循环执行 5 次终止后可得结果如表 1 所示。

表 1　DFA M' 的转换表

NFA 状态集	对应 DFA 状态	a	b
{0,1,2,4,7}	A	B	C
{1,2,3,4,6,7,8}	B	B	D
{1,2,4,5,6,7}	C	B	C
{1,2,4,5,6,7,9}	D	B	E
{1,2,4,5,6,7,10}	E	B	C

DFA M' = (Q', Σ, δ', q_0', F'),其中 Q' = {A, B,C,D,E},Σ = {a,b},δ' = {$\delta'(A,a)=B$, $\delta'(A,b)=C$, $\delta'(B,a)=B$, $\delta'(B,b)=D$, $\delta'(C,a)=B$, $\delta'(C,b)=C$, $\delta'(D,a)=B$, $\delta'(D,b)=E$, $\delta'(E,a)=B$, $\delta'(E,b)=C$},q_0' = A,F' = {E}。

所得 DFA M' 与子集法所得结果[10]一致,而且避免了重复判断和重复计算,如表 2 所示。

表 2　子集法与优化算法的计算量比较

NFA 状态集	Ja	Jb	Ca	Cb
{0,1,2,4,7}	6/2	5/1	12/7	11/5
{1,2,3,4,6,7,8}	8/2	8/2	12/7	12/6
{1,2,4,5,6,7}	7/2	6/1	12/7	11/5
{1,2,4,5,6,7,9}	8/2	8/2	12/7	12/6
{1,2,4,5,6,7,10}	8/2	7/1	12/7	11/5

表 2 中 Ja 表示对 NFA 某状态集的 I_a[15] 中 J 的计算量,Jb 表示对 NFA 某状态集的 I_b[15] 中 J 的计算量,Ca 表示在 I_a 中求 ε - closure 的计算量,Cb 表示在 I_b 中求 ε - closure 的计算量,r_1/r_2 中的 r_1 表示子集法的计算量,r_2 表示优化算法的计算量 。

5　结束语

应用子集构造法进行 NFA 确定化的过程易于理解,但存在重复计算的问题。基于子集构造法,结合识别符的有效引出状态集概念和 ε -closure 的并结论,给出了优化的 NFA 确定化算法。通过算法的实例应用,说明了算法对避免重复计算的有效性。进一步的工作是基于 ε -closure 状态性质分析的 NFA 的约简。

参考文献:

[1] 蔡增玉, 刘书如, 张建伟,等. 汉字模糊有空自动机的研究[J]. 计算机技术与发展,2008,18(3):89-91.

[2] 陈　倩. 一种基于有限自动机的快速串匹配算法[J]. 计算机技术与发展,2009,19(1):131-133.

[3] Ilie L, Yu S. Reducing NFAs by Invariant Equivalences [J]. Theoret. Comput Sci,2003,306:373-390.

[4] Melnikov B F. Once More about the State-Minimization of the Nondeterministic Finite Automata [J]. Korean J. Comput. Appl. Math,2000,7(3):655-662.

[5] 周启海. NFA→FA→GFA 自动机转换算法[J]. 电子科技大学学报,2005,34(3):363-365.

[6] 张超群, 周永权. 两种 εNDFA 确定化文法的讨论[J]. 广西民族学院学报,2006,12(4):77-80.

[7] 毛红梅, 聂承启. 一种将 NFA 到最小化 DFA 的方法[J]. 计算机与现代化,2004(10):6-7.

[8] 孙玉强, 刘三阳, 王明斐,等. 有限状态自动机的并行确定化及过程分析[J]. 计算机科学,2006,33(10):293-295.

[9] Gertjan van Noord. Treatment of Epsilon Moves in Subset Construction [J]. Computational Linguistics, 2000, 26(1): 61-76.

[10] Aho A V, Lam M S, Sethi R, et al. Compilers: Principles, Techniques, and Tools [M]. 2nd edition. [s. l.]. Addison-Wesley, 2007:152-156.

[11] Peter Linz. An Introduction to Formal Languages and Automata[M]. 3rd edition. [s. l.]. Jones and Bartlett Publishers, 2001:48-49.

[12] 周时阳, 祝建华. DFA 最小化算法研究[J]. 计算机工程与科学,2007,29(3):60-62.

[13] Chen C, Ren Pinghong, Yu Jiguo, et al. An Algorithm for Minimizing a Completely Deterministic Finite Automaton Based on State Transition Inverse Mapping [C]//In: Feng Gao, Xijun Zhu, ed. Proceedings of International Workshop on Information Security and Application. Qingdao: Academy Publisher, 2009:417-420.

[14] 陈意云,张　昱. 编译原理[M]. 第 3 版. 北京:高等教育出版社, 2008.

[15] 陈火旺,刘春林,谭庆平,等. 程序设计语言编译原理[M]. 第 3 版. 北京:国防工业出版社,2006.