[16] Zhao J, Li X, Zheng T, Zheng G. Removing irrelevant atomic formulas for checking timed automata efficiently. In: Proc. of the Int'l Conf. on Formal Modeling and Analysis of Timed Systems. Berlin, Heidelberg: Springer-Verlag, 2003. 34–45. [doi: 10.1007/978-3-540-40903-8_4]

[17] Zhao J, Li X, Zheng G. A quadratic-time DBM-based successor algorithm for checking timed automata. Information Processing Letters, 2005,96(3):101–105. [doi: 10.1016/j.ipl.2005.05.027]

[18] Lin H, Yi W. A proof system for timed automata. In: Proc. of the Int'l Conf. on Foundations of Software Science and Computation Structures. Berlin, Heidelberg: Springer-Verlag, 2000. 208–222. [doi: 10.1007/3-540-46432-8_14]

[19] Lin H, Yi W. Axiomatising timed automata. Acta informatica, 2002,38(4):277–305. [doi: 10.1007/s236-002-8035-2]

[20] Doyen L, Juhl L, Larsen KG, Markey N, Shirmohammadi M. Synchronizing words for weighted and timed automata. In: Proc. of the Int'l Conf. on Foundation of Software Technology and Theoretical Computer Science. 2014. 121–132. [doi: 10.4230/LIPIcs.FSTTCS.2014.121]

[21] Doyen L, Juhl L, Larsen KG, Markey N, Shirmohammadi M. Synchronizing words for timed and weighted automata. Research Report, LSV-13-15(version 2), Laboratoire Spécification et Vérification, ENS Cachan, France, 2014. 28.

[22] Eppstein D. Reset sequences for monotonic automata. SIAM Journal on Computing, 1990,19(3):500–510. [doi: 10.1137/0219033]

[23] Laroussinie F, Markey N, Schnoebelen P. Model checking timed automata with one or two clocks. In: Proc. of the Int'l Conf. on Concurrency Theory. Berlin, Heidelberg: Springer-Verlag, 2004. 387–401. [doi: 10.1007/978-3-540-28644-8_25]

[24] Fearnley J, Jurdziński M. Reachability in two-clock timed automata is PSPACE-complete. Information and Computation, 2015,243: 26–36. [doi: 10.1016/j.ic.2014.12.004]

[25] Courcoubetis C, Yannakakis M. Minimum and maximum delay problems in real-time systems. Formal Methods in System Design, 1992,1(4):385–415. [doi: 10.1007/BF00709157]

[26] Haase C, Ouaknine J, Worrell J. On the relationship between reachability problems in timed and counter automata. In: Proc. of the Int'l Workshop on Reachability Problems. Berlin, Heidelberg: Springer-Verlag, 2012. 54–65. [doi: 10.1007/978-3-642-33512-9_6]

[27] Haase C, Ouaknine J, Worrell J. Relating reachability problems in timed and counter automata. Fundamenta Informaticae, 2016,143(3-4):317–338. [doi: 10.3233/FI-2016-1316]

[28] Martyugin P. Complexity of problems concerning carefully synchronizing words for PFA and directing words for NFA. In: Proc. of the Computer Science Symp. in Russia. 2010. 288–302. [doi: 10.1007/978-3-642-13182-0_27]

[29] Martyugin P. Computational complexity of certain problems related to carefully synchronizing words for partial automata and directing words for nondeterministic automata. Theory of Computing Systems, 2014,54(2):293–304. [doi: 10.1007/s00224-013-9516-6]

[30] Babari P, Quaas K, Shirmohammadi M. Synchronizing data words for register automata. In: Proc. of the LIPIcs-Leibniz Int'l in Informatics. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016. 58. [doi: 10.4230/LIPIcs.MFCS.2016.15]

[31] Figueira D, Hofman P, Lasota S. Relating timed and register automata. Mathematical Structures in Computer Science, 2016,26(6): 993–1021. [doi: 10.1017/S0960129514000322]

[32] Imreh B, Steinby M. Directable nondeterministic automata. Acta Cybernetica, 1999,14(1):105–115.

[33] Olschewski J, Ummels M. The complexity of finding reset words in finite automata. Mathematical Foundations of Computer Science, 2010, 568–579. [doi: 10.1007/978-3-642-15155-2_50]

[34] Pin J. On two combinatorial problems arising from automata theory. North-holland Mathematics Studies, 1983, 535–548. [doi: 10.1016/S0304-0208(08)73432-7]

[35] Berlinkov MV. Approximating the minimum length of synchronizing words is hard. In: Proc. of the Computer Science Symp. in Russia. 2010. 37–47. [doi: 10.1007/978-3-642-13182-0_4]

[36] Gazdag Z, Iván S, Nagy-György J. Improved upper bounds on synchronizing nondeterministic automata. Information Processing Letters, 2009,109(17):986–990. [doi: 10.1016/j.ipl.2009.05.007]

[37] Doyen L, Massart T, Shirmohammadi M. Infinite synchronizing words for probabilistic automata. In: Proc. of the Int'l Symp. on Mathematical Foundations of Computer Science. Berlin, Heidelberg: Springer-Verlag, 2011. 278–289. [doi: 10.1007/978-3-642-22993-0_27]

[38] Caucal D. Synchronization of pushdown automata. In: Proc. of the Int'l Conf. on Developments in Language Theory. Berlin, Heidelberg: Springer-Verlag, 2006. 120–132. [doi: 10.1007/11779148_12]

[39] Chistikov D, Martyugin P, Shirmohammadi M. Synchronizing automata over nested words. In: Proc. of the Int'l Conf. on Foundations of Software Science and Computation Structures. Berlin, Heidelberg: Springer-Verlag, 2016. 252–26. [doi: 10.1007/978-3-662-49630-5_15]
[40] Quaas K, Shirmohammadi M, Worrell J. Revisiting reachability in timed automata. In: Proc. of the 32nd Annual ACM/IEEE Symp. on Logic in Computer Science (LICS). IEEE, 2017. 1–12. [doi: 10.1109/LICS.2017.8005098]
[41] Bersani MM, Rossi M, San Pietro P. A logical characterization of timed regular languages. Theoretical Computer Science, 2017, 658:46–59. [doi: 10.1016/j.tcs.2016.07.020]
[42] Maler O, Nickovic D, Pnueli A. From MITL to timed automata. In: Proc. of the Int'l Conf. on Formal Modeling and Analysis of Timed Systems. Berlin, Heidelberg: Springer-Verlag, 2006. 274–289. [doi: 10.1007/11867340_20]
[43] Ničković D, Piterman N. From MTL to deterministic timed automata. In: Proc. of the Int'l Conf. on Formal Modeling and Analysis of Timed Systems. Berlin, Heidelberg: Springer-Verlag, 2010. 152–167. [doi: 10.1007/978-3-642-15297-9_13]
[44] Papadimitriou CH. Computational Complexity. John Wiley and Sons Ltd., 2003.
[45] Schmitz S. Complexity hierarchies beyond elementary. ACM Trans. on Computation Theory (TOCT), 2016,8(1):3:1–3:36. [doi: 10.1145/2858784]

附中文参考文献:

[7] 宋富,吴志林.面向无穷数据的形式模型综述.软件学报,2016,27(3):1–9. http://www.jos.org.cn/1000-9825/4989.htm [doi: 10.13328/j.cnki.jos.004989]

附录

引理 1 的证明:只有经过*-变迁才能将 $\mathcal{A}_1,\ldots,\mathcal{A}_k$ 中的状态最终融合到 end 位置.因此,$(t_2,*)$应出现在 u 中.简单起见,可令 $t_2=0$.显然,如果 $post(L_B\times C_B,(0,*))$有定义,则 $post(L_B\times C_B,(0,*))=(\text{end},\vec{0})$, 它是单元集.如果对于 $n\leqslant|u|,u[n]=(t_2,*)$,那么,$u$ 的前缀$u[1,n]$也是 $\mathcal{B}$ 的一个仔细重置序列.由于假设u是最短的仔细重置序列,那么,$n=|u|$且$(t_2,*)$在 u 中仅出现 1 次,它位于 u 的末端.

因为 u 必须从 begin 位置开始,Σ中的字母和符号*在 begin 位置上均没有定义,所以 $u[1]=(t_1,\#)$,简单起见,可令 $t_1=0$,于是对于某个 $w\in((\mathbb{R}_{\geqslant 0},\Sigma\cup\{\#\}))^*$, u 可以被表示为 $u=(0,\#)\cdot w\cdot(t_2,*)$ 的形式.因为对每个 $i\in\{1,\ldots,k\}$,满足 $post(L_B\times C_B,(0,\#))\cap(L_i\times C_B)=\{(l_i,\vec{0})\}$, 所以 $|post(L_B\times C_B,(0,\#))\cap(L_i\times C_B)|=1$. Σ中的字母和符号#都定义在集合$L_i\times C_B$上,并且这些变迁将$L_i\times C_B$映射到$L_i\times C_B$自身.因此,对 $\forall m\in\{1,\ldots,|u|-1\}$, 有 $|loc(post(L_B\times C_B,u[1,m]))\cap L_i)|=1$. 于是,对于某个 m,如果 $u[m]=(0,\#)$,那么, $loc(post(L_B\times C_B,u[1,m]))=\{l_1,\ldots,l_k\}=loc(post(L_B\times C_B,(0,\#)))$, 这样, $(0,\#)\cdot u[m+1,|u|]$ 也是 $\mathcal{B}$ 的一个仔细重置序列.由于已假设 u 是最短的仔细重置序列,所以,$(0,\#)$变迁在 u 中只会出现 1 次,因此,$(0,\#)$不会出现在 w 中.

于是, $u=(t_1,\#)\cdot w\cdot(t_2,*)$, 其中, $w\in((\mathbb{R}_{\geqslant 0},\Sigma))^*,t_1,t_2\in\mathbb{R}_{\geqslant 0}$. □

命题 1 的证明:由于时间自动机对于交运算是封闭的(见文献[9]中的定理 3.15),所以 k 个时间自动机 $\mathcal{A}_1\ldots\mathcal{A}_k$ 的交运算形成新的时间自动机 $\mathcal{A}=\bigcap_{i=1}^{k}\mathcal{A}_i$. 由文献[9]定理 3.15 证明中的构造过程得到,该乘积自动机 $\mathcal{A}$ 的状态个数是 $k\cdot\prod_{i=1}^{k}|L_i|$, 时钟个数是 $\sum_{i=1}^{k}|C_i|$, 变迁集的大小是 $k\cdot\prod_{i=1}^{k}|E_i|$. 注意,假设对常量采用二进制编码,$|E|$包括时钟约束的长度.

再由时间正则语言 $\mathcal{L}(\mathcal{A})$ 非空的判定问题是 PSPACE-完全的(具体细节参考文献[9]中的定理 4.17 的证明)可证.

注意,该命题 PSPACE 成员性可以看成是“PSPACE 对交运算封闭”[44]的具体表现形式. □

引理 2 的证明:(1)~(3)均可由变迁函数的定义直接得出. □

注意,(1)和(2)意味着 a-变迁不改变位置所在列号,但会使位置的行号加 1,如果位置已在最后一行,那么该位置形成自环.(3)意味着 b-变迁会使位置进入第 0 行.

引理 3 的证明:

(1) 由于 $u \in ((\mathbb{R}_{\geqslant 0}, \{a,b\}))^* \setminus ((\mathbb{R}_{\geqslant 0}, \{a\}))^*$, (t_1,b)-变迁出现在序列 u 中, $t_1 \in \mathbb{R}_{\geqslant 0}$. 设 $u[h_1]$是(t_1,b)在 u 中的首次出现.由引理 2 的(3)得到 $loc(post(L_C \times C_C, b)) \subseteq Row_0$. 于是, $loc(post(L_C \times C_C, u[1,h_1])) \subseteq Row_0$, 所以, $|Col_i \cap loc(post(L_C \times C_C, u[1,h_1]))| \leqslant 1$. 假设对于某个 $j \in \{1,...,|u|-1\}$, 有 $|Col_i \cap loc(post(L_C \times C_C, u[1,j]))| \leqslant 1$. 如果 $u[j+1] = (t_2,a)$, $t_2 \in \mathbb{R}_{\geqslant 0}$, 那么,由引理 2 得到,对每个 $i \in \{1,\dots,n\}$, $loc(post(Col_i \times C_C, (t_2,a))) \subseteq Col_i$. 因此,

$$|loc(post(L_C \times C_C, u[1,j] \cdot (t_2,a))) \cap Col_i| \subseteq |loc(post(L_C \times C_C, u[1,j])) \cap Col_i| \leqslant 1.$$

如果 $u[j+1] = (t_1,b)$, 那么, $Col_i \cap loc(post(L_C \times C_C, u[j+1])) \subseteq Row_0$. 这意味着 $|loc(post(L_C \times C_C, u[j+1])) \cap Col_i| \leqslant 1$, 故 $|Col_i \cap loc(post(L_C \times C_C, u))| \leqslant 1$.

(2) 设 $i \in \{1,\dots,n\}$.由(1)得到 $|Col_i \cap loc(post(L_C \times C_C, u))| \leqslant 1$. 由引理 2 的(1)得到:对于 $i \in \{1,\dots,n\}$, $loc(post(Col_i \times C_C, a)) \subseteq Col_i$. a-变迁在 L_C 中的任何位置上都有定义.因此,如果 $|Col_i \cap loc(post(L_C \times C_C, u))| = 1$, 那么 $|Col_i \cap loc(post(L_C \times C_C, u \cdot (t,a)))| = 1$, $t \in \mathbb{R}_{\geqslant 0}$. 于是,

$$\begin{aligned}|loc(post(L_C \times C_C, u))| &= \sum_{i=1}^{n} |loc(post(L_C \times C_C, u)) \cap Col_i| \\ &= \sum_{i=1}^{n} |loc(post(L_C \times C_C, u \cdot (t,a))) \cap Col_i| \\ &= |loc(post(L_C \times C_C, u \cdot (t,a)))|.\end{aligned}$$

(3) 由引理 2 的(3)得到: $loc(post(L_C \times C_C, b)) \subseteq Row_0$. 因此 $loc(post(L_C \times C_C, u \cdot (t,b))) \subseteq loc(post(L_C \times C_C, b)) \subseteq Row_0$.

引理 4 的证明:设 $u \in ((\mathbb{R}_{\geqslant 0}, \{a,b\}^* b))^*$, 那么对于某些 $h_1,...,h_r \in \mathbb{N}$, 有 $u = (t_1, a^{h_1} b)(t_2, a^{h_2} b)...(t_r, a^{h_r} b)$ 成立,其中, $t_1,...,t_r \in \mathbb{R}_{\geqslant 0}$. 对于某个 $k \in \{1,\dots,r\}$, 设 $u_k = (t_1, a^{h_1} b)(t_2, a^{h_2} b)...(t_k, a^{h_k} b)$.

归纳基础.当 k=1 时,如果 $post(L_C \times C_C, u)$ 有定义,那么 $post(L_C \times C_C, (t_1, a^{h_1} b)) = post(L_C \times C_C, u_1)$ 也有定义.这只能出现在 $h_1 \geqslant m+1$ 的情况下,因为如果 $h_1 < m+1$,那么 $l_{i,j}^C \in loc(post(L_C \times C_C), a^{h_1})$, 随后的 b-变迁在位置 $l_{i,j}^C$ 上不一定有定义,矛盾.因此, $g(u_1) = g(t_1, a^{h_1} b) = f(t_1, a^{h_1} b) = (t_1, c_{m+1}) = (t_1, \#)$. 可以验证 $I(u_1) = I((t_1, a^{h_1} b)) = J((t_1, c_{m+1}))$.

归纳步骤.当 k>1 时,假设 $I(u_{k-1}) = J(g(u_{k-1}))$, 证明 $I(u_k) = J(g(u_k))$. 设 $i \in I(u_{k-1}) = J(g(u_{k-1})) = J(f_1 \circ \dots \circ f_{k-1})$, 这意味着 $l_{0,i}^C \in loc(post(L_C \times C_C, u_{k-1}))$ 和 $l_i^B \in post(L_B \times C_B, g(u_{k-1}))$, 即 $\mathcal{C}$ 的第 0 行第 i 列上的位置 $l_{0,i}^C$ 对应 $\mathcal{B}$ 的位置 l_i^B. 如果 $post(l_{0,i}^C \times C_C, (t_k, a^k b))$ 有定义,那么对于某个 $m \in I(u_k)$, $loc(post(l_{0,i}^C \times C_C, a^k b)) = l_{0,m}^C$, 即 $l_{0,i}^C$ 经过$(t_k, a^k b)$后可以到达 $l_{0,m}^C$ 位置.因此,从 $\mathcal{C}$ 的变迁函数定义可以得到:对于 $\mathcal{B}$, $post(l_i^B, f(t, a^{h_k} b))$ 有定义,并且 $loc(post(l_i^B, f(t, a^{h_k} b))) = l_m^B$. 因此, $m \in J(g(u_k))$ 和 $I(u_k) \subseteq J(g(u_k))$. 对包含关系 $I(u_k) \supseteq J(g(u_k))$ 的证明类似.于是 $I(u_k) = J(g(u_k))$. 这就意味着 $I(u) = J(g(u))$.

引理的第 2 句可类似地归纳证明. □

推论 1 的证明:由定理 1 的证明中提到的:时钟个数不同时,时间自动机的可达性问题的复杂性结论和定理 2 的证明中提到的:一般情况下部分规约的时间自动机的重置问题的复杂性结论可证. □

推论 2 的证明:采用定理 3 的证明中使用的多字母表到 2-字母表的映射方法可证. □

引理 5 的证明:可由 D_i-重置序列(i=1,2,3)的定义直接得到. □

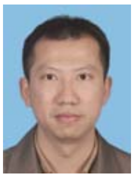
命题 2 的证明:首先证明,对于问题 A 和问题 B,如果 $A \leqslant_E B$ 且 $A \in$NEXPTIME,那么 $B \in$NEXPTIME.设 M 是判定 B 的算法,f 是从 A 到 B 的指数时间归约.判定 A 的算法 N 可以描述为

N="对于输入 w:

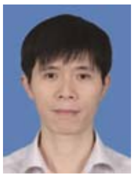
(1) 计算 $f(w)$;

(2) 以 $f(w)$为输入,运行 M,输出 M 的输出."算法 N 判定问题 A 当且仅当算法 M 判定问题 B.如果 w 是 A 的输入,那么 $f(w)$是 B 的输入.因为 f 是从 A 到 B 的指数时间归约,于是只要 M 接受 $f(w)$,N 就接受 w.此外,步骤(1)是在 $O(2^{n^{k_1}})$ 时间内运行的,其中, $k_1 \in \mathbb{N}$. 算法 N 的时间消耗是步骤(1)和步骤(2)消耗的总时间,而已知 N 是非确定的指数时间.由于 NEXPTIME $= \bigcup_{k>0}$ NTIME(2^{n^k}) 和 EXPTIME $\subseteq$ NEXPTIME[44], 所以步骤(2)(即算法 M)是非确定的指数时间算法.所以 $B \in$NEXPTIME.

然后证明:对于问题 A 和问题 B,如果 $A\leqslant_E B$ 且 $A\in$ACK,则 $B\in$ACK.由于 ACK 是超过 ELEMENTARY 的快速增长的复杂性类[45],显然,它相对指数时间归约是封闭的. □



朱凯(1979－),男,湖北十堰人,讲师,主要研究领域为理论计算机科学,形式化方法.



吴理华(1976－),男,博士,讲师,主要研究领域为自动机理论,离散事件系统.

毋国庆(1954－),男,教授,博士生导师,CCF 高级会员,主要研究领域为软件理论,软件工程,需求工程.

袁梦霆(1976－),男,博士,副教授,CCF 专业会员,主要研究领域为程序语言理论,编译优化,软件工程.