软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(12):3605−3621 [doi: 10.13328/j.cnki.jos.005611]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

格值交替树自动机*

魏秀娟[1], 李永明[1,2]

[1](陕西师范大学 数学与信息科学学院,陕西 西安 710119)
[2](陕西师范大学 计算机科学学院,陕西 西安 710119)
通讯作者: 李永明, E-mail: liyongm@snnu.edu.cn

摘 要: 交替(树)自动机因其本身关于取补运算的简洁性及其与非确定型(树)自动机的等价性,成为自动机与模型检测领域研究的一个新方向.在格值交替自动机与经典交替树自动机概念的基础上,引入格值交替树自动机的概念,并研究了格值交替树自动机的代数封闭性和表达能力.首先,证明了对格值交替树自动机的转移函数取对偶运算,终止权重取补之后所得自动机与原自动机接受语言互补这一结论.其次,证明了格值交替树自动机关于交、并运算的封闭性.最后,讨论了格值交替树自动机和格值树自动机、格值非确定型自动机的表达能力;证明了格值交替树自动机与格值树自动机的等价性,并给出了二者相互转化的算法及其复杂度分析;同时,提供了用格值非确定型自动机来模拟格值交替树自动机的方法.

关键词: 格值交替树自动机;格值正布尔公式;对偶运算;格值计算树;接受运行
中图法分类号: TP301

中文引用格式: 魏秀娟,李永明.格值交替树自动机.软件学报,2019,30(12):3605−3621. http://www.jos.org.cn/1000-9825/5611.htm
英文引用格式: Wei XJ, Li YM. *L*-valued alternating tree automata. Ruan Jian Xue Bao/Journal of Software, 2019,30(12): 3605−3621 (in Chinese). http://www.jos.org.cn/1000-9825/5611.htm

L-valued Alternating Tree Automata

WEI Xiu-Juan[1], LI Yong-Ming[1,2]

[1](College of Mathematics and Information Science, Shaanxi Normal University, Xi'an 710119, China)
[2](College of Computer Science, Shaanxi Normal University, Xi'an 710119, China)

Abstract: Because of the simplicity of taking complement operation on alternating (tree) automata and the equivalence relationship between alternating (tree) automata and nondeterministic (tree) automata, the study on alternating (tree) automata becomes a new research area of automata and model checking. Based on notions of *L*-valued alternating automata and alternating tree automata, the notion of *L*-valued alternating tree automata is introduce, and closure properties and expressive power of *L*-valued alternating tree automata are studied. Firstly, it is proved that after taking dual operations on transitions and changing the weight of each final state to its complement, a new *L*-valued alternating tree automaton is achieved which is the complement of the starting one. Afterwards, the closure is illustrated under conjunction and disjunction of languages accepted by *L*-valued alternating tree automata. Finally, the expressive power of *L*-valued alternating tree automata, *L*-valued tree automata, and *L*-valued nondeterministic automata are discussed. The equivalence relationship is proved between *L*-valued alternating tree automata and *L*-valued tree automata, the algorithms are given between them and complexities are discussed of algorithms; simultaneously, a method is provided to show how to use *L*-valued nondeterministic automata to simulate *L*-valued alternating tree automata.

Key words: *L*-valued alternating tree automata; *L*-valued positive Boolean formula; dual operation; *L*-valued computation tree; accepting run

* 基金项目: 国家自然科学基金(11671244, 11271237); 高等学校博士学科点专项科研基金(20130202110001)
Foundation item: National Natural Science Foundation of China (11671244, 11271237); Research Fund for the Doctoral Program of Higher Education of China (20130202110001)
收稿时间: 2016-09-18; 修改时间: 2018-03-20; 采用时间: 2018-05-29

非确定性在计算理论中有着重要意义[1−5].从逻辑层面看,非确定计算只涉及存在量词,而作为非确定的推广,“交替”在存在量词的基础上又增加了全称量词.Chandra 将交替的概念与自动机相结合,提出了交替自动机的概念[6],随后,这一类型的自动机在形式化证明中被作为一种有用的模型普遍使用[7−14].

Zhou[15]在原有交替ω-有穷自动机接受条件的基础上定义了 6 种新形式的接受条件,并研究了交替ω-有穷自动机在这些条件下接受语言的能力.Vardi 在研究线性时序逻辑[14]时,给出了用自动机理论方法来研究模型检测的新思路,即,把模型检测的可满足性问题转化为判断自动机语言是否为空的问题来讨论.Vardi 运用 Muller 等人给出的交替(Büchi)自动机与非确定型(Büchi)自动机的等价性,为任意给定的线性时序逻辑公式构造相应的交替 Büchi 自动机,使得该自动机接受的语言恰好等于满足原线性时序逻辑公式的计算的全体.

Kupferman 等人[16,17]定量地对交替自动机展开研究,提出了实数集上加权交替自动机的概念,并讨论了特殊语义(如 Max,Sum,Sup,Lim sup 语义)下加权交替(Büchi)自动机的表达能力,同时研究了特殊语义下加权交替(Büchi)自动机的代数封闭性.鉴于权值取值和语义选取的局限性,实数集上加权交替自动机的讨论较为特殊.另外,终止状态的影响并未考虑在内,这也体现了 Kupferman 等人[16,17]讨论的局限性.基于这些问题,Wei 等人提出了格值交替自动机的概念[18],概念的创新性体现于权值的设置.文献[18]将转移的权值作为运行树叶子节点的标记,使得树语言计算简洁化,同时保证了对转移取对偶运算、终止权重取补后所得的自动机与原自动机接受语言互补这一性质;此外,Wei 等人比较了格值交替自动机与格值非确定型自动机的表达能力.

树作为重要的非线性数据结构,在计算机科学中应用非常广泛.在编译源程序时,树可被用来表示源程序的语法结构;在数据库系统中,树可作为信息的重要组织形式.因此,关于输入为树结构的自动机研究也是非常有意义的.但到目前为止,针对交替树自动机的研究仅有较少学者涉及[11,19].Muller 等人在文献[10,11]中针对经典情形下的交替(Büchi)(树)自动机展开讨论时指出:对交替(Büchi)(树)自动机的转移函数取对偶运算,并将终止状态和非终止状态互换所得的自动机与原自动机接受的语言互补;同时讨论了交替(Büchi)(树)自动机与非确定型(Büchi)(树)自动机的表达能力.

量化情形下输入为树(k 元Σ-树)结构的交替自动机暂未有人展开讨论,针对此类计算模型是否有如上结论,本文即从此初衷展开研究且仅考虑有限输入树的情形.这里将 Muller 等人提出的交替树自动机的概念[10,11]和 Wei 等人在格值交替自动机[18]上的权值设置方式相结合,引入了格值交替树自动机的概念,讨论了其代数封闭性和表达能力.特别地,将状态转移函数取对偶运算,终止权重取补之后即可得到与原自动机接受语言互补的格值交替树自动机.此外,讨论了格值交替树自动机与格值树自动机之间的表达能力,说明了二者的等价性.最后指出可用格值非确定型自动机来模拟格值交替树自动机.

1 预备知识

首先介绍本文的预备知识,更多详细内容参见文献[18−21].

1.1 树的相关概念

定义 1[20]. 元素组(Σ,rk)称为一个分级的字母表,其中,Σ为有限集合,rk 为Σ到自然数集合 N 上的一个映射(在这里称为级映射).

在不引起混淆的情况下,通常用Σ表示(Σ,rk),简称字母表.对任意的 $k\geqslant 0,\Sigma^{(k)}=\{\sigma\in\Sigma|rk(\sigma)=k\}$.

定义 2[20]. 设 $H\cap\Sigma=\varnothing$.H 上Σ-项(Σ-树或简称树)构成的集合 $T_\Sigma(H)$为满足下面两个条件的最小的集合 T:

(1) $\Sigma^{(0)}\cup H\subseteq T$;

(2) 如果 $k\geqslant 1,\sigma\in\Sigma^{(k)}$,且$\xi_1,\ldots,\xi_k\in T$,则$\sigma(\xi_1,\ldots,\xi_k)\in T$.

当 $H=\varnothing$,$T_\Sigma(\varnothing)$简记为 T_Σ.

定义 3[20]. 树的高度 $Height$ 和位置集 Pos 是从集合 T_Σ分别到自然数集 N 和由自然数集生成的幺半群的幂集$\mathcal{P}(N^*)$上的映射,递归定义如下.

(1) 对任意的$\alpha\in\Sigma^{(0)}$,$Pos(\alpha)=\{\varepsilon\}$,$Height(\alpha)=0$;

(2) 对任意的$\xi=\sigma(\xi_1,\dots,\xi_k)\in T_\Sigma,k\geqslant 1$:

$$Pos(\xi)=\{\varepsilon\}\cup\{iv|1\leqslant i\leqslant k,v\in Pos(\xi_i)\};Height(\xi)=1+\max\{Height(\xi_i)|1\leqslant i\leqslant k\}.$$

定义 4[20]. 设$\xi\in T_\Sigma,w\in Pos(\xi),\xi(w)$表示$\xi$在 w 处的标记,递归定义如下.

(1) 对任意的$\alpha\in\Sigma^{(0)},\alpha(\varepsilon)=\alpha$;

(2) 对任意的$\xi=\sigma(\xi_1,\dots,\xi_k)\in T_\Sigma,k\geqslant 1,\xi(\varepsilon)=\sigma$;且对于任意的 $1\leqslant i\leqslant k$ 和 $v\in Pos(\xi_i),\xi(iv)=\xi_i(v)$.

规定位置 w 所在的层数称为树的第$|w|$层(这里的$|w|$表示串 w 的长度).

1.2 格的相关概念

定义 5[21]. 集合 X 上的二元关系$\leqslant$如果满足以下条件,则称$\leqslant$为集合 X 上一个序关系,同时称 X 为一个偏序集.

(1) 对任意的 $x\in X$,有 $x\leqslant x$ 成立;

(2) 对任意的 $x,y\in X$,若 $x\leqslant y$ 和 $y\leqslant x$ 成立,则有 $x=y$ 成立;

(3) 对任意的 $x,y,z\in X$,若 $x\leqslant y$ 和 $y\leqslant z$ 成立,则有 $x\leqslant z$ 成立.

基于集合 X 上的序关系$\leqslant$,可以定义 X 的子集 X'的上(下)界:设 $x\in X$,如果对任意的 $y\in X'$都有 $y\leqslant x(x\leqslant y)$,则称 x 为 X'的一个上界(下界).

X'的上确界(下确界)指的是 X'的最小(大)上界(下界),用 $sup(X')(inf(X'))$表示.特别地,X 作为自身的一个子集,若 X 存在上确界(下确界),则用 1(0)表示 $sup(X)(inf(X))$,并称其为 X 的最大(小)元.

对任意的 $x,y\in X$,可以用$\vee,\wedge$运算表示上下确界:$x\vee y$ 表示 $sup\{x,y\}$;$x\wedge y$ 表示 $inf\{x,y\}$.

定义 6[21]. 设 L 为一个非空的偏序集,若对任意的 $x,y\in L,x\vee y$ 和 $x\wedge y$ 同时存在,则称 L 为一个格.若格 L 有上下确界(同样用 1,0 表示),则称 L 为有界格.若格 L 中的任意元素 x,y,z 满足 $x\wedge(y\vee z)=(x\wedge z)\vee(y\wedge z)$,则称 L 为分配格.

例如,$L=\{0,1,b_1,b_2,b_3\}$(如图 1 所示),满足:$b_1\vee b_2=b_1;b_1\vee b_3=1;b_2\vee b_3=b_3;b_1\wedge b_2=b_2;b_1\wedge b_3=b_2;b_2\wedge b_3=b_2$.对任意的$x\in L$,有 $x\vee 1=1,x\wedge 1=x,x\vee 0=x,x\wedge 0=0$ 成立.

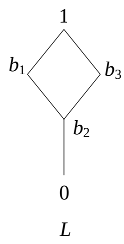

Fig.1 Lattice L
图 1 格 L

容易验证,L 为一个有界分配格.通常情况下,用符号 $1\oplus 2^2$ 来表示上述格结构.

1.3 格值树自动机

定义 7[20]. L 上的格值树自动机是一个四元组(Q,Σ,μ,ν),其中,Q 是有限状态集;Σ是一个分级的字母表;$\mu=(\mu_k|k\in N)$是一簇状态转移函数, $\mu_k:\Sigma^{(k)}\to L^{Q^k\times Q}$;$\nu$是格值根权向量,$\nu\in L^Q$,即 $\nu:Q\to L$.

格值树自动机接受的树语言(树级数)是从 T_Σ到格 L 上的映射 r,对任意的$\xi\in T_\Sigma$:

$$r(\xi)=h_\mu(\xi)\wedge\nu=\bigvee_{q\in Q}(h_\mu(\xi)_q\wedge\nu_q).$$

这里的 h_μ为从 T_Σ到 L^Q 上唯一的Σ-代数,即 h_μ为从 T_Σ到 L^Q 上的映射且满足:对任意的$\sigma(\xi_1,\dots,\xi_k)\in T_\Sigma$和 $q\in Q$,

$$h_\mu(\sigma(\xi_1\dots\xi_k))_q=\bigvee_{q_1,\dots,q_k\in Q}(h_\mu(\xi_1)_{q_1}\wedge\dots\wedge h_\mu(\xi_k)_{q_k}\wedge\mu_k(\sigma)_{q_1\dots q_k,q})..$$

例如,$A=(Q,\Sigma,\mu,\nu)$为 $L=1\oplus 2^2$ 上的一格值树自动机,其中,$Q=\{q_1,q_2\}$;$\Sigma=\{\sigma,\alpha\},\sigma\in\Sigma^{(2)},\alpha\in\Sigma^{(0)}$;非零的状态转移函数如下:

$$\mu_0(\alpha)_{\varepsilon,q_1}=\mu_0(\alpha)_{\varepsilon,q_2}=1,\mu_2(\sigma)_{q_1q_2,q_1}=\mu_2(\sigma)_{q_2q_1,q_1}=b_2,\mu_2(\sigma)_{q_2q_2,q_2}=1;\nu_{q_1}=b_3,\ \nu_{q_2}=b_1,$$

则 A 所对应的树语言 r 接受树$\sigma(\alpha,\alpha)$的程度如下:

$$\begin{aligned}r(\sigma(\alpha,\alpha))&=(h_\mu(\sigma(\alpha,\alpha))_{q_1}\wedge\nu_{q_1})\vee(h_\mu(\sigma(\alpha,\alpha))_{q_2}\wedge\nu_{q_2})\\&=(h_\mu(\alpha)_{q_1}\wedge h_\mu(\alpha)_{q_2}\wedge\mu_2(\sigma)_{q_1q_2,q_1}\wedge\nu_{q_1})\vee(h_\mu(\alpha)_{q_2}\wedge h_\mu(\alpha)_{q_1}\wedge\mu_2(\sigma)_{q_2q_1,q_1}\wedge\nu_{q_1})\vee\\&\quad(h_\mu(\alpha)_{q_2}\wedge h_\mu(\alpha)_{q_2}\wedge\mu_2(\sigma)_{q_2q_2,q_2}\wedge\nu_{q_2})\\&=(\mu_0(\alpha)_{\varepsilon,q_1}\wedge\mu_0(\alpha)_{\varepsilon,q_2}\wedge\mu_2(\sigma)_{q_1q_2,q_1}\wedge\nu_{q_1})\vee(\mu_0(\alpha)_{\varepsilon,q_2}\wedge\mu_0(\alpha)_{\varepsilon,q_1}\wedge\mu_2(\sigma)_{q_2q_1,q_1}\wedge\nu_{q_1})\vee\\&\quad(\mu_0(\alpha)_{\varepsilon,q_2}\wedge\mu_0(\alpha)_{\varepsilon,q_2}\wedge\mu_2(\sigma)_{q_2q_2,q_2}\wedge\nu_{q_2})\\&=(1\wedge1\wedge b_2\wedge b_3)\vee(1\wedge1\wedge b_2\wedge b_3)\vee(1\wedge1\wedge1\wedge b_1)\\&=b_1.\end{aligned}$$

1.4 交替树自动机的相关概念

对于给定的集合 X,设 $B^+(X)$为 X 上所有正布尔公式(即 X 中的元素通过$\vee,\wedge$运算生成的布尔公式)构成的集合,此外,要求公式 true,false 也在 $B^+(X)$中.

令 $Y\subseteq X,\theta\in B^+(X)$.将公式$\theta$中属于 Y 的元素赋值为真,将 X/Y 中元素赋值为假,如果将所有元素赋值之后,最终结果为真,则称 Y 满足公式θ.进一步地,若 Y 的任意子集都不满足公式θ,则称 Y 以极小方式满足θ.

定义 8[19]. Σ上的交替树自动机是一个四元组(Q,Σ,I,Δ),其中,Q,Σ分别为有限状态集和输入字母表;I 为初始状态集;$\Delta:Q\times\Sigma\to B^+(Q\times N)$为状态转移函数,且对任意的 $q\in Q,\sigma\in\Sigma$,有$\Delta(q,\sigma)\in B^+(Q\times\{1,\ldots,rk(\sigma)\})$成立.$B^+(Q\times N)$中的 N 为自然数集.

定义 9[19]. 设$\xi\in T_\Sigma$,A 为Σ上的一个交替树自动机.A 在ξ上的一个运行定义为一棵节点标记属于 $Q\times N^*$的树 t 且满足:

设 $t(w)=(q,w')$,$\xi(w')=\sigma$并且$\delta(q,\sigma)=\theta$,若存在 $Q\times\{1,\ldots,rk(\sigma)\}$的子集$\{(q_1,i_1),\ldots,(q_n,i_n)\}$满足$\theta$,则 w 在 t 中有后继位置 $w_1,\ldots,w_n$,且这些位置的标记依次为$(q_1,w'i_1),\ldots,(q_n,w'i_n)$.

如果运行树根节点的标记为(q,ε)且 $q\in I$,则称该运行是可被 A 接受的.

1.5 格值正布尔公式

对于给定的集合 X,$B^+(L\cup X)$表示 X 上所有格值正布尔公式(即 $L\cup X$ 上的正布尔公式)构成的集合.此外,要求公式 true,false 在 $B^+(L\cup X)$中.

任给集合 $Y\subseteq X,\theta\in B^+(L\cup X)$.定义格值元素 $v(\theta,Y)$:将公式θ中属于 Y 的元素赋值为 1,将 X/Y 中的元素赋值为 0,令 $v(\theta,Y)$为θ赋值后的结果.设$\theta_1,\theta_2\in B^+(L\cup X)$,若对任意的 $Y\subseteq X$,都有 $v(\theta_1,Y)=v(\theta_2,Y)$成立,则称$\theta_1$ 与θ_2 等价,记为$\theta_1\equiv\theta_2$.例如,$\theta_1=(0.5\vee(0.3\wedge(0.8\vee x_1)))\wedge0.2$,$\theta_2=0.2\wedge x_1$,按定义易证$\theta_1\equiv\theta_2$.

由文献[18]可知:对于任意的格值正布尔公式θ而言,均可以找到与其等价的最简终展开式,用θ^S 表示.这里的最简终展开式意味着公式的析取范式中,析取连接词连接的任意两项不能存在包含关系,即:任意的两项 $l\wedge(\underset{i\in I}{\wedge}x_i))$ 和 $l'\wedge(\underset{j\in J}{\wedge}x_j))$,不能有 $l\leqslant l'$和 $J\leqslant I$ 同时成立.若 $l''\wedge(\underset{s\in S}{\wedge}x_s)$ 为θ的最简终展开式中的一项,则称集合$\{x_s|s\in S\}$以极小方式满足θ的权重为 l''.对于上述θ_1,θ_2,可知二者有相同的最简终展开式 $0.2\wedge x_1$.事实上,$\theta_1\equiv\theta_2$ 当且仅当θ_1 和θ_2 有相同的最简终展开式.特别地,true≡1,false≡0.

2 格值交替树自动机

本节引入格值交替树自动机的概念,讨论其闭包性质以及格值交替树自动机与格值树自动机、格值非确定自动机之间的表达能力.

2.1 格值交替树自动机的概念

下文中,如果没有特别说明,L 均为分配格,且包含最大元 1 和最小元 0.

定义 10. k 元Σ-树上的格值交替树自动机(简称格值交替树自动机)是一个五元组 $A=(Q,\Sigma,I,\delta,K)$,其中,Q,Σ分

别为有限状态集和输入字母表,I 为格值初始状态集,δ是从 $Q\times\Sigma$到 $B^{+}(L\cup(Q\times K))$上的映射.$Q\times K$ 是所有形如(q,i)的元素构成的集合,$q\in Q,i\in K,K=\{1,\dots,k\}$是方向集.

定义 11. 格值交替树自动机 A 在给定输入树$\xi(\in T_{\Sigma})$上的运行为一棵节点标记属于$L\cup(Q\times K^{*})$的树 t,且满足:

(1) 如果 $t(\varepsilon)=(q,\varepsilon)$,则 $I(q)\neq 0$.

(2) 设 $t(w)=(q,w')$,$\xi(w')=\sigma(\sigma\in\Sigma^{(k)})$且$\delta(q,\sigma)=\theta(\theta\neq\text{true})$.若存在 $Q\times\{1,\dots,rk(\sigma)\}$的某个子集$\{(q_1,i_1),\dots,(q_{s-1},i_{s-1})\}$以极小方式满足$\theta$的权重为 $l(l\neq 1)$,则 w 在 t 中的后继位置有 $w_1,\dots,w_s$,且这些位置的标记分别为$l,(q_1,w'i_1),\dots,(q_{s-1},w'i_{s-1})$.若存在集合$\{(q_1,i_1),\dots,(q_{s-1},i_{s-1})\}$以极小方式满足$\theta$的权重为 1,则 w 在 t 中有后继位置 $w_1,\dots,w_{(s-1)}$,且这些位置的标记分别为$(q_1,w'i_1),\dots,(q_{s-1},w'i_{s-1})$(后一种情形下相当于 $l=1$,从下文格值交替树自动机接受语言的定义可以看出,此处的 1 并不影响接受语言的程度,为了计算方便略去不标).

(3) 如果 $t(w)=(q,w')$,$\xi(w')=\sigma(\sigma\in\Sigma^{(k)})$且$\delta(q,\sigma)=\text{true}$,则 w 在 t 中只有 1 个后继位置 w_1,节点标记为 1(此处的 1 不省略是为了使所有叶子节点都有格值标记).

(4) 如果 $t(w)=l$,则 w 没有后继位置.

设 A 为格值交替树自动机,A 所接受的语言 r_A 为从 T_{Σ}到格 L 上的映射,定义如下:对任意的$\xi\in T_{\Sigma}$,

$$r_A(\xi)=\bigvee_{\substack{t\in R_A(\xi);\\ t(\varepsilon)=(q,\varepsilon)}}(I(q)\wedge wt(t)).$$

其中,$R_A(\xi)$表示 A 在给定输入树ξ上的所有运行树构成的集合,(q,ε)为运行树 t 的根节点的标记.$wt(t)$表示运行树 t 的权重,该值等于 t 的所有叶子节点标记的合取值.根据运行树的定义可知,t 的所有叶子节点的标记均为格值元素,即 $wt(t)$等于 t 中出现的所有格值元素的合取值.

若运行 t 使得 $t(\varepsilon)=(q,\varepsilon)$且 $I(q)\wedge wt(t)\neq 0$ 成立,则称 t 为可被 A 接受的运行.

例 1:令 $L=1\oplus 2^2$(如图 1 所示).设 $A=(Q,\Sigma,I,\delta,K)$,其中,$Q=\{q_0,q_1,q_2\}$,$\Sigma=\{\sigma,\alpha\}$,$K=\{1,2\}$.其中,$\sigma\in\Sigma^{(2)}$,$\alpha\in\Sigma^{(0)}$,初始权重为:$I(q_0)=b_1$,$I(q_1)=I(q_2)=0$.状态转移函数为:$\delta(q_0,\sigma)=(b_1\wedge(q_1,1)\wedge(q_1,2))\vee(b_2\wedge(q_2,1))\vee b_3$;$\delta(q_0,\alpha)=\text{true}$;$\delta(q_1,\sigma)=b_2\wedge(q_2,1)$;$\delta(q_1,\alpha)=\text{true}$;$\delta(q_2,\sigma)=b_1\wedge(q_1,2)$;$\delta(q_2,\alpha)=\text{false}$.

给定输入树$\xi=\sigma(\sigma(\alpha,\alpha),\alpha)$,由定义 11 可知$\xi$上的接受运行树有两棵,记为 t_1,t_2,如图 2 所示.

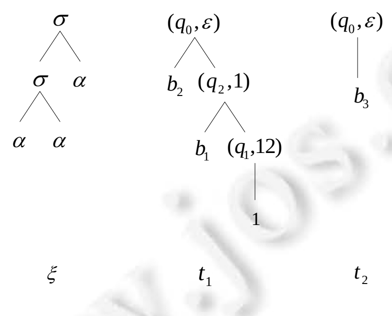

Fig.2 All accepting runs of A on ξ

图 2 A 在ξ上的接受运行树

格值交替树自动机 A 接受ξ的程度为

$$r_A(\xi)=\bigvee_{\substack{t\in R_A(\xi);\\ t(\varepsilon)=(q,\varepsilon)}}(I(q)\wedge wt(t))=(I(q_0)\wedge wt(t_1))\vee(I(q_0)\wedge wt(t_2))=(b_1\wedge b_2\wedge b_1\wedge 1)\vee(b_1\wedge b_3)=b_2.$$

为了下文叙述方便,将 t 中非叶子节点的标记(状态和位置构成的二元对)中的第 1 分量称为状态指标,第 2 分量称为位置指标.

2.2 格值交替树自动机的闭包性质

从代数角度研究各种不同类型的自动机[22−25],讨论其是否构成一个封闭的代数系统是对自动机本身性质

的探讨,也是保证其运算合理性的前提.这一节针对本文提出的格值交替树自动机的封闭性问题进行研究.

对于经典的交替树自动机而言,文献[10]借助计算树的概念和 Game-理论证明了对交替树自动机的转移函数取对偶运算,将终态和非终态互换即可得到与原交替自动机互补的另一交替自动机.本节旨在文献[10]的基础上研究格值交替树自动机的补运算,并证明其封闭性.由于要考虑补运算,所以这一节中用到的格含有补运算记作 c,即 c 是格 L 上的一元运算且满足:对任意的 $l,l_1,l_2\in L$,有 $l=c(c(l))$和 $l_1\leqslant l_2\Leftrightarrow c(l_2)\leqslant c(l_1)$成立[21].

引理 1. 对于任意的格值交替树自动机,均存在与其等价的且具有唯一初始状态的格值交替树自动机.

证明:设 A 为格值交替树自动机(Q,Σ,I,δ,K)构造 $A'=(Q\cup\{q^0\},\Sigma,q_0,\delta',K)$:$A'$在 A 的状态集基础上添加一个额外状态 q_0,作为 A'的唯一初始状态;对任意的$\sigma\in\Sigma$,添加转移函数 $\delta'(q^0,\sigma)=\bigvee_{I(q)\neq 0}(I(q)\wedge\delta(q,\sigma))$;对任意的 $q\in Q$, $\sigma\in\Sigma$,令$\delta'(q,\sigma)=\delta(q,\sigma)$.

由 A'的构造可知,初始自动机 A 与目标自动机 A'在任意给定的输入树上的运行树一一对应:事实上,A 在任意的输入树ξ上的运行树 t,其中,$t(\varepsilon)=q'$且 $I(q')\neq 0$,对应于 A'在ξ上的运行树 t',其中,$t'(\varepsilon)=q^0$ 且 $wt(t')=I(q^0)\wedge wt(t)$,因此 A 与 A'等价:

$$r_A(\xi)=\bigvee_{\substack{t\in R_A(\xi);\\ t(\varepsilon)=q}}(I(q)\wedge wt(t))=\bigvee_{\substack{t'\in R_{A'}(\xi);\\ t'(\varepsilon)=q^0}}wt(t')=r_{A'}(\xi).$$ □

有引理 1 的保证,在接下来的讨论中如果没有特殊说明,则仅考虑具有唯一初始状态的格值交替树自动机.

定理 1. k 元Σ-树上的格值交替树自动机关于交、并运算封闭.

证明:设 $A_i=(Q_i,\Sigma,q_i^0,\delta_i,K)$ $(i=1,2)$为两个状态不交($Q_1\cap Q_2=\varnothing$)的格值交替树自动机,其中,q_1^0,q_2^0 分别为二者的唯一的初始状态,δ_1,δ_2 分别是 A_1,A_2 的转移函数.

定义 k 元Σ-树上的格值交替树自动机:

$$A_\wedge=(Q_1\cup Q_2\cup\{q_\wedge^0\},\Sigma,q_\wedge^0,\delta_\wedge,K),A_\vee=(Q_1\cup Q_2\cup\{q_\vee^0\},\Sigma,q_\vee^0,\delta_\vee,K).$$

其中,

- 对任意的$\sigma\in\Sigma$, $\delta_\wedge(q_\wedge^0,\sigma)=\delta_1(q_1^0,\sigma)\wedge\delta_2(q_2^0,\sigma);\delta_\vee(q_\vee^0,\sigma)=\delta_1(q_1^0,\sigma)\vee\delta_2(q_2^0,\sigma)$.
- 对任意的 $q\in Q_1\cup Q_2$, $\delta_{\wedge(\vee)}(q,\sigma)=\begin{cases}\delta_1(q,\sigma), & 若 q\in Q_1\\ \delta_2(q,\sigma), & 若 q\in Q_2\end{cases}$.

易证 $r_{A_\wedge}(\xi)=r_1(\xi)\wedge r_2(\xi),r_{A_\vee}(\xi)=r_1(\xi)\vee r_2(\xi)$. □

定理 2. k 元Σ-树上的格值交替树自动机关于补运算封闭.

为了证明定理 2,需要借鉴参考文献[10]的记号和定义而引入一些新的概念.在介绍新概念之前,首先介绍 $Q\times K$ 上的格值正布尔公式($L\cup(Q\times K)$上的正布尔公式)的对偶运算.

对于 $Q\times K$ 上的格值正布尔公式,定义对偶运算:

(1) 对任意的 $(q,i)\in Q\times K,\overline{(q,i)}=(q,i)$;

(2) 对任意的 $l\in L,\overline{l}=c(l)$ (特别地, $\overline{0}=1,\overline{1}=0$);

(3) $\overline{\theta_1\vee\theta_2}=\overline{\theta_1}\wedge\overline{\theta_2};\overline{\theta_1\wedge\theta_2}=\overline{\theta_1}\vee\overline{\theta_2}$;

(4) $\overline{\text{true}}=\text{false};\overline{\text{false}}=\text{true}$.

对格值交替树自动机 A 的所有转移函数取对偶运算、终止权重用其补元代替,即可得到一个新的格值交替树自动机,用 $\overline{A}$ 表示.定理 2 可等价表述为 $\overline{A}$ 是一个与 A 互补的 k 元Σ-树上的格值交替树自动机.

定义 12. 格值的 n 步 ID($n\geqslant 0$),记作 FH_n,为所有形如 h_1,h_2 的元素构成的集合,其中,

- $h_1=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\dots k_n(q_n,k_1\dots k_n)$;
- $h_2=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\dots k_t(q_t,k_1\dots k_t)k'(l),t\leqslant n-1,k_i\in K(i=1,\dots,n),k'=0$.

这里的 0 表示自然数 0,并非格中的零元),$l\in L$.

注意到,$FH_n\cap FH_{n+1}\neq\varnothing$.

定义 13. 对于 FH_n 中的任意元 h 以及 $L\cup(Q\times K)$中的任意元 g,定义 h,g 的连接.

(1) 若 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s),g=(q,k)$,则

$$hg=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s)k(q,k_1\ldots k_sk);$$

(2) 若 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s),g=l$,则 $hg=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s)0(l)$;

(3) 若 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s)0(l)$,对任意的 g,hg 无意义.

FH_n 中的任意元 h 以及$\mathcal{L}^*(L\cup(Q\times K))$中的任意元 g 的连接运算定义如下.

- 若 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s),\ g=\vee_{j\in J}(l_j\wedge(\wedge_{i\in I_j}(q_i',k_i')))$,则

$$hg=\vee_{j\in J}\left((q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s)0(l_j)\wedge\left(\wedge_{i\in I_j}(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s)k_i'(q_1',\ k_1\ldots k_sk_i')\right)\right);$$

- 若 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s)0(l)$,则对任意的 g,hg 无意义.

注意,这里的$\mathcal{L}^*(L\cup(Q\times K))$表示由 $L\cup(Q\times K)$中元素生成的自由分配格.

对于转移函数δ,每固定一个输入字符σ,可以得到从 Q 到 $B^+(L\cup(Q\times K))$上的映射δ_σ.进一步,δ_σ可以扩展成从$\mathcal{L}^*(Q)$到 $B^+(L\cup(Q\times K))$上的同态映射,仍用记号δ_σ表示.同样地,δ_σ的对偶同态 $\overline{\delta_\sigma}$ 可相应定义如下:若$\delta_\sigma(q)=\theta$,则 $\overline{\delta_\sigma}(q)=\overline{\theta}.\overline{\theta}$ 是对θ取对偶运算后得到的格值正布尔公式.

将 $\overline{\delta_\sigma}$ 扩展成从$\mathcal{L}^*(Q)$到 $B^+(L\cup(Q\times K))$上的同态映射,仍记为 $\overline{\delta_\sigma}$.下面说明:对任意的 $e\in\mathcal{L}^*(Q)$,都有性质 $\overline{\delta_\sigma(e)}=\overline{\delta_\sigma}(\overline{e})$ 成立.事实上,假设 $e=\vee_{i\in I}(\wedge_{j\in J_i}s_{i_j})$,其中,$s_{i_j}\in Q$:

$$\overline{\delta_\sigma(e)}=\overline{\delta_\sigma(\vee_{i\in I}(\wedge_{j\in J_i}s_{i_j}))}=\overline{\vee_{i\in I}(\wedge_{j\in J_i}\delta_\sigma(s_{i_j}))}=\wedge_{i\in I}(\vee_{j\in J_i}\overline{\delta_\sigma(s_{i_j})})=\wedge_{i\in I}(\vee_{j\in J_i}\overline{\theta}_{i_j})=\wedge_{i\in I}(\vee_{j\in J_i}\overline{\delta_\sigma}(s_{i_j}))=\overline{\delta_\sigma}(\wedge_{i\in I}(\vee_{j\in J_i}s_{i_j}))=\overline{\delta_\sigma}(\overline{e}).$$

对于给定的运行树 t,根据同态映射簇$\{\delta_\sigma|\sigma\in\Sigma\}$可以定义从 $FH_n|_t$ 到$\mathcal{L}^*(FH_{n+1})$上的映射δ_n,其中,$FH_n|_t$ 表示限制在 t 上的格值 n 步 ID.这些 n 步 ID 对应于 t 的某些分支的(长度≤n 的)前缀.同上,这里的记号$\mathcal{L}^*(FH_{n+1})$表示由 FH_{n+1} 中元素生成的自由分配格.

如例 1 中的 t_1:

- $n=1$ 时, $FH_n|_{t_1}=\{(q_0,\varepsilon)1(q_2,1),(q_0,\varepsilon)0(b_2),(q_0,\varepsilon)0(b_3)\}$;
- $n=2$ 时, $FH_n|_{t_1}=\{(q_0,\varepsilon)1(q_2,1)2(q_1,12),(q_0,\varepsilon)0(b_2),(q_0,\varepsilon)0(b_2),(q_0,\varepsilon)1(q_2,1)0(b_1)\}$.

若 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_n(q_n,k_1\ldots k_n)$且 $t(k_1\ldots k_s)=\sigma$,定义$\delta_n(h)=h\delta_\sigma(q_n)$;

若存在 $s\leqslant n-1$,使得 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s)0(l)$,定义$\delta_n(h)=h$.

类似地,可以将其扩展成从$\mathcal{L}^*(FH_n|_t)$到$\mathcal{L}^*(FH_{n+1})$上的同态映射,仍用记号δ_n表示,其对偶同态用$\overline{\delta_n}$ 表示.

若 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_n(q_n,k_1\ldots k_n)$且 $t(k_1\ldots k_s)=\sigma$,定义 $\overline{\delta_n}(h)=h\overline{\delta_\sigma(q_n)}$;

若存在 $s\leqslant n-1$,使得 $h=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_s(q_s,k_1\ldots k_s)0(l)$,定义 $\overline{\delta_n}(h)=h$.

在 n 步 ID 上定义对偶运算:

- 若 $h_1=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_n(q_n,k_1\ldots k_n)$,定义 $\overline{h_1}=h_1$;
- 若 $h_2=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_t(q_t,k_1\ldots k_t)0(l)$, $\overline{h_2}=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_t(q_t,k_1\ldots k_t)0(c(l))$.

引理 2. 对于 $\mathcal{L}^*(FH_n|_{R_A(\xi)})$ 中任意的元素 h^*恒有 $\overline{\delta_n(h^*)}=\overline{\delta_n}(\overline{h^*})$ 成立,其中,记号 $\mathcal{L}^*(FH_n|_{R_A(\xi)})$ 表示:

$$\mathcal{L}^*(\cup_{t\in R_A(\xi)}FH_n|_t).$$

证明:不妨设 $h^*=\vee_{i\in I}\wedge_{j\in J}h_{i_j}$,其中, $h_{i_j}\in FH_n|_t$.由于δ_n和 $\overline{\delta_n}$ 是同态映射,则

$$\overline{\delta_n}(\overline{h^*})=\overline{\delta_n}(\overline{\vee_{i\in I}\wedge_{j\in J}h_{i_j}})=\overline{\delta_n}(\wedge_{i\in I}\vee_{j\in J}\overline{h_{i_j}})=\wedge_{i\in I}\vee_{j\in J}\overline{\delta_n}(\overline{h_{i_j}}),\overline{\delta_n(h^*)}=\overline{\delta_n(\vee_{i\in I}\wedge_{j\in J}h_{i_j})}=\overline{\vee_{i\in I}\wedge_{j\in J}\delta_n(h_{i_j})}=\wedge_{i\in I}\vee_{j\in J}\overline{\delta_n(h_{i_j})}.$$

下面只需证明 $\overline{\delta_n}(\overline{h_{i_j}})=\overline{\delta_n(h_{i_j})}$,引理 2 即得证.

如果 $h_{i_j}=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_n(q_n,k_1\ldots k_n)$,$t(k_1\ldots k_n)=\sigma$,则 $\delta_n(\overline{h_{i_j}})=\delta_n(h_{i_j})=h_{i_j}\delta_\sigma(q_n)$.因此,

$$\overline{\delta_n}(\overline{h_{i_j}})=h_{i_j}\overline{\delta_\sigma(q_n)}.$$

另一方面,$\overline{\delta_n(h_{i_j})}=\overline{h_{i_j}\delta_\sigma(q_n)}=h_{i_j}\overline{\delta_\sigma(q_n)}$.故结论成立.

情况 2:如果 $h_{i_j}=(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)...k_s(q_s,k_1...k_s)0(l),\delta_n(\overline{h_{i_j}})=\overline{h_{i_j}}$,因此 $\overline{\delta_n}(\overline{h_{i_j}})=\overline{h_{i_j}}$.

另一方面,由 $\overline{\delta_n}$ 定义知 $\overline{\delta_n}(\overline{h_{i_j}})=\overline{h_{i_j}}$.结论成立. □

下面引入格值计算树的概念.

定义 14. 设 A 为一格值交替树自动机,ξ为给定的输入树,递归定义 A 在ξ上的格值计算树 $T(A,\xi)$如下.

(1) $T(A,\xi)$的根节点的标记为(q_0,ε),其中,q_0 是 A 的初始状态.

(2) 节点 u 的标记为 h',将 h'唯一表示成$(\underset{i\in I}{\wedge}h_i)\wedge(\underset{j\in J}{\wedge}h_j)\wedge(\underset{p\in P}{\wedge}h_p)$的形式,其中,

- h_i 形如$(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_n(q_n,k_1\ldots k_n)$,且$\delta(q_n,\xi(k_1\ldots k_n))\neq$true;
- h_j 形如$(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_n(q_n,k_1\ldots k_n)$,且$\delta(q_n,\xi(k_1\ldots k_n))$=true;
- h_p 形如$(q_0,\varepsilon)k_1(q_1,k_1)k_2(q_2,k_1k_2)\ldots k_n(q_n,k_1\ldots k_n)0(l),s\leqslant n-1$.

(3) 计算 $\delta_n(h'):\delta_n(h')=(\underset{i\in I}{\wedge}h_i\delta_{\sigma_i}(q_n))\wedge(\underset{j\in J}{\wedge}h_j0(1))\wedge(\underset{p\in P}{\wedge}h_p)$,并将$\delta_n(h')$展开成析取范式.如果析取范式为$\underset{i\in I'}{\vee}(\underset{j\in J_i'}{\wedge}h_{i_j})$,则 u 有$|I'|$个后继节点,标记分别为 $\underset{j\in J'}{\wedge}h_{i_j}$,$i\in I'$.

例 2:设 $A=(Q,\Sigma,q_0,\delta,K)$为一个具有唯一初始状态的格值交替树自动机,$Q,\Sigma,\delta,K$ 的定义与例 1 一致.求 A 在输入树ξ上的格值计算树.根据定义 14 可得 $T(A,\xi)$,如图 3 所示.

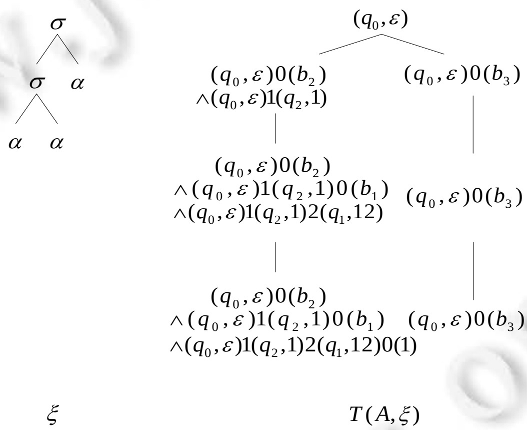

Fig.3 Computation tree of A on ξ

图 3 A 在给定的输入树ξ上的格值计算树

除了按照定义 11 求格值交替树自动机接受的语言 $r_A(\xi)$外,可以根据 A 在ξ上的格值计算树 $T(A,\xi)$的最后一层节点的标记得到 $r_A(\xi)$,只需将 $T(A,\xi)$每条分支上标记结尾处的格值合取,然后再将不同分支上所得结果取析取运算.对例 2 用这种方法,有结果$(b_2\wedge b_1\wedge 1)\vee b_3=b_3$.

设 $T(A,\xi)$是 A 在ξ上的格值计算树.取任意的 $i(0\leqslant i\leqslant Height(t)+1)$,且 $h',\ldots,h^{(k)}$分别为 $T(A,\xi)$第 i 层的 k 个节点的标记,则称 $h'\vee\ldots\vee h^{(k)}$为 $T(A,\xi)$第 i 层的总表达式.借助该定义可将求证定理 2 转化为求证 $T(A,\xi)$第 $Height(\xi)+1$ 层的总表达式与 $T(\bar{A},\xi)$ 第 $Height(\xi)+1$ 层的总表达式互为对偶关系.

现在来证明该结论.

证明:设 h_i 为 $T(A,\xi)$第 i 层的总表达式,$0\leqslant i\leqslant Height(\xi)+1$.由定义 14 可得到 h_i 与 h_{i+1} 有关系$\delta_i(h_i)=h_{i+1}$.

- 当 i=0 时,显然有 $h_0=(q_0,\varepsilon)=\overline{h_0}$.
- 假设当 $i\leqslant k$ 时,$\overline{\delta_i(h_i)}=\overline{\delta_i}(\overline{h_i})$成立.
- 当 $i=k+1$ 时,由定义可知 $\overline{\delta_{k+1}(h_{k+1})}=\overline{\delta_{k+1}(\delta_k(h_k))}$.又根据引理 2“$\mathcal{L}^*(FH_n|_{R_A(\xi)})$中任意的元素 h^*都有

$\overline{\delta_n(h^*)} = \overline{\delta_n}(\overline{h^*})$ 成立”,因此 $\overline{\delta_{k+1}}(\overline{\delta_k(h_k)}) = \overline{\delta_{k+1}(\delta_k(h_k))} = \overline{\delta_{k+1}(h_{k+1})}$.

从而 $\overline{\delta_{k+1}(h_{k+1})} = \overline{\delta_{k+1}}(\overline{h_{k+1}})$ 成立.故对于任意的 $i \leqslant Height(t)+1$,恒有 $\overline{\delta_i(h_i)} = \overline{\delta_i}(\overline{h_i})$ 成立.

特别地,当 $i=Height(t)+1$ 时,$\overline{\delta_i(h_i)} = \overline{\delta_i}(\overline{h_i})$.即 $T(A,\xi)$第 $Height(\xi)+1$ 层的总表达式与 $T(\overline{A},\xi)$ 第 $Height(\xi)+1$ 层的总表达式互为对偶关系.

该结论说明,对任意的输入树ξ, $r_A(\xi) = c(r_{\overline{A}}(\xi))$. □

2.3 格值交替树自动机的表达能力

这一节主要讨论格值交替树自动机、格值树自动机以及格值非确定自动机之间的表达能力.首先证明格值交替树自动机和格值树自动机有相同的表达能力.

定理 3. 对于任意的格值树自动机,存在与其等价的格值交替树自动机.

下面给定理 3 对应的算法表示.

算法 1.

输入:格值树自动机 $A=(Q,\Sigma,q_0,\mu)$;

输出:格值交替树自动机 $A'=(Q,\Sigma,q_0,\delta,K)$.

计算 $\max\{rk(\sigma)|\sigma\in\Sigma\}$

FOR $q\in Q$ **DO**

  FOR $\sigma\in\Sigma^{(k)}$ **DO** //$0\leqslant k\leqslant\max\{rk(\sigma)|\sigma\in\Sigma\}$

    FOR $(q_1,\ldots,q_k)\in Q^k$ **DO**

      $\delta(q,\sigma)$=false //初始化

      IF $\mu_k(\sigma)_{q_1\ldots q_k,q} \neq 0$

      THEN $\delta(q,\sigma) = false \vee (\mu_k(\sigma)_{q_1\ldots q_k,q} \wedge (q_1,1) \wedge \ldots \wedge (q_k,k))$

    END

  END

END

计算 $K=\{1,\ldots,\max\{rk(\sigma)|\sigma\in\Sigma\}\}$ //方向集

算法 1 的时间复杂度完全取决于δ的构造,而δ的构造同时取决于 q,σ以及 $q_1,\ldots,q_k$ 的选取,则该算法的时间复杂度为 $O(|Q|\times|\Sigma|\times|Q|^m)$,即 $O(|Q|^{m+1}\times|\Sigma|)$,其中 $m=\max\{rk(\sigma)|\sigma\in\Sigma\}$.

下面说明算法 1 输出的自动机即为与输入的格值树自动机等价的格值交替树自动机.

对任意的输入树ξ,只需证明 A 在ξ上的接受运行与 A'在ξ上的接受运行一一对应,且二者权重相等.

事实上,将 A 在ξ上的任意接受运行的叶子(格值标记)节点和非叶子节点标记的第 2 指标去掉(只留状态指标)后,即得到 A'在ξ上的一个接受运行,且二者权重相等.反之,将 A'在ξ上的任意接受运行的每个非叶子节点标记换成元素对,令元素对的第 1 指标等于该位置的原始标记,第 2 指标等于该节点的位置,并将每步转移的权重标出.即将转移 $\mu_k(\sigma)_{q_1\ldots q_k,q}$ 标在(q,w)的下一行的最左端,作为叶子节点标记,与 $\mu_k(\sigma)_{q_1\ldots q_k,q}$ 同行的节点标记从左至右依次为$(q_1,w_1),\ldots,(q_k,w_k)$.经过上述转化,即得到$A$ 在ξ上的一个接受运行,且与 A'在ξ上的接受运行权值相同.

例 3:设 $A=(Q,\Sigma,q_1,\mu)$为格值树自动机,其中,$L=1\oplus2^2$,$Q=\{q_1,q_2\}$,$\Sigma=\{\sigma,\alpha\}$,$\sigma\in\Sigma^{(2)}$,$\alpha\in\Sigma^{(0)}$; $\mu_2(\sigma)_{q_1q_2,q_1} = \mu_2(\sigma)_{q_2q_1,q_1} = b_1; \mu_2(\sigma)_{q_2q_2,q_2} = b_3; \mu_0(\alpha)_{\varepsilon,q_1} = b_2; \mu_0(\alpha)_{\varepsilon,q_2} = 1$.下面构造与 A 等价的格值交替树自动机 A'.

令 $A'=(Q,\Sigma,q_1,\delta,K)$,其中,$K=\{1,2\}$;$\delta(q_1,\sigma)=(b_1\wedge(q_1,1)\wedge(q_2,2))\vee(b_1\wedge(q_2,1)\wedge(q_1,2))$;$\delta(q_2,\sigma)=b_3\wedge(q_2,1)\wedge(q_2,2)$;$\delta(q_1,\alpha)=b_2$;$\delta(q_2,\alpha)=1$.

取$\xi=\sigma(\sigma(\alpha,\alpha),\alpha)$,$A$ 在ξ上的接受运行记为 t_1,t_2,t_3,相应地,A'在ξ上的接受运行为 t'_1,t'_2,t'_3(如图 4 所示).

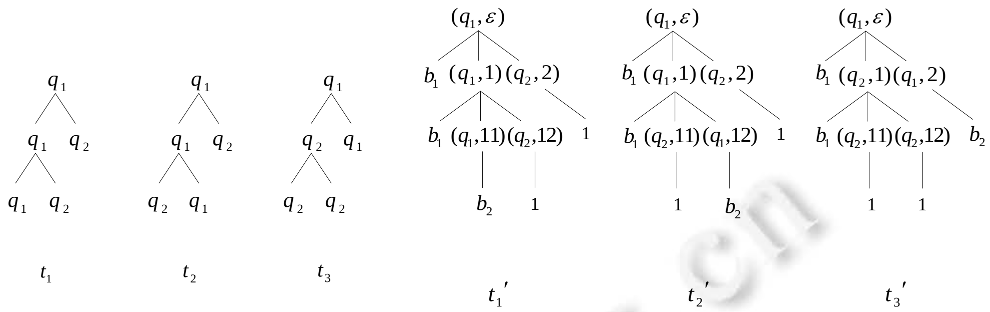

Fig.4 All accepting runs of A, A' on the input tree ξ

图 4 A,A'在给定的输入树ξ上的所有接受运行

定理 4. 对于任意的格值交替树自动机,存在与其等价的格值树自动机.

由于定理 4 的证明并不直观,故在给出其算法和形式化证明之前先用例子来说明构造思路.

同样地,这里仅考虑具有唯一初始状态的格值交替树自动机 $C=(Q,\Sigma,q_2,\delta,K)$.其中,$L=[0,1]$;$Q=\{q_1,q_2,q_3,q_4,q_5\}$,$\Sigma=\{\sigma,\alpha,\beta\}$,$\sigma\in\Sigma^{(2)}$,$\alpha,\beta\in\Sigma^{(0)}$,$K=\{1,2\}$.状态转移函数见表 1.

Table 1 Transition functions

表 1 状态转移表

δ	σ	α	β
q_2	$0.2\vee(((0.4\wedge(1,q_1)\wedge(2,q_2))\vee(0.3\wedge(2,q_1)\wedge(1,q_2)))\wedge(1,q_4)$	true	false
q_1	$(0.5\wedge(q_2,1)\wedge(q_2,2))\vee((q_1,2)\wedge(q_1,1))$	false	true
q_3	$(q_1,2)\wedge(q_4,1)\wedge(q_5,1)$	false	true
q_4	$(0.2\wedge(q_3,1)\wedge(q_3,2))\vee((q_4,1)\wedge(q_4,2))$	true	true
q_5	false	true	false

对于给定的输入树ξ,将 C 在ξ上的任意接受运行 t 按下列方法转化为 t'的形式.

(1) 令 $t'(\varepsilon)$等于 $t(\varepsilon)$二元组中的状态指标 q_2.

(2) 将第 1 层所有非格值元素按第 2 个指标分类,指标一致的放在一起,且将不同组的元素按第 2 指标的字典序从左到右排列.将排列好的每组中元素位置指标省去,形成新的多元状态组 $P_1,\dots,P_{rk(\xi(\varepsilon))}$,并将这些多元状态组作为 $t'(\varepsilon)$的后继状态.令 $\mu_{rk(\xi(\varepsilon))}(\xi(\varepsilon))_{p_1\cdots p_{rk(\xi(\varepsilon))},t(\varepsilon)}$ 等于第 1 层出现的格值元素;若没有格值元素出现则令其为 1,其中,P_i 为新形成的第 i 个状态组($i=1,\dots,rk(\xi(\varepsilon))$).即将 q_2,q_4 放在一起为一个二元状态组,记为(q_2,q_4);q_1 单独为一组.令(q_2,q_4)和 q_1 为 $t'(\varepsilon)$的后继状态,且 $\mu_2(\sigma)_{(q_2,q_4)q_1,q_2}=0.3$.

(3) 将 t 第 1 层中同属于第 i 个状态组的元素全部拿来,把它们的非格值后继按步骤(2)中的方法分类合并排列,略去第 2 指标,并将这些新的多元状态组作为 $t'(i)$的后继.令转移权重等于第 i 个状态组的所有格值后继的合取值;若没有格值后继,令权重为 1.即(q_2,q_4)的后继从左至右为(q_1,q_3,q_4)和(q_2,q_3),且

$$\mu_2(\sigma)_{(q_1,q_3,q_4)(q_2,q_3),(q_2,q_4)}=0.2.$$

若某个状态组仅有格值后继,例如节点标记$(q_1,2)$,则令 $\mu_0(\xi(2))_{\varepsilon,q_1}=\mu_0(\beta)_{\varepsilon,q_1}$ 等于其格值后继.

(4) 对每层元素重复过程(3),直到 t 的第 $Height(t)$层元素全部按照步骤(3)转化后立即停止(结果如图 5、图 6 所示).

将 C 在ξ上的接受运行 t 转化为 t'的过程中所构造的状态转移函数有:

$$\mu_2(\sigma)_{(q_1,q_3,q_4)(q_2,q_3),(q_2,q_4)}=0.2;\mu_2(\sigma)_{(q_2,q_4)q_1,q_2}=0.3;$$

$$\mu_2(\sigma)_{(q_2,q_4,q_5)q_1,(q_2,q_3)}=0.3;\mu_0(\alpha)_{\varepsilon,(q_2,q_4,q_5)}=1;\mu_0(\beta)_{\varepsilon,q_1}=1.$$

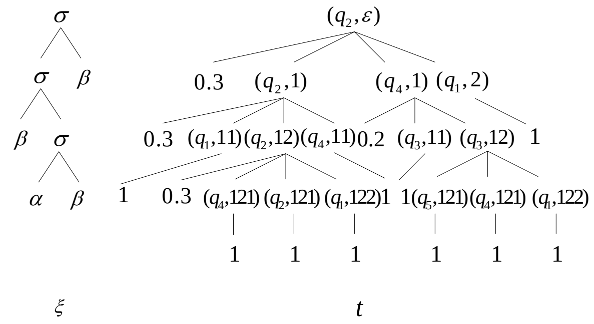

Fig.5 An accepting run of C on input tree ξ, denoted by t

图 5 C 在给定的输入树ξ上的某一接受运行 t

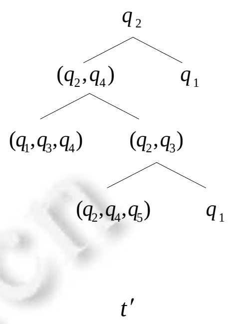

Fig.6 A run corresponding to t

图 6 将 t 按上述方法转化而得的 t'

注意到,第(2)步形成的新的元素组个数不大于 $rk(\xi(\varepsilon))$.若个数小于 $rk(\xi(\varepsilon))$,则需引入额外状态,使得加入额外状态之后总的后继个数等于 $rk(\xi(\varepsilon))$.同理,第(3)步、第(4)步两个步骤中也可能遇到这样的情况,也要做相同的处理,具体做法见算法 2.算法中提到的 q^*即为引入的额外状态.算法 2 为定理 4 所对应的将给定的格值交替树自动机转化为与之等价的格值树自动机的算法表示.

注:下文中出现的$\delta^S(q,\sigma)$为$\delta(q,\sigma)$的最简终展开式,相同的表示不再赘述.

算法 2.

输入:格值交替树自动机 $A=(Q,\Sigma,q_0,\delta,K)$;

输出:格值树自动机 $A'=(Q',\Sigma,q_0,\mu)$.

$Q'=\{q^*\}$. //初始化且 $q^*\notin Q$

FOR n **DO** //$1\leqslant n\leqslant|Q|$

  $Q'=Q'\cup Q^n$ // $Q^n=\underbrace{Q\times\ldots\times Q}_{n\text{-times}}$

END

FOR $(q_1,\ldots,q_k)\in Q^n$ **DO** //$1\leqslant n\leqslant|Q|$

  FOR $\sigma\in\Sigma^{(k')}$ **DO** //$0\leqslant k'\leqslant\max\{rk(\sigma)|\sigma\in\Sigma\}$

  $G_p=\{u_{p,1},\ldots,u_{p,p_s}\}$ //$1\leqslant p\leqslant n, u_{p,1},\ldots,u_{p,p_s}$ 为$\delta^S(q_p,\sigma)$的项

  $G=G_1\times\ldots\times G_n$

    FOR $(u_1,\ldots,u_n)\in G$ **DO**

      对 u_i 中的元素对$(q^{(m)},j)$

      $A_i^{(u_1,\ldots,u_n)}=\varepsilon$ //初始化

      IF $j=m$

      THEN $A_j^{(u_1,\ldots,u_n)}=(A_j^{(u_1,\ldots,u_n)},q^{(m)})$

    END

    IF $A_{i_0}=\varepsilon$

    THEN $A_{i_0}=q^*$ //赋值

  $\mu_{k'}(\sigma)_{A_1^{(u_1,\ldots,u_n)}\ldots A_{k'}^{(u_1,\ldots,u_n)},(q_1,\ldots,q_k)}=l_1\wedge\ldots\wedge l_{k'}$ //$l_1,\ldots,l_{k'}$为项 $u_1,\ldots,u_{k'}$的系数

      END

  END

END

$\mu_k(\sigma)_{q^*\ldots q^*,q^*}=1,\mu_0(\alpha)_{\varepsilon,q^*}=1.$

算法的时间复杂度完全取决于μ的构造,而μ的构造同时取决于状态、字母表元素和项 $u_1,\dots,u_n$ 的选取.在构造 A_i 的过程中要进行$|K|\cdot|Q|$次比对,故该算法的时间复杂度为 $O((|K|^{|Q|}+1)^{|Q|}\times|Q|\times|\Sigma|\times|K|)$.

文献[19]中交替树自动机到树自动机的转换,构造的树自动机的状态集为初始输入的交替树自动机状态集的幂集,转移函数为 $f(S_1,\dots,S_n)\to\{q\in Q|S_1\times\{1\}\cup\dots\cup S_n\times\{n\}$满足$\Delta(q,f)\}$.构造过程要考虑任意的 $n(1\leqslant n\leqslant \max\{rk(f)|f\in\Sigma\})$元状态组的转移.这里,状态组中的每个分量 S_i 为初始输入的交替树自动机状态集的幂集的子集.上述算法 2 的构造,格值树自动机的状态集是以初始格值交替树自动机的状态为基础的 $n(1\leqslant n\leqslant|Q|)$元元素组构成的集合,且构造转移函数过程中需要考虑每个这样的元素组.

由算法 2 可得:对于 A 的任意一棵运行树 t,对应地可找到 A'的若干棵运行树 $t_1,\dots,t_p$,使得 $wt(t_1)\vee\dots\vee wt(t_p)=wt(t)$;反过亦有类似结论.

事实上,对于 A'的任意运行树 t',与 t'等价的 A 的运行树 t 满足:

(1) $t(\varepsilon)=(t'(\varepsilon),\varepsilon)$.

(2) 设 t'的第 1 层的标记从左至右依次为 $P_1,\dots,P_s$,若 $P_1,\dots,P_s$ 中没有等于 q^*的,则(q,i)为 $t(\varepsilon)$的后继节点的标记,其中,$q\in P_i(i=1,\dots,s)$, $\mu_s(\xi(\varepsilon))_{P_1\dots P_s,t'(\varepsilon)}$ 为 t 的第 1 层中唯一的格值标记,且放置在第 1 层的最左端;若 $P_1,\dots,P_s$ 中存在 P_j 等于 q^*,则(q,i)为 $t(\varepsilon)$的后继节点的标记,其中,$q\in P_i$(对于任意的 $i\neq j$),同样地,$\mu_s(\xi(\varepsilon))_{P_1\dots P_s,t'(\varepsilon)}$ 为 t 的第 1 层中唯一的格值标记,且放置在第 1 层最左端.

(3) 对于 t 的第 1 层中任意的(q,i),找到$\delta^S(q,\sigma)$中项 $u_j=l_j\wedge(\bigwedge_{i'\in J,i''\in K}(q_{i'},i''))$,使得任意的 $q_{i'}$处在 t'的第 ii''个位置的多元状态集中,则$(q_{i'},ii'')$为(q,i)的一个后继节点标记,l_j 为(q,i)的后继节点中唯一的格值标记,且放置在(q,i)所有后继节点的最左端.依次考虑其他状态,直到全部状态均满足上述条件完毕,得到 t.

例 4:设 $L=[0,1]$,$A=(Q,\Sigma,I,q_0,K)$是一格值交替树自动机,其中,$Q=\{q_0,q_1\}$,$\Sigma=\{\sigma,\alpha\}$,$\sigma\in\Sigma^{(2)}$,$\alpha\in\Sigma^{(0}K=\{1,2\}$,状态转移函数为$\delta(q_0,\sigma)=(0.2\wedge(q_0,1)\wedge(q_1,1)\wedge(q_1,2))\vee(0.3\wedge(q_0,2))$;$\delta(q_0,\alpha)=$true;$\delta(q_1,\sigma)=0.5\wedge(q_0,2)$;$\delta(q_1,\alpha)=$false.

按定理 4 方法构造与 A 等价的格值树自动机 A'如下.

$A'=(Q',\Sigma,q_0,\mu)$,其中,$Q'=Q\cup Q^2\cup\{q^*\}$;转移函数定义如下:

$$\mu_2(\sigma)_{(q_0,q_1)(q_0,q_1),(q_0,q_1)}=0.2;\mu_2(\sigma)_{q^*q_0,(q_0,q_1)}=0.3;\mu_2(\sigma)_{(q_0,q_1)q_1,q_0}=0.2;$$

$$\mu_2(\sigma)_{q^*q_0,q_0}=0.3;\mu_2(\sigma)_{q^*q_0,q_1}=0.5;\mu_2(\sigma)_{q^*q^*,q^*}=1;\mu_0(\alpha)_{\varepsilon,q_0}=1;\mu_0(\alpha)_{\varepsilon,q^*}=1.$$

其他没有提到的转移权重为 0.

以输入树$\xi=\sigma(\sigma(\alpha,\alpha),\sigma(\alpha,\alpha))$为例说明 A 在ξ上的运行与 A'在ξ上运行的对应关系,如图 7、图 8 所示.

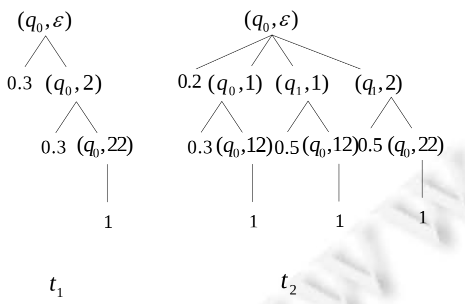

Fig.7 All accepting runs of A on input tree ξ, denoted by t_1, t_2

图 7 A 在给定的输入树ξ上的所有接受运行 t_1,t_2

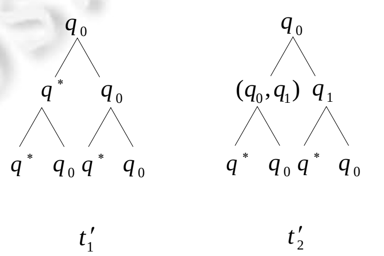

Fig.8 All accepting runs of A' on input tree ξ, t_1,t_2 denoted by t_1',t_2'

图 8 A'在给定的输入树ξ上的所有接受运行 t_1',t_2'

A 接受ξ与 A'接受ξ的程度分别为:

$r_A(\xi)=wt(t_1)\vee wt(t_2)=0.2\vee0.3=0.3$;

$r_{A'}(\xi)=h_\mu(\xi)_{q_0}=(h_\mu(\xi_1)_{(q_0q_1)}\wedge h_\mu(\xi_2)_{q_1}\wedge\mu_2(\sigma)_{(q_0q_1)q_1,q_0})\vee(h_\mu(\xi_1)_{q^*}\wedge h_\mu(\xi_2)_{q_0}\wedge\mu_2(\sigma)_{q^*q_0,q_0})=0.2\vee0.3=0.3.$

其中,$\xi_1=\xi_2=\sigma(\alpha,\alpha)$.

接下来讨论如何利用格值非确定自动机来模拟格值交替树自动机.这里的模拟是指:对于格值交替树自动机的任意输入树,存在格值非确定自动机中的输入字符串使得前者的接受程度等于后者;同样地,对于格值非确定自动机的任意输入字符串,存在格值交替树自动机的输入树,使得二者被接受的程度相等.这里的模拟包含着“等价”的含义,但并非实质上的等价,因为前者接受的是树,后者接受的是字符串.

定理 5. 设 A 是一格值交替树自动机,则可找到格值非确定自动机来模拟 A.

类似地,在给出定理 5 的形式化证明前,先通过一个例子来说明构造思想.

这里仍然采用定理 4 证明前的例子.将ξ上的运行 t 转化成串 w 上的路径 P,方法如下.

(1) 将$\xi=\sigma(\sigma(\beta,\sigma(\alpha,\beta)),\beta)$每层字符按照其位置的字典序排列,并将其位置标于该字符右下角处.将每层所得到的字符串依次连接,得到新的字符串 $w=\sigma_\varepsilon(\sigma_1\beta_2)(\beta_{11}\sigma_{12})(\alpha_{121}\beta_{122})$.每个字符组可看作单个字符属于 $(\Sigma_{K^*})^*$ 的情况,例如 $\sigma_1\beta_2\in(\Sigma_{K^*})^*$.

(2) 将$\{(q_2,\varepsilon)\}$作为路径 P 的初始状态;把 t 的第 1 层的非格值元素放在一起作为一个整体,将其作为$\{(q_2,\varepsilon)\}$经过字符σ_ε输入后可能到达的后继状态,并令转移权重为该层出现的所有格值元素的合取值.$\{(q_2,\varepsilon)\}$经过 σ_ε输入,可能到达状态$\{(q_2,1),(q_4,1),(q_1,2)\}$,且转移权重 $\delta_n(\{(q_2,\varepsilon)\},\sigma_\varepsilon,\{(q_2,1),(q_4,1),(q_1,2)\})=0.3$.对每层元素重复过程(2),直到 t 的第 $Height(t)+1$ 层停止.

将 t 按上述方法转化得到路径 P(P 可看作某一格值非确定自动机的运行路径):

$$\{(q_2,\varepsilon)\}\xrightarrow{\sigma_\varepsilon/0.3}\{(q_2,1),(q_4,1),(q_1,2)\}\xrightarrow{\sigma_1\beta_2/0.2}\{(q_1,11),(q_2,12),(q_4,11),(q_3,11),(q_3,12)\}$$
$$\xrightarrow{\beta_{11}\sigma_{12}/0.3}\{(q_4,121),(q_2,121),(q_1,122),(q_5,121)\}\xrightarrow{\alpha_{121}\beta_{122}/1}\{\varnothing\}.$$

另外,在定理 5 证明之前需引入完备的格值交替自动机的概念,该概念在证明中有重要作用.

定义 15. 若格值交替树自动机满足:对任意的 $q\in Q,\sigma\in\Sigma$,当$\delta(q,\sigma)\neq$true 且$\delta(q,\sigma)\neq$false 时,对于$\delta^S(q,\sigma)$中的任意一项,任取 $0\leqslant k\leqslant rk(\sigma)$,均存在第 2 指标为 k 的元素对出现在该项中,则称该格值交替树自动机是完备的.

注意到,例 1 中提到的格值交替树自动机不是完备的,比如对于$\delta^S(q_0,\sigma)$的第 2 项 $b_2\wedge(q_2,1)$,该项缺少第 2 指标为 2 的元素对;对于第 3 项 b_3,该项既缺少第 2 指标为 1 又缺少指标为 2 的元素对.

可以证明,格值交替树自动机和完备的格值交替树自动机有如下关系.

命题 5. 对于任意的格值交替树自动机,存在与其等价的完备的格值交替树自动机.

证明:设 $A=(Q,\Sigma,q_0,\delta,K)$为任意的格值交替树自动机.构造 $A'=(Q\cup\{q^\Delta\},\Sigma,q_0,\delta',K)$:

(1) 将破坏 A 完备性的转移全部取出,做如下处理:若$\delta^S(q',\sigma)$中存在某项 $u=l\wedge(\bigwedge_{i\in I}(q_i,i))$,使得 u 中不存在第 2 指标为 $j(j\in\{1,\ldots,rk(\sigma)\})$的元素对,则将$(q^\Delta,j)$添加到 u 中.重复上述过程,直到最后得到的 u'中存在第 2 指标为 $j'(j'=1,\ldots,rk(\sigma))$的元素对时结束.将$\delta^S(q',\sigma)$中每一项进行改造得到$\delta'(q',\sigma)$;

(2) 增加转移:对任意的$\sigma\in\Sigma$,令$\delta'(q^\Delta,\sigma)$=true.

从构造中不难看出,步骤(1)、步骤(2)两步之后使得某些运行树仅增加了一些叶子节点为 1 的分支,故不改变整体权重,从而 A 和 A'等价. □

例 5:将例 2 中的格值交替树自动机转化为与其等价的完备的格值交替树自动机.

$\delta^S(q_0,\sigma)$中第 2 项 $b_2\wedge(q_2,1)$缺少第 2 指标为 2 的元素对,将第 2 项中添加$(q^\Delta,2)$,使其变为 $b_2\wedge(q_2,1)\wedge(q^\Delta,2)$;同理,第 3 项 0.2 既缺少第 2 指标为 1 又缺少指标为 2 的元素对,则添加$(q^\Delta,1)$和$(q^\Delta,2)$,使其变为 $b_3\wedge(q^\Delta,1)\wedge(q^\Delta,2)$.令添加过后的公式等于$\delta'(q_0,\sigma)$.

按照上述做法得到的另一格值树自动机 $A'=(Q\cup\{q^\Delta\},\Sigma,q_0,\delta',K)$的转移函数如下.

- $\delta'(q_0,\sigma)=(b_1\wedge(q_1,1)\wedge(q_1,2))\vee(b_2\wedge(q_2,1)\wedge(q^\Delta,2))\vee(b_3\wedge(q^\Delta,1)\wedge(q^\Delta,2))$;
- $\delta'(q_0,\alpha)$=true;$\delta'(q_1,\sigma)=b_2\wedge(q_2,1)\wedge(q^\Delta,2)$;
- $\delta'(q_1,\alpha)$=true;$\delta'(q_2,\sigma)=b_1\wedge(q^\Delta,1)\wedge(q_1,2)$;
- $\delta'(q_2,\alpha)$=true;$\delta'(q^*,\sigma)$=true;$\delta'(q^*,\alpha)$=true.

ξ上的接受运行 t_1',t_2' 如图 9 所示.

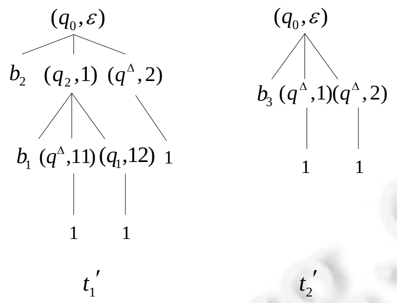

Fig.9 All accepting runs of A' on input tree ξ, denoted by t_1',t_2'

图 9 A'在给定的输入树ξ上的所有接受运行 t_1',t_2'

注意到, t_1',t_2' 分别对应于图 2 中的 t_1,t_2,且与 t_1,t_2 相比分别增加分支:

$$\begin{array}{ccccccc} (q_0,\varepsilon) & & & & & & \\ \downarrow & & & & & & \\ (q_2,1) & & (q_0,\varepsilon) & & (q_0,\varepsilon) & & (q_0,\varepsilon) \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ (q^{\Delta},11) & & (q^{\Delta},2) & & (q^{\Delta},1) & & (q^{\Delta},2) \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ 1 & , & 1 & \text{和} & 1 & , & 1\ . \end{array}$$

同时, $r_{A'}(\xi)=wt(t_1')\vee wt(t_2')=b_3$.

下面来证明定理 5,这里只需考虑完备的格值交替树自动机即可.

证明:设 $A=(Q,\Sigma,q_0,\delta,K)$为任意的完备的格值交替树自动机.构造格值非确定自动机:

$$A_n=(2^{Q\times K^*},(\Sigma_{K^*})^*,(q_0,\varepsilon),\delta_n,F).$$

这里与一般情形不同的是,A_n 的单个输入字符为 $(\Sigma_{K^*})^*$ 中元素,输入串属于 $((\Sigma_{K^*})^*)^*$.$F=\{\varnothing\}$.

为了使Σ上的输入树与 $(\Sigma_{K^*})^*$ 上的输入串联系起来,做如下构造.

对任意的非空集合 $T\in 2^{Q\times K^*}$,不妨设 $T=\{(q_1,w_1),\dots,(q_k,w_k)\}$,当输入字符 $a=\sigma_{w'}\sigma_{w''}\dots\sigma_{w^{(m)}}$ ($i=w',\dots,w^{(m)}$为按字典序排列的 $(\Sigma_{K^*})^*$ 中的元素)满足:对于任意的$(q_i,w_i)(i=w',\dots,w^{(m)})$,存在 $j_i\in\{1,\dots,m\}$使得 $w^{(j_i)}=w_i,|w_1|=\dots=|w_k|$,且 T'为满足 $\mathop{\wedge}\limits_{i=1}^{k}\delta^s(q_i,\sigma_{w_i})$ 的任意集合,T 到 T'有下列转移:

$$\delta_n(T,a,T')=\bigvee_{\{p_{w_ik_1},\dots,p_{w_ik_{m_i'}}\}}\mathop{\wedge}\limits_{i=1}^{k}l_{m_i'}(\sigma_{w_i})_{p_{w_ik_1},\dots,p_{w_ik_{m_i'}},q_i},$$

其中,

- $\{p_{w_ik_1},\dots,p_{w_ik_{m_i'}}\}\subseteq T',m_i=rk(\sigma_{w_i}),m_i\leqslant m_i',p_{w_ik_j}$ 表示某个第 2 指标为 w_ik_j 的元素对(形如$(q^{(j)},w_ik_j)$);
- $(q^{(1)},k_1),\dots,(q^{(m_i')},k_{m_i'})$ 以极小方式满足 $\delta(q_i,\sigma_{w_i'})$ 的权重为 $l_m(\sigma_{w_i})_{p_{w_ik_1},\dots,p_{w_ik_{m_i'}},q_i}$;
- 另外, $\mathop{\cup}\limits_{i=1}^{k}\{p_{w_ik_1},\dots,p_{w_ik_{m_i'}}\}=T'$.

若 $\{p_{w_ik_1},\dots,p_{w_ik_{m_i'}}\}$ 为空集,则意味着 $\delta(q_i,\sigma_{w_i})=\text{true}$,即$\varnothing$以极小方式满足 $\delta(q_i,\sigma_{w_i})$ 的权重为 1.

若 $T=\varnothing$,则对于任意的输入字符 $a\in(\Sigma_{K^*})^*$,定义$\delta_n(\varnothing,a,\varnothing)=1$.其他没有提到的转移权重均为 0.

下面说明 A_n 即为模拟 A 的格值非确定自动机.

事实上,对于任意使得 $r_A(t)\neq 0$ 的运行树 t,构造 $((\Sigma_{K^*})^*)^*$ 上的串 $w=a_1a_2\dots a_{Height(t)+1}$,其中, $a_i\in(\Sigma_{K^*})^*$,$a_1=t(\varepsilon)$,

将树第 $i(i=0,\ldots,Height(t))$层字符从左到右排列并把其在 t 中的位置作为下标标在字符右下方,并令 a_{i+1} 等于这个排列.由构造可知:t 上的每个运行唯一地对应于 w 上的某个运行,且权重相等,则 $L_{A_n}(w)=r_A(t)$.

对任意的 $w'=a_1a_2\ldots a_{Height(t)+1}w''$, $w''\in((\Sigma_{K^*})^*)^*$,亦有 $L_{A_n}(w')=r_A(t)$. 因为 w''上的转移必为$\delta_n(\varnothing,w'',\varnothing)=1$,这意味着,任给 $w\in((\Sigma_{K^*})^*)^*$,若 $L_{A_n}(w)\neq 0$,则 w 与一个 t 对应或者 w 的某个前缀与一个 t 对应,即将 w 中每个字符按右下角位置排列形成树 t,或者将 w 的某个前缀中每个字符按右下角位置排列形成树 t.

类似地,有 $r_A(t)=L_{A_n}(w)$.

综上所述,A_n 即为模拟格值交替树自动机 A 的格值非确定自动机. □

A_n 的状态个数跟 K^*的基数有关,所以定理 5 构造的格值非确定自动机的“有用”状态可能无限.由于定理 5 的研究是为了在理论上揭示交替树自动机和非确定自动机的内部联系,故即使出现无限状态,也并不会到影响二者的联系和转化过程.

例 6:令 $L=1\oplus 2^2$.设 $A=(Q,\Sigma,q_1,\delta,K)$,其中,$K=\{1,2\}$;$\delta(q_1,\sigma)=(b_2\wedge(q_2,1)\wedge(q_3,2))\wedge(b_3\wedge(q_3,1)\wedge(q_2,2))$;$\delta(q_2,\sigma)=b_1\wedge(q_3,1)\wedge(q_3,2)$;$\delta(q_3,\sigma)=\text{false}$;$\delta(q_1,\alpha)=\text{false}$;$\delta(q_2,\alpha)=\text{true}$;$\delta(q_3,\alpha)=\text{true}$.

A 的所有接受运行树有 4 棵,记为 t_1,t_2,t_3,t_4,它们分别是在输入树ξ_1,ξ_1,ξ_2,ξ_3 上的接受运行(如图 10 所示).

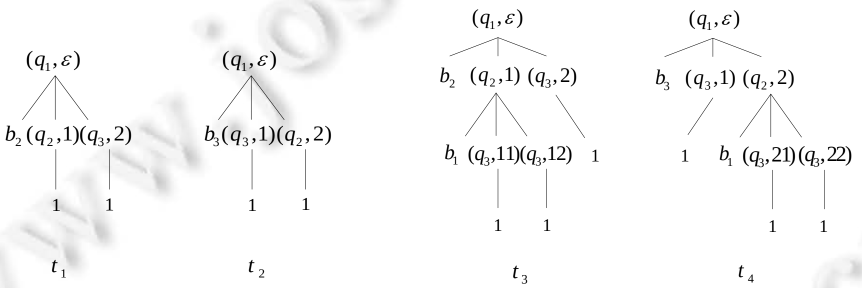

Fig.10 All accepting runs of A, denoted by t_1, t_2, t_3, t_4

图 10 A 的所有接受运行 t_1,t_2,t_3,t_4

对应的输入树分别为:$\xi_1=\sigma(\alpha,\alpha)$,$\xi_2=\sigma(\sigma(\alpha,\alpha),\alpha)$,$\xi_3=\sigma(\alpha,\sigma(\alpha,\alpha))$且 $r_A(\xi_1)=wt(t_1)\vee wt(t_2)=b_2\vee b_3=b_3$;$r_A(\xi_2)=wt(t_3)=b_2\wedge b_1=b_2$;$r_A(\xi_3)=wt(t_4)=b_3\wedge b_1=b_2$.

构造 $A_n=(2^{Q\times K^*},(\Sigma_{K^*})^*,(q_1,\varepsilon),\delta_n,F)$,其中,$F=\{\varnothing\}$.

- $\delta_n(\{(q_1,\varepsilon)\},\sigma_\varepsilon,\{(q_2,1),(q_3,2)\})=b_2$;$\delta_n(\{(q_1,\varepsilon)\},\sigma_\varepsilon,\{(q_3,1),(q_2,2)\})=b_3$;
- $\delta_n(\{(q_2,1),(q_3,2)\},\alpha_1\alpha_2,\varnothing)=1$;$\delta_n(\{(q_3,1),(q_2,2)\},\alpha_1\alpha_2,\varnothing)=1$;
- $\delta_n(\{(q_2,1),(q_3,2)\},\sigma_1\alpha_2,\{(q_3,11),(q_3,12)\})=b_1$;$\delta_n(\{(q_3,1),(q_2,2)\},\sigma_1\sigma_2,\{(q_3,21),(q_3,22)\})=b_1$;
- $\delta_n(\{(q_3,11),(q_3,12)\},\alpha_{11}\alpha_{12},\varnothing)=1$;$\delta_n(\{(q_3,21),(q_3,22)\},\alpha_{21}\alpha_{22},\varnothing)=1$;

$\delta_n(\varnothing,a,\varnothing)=1$,对任意的 $a\in(\Sigma_{K^*})^*$.

A_n 的所有接受路径有 4 类 P_1,P_2,P_3,P_4,分别对应于 4 类输入串 W_1,W_1,W_2,W_3.

P_1 是以$\{(q_1,\varepsilon)\}\xrightarrow{\sigma_\varepsilon/b_2}\{(q_2,1),(q_3,2)\}\xrightarrow{\alpha_1\alpha_2/1}\varnothing$ 为最开始的 2 步转移,从$\varnothing$后可能发生自循环的一类有限长度路径,形如$\{(q_1,\varepsilon)\}\xrightarrow{\sigma_\varepsilon/b_2}\{(q_2,1),(q_3,2)\}\xrightarrow{\alpha_1\alpha_2/1}\varnothing(\xrightarrow{a/1}\varnothing\xrightarrow{b/1}\varnothing\xrightarrow{c/1}\ldots)$, 对任意 $a,b,c,\ldots\in(\Sigma_{K^*})^*$. P_1 中的路径分别对应于输入串$\sigma_\varepsilon(\alpha_1\alpha_2),\sigma_\varepsilon(\alpha_1\alpha_2)a,\sigma_\varepsilon(\alpha_1\alpha_2)ab,\sigma_\varepsilon(\alpha_1\alpha_2)abc,\ldots$,一类以$\sigma_\varepsilon(\alpha_1\alpha_2)$为前缀的输入串. W_1 表示这类输入串的集合(上述括号内的转移可能发生也可能不发生,以下类似).

同理,P_2 是形如$\{(q_1,\varepsilon)\}\xrightarrow{\sigma_\varepsilon/b_3}\{(q_3,1),(q_2,2)\}\xrightarrow{\alpha_1\alpha_2/1}\varnothing(\xrightarrow{a/1}\varnothing\xrightarrow{b/1}\varnothing\xrightarrow{c/1}\ldots)$的一类有限长度路径,对任意的 $a,b,c,\ldots\in(\Sigma_{K^*})^*$.$P_2$ 中的路径分别对应于输入串$\sigma_\varepsilon(\alpha_1\alpha_2),\sigma_\varepsilon(\alpha_1\alpha_2)a,\sigma_\varepsilon(\alpha_1\alpha_2)ab,\sigma_\varepsilon(\alpha_1\alpha_2)abc,\ldots$,一类以$\sigma_\varepsilon(\alpha_1\alpha_2)$为前缀的输入串.$W_1$ 表示这类输入串的集合.

P_3 是形如$\{(q_1,\varepsilon)\}\xrightarrow{\sigma_\varepsilon/b_2}\{(q_2,1),(q_3,2)\}\xrightarrow{\sigma_1\alpha_2/b_1}\{(q_3,11),(q_3,12)\}\xrightarrow{\alpha_{11}\alpha_{12}/1}\varnothing(\xrightarrow{a/1}\varnothing\xrightarrow{b/1}\varnothing\xrightarrow{c/1}\ldots)$

的一类有限长度路径(对任意的 $a,b,c,\ldots\in(\Sigma_{K^*})^*$).P_3 中的路径分别对应于输入串 $\sigma_\varepsilon(\sigma_1\alpha_2)(\alpha_{11}\alpha_{12}),\sigma_\varepsilon(\sigma_1\alpha_2)(\alpha_{11}\alpha_{12})a,\sigma_\varepsilon(\sigma_1\alpha_2)(\alpha_{11}\alpha_{12})ab,\sigma_\varepsilon(\sigma_1\alpha_2)(\alpha_{11}\alpha_{12})abc,\ldots$,一类以 $\sigma_\varepsilon(\sigma_1\alpha_2)(\alpha_{11}\alpha_{12})$ 为前缀的输入串.W_2 表示这类输入串的集合.

P_4 是形如 $\{(q_1,\varepsilon)\}\xrightarrow{\sigma_\varepsilon/b_3}\{(q_3,1),(q_2,2)\}\xrightarrow{\alpha_1\sigma_2/b_1}\{(q_3,21),(q_3,22)\}\xrightarrow{\alpha_{21}\alpha_{22}/1}\varnothing(\xrightarrow{a/1}\varnothing\xrightarrow{b/1}\varnothing\xrightarrow{c/1}\ldots)$ 的一类有限长度路径,对任意的 $a,b,c,\ldots\in(\Sigma_{K^*})^*$ .P_4 中的路径且分别对应于输入串 $\sigma_\varepsilon(\alpha_1\sigma_2)(\alpha_{21}\alpha_{22}),\sigma_\varepsilon(\alpha_1\sigma_2)(\alpha_{21}\alpha_{22})a,\sigma_\varepsilon(\alpha_1\sigma_2)(\alpha_{21}\alpha_{22})ab,\sigma_\varepsilon(\alpha_1\sigma_2)(\alpha_{21}\alpha_{22})abc,\ldots$,一类以 $\sigma_\varepsilon(\alpha_1\sigma_2)(\alpha_{21}\alpha_{22})$ 为前缀的输入串.W_3 表示这类输入串的集合.

$$\begin{aligned}
L_{A_n}(w_1)&=(\delta_n(\{(q_1,\varepsilon)\},\sigma_\varepsilon,\{(q_2,1),(q_3,2)\})\wedge\delta_n(\{(q_2,1),(q_3,2)\},(\alpha_1\alpha_2)w_1',\varnothing))\vee\\
&\quad(\delta_n(\{(q_1,\varepsilon)\},\sigma_\varepsilon,\{(q_3,1),(q_2,2)\})\wedge\delta_n(\{(q_3,1),(q_2,2)\},(\alpha_1\alpha_2)w_1',\varnothing))\\
&=b_2\vee b_3=b_3,\\
L_{A_n}(w_2)&=\delta_n((\{(q_1,\varepsilon)\},\sigma_\varepsilon(\sigma_1\alpha_2)(\alpha_{11}\alpha_{12})w_2',\varnothing))=b_2\wedge b_1=b_2,\\
L_{A_n}(w_3)&=\delta_n((\{(q_1,\varepsilon)\},\sigma_\varepsilon(\alpha_1\sigma_2)(\alpha_{21}\alpha_{22})w_3',\varnothing))=b_3\wedge b_1=b_2,
\end{aligned}$$

其中,$w_1\in W_1,w_2\in W_2,w_3\in W_3$ 且 $w_1=\sigma_\varepsilon(\alpha_1\alpha_2)w_1',w_2=\sigma_\varepsilon(\sigma_1\alpha_2)(\alpha_{11}\alpha_{12})w_2',w_3=\sigma_\varepsilon(\alpha_1\sigma_2)(\alpha_{21}\alpha_{22})w_3'$.

综上所述,A 在 W_1 中任意串上的路径 $p_1\in P_1(p_2\in P_2)$ 模拟 A' 在 ξ_1 上的运行 $t_1(t_2)$,A 在 W_2 中任意串上的路径 $p_3\in P_3$ 模拟 A' 在 ξ_2 上的运行 t_3,A 在 W_3 中任意串上的路径 $p_4\in P_4$ 模拟 A' 在 ξ_3 上的运行 t_4.

3 总 结

本文引入格值交替树自动机的概念并对其闭包性质和表达能力展开研究.证明了格值交替树自动机关于交、并、补运算封闭,特别地,利用格值计算树的概念证明了只需对状态转移函数取对偶运算、将终止权重取补即可得到与原格值交替树自动机互补的另一格值交替树自动机.此外,证明了格值交替树自动机与格值树自动机的等价性.这为格值树自动机的补问题提供了另一思路.进一步地,本文指出,可以利用格值非确定型自动机来模拟格值交替树自动机.这一结论体现了非确定型自动机和交替树自动机这两类不同模型之间的内在联系.如何将这些结论推广到无限树上,且如何利用这些自动机理论的结果去讨论计算树逻辑模型检测问题,有待于进一步探讨.

References:

[1] Cao YZ, Ezawa YS. Nondeterministic fuzzy automata. Information Sciences, 2012,191:86−97.

[2] Hopcroft JE, Ullman JD. Formal Languages and Their Relation to Automata. Addison-Wesley, 1969.

[3] Karp RM. Reducibility among combinatorial problems, complexity of computer computations. In: Proc. of the Symp. on Complexity of Computer Computations. New York, 1972. 85−103.

[4] Aho AV, Hopcroft JE. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1974.

[5] Cook SA. The complexity of theorem proving procedures. In: Proc. of the 3rd Annual ACM Symp. on Theory of Computing. Ohio, 1971. 151−158.

[6] Chandra AK, Kozen DC, Stockmeyer LJ. Alternation. Journal of the Association for Computing Machinery, 1981,28(1):114−133.

[7] Kupferman O, Lustig Y. Weak alternating automata are not that weak. ACM Trans. on Computational Logic, 2001,2(3):408−429.

[8] Miyano S, Hayashi T. Alternating finite automata on ω-words. Theoretical Computer Science, 1984,32:321−330.

[9] Leiss E. Note succinct representation of regular languages by Boolean automata. Theoretical Computer Science, 1981,13:323−330.

[10] Muller DE, Schupp PE. Alternating automata on infinite trees. Theoretical Computer Science, 1987,54:267−276.

[11] Muller DE, Schupp PE. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. Theoretical Computer Science, 1995,141:69−107.

[12] Brozozowski JA, Leiss E. On equations for regular languages, finite automata, and sequential networks. Theoretical Computer Science, 1980,10:19−35.

[13] Kupferman O. Automata theory and model checking. http://www.cs.huji.ac.il/~ornak/publications/hbmc15.pdf

[14] Vardi MY. An automata-theoretic approach to linear temporal logic. Lecture Notes in Computer Science, 2005,1043:238−266.

[15] Zhou QL, Zhou WJ, Zhuang L, Su JX. On acceptance conditions of alternating ω-FA. Ruan Jian Xue Bao/Journal of software, 1994, 5(9):56−58 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19940909.htm [doi: 10.13328/j.cnki.jos.1994.09.009]

[16] Almagor S, Kupferman O. Max and sum semantics for alternating weighted automata. Lecture Notes in Computer Science, 2011, 6996:13−27.

[17] Chatterjee M, Doyen L, Henzinger TA. Alternating weighted automata. Lecture Notes in Computer Science, 2009,5699:3−13.

[18] Wei XJ, Li YM. Fuzzy alternating automata over distributive lattices. Information Sciences, 2018,425:34−47.

[19] Comon H, Dauchet M, Gilleron R, Jacquemard F, Lugiez D, Löding C, Tison S, Tommasi M. Tree automata: Techniques and applications. 2007. http://tata.gforge.inria.fr

[20] Droste M, Kuich W, Vogler H. Handbook of Weighted Automata. Berlin, Heidelberg: Springer-Verlag, 2009.

[21] Davey BA, Priestley HA. Introduction to Lattice and Order. 2nd ed., Cambridge: Cambridge University Press, 2002.

[22] Han ZW, Li YM. Quantum Müller automata and monadic second-order quantum logic. Ruan Jian Xue Bao/Journal of Software, 2014,25(1):27−36 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4406.htm [doi: 10.13328/j.cnki.jos.004406]

[23] Li YM. Fuzzy finite automta and monodic second-order Lukasiewicz logic. Chinese Journal of Computers, 2008,31(10):1788−1794 (in Chinese with English abstract).

[24] Xie ZW, Zhai Y, Deng PM, Yi Z. Algebraic properties of probabilistic finite state automta. Journal of Computer Reasearch and Development, 2013,50(12):2691−2698 (in Chinese with English abstract).

[25] Li YM, Pedrycz W. Fuzzy finite automata and fuzzy regular expressions with membership values in lattice-ordered monoids. Fuzzy Sets and Systems, 2005,156(1):68−92.

附中文参考文献:

[15] 周清雷,周文俊,庄雷,苏锦祥.关于交替的ω-有穷自动机的接受条件.软件学报,1994,5(9):56−58. http://www.jos.org.cn/1000-9825/19940909.htm [doi: 10.13328/j.cnki.jos.1994.09.009]

[22] 韩召伟,李永明.量子 Müller 自动机与单体二阶量子逻辑.软件学报,2014,25(1):27−36. http://www.jos.org.cn/1000-9825/4406.htm [doi: 10.13328/j.cnki.jos.004406]

[23] 李永明.模糊有穷自动机与单体二阶 Lukasiewicz 逻辑.计算机学报,2008,31(10):1788−1794.

[24] 谢正卫,翟莹,邓培民,易忠.概率有限状态自动机的代数性质.计算机研究与发展,2013,50(12):2691−2698.

魏秀娟(1990－),女,河南郑州人,博士,主要研究领域为计算智能,加权自动机.

李永明(1966－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算智能,加权自动机理论,模型检测,量子计算与量子信息.