ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.1, January 2010, pp.34−46
doi: 10.3724/SP.J.1001.2010.03456

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

PSL构造双向交换自动机及非确定自动机的方法*

虞 蕾[1,2+], 陈火旺[1]

[1](国防科学技术大学 计算机学院 博士后流动站,湖南 长沙 410073)

[2](第二炮兵工程学院 计算机系,陕西 西安 710025)

Method of Constructing Two-Way Alternating Automata for PSL and Translation to Nondeterministic Automata

YU Lei[1,2+], CHEN Huo-Wang[1]

[1](Postdoctoral Station, School of Computer, National University of Defense Technology, Changsha 410073, China)

[2](Department of Computer Science, Second Artillery Engineering College, Xi'an 710025, China)

+ Corresponding author: E-mail: yuleizj@163.com

Yu L, Chen HW. Method of constructing two-way alternating automata for PSL and translation to nondeterministic automata. *Journal of Software,* 2010,21(1):34−46. http://www.jos.org.cn/1000-9825/3456.htm

Abstract: PSL (property specification language) is a property specification language to describe parallel systems and can be divided into two parts, FL (foundation language) and OBE (optional branching extension). Since OBE is essentially the temporal logic CTL (computation tree logic), and PSL formulas with clock statements can be easily rewritten to unclocked formulas, this paper plays an emphasis on the unclocked FL logic. In order to be model-checked, each FL formula needs to be translated into a verifiable form, usually as an automaton (nondeterministic automaton). The translation into nondeterministic automata can be realized mainly by the construction of alternating automata. The translation rules for the two-way alternating automata from unclocked FL logic are explained in detail in this paper. The core logic of the construction rules is not only limited to an extension of LTL (linear temporal logic) with regular expressions, but considers overall FL operators adequately. A translation method from two-way alternating automata to nondeterministic automata is also provided. Finally, a translation tool from PSL formulas to the above two automata has been written. The complexity of the construction rules for the two-way alternating automata grows linearly with the length of the FL formulas, and at the same time, the correctness of the rules is verified. It is also proved that the two-way alternating automata and its corresponding nondeterministic automata accept the same language. The work above has important theoretical and application values for the modeling and model checking for the complex parallel systems.

Key words: PSL (property specification language); FL (foundation language); two-way alternating automata; nondeterministic automata; model checking

摘 要: PSL(property specification language)是一种用于描述并行系统的属性规约语言,包括线性时序逻辑

* Supported by the National Natural Science Foundation of China under Grant No.60503032 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA010301 (国家高技术研究发展计划(863))

Received 2008-01-31; Accepted 2008-08-11

FL(foundation language)和分支时序逻辑OBE(optional branching extension)两部分.由于OBE就是CTL(computation tree logic),并且具有时钟声明的公式很容易改写成非时钟公式,因此重点研究了非时钟FL逻辑.为便于进行模型检验,每个FL公式必须转化成为一种可验证形式,通常是自动机(非确定自动机).构造非确定自动机的过程主要是通过中间构建交换自动机来实现.详细给出了由非时钟FL构造双向交换自动机的构造规则.构造规则的核心逻辑不仅仅局限于是在LTL(linear temporal logic)基础上的正规表达式,而且全面而充分地考虑了各种FL操作算子的可能性.并且给出了将双向交换自动机转化为非确定自动机的一种方法.最后,编写了将PSL转化为上述自动机的实现工具.FL双向交换自动机的构造规则计算复杂度仅是FL公式长度的线性表达式,验证了构造规则的正确性.在此基础上,证明了双向交换自动机与其转化的等价的非确定自动机接受的语言相同.上述工作对解决复杂并行系统建模和模型验证问题具有重要的理论意义和应用价值.

关键词: PSL(property specification language);FL(foundation language);双向交换自动机;非确定自动机;模型检验

中图法分类号: TP301 **文献标识码:** A

1 引 言

PSL(property specification language)[1]是用于描述并发系统的IEEE标准属性规约语言(IEEE-1850).IBM的Sugar语言是形成PSL的基础[2,3].PSL主要有三大用途:(1) 形式化验证;(2) 动态验证仿真;(3) 功能规约描述(作为功能规约文档).由于PSL易于读写、语法精简、语义清晰、表达能力强,因此作为一种功能描述语言得到了广泛的应用.

PSL是一种分层语言,包括4层:1) 布尔层(Boolean layer),用于构造布尔表达式,是构筑时序表达式的基础;2) 时序层(temporal layer),是语言的核心层,用于实现语言的主要功能,即描述设计模块的时序属性;3) 验证层(verification layer),是与验证工具紧密结合的语言层,用以指示验证工具验证时序层所描述的模型属性规约;4) 建模层(modeling layer),为验证过程所需的测试输入模块或辅助硬件模块建模.时序层可分为FL(foundation language,基础语言)和OBE(optional branching extension,可选分支扩展)两部分.FL如同LTL,是线性时序逻辑,而OBE是一种分支时序逻辑.时序层的FL包含命题操作子、LTL操作子、定义时间粒度的时钟操作子、连续扩展正规表达式(sequential extended regular expressions,简称SEREs)以及abort操作子.FL公式的时钟声明可视为语法Sugar,具有时钟声明的公式很容易改写成非时钟公式(改写规则见官方语言标准[4]).由于OBE就是CTL时序逻辑[5],本文重点研究线性非时钟逻辑FL构造双向交换自动机及非确定自动机的方法.

1.1 逻辑与自动机转化的研究历史

20世纪60年代,Büchi[6]和Rabin[7]首先指出逻辑和自动机的联系.20世纪80年代,自动机广泛应用在模型与时序逻辑程序决策过程中.Streett[8]将自动机应用在扩展命题动态逻辑决策问题上;Vardi,Wolper和Sistla[9,10]使用自动机理论技术应用于扩展线性时态逻辑;Emerson和Sistla[11]提出了CTL*的自动机理论决策过程.作为构建自动机的一个特例,Vardi和Wolper提出了LTL的显示决策和模型检验过程[12].1995年,Gerth,Plede等人提出基于on-the-fly表格的、将furture-time LTL转化为具有扩展接收的不确定自动机的构建方法.Gerth等人是利用显式过程建立只与问题相关部分的自动机,这种方法的优势使其广泛应用于相关的转化过程中,如显式的表格(tableau)构建的改进[13]到支持符号化表示的自动机的转换[14].事实上,许多应用于LTL模型检验的符号化表格方法可视为是自动机转化过程的另一个发展分支,这个分支主要借助于作为非确定的自动机的直接表格解释方法,但这些工作都类似于早期的自动机,并不关心自动机的最小化问题.

Muller,Saoudi和Schupp[15]通过将分支时间扩展逻辑转化成弱交换自动机,描述了交换自动机与时序逻辑的联系.Vardi[16]和Isli[17]给出了LTL的显式自动机构造方法.Miyano和Hayashi给出了在无穷字上将交换自动机转化为非确定自动机的方法[18];然后,Muller,Saoudi和Schupp[19]将这一方法推广到树交换自动机上.Isli[17]进一步研究了有穷和无穷字上的自动机非确定优化方法.

将LTL转化为自动机的最小化问题引起广泛的关注,除了使用自动机的特殊结构属性技术[20]以外,还提出针对非确定[21]和交换自动机[22,23]的各种最小化仿真关系.另外,除了提出上述LTL和CTL*等逻辑的自动机以

外,还有扩展的表达逻辑即命题μ-演算转化为自动机的研究成果[24].

1.2 相关工作

Bustan等人[25]将PSL的核心逻辑定义为LTL_WR,即在LTL的基础上再定义正规表达式REs,并且证明每个LTL_WR公式f都存在一个非确定的Büchi自动机(NBA),复杂度是f的双指数.在表达时序逻辑方面,交换自动机(AFA)比非确定自动机(NFA)的复杂度呈指数形式精简.因此,为构造NFA,Bustan等人首先构造LTL_WR的AFA.但是,由于交换自动机只能产生每个分支都可接收的运行,而事实上,运行遍历的状态既包括可接收也包括不可接收.基于这种情况,文献[25]进一步将AFA转化为NFA.文献[26]介绍了将PSL转化为非确定Büchi自动机(NBA)的传统方法:将PSL中的SEREs首先转化为最小非确定自动机(NFA),而后NFA组织在一起,将PSL公式转变为交换Büchi自动机ABA,最后使用Miyano-Hayashi(MH)方法[18]将ABA转化为NBA.文献[27]指出,这种方法就是基于SAT的有界模型检验,前提是弱交换自动机.文献[28]提出一种基于MH、由PSL属性构造ABA而后转化为对应的NBA的符号化编码方法.文献[27,28]提出的两种方法都试图限制编码大小,但它们都依赖于一个旨在最小化ABA并执行优化方法的库,ABA最小化代价高.即便是对于中等长度的PSL规约,这两种方法也都不能在可接受的时间内完成转化任务.文献[29]提出将PSL转化为符号化表示的NBA的直接编码,主要基于SONF(suffix operator normal form,后缀操作符范式).实验结果表明,SONF结构是一种快速构造NBA的有效方法.文献[30]给出了将PSL的一个包含安全属性的子集SafetyPSL^{det}直接转化为co-universal自动机的方法.由于有穷自动机能够发现违反安全属性的反例,因此此时的模型检验可以简化成不变式"自动机A不处于接收状态"的验证过程.不变式检验不但简单,而且许多模拟工具在不变式中会执行得更好.其缺点是:虽然SafetyPSL^{det}的大部分操作子都能建立相应的非确定CUA,但弱正规表达式r建立的自动机必须先确定化,这样就明显地增加了构建自动机的复杂性.

本文的主要工作与创新是:(1) 在基于文献[1]定义的 PSL 及 FL 的基础上,提出各种非时钟 FL 算子构造双向交换自动机的规则.与文献[25]比较,核心逻辑不局限于是 LTL 基础上的正规表达式,而是全面、充分地考虑了各种 FL 操作算子的可能性,并进行了正确性证明,给出构造规则的复杂度仅是 FL 公式长度的线性表达式;(2) 给出将双向交换自动机转化为非确定自动机的一种方法,证明两者接受的语言相同,并列举了转化的具体实例;(3) 根据上述规则及定理,在 Linux 环境下,利用 C++语言编写了将 PSL 转化为上述自动机的实现工具,并在一个具体实例的背景下,给出了实验结果.这些都为下一步判空作准备,以执行 FL 的模型检验.本文将 FL 首先转化为双向交换自动机,而后构造非确定自动机的理由是:若将 FL 直接转化成非确定自动机会失去 FL 和自动机表达能力的双向(two-way)一致性(correspondence).从本文看,FL 和双向交换自动机在表达能力上等价,这种逻辑和自动机的一致性允许进一步简化和进行非确定构造以提高向非确定自动机的全面转化效率.而且,双向交换自动机的简洁同时意味着可以通过对其有效的直接操作和分析,避免对非确定自动机的直接使用.

本文第 2 节简单介绍 PSL 及 FL 的语法与语义.第 3 节是全文的主体,介绍构造 FL 双向交换自动机及转化为非确定自动机的一种方法.第 4 节介绍将 PSL 转化为双向交换自动机及非确定自动机的实现工具,给出一个队列控制电路的实例,并利用实现工具得到实验结果.最后,总结全文.

2 PSL语法与语义[1]

2.1 PSL语法

定义 1(连续扩展正规表达式(sequential extended regular expression,简称 SERE)).

- 每个布尔表达式 b 是一个 SERE;
- 如果r,r_1,r_2是SEREs,则以下各项都是SERE:

$\bullet\{r\}$ $\bullet r_1;r_2$ $\bullet r_1:r_2$ $\bullet r_1|r_2$ $\bullet r_1\&\&r_2$ $\bullet[*0]$ $\bullet r[*]$

定义 2(基础语言公式(foundation language formulas,简称 FL formulas)).

- 如果 b 是布尔表达式,则 b 和 $b!$都是 FL formulas;

- 如果 φ和ψ是 FL formulas,r 是 SERE,b 是布尔表达式,则以下各项都是 FL formulas:

$$\bullet(\varphi) \quad \bullet\neg\varphi \quad \bullet\varphi\wedge\psi \quad \bullet r! \quad \bullet r \quad \bullet X!\varphi \quad \bullet[\varphi\cup\psi] \quad \bullet\varphi \text{ abort } b \quad \bullet r|\!\rightarrow\varphi$$

定义 3(可选分支扩展公式(optional branching extension formulas,简称 OBE formulas)).

- 每个布尔表达式是一个 OBE 公式;
- 如果f,f_1,f_2是OBE公式,则以下是OBE公式:

$$\bullet(f) \quad \bullet\text{EX } f \quad \bullet\neg f \quad \bullet E[f_1\cup f_2] \quad \bullet f_1\wedge f_2 \quad \bullet\text{EG } f$$

定义 4(Accellera PSL 公式).

- 每个 FL 公式都是一个 Accellera PSL 公式;
- 每个 OBE 公式都是一个 Accellera PSL 公式.

2.2 PSL的语义

2.2.1 非时钟 SEREs 语义

$w\vdash r$表示w紧满足(tightly satisfies)r,其中,w是定义在$\Sigma=2^p\cup\{\bot,\top\}$上的有穷字,原子命题$p\in AP$,$b$为布尔表达式,$r,r_1,r_2$是非时钟SEREs,则:

(1) $w\vdash\{r\}\Leftrightarrow w\vdash r$;

(2) $w\vdash b\Leftrightarrow|w|=1$ 且$w^0\vdash b$;

(3) $w\vdash r_1;r_2\Leftrightarrow\exists w_1,w_2$, s.t. $w=w_1w_2,w_1\vdash r_1$且$w_2\vdash r_2$;

(4) $w\vdash r_1:r_2\Leftrightarrow\exists w_1,w_2,l$, s.t. $w=w_1lw_2,w_1l\vdash r_1$且$lw_2\vdash r_2$;

(5) $w\vdash r_1|r_2\Leftrightarrow w\vdash r_1$或$w\vdash r_2$;

(6) $w\vdash r_1\&\&r_2\Leftrightarrow w\vdash r_1$且$w\vdash r_2$;

(7) $w\vdash[*0]\Leftrightarrow w=\varepsilon$;

(8) $w\vdash r[*]\Leftrightarrow w\vdash[*0]$或$\exists w_1,w_2$, s.t. $w_1\neq\varepsilon,w=w_1w_2,w_1\vdash r$且$w_2\vdash r[*]$.

2.2.2 非时钟 FL 公式语义

公式 $w\vDash\varphi$表示 w 满足φ,其中,w 是一个有穷或无穷字,b 是布尔表达式,r 是非时钟 SERE,φ和ψ是非时钟 FL 公式,则:

(1) $w\vDash(\varphi)\Leftrightarrow w\vDash\varphi$;

(2) $w\vDash\neg\varphi\Leftrightarrow\neg w\vDash\varphi$;

(3) $w\vDash\varphi\wedge\psi\Leftrightarrow w\vDash\varphi$且 $w\vDash\psi$;

(4) $w\vDash b!\Leftrightarrow|w|>0$ 且$w^0\vdash b$;

(5) $w\vDash b\Leftrightarrow|w|=0$ 或$w^0\vdash b$;

(6) $w\vDash r!\Leftrightarrow\exists j<|w|$,s.t. $w^{0\cdots j}\vdash r$;

(7) $w\vDash r\Leftrightarrow\forall j<|w|$,s.t. $w^{0\cdots j}T^\omega\vDash r!$;

(8) $w\vDash X!\varphi\Leftrightarrow|w|>1$ 且$w^{1\cdots}\vDash\varphi$;

(9) $w\vDash[\varphi\cup\psi]\Leftrightarrow\exists k<|w|$,s.t. $w^{k\cdots}\vDash\psi$且$\forall j<k,w^{j\cdots}\vDash\varphi$;

(10) $w\vDash\varphi$ abort $b\Leftrightarrow w\vDash\varphi$或$\exists j<|w|$,s.t. $w^j\vdash b$且$w^{0\cdots j-1}T^\omega\vDash\varphi$;

(11) $w\vDash r\!\rightarrow\varphi\Leftrightarrow\forall j<|w|$,s.t. $w^{0\cdots j}\vdash r,w^{j\cdots}\vDash\varphi$.

3 构造 FL 的双向交换自动机及转化为非确定自动机的方法

3.1 非确定自动机与双向交换自动机定义[31]

3.1.1 非确定自动机

一个非确定自动机是五元序偶$A=\langle\Sigma,Q,Q_0,\rho,F\rangle$,其中,$\Sigma$是有限字母表,$Q$是有穷状态集,$Q_0\subseteq Q$是初始状态集合,$F$是接收状态集$F\subseteq Q$,$\rho$:$Q\times\Sigma\rightarrow 2^Q$是迁移函数,对$\forall(q,a)\in Q\times\Sigma,\rho(q,a)=p_1\vee p_2\vee\ldots\vee p_m$,表示$A$在状态$q$读入字符$a$,

可以有选择地将状态变成$p_1,p_2,\ldots$或p_m,并将读头右移一个方格而指向输入字符串的下一个字符.

非确定自动机A在字$w=w_1w_2...\in\Sigma^w$上的一个运行是一个无穷路径$r=q_0,q_1,\ldots$,其中,$q_0\in Q_0$,且对于任意$i\geq 0$,满足$q_{i+1}\in\rho(q_i,w_i)$.

3.1.2 双向交换自动机(two-way alternating automata)

一个双向交换自动机是五元序偶$A=\langle\Sigma,Q,q_0,\rho,F\rangle$,其中,$\Sigma,Q,F$与非确定自动机的相同,$q_0$是初始状态,迁移函数$\rho:Q\times\Sigma\to B^+(\{-1,0,1\}\times Q)$.其中,$B^+(\{-1,0,1\}\times Q)$为集合$\{-1,0,1\}\times Q$上有限次应用"∧"与"∨"的布尔公式.当读头位于字w的第i位置时,读头可移动到$i-1,i,i+1$处.

例 1:$\rho(s_0,a)=(-1,s_1)\wedge(1,s_2)\vee(0,s_3)$表示当自动机在状态$s_0$时,读头位置$i$所指的字母是$a$,将状态变为$s_1$,并将读头向左移动一个带方格而指向$i-1$ 即输入字符串中的前一个字符,并且将状态变为s_2,且读头右移一格指向$i+1$即输入字符串的下一个字符,或者将状态变为s_3,读头保持原位i不动.若$i=0$,自动机选择第 2 种情况.

双向交换自动机A在字$w\in\Sigma^w$上的一个运行是一棵标记为$(Q\times IN)$的树(T,r),$k\in IN$是读头位置,$r:T\to Q\times IN$,ε是根节点,$r(\varepsilon)=(q_0,0)$,对于任意$x\in T$,若$r(x)=(q_p,k)$,则集合$\{(q,\Delta)|c\in IN,x\cdot c\in T,$且$r(x\cdot c)=(q,k+\Delta)\}$满足公式$\rho(q_p,w_k)$.

3.2 FL的双向交换自动机的构造规则

(1) 原子公式

令$\varphi\in\{\bot,\top,p,\neg p\}$,其中,原子命题$p\in AP$,则构造$\varphi$的双向交换自动机$A=\langle\Sigma,Q,\rho,q_I,f\rangle$满足$Q=\{q_I,f\}$,$\rho(q_I,\varphi)=(0,f)$,表示$A$在状态$q_I$读入字符$\varphi$,将状态变成$f$,读头保持在原位不动.

(2) $\varphi=\neg\varphi_1$

设φ_1的双向交换自动机$A[\varphi_1]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,则$\varphi=\neg\varphi_1$的双向交换自动机$A[\neg\varphi_1]$即求$A[\varphi_1]$的补集:将$A[\varphi_1]$的接收状态和非接收状态交换,且$\rho_1$的定义中∧与∨对换[32].

(3) $\varphi=r_1\&\&r_2$

假设r_1双向交换自动机$A[r_1]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,$r_2$的双向交换自动机$A[r_2]=\langle\Sigma,Q_2,\rho_2,q_2,F_2\rangle$,则$r_1\&\&r_2$的双向交换自动机$A[r_1\&\&r_2]=\langle\Sigma,Q,\rho,q_I,F\rangle$满足$Q=Q_1\cup Q_2\cup\{q_I\}$,$F=F_1\cup F_2$,$\rho$的定义为

① $\rho(q_I,a)=(0,q_1)\wedge(0,q_2)$,其中,$a\in\Sigma$;

② 对$\forall q\in Q_1\backslash F_1,a\in\Sigma,\rho(q,a)=\rho_1(q,a)$;

③ 对$\forall q\in Q_2\backslash F_2,a\in\Sigma,\rho(q,a)=\rho_2(q,a)$.

"&&"是长度匹配合取连接符.为$r_1\&\&r_2$构造双向交换自动机时,必须严格要求每条路径长度一致.

(4) $\varphi=r_1|r_2$

假设r_1双向交换自动机$A[r_1]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,$r_2$的双向交换自动机$A[r_2]=\langle\Sigma,Q_2,\rho_2,q_2,F_2\rangle$,则$r_1|r_2$的双向交换自动机$A[r_1|r_2]=\langle\Sigma,Q,\rho,q_I,F\rangle$满足$Q=Q_1\cup Q_2\cup\{q_I\}$,$F=F_1\cup F_2$,$\rho$定义为

① $\rho(q_I,a)=(0,q_1)\vee(0,q_2)$,其中,$a\in\Sigma$;

② 对$\forall q\in Q_1\backslash F_1,a\in\Sigma,\rho(q,a)=\rho_1(q,a)$;

③ 对$\forall q\in Q_2\backslash F_2,a\in\Sigma,\rho(q,a)=\rho_2(q,a)$.

(5) $\varphi=\varphi_1$ abort b

假设φ_1的双向交换自动机$A[\varphi_1]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,则$\varphi_1$ abort b的双向交换自动机$A[\varphi_1$ abort $b]=\langle\Sigma,Q,\rho,q_1,F\rangle$满足$Q=Q_1\cup\{q_f\}$,$F=F_1\cup\{q_f\}$,$\rho$的定义为:对$\forall q\in Q$,

$$\rho(q,a)=\begin{cases}\rho_1(q,a), & \text{若}a\neq b\\ \rho_1(q,a)\vee(1,q_f), & \text{若}a=b\end{cases},$$

$$\rho(q_f,a)=(1,q_f),a\in\Sigma^*.$$

例 2:假设图 1(a)为φ_1双向交换自动机,则满足φ_1 abort b的双向交换自动机构造如图 1(b)所示.

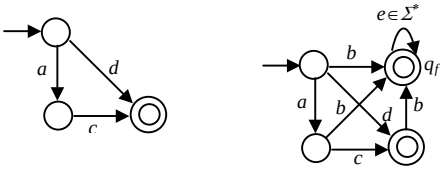
(6) $\varphi=\eta\cup\psi$

假设η的双向交换自动机$A[\eta]=\langle\Sigma,Q_1,\rho_1,q_\eta,F_1\rangle$,$\psi$的双向交换自动机$A[\psi]=\langle\Sigma,Q,\rho_2,q_\psi,F_2\rangle$,则$\varphi=\eta\cup\psi$的双向

交换自动机$A[\eta\cup\psi]=\langle\Sigma,Q,\rho,q_I,F\rangle$满足$Q=Q_1\cup Q_2\cup\{q_I\}$,$F=F_1\cup F_2$且$\rho$的定义为

① $\rho(q_I,\varepsilon)=(0,q_\psi)\vee((0,q_\eta)\wedge(1,q_I))$;

② 对$\forall q\in Q_1\backslash F_1,a\in\Sigma,\rho(q,a)=\rho_1(q,a)$;对$\forall q\in Q_2\backslash F_2,a\in\Sigma,\rho(q,a)=\rho_2(q,a)$.



(a) φ_1 (b) φ_1 abort b

Fig.1 Two-Way alternating automata

图 1 双向交换自动机

(7) $\varphi=X!\varphi_1$

假设φ_1的双向交换自动机$A[\varphi_1]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,则$\varphi=X!\varphi_1$的双向交换自动机$A[X!\varphi_1]=\langle\Sigma,Q,\rho,q_I,F\rangle$满足$Q=Q_1\cup\{q_I\}$,$F=F_1$,$\rho$的定义为:

① $\rho(q_I,a)=(1,q_1),a\in\Sigma$;

② 对$\forall q\in Q_1\backslash F_1,a\in\Sigma,\rho(q,a)=\rho_1(q,a)$.

(8) $\varphi=r\mapsto\varphi_1$

先求违反$r\mapsto\varphi_1$即$\neg r\mapsto\varphi_1$的双向交换自动机$A[\neg r\mapsto\varphi_1]$,即$\exists j\in\{0,\ldots,|w|\}$,s.t. $w^{0\cdots j}|\equiv r,w^{j\cdots}\not\models\neg\varphi_1$.假设$r$的双向交换自动机为$A[r]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,$\neg\varphi_1$的双向交换自动机为$A[\neg\varphi_1]=\langle\Sigma,Q_2,\rho_2,q_2,F_2\rangle$,则$A[\neg r\mapsto\varphi_1]=\langle\Sigma,Q,\rho,q,F\rangle$满足$Q=Q_1\cup Q_2$,$F=F_2$,$\rho$的定义为

① $\forall q_f\in F_1,\rho(q_f,\varepsilon)=(-1,q_2)$;

② 对$\forall q\in Q_1\backslash F_1,a\in\Sigma,\rho(q,a)=\rho_1(q,a)$;对$\forall q\in Q_2\backslash F_2,a\in\Sigma,\rho(q,a)=\rho_2(q,a)$.

最后,再求$A[\neg r\mapsto\varphi_1]$的补集即得$r\mapsto\varphi_1$的双向交换自动机.

下面考虑连接、交叠连接和闭包操作转化为双向交换自动机的构造方法.构造前,首先考虑一种错误情况,并给出避免错误的方法.

根据第 3.2 节的构造规则(3),图 2(a)不是正确的双向交换自动机构造方法,$A[r_1]$和$A[r_2]$之间存在长度不一致的路径p_1和p_2,$r_1\&\&r_2$不成立;但倘若再连接r_3即$(r_1\&\&r_2);r_3$,双向交换自动机构造如图 2(b)所示:将$A[r_1]$和$A[r_2]$的接收状态q_{f_1},q_{f_2}转变为非接收状态q'_{f_1},q'_{f_2},而后同时读入空字ε_i,将状态同时变为$A[r_3]$的初始状态q_3.这时,$A[r_1]$的路径p_1和$A[r_3]$的某路径长度之和有可能与$A[r_2]$的路径长度p_2和$A[r_3]$的某路径长度之和一致.为避免此种情况的发生,只要规定连接操作即读某空字ε_i前的所有节点的读头位置k一致即可,具体约束见下面的构造规则(9).

(9) $\varphi=r_1;r_2$

假设r_1的双向交换自动机$A[r_1]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,假设$r_2$的双向交换自动机$A[r_2]=\langle\Sigma,Q_2,\rho_2,q_2,F_2\rangle$,则$\varphi=r_1;r_2$的双向交换自动机$A[r_1;r_2]=\langle\Sigma,Q,\rho,q_1,F\rangle$满足约束:$\forall\varepsilon_i,\exists k$,若$x\in T,r(x)\in Q\times\{k\}$,有$w_k=a_k=\varepsilon_i$且$Q=Q_1\cup Q_2$,$F=F_2$,$\rho$的定义为:$\forall\ q_{f_i}\in F_1(i\geq1)$,其对应的节点$x_{q_{f_i}}$有$r(x_{q_{f_i}})=(q_{f_i},k_j)$且$\rho(q_{f_i},\varepsilon_j)=(0,q_2)$,则$\rho=\rho_1\cup\rho_2\cup\rho(q_{f_i},\varepsilon_j)$.

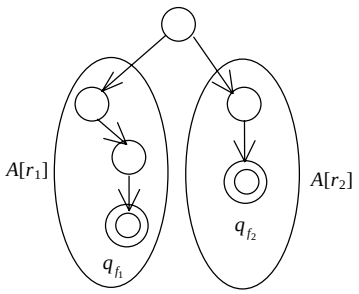
(10) $\varphi=r_1:r_2$

假设r_1的双向交换自动机$A[r_1]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,$r_2$的双向交换自动机$A[r_2]=\langle\Sigma,Q_2,\rho_2,q_2,F_2\rangle$,则$\varphi=r_1:r_2$的双向交换自动机$A[r_1:r_2]=\langle\Sigma,Q,\rho,q_1,F\rangle$满足约束同构造规则(9)且$Q=Q_1\cup Q_2$,$F=F_2$,$\rho$的定义为:$\forall\ q_{f_i}\in F_1(i\geq1)$,其对应的节点$x_{q_{f_i}}$必须满足$r(x_{q_{f_i}})=(q_i,k_j)$,且$\rho$定义为:$\exists l$,有$\rho(q,l)=(0,q_{f_i})$且$\rho(q_{f_i},\varepsilon_j)=(-1,q_2)$,则$\rho=\rho_1\cup\rho_2\cup\rho(q_{f_i},\varepsilon_j)$.

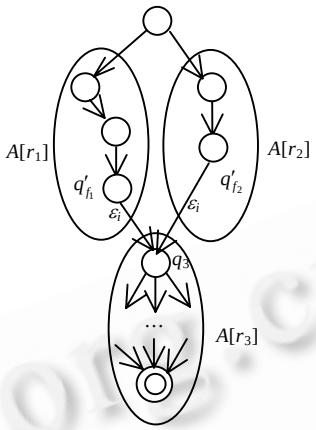
(11) $\varphi=r_1[*]$

假设r_1的双向交换自动机$A[r_1]=\langle\Sigma,Q_1,\rho_1,q_1,F_1\rangle$,则$\varphi=r_1[*]$的双向交换自动机$A[r_1[*]]=\langle\Sigma,Q,\rho,q_1,F\rangle$满足约束

同构造规则(9),且$Q=Q_1,F=F_1,\rho$的定义为:$\forall q_{f_i}\in F_1(i\geq 1)$,其对应的节点$x_{q_{f_i}}$有$r(x_{q_{f_i}})=(q_{f_i},k_j)$且$\rho(q_{f_i},\varepsilon_j)=(0,q_1)$,则$\rho=\rho_1\cup((0,q_1)\cup\rho)$.



(a) Invalid two-way alternating automata construction for $r_1\&\&r_2$
(a) 不成立的"$r_1\&\&r_2$"的构建



(b) Valid construction for $r_1\&\&r_2$ after adding "$;r_3$"
(b) 添加"$;r_3$"后$r_1\&\&r_2$可能转为成立

Fig.2
图 2

3.3 正确性证明

定理 1. φ是FL公式,$A[\varphi]$是依据第 3.2 节的规则构造的双向交换自动机,则对任意$w\in(\Sigma)^w$,$w\in L(A[\varphi])$当且仅当 $w\vDash\varphi$.

证明(用关于公式结构的归纳法):

- 当$\varphi=\neg\varphi_1$,$w\vDash\neg\varphi_1\Leftrightarrow\neg w\vDash\varphi_1\Leftrightarrow w\notin L(A[\varphi_1])$(归纳假设)$\Leftrightarrow w\in L(\overline{A\varphi_1})$,由此可知,求$\varphi=\neg\varphi_1$的双向交换自动机可转换成为求$\varphi_1$的双向交换自动机的补.
- 当$\varphi\in\{\bot,\top,p,\neg p\}$,其中原子命题 $p\in AP$,$w\in L(A[\varphi])\Leftrightarrow|w|=1,w=\varphi$(依据构造规则)$\Leftrightarrow w\vDash\varphi$(依据第 2.2 节的定义).
- 当$\varphi=r_1\&\&r_2$

假设r_1,r_2存在双向交换自动机$A[r_1]$和$A[r_2]$且满足性质:① 当$\rho(q_I,a)=(0,q_1)\wedge(0,q_2),\forall w_1,w_2,w_1\in L(A[r_1])\Leftrightarrow w_1\vdash r_1$且$w_2\in L(A[r_2])\Leftrightarrow w_2\vdash r_2$(归纳假设),且有$|w_1|=|w_2|$时,令$w\in\{w_1,w_2\}$,则$w\in L(A[r_1\&\&r_2])\Leftrightarrow w\vdash r_1$且$w\vdash r_2$成立(构造规则)$\Leftrightarrow w\vdash r_1\&\&r_2$(定义);② 当$\forall q\in Q_1\backslash F_1$时,有$L(A[r_1\&\&r_2])=L(A[r_1])$,则$\forall w_1,w_2,w_1\in L(A[r_1\&\&r_2])\Leftrightarrow w_1\in L(A[r_1])\Leftrightarrow w_1\vdash r_1,w_2\in L(A[r_1\&\&r_2])\Leftrightarrow w_2\in L(A[r_1])\Leftrightarrow w_2\vdash r_1$,且有$|w_1|=|w_2|$,令$w\in\{w_1,w_2\}$,则$w\vdash r_1$成立,又$\forall w$有$w\vdash r_2$成立(构造规则和归纳假设),所以$w\in L(A[r_1\&\&r_2])\Leftrightarrow w\vdash r_1$且$w\vdash r_2\Leftrightarrow w\vdash r_1\&\&r_2$(定义);③ 当$\forall q\in Q_2\backslash F_2$时,证明方法同当$\forall q\in Q_1\backslash F_1$时的情况.

- 当$\varphi=r_1|r_2$

假设r_1,r_2存在双向交换自动机$A[r_1]$和$A[r_2]$且满足性质.$w\in L(A[r_1])\Leftrightarrow w\vdash r_1$且$w\in L(A[r_2])\Leftrightarrow w\vdash r_2$(归纳假设),则$w\in L(A[r_1|r_2])\Leftrightarrow w\vdash r_1$或$w\vdash r_2$(构造规则)$\Leftrightarrow w\vdash r_1|r_2$(定义).

- 当$\varphi=\varphi_1$ abort b

假设φ_1存在双向交换自动机$A[\varphi_1]$且满足性质.$w\in L(A[\varphi_1\ \text{abort}\ b])\Leftrightarrow$当$a\neq b$时,$w\in L(A[\varphi_1])$或当$a=b$时,$\exists j<|w|,\exists w',|w|=|w'|,w^{0\cdots j-1}T^*=w'$,则 $w'\in L(A[\varphi_1])$ 且 $w^j\vdash b$(构造规则)$\Leftrightarrow w\vDash\varphi_1$ 或 $\exists j<|w|,w^{0\cdots j-1}\vDash\varphi_1$ 且 $w^j\vdash b$(归纳假设)$\Leftrightarrow w\vDash\varphi_1$ abort b(定义).

- 当$\varphi=\eta\cup\psi$

假设η与ψ存在双向交换自动机 $A[\eta]$与 $A[\psi]$且满足性质.$w\in L(A[\eta\cup\psi])\Leftrightarrow w\in L(A[\psi])$或$\exists k<|w|,\forall j<k$,$w^{k\cdots}\in L(A[\psi])$且$w^{j\cdots}\in L(A[\eta])$(根据构造规则)$\Leftrightarrow w\vDash\psi$或$\exists k<|w|,\forall j<k$,有$w^{k\cdots}\vDash\psi$且$w^{j\cdots}\vDash\eta$(归纳假设)$\Leftrightarrow w\vDash\eta\cup\psi$("$\cup$"

定义).

- 当$\varphi=X!\varphi_1$

假设φ_1存在双向交换自动机$A[\varphi_1]$且满足性质.$w\in L(A[X!\varphi_1])\Leftrightarrow w^{1\cdots}\in L(A[\varphi_1])$(依据构造规则)$\Leftrightarrow w^{1\cdots}\vDash\varphi_1$(归纳假设)$\Leftrightarrow w^{0\cdots}\vDash X!\varphi_1$即$w\vDash X!\varphi_1$(定义).

- 当$\varphi=\neg r\mapsto\varphi_1$

r与$\neg\varphi_1$存在双向交换自动机$A[r]$和$A[\neg\varphi_1]$且满足性质.$w\in L(A[\varphi])=L(A[\neg r\mapsto\varphi_1])\Leftrightarrow\exists j\le|w|,w^{0\cdots j}\in L(A[r])$,且$w^{j\cdots}\in L(A[\neg\varphi_1])$(构造规则,其中,字母$w^j$是$A[r]$的最后一个输入字符,读头回退一格,$w^j$又重新读一遍,是$A[\neg\varphi_1]$的第 1 个输入字符)$\Leftrightarrow\exists j\in\{0,\ldots,|w|\},w^{0\cdots j}\vDash r$,且$w^{j\cdots}\vDash\neg\varphi_1$(归纳假设)$\Leftrightarrow w\vDash\neg r\mapsto\varphi_1$(定义).

- 当$\varphi=r_1;r_2$

假设r_1,r_2存在双向交换自动机$A[r_1]$和$A[r_2]$且满足性质,$w\in L(A[r_1;r_2])\Leftrightarrow\exists w_1,w_2,w=w_1w_2=w_1\varepsilon w_2$,有$w_1\in L(A[r_1])$,且$\varepsilon w_2=w_2\in L(A[r_2])$(依据构造规则,$\rho=\rho_1\cup(\rho(q_{f_i},\varepsilon_j)\cup\rho_2)=\rho_1\cup((0,q_2)\cup\rho_2)$,且$Q=Q_1\cup Q_2)\Leftrightarrow\exists w_1,w_2$,$w=w_1w_2$,有$w_1\vdash r_1$,且$w_2\vdash r_2$(归纳假设)$\Leftrightarrow w\vdash r_1;r_2$(定义).

- 当$\varphi=r_1:r_2$

假设r_1,r_2存在双向交换自动机$A[r_1]$和$A[r_2]$且满足性质,$w\in L(A[r_1:r_2])\Leftrightarrow\exists w_1,w_2,l,w=w_1lw_2,w_1l\in L(A[r_1])$,且$lw_2\vdash L(A[r_2])$(构造规则,其中,字母$l$是$A[r_1]$的最后一个输入字符,读头回退一格,$l$又重新读一遍,是$A[r_2]$的第 1 个输入字符)$\Leftrightarrow\exists w_1,w_2,l$,s.t. $w=w_1lw_2$,有$w_1l\vdash r_1$且$lw_2\vdash r_2$(归纳假设)$\Leftrightarrow w\vdash r_1:r_2$(定义).

- 当$\varphi=r_1[*]$

假设r_1存在双向交换自动机$A[r_1]$且满足性质.$w\in L(A[r_1[*]])\Leftrightarrow w=\varepsilon$,或$\exists w_1\ne\varepsilon,\exists w_2,w=w_1w_2=w_1\varepsilon_jw_2$,有$w_1\in L(A[r_1])$,且$\varepsilon_jw_2\in L(A[r_1[*]])$(依据构造规则,$\rho=\rho_1\cup(\rho(q_{f_i},\varepsilon_j)\cup\rho))=\rho_1\cup((0,q_1)\cup\rho)$,且$Q=Q_1)\Leftrightarrow w\vdash[*0]$或$\exists w_1,w_2$,s.t. $w_1\ne\varepsilon,w=w_1w_2$,有$w_1\vdash r_1$且$w_2\vdash r_1[*]$(归纳假设)$\Leftrightarrow w\vdash r_1[*]$(定义). □

3.4 计算复杂度

FL 转化为双向交换自动机,时间复杂度为 $O(L)$,其中,L 是公式φ长度.

$$L=|\varphi|=\begin{cases}1, & \text{若}\varphi\in\{\bot,\top,p,\neg p\},p\in AP\\ |\varphi_1|, & \text{若}\varphi=\neg\varphi_1\\ 1+|\varphi_1|, & \text{若}\varphi=X!\varphi_1\text{或}\varphi=\varphi_1\ \text{abort}\ b\\ 1+|\varphi_1|+|\varphi_2|, & \text{若}\varphi=\varphi_1\cup\varphi_2\\ |r_1|, & \text{若}\varphi=r_1[*]\\ |r_1|+|r_2|, & \text{若}\varphi=(r_1\circ r_2),\circ\in\{;,:\}\\ 1+|r_1|+|r_2|, & \text{若}\varphi=(r_1\circ r_2),\circ\in\{|,\&\&\}\\ |r_1|+|\varphi_1|, & \text{若}\varphi=\neg r_1\mapsto\varphi\end{cases}.$$

3.5 双向交换自动机向非确定自动机的转化

定理 2. 令$A=\langle\Sigma,Q,q_0,\rho,F\rangle$是一个双向交换自动机,其中,$\rho:Q\times\Sigma\to B^+(\{-1,0,1\}\times Q)$.定义自动机$B=\langle\Sigma^n,(Q\times IN)^n,\{(q_0,0)\},\rho',F'\rangle$,$n$表示每个状态中的元素$(q,k)\in Q\times IN$的个数,$k\in IN$为读头位置,迁移函数$\rho':(Q\times IN)^n\times\Sigma^n\to\wedge^nB^+(Q\times IN)$,其中,$B^+(Q\times IN)$为集合$Q\times IN$上有限次应用“∧”与“∨”的布尔公式,$\wedge^nB^+(Q\times IN)$是将$n$个$B^+(Q\times IN)$进行合取运算,得到多个合取项的析取形式(即析取范式),每个合取项将作为一个新状态,F'满足:$F'\subseteq(Q\times IN)^n$,且$\exists q\in F$,$k\in IN$,有$(q,k)$为$F'$的子公式,则$B$是非确定自动机,且满足$L(A)=L(B)$.

先看一个实例.

例 3:已知某一双向交换自动机A初始状态为s_0,

$$\rho(s_0,w_0)=(1,s_1)\wedge(1,s_2)\vee(0,s_3),\rho(s_1,w_1)=(-1,s_3)\wedge(1,s_4)\vee(0,s_5),\rho(s_2,w_1)=(-1,s_6)\wedge(1,s_7)\vee(0,s_8),\ldots$$

将其转换成非确定自动机B,如图 3 所示.图中,A的初始状态s_0读入字符$w_0\in\Sigma^1$后将“∧(合取)”连接的子公式集合作为B的一个状态(整体),如图 3 中左状态“$(s_1,1)\wedge(s_2,1)$”,与右状态“$(s_3,0)$”的关系为“∨(析取)”.对于B的状态

“$(s_1,1)\wedge(s_2,1)$”,读入字符串w_1w_1后对“$(s_3,0)\wedge(s_4,2)\vee(s_5,1)$”和“$(s_6,1)\wedge(s_7,1)\vee(s_8,0)$”进行“$\wedge$”运算,有

$$((s_3,0)\wedge(s_4,2)\vee(s_5,1))\wedge((s_6,1)\wedge(s_7,1)\vee(s_8,0))=((s_5,1)\wedge(s_6,1)\wedge(s_7,1))\vee((s_5,1)\wedge(s_8,0))\vee((s_3,0)\wedge(s_4,2))\wedge(s_6,1)\wedge(s_7,1))\vee((s_3,0)\wedge(s_4,2)\wedge(s_8,0)).$$

这样得到多个合取项的析取形式(即析取范式).把每一合取项视为B的一个新状态(整体),有 4 个合取项,在图中表示成 4 个新状态.当一个状态中包含n个子公式$(q_1,k_1),(q_2,k_2),\ldots,(q_n,k_n)$,其中,$k_1\le k_2\le\ldots\le k_n$时,其后同时读入字符串 $w_{k_1}w_{k_2}...w_{k_n}\in\Sigma^n$.

证明:(定理 2)先证明$L(A)\subseteq L(B)$.设$w\in L(A)$.令 $w=(w_{k_0})(w_{k_{10}}w_{k_{11}}...w_{k_{1n_1}})(w_{k_{20}}w_{k_{21}}...w_{k_{2n_2}})(w_{k_{i0}}w_{k_{i1}}...w_{k_{in_i}})$,…,其中,$(w_{k_{i0}}w_{k_{i1}}...w_{k_{in_i}})$ 表示双向交换自动机A的一个运行的第i层($i\ge0$)的n_i个节点对应状态分别读入的字符,由于这些节点的对应状态之间是合取关系,因此 $(w_{k_{i0}}w_{k_{i1}}...w_{k_{in_i}})$ 必须同时读入.而B的构造是将第i层($i\ge0$)各元素的合取项视为一个状态(整体),每个状态必须同时读入合取项中各元素$(q,k)\in Q\times IN$ 对应的 A 中各状态 q 输入的字符.因此,A 中状态读入的字符串与 B 状态的字符串一致,所以 $w\in L(B)$.因此 $L(A)\subseteq L(B)$.

再证明$L(B)\subseteq L(A)$.设$w\in L(B)$.由于B迁移$\wedge^nB^+(Q\times IN)$是通过读入 $(w_{k_0}w_{k_1}...w_{k_{n-1}})$ 后求B中每个状态中具有合取关系的 n 个元素$(q,k)\in Q\times IN$ 对应的 A 中 n 个状态 q 的迁移的“与(合取)”运算,这样得到多个合取项的析取形式即析取范式(如例 3),每个合取项作为一个 B 的新状态.这样,B 的多个状态对应的输入字符串 $(w_{k_0}w_{k_1}...w_{k_{n-1}})$ 与 A 中状态的字符串相一致,所以 $w\in L(A)$.因此 $L(B)\subseteq L(A)$.

综上所述,$L(A)=L(B)$. □

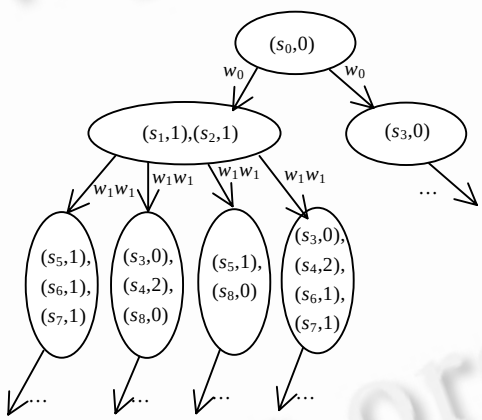


Fig.3 Nondeterministic automata for the Example 3

图 3 例 3 中转化后的非确定自动机

4 工具实现与实例

4.1 工具实现

根据第 3.2 节的构造规则及第 3.5 节的定理 2,给出 PSL 公式构造双向交换自动机及转化为非确定自动机的实现工具的算法框架如下.程序执行的硬件环境为 Intel P4 2.80GHz 处理器和 1.00GB 的内存,且工具是在 Linux 下,利用 C++编写的.

```
1   TwowayAlter_Automaton_t*   TwowayAlter_AutomatonInit(void);
2   if (delete) {
3   void   TwowayAlter_AutomatonFree(TwowayAlter_Automaton_t* aut);
4   goto 23;
    }
5   TwowayAlter_Automaton_t*   TwowayAlter_AutomatonGenerate(FL_Formula_t* flformula);
```

```
6   TwowayAlter_Automaton_t*
    DelAlternationfromTwowayAlter_Automaton(TwowayAlter_Automaton_t* aut);
7   if (deleteNode)
8       void   TwowayAlter_AutomatonFreeNode(FL_Vertex_t* node);
9   FL_Vertex_t*   TwowayAlter_AutomatonNewNode(TwowayAlter_Automaton_t* aut);
10  if (deleteEdge)
11      void   TwowayAlter_AutomatonFreeEdge(FL_Edge_t* edge);
12  FL_Edge_t*   TwowayAlter_AutomatonNewEdge(TwowayAlter_Automaton_t* aut,
                 FL_Vertex_t* source,array_t* dest,array_t* label);
13  FL_Edge_t*   TwowayAlter_AutomatonNewTrueEdgeLoop(TwowayAlter_Automaton_t* aut,
                 FL_Vertex_t* node);
14  void   TwowayAlter_AutomatonAddAbortEdge(TwowayAlter_Automaton_t* aut,FL_Vertex_t* node,
           FL_Vertex_t* abort_node);
15  void   TwowayAlter_AutomatonPruneStates(TwowayAlter_Automaton_t* aut);
16  array_t*
    TwowayAlter_AutomatonMergeLabelArray(array_t* label_src_array,array_t* label_dst_array);
17  array_t*   TwowayAlter_AutomatonMinimizeTermArray(array_t* term_array);
18  void TwowayAlter_AutomatonConcat(TwowayAlter_Automaton_t* aut_left,
        TwowayAlter_Automaton_t* aut_right, int typeOverlap);
19   NonDeterministic_Automaton_t*
     TwowayAlter2NonDeterministic_Automaton(TwowayAlter_Automaton_t* aut);
20  void printNode(FL_Vertex_t* node);
21  void printEdge(FL_Edge_t* edge);
22  void printLabelArray(array_t* array);
23  exit(0);
```

在上述算法框架中,步骤 1 是指初始化双向交换自动机,且置空.步骤 2~步骤 4 判断是否删除初始化的双向交换自动机,若是,则删除且转步骤 23.步骤 5 是指据第 3.2 节的规则,由 PSL 公式(包括 SERE 及 FL 公式)构造双向交换自动机.步骤 6 删除构造的双向交换自动机中的交换性(alternation).步骤 7~步骤 9 是指是否要删除节点,若是,则删除一个节点,否则新建一个节点.步骤 10~步骤 12 是指是否要删除一边,若是,则删除一边,否则新建一边.步骤 13 添加双向交换自动机中节点的 TRUE 循环(TRUE-loop).步骤 14 在双向交换自动机的节点上添加 aborting 边,即读入标记及某节点的后继,建立公式的否定形式的新边.例如,某个具有标记“*a*&*b*”的节点将会产生两条到 abort 状态的边,一条标记为“!*a*”,另一条为“!*b*”.而一个具有两条边、标记分别为“*a*”和“*b*”的节点将会得到一个标记为“!*a*&!*b*”的节点.步骤 15 为删除已得到的双向交换自动机的状态数,有 5 种方法.方法 1:优化边上定义的标志(排序、排除重复标志、排除不必要的 true、删除相互矛盾的标志),删除带 FALSE 的边.方法 2:删除在非确定的状态上自循环的交互的边.方法 3:删除不能迁移到接收状态的所有状态.方法 4:删除初始状态不可到达的状态.方法 5:简化某个状态的所有边集(删除冗余边).步骤 16 合并从源矩阵到目标矩阵的标记,即产生从源到目标的无重复的标记,同时筛选带有 TRUE 的标记,若一标记为 FALSE,则返回一个带标记为 FALSE 的矩阵.步骤 17 删除从确定范式 DNF 中非必要的项(terms),即删除所有被其他项覆盖的项,其中一个项是指指向顶点结构的指针矩阵.例如公式(*q*_1 && *q*_6)||(*q*_6),其首项(*q*_1 && *q*_6)是不必要的,应被删除.步骤 18 完成两个双向交换自动机连接,分 3 种情况(由 typeOverlap 表示):① 非交叠,可将右自动机直接连接在左自动机上;② 交叠,右双向交换自动机从初始状态引出的出边与左自动机的接收状态入边进行 AND 连接;③ 具有交换性的交叠,与情况②类似,但左双向交换自动机的目标节点没有被替换,而是与右自动机的目标节点相联合.步骤 19 根据第 3.5 节的定理 2,将由上述步骤得到的双向交换自动机转化为非确定自动机.步骤 20~步骤 22 显示(打

印)输出(包括状态、边、边上的标记).步骤 23 表示结束.

4.2 实例——队列控制电路

图 4 描述了队列控制电路图,包括队列和相应的控制芯片.队列具有两个指针:qFirst 指示第 1 数据单元位置,qLast 指示最后数据单元位置.控制芯片 cntrl 包含两个输入信号 qInsert(插入数据)、qRemove(取数据)以及 3 个输出信号:qFull(队列已满)、qEmpty(队列空)和 qError(队列上溢或下溢).

对于队列有两种情况必须避免:(1) 队列已满,仍执行插入数据操作,称为上溢;(2) 队列已空,仍执行取数操作,称为下溢.

图 5 给出队列控制电路部分代码及各信号指派.

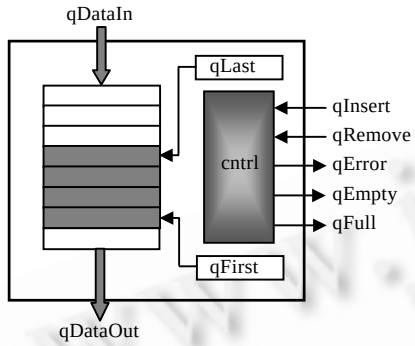


Fig.4 Diagram of a queue controlling circuit

图 4 队列控制电路图

```
function [3:0] qNext
        input [3:0] p;
        qNext=((p+1) mod qSize);
end function
assign   qFull=(qNext(qLast)==qFirst);
assign   qEmpty=(qLast==qFirst);
assign   qInsert=(qNext==qNext+1);
assign   qRemove=(qNext==qNext−1);
assign   qError=(qFull && qInsert)|(qEmpty && qRemove)
```

Fig.5 Some codes of a queue controlling circuit and its signals' assignments

图 5 队列控制电路部分代码及各信号指派

分析以上代码,得到电路的两则用 PSL 表达的属性:

1) (qFull && qInsert→X!(¬qEmpty)) abort (¬rstN);2) (qEmpty && qRemove→X!(¬qFull)) abort (¬rstN).

rstN 是表征电路重初始化操作符号.语句 1)表示不执行初始化操作时,性质"队列满并且插入数据,则下一步队列非空"不要求必须成立;语句 2)表示在不执行初始化操作时,性质"队列空且取数据,则下一步队列非满"不要求必须成立.

(qFull && qInsert→X!(¬qEmpty)) abort (¬rstN)=(¬(qFull && qInsert)|X!(¬qEmpty)) abort (¬rstN),

(qEmpty && qRemove→X!(¬qFull)) abort (¬rstN)=(¬(qEmpty && qRemove)|X!(¬qFull)) abort (¬rstN).

分别应用第 3.2 节中的规则(2)~规则(5)、规则(7)可构造上面属性 1)和属性 2)对应的双向交换自动机,再根据定理 2 转化为非确定自动机.

利用第 4.1 节给出的工具,得到属性 1)和属性 2)的双向交换自动机及非确定自动机.以属性 1)为例,最终转化为非确定自动机的实验结果为

状态	入边数	出边数
q_0:	IN 0	OUT 1
q_1:	IN 0	OUT 1
q_2:	IN 0	OUT 2
q_3:	IN 1	OUT 2
q_4:	IN 4	OUT 1
q_5:	IN 3	OUT 1

边(edge)及标记(label):	
q_2→q_3	Label: TRUE
q_3→q_4	Label: !'qEmpty'
q_4→q_4	Label: TRUE
q_1→q_4	Label: !'qInsert'
q_0→q_4	Label: !'qFull'
q_3→q_5	Label: 'qEmpty' & !'rstN'
q_2→q_5	Label: !('qFull' & 'qInsert')
q_5→q_5	Label: TRUE

初始状态:{q_0,q_1,q_2}
接收状态:{q_2,q_4,q_5}
状态数:6
符号数:3
边数:8
初始状态数:{1,1,1}
接收状态数:3

5 结束语

自动机是连接系统正确性推理与验证形式逻辑规约可满足性以及系统自动验证的决策过程设计之间的重

要纽带(即通常所指的模型检验的任务之一)[33].本文在文献[1]中定义的PSL及FL的基础上,给出了针对FL各种操作算子构造双向交换自动机的构造规则,其计算复杂度仅是FL公式长度的线性表达式,并验证了构造规则的正确性.在此基础上,又给出将双向交换自动机转化为等价的非确定自动机的转化方法,证明两类自动机接受语言相同,并给出了转化的具体实例.最后,编写了将PSL转化为上述自动机的实现工具,并在一个队列控制电路具体实例的背景下,给出其实验结果.以上步骤为下一步判空作准备,以执行FL的模型检验.这对解决复杂并行系统建模和模型验证问题具有重要的理论意义和应用价值.有待进一步研究的问题是:将双向交换自动机转化为非确定自动机时可考虑使用一些优化技术,如使用文献[34]中的PSL语法优化技术(syntactic optimization)等,以进一步提高转化效率.

致谢 本文在撰写过程中与刘万伟和赵常智同学进行过深入讨论,在此表示感谢.

References:

[1] Accellera Organization, Inc. Formal syntax and semantics of Accellera property specification language. Appendix B. 2004. 109−119. http://www.eda.org/vfv/docs/PSL-v1.1.pdf

[2] Accellea. Property specification language reference manual (version1.0.1). 2003. http://www.haifa.il.ibm.com/projects/verification/sugar

[3] Beer I, Ben-David S, Eisner C, Fisman D, Gringauze A, Rodeh Y. The temporal logic sugar. In: Gerard B, Hubert C, Alain F, eds. Proc. of the 13th Int'l Conf. on Computer Aided Verification (CAV). LNCS 2102, Heidelberg: Springer-Verlag, 2001. 363−367.

[4] Accellea. Property specification language reference manual (version1.1). 2004. http://www.eda.org/vfv/docs/PSL-v1.1.pdf

[5] Emerson EA, Clarke EM. Using branching-time temporal logic to synthesize synchronization skeletons. Science of Computer Programming, 1982,2(3):241−266.

[6] Büchi JR. On a decision method in restricted second order arithmetic. In: Suppes P, Tarski A, eds. Proc. of the '60 Int'l Congress on Logic, Methodology and Philosophy of Science. Stanford: Stanford University Press, 1962. 1−11.

[7] Rabin MO. Decidability of second-order theories and automata on infinite trees. Trans. of the American Mathematical Society, 1968,74(5):1025−1029.

[8] Streett RE. Propositional dynamic logic of looping and converse is elementarily decidable. Information and Control, 1982,54(1/2): 121−141.

[9] Vardi MY, Wolper P. Reasoning about infinite computations. Information and Computation, 1994,115(1):1−37.

[10] Wolper P, Vardi MY, Sistala AP. Reasoning about infinite computation paths. In: Proc. of the 24th Annual Symp. on Foundations of Computer Science (FOCS'83). Washington: IEEE Computer Society Press, 1983. 185−194. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4568076

[11] Emerson EA, Sistala AP. Deciding branching time logic. In: Proc. of the 16th Annual ACM Symp. on Theory of Computing. New York: ACM, 1984. 14−24. http://portal.acm.org/citation.cfm?id=808661

[12] Vardi MY, Wolper P. An automata-theoretic approach to automatic program verification. In: Proc. of the Symp. on Logic in Computer Science (LICS'86). Cambridge: IEEE Computer Society Press, 1986. 332−344. http://spinroot.com/spin/Doc/lics86.pdf

[13] Sebastiani R, Tonetta S. "More deterministic" vs. "Smaller" Büchi automata for efficient LTL model checking. In: Daniel G, Enrico T, eds. Proc. of the 12th Advanced Research Working Conf. on Correct Hardware Design and Verification Methods (CHARME 2003). LNCS 2860, Heidelberg: Springer-Verlag, 2003. 126−140.

[14] Schneider K. Improving automata generation for linear temporal logic by considering the automaton hierarchy. In: Robert N, Andrei V, eds. Proc. of the 8th Int'l Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2001). LNCS 2250, Heidelberg: Springer-Verlag, 2001. 39−54.

[15] Muller DE, Saoudi A, Schupp PE. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In: Proc. of the 3rd Annual Symp. on Logic in Computer Science (LICS'88). Edinburgh: IEEE Computer Society Press, 1988. 422−427. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5139

[16] Vardi MY. Nontraditional applications of automata theory. In: Masami H, Mitchell John C, eds. Proc. of the TACS'94. LNCS 789, Heidelberg: Springer-Verlag, 1994. 575−597.

[17] Isli A. Mapping an LPTL formula into a Büchi alternating automaton accepting its models. Research Report, MPI-I-94-230, Max-Planck-Institut für Informatik, 1994. 85−90.

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

[19] Muller DE, Saoudi A, Schupp PE. Alternating automata, the weak monadic theory of trees and its complexity. Theoretical Computer Science, 1992,97(2):233−244.

[20] Gastin P, Oddoux D. Fast LTL to Büchi automata translation. In: Gerard B, Hubert C, Alain F, eds. Proc. of the 13th Int'l Conf. on Computer Aided Verification (CAV 2001). LNCS 2102, Heidelberg: Springer-Verlag, 2001. 53−65.

[21] Etessami K. A hierarchy of polynomial-time computable simulations for automata. In: Brim L, Jancar P, Kretinsky M, Kucera A, eds. Proc. of the COMUR 2002. LNCS 2421, Heidelberg: Springer-Verlag, 2002. 131−144.

[22] Fritz C. Constructing Büchi automata from linear temporal logic using simulation relations for alternating Büchi automata. In: Ibarra OH, Dang Z, eds. Proc. of the 8th Int'l Conf. on Implementation and Application of Automata (CIAA 2003). LNCS 2759, Heidelberg: Springer-Verlag, 2003. 35−48.

[23] Fritz C, Wilke T. State space reductions for alternating Büchi automata: Quotienting by simulation equivalences. In: Manindra A, Anil S, eds. Proc. of the 22nd Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2002). LNCS 2556, Heidelberg: Springer-Verlag, 2002. 157−168.

[24] Emerson EA, Jutla CS. Tree automata, mu-calculus and determinacy. In: Proc. of the 32nd Annual Symp. on Foundations of Computer Science (FOCS'99). Washington: IEEE Computer Society Press, 1999. 368−377. http://portal.acm.org/citation.cfm?id=123229.123332

[25] Bustan D, Fisman D, Havlicek J. Automata construction for PSL. Technical Report, MCS05-04, Rehovot: Computer Science and Applied Mathematics, The Weizmann Institute of Science, 2005. http://www.wisdom.weizmann.ac.il/~dana/publicat/automata_constructionTR.pdf

[26] Ben-David S, Bloem R, Fisman D, Griesmayer A, Pill I, Ruah S. Automata construction algorithms optimized for PSL. Technical Report, Delivery 3.2/4, PROSYD, 2005.

[27] Heljanko K, Junttila T, Keinänen M, Lange M, Latvala T. Bounded model checking for weak alternating Büchi automata. In: Thomas B, Jones RB, eds. Proc. of the 18th Int'l Conf. on Computer Aided Verification (CAV 2006). LNCS 4144, Heidelberg: Springer-Verlag, 2006. 95−108.

[28] Bloem R, Cimatti A, Pill I, Roveri M, Semprini S. Symbolic implementation of alternating automata. In: Ibarra OH, Yen H-C, eds. Proc. of the 11th Int'l Conf. on Implementation and Application of Automata (CIAA 2006). LNCS 4094, Heidelberg: Springer-Verlag, 2006. 208−218.

[29] Cimatti A, Roveri M, Semprini S, Tonetta S. From PSL to NBA: A modular symbolic encoding. In: Proc. of the Formal Methods in Computer Aided Design (FMCAD 2006). Washington: IEEE Computer Society Press, 2006. 125−133. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4021018

[30] Ruah S, Fisman D, Ben-David S. Automata construction for on-the-fly model checking PSL safety simple subset. Technical Report, H-0234, Haifa: IBM Haifa Research Lab., 2005.

[31] Kupferman O, Piterman N, Vardi MY. Extended temporal logic revisited. In: Larsen KG, Mogens N, eds. Proc. of the 12th Int'l Conf. on Concurrency Theory (CONCUR 2001). LNCS 2154, Heidelberg: Springer-Verlag, 2001. 519−535.

[32] Muller D, Schupp P. Alternating automata on infinite trees. Theoretical Computer Science, 1987,54(2-3):267−276.

[33] Yi J, Zhang WH. Efficient translation from transition-based generalized Büchi automata to Büchi automata. Journal of Software, 2006,17(4):720−728 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/720.htm

[34] Cimatti A, Roveri M, Tonetta S. Syntactic optimizations for PSL verification. In: Orna G, Michael H, eds. Proc. of the TACAS 2007. LNCS 4424, Heidelberg: Springer-Verlag, 2007. 505−518.

附中文参考文献:

[33] 易锦,张文辉.从基于迁移的扩展 Büchi 自动机到 Büchi 自动机.软件学报,2006,17(4):720−728. http://www.jos.org.cn/1000-9825/17/720.htm

虞蕾(1978－),女,浙江浦江人,博士,讲师,主要研究领域为模型检验,航迹规划.

陈火旺(1936－2008),男,教授,博士生导师,中国工程院院士,CCF 高级会员,主要研究领域为形式化方法,人工智能技术.