以 DNA 为载体的线性时序逻辑模型检测

朱维军,周清雷,李永亮
(郑州大学信息工程学院,河南郑州 450001)

摘 要: 线性时序逻辑模型检测被广泛应用于处理器设计与验证、网络协议验证、安全协议验证等领域.然而到目前为止,该技术只能在电子计算的平台上实现.为了以脱氧核糖核酸(Deoxyribo Nucleic Acid,DNA)为载体对线性时序逻辑(Linear Temporal Logic,LTL)实施模型检测,给出了使用粘贴自动机实现 Until 算子模型检测的方法.首先,使用粘贴自动机对 Until 公式的有穷状态自动机(Finite State Automata,FSA)模型进行编码;然后,将系统模型转换为粘贴自动机的输入字符串;最后,用粘贴自动机验证系统是否满足公式.仿真实验结果证实,新方法可实现对 LTL 逻辑时序算子的检测.

关键词: 模型检测; 脱氧核糖核酸; 线性时序逻辑; 粘贴自动机
中图分类号: TP301; TP384 **文献标识码:** A **文章编号:** 0372-2112 (2016)06-1265-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.06.001

Conduct Linear Temporal Logic Model Checking via DNA Molecules

ZHU Wei-jun,ZHOU Qing-lei,LI Yong-liang
(*School of Information Engineering,Zhengzhou University,Zhengzhou,Henan* 450001,*China*)

Abstract: The linear temporal logic (LTL) model checking is widely used in processor design and verification,network protocol verification and security protocol verification. Up to now,this technique can only be realized on the platform of electronic computer. In order to conduct LTL model checking under the circumstance of deoxyribo nucleic acid (DNA), we proposed a method to check the Until constructor via a sticker automaton. We encode a finite state automaton (FSA) which is a model of the formula Until,with a DNA sticker automaton. And we convert a model of a system into its paths,as the input strings of the sticker automaton. We verify whether the system satisfies the formula or not,by using the sticker automaton. In this way,the formula Until can be checked with the double strand DNA molecules. The simulation results show that the method can successfully check the basic temporal formulas.

Key words: model checking;deoxyribo nucleic acid;linear temporal logic;sticker automaton

1 引言

模型检测由图灵奖得主 Clarke 教授等人提出[1].它是一种能够自动验证系统是否满足给定性质的形式化核心方法与技术,被广泛地应用于硬件验证[2]、网络协议验证、安全协议验证[3]等领域.在时序逻辑模型检测中,线性时序逻辑[4](Linear Temporal Logic,LTL)、计算树逻辑[5](Computation Tree Logic,CTL)分别被用来描述系统需满足的线性时序性质和分支时序性质,有穷状态自动机(Finite State Automata,FSA)则被用来为系统建模,模型检测算法自动检测模型是否满足需求性质.

脱氧核糖核酸(Deoxyribo Nucleic Acid,DNA)计算以 DNA 分子为载体,以生物酶和生物操作作为信息处理工具,是一种与电子计算不同的计算模型.1994 年,图灵奖得主 Adleman 教授发表在《Science》上的一篇文章用 DNA 实验求解了一个小规模的哈密顿路径问题[6],被认为是 DNA 计算领域的开创性工作.由于 DNA 分子在实验操作上相对容易,分子机构上便于信息处理,DNA 计算得到快速发展.学者们提出了诸如过滤模型[7]、粘贴自动机[8]等等 DNA 计算模型,并用这些模型成功解决了一系列 NP 难题:中国邮递员问题和 0-

收稿日期:2014-11-15;修回日期:2015-01-27;责任编辑:覃怀银
基金项目:国家自然科学基金(No. 61250007, No. U1204608, No. U1304606, No. 61572444);中国博士后科学基金(No. 2012M511588, No. 2015M572120);河南省高等学校青年骨干教师资助计划项目(No. 2014GGJS-001)

1 规划问题的 DNA 算法[9,10]、图的最大团问题和最小覆盖问题的 DNA 算法[11,12]、图着色问题的 DNA 算法[13]. 特别是,许进教授等人用并行 DNA 计算模型解决了 61 个顶点的 3 着色问题[14]. 最新的一系列研究[15~17]更是涉及到了纳米器件与 DNA 自组装.

在经典计算中的模型检测,状态空间与效率始终是困扰研究人员的难题. DNA 计算具有强大的并行处理能力,可为模型检测效率提升提供新的思路. 2006 年,图灵奖得主 Emerson 教授提出了一种用 DNA 分子进行 CTL 模型检测的思路与构想,给出了 CTL 公式 EFp 的 DNA 模型检测方法[18]. 这是使用 DNA 计算实现模型检测的一个尝试. 然而,该方法不能对一般 CTL 公式实施检测,更不能对 LTL 公式进行检测. 为此,我们研究以 DNA 分子为载体的 LTL 模型检测方法.

2 线性时序逻辑与 FSA

2.1 线性时序逻辑

LTL 逻辑的一阶部分不可判定,因此我们只需研究其命题部分(Propositional LTL,PLTL).

定义 1 设 AP 是原子命题集合,如果 φ 和 ψ 是 PLTL 公式,那么,$p \in$ AP、$\neg\varphi$、$\varphi \vee \psi$、$\varphi \rhd \psi$ 都是 PLTL 公式.

定义 2 一个 PLTL 模型是一个三元组 $M=(S,R,L)$,其中:

(1)S 是非空有穷状态集,$s \in S$ 是状态;

(2)$R: S \to S$,$R(s)$ 是状态 s 的唯一后继状态;

(3)$L: S \to 2^{AP}$,$L(s)$ 是在状态 s 中成立的原子命题的集合.

定义 3 满足关系 $\models$ 定义如下:

(1)$s \models p$ 当且仅当 $p \in L(s)$;

(2)$s \models \neg\varphi$ 当且仅当 $\neg(s \models \varphi)$;

(3)$s \models \varphi \vee \psi$ 当且仅当 $(s \models \varphi) \vee (s \models \psi)$;

(4)$s \models \varphi \rhd \psi$ 当且仅当 $\exists j \geq 0.\ R_j(s) \models \psi \wedge (\forall 0 \leqslant k < j.\ R_k(s) \models \varphi)$,其中,$R_0(s)=s$,$R_{n+1}(s)=R(R_n(s))$.

2.2 FSA

定义 4 一个 FSA A 是一个五元组(Σ, Q, T, q_0, F),其中:

(1) Σ 是一个有穷非空字母表;

(2) Q 是一个有穷非空状态集;

(3) 转移函数 $T: Q \times \Sigma \to R(Q)$;

(4) $q_0 \in Q$ 是起始状态;

(5)$F \subseteq Q$ 是接受状态集.

定义 5 设 $w=y_1,\cdots,y_m$ 是定义在 Σ 上的字符串,如果存在 Q 中状态序列 $r_0, r_1, \cdots, r_m$ 满足:

(1)$r_0 = q_0$

(2)$r_{i+1} \in T(r_i, y_{i+1})$,$i=0,1,\cdots,m-1$

(3)$r_m \in F$

则称 w 是被 A 接受的文字.

定义 6 A 识别的语言是被 A 接受的文字的集合.

3 粘贴自动机

粘贴自动机[8]是一种实现 FSA 的 DNA 计算模型,给定输入字符串的 DNA 链,粘贴自动机模型能够判断字符串是否被自动机接受.

3.1 FSA 和输入字符串的编码方式

假设 $M=(\Sigma,S,T,s_0,F)$ 是一个 FSA,字母表 Σ 中的任意一个字符 a 都可以用单链 DNA 编码为 $C(a)$.

(1)Σ 上的输入字符串 $w=a_1,\cdots,a_n$ 用一个 DNA 单链编码:$5'\ I_1\ X_0\cdots X_m\ C(a_1)\cdots X_0\cdots X_m\ C(a_n)\ X_0\cdots X_m\ I_2\ 3'$,其中,$I_1$ 是启动区序列,$X_0\cdots X_m$ 是间隔序列,用来间隔字符 a_i 的编码 $C(a_i)$,I_2 是终止区序列.

(2) 状态转移函数 $T(s_i,a)=s_j$ 用下面的 DNA 编码:$3'\ \overline{X_{i+1}\cdots X_m}\,\overline{C(a)}\,\overline{X_0\ldots X_j}\ 5'$,其中 $\bar{X}$ 表示核苷酸 X 的 Watson-Crick 补链,$\overline{C(a)}$ 表示 a 的 DNA 序列的 Watson-Crick 补链.

(3)每个初态 s_i 的编码:$3'\ \overline{I_1}\ \overline{X_0\cdots X_i}\ 5'$

(4)每个终态 s_j 的编码:$3'\ \overline{X_{j+1}\cdots X_m}\ \overline{I_2}\ 5'$

3.2 计算过程

粘贴自动机模型的计算过程分为 3 步:

第一步:数据预处理

(1) 合成输入字符串和自动机的编码链.

(2) 将所有的 DNA 链放到试管 T 中,使互补链充分结合.

(3) 在试管中 T 加入 DNA 连接酶,得到(部分)双链 DNA 分子.

第二步:计算

经过第一步的数据处理,如果输入字符串被自动机接受,那么试管 T 中应该是以启动区开始以终止区结束的完全双链的 DNA 分子;否则,试管 T 中的 DNA 不是完全双链,要么终止区是部分双链,要么启动区和终止区中的某部分是部分双链的. 根据上述情况,可以向试管 T 中加入 Mung Bean 核酶,降解其中的单链 DNA 片段,保留完全双链的 DNA 分子.

第三步:输出结果

向试管 T 中加入 Mung Bean 核酶后,可以利用电泳技术分离不同长度的 DNA 分子,如果存在两种或者两种以上的不同的 DNA 分子质量,说明在加入核酶前,试管 T 中存在部分双链的 DNA 分子,输入字符串不被自动机接受;否则,在加入核酶前,T 中全部是完全双链 DNA 分子,输入字符串是被自动机接受的.

4　用粘贴自动机实现 PLTL 模型检测

4.1　将 PLTL 公式转换为 FSA

PLTL 的基本核心公式如下:

$$p \triangleright q \tag{1}$$

图 1 是式(1)的 FSA 模型. 按照粘贴自动机模型中 FSA 的编码方式编码图 1 中的自动机,将编码好的 DNA 单链放到试管 R 中.

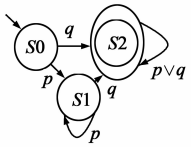

图1　式(1)对应的FSA

4.2　生成系统模型的符合粘贴自动机的输入字符串格式的运行

首先,我们用图 2 所示的单链 DNA 对系统的状态进行编码. 其中:$X_0 \cdots X_m$ 是粘贴自动机中的间隔序列,为满足粘贴自动机输入字符串的格式而增加;$L(s)$ 是状态标签函数;Y 用来调节编码,使每个状态编码的长度为相等的偶数,并且保证状态和状态编码之间一一对应.

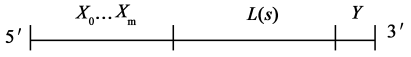

图2　状态的DNA编码

在状态编码的基础上,我们可以使用算法 1 生成系统自动机模型中以开始状态为始点以终止状态为终点的所有路径. 算法描述如下.

算法 1　系统模型自动机的运行路径生成算法

Input:公式(1)的自动机模型 M

Output:表达 M 所有路径的所有 DNA 分子

BEGIN

第 1 步:对于每个状态 s,向试管 T 中加入 s 的编码;对于每条边 e,向试管 T 中加入 e 的编码. 经过退火和连接反应,生成系统的 FSA 模型中的随机路径.

第 2 步:对试管 T 执行提取前缀操作,使试管 T 中只含有以开始状态为起点的路径.

第 3 步:对试管 T 执行提取后缀操作,使试管 T 中只含有以终止状态为终点的路径.

第 4 步:在试管 T 中加入 DNA 分子 $\frac{I_1}{I_1 H_0}$ 和 $\frac{X_0 \cdots X_m I_2}{B X_0 \cdots X_m I_2}$,其中 I_1 是启动区,H_0 是初始状态编码的前半段,B 是终止状态编码的后半段,$X_0 \cdots X_m$ 是间隔序列,I_2 是终止区. 经过退火和连接反应,在单链 DNA 分子 5′端添加启动区,并且在单链 DNA 分子 3′端添加间隔区和终止区.

第 5 步:通过亲和纯化技术,过滤出试管 T 中所有包含 I_1 和 I_2 序列的单链 DNA 分子.

END

其中,第 5 步除去第 4 步中多余的 DNA 分子,得到满足粘贴自动机输入字符串格式要求的单链 DNA 分子,每条单链 DNA 对应系统的一个运行.

4.3　用粘贴自动机验证系统是否满足 PLTL 公式

4.1 节获得表征式(1)模型的单链 DNA 分子;4.2 节的算法则得到了表征系统模型的单链 DNA 分子. 在此基础上,我们通过让上述两类单链 DNA 分子充分地进行生化反应,即可检测系统是否满足式(1). 检测方法如算法 2 所示. 需要补充说明的是,我们可以证明:要验证一个系统是否满足式(1),只需验证系统的长度小于 $|V| * 2^{|V|-1} + |E|$ 的所有运行是否满足式(1). 篇幅所限,不再给出详细证明过程.

算法 2　公式(1)的 DNA 模型检测算法

Input:4.1 节的试管 R,即式(1)对应的粘贴自动机的编码;算法 1 获得的试管 T,即系统的所有运行

Output:系统模型是否满足公式

BEGIN

第 1 步:用长度分离操作提出试管 T 中长度小于 $|V| * 2^{|V|-1} + |E|$ 的单链 DNA.

第 2 步:合并试管 R 到 T,在 T 中加入 DNA 连接酶,使互补链充分反应.

第 3 步:充分反应后,在试管 T 中加入 Mung Bean 核酶,降解所有单链 DNA.

第 4 步:对 T 执行分离操作,得到试管 T_1 和 T_2,T_1 中是含有启动区的 DNA 链,T_2 中则是不含的.

第 5 步:检测 T_2. 若 T_2 中有 DNA 链,则它不含启动区,它是第 2 步中的部分双链 DNA 分子被第 3 步中的核酶切开后的产物,第 2 步含有部分双链 DNA 分子,系统有不满足式(1)的运行. 否则,执行下一步.

第 6 步:对 T_1 执行分离操作,得到试管 R_1 和 R_2,R_1 中是含有终止区的 DNA 链,R_2 中则是不含的.

第 7 步:检测 R_2,如果 R_2 中有 DNA 链,那么它只含有启动区而不含有终止区,它是第 2 步中的部分双链 DNA 分子被第 3 步中的 Mung Bean 核酶切开后的产物. 第 2 步含有部分双链 DNA 分子,说明系统有不满足式(1)的运行. 否则,系统的所有运行都被粘贴自动机接受,系统满足式(1).

END

式(1)对应的自动机接受的运行会产生完全双链 DNA 分子,它既含有启动区又含有终止区,不会被 Mung Bean 核酶切开;而不被自动机接受的运行会产生部分双链 DNA 分子,将被 Mung Bean 核酶切开,切后的产物要么只含有启动区,要么只含有终止区. 算法 2 就是根据这个事实检测的.

4.4　复杂度分析

算法 2 有两个输入:系统模型的符合粘贴自动机输入字符串格式的运行;式(1)对应的粘贴自动机模型.

假设系统模型有 x 个状态、y 条边,算法 1 生成系统

模型的符合粘贴自动机输入字符串格式的运行. 第 1 步要合成所有状态和边的编码,此步共需执行 $x+y$ 个操作步骤;第 2、3、5 步,每步均需执行 1 个操作步骤;第 4 步需执行2 个操作步骤,(分别为加入两个 DNA 分子). 因此,生成系统模型的符合粘贴自动机输入字符串格式的运行的时间复杂度是 $O(x+y)+O(3)+O(2)=O(x+y)$.

假设有限状态自动机 M 有 m 个状态、n 条边,在建立其对应的粘贴自动机模型的过程中,合成所有状态和边的编码共需 $m+n$ 个操作步骤,而式(1)对应的自动机中,$m=3$、$n=6$,因此,建立式(1)对应的粘贴自动机模型共需执行 9 个操作步骤.

最后,在用粘贴自动机模型验证系统运行是否满足式(1)的算法 2 中,这 7 步中的每一步都需要执行 1 个操作步骤. 综上所述,算法 2 的时间复杂度是 $O(x+y)+O(9)+O(7)=O(x+y)$.

表 1 给出了新算法与相关方法的复杂度比较.

表 1 时序逻辑模型检测的 DNA 方法的检测效率之比较

	检测 LTL 的 DNA 算法(新算法)	检测 CTL 的 DNA 方法[18]	检测 LTL 的经典方法[1]
复杂度	$O(x+y)$个生物操作元步骤	$O(x+y)$个生物操作元步骤	PSPACE 个电子元步骤

5 仿真实验

我们在 Win7 操作系统上,用 VB 语言编写实验程序以实现算法. 新方法涉及的所有分子生物元操作,均已在现有的 DNA 计算相关工作中被使用,这些元操作的有效性也已被前人工作中的真实分子生物实验所证实. 因此,本文采用仿真实验代替真实生物实验,获得的结果是可信的.

表 2 和表 3 给出了实验的编码方式. 在此基础上,我们实施了两个仿真实验.

表 2 输入字符串的编码规则

(即系统的运行,按照粘贴自动机输入字符串的编码方式编码)

编码对象	编码
启动区	$I_1=5'$ATAT $3'$
间隔区	$X_0=5'$CA $3'$,$X_1=5'$AG $3'$,$X_2=5'$CT $3'$,$X_3=5'$AC $3'$
终止区	$I_2=5'$ CGCG $3'$
原子命题(输入字符)	$p=5'$ GG $3'$,$q=5'$TT $3'$,$r=5'$GT $3'$

表 3 式(1)对应的自动机

(按照粘贴自动机的编码方式编码)

编码对象	编码
开始状态 s_0	$3'$TATAGT $5'$
结束状态 s_2	$3'$TGGCGC $5'$
转移规则 $t(s_0,p)=s_1$	$3'$TCGATGCCGTTC $5'$
转移规则 $t(s_0,q)=s_2$	$3'$TCGATGAAGTTCGA $5'$
转移规则 $t(s_1,p)=s_1$	$3'$GATGCCGTTC $5'$
转移规则 $t(s_1,q)=s_2$	$3'$GATGAAGTTCGA $5'$
转移规则 $t(s_2,p)=s_2$	$3'$TGCCGTTCGA $5'$
转移规则 $t(s_2,q)=s_2$	$3'$TGAAGTTCGA $5'$

实验 1 系统模型如图 3,$|V|=3$,$|E|=3$,要验证的系统运行的最大长度是 15. 由图 3 可知:在状态 0,p 和 q 同时满足;由表 2 最后一行可知,p 对应的编码是 GG,q 对应的编码是 TT;因此状态 0 对应的编码是 GG/TT,其中斜线表示“或”,即不同的 DNA 分子可在 p 和 q 任选其一. 依次类推,状态 1 对应的编码是 GG,状态 2 对应的编码是 TT. 表 4 是系统模型长度不超过 15 的所有 7 个运行. 算法 2 让表 3 中单链 DNA 分子与表 4 中的单链 DNA 分子进行充分的生化反应,得到的双链 DNA 分子如表 5 所示,即为系统运行 7 条路径的检测结果. 由表 5 可见,所有路径都是完整的双链 DNA 分子,这表明所有运行均满足 $p\triangleright q$,因此系统满足 $p\triangleright q$.

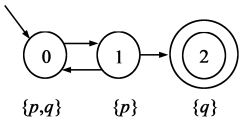

图3 实验1的系统模型

表 4 实验 1 的系统运行

路径	路径的 DNA 编码或者路径经过的顶点顺序
1 号路径的编码	ATAT CAAGCTAC GG/TT CAAGCTAC GG CAAGCTAC TT CAAGCTAC CGCG
1 号路径的顶点顺序	0,1,2
2 号路径的编码	ATAT CAAGCTAC (GG/TT CAAGCTAC GG CAAGCTAC)^2 TT CAAGCTAC CGCG
2 号路径的顶点顺序	0,1,0,1,2
k 号路径的编码	ATAT CAAGCTAC (GG/TT CAAGCTAC GG CAAGCTAC)^k TT CAAGCTAC CGCG
k 号路径的顶点顺序	$(0,1)^k$,2
7 号路径的编码	ATAT CAAGCTAC (GG/TT CAAGCTAC GG CAAGCTAC)^7 TT CAAGCTAC CGCG
7 号路径的顶点顺序	0,1,0,1,0,1,0,1,0,1,0,1,0,1,2

表 5　实验 1 的系统运行的检测结果

路径号	路径的 DNA 上链或 DNA 下链
1 号路径的上链	ATAT CAAGCTAC GG CAAGCTAC GG CAAGCTAC TT CAAGCTAC CGCG
1 号路径的下链	TATA GTTCGATG CC GTTCGATG CC GTTCGATG AA GTTCGATG GCGC
2 号路径的上链	ATAT CAAGCTAC GG CAAGCTAC GG CAAGCTAC TT CAAGCTAC GG CAAGCTAC TT CAAGCTAC CGCG
2 号路径的下链	TATA GTTCGATG CC GTTCGATG CC GTTCGATG AA GTTCGATG CC GTTCGATG AA GTTCGATG GCGC
k 号路径的上链	ATAT CAAGCTAC GG CAAGCTAC GG CAAGCTAC TT CAAGCTAC $(\text{GG CAAGCTAC})^{2k-3}$ TT CAAGCTAC CGCG
k 号路径的下链	TATA GTTCGATG CC GTTCGATG CC GTTCGATG AA GTTCGATG $(\text{CC GTTCGATG})^{2k-3}$ AA GTTCGATG GCGC
7 号路径的上链	ATAT CAAGCTAC GG CAAGCTAC GG CAAGCTAC TT CAAGCTAC $(\text{GG CAAGCTAC})^{11}$ TT CAAGCTAC CGCG
7 号路径的下链	TATA GTTCGATG CC GTTCGATG CC GTTCGATG AA GTTCGATG $(\text{CC GTTCGATG})^{11}$ AA GTTCGATG GCGC

实验 2　系统模型如图 4,$|V|=3$,$|E|=4$,要验证的系统运行的最大长度是 16. 由图 4 可知:状态 0 对应的编码是 GG/TT,状态 1 对应的编码是 GT,状态 2 对应的编码是 TT. 表 6 是系统模型长度不超过 16 的所有 29 个运行. 算法 2 让表 3 中的单链 DNA 分子与表 6 中的单链 DNA 分子进行充分的生化反应,得到的双链 DNA 分子如表7所示,即为系统运行29条路径的检测结果.

由表 7 可见,存在一部分路径不是完全的双链 DNA 分子,说明这些路径不满足 $p \triangleright q$. 因此系统不满足 $p \triangleright q$.

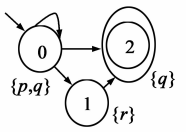

图4　实验2的系统模型

表 6　实验 2 的系统运行

路径	路径的 DNA 编码或者路径经过的顶点顺序
1 号路径的编码	ATAT CAAGCTAC GG/TT CAAGCTAC TT CAAGCTAC CGCG
1 号路径的顶点顺序	0,2
2 号路径的编码	ATAT CAAGCTAC GG/TT CAAGCTAC GG/TT CAAGCTAC TT CAAGCTAC CGCG
2 号路径的顶点顺序	0,0,2
3 号路径的编码	ATAT CAAGCTAC GG/TT CAAGCTAC GT CAAGCTAC TT CAAGCTAC CGCG
3 号路径的顶点顺序	0,1,2
$2k$ 号路径的编码	ATAT CAAGCTAC $(\text{GG/TT CAAGCTAC})^{k+1}$ TT CAAGCTAC CGCG
$2k$ 号路径的顶点顺序	0^{k+1},2
$2k+1$ 号路径的编码	ATAT CAAGCTAC$(\text{GG/TT CAAGCTAC})^{k}$ GT CAAGCTAC TT CAAGCTAC CGCG
$2k+1$ 号路径的顶点顺序	0^{k},1,2
28 号路径的编码	ATAT CAAGCTAC $(\text{GG/TT CAAGCTAC})^{15}$ TT CAAGCTAC CGCG
28 号路径的顶点顺序	0,0,0,0,0,0,0,0,0,0,0,0,0,0,2
29 号路径的编码	ATAT CAAGCTAC $(\text{GG/TT CAAGCTAC})^{14}$ GT CAAGCTAC TT CAAGCTAC CGCG
29 号路径的顶点顺序	0,0,0,0,0,0,0,0,0,0,0,0,0,1,2

表 7　实验 2 的系统运行的检测结果

路径号	路径的 DNA 上链或 DNA 下链
1 号路径上链	ATAT CAAGCTAC GG CAAGCTAC TT CAAGCTAC CGCG
1 号路径下链	TATA GTTCGATG CC GTTCGATG AA GTTCGATG GCGC
2 号路径上链	ATAT CAAGCTAC GG CAAGCTAC TT CAAGCTAC TT CAAGCTAC CGCG
2 号路径下链	TATA GTTCGATG CC GTTCGATG AA GTTCGATG AA GTTCGATG GCGC
3 号路径上链	ATAT CAAGCTAC GG/TT CAAGCTAC GT CAAGCTAC TT CAAGCTAC CGCG
3 号路径下链	TATA GT　　TG GCGC

续表

$2k$ 号路径上链	ATAT CAAGCTAC GG CAAGCTAC TT CAAGCTAC (GG CAAGCTAC)$^{k-1}$ TT CAAGCTAC CGCG
$2k$ 号路径下链	TATA GTTCGATG CC GTTCGATG AA GTTCGATG (CC GTTCGATG)$^{k-1}$ AA GTTCGATG GCGC
$2k+1$ 号路径上链	ATAT CAAGCTAC (GG/TT CAAGCTAC)^{k} GT CAAGCTAC TT CAAGCTAC CGCG
$2k+1$ 号路径下链	TATA GT TG GCGC
28 号路径上链	ATAT CAAGCTAC GG CAAGCTAC TT CAAGCTAC (GG CAAGCTAC)^{13} TT CAAGCTAC CGCG
28 号路径下链	TATA GTTCGATG CC GTTCGATG AA GTTCGATG (CC GTTCGATG)^{13} AA GTTCGATG GCGC
29 号路径上链	ATAT CAAGCTAC (GG/TT CAAGCTAC)^{14} GT CAAGCTAC TT CAAGCTAC CGCG
29 号路径下链	TATA GT TG GCGC

实验 1 和实验 2 分别针对系统满足公式和系统不满足公式这两种情况进行仿真检测. 把有关这些实验所获结果的表格(表 4、表 5、表 6 和表 7)综合起来,我们可以看出:无论系统模型是否满足式(1),该式都可以使用 DNA 分子来实现模型检测;特别是,当系统不满足式(1)时,新算法可以给出反例,以具体指出是系统中的哪些路径不满足公式,正如表 7 中的单号路径所示.

进一步地,我们把有关新方法的上述实验结果与相关方法进行比较,结果如表 8 所示. 从表中不难看出:文献[18]给出的是针对 CTL 公式的 DNA 模型检测方法;而本文给出的是针对 LTL 公式的 DNA 模型检测方法. 由于 Until 是 LTL 公式的核心算子,Final 与 Always 等其它时序算子表达的公式均可转化为 Until 算子表达的公式,因此,时序算子加原子命题组成的 LTL 公式就可以在 DNA 分子上实施模型检测. 这是新算法相对于文献[18]中方法的比较优势.

表 8 时序逻辑模型检测的 DNA 方法的检测能力之比较

公式类型	本文算法	文献[18]给出的方法
EF p	不能检测	能检测,不能给出反例
$p \triangleright q$	能检测,可以给出反例	不能检测
Always p	能检测,可以给出反例	不能检测
Final p	能检测,可以给出反例	不能检测
其他情况	不能检测	不能检测

6 结论

本文的主要成果是算法 2. 使用该算法即可在 DNA 计算框架内实现 LTL 逻辑的核心算子模型检测. 这是本文工作的贡献. 一方面,基于 DNA 计算的方法可利用其巨大的天然的并行计算优势缓解 LTL 模型检测当前面临的状态空间爆炸这一瓶颈,这是新算法在计算机科学中的潜在应用价值. 另一方面,在细胞内进行的 LTL 模型检测计算,将使得对肿瘤分子标志发展过程的动态识别与动态地自动调整药物用法与用量成为可能,进而有望为人类疾病的分子诊疗提供更好的自治智能方法,这是新方法在分子生物学与医学中的潜在应用价值.

参考文献

[1] CLARKE E. Model Checking[M]. Massachusetts: MIT Press, 1999.

[2] BAMAT J, BAUCH P, BRIM L, et al. Designing fast LTL model checking algorithms for many-core GPUs[J]. Journal of Parallel and Distributed Computing, 2012, 72(9): 1083 – 1097.

[3] CARBONE R. LTL model-checking for security protocols [J]. AI Communications, 2011, 24(3): 281 – 283.

[4] GOTZHEIN R. Temporal logic and applications-a tutorial [J]. Computer Networks and ISDN Systems, 1992, 24(3): 203 – 218.

[5] BENARI M, PNUELI A, MANNA Z. The temporal logic of branching time[J]. Acta Informatica, 1983, 20(3): 207 – 226.

[6] ADLEMAN L. Molecular computation of solutions to combinatorial problems[J]. Science, 1994, 266(5187): 1021 – 1023.

[7] ADLEMAN L. On constructing a molecular computer[J]. Discrete Mathematics and Theoretical Computer Science, 1995, 27: 1 – 21.

[8] ZIMMERMANN K, IGNATOVA Z, PEREZ M. DNA Computing Models[M]. Berlin: Springer, 2008.

[9] YIN Z X, ZHANG F Y, XU J. A Chinese postman problem based on DNA computing[J]. Journal of Chemical Information and Computer Sciences, 2002, 2(42): 222 – 224.

[10] YIN Z X, ZHANG F Y, XU J. The general form of 0-1 programming problem based on DNA computing[J]. Biosystems, 2003, 70(1): 73 – 79.

[11] PAN L Q, XU J, LIU Y C. A surface-based DNA algorithm for the maximal clique problem[J]. Chinese Journal of Electronics, 2002, 11(4): 469 – 471.

[12] PAN L Q, XU J. A surface-based DNA algorithm for the

minimal vertex cover problem[J]. Progress in Natural Science,2003,13(1):81 - 84.

[13] 高琳,许进. 图的顶点着色问题的 DNA 算法[J]. 电子学报,2003,31(4):494 - 497.
GAO Lin,XU Jin. A DNA algorithm for graph vertex coloring problem[J]. Acta Electronica Sinica,2003,31(4):494-497. (in Chinese)

[14] XU J,QIANG X L,ZHANG K,et al. A parallel type of DNA computing model for graph vertex coloring problem [A]. 2010 IEEE Fifth International Conference on Bio-inspired Computing:Theories and Applications[C]. Changsha:IEEE Press,2010. 231 - 235.

[15] YANG J,DONG C,DONG Y,et al. Logic Nanoparticle beacon triggered by the binding induced effect of multiple inputs[J]. ACS Applied Material Interfaces, 2014, 6(16):14486 - 14492.

[16] ZHANG C,WU L,YANG J,et al. A molecular logical switch beacon controlled by thiolated DNA signals[J]. Chemical Communications,2013,49:11308 - 11310.

[17] ZHANG C,MA J,YANG J. Nanoparticle aggregation logic computing controlled by DNA branch migration[J]. Applied Physics Letters,2013,103(9):093 - 106.

[18] EMERSON E A,HAGER K D,KONIECZKA J H. Molecular model checking[J]. International Journal of Foundations of Computer Science,2006,17(04):733 - 741.

作者简介

朱维军　男,1976 年生于河南郑州,现为郑州大学副教授. 主要研究方向:DNA 计算、形式化方法.
E-mail:zhuweijun76@163.com

周清雷　男,1962 年生于河南新乡,现为郑州大学教授、博士生导师. 主要研究方向:DNA 计算、形式化方法.
E-mail:ieqlzhou@zzu.edu.cn