

基于 DNA Tiles 自组装的布尔逻辑运算

黄玉芳¹⁾ 程 珍¹⁾ 周 康²⁾ 肖建华³⁾ 石晓龙¹⁾

¹⁾(华中科技大学控制科学与工程系分子生物计算实验室 武汉 430074)

²⁾(武汉工业学院数理科学系 武汉 430023)

³⁾(南开大学现代物流研究中心 天津 300071)

摘 要 大量研究工作表明, DNA tiles 自组装现象是分子生物计算过程中一个很重要的计算方式. 分子自组装的基本特点在于由许多小分子在一定机理的作用下, 自动形成更大规模的超级分子结构的过程. 自组装用于计算, 在于这种组装模式可以抽象成一个自动化的系统, 只需根据问题的需要设计好输入, 再将其输入到运算系统, 经过分子自组装过程, 最后能生成问题的解. 文中基于这样的运算机理, 在 DNA tiles 自组装这个计算平台上, 尝试做布尔逻辑运算, 针对 4 变量 4 句子的布尔逻辑问题, 提出一个 DNA tiles 自组装自动化运算系统.

关键词 自组装; DNA tiles; 分子计算; 布尔逻辑计算; 自动化系统

中图法分类号 TP301 DOI号: 10.3724/SP.J.1016.2009.02347

Boolean Logic Computation Based on Dan Tiles Self-Assembly

HUANG Yu-Fang¹⁾ CHENG Zhen¹⁾ ZHOU Kang²⁾ XIAO Jian-Hua³⁾ SHI Xiao-Long¹⁾

¹⁾(Research Institute of Bio-molecular Computing, Department of Control Science and Engineering,

Huazhong University of Science and Technology, Wuhan 430074)

²⁾(Department of Mathematics and Physic, Wuhan Polytechnic University, Wuhan 430023)

³⁾(Research Center of Logistics, Nankai University, Tianjin 300071)

Abstract A large number of work has demonstrated that self-assembly of DNA tiles is a significant method among molecular computations. Self-assembly is a process in which small objects autonomously associate with each other to form larger complexes. The simple binary arithmetic and logical operations can be computed by the process of self assembly of DNA tiles, yielding the results of the problem. In this paper, the authors consider taking advantage of the self-assembly of DNA tiles for logical evaluation, and propose a procedure to compute a 4-variable 4-clause 3-conjunctive (3-CNF) Boolean computational problem. The procedure enables any Boolean operations whose inputs and outputs are defined by a truth table, and evaluates any different kinds of Boolean logical formula simultaneously.

Keywords self-assembly; DNA tiles; molecular computation; logical evaluation; SAT problem

1 引 言

基于 DNA tiles 自组装技术设计算法求解数学

问题有可能成为突破 DNA 计算瓶颈的有效方式之一. 众所周知, 传统的电子计算机是在硅的物理基础上建立起来的, 随着生产工艺的日渐成熟与电子器件的逐渐微小化技术的发展, 这种微型技术似乎已

收稿日期: 2008-10-20; 最终修改稿收到日期: 2009-10-22. 本课题得到国家自然科学基金(60674106, 30870826, 60703047, 60533010, 60803113, 60903105)、国家“八六三”高技术研究发展计划项目基金(2009AA012413)、湖北省教育厅科学技术研究项目(重点)资助课题(D20091805)、湖北省自然科学基金项目资助课题(2007ABA407)资助. 黄玉芳, 女, 1982 年生, 博士研究生, 研究方向为分子生物计算、信息安全、图论、高效算法与设计. E-mail: yufangh369@163.com. 程 珍, 女, 1982 年生, 博士研究生, 研究方向为分子生物计算、信息安全与图论. 周 康, 男, 1965 年生, 副教授, 研究方向为分子生物计算、最优化理论研究. 肖建华, 男, 1979 年生, 讲师, 研究方向为生物计算和物流系统优化. 石晓龙, 男, 1975 年生, 副教授, 研究方向为模式识别、DNA 计算、分子生物自动机. E-mail: sxldna@126.com.

经接近瓶颈期,进而导致电子计算机的存储能力也在求解一些需要超大运算量的问题上(比如, NP 难问题)显得无能为力,微型化便在此时被提出来了. 因为这类问题往往随着运算规模的增大,其所需的运算量也将呈现指数级的增长. 早在 1959 年, Feynman^[1]就提出了建立亚微观计算机的设想,并对其可行性进行了研究. 于是,科学家们开始探索并设计基于其他的物质为基本运算介质的新型计算模式,目前为止,这方面的研究工作主要围绕着分子生物计算机以及量子计算机的研制展开. 在此,我们主要从分子生物计算机的研究出发,首先回顾这种计算模式的发展起源.

为分子生物计算机的发展做的奠基性的工作应该要追溯到 1994 年, Adleman^[2]用 DNA 计算的方法解决了有向 Hamilton 路径问题,并成功地利用现代分子生物技术在装有 DNA 溶液的试管中进行了实验. 该工作尽管只解决了一个规模仅为 7 个顶点的有向图的 Hamilton 路径问题,但是它所带来的突破性的开创意义是不可忽视的. 因为它的重要意义不仅仅在于算法与速度,更在于采用了一种全新的介质作为计算要件,提供了一种以生物技术来实现电子计算机在多项式时间内难以解决的困难问题的计算方式. Adleman 的分子生物计算模型主要反映了以下两个事实:(1) DNA 分子链的巨大并行性;(2) Watson-Crick 的互补结构. 再加上,许多著名的难解计算问题在尚未找到有效的多项式时间算法的前提下,可以通过穷尽所有可能解来求解. 然而,这样的搜索规模过于庞大,利用当前的技术是无法完成的. 但是, DNA 分子链能够高密度的存储信息,并且能轻而易举地进行大量的拷贝,使得穷尽搜索解决这类问题成为可能. 而 Watson-Crick 的互补结构保证了 DNA 分子可以通过互补或解链来实现各种计算,如果再建立良好的数据结构, DNA 计算便能用来求解更多的更复杂的数学问题. 随后,在 Adleman 的计算模型基础上,威斯康辛大学、普林斯顿大学、斯坦福大学以及加州理工学院等科研单位均开展了这方面的研究工作^[3-4].

由于一切的复杂困难计算问题最终都离不开基本的代数运算,所以为了研制出可以求解困难问题的分子生物计算机,就有必要研究如何在分子尺度上执行代数运算. 除了将 DNA 计算用于求解数学困难问题之外,科学家们尝试用这种分子生物计算模式执行最基本的代数运算. Guarnieri 等人^[5]采用类似于电子计算机中的位交换方法,来实现二进制

数的存取和进位,首次运用一系列引物延伸反应实验完成了两个二进制数的加法运算,并推广到任意两个 n 进制数的加法,其结果证实了 DNA 计算机具备执行数学基本运算的能力. Fujiwara 等人^[6]进一步提出了可寻址的 DNA 逻辑运算及四则运算的方案.

Wang^[7]在其著作中,从数学的角度提出了“Wang Tile”的概念,并指出 Wang Tile 可以通过自组装过程,在二维平面形成周期性格局,而且该过程具有图灵机等价性. Wang Tile 理论显示,如果给定一组方块形的 Tiles,每个 Tile 的边都涂上不同的颜色,则不同 Tiles 在边颜色相同的地方会粘合在一起. 1982 年,纽约大学的 Seeman 教授致力于 DNA 分子结构的研究,对 Tile 结构的物理化学性质作了深入细致的探索,提出利用 Tile 分子构建各种简单分子器件的设想^[8]. 随后, Winfree 和 Seeman^[9-10]在研究分子纳米结构的基础上,受到 Wang Tiling 的思想的启发,提出通过 Tile 分子自组装的过程来实现计算的思想,并证明 DNA Tile 自组装计算具有图灵等价性.

本文主要讨论基于 DNA Tile 实现布尔逻辑运算;第 2 节主要回顾 DNA Tile 自组装机理以及 DNA Tile 自组装运算的一些应用方面的研究进展;第 3 节提出基于 DNA Tile 自组装实现布尔逻辑运算;最后,我们给出进一步的结论和讨论.

2 DNA Tile 自组装的机理及其应用

分子自组装代表了一种“自下而上”的制造高精度复合物的方法. 这个方法可以分为两步实现:首先,通过合成化学的方法构造出单个的分子;然后,把单个分子组装成大的结构. DNA 分子由于其结构特点和特有的分子间相互作用特性(Watson-Crick 互补规则),尤其适合于充当分子生物计算的基本材料.

DNA 自组装计算模型使用的 DNA Tile 是一簇具有分支结构的 DNA 交叉分子,每一个分子都有一个粘贴末端,可以与具有互补粘贴末端的 Tile 嵌套结合,逐步形成 DNA 分子网络结构. DNA Tile 自组装用于运算的核心思想在于,对于一个给定的问题,首先就问题建立算法,然后依据问题特点以及算法结构,设计 DNA Tile 集,编码原则严格按照算法结构进行,从而保证利用 DNA Tile 自组装过程求解问题的解具有完备性.

当前, DNA Tile 计算所用到的自组装形式主要从三个方面进行讨论, 即一维自组装(又称线性自组装)、二维自组装以及三维自组装. 其中, 分子生物计算的早期研究工作主要是利用了一维线性自组装, 例如粘帖模型、剪切模型等. 后来, 由于一维线性自组装计算模型中的实验操作步数太多, 以及生化反应中的 DNA 链错配率过高, 从而影响了求解问题时生成解的有效性. 为了克服这种不足, Winfree 等人提出二维的自组装计算模型, 该模型较之先前的一维自组装模型的最大优点就在于, 在整个自组装过程中人为操作数大大减少, 这也直接降低了在生物实验中人为导致的错误. 另外, 二维自组装的组空间更大, 使得其所能兼容的计算结构也更复杂, 因此, 更适合用它来解决复杂的困难问题. 这期间也有人提出三维自组装模型^[11-12], 但是这方面的实验还有待进一步的研究. 本文将重点放在探讨如何应用二维自组装进行计算.

目前为止, 围绕 DNA Tile 自组装作算术运算的研究已经广泛展开. 2000 年, LaBean^[13] 提出一种更复杂的 DNA Tile 分子构型, 即三交叉 Tile 分子, 他指出这类分子较以往的 Tile 类型在生化反应中更具刚性, 更稳定, 并通过生物实验验证了这种 Tile 分子自组装过程可以完成四步累积异或运算. Brun^[14] 提出一个抽象的 Tile 自组装算术模型, 并用它来实现二进制数的加法和乘法计算, 并用其乘法系统分解了大整数. Zhang^[15] 在 Brun 工作的基础上, 根据减法和除法操作的运算机理, 按照除法的运算过程, 基于 DNA Tile 自组装构建了 3 个子系统: 比较子系统、复制子系统和减法子系统用于完成除法运算, 同时也用该除法系统分解整数. 除此之外, Barua^[16] 将 DNA Tile 自组装计算模型用于建立有限域乘法和加法系统.

3 基于 DNA Tiles 自组装的布尔逻辑计算

3.1 布尔逻辑问题

布尔逻辑得名于 George Boole, 这位英国数学家在 19 世纪中叶首次定义了逻辑的代数系统. 现在, 布尔逻辑在电子学、计算机硬件和软件等方面中都有着广泛的应用. 在计算机中, 布尔逻辑定义若干布尔逻辑函数, 有时候称作操作符. 每个函数根据一个或者更多的输入, 用一个逻辑算法来计算输出值. 该算法根据输入所取真和假的组合来决定什么时候

输出真(0 真 1 假; 1 真 0 假, 相对的). 每个逻辑函数类似于一个现实世界的逻辑运算, 可以用来定义各种逻辑的情况. 本文将在分子尺度上, 在 DNA Tile 自组装的技术支持下, 实现对布尔逻辑表达式进行求值计算, 该方法以后将有可能被应用到求解数学困难问题, 比如可满足性问题.

本文重点讨论这类布尔逻辑表达式的求值问题, 不失一般性, 考虑如下的合取范式(CNF),

$$W = (x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee x_3 \vee \bar{x}_4) \wedge (x_2 \vee \bar{x}_3 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4),$$

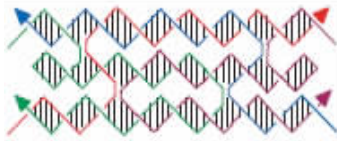
其中, $x_i (i=1, 2, 3, 4)$ 是独立命题变元, \bar{x}_i 表示变元 x_i 的非, 每个变元的取值范围为 $\{0, 1\}$, 数 0 和 1 逻辑意义分别为假和真. 而且, 符号“ \vee ”表示逻辑或运算, 即 $x_i \vee x_j = 0$ 当且仅当 $x_i = x_j = 0$, 符号“ \wedge ”表示逻辑与运算, 即 $x_i \wedge x_j = 1$ 当且仅当 $x_i = x_j = 1$. 本文的研究工作主要考虑对于逻辑变元组 (x_1, x_2, x_3, x_4) 任意给定一组赋值, 然后求解布尔逻辑表达式 W 是真还是假.

3.2 算子 Tile 的构造

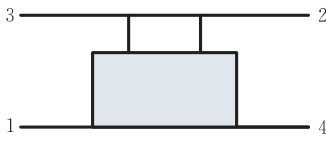
自组装技术目前面临的一个关键问题就是如何预先设计基本的 Tile 分子, 使得所有这些 Tiles 能按照我们的意愿在生化反应结束时生成目标问题的解. 为了解决这个问题, 我们要做的是, 当给定一个问题时, 首先分析问题本身, 结合 DNA Tile 自组装的特点, 设计相应的算法结构, 这个过程通常也是生成求解问题所需的算子 Tile 的过程, 这里的算子 Tile 指的是在自组装过程中, 具有粘帖末端, 而且其上承载着问题的信息. 为了让不同的 Tile 完成不同的函数功能, 就有必要巧妙设计每个 Tile 的粘帖末端, 这将有赖于 DNA 编码技术的不断发展, 从而最大限度地减少 Tile 在自组装过程中生成非解的几率.

首先, 我们来介绍一下本文即将用到的基本 Tile 类型. 三交叉 (triple crossover) DNA Tile 分子, 也简称为 TX 分子, 其分子结构形式上可以描述为图 1 中的(a), 它是由 4 条 DNA 单链通过碱基互补配对原则所形成的二维结构分子, 它最多可以伸展出 4 个粘性末端, 为了方便运算的表达, 我们将这种分子进一步抽象为图 1 中(b)的结构, 数字 1、2、3 和 4 分别表示 TX 分子的 4 个粘帖末端. 在下面的讨论中, 我们将根据计算的需要, 有选择性地选择其中一些粘帖末端, 分别设计其运算功能, 用于完成不同的运算.

我们知道, 衡量 DNA 计算性能高低的判断标准之一, 在于其能在多大程度上实现并行处理. 由于



(a) 三交叉(tripple crossover)DNA Tile 分子



(b) 抽象的运算 Tile 单元

图 1

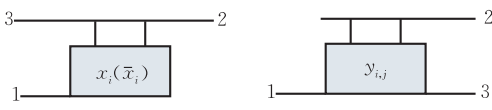
所讨论的布尔逻辑表达式 W 有 4 个语句,为了充分利用 DNA Tile 自组装计算的并行性,我们针对每个子句分别建立一个 tile 自组装运算子系统,每个子系统将同时独立执行其中一个语句的运算,分别记这 4 个子系统为 C_1, C_2, C_3 和 C_4 .

为了启动每个子系统的运算,我们采用类似文献[13]中的方法,对每个自组装运算子系统分别构造其种子框架,用于启动整个自组装运算. 设 $C_{i,i-1}$ 和 C_{ii} 分别为第 i 个子系统的启动 tile,其中 $i=1, 2, 3, 4$ 的分子结构如图 2(a)所示. C_{10}, C_{21}, C_{32} 和 C_{43} 为顶上左右两边各有一个粘性末端,在随后的讨论中,我们可以看到,这种 tile $C_{i,i-1}$ ($i=1, 2, 3, 4$) 的左上方的粘性末端将分别用于连接另外一个启动分子 C_{ii} ($i=1, 2, 3, 4$), 右上方的粘性末端将用于连接 W 中的第一个参加运算的变元(或变元非). 类似可知, tile C_{ii} ($i=1, 2, 3, 4$) 在其右上方和右下方分别有一个粘性末端,这样的设计将使得 tile C_{ii} 在第 i 个子系统中,通过右下方与另一启动 tile $C_{i,i-1}$ 相连, 右上方的末端将与运算 tile 相结合.

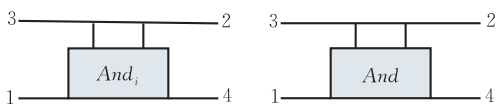
由布尔逻辑表达式 W 可知,其共有 4 个变元 x_i ($i=1, 2, 3, 4$), 另外参加计算的还有变元的非 \bar{x}_i , 其分子结构统一为图 2(b)的左边分子. 除却右下方位



(a) 两个起始tile用于连结输入的变元tile以及运算tile



(b) 变元(或变元非)tile以及作逻辑OR运算的tile



(c) 作逻辑AND运算的tile

图 2

置以外,剩下的 3 个角上分别都有一个突出的粘性末端,其中粘性末端 1 和 2 是用于在运算过程中,连接其它变元的 tile,而末端 3 则是向外输出该变元 tile 的逻辑值. 也就是说,对于变元(或变元非)的 tile 而言,其中的粘性末端 1 和 2 可以视作变元识别区域,其具体的编码设计将视该变元在逻辑语句中与什么变元发生逻辑运算而定. 我们的目标是,让适当的变元 tile 相对于整个运算结构,生长在适当的位置,从而保证能实现对目标问题的逻辑求值.

为了运算的需要,我们对于每个子系统 C_i , 分别设计一个 tile y_{i1} ($i=1, 2, 3, 4$) (见图 2(b)所示), 令 y_{i1} 的逻辑取值直接等于该系统中第一个变元的值. 这样的处理将使得每个子句的逻辑运算顺利进行. 类似地,我们设计 tile y_{ij} 用于执行每个语句内部的逻辑 OR 运算,其关系式如下所示,

$$\begin{aligned} y_{11} &= x_1, & y_{12} &= y_{11} \vee x_2, & y_{13} &= y_{12} \vee \bar{x}_3; \\ y_{21} &= x_1, & y_{22} &= y_{21} \vee x_3, & y_{23} &= y_{22} \vee \bar{x}_4; \\ y_{31} &= x_2, & y_{32} &= y_{31} \vee \bar{x}_3, & y_{33} &= y_{32} \vee x_4; \\ y_{41} &= \bar{x}_1, & y_{42} &= y_{41} \vee \bar{x}_2, & y_{43} &= y_{42} \vee \bar{x}_4. \end{aligned}$$

它们也同时记录了每个子运算系统运算过程中产生的临时结果,其分子结构如图 2(b)所示,其中粘性末端 1 用于连结上次运算的结果,粘性末端 3 用于连结参加下一步逻辑运算的变元(或变元非),粘性末端 2 将记录本次运算的结果. 且具体的 OR 运算规则如图 3(a)所示.

在每个子系统分别执行完各个子语句运算之后,我们将建立二级并行运算系统. 这里为了方便叙述,我们将上面提到的 4 个子系统统一记作一级并行运算系统. 令二级并行运算系统记作 D_1, D_2 , 其中, D_1 用于对子系统 C_1 和 C_2 的运算结果进一步作逻辑 AND 运算,其结果记作 And_1 , 且 $And_1 = y_{13} \wedge y_{23}$. 同时, D_2 用于对子系统 C_3 和 C_4 的运算结果进一步作逻辑 AND 运算,其结果记作 And_2 , 且 $And_2 = y_{33} \wedge y_{43}$. 具体的算子 tile 结构见图 2(c)左边分子所示. 粘性末端 1, 4 用于连结一级子系统的运算结果,粘性末端 3 用于标识这是第二级并行系统,这一点将在执行进一步的 AND 运算中发挥显著作用. 且其具体的 AND 运算规则见图 3(b).

在执行完二级并行运算系统之后,我们将建立三级并行运算系统 E_1 . 因为二级并行运算系统的最终运算结果只有两个,故这里只需设计出一个运算系统. 其算子 tile 结构见图 2(c)右方分子,左下方和右下方的两个粘性末端将用于识别上级并行运算系统的结果,顶端的两个粘性末端同时对外输出本次

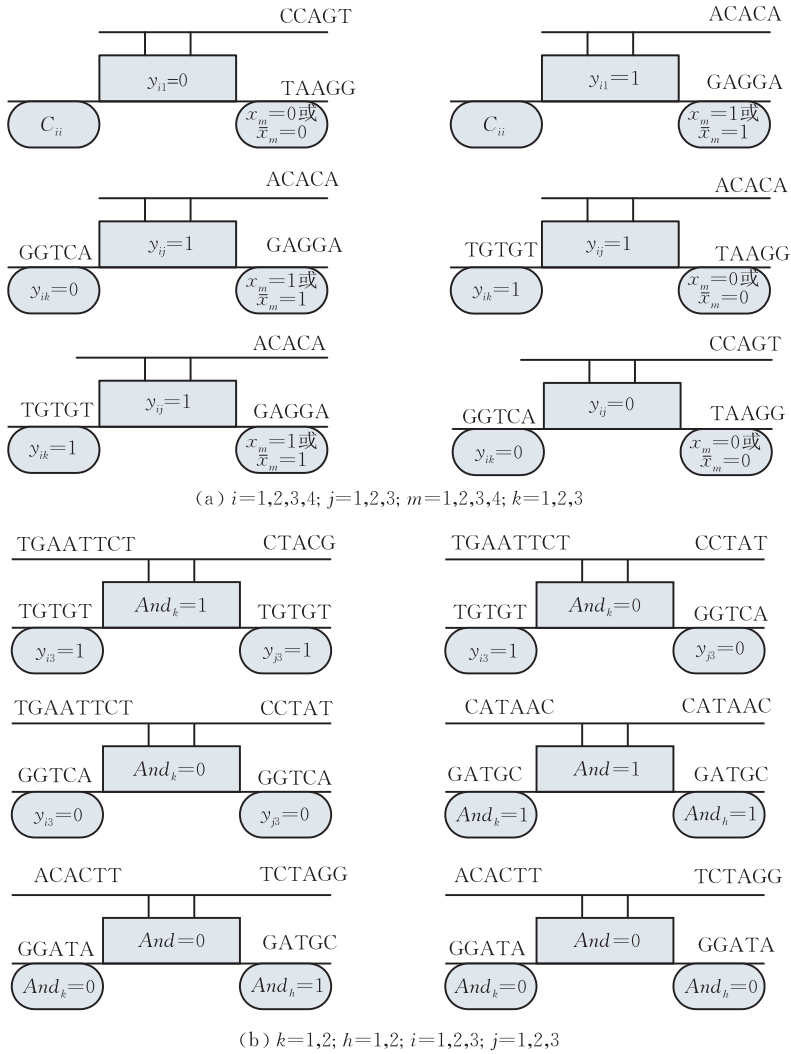


图 3 算子 Tile 的基本自组装规则((a) 运算 tile 执行逻辑 OR 运算的基本规则, 下方的两个方椭圆形结构 tile 作为输入, 当且仅当两个输入同时为 0 时, OR 运算的结果才为 0; (b) 运算 tile 执行逻辑 AND 运算的基本规则, 当且仅当下方的两个输入同时为 1 时, 其 AND 运算的结果才为 1)

AND 运算的结果, $And = And_1 \wedge And_2$, 且其具体的 AND 运算规则见图 3(b).

本文除却两类启动 tile(图 2(a))外, 其余的 tile 都是具有一定的布尔逻辑值的分子结构. 首先, 对变元 tile $x_i (i=1, 2, 3, 4)$ 或者变元非 tile $\bar{x}_i (i=1, 2, 3, 4)$ 而言, 当 $x_i = 1$ 时(见图 2(b)), 我们令其左上方的粘性末端为碱基序列 $5'-CTCCT-3'$; 当 $x_i = 0$ 时, 我们令其左上方的粘性末端为碱基序列 $5'-ATTCC-3'$. 类似地, 对于运算 tile y_{ij} , 当 $y_{ij} = 1$ 时, 令其右上方的粘性末端为碱基序列 $3'-ACACA-5'$; 当 $y_{ij} = 0$ 时, 令其右上方的粘性末端为碱基序列 $3'-CCAGT-5'$.

不失一般性, 对布尔逻辑问题 W 任给一组取值 $(x_1, x_2, x_3, x_4) = (1, 0, 0, 1)$, 通过一级子系统可以同时生成 4 个子系统, 如图 4 所示. 当构造二级并行

子系统时, 为了使一级子系统 C_1 和 C_2 形成二级子系统 D_1 进一步作逻辑 AND 运算, 我们设计探针 1, 其碱基序列根据变元非 \bar{x}_3 突出的粘性末端为碱基序列 $5'-TCTACAA-3'$, 变元非 \bar{x}_4 突出的粘性末端为碱基序列 $5'-GTCATCA-3'$, 于是探针 1 的碱基序列可以设计如下:

$$3'-NNNNNAGATGTTTCAGTAGTNNNNN-5'$$

类似地, 为了使一级子系统 C_3 和 C_4 形成二级子系统 D_2 进一步作逻辑 AND 运算, 我们设计探针 2, 其碱基序列设计如下:

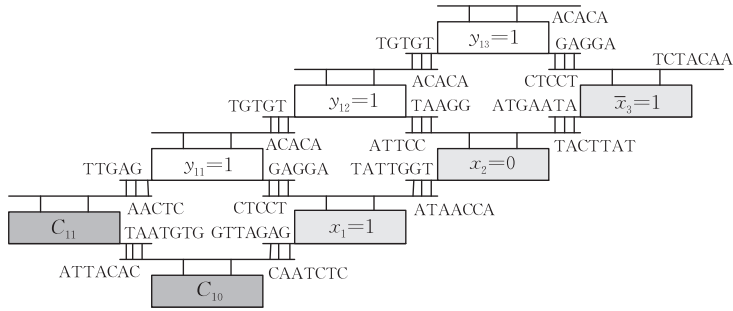
$$3'-NNNNNAATTGCCAGTAGTNNNNN-5'$$

在 T4 连接酶作用下, 将自动组装成二级子系统 D_1 和 D_2 , 见图 5 所示.

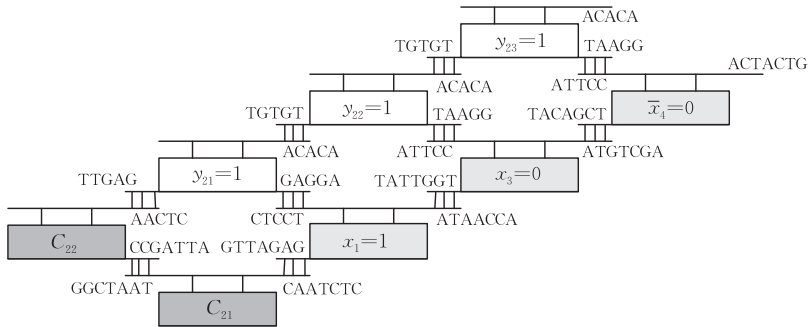
进一步地, 我们设计探针 3, 其碱基序列为 $3'-NNNNNNA-GAATTCAACTTAAGANNNNN-$

5', 用于连结二级子系统 D_1 和 D_2 , 以进一步作逻辑 AND 运算, 其运算结果如图 6 所示, 最后通过一定

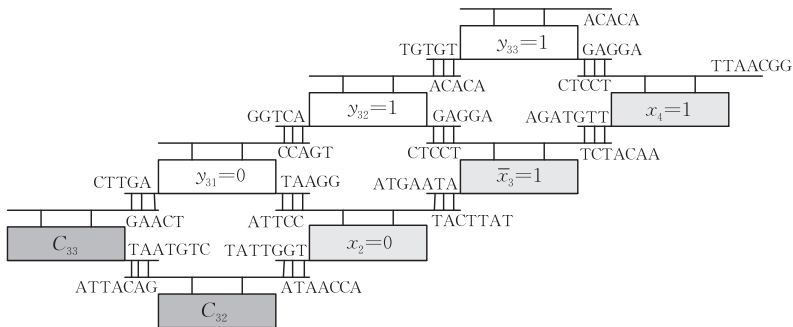
的生物操作可以提取出最终的结果 $And=1$, 即布尔逻辑问题的结果为 $W=1$.



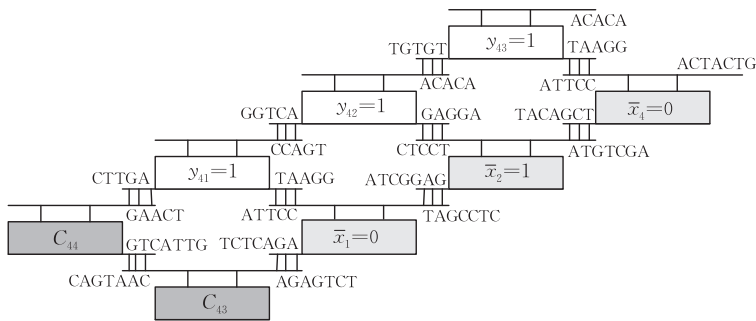
(a) 第 1 级自组装子系统中的 C_1 完成对布尔逻辑问题 W 的第 1 个语句 $x_1 \vee x_2 \vee \bar{x}_3$ 的运算



(b) 第 1 级自组装子系统中的 C_1 完成对第 2 个语句 $x_1 \vee x_3 \vee \bar{x}_4$ 的运算



(c) 第 1 级自组装子系统中的 C_3 完成对第 3 个语句 $x_2 \vee \bar{x}_3 \vee x_4$ 的运算



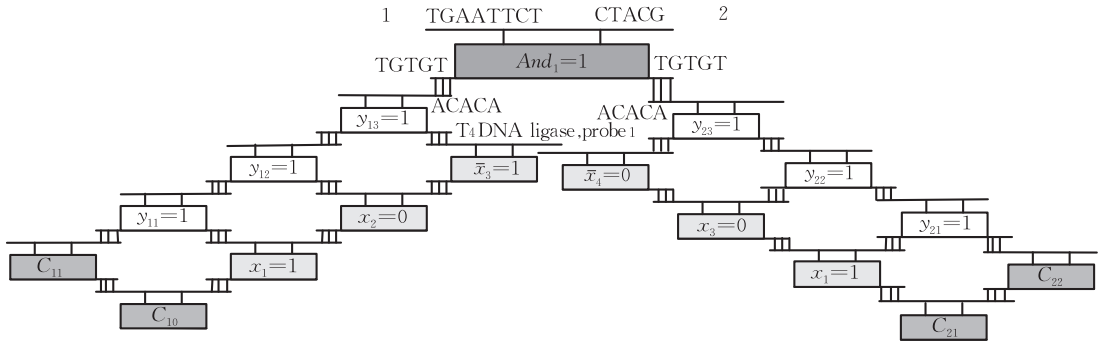
(d) 第 1 级自组装子系统中的 C_4 完成对第 4 个语句 $\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_4$ 的运算

图 4

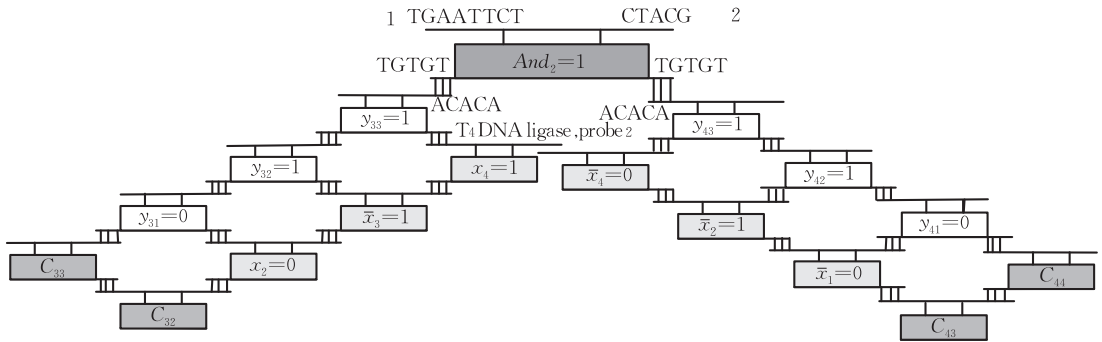
4 讨 论

本文在深入理解分子自组装理论的基础上, 借

助于已有的三交叉 DNA tile 分子结构类型, 将待运算的信息通过编码与 tile 的粘性末端相关联, 用 DNA Tile 自组装技术对给定的布尔逻辑问题进行求值运算. 将基本 tile 类型视作构建计算的算子



(a) 第 2 级自组装子系统 D_1 对第 1 级子系统中的 C_1 和 C_2 的结果进一步作逻辑 AND 运算



(b) 第 2 级自组装子系统 D_2 对第 1 级子系统中的 C_3 和 C_4 的结果作逻辑 AND 运算

图 5

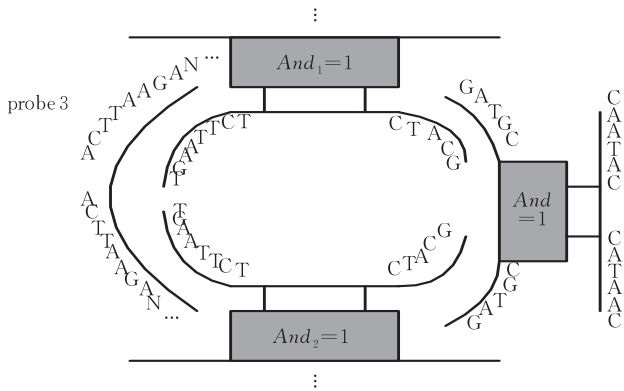


图 6 第 3 级 tile 自组装子系统 E_1 完成最后一步的逻辑 AND 运算,同时通过两条突出的粘性末端输出解 1

tile, 存储运算过程中的数值, 用以完成布尔逻辑运算. 具体的, 我们就具有一定形式的布尔逻辑表达式进行求值, 即 4 个变元, 由 4 个语句的合取范式构成的布尔逻辑计算问题, 设计了 4 个 tile 自组装一级子系统, 并在此基础上构建两个二级子系统, 随后进一步构建一个三级子系统. 每个子系统对应着 4 个合取范式中的一个语句, 相互之间独立地完成逻辑运算. 不同级别的子系统之间存在着承上启下的相互传承的关系, 即第 $(i+1)$ 个子系统是在第 i 个子系统的结果上作进一步的自组装, 以完成逻辑运算. 子系统的个数设计原则上是根据问题语句的个数来确定, 而且子系统的级数(即多少层子系统)则是由

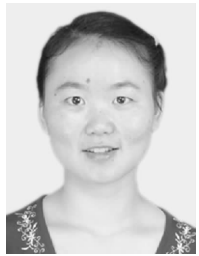
第一级子系统的个数逐层除二, 直到最后一级的子系统的自组装模块数为 1 才停止, 与此同时, 最后一级子系统将输出问题的运算结果. 通过构建第一级自组装子系统, 我们将已知问题输出到整个运算系统, 经过系统的自组装运算, 自组装系统将最后输出问题的解.

该方法原则上可以用于求解由任意个语句任意个变量所构成的布尔逻辑问题, 利用 tile 自组装运算的高度并行性, 能快速对问题求解. 该方法可以进一步用于解决 NP 完全问题中的可满足性问题, 不过面临的问题是, 如何对其中参加运算的 tile 给出恰当的编码设计, 近而能有效求解问题. 当然这也同样是困扰着 tile 自组装运算的一个话题, 相信这方面的研究工作能尽早解决编码问题中所遇到的困难.

参 考 文 献

- [1] Feynman R P, Gilbert D H. In Minaturization. New York: Reinhold Publishing Corporation, 1961: 282-296
- [2] Adleman L. Molecular computation of solutions to combinatorial problems. Science, 1994, 266(5187): 1021-1024
- [3] Lipton R J. DNA solutions of hard computational problems. Science, 1995, 268(28): 542-545
- [4] Braich R S, Chelyapov N, Johnson C, Rothmund P W K, Adleman L. Solutions of 20-variable 3-SAT problem on a

- DNA computer. *Science*, 2002, 296(5567): 499-502
- [5] Guarnieri F, Fliss M, Bancroft C. Making DNA add. *Science*, 1996, 273(5272): 220-223
- [6] Fujiwara A, Matsumoto K, Chen W. Addressable procedures for logic and arithmetic operations with DNA strands// *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*. Washington, DC, USA, 2003; 162.1
- [7] Wang H. Dominoes and the AEA case of the decision problem// *Proceeding of the Symposium Mathematical Theory of Automata*. New York, 1963; 23-55
- [8] Seeman N C. Nucleic-acid junctions and lattices. *Journal of Theoretical Biology*, 1982, 99(11): 237-247
- [9] Winfree E, Yang X P, Seeman N C. Universal computation via self-assembly of DNA: Some theory and experiments// *Proceedings of the 2nd DIMACS Meeting on DNA Based Computers*. Landweber, Lipton, 1996
- [10] Winfree E. Algorithmic self-assembly of DNA[D]. California Institute of Technology, Pasadena, California, 1998
- [11] Kao M, Ramachandran V. DNA self-assembly for constructing 3D boxes// *Lecture Notes in Computer Science 2223*. Springer Berlin, Heidelberg, 2001; 429-441
- [12] Essam A, Belal Z, Feras A. 3D DNA nanostructures for vector multiplication. *American Journal of Scientific Research*, 2009, (1): 5-11
- [13] Mao C D, LaBean T, Reif J, Seeman N. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, 2000, 407(6803): 493-496
- [14] Brun Y. Arithmetic computation in the tile assembly model: Addition and multiplication. *Theoretical Computer Science*, 2007, 378(1): 17-31
- [15] Zhang X C, Wang Y F, Chen Z H, Xu J, Cui G Z. Arithmetic computation using self-assembly of DNA tiles: Subtraction and division. *Progress in Natural Science*, 2009, 19(3): 377-388
- [16] Barua R, Das S. Finite field algorithmic using self-assembly of DNA tilings// *Proceedings of the 2003 Congress on Evolutionary Computation*. Canberra, Australia, 2003; 2529-2536



HUANG Yu-Fang, born in 1982, Ph. D. candidate. Her research interests include bio-computer, graph theory, cryptology and system engineering.

CHENG Zhen, born in 1982, Ph. D. candidate. Her research interests include bio-computer and cryptology.

Background

The research group mainly devotes themselves to the field including the Graph Theory, Information Security, Model and Method of DNA Computer etc. They have published more than 150 papers on DNA computing and DNA computer. The paper proposes an arithmetic model to evaluate a four-variable four-clause Boolean computational problem based on the DNA tiles self-assembly. The smart "DNA tiles" invented by Winfree are the most promising approaches among DNA computing methods. Winfree's brainstorm is to create nano-scopic building blocks out of DNA that not only can store data but are designed- Winfree likes to say "programmed"-to carry out mathematical operations by fitting together in specific ways. Normally, DNA exists as two intertwined strands of the chemical letters A, G, C and T-the familiar double helix. But Winfree's DNA tiles are made by knotting together three or more of these strands, forming "tiles" about 15 nanometers (billionths of a meter) on their longest side. Taking advantage of DNA's ability to selectively recognize other strands of DNA, Winfree has "coded" the edges of these tiles so that they come together in just the right way to form tiny built-to-order structures. Then, Mao

ZHOU Kang, born in 1965, Ph. D. candidate, associate professor. His research interests include bio-computer and operational research.

XIAO Jian-Hua, born in 1979, Ph. D., lecturer. His research interests include Bio-inspired computation and logistics optimization.

SHI Xiao-Long, born in 1975, associate professor. His research interests include pattern recognition, DNA computing and bio-molecular automaton.

and his colleagues have used DNA triple crossover molecules with diagonal reporter strands as tiles to perform four steps of a cumulative exclusive OR (XOR) operation in two independent assemblies in their laboratory. Brun has defined an arithmetic system of tile assembly, and tackled these issues to compute the sum and product of two numbers in binary. All these work have demonstrated that the DNA tiles can program in light of our careful designs to accomplish a variety of computations. This model proposed in this paper has demonstrated the huge potential for computing inherent in DNA tiles. It shows that the great potential of algorithmic self-assembly of DNA tiles can be used to solve the SAT problem. When the conventional computer based on silicon is facing more and more seriously hard computing problems, at least needing quite a long time to get the result, human beings have to turn to other computing methods for help. Algorithm basing on the self-assembling of DNA tiles may be one of the possible ways to break the limit of brute-force method in DNA computing. Many scientists have been paying their attentions on this field for solving NP problems using DNA tiles self-assembly.