

基于线性自组装的 DNA 减法模运算

方习文, 来学嘉

上海交通大学计算机科学与工程系, 上海 200240

E-mail: cieven@sjtu.edu.cn

2009-11-20 收稿, 2010-01-19 接受

国家自然科学基金(批准号: 60773092, 60573032)和教育部博士点基金(编号: 20090073110027)资助项目

摘要 使用线性自组装方法, 提出了两个非负二进制整数减法模运算的 DNA 算法. 对于两个表示为 n 位的二进制数 A 与 B , 算法给出 $A-B$ 在模 2^n 情况下的运算结果. 算法中包含反应被减数与减数大小关系的扩展借位信息, 从而在计算前不必对 A 与 B 的大小关系进行预分类. 结果反应链中包含运算结果、每一步借位信息、参与运算的数值、判断被减数与减数大小的标志位等信息. 算法充分利用 DNA 反应的并行特性, 在给定两个被减数集与减数集时, 可进行两个集合的减法模运算的并行计算. 算法的可行性基于已知的 DNA 算法实验. 算法具有良好的自发反应特性, 避免了人工操作随运算数值位数增长的情况, 对于计算位数 n , 在本算法中参与反应的单链库规模为 $O(n)$, 生物操作复杂度为常数.

关键词

DNA 计算
DNA 减法
模运算
自组装

自 1994 年 Adleman^[1]首次使用 DNA 计算方法提出汉密尔顿路径问题的解法以来, DNA 计算成为了一项新的研究热点^[2-4], 各种计算模型被陆续提出, 如黏帖模型^[5-8], 剪接模型^[9,10]等. 关于 DNA 计算解决组合问题已有很多讨论^[11-13], 且 DNA 技术也被应用到了密码系统中^[14,15].

在 DNA 计算的具体运算模块设计方面, 目前 DNA 加法运算的算法已有若干讨论. 自 Guarnieri 等最早提出一种 DNA 加法算法^[16]以来, Gupta 等^[17], Wasiewicz 等^[18]陆续提出若干种加法算法方案, 但这些方案最大的弱点是实验中生物操作步骤会随着参与运算的位数的增加而增加, 这是实验可操作性上的致命影响.

若在 DNA 计算方法中使用自组装方法, 则可能使之具备自发反应特性. 1998 年 Winfree 等构造出若干种 DNA 链互相拼接而成的自组装模型(DX 模块)^[19], 2004 年 Rothmund 等使用自组装模型完成了 DNA 计算的 XOR 操作^[20], 使用自组装模型在运算过程中可有效控制人为生物操作步骤, 赵健等基于线性自组装模型提出了加法算法^[21], 其生物操作步骤

为常数, 算法特性优于此前的 DNA 加法方案.

虽然在计算机传统计算方法中, 减法运算可以通过转换为补码按照与加法相同的规则实现, 但在 DNA 计算中, 若按照同样的方式执行运算, 存在两个弊端: 首先, 需另外设计增加运算前转换补码模块, 以及运算后将结果转换原码模块, 加之加法运算模块构成整体, 而这 3 个模块之间的连接需要人工操作, 限制整个过程的自组装自发连续性; 其次, 补码与原码的互相转换若使用 DNA 计算实现, 其反应规模对于整个减法运算来说远不是一个可忽略过程, 人为增加了反应的难度. 因此, 针对于减法运算模块特定编码的研究仍然具有其实际意义.

减法作为最重要的运算模块之一, 模减法运算则在密码系统中更为重要. 其易于实验操作的有效方案尚鲜有提及, 其难点在于减法借位约束关系, 被减数与减数大小的比较, 以及模运算的处理. 不能直接改换 DNA 加法方案的编码数据到 DNA 减法运算, 需给出新的编码方案来处理减法借位时的状态和模运算.

本文提出一种基于线性自组装方法的 DNA 减法

模运算, 给出 DNA 减法的编码方法, 该算法可在简洁的实验步骤下完成 DNA 减法运算, 最终反应生成链中除包含结果外还包含被减数与减数的每一位数值、每一位借位信息及扩展借位信息, 其中扩展借位信息标志着被减数与减数的大小判断, 且整个实验过程中, 对于运算位数 n , 整个单链库的规模为 $O(n)$, 生物操作复杂度为常数.

1 算法原理

1.1 表示方法及约束条件定义

在本算法中, 被减数与减数分别标记为 A 和 B , 均为非负整数. $A-B$ 的结果以 S 表示, 其中 A, B, S 均为用 n 位表示的二进制数, 高位不足 n 位的以 0 补满.

在算法描述中间过程中, 记被减数、减数、结果的第 i 位分别为 A_i, B_i, S_i . 在进行第 i 位上的运算时, 记当前位的借位为 C_i .

第 i 位上的减法计算由下列两步组合而成:

I) 以 A_i 与当前位上的借位 C_i 相减, 得出中间结果 H_i , 其中可能发生下一位的借位, 以 C_{i+1} 记录下一位的借位情况;

II) 综合中间结果 H_i 与下一位的借位情况 C_{i+1} , 令 H_i 与 B_i 相减, 并更新 C_{i+1} 的值. 其中, 在第一步求得中间结果的过程中, 对于不同的 A_i 与借位 C_i 的组合, 其产生的结果如表 1 所示.

从表中易知, H_i 与 C_{i+1} 只可能有 (0,0), (1,0), (1,1) 三种组合.

在第二步计算中, 最终的结果 C_{i+1}, S_i 由 B_i 及上一步中间结果 H_i, C_{i+1} 共同决定. B_i 只有 0 和 1 两种取值, 每种取值对应 H_i 与 C_{i+1} 三种组合, 按照减法法则, 其计算结果如表 2 所示.

由表 1、表 2 可知, 在第 i 位减法计算的第二步

$$\begin{array}{r} A_{n-1} \dots A_i A_0 \\ - B_{n-1} \dots B_i B_0 \\ \hline S_{n-1} \dots S_i S_0 \end{array}$$

图 1 二进制减法中被减数、减数、结果的表示

$$\begin{array}{r} A_i \quad (C_{i+1}) \quad H_i \\ - C_i \quad - B_i \\ \hline (C_{i+1}) \quad H_i \quad (C_{i+1}) \quad S_i \\ \text{(I)} \quad \quad \quad \text{(II)} \end{array}$$

图 2 第 i 位上减法计算的两个步骤

骤中, 所使用到的 H_i 与 C_{i+1} 由第一步给定. 而进行到第 $i+1$ 位时, 计算的第一步骤中所需的 C_{i+1} 又可以从第 i 位减法计算第二步的结果得出. 因此, 第 i 位运算中第一个表格的输出可作为第二个表格的输入, 而第二个表格的输出可作为第 $i+1$ 运算第一个表格的输入.

自组装反应将根据这个特征来设计流程. 对于第 i 位被减数, 根据图 4 中输出条件组合的不同取值给出约束条件; 对于第 $i+1$ 位减数, 根据图 3 中输出条件组合的不同取值给出约束条件. 将这些约束条件以真值表形式列出如表 3 所示.

对于第 0 位, 由于 C_0 和 S_1 都为空值, 则第 0 位上对于 A_0 的不同取值各包含 1 种约束条件, 中间结果 H_0 等于 A_0 , 且不发生下一位上的借位, 因此这一位的输出结果 H_0 与 C_1 的组合只有两种情况, 故对于 B_0 的不同取值各包含 2 种约束条件. 而对于 $i=1$ 时, 令上一步的输出为约束条件, S_0C_1 有 00, 10, 11 三种组合, 故 A_1 的不同取值分别包含 3 种约束条件. 由于每位的运算规则不变, 当 $2 < i \leq n$ 时, 对于 A_i 的每个取值包含 4 种约束条件, B_i 包含 3 种约束条件.

1.2 DNA 单链库的设计及反应原理

根据上一节的讨论, 对于 A_i 和 B_i 的每个取值都包含有若干约束条件, DNA 单链库的设计可依照在给定取值的一位情况下, 每条单链表示一种约束条件的原则进行.

对于被减数的第 i 位值, 由上一节讨论知可将 $S_{i-1}C_i$ 视为约束条件, 此处以单链 $\underline{S_{i-1}C_i[xy]}-A_i[z]-H_iC_{i+1}[wt]$ 表示这个约束条件; 对于减数的第 i 位值, 由上一节讨论知可将 H_iC_{i+1} 视为约束条件, 此处以单

表 1 被减数、借位与中间结果的对应表

A_i	C_i	H_i	C_{i+1}
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

表 2 减数、中间结果与最终结果的对应表

B_i	H_i	C_{i+1}	S_i	C_{i+1}
0	0	0	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	1	0	0	0
1	1	1	0	1

表 3 约束条件真值表

$A_0=0$	S_{-1}	C_0	H_0	C_1
	Null	Null	0	0
$B_0=0$	H_0	C_1	S_0	C_1
	0	0	0	0
	1	0	1	0
$A_1=0$	S_0	C_1	H_1	C_2
	0	0	0	0
	1	0	0	0
	1	1	1	1
$B_1=0$	H_1	C_2	S_1	C_2
	0	0	0	0
	1	0	1	0
	1	1	1	1
$A_2=0$	S_1	C_2	H_2	C_3
	0	0	0	0
	0	1	1	1
	1	0	0	0
	1	1	1	1
$B_2=0$	H_2	C_3	S_2	C_3
	0	0	0	0
	1	0	1	0
	1	1	1	1

$A_0=1$	S_{-1}	C_0	H_0	C_1
	Null	Null	1	0
$B_0=1$	H_0	C_1	S_0	C_1
	0	0	1	1
	1	0	0	0
$A_1=1$	S_0	C_1	H_1	C_2
	0	0	1	0
	1	0	1	0
	1	1	0	0
$B_1=1$	H_1	C_2	S_1	C_2
	0	0	1	1
	1	0	0	0
	1	1	0	1
$A_2=1$	S_1	C_2	H_2	C_3
	0	0	1	0
	0	1	0	0
	1	0	1	0
	1	1	0	0
$B_2=1$	H_2	C_3	S_2	C_3
	0	0	1	1
	1	0	0	0
	1	1	0	1

...

$2 < i \leq n$ 时, A_i, B_i 的约束条件与 A_2, B_2 相同

链 $H_iC_{i+1}[xy]-B_i[z]-S_iC_{i+1}[wt]$ 表示这个约束条件. 其中下标 i 表示第 i 位, $[]$ 内则表示该位上的取值(0 或 1), 下划线表示互补序列(设计互补的目的是为使反应自发进行). 其中, 被减数单链中 $H_iC_{i+1}[wt]$ 段的内容通过查表由 $S_{i-1}C_i[xy]-A_i[z]$ 段决定, 减数单链中 $S_iC_{i+1}[wt]$ 段的内容则通过查表由 $H_iC_{i+1}[xy]-B_i[z]$ 决定. 例如, 单链 $H_2C_3[10]-B_3[1]-S_3C_4[00]$, 对应 $B_3=1$, 其约束条件是 $H_2=1, C_3=0$, 通过查表知结果 $S_3=0$, 后一位上的借位 $C_4=0$, 以 $S_3C_4[00]$ 编码序列表示并连接在之后.

为了 PCR 扩增方法能够读出最终结果, 令表示 A_0 的单链的 $S_{-1}C_0[xy]$ 部分替换为开始引物 PrimerL, 而在结束组连接上结束引物 PrimerR, 这样, 在反应结束后以 PrimerL 和 PrimerR 为引物进行扩增即可得到最终结果.

根据运算的总位数, 对于每一不同的位置下标都以 m 个碱基的长度的 DNA 序列表示. 根据约束条件, 当 $i \geq 1$ 时, 减数各个位所代表的约束条件组中表示数值的编码固定, 当 $i \geq 2$ 时, 被减数各个位所代表的约束条件组中表示数值的编码固定, 不同位置上仅是表示位置的编码改变. 因此, 当 $i \geq 1$ 时, 减数具有统一的表达形式, 当 $i \geq 2$ 时, 被减数具有统一的表达形式, 单链库中所有的单链元素图 3 所示.

该单链库一经生成, 即已包含了减法了运算规则, 可反复使用. 每次运算时无需重新考虑各位的运算限制, 只需从单链库中选取相应的单链集混合反

应即可. 单链库中单链自发组合形成的链条具有如图 4 的结构形式.

考虑到需对最终结果进行 PCR 扩增, 同时建立如图 5 所示的运算数值互补单链库, 其可以填补单链库自组装反应的空缺段, 使得最终反应结果可以以如图 6 所示结构的双链形式存在.

1.3 对于被减数小于减数情况的处理

减法区别于加法运算的一点是被减数小于减数情况下的处理. 本算法中并不对被减数与减数的大小比较做特殊处理, 而是设置 C_n 这一符号位, 以区分结果是 $A \geq B$ 所得还是 $A < B$ 所得.

令 A'_i 表示 A 的最低 i 位所表示的二进制数, 即

$$A'_i = \sum_{j=0}^{i-1} 2^j \times A_j, \quad (1)$$

B'_i 意义相同.

在减法进行过程中, 存在一个基本事实: 若 $C_i=0$, 则说明 A 的最低 i 位减去 B 的最低 i 位够减, 即 $A'_i \geq B'_i$, 反之则说明 $A'_i < B'_i$. 因此, 虽然运算针对 n 位二进制数进行, 在其过程中不但包括 $C_0, C_1, \dots, C_{n-1}$ 等 n 位借位信息, 还包括一位扩展借位信息 C_n (扩展指此位已超出 n 位, A_n, B_n 不存在), $C_n=0$ 表示 $A \geq B$, $C_n=1$ 表示 $A < B$, 被减数与减数的大小比较情况可以从 C_n 上得出.

$A_0 = 0$	$A_1 = 0$	$A_i = 0$	$B_i = 0$
$\{5, \overline{\text{PrimerL}} A_0[0] H_0C_1[00]_3, \}$	$\{5, \overline{S_0C_1[00]} A_1[0] H_1C_2[00]_3, \}$	$\{5, \overline{S_{i-1}C_i[00]} A_i[0] H_iC_{i+1}[00]_3, \}$	$\{3, \overline{H_iC_{i+1}[00]} B_i[0] S_iC_{i+1}[00]_5, \}$
$A_0 = 1$	$\{5, \overline{S_0C_1[10]} A_1[0] H_1C_2[00]_3, \}$	$\{5, \overline{S_{i-1}C_i[01]} A_i[1] H_iC_{i+1}[11]_3, \}$	$\{3, \overline{H_iC_{i+1}[10]} B_i[0] S_iC_{i+1}[10]_5, \}$
$\{5, \overline{\text{PrimerL}} A_0[1] H_0C_1[10]_3, \}$	$\{5, \overline{S_0C_1[11]} A_1[0] H_1C_2[11]_3, \}$	$\{5, \overline{S_{i-1}C_i[10]} A_i[1] H_iC_{i+1}[00]_3, \}$	$\{3, \overline{H_iC_{i+1}[11]} B_i[0] S_iC_{i+1}[11]_5, \}$
		$\{5, \overline{S_{i-1}C_i[11]} A_i[1] H_iC_{i+1}[11]_3, \}$	$B_i = 1$
$B_0 = 0$	$A_1 = 1$	$A_i = 1$	$\{3, \overline{H_iC_{i+1}[00]} B_i[1] S_iC_{i+1}[11]_5, \}$
$\{3, \overline{H_0C_1[00]} B_0[0] S_0C_1[00]_5, \}$	$\{5, \overline{S_0C_1[00]} A_1[1] H_1C_2[10]_3, \}$	$\{5, \overline{S_{i-1}C_i[00]} A_i[1] H_iC_{i+1}[10]_3, \}$	$\{3, \overline{H_iC_{i+1}[10]} B_i[1] S_iC_{i+1}[00]_5, \}$
$\{3, \overline{H_0C_1[10]} B_0[0] S_0C_1[10]_5, \}$	$\{5, \overline{S_0C_1[10]} A_1[1] H_1C_2[10]_3, \}$	$\{5, \overline{S_{i-1}C_i[01]} A_i[1] H_iC_{i+1}[00]_3, \}$	$\{3, \overline{H_iC_{i+1}[11]} B_i[1] S_iC_{i+1}[01]_5, \}$
$B_0 = 1$	$\{5, \overline{S_0C_1[11]} A_1[1] H_1C_2[00]_3, \}$	$\{5, \overline{S_{i-1}C_i[10]} A_i[1] H_iC_{i+1}[10]_3, \}$	End group
$\{3, \overline{H_0C_1[00]} B_0[1] S_0C_1[11]_5, \}$		$\{5, \overline{S_{i-1}C_i[11]} A_i[1] H_iC_{i+1}[00]_3, \}$	$\{5, \overline{S_nC_{n+1}[00]} \text{PrimerR}_3, \}$
$\{3, \overline{H_0C_1[10]} B_0[1] S_0C_1[00]_5, \}$			$\{5, \overline{S_nC_{n+1}[01]} \text{PrimerR}_3, \}$
			$\{5, \overline{S_nC_{n+1}[10]} \text{PrimerR}_3, \}$
			$\{5, \overline{S_nC_{n+1}[11]} \text{PrimerR}_3, \}$

图 3 DNA 单链库设计

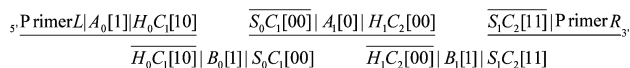


图 4 单链库自组装形式示例

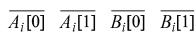


图 5 运算数值互补单链库

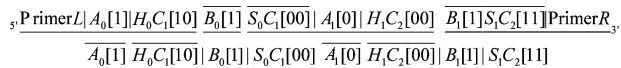


图 6 最终结果双链示例

对于 $A < B$ 的情况, 其被减数单链库与减数单链库均与 $A \geq B$ 的情况相同, 即: $A < B$ 时也按照 $A \geq B$ 时的运算规则进行. 由于 C_n 的存在, 此时在实际意义上相当于进行的是 $2^n + A - B$ 的运算, 即: $A - B$ 模 2^n 的结果. 因此, 无需特意设置被减数与减数的顺序, 按照相同的运算规则即可得出减法模运算的结果.

1.4 DNA 减法反应步骤

整个减法运算分为如下 5 个步骤进行:

- 根据给定的被减数和减数每一位的数值, 从单链库中选取各个位代表的 DNA 约束链组;
- 将 i) 中选取的单链按照适当浓度在试管中混合, 同时加以运算数值互补单链库, 在适当的实验环境中, 如适宜的温度下等, 发生线性自组装反应;
- 以 T4 DNA 连接酶对自组装链的缺口进行修补;

iv) 以 PrimerL 和 PrimerR 为引物, 以得出最终解 DNA 双链为目标, 通过 PCR 进行扩增;

v) 扩增后对产物进行测序, 按位读出减法结果的每一位值.

1.5 算法的正确性和实验室验证

根据上文对减法约束条件的分析, 在减法运算的过程中, 对于每一位运算数值与当前借位信息的组合, 都对应着一个包含下一步的借位信息、中间结果、最终结果的组合. 该对应以约束条件表现在参与运算数的 DNA 单链中, 且约束条件以互补形式反映在按位运算的单链上. 因此, 根据 DNA 的碱基互补原则, 算法在理论上的运算过程将严格遵循互补原则, 亦即严格遵循图 3 所示的减法约束条件, 从而得出了算法在运算逻辑上的正确性.

在算法实验室的可行性方面, 对于线性自组方法设计算法的过程, 最本质的部分是单链库的设计, 单链库设计不同决定了实验目的不同. 在单链库设计之外, 其他步骤已有通用性的实践模板. 因此, 对于本文提出算法的实验步骤设计, 只需替换相似规模实验中的单链库即可.

在以往的 DNA 计算研究中, 赵健等提出了基于线性自组装的 DNA 加法并设计实验验证了算法的正确性^[21], 其算法复杂度、单链库规模与本文提出算法类似. 在本算法的实施中, 只需将赵健等提出的实验^[21]中基于加法规则的单链库替换为本文图 3 所示的

基于减法规则的单链库,即可使用同样的实验步骤进行实验室验证.赵健等成功验证其提出的算法在进行4位数加法正确性的实验步骤如下^[21]:

(1) 混合:将表示给定运算数值的DNA单链在从DNA库中选出,在同一个试管中混合,每条DNA单链的浓度为50 pmol.除了 A_0 对应的DNA单链以外,所有的DNA单链都在5'端加了磷酸,为了实现之后的连接反应.

(2) 杂交:室温杂交30 min,序列互补的DNA单链会自发的同时的自组装在一起.

(3) 连接:T4 DNA连接酶室温连接1 h.

(4) PCR:用磁珠吸附连接产物(A_0 对应的DNA单链的5'端有生物素标记)进行纯化,将纯化后的连接产物以PrimerL和PrimerR为引物做PCR.

(5) 测序:将PCR产物测序.根据该序列查找对应的表示结果的段,即可得到加法结果.

由以上实验步骤可知,本文算法与其算法的本质差异——DNA单链库的设计,在整个实验过程中并不反映在实验步骤的差异上,无论是混合、杂交或连接、PCR扩增、测序,都是通用性的生物实验方法,在实验规模相同的条件下,并不受实验材料碱基序列如何编排的影响.

此处容易看出,实验中实施整个算法按位计算的流程从而获得实验结果的关键在于步骤(2),序列互补的DNA单链自发自组装从而得出正确的反应结果——而此步骤中反应结果的正确性是由该DNA单链库设计在加法约束条件逻辑上的正确性决定的.而根据上文对本文提出的算法在减法约束条件逻辑上的正确性的说明可知,若将实验中的单链库替换成本文的单链库,反应亦将得出正确的结果.同样的,这是由算法在按位进行减法运算约束条件逻辑上的正确性决定的.因此,本文的算法可以利用赵健等的实验流程^[21]进行实验室验证.

2 范例

此处以1101-0111为例,说明本算法中单链库的形式以及反应结果.

将表示 $A_3=1, A_2=1, A_1=0, A_0=1, B_3=0, B_2=1, B_1=1, B_0=1$ 的8组单链从DNA库中挑选出,如图7所示.

加之图5所示的运算数值互补单链库,按照碱基互补原则自组装反应后,其中最长的链,亦即以PrimerL引物开头、PrimerR引物结尾的链既为计算结果,如图8所示.

从该链中读出最终结果: $S_0=0, S_1=1, S_2=1, S_3=0$,

$A_0 = 1$	$A_1 = 0$	$A_2 = 1$	$A_3 = 1$
$\{ \text{PrimerL} A_0[1] H_0C_1[10] \}_3$	$\{ \overline{S_0C_1[00]} A_1[0] H_1C_2[00] \}_3$	$\{ \overline{S_1C_2[00]} A_2[1] H_2C_3[10] \}_3$	$\{ \overline{S_2C_3[00]} A_3[1] H_3C_4[10] \}_3$
	$\{ \overline{S_0C_1[10]} A_1[0] H_1C_2[00] \}_3$	$\{ \overline{S_1C_2[01]} A_2[1] H_2C_3[10] \}_3$	$\{ \overline{S_2C_3[01]} A_3[1] H_3C_4[00] \}_3$
	$\{ \overline{S_0C_1[11]} A_1[0] H_1C_2[11] \}_3$	$\{ \overline{S_1C_2[10]} A_2[1] H_2C_3[10] \}_3$	$\{ \overline{S_2C_3[10]} A_3[1] H_3C_4[10] \}_3$
		$\{ \overline{S_1C_2[11]} A_2[1] H_2C_3[00] \}_3$	$\{ \overline{S_2C_3[11]} A_3[1] H_3C_4[00] \}_3$
$B_0 = 1$	$B_1 = 1$	$B_2 = 1$	$B_3 = 0$
$\{ \overline{H_0C_1[00]} B_0[1] S_0C_1[11] \}_3$	$\{ \overline{H_1C_2[00]} B_1[1] S_1C_2[11] \}_3$	$\{ \overline{H_2C_3[00]} B_2[1] S_2C_3[11] \}_3$	$\{ \overline{H_3C_4[00]} B_3[0] S_3C_4[00] \}_3$
$\{ \overline{H_0C_1[10]} B_0[1] S_0C_1[00] \}_3$	$\{ \overline{H_1C_2[10]} B_1[1] S_1C_2[00] \}_3$	$\{ \overline{H_2C_3[10]} B_2[1] S_2C_3[00] \}_3$	$\{ \overline{H_3C_4[10]} B_3[0] S_3C_4[10] \}_3$
	$\{ \overline{H_1C_2[11]} B_1[1] S_1C_2[01] \}_3$	$\{ \overline{H_2C_3[11]} B_2[1] S_2C_3[01] \}_3$	$\{ \overline{H_3C_4[11]} B_3[0] S_3C_4[11] \}_3$
			$\{ \overline{S_3C_4[00]} \text{PrimerR} \}_3$
			$\{ \overline{S_3C_4[01]} \text{PrimerR} \}_3$
			$\{ \overline{S_3C_4[10]} \text{PrimerR} \}_3$
			$\{ \overline{S_3C_4[11]} \text{PrimerR} \}_3$

图7 1101-0111为例,从减法单链库中挑选出的单链集

$$\text{PrimerL} | A_0[1] | H_0C_1[10] | B_0[1] | \overline{S_0C_1[00]} | A_1[0] | H_1C_2[00] | B_1[1] | \overline{S_1C_2[11]} | A_2[1] | H_2C_3[00] | B_2[1] | \overline{S_2C_3[11]} | A_3[1] | H_3C_4[00] | B_3[0] | \overline{S_3C_4[00]} | \text{PrimerR}$$

$$A_0[1] | H_0C_1[10] | B_0[1] | \overline{S_0C_1[00]} | A_1[0] | H_1C_2[00] | B_1[1] | \overline{S_1C_2[11]} | A_2[1] | H_2C_3[00] | B_2[1] | \overline{S_2C_3[11]} | A_3[1] | H_3C_4[00] | B_3[0] | \overline{S_3C_4[00]}$$

图8 表示1101-0111计算结果的链

可知结果为0110, 由 $C_4=0$ 知所进行的是 $A>B$ 的运算, 且 $A-B=0110 \pmod{16}$, 这与逻辑运算的正确结果是一致的.

3 算法的性质

性质 1: 给定 n 位($n>2$)的运算数值 A 与 B , 进行反应的单链数值是 $O(n)$ 的.

说明: 根据图 5 中的约束条件真值表, A_0 的给定数值产生 1 条链, B_0 的给定数值产生 2 条链; A_1 的给定数值产生 3 条链, B_1 的给定数值产生 3 条链; 当 $i>1$ 时, A_i 的给定数值产生 4 条链, B_i 的给定数值产生 3 条链, 结束组包含 4 条链. 故反应单链的总数为 $9+7(n-2)+4=7n-1$ 条单链. 参加反应的单链总数是 $O(n)$ 的.

性质 2: 给定 n 位($n>2$)的运算数值 A 与 B , 进行反应直至产生结果中的人为参与生物操作步骤是常数.

说明: 在挑选出参加反应的单链后, 每条单链都表示一种约束条件, 表示正确结果链的组成部分单链必有互补段, 因此整个反应过程将通过自组装自发进行, 其中可能的人为参与步骤仅为调节实验进行中的温度, 进行 PCR 扩增等, 与位数 n 无关. 因此, 对于位数 n , 人为参与步骤是常数.

性质 3: 算法不仅可以进行两数相减运算, 还能对两个 n 位二进制数集合进行两两相减的并行运算, 得出被减数集与减数集中任意组合模 2^n 的差.

说明: 令 A 表示被减数集, 令 B 表示减数集, 将表示 A, B 中所有数值的单链分别从图 6 表示的被减数与减数单链库中选出若干个拷贝, 加之结束组的若干个拷贝, 按照适当浓度混合, 基于 DNA 的并行反应特性, 将得出 $|A|*|B|$ 个运算结果, 其结果链集合表示的数值以集合 S 表示:

$$S = \{s \mid s = x - y \pmod{2^n}, x \in A, y \in B\}, \quad (2)$$

其中通过读取结果链中 C_n 段上的信息, 可知计算中被减数与减数的大小关系.

性质 4: 实验中能够进行的运算位数与编码的关系.

说明: 在将数值按照约束条件编码成单链时, 例如图 9 所示. 其中 $S_{i-1}C_iA_iH_iC_{i+1}$ 的编码包含两部分信息, 一部分标识当前段所代表的元素, 如被减数、中间结果等; 另一部分表示位置标识, 整个反应中所能进行的计算位数由表示位置标识的长度而定.

$$\{\overline{S_{i-1}C_i[00]} \mid A_i[0] \mid H_iC_{i+1}[00]_3\}$$

图 9 单链编码构成示例

以 m 位碱基序列表示每个位置标识, 则整个过程能进行计算位数为 4^m 位. 例如, 以 20 位碱基序列表示每个位置标识, 则能够计算 4^{20} , 既 2^{40} 位二进制数的减法计算.

4 结论

本文提出的 DNA 减法模运算算法, 其具有以下特性:

(1) 整体编码方案一致. 不再对被减数与减数的大小进行分类而增加冗余.

(2) 设置扩展借位信息, 以扩展借位信息为被减数与减数大小判断的标志, 并在结果中包含这一标志.

(3) 自发反应性强. 反应中人为生物操作不随运算位数而增长. 反应结束后对包含开始引物与结束引物的链进行扩增测序即可得所需结果.

(4) 充分利用 DNA 反应的并行特性. 算法不仅可以进行两个数的减法模运算, 还能对指定的被减数集合与减数集合的任意元素两两进行减法模运算, 得出遍历反应结果.

DNA 因其巨大的并行反应特性, 作为一种计算材料可以具有相当的计算潜力. 若在计算中能够充分利用 DNA 并行特性, 将可以得到较为可观的并行计算能力. 目前, DNA 计算理论尚未形成统一完备的体系, 但其强大的并行反应特性所带来的并行计算能力在本文提出的算法中已有体现. 此领域的研究在未来必将有更大的发展潜力.

参考文献

- 1 Adleman L M. Molecular computation of solutions to combinatorial problems. Science, 1994, 266: 1021—1024
- 2 Kari L. DNA computing: Arrival of biological mathematics. Math Intell, 1997, 19: 9—22
- 3 Praun G, Rozenberg G, Salomaa A. DNA Computing—New Computing Paradigms. Berlin: Springer-Verlag, 1998

- 4 俞洋, 缪淮扣, 宋世平, 等. DNA 分子计算与 DNA 计算机的研究进展. 科学通报, 2008, 53: 497—502
- 5 Roweis S, Winfree E, Burgoyne R, et al. A sticker based architecture for DNA computation. In: Landweber L F, Baum E B, eds. DNA Based Computers II, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Providence: American Mathematical Society, 1996. 1—29
- 6 Kari L, Paun G, Rozenberg G, et al. DNA computing, sticker systems, and universality. Acta Inform, 1998, 35: 401—420
- 7 许进, 董亚非, 魏小鹏. 粘贴 DNA 计算机模型(I): 理论. 科学通报, 2004, 49: 205—212
- 8 许进, 李三平, 董亚非, 等. 粘贴 DNA 计算机模型(II): 应用. 科学通报, 2004, 49: 299—307
- 9 Păun G, Salomaa A. DNA Computing based on the splicing operation. Math Japo, 1996, 43: 607—632
- 10 Păun G, Freund R, Kari L. DNA Computing based on splicing: The existence of universal computers. Theor Comput Syst, 1999, 32: 69—112.
- 11 Ouyang Q, Kaplan P D, Liu S, et al. DNA solution of the maximal clique problem. Science, 1997, 278: 446—449
- 12 Braich R S, Chelyapov N, Johnson C, et al. Solution of a 20-variable 3-SAT problem on a DNA computer. Science, 2002, 296: 499—502
- 13 许进, 强小利, 方刚, 等. 一种图顶点着色 DNA 计算机模型. 科学通报, 2006, 51: 480—487
- 14 肖国镇, 卢明欣, 秦磊, 等. 密码学的新领域——DNA 密码. 科学通报, 2006, 51: 1139—1144
- 15 卢明欣, 来学嘉, 肖国镇, 等. 基于 DNA 技术的对称加密方法. 中国科学 E 辑: 信息科学, 2007, 37: 175—182
- 16 Guarnieri F, Fliss M, Bancroft C, et al. Making DNA add. Science, 1996, 273: 220—223
- 17 Gupta V, Parthasarathy S, Zaki M J. Arithmetic and logic operations with DNA. In: Rubin H, Wood D H, eds. DNA Based Computer III, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Providence: American Mathematical Society, 1997. 212—220
- 18 Wasiewicz P, Mulawka J J, Rudnicki W R, et al. Adding numbers with DNA. In: Proceedings of 2000 IEEE International Conference on Systems, Man and Cybernetics, 2000 Oct 8—11, Nashville. Tennessee: IEEE Press, 2000, 1. 265—270
- 19 Winfree E, Liu F, Wenzler L A, et al. Design and self-assembly of two-dimensional DNA crystals. Nature, 1998, 394: 539—544
- 20 Rothmund P W K, Papadakis N, Winfree E. Algorithmic self-assembly of DNA Sierpinski triangles. PLoS Biol, 2004, 2: 2041—2053
- 21 赵健, 钱璐璐, 刘强, 等. 基于线性自组装的 DNA 加法. 科学通报, 2006, 51: 2485—2489