基于剪接系统的有向哈密顿路问题分析

王淑栋[1,2], 许 进[2], 刘文斌[2]
(1. 山东科技大学信息科学与工程学院, 山东泰安 271019; 2. 华中科技大学控制科学与工程系, 湖北武汉 430074)

摘 要: 首先给出了剪接系统模拟有向哈密顿路问题的思想; 然后通过此剪接系统所产生语言的性质对有向哈密顿路问题进行分析, 给出了有向图存在哈密顿路的充要条件. 在我们的构造中, 模拟问题的剪接系统至多运行 n-2 步, 其中 n 是模拟问题的规模.

关键词: DNA 计算; 剪接系统; 有向哈密顿路问题
中图分类号: TP18 文献标识码: A 文章编号: 0372-2112 (2005) 05-0774-04

Analysis for Directed Hamilton Path Problems Based on Splicing Systems

WANG Shu-dong[1,2], XU Jin[2], LIU Wen-bin[2]
(1. *College of Information Science & Engineering, Shandong University of Science and Technology, Taian, Shandong 271019, China;*
2. *Dept of Control Science & Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China*)

Abstract: The ideas of simulating directed Hamilton path problems by splicing systems are showed; then some properties of directed graphs are presented based on the analysis of directed Hamilton path problems according to the properties of languages generated by the splicing systems. In our construction, the splicing systems simulating problems run at most n-2 steps, where n is the size of problems.

Key words: DNA computing; splicing system; directed Hamilton path problem

1 引言

目前, 科学家研究的 DNA 计算形式模型有很多类, 例如: 粘贴系统, 剪接系统, 等量检测系统和插入删除系统等, 其中粘贴系统和剪接系统是最常见的 DNA 计算模型. 粘贴系统模型是建立在粘贴运算基础上的 DNA 计算模型, 而剪接系统是基于剪接运算上的 DNA 计算模型, 它切割双链 DNA 分子并在切割处连接不同 DNA 分子的切割部分形成一个新的 DNA 分子. 1987 年, Tom Head 在文献[1]中引入了剪接的概念, 并利用形式语言理论分析了 DNA 分子重组模型的生成能力. 1992 年, 他又利用形式语言技术给出了分子计算的抽象模型 - 剪接系统[2]. 之后, 有关剪接系统大部分工作重点还是试验技术基础上的形式语言理论. 1996 年, E. Csuhaj-Varju, L. Freund, L. Kari 和 Gh. Păun 在夏威夷举行的第一届生物计算讨论会上作了《剪接基础上的 DNA 计算》的报告[3]. 同年, Gh. Păun 和 A. Salomaa 从递归可枚举语言的特点出发, 进一步论证了剪接运算基础上 DNA 计算的可能性[4]. 1999 年, R. Freund, L. Kari 和 Gh. Păun 论述了剪接系统上的 DNA 计算, 并证明了通用 DNA 计算机的存在性[5]. 2001 年, Y. Benenson 等人通过剪接系统构造了一台可编程的有穷自动机, 这充分说明了剪接系统是计算完备的, 也就是说每个 PASCAL 程序都可以通过合适的剪接系统来模拟, 反之亦然. 另外, 他还证明了通用剪接系统的存在性, 即剪接运算基础上的通用可编程的 DNA 计算机的存在性[6]. 直到现在, 有关剪接系统的研究主要集中在它所产生的形式语言的性质(如封闭性)及其与乔姆斯基体系中其它形式语言的关系. 在文献[7]中, Katrin Erk 利用剪接系统的高度并行性, 将布尔电路作为剪接系统的词进行模拟, 在有穷的时间内得到了布尔电路的值.

2 剪接系统

数学上, 对在限制性内切酶、DNA 连接酶、DNA 聚合酶和外切酶作用下, DNA 链进行重组过程抽象为剪接运算. 剪接系统就是将剪接运算当作基本算子的一种语言生成器, 也是对下述生物过程的一种建模: 酶对 DNA 分子中的特定片段进行切割, 然后再对所切割的片段进行重新连接.

下面我们给出剪接系统的定义.

设 V 是一个有穷字母表, 其中字符 $\#, \$ \notin V$.

V 上的剪接规则是形为 $r = u_1\#u_2\$u_3\#u_4$ 的词, 其中 $u_1, u_2, u_3, u_4 \in V^*$. 对于剪接规则 $r = u_1\#u_2\$u_3\#u_4$ 和词 $v, w \in V^*$, 如果存在词 $v', v'', w', w'' \in V^*$ 使得 $v = v'u_1u_2v''$,

收稿日期: 2004-05-14; 修回日期: 2004-12-10
基金项目: 国家自然科学基金(No. 60174047, No. 60274026, No. 60403002)

$w = w' u_3 u_4 w''$, 那么可以将规则 r 应用到 v, w 上, 并产生两个新词 $\overline{v} = v' u_1 u_4 w''$ 和 $\overline{w} = w' u_3 u_2 v''$, 将这种剪接过程记为 $(v, w) \vDash r(\overline{v}, \overline{w})$. 词 v, w 称为剪接项. 在已知上下文的情况下可以省略 r, 用 $\vDash$ 代替 $\vDash r$. 通过剪接规则 r, 从 v, w 到 $\overline{v}$, $\overline{w}$ 的传递过程如图 1 所示. 我们将系统中不能进行进一步剪接运算的词称为垃圾词.

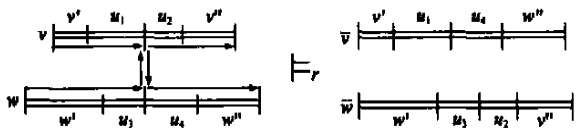

图 1　剪接运算

字母表 V 上的剪接系统是一个三元组 $S = (V, I, R)$, 其中 $I \subseteq V^*$ 是初始语言, $R \subseteq V^* \# V^* \$ V^* \# V^*$ 是剪接规则集. 将剪接规则 $u_1 \# u_2 \$ u_3 \# u_4$ 记为 $(u_1, u_2; u_3, u_4)$.

$\sigma = (V, R)$ 称为 H 方案, 其中 V 是一个字母表, $R \subseteq V^* \# V^* \$ V^* \# V^*$ 是剪接规则集. 对一个 H 方案 $\sigma = (V, R)$ 和一个语言 $I \subseteq V^*$, 定义

$\sigma^0(I) = I$,

$\sigma^1(I) = \sigma(I) = \{w \in V^* \mid (x, y) \vDash r(w, z)$ 或 $(x, y) \vDash r(z, w)$, 其中 $x, y \in I, r \in R\}$,

$\sigma^{i+1}(I) = \sigma^i(I) \cup \sigma(\sigma^i(I)), i \geqslant 0$,

并且 $\sigma^*(I) = \bigcup_{i \geqslant 0} \sigma^i(I)$.

令 $k = \min\{i \mid \sigma^i(I) = \sigma^{i+1}(I) = \sigma^*(I)\}$, 我们称剪接系统运行了 k 步停止. 这样, 剪接系统 $S = (V, I, R)$ 产生的语言 $L(S) = \sigma^k(I) = \sigma^*(I)$. 显然 $L(S)$ 是最小集 L 使得 $I \subseteq L$, 并且如果 $v, w \in L$ 且 S 中存在一个剪接规则 r 使得 $(v, w) \vDash r(\overline{v}, \overline{w})$, 那么 $\overline{v}, \overline{w} \in L$.

在实际生物操作中, 剪接系统中有穷字母表 V 上的初始语言对应着初始试管中的 DNA 分子. 如果初始语言的两个词进行了剪接运算, 那么初始试管中这两个词对应的两个 DNA 分子就在酶的作用下进行切割和连接等重组操作. 剪接规则对应着生物操作中的限制性内切酶、DNA 连接酶、DNA 聚合酶和外切酶等. 因为酶可以对试管中所有包含特定片段的 DNA 分子同时进行切割和连接运算, 所以在生物操作的意义下, 剪接系统具有并行性. 在剪接系统中, 语言中的词模拟 DNA 分子, 允许在两个词的前缀和后缀间进行交换的剪接规则模拟重组过程, 垃圾词对应的 DNA 分子中没有酶识别的特定片段.

3　有向哈密顿路问题

设 $G = (V, E)$ 是一个简单有向图, 其中 $V = \{v_1, v_2, \cdots, v_n\}$ 是 G 的顶点集, $E = \{v_iv_j \mid v_i$ 与 v_j 相邻, $v_i, v_j \in V\}$ 是 G 的边集. 顶点 v 的出度等于 G 中从 v 到其它顶点的边数, 其入度等于从其它顶点到 v 的边数. 恰好经过 G 中所有顶点一次的路叫做有向哈密顿路, 有向哈密顿路问题就是在 G 中寻找有向哈密顿路的问题, 这是图论中的一个 NP-完全问题. 到目前为止, 图论专家还没有给出一个较好的求解有向哈密顿路问题的充要条件. 在电子计算机上解决有向哈密顿路问题的所需时间随着图规模的扩大呈指数增长. 1994 年, Adleman[8] 突破了传统计算机结构体系的束缚, 开创性地提出用分子生物技术解决有向哈密顿路问题, 将所需时间由以前的指数时间缩短到多项式时间, 并论述了分子计算的高密度、高并行性的潜在能力. 本文利用剪接系统的高度并行性, 对有向哈密顿路问题进行了模拟; 然后通过该剪接系统所产生语言的性质对有向哈密顿路问题进行分析, 给出了一个有向图存在哈密顿路的充要条件. 在我们的构造中, 模拟问题的剪接系统至多运行 $n-2$ 步停止, 其中 n 是模拟问题的规模.

4　基于剪接系统的有向哈密顿路问题分析

设 $G = (V', E)$ 是一个简单有向图, 其中 $V' = \{v_1, v_2, \cdots, v_n\}$ 是 G 的顶点集, $E = \{v_iv_j \mid v_i$ 与 v_j 相邻, $v_i, v_j \in V\}$ 是 G 的边集. 我们用剪接系统 $S = (V, I, R)$ 模拟 G 的哈密顿路问题, 其中 $V = \{Y, v_1, v_2, \cdots, v_n\}$, $I = \{v_iv_jY \mid v_iv_j \in E\}$, Y 是标记, $R = (T_1v_i, Y; v_i, T_2)$, 其中 T_1 与 T_2 不含相同的字符, 即不存在字符 $v \in V'$ 使得 v 是 T_1 和 T_2 的子词且 $T_1 \neq \varepsilon, T_2 \neq Y$.

由上述构造, 我们得到主要结果如下:

定理 1　$L(S)$ 中词的长度不超过 $n+1 (n \geqslant 2)$.

证明　设 $T \in L(S)$.

若 T 是初始语言 I 中的词, 则 $|T| = 3 \leqslant n+1$.

若 T 是经过剪接运算得到的词, 则 $|T| = 2$ 或者 $|T| \geqslant 4$. 显然有 $|T| = 2 \leqslant n+1$.

下面我们讨论 $|T| \geqslant 4$ 的情况.

假设 $|T| = l \geqslant n+2$. 由剪接规则的构造知: $L(S)$ 中的所有词都以标记 Y 结尾. 设 $T = T_1Y$, $T_1 \in \{v_1, v_2, \cdots, v_n\}^*$. 因为 $|T_1| \geqslant n+1$, 所以 T_1 中含有两个相同的字符, 设为 v_m, 即 T_1 可以写成 $v_{i_1}v_{i_2}\cdots, v_{i_j}v_mv_{i_{j+1}}\cdots v_{i_k}v_mv_{i_{k+1}}\cdots v_{l-3}$, 所以 $T = v_{i_1}v_{i_2}\cdots v_{i_j}v_mv_{i_{j+1}}\cdots v_{i_k}v_mv_{i_{k+1}}\cdots v_{l-3}Y$, 这样在 T 形成的剪接过程中一定存在剪接运算

$(v_{i_1}v_{i_2}\cdots v_{i_j}v_mv_{i_{j+1}}\cdots v_{i_{j+p}} \mid Y,\ U_{i_{j+p}} \mid v_{i_{j+p+1}}\cdots v_{i_k}v_mv_{i_{k+1}}\cdots v_{l-3}Y)$
$\vDash r(v_{i_1}v_{i_2}\cdots v_{i_j}v_mv_{i_{j+1}}\cdots v_{i_{j+p}}v_{i_{j+p+1}}\cdots v_{i_k}v_mv_{i_{k+1}}\cdots v_{l-3}Y, v_{i_{j+p}}Y)$.

这样 $S_1 = v_{i_1}v_{i_2}\cdots v_{i_j}v_mv_{i_{j+1}}\cdots v_{i_{j+p-1}}$ 和 $S_2 = v_{i_{j+p+1}}\cdots v_{i_k}v_mv_{i_{k+1}}\cdots v_{l-3}Y$ 就出现了相同的子词 v_m, 与所构造的剪接规则矛盾. 所以假设错误, 即 $|T| \leqslant n+1$. (注: "|" 表示剪接位点).

定理 2　$v_{i_1}v_{i_2}\cdots v_{i_m}Y$ 是 $L(S)$ 的一个词当且仅当 $v_{i_1}v_{i_2}\cdots v_{i_m}$ 是 G 中的一条有向路.

证明 (必要性)　由剪接规则的构造知: $L(S)$ 中的所有词都以标记 Y 结尾. 不妨设 $T = v_{i_1}v_{i_2}\cdots v_{i_m}Y$ 是 $L(S)$ 中一个长度为 $m+1$ 的词.

我们对 m 进行归纳.

$m = 1$ 时, 显然结论成立.

$m = 2$ 时, $T = v_{i_1}v_{i_2}Y \in I$, 即 $v_{i_1}v_{i_2} \in E(G)$, 显然结论成立.

假设 $m \leqslant k (k \geqslant 3)$ 时, $v_{i_1}v_{i_2}\cdots v_{i_m}$ 是 G 中一条有向路.

下面考虑 $m = k+1$ 的情况.

因为 $|T| \geqslant 4$, 所以 T 是通过一系列剪接运算, 设为 $r_1, r_2, \cdots, r_s (s \geqslant 1)$ 得到的. 不妨设 $r_s = (v_{i_1}v_{i_2}\cdots v_{i_{l+1}}, Y; v_{i_{l+1}}, v_{i_{l+2}}$

$\cdots v_{i_m}Y)(l\geqslant 1)$. 这说明了 $v_{i_1}v_{i_2}\cdots v_{i_l}v_{i_{l+1}}Y$ 和 $v_{i_{l+1}}v_{i_{l+2}}\cdots v_{i_m}Y$ 是 $L(S)$ 的词, 由归纳假设知: $v_{i_1}v_{i_2}\cdots v_{i_l}v_{i_{l+1}}$ 和 $v_{i_{l+1}}v_{i_{l+2}}\cdots v_{i_m}$ 分别是 G 中长度为 l 和 $m-l-1$ 的有向路, 将这两条路连接起来, 就得到了 G 中长度为 $m-1$ 的一条有向路 $v_{i_1}v_{i_2}\cdots v_{i_l}v_{i_{l+1}}\cdots v_{i_m}$.

(充分性)　设 $v_{i_1}v_{i_2}\cdots v_{i_l}v_{i_{l+1}}\cdots v_{i_m}$ 是 G 中一条有向路, 则 $v_{i_k}v_{i_{k+1}}\in E(G)$, $k=1,2,\cdots,m-1$. 这样, 我们就可以从初始语言 I 中的词 $v_{i_1}v_{i_2}$ 开始, 运用剪接规则序列 $r_1, r_2, \cdots, r_{m-2}$, 其中 $r_s=(v_{i_1}v_{i_2}\cdots v_{i_{s+1}}, Y; v_{i_{s+1}}, v_{i_{s+2}}Y)$, $(s=1,2,\cdots,m-2)$, 得到词 $v_{i_1}v_{i_2}\cdots v_{i_s}v_{i_{s+1}}\cdots v_{i_m}Y$.

推论 1　$L(S)$ 中存在长度为 $n+1$ 的词当且仅当 G 中有哈密顿路.

证明　充分性是显然的.

(必要性)　设 $T=v_{i_1}v_{i_2}\cdots v_{i_n}Y$ 是 $L(S)$ 中长度为 $n+1$ 的词. 由定理 2 知: $v_{i_1}v_{i_2}\cdots v_{i_n}$ 是 G 中的一条有向路. 又因为 v_{i_1}, v_{i_2}, $\cdots$, v_{i_n} 是 n 个互不相同的字符, 当然有 $\{v_{i_1}, v_{i_2}, \cdots, v_{i_n}\}=\{v_1, v_2, \cdots, v_n\}=V$, 所以 $v_{i_1}v_{i_2}\cdots v_{i_n}$ 是 G 中的一条有向哈密顿路.

推论 2　$L(S)$ 中长度为 $n+1$ 的词的数目即是 G 中哈密顿路的条数.

定理 3　$L(S)$ 中含有长度为 2 垃圾词 v_iY, $i\in\{1,2,\cdots,n\}$ 当且仅当 G 中顶点 v_i 的出度和入度都大于 0.

证明**:** **(必要性)**　因为初始语言 I 中所有词的长度都为 3, 所以长度为 2 的垃圾词 v_iY 一定是某些剪接运算的产物. 因为剪接规则 $R=(T_1v_i, Y; v_i, T_2)$, 其中 T_1 与 T_2 不含相同的字符, 即不存在字符 $v\in V'$ 使得 v 是 T_1 和 T_2 的子词且 $T_1\neq\varepsilon$, $T_2\neq Y$. 由其构造知: 所有长度都为 2 的垃圾词 v_iY 都是通过剪接运算② $(S_1v_i|Y, v_i|S_2Y)\vdash r(S_1v_iS_2Y, v_iY)$ 得到的, 其中 $S_1\neq\varepsilon$, $S_2\neq\varepsilon$. 也就是说存在 $v_m, v_k\in V'$ 使得 $L(S)$ 中一定含有词 $T_1=S_1'v_mv_iY$ 和 $T_2=v_iv_kS_2'Y$, 其中 $S_1=S_1'v_m$, $S_2=v_kS_2'$. 由定理 2 知: $S_1'v_mv_i$ 和 $v_iv_kS_2'$ 都是 G 中的有向路. 所以 v_mv_i, $v_iv_k\in E(G)$. 因此 G 中顶点 v_i 的出度和入度都大于 0.

(充分性)　因为 G 中顶点 v_i 的出度和入度都大于 0, 所以存在 $v_m, v_k\in V'$ 使得 $v_mv_i, v_iv_k\in E(G)$. 这样初始语言 I 中含有词 v_mv_iY 和 v_iv_kY. 因此通过剪接运算② $(v_mv_i|Y, v_i|v_kY)\vdash r(v_mv_iv_kY, v_iY)$ 得到长度为 2 垃圾词 v_iY.

定理 4　剪接系统 S 至多运行 $n-2(n\geqslant 2)$ 步就会停止.

证明　当 $n=2$ 时, G 中含有两个顶点. 剪接系统不会进行任何剪接运算.

当 $n\geqslant 3$ 时, 任取 $T=v_{i_1}v_{i_2}\cdots v_{i_m}Y\in L(S)$, 只要证明剪接系统 S 至多运行 $n-2(n\geqslant 2)$ 步就能得到 T 即可.

我们对 T 的长度进行讨论.

(1) $|T|=2$. 设 $T=v_{i_1}Y$, $i_1\in\{1,2,\cdots,n\}$. 由定理 3 知: G 中顶点 v_{i_1} 的出度和入度都大于 0. 不妨设 $v_{i_1}v_k, v_mv_{i_1}\in E(G)$, 由剪接系统中对初始语言的定义, 有 $v_{i_1}v_kY, v_mv_{i_1}Y\in I$. 这样, 通过剪接运算 $(v_mv_{i_1}|Y, v_{i_1}|v_kY)\vdash r(v_mv_{i_1}v_kY, v_{i_1}Y)$ 就可以得到 $T=v_{i_1}Y$. 因此剪接系统 S 只需进行 1 步就能得到 T.

(2) $|T|=3$. 设 $T=v_{i_1}v_{i_2}Y$, $i_1, i_2\in\{1,2,\cdots,n\}$, 且 $i_1\neq i_2$. 由剪接系统中对初始语言的定义知: $v_{i_1}v_{i_2}Y\in I$. 所以剪接系统 S 不需进行任何运算就可得到 T.

(3) $|T|\geqslant 4$. 设 $T=v_{i_1}v_{i_2}\cdots v_{i_m}Y(m\geqslant 3)$. 由定理 2 的充分性证明知: $L(S)$ 中词 T 可以从 $v_{i_1}v_{i_2}Y$ 开始, 运用剪接规则序列 $r_1, r_2, \cdots, r_{m-2}$ 得到. 由定理 1 知: $|T|\leqslant n+1$, 所以 $|T|=m+1\leqslant n+1$, 即 $m\leqslant n$. 因此剪接系统 S 至多进行 $n-2(n\geqslant 2)$ 步就可得到 T.

由上述定理可知, 对于剪接系统 S 来说, 它模拟的有向图 G 与其产生的语言 $L(S)$ 是一一对应的.

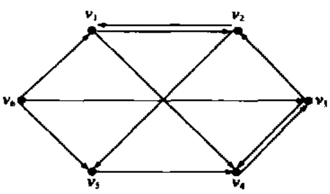

图 2　6 个顶点 11 条边的有向图

下面给出一个实例(如图 2)来说明剪接系统模拟哈密顿路问题的整个过程.

模拟图 2 的剪接系统的字母表 $V=\{v_1, v_2, \cdots, v_6, Y\}$, 初始语言 $I=\{v_iv_jY|v_iv_j\in E(G)\}$, 规则 R 同上.

第一步剪接运算:

$(v_1v_2|Y, v_2|v_5Y)\vdash r(v_1v_2v_5Y, v_2Y)$, $(v_2v_1|Y, v_1|v_4Y)\vdash r(v_2v_1v_4Y, v_1Y)$, $(v_3v_2|Y, v_2|v_1Y)\vdash r(v_3v_2v_1Y, v_2Y)$, $(v_3v_2|Y, v_2|v_5Y)\vdash r(v_3v_2v_5Y, v_2Y)$, $(v_4v_3|Y, v_3|v_2Y)\vdash r(v_4v_3v_2Y, v_3Y)$, $(v_5v_4|Y, v_4|v_3Y)\vdash r(v_5v_4v_3Y, v_4Y)$, $(v_6v_5|Y, v_5|v_4Y)\vdash r(v_6v_5v_4Y, v_5Y)$, $(v_6v_1|Y, v_1|v_2Y)\vdash r(v_6v_1v_2Y, v_1Y)$, $(v_6v_1|Y, v_1|v_4Y)\vdash r(v_6v_1v_4Y, v_1Y)$, $(v_6v_3|Y, v_3|v_2Y)\vdash r(v_6v_3v_2Y, v_3Y)$, $(v_6v_3|Y, v_3|v_4Y)\vdash r(v_6v_3v_4Y, v_3Y)$, $(v_1v_4|Y, v_4|v_3Y)\vdash r(v_1v_4v_3Y, v_4Y)$ $(v_2v_5|Y, v_5|v_4Y)\vdash r(v_2v_5v_4Y, v_5Y)$.

令

$I_1=\{v_iv_jv_kY|(v_iv_j|Y, v_j|v_kY)\vdash r(v_iv_jv_kY), i,j,k\in\{1,2,\cdots,6\}$, 且 i,j,k 两两不等$\}$;

$I'_1=\{v_jY|(v_iv_j|Y, v_j|v_kY)\vdash r(v_iv_jv_kY, v_jY), i,j,k\in\{1,2,\cdots,6\}$, 且 i,j,k 两两不等$\}$.

显然 I'_1 中的词不能进行进一步的剪接运算, 即为垃圾词. 这样, 经过第一步剪接运算后得到 $\sigma(I)=I\cup I_1\cup I'_1$.

根据剪接规则 $R=(S_1v_i, Y; v_i, S_2Y)$ 的不同, 第二步剪接运算可分为四类:

(1) $|S_1|=|S_2|=1$. 这类运算与第一步剪接中的运算相同, 它不会给 $\sigma^2(I)$ 中增加新词.

(2) $|S_1|=1$, $|S_2|\geqslant 2$. 例: 词 v_6v_1Y 与第一步剪接得到的词 $v_1v_2v_5Y$ 在规则 $r\in R$ 下发生运算 $(v_6v_1|Y, v_1|v_2v_5Y)\vdash r(v_6v_1v_2v_5Y, v_1Y)$.

(3) $|S_1|\geqslant 2$, $|S_2|=1$. 例: 第一步剪接得到的词 $v_1v_2v_5Y$ 与词 v_5v_4Y 在规则 $r\in R$ 发生运算 $(v_1v_2v_5|Y, v_5|v_4Y)\vdash r(v_1v_2v_5v_4Y, v_5Y)$.

(4) $|S_1|\geqslant 2$, $|S_2|\geqslant 2$. 例: 第一步剪接得到的词 $v_1v_2v_5Y$

和 $v_5v_4v_3Y$ 在规则 $r\in R$ 下发生运算 $(v_1v_2v_5 | Y, v_5 | v_4v_3Y) \vdash_r (v_1v_2v_5v_4v_3Y, v_5Y)$.

第三步剪接运算与第二步剪接类似.

下面我们看一下第四步也就是最后一步剪接得到的结果. 长度为 T 的词有: $v_6v_5v_4v_3v_2v_1Y$, $v_6v_1v_2v_5v_4v_3Y$, $v_6v_1v_4v_3v_2v_5Y$. 由推论 2: 图 2 中有三条不同的哈密顿路, 再由定理 2 或推论 1 知: 这三条哈密顿路分别为 $v_6v_5v_4v_3v_2v_1$, $v_6v_1v_2v_5v_4v_3$, $v_6v_1v_4v_3v_2v_5$.

5　结论

本文将 DNA 计算中的形式模型应用于图论中的 NP 完全问题. 利用剪接系统的高度并行性, 对有向哈密顿问题进行了模拟. 通过构造的剪接系统所产生语言的性质对有向哈密顿路问题进行了分析, 给出了有向图存在哈密顿路的充要条件.

参考文献:

[1] T Head. Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors[J]. Bull. Math. Biol, 1987, 49: 737- 759.

[2] T Head. Splicing schemes and DNA[J]. Nanobiology, 1992, 1: 335- 342.

[3] E Csuhaj-Varju, L Freund, L Kari, Gh Păun. DNA computing based on splicing: universality results[A]. Proceedings of 1st Annual Pacific Symposium on Biocomputing[C]. Hawaii, Jan. 1996.

[4] Gh Păun, A Salomaa. DNA computing based on the splicing operation[J]. Math. Japonica, 1996, 43(3): 607- 632.

[5] R Freund, L Kari, Gh Păun. DNA computing based on splicing: The existence of universal computers[J]. Theory of Computing Systems, 1999, 32: 69- 112.

[6] Y Benenson, T Paz-Elizur, et al. Programmable and autonomous computing machine made of biomolecules[J]. Nature, 2001, 414(22): 430 - 434.

[7] Katrin Erk. Simulating boolean circuits by finite splicing[A]. In Proceedings of the Congress on Evolutionary Computation (CEC' 99)[C]. Washington D. C, USA, July 6- 9, 1999.

[8] L M Adleman. Molecular computation of solution to combinatorial problems[J]. Science 1994, 266: 1021- 1024.

[9] J A Bondy, U S R Murty. Graph Theory with Application[M]. The Macmillan Press Ltd, 1976.

作者简介:

王淑栋　1973 年生, 1997 年获山东矿业学院应用数学与软件工程系理学学士学位, 2000 年获山东科技大学信息科学与工程学院运筹学与控制论理学硕士学位, 2004 年获华中科技大学系统工程研究所工学博士学位, 现为山东科技大学信息科学与工程学院副教授, 目前从事 DNA 计算、遗传算法、图与组合优化领域的研究. E-mail: wangshudong@mail.hust.edu.cn.

许　进　1959 年生, 教授, 博士导师. 现任中国数学会图论分会理事, 中国电子学会图论与系统优性专业委员会理事、委员以及核心组成人员, 目前主要从事 DNA 计算、神经网络、图论及遗传算法领域的研究.