

DNA 纳米结构自装配的语言能力分析

陈 燕¹⁾ 傅 岩²⁾ 朱 萌³⁾

¹⁾(北京工业大学应用数理学院 北京 100124)

²⁾(弗吉尼亚理工学院基因、生物信息学和计算生物学实验室 布莱克斯堡 VA 24061 美国)

³⁾(克莱姆森大学计算学院 克莱姆森 SC 29634 美国)

摘 要 文中研究了 DNA 分子杂交的内在计算能力,一方面,文中基于 Winfree 先前关于线性分子自装配仅能产生正则语言工作的基础上,进一步扩展证明了线性自装配通过杂交分别表示左、右线性派生的线性分子,也能产生线性语言.另一方面,文中定义了一种新的通过上下文无关语言通过自装配产生线性语言的方法,即证明了等价于上下文无关语言的特定序列集能通过 1-, 2-, 3- 粘头分子的混合自装配产生线性语言,同时,这是对 Winfree 关于树状纳米结构自装配等价于上下文无关语言理论的一个较好的补充.

关键词 自装配;语言能力;粘头;分子

中图法分类号 TP301

Language Capability Analysis of DNA Nanostructure's Self-Assembly

CHEN Yan¹⁾ FU Yan²⁾ ZHU Meng³⁾

¹⁾(Applied Science College, Beijing University of Technology, Beijing 100124)

²⁾(Genetics, Bioinformatics, and Computational Biology, Virginia Tech, Blacksburg, VA 24061, USA)

³⁾(School of Computing, Clemson University, Clemson, SC 29634, USA)

Abstract The computational capabilities inherent in the hybridization of DNA molecules were examined. First, based on the Winfree's previous work which demonstrated the self-assembly of linear molecules could generate only regular language, it was proven that the linear self-assembly can also generate linear languages, by hybridizing the linear molecules which respectively represent the left and right linear derivations. Then a new way was defined to prove that the unique set of sequences equivalent to context-free languages can be obtained by mixed self-assembly of molecules with 1-, 2-, 3- sticky ends, which is a supplement for Winfree's theory that the self-assembly of dendrimer nanostructures is equivalent to context-free language.

Keywords self-assembly; language capability; sticky ends; molecule

1 Introduction

The basic structure of DNA molecule is double-helix, whereas some nonlinear structures of DNA enlighten people to construct DNA molecules with different structures using Watson-Crick complementarity theory. Seeman's experiment composed complicated DNA structure for DNA Nano-

technology. Adleman has applied linear self-assembly to solve the Hamiltonian Path Problem. The self-assembly of the molecules with different structures presents different computation abilities. Due to the essential way of measuring the computation ability of a computation system is to discuss the formal language it generates, the complexity of the formal language indicates a classification of the

systems in terms of computational ability. Therefore, if we can prove the relationship between different DNA self-assembly and the formal language, then we could determine the computational ability of DNA molecular computer constructed by the self-assembly which is beneficial in our construction of molecular computers.

Winfree has examined the computational capabilities inherent in the self-assembly hybridization of DNA molecules^[1]. First he considered theoretical models, and showed that the self-assembly of oligonucleotides into linear duplex DNA can only generate sets of sequences equivalent to regular languages. If branched DNA is used for self-assembly of dendrimer structures, only sets of sequences equivalent to context-free languages can be achieved. However, it is noted that this form of self-assembly has not been widely studied in the lab, and that full self-assembly would be limited not only by material but also by geometric interferences and volumetric constraints. In contrast, the self-assembly of double crossover molecules can generate two dimensional sheets or three dimensional solids, or slot-filling model. These nanostructures resemble a single permanent binary event that involves two binding regions. Therefore theoretically, the two dimensional self-assembly is equivalent to a recursively enumerable language indicating a universal computation. And more remarkably, the proof relies on a very direct simulation of a universal class of cellular automaton. On the other hand, although being faster and more efficient than 1-tape Turing Machines because of the parallelism, the one dimensional cellular automaton model is not seem to be convenient model for computing functions.

Turing machine is the basic abstract computing model of electronic computer. Wang has testified that Turing machine can be simulated by self-assembly model of tiles covered by plane^[2]. Admittedly, Wang didn't concern concrete operation steps, even so it has been proved that given a Turing machine T and input X , if the Turing machine can stop, then a finite set with tile styles must be present consequentially, and existing effective operation process to make sure these tiles cover the plane non-overlapped and non-interspaced. Hence, to ascertain whether Turing machine can be simulated by molecular self-assembly model is a meaningful work. As a matter of fact, Winfree^[3-4] has demonstrated that the surprisingly sophisticated universal computation can be performed using tile

self-assembly model, namely tile self-assembly model possess Turing-universal.

Computation by linear assemblies of complex DNA tiles, which we call string tiles, was investigated thoroughly^[5]. By keeping track of the strands as they weave back and forth through the assembly, it can be shown that surprisingly sophisticated calculations can be carried out using linear string tile self-assembly. Thus, we can conclude that linear self-assembly of string tiles can generate the output languages of finite-visit Turing Machines. Surprisingly, a CNF-SAT problem of N clauses and M variables is solved using an initial set of $2M+2$ hairpin tiles of width N , which assemble to form all 2^M distinct tile assemblies of length $M+2$.

In this paper, we design self-assembly of other linear molecules on the basis of Winfree's work, which is different from his and takes on distinct calculation effect. So, the result could be considered to be a good supplement to the conclusion of linear molecule and computation capacity. Analysing DNA linear molecule self-assembly computation model, the rapid expansion of the operations applied to DNA computing leads to exigent need for modifying and complementing. There, we'll beneficially extend the language capability of the linear self-assembly model which based on the Winfree's work, furthermore discuss linear molecule with end self-assembly model and give a strict mathematical proof for what Winfree didn't. Meanwhile, we found Winfree's method has limitation of the conjunct with linear molecules of 2 sticky ends since the theory that the self-assembly of oligonucleotides into linear DNA molecule can generate regular languages was extended to be proved, for he didn't consider left or right derivation, and the property that linear molecules could be assembled to generate linear languages through left and right linear derivation, particularly this issue of the paper is accordant to his result that the self-assembly of molecules with 2 sticky ends is equivalent to regular languages.

On the other hand, Paul invested in details about the eight types of self-assembly of oligonucleotides into full single-strand and molecules with 1 sticky end^[6], which is called sticky computation. Theorem 1~4 in the following sections respectively studies the languages which are generated by single and mixed self-assembly into linear molecules with 1 and 2 sticky ends. So they could be some extension of other molecules with more stick-

y ends for sticky computation.

In the end, we define a new way to prove that the unique set of sequences equivalent to context-free languages can be obtained by mixed self-assembly of molecules with 1, 2, 3 sticky ends, which is a best complement to the theory that the self-assembly of dendrimer structures is equivalent to context-free languages. What’s more, the assumption and denotation in the demonstration are consistent with that about linear self-assembly generating regular languages.

2 Linear Self-Assembly is Equivalent to Regular Languages

Linear self-assembly begins with oligonucleotides or duplex DNA with sticky ends, and proceeds at a constant temperature, allowing only permanent binary events with a single perfectly complementary hybridization site and no intramolecular hybridization. Therefore, we must assume the whole DNA self-assembly process: Synthesize several DNA sequences. Mix the DNA together in solution. Heat the solution up and slowly cool it down, allowing the complexes of DNA to form. Chemically or enzymatically ligate adjacent strands. Denature the DNA again, and search whether single-stranded (or circular-stranded) DNA sequences are now present in the solution. There are three operations including hybridization, ligation, and Denaturation under the single self-assembly concept; after proceeding of the three operations in the sequence may generate the linear (or circular) DNA sequences (languages). The sequence is the string that DNA computing can attain under certain grammar, which presents its computational ability.

To describe the languages generated by linear self-assembly of different molecules, we introduce some denotations here. Consider $\Lambda_{\mathcal{R},d}$ to be the languages generated by self-assembly of the molecules with d sticky ends in the grammar derivation rule set \mathcal{R} . To be concise, we hypothesize that the length of the sticky ends does not affect the computational ability and thus ignore the discussion about it. As a matter of fact, the length of sticky ends can be completely varying, but whether more computing capabilities could be achieved still need further study.

Theorem 1. Self-assembly of the molecules with 1 sticky end generates the phrase $\Sigma_V \rightarrow \Sigma_T^*$.

Considering the grammar possessing one-step derivation format $\Sigma_V \rightarrow \Sigma_T^* \in \{ \Sigma_V, \Sigma_T, \mathcal{R}, S \}$, which

can be simulated by the following formal molecule (1) with 1 sticky end in Fig. 1 for $\forall V \rightarrow p \in \mathcal{R}$.

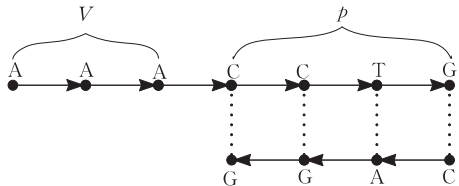


Fig. 1 Molecule (1) with 1 sticky end for $\forall V \rightarrow p \in \mathcal{R}$ used to be simulation, where the left side of molecules skicky-end generation formula is nonterminal, the right side of duplex generation formula without sticky-end is terminal

However, the molecule (2) with 1 sticky end in Fig. 2 can be simulated the condition when $\forall S$ is as follows.

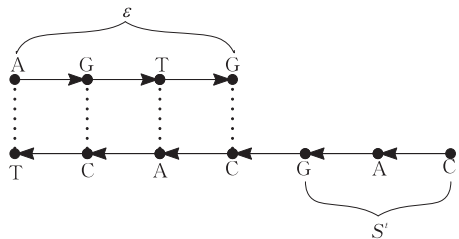


Fig. 2 Molecule (2) with 1 sticky end for any S (namely $S=V$) used to be simulation, where duplex strands segment of molecules without sticky-end may encode the grammar sequences, in addition to empty symbol, and the complementary strand of S is S' during encoding original symbols with sticky-ends

Actually, sticky-ended hybridization process driven by the second thermodynamics reaction can link molecule (1) with (2) with complementary sticky-ends, and further more the linking process is the derivation process of $\Sigma_V \rightarrow \Sigma_T^*$ in itself, as illuminated in Fig. 3.

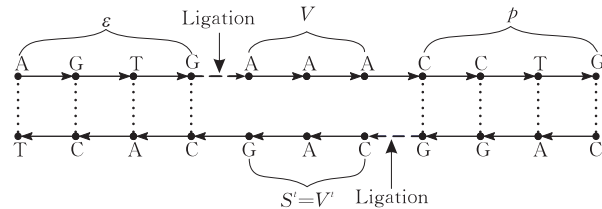


Fig. 3 A DNA molecule synthesized by molecules (1) and (2) through sticky-ended hybridization process driven by the second thermodynamics reaction

If self-assembly can generate duplex linear molecules without sticky-ends after balancing the hybridization reaction, then we can extract the terminal string $\Lambda_{\mathcal{R},1} = p$ of the DNA sequences using certain algorithm. Supposing there are simultaneously linear molecules (1)(2)(3)(4) with 1, 2 sticky ends (linear molecules (3)(4) exist in Theo-

rem 2 and Inference 1) participating in self-assembly. After hybridization reaction reaches equilibrium, if self-assembly forms an integrated duplex, and there will be a complete derivation of a string from the original S to the terminal. Note the generated DNA sequence as to be $\Delta_{\mathcal{R},2(L,R)}^{pre}$ (if one-side self-assembly happens, the sequence is noted as $\Delta_{\mathcal{R},2(L)}^{pre}$ or $\Delta_{\mathcal{R},2(R)}^{pre}$ accordingly, as follows), which could sequence through gel electrophoresis precisely. Apparently, $\Delta_{\mathcal{R},2(L,R)}^{pre}$ is the sequence that is formed of accumulated unit segment as $(V, L/R, p, V')$, which not only memorizes the grammars derive terminal string, but also memorizes the state transitions of the corresponding grammatical automaton during the whole derivation process. Therefore, we need to design an effective algorithm, which can extract terminal strings from $\Delta_{\mathcal{R},2(L,R)}^{pre}$, that is the language $\Delta_{\mathcal{R},2(L,R)}$ of grammar derivation.

Algorithm 1. $\Delta_{\mathcal{R},2(L,R)}^{pre} \mapsto \Delta_{\mathcal{R},2(L,R)}$.

Input a string $\Delta_{\mathcal{R},2(L,R)}^{pre}$,

1. Scan $\Delta_{\mathcal{R},2(L,R)}^{pre}$ from left to right, delete all characters of the string belonging to the set Σ_V .
2. Add original ending identified symbol φ in the end of $\Delta_{\mathcal{R},2(L,R)}^{pre}$.
3. Scan $\Delta_{\mathcal{R},2(L,R)}^{pre}$ from left to right again, insert non-terminal string p on the right of L and the left of R in turn.
4. Delete all meaningless characters φ, ϵ, L, R .

Output the string $\Delta_{\mathcal{R},2(L,R)}^{pre}$, which is our result $\Delta_{\mathcal{R},2(L,R)}$.

Specially, it follows that even though the self-assembly of molecules with 1 sticky end is simple, it is the basis for the generation of more senior DNA sequence grammars.

Definition 1. Consider the unit of self-assembly is u ('unit'), and one-side self-assembly of molecules with 2 sticky ends is defined as u satisfies the self-assembly of $\{u \mid u \in R, d(u) = 2\}$ (named as right side self-assembly); or satisfies the self-assembly of $\{u \mid u \in L, d(u) = 2\}$ (named as left side self-assembly).

Note that, the direction of Definition 1 is not the growing direction of the self-assembly into crystal lattice, but the location which of the string the nonterminal is replaced.

Theorem 2. Mixed one-side self-assembly of molecules with 1, 2 sticky ends is equivalent to regular languages.

We can only prove the relationship between the language $\Delta_{\mathcal{R},2(R)}$ generated by right-side self-assembly and the regular language RL ; and the left-side process is similar. The proof can be shown as

three steps to prove: $\Delta_{\mathcal{R},2(R)} \Leftrightarrow RL$.

(1) First of all, prove linear molecules with 2 sticky ends can encode the production of right-linear grammars.

Consider any right linear grammars $\{\Sigma_V, \Sigma_T, \mathcal{R}, S\}$, where $\mathcal{R}: \Sigma_V \rightarrow \Sigma_T^+ \cup \Sigma_T^+ \Sigma_V$, where $+$ is the positive closure. Encode linear molecule (3) with 2 sticky ends as the following Fig. 4, and for any production $V \rightarrow pV'$ of \mathcal{R} , there exists only one molecule with 2 sticky ends to simulate it.

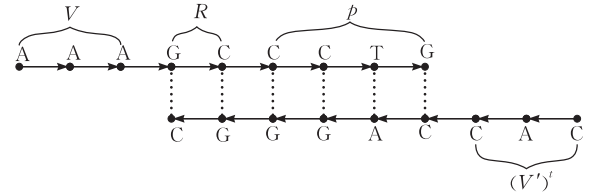


Fig. 4 Linear molecule (3) with 2 sticky ends encoded by any production $V \rightarrow pV'$ in \mathcal{R} , where the two sticky ends of that respectively encode the nonterminal around the production, and the duplex segment encodes the right linear derivation representative R and the terminal string p in the right of production. The code of terminal V and non-terminal string p is just functional representative, which can be changed. $(\bullet)'$ denotes the Watson-Crick complement of the string in the bracket

(2) Molecules with 1 sticky end can simulate the production formed in $V \rightarrow p$ from Theorem 1. Here we still need to use the molecule (1) with 1 sticky end described in Theorem 1 to simulate $V \rightarrow p$, and the molecule to simulate the one-step derivation with the beginning of original character S .

Obviously, the hybridization of sticky ends which is driven by the second thermodynamics reaction can put molecules with 1 or 2 sticky ends to link together. And this is also a process that grammar $\{\Sigma_V, \Sigma_T, \mathcal{R}, S\}$ derives languages in itself. If self-assembly can generate duplex linear molecules without sticky-ends after reaching hybridization reaction equilibrium, we can extract the terminal string $\Delta_{\mathcal{R},2(R)}$ of the DNA sequence by Algorithm 1, that is to say, the sequence encodes the languages by the right linear grammars derivation.

(3) Because the right linear grammar is equivalent to the regular grammars^[7], so it is obtained that: A necessary and sufficient condition for equality between Δ and RL is that there exists a grammar that generates language Δ , and its production may be as $A \rightarrow a$ or as $A \rightarrow aB$, which is the right linear grammars production, where A, B are grammars variables, a is the terminal string. That means that the right linear derivation is equivalent

to regular grammars, and generates regular languages.

In conclusion of the verification about (1), (2), (3), we can get $\Lambda_{\mathcal{R},2(\mathcal{R})} \Leftrightarrow RL$.

Inference 1. $\Lambda_{\mathcal{R},2(L)} \Leftrightarrow RL$.

The proof is similar as that of Theorem 2. Note that the left linear grammar $R: \Sigma_V \rightarrow \Sigma_T^+ \cup \Sigma_V \Sigma_T^+$ is equivalent with the right linear grammar. The linear molecule with 2 sticky ends simulating the left linear grammar is illuminated in Fig. 5. Similarly, the languages generated by self-assembly could extract the meaningful string $\Lambda_{\mathcal{R},2(L)}$ using Algorithm 1.

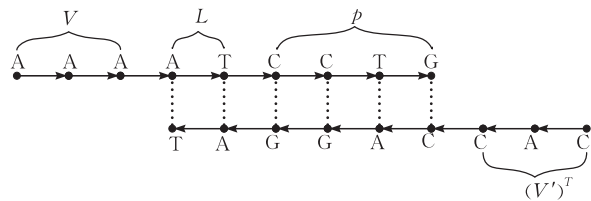


Fig. 5 Linear molecule (4) with 2 sticky ends to simulate the left linear grammar

Theorem 3. $\Lambda_{\mathcal{R},2(L,R)} \Leftrightarrow LIN$.

Viz.: Mixing double-side self-assembly into 1, 2 sticky-ended molecules can generate linear language.

Considering the definition of LIN is the format of $u \rightarrow v$, where u is equal to N , v is equal to $T^* \cup T^* N T^*$, then linear derivation can be formed by the combination of the right linear derivation and the left linear derivation; but on the other hand, the self-assembly of linear molecules with 2 sticky ends is equivalent to the right linear derivation, besides the left linear derivation. Thus, the mixed self-assembly of linear molecules with 2 sticky ends can simulate the linear derivation process so that the analysis can move on.

So, the proof of the theorem can be carried on in three steps:

(1) Linear languages LIN is one of context-free languages from the definition, containing regular languages, derived by linear grammars. If the production has the form of $\Sigma_V \rightarrow \Sigma_T^* \cup \Sigma_T^* \Sigma_V \Sigma_T^*$, then the quadruple $[\Sigma_V, \Sigma_T, \mathcal{R}, S]$ is a type of linear grammars.

(2) On the other hand, we can consider that in the perspective of property:

On the one hand, we have $LIN \subset \mathcal{R}_{(L,R)}$, which means any linear grammars can be obtained by linking a group of right linear grammars with the left linear grammars accordingly. In fact, if that can be rewritten two rules as follows: $A \rightarrow$

pC , $C \rightarrow Bq$ for $\forall A \rightarrow pBq \in \mathcal{R}$, $p, q \in \Sigma_T^*$, $A, B \in \Sigma_V$, B can be empty, where C is a newly additional nonterminal, and only can be used in the group of production.

On the other hand, we have $\mathcal{R}_{(L,R)} \subset LIN$, which means any combination linking the right linear production and the left cannot go beyond the category of linear grammars, i. e. that cannot generate the context-free production. Actually, Any context-free grammar \mathcal{R} has its most brief form as $\Sigma_V \rightarrow \Sigma_T^* \cup \Sigma_V \Sigma_V$. However, the four kinds of any combination of the linkable right linear production and the left only just generate the form of $\Sigma_T^* \Sigma_V \Sigma_T^*$, the proof of which is given as follows:

(i) Left+Left

Consider Left to be $A \rightarrow Cq$ and $C \rightarrow Bq'$ as well, then (Left+Left) is $A \rightarrow Cq \rightarrow Bq'q$.

(ii) Left+Right

Consider Left to be $A \rightarrow Cq$, and Right to be $C \rightarrow pB$, then (Left+Right) is $A \rightarrow Cq \rightarrow pBq$.

(iii) Right+Left

Consider Right to be $A \rightarrow pC$, Left to be $C \rightarrow Bq$, then (Right+Left) is $A \rightarrow pC \rightarrow pBq$.

(iv) Right+Right

Consider Right to be $A \rightarrow pC$ and $C \rightarrow p'B$ as well, then (Right+Right) is $A \rightarrow pC \rightarrow pp'B$.

From the above, the case of (Left+Left) or (Right+Right) is equivalent to regular grammars, but the mixing of left and right can generate more advanced languages than regular grammars. In a word, the four kinds combination cannot generate the grammar in the form of $\Sigma_V \rightarrow \Sigma_V \Sigma_V$.

And then, the part has proved that the mixing of the left and the right linear grammar is equivalent to linear grammar.

(3) Furthermore, the single-side self-assembly of linear molecules with 2 sticky ends corresponds to the right linear grammar or the left from Theorem 1 and Inference 1, and then we can achieve that language $\Lambda_{\mathcal{R},2(L,R)}$ generated by the mixed double side self-assembly is equivalent to the linear language LIN described in (2), so the proof of $\Lambda_{\mathcal{R},2(L,R)} \Leftrightarrow LIN$ is obtained.

In the proof of Theorem 1, the self-assembly of linear molecules (1)(2)(3) can simulate the right linear derivation; moreover molecules (1)(2)(4) can simulate the left. Mix (1)(2)(3)(4) and proceed process similar to Theorem 1. The sticky-ends hybridization reaction driven by the second thermodynamics reaction can make molecules with 1 or 2 complementary sticky ends link together, and the linking process is the process that mixed

left and right linear grammar derives languages in itself. If self-assembly can generate duplex linear molecules without sticky ends after the hybridization reaction reaches equilibrium, then we can extract the terminal string $\Delta_{\mathcal{R}, 2(L, R)}$ from the DNA sequence using Algorithm 1, which encodes the languages of mixed left and right linear grammar derivation, and that is linear languages LIN .

3 Dendrimer Self-Assembly is Equivalent to Context-Free Languages

Dendrimer self-assembly is an efficient approach to construct sizable, soft-matter nanostructures without paying a large entropic cost, as the assembling units or building blocks are already macromolecules of considerable size. DNA self-assembly process can be generalized to set operations by applying the operation to each branch junction molecule in the original set, and taking the union of dendrimer complexes that result. Thus, the self-assembly begins with duplexes, hairpins, and 3-armed junctions with sticky ends, and proceeds at a constant temperature, allowing only permanent binary events with a single perfectly complementary hybridization site and no intramolecular hybridization.

On the other hand, Abrahams-Gessel^[8] suspects that dendrimer self-assembly seems formally identical to context-free grammars. Therefore, by the inspiration of the Winfree's work, we propose a new way to prove the mix self-assembly of molecules with 1, 2, 3 sticky ends is equivalent to context-free languages, which is a better theoretical supplement for the theory that dendrimer self-assembly is equivalent to context-free languages. The assumptions and symbols of the following proof are identical to that of last section.

Theorem 4. $\Delta_{\mathcal{R}, 3} \Leftrightarrow CFL$.

Here, we need to prove that the mix self-assembly of molecules with 1, 2, 3 sticky ends is equivalent to context-free languages.

This can be proved in three steps:

- (1) Linear molecules with 3 sticky ends can encode CFG production $A \rightarrow BC$;
- (2) The self-assembly of linear molecules with 1, 2, 3 sticky ends is equivalent to context-free grammar derivation;
- (3) CFG generates CFL.

Consider any context-free grammar $\{\Sigma_V, \Sigma_T, \mathcal{R}, S\}$, where the simplest form of production \mathcal{R} in Chomsky normal forms is $\mathcal{R}: \Sigma_V \rightarrow \Sigma_T \cup \Sigma_V \Sigma_V$. We can encode the molecules with 3 sticky ends in

the following way of Fig.6 (i.e. three-armed branched junction DNA) so that there exists a unique molecule with 3 sticky ends that could simulate it.

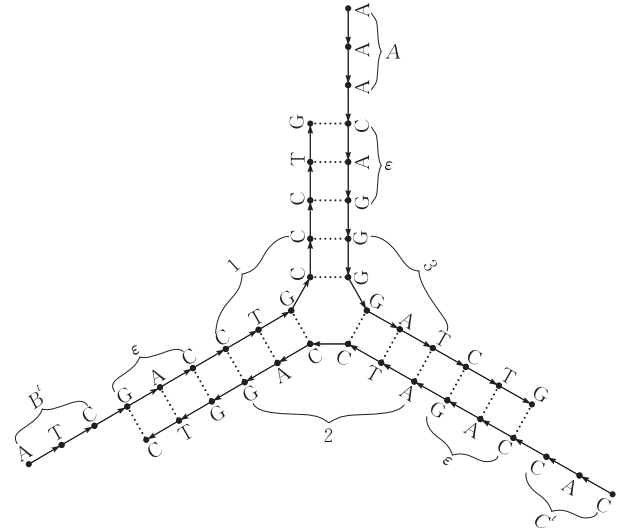


Fig. 6 There exists a unique molecule (5) with 3 sticky ends for any production $A \rightarrow BC$ of \mathcal{R} , where $A, B, C \in \Sigma_V$, three sticky ends of the molecule encode nonterminal A, B, C around production respectively, and three arms segment of that encode meaningless symbol ϵ and serial numbers "1, 2, 3"

Three-armed junction encodes the serial numbers of the nonterminal which arise in the context-free grammar production, such as the serial numbers of A, B, C in $A \rightarrow BC$ are 1, 2, 3 respectively. Thus, a corresponding three-armed branched junction molecule (5) can be found to simulate all the production of the format of $A \rightarrow BC$.

It is shown that the molecules with 1 sticky end can simulate the format of $V \rightarrow p$ production from Theorem 1. Here, Fig. 7 use another molecule with 1 sticky end to simulate $V \rightarrow p$, and the meaningless ϵ can be encoded in the part of hairpin for the originals so that deduction could be unaffected.

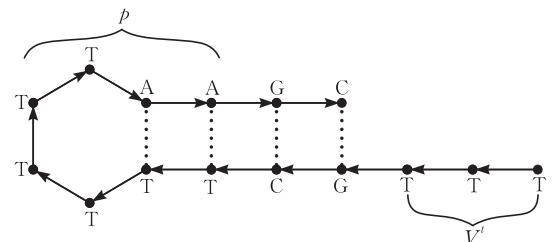


Fig. 7 The molecule (6) with 1 sticky end to simulate $V \rightarrow p$, where the circular segment encode the terminal p , the sticky-end segment encode the non-terminal on the left of the production

Obviously, the sticky-ends hybridization reaction driven by the second thermodynamics reaction

can make the molecules with 1, 2 or 3 sticky ends with complementary sticky ends link together, and the linking process is the very process that the grammar $\{\Sigma_V, \Sigma_T, \mathcal{R}, S\}$ derives languages in itself. Derivation begins with the arm labeled 1 of molecules with 3 sticky ends, and moves to two branches direction. If there is context-free derivation after branching, the arm labeled 2 or 3 can link with a new 3-armed molecule, and generate new branches; if there is linear derivation after branching, then the molecules with 2 sticky ends stick to the sticky-ends of the molecules with 3 and derive sticky-ends of the next nonterminal. The hybridization process is able to persist to the point that sticky-ends of all branches are combined with corresponding hairpin molecules with 1 sticky end and they form a integrated circular duplex, which means all nonterminals are replaced by the terminal in the process of derivation. Add junction enzyme and heat the solution up after hybridization reaction reaching equilibrium, if the single strands circular molecules have been formed (sequence of this molecule can be read by circular molecules gel electrophoresis), then we can extract terminal string $\Lambda_{\mathcal{R},3}$ of the circular DNA sequences using Algorithm 2.

Supposing there exists simultaneously molecules with 1, 2, 3 sticky ends involving in self-assembly such as (3)(4)(5)(6) in the solution at the same time. Add junction enzyme and heat it up after hybridization reaction reaching equilibrium, if the self-assembly forms integrated circular strands, then it indicates that there exists an integrated context-free derivation from the initial S to the terminal string. Denote the generated DNA sequence as $\Lambda_{\mathcal{R},3}^{pre}$, where $\Lambda_{\mathcal{R},3}^{pre}$ is a circular molecule, which could be sequenced through circular gel electrophoresis. Similar to regular grammars and linear grammars, $\Lambda_{\mathcal{R},3}^{pre}$ contains not only the terminal string derived by grammars, but also meaningless symbol, serial number information and the replaced nonterminal. So we have to design an algorithm to extract the language $\Lambda_{\mathcal{R},3}$ derived by grammars from $\Lambda_{\mathcal{R},3}^{pre}$. Unlike Algorithm 1, we'll deal with a circular string here.

Algorithm 2. $\Lambda_{\mathcal{R},3}^{pre} \mapsto \Lambda_{\mathcal{R},3}$.

Input circular string $\Lambda_{\mathcal{R},3}^{pre}$,

1. Cut $\Lambda_{\mathcal{R},3}^{pre}$ at index 1 that corresponds to the initial character, and form linear string.
2. Iterate through $\Lambda_{\mathcal{R},3}^{pre}$ from left to right and delete all characters belonging to Σ_V .
3. Set adjoining layer serial numbers 1, 2, 3

to be branching locating symbols, and every branches to be a palindrome. Keep point location of every branches steady, and search leftward and rightward simultaneously:

(1) If symbol L is found, then fold the string between the symmetrical place L to the right with the original branch midpoint as symmetrical point, and the folded parts are not folded again.

(2) If symbol R is found, then fold the string between the symmetrical place R to the left with the original branch midpoint as symmetrical point, and the folded parts are not folded again.

(3) If the same layer serial number 1, 2, 3 are found, then stop look up, and go into the above layer.

4. Delete the serial numbers and meaningless symbol ϵ .

Output the string $\Lambda_{\mathcal{R},3}$.

Well then, the sequence of circular string $\Lambda_{\mathcal{R},3}^{pre}$ encodes the languages derived by context-free grammars. That is $\Lambda_{\mathcal{R},3} \Leftrightarrow CFL$.

4 Discussion

The analysis of all types of DNA computational model demonstrates that the self-assembly of different DNA nanostructures molecules by self-Assembly can generate different types of languages, which then correspond to different computational capabilities and the universal computation. Based on Winfree's works, we further analysed the language capability and computing function of linear molecules, 3-arm branched junction dendrimer DNA molecules. In addition, we assume that the mixed self-assembly of the molecules with 4 sticky ends or higher pad number cannot generate context-free languages, which is because when junction molecules join together, there is a common character that only one pad can hybridize, namely join together. Therefore, there are two choices to represent the context: one is to label all context, the other is to replace the nonterminal by terminal string through special operations. However, the former is infeasible while the latter deserves further study.

On the other hand, Winfree has achieved some inspiring experimental results in the two dimensional self-assembly models, which shows that the potentiality of universal computation has prevailing meaning in nature; and still analyzed the computing capability of one dimensional string tiles and two dimensional double-crossover (DX) molecules by self-assembly, which has guiding effect to de-

sign other applied models of DNA computation.

With the generation of new biological technology, we can design more DNA computing operations, modify the original self-assembly computing models and have further supplement, so that more powerful computation capability can be achieved. Besides, the self-assembly process can be used as an interesting universal computing device, but whether that is useful still deserves further consideration and study. Perhaps it can be applied in future nanotechnology.

References

[1] Winfree Erik, Yang Xiao-Ping, Seeman N C. Universal computation via self-assembly of DNA: Some theory and experiments//Landweber L F, Baum E B eds. DNA Based Computers II; DIMACS Workshop, 1996, 44, Providence, RI, 1998

[2] Wang H. Proving theorems by pattern recognition II. Bell Systems Technical Journal, 1961, 40: 1-42

[3] Winfree Erik, Liu Fu-Rong, Wenzler L A, Seeman N C. Design and self-assembly of two-dimensional DNA crystals. Nature, 1998, 394: 539-544

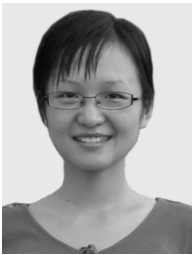
[4] Winfree Erik. Algorithmic self-assembly of DNA[Ph. D. dissertation]. California Institute of Technology, Pasadena, California, USA, 2003

[5] Winfree Erik, Eng Tony, Rozenberg Grzegorz. String tile models for DNA computing by self-assembly//Condon A ed. Proceedings of the DNA 2000. LNCS 2054. Springer, 2001: 63-88

[6] Paun, Rozenberg G, Salomaa A. DNA Computing: New Computing Paradigms. Berlin; Springer, 2002

[7] Jiang Zong-Li, Jiang Shou-Xu. Formal Languages and Automata Theory. Beijing; Tsinghua University Press, 2002: 70

[8] Dan Abrahams-Gessel (Dartmouth), Personal Communication. Berlin; Springer, 1996



CHEN Yan, born in 1983, M. S. candidate. Her research direction is DNA computing and bioinformation.

FU Yan, born in 1982, Ph. D. candidate. Her research interests include DNA computing and bioinformation.

ZHU Meng, born in 1982, Ph. D. candidate. His research interests include pattern recognition and bioinformation.