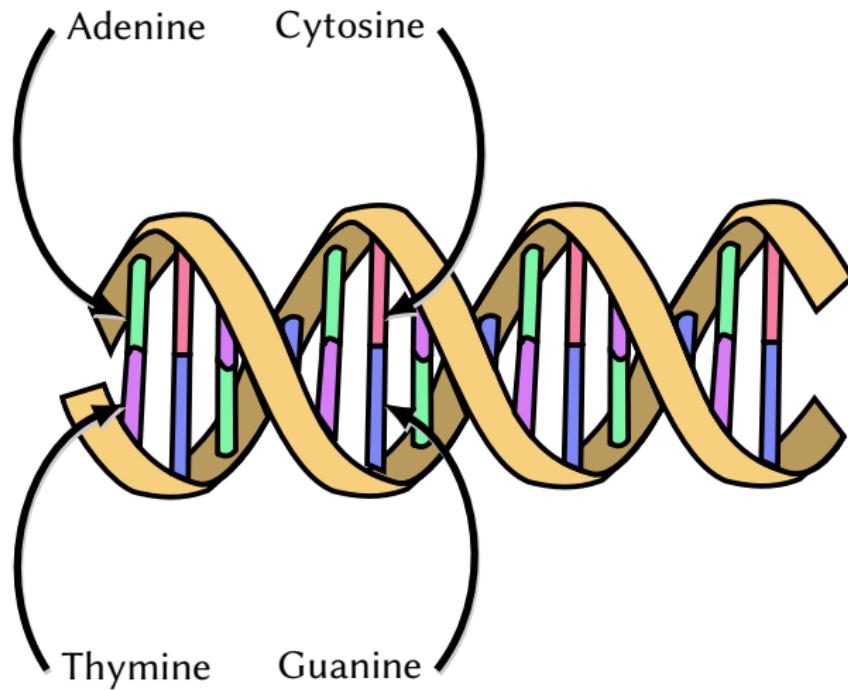


Toehold DNA Languages are Regular

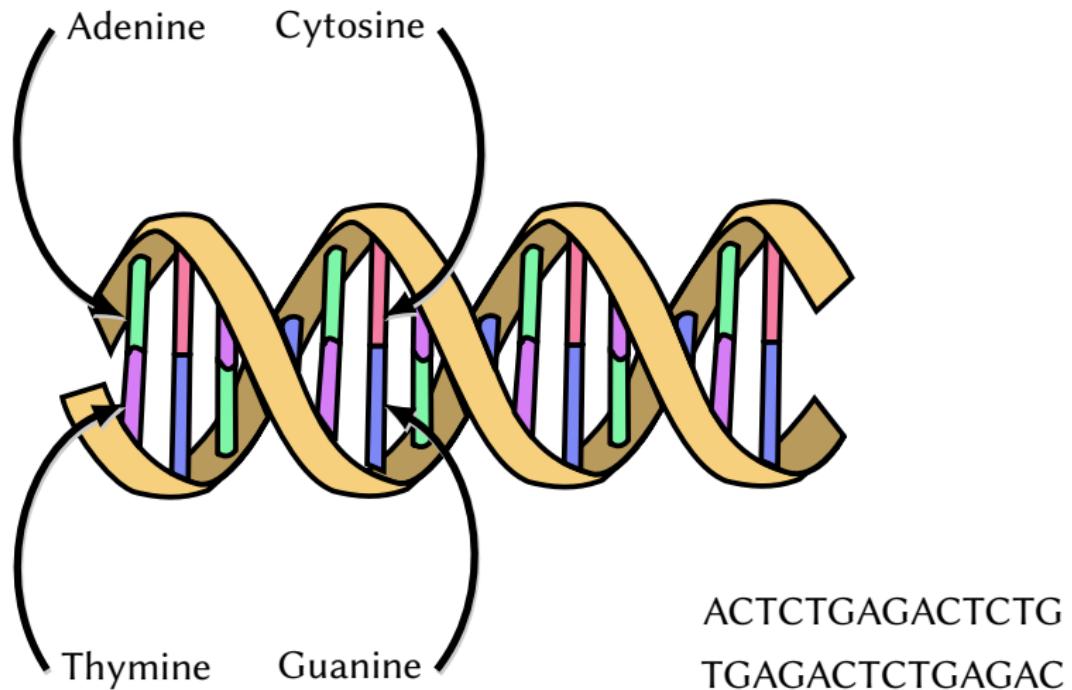


Sebastian Brandt Nicolas Mattia Jochen Seidel Roger Wattenhofer

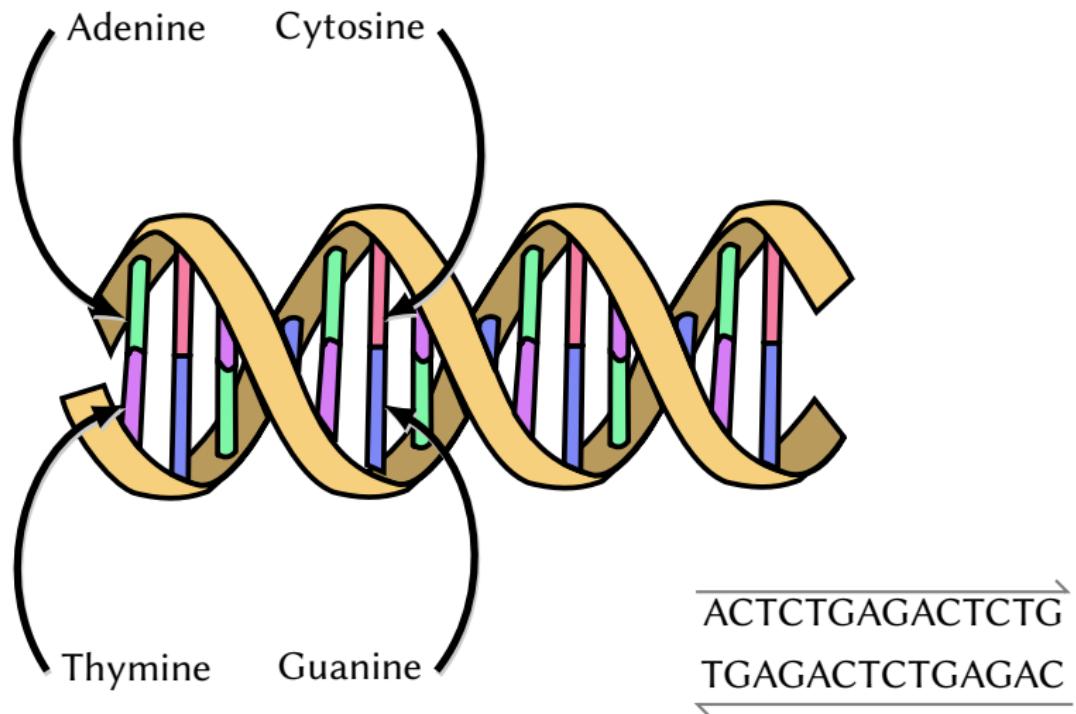
DNA



DNA



DNA



Domains & Complements

$$\overrightarrow{\mathbf{a}} \equiv \overrightarrow{\text{ATCG}}$$

$$\overrightarrow{\mathbf{b}} \equiv \overrightarrow{\text{ATT}}$$

$$\overrightarrow{\mathbf{c}} \equiv \overrightarrow{\text{CTC}}$$

$$\overrightarrow{\overline{\mathbf{a}}} \equiv \overrightarrow{\overrightarrow{\text{ATCG}}}$$

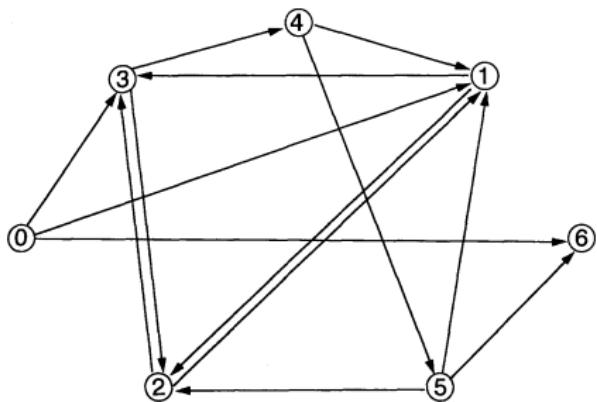
$$\overrightarrow{\overline{\mathbf{b}}} \equiv \overrightarrow{\overrightarrow{\text{ATT}}}$$

$$\overrightarrow{\overline{\mathbf{c}}} \equiv \overrightarrow{\overrightarrow{\text{CTC}}}$$

$$\begin{array}{c} \overrightarrow{\mathbf{abc}} \\ \overleftarrow{\mathbf{abc}} \end{array} \equiv \begin{array}{c} \overrightarrow{\text{ATCGATTCTC}} \\ \overleftarrow{\text{TGAGAAGAG}} \end{array}$$

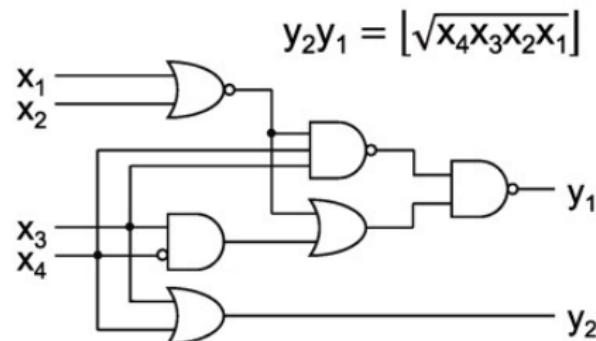


Hamiltonian Path



Adleman 1994

Square Root



Qian, Winfree 2011

Strand Binding

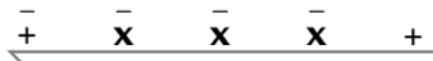
+

x x x x x x x x x *

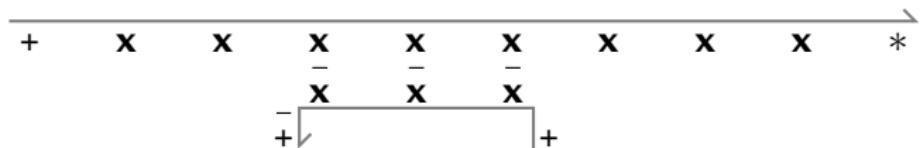


- - - - +

x x x x



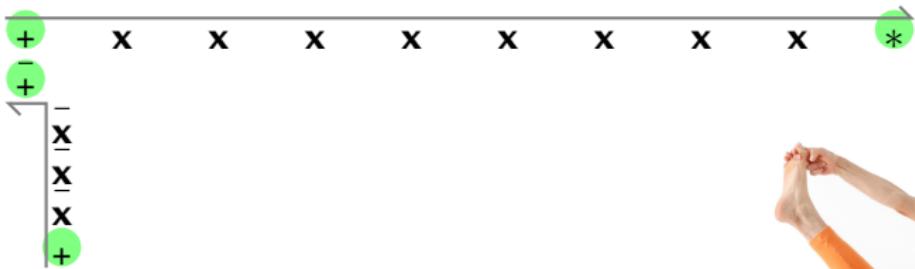
Strand Binding



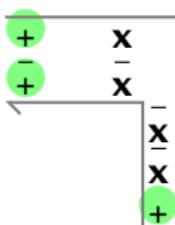
Strand Binding, Toeholds



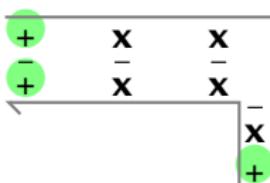
Strand Binding, Toeholds



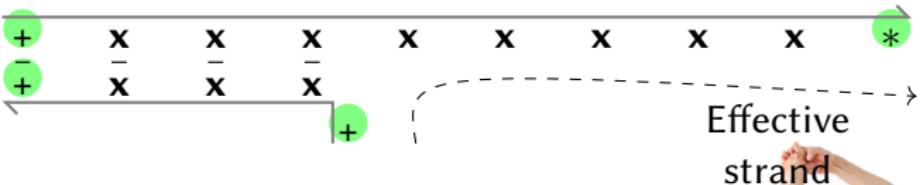
Strand Binding, Toeholds



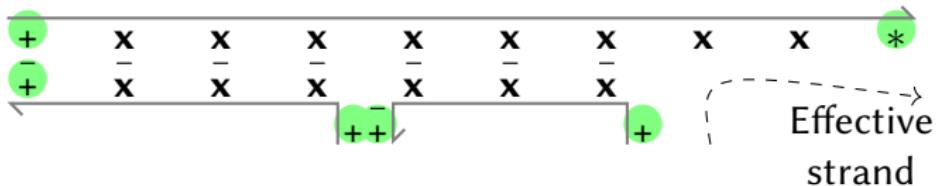
Strand Binding, Toeholds



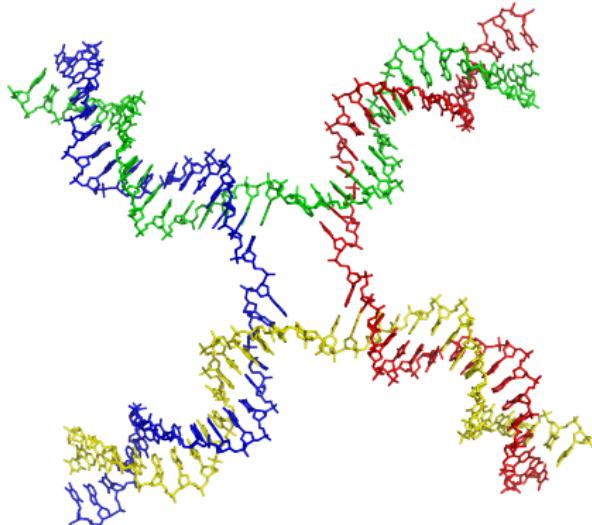
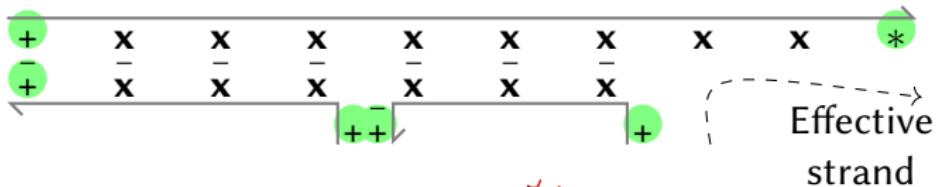
Strand Binding, Toeholds, Effective Strands



Strand Binding, Toeholds, Effective Strands



Strand Binding, Toeholds, Effective Strands





Algorithm

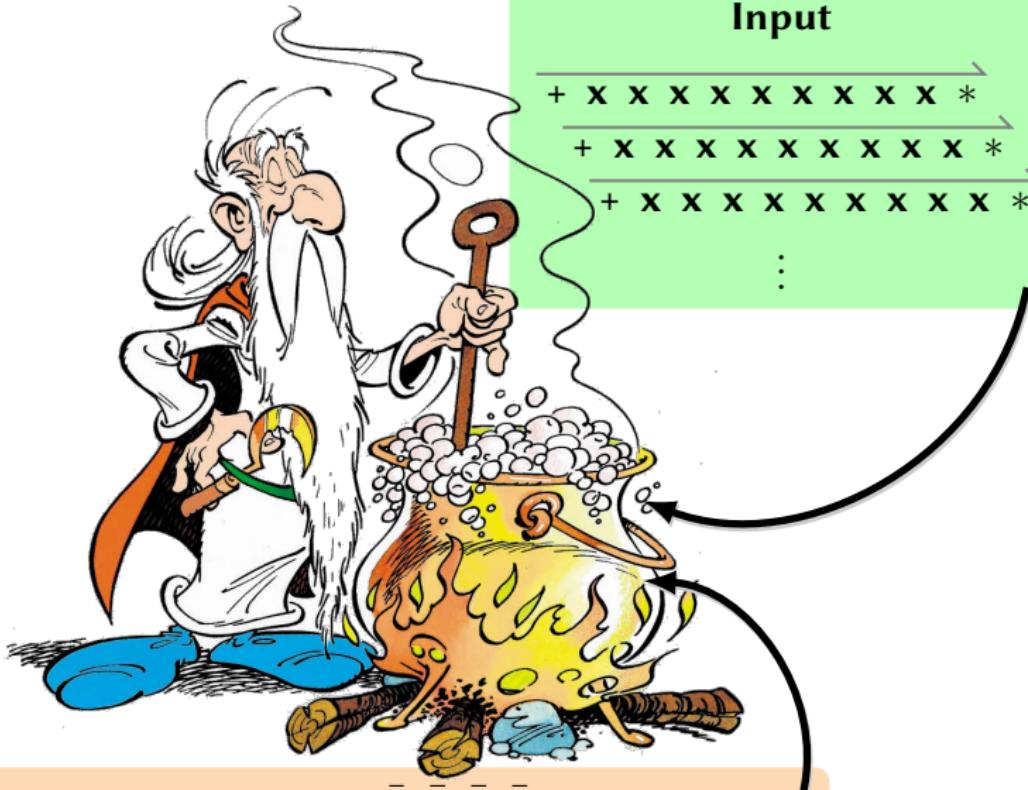
$$\begin{array}{r} + \ x \ x \ x \ + \\ - \ - \ - \\ + \ x \ x \ x \ + \\ - \ - \ - \\ + \ x \ x \ x \ + \\ - \ - \ - \\ \hline \end{array} \quad \dots$$

Below the first row of the algorithm, there is a horizontal arrow pointing to the right under the '+' sign, with the letters 'a' and 'b' written above it. Below the second row, there is another horizontal arrow pointing to the right under the '+' sign, with the letter '*' written above it.

Input

$$\begin{array}{r} + \times \times \times \times \times \times \times \times \times * \\ + \times \times \times \times \times \times \times \times \times * \\ \hline + \times \times \times \times \times \times \times \times \times * \end{array}$$

⋮



Algorithm

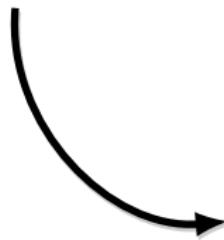
$$\begin{array}{r} - \quad - \quad - \\ + \times \times \times + \\ - \quad - \quad - \\ + \times \times \times + \\ - \quad - \quad - \\ + \times \times \times + \end{array} \quad \dots$$

Below the first column of the algorithm, there is a horizontal arrow pointing left under the minus sign, with the letters "a b *" written below it. This indicates that the algorithm is being applied to the multiplication problem shown in the green box above.

- - - -
+ x x x +

- -
+ a b *

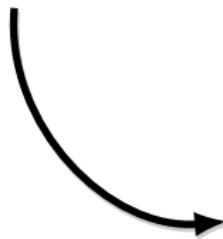
+ x x x x x x x x x *



- - - -
+ x x x +

- -
+ a b *

+ x x x x x x x x x *

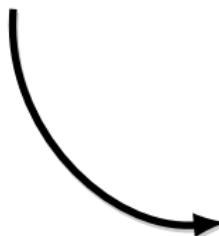


- - - -
+ x x x | - a b *
| ++ \

$\begin{array}{r} - \\ + \end{array} \begin{array}{c} x \\ x \\ x \\ x \end{array} \begin{array}{r} - \\ + \end{array}$

$\begin{array}{r} - \\ + \end{array} \begin{array}{c} a \\ b \end{array} \begin{array}{r} * \\ \end{array}$

$\begin{array}{r} \overbrace{\hspace{10cm}} \\ + \end{array} \begin{array}{cccccccccc} x & x & x & x & x & x & x & x & x & x & * \end{array}$



$\begin{array}{r} - \\ + \end{array} \begin{array}{c} x \\ x \\ x \end{array} \begin{array}{r} - \\ + \end{array} \begin{array}{c} a \\ b \end{array} \begin{array}{r} * \\ \end{array}$

$\begin{array}{r} \overbrace{\hspace{3cm}} \\ + \end{array} \begin{array}{c} x \\ x \\ x \end{array} \begin{array}{r} - \\ + \end{array} \begin{array}{c} x \\ x \\ x \end{array} \begin{array}{r} - \\ + \end{array} \begin{array}{c} x \\ x \\ x \end{array} \begin{array}{r} + \\ \end{array}$





$$\begin{array}{cccccccccc} + & \times & * \\ - & - & - & - & - & - & - & - & \\ + & \times & \\ \swarrow & \searrow & \swarrow & \searrow & \swarrow & \searrow & \swarrow & \searrow & \end{array}$$

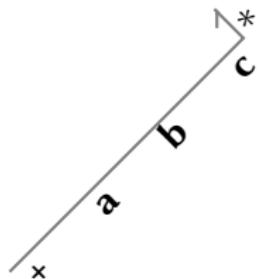
The diagram shows a sequence of eight symbols: a plus sign (+), a multiplication sign (×), a minus sign (-), another multiplication sign (×), another minus sign (-), another multiplication sign (×), another minus sign (-), and another multiplication sign (×). Above the sequence, there are three rows of operators: a plus sign (+) above the first symbol, a minus sign (-) above the second symbol, and a plus sign (+) above the third symbol. Below the sequence, there are three pairs of arrows pointing diagonally: a pair of arrows pointing up and to the left between the first two symbols, a pair of arrows pointing down and to the right between the second and third symbols, and a pair of arrows pointing up and to the left between the fourth and fifth symbols.



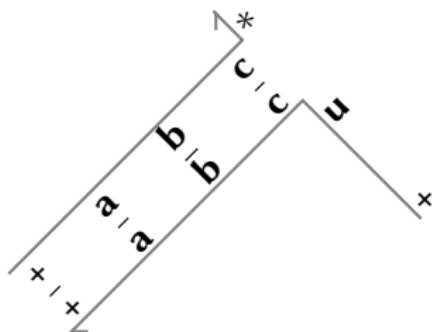
$$\begin{array}{cccccccccc} + & \times & * \\ - & - & - & - & - & - & - & - & \\ + & \times & \\ \swarrow & \searrow & \swarrow & \searrow & \swarrow & \searrow & \swarrow & \searrow & \end{array}$$

The diagram shows a sequence of operations: addition (+), multiplication (×), subtraction (-), addition (+), multiplication (×), subtraction (-), addition (+), multiplication (×), subtraction (-), and multiplication (*). Brackets group the operations into pairs: (addition, multiplication), (subtraction, multiplication), (addition, multiplication), (subtraction, multiplication), and (addition, multiplication).

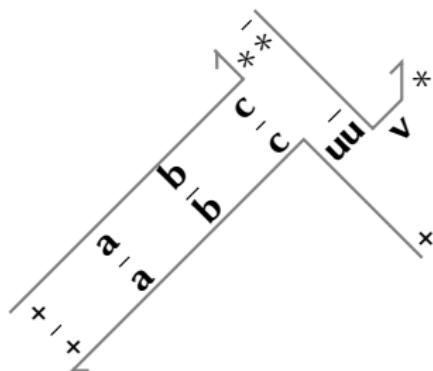
Substitution: $+abc^* \rightarrow +z^*$



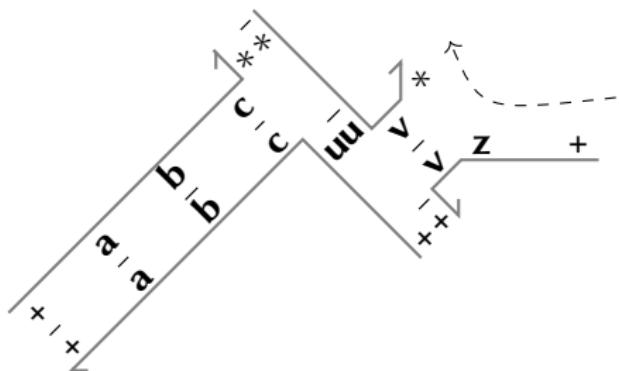
Substitution: $+abc^* \rightarrow +z^*$



Substitution: $+abc^* \rightarrow +z^*$



Substitution: $+abc^* \rightarrow +z^*$

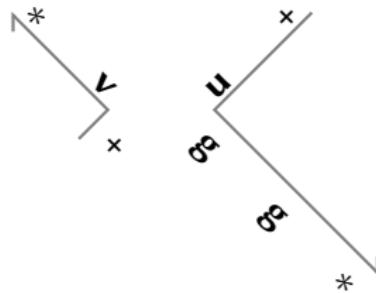


Aggregation: $+abc^* \wedge +def^* \rightarrow +gg^*$

1. $+abc^* \rightarrow +ugg^*$

2. $+def^* \rightarrow +vv^*$

3. $\overline{\overline{\overline{\overline{*v u +}}}}$

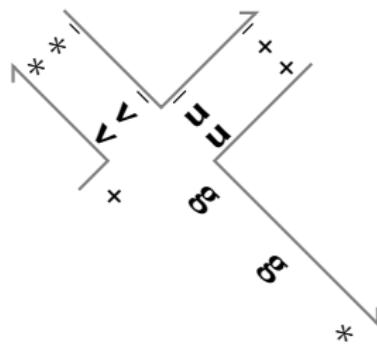


Aggregation: $+abc^* \wedge +def^* \rightarrow +gg^*$

1. $+abc^* \rightarrow +ugg^*$

2. $+def^* \rightarrow +vv^*$

3. $\overline{\overline{v}u+}$

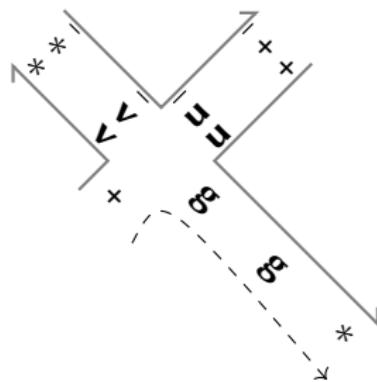


Aggregation: $+abc^* \wedge +def^* \rightarrow +gg^*$

1. $+abc^* \rightarrow +ugg^*$

2. $+def^* \rightarrow +vv^*$

3. $\overline{v} \overline{u}$



Primality

For $2 \leq j \leq \sqrt{n}$ and $k \leq j - 1$:

$$1. \xrightarrow{+b^j xx+}$$

$$2. \xrightarrow{+b^j x^j b^j +}$$

$$3. \xrightarrow{+b^j x^k *} \rightarrow \xrightarrow{+b^j f*}$$

$$4. \xrightarrow{+b^2 f*} \wedge \xrightarrow{+b^3 f*} \wedge \cdots \wedge \xrightarrow{+b^{\sqrt{n}} f*} \rightarrow \xrightarrow{+*}$$

Primality

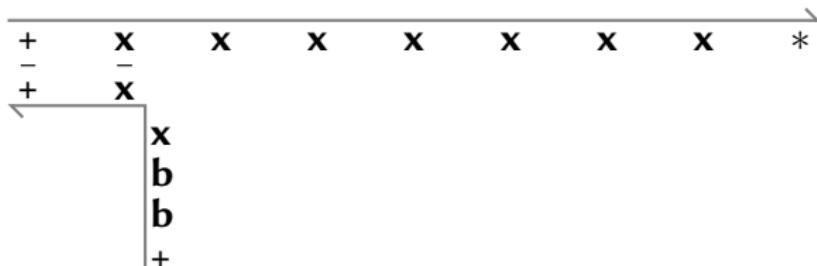
For $2 \leq j \leq \sqrt{n}$ and $k \leq j - 1$:

$$1. \overrightarrow{+b^j \cancel{xx} +}$$

$$2. \overrightarrow{+b^j \cancel{x^j} \cancel{b^j} +}$$

$$3. \overrightarrow{+b^j x^k *} \rightarrow \overrightarrow{+b^j f *} \quad \text{---}$$

$$4. \overrightarrow{+b^2 f *} \wedge \overrightarrow{+b^3 f *} \wedge \cdots \wedge \overrightarrow{+b^{\sqrt{n}} f *} \rightarrow \overrightarrow{+*}$$



Primality

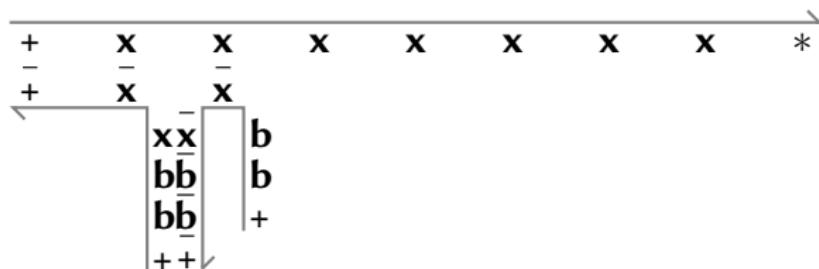
For $2 \leq j \leq \sqrt{n}$ and $k \leq j - 1$:

$$1. \overrightarrow{+b^j \underline{x} \underline{x} +}$$

$$2. \overrightarrow{+b^j \underline{x}^j \underline{b}^j +}$$

$$3. \overrightarrow{+b^j \underline{x}^k *} \rightarrow \overrightarrow{+b^j f *}$$

$$4. \overrightarrow{+b^2 f *} \wedge \overrightarrow{+b^3 f *} \wedge \cdots \wedge \overrightarrow{+b^{\sqrt{n}} f *} \rightarrow \overrightarrow{+*}$$



Primality

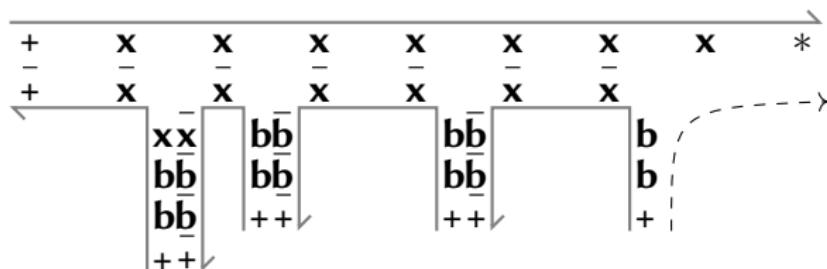
For $2 \leq j \leq \sqrt{n}$ and $k \leq j - 1$:

1. $\overrightarrow{+b^j \bar{x} \bar{x} +}$

2. $\overrightarrow{+b^j \bar{x}^j \bar{b}^j +}$

3. $\overrightarrow{+b^j \bar{x}^k *} \rightarrow \overrightarrow{+b^j f *}$

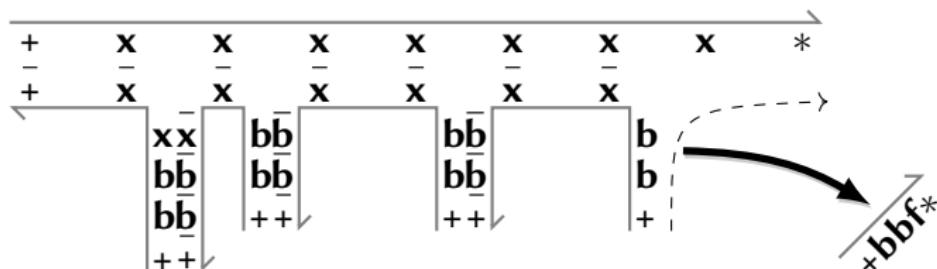
4. $\overrightarrow{+b^2 f *} \wedge \overrightarrow{+b^3 f *} \wedge \cdots \wedge \overrightarrow{+b^{\sqrt{n}} f *} \rightarrow \overrightarrow{+*}$



Primality

For $2 \leq j \leq \sqrt{n}$ and $k \leq j - 1$:

1. $\overrightarrow{+b^j \bar{x} \bar{x} +}$
2. $\overrightarrow{+b^j \bar{x}^j \bar{b}^j +}$
3. $\overrightarrow{+b^j \bar{x}^k *} \rightarrow \overrightarrow{+b^j f *}$
4. $\overrightarrow{+b^2 f *} \wedge \overrightarrow{+b^3 f *} \wedge \cdots \wedge \overrightarrow{+b^{\sqrt{n}} f *} \rightarrow \overrightarrow{+*}$



Primality

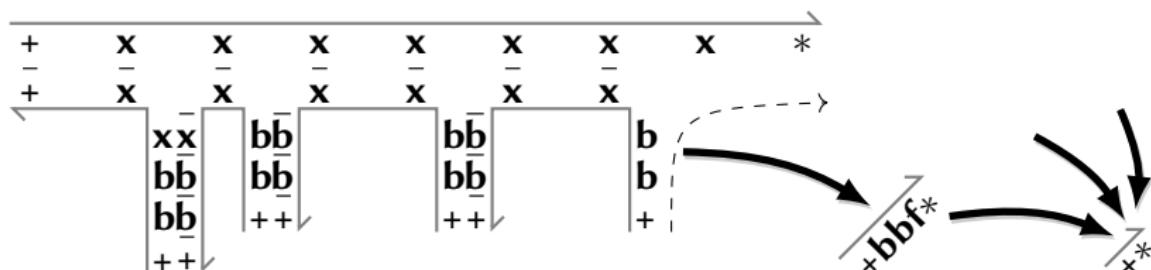
For $2 \leq j \leq \sqrt{n}$ and $k \leq j - 1$:

1. $\overrightarrow{+b^j \underline{x} \underline{x} +}$

2. $\overrightarrow{+b^j \underline{x}^j \underline{b}^j +}$

3. $\overrightarrow{+b^j \underline{x}^k *} \rightarrow \overrightarrow{+b^j f *}$

4. $\overrightarrow{+b^2 f *} \wedge \overrightarrow{+b^3 f *} \wedge \cdots \wedge \overrightarrow{+b^{\sqrt{n}} f *} \rightarrow \overrightarrow{+*}$



Primality

For $2 \leq j \leq \sqrt{n}$ and $k \leq j - 1$:

1. $\overrightarrow{+b^j xx+}$
2. $\overrightarrow{+b^j x^j b^j +}$
3. $\overrightarrow{+b^j x^k *} \rightarrow \overrightarrow{+b^j f*}$
4. $\overrightarrow{+b^2 f*} \wedge \overrightarrow{+b^3 f*} \wedge \cdots \wedge \overrightarrow{+b^{\sqrt{n}} f*} \rightarrow \overrightarrow{+*}$

- ▶ $O(n)$ strands
- ▶ Binary Encoding: $O(\sqrt{n} \cdot \log n)$ strands
- ▶ Good: $O(\sqrt{n} \cdot \log n) \subsetneq O\left(\frac{n}{\log n}\right)$

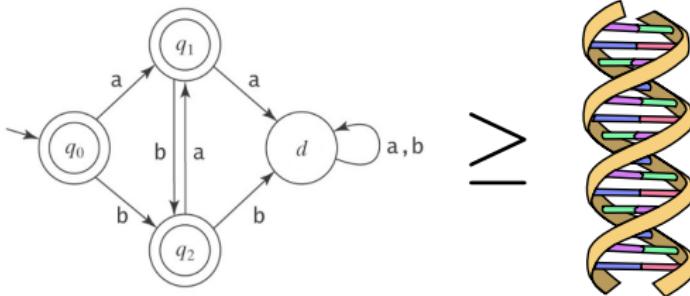
Primality

For $2 \leq j \leq \sqrt{n}$ and $k \leq j - 1$:

1. $\overrightarrow{+b^j xx+}$
2. $\overrightarrow{+b^j x^j b^j +}$
3. $\overrightarrow{+b^j x^k *} \rightarrow \overrightarrow{+b^j f*}$
4. $\overrightarrow{+b^2 f*} \wedge \overrightarrow{+b^3 f*} \wedge \cdots \wedge \overrightarrow{+b^{\sqrt{n}} f*} \rightarrow \overrightarrow{+*}$

- ▶ $O(n)$ strands
- ▶ Binary Encoding: $O(\sqrt{n} \cdot \log n)$ strands
- ▶ Good: $O(\sqrt{n} \cdot \log n) \subsetneq O\left(\frac{n}{\log n}\right)$

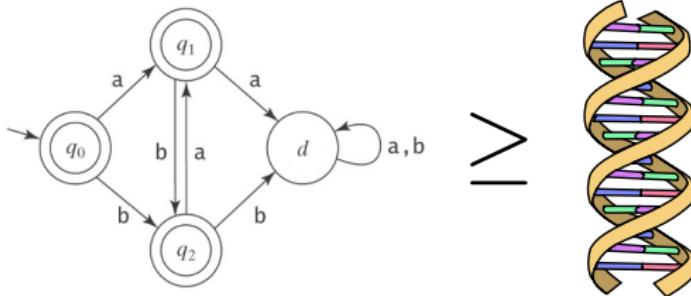
Constant Size Algorithms?



Theorem

Let \mathcal{A} be a DNA algorithm.

*If \mathcal{A} , interpreted as a set of strands, has finite size,
then \mathcal{A} decides a regular language.*



Theorem

Let \mathcal{A} be a DNA algorithm.

If \mathcal{A} , interpreted as a set of strands, is a regular language,
then \mathcal{A} decides a regular language.

Basic idea: Define language \mathcal{I} such that

1. \mathcal{I} is regular,
2. every $\iota \in \mathcal{I}$ is accepted by \mathcal{A} , and
3. every ι accepted by \mathcal{A} is in \mathcal{I} .

The Language \mathcal{I} ...

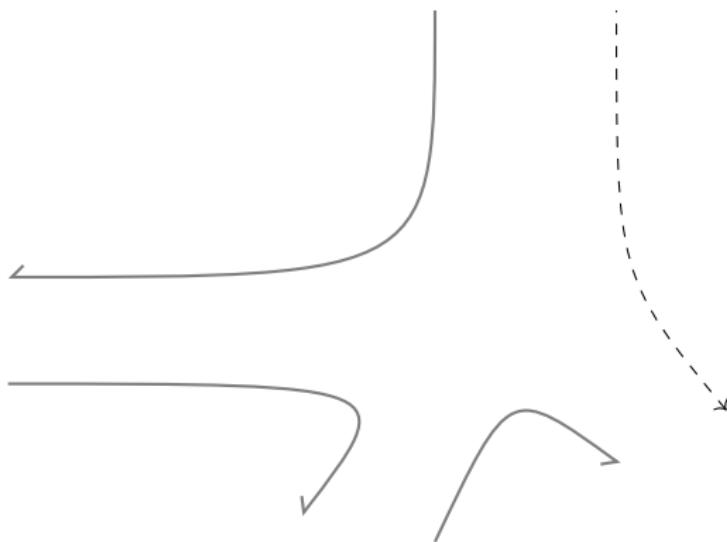
$$\mathcal{I} := \bigcup_{w: \delta(w, *) \in F} \mathcal{X}(\delta(s, +), w, Q \times Q \setminus \{(\delta(s, +), w)\}),$$

$$\mathcal{X}(v, w, J) := \bigcup_{q \in I(v, w, J)} \left(\bigcap_{\substack{(q_i, q_{i+1}): \\ q_i \neq \text{NULL} \neq q_{i+1}}} C(q_i, q_{i+1}, J) \cap \bigcap_{i: q_i = \text{NULL}} D(q_{i-1}, q_{i+1}, J) \right)$$

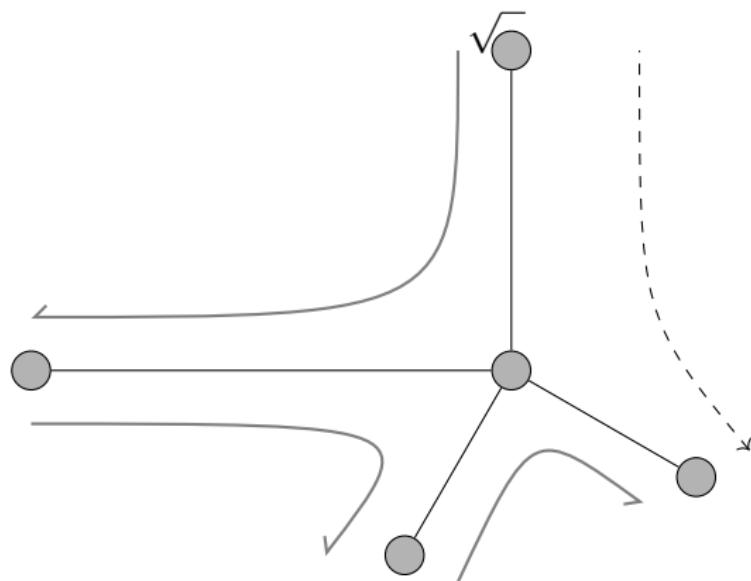
$$C(x, y, J) := \begin{cases} +\Sigma^* *, & \text{if } \exists(z_1, z_2) \in H_{x, y, J} \text{ s.t. } z_2 = s \text{ and } z_1 \in F \\ \bigcup_{(z_1, z_2) \in H_{x, y, J}} \mathcal{X}(z_1, z_2, J \setminus \{(x, y), (z_1, z_2)\}), & \text{otherwise.} \end{cases}$$

$$D(x, y, J) := \bigcup_{K \subseteq J \setminus \{(x, y)\}} \left(\overline{\mathcal{L}(B_{K, x, y})} \cap \bigcap_{(z_1, z_2) \in K} \mathcal{X}(z_1, z_2, J \setminus \{(x, y), (z_1, z_2)\}) \right)$$

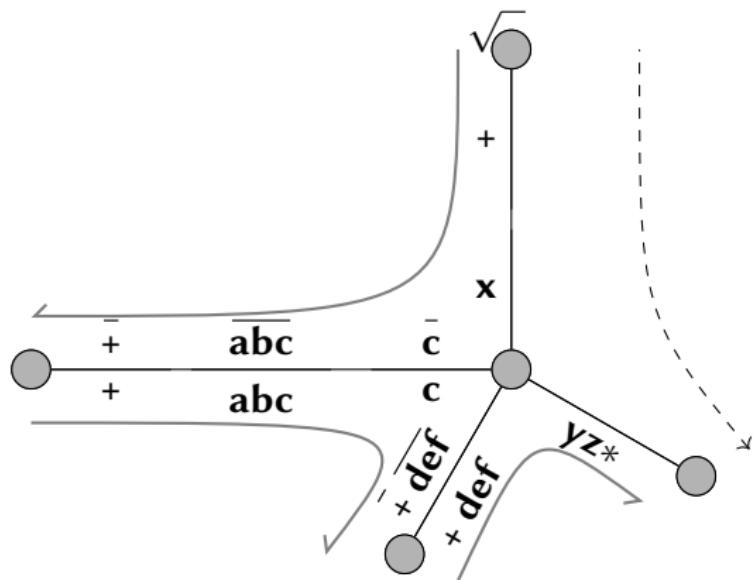
Assemblies



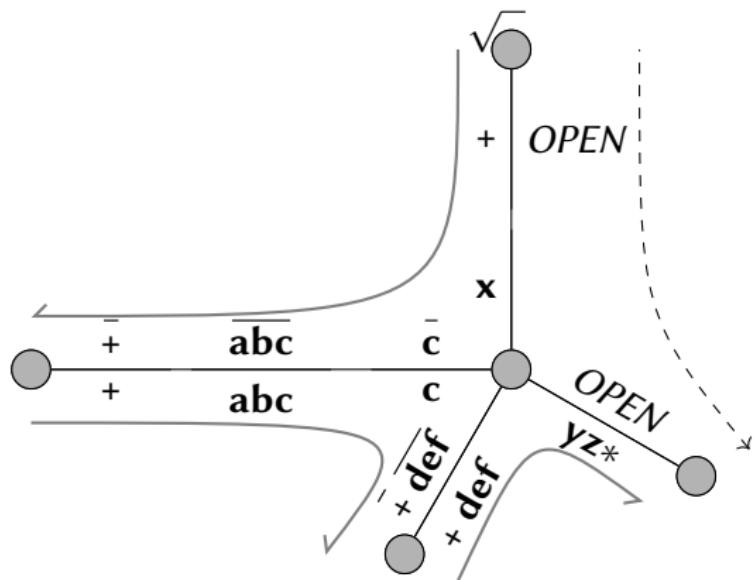
Assemblies



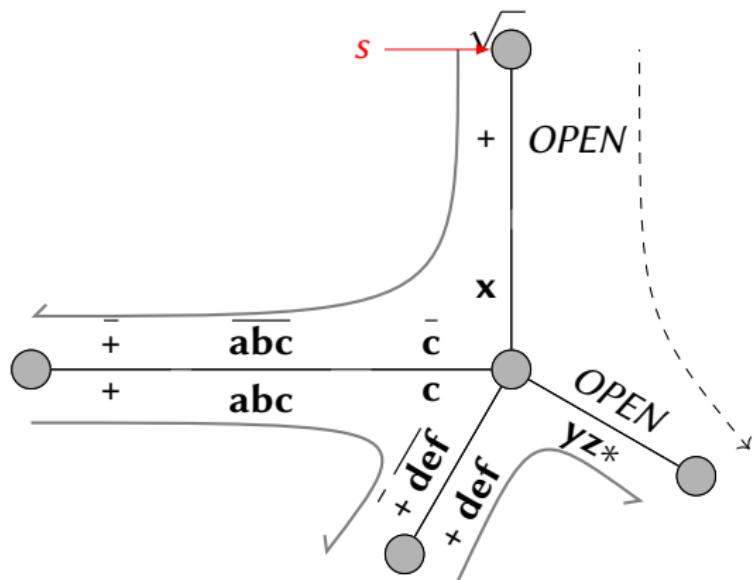
Assemblies



Assemblies

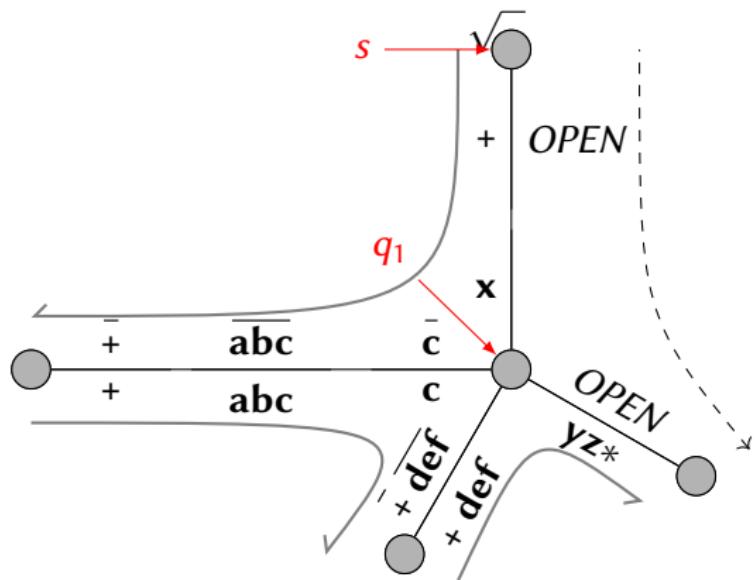


Assemblies & Automata



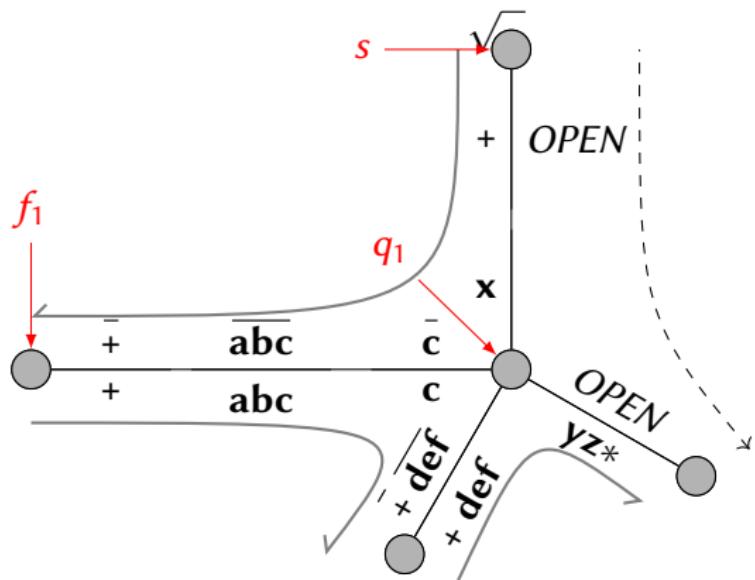
DNA Algorithm \approx Finite Automaton
 $(Q, \Sigma, \delta, s, F)$

Assemblies & Automata



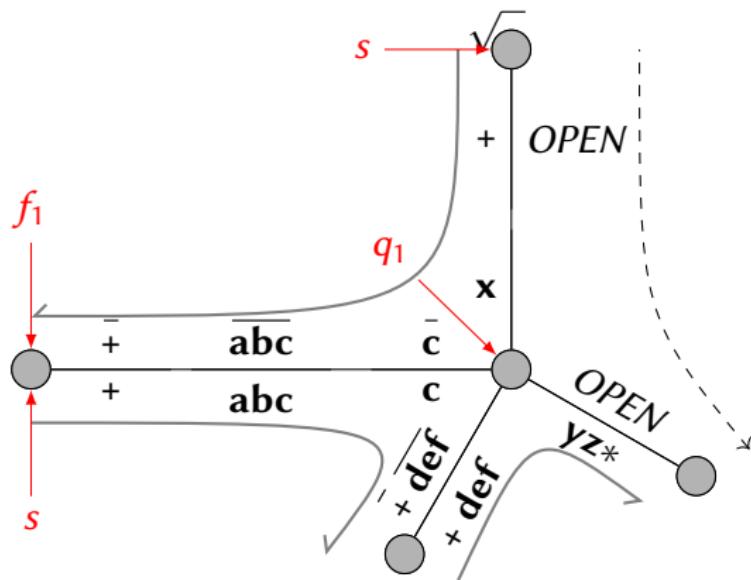
DNA Algorithm \approx Finite Automaton
 $(Q, \Sigma, \delta, s, F)$

Assemblies & Automata



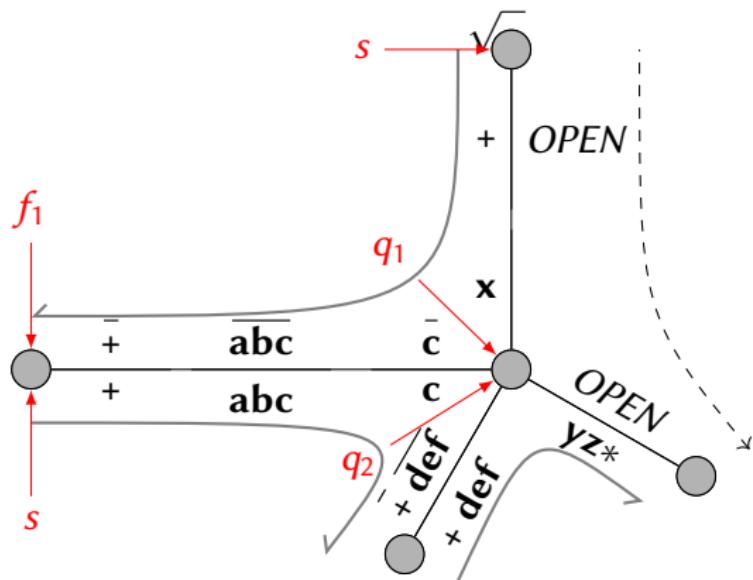
DNA Algorithm \approx Finite Automaton
 $(Q, \Sigma, \delta, s, F)$

Assemblies & Automata



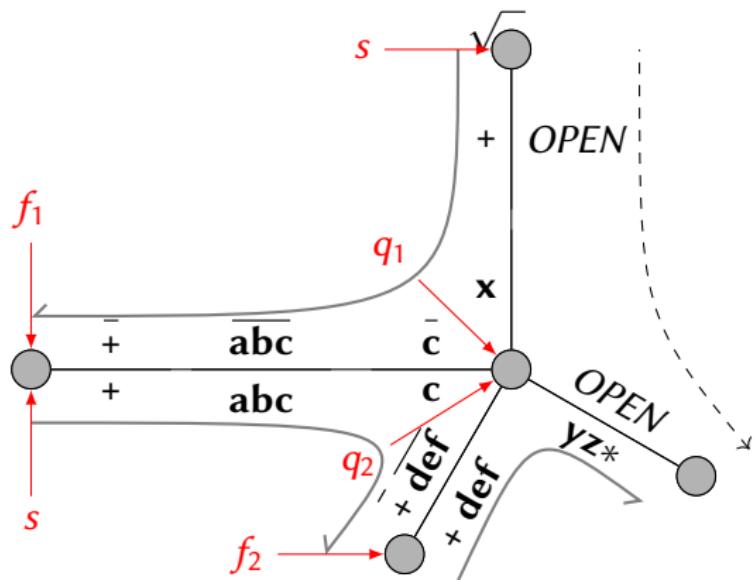
DNA Algorithm \approx Finite Automaton
 $(Q, \Sigma, \delta, s, F)$

Assemblies & Automata



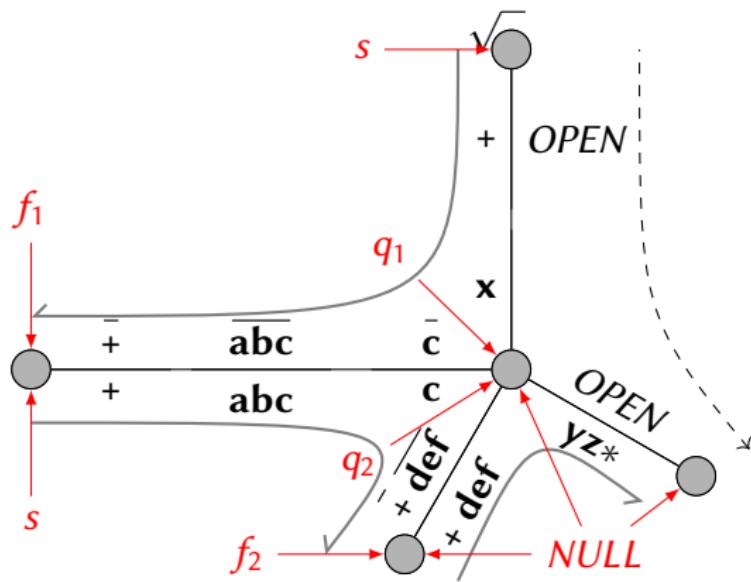
DNA Algorithm \approx Finite Automaton
 $(Q, \Sigma, \delta, s, F)$

Assemblies & Automata



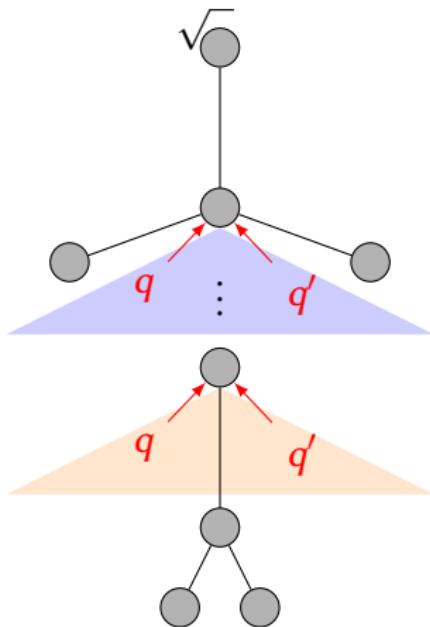
DNA Algorithm \approx Finite Automaton
 $(Q, \Sigma, \delta, s, F)$

Assemblies & Automata



DNA Algorithm \approx Finite Automaton
 $(Q, \Sigma, \delta, s, F)$

Minimal Assemblies



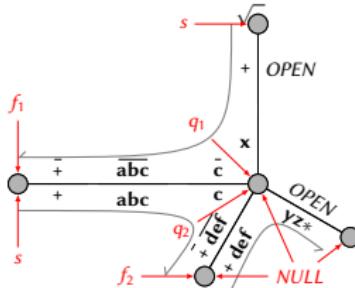
The Laborious Part ...

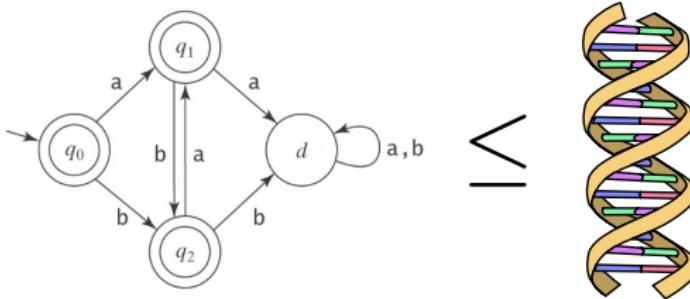
$$\mathcal{I} := \bigcup_{w: \delta(w, *) \in F} \mathcal{X}(\delta(s, +), w, Q \times Q \setminus \{(\delta(s, +), w)\})$$

$$\mathcal{X}(v, w, J) := \bigcup_{q \in I(v, w, J)} \left(\bigcap_{\substack{(q_i, q_{i+1}): \\ q_i \neq \text{NULL} \neq q_{i+1}}} C(q_i, q_{i+1}, J) \cap \bigcap_{i: q_i = \text{NULL}} D(q_{i-1}, q_{i+1}, J) \right)$$

$$C(x, y, J) := \begin{cases} +\Sigma^* *, & \text{if } \exists (z_1, z_2) \in H_{x, y, J} \text{ s.t. } z_2 = s \text{ and } z_1 \in F \\ \bigcup_{(z_1, z_2) \in H_{x, y, J}} \mathcal{X}(z_1, z_2, J \setminus \{(x, y), (z_1, z_2)\}), & \text{otherwise.} \end{cases}$$

$$D(x, y, J) := \bigcup_{K \subseteq I \setminus \{(x, y)\}} \left(\overline{\mathcal{L}(B_{K, x, y})} \cap \bigcap_{(z_1, z_2) \in K} \mathcal{X}(z_1, z_2, J \setminus \{(x, y), (z_1, z_2)\}) \right)$$

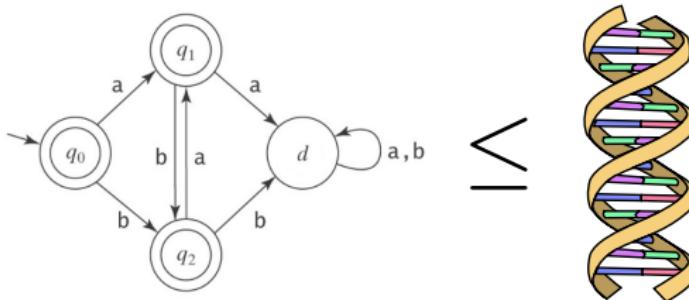




Theorem

Let \mathcal{L} be a regular language.

There is a constant size DNA algorithm that decides $+\mathcal{L}^*$.



Theorem

Let \mathcal{L} be a regular language.

There is a constant size DNA algorithm that decides $+\mathcal{L}^*$.

- ▶ Regular Language \rightarrow FA $(Q, \Gamma, \delta, s, F)$
- ▶ $\xrightarrow{+s+}$ for the starting state s
- ▶ $\xrightarrow{+q'xq+}$ for all possible transitions $\delta(q, x) = q'$
- ▶ $\xrightarrow{*f*}$ for all $f \in F$.

Summary

