De Bruijn Superwalk with Multiplicities Problem is NP-hard

Evgeny Kapun Fedor Tsarev
Genome Assembly Algorithms Laboratory
University ITMO, St. Petersburg, Russia

April 11, 2013

Genome Assembly Models

- Shortest Common Superstring – NP-hard (Gallant et al., 1980).
- Shortest de Bruijn Superwalk – NP-hard (Medvedev et al., 2007).

Genome Assembly Models

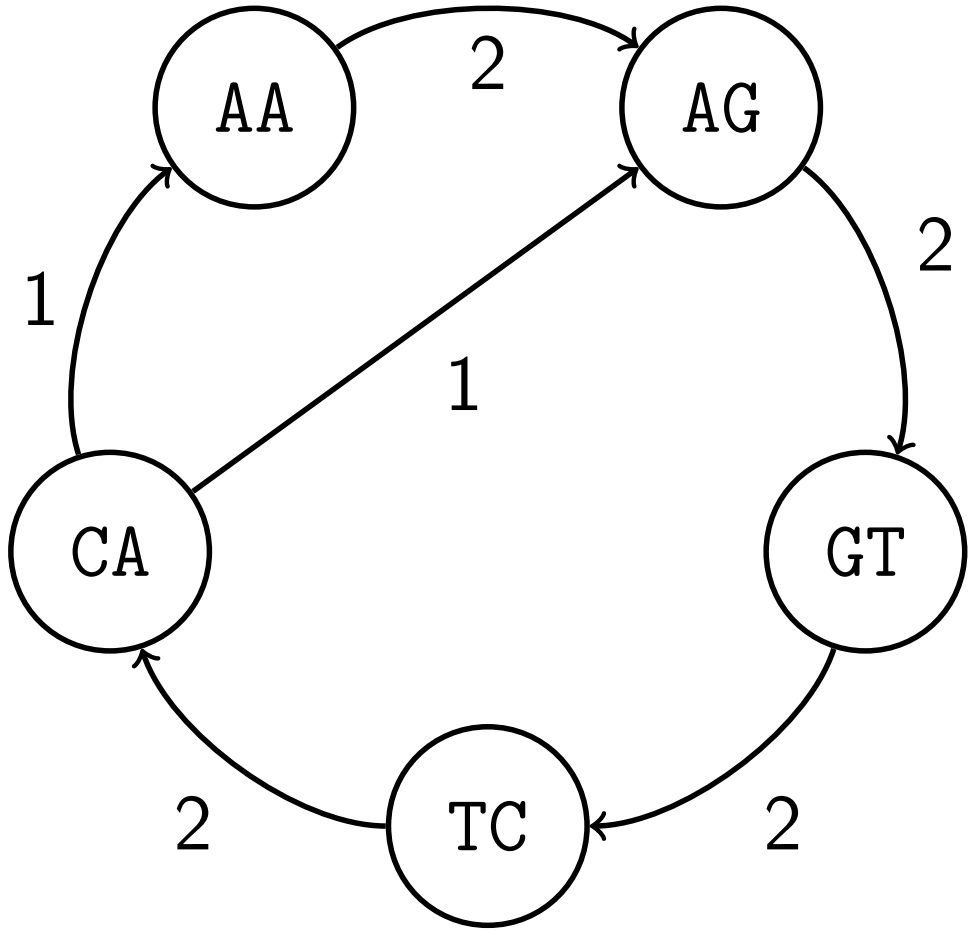
- Maximum likelihood approach – to find edges' multiplicities (Medvedev and Brudno, 2009; Varna et al., 2011).
- De Bruijn Superwalk with Multiplicities – complexity was not known. This talk – we prove it is NP-hard.

De Bruijn Superwalk with Multiplicities Problem

Find a walk in the de Bruijn graph containing several walks as subwalks and passing through each edge the exactly predefined number of times.

Example

- Reads = subwalks: `AAGT`, `AGTCA`, `TCAA`



- Superwalk: `AAGTCAGTCAAG`

NP-hardness Proof Outline

1. Reduce Shortest Common Superstring problem to Common Superstring with Multiplicities problem.
2. Reduce Common Superstring with Multiplicities problem to De Bruijn Superwalk with Multiplicities problem.

NP-hardness Proof Outline

1. Reduce Shortest Common Superstring problem to Common Superstring with Multiplicities problem.

Common Superstring with Multiplicities Problem

Find a string containing several strings as substrings and containing each character the exactly predefined number of times.

Example

- Strings: `AAGT`, `AGTCA`, `TCAA`
- Multiplicities: $m(\texttt{A}) = 5$, $m(\texttt{C}) = 2$, $m(\texttt{G}) = 3$, $m(\texttt{T}) = 2$
- Solution for SCS: `AAGTCAA`
- Solution for CSM: `AAGTCAGTCAAG` or just `AAGTCAAACGGT`

Reducing SCS to CSM

Given an instance of SCS with $\Sigma = \{0, 1\}$ in decision form ("Is there such a string that ..."), substitute

$$0 \longrightarrow T_0 = 000111$$

$$1 \longrightarrow T_1 = 001011$$

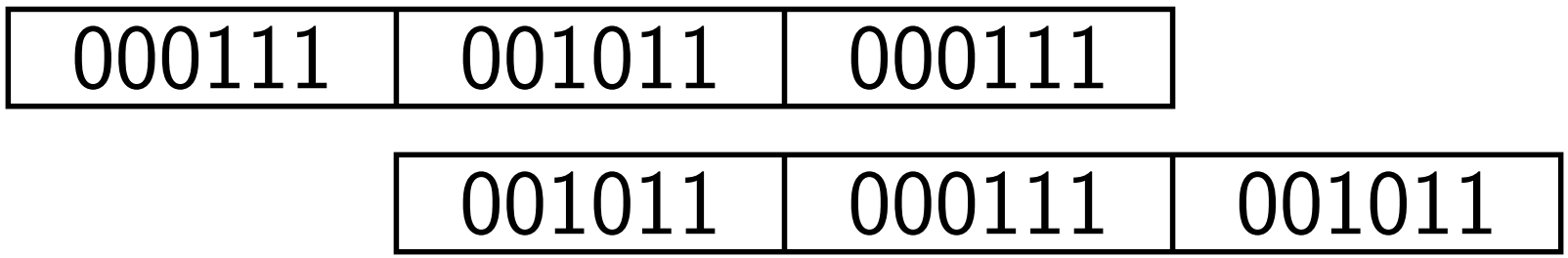
and make the multiplicities of 0 and 1 equal to 3 times the desired superstring length.

Properties of T_0 and T_1

- T_0 and T_1 have the same length.
- Furthermore, number of occurrences of each character is the same in T_0 and T_1.
- No proper suffix of either T_0 or T_1 is equal to any of the proper prefixes of either T_0 or T_1.

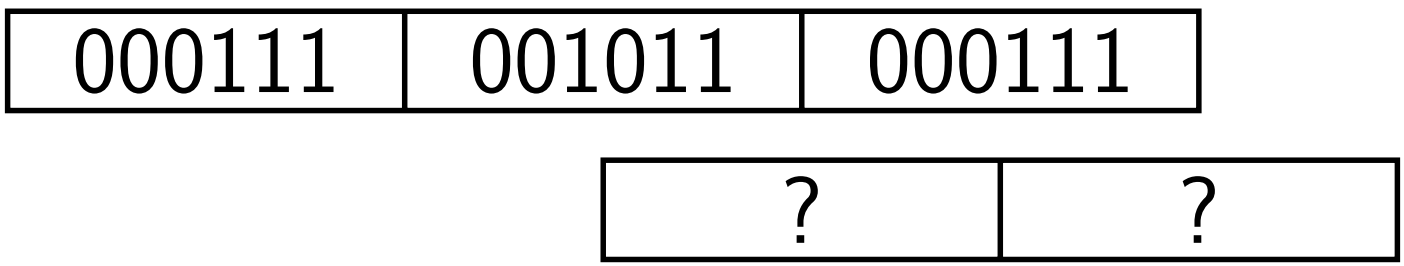
Properties of T_0 and T_1

As a result, all overlaps of the transformed strings are aligned.



Properties of T_0 and T_1

Unaligned overlaps are impossible because no proper prefix of T_0 and T_1 is equal to any proper suffix.



Reducing SCS to CSM

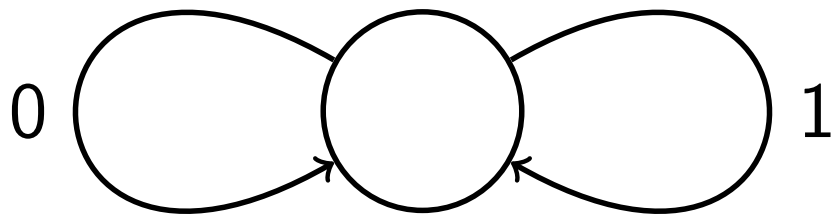
As a result, the shortest common superstring of the transformed strings would be equal to the transformed shortest common superstring of the original strings.

NP-hardness Proof Outline

2. Reduce Common Superstring with Multiplicities problem to De Bruijn Superwalk with Multiplicities problem.

Reducing CSM to DBSM

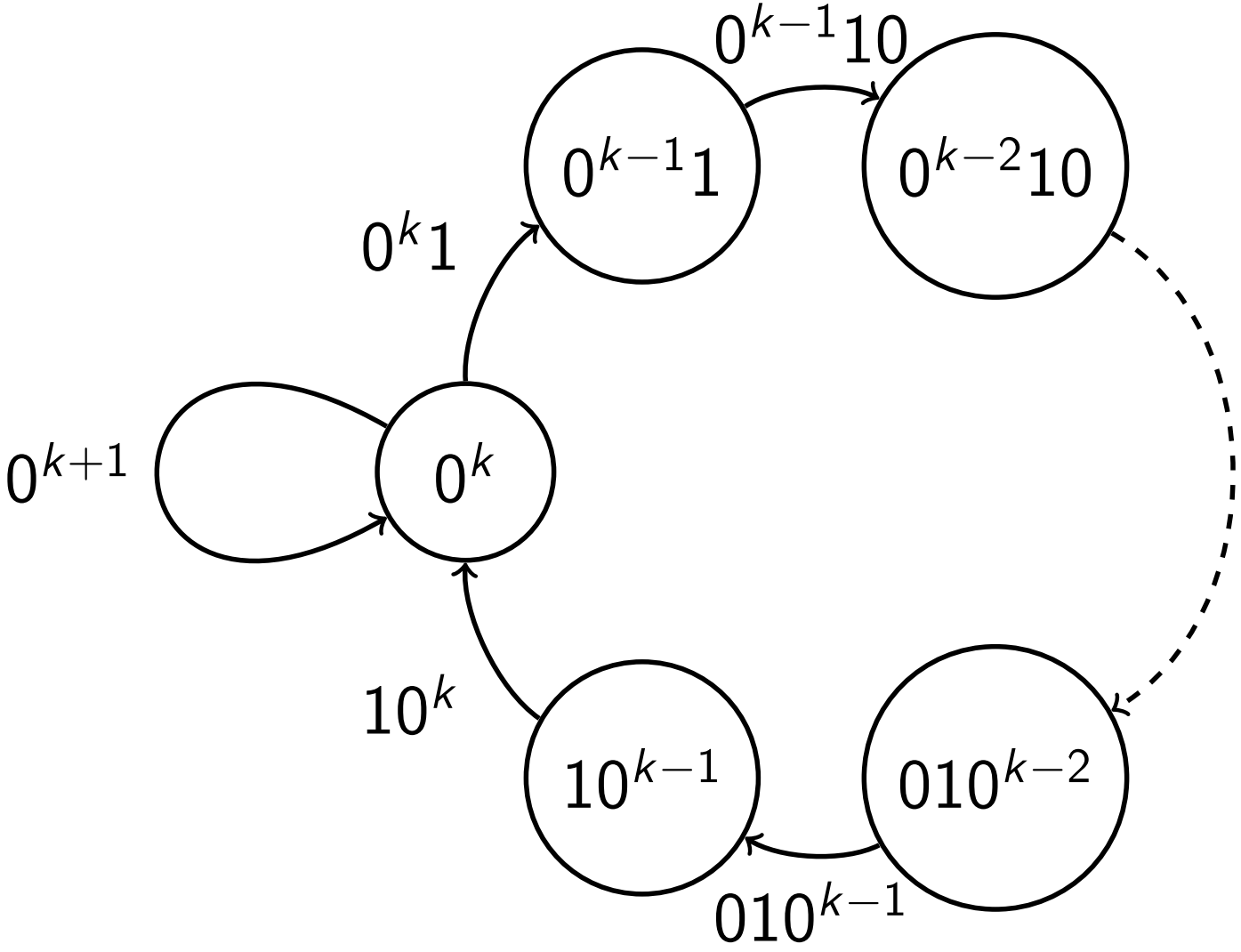
Trivial reduction ($\Sigma = \{0, 1\}$, $k = 0$):



Strings become walks, multiplicities of characters become multiplicities of edges.

Reducing CSM to DBSM

Generalization for any k:



Result

De Bruijn Superwalk with Multiplicities problem is NP-hard for any $|\Sigma| \geq 2$ and any k. Since the case $|\Sigma| = 1$ is trivial, the problem is NP-hard for all nontrivial cases.

Ackowledgements

Funding:

- Ministry of Education and Science of Russian Federation (contract 16.740.11.0495, agreement 14.B37.21.0562)
- University ITMO (research project 610455)

Thank you! Questions?