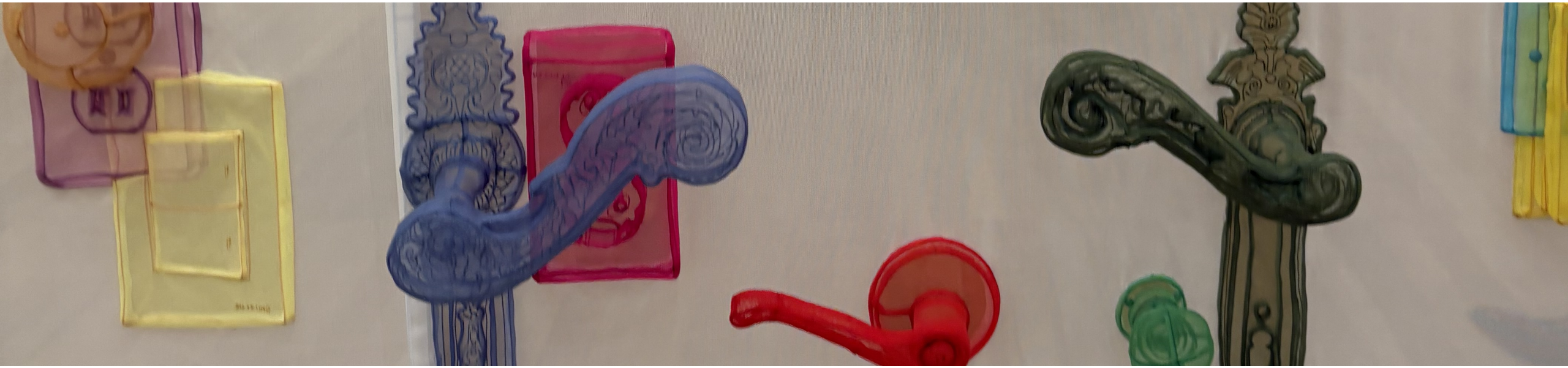
CWI
Research institute for mathematics & computer science in the Netherlands

VU VRIJE UNIVERSITEIT AMSTERDAM

When is String Reconstruction using de Bruijn Graphs Hard?

ESA 2025

Ben Bals
Centrum Wiskunde & Informatica and VU Amsterdam



What math has done for physics, computer science will do for biology.

Stanislaw Ulam

String Reconstruction

ATTAATTATA

k=3

ATT TAA ATT TAT TTA TTA ATA AAT

AT TT TA AA

Theorem. Eulerian trails in de Bruijn graphs are exactly the string reconstructions.

Domain Knowledge

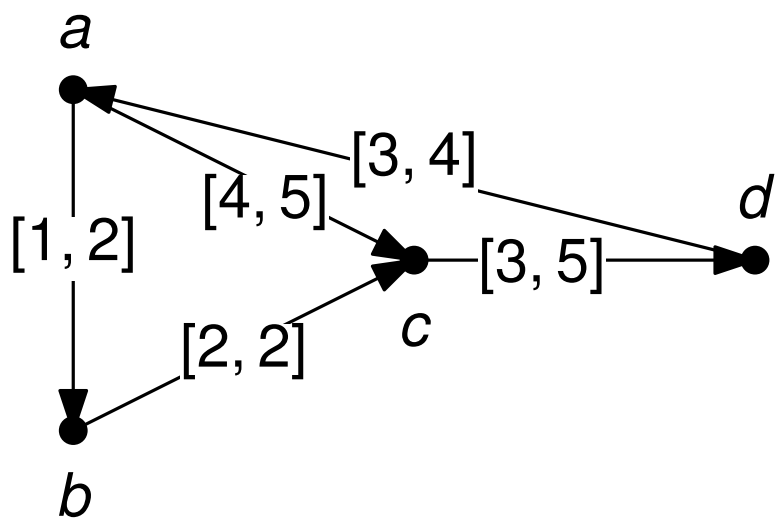
What if I know that `ATT` always appears towards the start?

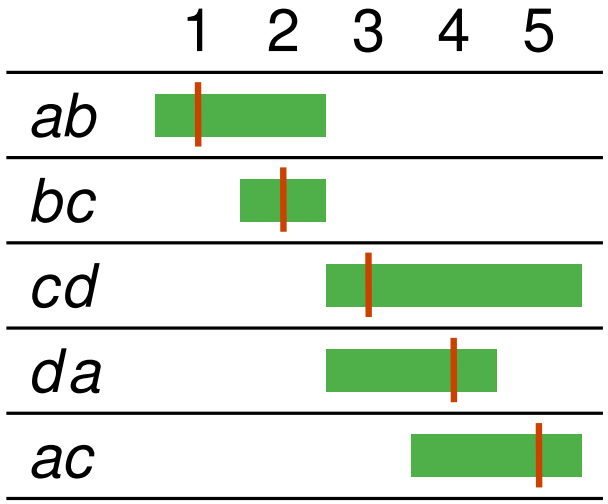
Problem. Eulerian Trails in Digraphs with Interval Cost functions (diET)

Given: Directed graph $G = (V, E)$, interval function $c\colon E \to \mathcal{I}_m$

Decide: Is there an Eulerian trail $P = e_1 \dots e_m$ in G such that for all $t \in [m]$, $t \in c(e_t)$?

Map edge e to interval $c(e)$ from $[1, m]$





For some timestep t, we call an edge e **available at** t if $t \in c(e)$.

Our Results

CWI Research institute for mathematics & computer science in the Netherlands

VU VRIJE UNIVERSITEIT AMSTERDAM

Parameters

- w Width of the largest interval in c
- k Fragment length for the dBG

	Known	Our Results
Algorithmic	$\mathcal{O}(m \cdot 4^w)$ [BPSS02] [BM23]	$\mathcal{O}(m \cdot 4^{w \log w/k})$
Hardness	NP-hard for dBG with $\lvert\Sigma\rvert \geq 4$ [BPSS02]	NP-hard for dBG with $\lvert\Sigma\rvert \geq 2$

(Hiding non-exponential factors of w.)

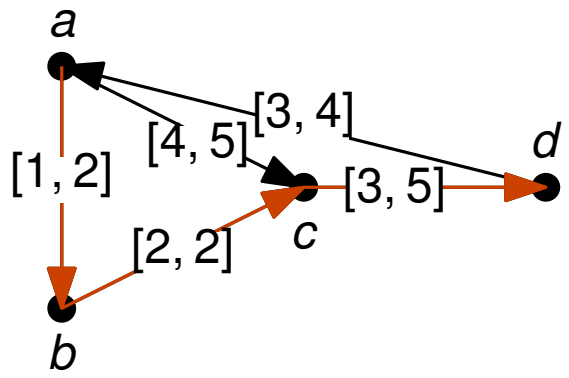
It's enough if w/k is small!
In practice, k is often large
(e.g., accurate DNA reads)

Not related to k at all

Also in the paper

Counting extensions for state-of-the-art and our new algorithms

Ben-Dor et al. Algorithm



Theorem. There is a $\mathcal{O}(m \cdot w^{1.5} 4^w)$–time algorithm for diET.

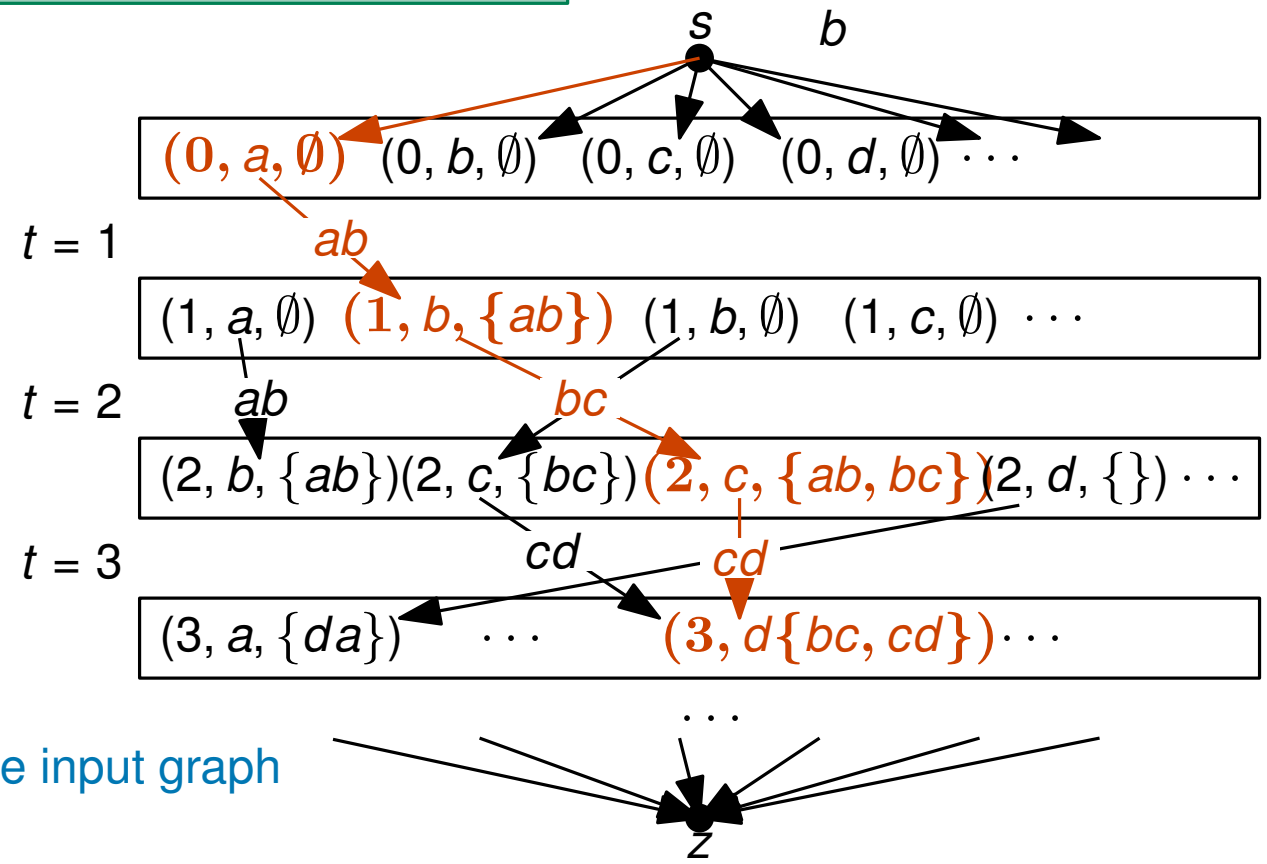
Construct a state graph

Every state (t, v, S) consists of

- t — a timestep
- v — a vertex
- $S \subseteq E$ — a set of unavailable edges

Add the state edge $(t, v, S) \to (t + 1, u, S')$ if

- vu is an edge available at $t + 1$
- vu is not forbidden (i.e., not in S)
- S' includes $v \to u$ and all edges from S that are still available at $t + 1$



Path through the state graph ≡ Eulerian trail in the input graph

Core insight. We do not need to remember edges that are not available anymore.

Ben-Dor et al. Algorithm—Analysis

Theorem. There is a $\mathcal{O}(m \cdot w^{1.5} 4^w)$–time algorithm for diET.

Core insight. We do not need to remember edges that are not available anymore.

Lemma. At any timestep t, there are at most $2w - 1$ available edges.

Proof idea.

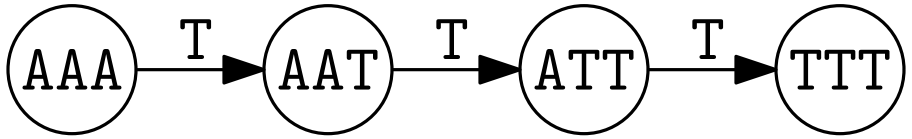
- Any edge is available for at most w timesteps.
- Therefore, if it is available in t, it must be used within the interval $[t - w + 1, t + w]$.
- This interval has width $2w - 1$.

Corrollay. At any timestep, there are at most $2^{2w-1} \leq 4^w$ choices for S.

Doing Better With Combinatorics

In dBGs, a node already encodes the labels of the last $k-1$ edges!

How many paths of length 3 are there between these two nodes?



$k = 4$

Core insight. We can remember the last w edges as a string!

Every state (t, v, S) consists of

- t — a timestep
- $\alpha \in \Sigma^{w+k-1}$ — a string
  - The substrings of length k uniquely describe the the last w edges
  - Naively describing these edges would take kw characters

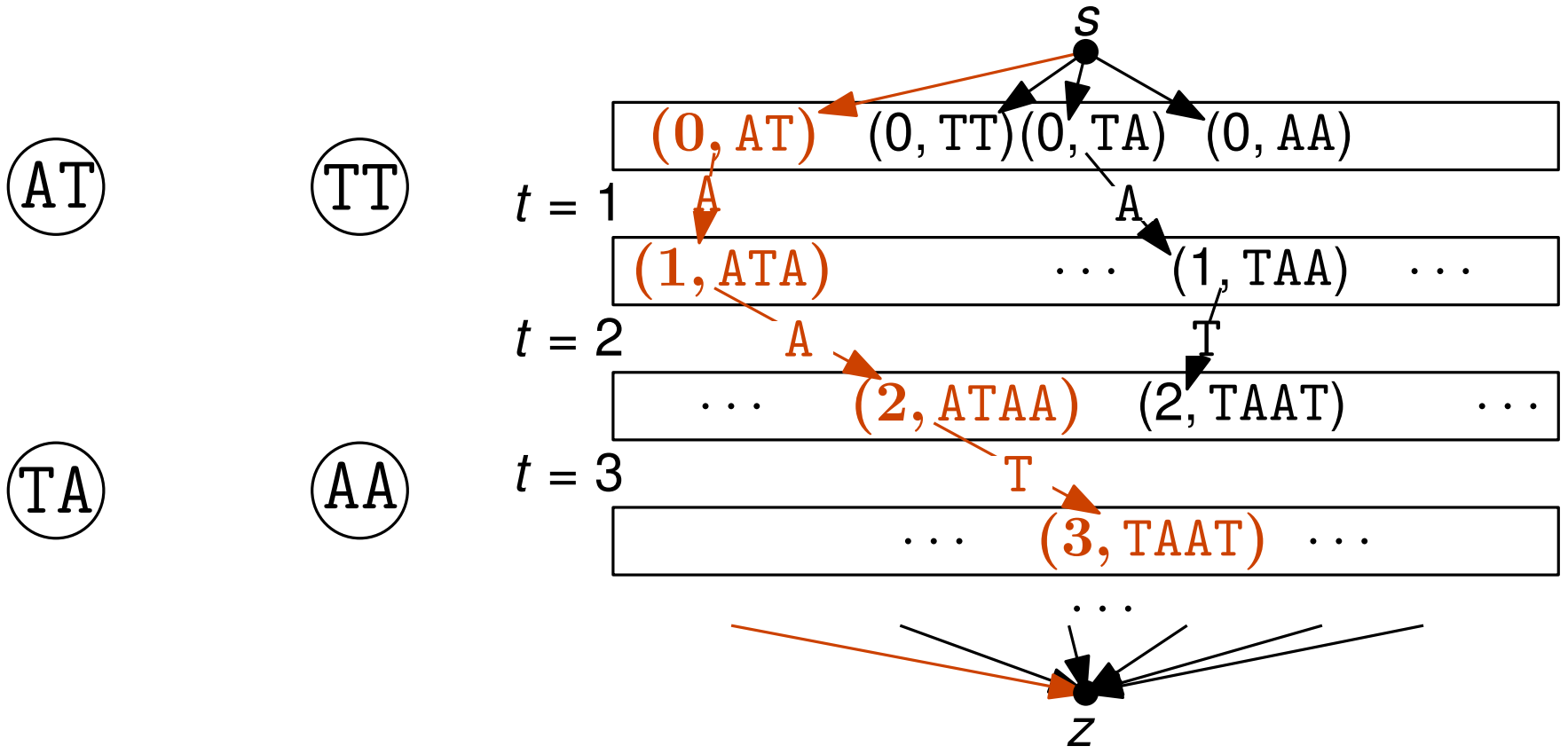
This is where the improvement by a factor k comes from!

Theorem. There is a $\mathcal{O}(m \cdot 4^{\frac{w \log w}{k}})$–time algorithm for diET on dBGs.

Doing Better With Combinatorics

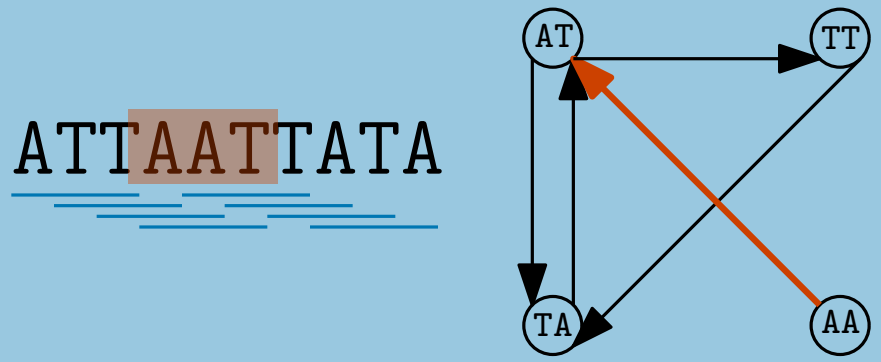
Core insight. We can remember the last w edges as a string!

Theorem. There is a $\mathcal{O}(m \cdot 4^{\frac{w \log w}{k}})$–time algorithm for diET on dBGs.

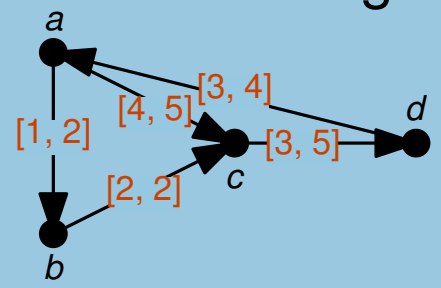


Summary

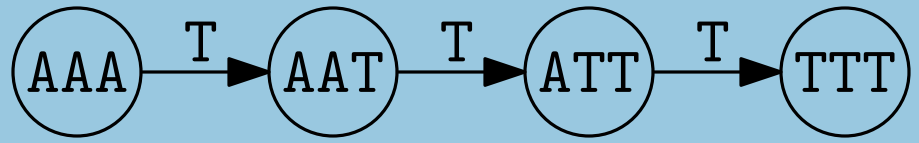
String Reconstruction using dBGs



Use Parametrization to Capture Domain Knowledge



Combinatorial Insights for Faster Algorithms



References

[BM23] Bumpus, Benjamin Merlin, and Kitty Meeks. “Edge Exploration of Temporal Graphs.” arXiv:2103.05387. Preprint, arXiv, November 17, 2021. https://doi.org/10.48550/arXiv.2103.05387.

[BPSS02] Ben-Dor, A., I. Pe’er, R. Shamir, and R. Sharan. “On the Complexity of Positional Sequencing by Hybridization.” Journal of Computational Biology 8, no. 4 (2001): 361–71. https://doi.org/10.1089/106652701752236188.

[HFLSP96] Hannenhalli, Sridhar, William Feldman, Herbert F. Lewis, Steven S. Skiena, and Pavel A. Pevzner. “Positional Sequencing by Hybridization.” Bioinformatics 12, no. 1 (1996): 19–24. https://doi.org/10.1093/bioinformatics/12.1.19.

Bonus: Another Application

Differential Privacy

Goal: Share data while

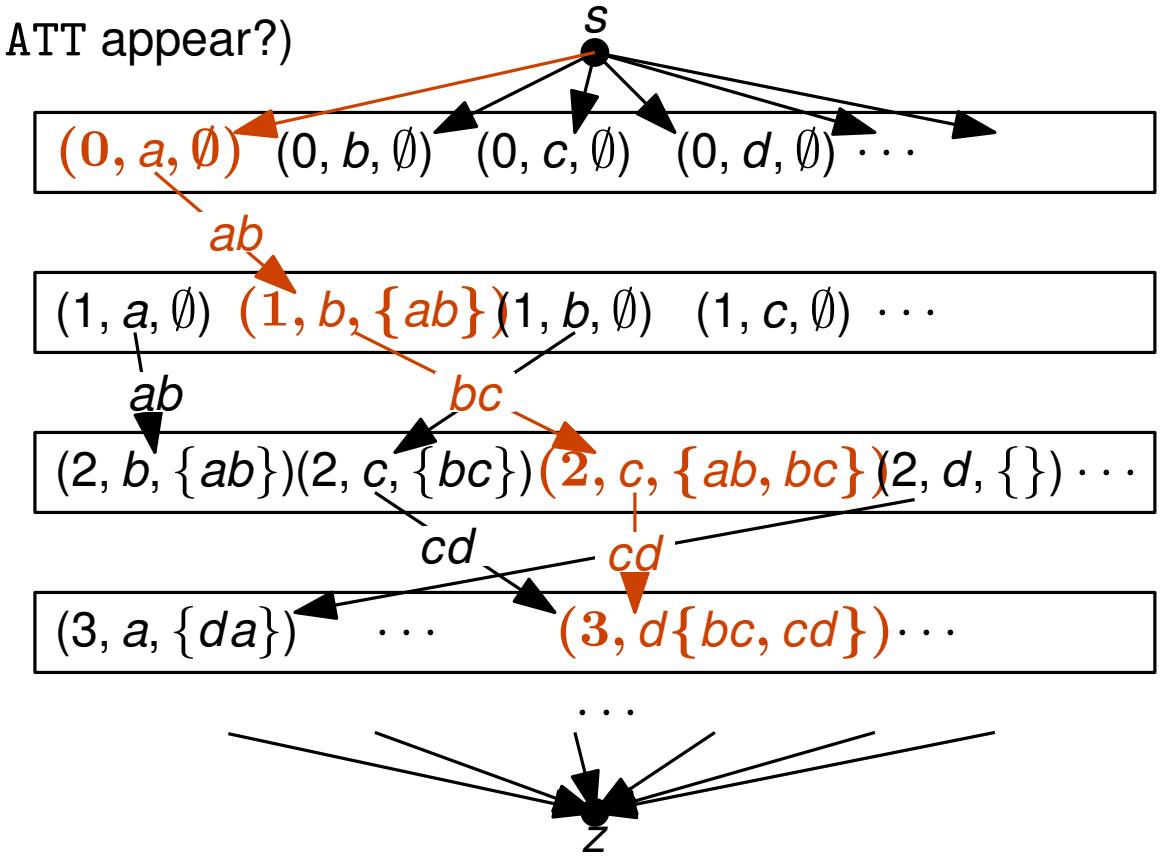
- Allowing pattern matching (e.g., ”how many times does `ATT` appear?)
- Preventing reconstruction

Idea: Share all substrings of length k such that

- There are many possible reconstructions (over some threshold z)
- $\equiv$ where there are many Eularian trails in the dBG
- This should also hold given some domain knowledge c

Approach: Count ETs (with or without c)

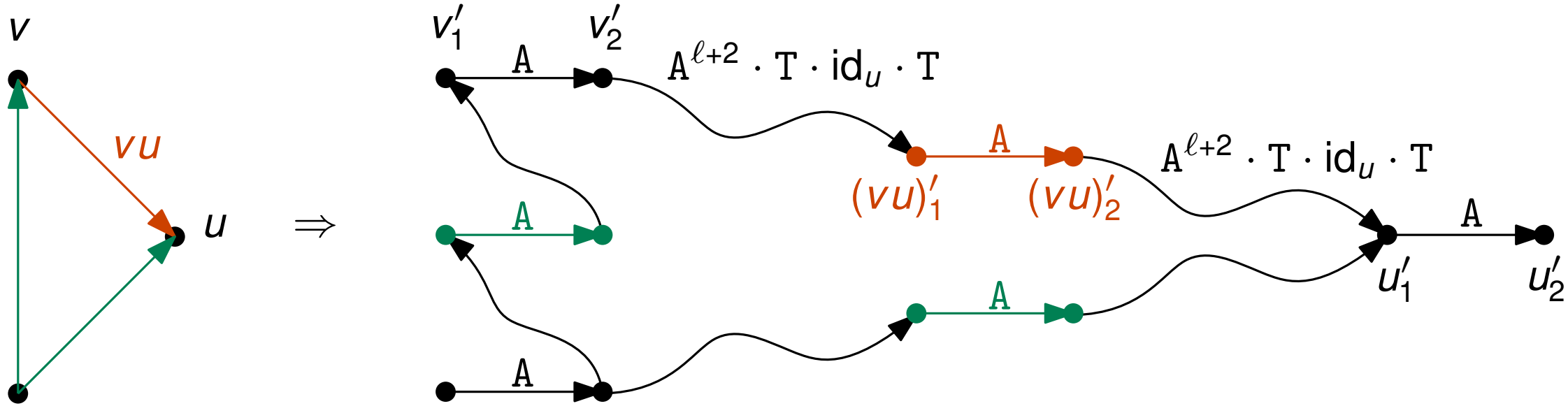
- Count s-t-paths in state graph



Bonus: NP-Hardness for $|\Sigma| = 2$

Set $\ell \coloneqq \lceil \log |V| \rceil$. Let $id_v \in \{\mathtt{A}, \mathtt{T}\}^\ell$ be a unique id for each $v \in V$

$v'_1 \coloneqq$	$\mathtt{A}^{\ell+3}$	$\mathtt{T}$	id_v	$\mathtt{T}$	$\mathtt{A}^{\ell+3}$	$\mathtt{T}$	id_v	$\mathtt{T}$	
$v'_2 \coloneqq$	$\mathtt{A}^{\ell+2}$	$\mathtt{T}$	id_v	$\mathtt{T}$	$\mathtt{A}^{\ell+3}$	$\mathtt{T}$	id_v	$\mathtt{T}$	$\mathtt{A}$
$(vu)'_1 \coloneqq$	$\mathtt{A}^{\ell+3}$	$\mathtt{T}$	id_v	$\mathtt{T}$	$\mathtt{A}^{\ell+3}$	$\mathtt{T}$	id_u	$\mathtt{T}$	
$(vu)'_2 \coloneqq$	$\mathtt{A}^{\ell+2}$	$\mathtt{T}$	id_v	$\mathtt{T}$	$\mathtt{A}^{\ell+3}$	$\mathtt{T}$	id_u	$\mathtt{T}$	$\mathtt{A}$



Bonus: NP-Hardness for $|\Sigma| = 2$

Three types of edges e

- Always available c(e) = [1,m]
- Only early $c(e) = [1, \tau]$
- Only late $c(e) = [\tau + 1, m]$

Idea. Choose τ such that is exactly enough time to visit n nodes with $n - 1$ edges