The Computational Complexity of Factored Graphs

Shreya Gupta **Boyang Huang** Russell Impagliazzo Stanley Woo Christopher Ye

University of California San Diego

Factored Graph

A **factored graph** is a **succinct graph representation** $G = f(G_1, \ldots, G_k)$ that combines input graphs $G_1, \ldots, G_k$ into a single graph G using **graph operations**.

Number

Arithmetic operation

$$(2 + 0) \times 2 - 5$$

Arithmetic Expression

Graph

Graph operation

$$(G_1 \square (G_2 \cup G_3)) \times G_4$$

Factored Graph Representation

Graph Operations

Given two graphs G and H, we consider the following three **binary graph operations**:

		Vertex Set	Condition for $(g_1, h_1) \sim (g_2, h_2)$
Cartesian product	$G \square H$	$V(G) \times V(H)$	$(g_1 = g_2 \wedge h_1 \sim h_2) \vee (h_1 = h_2 \wedge g_1 \sim g_2)$
Tensor product	$G \times H$	$V(G) \times V(H)$	$g_1 \sim g_2 \wedge h_1 \sim h_2$

		Vertex Set	Edge Set
Union	$G \cup H$	$V(G) \cup V(H)$	$E(G) \cup E(H)$

Applications

- Practical instances of graphs and data structures are often highly structured
  - Road networks
  - Databases
  - Compounds in molecular geometry
  - Finite automata

Question. In addition to being a method to compress graph data, *when can we also leverage the factored structure to derive a **"better"** algorithm?*

Measuring Factored Graph Complexity

We say that a factored graph $G = f(G_1, \dots, G_k)$ is **of complexity $(\boldsymbol{n}, \boldsymbol{k})$** if

1) each $\boldsymbol{G_i}$ **has at most $\boldsymbol{n}$ vertices**
2) the formula f uses **$\boldsymbol{k}$ input graphs***

*counting with multiplicities

Observation. Every graph G has two *trivial* factored graph representations:

1) complexity $(|V(G)|, 1)$

$$G = G.$$

2) complexity $(2, |E(G)|)$

$$G = \bigcup_{e \in E(G)} e\,.$$

Interesting factored graph representations **balance** n and k in some meaningful way.

Parameterized Complexity

For any graph problem, we can define a version where the **input is given as a factored graph**.

Observation. A factored graph of complexity (n, k) could have an explicit size of $\Omega(n^k)$.

Question. How does the factored representation affect the difficulty of a problem?

Parameterized Complexity

A parameterized problem is …

in XP if it can be solved in time $\boldsymbol{O\left(n^{f(k)}\right)}$ for some function f.

$\supsetneq$

fixed-parameter tractable (in FPT) if it can be solved in time $\boldsymbol{O\left(f(k)n^{O(1)}\right)}$ for some function f.

Observation. Any graph problem with poly-time algorithm has an $n^{O(k)}$-time algorithm on factored graphs of complexity (n, k) (i.e., in **XP**)

Question. Can we do better, i.e., in **FPT**?

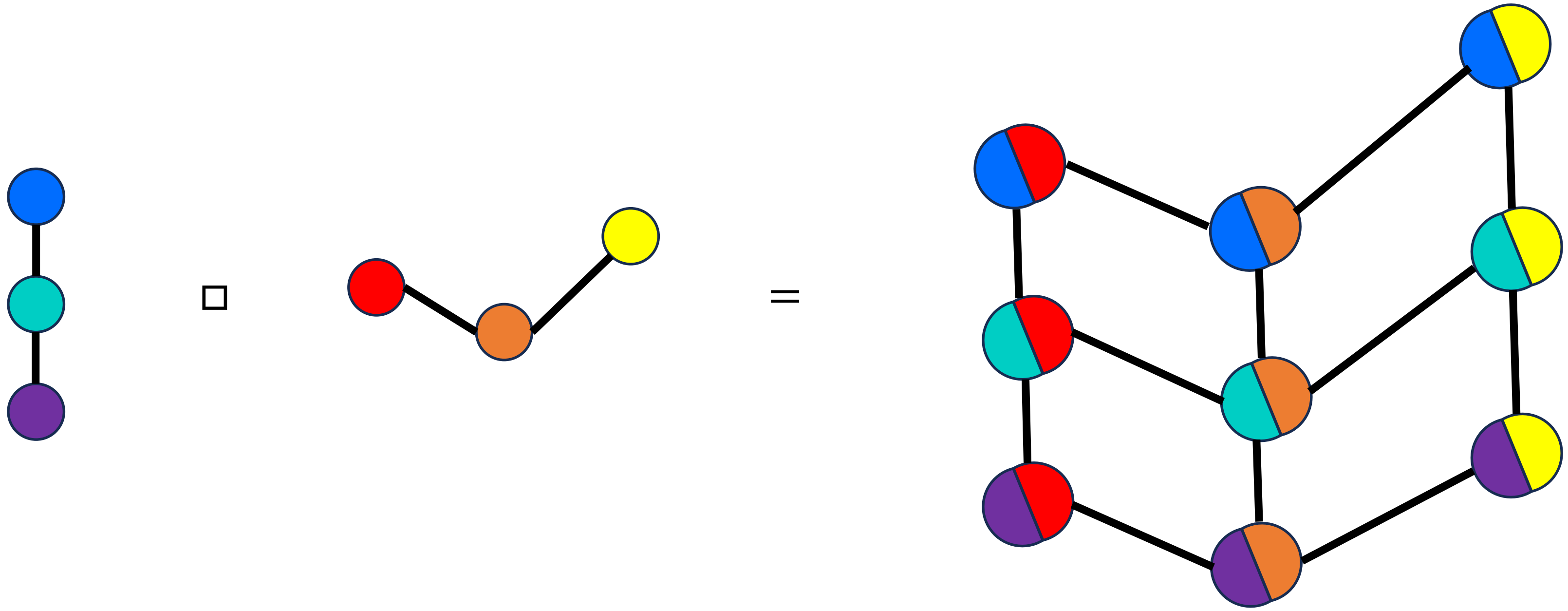
Our Results - Overview

Theorem. On factored graph inputs,
1) **Lexicographically First Maximal Independent Set (LFMIS)** is not in FPT.
2) **Counting Cliques** is in FPT.
3) **Reachability** is in FPT if and only if $\mathrm{NL} \subseteq \mathrm{DTIME}(n^C)$ for some absolute constant C.

Outline for the rest of the talk:
- More detailed definition of the graph operations
- Proof overview for Theorems 1) and 3)

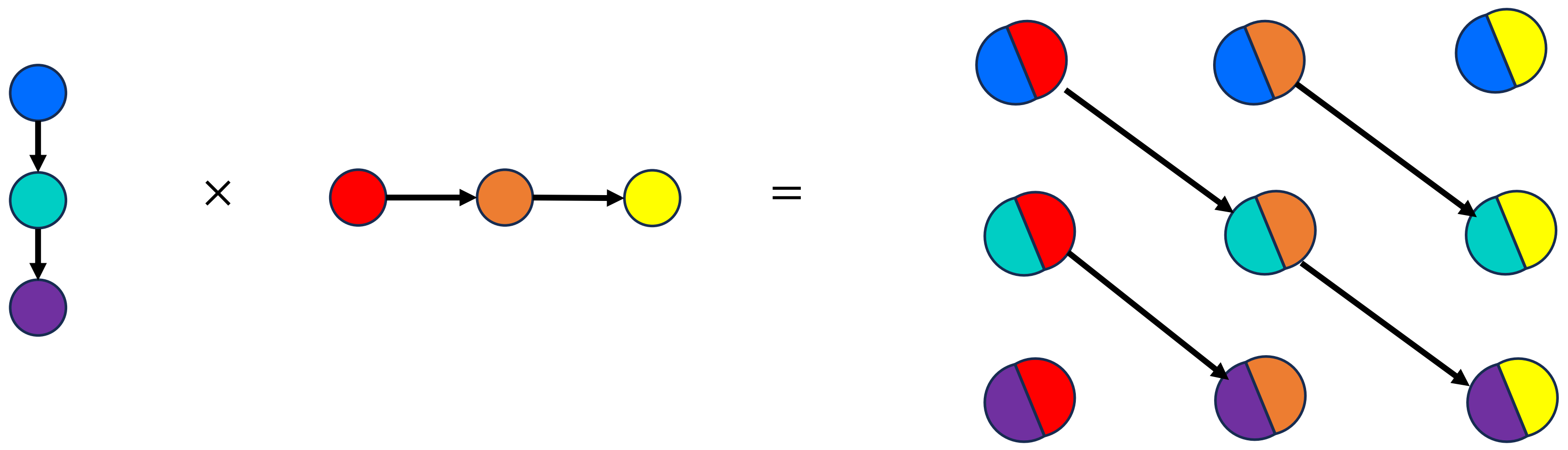
Graph Operations

The **Cartesian product** $G\square H$ of two directed graphs G and H has vertex set $V(G)\times V(H)$ and directed edges $\big((g_1,h_1),(g_2,h_2)\big)$ if
either $g_1 = g_2$ and $(h_1,h_2) \in E(H)$
or $h_1 = h_2$ and $(g_1,g_2) \in E(G)$.



Graph Operations

The **tensor product** $G \times H$ of two directed graphs G and H has vertex set $V(G) \times V(H)$ and directed edges $((g_1, h_1), (g_2, h_2))$ if $(g_1, g_2) \in E(G)$ and $(h_1, h_2) \in E(H)$.



One way of thinking about tensor products: **conjunction of edge conditions**

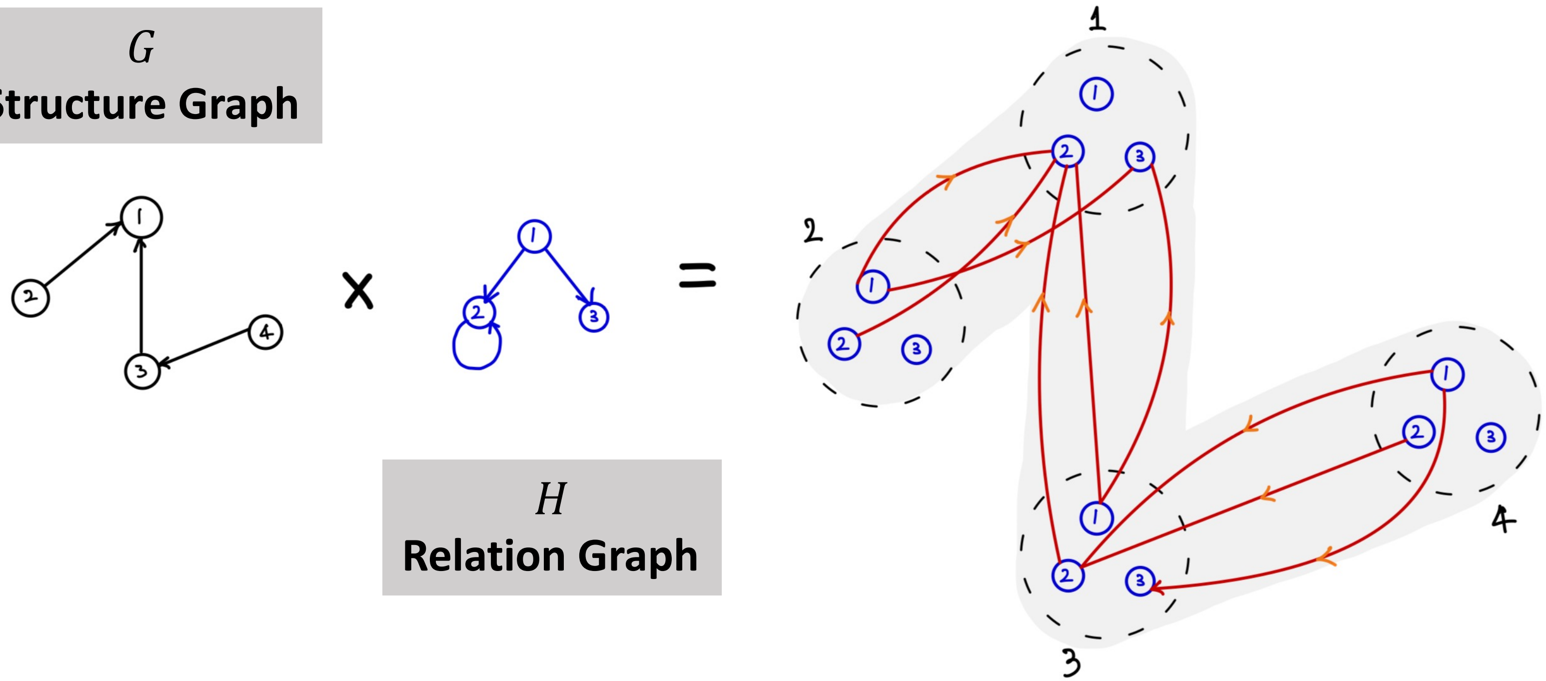
Graph Operations

The **tensor product** $G \times H$ of two directed graphs G and H has vertex set $V(G) \times V(H)$ and directed edges $((g_1, h_1), (g_2, h_2))$ if $(g_1, g_2) \in E(G)$ and $(h_1, h_2) \in E(H)$.

G
Structure Graph

H
Relation Graph

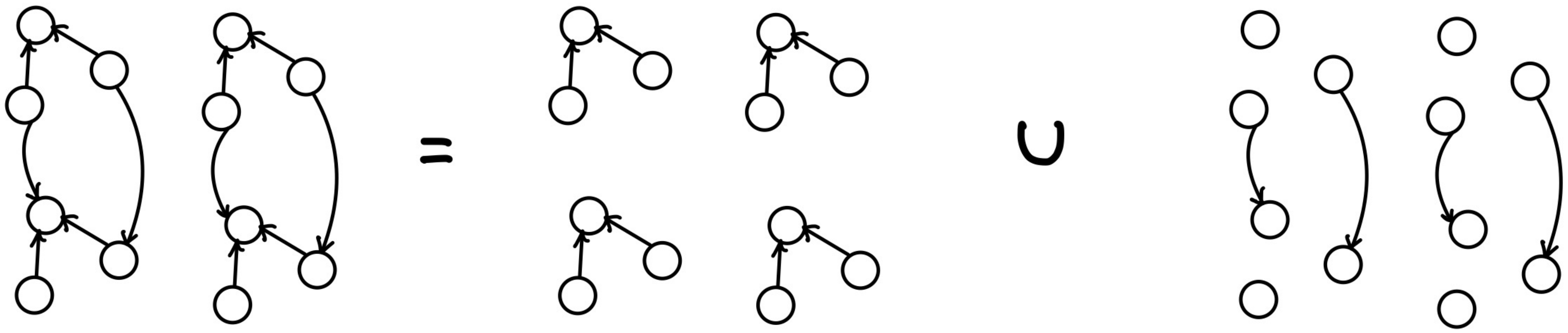
$G \times H$



Another way of thinking about tensor products: **embedding relations into structure**

Graph Operations

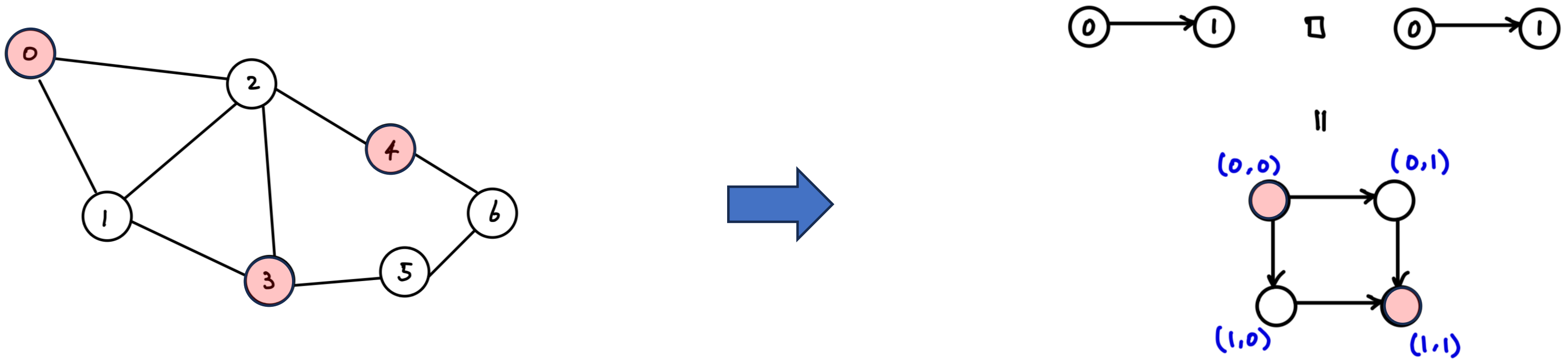
The **union** $G \cup H$ of two directed graphs G and H has
vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H)$.



Useful to decompose graph into **repetitive sub-structures**.

Lexicographically First Maximal Independent Set

Input: graph $G = (V, E)$, with vertex indices $V = \{0, 1, \dots, |V| - 1\}$, and a special vertex $s \in V$.
Output: whether s belongs to the lexicographically first maximal independent set of G



Standard Version

Factored Version

Same except that the input is a factored graph $G = f(G_1, \dots, G_k)$
with indices given for each input graph $V(G_i) = \{0, 1, \dots, |V(G_i)| - 1\}$.

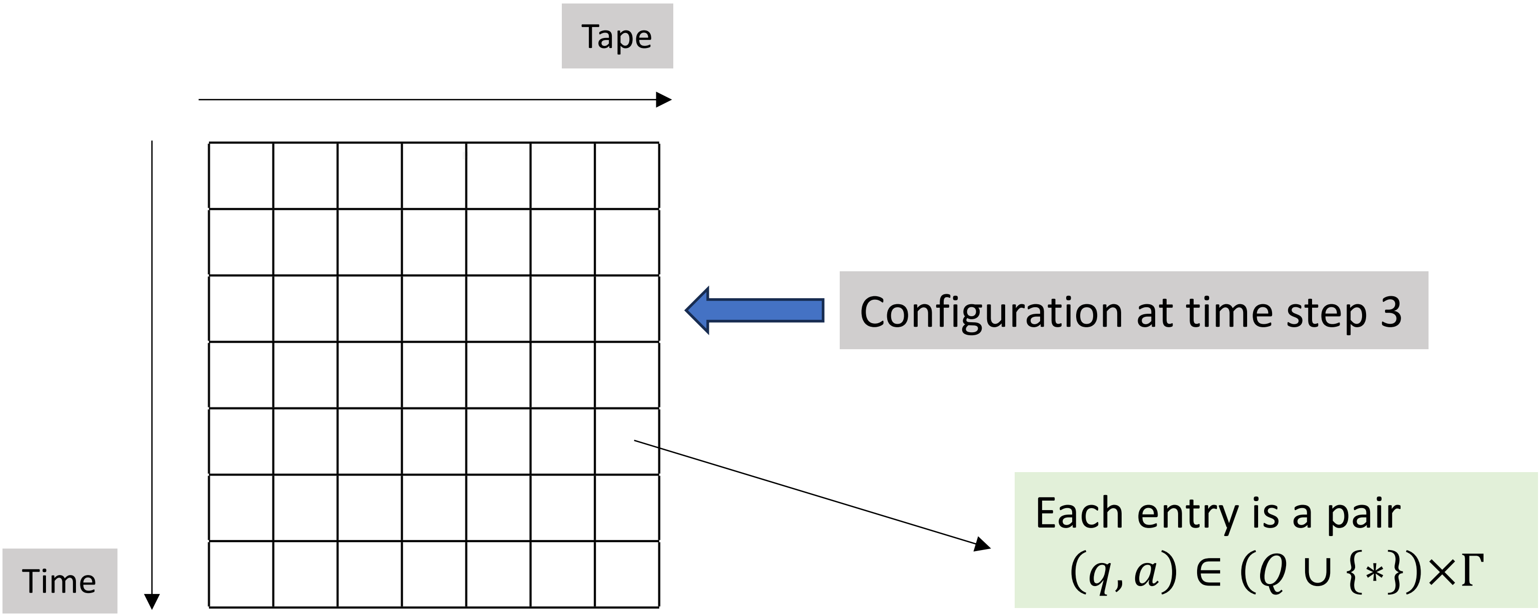
Theorem 1. Lexicographically First Maximal Independent Set on factored graphs is unconditionally **not in FPT**.
In particular, it is **XP-complete under FPT-reductions** and requires $\boldsymbol{n^{\Omega(\sqrt{k})}}$-time on factored graphs of complexity (n, k).

Proof Ideas - LFMIS

Observation. In the Lexicographically First Maximal Independent Set problem:

1) lexicographic first → sequential order
2) independent set → constraint

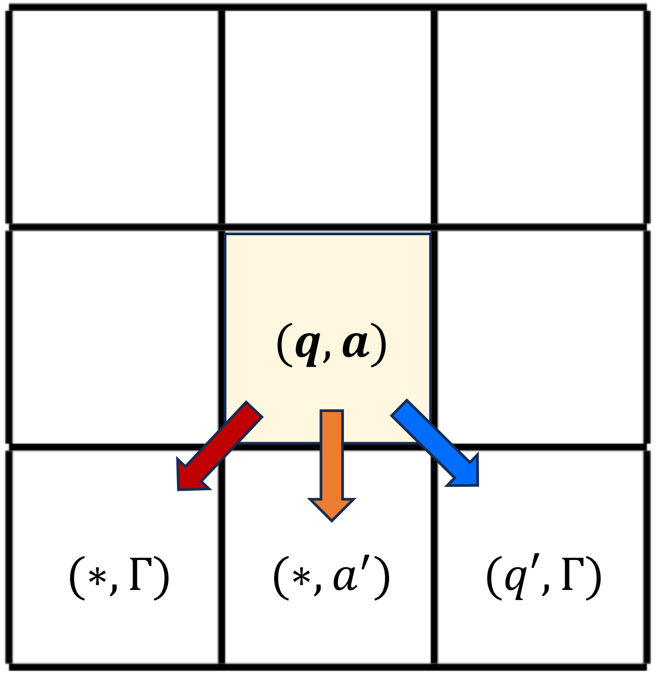
Key Idea. Given a TM M, construct a graph G whose LFMIS inductively recovers the computation of M



Computational History Matrix of a TM

Graph Construction

What can we say if the center entry is (q, a) and $\delta(q, a) = (q', a', R)$?

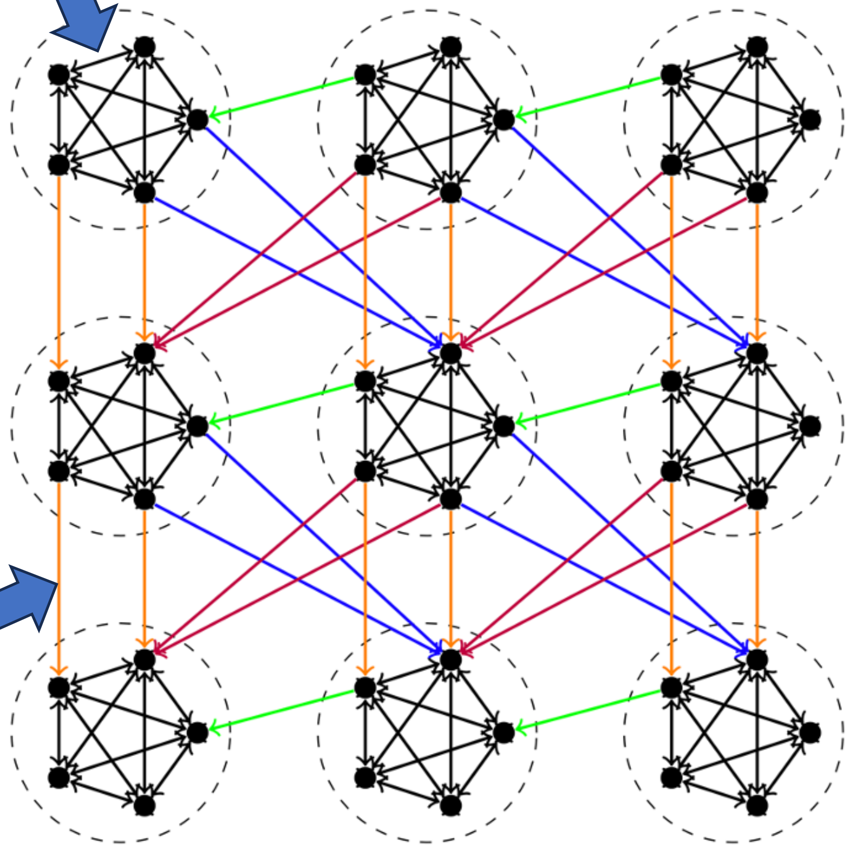


Computational History Matrix of a TM Running Time $T = 3$

Cluster: set of all possible choices for the corresponding entry

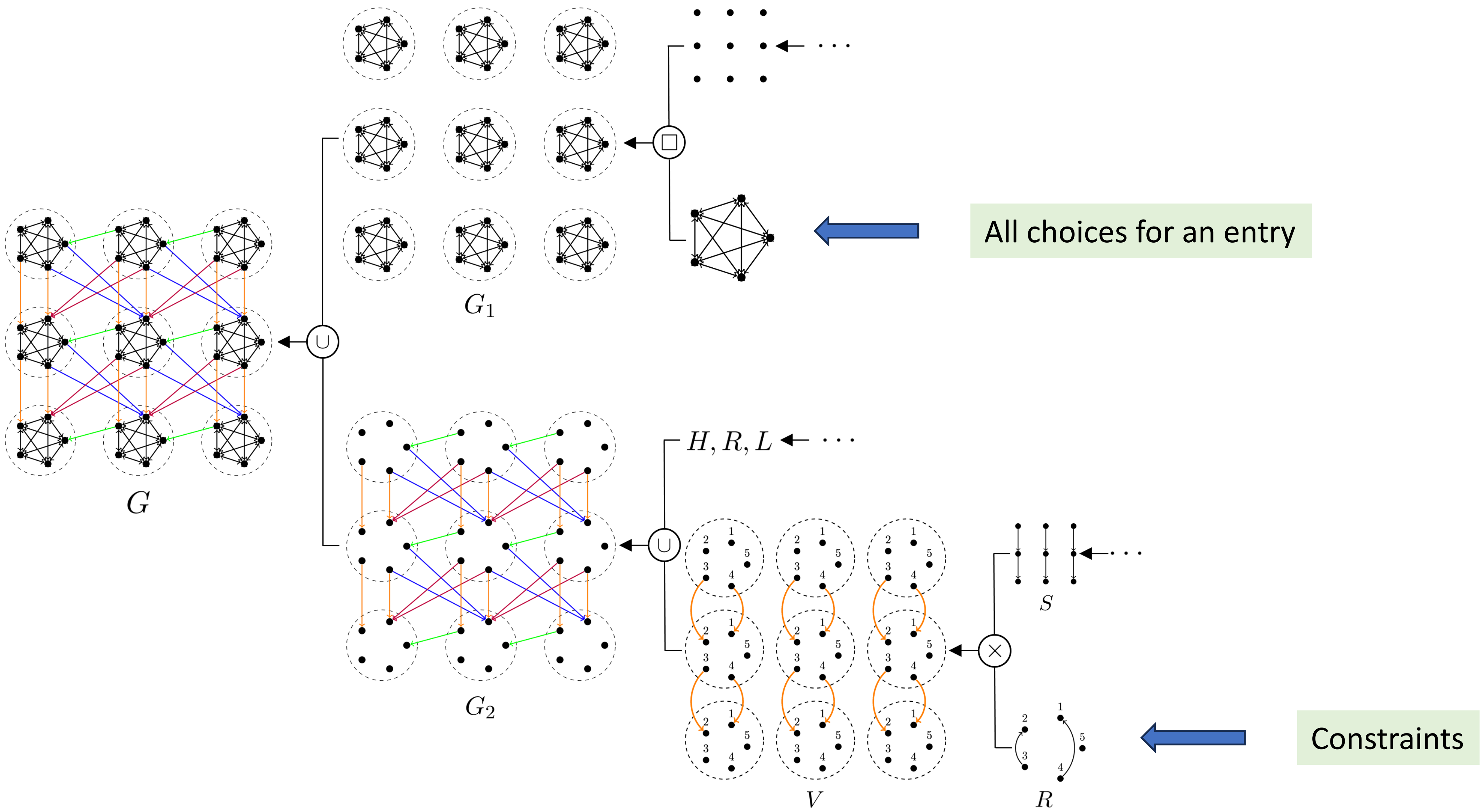
Intra-cluster edges

Inter-cluster edges



G

Factorization



Reachability

Theorem 3. Reachability on factored graphs is **XNL-complete under FPT-reductions.** Moreover, the following are equivalent:

1) $\mathbf{XNL} \subseteq \mathbf{FPT}$ (in particular, reachability)
2) $\mathbf{NL} \subseteq \mathbf{DTIME}(\boldsymbol{n}^{\boldsymbol{C}})$ for some absolute constant C.

A parameterized problem is in **XNL** if it can be solved in $\boldsymbol{O}(\boldsymbol{f}(\boldsymbol{k})\,\mathbf{log}\,\boldsymbol{n})$ **nondeterministic space**.

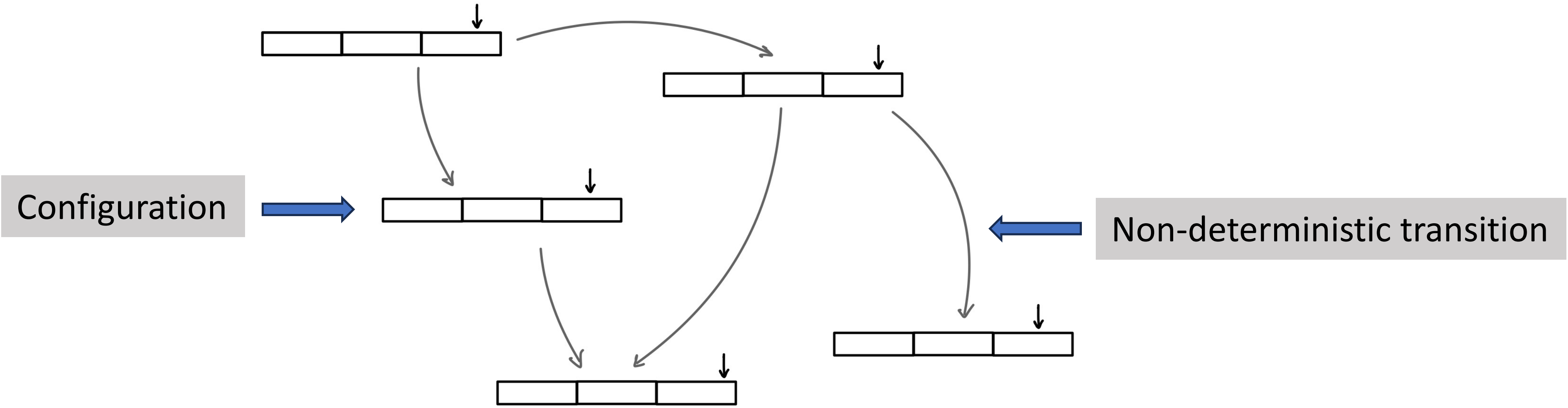
Remark. Recall that (standard) Reachability is complete for the class **NL.** [Sip96]

Proof Ideas - Reachability

Theorem 3. Reachability on factored graphs is **XNL-complete under FPT-reductions.**

NL-Complete Proof for Standard Reachability. [Sip96]

- For a language $L \in \mathbf{NL}$, there is a **non-deterministic TM $\boldsymbol{M}$** that decides L in $\boldsymbol{S \log n}$ space (S constant).
- Construct a **configuration graph** of M on input x.
- $x \in L \Leftrightarrow$ there is a path from initial configuration to accepting configurations.



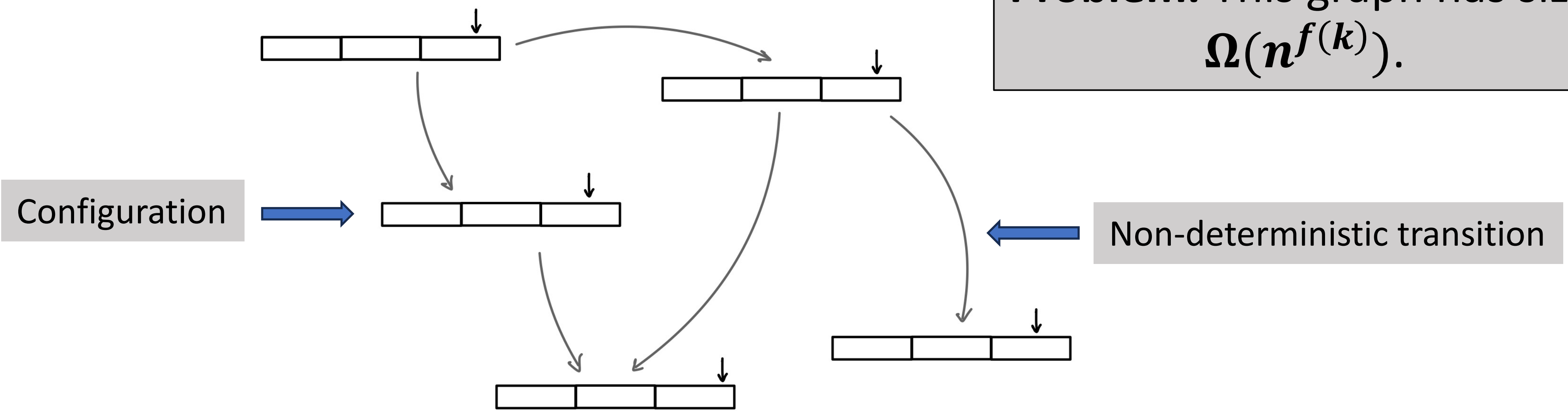
Example configuration graph with $S = 3$.

Proof Ideas - Reachability

Attempt: XNL-Complete Proof for Reachability on Factored Graphs.

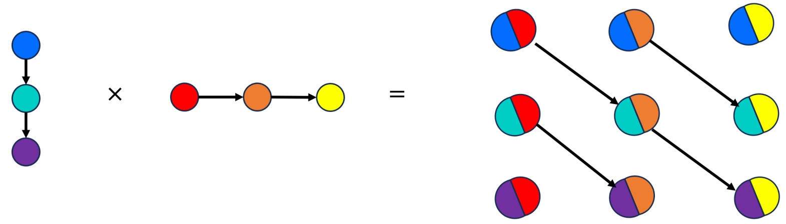
- For a language $L \in \mathbf{XNL}$, there is a **non-deterministic TM $\boldsymbol{M}$** that decides L in $\boldsymbol{f(k)}\ \mathbf{log}\ \boldsymbol{n}$ space.
- Construct a **configuration graph** of M on input x.
- $x \in L \Leftrightarrow$ there is a path from initial configuration to accepting configurations.

Problem. This graph has size $\boldsymbol{\Omega(n^{f(k)})}$.



Example configuration graph with $f(k) = 3$.

Recall tensor products

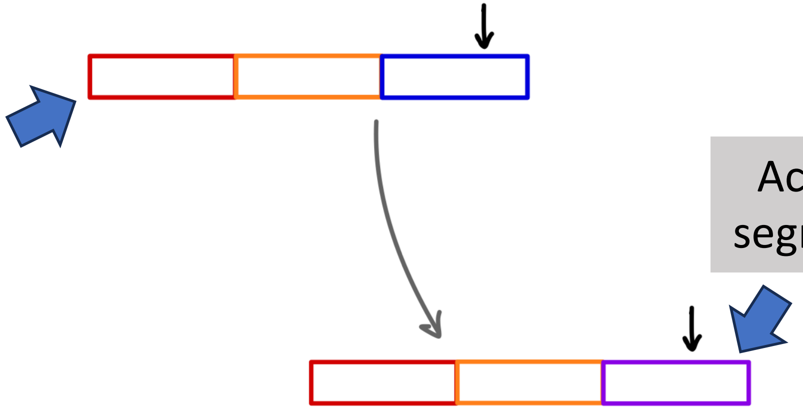


Proof Ideas - Reachability

Problem. # of configurations on an $f(k)\log n$-sized work tape grows exponentially in $f(k)$.

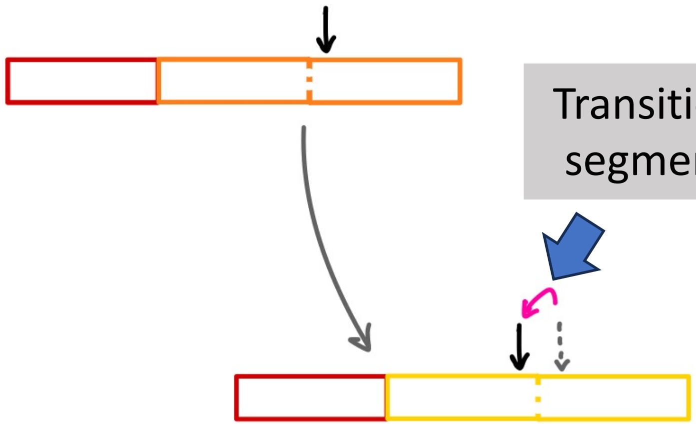
Solution. Decompose the work tape into $f(k)$ segments of size $\log n$, then use graph products to combine.

Inactive segment



Active segment

In-segment transition



Transition segment

Cross-segment transition

Upshot. The configuration graph has a factored graph rep. of complexity $(\boldsymbol{n^{O(1)}}, \boldsymbol{g(f(k))})$ for some function g.

Wrapup

- We studied the computational complexity of various problems on factored graphs.
  - Lexicographically First Maximal Independent Set (not FPT)
  - Clique Counting (FPT)
  - Reachability (open)
- Future Work
  - Do similar results hold for other complete problems?
  - More *fine-grained* analysis: there is still room for improvement from the naïve $n^{O(k)}$-time algorithm.
  - Can we define natural factored instances on other interesting objects?

Thank you!

arXiv:2407.19102