Query answering is the most fundamental problem in DB

Database D

R	A	B
	u	x
	v	x

S	B	C
	x	y
	x	z

Query Q

```
SELECT R.A, S.C
FROM R, S
WHERE R.B = S.B
```

Result $Q(D)$

Q(D)	A	C
	u	y
	u	z
	v	y
	v	z

Three crucial problems for query answering

R	A	B
	u	x
	v	x

S	B	C
	x	y
	x	z

```
SELECT R.A, S.C
FROM R, S WHERE R.B = S.B
```

Q(D)	A	C
	u	y
	u	z
	v	y
	v	z

1. Enumeration

$$(u, y),\ (u, z),\ (v, y),\ (v, z)$$

2. Uniform generation

$$(u, y) : \frac{1}{4},\ (u, z) : \frac{1}{4},\ (v, y) : \frac{1}{4},\ (v, z) : \frac{1}{4}$$

3. Counting

$$|Q(D)| \ = \ 4$$

In this paper, we study log-space complexity classes

We consider the class $\textsc{RelationNL}$ and
show that it has good algorithmic properties in terms of:

- Enumeration.
- Approximate counting.
- Approximate uniform generation.

We consider the subclass $\textsc{RelationUL}$ and
show that it has better algorithmic properties in terms of:

- Constant delay enumeration (polynomial time preprocessing).
- Exact counting.
- Exact uniform generation.

We show **applications** of these results in
information extraction, graph databases, and among others.

Efficient log-space classes for enumeration, counting, and uniform generation

Marcelo Arenas
Luis Alberto Croqueville
Cristian Riveros

PUC & IMFD Chile

Rajesh Jayaram

Carnegie Mellon University

Outline

The class RelationNL

FPRAS for RelationNL

Conclusions

Outline

The class RelationNL

FPRAS for RelationNL

Conclusions

Relations as instances of problems

Let Σ be a finite alphabet.

Definitions

A **problem** is a relation $R \subseteq \Sigma^* \times \Sigma^*$.

- If $(x, y) \in R$, then x is an **input** and y is a **solution**.

We restrict to ***p*-relations** R where for every $(x, y) \in \Sigma^* \times \Sigma^*$:

1. if $(x, y) \in R$, then y is of polynomial size with respect to x.
2. $(x, y) \in R$ can be verified in polynomial time.

Three main problems associated to a p-relation

Given an input x we denote by $W_R(x)$ the set of solutions or **witnesses**:

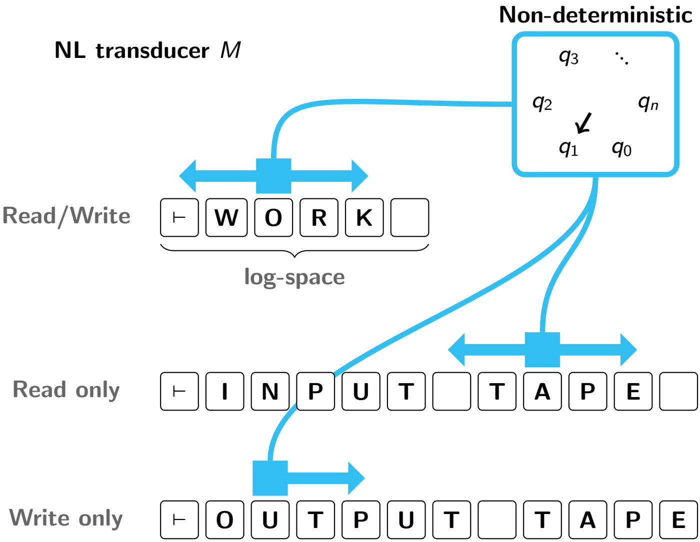
$$W_R(x) \;=\; \{y \in \Sigma^* \mid (x,y) \in R\}$$

Problem:	$\textsc{Enum}(R)$
Input:	A word $x \in \Sigma^*$
Output:	Enumerate all $y \in W_R(x)$ without repetitions

Problem:	$\textsc{Count}(R)$
Input:	A word $x \in \Sigma^*$
Output:	The size $\lvert W_R(x)\rvert$

Problem:	$\textsc{Gen}(R)$
Input:	A word $x \in \Sigma^*$
Output:	Generate uniformly at random a word in $W_R(x)$.

A log-space complexity class: RELATIONNL



A log-space complexity class: RelationNL

Given an NL-transducer M and an input x, we define its **set of outputs**:

$$M(x) = \{y \in \Sigma^* \mid \text{there exists a run of } M \text{ on } x \text{ that halts in an accepting state with } y \text{ in the output}\}$$

Definition of RelationNL

A relation R is in RelationNL iff there exists an NL-transducer M s.t.:

$$R = \{(x, y) \in \Sigma^* \times \Sigma^* \mid y \in M(x)\}$$

Main results for $\textsc{RelationNL}$

Theorem

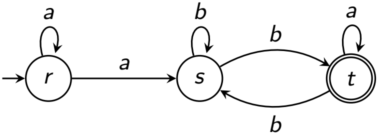
If $R \in \textsc{RelationNL}$ then:

1. $\textsc{Enum}(R)$ can be solved with **polinomial delay**.
2. $\textsc{Count}(R)$ admits an **FPRAS**
   (fully polynomial-time randomized approximation scheme).
3. $\textsc{Gen}(R)$ admits a **polynomial time "Las Vegas" uniform generator**.

We introduce a subclass $\textsc{RelationUL}$ that has good properties w.r.t. constant delay enumeration, exact counting, and uniform gen.

Outline

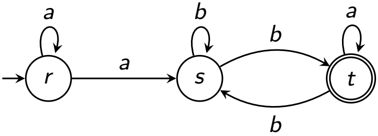
A complete problem for RelationNL



$$\underbrace{000000000\cdots00}_{n}$$

How many words of length n are accepted
by a non-deterministic finite state automaton (NFA)?

A complete problem for RelationNL



$$\underbrace{000000000\cdots00}_{n}$$

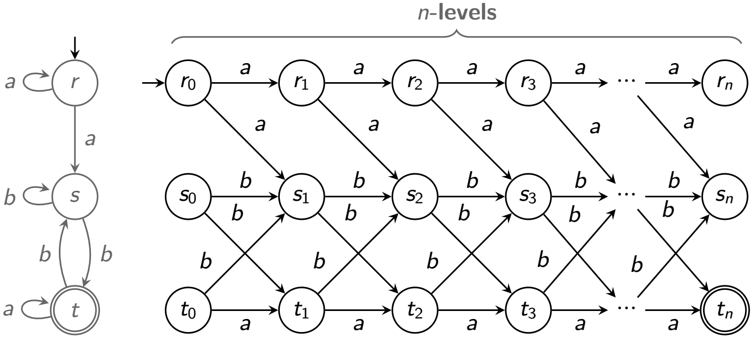
Problem: #NFA
Input: A NFA $\mathcal{A} = (Q, \Sigma, \Delta, q_0, F)$ and 0^n.
Output: $|\{w \mid w \in \mathcal{L}(\mathcal{A}) \text{ and } |w| = n\}|$.

Proposition

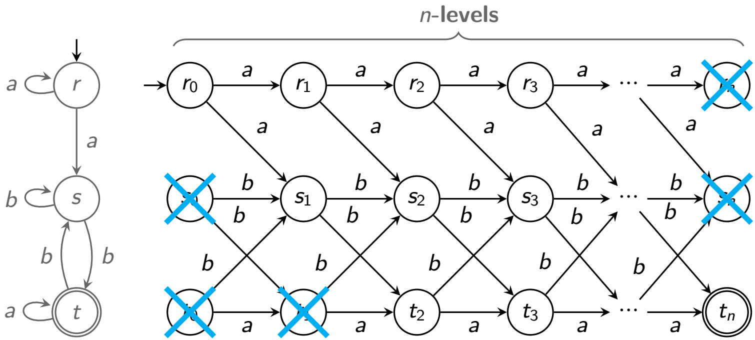
For every $R \in$ RelationNL,
there exists a **parsimonious reduction** from Count(R) to #NFA .

If we find an FPRAS for #NFA,
we have an FPRAS for every $R \in$ RelationNL.

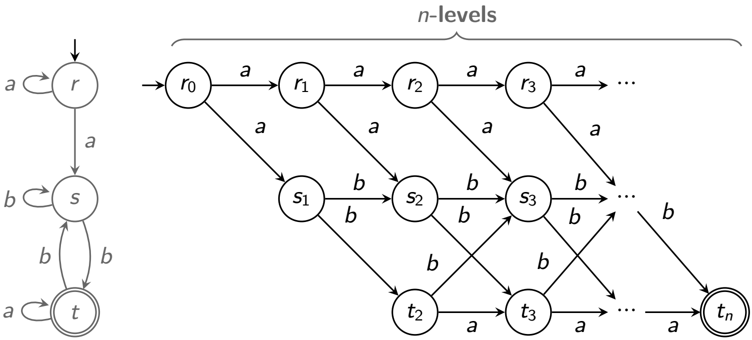
Main ideas of FPRAS: Unfold the NFA until level n



Main ideas of FPRAS: Unfold the NFA until level n

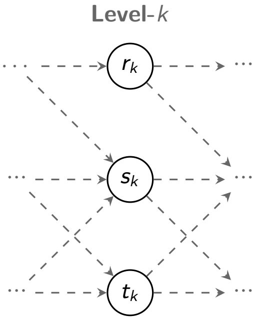


Main ideas of FPRAS: Unfold the NFA until level n



The problem is reduced to approximate
the number of **label-paths** from the initial state to the final states.

Main ideas of FPRAS: languages at level k



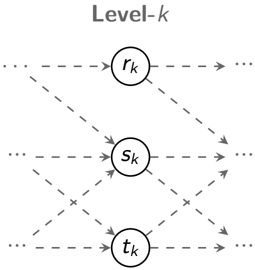
Let Q_k be the set of states at level k. For each $P \subseteq Q_k$:

$$\mathcal{L}(P) \;=\; \text{all words that reach any state in } P \text{ from the initial state.}$$

We want to **approximate** the size $|\mathcal{L}(P)|$ for any $P \subseteq Q_k$.

…we want to approximate $|\mathcal{L}(F)|$ where $F \subseteq Q_n$.

Main ideas of FPRAS: a sketch for each level



For every $q \in Q_k$

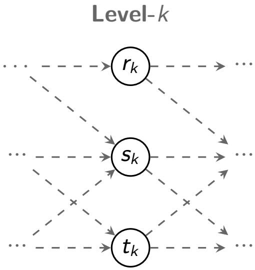
- $N(q)$: $N(q) \sim |\mathcal{L}(q)|$
  an $(1 \pm \epsilon)$-approximation.
- $S(q)$: $S(q) \subseteq \mathcal{L}(q)$
  uniform sample of poly-size.

For every $P \subseteq Q_k$ and for any total order $<$ of P:

$$
\begin{aligned}
|\mathcal{L}(P)| &= \sum_{q \in P} |\mathcal{L}(q)| \cdot \frac{|\mathcal{L}(q) \setminus \mathcal{L}(\{p \in P \mid p < q\})|}{|\mathcal{L}(q)|} \\
&\sim \sum_{q \in P} N(q) \cdot \frac{|S(q) \setminus \mathcal{L}(\{p \in P \mid p < q\})|}{|S(q)|}
\end{aligned}
$$

This approximation can be computed in poly-time from $N(q)$ and $S(q)$

Main ideas of FPRAS: a sketch for each level



For every $q \in Q_k$

- $N(q)$: $N(q) \sim |\mathcal{L}(q)|$ an $(1 \pm \epsilon)$-approximation.
- $S(q)$: $S(q) \subseteq \mathcal{L}(q)$ uniform sample of poly-size.

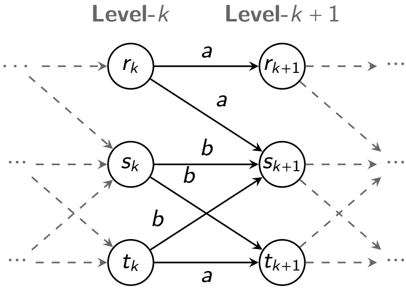
For every $P \subseteq Q_k$ and for any total order $<$ of P:

$$|\mathcal{L}(P)| \;\sim\; N(P) = \sum_{q \in P} N(q) \cdot \frac{|S(q) \setminus \mathcal{L}(\{p \in P \mid p < q\})|}{|S(q)|}$$

For every $P \subseteq Q_k$ and $q \in Q_k - P$ (by Hoeffding's inequality):

$$\left| \frac{|S(q) \setminus \mathcal{L}(P)|}{|S(q)|} - \frac{|\mathcal{L}(q) \setminus \mathcal{L}(P)|}{|\mathcal{L}(q)|} \right| \leq \epsilon \quad \text{with (exponentially) high prob.}$$

Main ideas of FPRAS: update the sketch to the next level



For every $q \in Q_k$

- $N(q)$: $N(q) \sim |\mathcal{L}(q)|$
  an $(1 \pm \epsilon)$-approximation.
- $S(q)$: $S(q) \subseteq \mathcal{L}(q)$
  uniform sample of poly-size.

For every $q \in Q_{k+1}$ let $P_c = \{p \in Q_k \mid (p, c, q) \in \Delta\}$ for $c \in \{a, b\}$:

$$N(q) = N(P_a) + N(P_b)$$

To generate $S(q)$ we use a technique from **Jerrum, Valiant, and Vazirani** for generating a uniform sample by using the $(1 \pm \epsilon)$-approximations:

$$\{N(P)\}_{P \subseteq Q_{k'}} \quad \text{for every } k' \leq k.$$

Outline

The class RelationNL

FPRAS for RelationNL

Conclusions

Conclusions and future work

1. We provide complexity classes that has good properties in terms of **enumeration**, **counting**, and **uniform generation**.
2. RelationNL is the first complexity class with a simple definition based on TM and where each problem **admits an FPRAS**.

Future work:

1. Find an FPRAS for #NFA that can be used in practice with better polynomial factors and constants.
2. Find an FPRAS for #CFG.

Thanks!