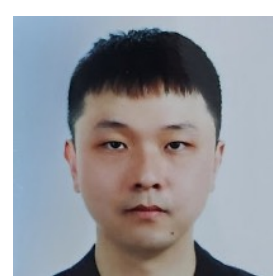
Università
Ca'Foscari
Venezia

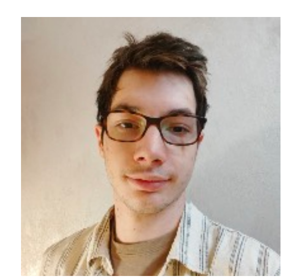
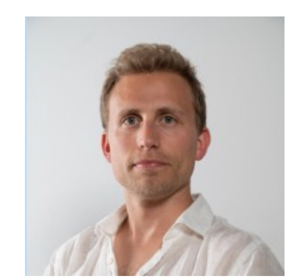
Random Wheeler Automata

CPM 2024

Fukuoka, Japan, 25 June 2024





Ruben Becker Davide Cenzato Sung-Hwan Kim Bojana Kodric Riccardo Maso Nicola Prezza

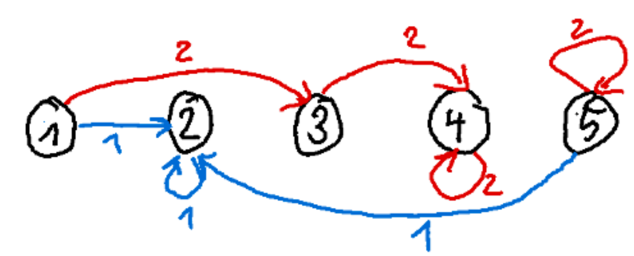
Setting and Contribution

Setting:
- Wheeler Automata constitute class of Automata, s.t. we can
  1. store them concisely ($O(1)$ space per edge)
  2. do pattern matching on their accepted language

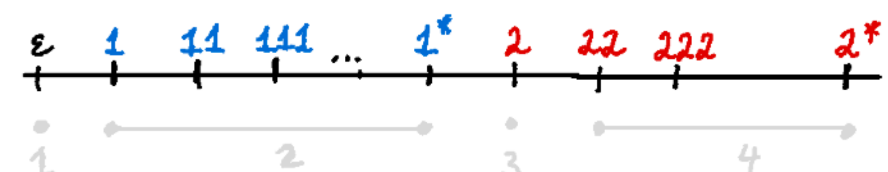
→ generalizing FM index from strings to (some) graphs

- Here: Focus on DFAs, then:

$\boxed{\text{Wheeler DFAs} = \text{DFAs whose states can be ordered consistently with co-lex-order of accepted strings.}}$



1	ε
2	1^k for $k \geq 1$
3	2
4	2^k for $k \geq 2$



- Plenty of algorithmic / language-theoretic research, but few available data sets.

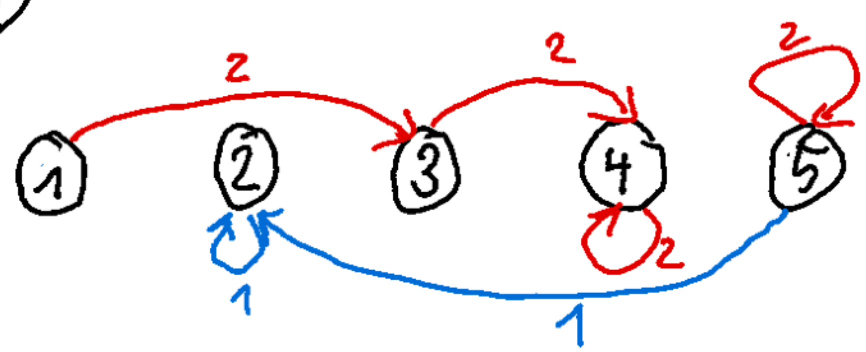
Contribution: Algorithm to generate uniform Wheeler DFA from $\mathcal{D}_{n,m,\sigma}$ in constant space & expected linear time

all Wheeler DFAs with n nodes, m edges, alphabet size σ

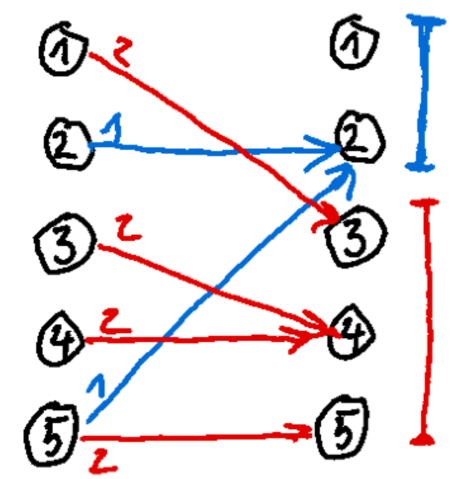
Wheeler DFAs

Def.: Wheeler DFA $D = (\overset{=[n]}{Q}, \delta, \Sigma)$ is s.t. $<$ satisfies

D



"Bipartite Representation"



(i) ranges

(ii) edges of same color do not cross

$O =$

	1	2
1		
2		
3		
4		
5		

$I =$

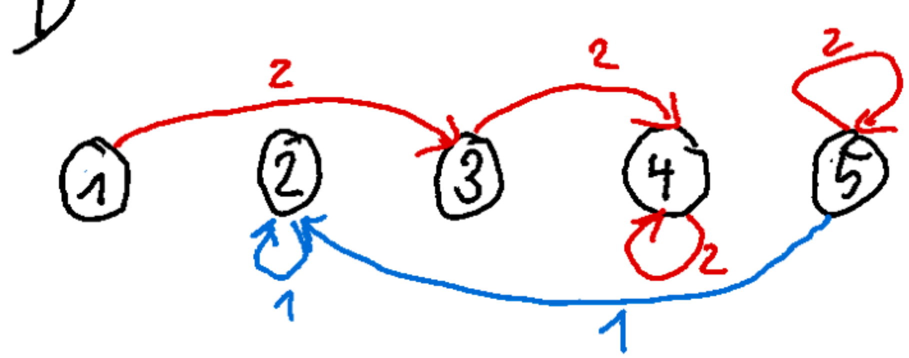
Wheeler DFAs

Def.: Wheeler DFA $D = (\overset{=[n]}{Q}, \delta, \Sigma)$ is s.t. $<$ satisfies

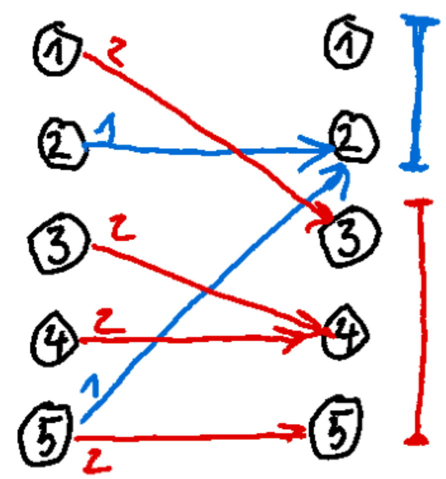
(i) $u' = \delta_a(u),\ v' = \delta_{a'}(v),\ a \prec a' \implies u' < v'$

(ii) $u' = \delta_a(u) \neq v' = \delta_a(v)\quad u < v \implies u' < v'$

D



"Bipartite Representation"



(i) ranges $\mathrm{I}\ \mathrm{I}$

(ii) edges of same color do not cross

$O =$

	1	2
1	0	
2	1	
3	0	
4	0	
5	1	

$I =$

2					
1	0				

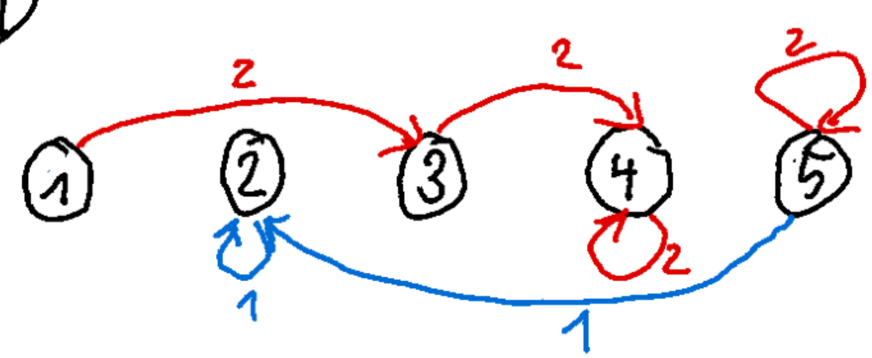
Wheeler DFAs

Def.: Wheeler DFA $D = (\overset{=[n]}{Q}, \delta, \Sigma)$ is s.t. $<$ satisfies

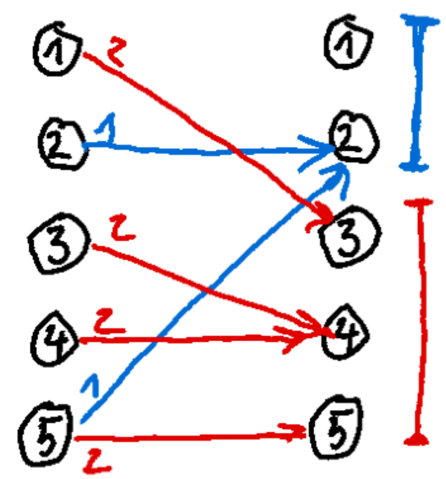
(i) $u' = \delta_a(u),\ v' = \delta_{a'}(v),\ a \prec a' \implies u' < v'$

(ii) $u' = \delta_a(u) \neq v' = \delta_a(v)\ \ u < v \implies u' < v'$

D



"Bipartite Representation"



(i) ranges

(ii) edges of same color do not cross

$O =$

	1	2
1	0	1
2	1	
3	0	
4	0	
5	1	

$I =$

2		3			
1	0	1			

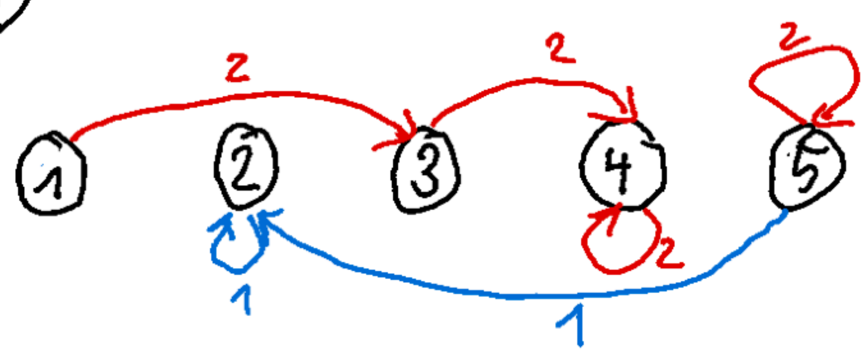
Wheeler DFAs

Def.: Wheeler DFA $D = (\overset{=[n]}{Q}, \delta, \Sigma)$ is s.t. $<$ satisfies

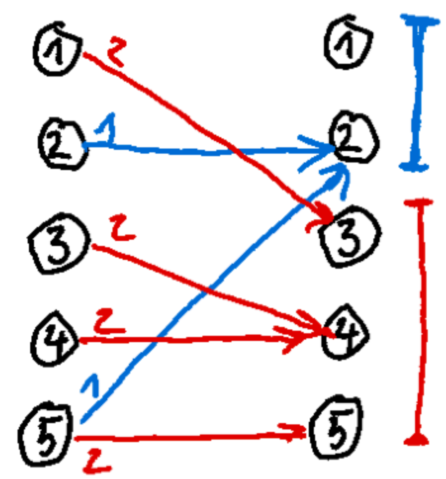
ⓘ $u' = \delta_a(u)$, $v' = \delta_{a'}(v)$, $a \prec a' \implies u' < v'$

ⓘⓘ $u' = \delta_a(u) \neq v' = \delta_a(v)$ $u < v \implies u' < v'$

D



"Bipartite Representation"



(i) ranges

(ii) edges of same color do not cross

$O =$

	1	2
1	0	1
2	1	0
3	0	1
4	0	1
5	1	

$I =$

2		3	4		
1	0	1	1	0	

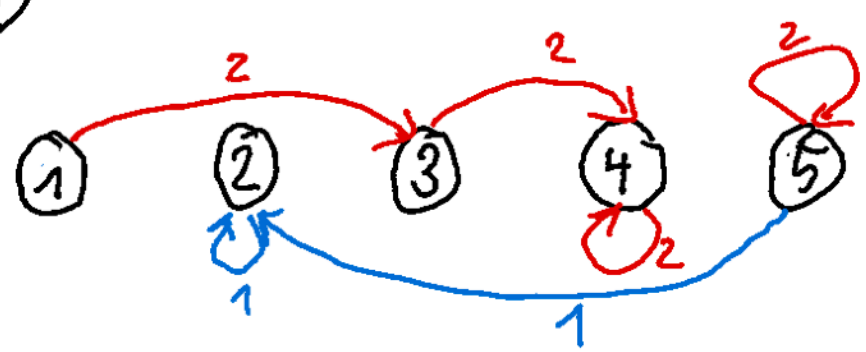
Wheeler DFAs

Def.: Wheeler DFA $D = (\overset{=[n]}{Q}, \delta, \Sigma)$ is s.t. $\prec$ satisfies

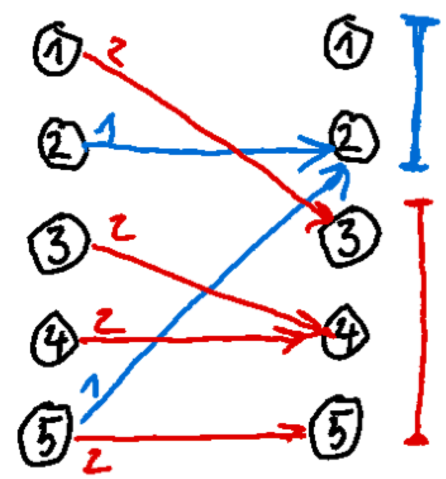
(i) $u' = \delta_a(u),\ v' = \delta_{a'}(v),\ a \prec a' \Rightarrow u' \prec v'$

(ii) $u' = \delta_a(u) \neq v' = \delta_a(v)\ \ u \prec v \Rightarrow u' \prec v'$

D



"Bipartite Representation"



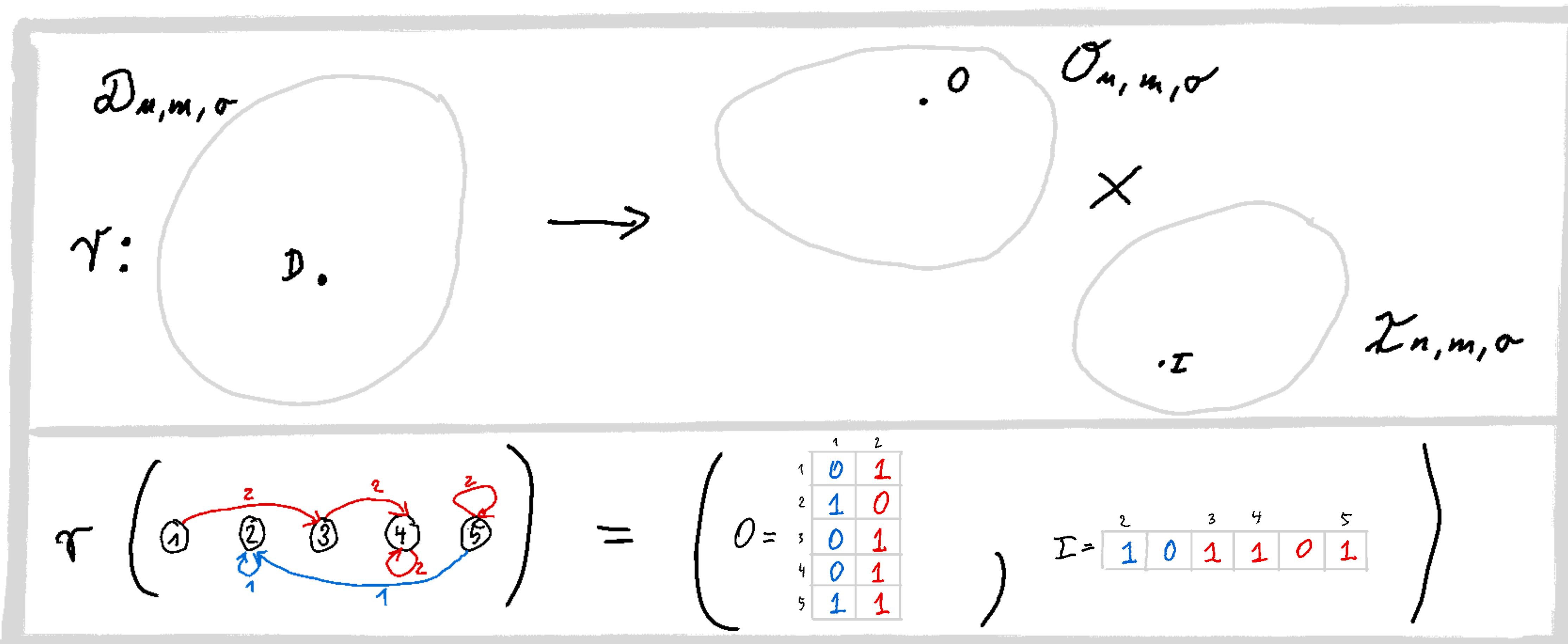
(i) ranges

(ii) edges of same color do not cross

$$O = \begin{array}{c|cc|} & 1 & 2 \\ \hline 1 & 0 & 1 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \\ 4 & 0 & 1 \\ 5 & 1 & 1 \end{array}$$

$$I = \begin{array}{|c|c|c|c|c|c|} 2 & & 3 & 4 & & 5 \\ \hline 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

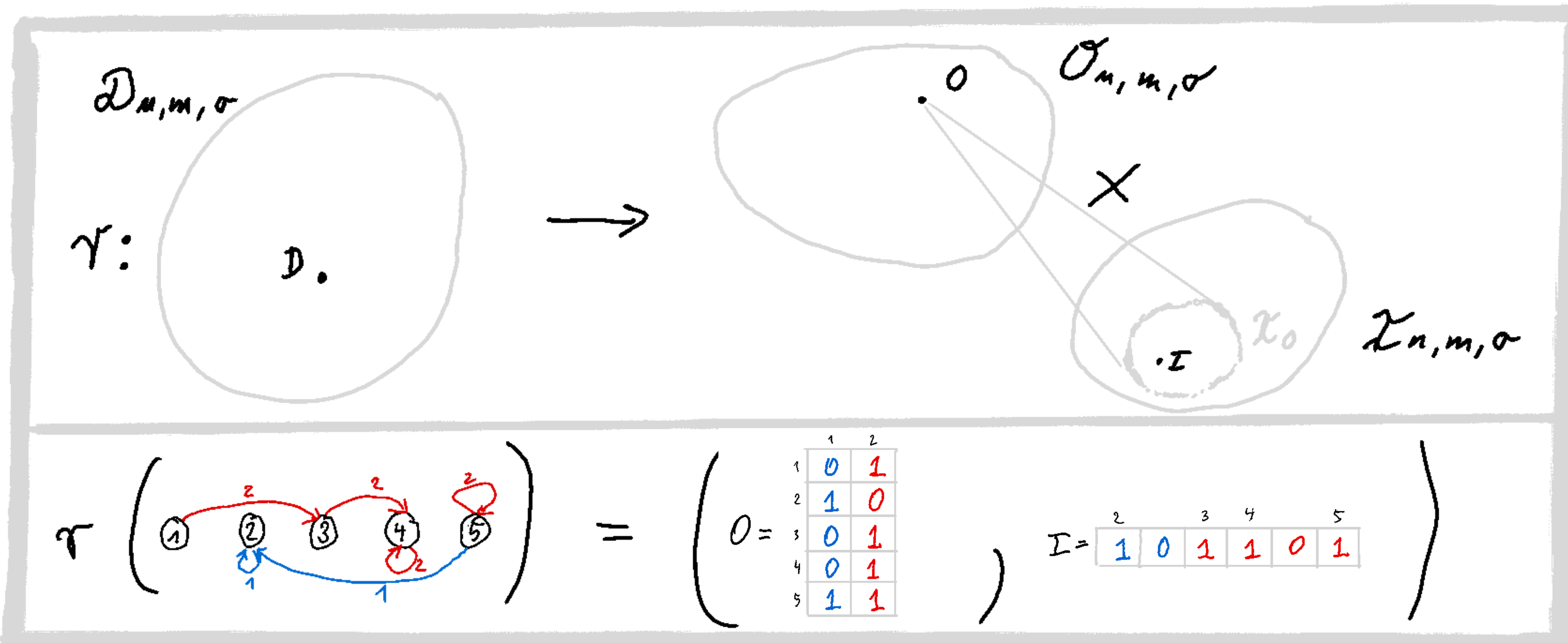
Sampling Wheeler DFAs



Idea: Sample from $\mathcal{D}_{n,m,\sigma}$ by sampling from $\mathcal{O}_{n,m,\sigma} \times \mathcal{I}_{n,m,\sigma}$

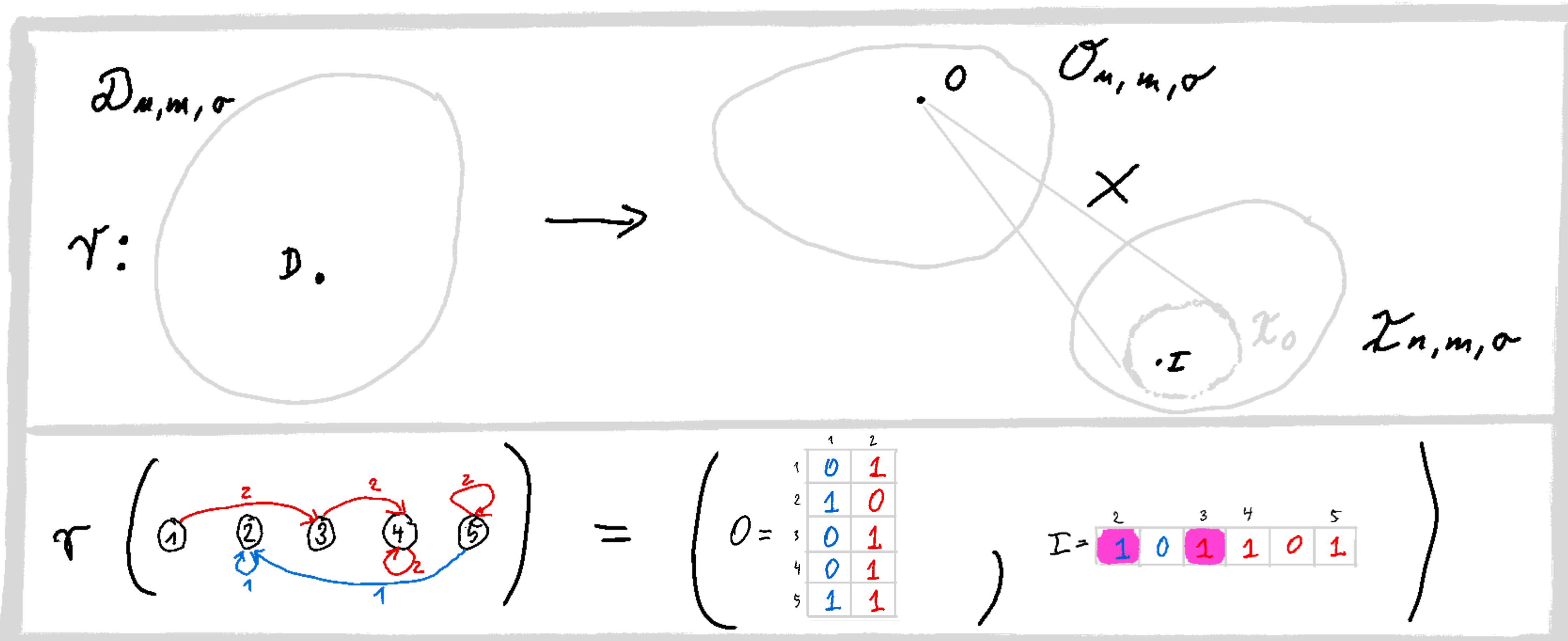
- τ is injective ✓ [a pair corresponds to at most one DFA]
- τ is not surjective ✗ [not every pair corresponds to a DFA]

Sampling Wheeler DFAs ctd.



$$\mathcal{I}_0 := \{ I \in \mathcal{I}_{n,m,\sigma} : I_{1+\sum_{k=1}^{j-1} \|O_k\|_1} = 1 \quad \forall j \in [\sigma] \}$$

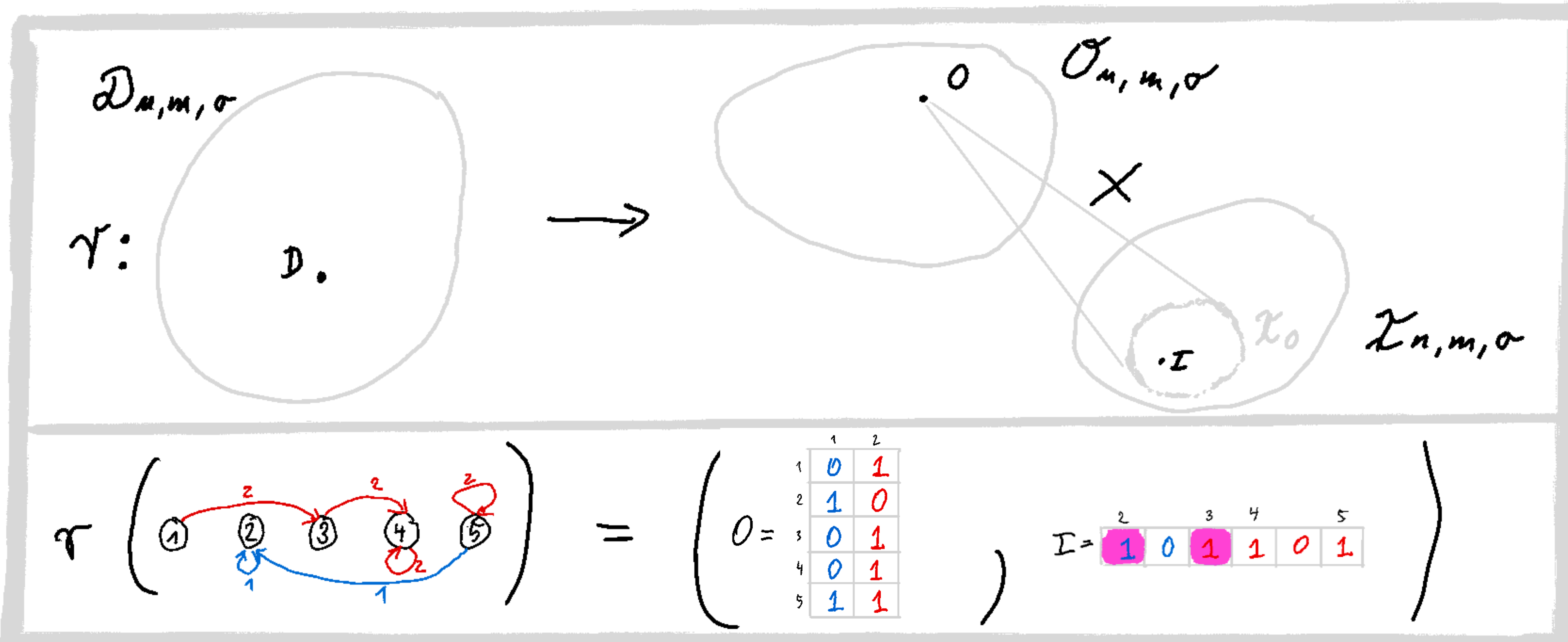
Sampling Wheeler DFAs ctd.



$$\mathcal{I}_0 := \left\{ I \in \mathcal{I}_{n,m,\sigma} : \ I_{1+\sum_{k=1}^{j-1} \|O_k\|_1} = 1 \quad \forall j \in [\sigma] \right\}$$

$$\mathcal{R}_{n,m,\sigma} := \left\{ (O, I) : \ O \in \mathcal{O}_{n,m,\sigma} \text{ and } I \in \mathcal{I}_0 \right\}$$

Sampling Wheeler DFAs ctd.



$$\mathcal{I}_0 := \left\{ I \in \mathcal{I}_{n,m,\sigma} : I_{1+\sum_{k=1}^{j-1} \|O_k\|_1} = 1 \quad \forall j \in [\sigma] \right\}$$

$$R_{n,m,\sigma} := \{ (O, I) : O \in \mathcal{O}_{n,m,\sigma} \text{ and } I \in \mathcal{I}_0 \}$$

$r : \mathcal{D}_{n,m,\sigma} \longrightarrow R_{n,m,\sigma}$ is bijective !

$\longrightarrow$ sample from $\mathcal{D}_{n,m,\sigma}$ by sampling from $R_{n,m,\sigma}$

$\hookrightarrow$ uniformity follows from $|\mathcal{I}_0| = |\mathcal{I}_0'|$ $\forall O, O' \in \mathcal{O}_{n,m,\sigma}$

Sampling from $R_{n,m,\sigma}$

$$R_{n,m,\sigma} := \{(O,I) : O \in \mathcal{O}_{n,m,\sigma} \text{ and } I \in \mathcal{I}_O\},$$

$$\text{where } \mathcal{I}_O := \{ I \in \mathcal{I}_{n,m,\sigma} : I_{1+\sum_{k=1}^{j-1} \|O_k\|_1} = 1 \quad \forall j \in [\sigma] \}$$

sample_O

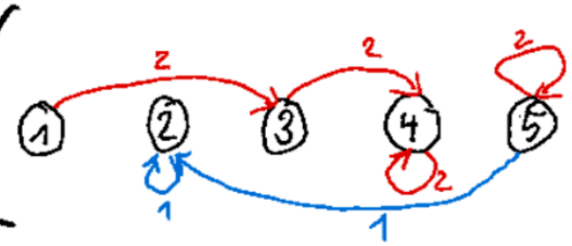
repeat
  $O := \texttt{reshape}_{n,\sigma}(\texttt{shuffle}(1^m 0^{n\sigma - m}))$
until $\|O_j\| \geq 1$ *for all* $j \in [\sigma]$
return O

sample_I(O)

$\text{mask} := 1\#^{\|O_1\|-1} 1\#^{\|O_2\|-1} \ldots 1\#^{\|O_\sigma\|-1}$
$I := \texttt{fill}(\text{mask}, \texttt{shuffle}(1^{n-\sigma-1} 0^{m-n+1}))$
return I

build-D(O,I)

$\delta := \emptyset,\ i := 1,\ v := 1$
for $j = 1, 2, \ldots, \sigma$ **do**
  for $u = 1, 2, \ldots, n$ **do**
    if $O_{u,j} = 1$ **then**
      if $I_i = 1$ **then** $v := v + 1$
      $\delta := \delta \cup \{((u,j), v)\}$
      $i := i + 1$

$r\Big($  $\Big) = \left(O = \begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 0 & 1 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \\ 4 & 0 & 1 \\ 5 & 1 & 1 \end{array},\quad I = \begin{array}{cccccc} 2 & & 3 & 4 & & 5 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{array} \right)$

Sampling from $\mathcal{R}_{n,m,\sigma}$

$$\mathcal{R}_{n,m,\sigma} := \{ (O, I) : O \in \mathcal{O}_{n,m,\sigma} \text{ and } I \in \mathcal{I}_O \},$$

$$\text{where } \mathcal{I}_O := \{ I \in \mathcal{I}_{n,m,\sigma} : I_{1+\sum_{k=1}^{j-1} \|O_k\|_1} = 1 \quad \forall j \in [\sigma] \}$$

sample_O

repeat
$\quad O := \texttt{reshape}_{n,\sigma}(\texttt{shuffle}(1^m 0^{n\sigma - m}))$
until $\|O_j\| \geq 1$ *for all* $j \in [\sigma]$
return O

sample_I (O)

mask $:= 1\#^{\|O_1\|-1} 1\#^{\|O_2\|-1} \ldots 1\#^{\|O_\sigma\|-1}$
$I := \texttt{fill}(\text{mask}, \texttt{shuffle}(1^{n-\sigma-1} 0^{m-n+1}))$
return I

build-D (O, I)

$\delta := \emptyset,\ i := 1,\ v := 1$
for $j = 1, 2, \ldots, \sigma$ **do**
$\quad$ **for** $u = 1, 2, \ldots, n$ **do**
$\quad\quad$ **if** $O_{u,j} = 1$ **then**
$\quad\quad\quad$ **if** $I_i = 1$ **then** $v := v + 1$
$\quad\quad\quad$ $\delta := \delta \cup \{((u, j), v)\}$
$\quad\quad\quad$ $i := i + 1$

of rejections in sample_O:

$m = \Omega(\sigma \ln \sigma)$

- $\Pr[\text{column } j \text{ empty}] = \prod_{i=1}^{m} \left(1 - \frac{n}{n\sigma - (i-1)}\right) \leq \left(1 - \frac{1}{\sigma}\right)^m = O(1/\sigma)$
- $\Pr[\exists \text{ empty column}] = O(1)$ [union bound]
- $\mathbb{E}[\#\text{iterations}]$ is constant and #iterations is $O(\log m)$ w. pr. $1 - m^{-c}$

Constant Space Implementation

- build_D accesses O column- and I bit-wise
- can generate non-zero entries of O and I on the fly in constant space using sequential_shuffler(N, k) of [Shekelyan & Cormode]
  ↳ • generate k uniform integers from $[N]$ in ascending order within constant space
  • call with $N = n\sigma$ $k = m$ for O
  $N = m - \sigma$ $k = n - \sigma - 1$ I
- sequential_shuffler(N, k) is essentially a clever implementation of Knuth's shuffler

Theorem: Generate a uniform Wheeler DFA from $\mathcal{D}_{n,m,\sigma}$ in $O(1)$ space $O(m)$ exp. time, $O(m \log m)$ time w.h.pr. if $\sigma \leq m/en$.

Implementation: • Generates DFA with $n = 64 \cdot 10^6$ $m = 8 \cdot 10^9$ in ≈ 10 mins
• Throughput: $\geq 8 \cdot 10^6$ edges per second

Add-On: Bound on Number of WDFAs

Our Wheeler DFA representation implies a bound on number of Wheeler DFAs $\mathcal{D}_{n,\sigma}$ with n nodes and alphabet size σ.

Theorem: For $\varepsilon \in (0, 1/2]$, $n \geq 2/\varepsilon$, and $\sigma \leq (1-\varepsilon) \cdot n$

$$\log |\mathcal{D}_{n,\sigma}| \geq n\sigma + (n-\sigma)\log\sigma - (n + \log\sigma)$$

$$\log |\mathcal{D}_{n,\sigma}| \leq n\sigma + (n-\sigma)\log\sigma + O(n)$$

information-theoretic worst case #bits to encode WDFA from $\mathcal{D}_{n,\sigma}$

Our representation (opportunely encoded using succinct bitvectors) gives an encoding of size $n\sigma + (n-\sigma)\log\sigma$, thus being optimal up to an additive $O(n)$ term.

Conclusion & Future Work

Theorem: Generate a uniform Wheeler DFA from $\mathcal{D}_{n,m,\sigma}$ in $O(1)$ space
$O(m)$ exp. time
$O(m \log m)$ time w.h.pr. if $\sigma \leq m/en$.

Implementation:
- Generates DFA with $n = 64 \cdot 10^6$ $m = 8 \cdot 10^9$ in ≈ 10 mins
- Throughput: $\geq 8 \cdot 10^6$ edges per second

Future Work:
- Wheeler NFAs
- Automata of colex width p
- analyse threshold phenomena on $\mathcal{D}_{n,m,\sigma}$

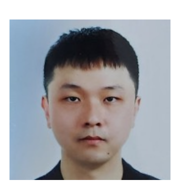
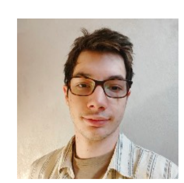
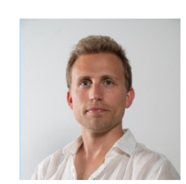
Università Ca'Foscari Venezia

That's all. Thank you! Questions?

Random Wheeler Automata

CPM 2024

Fukuoka, Japan, 25 June 2024

Ruben Becker, Davide Cenzato, Sung-Hwan Kim, Bojana Kodric, Riccardo Maso, and Nicola Prezza

This work has been funded by the European Union (ERC, REGINDEX, 101039208).Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.