Indexing and compressing regular languages

Nicola Prezza, Ca’ Foscari university of Venice, Italy

Joint work with: Nicola Cotumaccio (GSSI), Giovanna D’Agostino (uniud), Alberto Policriti (uniud), Jarno Alanko (university of Helsinki), Davide Martincigh (uniud)

Università
Ca’Foscari
Venezia

On the menu

1. **Foundations: a theory of ordered regular languages**

    a. Wheeler NFAs
    b. Wheeler languages
    c. Sorting any regular language: partial co-lex orders

2. **Sortability hierarchies of regular languages**

3. **Complexity of sorting regular languages**

4. **Open problems**

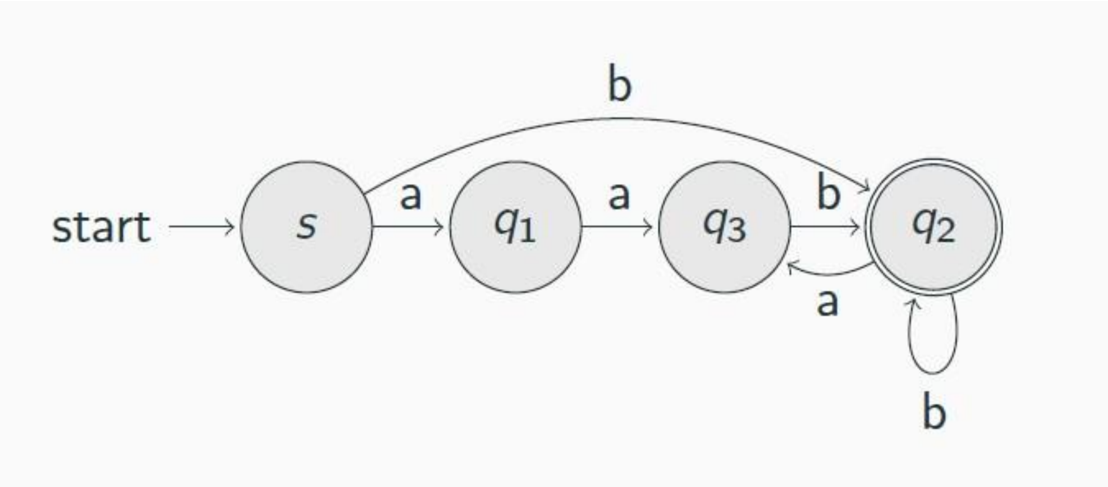
1.a Wheeler NFAs

Wheeler NFAs

[Gagie, Manzini, Sirén. "Wheeler graphs: A framework for BWT-based data structures." TCS'17]

WNFA = NFA whose states can be sorted in a **total order** respecting the co-lex axioms:

1. in(u) < in(v) ⇒ u < v
2. u < v & (u,u',a), (v,v',a) ∈ E ⇒ u' ≤ v'

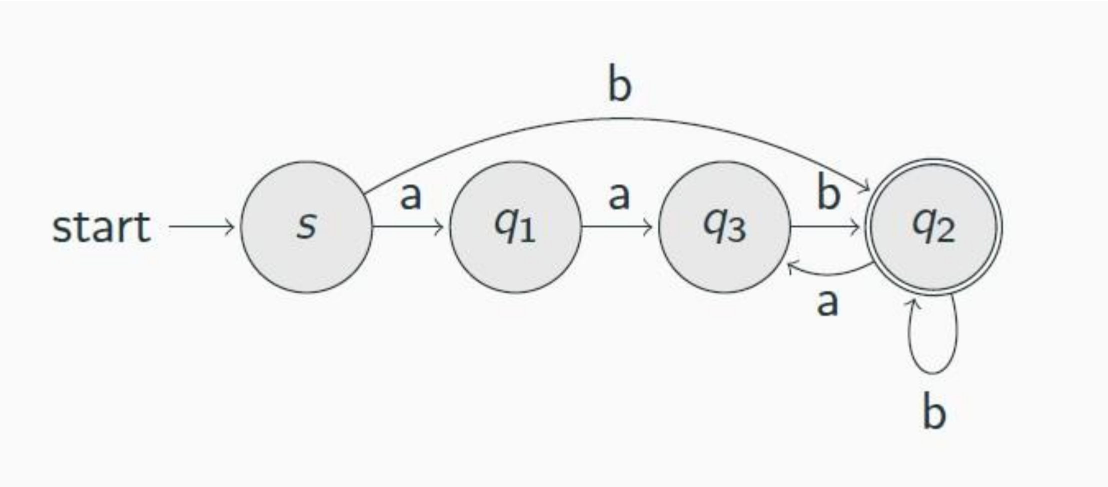


Wheeler NFAs

[Gagie, Manzini, Sirén. "Wheeler graphs: A framework for BWT-based data structures." TCS’17]

WNFAs:

- Generalize the concept of *prefix sorting* from strings to labeled graphs
- Can be stored using log(σ) + O(1) bits per edge
- Support fast subpath queries



Subpath queries on WNFAs

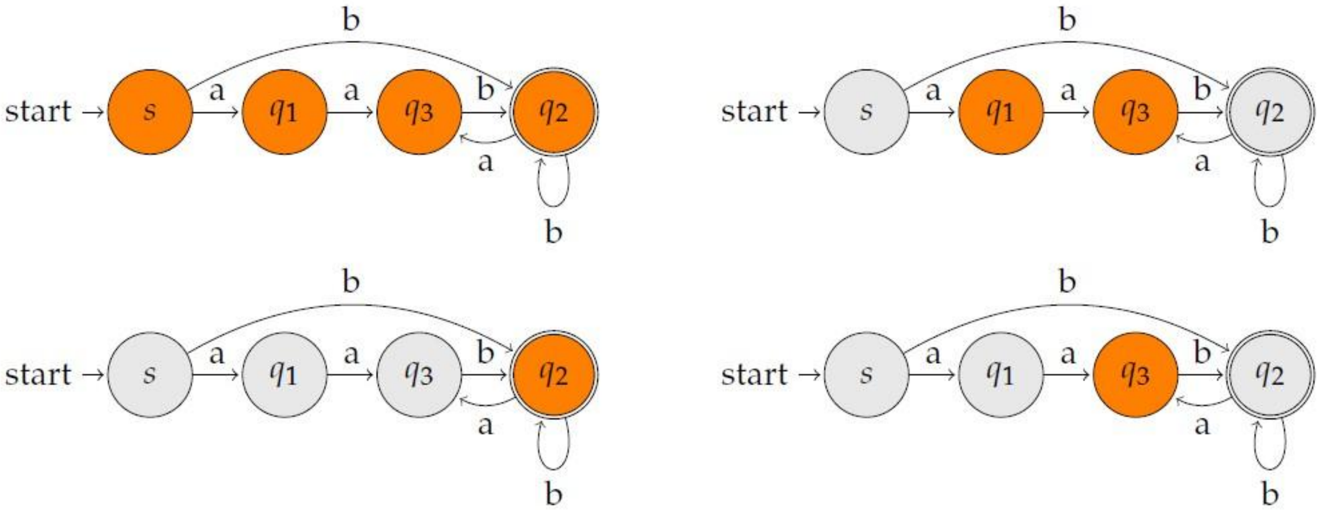


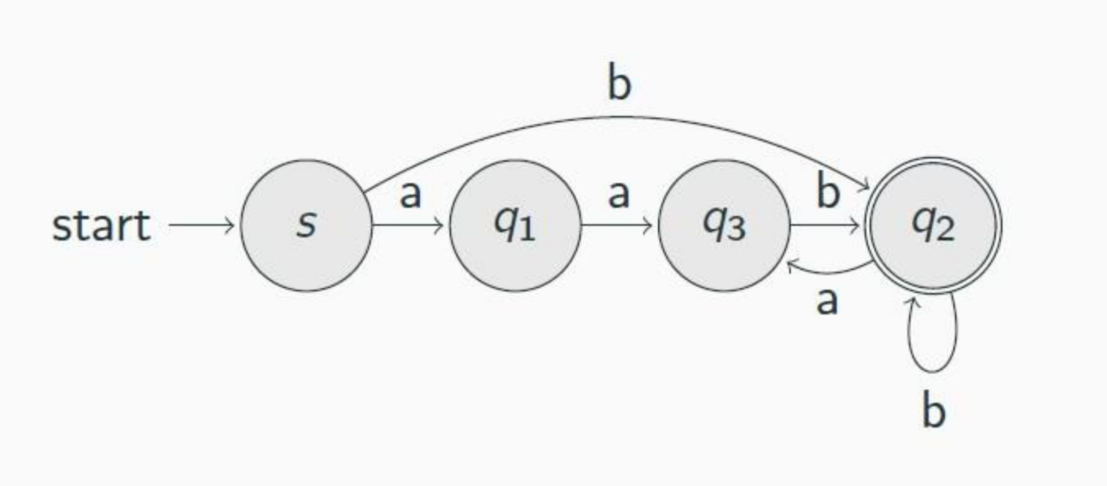
Figure 11. Searching nodes reached by a path labeled "aba" in a Wheeler graph. Top left: we begin with the nodes reached by the empty string (full range). Top right: range obtained from the previous one following edges labeled 'a'. Bottom left: range obtained from the previous one following edges labeled 'b'. Bottom right: range obtained from the previous one following edges labeled 'a'. This last range contains all nodes reached by a path labeled "aba"

1.b From Sorting NFAs to Regular Languages

A language-theoretical approach

[Alanko, D'Agostino, Policriti, P.. Regular languages meet prefix sorting. SODA'2020]

Let's take a step back, and study the problem as **a problem on regular languages**.

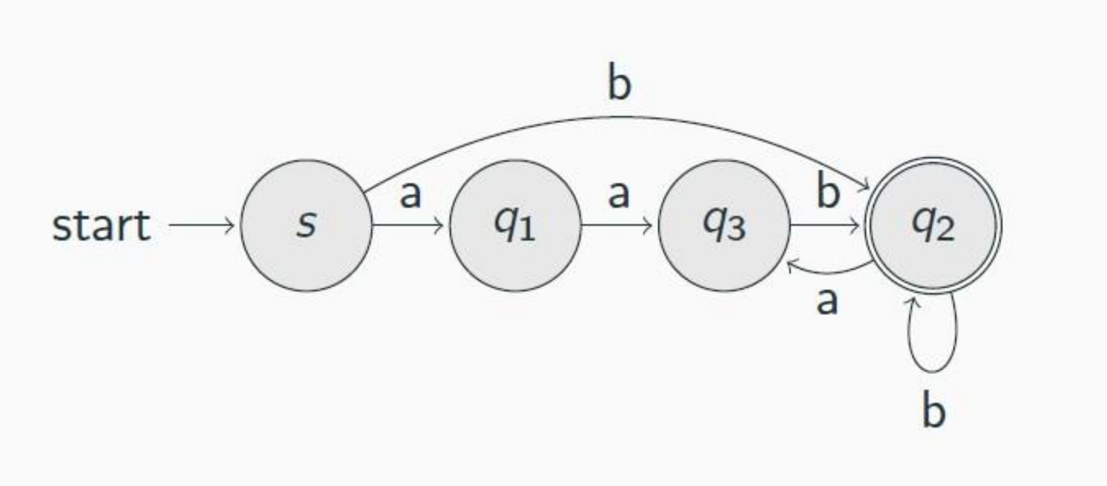


L = (ε|aa)b(ab|b)*

A language-theoretical approach

[Alanko, D'Agostino, Policriti, P.. Regular languages meet prefix sorting. SODA'2020]

- L (regular, infinite) can be finitely represented as a DFA A.
- Sort co-lexicographically all prefixes of words in L.
- Map this information on A (WDFA). What happens?



L = (ε|aa)b(ab|b)*

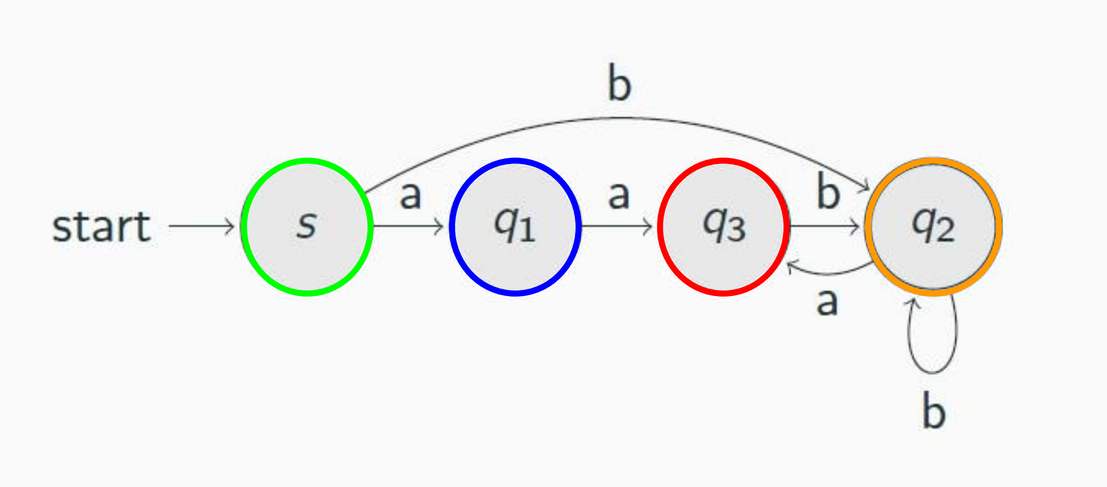
L =

ε
a
aa
ba
aaba
aababa
...
b
aab
bab
aabab
babab
...
bb
...
bbbb
....

A language-theoretical approach

[Alanko, D'Agostino, Policriti, P.. Regular languages meet prefix sorting. SODA'2020]

- L (regular, infinite) can be finitely represented as a DFA A.
- Sort co-lexicographically all prefixes of words in L.
- Map this information on A (WDFA). What happens?



L = (ε|aa)b(ab|b)*

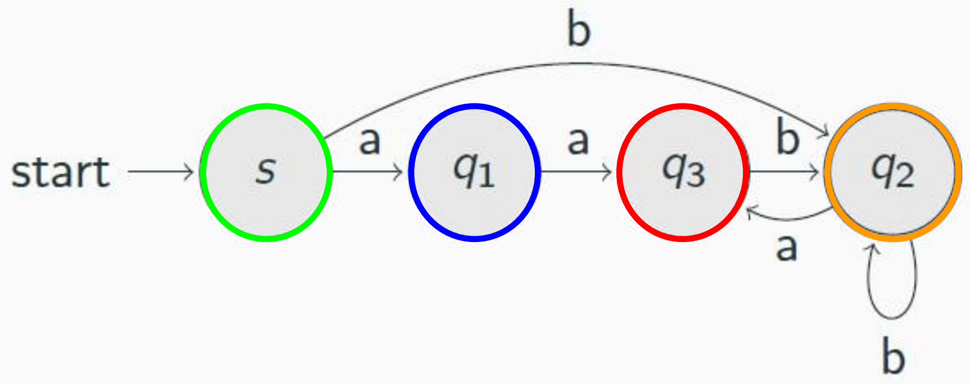
L =	ε	s
	a	q_1
	aa	
	ba	
	aaba	q_3
	aababa	
	...	
	b	
	aab	
	bab	
	aabab	
	babab	q_2
	…	
	bb	
	…	
	bbbb	
	....	

A language-theoretical approach

[Alanko, D’Agostino, Policriti, P.. Regular languages meet prefix sorting. SODA’2020]

- L (regular, infinite) can be finitely represented as a DFA A.
- Sort co-lexicographically all prefixes of words in L.
- Map this information on A (WDFA). What happens?

States form intervals and we re-obtain the Wheeler order!



L = (ε|aa)b(ab|b)*

L =

ε	**s**
a	**q_1**
aa ba aaba aababa ...	**q_3**
b aab bab aabab babab … bb … bbbb	**q_2**

Wheeler languages

L = (ε|aa)b(ab|b)*

Not a coincidence. From [Alanko et al. SODA'20]:

> **Theorem** [Myhill-Nerode theorem for W. languages]:
>
> *A regular language is Wheeler*
>
> ⇐⇒
>
> *its Myhill-Nerode equivalence classes (≡ states of minimum DFA) form a* ***finite number of intervals in co-lex order.***

ε	**[ε]**
a	**[a]**
aa ba aaba aababa ...	**[aa]**
b aab bab aabab babab … bb … bbbb	**[b]**

Wheeler languages

$L = (\varepsilon|aa)b(ab|b)^*$

Not a coincidence. From [Alanko et al. SODA'20]:

> **Theorem** [Myhill-Nerode theorem for W. languages]:
>
> *A regular language is Wheeler*
>
> $\Longleftarrow\!\!\Longrightarrow$
>
> *its Myhill-Nerode equivalence classes (≡ states of minimum DFA) form a* ***finite number of intervals in co-lex order.***

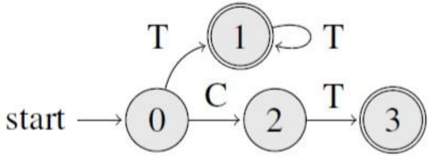
Wheeler languages = regular languages recognized by Wheeler NFAs
= regular languages recognized by Wheeler DFAs

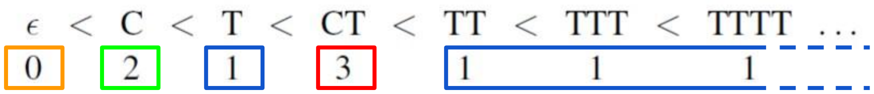
ε	**[ε]**
a	**[a]**
aa ba aaba aababa ...	**[aa]**
b aab bab aabab babab … bb … bbbb	**[b]**

Wheeler languages

Note that also the following situation could occur:

- Some MN classes are split into multiple intervals (in the example: class 1)
- Still, the number of MN *intervals* is *finite*



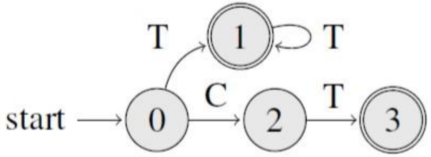


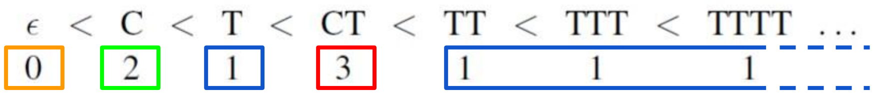
Finite number of MN intervals on the total order ≡ Wheeler language

Wheeler languages

Note that also the following situation could occur:

- Some MN classes are split into multiple intervals (in the example: class 1)
- Still, the number of MN *intervals* is *finite*





Finite number of MN intervals on the total order ≡ Wheeler language

- In this case, the **DFA is not Wheeler**, but **the language is**.
- 5 intervals ≡ 5 states of a **minimum Wheeler DFA** for the language.
- The gap between min-DFA and min-WDFA could be exponential

Wheeler languages

Another observation: previous examples concerned **DFAs**.

On **NFAs**, intervals could **overlap** in a prefix/suffix manner. In general, the picture becomes:

Wheeler languages

Another observation: previous examples concerned **DFAs**.

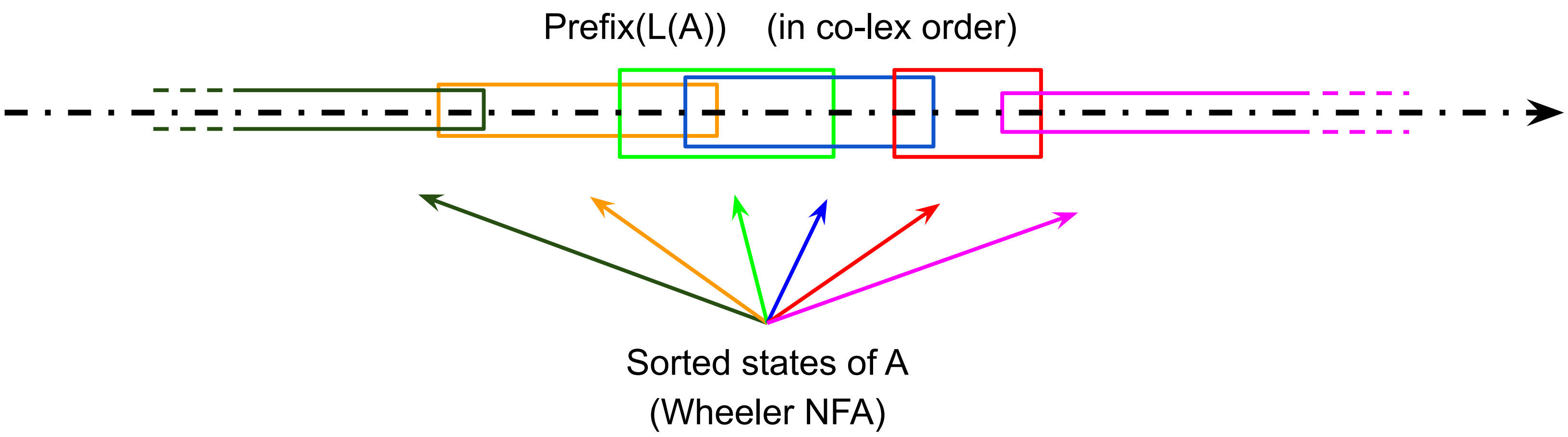
On **NFAs**, intervals could **overlap** in a prefix/suffix manner. In general, the picture becomes:

Prefix(L(A))   (in co-lex order)

Wheeler languages

Another observation: previous examples concerned **DFAs**.

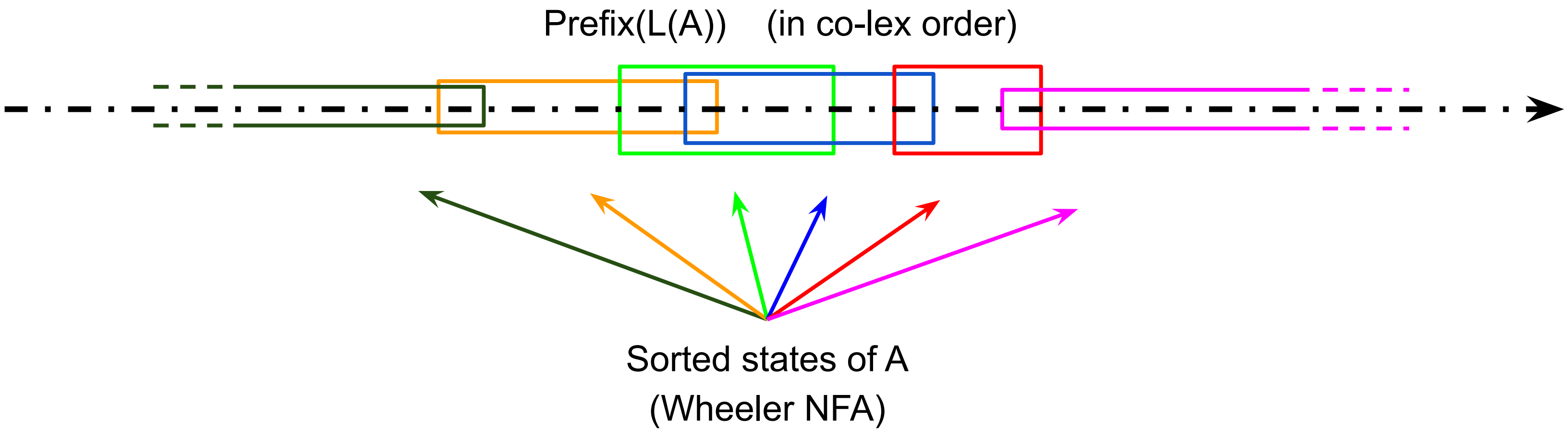
On **NFAs**, intervals could **overlap** in a prefix/suffix manner. In general, the picture becomes:



Wheeler languages

Another observation: previous examples concerned **DFAs**.

On **NFAs**, intervals could **overlap** in a prefix/suffix manner. In general, the picture becomes:



However, not all NFAs/languages are Wheeler! **can we index arbitrary NFAs/languages?**

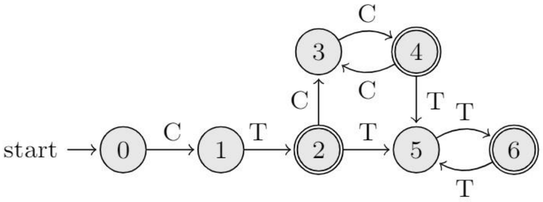
1.c Partial co-lex orders

co-lex orders

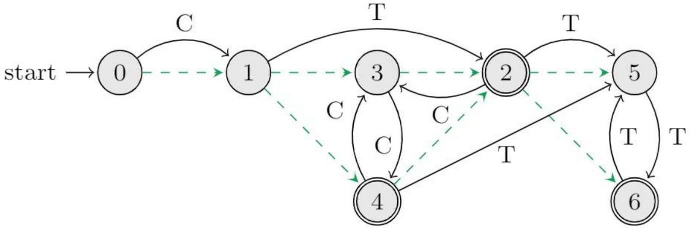
[Cotumaccio and P. *On Indexing and Compressing Finite Automata*. SODA 2021]

Solution: abandon total orders, embrace **partial orders**.

Result: any NFA admits a *partial co-lex order* of its nodes.



sort

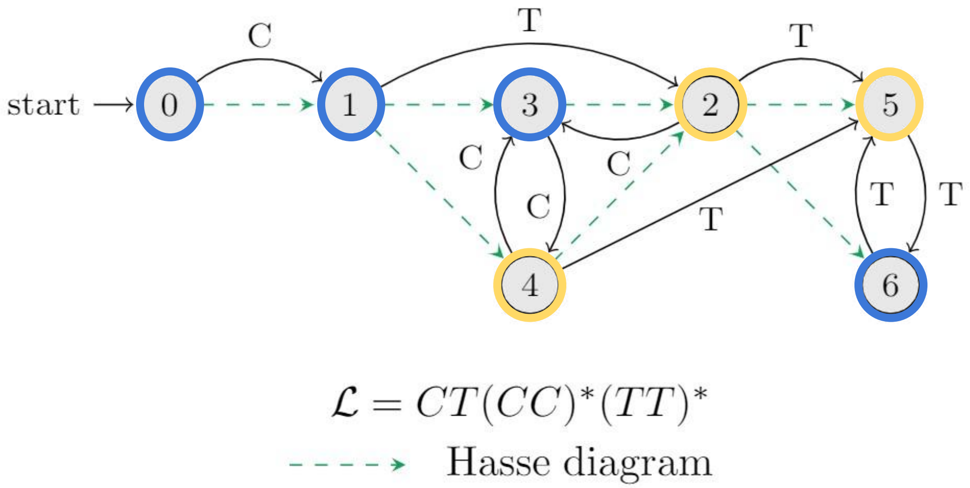


$\mathcal{L} = CT(CC)^*(TT)^*$

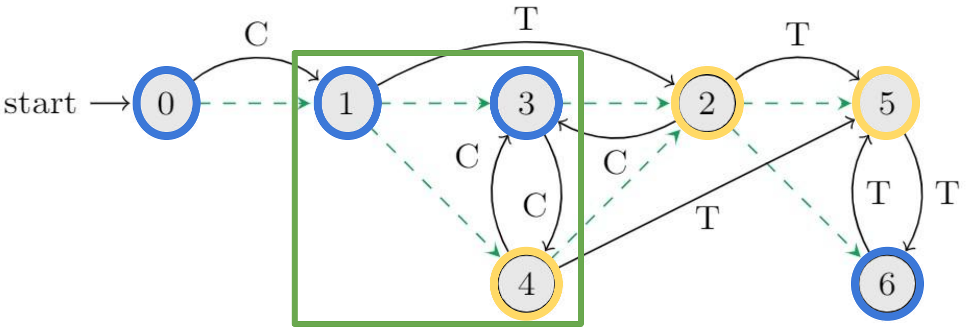
Hasse diagram

co-lex orders

- We can partition states of A into **p** totally-ordered chains.
- The smallest **p** is the order's **width** (in the example below, p = 2: {blue, yellow})



co-lex orders



$$\mathcal{L} = CT(CC)^*(TT)^*$$

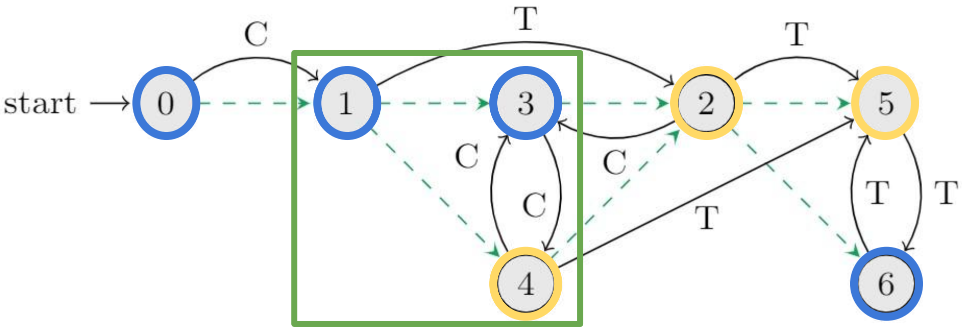
Hasse diagram

Indexing and compression still work!

Indexing ≡ states reached by any string ("C") always form a ***convex set*** *in the partial order*.

Convex set = p intervals on the p (totally-sorted) chains

co-lex orders



$$\mathcal{L} = CT(CC)^*(TT)^*$$

Hasse diagram

Indexing and compression still work!

Indexing ≡ states reached by any string ("C") always form a ***convex set*** *in the partial order*.

Convex set = p intervals on the p (totally-sorted) chains

Compression: |BWT| = O(log p) bits per edge

BWT(A) = (IN,OUT)

OUT	IN	[]	[1]	[2,2]	[2]	[1]	[1]	[1,2,2]
		0	1	3	6	4	2	5
[(1,C)]	0		(1,1,C)					
[(2,T)]	1						(1,2,T)	
[(2,C)]	3					(1,2,C)		
[(2,T)]	6							(1,2,T)
[(1,C),(2,T)]	4			(2,1,C)				(2,2,T)
[(1,C),(2,T)]	2			(2,1,C)				(2,2,T)
[(1,T)]	5				(2,1,T)			

co-lex orders

Let n = number of states, m = number of edges.

[Cotumaccio, P. SODA'21] p = width(A,<) is a fundamental parameter for NFAs:

- Powerset determinization explodes with **2^p** (rather than 2^n)*

*consequence: NFA equivalence / universality (PSPACE-complete) are FPT w.r.t. p!

co-lex orders

Let n = number of states, m = number of edges.

[Cotumaccio, P. SODA’21] p = width(A,<) is a fundamental parameter for NFAs:

- Powerset determinization explodes with $\mathbf{2^p}$ (rather than 2^n)*
- NFA compression: O(**log p**) bits per edge (rather than log n)

*consequence: NFA equivalence / universality (PSPACE-complete) are FPT w.r.t. p!

co-lex orders

Let n = number of states, m = number of edges.

[Cotumaccio, P. SODA'21] $p = \mathrm{width}(A,<)$ is a fundamental parameter for NFAs:

- Powerset determinization explodes with $\mathbf{2^p}$ (rather than 2^n)*
- NFA compression: $O(\mathbf{\log p})$ bits per edge (rather than $\log n$)
- NFA membership / pattern matching: $O(\mathbf{p^2})$ time per character (rather than m)

*consequence: NFA equivalence / universality (PSPACE-complete) are FPT w.r.t. p!

2. Sortability Hierarchies of Regular Languages

Widths of a language

From [Cotumaccio, D’Agostino, Policriti, P. (ongoing work)]:

Definition Deterministic width $\text{width}^D(L)$ of L: smallest p such that there exists A DFA with:

- width(A) = p
- L(A) = L

Widths of a language

From [Cotumaccio, D'Agostino, Policriti, P. (ongoing work)]:

Definition Deterministic width $\text{width}^D(L)$ of L: smallest p such that there exists A DFA with:

- width(A) = p
- L(A) = L

Definition Nondeterministic width $\text{width}^N(L)$ of L. Smallest p such that there exists A NFA with:

- width(A) = p
- L(A) = L

Widths of a language

From [Cotumaccio, D'Agostino, Policriti, P. (ongoing work)]:

Some results:

- Non-unicity of the smallest-width DFA (Myhill-Nerode theorem for any $\text{width}^D(L)$)
- $\text{width}^N(L) = \text{width}^D(L) = 1$ (total order) iff L is Wheeler.

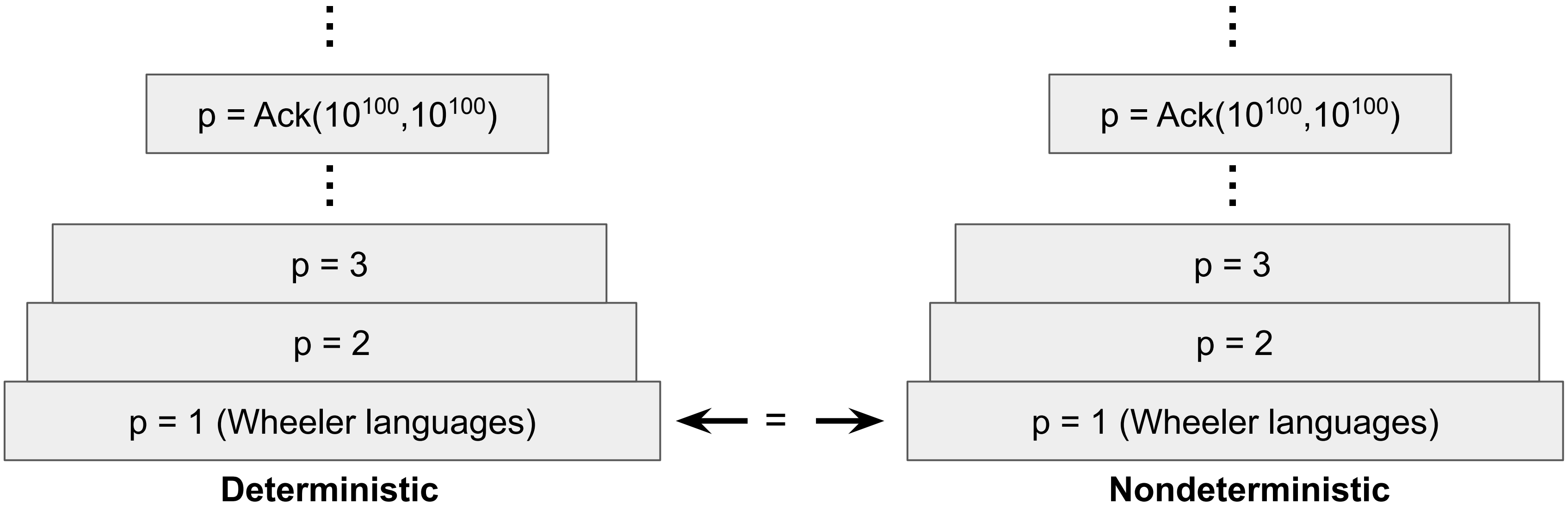
Widths of a language

Which relations exist between $\mathrm{width}^N(L)$ and $\mathrm{width}^D(L)$? We prove:

Widths of a language

Which relations exist between $\text{width}^N(L)$ and $\text{width}^D(L)$? We prove:

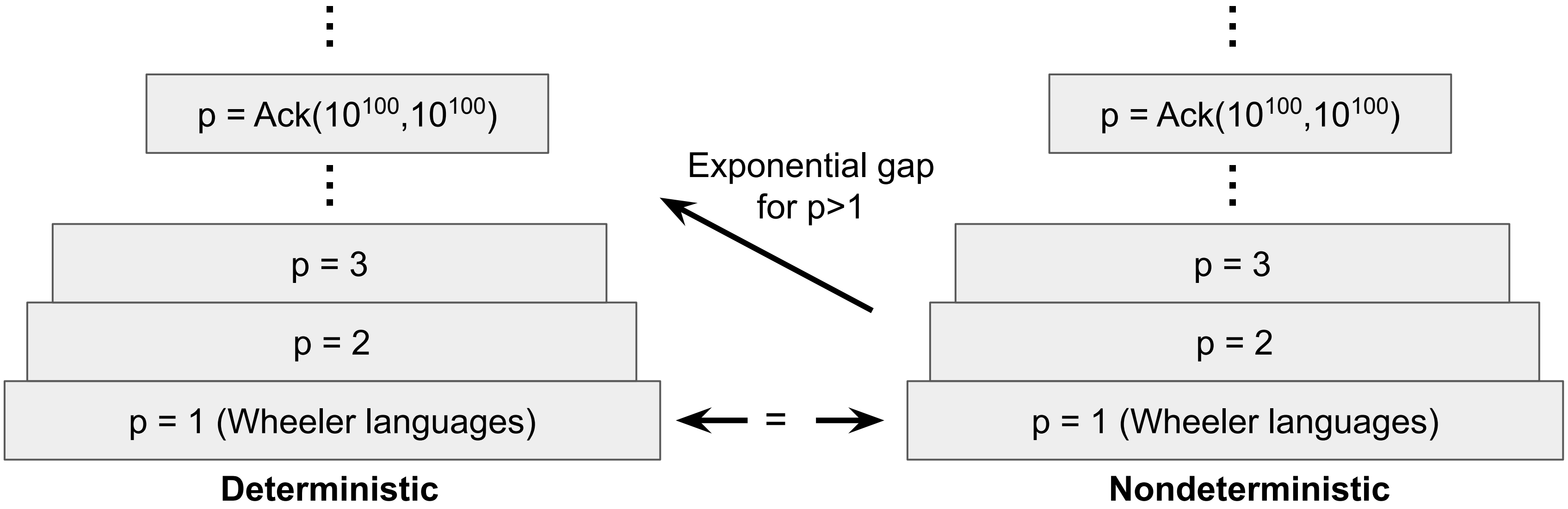
1. Both hierarchies are proper and do not collapse: for every p, there exists L such that $\text{width}^N(L) = \text{width}^D(L) = p$



Widths of a language

Which relations exist between $\text{width}^N(L)$ and $\text{width}^D(L)$? We prove:

2. $\text{width}^N(L) \leq \text{width}^D(L) \leq 2^{\text{width}^N(L)} - 1$
3. There exist infinitely many L such that $\text{width}^D(L) \geq e^{\sqrt{\text{width}^N(L)}}$



3. Complexity

Complexity

How hard is it to compute width(A) and width(L(A))?

given / compute	A: DFA	A: NFA
width(A)	$O(m^2 + n^{5/2})$ [1]	NP-hard [2]*
width(L(A))	$n^{O(\mathrm{width}(L(A)))}$ [4]**	PSPACE-hard [3]*

[1] Cotumaccio and P. On Indexing and Compressing Finite Automata. SODA'21.
[2] Gibney and Thankachan. On the hardness and inapproximability of recognizing Wheeler graphs. ESA'19
[3] D'Agostino, Martincigh, Policriti. Ordering regular languages: a danger zone. ICTCS'21
[4] Cotumaccio, D'Agostino, Policriti, P. Ongoing work.

* completeness holds in the Wheeler (p=1) case.
** note: in P for Wheeler L(A).

Complexity

How hard is it to **index** a NFA A with the optimal width(A)?

[Cotumaccio, D'Agostino, Policriti, P. Ongoing work]

Note: computing width(A) is NP-hard, but it is actually possible to side-step this problem by using a different order (of no worse width and computable in polytime):

Complexity

How hard is it to **index** a NFA A with the optimal width(A)?

[Cotumaccio, D'Agostino, Policriti, P. Ongoing work]

Note: computing width(A) is NP-hard, but it is actually possible to side-step this problem by using a different order (of no worse width and computable in polytime):

Definition (glocal order) Let $q \trianglelefteq q'$ iff $(q \leq_1 q_1 \leq_2 q_2 \ldots \leq_k q')$ for some co-lex pre-orders $\leq_1, \leq_2, \ldots, \leq_k$ and some states $q_1 \ldots q_{k-1}$.

Complexity

How hard is it to **index** a NFA A with the optimal width(A)?

[Cotumaccio, D'Agostino, Policriti, P. Ongoing work]

Note: computing width(A) is NP-hard, but it is actually possible to side-step this problem by using a different order (of no worse width and computable in polytime):

Definition (glocal order) Let $q \trianglelefteq q'$ iff ($q \leq_1 q_1 \leq_2 q_2 \ldots \leq_k q'$) for some co-lex pre-orders $\leq_1, \leq_2, \ldots, \leq_k$ and some states $q_1 \ldots q_{k-1}$.

Thm. It is possible to index a NFA A for the optimal width(A) in polynomial $O(|A|^6)$ time.

(infinite, unordered) list of open problems

1. Approximation algorithms for width(A) / width(L(A))
2. How does width(L) change with regexp operations?
3. Logical characterization of p-sortable languages (see Büchi's theorem: MSO ≡ REG)
4. Indexability lower bounds as a function of width(A) (fine-grained complexity)
5. Zoo of NFA orders (complexity, relations between different notions of width,...)
6. Algorithms for minimizing width(A) and/or number of states
7. Repetitive graph compression: run-length BWT / graph attractors
8. Dynamic data structures: maintain small width upon edge insertions/deletions
9. Generalizations: string-labeled edges, sorting context-free languages, ...
10. ...