Space Efficient Merging of Compressed Indices

Lavinia Egidi and Giovanni Manzini

learned.di.unipi.it

3rd meeting of the PRIN project
"Multicriteria Data Structures and Algorithms: from compressed to learned indexes, and beyond"

12 MARCH 2021
Online

Wheeler graphs/automata

- Born as a unifying vision of BWT variants
  [Gagie, Manzini, Sirén, 2017]
  - BWT
  - XBWT
  - grafi di de Bruijn
  - ...
- Grown as a way to lift pattern matching from strings to languages
  [Alanko, D'Agostino, Policriti, Prezza, 2020]
  [Cotumaccio, Prezza, 2021]

Wheeler automaton

[Gagie, Manzini, Sirén, 2017]

- Let $A=(V,E,\Sigma,s,F)$ be an automaton with $L(A) \subseteq \Sigma^*$

  (totally ordered alphabet Σ)

- A is Wheeler iff it admits a total order of V s.t.

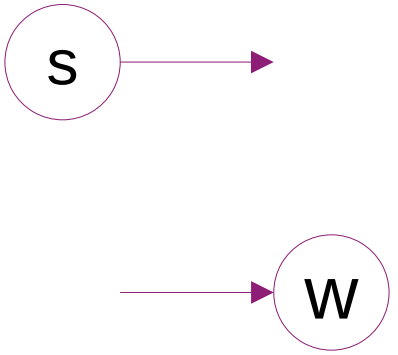
Wheeler automaton

[Gagie, Manzini, Sirén, 2017]

- Let $A=(V,E,\Sigma,s,F)$ be an automaton with $L(A) \subseteq \Sigma^*$

  (totally ordered alphabet Σ)

- A is Wheeler iff it admits a total order of V s.t.
  - states with in-degree 0 are smallest



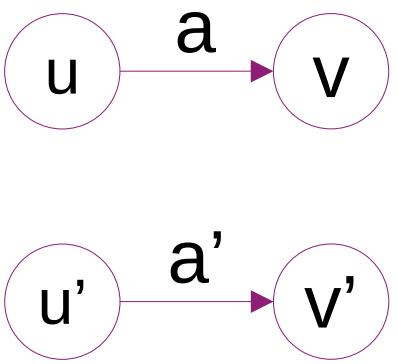
Wheeler automaton

[Gagie, Manzini, Sirén, 2017]

- Let $A=(V,E,\Sigma,s,F)$ be an automaton with $L(A) \subseteq \Sigma^*$

  (totally ordered alphabet Σ)

- A is Wheeler iff it admits a total order of V s.t.
  - states with in-degree 0 are smallest
  - for (u,v,a),(u’,v’,a’) transitions, if a<a’ then v<v’



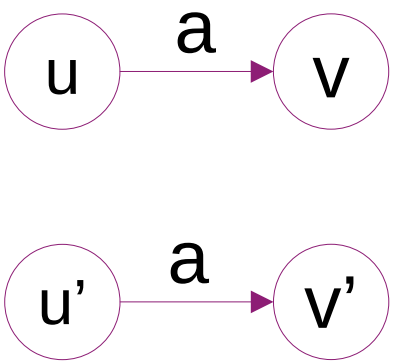
Wheeler automaton

[Gagie, Manzini, Sirén, 2017]

- Let $A=(V,E,\Sigma,s,F)$ be an automaton with $L(A) \subseteq \Sigma^*$

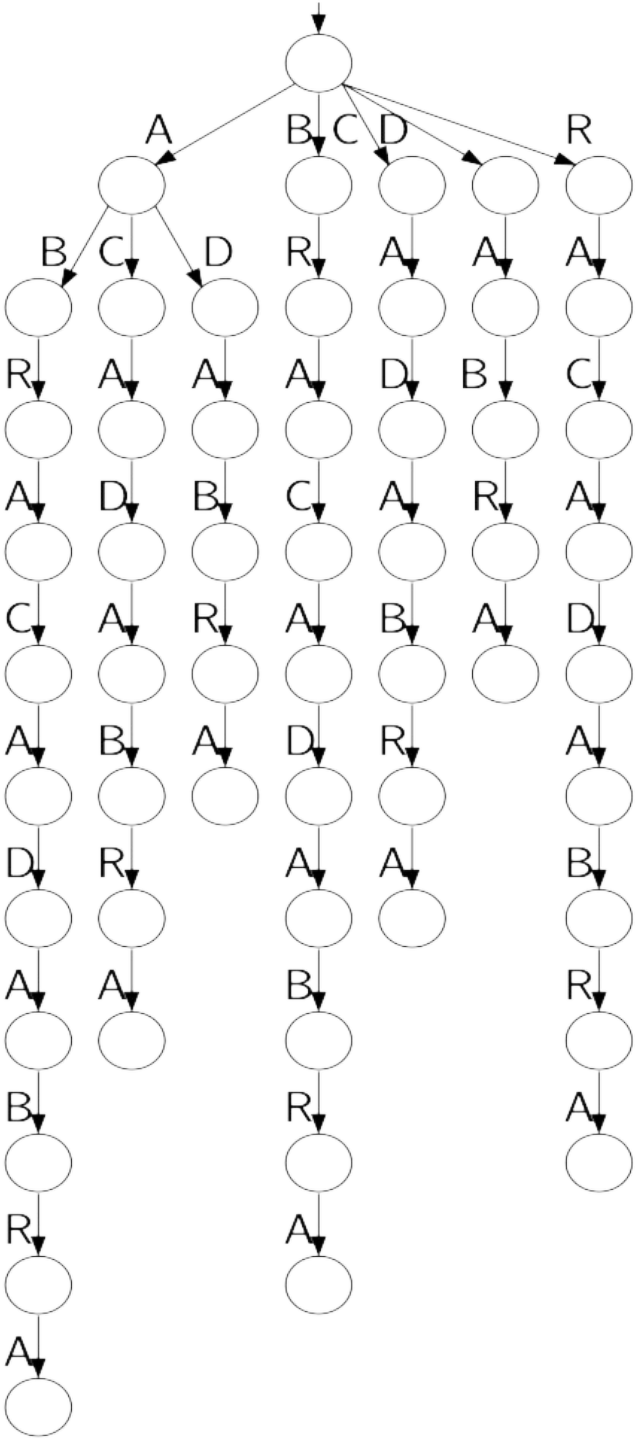
  (totally ordered alphabet Σ)

- A is Wheeler iff it admits a total order of V s.t.
  - states with in-degree 0 are smallest
  - for (u,v,a),(u',v',a') transitions, if a<a' then v<v'
  - for (u,v,a),(u',v',a) transitions, if u<u' then $v \leq v'$



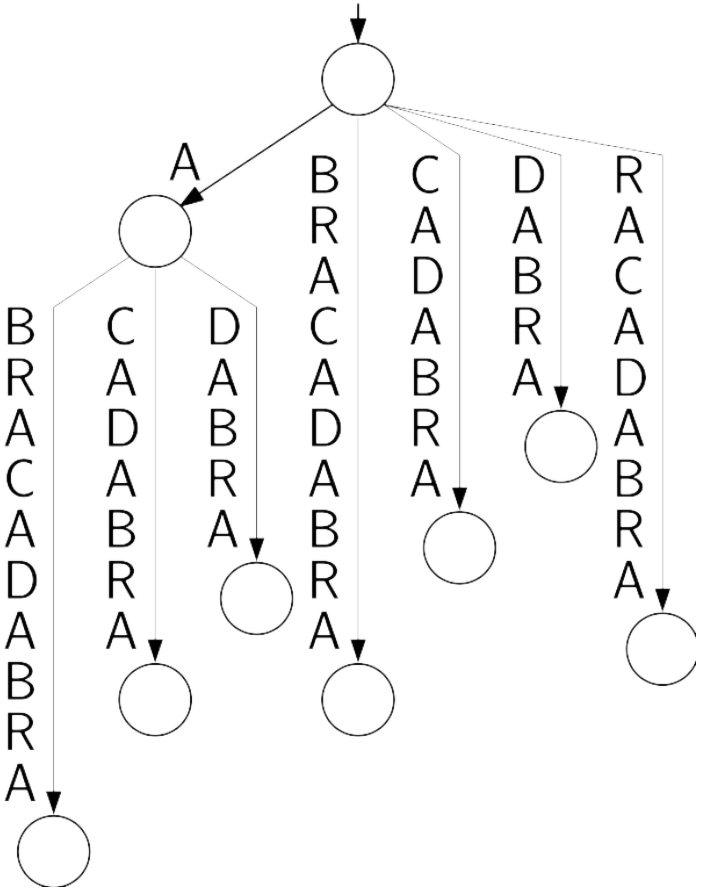
Searching for substrings of ABRACADABRA

We can use a DFA



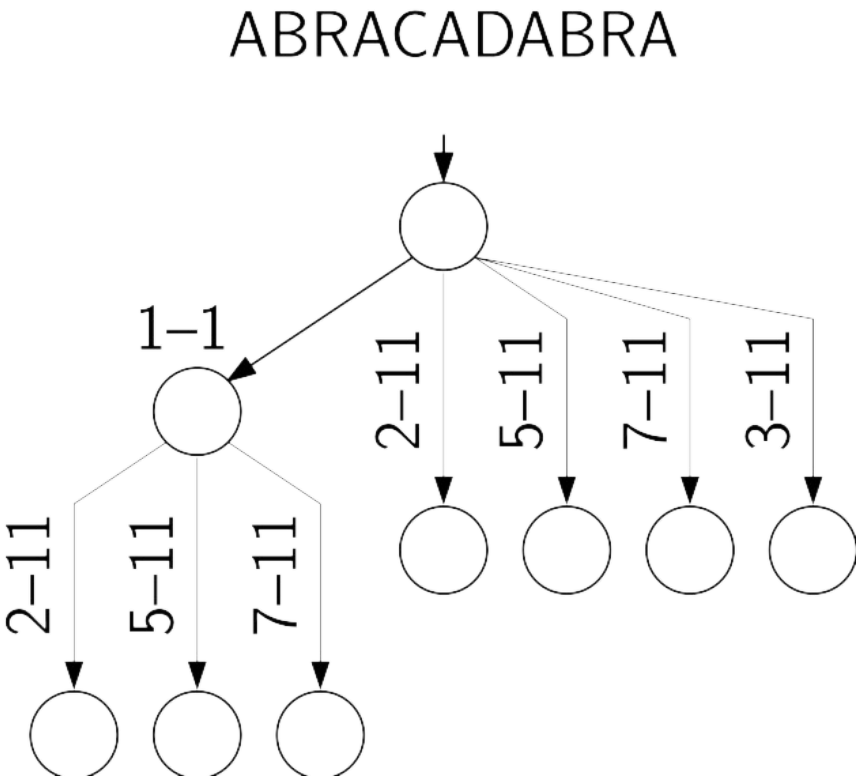
Searching for substrings of ABRACADABRA

Compacted Trie



More space efficient!

Suffix Tree

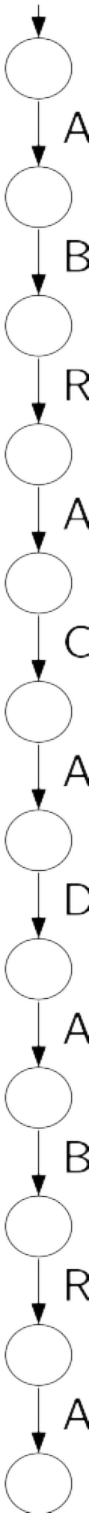


“theoretically” space efficient

Searching for substrings of ABRACADABRA

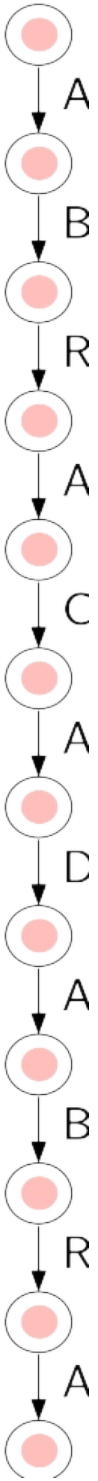
We can use a NFA!
Every state initial & final

Extremely space efficient!



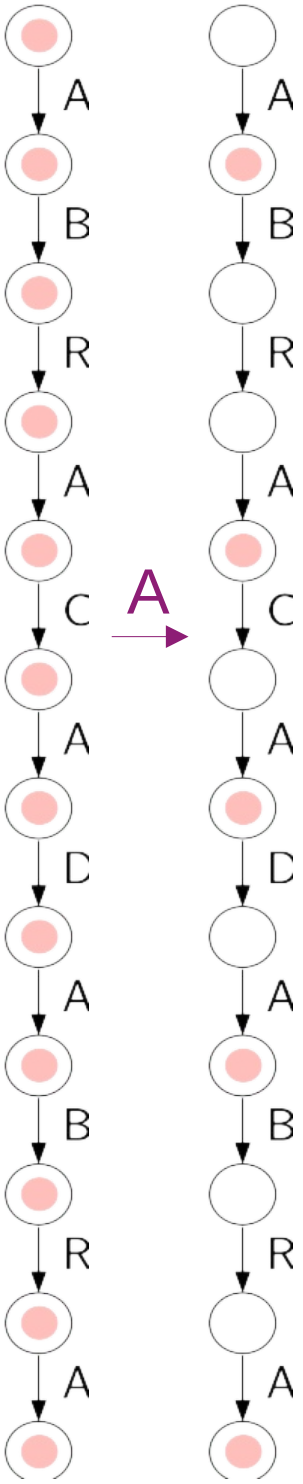
Searching for substrings of ABRACADABRA

Example:
Searching ABR



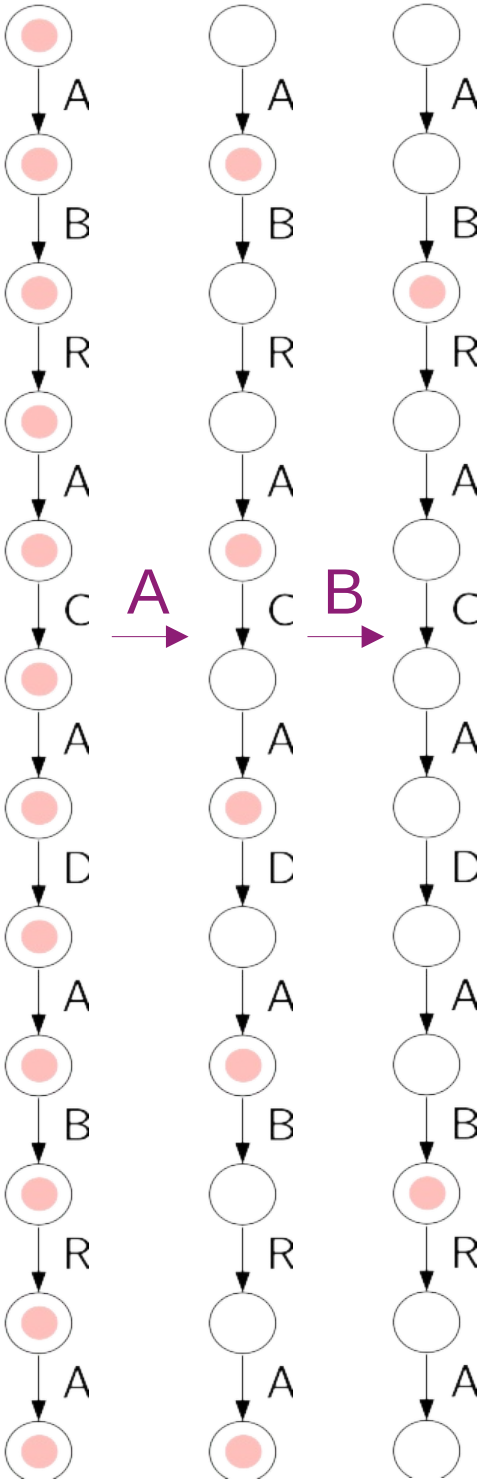
Searching for substrings of ABRACADABRA

Example:
Searching ABR



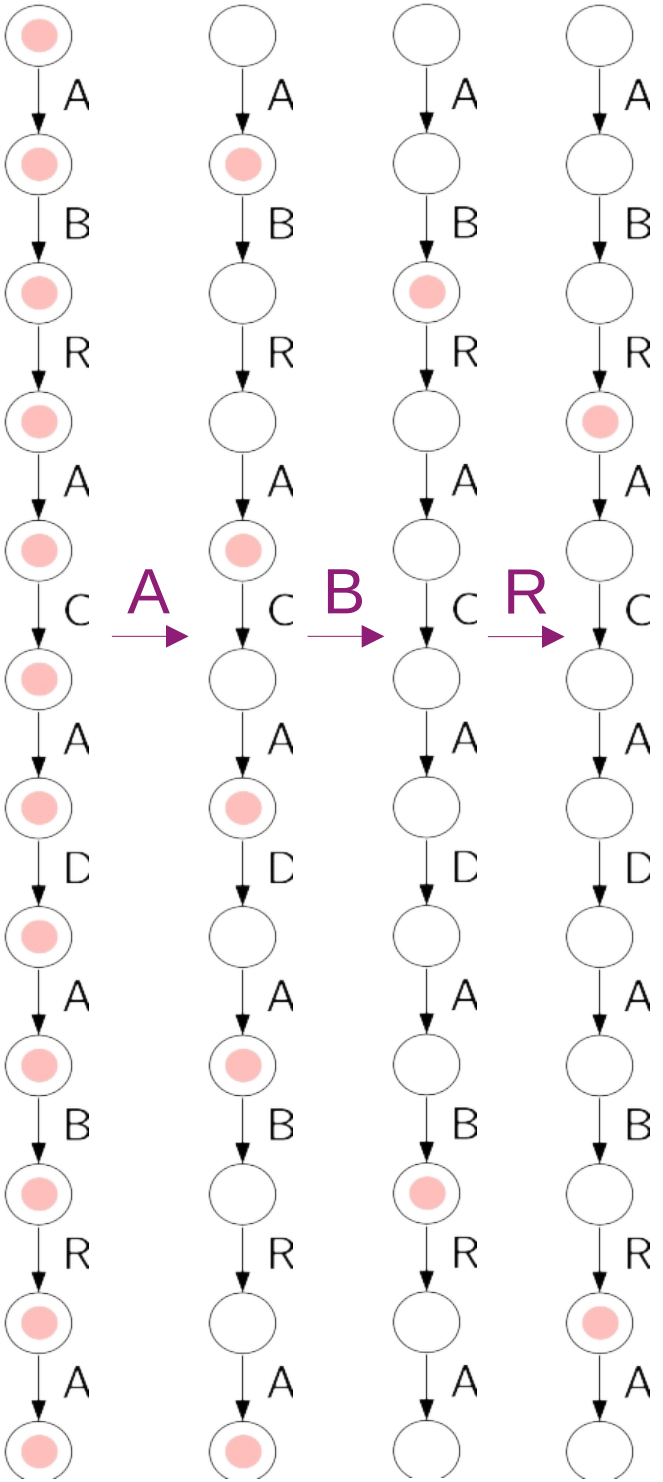
Searching for substrings of ABRACADABRA

Example:
Searching ABR



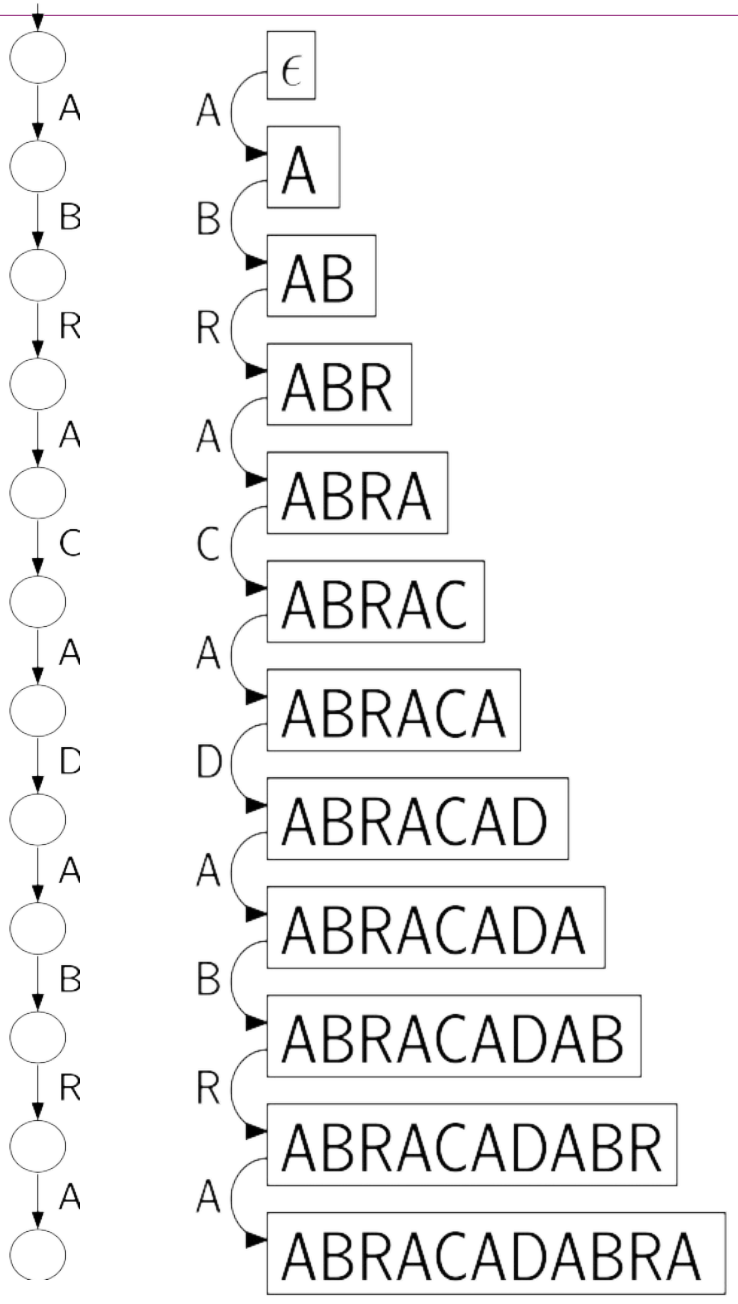
Searching for substrings of ABRACADABRA

Example:
Searching ABR



NFAs made simpler

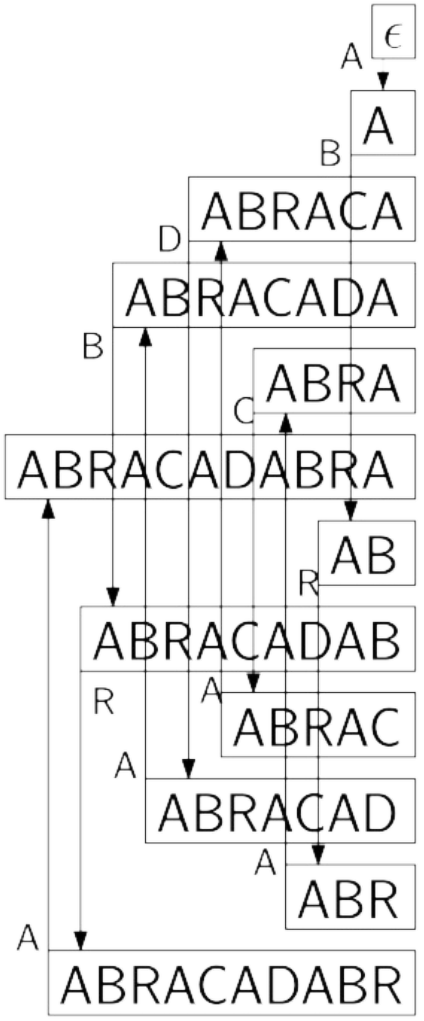
"Naturally" assign
a label to each state



NFAs made simpler

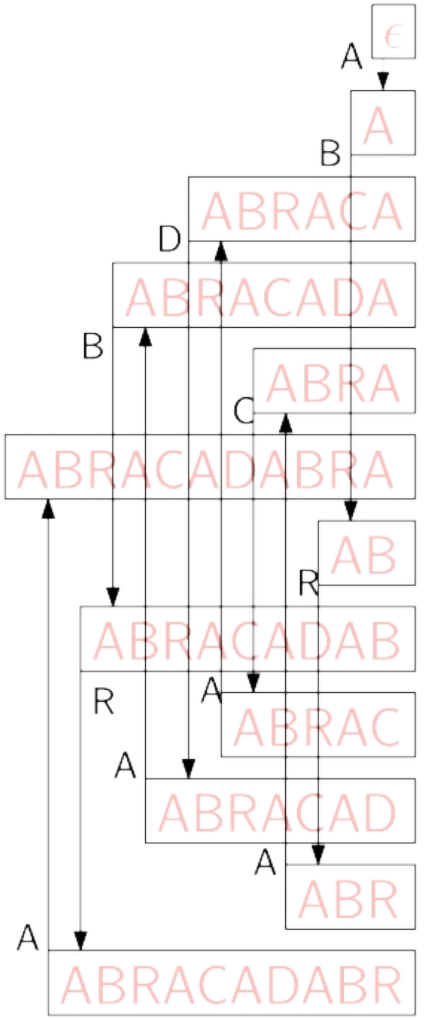
"Naturally" assign
a label to each state

Arrange NFA states
according to labels



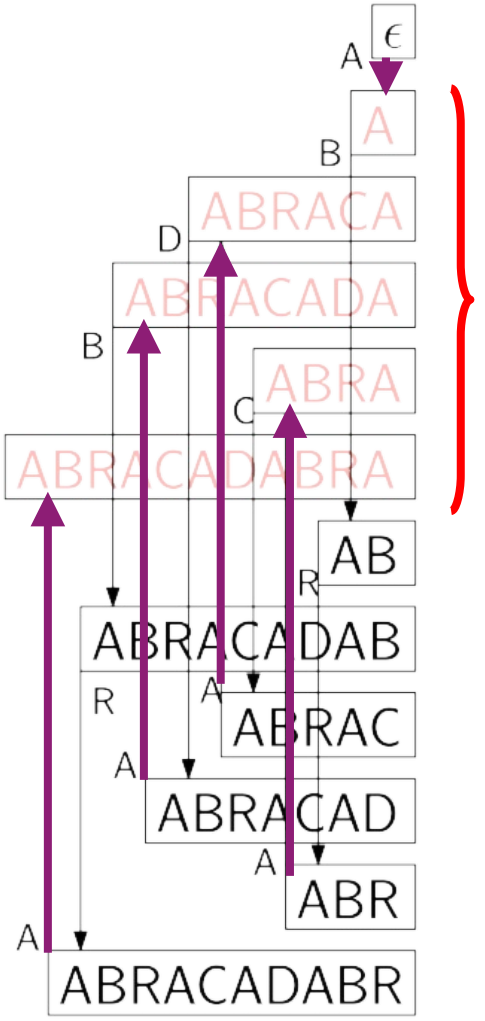
Searching in a sorted NFA

Example:
Searching ABR



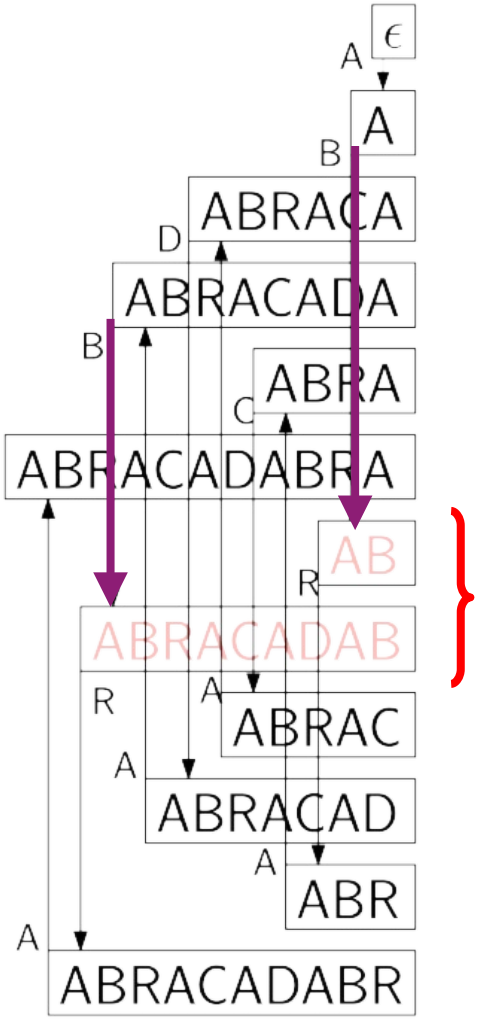
Searching in a sorted NFA

Example:
Searching ABR



Searching in a sorted NFA

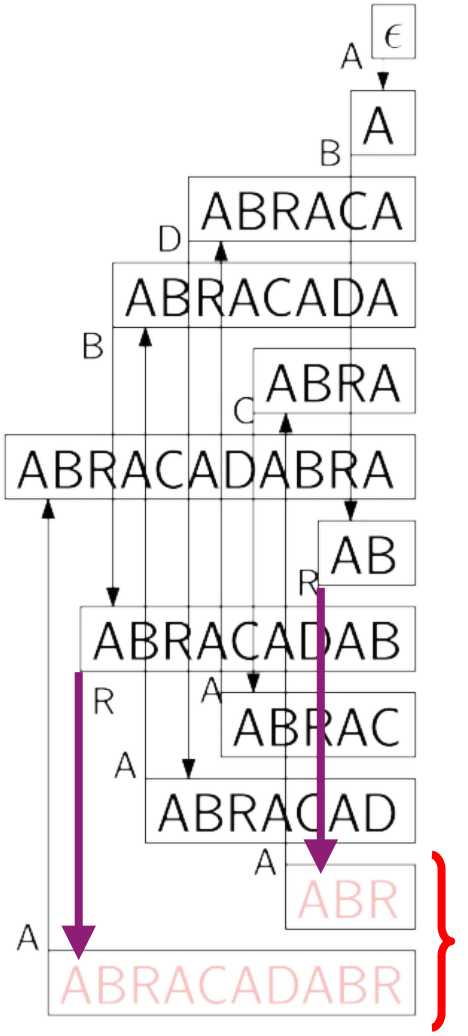
Example:
Searching ABR



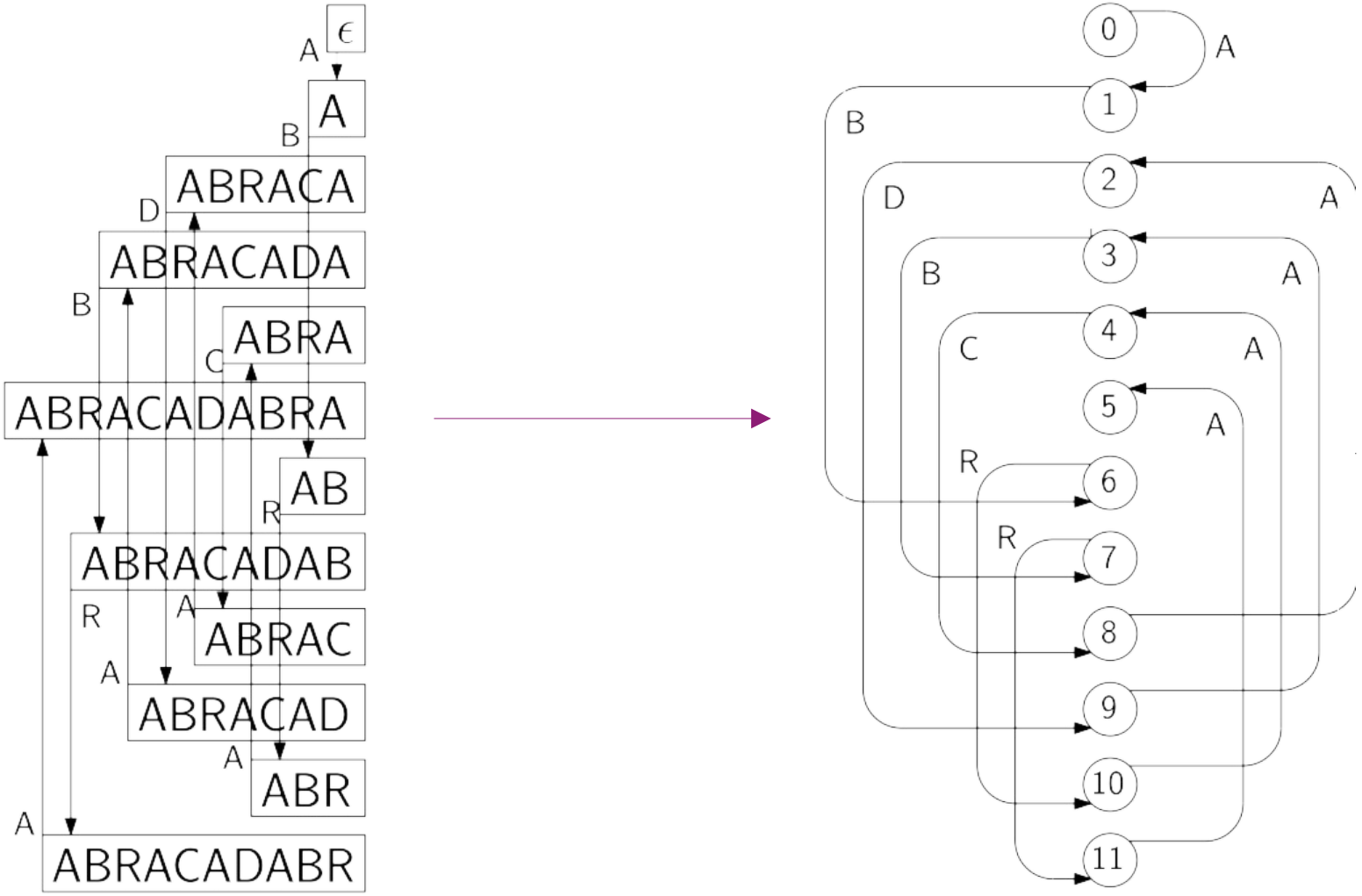
Searching in a sorted NFA

Example:
Searching ABR

Two occurrences found!



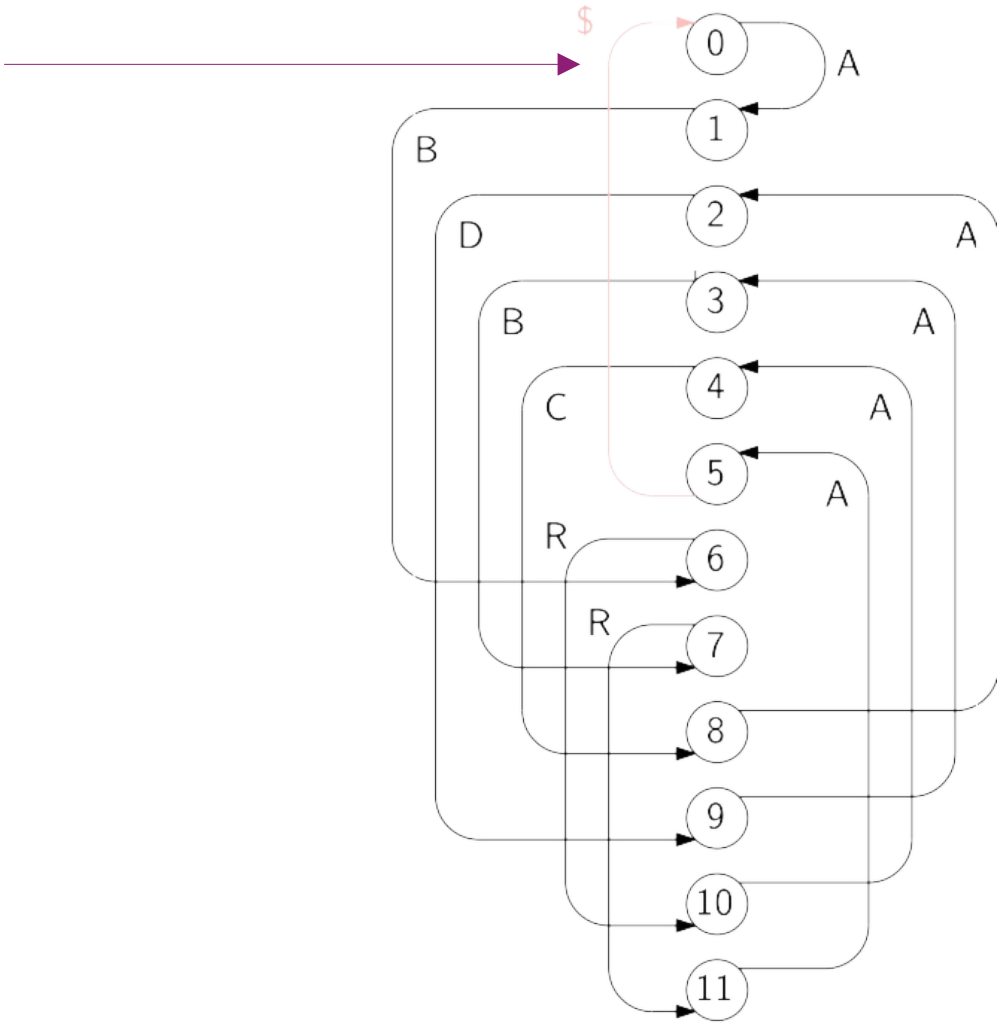
NFAs made simpler



No need to store the state labels

NFAs made simpler

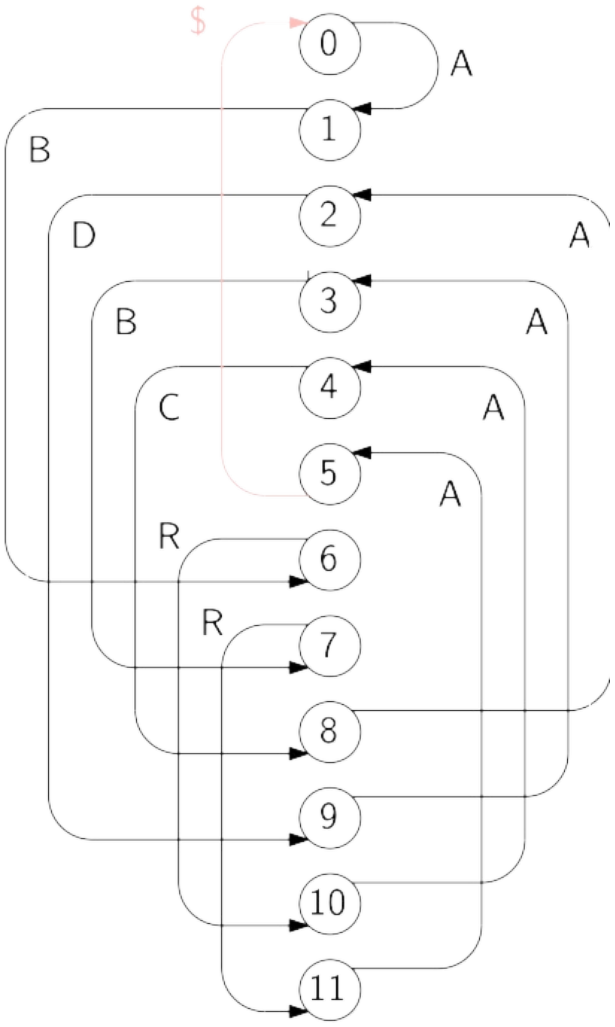
Add one arc
for symmetry



NFAs made simpler

A is Wheeler iff it admits a total order of V s.t.

1) ~~states with in-degree 0 are smallest~~
2) for (u,v,a),(u',v',a') transitions, if a<a' then v<v'
3) for (u,v,a),(u',v',a) transitions, if u<u' then $v \leq v'$



NFAs made simpler

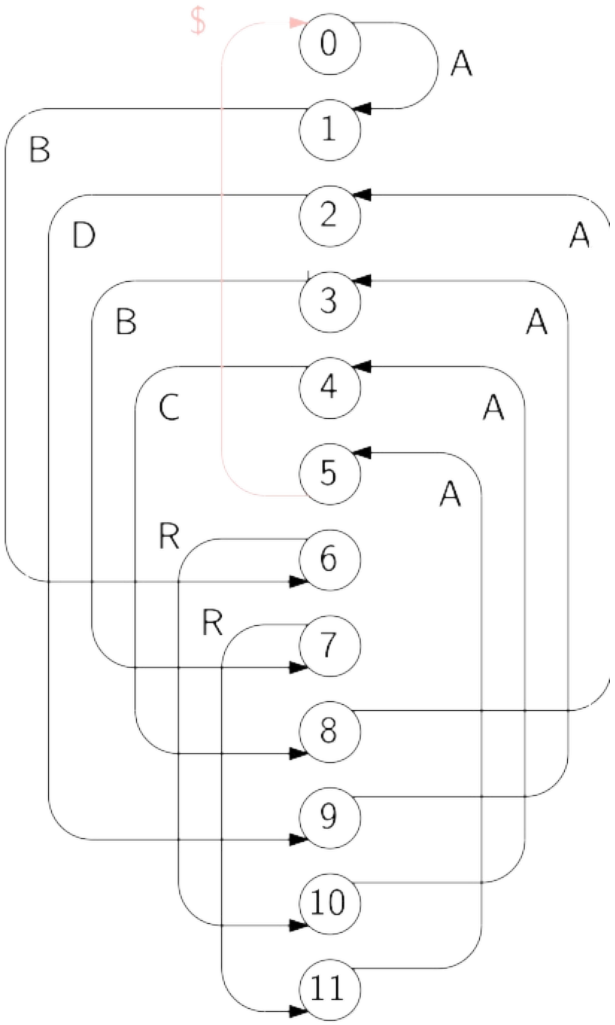
A is Wheeler iff it admits a total order of V s.t.

1) ~~states with in-degree 0 are smallest~~
2) for (u,v,a),(u',v',a') transitions, if a<a' then v<v'
3) for (u,v,a),(u',v',a) transitions, if u<u' then $v \leq v'$

Enough storing
the edge labels:
ABDBC$RRAAAA

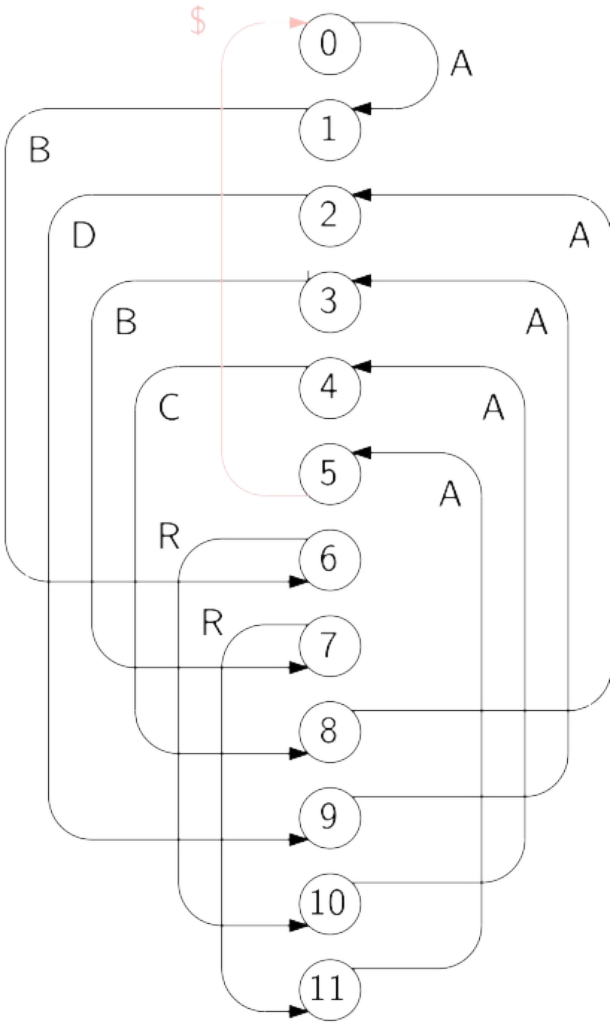
↓

BWT of
$(ABRACADABRA)^R$



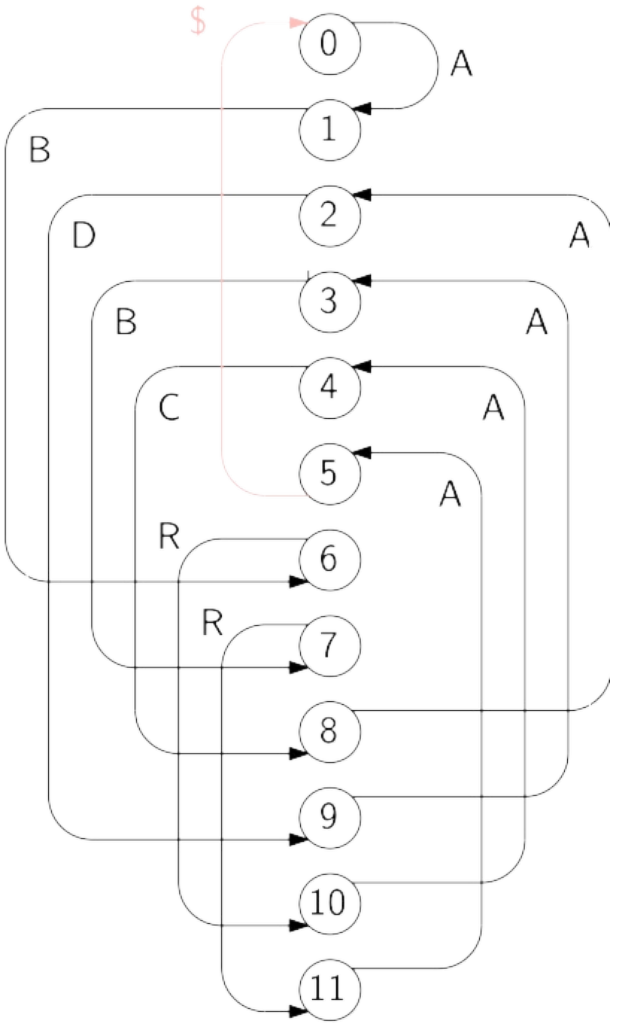
NFAs made simpler

each state has
in-degree 1
and
out-degree 1

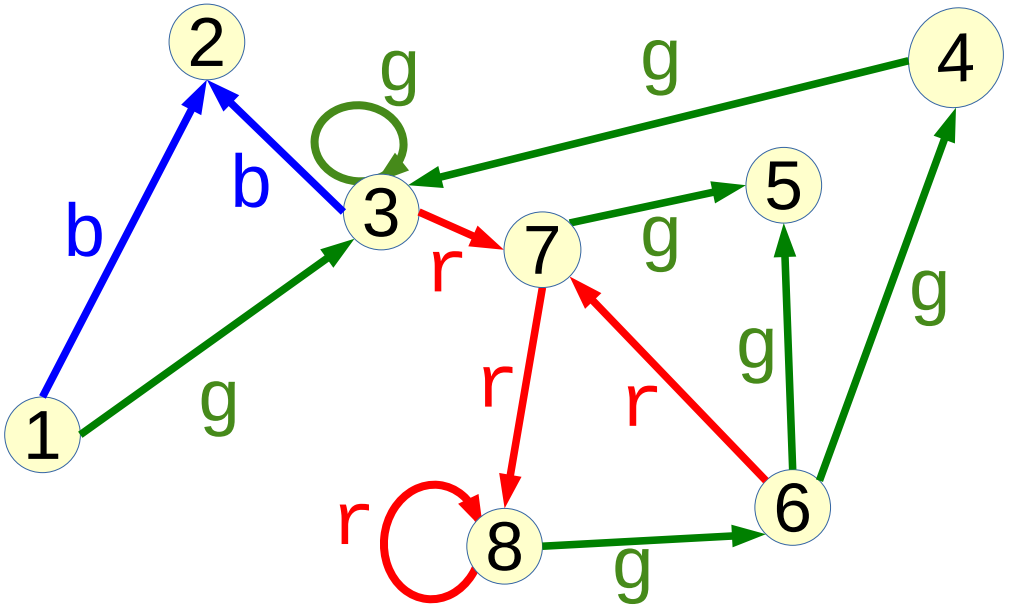


NFAs made simpler

~~each state has in-degree 1 and out-degree 1~~



A colorful 8-state Wheeler automaton

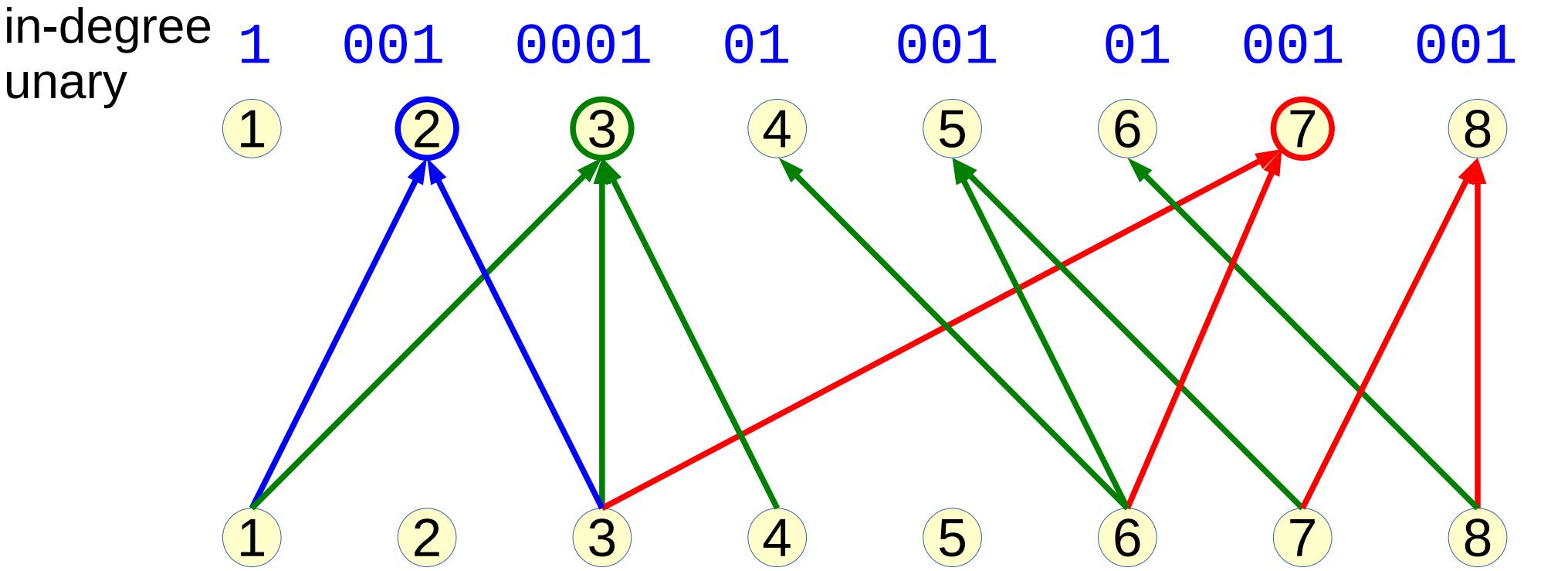


WO: 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8

Incoming labels: b g r

Succinct representation of a Wheeler automaton

inspired by [Bowe, Onodera, Sadakane, Shibuya 2012]



labels: bg bgr g ggr gr gr

unary out-degree: 001 1 0001 01 1 0001 001 001

starting states for incoming labels : b → 2 g → 3 r → 7

Succinct representation of a Wheeler automaton

[Gagie, Manzini, Sirén, 2017]

- succint representation

  $2(|V|+|E|) + |E| \log|\Sigma|+ |\Sigma| \log |E| + l.o.t.$
- forward/backward traversal of an edge in time $O(\log|\Sigma|)$

Succinct representation of a Wheeler automaton

[Gagie, Manzini, Sirén, 2017]

- succint representation

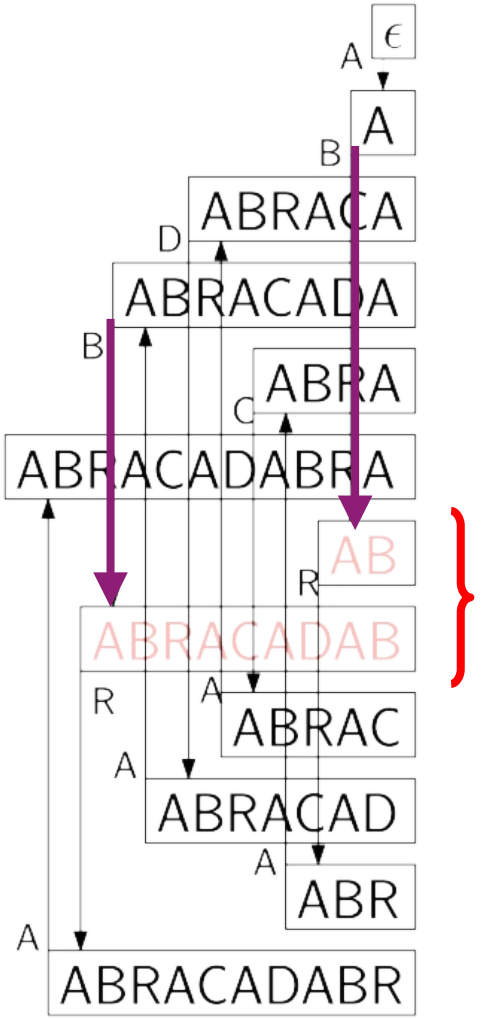
  $2(|V|+|E|) + |E| \log|\Sigma|+ |\Sigma| \log |E| + l.o.t.$
- forward/backward traversal of an edge in time $O(\log|\Sigma|)$

Why does it work?

1) in a WA the set of states that are reached via a string is an interval (wrt the ordering)

Searching in a sorted NFA

Example:
Searching ABR



Succinct representation of a Wheeler automaton

[Gagie, Manzini, Sirén, 2017]

- succinct representation

  $2(|V|+|E|) + |E| \log|\Sigma| + |\Sigma| \log |E| + l.o.t.$
- forward/backward traversal of an edge in time $O(\log|\Sigma|)$

Why does it work?

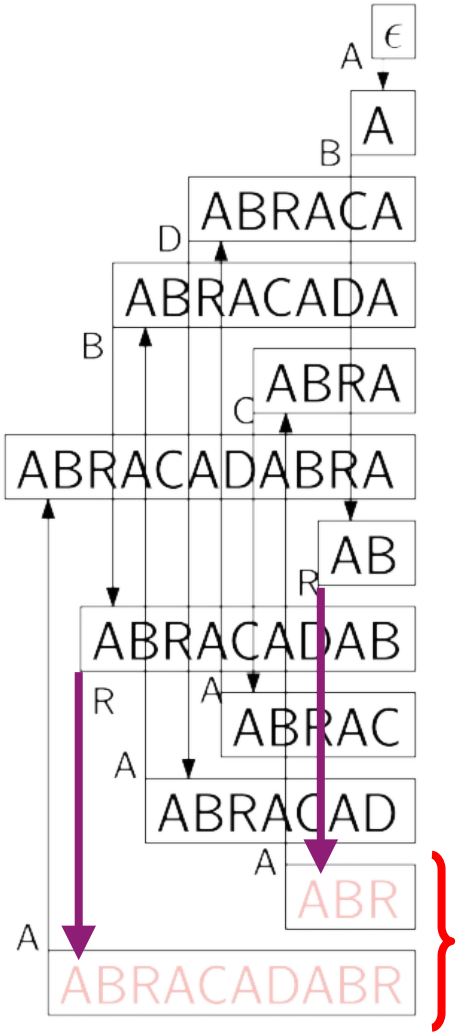
1) in a WA the set of states that are reached via a string is an interval (with respect to the Wheeler order)

2) adding a character $c \rightarrow$ other interval

Searching in a sorted NFA

Example:
Searching ABR

Two occurrences found!



Succinct representation of a Wheeler automaton

[Gagie, Manzini, Sirén, 2017]

- succinct representation

  $2(|V|+|E|) + |E| \log|\Sigma| + |\Sigma| \log |E| + l.o.t.$
- forward/backward traversal of an edge in time $O(\log|\Sigma|)$

Why does it work?

1) in a WA the set of states that are reached via a string is an interval (with respect to the Wheeler order)

2) adding a character $c \rightarrow$ other interval

3) in a total order an interval can be denoted by its endpoints

Recognizing Wheelerness of A

- is NFA A Wheeler? find an ordering:
  - decidable: NP-complete if $d \geq 5$
    [Gibney, Thankachan, 2019]
  - $\in$ P if $d \leq 2$
    [Alanko, D'Agostino, Policriti, Prezza, 2020]

Wheeler languages

L is wheeler if there is a Wheeler automaton A that decides it

- Some properties:
  - finite and cofinite languages are Wheeler
  - closed for:
    - intersections ✓
  - not closed for:
    - unions ✗
    - complements ✗
    - concatenations ✗
    - Kleene star ✗

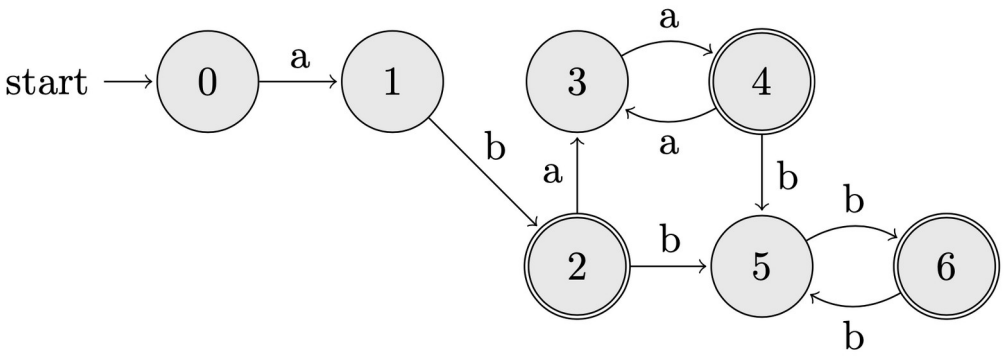
Recognizing Wheelerness of L

[Alanko, D'Agostino, Policriti, Prezza, 2020]

- is L Wheeler?
  - decidable: if it is expressed by an NFA
  - $\in$ P: if it is expressed by a DFA
  - minimum Wheeler DFA equivalent to a DFA with n states has $\Omega(2^{n/4})$ states

Extension to arbitrary automata

[Cotumaccio, Prezza, 2021]

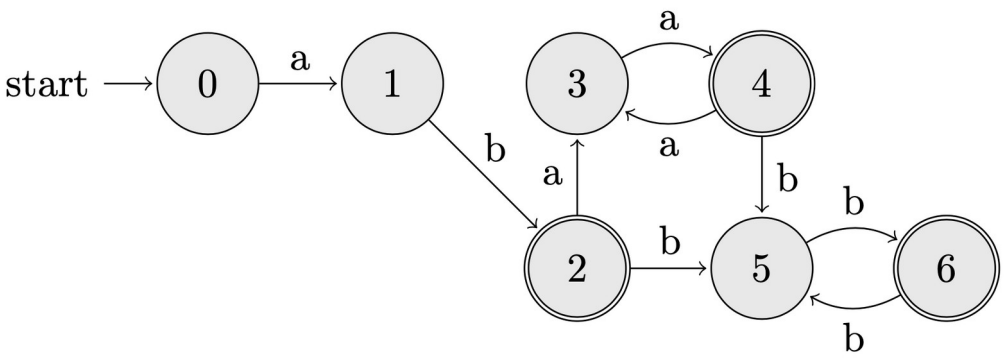


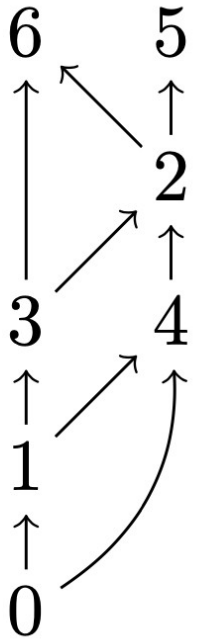
not Wheeler:
by 3), if 3 < 4, (3,4,a) and (4,3,a) $\Rightarrow 4 \leq 3$

A is Wheeler iff it admits a total order of V s.t.
1) states with in-degree 0 are smallest
2) for (u,v,a),(u',v',a') transitions, if a<a' then v<v'
3) for (u,v,a),(u',v',a) transitions, if u<u' then $v \leq v'$

Extension to arbitrary automata

[Cotumaccio, Prezza, 2021]





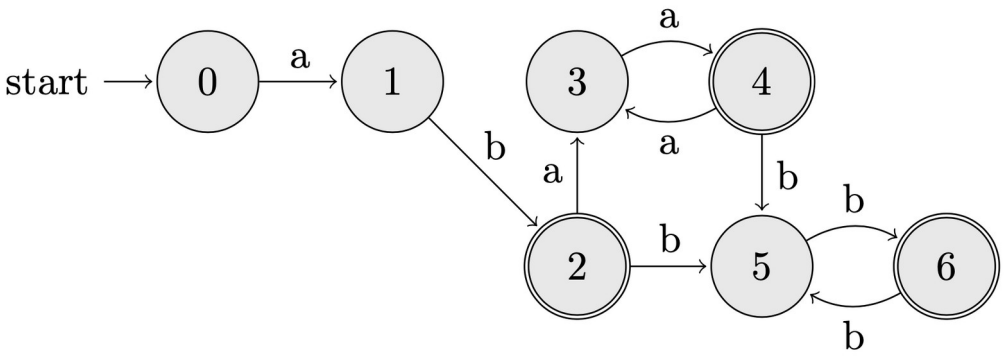
a co-lexicographic partial order of the states

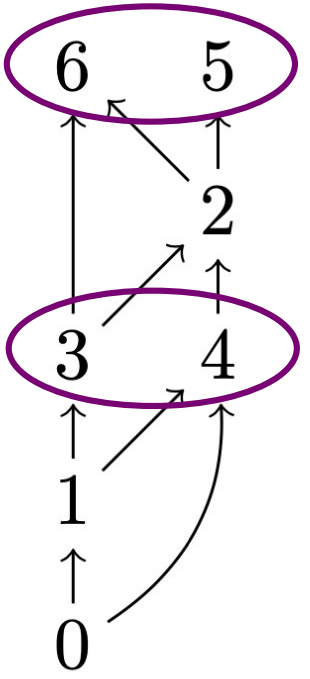
A is Wheeler iff it admits a total order of V s.t.

1) states with in-degree 0 are smallest
2) for (u,v,a),(u',v',a') transitions, if a<a' then v<v'
3) for (u,v,a),(u',v',a) transitions, if u<u' then $v \leq v'$

Extension to arbitrary automata

[Cotumaccio, Prezza, 2021]





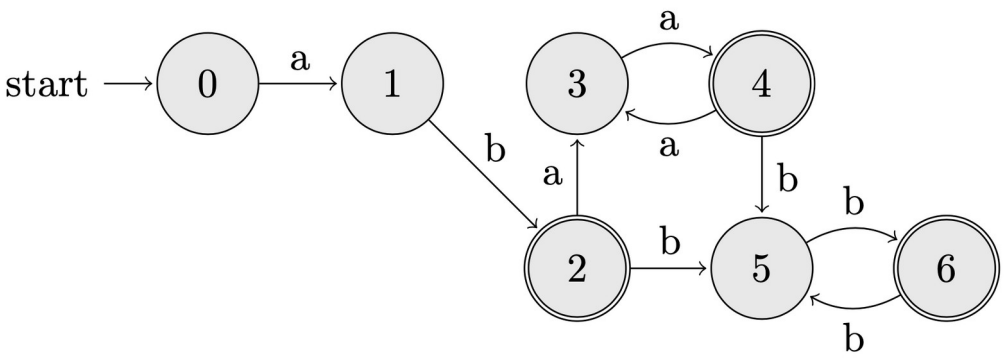
p-sortable automaton ⇐ a co-lexicographic partial order of the states of width p=2

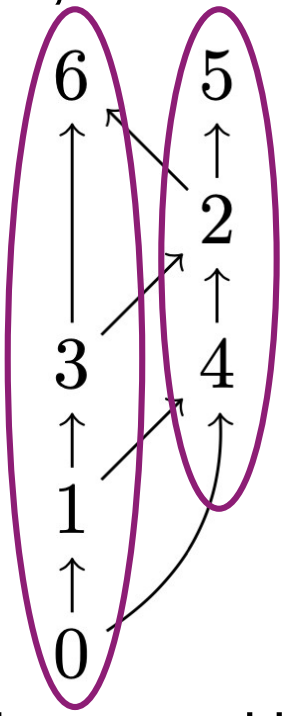
A is Wheeler iff it admits a total order of V s.t.

1) states with in-degree 0 are smallest
2) for (u,v,a),(u',v',a') transitions, if a<a' then v<v'
3) for (u,v,a),(u',v',a) transitions, if u<u' then v ≤ v'

Extension to arbitrary automata

[Cotumaccio, Prezza, 2021]





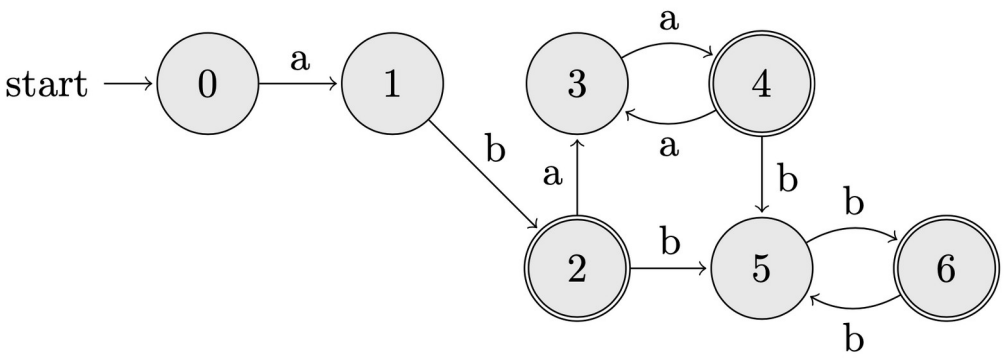
p-sortable automaton ⟸⟹ chain decomposition p=2 chains

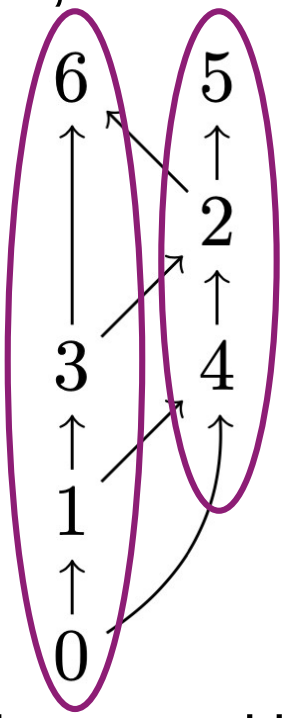
A is Wheeler iff it admits a total order of V s.t.

1) states with in-degree 0 are smallest
2) for (u,v,a),(u',v',a') transitions, if a<a' then v<v'
3) for (u,v,a),(u',v',a) transitions, if u<u' then v ≤ v'

Extension to arbitrary automata

[Cotumaccio, Prezza, 2021]





Each chain is a total order and has properties 1)-3) of a Wheeler automaton

chain decomposition
p=2 chains

Why does it work?

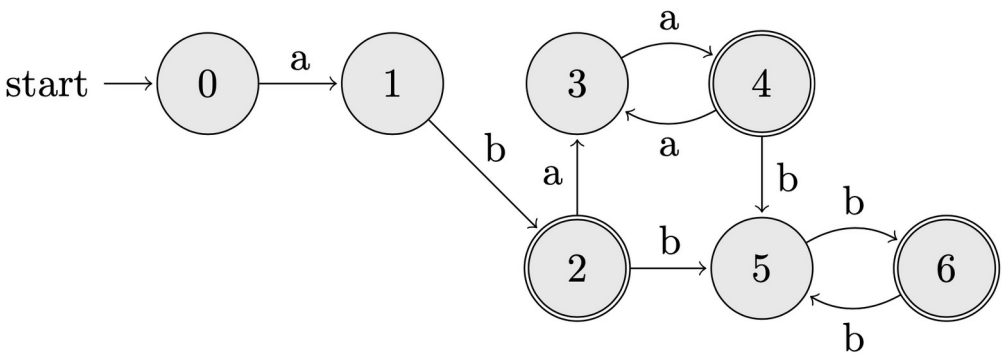
1) in a WA the set of states that are reached via a string is an interval

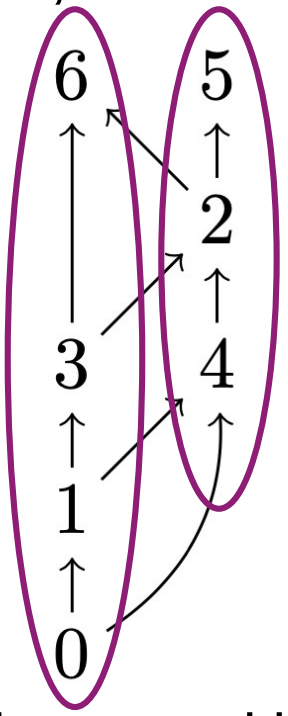
2) adding a character *c* → other interval

3) total order → specify interval through its endpoints

Extension to arbitrary automata

[Cotumaccio, Prezza, 2021]





Each chain is a total order and has properties 1)-3) of a Wheeler automaton

⇓

essentially encode each chain as a WA

chain decomposition
p=2 chains

Why does it work?

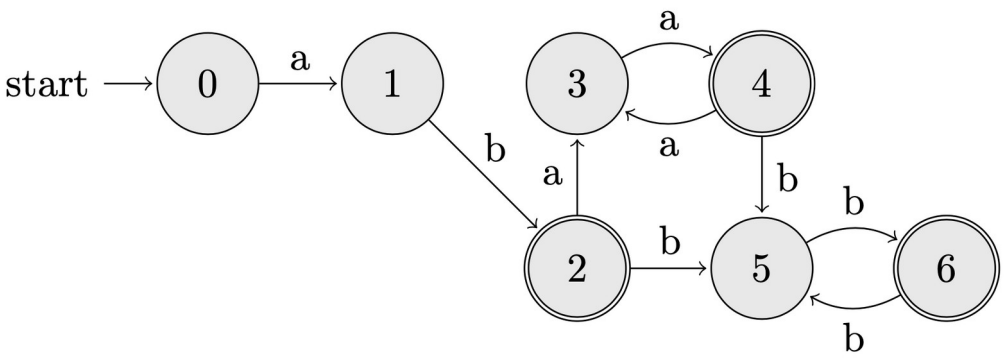
1) in a WA the set of states that are reached via a string is an interval

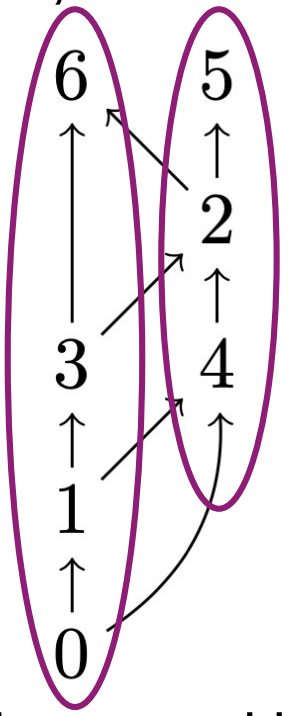
2) adding a character *c* → other interval

3) total order → specify interval through its endpoints

Extension to arbitrary automata

[Cotumaccio, Prezza, 2021]





succint representation

- O(|E| (log |Σ| + log p) + |V|) space
- O(p^2 log(p|Σ|)) time for forward traversal of edges

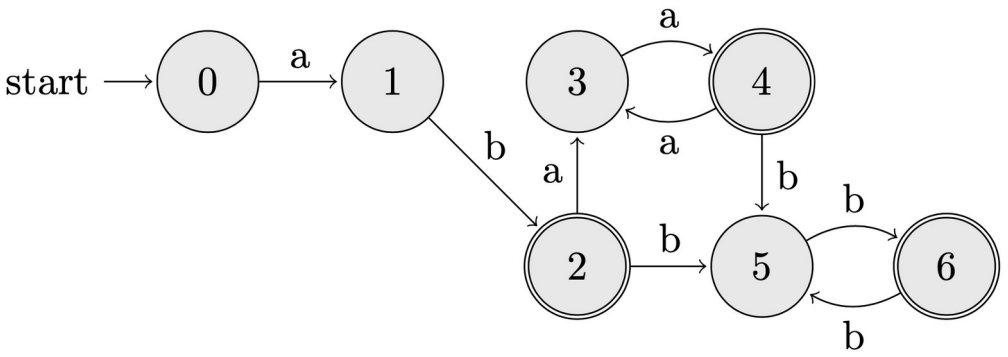
chain decomposition
p=2 chains

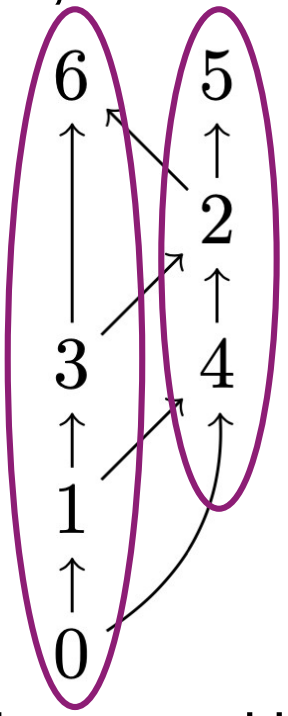
Wheeler automata [Gagie et al., 2017]

- succint representation, space:
  2(|V|+|E|) + |E| log|Σ|+ |Σ|log |E| + l.o.t.
- forward/backward traversal of edge in time O(log|Σ|)

Extension to arbitrary automata

[Cotumaccio, Prezza, 2021]





chain decomposition
p=2 chains

A *p*-sortable NFA with |V| states
via powerset construction

⇓

DFA A' with $|V^*| \leq 2^p(|V|-p+1)-1$ stati

Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

accepted with minor revisions

Merging succint representations:

given succint representations of collections A and B, obtain a succint representation of A∪B

Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

accepted with minor revisions

Merging succint representations:

given succint representations of collections A and B, obtain a succint representation of A∪B

expanding A and B to compute the union

Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

accepted with minor revisions

Merging succint representations:

given succint representations of collections A and B, obtain a succint representation of A∪B

Problems:

expanding A and B to compute the union

😐 requires too much space

Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

accepted with minor revisions

Merging succint representations:

given succint representations of collections A and B, obtain a succint representation of A∪B

Problems:

expanding A and B to compute the union

😐 requires too much space

☹ if the representation is lossy, it’s not possible at all!

Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

accepted with minor revisions

Merging succint representations:

given succint representations of collections A and B, obtain a succint representation of A∪B

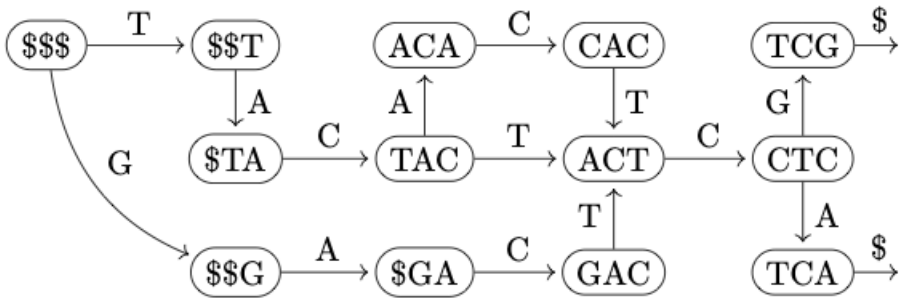
Do not expand! Merge directly! ☺

Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

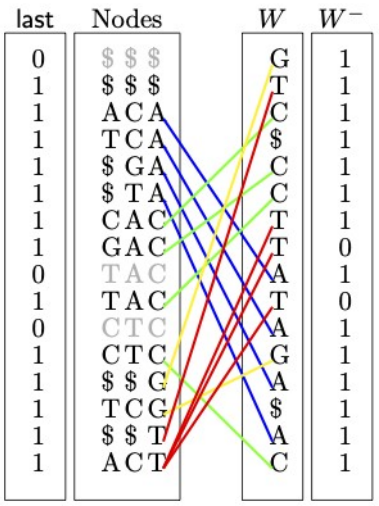
accepted with minor revisions

de Bruijn graph (k=3)



BOSS representation (lossy)

[Bowe et al. 2012]



merging:

- O(|E| k) time
- 4 |V| bits + O(|Σ|) space

state of the art:
[Muggli et al, 2017,2019]

- same time
- space: 2(|E|log|Σ|+|E|+|V|)+O(|Σ|)

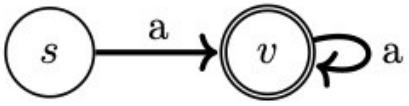
Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

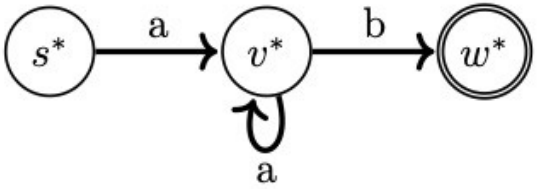
accepted with minor revisions

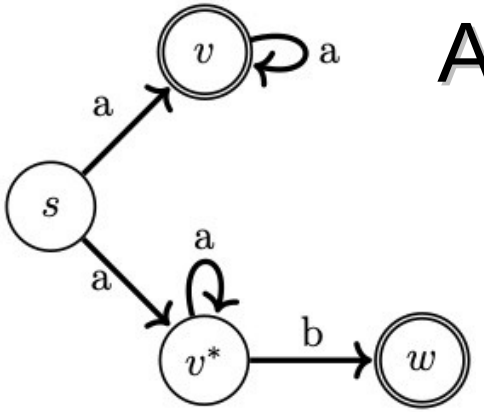
union of Wheeler automata

A_0



A_1





$A_0 \cup A_1$

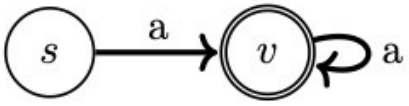
Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

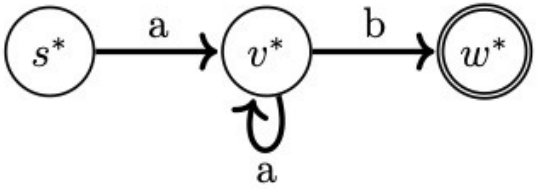
accepted with minor revisions

union of Wheeler automata

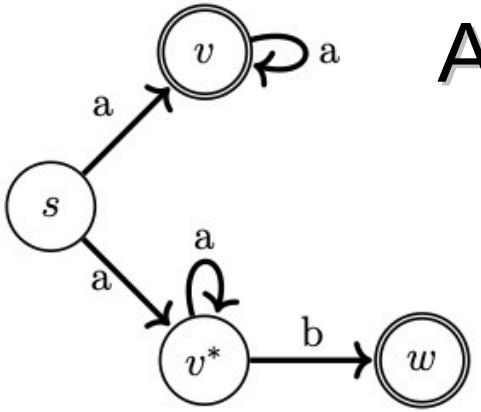
A_0



A_1



$A_0 \cup A_1$



Wheeler languages not closed for unions

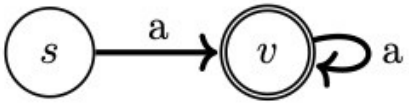
Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

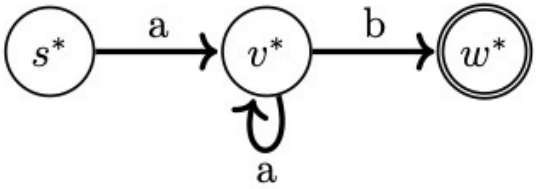
accepted with minor revisions

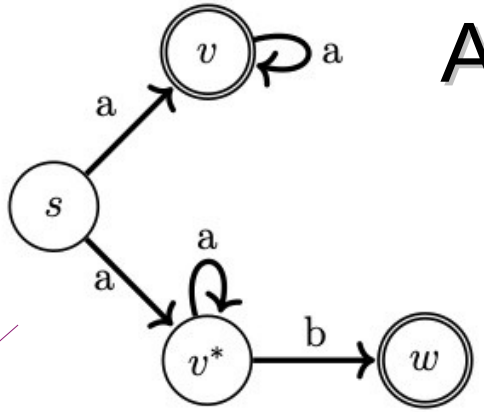
union of Wheeler automata

A_0



A_1





$A_0 \cup A_1$

union language Wheeler but union automaton not Wheeler:
$s < v, v^*$ and $v \neq v^*$

- $(s,v,a), (v^*,v^*,a) \Rightarrow v \leq v^*$
- $(s,v^*,a), (v,v,a) \Rightarrow v^* \leq v$

A is Wheeler iff it admits a total order of V s.t.

1) states with in-degree 0 are smallest
2) for (u,v,a),(u',v',a') transitions, if a<a' then v<v'
3) for (u,v,a),(u',v',a) transitions, if u<u' then $v \leq v'$

Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

accepted with minor revisions

- reduction to 2-SAT

in time $O(|E_0||E_1|)$

- find a Wheeler order of $A_0 \cup A_1$
- or report it doesn't exist

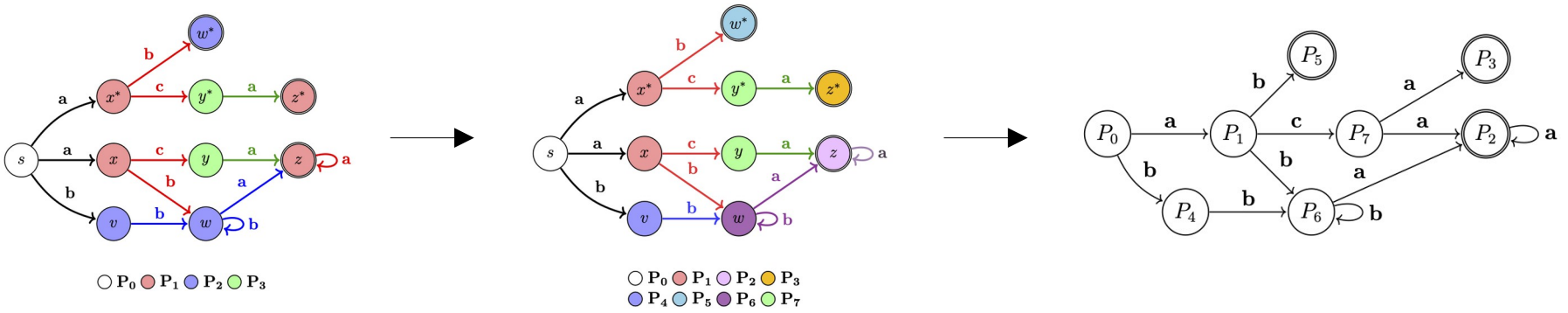
(uses more space than the succint representation)

Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

accepted with minor revisions

- refining algorithm: output a possibly smaller Wheeler automaton for the same language, or that no WO for the union automaton exists



Merging deBruijn graphs and Wheeler automata

[Egidi, Louza, Manzini, 2021]

accepted with minor revisions

- refining algorithm: output a possibly smaller Wheeler automaton for the same language, or that no WO for the union automaton exists

- time $O(|V|^2)$
- space $4|V|+o(|V|)$ bits

Conclusions

- Wheeler automata promising, also in view of recent work
- future work: extend our approach to generic automata through ideas in [Cotumaccio et al., 2021]