Indexing and compression: from Wheeler graphs to arbitrary graphs

Nicola Cotumaccio
nicola.cotumaccio@gssi.it

Nicola Prezza
nicola.prezza@unive.it

[1]GSSI, L'Aquila, Italy

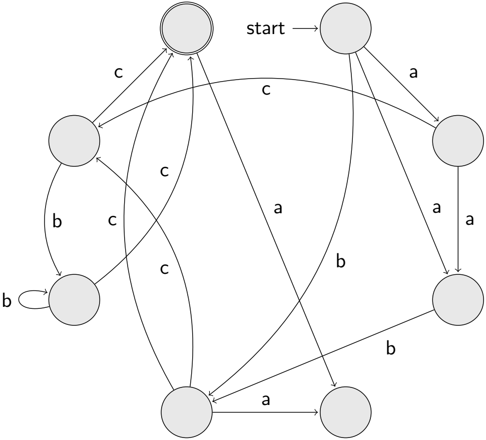
[2]Ca' Foscari University, Venice, Italy

GRAN SASSO SCIENCE INSTITUTE

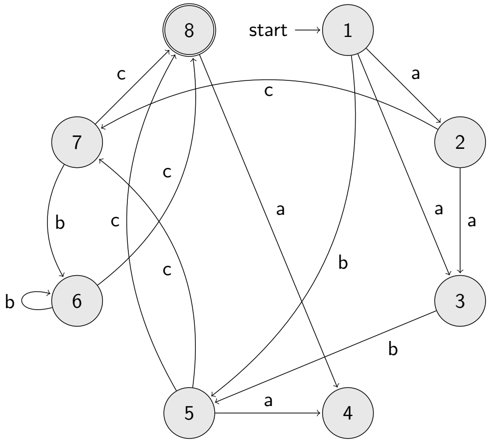
1. In 2017 a new class of automata - the class of Wheeler automata - was introduced [1].
2. Wheeler automata:
   1. can be compactly stored;
   2. allow to efficiently compute the set of states reachable from the initial state.
   3. capture most compression techniques based on the celebrated Burrows-Wheeler transform.

[1] Travis Gagie, Giovanni Manzini, Jouni Sirén, Wheeler graphs: A framework for BWT-based data structures, Theoretical Computer Science, Volume 698, 2017, Pages 67-78.

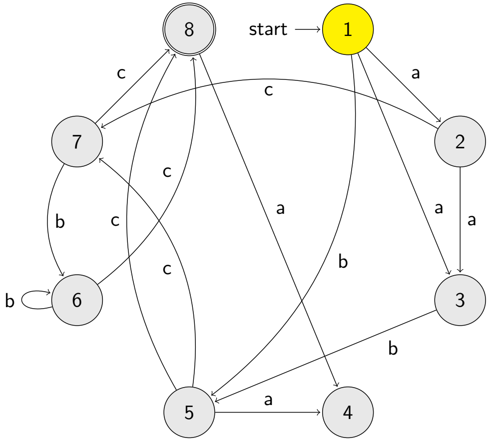
We start from an edge-labeled automaton



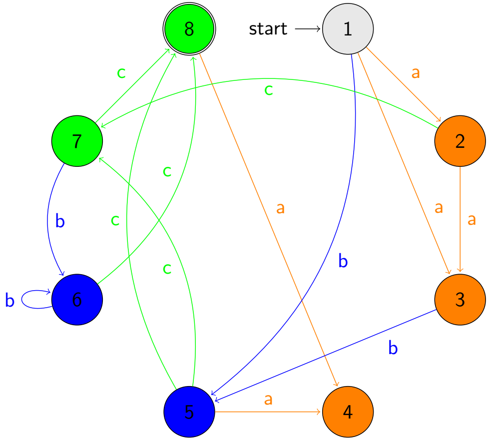
The automaton is Wheeler if there exists a total order on the set of states with the following properties:



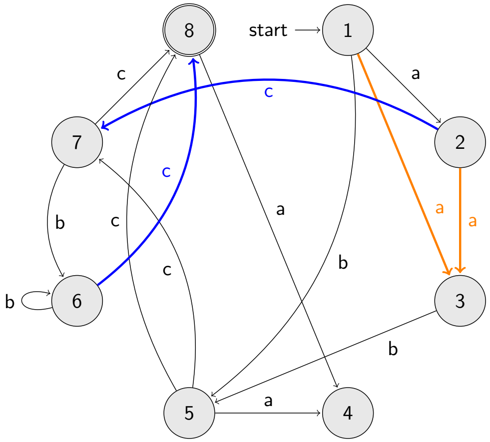
1) The initial state must come first.



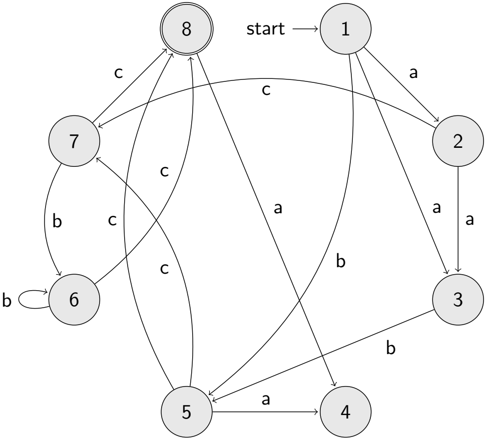
2) All states reached by edges labeled *a* come before all states reached by edges labeled *b*, which come before all states reached by edges labeled *c*, and so on.

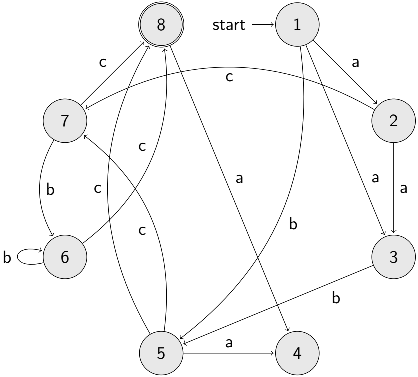


3) If we consider two edges with the same label, the mutual order of the start states is equal to the mutual order of the end states.

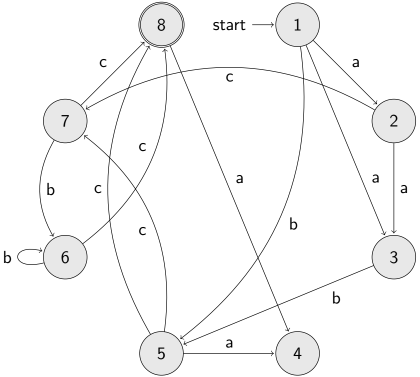


In a Wheeler automaton, the strings recognized by state i are co-lexicographically smaller than the strings recognized by state $i+1$, up to intersections.





- $I_1 = \{\epsilon\}$
- $I_2 = \{a\}$
- $I_3 = \{a, aa\}$
- $I_4 = \{ba, aba, aaba, bca, abca, aabca, \ldots, acbbca, bcbbca, abcbbca, aabcbbca, acbca, bcbca, abcbca, aabcbca, acca, bcca, abcca, aabcca\}$



- $I_5 = \{b, ab, aab\}$
- $I_6 = \{\ldots, acbb, bcbb, abcbb, aabcbb, acb, bcb, abcb, aabcb\}$
- $I_7 = \{ac, bc, abc, aabc\}$
- $I_8 = \{bc, abc, aabc, \ldots, acbbc, bcbbc, abcbbc, aabcbbc, acbc, bcbc, abcbc, aabcbc, acc, bcc, abcc, aabcc\}$

1. The main limitation of Wheeler automata is that they capture only a small subclass of regular languages.

1. The main limitation of Wheeler automata is that they capture only a small subclass of regular languages.
2. For example, unary languages are Wheeler if and only if they are finite or cofinite.

1. The main limitation of Wheeler automata is that they capture only a small subclass of regular languages.
2. For example, unary languages are Wheeler if and only if they are finite or cofinite.
3. In the paper, we show how to generalize Wheeler automata to arbitrary automata, and so to the whole class of regular languages.

1. The main limitation of Wheeler automata is that they capture only a small subclass of regular languages.
2. For example, unary languages are Wheeler if and only if they are finite or cofinite.
3. In the paper, we show how to generalize Wheeler automata to arbitrary automata, and so to the whole class of regular languages.
4. In the remaining of this presentation, we describe some enjoyable properties of Wheeler automata and we outline how they extend to generic automata.

Here is the standard definition of Wheeler order.

Here is the standard definition of Wheeler order.

Definition

Let $\mathcal{A} = (Q, E, \Sigma, s, F)$ be an NFA. A *Wheeler order* of $\mathcal{A}$ is a total order $\leq$ on Q that satisfies the following two axioms:

1. (Axiom 1) For every $u, v \in Q$, if $\lambda(u) \prec \lambda(v)$, then $u < v$ (in particular, states with no incoming edges come before all remaining states);
2. (Axiom 2) For all edges $(u', u), (v', v) \in E$, if $\lambda(u) = \lambda(v)$ and $u' < v'$, then $u \leq v$.

- In the above definition $\lambda(u)$ is the label of state u.

Here is the standard definition of Wheeler order.

Definition

Let $\mathcal{A} = (Q, E, \Sigma, s, F)$ be an NFA. A *Wheeler order* of $\mathcal{A}$ is a total order $\leq$ on Q that satisfies the following two axioms:

1. (Axiom 1) For every $u, v \in Q$, if $\lambda(u) \prec \lambda(v)$, then $u < v$ (in particular, states with no incoming edges come before all remaining states);
2. (Axiom 2) For all edges $(u', u), (v', v) \in E$, if $\lambda(u) = \lambda(v)$ and $u' < v'$, then $u \leq v$.

- In the above definition $\lambda(u)$ is the label of state u.

Only some automata admit a Wheeler order.

In our paper, we give the definition of co-lexicographic order.

In our paper, we give the definition of co-lexicographic order.

Definition

Let $\mathcal{A} = (Q, E, \Sigma, s, F)$ be an NFA. A ~~*Wheeler*~~ *co-lexicographic order* of $\mathcal{A}$ is a ~~total~~ partial order $\leq$ on Q that satisfies the following two axioms:

1. (Axiom 1) For every $u, v \in Q$, if $\lambda(u) \prec \lambda(v)$, then $u < v$ (in particular, states with no incoming edges come before all remaining states);
2. (Axiom 2) For all edges $(u', u), (v', v) \in E$, ~~if $\lambda(u) = \lambda(v)$ and $u' < v'$, then $u \leq v$~~ if $\lambda(u) = \lambda(v)$ and $u < v$, then $u' \leq v'$.

In our paper, we give the definition of co-lexicographic order.

Definition

Let $\mathcal{A} = (Q, E, \Sigma, s, F)$ be an NFA. A ~~*Wheeler*~~ *co-lexicographic order* of $\mathcal{A}$ is a ~~total~~ partial order $\leq$ on Q that satisfies the following two axioms:

1. (Axiom 1) For every $u, v \in Q$, if $\lambda(u) \prec \lambda(v)$, then $u < v$ (in particular, states with no incoming edges come before all remaining states);
2. (Axiom 2) For all edges $(u', u), (v', v) \in E$, ~~if $\lambda(u) = \lambda(v)$ and $u' < v'$, then $u \leq v$~~ if $\lambda(u) = \lambda(v)$ and $u < v$, then $u' \leq v'$.

Every automaton admits a co-lexicographic order!

Definition

Let $\mathcal{A} = (Q, E, \Sigma, s, F)$ be an NFA. A *co-lexicographic order* of $\mathcal{A}$ is a partial order $\leq$ on Q that satisfies the following two axioms:

1. (Axiom 1) For every $u, v \in Q$, if $\lambda(u) < \lambda(v)$, then $u < v$ (in particular, states with no incoming edges come before all remaining states);
2. (Axiom 2) For all edges $(u', u), (v', v) \in E$, if $\lambda(u) = \lambda(v)$ and $u < v$, then $u' \leq v'$.

Definition

Let $\mathcal{A} = (Q, E, \Sigma, s, F)$ be an NFA. A *co-lexicographic order* of $\mathcal{A}$ is a partial order $\leq$ on Q that satisfies the following two axioms:

1. (Axiom 1) For every $u, v \in Q$, if $\lambda(u) < \lambda(v)$, then $u < v$ (in particular, states with no incoming edges come before all remaining states);
2. (Axiom 2) For all edges $(u', u), (v', v) \in E$, if $\lambda(u) = \lambda(v)$ and $u < v$, then $u' \leq v'$.

- Now the order can be partial (but, as we will see, the more is "approximately total", the better).

Definition

Let $\mathcal{A} = (Q, E, \Sigma, s, F)$ be an NFA. A *co-lexicographic order* of $\mathcal{A}$ is a partial order $\leq$ on Q that satisfies the following two axioms:

1. (Axiom 1) For every $u, v \in Q$, if $\lambda(u) < \lambda(v)$, then $u < v$ (in particular, states with no incoming edges come before all remaining states);
2. (Axiom 2) For all edges $(u', u), (v', v) \in E$, if $\lambda(u) = \lambda(v)$ and $u < v$, then $u' \leq v'$.

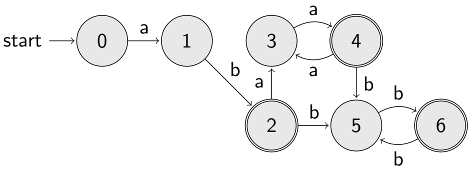
- Now the order can be partial (but, as we will see, the more is "approximately total", the better).
- In Axiom 2, we have just considered the contrapositive statement (which is not equivalent to the old statement if the order is not total).

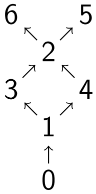
Definition

Let $\mathcal{A} = (Q, E, \Sigma, s, F)$ be an NFA. A *co-lexicographic order* of $\mathcal{A}$ is a partial order $\leq$ on Q that satisfies the following two axioms:

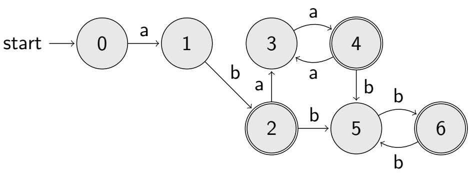
1. (Axiom 1) For every $u, v \in Q$, if $\lambda(u) < \lambda(v)$, then $u < v$ (in particular, states with no incoming edges come before all remaining states);
2. (Axiom 2) For all edges $(u', u), (v', v) \in E$, if $\lambda(u) = \lambda(v)$ and $u < v$, then $u' \leq v'$.

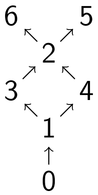
- Now the order can be partial (but, as we will see, the more is "approximately total", the better).
- In Axiom 2, we have just considered the contrapositive statement (which is not equivalent to the old statement if the order is not total).
- The intuition is that co-lexicographic order compare strings by comparing the last letter and possibly proceeding backward.





1. This automaton does not admit a Wheeler order (or equivalently, a total co-lexicographic order).





1. This automaton does not admit a Wheeler order (or equivalently, a total co-lexicographic order).
2. However, it can be shown that the partial order given by the Hasse diagram is a co-lexicographic order.

$$\begin{array}{ccccc} 6 & & & & 5 \\ & \nwarrow & & \nearrow & \\ & & 2 & & \\ & \nearrow & & \nwarrow & \\ 3 & & & & 4 \\ & \nwarrow & & \nearrow & \\ & & 1 & & \\ & & \uparrow & & \\ & & 0 & & \end{array}$$

1. This total order admits a chain partition of cardinality equal to two (for example $\{\{0, 1, 3, 6\}, \{4, 2, 5\}\}$).

$$
\begin{tikzcd}
6 & & 5 \\
& 2 \arrow[ul] \arrow[ur] & \\
3 \arrow[ur] & & 4 \arrow[ul] \\
& 1 \arrow[ul] \arrow[ur] & \\
& 0 \arrow[u] &
\end{tikzcd}
$$

1. This total order admits a chain partition of cardinality equal to two (for example $\{\{0, 1, 3, 6\}, \{4, 2, 5\}\}$).
2. By Dilworth theorem, two is also the cardinality of a largest antichain.

$$
\begin{tikzcd}
6 & & 5 \\
& 2 \arrow[ul] \arrow[ur] & \\
3 \arrow[ur] & & 4 \arrow[ul] \\
& 1 \arrow[ul] \arrow[ur] & \\
& 0 \arrow[u] &
\end{tikzcd}
$$

1. This total order admits a chain partition of cardinality equal to two (for example $\{\{0, 1, 3, 6\}, \{4, 2, 5\}\}$).
2. By Dilworth theorem, two is also the cardinality of a largest antichain.
3. As a consequence, a measure of the complexity of an automaton is the smallest p for which there exists a co-lexicographic order that admits a chain partition of cardinality p.

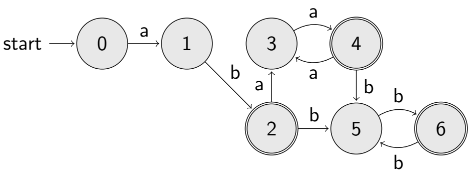
$$\begin{array}{ccccc}
6 & & & & 5 \\
& \nwarrow & & \nearrow & \\
& & 2 & & \\
& \nearrow & & \nwarrow & \\
3 & & & & 4 \\
& \nwarrow & & \nearrow & \\
& & 1 & & \\
& & \uparrow & & \\
& & 0 & &
\end{array}$$

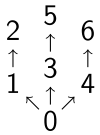
1. This total order admits a chain partition of cardinality equal to two (for example $\{\{0, 1, 3, 6\}, \{4, 2, 5\}\}$).
2. By Dilworth theorem, two is also the cardinality of a largest antichain.
3. As a consequence, a measure of the complexity of an automaton is the smallest p for which there exists a co-lexicographic order that admits a chain partition of cardinality p.
4. An automaton is Wheeler if and only if $p = 1$ (the order must be total).

1. Our compression and indexing results depends on p, and the lower p, the better (as one expects).

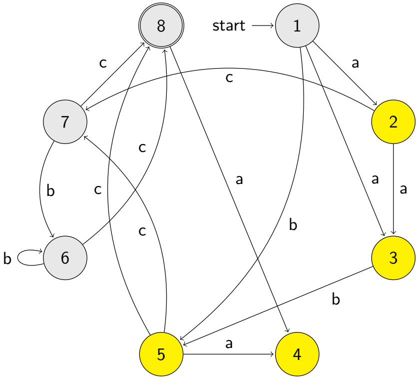
1. Our compression and indexing results depends on p, and the lower p, the better (as one expects).
2. Every automaton has its own p, because every automaton admits a trivial co-lexicographic order, namely:

$$\begin{aligned} \leq \; := \; & \{(u,u) \in Q \times Q \mid u \in Q\} \cup \\ & \cup \{(u,v) \in Q \times Q \mid \lambda(u) < \lambda(v)\}. \end{aligned}$$



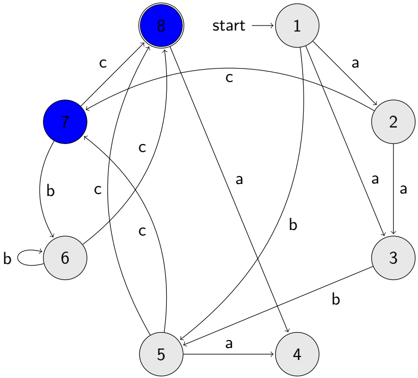


The key property of Wheeler automata is that if we start from consecutive states and we read any string, we end up in consecutive states.



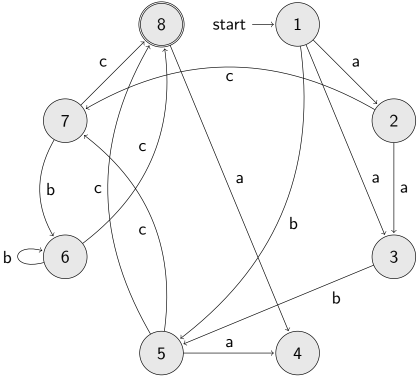
Start from the range 2-3-4-5.
Let us read the letter "c".
We have reached the range 7-8.

The key property of Wheeler automata is that if we start from consecutive states and we read any letter, we end up in consecutive states.

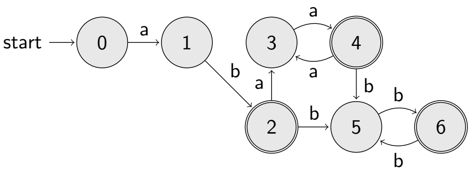


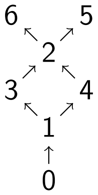
Start from the range 2-3-4-5.
Let us read the letter "c".
We have reached the range 7-8.

1. The key property of Wheeler automata is that if we start from consecutive states and we read any letter, we end up in consecutive states.
2. By induction the same works if we read any string.



In arbitrary automata, one finds out that everything works analogously, but we must keep track of p intervals.





Wheeler automata	Arbitrary automata
Compact representation of Wheeler automata	Compact representation of arbitrary automata
Burrows Wheeler transform of Wheeler automata	Burrows Wheeler transform of arbitrary automata
Return the states reached from initial state by reading $\alpha \in \Sigma^m$ in $O(m \log(\|\Sigma\|))$ time	Return the states reached from initial state by reading $\alpha \in \Sigma^m$ in $O(mp^2 \log(p\|\Sigma\|))$ time
Determining whether an NFA is Wheeler is NP-complete	Determining p for an NFA is NP-hard
Determining whether a DFA is Wheeler is linear	Determining p for a DFA is polynomial

Wheeler automata

The powerset construction
transform an NFA
with n states
into a DFA
with $\leq 2n - 1$ states

Arbitrary automata

The powerset construction
transform an NFA
with n states
into a DFA
with $\leq 2^p(n - p + 1) - 1$ states

Wheeler automata

The powerset construction transform an NFA with n states into a DFA with $\leq 2n - 1$ states

Arbitrary automata

The powerset construction transform an NFA with n states into a DFA with $\leq 2^p(n - p + 1) - 1$ states

1. The powerset construction turns out to be exponential in p, not in n.

Wheeler automata	Arbitrary automata
The powerset construction transform an NFA with n states into a DFA with $\leq 2n - 1$ states	The powerset construction transform an NFA with n states into a DFA with $\leq 2^p(n - p + 1) - 1$ states

1. The powerset construction turns out to be exponential in p, not in n.
2. Problems that are difficult on NFAs but easy on DFAs are fixed-parameter tractable with respect to p.

Wheeler automata

The powerset construction transform an NFA with n states into a DFA with $\leq 2n-1$ states

Arbitrary automata

The powerset construction transform an NFA with n states into a DFA with $\leq 2^p(n-p+1)-1$ states

1. The powerset construction turns out to be exponential in p, not in n.
2. Problems that are difficult on NFAs but easy on DFAs are fixed-parameter tractable with respect to p.
3. For example, one can check the equivalence between two NFAs by simply transforming them into DFAs and then checking the equivalence of the resulting DFAs. This yields an algorithm exponential in p (not in n) for a P-SPACE complete problem.

Future research:

1. Studying the hierarchy of regular languages induced by co-lexicographic orders. What role does p play?
2. Determining the relationship between intersection, union, ... of regular languages and p.
3. Extending more indexing techniques to arbitrary automata (tunneling...)
4. Describing more well-known problems being fixed-parameter tractable with respect to p.

Indexing and compression: from Wheeler graphs to arbitrary graphs

Nicola Cotumaccio
nicola.cotumaccio@gssi.it

Nicola Prezza
nicola.prezza@unive.it

[1]GSSI, L'Aquila, Italy

[2]Ca' Foscari University, Venice, Italy

GRAN SASSO SCIENCE INSTITUTE