

On Co-lexicographically Ordered Regular Languages

Giovanna D'Agostino, DMIF, University of Udine

reporting work in collaboration with:

Jarno Alanko,
Nicola Cotumaccio,
Davide Martincigh,
Alberto Policriti,
Nicola Prezza.



Explaining the title

Regular Languages

Co-lexicographic order on words



Explaining the title

Regular Languages

Co-lexicographic order on words

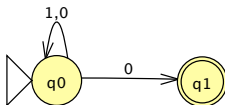
How and why we shall mix them



Finite Automata recognizing Languages

A language is a subset of finite strings on a finite alphabet Σ .

A regular language is a language for which there is a finite device (a finite automaton) accepting exactly the strings (words) belonging to the language.



a (non deterministic) automaton \mathcal{B} accepting the language

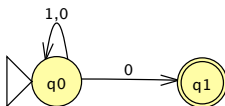
$$L(\mathcal{B}) = \{\alpha \in \Sigma^* : \alpha \text{ ends with } 0\}.$$



Finite Automata recognizing Languages

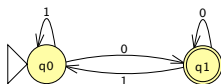
A language is a subset of finite strings on a finite alphabet Σ .

A regular language is a language for which there is a finite device (a finite automaton) accepting exactly the strings (words) belonging to the language.



a (non deterministic) automaton \mathcal{B} accepting the language

$$L(\mathcal{B}) = \{\alpha \in \Sigma^* : \alpha \text{ ends with } 0\}.$$



\mathcal{A} (deterministic)

$$L(\mathcal{A}) = L(\mathcal{B})$$



Co-lex Order on words

Words in Σ^* for a finite ordered alphabeth $a \prec b \prec c \dots$ can be ordered **lexicographically** as in a dictionary:

$$alma \prec_{\ell} ama \prec_{\ell} amare$$

or **co-lexicographically** by first reversing the words and then comparing them lexicographically (as in arabian dictionary?)

$$ama \prec_{c\ell} alma \prec_{c\ell} amare$$



Co-lex Order on words

Words in Σ^* for a finite ordered alphabeth $a \prec b \prec c \dots$ can be ordered **lexicographically** as in a dictionary:

$$alma \prec_{\ell} ama \prec_{\ell} amare$$

or **co-lexicographically** by first reversing the words and then comparing them lexicographically (as in arabian dictionary?)

$$ama \prec_{cl} alma \prec_{cl} amare$$

Using the co-lex order on words, we give priority to final letters of the words read by the automaton.



Automata and Orders have already met many times and attracted attention because of their relation with logical, combinatorial, and algebraic characterization of languages, e.g.:

Ordered automata and associated languages, H.-J. Shyr and G. Thierrin, Tamkang J. Math, 1974,

Partially Ordered Automata and Piecewise Testability, Tomás Masopust and Markus Krötzsch, Log. Methods Comput. Sci. 17:2, 2021,

Partially-Ordered Two-Way Automata: a New Characterization of DA, Thomas Schwentick and Denis Thérien and Heribert Vollmer, DLT 2001, Vienna, Austria.

...



A recent development:

Wheeler graphs: a framework for BWT-based data structures, Travis Gagie and Giovanni Manzini and Jouni Sirén, Theoretical Computer Science, (2017);

A simple and unified perspective on several algorithmic techniques related to suffix sorting, in particular, to a string transformation called the **Burrows-Wheeler transform (BWT)**.

The general idea is to enforce and exploit a total order on the states of an automaton, induced by the co-lex order of strings arriving in the states: this requires an a priori fixed order of its underlying alphabet which propagates through the automaton's transition relation.

The dependence on a fixed order of the alphabet marks the difference between this approach and the ones before.



The resulting automata are called

Wheeler Automata

Thanks to the order imposed on their states Wheeler automata admit efficient data structures for solving string matching on the automaton's paths.

In general, words read by an automaton and arriving in a state p can be messy.

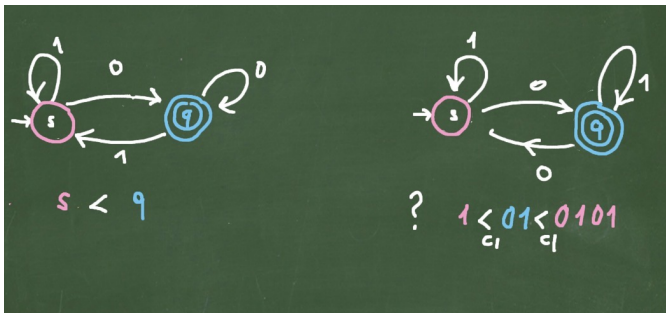
In Wheeler automata the words arriving in a given state form an interval in the co-lex order, and this fact allows for more efficient algorithms (e.g. to determine whether a word is accepted by the automaton).



Imposing a Co-lex Order on the Automaton' States: Deterministic Case

General idea: use the co-lex order of words arriving in the states to (partial) order the states of a DFA \mathcal{A} :

$$p <_{\mathcal{A}} q \Leftrightarrow \forall \alpha \rightsquigarrow p \forall \beta \rightsquigarrow q \quad \alpha \prec_{cl} \beta$$



In the first automaton the states are totally ordered;

In the second automaton the states are not totally ordered



Wheeler DFA

A **Wheeler DFA** \mathcal{A} is a DFA such that the partial order defined by

$$p <_{\mathcal{A}} q \Leftrightarrow \forall \alpha \rightsquigarrow p \forall \beta \rightsquigarrow q \quad \alpha \prec_d \beta$$

is a total order.

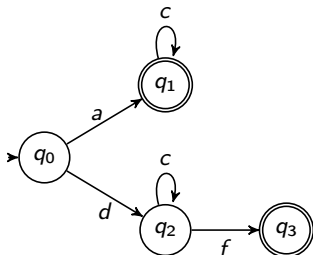


Wheeler DFA

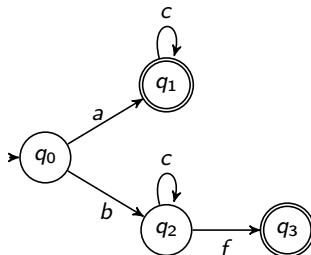
A **Wheeler DFA** \mathcal{A} is a DFA such that the partial order defined by

$$p <_{\mathcal{A}} q \Leftrightarrow \forall \alpha \rightsquigarrow p \forall \beta \rightsquigarrow q \quad \alpha \prec_{cl} \beta$$

is a total order.



(c) A Wheeler DFA \mathcal{A} with $q_0 <_{\mathcal{A}} q_1 <_{\mathcal{A}} q_2 <_{\mathcal{A}} q_3$



(d) A non Wheeler DFA \mathcal{B} .

in \mathcal{B} :

ac	\prec_{cl}	bc	\prec_{cl}	acc	\prec_{cl}	bcc	\dots
q_1		q_2		q_1		q_2	\dots



Wheeler Languages

Definition

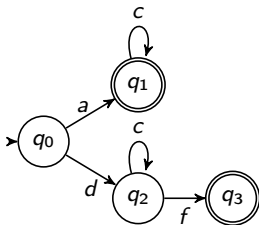
A regular language \mathcal{L} is a Wheeler Language if there exists a Wheeler DFA \mathcal{A} recognizing \mathcal{L}

Warning Every regular language is recognized by a special DFA (the minimum DFA for the language) with fewer states as possible.

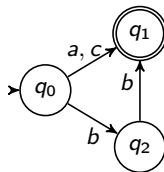
However, Wheeler order often conflicts with few states and it can be that the minimum DFA recognizing a Wheeler language is not Wheeler



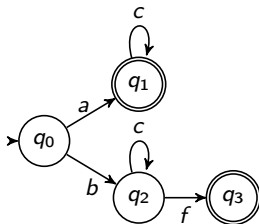
Wheeler/ non Wheeler DFA and Languages



(e) A Wheeler DFA \mathcal{A} recognizing a Wheeler Language.



(f) A non Wheeler DFA \mathcal{C} recognizing a (finite) Wheeler Language .



(g) A non Wheeler DFA \mathcal{B} recognizing a non Wheeler Language .



Complexity of Wheelerness I

What is the complexity of deciding whether a DFA is a Wheeler DFA?

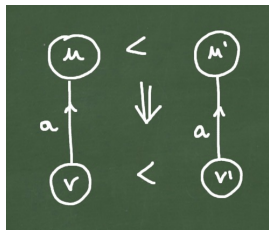


Global and Local Definition of a Wheeler Order

The $<_{\mathcal{A}}$ partial order over a DFA can be characterized as the maximum partial order $<$ such that the following holds:

- (1) If $u < u'$ then $\forall e \in \lambda(u) \forall e' \in \lambda(u') \ e \preceq e'$;
- (2) if $u < u'$ and $u = \delta(v, a)$, $u' = \delta(v', a)$ then $v \leq v'$

We call them the co-lex axioms. They tell us that if $u < u'$ then the local property (1) propagates backwards.



Is this characterization that allows to determine $<_{\mathcal{A}}$ in polynomial time over DFA's, and to check if it is total (i.e. to check Wheelerness over DFA's)



Complexity of Wheelerness II

What is the complexity of deciding whether a regular language is Wheeler?



Wheeler Languages and Complexity

Wheeler-ness of a regular language (described by a DFA) can be checked in polynomial time.

To see this we first give some characterizations of Wheeler languages in terms of monotone sequence in the co-lexicographical order.



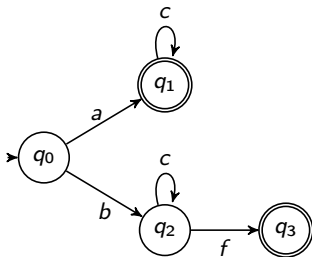
Monotone sequence in a Regular Language

Myhill-Nerode equivalence for a Regular language:

$$\alpha \equiv_{\mathcal{L}} \beta \iff (\forall \gamma \ \alpha\gamma \in \mathcal{L} \leftrightarrow \beta\gamma \in \mathcal{L})$$

Classes of the Myhill-Nerode equivalence correspond to states of the minimum automaton for \mathcal{L} .

In a Regular Language monotone sequences of words may jump between $\equiv_{\mathcal{L}}$ -classes indefinitely:



in \mathcal{B} :
 $ac \prec_{cl} bc \prec_{cl} acc \prec_{cl} bcc \dots$
 $q_1 \quad q_2 \quad q_1 \quad q_2 \quad \dots$

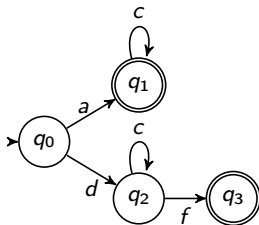


Monotone sequence in a Wheeler Language

Theorem

*A regular language is Wheeler iff the $\equiv_{\mathcal{L}}$ -classes **can be divided into a finite number of intervals** in the co-lex order.*

A regular language is Wheeler iff every co-lex monotone sequence eventually ends in only one $\equiv_{\mathcal{L}}$ -class (is eventually constant modulo $\equiv_{\mathcal{L}}$).



$$ac^* \prec dc^*$$



Deciding Wheelerness on Languages

Theorem

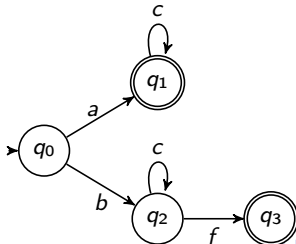
A regular language $\mathcal{L} = \mathcal{L}(\mathcal{A}_{min})$ is not Wheeler if and only if there exist strings μ, ν , and γ such that, in \mathcal{A}_{min}

- (1) μ and ν label paths from s to states q_1 and q_2 , respectively, with $q_1 \neq q_2$;
- (2) γ labels two cycles, one starting from q_1 and one starting from q_2 ;
- (3) $\mu, \nu \prec \gamma$ or $\gamma \prec \mu, \nu$.

Moreover, the length of the words μ, ν, γ satisfying the above properties can be bounded:

$$|\mu|, |\nu| < |\gamma| \leq 2 + |\mathcal{A}| + 2|\mathcal{A}|^2 + |\mathcal{A}|^3,$$

where $|\mathcal{A}|$ is the number of states of the automaton \mathcal{A} .



Non Determinism and Wheelerness

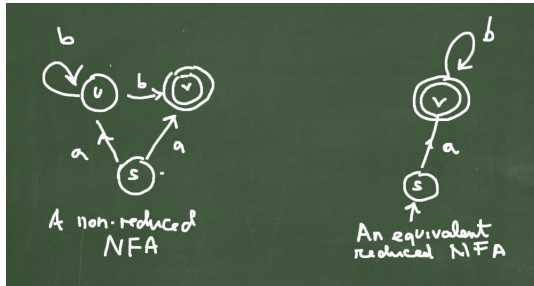
Up to now we considered only DFA's. What about **Wheeler NFA**?

Words in an NFA may end up in different states so that

$$p < q \Leftrightarrow \forall \alpha \rightsquigarrow p \ \forall \beta \rightsquigarrow q \ \alpha \prec_{cl} \beta$$

is too weak for them (intersecting states would always be incomparable).

For simplicity we consider reduced NFA: different states are reached by different sets of words.



A co-lex order over a reduced NFA \mathcal{N} is a partial order $<$ over the states satisfying:

(1) If $u < u'$ then $\forall e \in \lambda(u) \forall e' \in \lambda(u') \ e \preceq e'$;

(2) if $u < u'$ and $u \in \delta(v, a)$, $v \in \delta(v', a)$ then $v \leq v'$

As with DFA's, a reduced NFA \mathcal{A} always have a maximum co-lex order $<_{\mathcal{A}}$ that can be determined in polynomial time.

Definition

A reduced NFA is Wheeler if the maximum co-lex order $<_{\mathcal{A}}$ is a total order.

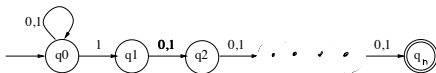
Do we get more languages using Wheeler NFA's?



Determinization

The powerset construction applied to an NFA returns an equivalent DFA with a possibly exponential blow-up of the states.

There are languages where the minimum DFA recognizing the language is exponentially bigger than a minimal state equivalent NFA



We can prove that the above NFA is not Wheeler...



Determinization of a Wheeler NFA

Theorem

If an NFA with n states is Wheeler then the powerset construction applied to it returns an equivalent Wheeler DFA with a number of states which is linearly related to n .

Proof The states of the powerset automaton correspond to sets

$I_\alpha = \{u \in Q : \alpha \rightsquigarrow u\}$, for words α .

In a Wheeler NFA \mathcal{A} the family

$$\{I_\alpha : \alpha \in \Sigma^*\}$$

is a prefix-suffix family of intervals in the order $<_{\mathcal{A}}$

(if $I \subseteq J$ then I is either a suffix or a prefix of J).

A prefix-suffix family can be linearly ordered and its cardinality is linearly related to n .

Corollary

The class of regular languages recognized by Wheeler NFA coincide with the class of regular languages recognized by a Wheeler DFA.



Wheeleriness: PRO

Wheeler Automata and languages have a nice "interval" structure allowing an efficient storage and pattern matching.

They admit a Myhill-Nerode Theorem (the "interval" variation of the usual Myhill-Nerode Theorem for Regular Languages).

Determinization is not exponential.

Difficult questions on regular languages such as universality, containment, etc become polynomial on Wheeler languages.



Wheeleriness: CONS

Wheeler languages are few. E.g. over a one letter alphabet, only finite and cofinite languages are Wheeler.

Wheeler Languages are strictly contained in the class of star-free languages, i.e. the class of languages which can be obtained from the finite languages using boolean operators (including complementation), and concatenation (but no Kleene star).

Contrary to Star Free Languages, Wheeler languages are not well behaved w.r.t. closure under operations: they are closed under intersection but not under negation (hence they are not closed under union) or concatenation.

Hence is not clear what a Wheeler regular expression should be.

Star Free Language have very nice algebraic and logic characterization (aperiodic monoids and $FO(<)$).

Similar characterizations for Wheeler Languages seems difficult to find.



Chi troppo vuole, nulla stringe!

The class of Wheeler Languages is computationally well behaved but small.

Can we ask less, maintaining the well behaviour?



Better Tidy or Messy?

The $<_{\mathcal{A}}$ partial order over a DFA is always defined.

Wheeler DFA happens to have a very tidy structure because $<_{\mathcal{A}}$ is total.

However, between a tidy guy and a messy guy there are lots of other possibilities.

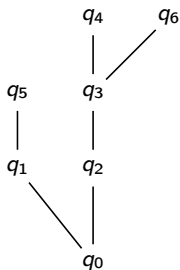
We can measure how far is a DFA from being Wheeler using the width of the partial order $<_{\mathcal{A}}$



Digression: the Width of a Partial Order

Given a partial order, its width is the minimum number of chains in which it can be partitioned.

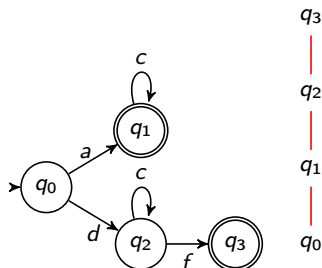
By Dilworth Theorem the width is also the maximum number of pairwise incomparable states.



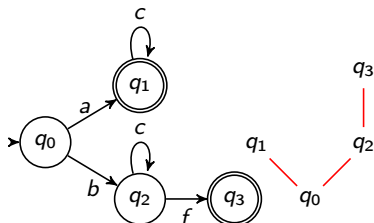
Hasse Diagram of a Partial order of width 3



The Width of a DFA



(h) A Wheeler DFA \mathcal{A} and its Hasse Diagram : width =1



(i) A non Wheeler DFA \mathcal{B} and its Hasse diagram: width=2

We expect that the less the width of the DFA the more it will have a Wheeler like behaviour.

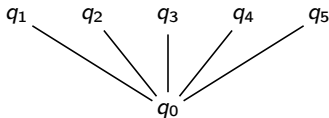
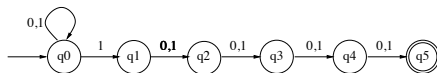


Width of Automata

Definition

The **width of a DFA** is the width of the partial order $<_{\mathcal{A}}$.

The **width of a (reduced) NFA** is the width of its maximum-colex order .



A Meaningful Hierarchy of Automata: Compression

Cotumaccio and Prezra proved that the better the width, the better we can compress the automaton (use less space than for general NFA to store the automaton).

Theorem

An NFA of width p over an alphabet of cardinality σ with m transitions can be stored using $O(m\log(p) + m\log(\sigma))$ -bits.

It is possible to prove that in general an NFA requires at least $m\log(n) + m\log(\sigma)$ -bits. Hence, if $p \ll n$ the automaton can be compressed (in such a way that pattern matching on the automaton paths can be done directly and fast without decompressing).



A Meaningful Hierarchy of Automata: Determinization

Cotumaccio and Prezza also showed that determinization is fixed parameter tractable with respect to the width of an NFA:

Theorem

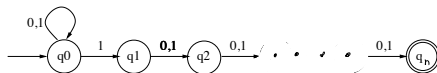
If \mathcal{D} is the DFA obtained from \mathcal{N} via the powerset construction and $\text{width}(\mathcal{N}) = p$ then:

$$|\mathcal{D}| \leq 2^p(n - p + 1) - 1$$

Hence, the important parameter for determinization is not states cardinality, but rather the width of the NFA.

E.g. There are very big Wheeler automata having linearly comparable determinizations.

On the other hand, the width of the following NFA is comparable to its cardinality; hence determinization produces an exponential blow-up in state cardinality.



Hierarchies of Regular Languages based on the Notion of Width

The **deterministic width of a regular language** is the minimum among the width of DFAs recognizing the language:

$$\text{width}^d(\mathcal{L}) = \min\{\text{width}(\mathcal{A}) : \mathcal{A} \text{ is a DFA with } \mathcal{L}(\mathcal{A}) = \mathcal{L}\}$$

The **non-deterministic width of a regular language** is the minimum among the width of NFAs recognizing the language:

$$\text{width}^{nd}(\mathcal{L}) = \min\{\text{width}(\mathcal{A}) : \mathcal{A} \text{ is an NFA with } \mathcal{L}(\mathcal{A}) = \mathcal{L}\}$$

Note: in general, $\text{width}^{nd}(\mathcal{L}) \leq \text{width}^d(\mathcal{L})$.

$$\mathcal{L} \text{ is a Wheeler Language} \Leftrightarrow \text{width}^d(\mathcal{L}) = 1 \Leftrightarrow \text{width}^{nd}(\mathcal{L}) = 1$$



Questions

The deterministic and non-deterministic hierarchies are proper, i.e. it is true that all levels are non empty?

The levels of the two hierarchies coincide (as in their first level)?

Given a Regular Language \mathcal{L} can we effectively determine its deterministic/non-deterministic width?

If yes, can we do it in polynomial time?



Two Non-Collapsing Hierarchies

The language $\mathcal{L}_n = \{a^{kn} : k \geq 0\}$ is recognized by a n -cycle DFA:

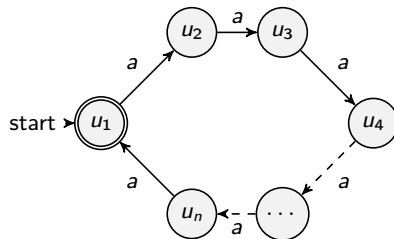


Figure: A DFA recognizing \mathcal{L}_n .

Hence $\text{width}^d(\mathcal{L}_n) \leq n$.

Moreover, one can prove that any NFA recognizing \mathcal{L}_n must contain a cycle bigger than n and all states along this cycle must be incomparable.

Hence $\text{width}^{nd}(\mathcal{L}_n) \geq n$ and

$$\text{width}^{nd}(\mathcal{L}_n) = \text{width}^d(\mathcal{L}_n) = n$$



Two Different Hierarchies

Let p_1, \dots, p_k be different primes and $\mathcal{L} = \{a^r : \exists i \ p_i | r\}$.

For this language it holds

$$\text{width}^d(\mathcal{L}) \geq \prod_{i=1}^k p_i$$

$$\text{width}^{nd}(\mathcal{L}) \leq \sum_{i=1}^k p_i.$$

Hence, here are languages with

$$\text{width}^{nd}(\mathcal{L}) < \text{width}^d(\mathcal{L})$$

Theorem

If \mathcal{L} is a regular language then

$$\text{width}^{nd}(\mathcal{L}) \leq \text{width}^d(\mathcal{L}) \leq 2^{\text{width}^{nd}(\mathcal{L})} - 1$$



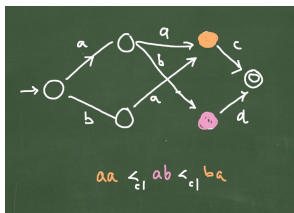
Calculating the Deterministic Width of a Regular Language

$$\text{width}^d(\mathcal{L}) = \min\{\text{width}(\mathcal{A}) : \mathcal{A} \text{ is a DFA with } \mathcal{L}(\mathcal{A}) = \mathcal{L}\}$$

Suppose we have $\mathcal{L} = \mathcal{L}(\mathcal{A})$ for a DFA \mathcal{A} .

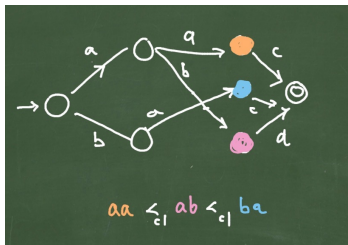
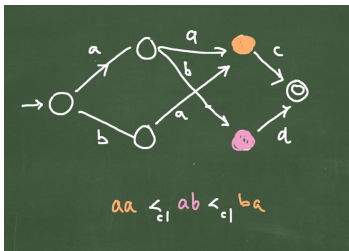
We know how to calculate $k = \text{width}(\mathcal{A})$ in polynomial time, but k is only an upper bound on the width of the language.

To check the width of all automata having state cardinality as \mathcal{A} or smaller, will not be enough: the following (minimum) DFA \mathcal{A}_{\min} recognize a finite language, hence $\text{width}(\mathcal{L}(\mathcal{A}_{\min})) = 1$ but $\text{width}(\mathcal{A}_{\min}) > 1$:



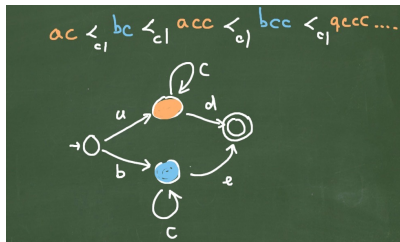
Two contrasting Objectives

There is a conflict between minimum width and minimum state cardinality. In order to achieve the width of the language it is sometime necessary to create new states:



Two contrasting Objectives

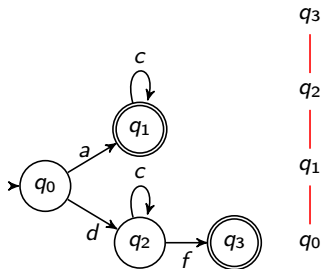
However, there are non solvable incomparabilities:



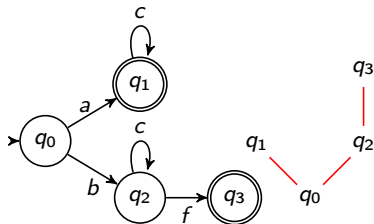
The entanglement number of a DFA

States q_1, \dots, q_k are entangled in a DFA \mathcal{A} if there is a monotone sequence of words arriving infinitely many times in each state.

The entanglement number $\text{ent}(\mathcal{A})$ of a DFA \mathcal{A} is the maximum number k for which there are k -entangled states q_1, \dots, q_k .



(a) A Wheeler DFA \mathcal{A} with $\text{ent}(\mathcal{A}) = 1$



(b) A non Wheeler DFA \mathcal{B} with $\text{ent}(\mathcal{B}) = 2$



Entangled states are always incomparable in any DFA. Hence they contribute to the width of the DFA.

k entangled states in the minimum automaton $\mathcal{A}_{\mathcal{L}}$ of a regular language \mathcal{L} generate k entangled states in any equivalent DFA. Hence

$$\text{ent}(\mathcal{A}_{\mathcal{L}}) \geq \text{width}^d(\mathcal{L})$$

Theorem

$$\text{ent}(\mathcal{A}_{\mathcal{L}}) = \text{width}^d(\mathcal{L})$$

Proof.

Starting from $\text{ent}(\mathcal{A}_{\mathcal{L}})$ and chopping states when they are only finitely interleaved we obtain a new equivalent DFA \mathcal{H} with

$$\text{ent}(\mathcal{A}_{\mathcal{L}}) = \text{ent}(\mathcal{H}) = \text{width}(\mathcal{H}) \leq \text{width}^d(\mathcal{L}) \leq \text{ent}(\mathcal{A}_{\mathcal{L}})$$



Open Problems and Generalizations

Can we calculate effectively the non-deterministic width of a language?






Rational Expression for Wheeler Language? Logical Characterization?
Algebraic Characterization?

Other kind of automata? Alternating finite automata?

Languages of infinite words?



References

-  T. Gagie, G. Manzini and J. Sirén Wheeler graphs: a framework for BWT-based data structures, Theoretical Computer Science, 698: 67 - 78, 2017.
-  N. Cotumaccio, N. Prezza, On Indexing and Compressing Finite Automata, SODA 2021, pp 2585-2599.
-  J. Alanko, G. D'Agostino, A. Policriti and N. Prezza, Wheeler languages, Information and Computation 281, 2021
-  N. Cotumaccio, G. D'Agostino, A. Policriti and N. Prezza, Co-lexicographically ordering automata and regular languages. Part I. CoRR abs/2208.04931 (2022) (submitted)
-  G.D'Agostino, D. Martincigh and A. Policriti, Ordering Regular Languages: a Danger Zone, Proceedings of the 22nd Italian Conference on Theoretical Computer Science, Bologna, Italy, September 13-15, 2021,

