Lower and Upper Bound Results for Hard Problems Related to Finite Automata

Henning Fernau
Universität Trier, Germany
fernau@informatik.uni-trier.de

Andreas Krebs
Universität Tübingen, Germany
krebs@informatik.uni-tuebingen.de

Berkeley, November 2015

Overview

1. Three classical problems on finite automata
2. More problems on finite automata
3. Jumping finite automata
4. Boustrophedon finite automata
5. Conclusions

ETH and SETH A simplified view

Exponential Time Hypothesis: 3-SAT instances (with n variables and m clauses) cannot be solved in time $O^*(2^{o(n)})$.
Sparsification Lemma: If ETH holds, then 3-SAT instances cannot be solved in time $O^*(2^{o(n+m)})$.

Strong Exponential Time Hypothesis: SAT instances on n variables cannot be solved in time $O^*((2-\varepsilon)^n)$ for any $\varepsilon > 0$.

Known: SETH implies ETH
ETH implies: FPT $\neq$ W[1]
There is a 1-1 correspondence between SUBEXP vs. EXP and FPT vs. XP

Problems on Finite Automata

non-universality

Given an automaton A with input alphabet Σ, is $L(A) \neq \Sigma^*$?

inequivalence

Given two automata A_1, A_2, is $L(A_1) \neq L(A_2)$?

intersection non-emptiness

Given k automata $A_1, \ldots, A_k$, is $\bigcap_{i=1}^{k} L(A_i) \neq \emptyset$?

Classical Status

	DFA	NFA
non-universality	poly-time	PSPACE-complete
inequivalence	poly-time	PSPACE-complete
intersection $\neq \emptyset$	PSPACE-complete	PSPACE-complete

$\rightsquigarrow$ Focus on **N**FAs.

Classical Status for NFA-problems

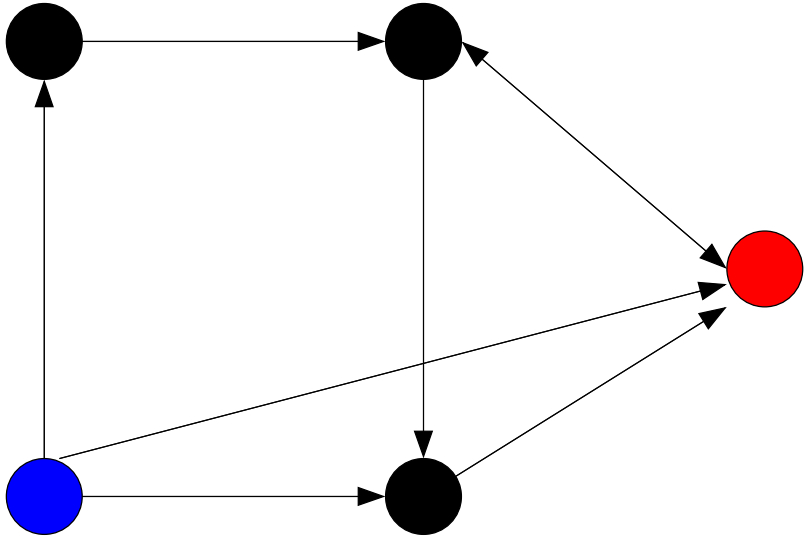
NFA	unary	binary
non-universality	NP-complete	PSPACE-complete
inequivalence	NP-complete	PSPACE-complete
intersection $\neq \emptyset$	NP-complete	PSPACE-complete

Even unary input alphabets are interesting.

intersection $\neq \emptyset$ is then NP-complete also for DFAs.

Previously unknown: Complexity status under ETH $\rightsquigarrow$ main topic of the talk

Tally NFAs (unary input alphabets)
nothing else than directed graphs (edge labels not interesting)
UNIVERSALITY hence models the folllowing scenary:
Can somebody living in the blue circle visit the red one in **any** number of steps?



This question is trivial for undirected graphs. . .

Non-universality for Tally NFAs: NP-hardness by Stockmeyer & Meyer 1973

Reduction from 3-SAT ☺ (n variables, m clauses)

Idea: Codify assignments by Chinese remainder.

Take the first n primes $p_1, \ldots, p_n$; Ex.: $p_1 = 2$, $p_2 = 3$, $p_3 = 5$.

a^z encodes assignment α if $z \equiv \alpha(x_i) \bmod p_i$;

Ex.: $\alpha(x_1) = 0$, $\alpha(x_2) = 1$, $\alpha(x_3) = 1 \rightsquigarrow z = 16$.

Recall: $p_n \sim n \ln n$.

$\exists$ NFA A_0 for $L_0 := \bigcup_{k=2}^{n} \bigcup_{j=2}^{p_k - 1} \{a\}^j \{a^{p_k}\}^*$ with $\leq np_n \sim n^2 \ln n$ many states.

L_0 collects words that do not encode assignments. Ex.: $aa \in L_0$.

$L_j := \{a^{z_{k_j}}\} \cdot \{a^{p_{i_j(1)} \cdots p_{i_j(|c_j|)}}\}^*$ with $0 \leq z_{k_j} < p_{i_j(1)} \cdots p_{i_j(|c_j|)}$ is uniquely determined by $z_{k_j} \equiv \alpha(x_r) \bmod p_{i_j(r)}$ for $r = 1, \ldots, |c(j)|$ s.t. α falsifies c_j.

$i_j(\ell)$ is the index of the ℓth variable in clause c_j.

As $p_{i_j(1)} \cdots p_{i_j(|c_j|)} \leq p_n^3$ (3-SAT), L_j is accepted by a DFA with $\leq p_n^3$ states.

Altogether, the NFA has at most mp_n^3 states: quite a many! ☹

Non-universality for Tally NFAs: NP-hardness (Stockmeyer & Meyer 1973)

Cor.: Unless ETH fails, for any $\epsilon > 0$, there is no $O^*(2^{o(q^{1/4-\epsilon})})$-time algorithm for deciding, given a tally NFA A on q states, whether $L(A) = \{a\}^*$.

The algorithmic side:
Textbook algorithm (conversion into DFAs) yields $O^*(2^q)$-time algorithm.
Chrobak 1986 (conversion into DFAs): improvement to $O^*(2^{\Theta(\sqrt{q \log q})})$.

Can we bring these bounds together?

Let us improve on the lower bound.

An auxiliary result yields an improved lower bound

Thm.: Unless ETH fails, there is no $O^*(2^{o(m)})$-time algorithm for deciding if a given m-edge graph has a (proper) 3-coloring.

See Fedor's talk . . . / W.l.o.g., $m = O(n)$.

Now, mimic proof of Stockmeyer / Meyer, reducing from 3-COLORING.
variables $\approx$ vertices / clauses $\approx$ edges
(a) Coloring condition has two vertices, not three variables $\rightsquigarrow$ improvement
(b) Moreover, more efficient encoding $\rightsquigarrow$ no ϵ-term

Thm.: Unless ETH fails, there is no $O^*(2^{o(q^{1/3})})$-time algorithm for deciding, given a tally NFA A on q states, whether $L(A) = \{a\}^*$.

Open: Match LB $O^*(2^{o(\sqrt[3]{q})})$ with UB $O^*(2^{\Theta(\sqrt{q \log q})})$.

Inequivalence for Tally NFAs

Cor.: Unless ETH fails, there is no $O^*(2^{o(q^{1/3})})$-time algorithm for deciding, given two tally NFAs A_1, A_2, each on q states, whether $L(A_1) = L(A_2)$.

The algorithmic side:
Convert both NFAs into DFAs (Chrobak 1986);
then use complementation and emptiness tests.
Complexity: $O^*(2^{\Theta(\sqrt{q \log q})})$.

Open: Match LB $O^*(2^{o(\sqrt[3]{q})})$ with UB $O^*(2^{\Theta(\sqrt{q \log q})})$.

Same results for TALLY NFA NON-INCLUSION.

Intersection Non-emptiness for Tally FAs

Thm.: There is no algorithm that, given k tally DFAs (or NFAs) $A_1, \ldots, A_k$, each with at most q states, decides if $\bigcap_{i=1}^{k} L(A_i) \neq \emptyset$ in time $O^*(2^{o(\min(k,q^{1/2}))})$ unless ETH fails.

Revisit previous constructions:
$k \approx n + m$, $q \approx n^2$.

The algorithmic side: UB $O^*(q^k)$ is terribly far off. ☹

Non-Tally FAs with bounded alphabets

Thm.: Assuming ETH, there is no algorithm for solving UNIVERSALITY for q-state NFAs with binary input alphabets that runs in time $O(2^{o(q)})$.

Idea: Reduction from 3-COLORING.

This matches the DFA construction. ☺

(Basically) the same (matching) result for INEQUIVALENCE for binary NFAs.

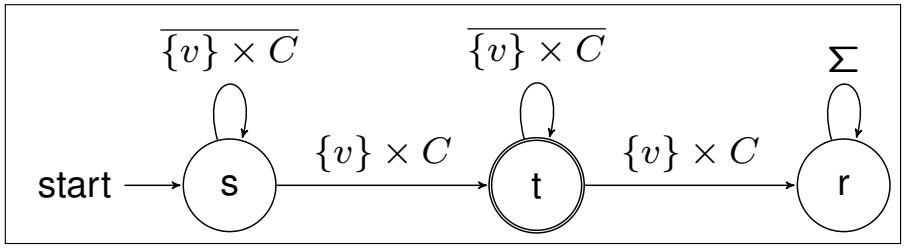
For INTERSECTION NONEMPTINESS of DFAs, we get a weaker result:
Cor.: There is no algorithm that, given k DFAs $A_1, \ldots, A_k$ with binary input alphabet, each with at most q states, decides in time $O^*(2^{o(\min\{k,2^q\})})$ if $\bigcap_{i=1}^{k} L(A_i) \neq \emptyset$ unless ETH fails.

Non-Tally FAs with unbounded alphabets

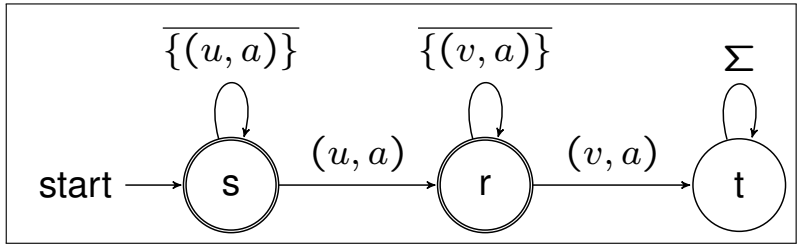
Thm.: There is no algorithm that, given k DFAs $A_1, \ldots, A_k$ with unbounded input alphabet, each with at most 3 states, decides in time $O^*(2^{o(k)})$ if $\bigcap_{i=1}^{k} L(A_i) \neq \emptyset$ unless ETH fails.
This matches the product automaton upper bound $O^*(3^k)$. ☺

Reduction from 3-COLORING:
Choose alphabet $\Sigma = V \times C$, $C = \{1, 2, 3\}$.

For all vertices v, A_v:



For all edges uv, $A_{uv,a}$:



Summary of the Classical Problems for q-State NFAs in $O^*(2^{f(\cdot)})$-estimates

input alphabet	UNIVERSALITY		EQUIVALENCE		k-NFA-INTERSECTION		Done?
	Lower	Upper	Lower	Upper	Lower	Upper	
unary	$o(\sqrt[3]{q})$	$\Theta(\sqrt{q \log q})$	$o(\sqrt[3]{q})$	$\Theta(\sqrt{q \log q})$	$o(\min(k, \sqrt{q}))$	$k \log q$	No
binary	$o(q)$	q	$o(q)$	q	$o(\min(k, 2^q))$	$k \log q$	No
unbounded	$o(q)$	q	$o(q)$	q	$o(k)$	$k \log q$	Yes

Overview

1. Three classical problems on finite automata
2. More problems on finite automata
3. Jumping finite automata
4. Boustrophedon finite automata
5. Conclusions

Aperiodicity

A regular language is *aperiodic* if it can be expressed, starting from finite sets, with the Boolean operations and with concatenation.

Known: A language accepted by some minimum-state DFA A is not aperiodic iff there is an input word u (star witness) and some state p such that $\delta^*(p, u) \neq p$, but for some $r > 1$, $\delta^*(p, u^r) = p$.

A reduction due to J. Stern (1985) shows:

Cor.: Assuming ETH, there is no algorithm for solving APERIODICITY for q-state DFAs on unbounded input alphabets that runs in time $O(2^{o(q)})$.
Slightly weaker for binary input alphabets.

The mentioned characterization of aperiodicity shows:

Propos.: APERIODICITY can be tested in time $O^*(q^q) = O^*(2^{q \log q})$ for q-state DFAs on unbounded input alphabets.

Still a small gap!

Synchronizing words

Given a deterministic finite semi-automaton, i.e., for each $a \in \Sigma$, a mappting $\mu_a : Q \to Q$, a set $Q_{sync} \subseteq Q$, a Q_{sync}-*synchronizing word* $w \in \Sigma^*$ enjoys

$$\forall q, q' \in Q_{sync} : \mu_w(q) = \mu_w(q') \,.$$

Q_{sync}-SW: Is there a Q_{sync}-*synchronizing word*? PSPACE-complete.
$Q_{sync} = Q$: Related to Černý's Conjecture.
Multi-parameter analysis in F., Heggernes, Villanger, JCSS 2015
Q-SW "only" NP-complete (with length bound on synchr. word).

Thm.: There is an algorithm for solving Q_{sync}-SW on unbounded input alphabets that runs in time $O^*(2^q)$ for q-state deterministic finite semi-automata. Conversely, assuming ETH, there is no $O^*(2^{o(q)})$-time algorithm for this task, even on **bounded** input alphabets.

Consequences from previous SETH hardness results on SW

Known: No $O^*((|\Sigma| - \varepsilon)^\ell)$-time algorithm solving SW (for any $\varepsilon > 0$) unless SETH fails.

Consequences:
There is a straightforward algorithm with running time $O^*(|\Sigma|^\ell)$ that, given k DFAs over the input alphabet Σ and an integer ℓ, decides whether or not there is a word $w \in \Sigma^{\leq \ell}$ accepted by all these DFAs.
Conversely, there is no algorithm that solves this problem in time $O((|\Sigma| - \varepsilon)^\ell)$ for any $\varepsilon > 0$ unless SETH fails.

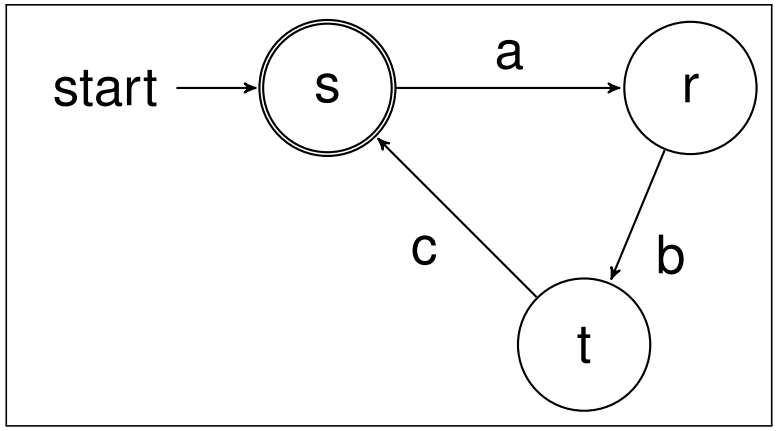
Similarly for UNIVERSALITY, EQUIVALENCE

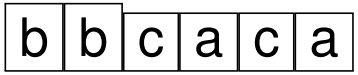
Overview

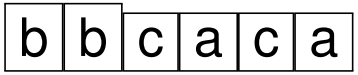
1. Three classical problems on finite automata
2. More problems on finite automata
3. Jumping finite automata
4. Boustrophedon finite automata
5. Conclusions

Jumping Finite Automata (JFAs)

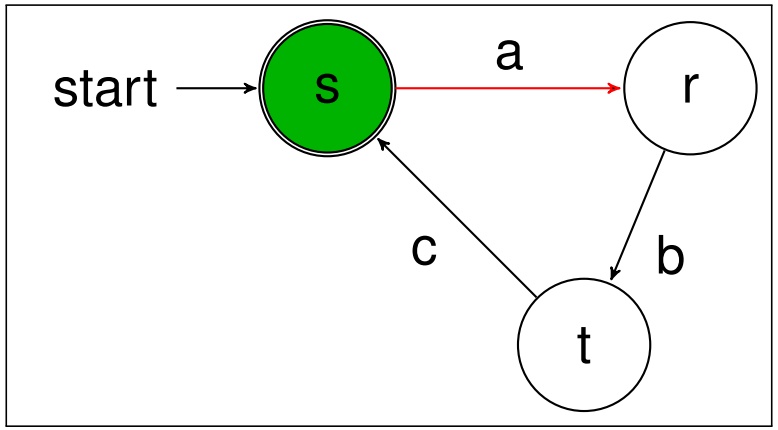
Meduna, Zemek IJFCS 2012; F., Paramasivan, Schmid CIAA 2015

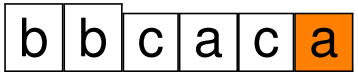




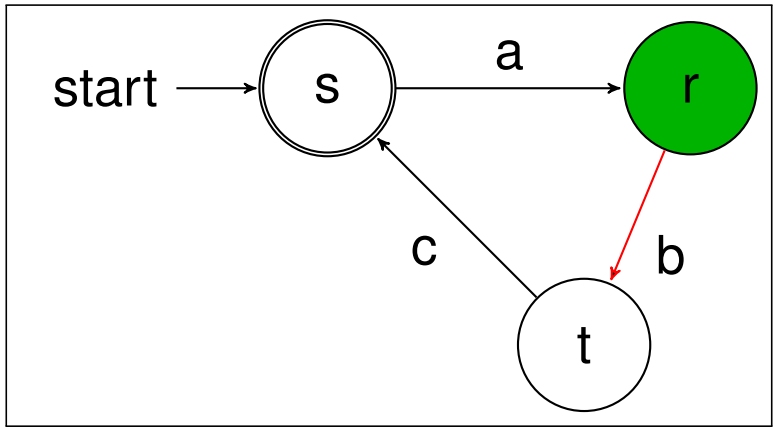


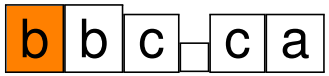
Jumping Finite Automata

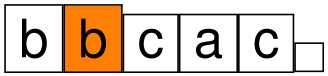




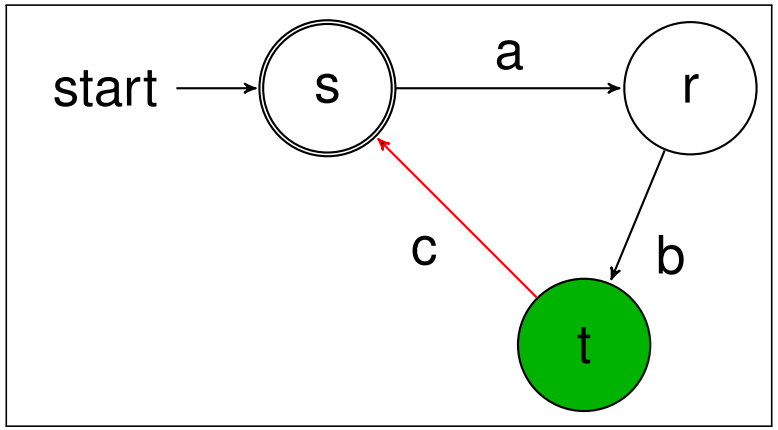
Jumping Finite Automata



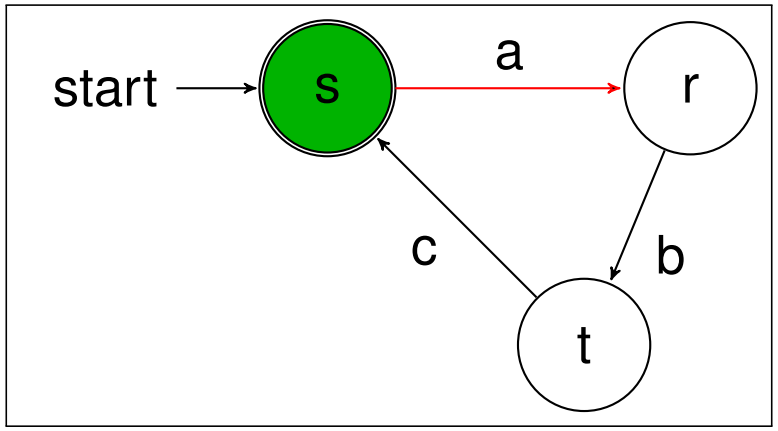




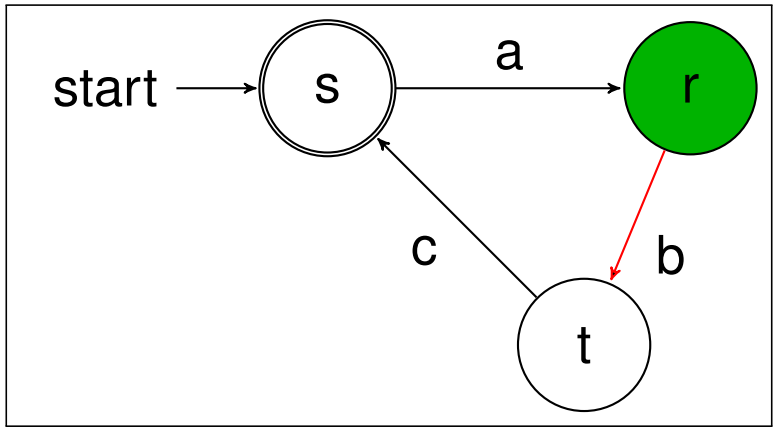
Jumping Finite Automata



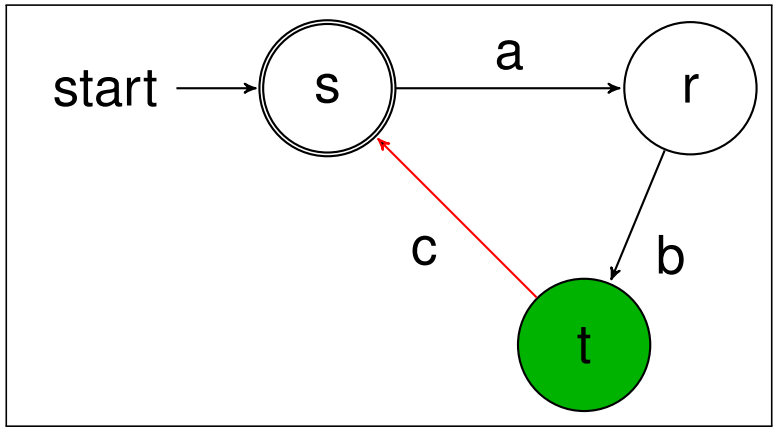
Jumping Finite Automata



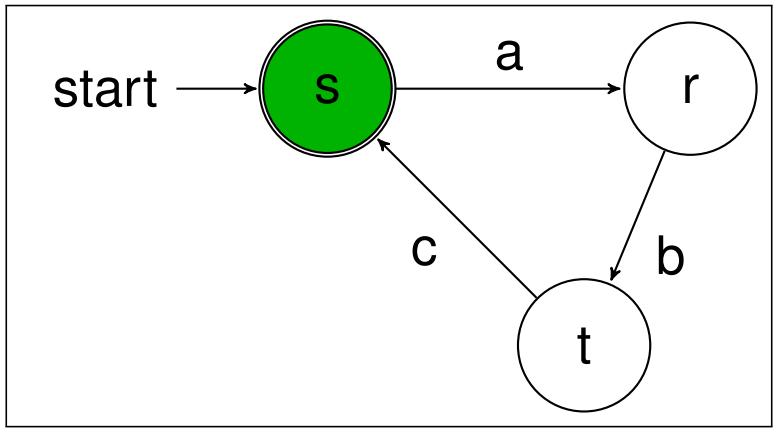
Jumping Finite Automata



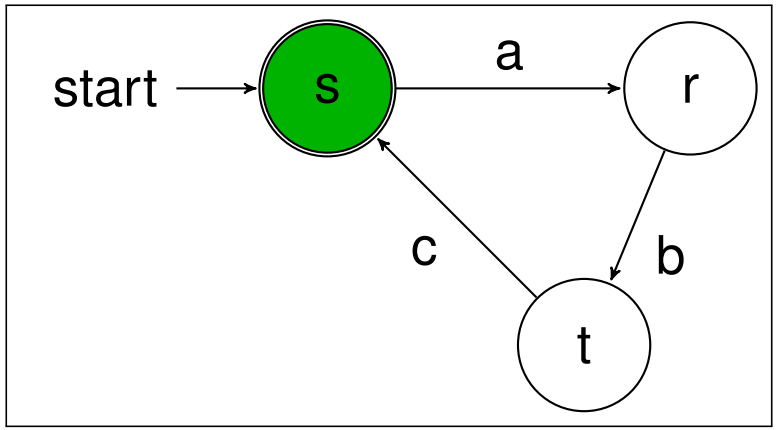
Jumping Finite Automata



Jumping Finite Automata



Jumping Finite Automata



Accepted language: $\{w \in \{a, b, c\}^* : |w|_a = |w|_b = |w|_c\}$

Hard questions for JFAs

All hardness results for tally NFAs transfer.

New specific hard questions:

UNIVERSAL MEMBERSHIP: NP-hardness reduction by V. Vorel as a starting point.

Different reductions by Mayer, Stockmeyer Inf. & Comput. 1994; F., Paramasivan, Schmid 2015

Thm.: Under ETH, there is no algorithm solving 1-in-3-SAT in time $O^*(2^{o(n)})$ or $O^*(2^{o(m)})$ on CNF formulae with n variables and m clauses.

Cor.: Under ETH, there is no algorithm solving UNIVERSAL MEMBERSHIP for JFAs in time $O^*(2^{o(n)})$ or $O^*(2^{o(m)})$ on automata with at most n states and input words of length at most n, with input alphabet bounded by $m = |\Sigma|$.

Upper bound: $O^*(n!)$ (poly-space); alternatively DP yields $O^*(2^n)$.

Another interesting (hard) problem:

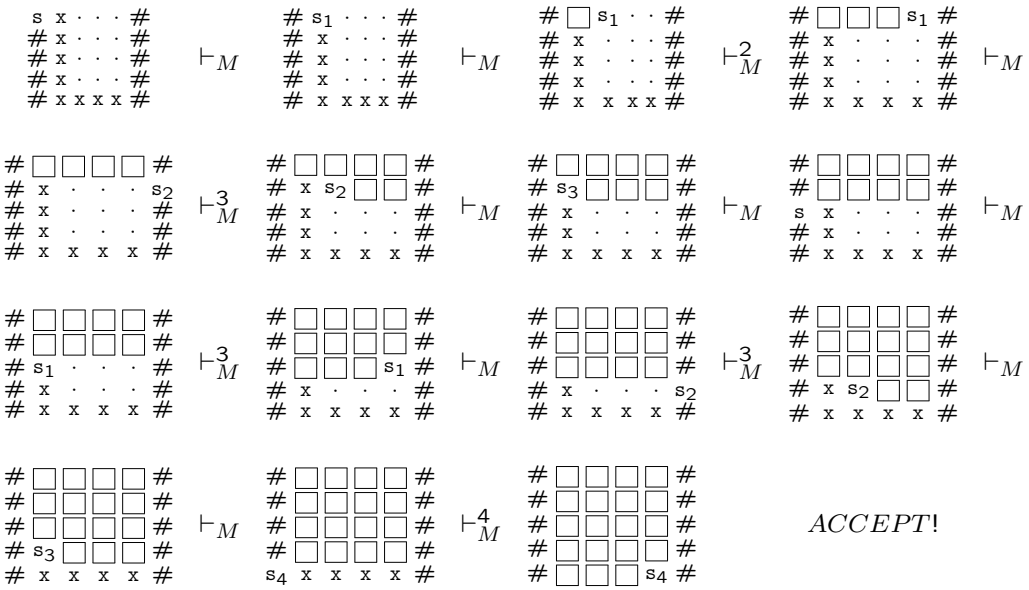
2-JFA-INTERSECTION-NONEMPTINESS (unbounded alphabets)

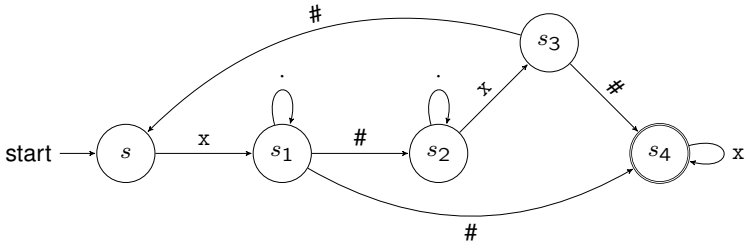
Overview

1. Three classical problems on finite automata
2. More problems on finite automata
3. Jumping finite automata
4. Boustrophedon finite automata
5. Conclusions

Example: BFAs (boustrophedon finite automata)

see F., Paramasivan, Schmid, Thomas IWCIA 2015





The non-emptiness problem for B(D)FAs is NP-complete.

The inequivalence problem for BDFAs is NP-complete.

State minimization for BDFAs is NP-hard.

The ETH perspective

NON-EMPTINESS for BFAs $\approx$ INTERSECTION NON-EMPTINESS OF TALLY FAS.

Recall previous constructions: 3-COLORING$\leq$INTERSECTION NON-EMPTINESS

$k \approx n + m$, $q \approx n^2$

$\rightsquigarrow$ the related BFA has $\approx (n + m)n^2$ many states

This can be improved to $\approx (n + m)n$ by building loops sequentially.

Thm.: There is no algorithm that, given some BFA A with at most q states, decides if $L(A) \neq \emptyset$ in time $O^*(2^{o(\sqrt{q})})$ unless ETH fails.

Conversely, this problem can be solved in time $O^*(q^q)$.

Open: Close the gap!

Conclusions

Finite automata do offer quite some interesting hard problems.

Hardly ever studied under ETH.

Many questions still open, trying to match upper and lower bounds.

Recall tally problems: Match LB $O^*(2^{o(\sqrt[3]{q})})$ with UB $O^*(2^{\Theta(\sqrt{q \log q})})$.
LB for INTERSECTION NONEMPTINESS of DFAs ($O^*(2^{o(\min\{k, 2^q\})})$) looks bad, compared to UB $O^*(q^k)$.
etc. . . .

What about measuring number of transitions, not states?

Thanks for your attention!

Thanks for support through