Complexities (and results) for One-Way Jumping Automata

Big VMI event @FMI, 2023

S. Z. Fazekas   Robert Mercaș   O. Wu

18th September 2023

Loughborough University

Jumping Finite Automata

Jumping finite automata [Meduna and Zemek, 2012]

- process the input in arbitrary order

- can match number of letters

- accepts permutation-closed semilinear languages, so incomparable to REG [Fernau et al., 2015]

Right one-way jumping finite automaton ($\circlearrowright_R$DFA) [Chigahara et al., 2016]

$M = (Q, \Sigma, R, s, F)$, as in a DFA.

Elements of R are transition rules $\mathbf{p}a \to \mathbf{q} \in R$

Configurations of M are strings in $Q\Sigma^*$.
$x, y \in \Sigma^*$, $a \in \Sigma$, $\mathbf{p}, \mathbf{q} \in Q$ and $\mathbf{p}a \to \mathbf{q} \in R$.

Right one-way jumping finite automaton ($\circlearrowright_R$DFA) [Chigahara et al., 2016]

$M = (Q, \Sigma, R, s, F)$, as in a DFA.

Elements of R are transition rules $\mathbf{p}a \to \mathbf{q} \in R$

Configurations of M are strings in $Q\Sigma^*$.
$x, y \in \Sigma^*$, $a \in \Sigma$, $\mathbf{p}, \mathbf{q} \in Q$ and $\mathbf{p}a \to \mathbf{q} \in R$.

The *right one-way jumping relation* $\circlearrowright_R$ over $Q\Sigma^*$, ($\circlearrowright_R$DFA M jumps from configuration $\mathbf{p}xay$ to $\mathbf{q}yx$):

$$\mathbf{p}xay \circlearrowright \mathbf{q}yx$$

if $x \in \{\Sigma \setminus \Sigma_p\}^*$, where

$$\Sigma_p = \{b \in \Sigma \mid (\mathbf{p}, b, \mathbf{p}') \in R \text{ for some } \mathbf{p}' \in Q\}.$$

Example

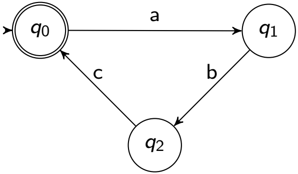
Let M be a $\circlearrowright_R$DFA given by

$$M = (\{\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2\}, \{a, b, c\}, R, \mathbf{q}_0, \{\mathbf{q}_0\}),$$

where R consists of the rules $\mathbf{q}_0 a \to \mathbf{q}_1$, $\mathbf{q}_1 b \to \mathbf{q}_2$ and $\mathbf{q}_2 c \to \mathbf{q}_0$.

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

$$\mathbf{q}_0 acbcab \circlearrowright_R \mathbf{q}_1 bcabc \circlearrowright_R \mathbf{q}_2 cabc \circlearrowright_R \mathbf{q}_0 abc \circlearrowright_R \mathbf{q}_1 bc \circlearrowright_R \mathbf{q}_2 c \circlearrowright_R \mathbf{q}_0$$



(Classic) example of $\circlearrowright_R$**DFA**

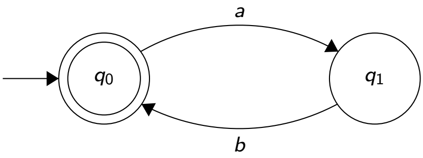


Figure: DFA for the regular language $(ab)^*$ and $\circlearrowright_R$**DFA** for the non-regular language $L(\mathcal{M}_{ab})\{w \mid |w|_a = |w|_b\}$

We have strict inclusion of languages, among **DFA**, $\circlearrowright_R$**DFA**, and **JFA**

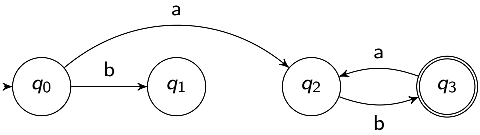


Figure: The automaton A with $L_D(A) \subsetneq L_R(A) \subsetneq L_J(A)$ [Beier and Holzer, 2018a]

$$L_D(A) = (ab)^+$$

$$L_R(A) = \{a\,w\,b \mid |w|_a = |w|_b\}$$

$$L_J(A) = \{a\,w \mid |w|_a + 1 = |w|_b\}$$

Who's who

- **REG** $\subsetneq$ $\circlearrowright_R$**DFA**.
- **CF** and $\circlearrowright_R$**DFA** are incomparable.
- $\circlearrowright_R$**DFA** $\subsetneq$ **CS**.

Not a very nice class (closure wise)

$\circlearrowright_R$**DFA** is not closed under "anything" ($\cup$, $\cap$, $\cdot$, $\overline{L}$, reversal, intersect. with reg. lang., homomorphisms, etc.)

Not a very nice class (closure wise)

$\circlearrowright_R$**DFA** is not closed under "anything" ($\cup$, $\cap$, $\cdot$, $\overline{L}$, reversal, intersect. with reg. lang., homomorphisms, etc.)

Not even unique minimal!

Figure: Different minimal $\circlearrowright_R$**DFA** accepting the language $\{w \in \{a,b\}^* \mid |w|_b > 0\}$.

Not a very nice class (closure wise)

$\circlearrowright_R$**DFA** is not closed under "anything" ($\cup$, $\cap$, $\cdot$, $\overline{L}$, reversal, intersect. with reg. lang., homomorphisms, etc.)

Not even unique minimal!

Figure: Different minimal $\circlearrowright_R$**DFA** accepting the language $\{w \in \{a, b\}^* \mid |w|_b > 0\}$.

Some subclasses admit nice characterizations [Beier and Holzer, 2018b]

Not a very nice class (closure wise)

$\circlearrowright_R$**DFA** is not closed under "anything" ($\cup$, $\cap$, $\cdot$, $\overline{L}$, reversal, intersect. with reg. lang., homomorphisms, etc.)

Not even unique minimal!

Figure: Different minimal $\circlearrowright_R$**DFA** accepting the language $\{w \in \{a, b\}^* \mid |w|_b > 0\}$.

Some subclasses admit nice characterizations [Beier and Holzer, 2018b]

Good decidability properties [Beier and Holzer, 2018a]

Not a very nice class (closure wise)

$\circlearrowright_R$**DFA** is not closed under "anything" ($\cup$, $\cap$, $\cdot$, $\overline{L}$, reversal, intersect. with reg. lang., homomorphisms, etc.)

Not even unique minimal!

Figure: Different minimal $\circlearrowright_R$**DFA** accepting the language $\{w \in \{a,b\}^* \mid |w|_b > 0\}$.

Some subclasses admit nice characterizations [Beier and Holzer, 2018b]

Good decidability properties [Beier and Holzer, 2018a]

$\circlearrowright_R$ processing mode studied with more general machine models: NFA, PDA, LBA, 2Way Jumping [Fazekas et al., 2019, Beier and Holzer, 2019, Fazekas et al., 2021]

Closed under	**REG**	$\circlearrowright_R$**DFA**	**JFA**
Union	yes	no	yes
Union with reg	yes	no	no
Intersection	yes	no	yes
Intersection with reg	yes	no	no
Complementation	yes	no	yes
Reversal	yes	no	yes
Concatenation	yes	no	no
Right conc. with reg	yes	no	no
Left conc. with reg	yes	yes	no
Kleene star or plus	yes	no	no
Homomorphism	yes	no	no
Inv. homomorphism	yes	no	no
Substitution	yes	no	no
Permutation	no	no	yes

Table: The Closure Properties of **REG**, $\circlearrowright_R$**DFA** and **JFA** [Beier and Holzer, 2018a]

1. Preliminaries
2. Model restrictions
3. Not again restrictions!
4. Conclusions

Some open questions

How do you get **REG**?

Some open questions

How do you get **REG**?

Theorem ([Fazekas and Yamamura, 2016])

For any $\circlearrowright_R$**DFA** $\mathcal{A} = (Q, \Sigma, R, \mathbf{s}, F)$ *and any constant* k*, the set of words accepted by* $\mathcal{A}$ *in at most* k *sweeps is regular.*

Some open questions

How do you get **REG**?

Theorem ([Fazekas and Yamamura, 2016])

For any $\circlearrowright_R$**DFA** $\mathcal{A} = (Q, \Sigma, R, \mathbf{s}, F)$ *and any constant* k*, the set of words accepted by* $\mathcal{A}$ *in at most* k *sweeps is regular.*

Conjecture ([Fazekas and Yamamura, 2016])

For any $\circlearrowright_R$**DFA** $\mathcal{A} = (Q, \Sigma, R, \mathbf{s}, F)$*, the language accepted by by* $\mathcal{A}$ *is regular if and only if it has constant sweep complexity.*

Some open questions

How do you get **REG**?

Theorem ([Fazekas and Yamamura, 2016])

For any $\circlearrowright_R$**DFA** $\mathcal{A} = (Q, \Sigma, R, \mathbf{s}, F)$ *and any constant* k*, the set of words accepted by* $\mathcal{A}$ *in at most* k *sweeps is regular.*

Conjecture ([Fazekas and Yamamura, 2016])

For any $\circlearrowright_R$**DFA** $\mathcal{A} = (Q, \Sigma, R, \mathbf{s}, F)$*, the language accepted by by* $\mathcal{A}$ *is regular if and only if it has constant sweep complexity.*

All known classes of languages accepted by $\circlearrowright_R$**DFA** had sweep complexity either constant or linear (which is also the upper bound).

Basics on $\circlearrowright_R$DFA

A $\circlearrowright_R$**DFA** jumping transition from $\mathbf{p}ax$ to $\mathbf{q}y$, denoted $\mathbf{p}ax \models \mathbf{q}y$:

(i) $\mathbf{p}ax \Rightarrow \mathbf{q}y$, where $x = y$ and $\mathbf{p}a \to \mathbf{q} \in R$, as defined earlier, or

(ii) $\mathbf{p}ax \circlearrowright \mathbf{p}xa$, when $a \in \Sigma \setminus \Sigma_p, \mathbf{p} = \mathbf{q}$ and $xa = y$. (*a jump*)

w accepted by M if $\mathbf{s}w \models^* \mathbf{f}$, and $L(M) = \{x \in \Sigma^* \mid \exists \mathbf{f} \in F : \mathbf{s}x \models^* \mathbf{f}\}$.

Basics on $\circlearrowright_R$DFA

A $\circlearrowright_R$**DFA** jumping transition from $\mathbf{p}ax$ to $\mathbf{q}y$, denoted $\mathbf{p}ax \models \mathbf{q}y$:

(*i*) $\mathbf{p}ax \Rightarrow \mathbf{q}y$, where $x = y$ and $\mathbf{p}a \to \mathbf{q} \in R$, as defined earlier, or

(*ii*) $\mathbf{p}ax \circlearrowright \mathbf{p}xa$, when $a \in \Sigma \setminus \Sigma_p, \mathbf{p} = \mathbf{q}$ and $xa = y$. (*a jump*)

w accepted by M if $\mathbf{s}w \models^* \mathbf{f}$, and $L(M) = \{x \in \Sigma^* \mid \exists \mathbf{f} \in F : \mathbf{s}x \models^* \mathbf{f}\}$.

Deficient states:

$\mathbf{p} \in Q$ is S-deficient if $\forall a \in S \subset \Sigma$ we have $\mathbf{p}a \to \mathbf{q} \notin R$ for every $\mathbf{q} \in Q$

Basics on $\circlearrowright_R$DFA

A $\circlearrowright_R$**DFA** jumping transition from $\mathbf{p}ax$ to $\mathbf{q}y$, denoted $\mathbf{p}ax \models \mathbf{q}y$:

(*i*) $\mathbf{p}ax \Rightarrow \mathbf{q}y$, where $x = y$ and $\mathbf{p}a \to \mathbf{q} \in R$, as defined earlier, or

(*ii*) $\mathbf{p}ax \circlearrowright \mathbf{p}xa$, when $a \in \Sigma \setminus \Sigma_p, \mathbf{p} = \mathbf{q}$ and $xa = y$. (*a jump*)

w accepted by M if $\mathbf{s}w \models^* \mathbf{f}$, and $L(M) = \{x \in \Sigma^* \mid \exists \mathbf{f} \in F : \mathbf{s}x \models^* \mathbf{f}\}$.

Deficient states:

$\mathbf{p} \in Q$ is S-deficient if $\forall a \in S \subset \Sigma$ we have $\mathbf{p}a \to \mathbf{q} \notin R$ for every $\mathbf{q} \in Q$

Sweeps:

position :	0	1	2	3	4	5	6	7
input	**a**	**a**	**a**	**a**	**b**	**b**	**b**	**b**
after sweep 1	ε	**a**	**a**	**a**	ε	**b**	**b**	**b**
after sweep 2	ε	ε	**a**	**a**	ε	ε	**b**	**b**
after sweep 3	ε	ε	ε	**a**	ε	ε	ε	**b**
after sweep 4	ε	ε	ε	ε	ε	ε	ε	ε

Figure: The computation table for a^4b^4 by $\mathcal{M}_{ab}$.

Definition of VMI for complexities [Fazekas et al., 2022]

Let $w \in L(M)$, and consider a computation in M on the input w:

$$C_M(w) : \mathbf{p}_0 w \models \mathbf{p}_1 w_1 \models \mathbf{p}_2 w_2 \models \ldots \models \mathbf{p}_m, \text{ where } \mathbf{p}_0 = s \text{ and } \mathbf{p}_m \in F,$$

$$E(C_M(w)) = \{i \geq 1 \mid \mathbf{p}_{i-1} w_{i-1} \circlearrowright \mathbf{p}_i w_i\}.$$

Definition of VMI for complexities [Fazekas et al., 2022]

Let $w \in L(M)$, and consider a computation in M on the input w:

$$C_M(w) : \mathbf{p}_0 w \models \mathbf{p}_1 w_1 \models \mathbf{p}_2 w_2 \models \ldots \models \mathbf{p}_m, \text{ where } \mathbf{p}_0 = s \text{ and } \mathbf{p}_m \in F,$$

$$E(C_M(w)) = \{i \geq 1 \mid \mathbf{p}_{i-1} w_{i-1} \circlearrowright \mathbf{p}_i w_i\}.$$

Define the *jumping complexity* of the computation of M on the word w by

$$jc_M(w) = \begin{cases} \min\{card(E(C_M(w))) \mid C_M(w) \text{ is a computation of } M \text{ on } w\} \\ 0, \text{ if } w \notin L(M). \end{cases}$$

Definition of VMI for complexities [Fazekas et al., 2022]

Let $w \in L(M)$, and consider a computation in M on the input w:

$$C_M(w) : \mathbf{p}_0 w \models \mathbf{p}_1 w_1 \models \mathbf{p}_2 w_2 \models \ldots \models \mathbf{p}_m, \text{ where } \mathbf{p}_0 = s \text{ and } \mathbf{p}_m \in F,$$

$$E(C_M(w)) = \{i \geq 1 \mid \mathbf{p}_{i-1} w_{i-1} \circlearrowright \mathbf{p}_i w_i\}.$$

Define the *jumping complexity* of the computation of M on the word w by

$$jc_M(w) = \begin{cases} \min\{card(E(C_M(w))) \mid C_M(w) \text{ is a computation of } M \text{ on } w\} \\ 0, \text{ if } w \notin L(M). \end{cases}$$

The *jumping complexity* of an automaton M is

$$jc_M(n) = \max\{jc_M(w) \mid |w| = n\}.$$

$\mathrm{JUMP}(f(n))$ class of langs. accepted by $\circlearrowright_R$**DFA** with $jc_M(n) = \mathcal{O}(f(n))$.

Definition of VMI for complexities [Fazekas et al., 2022]

Let $w \in L(M)$, and consider a computation in M on the input w:

$$C_M(w) : \mathbf{p}_0 w \models \mathbf{p}_1 w_1 \models \mathbf{p}_2 w_2 \models \ldots \models \mathbf{p}_m, \text{ where } \mathbf{p}_0 = s \text{ and } \mathbf{p}_m \in F,$$

$$E(C_M(w)) = \{i \geq 1 \mid \mathbf{p}_{i-1} w_{i-1} \circlearrowright \mathbf{p}_i w_i\}.$$

Define the *jumping complexity* of the computation of M on the word w by

$$jc_M(w) = \begin{cases} \min\{card(E(C_M(w))) \mid C_M(w) \text{ is a computation of } M \text{ on } w\} \\ 0, \text{ if } w \notin L(M). \end{cases}$$

The *jumping complexity* of an automaton M is

$$jc_M(n) = \max\{jc_M(w) \mid |w| = n\}.$$

$\mathrm{JUMP}(f(n))$ class of langs. accepted by $\circlearrowright_R$**DFA** with $jc_M(n) = \mathcal{O}(f(n))$.

The *sweep complexity* of an automaton M is $sc_M(n)$ is the maximum number of sweeps M makes on processing inputs $w \in L(M)$ of length n.
$\mathrm{SWEEP}(f(n))$ class of langs. accepted by $\circlearrowright_R$**DFA** with $sc_M(n) = \mathcal{O}(f(n))$.

Sweep complexities for bounded jumps [Fazekas et al., 2022]

Proposition

For any integer $k \geq 0$, the sweep complexity of any $ROWJFA_k$ is $O(\log n)$.

Sweep complexities for bounded jumps [Fazekas et al., 2022]

Proposition

For any integer $k \geq 0$, the sweep complexity of any $ROWJFA_k$ is $O(\log n)$.

Theorem

For every $k > 1$, there exists a $ROWJFA_{k-1}$ w/ sweep complexity $\Omega(\log n)$.

Deficit states [Fazekas et al., 2022]

Proposition

If a $\circlearrowright_R$**DFA** *has superconstant jump complexity, then it has two cyclic, reachable and co-reachable states* $\mathbf{p}$ *and* $\mathbf{q}$ *such that state* $\mathbf{p}$ *is a-deficient, for some* $a \in \Sigma$*, state* $\mathbf{q}$ *is reachable from* $\mathbf{p}$ *and* $\mathbf{q}aw \models^* \mathbf{q}$*, for some* $w \in \Sigma^*$*.*

Proposition

If a $\circlearrowright_R$**DFA** *has superconstant sweep complexity, then it has two reachable and co-reachable states* $\mathbf{p}$ *and* $\mathbf{q}$ *such that* $\mathbf{p}$ *is a-deficient,* $\mathbf{q}$ *is b-deficient, for some* $a, b \in \Sigma$ *with* $a \neq b$*, and* $\mathbf{p}buav \models^* \mathbf{q}av \models^* \mathbf{p}$*, for some* $u, v \in \Sigma^*$*.*

But then... [Fazekas and Mercaș, 2023]

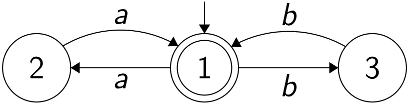


Figure: $L(\mathcal{B}) = \{w \in \{a, b\}^* \mid |w|_a \equiv 0 \bmod 2, |w|_b \equiv 0 \bmod 2\}$.

But then... [Fazekas and Mercaș, 2023]

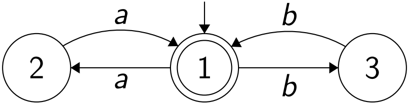


Figure: $L(\mathcal{B}) = \{w \in \{a, b\}^* \mid |w|_a \equiv 0 \bmod 2, |w|_b \equiv 0 \bmod 2\}$.

Proposition

$L(\mathcal{B})$ is regular.

But then... [Fazekas and Mercaș, 2023]

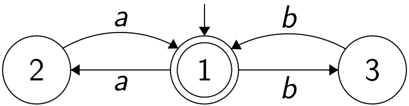


Figure: $L(\mathcal{B}) = \{w \in \{a,b\}^* \mid |w|_a \equiv 0 \bmod 2, |w|_b \equiv 0 \bmod 2\}$.

Proposition

$L(\mathcal{B})$ is regular.

Theorem

The sweep complexity of $\mathcal{B}$ is $\Theta(\log n)$.

And we continue

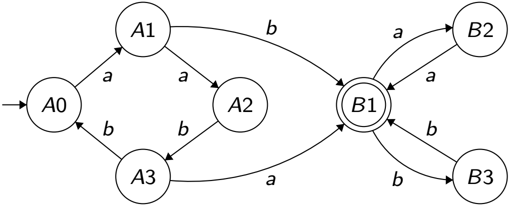


Figure: $L(\mathcal{C}) = \{w \in \{a,b\}^* \mid |w|_a \equiv 1 \mod 2, |w|_b \equiv 1 \mod 2\}$

And we continue

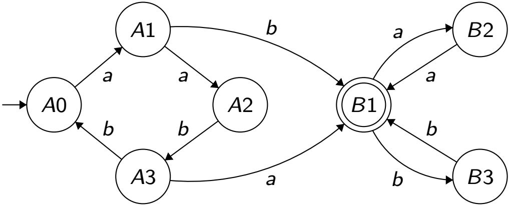


Figure: $L(\mathcal{C}) = \{w \in \{a,b\}^* \mid |w|_a \equiv 1 \mod 2, |w|_b \equiv 1 \mod 2\}$

Proposition

$L(\mathcal{C})$ is regular.

And we continue

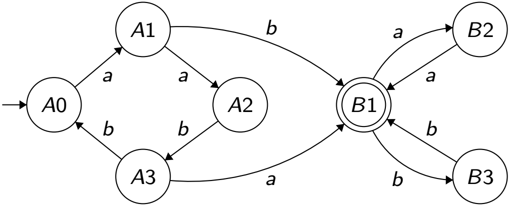


Figure: $L(\mathcal{C}) = \{w \in \{a,b\}^* \mid |w|_a \equiv 1 \mod 2, |w|_b \equiv 1 \mod 2\}$

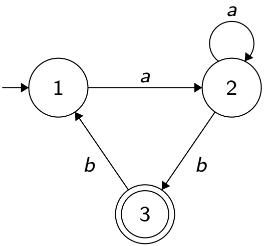
Proposition

$L(\mathcal{C})$ is regular.

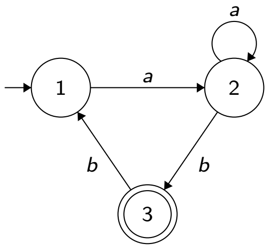
Proposition

The sweep complexity of $\mathcal{C}$ is $\Theta(n)$.

But **non-REG** languages also can have sublinear sweep complexity



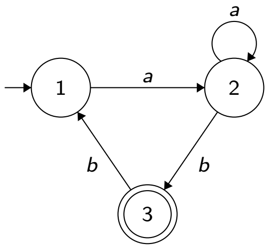
But **non-REG** languages also can have sublinear sweep complexity



Lemma

The ROWJFA $\mathcal{D}$ accepts a non-regular language.

But **non-REG** languages also can have sublinear sweep complexity

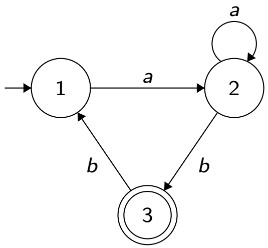


Lemma

The ROWJFA $\mathcal{D}$ accepts a non-regular language.

$\mathcal{D}$ accepts $\varphi^n(ab)$ in $n+1$ sweeps, where $\varphi(a) = abab$, $\varphi(b) = b$.

But **non-REG** languages also can have sublinear sweep complexity



Lemma

The ROWJFA $\mathcal{D}$ accepts a non-regular language.

$\mathcal{D}$ accepts $\varphi^n(ab)$ in $n+1$ sweeps, where $\varphi(a) = abab$, $\varphi(b) = b$.

Lemma

The sweep complexity of $\mathcal{D}$ is $\Theta(\log n)$.

Complexity classes separation

Theorem

$\mathit{SWEEP}(1) \subsetneq \mathit{SWEEP}(\log n)$

Lemma

Any automaton which accepts $L_{ab} = \{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$ *has sweep complexity* $\Theta(n)$.

Theorem

For any $f : \mathbb{N} \Rightarrow \mathbb{N}$ *with* $f(n) \in o(n)$ *we have* $\mathit{SWEEP}(f(n)) \subsetneq \mathit{SWEEP}(n)$.

Answered several questions regarding sweep complexity [Fazekas et al., 2022]

1. Is the language of each machine with $\omega(1)$ complexity non-regular? NO.
2. Is there a machine with sweep complexity between constant and linear, that is, $\omega(1)$ and $o(n)$? YES.
3. Is there a *language* with sweep complexity between constant and linear, that is, all machines accepting it have superconstant complexity and at least one has sublinear? YES.
4. Is there an upper bound in terms of sweep complexity on machines accepting regular languages? NO.
5. Are machines less complex in the case of binary alphabets? NO.

What is next?

- do machines with arbitrary (constructible) sublinear complexity exist ($\Theta(\log^k n)$ and $\Theta(n^\epsilon)$, for constants $k > 1$ and $\epsilon < 1$)?
- logarithmic complexity can produce non-regular languages, but is it doable with less of this 'non-regular' resource?
- is it decidable given a machine or language and a function $f(n)$, whether the machine/language has $\Theta(f(n))$ sweep complexity?
- allow nondeterminism
- find sweet spot between extensions/restrictions and good closure props
- solve the open problems regarding equivalence and regularity

Thank you, VICTOR!

Questions?

Beier, S. and Holzer, M. (2018a).
Decidability of right one-way jumping finite automata.
In Hoshi, M. and Seki, S., editors, *DLT*, pages 109–120. Springer.

Beier, S. and Holzer, M. (2018b).
Properties of right one-way jumping finite automata.
In Konstantinidis, S. and Pighizzini, G., editors, *DCFS*, volume 10952 of *LNCS*, pages 11–23. Springer.

Beier, S. and Holzer, M. (2019).
Nondeterministic right one-way jumping finite automata (extended abstract).
In Hospodár, M., Jirásková, G., and Konstantinidis, S., editors, *DCFS*, pages 74–85.

Chigahara, H., Fazekas, S. Z., and Yamamura, A. (2016).
One-way jumping finite automata.
International Journal of Foundations of Computer Science, 27(3):391–405.

Fazekas, S. Z., Hoshi, K., and Yamamura, A. (2019).
Enhancement of automata with jumping modes.
In Castillo-Ramirez, A. and de Oliveira, P. P. B., editors, *AUTOMATA 2019*, pages 62–76. Springer.

Fazekas, S. Z., Hoshi, K., and Yamamura, A. (2021).
Two-way deterministic automata with jumping mode.
Theoretical Computer Science, 864:92–102.

Fazekas, S. Z. and Mercaș, R. (2023).
Sweep complexity revisited.
In Nagy, B., editor, *CIAA*, volume 14151 of *LNCS*, pages 116–127. Springer.

Fazekas, S. Z., Mercaș, R., and Wu, O. (2022).
Complexities for jumps and sweeps.
Journal of Automata, Languages and Combinatorics, 27(1-3):131–149.

Fazekas, S. Z. and Yamamura, A. (2016).
On regular languages accepted by one-way jumping finite automata.
In *NCMA, short papers*, pages 7–14.

Fernau, H., Paramasivan, M., and Schmid, M. L. (2015).
Jumping finite automata: Characterizations and complexity.
In Drewes, F., editor, *CIAA*, volume 9223 of *LNCS*, pages 89–101. Springer.

Meduna, A. and Zemek, P. (2012).
Jumping finite automata.
International Journal of Foundations of Computer Science, 23(7):1555–1578.

DFA for even and odd

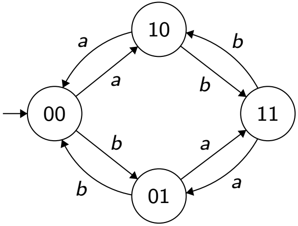


Figure: DFA accepting words with even (for **00** final state) or odd (for **11** final state) number of *a*'s and *b*'s.