Weight-reducing Hennie Machines and Their Descriptional Complexity[1]

Daniel Průša

Czech Technical University in Prague

International Conference on Language and Automata Theory and Applications (LATA) 2014

[1] The author was supported by the Grant Agency of the Czech Republic under the project P103/10/0783.

Outline

Regular languages and finite automata

Basic model: One-way deterministic finite-state automaton (1DFA).

Extensions:

- Nondeterminism (1NFA), alternation (1AFA).
- Two-way movement (2DFA, 2NFA).
- Usage of a pebble (2DPA).

All the models equal in power, however, they differ in succinctness of their descriptions.

Regular languages and finite automata

Basic model: One-way deterministic finite-state automaton (1DFA).

Extensions:

- Nondeterminism (1NFA), alternation (1AFA).
- Two-way movement (2DFA, 2NFA).
- Usage of a pebble (2DPA).

All the models equal in power, however, they differ in succinctness of their descriptions.

Size of automata description

- Number of states – frequently studied measure (for example: 1DFA needs 2^n states to simulate 1NFA in the worst case).
- Number of transitions – better suits our purposes.

Theorem ([Shannon 1956])

Each Turing machine has an equivalent with only two active states.

Size of automata description

- Number of states – frequently studied measure (for example: 1DFA needs 2^n states to simulate 1NFA in the worst case).
- Number of transitions – better suits our purposes.

Theorem ([Shannon 1956])

Each Turing machine has an equivalent with only two active states.

Hennie machine

- Bounded, single-tape Turing machine.

$\vdash$	a_1	a_2	a_3	a_4	a_5	a_6	$\dashv$

- The number of transitions performed over every tape field limited by a constant k.

Theorem ([Hennie 1965])

Each language accepted by a Hennie machine is regular language.

The condition can be further relaxed:

- Linear time [Hennie 1965].
- $\mathcal{O}(n \log n)$ time [Hartmanis 1968].

Hennie machine

- Bounded, single-tape Turing machine.

$\vdash$	a_1	a_2	a_3	a_4	a_5	a_6	$\dashv$

- The number of transitions performed over every tape field limited by a constant k.

Theorem ([Hennie 1965])

Each language accepted by a Hennie machine is regular language.

The condition can be further relaxed:

- Linear time [Hennie 1965].
- $\mathcal{O}(n \log n)$ time [Hartmanis 1968].

Hennie machine

- Bounded, single-tape Turing machine.

$\vdash$	a_1	a_2	a_3	a_4	a_5	a_6	$\dashv$

- The number of transitions performed over every tape field limited by a constant k.

Theorem ([Hennie 1965])

Each language accepted by a Hennie machine is regular language.

The condition can be further relaxed:

- Linear time [Hennie 1965].
- $\mathcal{O}(n \log n)$ time [Hartmanis 1968].

Hennie machine - properties

Nonrecursive trade-off with respect to 1DFA:

- Let $c(n)$ be the cost of the optimal simulation of Hennie machine by 1DFA.
- $c(n)$ is not bounded by any recursive function.

Nonconstructive:

- Given a Turing machine T, it is undecidable if T is a Hennie machine.

Hennie machine - properties

Nonrecursive trade-off with respect to 1DFA:

- Let $c(n)$ be the cost of the optimal simulation of Hennie machine by 1DFA.
- $c(n)$ is not bounded by any recursive function.

Nonconstructive:

- Given a Turing machine T, it is undecidable if T is a Hennie machine.

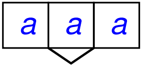
Our goal

1. Propose a constructive variant of Hennie machine.
2. Study the descriptional complexity of the model.

Weight-reducing property

Weight function defined on working symbols: $\mu : \Gamma \to \mathbb{N}$

Each transition has to decrease the weight of the scanned symbol $\Rightarrow$ the bound k is incorporated in Γ ($k \leq |\Gamma|$).



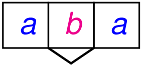
$\mu(b) < \mu(a)$

Sgraffito automaton [Prusa and Mraz 2012] - 2D automaton for recognition of picture languages, the same principle applied.

Weight-reducing property

Weight function defined on working symbols: $\mu : \Gamma \to \mathbb{N}$

Each transition has to decrease the weight of the scanned symbol $\Rightarrow$ the bound k is incorporated in Γ ($k \leq |\Gamma|$).



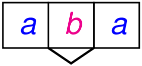
$\mu(b) < \mu(a)$

Sgraffito automaton [Prusa and Mraz 2012] - 2D automaton for recognition of picture languages, the same principle applied.

Weight-reducing property

Weight function defined on working symbols: $\mu : \Gamma \to \mathbb{N}$

Each transition has to decrease the weight of the scanned symbol $\Rightarrow$ the bound k is incorporated in Γ ($k \leq |\Gamma|$).



$\mu(b) < \mu(a)$

Sgraffito automaton [Prusa and Mraz 2012] - 2D automaton for recognition of picture languages, the same principle applied.

Formal definition of weight-reducing Hennie machine

$M = (Q, \Sigma, \Gamma, \delta, q_0, Q_F, \mu)$, where

- Σ is an input alphabet
- Γ is a working alphabet, $\Gamma \supseteq \Sigma$, $\Gamma \cap \{\vdash, \dashv\} = \emptyset$
- Q is a finite, non-empty set of states
- q_0 is the initial state, $q_0 \in Q$
- Q_F is the set of final states, $Q_F \subseteq Q$
- μ a weight function, $\mu : \Gamma \to \mathbb{N}$
- δ a transition relation, $\delta : (Q \setminus Q_F) \times (\Gamma \cup \{\vdash, \dashv\}) \to 2^{Q \times (\Gamma \cup \{\vdash, \dashv\}) \times \{\leftarrow, 0, \rightarrow\}}$
  - each transition over the input is weight-reducing

$$(q', a', d) \in \delta(q, a) \ \Rightarrow \mu(a') < \mu(a)$$
$$\text{for all } q, q' \in Q, d \in \{\leftarrow, 0, \rightarrow\}, a, a' \in \Gamma$$

  - automaton is bounded

Formal definition of weight-reducing Hennie machine

$M = (Q, \Sigma, \Gamma, \delta, q_0, Q_F, \mu)$, where

- Σ is an input alphabet
- Γ is a working alphabet, $\Gamma \supseteq \Sigma$, $\Gamma \cap \{\vdash, \dashv\} = \emptyset$
- Q is a finite, non-empty set of states
- q_0 is the initial state, $q_0 \in Q$
- Q_F is the set of final states, $Q_F \subseteq Q$
- μ a weight function, $\mu : \Gamma \to \mathbb{N}$
- δ a transition relation, $\delta : (Q \setminus Q_F) \times (\Gamma \cup \{\vdash, \dashv\}) \to 2^{Q \times (\Gamma \cup \{\vdash, \dashv\}) \times \{\leftarrow, 0, \rightarrow\}}$
  - each transition over the input is weight-reducing

    $$(q', a', d) \in \delta(q, a) \Rightarrow \mu(a') < \mu(a)$$

    for all $q, q' \in Q, d \in \{\leftarrow, 0, \rightarrow\}, a, a' \in \Gamma$
  - automaton is bounded

Formal definition of weight-reducing Hennie machine

$M = (Q, \Sigma, \Gamma, \delta, q_0, Q_F, \mu)$, where

- Σ is an input alphabet
- Γ is a working alphabet, $\Gamma \supseteq \Sigma$, $\Gamma \cap \{\vdash, \dashv\} = \emptyset$
- Q is a finite, non-empty set of states
- q_0 is the initial state, $q_0 \in Q$
- Q_F is the set of final states, $Q_F \subseteq Q$
- μ a weight function, $\mu : \Gamma \to \mathbb{N}$
- δ a transition relation, $\delta : (Q \setminus Q_F) \times (\Gamma \cup \{\vdash, \dashv\}) \to 2^{Q \times (\Gamma \cup \{\vdash, \dashv\}) \times \{\leftarrow, 0, \rightarrow\}}$
  - each transition over the input is weight-reducing

    $$(q', a', d) \in \delta(q, a) \Rightarrow \mu(a') < \mu(a)$$
    for all $q, q' \in Q, d \in \{\leftarrow, 0, \rightarrow\}, a, a' \in \Gamma$
  - automaton is bounded

Formal definition of weight-reducing Hennie machine

$M = (Q, \Sigma, \Gamma, \delta, q_0, Q_F, \mu)$, where

- Σ is an input alphabet
- Γ is a working alphabet, $\Gamma \supseteq \Sigma$, $\Gamma \cap \{\vdash, \dashv\} = \emptyset$
- Q is a finite, non-empty set of states
- q_0 is the initial state, $q_0 \in Q$
- Q_F is the set of final states, $Q_F \subseteq Q$
- μ a weight function, $\mu : \Gamma \to \mathbb{N}$
- δ a transition relation,
  $\delta : (Q \setminus Q_F) \times (\Gamma \cup \{\vdash, \dashv\}) \to 2^{Q \times (\Gamma \cup \{\vdash, \dashv\}) \times \{\leftarrow, 0, \rightarrow\}}$
  - each transition over the input is weight-reducing

    $$(q', a', d) \in \delta(q, a) \Rightarrow \mu(a') < \mu(a)$$
    for all $q, q' \in Q, d \in \{\leftarrow, 0, \rightarrow\}, a, a' \in \Gamma$
  - automaton is bounded

Formal definition of weight-reducing Hennie machine

$M = (Q, \Sigma, \Gamma, \delta, q_0, Q_F, \mu)$, where

- Σ is an input alphabet
- Γ is a working alphabet, $\Gamma \supseteq \Sigma$, $\Gamma \cap \{\vdash, \dashv\} = \emptyset$
- Q is a finite, non-empty set of states
- q_0 is the initial state, $q_0 \in Q$
- Q_F is the set of final states, $Q_F \subseteq Q$
- μ a weight function, $\mu : \Gamma \rightarrow \mathbb{N}$
- δ a transition relation, $\delta : (Q \setminus Q_F) \times (\Gamma \cup \{\vdash, \dashv\}) \rightarrow 2^{Q \times (\Gamma \cup \{\vdash, \dashv\}) \times \{\leftarrow, 0, \rightarrow\}}$
  - each transition over the input is weight-reducing

$$(q', a', d) \in \delta(q, a) \Rightarrow \mu(a') < \mu(a)$$
$$\text{for all } q, q' \in Q, d \in \{\leftarrow, 0, \rightarrow\}, a, a' \in \Gamma$$

  - automaton is bounded

Conversion to a weight-reducing Hennie machine

Theorem

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Q_F)$ be a Hennie machine with the number of transions performed over a tape field by k. There is a weight-reducing Hennie machine A such that $L(A) = L(M)$ and the working alphabet of A has no more than $(k+1)|\Gamma|$ symbols.

- If M is deterministic, then A is deterministic as well.

Focus on deterministic weight-reducing (det-wr) Hennie machines.

Conversion to a weight-reducing Hennie machine

Theorem

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Q_F)$ be a Hennie machine with the number of transions performed over a tape field by k. There is a weight-reducing Hennie machine A such that $L(A) = L(M)$ and the working alphabet of A has no more than $(k+1)|\Gamma|$ symbols.

- If M is deterministic, then A is deterministic as well.

Focus on deterministic weight-reducing (det-wr) Hennie machines.

Conversion to a weight-reducing Hennie machine

Theorem

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Q_F)$ be a Hennie machine with the number of transions performed over a tape field by k. There is a weight-reducing Hennie machine A such that $L(A) = L(M)$ and the working alphabet of A has no more than $(k+1)|\Gamma|$ symbols.

- If M is deterministic, then A is deterministic as well.

Focus on deterministic weight-reducing (det-wr) Hennie machines.

Comparison with 1DFA

Theorem

Each n-state m-working symbol det-wr Hennie machine can be simulated by a 1DFA with $2^{2^{\mathcal{O}(m \log n)}}$ states.

- Take crossing sequences of a Hennie machine as states of 1NFA, convert it to 1DFA.

Theorem

For each integer $n \geq 1$, there is a language B_n such that

- *B_n is accepted by a det-wr Hennie machine with $\mathcal{O}(1)$ states and $\mathcal{O}(n)$ working symbols,*
- *A 1DFA requires at least 2^{2^n} states to accept B_n.*

Comparison with 1DFA

Theorem

Each n-state m-working symbol det-wr Hennie machine can be simulated by a 1DFA *with* $2^{2^{\mathcal{O}(m \log n)}}$ *states.*

- Take crossing sequences of a Hennie machine as states of 1NFA, convert it to 1DFA.

Theorem

For each integer $n \geq 1$, there is a language B_n such that

- *B_n is accepted by a det-wr Hennie machine with $\mathcal{O}(1)$ states and $\mathcal{O}(n)$ working symbols,*
- *A* 1DFA *requires at least* 2^{2^n} *states to accept B_n.*

Comparison with 1DFA

Theorem

Each n-state m-working symbol det-wr Hennie machine can be simulated by a 1DFA *with* $2^{2^{\mathcal{O}(m \log n)}}$ *states.*

- Take crossing sequences of a Hennie machine as states of 1NFA, convert it to 1DFA.

Theorem

For each integer $n \geq 1$, there is a language B_n such that

- *B_n is accepted by a det-wr Hennie machine with $\mathcal{O}(1)$ states and $\mathcal{O}(n)$ working symbols,*
- *A* 1DFA *requires at least* 2^{2^n} *states to accept B_n.*

Languages B_n

B_n over $\Sigma = \{0, 1, \$\}$, consists of strings $v_1\$v_2\$\ldots\$v_j$, where

- $j \geq 2$, each $v_i \in \{0, 1\}^*$, $|v_j| \leq n$
- there is $\ell < j$ such that $v_\ell = v_j$

$$11\$011\$10110\$011 \in B_3, B_4, \ldots$$

1DFA needs $\Omega(2^{2^n})$ states. Apply Myhill-Nerode theorem.
There are 2^{2^n} subsets of $\{0, 1\}^n$.

$$000\$001\$101\$111\$001 \in B_3$$
$$000\$010\$101\$111\$001 \notin B_3$$

$001 is a distinguishing extension

Languages B_n

B_n over $\Sigma = \{0, 1, \$\}$, consists of strings $v_1\$v_2\$\dots\$v_j$, where

- $j \geq 2$, each $v_i \in \{0, 1\}^*$, $|v_j| \leq n$
- there is $\ell < j$ such that $v_\ell = v_j$

$$11\$011\$10110\$011 \in B_3, B_4, \dots$$

1DFA needs $\Omega(2^{2^n})$ states. Apply Myhill-Nerode theorem.
There are 2^{2^n} subsets of $\{0, 1\}^n$.

$$000\$001\$101\$111\$001 \in B_3$$
$$000\$010\$101\$111\$001 \notin B_3$$

$001 is a distinguishing extension

Languages B_n

B_n over $\Sigma = \{0, 1, \$\}$, consists of strings $v_1\$v_2\$\ldots\$v_j$, where

- $j \geq 2$, each $v_i \in \{0, 1\}^*$, $|v_j| \leq n$
- there is $\ell < j$ such that $v_\ell = v_j$

$$11\$011\$10110\$011 \in B_3, B_4, \ldots$$

1DFA needs $\Omega(2^{2^n})$ states. Apply Myhill-Nerode theorem.
There are 2^{2^n} subsets of $\{0, 1\}^n$.

$$000\$001\$101\$111\$001 \in B_3$$
$$000\$010\$101\$111\$001 \notin B_3$$

$\$001$ is a distinguishing extension

Simulation of 1NFA

Theorem

Each n-state 1NFA *can be simulated by a det-wr Hennie machine with the number of transitions polynomial in n.*

⇒ Efficient elimination of nondeterminism by two-way motion and restricted rewriting.

Problem ([Sakoda and Sipser 1978])

What is the cost, in terms of states, of the optimal simulation of

- 1NFA *by* 2DFA
- 2NFA *by* 2DFA

Conjecture: the trade-offs are exponential

Simulation of 1NFA

Theorem

Each n-state 1NFA can be simulated by a det-wr Hennie machine with the number of transitions polynomial in n.

$\Rightarrow$ Efficient elimination of nondeterminism by two-way motion and restricted rewriting.

Problem ([Sakoda and Sipser 1978])

What is the cost, in terms of states, of the optimal simulation of

- 1NFA *by* 2DFA
- 2NFA *by* 2DFA

Conjecture: the trade-offs are exponential

Simulation of 1NFA

Theorem

Each n-state 1NFA can be simulated by a det-wr Hennie machine with the number of transitions polynomial in n.

$\Rightarrow$ Efficient elimination of nondeterminism by two-way motion and restricted rewriting.

Problem ([Sakoda and Sipser 1978])

What is the cost, in terms of states, of the optimal simulation of

- 1NFA *by* 2DFA
- 2NFA *by* 2DFA

Conjecture: the trade-offs are exponential

Simulation of 1NFA

Theorem

Each n-state 1NFA *can be simulated by a det-wr Hennie machine with the number of transitions polynomial in n.*

$\Rightarrow$ Efficient elimination of nondeterminism by two-way motion and restricted rewriting.

Problem ([Sakoda and Sipser 1978])

What is the cost, in terms of states, of the optimal simulation of

- 1NFA *by* 2DFA
- 2NFA *by* 2DFA

Conjecture: the trade-offs are exponential

Simulation of 1NFA

Given an n-state 1NFA and an input $w \in \Sigma^*$.

1. $|w| \geq n$.
   - Enough space to record states reachable by 1NFA on the tape.
   - Blocks of length n, each used to simulate 1NFA when computing inside the block.

q_1	q_2	q_3	q_4
1	1	1	0

Simulation of 1NFA

Given an n-state 1NFA and an input $w \in \Sigma^*$.

1. $|w| \geq n$.

- Enough space to record states reachable by 1NFA on the tape.
- Blocks of length n, each used to simulate 1NFA when computing inside the block.

q_1	q_2	q_3	q_4
1	1	1	0

Simulation of 1NFA

Given an n-state 1NFA and an input $w \in \Sigma^*$.

1. $|w| \geq n$.
   - Enough space to record states reachable by 1NFA on the tape.
   - Blocks of length n, each used to simulate 1NFA when computing inside the block.

q_1	q_2	q_3	q_4
1	1	1	0

Simulation of 1NFA

2. $|w| < n$.

Undirected s-t connectivity problem.

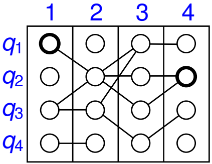
	1	2	3	4
q_1				
q_2				
q_3				
q_4				

- Solvable by a deterministic logarithmic-space algorithm [Reingold 2008].
- $\mathcal{O}(\log n)$ space simulated in states of det-wr Hennie machine.

Simulation of 1NFA

2. $|w| < n$.

Undirected s-t connectivity problem.

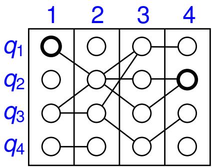


- Solvable by a deterministic logarithmic-space algorithm [Reingold 2008].
- $\mathcal{O}(\log n)$ space simulated in states of det-wr Hennie machine.

Simulation of 1NFA

2. $|w| < n$.

Undirected s-t connectivity problem.



- Solvable by a deterministic logarithmic-space algorithm [Reingold 2008].
- $\mathcal{O}(\log n)$ space simulated in states of det-wr Hennie machine.

det-wr Hennie machine → 2NFA (trade-off)

Theorem ([Kari and Moore 2001])

Let L be a finite unary language accepted by a 2NFA with n states. The longest string in L has length at most $n+2$.

$$U_n = \{a^{2^n}\},\ n \geq 1$$

- Each 2NFA accepting U_n has $\Omega(2^n)$ states.
- There is a det-wr Hennie machine with $\mathcal{O}(n)$ states and $\mathcal{O}(n)$ working symbols (i.e., $\mathcal{O}(n^2)$ transitions) accepting U_n, yielding a $2^{\Omega(\sqrt{n})}$ trade-off.

det-wr Hennie machine $\to$ 2NFA (trade-off)

Theorem ([Kari and Moore 2001])

Let L be a finite unary language accepted by a 2NFA *with n states. The longest string in L has length at most $n+2$.*

$$U_n = \{a^{2^n}\}, \ n \geq 1$$

- Each 2NFA accepting U_n has $\Omega(2^n)$ states.
- There is a det-wr Hennie machine with $\mathcal{O}(n)$ states and $\mathcal{O}(n)$ working symbols (i.e., $\mathcal{O}(n^2)$ transitions) accepting U_n, yielding a $2^{\Omega(\sqrt{n})}$ trade-off.

det-wr Hennie machine → 2NFA (trade-off)

Theorem ([Kari and Moore 2001])

Let L be a finite unary language accepted by a 2NFA with n states. The longest string in L has length at most $n+2$.

$$U_n = \{a^{2^n}\},\ n \geq 1$$

- Each 2NFA accepting U_n has $\Omega(2^n)$ states.
- There is a det-wr Hennie machine with $\mathcal{O}(n)$ states and $\mathcal{O}(n)$ working symbols (i.e., $\mathcal{O}(n^2)$ transitions) accepting U_n, yielding a $2^{\Omega(\sqrt{n})}$ trade-off.

det-wr Hennie machine → 2NFA (trade-off)

Theorem ([Kari and Moore 2001])

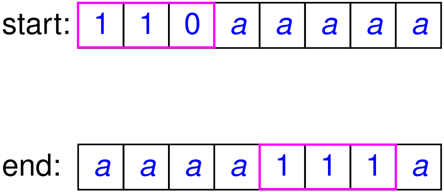
Let L be a finite unary language accepted by a 2NFA with n states. The longest string in L has length at most $n + 2$.

$$U_n = \{a^{2^n}\},\ n \geq 1$$

- Each 2NFA accepting U_n has $\Omega(2^n)$ states.
- There is a det-wr Hennie machine with $\mathcal{O}(n)$ states and $\mathcal{O}(n)$ working symbols (i.e., $\mathcal{O}(n^2)$ transitions) accepting U_n, yielding a $2^{\Omega(\sqrt{n})}$ trade-off.

Recognition of U_n

- Binary counter of length n used, initialized by n.
- Repeatedly incremented and shifted by one field to the right.
- Position of the right end checked when $2^n - 1$ is reached.



Simulation of 2NFA

2NFA with n states, input w. Similar approach as in the case of 1NFA can be applied.

- $|w| < n$: STCON problem (directed s-t connectivity) - deterministically in space $\mathcal{O}(\log^2 n)$
- $|w| \geq n$: $\mathcal{O}(n^2)$ steps of 2NFA to be simulated in each block of length n

$\Rightarrow$ det-wr Hennie machine with $\mathcal{O}(n^{\log n})$ states

Relation to 1AFA:

Theorem ([Birget 1993])

Each n-state 2NFA can be simulated by a n^2-state 1AFA.

Simulation of 2NFA

2NFA with n states, input w. Similar approach as in the case of 1NFA can be applied.

- $|w| < n$: STCON problem (directed s-t connectivity) - deterministically in space $\mathcal{O}(\log^2 n)$
- $|w| \geq n$: $\mathcal{O}(n^2)$ steps of 2NFA to be simulated in each block of length n

$\Rightarrow$ det-wr Hennie machine with $\mathcal{O}(n^{\log n})$ states

Relation to 1AFA:

Theorem ([Birget 1993])

Each n-state 2NFA can be simulated by a n^2-state 1AFA.

Simulation of 2NFA

2NFA with n states, input w. Similar approach as in the case of 1NFA can be applied.

- $|w| < n$: STCON problem (directed s-t connectivity) - deterministically in space $\mathcal{O}(\log^2 n)$
- $|w| \geq n$: $\mathcal{O}(n^2)$ steps of 2NFA to be simulated in each block of length n

$\Rightarrow$ det-wr Hennie machine with $\mathcal{O}(n^{\log n})$ states

Relation to 1AFA:

Theorem ([Birget 1993])

Each n-state 2NFA can be simulated by a n^2-state 1AFA.

Simulation of 2NFA

2NFA with n states, input w. Similar approach as in the case of 1NFA can be applied.

- $|w| < n$: STCON problem (directed s-t connectivity) - deterministically in space $\mathcal{O}(\log^2 n)$
- $|w| \geq n$: $\mathcal{O}(n^2)$ steps of 2NFA to be simulated in each block of length n

$\Rightarrow$ det-wr Hennie machine with $\mathcal{O}(n^{\log n})$ states

Relation to 1AFA:

Theorem ([Birget 1993])

Each n-state 2NFA *can be simulated by a n^2-state* 1AFA.

Simulation of deterministic one-pebble automata

Theorem

Each deterministic one-pebble n-state automaton can be simulated by a det-wr Hennie machine with $\mathcal{O}(n^6)$ transitions.

Results summary

1. det-wr Hennie machine $\rightarrow$ 1DFA:
   - double exponential trade-off (like in the case of 1AFA or 2DPA)
2. 1NFA, 2DPA $\rightarrow$ det-wr Hennie machine:
   - polynomial trade-off
3. det-wr Hennie machine $\rightarrow$ 2NFA: $2^{\Omega(\sqrt{n})}$
4. 2NFA $\rightarrow$ det-wr Hennie machine: $\mathcal{O}(n^{\log n})$

Results summary

1. det-wr Hennie machine $\rightarrow$ 1DFA:
   - double exponential trade-off (like in the case of 1AFA or 2DPA)
2. 1NFA, 2DPA $\rightarrow$ det-wr Hennie machine:
   - polynomial trade-off
3. det-wr Hennie machine $\rightarrow$ 2NFA: $2^{\Omega(\sqrt{n})}$
4. 2NFA $\rightarrow$ det-wr Hennie machine: $\mathcal{O}(n^{\log n})$

Results summary

1. det-wr Hennie machine $\rightarrow$ 1DFA:
   - double exponential trade-off (like in the case of 1AFA or 2DPA)
2. 1NFA, 2DPA $\rightarrow$ det-wr Hennie machine:
   - polynomial trade-off
3. det-wr Hennie machine $\rightarrow$ 2NFA: $2^{\Omega(\sqrt{n})}$
4. 2NFA $\rightarrow$ det-wr Hennie machine: $\mathcal{O}(n^{\log n})$

Results summary

1. det-wr Hennie machine $\rightarrow$ 1DFA:
   - double exponential trade-off (like in the case of 1AFA or 2DPA)
2. 1NFA, 2DPA $\rightarrow$ det-wr Hennie machine:
   - polynomial trade-off
3. det-wr Hennie machine $\rightarrow$ 2NFA: $2^{\Omega(\sqrt{n})}$
4. 2NFA $\rightarrow$ det-wr Hennie machine: $\mathcal{O}(n^{\log n})$

Thank you!