Nondeterministic Biautomata and Their Descriptional Complexity

Markus Holzer and Sebastian Jakobi

Institut für Informatik
Justus-Liebig-Universität
Arndtstr. 2, 35392 Gießen, Germany

23. Theorietag „Automaten und Formale Sprachen“
Ilmenau, 25.-27. September 2013

1. Biautomata
   - Nondeterministic Biautomata
   - Structural Properties of Biautomata
2. Descriptional Complexity
   - Complexity of the $\diamond$-Property
   - Conversion Problems for $\diamond$-F-Biautomata
   - Complexity of the F-Property
3. Conclusion

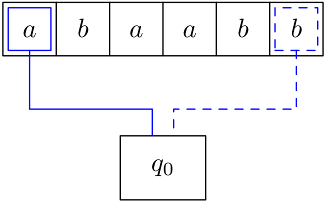
1. Biautomata
   - Nondeterministic Biautomata
   - Structural Properties of Biautomata

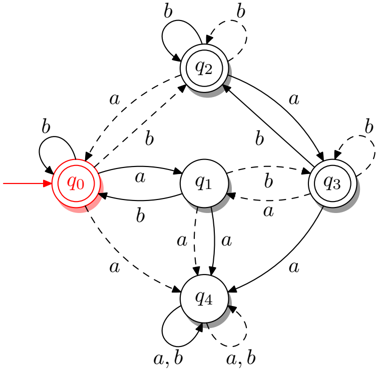
2. Descriptional Complexity
   - Complexity of the $\diamond$-Property
   - Conversion Problems for $\diamond$-F-Biautomata
   - Complexity of the F-Property

3. Conclusion

Biautomata ...as introduced in [Klíma, Polák, 2011]

A biautomaton $A = (Q, \Sigma, \cdot, \circ, q_0, F)$ is a deterministic finite automaton with two reading heads (left-to-right and right-to-left) ...



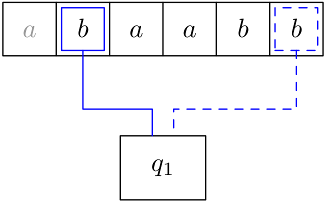


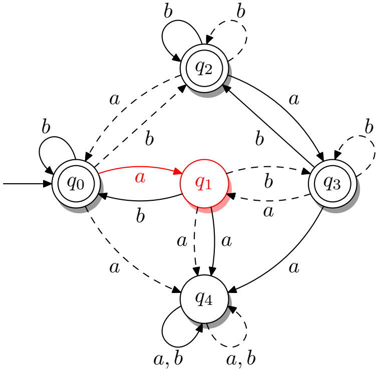
...that satisfies the following two restrictions:

1. heads move independently: $(q \cdot a) \circ b = (q \circ b) \cdot a$ ($\diamond$-property)
2. forward/backward acceptance: $q \cdot a \in F \iff q \circ a \in F$ (F-property)

Biautomata ...as introduced in [Klíma, Polák, 2011]

A biautomaton $A = (Q, \Sigma, \cdot, \circ, q_0, F)$ is a deterministic finite automaton with two reading heads (left-to-right and right-to-left) ...



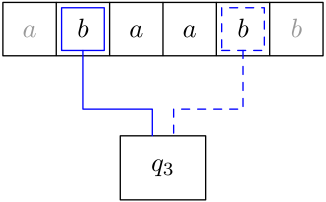


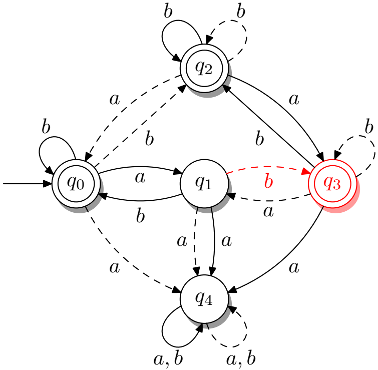
...that satisfies the following two restrictions:

1. heads move independently: $(q \cdot a) \circ b = (q \circ b) \cdot a$ ($\diamond$-property)
2. forward/backward acceptance: $q \cdot a \in F \iff q \circ a \in F$ (F-property)

Biautomata …as introduced in [Klíma, Polák, 2011]

A biautomaton $A = (Q, \Sigma, \cdot, \circ, q_0, F)$ is a deterministic finite automaton with two reading heads (left-to-right and right-to-left) …



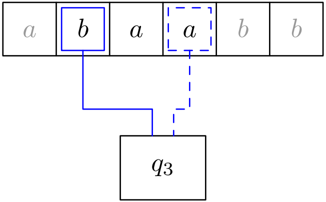


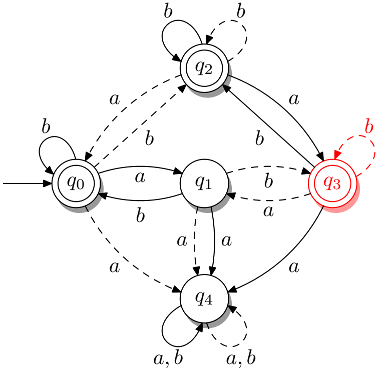
…that satisfies the following two restrictions:

1. heads move independently: $(q \cdot a) \circ b = (q \circ b) \cdot a$ ($\diamond$-property)
2. forward/backward acceptance: $q \cdot a \in F \iff q \circ a \in F$ (F-property)

Biautomata . . . as introduced in [Klíma, Polák, 2011]

A biautomaton $A = (Q, \Sigma, \cdot, \circ, q_0, F)$ is a deterministic finite automaton with two reading heads (left-to-right and right-to-left) . . .



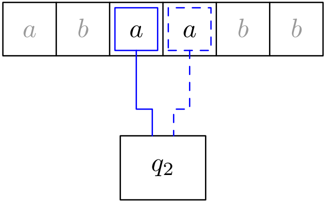


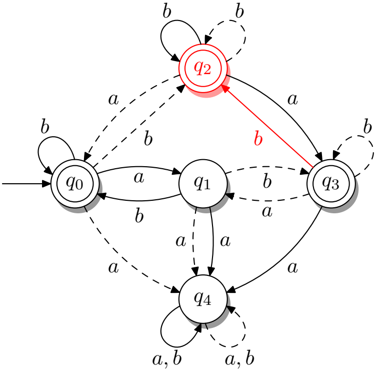
. . . that satisfies the following two restrictions:

1. heads move independently: $(q \cdot a) \circ b = (q \circ b) \cdot a$ ($\diamond$-property)
2. forward/backward acceptance: $q \cdot a \in F \iff q \circ a \in F$ (F-property)

Biautomata ... as introduced in [Klíma, Polák, 2011]

A biautomaton $A = (Q, \Sigma, \cdot, \circ, q_0, F)$ is a deterministic finite automaton with two reading heads (left-to-right and right-to-left) ...



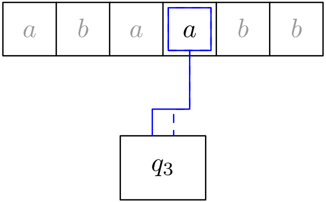


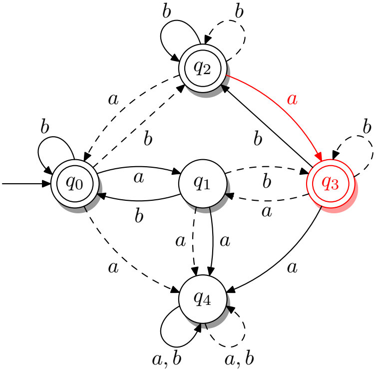
... that satisfies the following two restrictions:

1. heads move independently: $(q \cdot a) \circ b = (q \circ b) \cdot a$ ($\diamond$-property)
2. forward/backward acceptance: $q \cdot a \in F \iff q \circ a \in F$ (F-property)

Biautomata …as introduced in [Klíma, Polák, 2011]

A biautomaton $A = (Q, \Sigma, \cdot, \circ, q_0, F)$ is a deterministic finite automaton with two reading heads (left-to-right and right-to-left) …



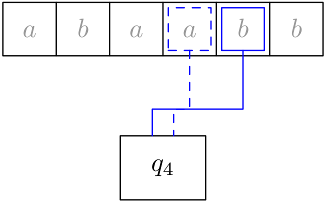


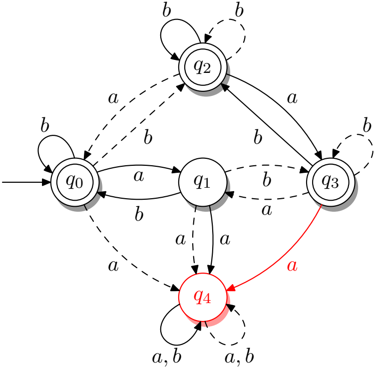
…that satisfies the following two restrictions:

1. heads move independently: $(q \cdot a) \circ b = (q \circ b) \cdot a$ ($\diamond$-property)
2. forward/backward acceptance: $q \cdot a \in F \iff q \circ a \in F$ (F-property)

Biautomata ...as introduced in [Klíma, Polák, 2011]

A biautomaton $A = (Q, \Sigma, \cdot, \circ, q_0, F)$ is a deterministic finite automaton with two reading heads (left-to-right and right-to-left) ...





...that satisfies the following two restrictions:

1. heads move independently: $(q \cdot a) \circ b = (q \circ b) \cdot a$ ($\diamond$-property)
2. forward/backward acceptance: $q \cdot a \in F \iff q \circ a \in F$ (F-property)

Results on Biautomata

[Klíma, Polák, 2011/12]

Similarities to ordinary deterministic finite automata:

- biautomata accept regular languages
- have unique minimal automata

Simple characterizations of regular language classes via biautomata:

- prefix-suffix testable languages
- piecewise testable and k-piecewise testable languages

Results on Biautomata

[Klíma, Polák, 2011/12]

Similarities to ordinary deterministic finite automata:

- biautomata accept regular languages
- have unique minimal automata

Simple characterizations of regular language classes via biautomata:

- prefix-suffix testable languages
- piecewise testable and k-piecewise testable languages

[Jirásková, Klíma, 2012]

Descriptional Complexity: NFA/DFA/syntactic monoid $\longrightarrow$ biautomaton

Results on Biautomata

[Klíma, Polák, 2011/12]

Similarities to ordinary deterministic finite automata:

- biautomata accept regular languages
- have unique minimal automata

Simple characterizations of regular language classes via biautomata:

- prefix-suffix testable languages
- piecewise testable and k-piecewise testable languages

[Jirásková, Klíma, 2012]

Descriptional Complexity: NFA/DFA/syntactic monoid $\longrightarrow$ biautomaton

[H., J., 2013]

Brzozowski-like minimization of biautomata

Characterizations of cyclic and commutative regular languages

Biautomata ...more general

nondeterministic biautomaton (NBiA) $A = (Q, \Sigma, \cdot, \circ, I, F)$:

- Q — finite set of states
- Σ — input alphabet
- $\cdot\colon Q \times \Sigma \to 2^Q$ — forward transition function
- $\circ\colon Q \times \Sigma \to 2^Q$ — backward transition function
- $I \subseteq Q$ — set of initial states
- $F \subseteq Q$ — set of final or accepting states

A accepts $w \in \Sigma^*$ if $w = u_1 u_2 \ldots u_k v_k \ldots v_2 v_1$ $(u_i, v_i \in \Sigma^*)$, and

$$[((\ldots((((I \cdot u_1) \circ v_1) \cdot u_2) \circ v_2) \ldots) \cdot u_k) \circ v_k] \cap F \neq \emptyset$$

deterministic biautomaton (DBiA):

- $|I| = 1$
- $|q \cdot a| = 1$ and $|q \circ a| = 1$ for all $q \in Q$, $a \in \Sigma$

Biautomata ... more general

nondeterministic biautomaton (NBiA) $A = (Q, \Sigma, \cdot, \circ, I, F)$:

- Q — finite set of states
- Σ — input alphabet
- $\cdot\colon Q \times \Sigma \to 2^Q$ — forward transition function
- $\circ\colon Q \times \Sigma \to 2^Q$ — backward transition function
- $I \subseteq Q$ — set of initial states
- $F \subseteq Q$ — set of final or accepting states

A accepts $w \in \Sigma^*$ if $w = u_1 u_2 \ldots u_k v_k \ldots v_2 v_1$ $(u_i, v_i \in \Sigma^*)$, and

$$[((\ldots((((I \cdot u_1) \circ v_1) \cdot u_2) \circ v_2) \ldots) \cdot u_k) \circ v_k] \cap F \neq \emptyset$$

deterministic biautomaton (DBiA):

- $|I| = 1$
- $|q \cdot a| = 1$ and $|q \circ a| = 1$ for all $q \in Q$, $a \in \Sigma$

Biautomata ... more general

nondeterministic biautomaton (NBiA) $A = (Q, \Sigma, \cdot, \circ, I, F)$:

- Q — finite set of states
- Σ — input alphabet
- $\cdot\colon Q \times \Sigma \to 2^Q$ — forward transition function
- $\circ\colon Q \times \Sigma \to 2^Q$ — backward transition function
- $I \subseteq Q$ — set of initial states
- $F \subseteq Q$ — set of final or accepting states

A accepts $w \in \Sigma^*$ if $w = u_1 u_2 \ldots u_k v_k \ldots v_2 v_1$ $(u_i, v_i \in \Sigma^*)$, and

$$[((\ldots((((I \cdot u_1) \circ v_1) \cdot u_2) \circ v_2) \ldots) \cdot u_k) \circ v_k] \cap F \neq \emptyset$$

deterministic biautomaton (DBiA):

- $|I| = 1$
- $|q \cdot a| = 1$ and $|q \circ a| = 1$ for all $q \in Q$, $a \in \Sigma$

Structural Properties of Biautomata

$\diamond$-property: for all $q \in Q$, $a, b \in \Sigma$: $(q \cdot a) \circ b = (q \circ b) \cdot a$

F-property: for all $q \in Q$, $a \in \Sigma$: $(q \cdot a) \cap F \neq \emptyset \iff (q \circ a) \cap F \neq \emptyset$

$\diamond$-F-DBiA are biautomata as introduced in [Klíma, Polák, 2011]

Structural Properties of Biautomata

$\diamond$-property: for all $q \in Q$, $a, b \in \Sigma$: $(q \cdot a) \circ b = (q \circ b) \cdot a$

F-property: for all $q \in Q$, $a \in \Sigma$: $(q \cdot a) \cap F \neq \emptyset \Longleftrightarrow (q \circ a) \cap F \neq \emptyset$

$\diamond$-F-DBiA are biautomata as introduced in [Klíma, Polák, 2011]

Simpler acceptance conditions

general: A accepts w if $w = u_1 u_2 \ldots u_k v_k \ldots v_2 v_1$ $(u_i, v_i \in \Sigma^*)$ and
$[((\ldots((((I \cdot u_1) \circ v_1) \cdot u_2) \circ v_2) \ldots) \cdot u_k) \circ v_k] \cap F \neq \emptyset$

with $\diamond$: A accepts w if $w = uv$ $(u, v \in \Sigma^*)$ and
$[(I \cdot u) \circ v] \cap F \neq \emptyset$ (equivalent: $[(I \circ v) \cdot u] \cap F \neq \emptyset$)

with $\diamond$, F: A accepts w if
$[I \cdot w] \cap F \neq \emptyset$ (equivalent: $[I \circ w] \cap F \neq \emptyset$)

Complexity of the $\diamond$-Property

Lemma

For any given n-state $\diamond$-NBiA A one can construct an equivalent NFA that has $O(n^2)$ states. In particular $L(A)$ is regular.

Complexity of the $\diamond$-Property

Lemma

For any given n-state $\diamond$-NBiA A one can construct an equivalent NFA that has $O(n^2)$ states. In particular $L(A)$ is regular.

Lemma

1. Let $G = (N, T, P, S)$ be a linear context-free grammar in normalform. Then one can construct an equivalent $(|N| + 1)$-state NiBA A. Automaton A may satisfy the F-property.
2. Let A be an n-state NBiA. Then one can construct an equivalent linear context-free grammar G in normalform with $(n + 1)$ nonterminals.

Complexity of the $\diamond$-Property

Lemma

For any given n-state $\diamond$-NBiA A one can construct an equivalent NFA that has $O(n^2)$ states. In particular $L(A)$ is regular.

Lemma

1. Let $G = (N, T, P, S)$ be a linear context-free grammar in normalform. Then one can construct an equivalent $(|N| + 1)$-state NiBA A. Automaton A may satisfy the F-property.
2. Let A be an n-state NBiA. Then one can construct an equivalent linear context-free grammar G in normalform with $(n + 1)$ nonterminals.

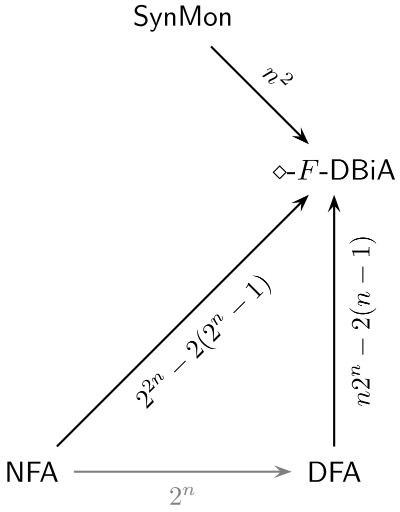
Theorem

The trade-off between NBiAs and $\diamond$-NBiAs is non-recursive.

Conversions to $\diamond$-F-DBiA

From [Jirásková, Klíma 2012]:

1. NFA $\to$ $\diamond$-F-DBiA
2. DFA $\to$ $\diamond$-F-DBiA
3. SynMon $\to$ $\diamond$-F-DBiA



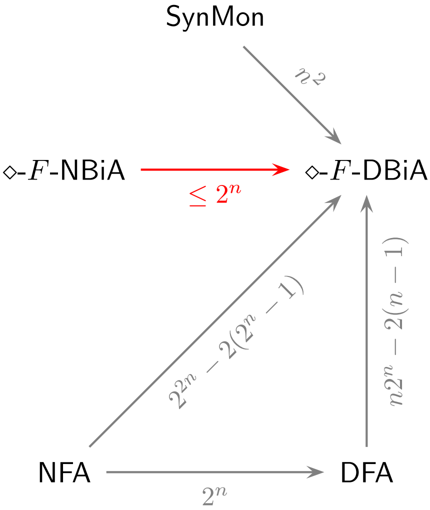
Conversions to $\diamond$-F-DBiA

Theorem

For any given n-state NBiA one can construct an equivalent DBiA that has 2^n states. Further, this construction preserves the $\diamond$- and F-properties.

Proof Idea:

Power-set construction. $\square$



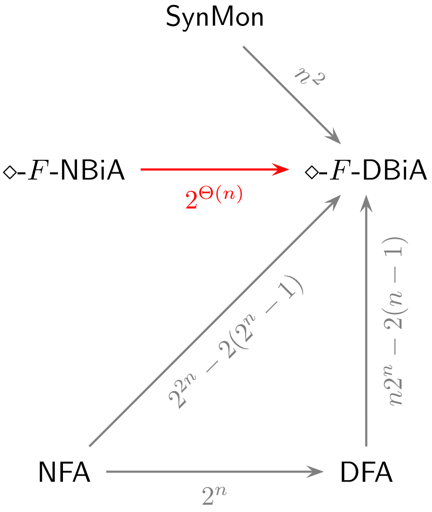
Conversions to $\diamond$-F-DBiA

Lemma

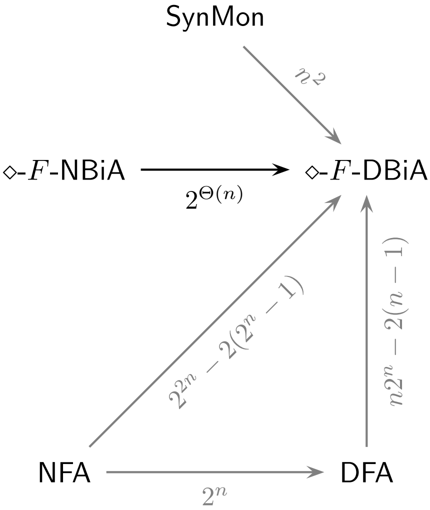
For all $m \geq 1$ there is a language L_m accepted by a $\diamond$-F-NBiA with $3m+2$ states and for which every $\diamond$-F-DBiA needs at least $2^{2m}+1$ states.

Proof Idea:

$L_m = \Sigma^* a \Sigma^{m-1} a \Sigma^*$ $\square$



Conversions to $\diamond$-F-NBiA



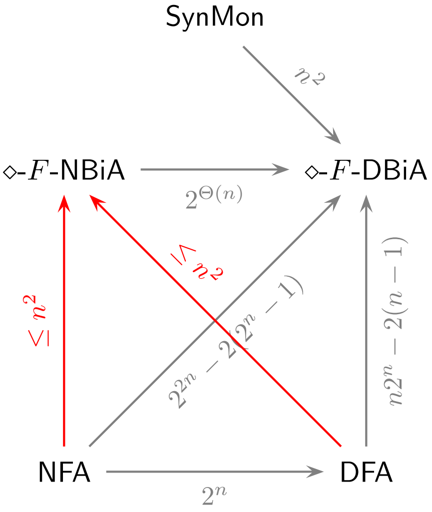
Conversions to $\diamond$-F-NBiA

Theorem

For any given n-state NFA one can construct an equivalent $\diamond$-F-NBiA that has n^2 states.

Proof Idea:

Construct product of forward and backward automata. $\square$



A Lower Bound Technique for $\diamond$-F-NBiA

Lemma

A set $S = \{\, (x_i, y_i, z_i) \mid x_i, y_i, z_i \in \Sigma^*, 1 \leq i \leq n \,\}$ is a *bi-fooling set* for $L \subseteq \Sigma^*$ if

1. for $1 \leq i \leq n$ it is $x_i \cdot y_i \cdot z_i \in L$, and
2. for $1 \leq i, j \leq n$, with $i \neq j$, it is $x_i \cdot y_j \cdot z_i \notin L$ or $x_j \cdot y_i \cdot z_j \notin L$.

Then any $\diamond$-F-NBiA for L has at least $|S|$ states.

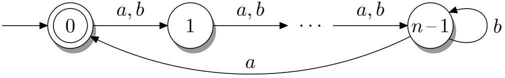
Proof Idea: Let $A = (Q, \Sigma, \cdot, \circ, I, F)$ be a $\diamond$-F-NBiA for L.
If $x_i y_i z_i \in L$ then there is $q_i \in [(I \cdot x_i) \circ z_i]$ such that $(q_i \cdot y_i) \cap F \neq \emptyset$.
If $|Q| < |S|$ then there are $i \neq j$ with $q_i = q_j$.
Then $x_i y_j z_i \in L$ and $x_j y_i z_j \in L$, a contradiction. $\square$

Lower Bound for DFA $\to$ $\diamond$-F-NBiA

Theorem

For every $n \geq 1$ there is a language L_n accepted by an n-state DFA, such that any $\diamond$-F-NBiA needs n^2 states to accept L_n.

Sketch of Proof: Consider A_n



with the bi-fooling set $S = \{\, (x_{i,j}, y_{i,j}, z_{i,j}) \mid 0 \leq i, j \leq n-1 \,\}$, where

$$x_{i,j} = a^i, \quad y_{i,j} = a^{n-i} b^{n-2} a^{j+2}, \quad z_{i,j} = a^{n-j}.$$

Note that $x_{i,j} \cdot y_{i,j} \cdot z_{i,j} = a^n b^{n-2} a^{n+2} \in L(A)$.

Consider $(i,j) \neq (i',j')$:

$i = i'$: Word $x_{i,j} \cdot y_{i',j'} \cdot z_{i,j} = a^n b^{n-2} a^{n+2+j'-j} \notin L(A)$.

$i \neq i'$: Assume $(j'-j) \bmod n \neq n-1$, then

$x_{i,j} \cdot y_{i',j'} \cdot z_{i,j} = a^{n-i'+i} b^{n-2} a^{n+2+j'-j} \notin L(A)$.

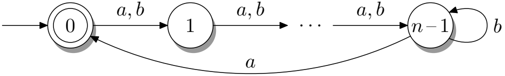
Case $(j'-j) \bmod n = n-1$ similar. $\square$

Lower Bound for DFA $\to \diamond$-F-NBiA

Theorem

For every $n \geq 1$ there is a language L_n accepted by an n-state DFA, such that any $\diamond$-F-NBiA needs n^2 states to accept L_n.

Sketch of Proof: Consider A_n



with the bi-fooling set $S = \{\, (x_{i,j}, y_{i,j}, z_{i,j}) \mid 0 \leq i, j \leq n-1 \,\}$, where

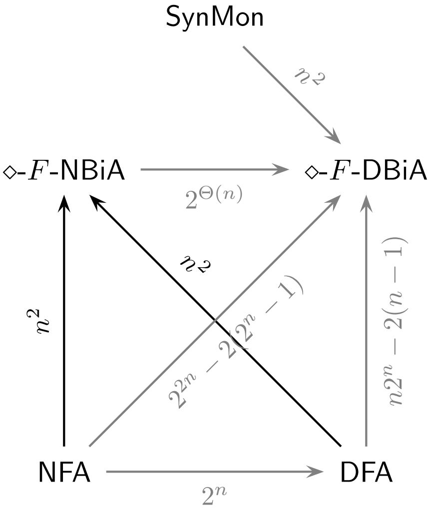
$$x_{i,j} = a^i, \quad y_{i,j} = a^{n-i} b^{n-2} a^{j+2}, \quad z_{i,j} = a^{n-j}.$$

Note that $x_{i,j} \cdot y_{i,j} \cdot z_{i,j} = a^n b^{n-2} a^{n+2} \in L(A)$.
Consider $(i, j) \neq (i', j')$:

$i = i'$: Word $x_{i,j} \cdot y_{i',j'} \cdot z_{i,j} = a^n b^{n-2} a^{n+2+j'-j} \notin L(A)$.

$i \neq i'$: Assume $(j' - j) \bmod n \neq n-1$, then
$x_{i,j} \cdot y_{i',j'} \cdot z_{i,j} = a^{n-i'+i} b^{n-2} a^{n+2+j'-j} \notin L(A)$.
Case $(j' - j) \bmod n = n - 1$ similar. $\square$

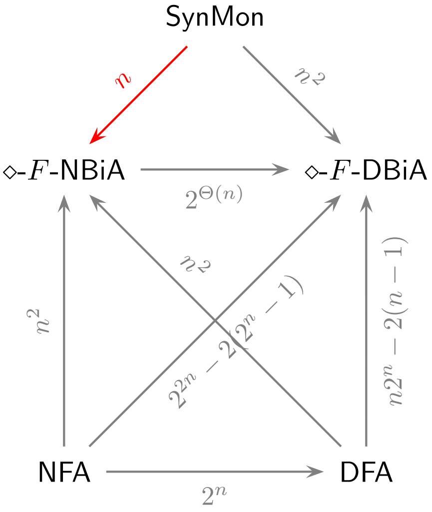
Conversions to $\diamond$-F-NBiA



Conversions to $\diamond$-F-NBiA

Theorem

Let L be regular language given by a syntactic monoid of size n. Then one can construct a $\diamond$-F-NBiA for L that has at most n states.
This bound can be reached for every $n \geq 1$.



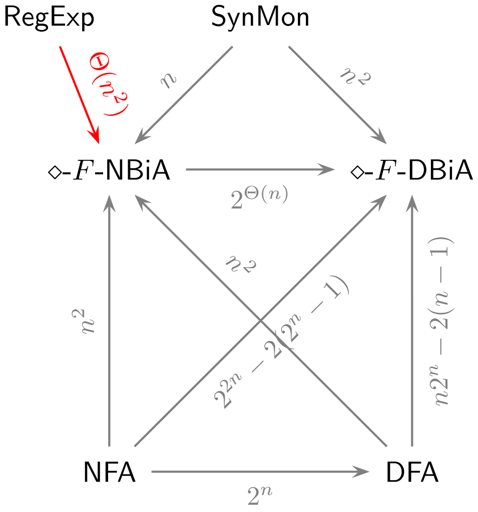
Conversions to $\diamond$-F-NBiA

Theorem

For any given regular expression of alphabetic width n one can construct an equivalent $\diamond$-F-NBiA that has $(n+1)^2$ states.

Lemma

For all $n \geq 1$ there is a regular language L_n with alphabetic width n, such that any $\diamond$-F-NBiA needs n^2 states to accept L_n.



Complexity of the F-Property

The F-property is easy for $\diamond$-NBiAs,

Lemma

For any given n-state $\diamond$-NBiA one can construct an equivalent $\diamond$-F-NBiA that has $O(n^4)$ states.

Complexity of the F-Property

The F-property is easy for $\diamond$-NBiAs,

Lemma

For any given n-state $\diamond$-NBiA one can construct an equivalent $\diamond$-F-NBiA that has $O(n^4)$ states.

...and hard for $\diamond$-DBiAs.

Lemma

For every $n \geq 1$ there is a regular language L_n accepted by a $\diamond$-DBiA with $O(n^2)$ states and for which every $\diamond$-F-DBiA needs $\Omega(2^n)$ states.

Conclusion

Summary

- nondeterminism for biautomata
- structural properties: $\diamond$, F
- descriptional complexity of conversion problems

Further Research

- tight bound for determinization of $\diamond$-F-NBiAs
- tight bounds for $\diamond$-BiA $\rightarrow$ $\diamond$-F-BiA and $\diamond$-BiA $\rightarrow$ NFA/DFA
- understanding the F-property (also concerning minimization problems)
- further characterizations of subregular language classes
- residual biautomata
- ...

Thank you for your attention!