Fine-grained Complexity of Ambiguity Testing of Automata

Anita Dürr

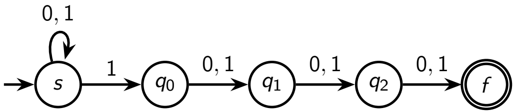
Saarland University and Max Planck Institute for Informatics

Joint work with Karolina Drabik, Fabian Frei, Filip Mazowiecki and Karol Węgrzycki

Warsaw University, CISPA, Max Planck Institute for Informatics

Finite Automata

Finite Automaton



$$\mathcal{A} = (Q, \delta, \Sigma, \{s\}, \{f\})$$

$$\mathcal{L}(\mathcal{A}) = (0+1)^*1(0+1)^3$$

- A word $w \in \Sigma^*$ is accepted by $\mathcal{A}$ if it labels an accepting run

DFA

✓ efficient for complementation, minimization, acceptance, containment, equivalence ...

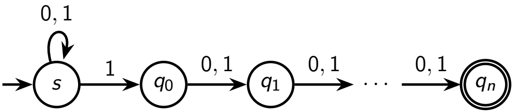
NFA (this talk: no ε-transitions)

✗ containment and equivalence are PSPACE-complete

✓ exponentially more succint

Finite Automata

Finite Automaton



$$\mathcal{A} = (Q, \delta, \Sigma, \{s\}, \{q_n\})$$
$$\mathcal{L}(\mathcal{A}) = (0+1)^*1(0+1)^n$$

- A word $w \in \Sigma^*$ is accepted by $\mathcal{A}$ if it labels an accepting run

DFA

✓ efficient for complementation, minimization, acceptance, containment, equivalence ...

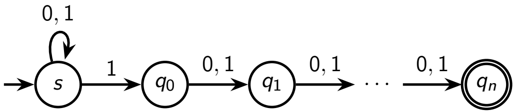
NFA (this talk: no ε-transitions)

✗ containment and equivalence are PSPACE-complete

✓ exponentially more succint

Finite Automata

Finite Automaton



$$\mathcal{A} = (Q, \delta, \Sigma, \{s\}, \{q_n\})$$
$$\mathcal{L}(\mathcal{A}) = (0+1)^*1(0+1)^n$$

- A word $w \in \Sigma^*$ is accepted by $\mathcal{A}$ if it labels an accepting run

DFA

✓ efficient for complementation, minimization, acceptance, containment, equivalence ...

- words label a unique accepting run

NFA (this talk: no ε-transitions)

✗ containment and equivalence are PSPACE-complete

✓ exponentially more succint

- words can label multiple accepting runs

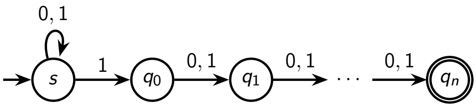
Ambiguity of Finite Automata

Definition

An NFA $\mathcal{A}$ is **unambiguous** if every word has $\leqslant 1$ accepting run. $\rightarrow$ UFA

UFA

- ✓ UFA is more succint than DFA
- ✓ containment and equivalence are still polytime in UFA



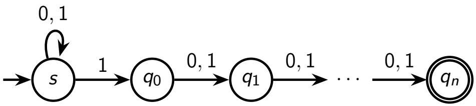
Ambiguity of Finite Automata

Definition

An NFA $\mathcal{A}$ is **unambiguous** if every word has $\leqslant 1$ accepting run. $\rightarrow$ UFA

UFA

- ✓ UFA is more succint than DFA
- ✓ containment and equivalence are still polytime in UFA



Given an NFA $\mathcal{A}$, is it unambiguous?

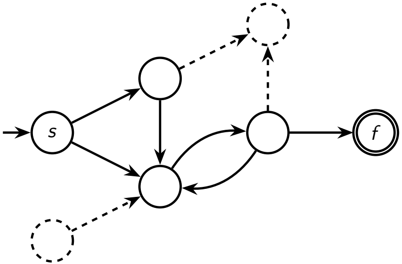
- Is decidable in n^2 time. [1991, 2001]
- Tight under OV or k-cycle. [our work]

$n = |Q| + |\delta| =$ size of $\mathcal{A}$

Reminder

Assumptions

- single starting state s and single accepting state f
- $\mathcal{A}$ is *trim*, i.e. every state is reachable from s and co-reachable from f

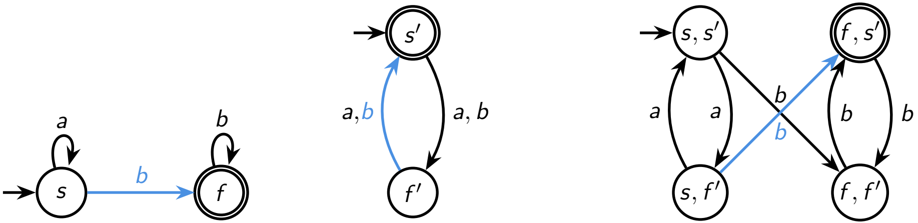


Reminder

Assumptions

- single starting state s and single accepting state f
- $\mathcal{A}$ is *trim*, i.e. every state is reachable from s and co-reachable from f

Intersection of NFAs

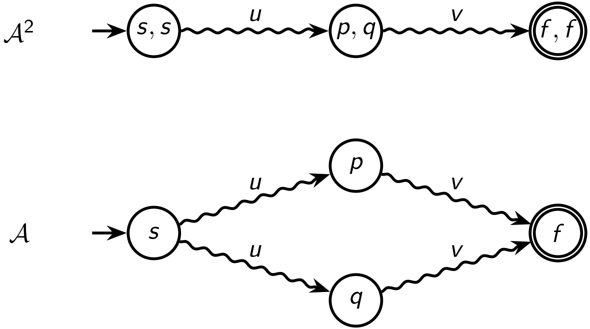


$\mathcal{L}(\mathcal{A}_1) = a^*b^*$ $\qquad$ $\mathcal{L}(\mathcal{A}_2) = (\Sigma\Sigma)^*$ $\qquad$ $\mathcal{L}(\mathcal{A}_1 \cap \mathcal{A}_2) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$

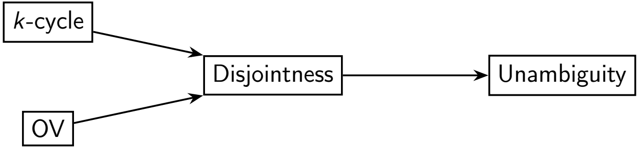
Unambiguity Test

Lemma

$\mathcal{A}$ is unambiguous iff every state in trimmed $\mathcal{A}^2 = \mathcal{A} \cap \mathcal{A}$ is of the form (q, q) where $q \in Q$.



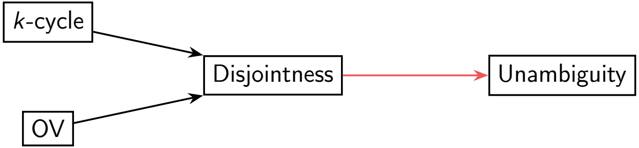
Lower-bounds



Problem (Disjointness)

Given two DFAs $\mathcal{D}_1$ and $\mathcal{D}_2$, decide whether $\mathcal{L}(\mathcal{D}_1 \cap \mathcal{D}_2) = \varnothing$.

Lower-bounds

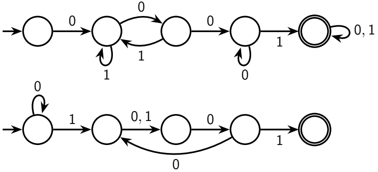


Problem (Disjointness)

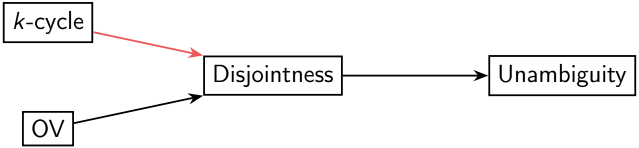
Given two DFAs $\mathcal{D}_1$ and $\mathcal{D}_2$, decide whether $\mathcal{L}(\mathcal{D}_1 \cap \mathcal{D}_2) = \varnothing$.

Disjointness $\to$ Unambiguity

- Given $\mathcal{D}_1$ and $\mathcal{D}_2$, let $\mathcal{A} \coloneqq \mathcal{D}_1 \cup \mathcal{D}_2$.
- $\mathcal{A}$ is unambiguous $\Leftrightarrow \mathcal{L}(\mathcal{D}_1 \cap \mathcal{D}_2) = \varnothing$



Lower-bounds



Problem (Disjointness)

Given two DFAs $\mathcal{D}_1$ and $\mathcal{D}_2$, decide whether $\mathcal{L}(\mathcal{D}_1 \cap \mathcal{D}_2) = \varnothing$.

***k*-cycle**

Given a k-partite graph $G = (V_1 \uplus \cdots \uplus V_k, E)$, decide if it contains a cycle of length k.
Hypothesis: $\forall \varepsilon > 0, \exists k \in \mathbb{N}$ not in time $\mathcal{O}(|E|^{2-\varepsilon})$

k-Cycle $\to$ Disjointness

Construction Given k-partite graph $G = (V_1 \uplus \cdots \uplus V_k, E)$, need to detect k-cycle.

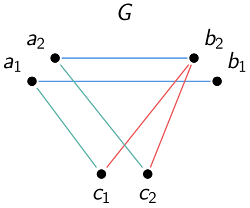
- $\mathcal{D}_1$ recognize paths $V_1 \to V_2 \cdots \to V_k \to V_1$,
- $\mathcal{D}_2$ recognize word of length $k+1$ starting and ending with the same vertex $v \in V_1$.

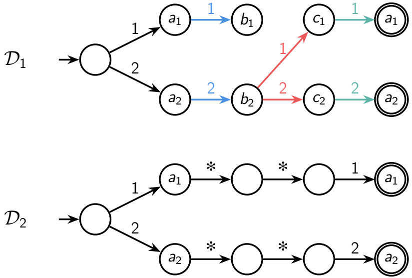
$\to$ Then any word $w \in \mathcal{L}(\mathcal{D}_1) \cap \mathcal{L}(\mathcal{D}_2)$ encodes a k-cycle in G.

Example

$k = 3$

$n = 2$





Size $\mathcal{O}(n+m)$ states and transitions

k-Cycle $\rightarrow$ Disjointness

Construction Given k-partite graph $G = (V_1 \uplus \cdots \uplus V_k, E)$, need to detect k-cycle.

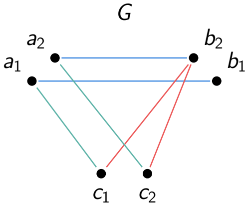
- $\mathcal{D}_1$ recognize paths $V_1 \rightarrow V_2 \cdots \rightarrow V_k \rightarrow V_1$,
- $\mathcal{D}_2$ recognize word of length $k+1$ starting and ending with the same vertex $v \in V_1$.

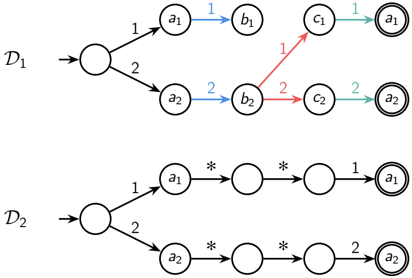
$\rightarrow$ Then any word $w \in \mathcal{L}(\mathcal{D}_1) \cap \mathcal{L}(\mathcal{D}_2)$ encodes a k-cycle in G.

Example

$k = 3$

$n = 2$





Issue

k-Cycle $\to$ Disjointness

Construction Given k-partite graph $G = (V_1 \uplus \cdots \uplus V_k, E)$, need to detect k-cycle.

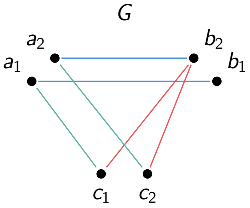
- $\mathcal{D}_1$ recognize paths $V_1 \to V_2 \cdots \to V_k \to V_1$,
- $\mathcal{D}_2$ recognize word of length $k+1$ starting and ending with the same vertex $v \in V_1$.

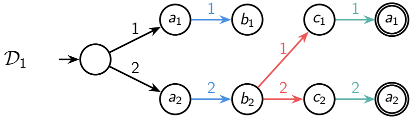
$\to$ Then any word $w \in \mathcal{L}(\mathcal{D}_1) \cap \mathcal{L}(\mathcal{D}_2)$ encodes a k-cycle in G.

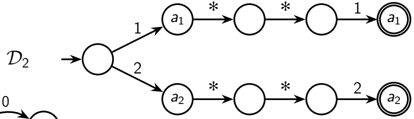
Example

$k = 3$

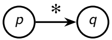
$n = 2$



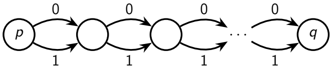




Fix



$\equiv$



k-Cycle $\rightarrow$ Disjointness

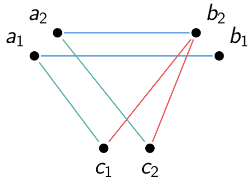
Construction Given k-partite graph $G = (V_1 \uplus \cdots \uplus V_k, E)$, need to detect k-cycle.

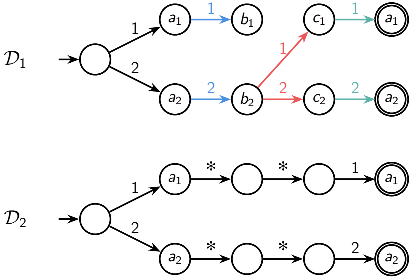
- $\mathcal{D}_1$ recognize paths $V_1 \rightarrow V_2 \cdots \rightarrow V_k \rightarrow V_1$,
- $\mathcal{D}_2$ recognize word of length $k+1$ starting and ending with the same vertex $v \in V_1$.

$\rightarrow$ Then any word $w \in \mathcal{L}(\mathcal{D}_1) \cap \mathcal{L}(\mathcal{D}_2)$ encodes a k-cycle in G.

Example
$k = 3$
$n = 2$

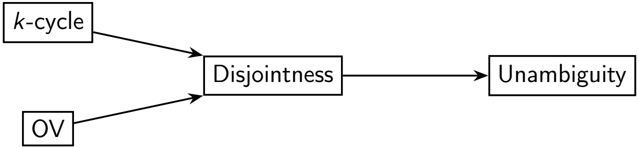
G





Size $\mathcal{O}((n+m)\log n)$ states and transitions

Lower-bounds



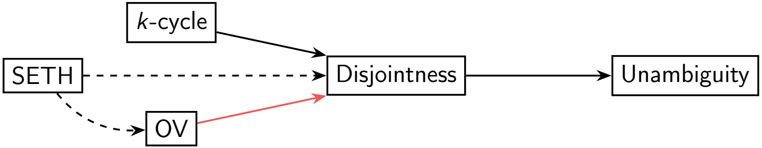
Problem (Disjointness)

Given two DFAs $\mathcal{D}_1$ and $\mathcal{D}_2$, decide whether $\mathcal{L}(\mathcal{D}_1 \cap \mathcal{D}_2) = \varnothing$.

***k*-cycle**

Given a k-partite graph $G = (V_1 \uplus \cdots \uplus V_k, E)$, decide if it contains a cycle of length k.
Hypothesis: $\forall \varepsilon > 0, \exists k \in \mathbb{N}$ not in time $\mathcal{O}(|E|^{2-\varepsilon})$

Lower-bounds



Problem (Disjointness)

Given two DFAs $\mathcal{D}_1$ and $\mathcal{D}_2$, decide whether $\mathcal{L}(\mathcal{D}_1 \cap \mathcal{D}_2) = \varnothing$.

Orthogonal Vector

Given sets $A, B \subseteq \{0,1\}^d$ of size n decide if

$$\exists a \in A, \exists b \in B : \ a \perp b.$$

Hypothesis: $\forall \varepsilon > 0, \exists c > 0$ not in time $\mathcal{O}(n^{2-\varepsilon})$ for $d = c \log n$

OV $\to$ Disjointness

Construction Given $A, B \subseteq \{0,1\}^d$ of size n, want to check whether

$$\exists a \in A, \exists b \in B : \ \forall \ell \in [d], a[\ell] = 0 \text{ or } b[\ell] = 0$$

Write $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_n\}$.
Let $\mathcal{D}_1$ and $\mathcal{D}_2$ recognize words $c_A c_B r_1 \ldots r_d$ such that:

- $c_A, c_B \in [n]$,
- $r_1, \ldots, r_d \in \{A, B\}$, and

$$\begin{array}{lll} (\text{for } \mathcal{D}_1) & \forall \ell \in [d] & : (r_\ell = A) \Rightarrow (a_{c_A}[\ell] = 0) \\ (\text{for } \mathcal{D}_2) & \forall \ell \in [d] & : (r_\ell = B) \Rightarrow (b_{c_B}[\ell] = 0) \end{array}$$

$\to$ Then any word $w \in \mathcal{L}(\mathcal{D}_1) \cap \mathcal{L}(\mathcal{D}_2)$ encodes a pair of orthogonal vectors in $A \times B$.

OV $\rightarrow$ Disjointness

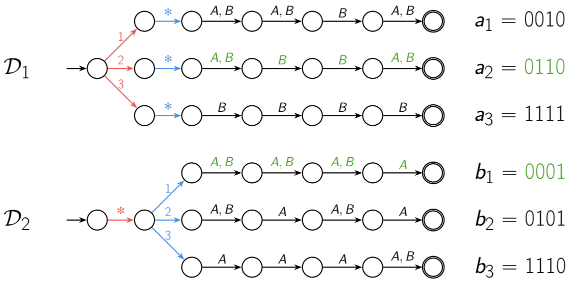
Construction Given $A, B \subseteq \{0,1\}^d$ of size n, want to check whether

$$\exists a \in A, \exists b \in B : \ \forall \ell \in [d], a[\ell] = 0 \text{ or } b[\ell] = 0$$

Example

$n = 3$

$d = 4$



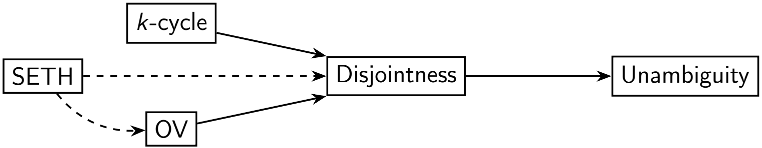
Size $\mathcal{O}(nd + n\log n)$ states and transitions

Recap

Given an NFA $\mathcal{A}$, is it unambiguous?

- Is decidable in n^2 time. [1991, 2001]
- Tight under OV or k-cycle. [our work]

$n = |Q| + |\delta| = \text{size of } \mathcal{A}$



Thank you!