On the Computational Complexity of Problems Related to Distinguishability Sets

Markus Holzer and Sebastian Jakobi

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

17th International Workshop on
Descriptional Complexity of Formal Systems
Waterloo, Ontario, Canada
June 27, 2015

Overview

Distinguishability Sets

distinguish between states of a DFA:

for states p, q, find a word w such that $\delta(p, w) \in F \iff \delta(q, w) \notin F$.

distinguish between words:

for words x, y find a word w such that $xw \in L \iff yw \notin L$.

Distinguishability Sets

distinguish between states of a DFA:

for states p, q, find a word w such that $\delta(p, w) \in F \iff \delta(q, w) \notin F$.

distinguish between words:

for words x, y find a word w such that $xw \in L \iff yw \notin L$.

[Câmpeanu, Moreira, Reis, 2014] study distinguishability sets

... for a DFA $A = (Q, \Sigma, \delta, q_0, F)$:

$$\mathsf{D}_A(p, q) = \{\, w \in \Sigma^* \mid \delta(p, w) \in F \Leftrightarrow \delta(q, w) \notin F \,\},\ \mathsf{D}(A) = \bigcup_{p,q \in Q} \mathsf{D}_A(p, q)$$

... and for a language $L \subseteq \Sigma^*$:

$$\mathsf{D}_L(x, y) = \{\, w \in \Sigma^* \mid xw \in L \Leftrightarrow yw \notin L \,\},\ \mathsf{D}(L) = \bigcup_{x,y \in \Sigma^*} \mathsf{D}_L(x, y)$$

Distinguishability Sets

distinguish between states of a DFA:

for states p, q, find a word w such that $\delta(p, w) \in F \iff \delta(q, w) \notin F$.

distinguish between words:

for words x, y find a word w such that $xw \in L \iff yw \notin L$.

[Câmpeanu, Moreira, Reis, 2014] study distinguishability sets

…for a DFA $A = (Q, \Sigma, \delta, q_0, F)$:

$$\mathsf{D}_A(p, q) = \{\, w \in \Sigma^* \mid \delta(p, w) \in F \Leftrightarrow \delta(q, w) \notin F \,\}, \ \mathsf{D}(A) = \bigcup_{p,q \in Q} \mathsf{D}_A(p, q)$$

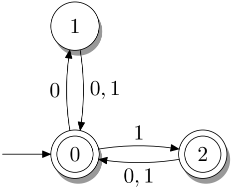
…and for a language $L \subseteq \Sigma^*$:

$$\mathsf{D}_L(x, y) = \{\, w \in \Sigma^* \mid xw \in L \Leftrightarrow yw \notin L \,\}, \ \mathsf{D}(L) = \bigcup_{x,y \in \Sigma^*} \mathsf{D}_L(x, y)$$

If all states in A are reachable and $L = L(A)$ then $\mathsf{D}(L) = \mathsf{D}(A)$.

An Example (from [CMR14])

A:

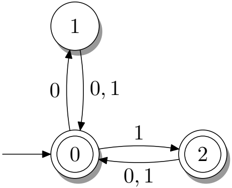


$$L(A) = (\{0,1\}^2)^* \cup \{0,1\}^*\{1\}$$

$$\mathsf{D}_A(1,0) = \{\lambda\} \cup \{0,1\}(\{0,1\}^2)^*\{0\} \\ \cup (\{0,1\}^2)^*\{0\}$$

An Example (from [CMR14])

A:

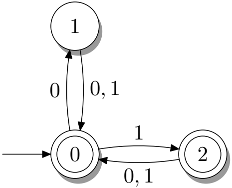


$$L(A) = (\{0,1\}^2)^* \cup \{0,1\}^*\{1\}$$

$$\begin{aligned} \mathsf{D}_A(1,0) &= \{\lambda\} \cup \{0,1\}(\{0,1\}^2)^*\{0\} \\ &\qquad \cup (\{0,1\}^2)^*\{0\} \\ \mathsf{D}_A(1,2) &= \{\lambda\} \\ \mathsf{D}_A(0,2) &= \mathsf{D}_A(1,0) \setminus \{\lambda\} \end{aligned}$$

An Example (from [CMR14])

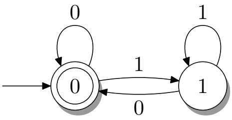
A:



$$L(A) = (\{0,1\}^2)^* \cup \{0,1\}^*\{1\}$$

$$\begin{aligned}
\mathsf{D}_A(1,0) &= \{\lambda\} \cup \{0,1\}(\{0,1\}^2)^*\{0\} \\
&\qquad \cup (\{0,1\}^2)^*\{0\} \\
\mathsf{D}_A(1,2) &= \{\lambda\} \\
\mathsf{D}_A(0,2) &= \mathsf{D}_A(1,0) \setminus \{\lambda\}
\end{aligned}$$

$\mathsf{D}(A)$:



$$\mathsf{D}(A) = \{\lambda\} \cup \{0,1\}^*\{0\}$$

Properties of Distinguishability Sets [CMR14]

$$\mathsf{D}_L(x,y) = \{\, w \in \Sigma^* \mid xw \in L \Leftrightarrow yw \notin L \,\},\ \mathsf{D}(L) = \bigcup_{x,y\in\Sigma^*} \mathsf{D}_L(x,y)$$

Alternative characterization of $\mathsf{D}(L)$:

$$\mathsf{D}(L) = \mathsf{suff}(L) \cap \mathsf{suff}(\overline{L}),$$

where $\mathsf{suff}(L) = \{\, v \in \Sigma^* \mid uv \in L \text{ for some } u \in \Sigma^* \,\}$ are the suffixes of words from L and $\overline{L} = \Sigma^* \setminus L$ is the complement of L.

Properties of Distinguishability Sets [CMR14]

$$\mathsf{D}_L(x,y) = \{\, w \in \Sigma^* \mid xw \in L \Leftrightarrow yw \notin L \,\},\ \mathsf{D}(L) = \bigcup_{x,y\in\Sigma^*} \mathsf{D}_L(x,y)$$

Alternative characterization of $\mathsf{D}(L)$:

$$\mathsf{D}(L) = \mathsf{suff}(L) \cap \mathsf{suff}(\overline{L}),$$

where $\mathsf{suff}(L) = \{\, v \in \Sigma^* \mid uv \in L \text{ for some } u \in \Sigma^* \,\}$ are the suffixes of words from L and $\overline{L} = \Sigma^* \setminus L$ is the complement of L.

- $\mathsf{D}(L)$ is suffix-closed.
- If L is regular then $\mathsf{D}(L)$ is regular.
  The worst case state complexity of $\mathsf{D}(L)$ is $2^n - n$

Properties of Distinguishability Sets [CMR14]

$$\mathsf{D}_L(x,y) = \{\, w \in \Sigma^* \mid xw \in L \Leftrightarrow yw \notin L \,\},\ \mathsf{D}(L) = \bigcup_{x,y\in\Sigma^*} \mathsf{D}_L(x,y)$$

Alternative characterization of $\mathsf{D}(L)$:

$$\mathsf{D}(L) = \mathsf{suff}(L) \cap \mathsf{suff}(\overline{L}),$$

where $\mathsf{suff}(L) = \{\, v \in \Sigma^* \mid uv \in L \text{ for some } u \in \Sigma^* \,\}$ are the suffixes of words from L and $\overline{L} = \Sigma^* \setminus L$ is the complement of L.

- $\mathsf{D}(L)$ is suffix-closed.
- If L is regular then $\mathsf{D}(L)$ is regular.
  The worst case state complexity of $\mathsf{D}(L)$ is $2^n - n$

Problem: D-SET-SIZE

Given a DFA A and an integer k, decide whether $\mathsf{sc}(\mathsf{D}(L(A))) \leq k$.

Overview

Deciding the State Complexity of $\mathsf{D}(L)$

Problem: D-Set-Size

Given a DFA A and an integer k, decide whether $\mathsf{sc}(\mathsf{D}(L(A))) \leq k$.

Lemma

D-Set-Size *is contained in* PSPACE*.*

Deciding $\mathsf{sc}(L(B)) \leq k$ for a DFA B is NL-complete.

Given an n-state DFA A, a DFA A_{D} for $\mathsf{D}(L(A))$ can be constructed with at most $2^n - n$ states. [CMR14]

Construct A_{D} on-the-fly and use the NL-algorithm on it.

Deciding the State Complexity of $\mathsf{D}(L)$

Problem: D-Set-Size

Given a DFA A and an integer k, decide whether $\mathrm{sc}(\mathsf{D}(L(A))) \leq k$.

Lemma

D-Set-Size *is contained in* PSPACE.

Deciding $\mathrm{sc}(L(B)) \leq k$ for a DFA B is NL-complete.

Given an n-state DFA A, a DFA A_{D} for $\mathsf{D}(L(A))$ can be constructed with at most $2^n - n$ states. [CMR14]

Construct A_{D} on-the-fly and use the NL-algorithm on it.

Lemma

D-Set-Size *is* PSPACE*-hard.*

D-Set-Size is PSPACE-hard

Reduction from the PSPACE-complete problem

Problem: DFA-Union-Universality

Given DFAs $A_1, A_2, \ldots, A_n$ with common input alphabet Σ, decide whether $\bigcup_{i=1}^{n} L(A_i) = \Sigma^*$.

D-SET-SIZE is PSPACE-hard

Reduction from the PSPACE-complete problem

Problem: DFA-UNION-UNIVERSALITY

Given DFAs $A_1, A_2, \ldots, A_n$ with common input alphabet Σ, decide whether $\bigcup_{i=1}^{n} L(A_i) = \Sigma^*$.

- We may assume $\{\lambda\} \cup \Sigma \subseteq \bigcup_{i=1}^{n} L(A_i)$ and add DFAs
  - A_{n+1} for language $\{\lambda\}$
  - A_{n+2} for language Σ
- Minimize DFAs $A_1, A_2, \ldots, A_{n+2}$ and (polynomial time reduction)
- add reset symbol $\#$ that takes A_i back to its initial state.

Obtain $A'_1, A'_2 \ldots, A'_{n+2}$ with $A'_i = (Q'_i, \Sigma_\#, \delta'_i, s'_i, F'_i)$, $\Sigma_\# = \Sigma \cup \{\#\}$

$$\bigcup_{i=1}^{n} L(A_i) = \Sigma^* \iff \bigcup_{i=1}^{n+2} L(A'_i) = \Sigma_\#^*$$

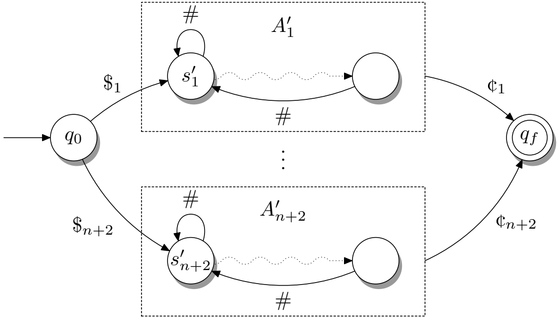
D-Set-Size is PSPACE-hard (2)

Combine the DFAs $A'_i = (Q'_i, \Sigma_\#, \delta'_i, s'_i, F'_i)$, $1 \leq i \leq n+2$, to one DFA $A = (Q, \Gamma, \delta, q_0, F)$:

$$\Gamma = \Sigma_\# \cup \{\, \$_i, ¢_i \mid 1 \leq i \leq n+2 \,\}$$

$$Q = \bigcup_{i=1}^{n+2} Q'_i \cup \{q_0, q_f, q_s\},$$

$$F = \bigcup_{i=1}^{n+2} F'_i \cup \{q_f\},$$



D-Set-Size is PSPACE-hard (2)

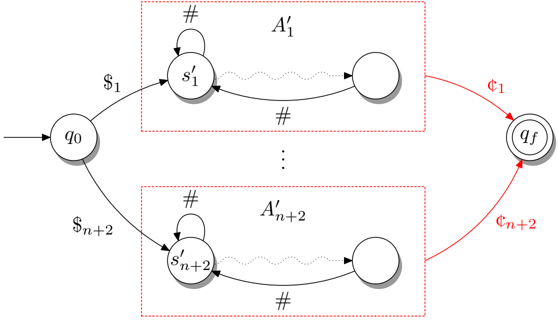
Combine the DFAs $A'_i = (Q'_i, \Sigma_\#, \delta'_i, s'_i, F'_i)$, $1 \leq i \leq n+2$, to one DFA $A = (Q, \Gamma, \delta, q_0, F)$:

$\Gamma = \Sigma_\# \cup \{\, \$_i, ¢_i \mid 1 \leq i \leq n+2 \,\}$

$Q = \bigcup_{i=1}^{n+2} Q'_i \cup \{q_0, q_f, q_s\}$,

$F = \bigcup_{i=1}^{n+2} F'_i \cup \{q_f\}$,

for all $q_i \in Q'_i$:
$\delta(q_i, ¢_i) = q_f$



D-Set-Size is PSPACE-hard (2)

Combine the DFAs $A'_i = (Q'_i, \Sigma_\#, \delta'_i, s'_i, F'_i)$, $1 \leq i \leq n+2$, to one DFA $A = (Q, \Gamma, \delta, q_0, F)$:

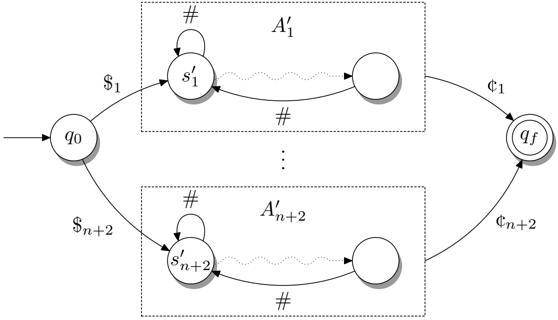
$\Gamma = \Sigma_\# \cup \{\, \$_i, ¢_i \mid 1 \leq i \leq n+2 \,\}$

$Q = \bigcup_{i=1}^{n+2} Q'_i \cup \{q_0, q_f, q_s\}$,

$F = \bigcup_{i=1}^{n+2} F'_i \cup \{q_f\}$,

for all $q_i \in Q'_i$:
$\delta(q_i, ¢_i) = q_f$

target number of states: $k = |Q| + 1$



Claim:

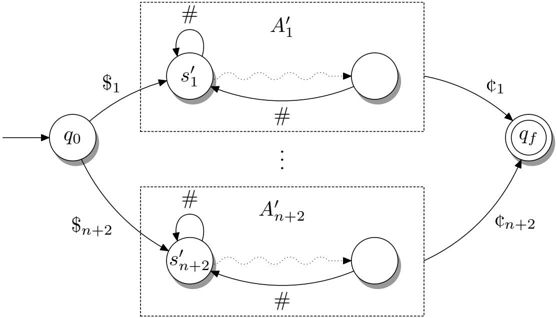
$\mathrm{sc}(\mathrm{D}(L(A))) \leq k$
if and only if
$\bigcup_{i=1}^{n+2} L(A'_i) = \Sigma_\#^*$

D-Set-Size is PSPACE-hard (3)

$$L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^{*} ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_{\#}^{*} \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^{*} ¢_i .$$

Let $w \in \mathsf{D}(L)$, then $\delta(q, w) \in F$ for some $q \in Q$

- if $q = q_0$: $w \in L$

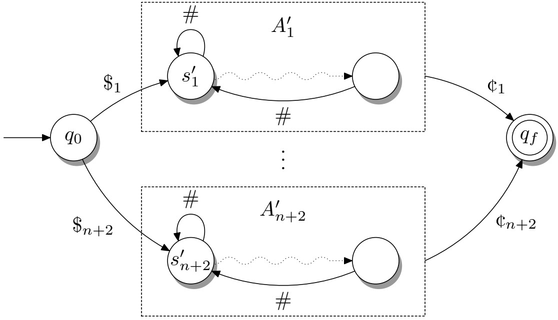


D-Set-Size is PSPACE-hard (3)

$$L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^{*} ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_{\#}^{*} \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^{*} ¢_i.$$

Let $w \in \mathsf{D}(L)$, then $\delta(q, w) \in F$ for some $q \in Q$

- if $q = q_0$: $w \in L$
- if $q = q_f$: $w = \lambda \in \Sigma_{\#}^{*}$

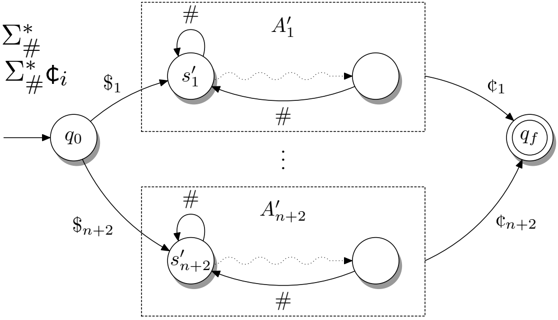


D-SET-SIZE is PSPACE-hard (3)

$$L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_\#^* \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i.$$

Let $w \in \mathsf{D}(L)$, then $\delta(q, w) \in F$ for some $q \in Q$

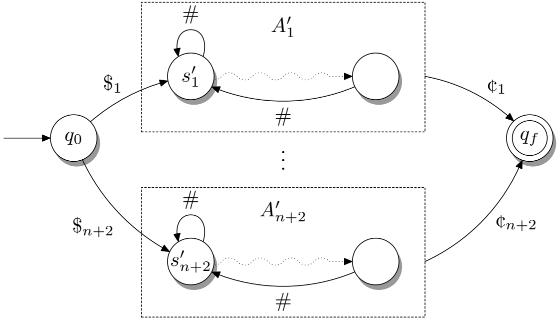
- if $q = q_0$: $w \in L$
- if $q = q_f$: $w = \lambda \in \Sigma_\#^*$
- if $q \in Q_i$:
  - if $\delta(q, w) \in F_i$: $w \in \Sigma_\#^*$
  - if $\delta(q, w) = q_f$: $w \in \Sigma_\#^* ¢_i$



D-SET-SIZE is PSPACE-hard (3)

$$L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_\#^* \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i.$$

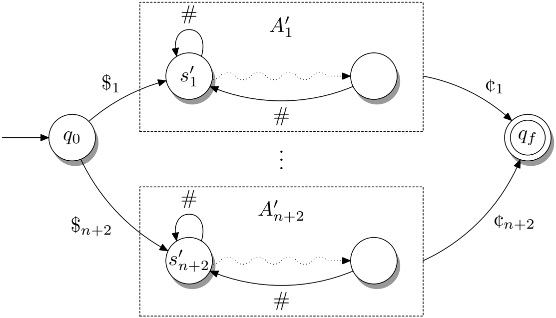
- $L \subseteq \mathsf{D}(L)$:
  A has a sink state.



D-Set-Size is PSPACE-hard (3)

$$L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_\#^* \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i.$$

- $L \subseteq \mathsf{D}(L)$:
  A has a sink state.
- $\bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i \subseteq \mathsf{D}(L)$:
  $\bigcup_{i=1}^{n+2} \$_i \Sigma_\#^* ¢_i \subseteq L \subseteq \mathsf{D}(L)$,
  and $\mathsf{D}(L)$ is suffix-closed.



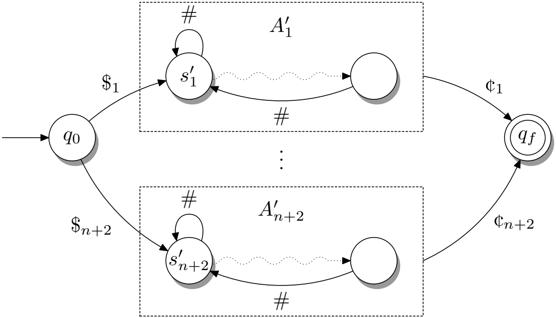
D-Set-Size is PSPACE-hard (3)

$$\Sigma_{\#}^* \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^* ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_{\#}^* \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^* ¢_i.$$

- $L \subseteq \mathsf{D}(L)$:
  A has a sink state.
- $\bigcup_{i=1}^{n+2} \Sigma_{\#}^* ¢_i \subseteq \mathsf{D}(L)$:
  $\bigcup_{i=1}^{n+2} \$_i \Sigma_{\#}^* ¢_i \subseteq L \subseteq \mathsf{D}(L)$,
  and $\mathsf{D}(L)$ is suffix-closed.

If $\bigcup_{i=1}^{n+2} L(A'_i) = \Sigma_{\#}^*$:

- $\Sigma_{\#}^* \subseteq \mathsf{D}(L)$:
  $\bigcup_{i=1}^{n+2} \$_i L(A'_i) \subseteq L \subseteq \mathsf{D}(L)$,
  and $\mathsf{D}(L)$ is suffix-closed.



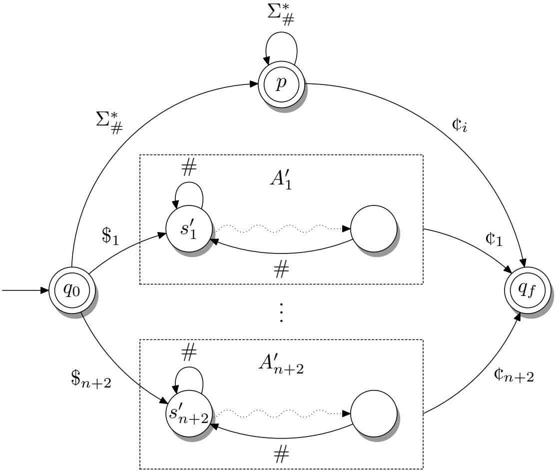
D-Set-Size is PSPACE-hard (3)

$$\Sigma_\#^* \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_\#^* \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i.$$

- $L \subseteq \mathsf{D}(L)$:
  A has a sink state.
- $\bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i \subseteq \mathsf{D}(L)$:
  $\bigcup_{i=1}^{n+2} \$_i \Sigma_\#^* ¢_i \subseteq L \subseteq \mathsf{D}(L)$,
  and $\mathsf{D}(L)$ is suffix-closed.

If $\bigcup_{i=1}^{n+2} L(A_i') = \Sigma_\#^*$:

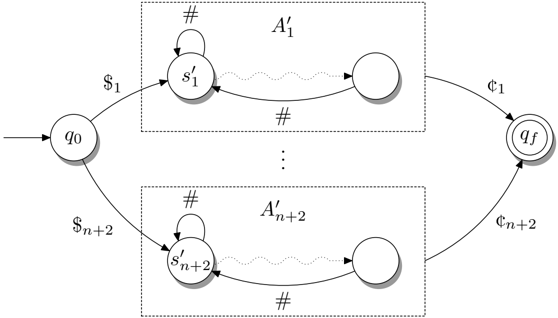
- $\Sigma_\#^* \subseteq \mathsf{D}(L)$:
  $\bigcup_{i=1}^{n+2} \$_i L(A_i') \subseteq L \subseteq \mathsf{D}(L)$,
  and $\mathsf{D}(L)$ is suffix-closed.



D-SET-SIZE is PSPACE-hard (4)

$$L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^{*} ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_{\#}^{*} \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^{*} ¢_i.$$

Let $w \in \Sigma_{\#}^{*}$ with $w \notin \bigcup_{i=1}^{n+2} L(A'_i)$ and B be a DFA with $L(B) = \mathsf{D}(L)$.



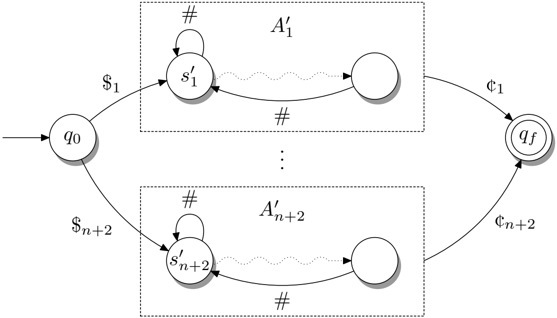
D-Set-Size is PSPACE-hard (4)

$$L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_\#^* \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_\#^* ¢_i.$$

Let $w \in \Sigma_\#^*$ with $w \notin \bigcup_{i=1}^{n+2} L(A'_i)$ and B be a DFA with $L(B) = \mathsf{D}(L)$.

Because $\$_i^{-1}\mathsf{D}(L) = \$_i^{-1}L$ and all A'_i are minimal, B "contains" A.

$L(A'_{n+1}) = \{\lambda\}$, $L(A'_{n+2}) = \Sigma \quad \Longrightarrow \quad \Sigma \subseteq \mathsf{D}(L)$



D-Set-Size is PSPACE-hard (4)

$$L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^{*} ¢_i \quad \subseteq \quad \mathsf{D}(L) \quad \subseteq \quad \Sigma_{\#}^{*} \cup L \cup \bigcup_{i=1}^{n+2} \Sigma_{\#}^{*} ¢_i.$$

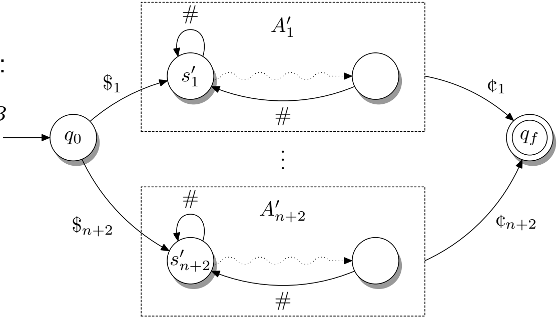
Let $w \in \Sigma_{\#}^{*}$ with $w \notin \bigcup_{i=1}^{n+2} L(A'_i)$ and B be a DFA with $L(B) = \mathsf{D}(L)$.

Because $\$_i^{-1}\mathsf{D}(L) = \$_i^{-1}L$ and all A'_i are minimal, B "contains" A.

$L(A'_{n+1}) = \{\lambda\}$, $L(A'_{n+2}) = \Sigma \quad \Longrightarrow \quad \Sigma \subseteq \mathsf{D}(L)$

One can show that B needs at least two additional states:

- $p_1 = \delta_B(q_{0,B}, \#w) \notin F_B$
- $p_2 = \delta_B(q_{0,B}, a) \in F_B$ for some $a \in \Sigma$.



D-SET-SIZE is PSPACE-complete

Theorem

D-SET-SIZE *is* PSPACE-*complete.*

Remark

Also logspace-reduction is possible:
Instead of minimizing the input DFAs A_i in the union universality instance, use additional symbols to ensure their minimality.

Recognizing Representations of $\mathsf{D}(L)$

Problem: $\textsc{L-Versus-D}$

Given two DFAs A and B with $L = L(A)$ and $L' = L(B)$, is $L' = \mathsf{D}(L)$?

Theorem

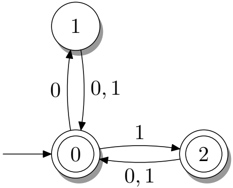
$\textsc{L-Versus-D}$ *is* PSPACE-*complete.*

Remark

Deciding $L' \supseteq \mathsf{D}(L)$ is NL-complete and $L' \subseteq \mathsf{D}(L)$ is PSPACE-complete.

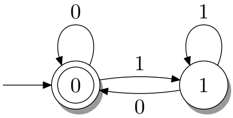
An Example (from [CMR14])

L:



$$L = (\{0,1\}^2)^* \cup \{0,1\}^*\{1\}$$

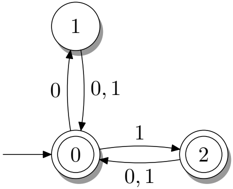
$\mathsf{D}(L)$:



$$\mathsf{D}(L) = \{\lambda\} \cup \{0,1\}^*\{0\}$$

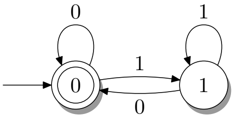
An Example (from [CMR14])

L:



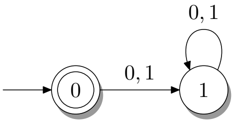
$L = (\{0,1\}^2)^* \cup \{0,1\}^*\{1\}$

$\mathsf{D}(L)$:



$\mathsf{D}(L) = \{\lambda\} \cup \{0,1\}^*\{0\}$

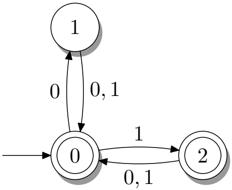
$\mathsf{D}^2(L)$:



$\mathsf{D}^2(L) = \mathsf{D}(\mathsf{D}(L)) = \{\lambda\}$

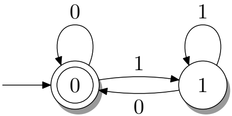
An Example (from [CMR14])

L:



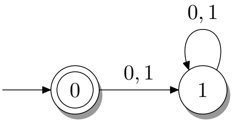
$L = (\{0,1\}^2)^* \cup \{0,1\}^*\{1\}$

$\mathsf{D}(L)$:



$\mathsf{D}(L) = \{\lambda\} \cup \{0,1\}^*\{0\}$

$\mathsf{D}^2(L)$:



$\mathsf{D}^2(L) = \mathsf{D}(\mathsf{D}(L)) = \{\lambda\}$

$\mathsf{D}^3(L)$:

$\mathsf{D}^2(L) = \mathsf{D}^3(L) = \ldots = \{\lambda\}$

The Hierarchy $L, \mathsf{D}(L), \mathsf{D}^2(L), \ldots$

We know from [Câmpeanu, Moreira, Reis, 2014]:

$$\mathsf{D}(L) \supseteq \mathsf{D}^2(L) = \mathsf{D}^{2+i}(L) \text{ for } i \geq 0,$$

and
- $L \supseteq \mathsf{D}(L)$ if L suffix-closed
- $L \subseteq \mathsf{D}(L)$ if L has $\emptyset$ as a quotient

Interestingly, theses are only *sufficient* conditions ($(a^n)^* \subseteq \mathsf{D}((a^n)^*)$), but:

$L = \mathsf{D}(L)$ *if and only if* L is suffix closed and has $\emptyset$ as a quotient.

The Hierarchy $L, \mathsf{D}(L), \mathsf{D}^2(L), \ldots$

We know from [Câmpeanu, Moreira, Reis, 2014]:

$$\mathsf{D}(L) \supseteq \mathsf{D}^2(L) = \mathsf{D}^{2+i}(L) \text{ for } i \geq 0,$$

and

- $L \supseteq \mathsf{D}(L)$ if L suffix-closed
- $L \subseteq \mathsf{D}(L)$ if L has $\emptyset$ as a quotient

Interestingly, theses are only *sufficient* conditions $((a^n)^* \subseteq \mathsf{D}((a^n)^*))$, but:

$L = \mathsf{D}(L)$ *if and only if* L is suffix closed and has $\emptyset$ as a quotient.

Decision Problems on the Hierarchy

Given a DFA A with $L = L(A)$,

is $L = \mathsf{D}(L)$?	L-EQUALS-D
is $\mathsf{D}(L) = \mathsf{D}^2(L)$?	D-EQUALS-DSQUARE
is $L = \mathsf{D}^2(L)$?	L-EQUALS-DSQUARE

L-EQUALS-D is NL-complete

Theorem

L-EQUALS-D *is* NL*-complete.*

$L = \mathsf{D}(L)$ *if and only if* L is suffix closed and has $\emptyset$ as a quotient. [CMR14]
Both properties can be checked in NL $\Longrightarrow$ containment in NL follows.

NL-hardness by a reduction from the graph reachability problem.

D-EQUALS-DSQUARE and Synchronization

A (not necessarily regular) language $L \subseteq \Sigma$ is *language-synchronizing* if there exists a *language-reset word* $w \in \Sigma^*$ such that

$$\text{for all } x, y, z \in \Sigma^*\text{: } xwz \in L \text{ if and only if } ywz \in L.$$

Theorem

A language $L \subseteq \Sigma^*$ is language-synchronizing if and only if $\mathsf{D}(L) = \mathsf{D}^2(L)$.

D-Equals-DSquare and Synchronization

A (not necessarily regular) language $L \subseteq \Sigma$ is *language-synchronizing* if there exists a *language-reset word* $w \in \Sigma^*$ such that

$$\text{for all } x, y, z \in \Sigma^*\text{: } xwz \in L \text{ if and only if } ywz \in L.$$

Theorem

A language $L \subseteq \Sigma^*$ is language-synchronizing if and only if $\mathsf{D}(L) = \mathsf{D}^2(L)$.

L is language-synchronizing with language-reset word w if and only if:

$$\text{for all } x, y, z \in \Sigma^*\text{: } wz \in x^{-1}L \text{ if and only if } wz \in y^{-1}L$$

$$\text{for all } z \in \Sigma^*\text{: } wz \notin \mathsf{D}(L)$$

$$w^{-1}\mathsf{D}(L) = \emptyset$$

$\textsc{D-Equals-DSquare}$ and Synchronization

A (not necessarily regular) language $L \subseteq \Sigma$ is *language-synchronizing* if there exists a *language-reset word* $w \in \Sigma^*$ such that

$$\text{for all } x, y, z \in \Sigma^*\text{: } xwz \in L \text{ if and only if } ywz \in L.$$

Theorem

A language $L \subseteq \Sigma^*$ is language-synchronizing if and only if $\mathsf{D}(L) = \mathsf{D}^2(L)$.

L is language-synchronizing with language-reset word w if and only if:

$$\text{for all } x, y, z \in \Sigma^*\text{: } wz \in x^{-1}L \text{ if and only if } wz \in y^{-1}L$$

$$\text{for all } z \in \Sigma^*\text{: } wz \notin \mathsf{D}(L)$$

$$w^{-1}\mathsf{D}(L) = \emptyset$$

$\mathsf{D}(L) = \mathsf{D}^2(L)$ if and only if $\mathsf{D}(L)$ has $\emptyset$ as a quotient. [CMR14]

D-EQUALS-DSQUARE is NL-complete

A DFA $A = (Q, \Sigma, \delta, q_0, F)$ is *synchronizing* if there exists $w \in \Sigma^*$, $q \in Q$ such that $\delta(p, w) = q$ for all $p \in Q$.

Theorem

A regular language $L \subseteq \Sigma^*$ is language-synchronizing if and only if the minimal DFA for L is synchronizing.

D-EQUALS-DSQUARE is NL-complete

A DFA $A = (Q, \Sigma, \delta, q_0, F)$ is *synchronizing* if there exists $w \in \Sigma^*$, $q \in Q$ such that $\delta(p, w) = q$ for all $p \in Q$.

Theorem

A regular language $L \subseteq \Sigma^*$ is language-synchronizing if and only if the minimal DFA for L is synchronizing.

Theorem

Deciding for a given DFA A, whether it is synchronizing, is NL*-complete (even if the problem instances are restricted to minimal DFAs).*

D-EQUALS-DSQUARE is NL-complete

A DFA $A = (Q, \Sigma, \delta, q_0, F)$ is *synchronizing* if there exists $w \in \Sigma^*$, $q \in Q$ such that $\delta(p, w) = q$ for all $p \in Q$.

Theorem

A regular language $L \subseteq \Sigma^*$ is language-synchronizing if and only if the minimal DFA for L is synchronizing.

Theorem

Deciding for a given DFA A, whether it is synchronizing, is NL*-complete (even if the problem instances are restricted to minimal DFAs).*

Theorem

D-EQUALS-DSQUARE *is* NL*-complete.*

L-Equals-DSquare is NL-complete

Theorem

L-Equals-DSquare *is* NL-*complete.*

Observe that $L = \mathsf{D}^2(L)$ if and only if $L = \mathsf{D}(L)$:

L-EQUALS-DSQUARE is NL-complete

Theorem

L-EQUALS-DSQUARE *is* NL-*complete.*

Observe that $L = \mathsf{D}^2(L)$ if and only if $L = \mathsf{D}(L)$:

Assume $L = \mathsf{D}^2(L)$, apply D $\quad\Longrightarrow\quad \mathsf{D}(L) = \mathsf{D}^3(L)$

Because $\mathsf{D}^2(L) = \mathsf{D}^3(L)$, we obtain $\mathsf{D}(L) = \mathsf{D}^2(L) = L$.

L-EQUALS-DSQUARE is NL-complete

Theorem

L-EQUALS-DSQUARE *is* NL*-complete.*

Observe that $L = \mathsf{D}^2(L)$ if and only if $L = \mathsf{D}(L)$:

Assume $L = \mathsf{D}^2(L)$, apply D $\implies$ $\mathsf{D}(L) = \mathsf{D}^3(L)$

Because $\mathsf{D}^2(L) = \mathsf{D}^3(L)$, we obtain $\mathsf{D}(L) = \mathsf{D}^2(L) = L$.

Conversely assume $L = \mathsf{D}(L)$, apply D $\implies$ $\mathsf{D}(L) = \mathsf{D}^2(L)$

We obtain $L = \mathsf{D}(L) = \mathsf{D}^2(L)$.

Overview

Conclusion

Summary

- PSPACE-complete decision problems even for DFAs:
  - D-SET-SIZE
  - L-VERSUS-D
- NL-complete decision problems on the hierarchy of $\mathsf{D}^i(L)$
  - link to synchronizing DFAs

Further Research

- Use prefixes or infixes instead of suffixes:

  $\mathsf{suff}(L) \cap \mathsf{suff}(\overline{L}), \qquad \mathsf{pref}(L) \cap \mathsf{pref}(\overline{L}), \qquad \mathsf{infix}(L) \cap \mathsf{infix}(\overline{L})$
- Use quasi-lexicographically minimal distinguishing words: [CMR14]

$$\underline{\mathsf{D}}_L(x,y) = \min\{\, w \mid w \in \mathsf{D}_L(x,y)\,\}, \quad \underline{\mathsf{D}}(L) = \{\, \underline{\mathsf{D}}_L(x,y) \mid x \not\equiv_L y \,\}$$

The $\underline{\mathsf{D}}^i(L)$-hierarchy is finite for every L, but fixed point may vary.

Thank you for your attention!