The Big-O Problem for Weighted Automata

THE UNIVERSITY OF WARWICK
Dmitry Chistikov

UNIVERSITY OF OXFORD
Stefan Kiefer

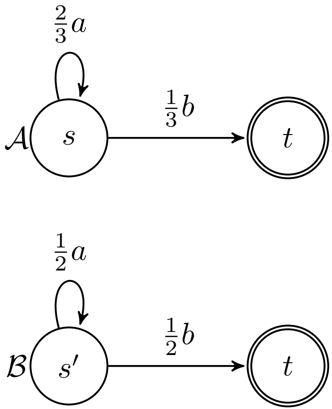
UNIVERSITY OF OXFORD
Andrzej Murawski

MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS
David Purser

Appeared at CONCUR'20

Model: Non-negative weighted automata

- $(\mathbb{Q}_{\geq 0}, +, \times)$-semi-ring
- LMCs (or HMMs), Probabilistic Automata
- From $\mathcal{A}$ gives weight $\mathcal{A}(w)$ to $w \in \Sigma^*$:
- $\mathcal{A}(a^n\ b) = \left(\frac{2}{3}\right)^n \cdot \frac{1}{3}$

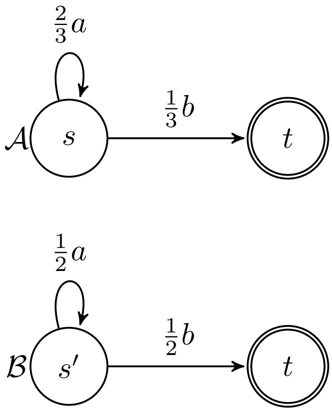


Big-O Problem

Definition (Big-O)

$\mathcal{A}$ is big-O $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

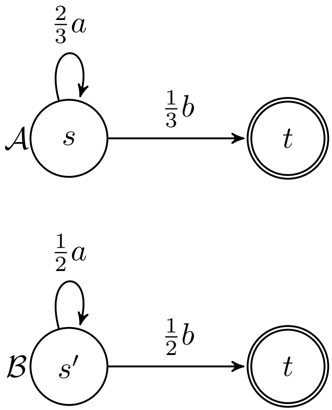


Big-O Problem

Definition (Big-O)

f is big-O g if $\exists c > 0$, s.t. $\forall n \in \mathbb{N}$

$$f(n) \leq c \cdot g(n)$$



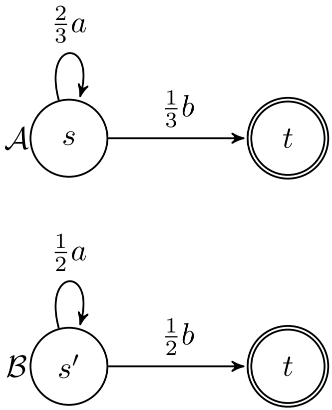
Big-O Problem

Definition (Big-O)

$\mathcal{A}$ is big-O $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

Problem: decide if $\mathcal{A}$ big-O $\mathcal{B}$



Big-O Problem

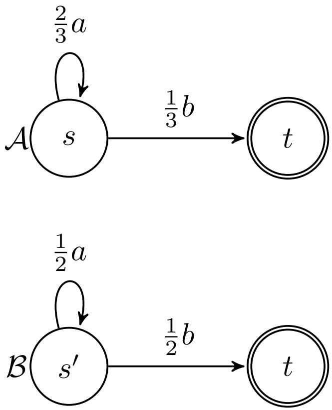
Definition (Big-O)

$\mathcal{A}$ is big-O $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

Problem: decide if $\mathcal{A}$ big-O $\mathcal{B}$

$$\frac{\frac{2}{3}^n \cdot \frac{1}{3}}{\frac{1}{2}^n \cdot \frac{1}{2}} \to \infty$$



Big-O Problem

Definition (Big-O)

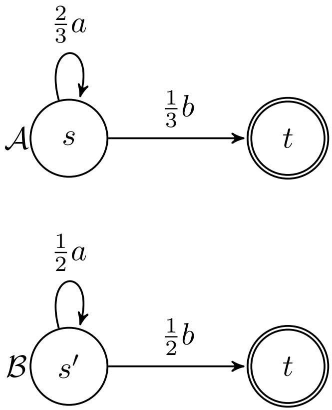
$\mathcal{A}$ is big-O $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

Problem: decide if $\mathcal{A}$ big-O $\mathcal{B}$

$$\frac{\frac{2}{3}^n \cdot \frac{1}{3}}{\frac{1}{2}^n \cdot \frac{1}{2}} \to \infty$$

Relaxation of containment



Summary

Definition (Big-O)

$\mathcal{A}$ is big-O of $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

Results:

- Big-O Problem is undecidable
- On unary weighted automata Big-O Problem is coNP-complete
- The relation to Total Variation Distance
- The relation to Differential Privacy
- Bounded Languages

The language containment condition

$$\begin{array}{rlll} \mathcal{A} \text{ is big-O } \mathcal{B} & \text{if } \exists c > 0, \text{ s.t.} & \forall w \in \Sigma^* & \mathcal{A}(w) \leq c \cdot \mathcal{B}(w) \\ \mathcal{A} \text{ is eventually-big-O } \mathcal{B} & \text{if } \exists c > 0, k > 0, \text{ s.t.} & \forall w \in \Sigma^{\geq k} & \mathcal{A}(w) \leq c \cdot \mathcal{B}(w) \end{array}$$

The language containment condition

$\mathcal{A}$ is big-O $\mathcal{B}$	if $\exists c > 0$, s.t.	$\forall w \in \Sigma^*$	$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$
~~$\mathcal{A}$ is eventually-big-O $\mathcal{B}$~~	~~if $\exists c > 0, k > 0$, s.t.~~	~~$\forall w \in \Sigma^{>k}$~~	~~$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$~~

Single w with $\mathcal{A}(w) > 0$ and $\mathcal{B}(w) = 0$ then $\mathcal{A}$ is NOT big-O of $\mathcal{B}$.

The language containment condition

$\mathcal{A}$ is big-O $\mathcal{B}$	if $\exists c > 0$, s.t.	$\forall w \in \Sigma^*$	$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$
~~$\mathcal{A}$ is eventually-big-O $\mathcal{B}$~~	~~if $\exists c > 0, k > 0$, s.t.~~	~~$\forall w \in \Sigma^{>k}$~~	~~$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$~~

Single w with $\mathcal{A}(w) > 0$ and $\mathcal{B}(w) = 0$ then $\mathcal{A}$ is NOT big-O of $\mathcal{B}$.

To prevent this, the *language containment condition* requires $L(\mathcal{A}) \subseteq L(\mathcal{B})$.

The language containment condition

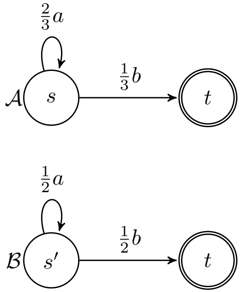
$\mathcal{A}$ is big-O $\mathcal{B}$	if $\exists c > 0$, s.t.	$\forall w \in \Sigma^*$	$\mathcal{A}(w) \le c \cdot \mathcal{B}(w)$
~~$\mathcal{A}$ is eventually-big-O $\mathcal{B}$~~	~~if $\exists c > 0, k > 0$, s.t.~~	~~$\forall w \in \Sigma^{>k}$~~	~~$\mathcal{A}(w) \le c \cdot \mathcal{B}(w)$~~

Single w with $\mathcal{A}(w) > 0$ and $\mathcal{B}(w) = 0$ then $\mathcal{A}$ is NOT big-O of $\mathcal{B}$.

To prevent this, the *language containment condition* requires $L(\mathcal{A}) \subseteq L(\mathcal{B})$.

The condition is necessary but not sufficient:

$$L(\mathcal{A}) = L(\mathcal{B}) = \{a^n b \mid n \in \mathbb{N}\}$$



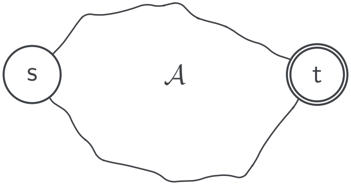
Undecidability

Theorem

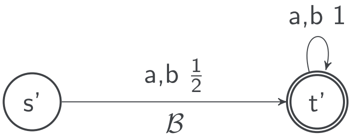
The Big-O Problem is undecidable.

Reduce from Emptiness of Probabilistic Automata ($\exists w \ \mathcal{A}(w) > \frac{1}{2}$)

Reduction from $\exists w\ \mathcal{A}(w) > \frac{1}{2}$?

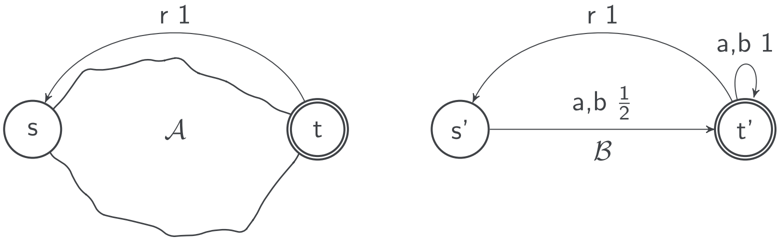


$$\mathcal{A}(w)$$



$$\mathcal{B}(w) = \frac{1}{2}\ \forall w \in \{a, b\}^+$$

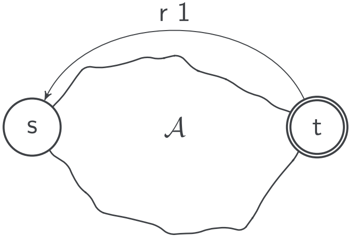
Reduction from $\exists w\ \mathcal{A}(w) > \frac{1}{2}$?

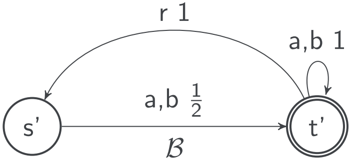


$\mathcal{A}(w)$

$\mathcal{B}(w) = \frac{1}{2}\ \forall w \in \{a, b\}^+$

Reduction from $\exists w\ \mathcal{A}(w) > \frac{1}{2}$?





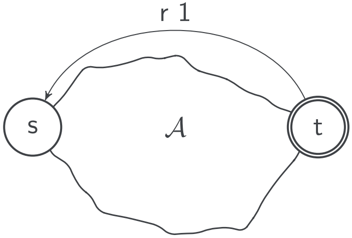
$\mathcal{A}(w)$

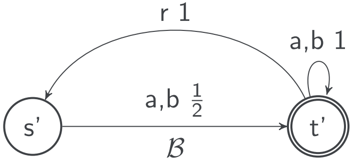
$\mathcal{A}(w\,r\,w) = \mathcal{A}(w)\mathcal{A}(w)$

$\mathcal{B}(w) = \frac{1}{2}\ \forall w \in \{a, b\}^+$

$\mathcal{B}(w\,r\,w) = \frac{1}{2}^2$

Reduction from $\exists w\ \mathcal{A}(w) > \frac{1}{2}$?





$\mathcal{A}(w)$

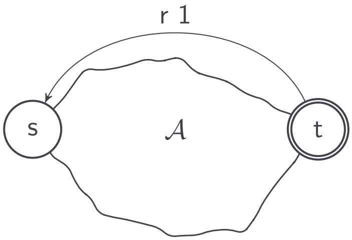
$\mathcal{A}(w\, r\, w) = \mathcal{A}(w)\mathcal{A}(w)$

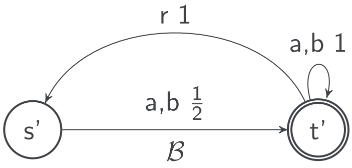
$\mathcal{B}(w) = \frac{1}{2}\ \forall w \in \{a, b\}^+$

$\mathcal{B}(w\, r\, w) = \frac{1}{2}^2$

$\frac{\mathcal{A}(w\, r\, w)}{\mathcal{B}(w\, r\, w)} = \frac{\mathcal{A}(w)^2}{1/2^2}$

Reduction from $\exists w\ \mathcal{A}(w) > \frac{1}{2}$?





$$\mathcal{A}(w)$$

$$\mathcal{A}(w\,r\,w) = \mathcal{A}(w)\mathcal{A}(w)$$

$$\vdots$$

$$\mathcal{A}((w\,r)^{i-1}w) = (\mathcal{A}(w))^i$$

$$\mathcal{B}(w) = \frac{1}{2}\ \forall w \in \{a, b\}^+$$

$$\mathcal{B}(w\,r\,w) = \frac{1}{2}^2$$

$$\vdots$$

$$\mathcal{B}((w\,r)^{i-1}w) = \frac{1}{2}^i$$

$$\frac{\mathcal{A}(w\,r\,w)}{\mathcal{B}(w\,r\,w)} = \frac{\mathcal{A}(w)^2}{1/2^2}$$

$$\vdots$$

$$\frac{\mathcal{A}((w\,r)^{i-1}w)}{\mathcal{B}((w\,r)^{i-1}w)} = (2\mathcal{A}(w))^i$$

Summary

Definition (Big-O)

$\mathcal{A}$ is big-O of $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

Results:

- ✓ Big-O Problem is undecidable
- On unary weighted automata Big-O Problem is coNP-complete
- The relation to Total Variation Distance
- The relation to Differential Privacy
- Bounded Languages

Unary Case

$|\Sigma| = 1$.

Theorem

If $\mathcal{A}$ and $\mathcal{B}$ are unary, then the Big-O problem is coNP*-complete*

Unary Case

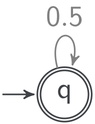
$|\Sigma| = 1$.

Theorem

If $\mathcal{A}$ and $\mathcal{B}$ are unary, then the Big-O problem is coNP*-complete*

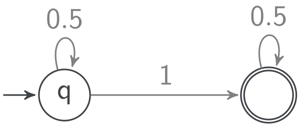
Hardness: Reduce from unary NFA universality.

Unary case



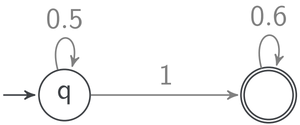
$$\mathcal{A}(a^n) = 0.5^n$$

Unary case



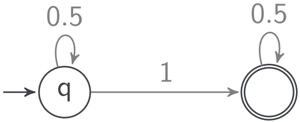
$\mathcal{A}(a^n) \approx 0.5^n n$

Unary case

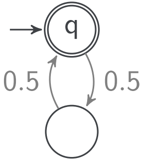


$\mathcal{A}(a^n) \approx 0.6^n$

Unary case

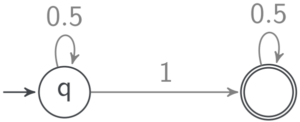


$$\mathcal{A}(a^n) \approx 0.5^n n$$

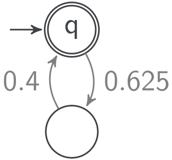


$$\begin{cases} \mathcal{A}(a^n) = 0.5^n & n \text{ even} \\ \mathcal{A}(a^n) = 0 & n \text{ odd} \end{cases}$$

Unary case

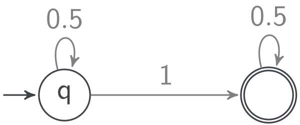


$$\mathcal{A}(a^n) \approx 0.5^n n$$

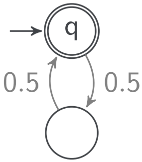


$$\begin{cases} \mathcal{A}(a^n) = 0.5^n & n \text{ even} \\ \mathcal{A}(a^n) = 0 & n \text{ odd} \end{cases}$$

Unary case

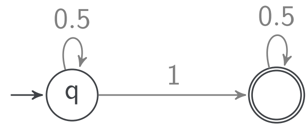


$$\mathcal{A}(a^n) \approx 0.5^n n$$

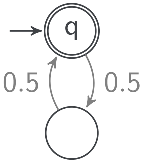


$$\begin{cases} \mathcal{A}(a^n) = 0.5^n & n \text{ even} \\ \mathcal{A}(a^n) = 0 & n \text{ odd} \end{cases}$$

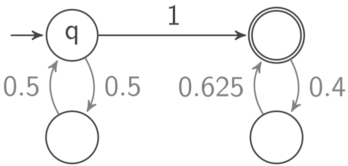
Unary case



$$\mathcal{A}(a^n) \approx 0.5^n n$$

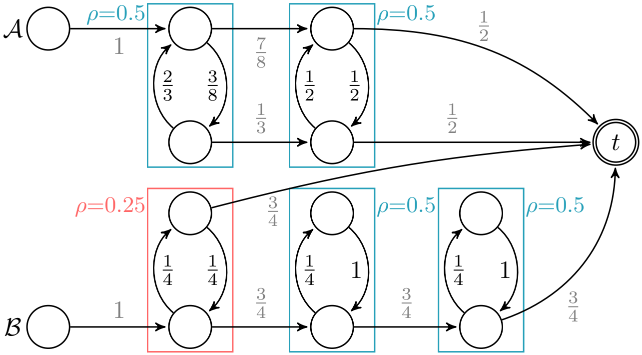


$$\begin{cases} \mathcal{A}(a^n) = 0.5^n & n \text{ even} \\ \mathcal{A}(a^n) = 0 & n \text{ odd} \end{cases}$$



$$\begin{cases} \mathcal{A}(a^n) \approx 0.5^n n & n \text{ odd} \\ \mathcal{A}(a^n) = 0 & n \text{ even} \end{cases}$$

Unary Case Example



$$\mathcal{A}(a^n) = \begin{cases} \Theta(0.5^n n) & n \geq 3 \\ 0 & n = 1, 2 \end{cases}$$

$$\mathcal{B}(a^n) = \begin{cases} \Theta(0.25^n) & n \geq 3, \text{ odd} \\ \Theta(0.5^n n) & n \geq 4, \text{ even} \\ 0 & n = 1, 2 \end{cases}$$

Unary Case

Define $f_{\mathcal{A}}(n) = (\rho, k)$ so $\mathcal{A}(a^n) = \Theta(\rho^n n^k)$

- ρ is the largest spectral radius seen on any path of length n

Unary Case

Define $f_{\mathcal{A}}(n) = (\rho, k)$ so $\mathcal{A}(a^n) = \Theta(\rho^n n^k)$

- ρ is the largest spectral radius seen on any path of length n
  - The spectral radius of an SCC is the largest eigenvalue of the SCC.

Unary Case

Define $f_{\mathcal{A}}(n) = (\rho, k)$ so $\mathcal{A}(a^n) = \Theta(\rho^n n^k)$

- ρ is the largest spectral radius seen on any path of length n
  - The spectral radius of an SCC is the largest eigenvalue of the SCC.
- ρ is seen $k+1$ times on some path

Unary Case

Define $f_{\mathcal{A}}(n) = (\rho, k)$ so $\mathcal{A}(a^n) = \Theta(\rho^n n^k)$

- ρ is the largest spectral radius seen on any path of length n
  - The spectral radius of an SCC is the largest eigenvalue of the SCC.
- ρ is seen $k + 1$ times on some path

$$(\rho, k) \leq (\rho', k') \iff \rho^n n^k = O(\rho'^n n^{k'})$$

Unary Case

Define $f_{\mathcal{A}}(n) = (\rho, k)$ so $\mathcal{A}(a^n) = \Theta(\rho^n n^k)$

- ρ is the largest spectral radius seen on any path of length n
  - The spectral radius of an SCC is the largest eigenvalue of the SCC.
- ρ is seen $k+1$ times on some path

$$(\rho, k) \leq (\rho', k') \iff \rho^n n^k = O(\rho'^n n^{k'}) \iff \rho < \rho' \ \vee\ (\rho = \rho' \wedge k \leq k')$$

Unary Case

Define $f_{\mathcal{A}}(n) = (\rho, k)$ so $\mathcal{A}(a^n) = \Theta(\rho^n n^k)$

- ρ is the largest spectral radius seen on any path of length n
  - The spectral radius of an SCC is the largest eigenvalue of the SCC.
- ρ is seen $k+1$ times on some path

$$(\rho, k) \leq (\rho', k') \iff \rho^n n^k = O(\rho'^n n^{k'}) \iff \rho < \rho' \lor (\rho = \rho' \land k \leq k')$$

Then $\mathcal{A}(a^n) \leq c \cdot \mathcal{B}(a^n)$ if and only if

- The language containment condition holds; and
- $f_{\mathcal{A}}(n) \leq f_{\mathcal{B}}(n)$ for all *but finitely many* n.

Unary Case

Verifying $f_{\mathcal{A}}(n) \leq f_{\mathcal{B}}(n)$ for all but finitely many n in coNP:

- for every (ρ, k), for a.b.f.m. n s.t. $f_{\mathcal{A}}(n) = (\rho, k)$ we require $f_{\mathcal{B}}(n) \geq (\rho, k)$

Unary Case

Verifying $f_{\mathcal{A}}(n) \le f_{\mathcal{B}}(n)$ for all but finitely many n in coNP:

- for every (ρ, k), for a.b.f.m. n s.t. $f_{\mathcal{A}}(n) = (\rho, k)$ we require $f_{\mathcal{B}}(n) \ge (\rho, k)$
- define $L_1 = \{a^n \mid f_{\mathcal{A}}(n) \ge (\rho, k)\}$ and $L_2 = \{a^n \mid f_{\mathcal{B}}(n) \ge (\rho, k)\}$

Unary Case

Verifying $f_{\mathcal{A}}(n) \leq f_{\mathcal{B}}(n)$ for all but finitely many n in coNP:

- for every (ρ, k), for a.b.f.m. n s.t. $f_{\mathcal{A}}(n) = (\rho, k)$ we require $f_{\mathcal{B}}(n) \geq (\rho, k)$
- define $L_1 = \{a^n \mid f_{\mathcal{A}}(n) \geq (\rho, k)\}$ and $L_2 = \{a^n \mid f_{\mathcal{B}}(n) \geq (\rho, k)\}$
- verify $L_1 \setminus L_2$ is finite

Unary Case

Verifying $f_{\mathcal{A}}(n) \leq f_{\mathcal{B}}(n)$ for all but finitely many n in coNP:

- for every (ρ, k), for a.b.f.m. n s.t. $f_{\mathcal{A}}(n) = (\rho, k)$ we require $f_{\mathcal{B}}(n) \geq (\rho, k)$
- define $L_1 = \{a^n \mid f_{\mathcal{A}}(n) \geq (\rho, k)\}$ and $L_2 = \{a^n \mid f_{\mathcal{B}}(n) \geq (\rho, k)\}$
- verify $L_1 \setminus L_2$ is finite
  - We show this can be done in coNP for two unary NFA.

Summary

Definition (Big-O)

$\mathcal{A}$ is big-O of $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

Results:

- ✓ Big-O Problem is undecidable
- ✓ On unary weighted automata Big-O Problem is coNP-complete
- ▶ The relation to Total Variation Distance
- ▶ The relation to Differential Privacy
- ▶ Bounded Languages

Relation to Total Variation Distance

Definition (Total variation distance)

$$tv(\mathcal{A},\mathcal{B}) = \sup_{E\subseteq\Sigma^*} \mathcal{A}(E) - \mathcal{B}(E)$$

Relation to Total Variation Distance

Definition (Total variation distance)

$$tv(\mathcal{A},\mathcal{B}) = \sup_{E\subseteq\Sigma^*} \mathcal{A}(E) - \mathcal{B}(E)$$

Definition (Ratio variation distance)

$$r(\mathcal{A},\mathcal{B}) = \sup_{E\subseteq\Sigma^*} \frac{\mathcal{A}(E)}{\mathcal{B}(E)}$$

Relation to Total Variation Distance

Definition (Total variation distance)

$$tv(\mathcal{A}, \mathcal{B}) = \sup_{E \subseteq \Sigma^*} \mathcal{A}(E) - \mathcal{B}(E)$$

Definition (Ratio variation distance)

$$r(\mathcal{A}, \mathcal{B}) = \sup_{E \subseteq \Sigma^*} \frac{\mathcal{A}(E)}{\mathcal{B}(E)} = \sup_{w \in \Sigma^*} \frac{\mathcal{A}(w)}{\mathcal{B}(w)}$$

Relation to Total Variation Distance

Definition (Total variation distance)

$$tv(\mathcal{A},\mathcal{B}) = \sup_{E\subseteq\Sigma^*} \mathcal{A}(E) - \mathcal{B}(E)$$

Definition (Ratio variation distance)

$$r(\mathcal{A},\mathcal{B}) = \sup_{E\subseteq\Sigma^*} \frac{\mathcal{A}(E)}{\mathcal{B}(E)} = \sup_{w\in\Sigma^*} \frac{\mathcal{A}(w)}{\mathcal{B}(w)}$$

$r(\mathcal{A},\mathcal{B})$ is either

- the smallest c such that $\mathcal{A}(w) \leq c\cdot\mathcal{B}(w)$ for every w; or
- ∞ if $\mathcal{A}$ not big-O $\mathcal{B}$

Differential Privacy

Definition

$\mathcal{A}, \mathcal{B}$ are ϵ-differentially private if $\forall w \in \Sigma^*$

$$\frac{\mathcal{A}(w)}{\mathcal{B}(w)} \leq e^{\epsilon}$$

To compute e^{ϵ}, we would like to know if it exists (Big-O Problem)

Differential Privacy

Definition

$\mathcal{A}, \mathcal{B}$ are ϵ-differentially private if $\forall w \in \Sigma^*$

$$\frac{\mathcal{A}(w)}{\mathcal{B}(w)} \leq e^{\epsilon}$$

To compute e^{ϵ}, we would like to know if it exists (Big-O Problem)

Theorem

$r(\mathcal{A}, \mathcal{B})$, and thus e^{ϵ}, cannot be approximated

Differential Privacy

Definition

$\mathcal{A}, \mathcal{B}$ are ϵ-differentially private if $\forall w \in \Sigma^*$

$$\frac{\mathcal{A}(w)}{\mathcal{B}(w)} \leq e^\epsilon$$

To compute e^ϵ, we would like to know if it exists (Big-O Problem)

Theorem

$r(\mathcal{A}, \mathcal{B})$, and thus e^ϵ, cannot be approximated

even under the promise that $r(\mathcal{A}, \mathcal{B}) < \infty$

Summary

Definition (Big-O)

$\mathcal{A}$ is big-O of $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

Results:

- ✓ Big-O Problem is undecidable
- ✓ On unary weighted automata Big-O Problem is coNP-complete
- ✓ The relation to Total Variation Distance
- ✓ The relation to Differential Privacy
- ▶ Bounded Languages

Bounded Languages

Definition

A language L is letter-bounded if there exists $a_1, \ldots, a_m \in \Sigma$ such that $L \subseteq a_1^* \ldots a_m^*$

Bounded Languages

Definition

A language L is letter-bounded if there exists $a_1, \ldots, a_m \in \Sigma$ such that $L \subseteq a_1^* \ldots a_m^*$

Definition

A language L is bounded if there exists $w_1, \ldots, w_m \in \Sigma^*$ such that $L \subseteq w_1^* \ldots w_m^*$

Bounded Languages

Definition

A language L is letter-bounded if there exists $a_1, \ldots, a_m \in \Sigma$ such that $L \subseteq a_1^* \ldots a_m^*$

Definition

A language L is bounded if there exists $w_1, \ldots, w_m \in \Sigma^*$ such that $L \subseteq w_1^* \ldots w_m^*$

Theorem

The big-O problem for weighted automata with bounded languages is decidable subject to Schanuel's conjecture

Bounded Languages

Definition

A language L is letter-bounded if there exists $a_1, \ldots, a_m \in \Sigma$ such that $L \subseteq a_1^* \ldots a_m^*$

Definition

A language L is bounded if there exists $w_1, \ldots, w_m \in \Sigma^*$ such that $L \subseteq w_1^* \ldots w_m^*$

Theorem

The big-O problem for weighted automata with bounded languages is decidable subject to Schanuel's conjecture†

†Or the decidability of the first order theory of the *reals* with exponential function

Summary

Definition (Big-O)

$\mathcal{A}$ is big-O of $\mathcal{B}$ if $\exists c > 0$, s.t. $\forall w \in \Sigma^*$

$$\mathcal{A}(w) \leq c \cdot \mathcal{B}(w)$$

Results:

- ✓ Big-O Problem is undecidable
- ✓ On unary weighted automata Big-O Problem is coNP-complete
- ✓ The relation to Total Variation Distance
- ✓ The relation to Differential Privacy
- ✓ Bounded Languages

Open Problems

- When can $r(\mathcal{A}, \mathcal{B})$ (optimal c) be computed/approximated?
- Find the boundary between decidability and undecidability.
  - polynomial-ambiguity is open
- Negative values
- $(\mathbb{N}, \max, +)$ open (we think decidable) and $(\mathbb{N}, \min, +)$ is decidable

2 Letter-bounded Case

Assume $L \subseteq a^*b^*$, then there are two matrices A and B s.t. $\mathcal{A}(a^n b^m) = (A^n \times B^m)_{s,t}$.

Want to know if there exists c such that $(A^n \times B^m)_{s,t} \leq c(A^n \times B^m)_{s',t}$ for all n, m?

2 Letter-bounded Case

Assume $L \subseteq a^* b^*$, then there are two matrices A and B s.t. $\mathcal{A}(a^n b^m) = (A^n \times B^m)_{s,t}$.

Want to know if there exists c such that $(A^n \times B^m)_{s,t} \leq c(A^n \times B^m)_{s',t}$ for all n, m?

$$(A^n \times B^m)_{s,t} = \sum_q (A^n)_{s,q} \cdot (B^m)_{q,t}$$

2 Letter-bounded Case

Assume $L \subseteq a^*b^*$, then there are two matrices A and B s.t. $\mathcal{A}(a^n b^m) = (A^n \times B^m)_{s,t}$.

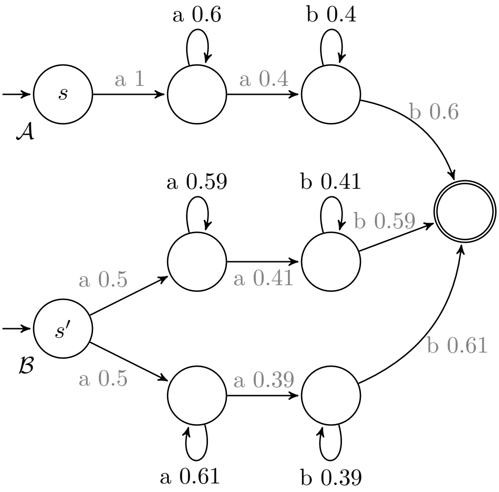
Want to know if there exists c such that $(A^n \times B^m)_{s,t} \le c(A^n \times B^m)_{s',t}$ for all n, m?

$$(A^n \times B^m)_{s,t} = \sum_q (A^n)_{s,q} \cdot (B^m)_{q,t}$$

So

- each $(A^n)_{s,q}$ has its own (ρ, k)-value; and
- each $(B^m)_{q,t}$ has its own (ρ, k)-value.

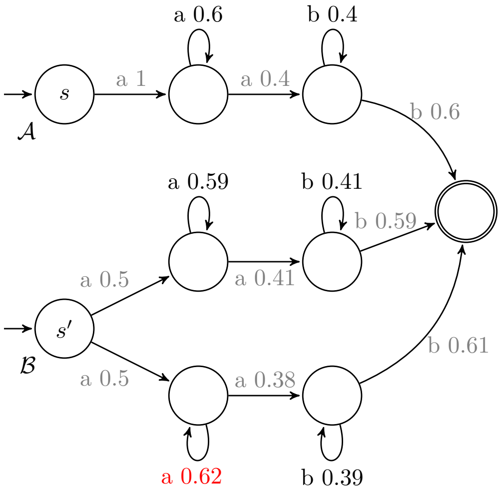
Why is it harder?



$\mathcal{A}(a^n b^m) = \Theta(0.6^n \cdot 0.4^m)$ and
$\mathcal{B}(a^n b^m) = \Theta(0.61^n \cdot 0.39^m + 0.59^n \cdot 0.41^m)$

$$\frac{\mathcal{A}(a^n b^{0.66n})}{\mathcal{B}(a^n b^{0.66n})} \xrightarrow[n \to \infty]{} \infty$$

Why is it harder?



$\mathcal{A}(a^n b^m) = \Theta(0.6^n \cdot 0.4^m)$ and
$\mathcal{B}(a^n b^m) = \Theta(0.61^n \cdot 0.39^m + 0.59^n \cdot 0.41^m)$

$$\frac{\mathcal{A}(a^n b^{0.66n})}{\mathcal{B}(a^n b^{0.66n})} \xrightarrow[n\to\infty]{} \infty$$

$$\mathcal{B}(a^n b^m) = \Theta(0.62^n \cdot 0.39^m + 0.59^n \cdot 0.41^m)$$

$$\frac{\mathcal{A}(a^n b^m)}{\mathcal{B}(a^n b^m)} \leq c \text{ for all } n, m$$

Solution

- Encode comparison in to First Order Theory of the *Reals*

Solution

- Encode comparison in to First Order Theory of the *Reals*
  - There is a real solution if and only if there is a integer solution
  - Can relax the real solution to the nearest integer solution, as we are looking at large words

Solution

- Encode comparison in to First Order Theory of the *Reals*
  - There is a real solution if and only if there is a integer solution
  - Can relax the real solution to the nearest integer solution, as we are looking at large words
- But encoding $\rho^n n^k$ requires *exponential* function.
  - Decidable subject to Schanuel's conjecture