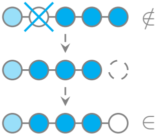
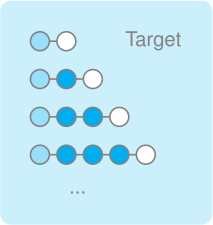
What do you do
if a computational object fails a specification?





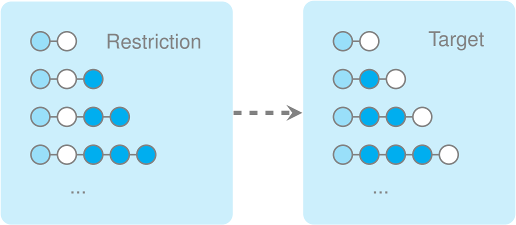
Over finite words and regular specifications:

1. Non-deterministic finite automata

2. Deterministic finite automata

“Regular repair of specifications”, in LICS 2011.

What do you do
if a computational object fails a specification?



Over finite words and regular specifications:

1. Non-deterministic finite automata
2. Deterministic finite automata

"Regular repair of specifications", in LICS 2011.

At what rate do we need to repair each word in the worst case?

Asymptotic cost

$$\lim_{n\to\infty} \sup \left\{ \underbrace{\frac{\mathrm{cost}(w, T)}{|w|}}_{\text{normalized cost}} \;\middle|\; w \in R,\ |w| \geq n \right\}$$

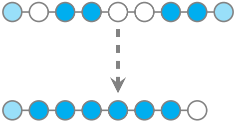
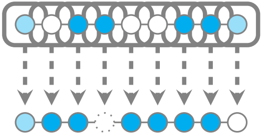
Example

$R : (aa)^*$		$T : (ab)^*$	$\rightarrow$ 1/2
$(\mathrm{a\ a})^N$	- - - →	$(\mathrm{a\ b})^N$	
$R : a^+ b^+$		$T : a^+ c^+$	$\rightarrow$ 1
$\mathrm{a\ b}^N$	- - - →	$\mathrm{a\ c}^N$	

At what rate do we need to repair each word in the worst case?

Asymptotic cost

Different ways of repairing:

Arbitrary	Streaming
	

At what rate do we need to repair each word in the worst case?

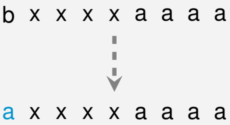
Asymptotic cost

Example

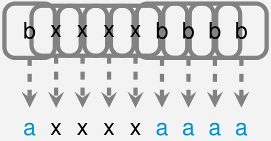
$R: \quad (a + b)\, x^* \,(a^* + b^*)$

$T: \quad a\, x^*\, a^* + b\, x^*\, b^*$

Arbitrary $\to 0$



Streaming $\to 1$



We study how to compute the asymptotic cost of two regular languages

1. The asymptotic cost is rational and can be effectively computed.
2. First algorithm to compute the asymptotic cost:
   - Distance automata and their determinization.
   - Cycle analysis.
3. Streaming asymptotic cost.
   - Reduction to mean-payoff games.
   - Complexity in PTIME.

We study how to compute the asymptotic cost of two regular languages

1. The asymptotic cost is rational and can be effectively computed.

2. First algorithm to compute the asymptotic cost:
   - Distance automata and their determinization.
   - Cycle analysis.

3. Streaming asymptotic cost.
   - Reduction to mean-payoff games.
   - Complexity in PTIME.

The cost of traveling between languages

Cristian Riveros
Michael Benedikt
Gabriele Puppis

University of Oxford
ICALP 2011

Outline

Setting

Edit distance automata

Determinization

Asymptotic Cost

Outline

Setting

Edit distance automata

Determinization

Asymptotic Cost

Repairability over regular languages

- Σ and Δ are alphabets.
- Two regular languages:
  - R (Restriction) over Σ^*, and
  - T (Target) over Δ^*.
- R and T are given by:
  - Deterministic finite automata (DFA), or
  - Non-deterministic finite automata (NFA).

In this talk, all automata are trim.

Repairability using edit operations

Edit operations: deletion, insertion, and relabeling.

- All operations have cost equal to 1.

Definition

For words u, v and language T:

$$\mathrm{dist}(u, v) \quad = \quad \text{shortest sequence of operations that transform } u \text{ into } v$$

$$\mathrm{dist}(u, T) \quad = \quad \min_{v \in T} \{ \mathrm{dist}(u, v) \}$$

Both computable in PTIME

(Wagner and Fisher 1974, Wagner 1974).

Asymptotic cost

Definition

$$\mathbf{A}(R, T) = \lim_{n \to \infty} \sup \{ \underbrace{\frac{\mathrm{dist}(w, T)}{|w|}}_{\text{normalized cost}} \mid w \in R, \ |w| \geq n \}$$

Example

$R : (aa)^*$ — $T : (ab)^*$

$(\mathrm{a}\ \mathrm{a})^N \dashrightarrow (\mathrm{a}\ \mathrm{b})^N$

$\mathbf{A}(R, T) = 1/2$

$R : a^+ b^+$ — $T : a^+ c^+$

$\mathrm{a}\ \mathrm{b}^N \dashrightarrow \mathrm{a}\ \mathrm{c}^N$

$\mathbf{A}(R, T) = 1$

Asymptotic cost

Definition

$$\mathbf{A}(R, T) = \lim_{n \to \infty} \sup \left\{ \underbrace{\frac{\mathrm{dist}(w, T)}{|w|}}_{\text{normalized cost}} \;\middle|\; w \in R,\ |w| \geq n \right\}$$

The asymptotic cost $\mathbf{A}(R, T)$ of regular languages R and T:

- always exists.
- is between 0 and 1.

In this talk:

We focus on how to compute $\mathbf{A}(\Sigma^*, T)$ where $R = \Sigma^*$.

Outline

Distance automata

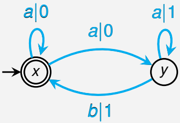
Definition

A distance automaton $\mathcal{D}$ is a non-deterministic finite automaton with transitions labeled with cost in $\mathbb{N} \cup \{\infty\}$.

$$\mathcal{D} : \ \Sigma^* \to \mathbb{N} \cup \{\infty\}$$

Example

$\mathcal{D}$:



$w \ = \ \text{a a b}$

$\rho_1 : \ x \xrightarrow{a|0} y \xrightarrow{a|1} y \xrightarrow{b|1} x \qquad \text{cost}(\rho_1) = 2$

$\rho_2 : \ x \xrightarrow{a|0} x \xrightarrow{a|0} y \xrightarrow{b|1} x \qquad \text{cost}(\rho_2) = 1$

$$\mathcal{D}(w) \ = \ \min\{ \ \text{cost}(\rho_1), \text{cost}(\rho_2) \ \} \ = \ 1$$

$$\mathcal{D}(w) \ = \ \min \{ \ \text{cost}(\rho) \ \mid \ \rho \text{ is a run of } w \text{ over } \mathcal{D} \ \}$$

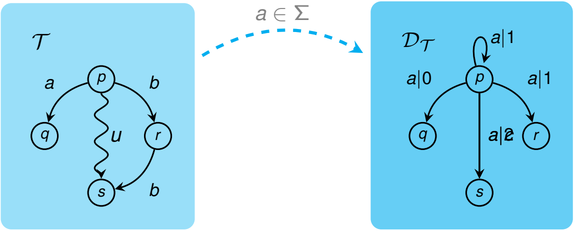
The edit-distance to a language can be computed by a distance automaton

Let $\mathcal{T} = (\Delta, Q, \delta, q_0, F)$ be a finite automaton.

Definition

We define the edit distance automaton $\mathcal{D}_{\mathcal{T}} = (\Sigma, Q, \delta^{\text{edit}}, q_0^{\text{edit}}, F^{\text{edit}})$:

$$\mathcal{D}_{\mathcal{T}}(w) = \text{dist}(w, \mathcal{T}) \quad \text{for all } w \in \Sigma^*$$



$$c = \min_{u \in \Sigma^*} \{ \text{dist}(a, u) \mid p \xrightarrow{u} s \text{ in } \mathcal{T} \}$$

The edit-distance to a language can be computed by a distance automaton

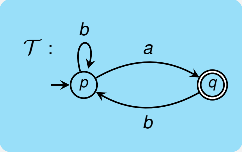
Let $\mathcal{T} = (\Delta, Q, \delta, q_0, F)$ be a finite automaton.

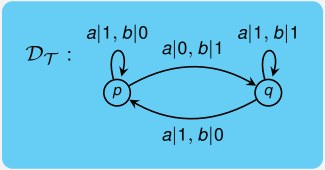
Definition

We define the edit distance automaton $\mathcal{D}_\mathcal{T} = (\Sigma, Q, \delta^{\text{edit}}, q_0^{\text{edit}}, F^{\text{edit}})$:

$$\mathcal{D}_\mathcal{T}(w) = \text{dist}(w, \mathcal{T}) \quad \text{for all } w \in \Sigma^*$$

Example





The asymptotic cost problem for a distance automaton

For any distance automaton $\mathcal{D}$:

$$\mathbf{A}(\mathcal{D}) = \lim_{n \to \infty} \sup \left\{ \frac{\mathcal{D}(w)}{|w|} \;\middle|\; w \in \mathcal{L}(\mathcal{D}),\ |w| \geq n \right\}$$

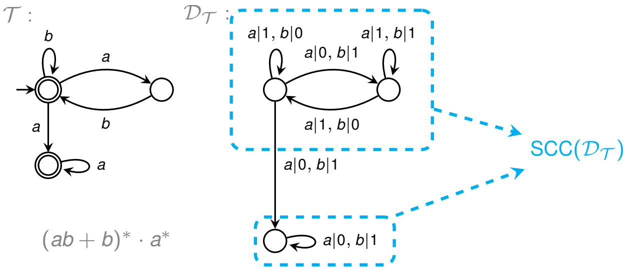
Theorem

The problem of deciding whether $\mathbf{A}(\mathcal{D}) \leq \frac{1}{2}$ is undecidable given an arbitrary distance automaton $\mathcal{D}$.

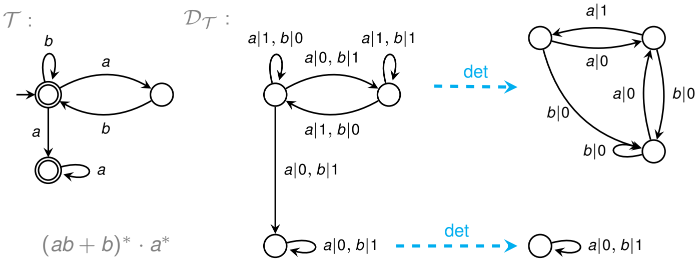
This is not the case for the edit distance automaton $\mathcal{D}_{\mathcal{T}}$.

Outline

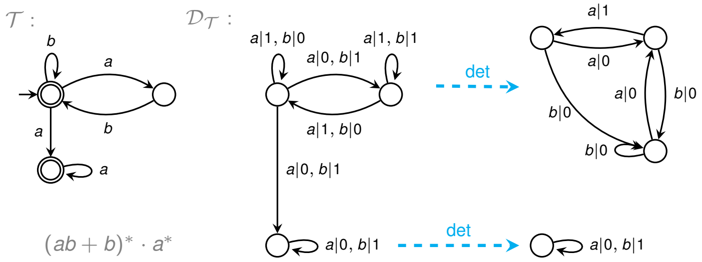
The strongly connected components (SCC) of $\mathcal{D}_{\mathcal{T}}$ are determinizable



The strongly connected components (SCC) of $\mathcal{D}_{\mathcal{T}}$ are determinizable

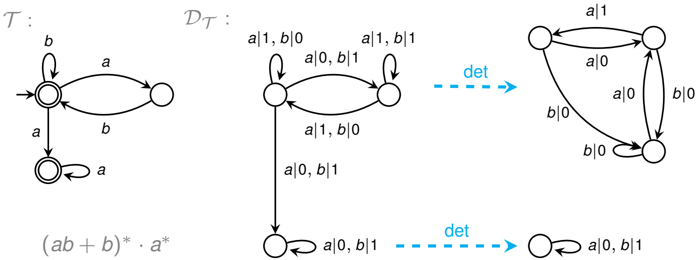


The strongly connected components (SCC) of $\mathcal{D}_\mathcal{T}$ are determinizable



$\mathcal{D}_\mathcal{T}$ is NOT always determinizable.

The strongly connected components (SCC) of $\mathcal{D}_\mathcal{T}$ are determinizable



$(ab + b)^* \cdot a^*$

Proposition

$\mathcal{D}_\mathcal{T}|C$ is determinizable for every $C \in \mathrm{SCC}(\mathcal{D}_\mathcal{T})$.

We can determinize $\mathcal{D}_\mathcal{T}|C$ using Mohri's procedure (Mohri 1997).

The asymptotic cost can be computed using the determinization of a distance automata

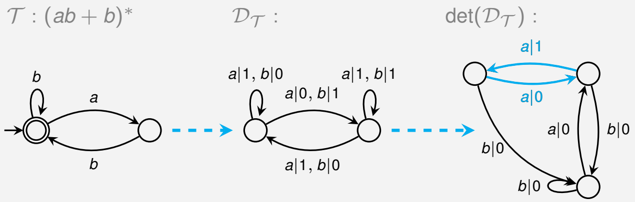
Proposition

Suppose that $\mathcal{D}_{\mathcal{T}}$ is a single strongly connected component:

$$\mathbf{A}(\mathcal{D}_{\mathcal{T}}) = \max\left\{ \frac{\text{cost}(L)}{|L|} \mid L \text{ is a simple cycle of } \det(\mathcal{D}_{\mathcal{T}}) \right\}$$

- $\text{cost}(L) =$ sum of the cost of the edges of L.

Example



Outline

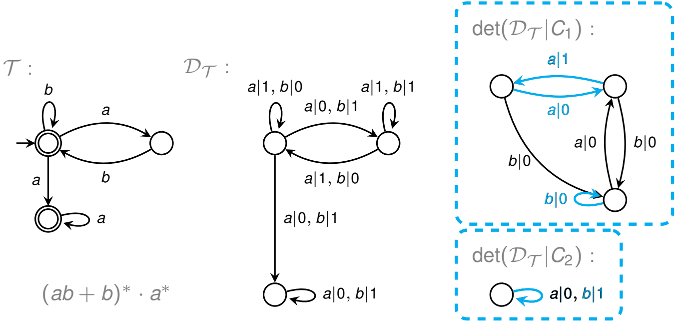
Setting

Edit distance automata

Determinization

Asymptotic Cost

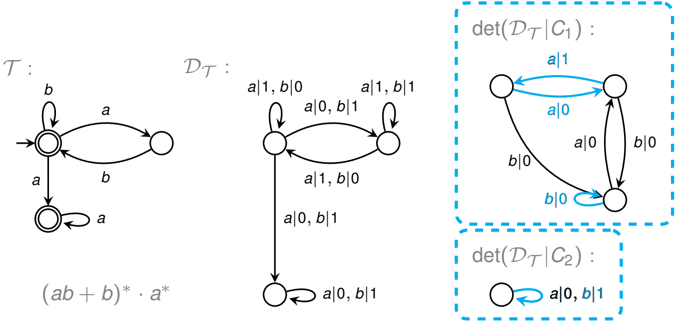
Moving to multiple components



Worst case for C_1: $(aa)\,(aa)\,(aa)\ \ldots$

- $\mathbf{A}((aa)^n, C_1) = \frac{1}{2}$.
- $\mathbf{A}((aa)^n, C_2) = 0$.

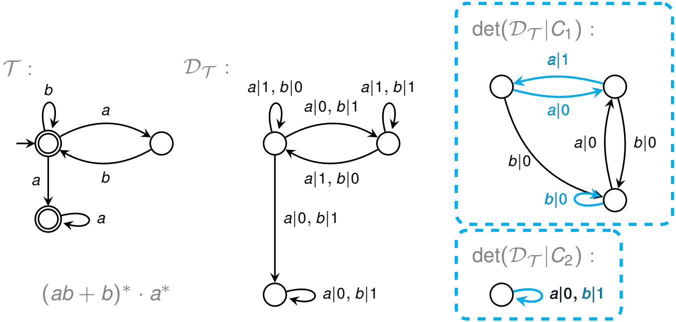
Moving to multiple components



Worst case for C_2: $b\ b\ b\ b\ \ldots$

- $\mathbf{A}(b^n, C_2) \ = \ 1.$
- $\mathbf{A}(b^n, C_1) \ = \ 0.$

Moving to multiple components



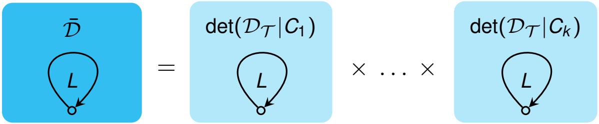
A($\mathcal{D}_\mathcal{T}$) cannot be computed as a function of the simple cycles of det($\mathcal{D}_\mathcal{T}|C$) for all $C \in \mathcal{D}_\mathcal{T}$.

We have to consider a combination of the common cycles of all components

Let $C_1, \ldots, C_k$ be the SCC of $\mathcal{D}_{\mathcal{T}}$.

Definition

We define the multi-distance automaton $\bar{\mathcal{D}}$:



Let $L_1, \ldots, L_m$ be all the simple cycles of $\bar{\mathcal{D}}$.

Definition

$\text{cost}(L_i, C) \quad = \quad$ cost of the projection of the simple cycle L_i into the component $\det(\mathcal{D}_{\mathcal{T}}|C)$ of $\bar{\mathcal{D}}$.

$\mathbf{A}(\mathcal{D}_{\mathcal{T}})$ is equal to a linear combination of its cycles

Theorem

$$\mathbf{A}(\mathcal{D}_{\mathcal{T}}) = \max_{r_1,\ldots,r_m \geq 0} \min_{C \in \mathrm{SCC}(\mathcal{T})} \frac{\sum_{i=1}^{m} r_i \cdot \mathrm{cost}(L_i, C)}{\sum_{i=1}^{m} r_i \cdot |L_i|}$$

- m: number of cycles in $\bar{\mathcal{D}}$.
- $r_1, \ldots, r_m$: number of repetitions of simple cycles $L_1, \ldots, L_m$.

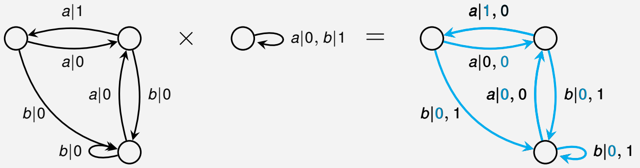
$\mathbf{A}(\mathcal{D}_{\mathcal{T}})$ is equal to a linear combination of its cycles

Theorem

$$\mathbf{A}(\mathcal{D}_{\mathcal{T}}) \;=\; \max_{r_1,\ldots,r_m \geq 0} \;\; \min_{C \in \mathrm{SCC}(\mathcal{T})} \; \frac{\sum_{i=1}^{m} r_i \cdot \mathrm{cost}(L_i, C)}{\sum_{i=1}^{m} r_i \cdot |L_i|}$$

Example $[\ \mathcal{T} :\ (ab + b)^* \cdot a^*\ ]$

$\det(\mathcal{D}_{\mathcal{T}} | C_1)$: $\qquad$ $\det(\mathcal{D}_{\mathcal{T}} | C_2)$: $\qquad$ $\bar{\mathcal{D}}$:



$$\mathbf{A}(\Sigma^*, \mathcal{T}) = \frac{1}{3}$$

Some remarks about computing $\mathbf{A}(\mathcal{D}_{\mathcal{T}})$

$$\mathbf{A}(\mathcal{D}_{\mathcal{T}}) = \max_{r_1,\ldots,r_m \geq 0} \; \min_{C \in \mathrm{SCC}(\mathcal{T})} \frac{\sum_{i=1}^{m} r_i \cdot \mathrm{cost}(L_i, C)}{\sum_{i=1}^{m} r_i \cdot |L_i|}$$

linear programming problem

$$\textbf{MAXIMIZE} \quad y \quad \textbf{SUBJECT TO} \quad \sum_{i=1}^{m} \frac{\mathrm{cost}(L_i, C)}{|L_i|} \cdot x_i \geq y \quad \forall\, C \in \mathrm{SCC}(\mathcal{T})$$

$$\sum_{i=1}^{m} x_i \leq 1$$

$\mathbf{A}(\mathcal{D}_{\mathcal{T}})$ is a rational number.

Some remarks about the complexity of computing $\mathbf{A}(\mathcal{D}_\mathcal{T})$

$\mathbf{A}(\mathcal{D}_\mathcal{T})$ can be computed in double exponential time.

- The size of the multiple-distance automata $\bar{\mathcal{D}}$ is exponential in $\mathcal{D}_\mathcal{T}$.
- The number m of simple cycles is exponential in $\bar{\mathcal{D}}$.
- $\mathbf{A}(\mathcal{D}_\mathcal{T})$ can be reduced to a linear programming problem of size double exponential.

The exact complexity of computing $\mathbf{A}(\mathcal{D}_\mathcal{T})$ is an open problem.

Conclusions and current work

1. The asymptotic cost is rational and can be effectively computed.
2. First algorithm to compute the asymptotic cost:
   - Distance automata and their determinization.
   - Cycle analysis.
3. Streaming asymptotic cost.
   - Reduction to mean-payoff games.
   - Complexity in PTIME.
4. Current work:
   - Repairing tree regular languages.

The cost of traveling between languages

Cristian Riveros
Michael Benedikt
Gabriele Puppis

University of Oxford
ICALP 2011