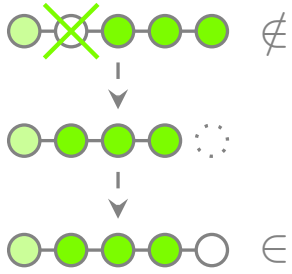
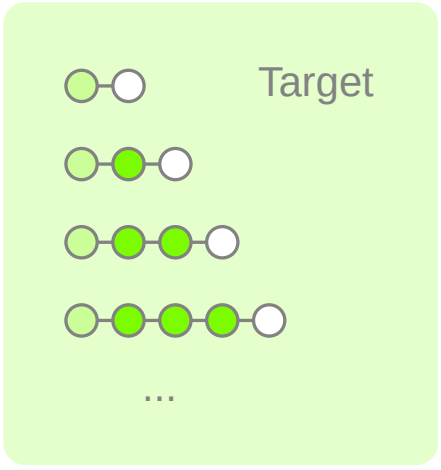
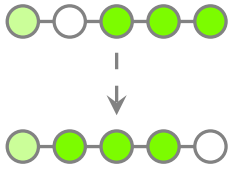
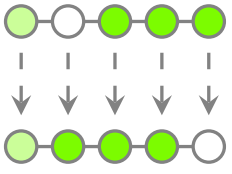
What do you do

if a computational object fails a specification?



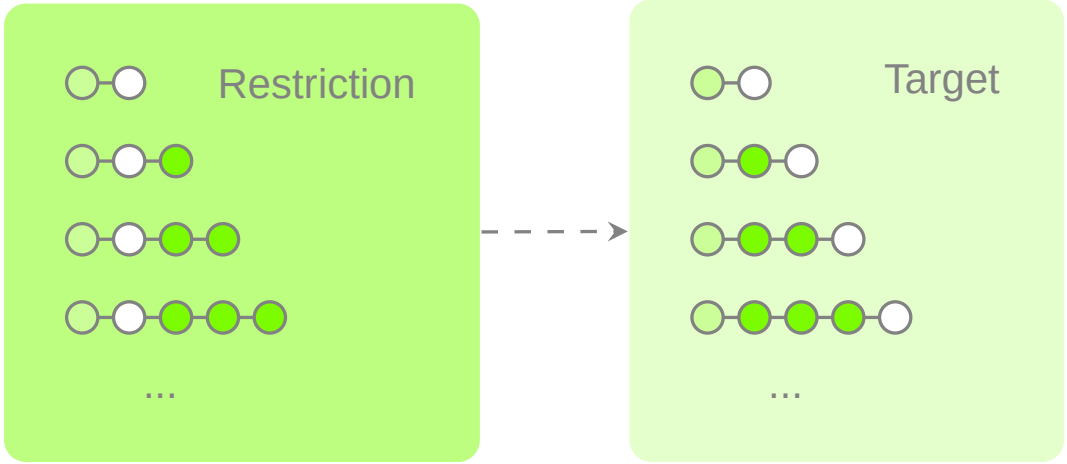


Different ways of repairing:

Arbitrary	Streaming
	

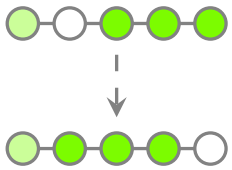
What do you do

if a computational object fails a specification?

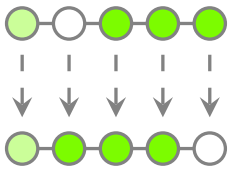


Different ways of repairing:

Arbitrary

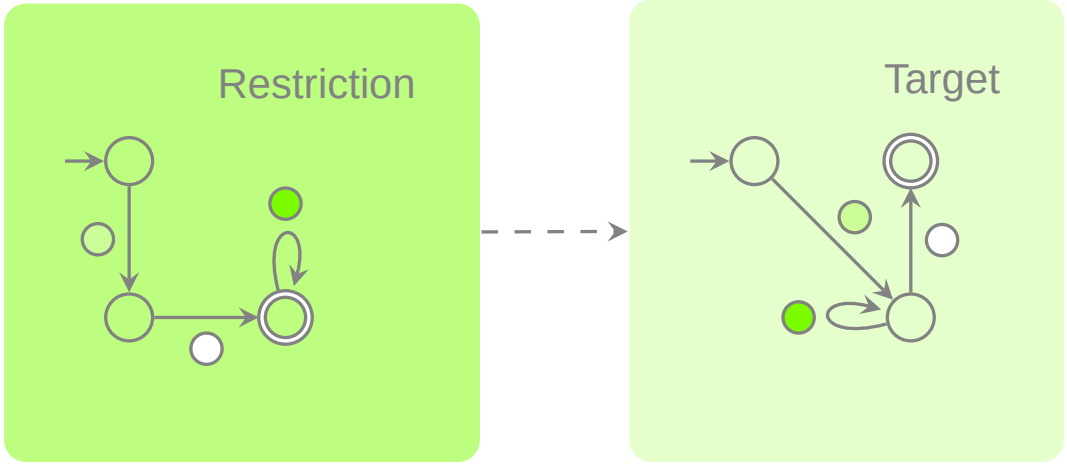


Streaming



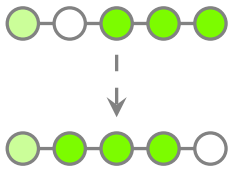
What do you do

if a computational object fails a specification?

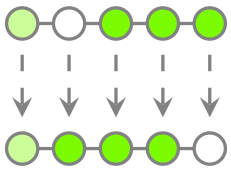


Different ways of repairing:

Arbitrary



Streaming



Applications of repairing regular languages

XML data

- Correct XML Documents.

Software verification

- Satisfability.

We study two main problems about repairing regular languages

1. Bounded Repairability
   - Can we repair each word with a bounded number of modifications?

Example

$$R : (ba)^* b \qquad\qquad T : (a^* b)^*$$
$$(\mathrm{b\ a})^N\ \mathrm{b} \quad \Rightarrow \quad \mathrm{a}\ (\mathrm{b\ a})^N\ \mathrm{b}$$

$$R : (a + b)^* \qquad\qquad T : (a + bb)^*$$
$$(\mathrm{a\ b})^N \quad \Rightarrow \quad (\mathrm{a\ b}\ \not{\mathrm{a}}\ \mathrm{b})^{\frac{N}{2}}$$

We study two main problems about repairing regular languages

1. Bounded Repairability
   - Can we repair each word with a bounded number of modifications?
2. Asymptotic cost
   - At what rate do we need to repair each word in the worst case?

Example

$$R : (a + b)^* \qquad T : (a + bb)^*$$

$$(\mathrm{a\ b\ a})^N \quad \Rightarrow \quad (\mathrm{a\ b\ b})^N \qquad \to \tfrac{1}{3}$$

$$R : a^+b^+ \qquad T : a^+c^+$$

$$\mathrm{a\ b}^N \quad \Rightarrow \quad \mathrm{a\ c}^N \qquad \to \mathbf{1}$$

We study two main problems about repairing regular languages

1. Bounded Repairability
   - Can we repair each word with a bounded number of modifications?
2. Asymptotic cost
   - At what rate do we need to repair each word in the worst case?

We also consider these problems in a streaming setting.

In this talk...

1. Bounded Repairability (LICS 2011)
   - Characterizations for the general and streaming setting.
   - Connections with reachability games in the streaming setting.
   - Complexity for the general and streaming setting.
2. Asymptotic cost (ICALP 2011)
   - Connections with distance automata.
   - Rationality and algorithm to compute the asymptotic cost.

The cost of repairing regular specifications

Cristian Riveros
M. Benedikt, G. Puppis

University of Oxford
7th of June

Outline

Setting

Bounded Repairability

Asymptotic cost

Repairability over regular languages

- Σ and Δ are alphabets.
- Finite automaton $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$.
- All the automata in this talk are trim.
- Two regular languages: R (Restriction) and T (Target).
  - $R = \mathcal{L}(\mathcal{R})$ where $\mathcal{R} = (\Sigma, Q, \delta, q_0, F)$, and
  - $T = \mathcal{L}(\mathcal{T})$ where $\mathcal{T} = (\Delta, Q', \delta', q_0', F')$.
- We consider repairability over:
  - Deterministic finite automata (DFA), and
  - Non-deterministic finite automata (NFA).

Repairability using edit operations

Edit operations: deletion, insertion, and relabeling.

- All operations have cost equal to 1.

Definition

For words u, v in Σ^*:

$$\begin{aligned} \text{dist}(u, v) &= \text{shortest sequence of operations that transform } u \text{ into } v \\ \text{dist}(u, T) &= \min_{v \in T} \{ \text{dist}(u, v) \} \end{aligned}$$

Repairing words

- A repair strategy is a function $f : R \to T$.
- A streaming repair strategy is a function $f : R \to T$ given by a sequential transducer.
- The cost of f over $u \in R$ is denoted by $\mathrm{dist}(u, f(u))$.

We study the set of repair strategies given R and T.

Outline

Setting

Bounded Repairability

Asymptotic cost

Bounded repairability

Definition

Given R and T, determine if there exists a (streaming) repair strategy $f : R \to T$ and $n \in \mathbb{N}$:

$$\text{dist}(u, f(u)) \leq n \quad \text{for all } u \in R$$

Example

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

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

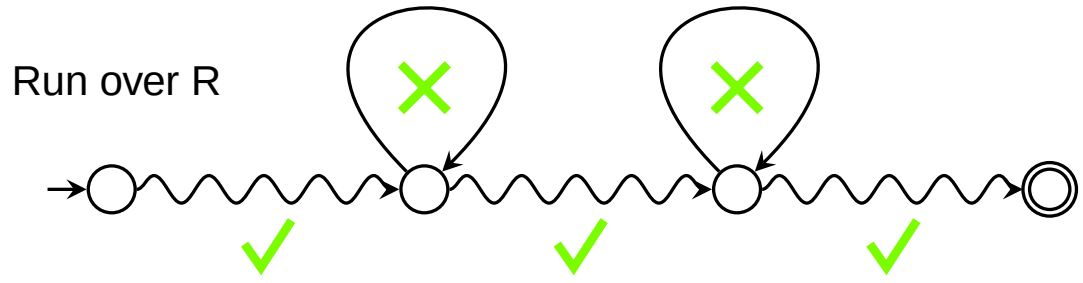
Arbitrary: b x x $\cdots$ x a a $\cdots$ a $\Rightarrow$ a x x $\cdots$ x a a $\cdots$ a ✓

Streaming: b x x $\cdots$ x b b $\cdots$ b $\Rightarrow$ a x x $\cdots$ x a a $\cdots$ a ✗

Generalization of language containment.

Intuition of bounded repairability

We should not repair during the cyclic behavior of *R*.



Intuition of bounded repairability

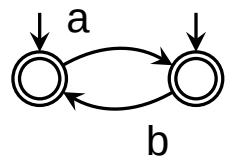
We should not repair during the cyclic behavior of R.

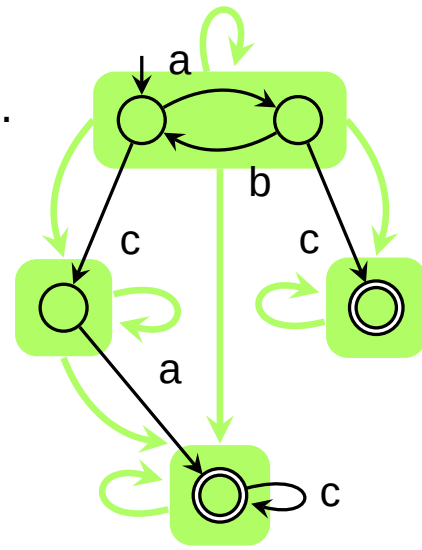
Definition

For an automaton $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$:

- SCC($\mathcal{A}$): strongly connected components of $\mathcal{A}$.
- dag($\mathcal{A}$): directed acyclic graph of SCC($\mathcal{A}$).
- dag*($\mathcal{A}$): transitive closure of dag($\mathcal{A}$).
- Given $C \in$ SCC($\mathcal{A}$), we define:

$$\mathcal{A}|C = (\Sigma, Q, \delta, C, C)$$





$\mathcal{L}(\mathcal{A}|C)$ contains the cyclic behavior of C in $\mathcal{A}$.

Path covering

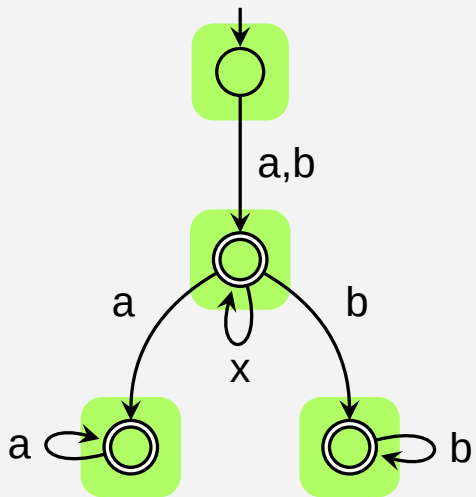
Definition

Given two NFA $\mathcal{R}$ and $\mathcal{T}$, a path $\pi = C_1 \dots C_k$ in $\mathrm{dag}(\mathcal{R})$ is covered by a path $\pi' = C'_1 \dots C'_k$ in $\mathrm{dag}^*(\mathcal{T})$ if:

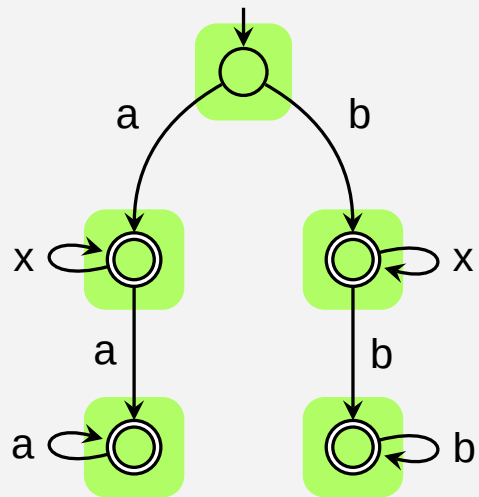
$$\mathcal{L}(\mathcal{R}|C_i) \subseteq \mathcal{L}(\mathcal{T}|C'_i) \quad \text{for all } i \leq k$$

Example

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



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

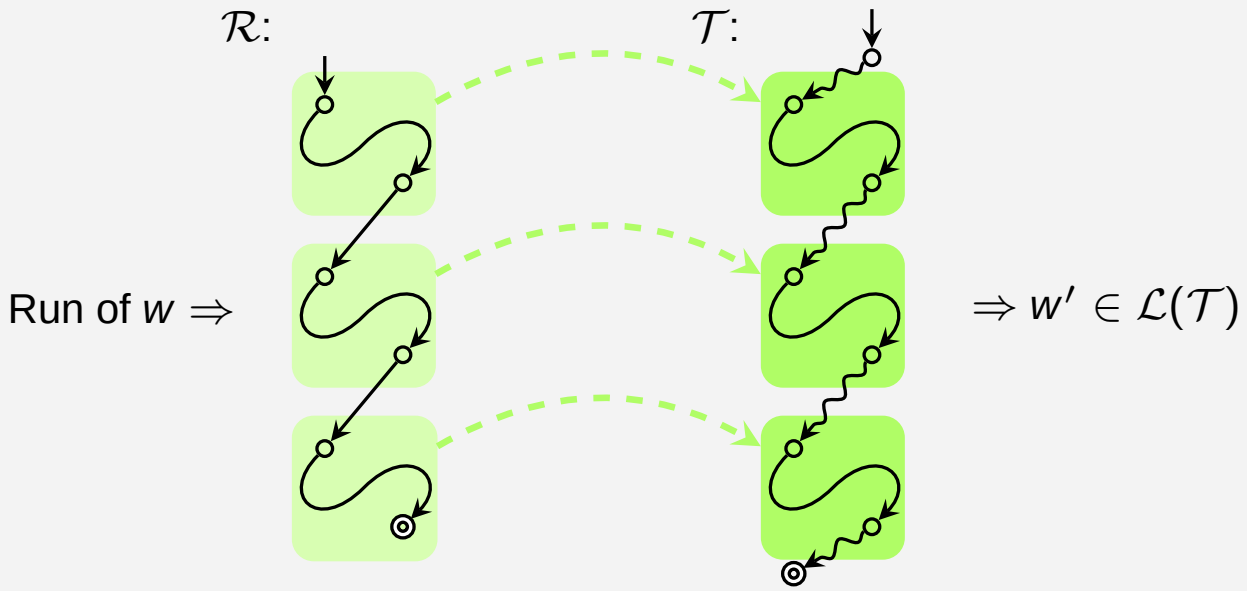


Characterization of bounded repairability

Theorem

Given two NFA $\mathcal{R}$ and $\mathcal{T}$, there is a repair strategy from $\mathcal{L}(\mathcal{R})$ into $\mathcal{L}(\mathcal{T})$ with uniformly bounded cost iff every path in $\mathrm{dag}(\mathcal{R})$ is covered by some path in $\mathrm{dag}^*(\mathcal{T})$.

Proof sketch ($\Leftarrow$)



Complexity results

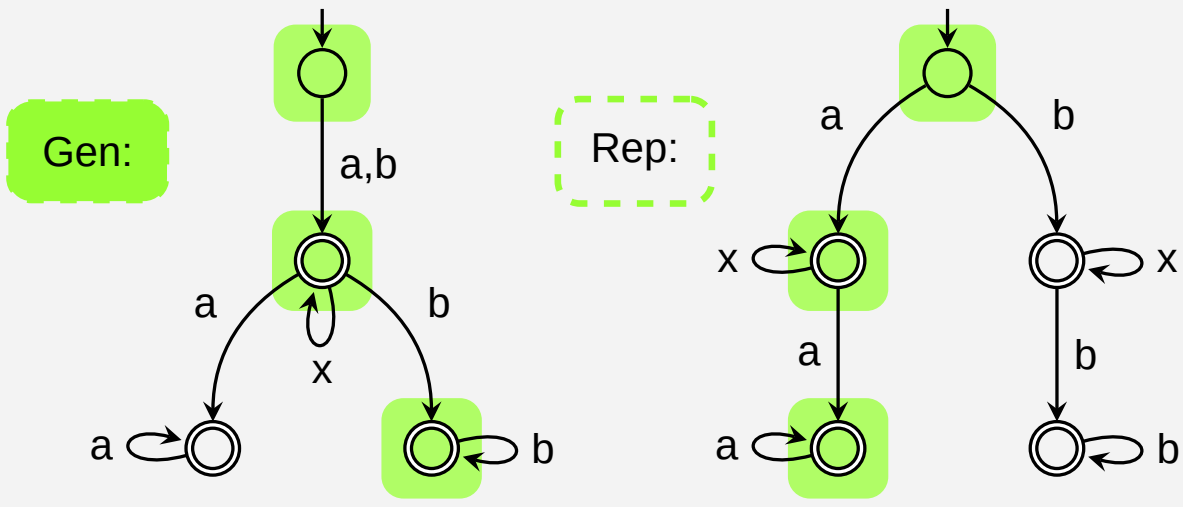
- NFA: the problem is PSPACE-complete.
  - Upper bound: Check all paths in $\mathrm{dag}(\mathcal{R})$ using polynomial space.
  - Lower bound: containment of NFA.
- DFA: the problem is coNP-complete.
  - Upper bound: Guess a path in $\mathrm{dag}(\mathcal{R})$ and check coverability.
  - Lower bound: validity of propositional formulas in DNF.
- Threshold problem is PSPACE-complete.

Streaming case

Game between a Generator (Gen) and Repairer (Rep).

Example

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



Streaming case

Game between a Generator (Gen) and Repairer (Rep).

Theorem

Given two DFA $\mathcal{R}$ and $\mathcal{T}$, there is a streaming repair strategy from $\mathcal{L}(\mathcal{R})$ into $\mathcal{L}(\mathcal{T})$ with uniformly bounded cost iff Repairer has a winning strategy over the reachability game defined for $\mathrm{dag}(\mathcal{R})$ and $\mathrm{dag}^*(\mathcal{T})$.

Complexity results in the streaming case

- DFA: the problem is PTIME-complete.
  - Upper bound: Solve the reachability game over $\mathrm{dag}(\mathcal{R})$ and $\mathrm{dag}^*(\mathcal{T})$.
  - Lower bound: Evaluation of boolean circuits.
- NFA: the problem is PSPACE-hard and in EXPTIME.
  - Upper bound: Direct subset construction.
  - Lower bound: Language containment.

The exact complexity for NFA is an open problem.

More results about bounded repairability

In the LICS paper "Regular repair of specifications":

- Connections with distance automata and energy games.
- Threshold problem.
- Complexity for LTL specifications.
- Infinite words.

Outline

Setting

Bounded Repairability

Asymptotic cost

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

$$\begin{array}{rcll} R : (a+b)^* & & T : (ab+b)^* & \\ (\mathrm{a\ a})^N & \Rightarrow & (\mathrm{a\ b})^N & \to \mathbf{A}(R,T) = \frac{1}{2} \\ & & & \\ R : a^+b^+ & & T : a^+c^+ & \\ \mathrm{a\ b}^N & \Rightarrow & \mathrm{a\ c}^N & \to \mathbf{A}(R,T) = \mathbf{1} \end{array}$$

We show that:

- $\mathbf{A}(R,T)$ is rational and can be effectively computed.
- In this talk: only focus on $\mathbf{A}(\Sigma^*,T)$.

Distance automata

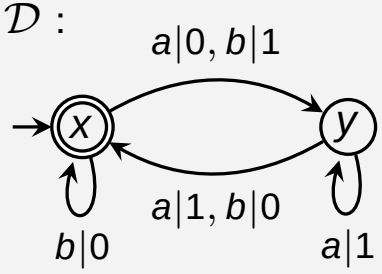
Definition

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

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

Example

$\mathcal{D}$:



$w =$ a a a b

$$\rho_1(w) = x \xrightarrow{a|0} y \xrightarrow{a|1} y \xrightarrow{a|1} x \xrightarrow{b|0} x$$
$$\rho_2(w) = x \xrightarrow{a|0} y \xrightarrow{a|1} x \xrightarrow{a|0} y \xrightarrow{b|0} x$$

} runs

$\text{cost}(\rho_1) = 2 \quad \text{cost}(\rho_2) = 1$

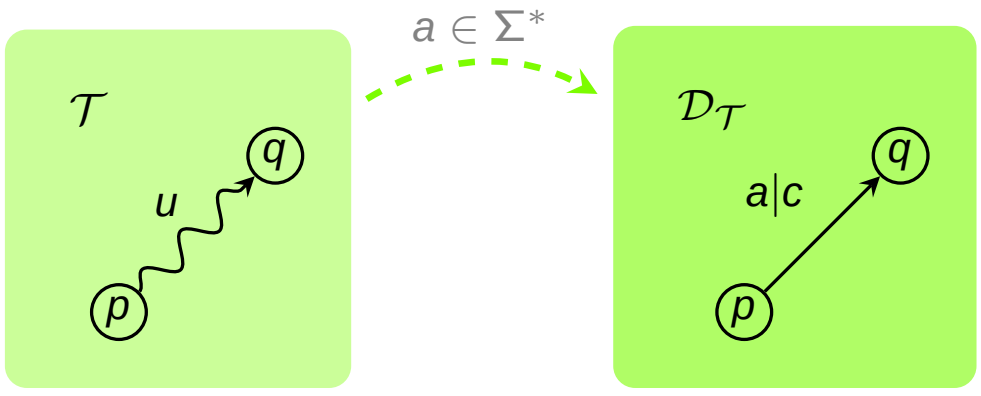
$\mathcal{D}(w) = \min\{\text{cost}(\rho) \mid \rho \text{ is a run of } w \text{ over } \mathcal{D}\}$

Edit-distance automata of a regular language

Given an automaton $\mathcal{T} = (\Delta, Q, \delta, q_0, F)$.

Definition

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



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

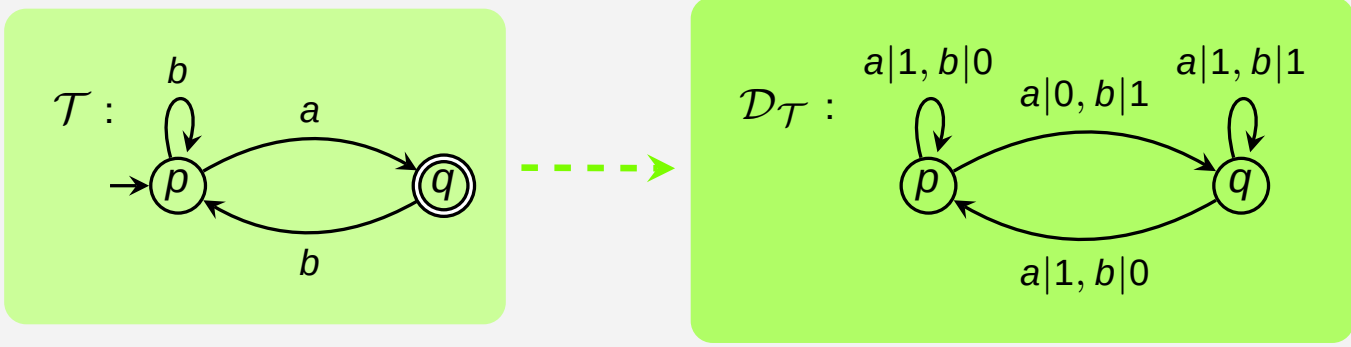
Edit-distance automata of a regular language

Given an automaton $\mathcal{T} = (\Delta, Q, \delta, q_0, F)$.

Definition

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

Example



Theorem

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

The asymptotic cost problem for a distance automaton is undecidable

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

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

Theorem

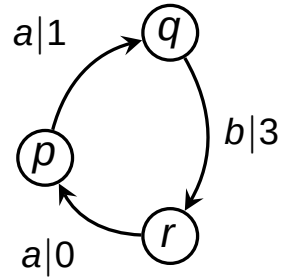
The problem of deciding whether $\mathbf{A}(\mathcal{D}) \leq \frac{1}{2}$ is undecidable given an arbitrary distance automaton $\mathcal{D}$.

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

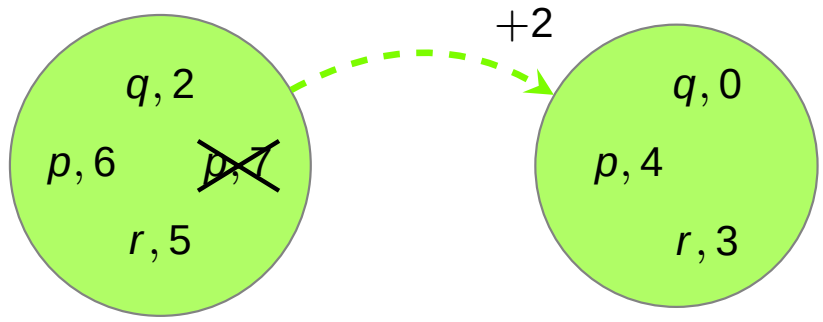
Determinization of distance automata

Extension of the subset construction by storing tuples (state, value).

$\mathcal{D}$:

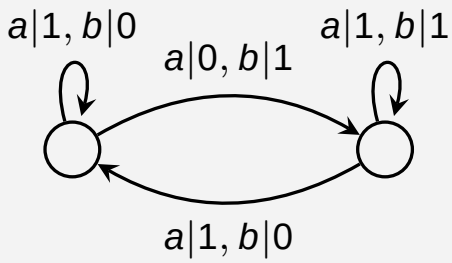


det($\mathcal{D}$) :

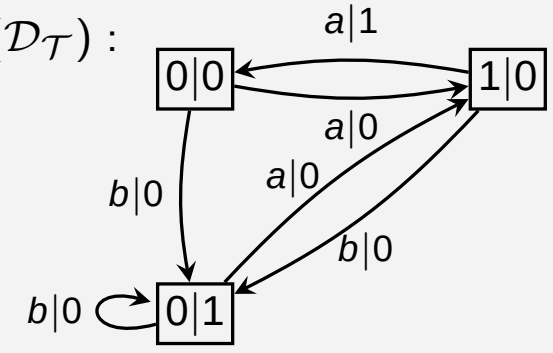


Example

$\mathcal{D}_\mathcal{T}$:



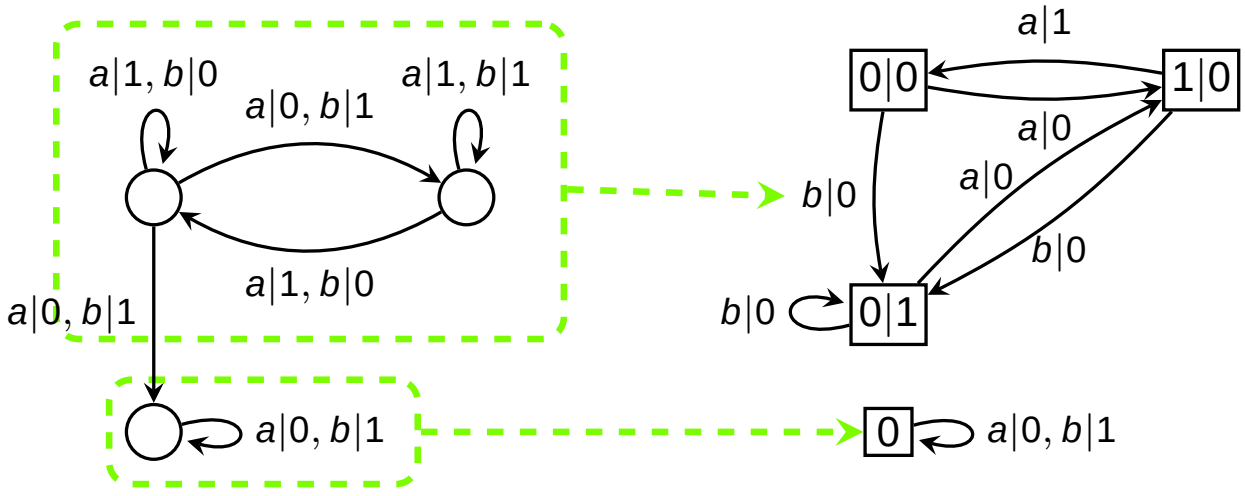
det($\mathcal{D}_\mathcal{T}$) :



Notice that det($\mathcal{D}$) could be infinite.

The strongly connected components of $\mathcal{D}_T$ are determinizable

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



Proposition

$\det(\mathcal{D}_T|C)$ is finite for every $C \in \mathrm{SCC}(\mathcal{D}_T)$.

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:

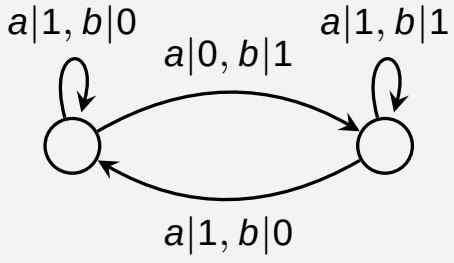
$$\mathbf{A}(\mathcal{D}_{\mathcal{T}}) = \max\left\{ \frac{\text{cost}(L)}{|L|} \mid L \text{ is a simple cycle of } \det(\mathcal{D}_{\mathcal{T}}) \right\}$$

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

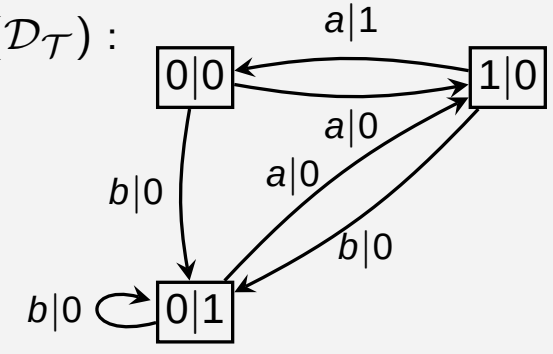
Example

$T : (ab + b)^*$

$\mathcal{D}_{\mathcal{T}}$:

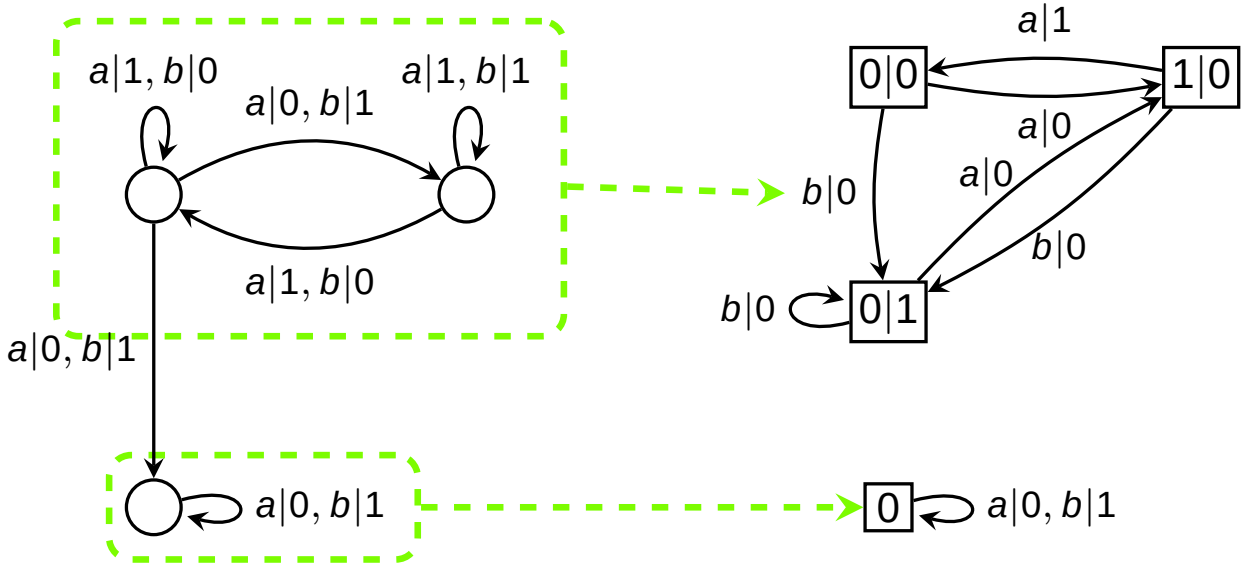


$\det(\mathcal{D}_{\mathcal{T}})$:



General case

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



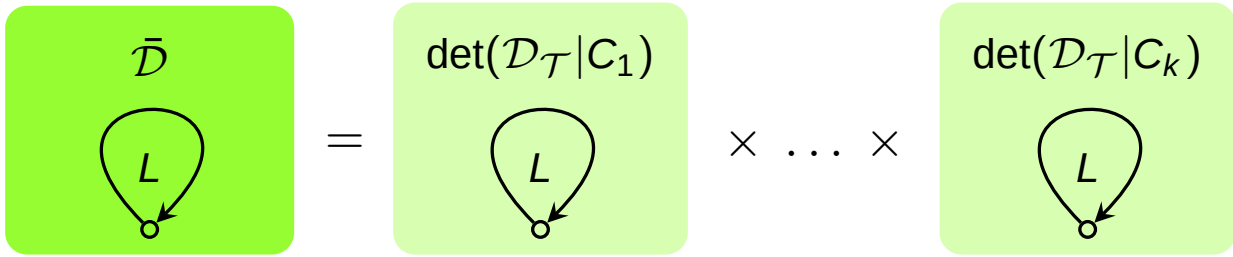
A$(\mathcal{D}_\mathcal{T})$ cannot be computed as a function of the simple cycles of $\det(\mathcal{D}_\mathcal{T}|C)$ for all $C \in \mathcal{D}_\mathcal{T}$.

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

$\text{cost}_j(L_i)$ = cost of the projection of the simple cycle L_i into the j-th component of $\bar{\mathcal{D}}$

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

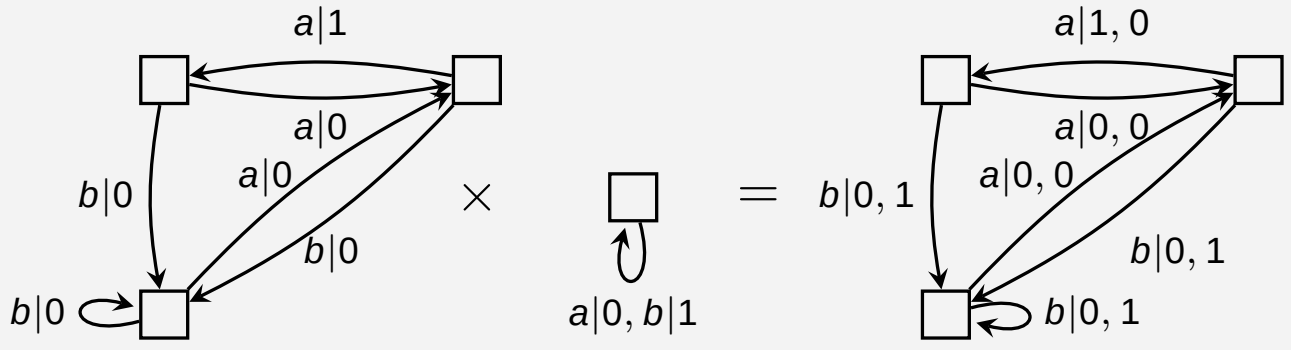
Theorem

$$\mathbf{A}(\mathcal{D}_\mathcal{T}) = \max_{\alpha_1,\ldots,\alpha_m \geq 0} \min_{1 \leq j \leq k} \frac{\sum_{1 \leq i \leq m} \alpha_i \cdot \mathrm{cost}_j(L_i)}{\sum_{1 \leq i \leq m} \alpha_i \cdot |L_i|}$$

- k: number of SCC of $\mathcal{D}_\mathcal{T}$.
- m: number of cycles in $\bar{\mathcal{D}}$.

Example

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



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

$$\mathbf{A}(\mathcal{D}_{\mathcal{T}}) = \underbrace{\max_{\alpha_1,\ldots,\alpha_m \geq 0} \min_{1 \leq j \leq k} \frac{\sum_{1 \leq i \leq m} \alpha_i \cdot \mathrm{cost}_j(L_i)}{\sum_{1 \leq i \leq m} \alpha_i \cdot |L_i|}}_{\mathbf{E}(\mathcal{D}_{\mathcal{T}})}$$

$\mathbf{E}(\mathcal{D}_{\mathcal{T}})$ can be seen as a linear programming problem.

$$\text{MAXIMIZE} \quad y \quad \text{SUBJECT TO} \quad \sum_{1 \leq i \leq m} c_{i,j} \cdot x_i \geq y \qquad \forall 1 \leq j \leq k$$
$$\sum_{1 \leq i \leq m} x_i \leq 1, \quad x_i \geq 0 \qquad \forall 1 \leq i \leq k.$$

- $c_{i,j} = \frac{\mathrm{cost}_j(L_i)}{|L_i|}$, and
- $x_1, \ldots, x_m$ represent the values $\alpha_1 \cdot |L_i|, \ldots, \alpha_m \cdot |L_i|$.

$\mathbf{A}(\mathcal{D}_{\mathcal{T}})$ 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 $\bar{\mathcal{D}}$ 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.

More results about the asymptotic cost

In the ICALP paper "The cost of traveling between languages":

- Asymptotic cost in the streaming case.
- Reduction to mean-payoff games.
- Complexity remains in PTIME.

Conclusions and current work

- Bounded Repairability:
  - Characterization using coverability of paths.
  - Connections with reachability games in the streaming setting.
- Asymptotic cost:
  - Connections with distance automata and its determinization.
  - The value is rational and can be effectively computed.
- Current work:
  - Repairing tree regular languages.