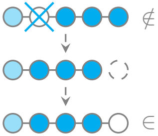
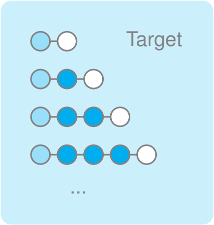
What do you do
if a computational object fails a specification?

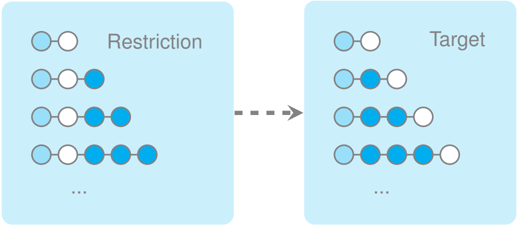




1. Non-deterministic finite automata
2. Deterministic finite automata
3. Linear Temporal Logic (LTL)

Only over finite words

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



1. Non-deterministic finite automata
2. Deterministic finite automata
3. Linear Temporal Logic (LTL)

Only over finite words

Can we repair each word
with a bounded number of modifications?

Bounded Repair Problem

Example

$R : (ba)^* b$		$T : (a^* b)^*$	
$(\mathrm{b\ a})^N\ \mathrm{b}$	⇢	$\mathrm{a}\ (\mathrm{b\ a})^N\ \mathrm{b}$	✓
$R : (a + b)^*$		$T : (a + bb)^*$	
$(\mathrm{a\ b})^N$	⇢	$(\mathrm{a\ b}\ \not{\mathrm{a}}\ \mathrm{b})^{\frac{N}{2}}$	✗

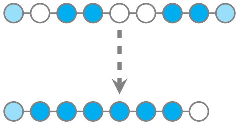
Can we repair each word

with a bounded number of modifications?

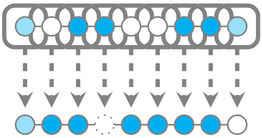
Bounded Repair Problem

Different ways of repairing:

Arbitrary



Streaming



Can we repair each word with a bounded number of modifications?

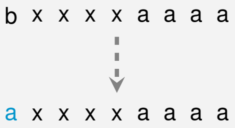
Bounded Repair Problem

Example

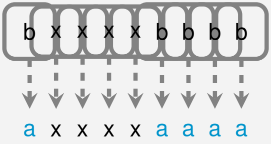
R : $(a + b)\, x^* \,(a^* + b^*)$

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

Arbitrary ✓



Streaming ✗



We study the bounded repair problem in deep

1. Non-streaming:
   - Characterization based on strongly connected components.
   - Tight complexity bounds.
2. Streaming:
   - Characterization based on reachability games.
   - Optimal repair strategies.
   - Independent of lookahead and variants of cost function.
   - Complexity bounds.
3. Connections with distance automata and energy games.

Regular Repair of Specifications

Cristian Riveros
Michael Benedikt
Gabriele Puppis

University of Oxford
LICS 2011

Outline

Setting

Non-streaming

Streaming

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),
  - Non-deterministic finite automata (NFA), or
  - Linear temporal logic (LTL).
- In this talk:
  - All automata are trim.
  - All LTL formulas are over finite strutures.

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:

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

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

Both computable in PTIME

(Wagner and Fisher 1974, Wagner 1974).

Bounded repairability

A repair strategy is a function $f : R \to T$.

Definition

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

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

Generalization of language containment.

Outline

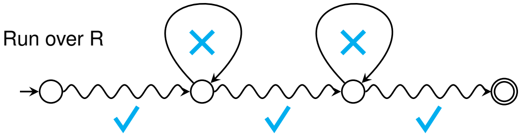
Setting

Non-streaming

Streaming

Intuition of bounded repairability

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



Intuition of bounded repairability

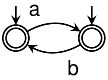
We should not repair during the cyclic behavior of R.

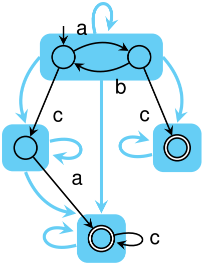
Definition

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

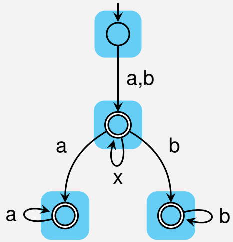
Definition

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

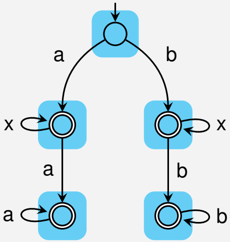
$$\mathcal{L}(\mathcal{R}|C_i) \subseteq \mathcal{L}(\mathcal{T}|C'_i) \quad \text{for all } i \leq k$$

Example

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



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

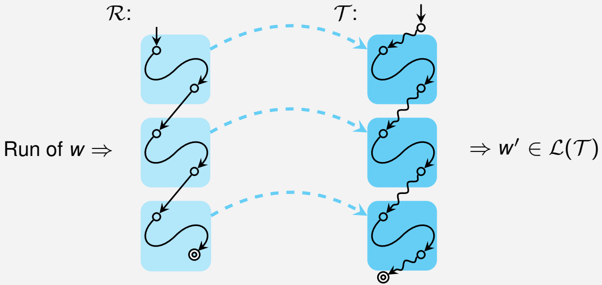


Characterization of bounded repairability

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 dag($\mathcal{R}$) is covered by some path in dag$^*(\mathcal{T})$.

Proof sketch (⇐)



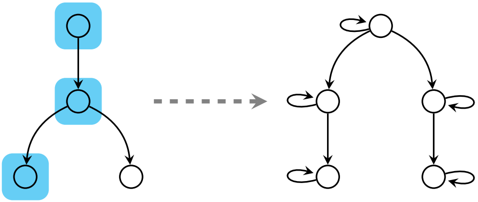
Complexity results

	fixed	DFA	NFA	LTL
fixed	Const	PTIME	PSPACE	PSPACE
DFA	PTIME	CoNP	PSPACE	PSPACE
NFA	PTIME	CoNP	PSPACE	PSPACE
LTL	PSPACE	PSPACE	PSPACE	CoNEXP

Upper bound intuition:

Restriction: dag($\mathcal{R}$)

Target: dag*($\mathcal{T}$)



Complexity results

	fixed	DFA	NFA	LTL
fixed	Const	PTIME	PSPACE	PSPACE
DFA	PTIME	CoNP	PSPACE	PSPACE
NFA	PTIME	CoNP	PSPACE	PSPACE
LTL	PSPACE	PSPACE	PSPACE	CoNEXP

Threshold problem: Given $k \in \mathbb{N}$, determine if:

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

Threshold problem is PSPACE-complete
for languages R and T given by DFA or NFA.

Outline

Setting

Non-streaming

Streaming

Streaming Repair Strategies

- 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,
  - with k-lookahead for some $k \in \mathbb{N}$.
- Two possible cost for a streaming repair strategy $f : R \to T$:
  - edit-cost$(u, f) = \text{dist}(u, f(u))$
  - aggregate-cost$(u, f) = \sum_{i=0}^{n} \text{dist}(u_i, v_i)$ with

$$q_0 \xrightarrow{u_1/v_1} q_1 \xrightarrow{u_2/v_2} \dots \xrightarrow{u_n/v_n} q_n$$

be a run of the sequential transducer.

Streaming case

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

Theorem

Given two DFA $\mathcal{R}$ and $\mathcal{T}$, the following condition are equivalent:

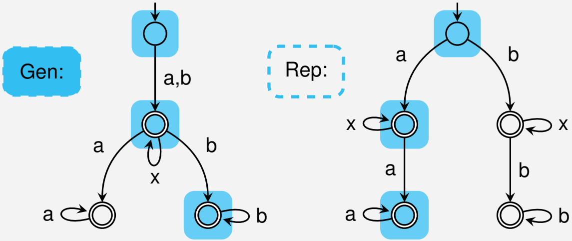
1. there is a k-lookahead streaming strategy with uniformly bounded edit cost,
2. Repairer has a winning strategy over a reachability game defined over $\text{dag}(\mathcal{R})$ and $\text{dag}^*(\mathcal{T})$,
3. there is a 0-lookahead streaming strategy with worst-case aggregate cost at most $(1 + |\text{dag}(\mathcal{R})|) \cdot |\mathcal{T}|$.

Streaming case

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

Example of the reachability game

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



Complexity results in the streaming case

	fixed	DFA	NFA	LTL
fixed	Const	PTIME	PSPACE	PSP, EXPSP
DFA	PTIME	PTIME	PSPACE	PSP, EXPSP
NFA	PT, PSP	PT, PSP	PSP, EXP	PSP, 2EXP
LTL	PSP, EXPSP	PSP, EXPSP	PSP, 2EXP	EXPSP, 2EXP

Upper bound:

- Solve the reachability game over dag($\mathcal{R}$) and dag($\mathcal{T}$).
- This is well known to be in PTIME.

Complexity results in the streaming case

	fixed	DFA	NFA	LTL
fixed	Const	PTIME	PSPACE	PSP, EXPSP
DFA	PTIME	PTIME	PSPACE	PSP, EXPSP
NFA	PT, PSP	PT, PSP	PSP, EXP	PSP, 2EXP
LTL	PSP, EXPSP	PSP, EXPSP	PSP, 2EXP	EXPSP, 2EXP

- Upper bound: Direct subset construction.
- Lower bound: Language containment.

The exact complexity for NFA is an open problem.

Connections with distance automata and energy games

Given regular languages R and T:

- There exists a distance automaton $\mathcal{D}_{R,T}$ such that:

  R is bounded repairable into T

  $\Longleftrightarrow$

  the cost function computed by $\mathcal{D}_{\mathcal{R},\mathcal{T}}$ is uniformly bounded.

- There exists an energy game $\mathcal{G}_{R,T}$ such that:

  R is streaming bounded repairable into T

  $\Longleftrightarrow$

  energy player has a winning strategy over $\mathcal{G}_{R,T}$.

Conclusion and future work

1. Non-streaming:
   - Characterization using coverability of paths.
   - Tight complexity bounds for DFA, NFA and LTL.
2. Streaming:
   - Characterization based on reachability games.
   - Optimal repair strategies.
   - Independent of lookahead and variants of cost function.
3. Future work:
   - “The cost of traveling between languages”, in ICALP 2011.
   - Repairing tree regular languages.

Regular Repair of Specifications

Cristian Riveros
Michael Benedikt
Gabriele Puppis

University of Oxford
LICS 2011