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)
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**

\[
\begin{align*}
R : (ba)^* b & \quad T : (a^* b)^* \\
(b a)^N b & \quad \rightarrow \quad a (b a)^N b \quad \checkmark \\
R : (a + b)^* & \quad T : (a + bb)^* \\
(a b)^N & \quad \rightarrow \quad (a b \ a b)^N \quad \times
\end{align*}
\]
Can we repair each word with a bounded number of modifications?
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**

\[ b x x x x a a a a \]

\[ a x x x x a a a a \]

**Streaming**

\[ b x x x x b b b b \]

\[ a x x x x a a a a \]
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 structures.
Repairability using edit operations

Edit operations: deletion, insertion, and relabeling.

- delete(2) → \(\text{\color{red}X}\) \(\rightarrow\) \(\text{\color{blue}•} - \text{\color{blue}•} - \text{\color{blue}•} - \text{\color{blue}•}\)
- insert(3, \(\text{\color{blue}•}\)) → \(\text{\color{blue}•} - \text{\color{blue}•} - \text{\color{blue}•} - \text{\color{blue}•} - \text{\color{blue}•}\)
- relabel(4, \(\text{\color{blue}•}\)) → \(\text{\color{blue}•} - \text{\color{blue}•} - \text{\color{blue}•} - \text{\color{blue}•}\)

- All operations have cost equal to 1.

Definition

For words \(u, v\) and language \(T\):

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

\[
\text{dist}(u, T) = \min_{v \in T} \{ \text{dist}(u, v) \}
\]

Both computable in PTIME
A repair strategy is a function $f : R \rightarrow T$.

**Definition**

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

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

Generalization of language containment.
Outline

Setting

Non-streaming

Streaming
Intuition of bounded repairability

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

Run over $R$
We should not repair during the cyclic behavior of $R$.

Definition

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

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

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

$L(A|C)$ contains the cyclic behavior of $C$ in $A$. 

\[ L(A|C) \]
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:

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

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

Proof sketch ($\iff$)

Run of $w \Rightarrow w' \in \mathcal{L}(\mathcal{T})$
### Complexity results

<table>
<thead>
<tr>
<th></th>
<th>fixed</th>
<th>DFA</th>
<th>NFA</th>
<th>LTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>Const</td>
<td>PTIME</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>DFA</td>
<td>PTIME</td>
<td>CoNP</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>NFA</td>
<td>PTIME</td>
<td>CoNP</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE</td>
<td>PSPACE</td>
<td>PSPACE</td>
<td>CoNEXP</td>
</tr>
</tbody>
</table>

**Upper bound intuition:**

Restriction: $\text{dag}(\mathcal{R})$  
Target: $\text{dag}^*(\mathcal{T})$
Complexity results

<table>
<thead>
<tr>
<th></th>
<th>fixed</th>
<th>DFA</th>
<th>NFA</th>
<th>LTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>Const</td>
<td>PTIME</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>DFA</td>
<td>PTIME</td>
<td>CoNP</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>NFA</td>
<td>PTIME</td>
<td>CoNP</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE</td>
<td>PSPACE</td>
<td>PSPACE</td>
<td>CoNEXP</td>
</tr>
</tbody>
</table>

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

$$\text{dist}(u, T) \leq k$$ 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
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 $f : R \to T$:

- $\text{edit-cost}(u, f) = \text{dist}(u, f(u))$
- $\text{aggregate-cost}(u, f) = \sum_{i=0}^{n} \text{dist}(u_i, v_i)$ with

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

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**

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

\[ T : a \, x^* \, a^* + b \, x^* \, b^* \]
Complexity results in the streaming case

<table>
<thead>
<tr>
<th></th>
<th>fixed</th>
<th>DFA</th>
<th>NFA</th>
<th>LTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>Const</td>
<td>PTIME</td>
<td>PSPACE</td>
<td>PSP, EXPSP</td>
</tr>
<tr>
<td>DFA</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PSPACE</td>
<td>PSP, EXPSP</td>
</tr>
<tr>
<td>NFA</td>
<td>PT, PSP</td>
<td>PT, PSP</td>
<td>PSP, EXP</td>
<td>PSP, 2EXP</td>
</tr>
<tr>
<td>LTL</td>
<td>PSP, EXPSp</td>
<td>PSP, EXPSp</td>
<td>PSP, 2EXP</td>
<td>EXPSp, 2EXP</td>
</tr>
</tbody>
</table>

Upper bound:

- Solve the reachability game over dag(\(R\)) and dag(\(T\)).
- This is well known to be in PTIME.
Complexity results in the streaming case

<table>
<thead>
<tr>
<th></th>
<th>fixed</th>
<th>DFA</th>
<th>NFA</th>
<th>LTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>Const</td>
<td>PTIME</td>
<td>PSP</td>
<td>PSP, EXPSP</td>
</tr>
<tr>
<td>DFA</td>
<td>PTIME</td>
<td>PTIME</td>
<td>PSP</td>
<td>PSP, EXPSP</td>
</tr>
<tr>
<td>NFA</td>
<td>PT, PSP</td>
<td>PT, PSP</td>
<td>PSP, EXP</td>
<td>PSP, 2EXP</td>
</tr>
<tr>
<td>LTL</td>
<td>PSP, EXPSP</td>
<td>PSP, EXPSP</td>
<td>PSP, 2EXP</td>
<td>EXPSP, 2EXP</td>
</tr>
</tbody>
</table>

- **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 $D_{R,T}$ such that:
  
  $R$ is bounded repairable into $T$

  \[ \iff \]

  the cost function computed by $D_{R,T}$ is uniformly bounded.

- There exists an energy game $G_{R,T}$ such that:

  $R$ is streaming bounded repairable into $T$

  \[ \iff \]

  energy player has a winning strategy over $G_{R,T}$. 

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:
   ▶ Repairing tree regular languages.