Inferring DFA without Negative Examples

Florent Avellaneda & Alexandre Petrenko

Computer Research Institute of Montréal (Canada)

ICGI 2018 - 6 September 2018

Inference problem

Input: Positive observations S_+ and negative observations S_-

Output: The "best" model consistent with observations

What means "best"?

"best": The model most likely to be the real one

Law of parsimony

Among competing hypotheses, the one with the fewest assumptions should be selected

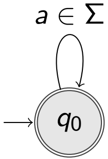
The law implies that the simplest conjecture should be the best. For DFA, we generally use the number of states as the unit of measurement for the complexity

Problem to Solve

Input: Positive observations S_+ ~~and negative observations S_-~~

Output: The "best" model consistent with observations

Constraint: It makes no sense to look for a DFA with a minimal number of states



Ideas for defining "simplest"

Idea 1: Try to minimize the recognized language

$$\mathcal{A} \leq \mathcal{A}' \text{ if and only if } L(\mathcal{A}) \subseteq L(\mathcal{A}')$$

Idea 2: Set a maximum number of states smaller than that in the given positive examples

Overview

1. Inferring Simplest DFA from Positive Examples
2. Checking the Uniqueness of a Solution
3. Finding Characteristics Positive Examples
4. Case Study
5. Conclusion & Perspectives

Overview

1. Inferring Simplest DFA from Positive Examples
2. Checking the Uniqueness of a Solution
3. Finding Characteristics Positive Examples
4. Case Study
5. Conclusion & Perspectives

Definitions

Definition (n-conjecture)

A DFA $\mathcal{A}$ is an n-conjecture for S_+ if $S_+ \subseteq L(\mathcal{A})$ and $\mathcal{A}$ has at most n states

Definition (simplest)

A minimal DFA $\mathcal{A}$ is a simplest n-conjecture for S_+ if for each n-conjecture $\mathcal{A}'$ for S_+, we have $L(\mathcal{A}') \not\subset L(\mathcal{A})$

Problem statement: Given an integer n and a set of positive examples S_+, find a simplest n-conjecture for S_+

Method

Let $\mathcal{A}$ be the chaos DFA and $\mathcal{A}'$ the empty DFA

1. If $S_+ \not\subseteq L(\mathcal{A}')$: Let $w \in S_+ \setminus L(\mathcal{A}')$ and add the constraint
   "***w* has to be accepted**"

2. If $S_+ \subseteq L(\mathcal{A}')$ and $L(\mathcal{A}') \not\subseteq L(\mathcal{A})$: Let $w \in L(\mathcal{A}') \setminus L(\mathcal{A})$ and add the constraint
   "***w* must not be accepted**"

3. If $S_+ \subseteq L(\mathcal{A}')$ and $L(\mathcal{A}) \subseteq L(\mathcal{A}')$: Replace $\mathcal{A}'$ by $\mathcal{A}$ and add constraint
   "**excluding solution** $\mathcal{A}'$"

Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$

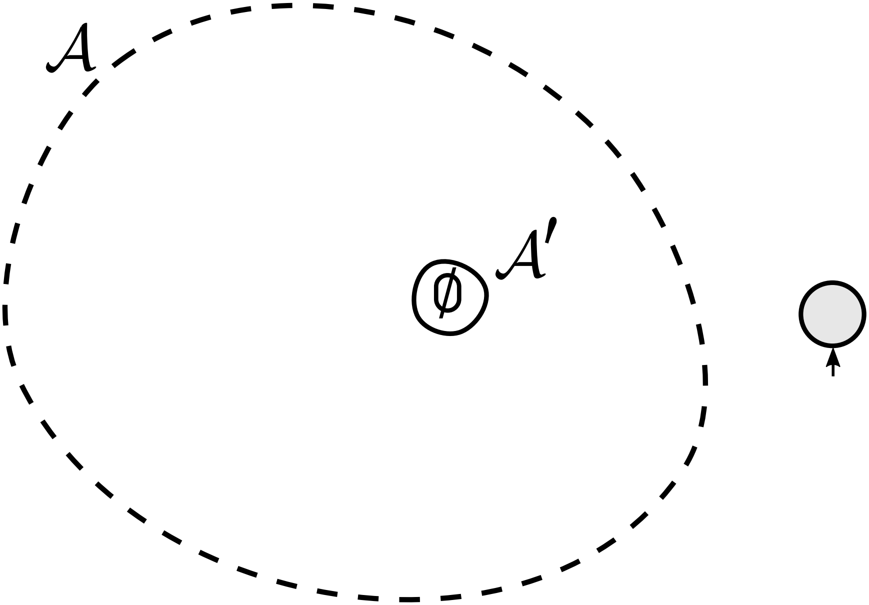


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$

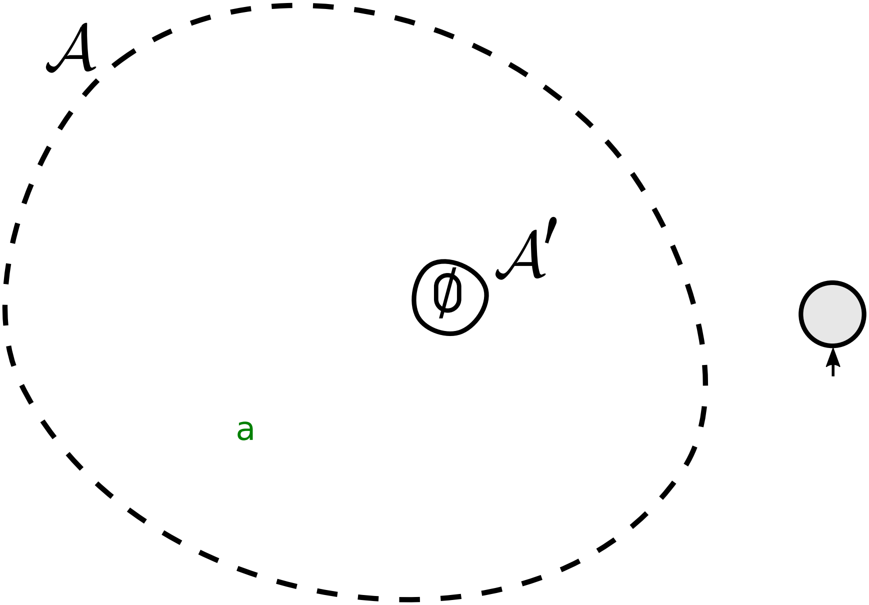
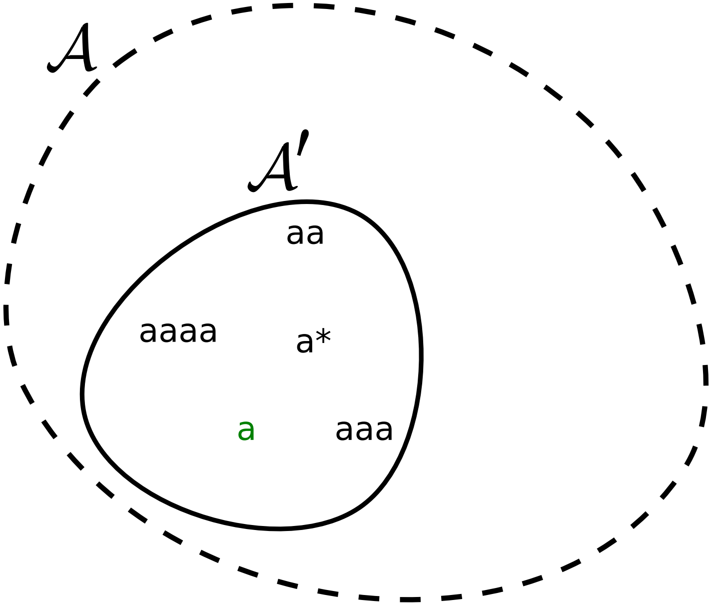


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$



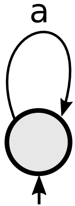
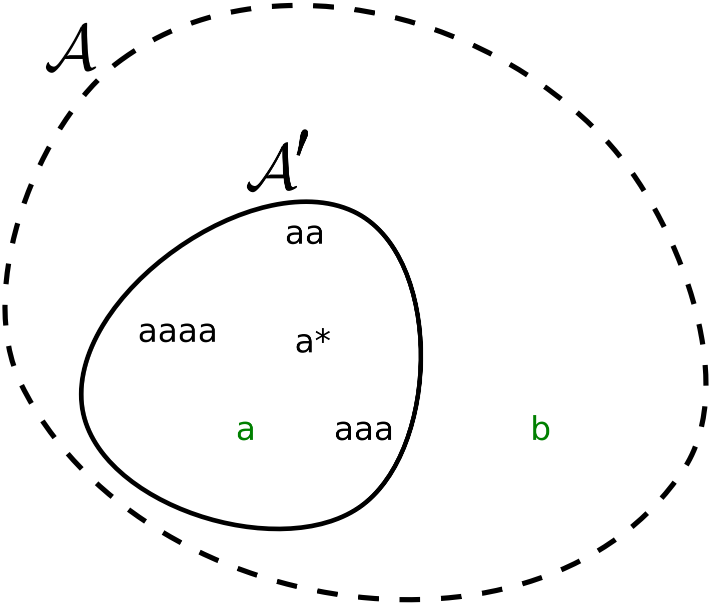


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$



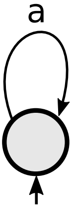
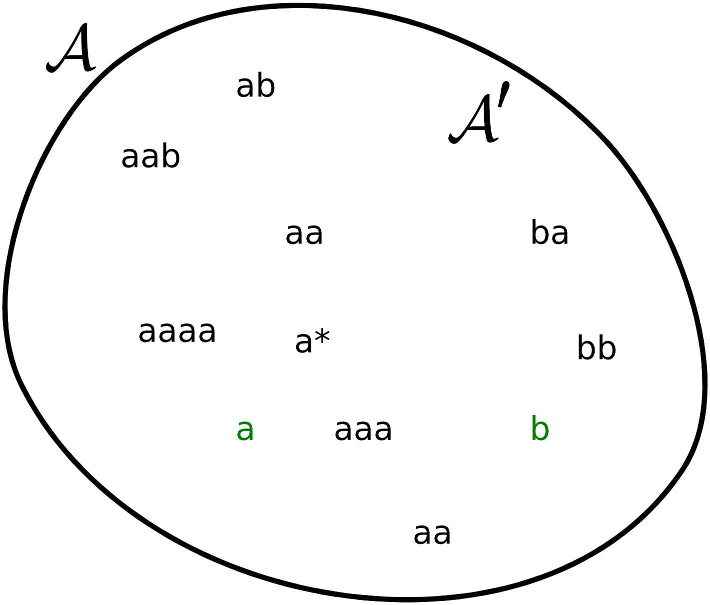


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$



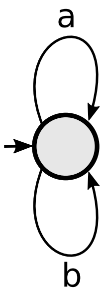


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$

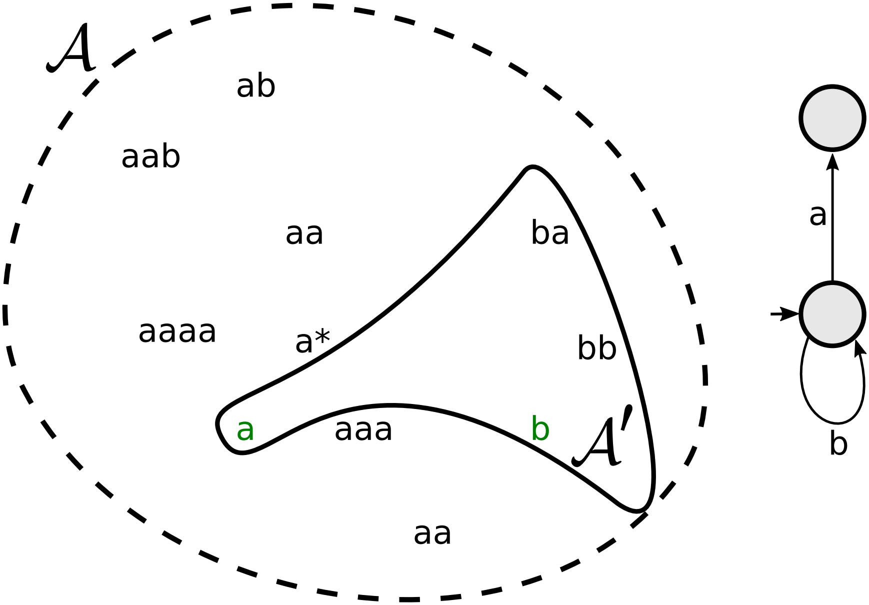


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$

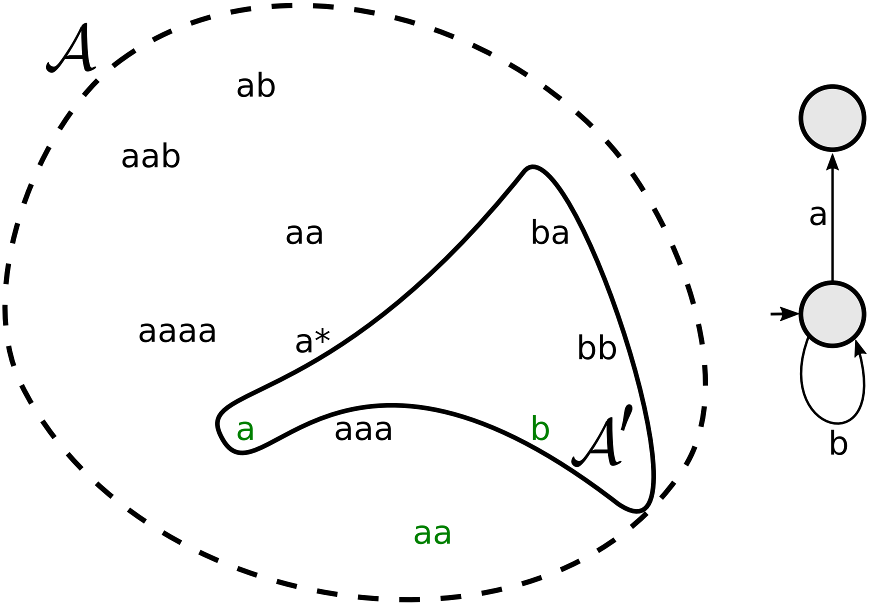


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$

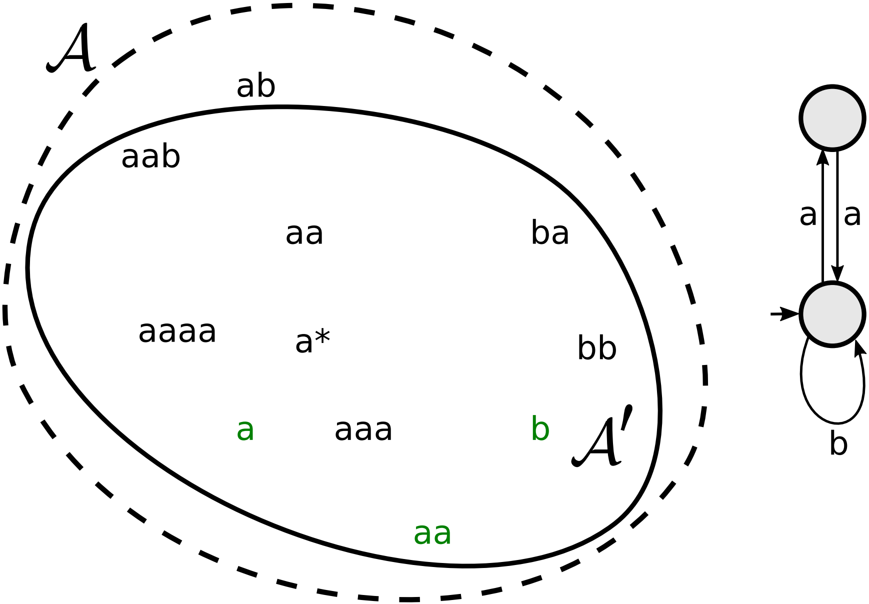


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$

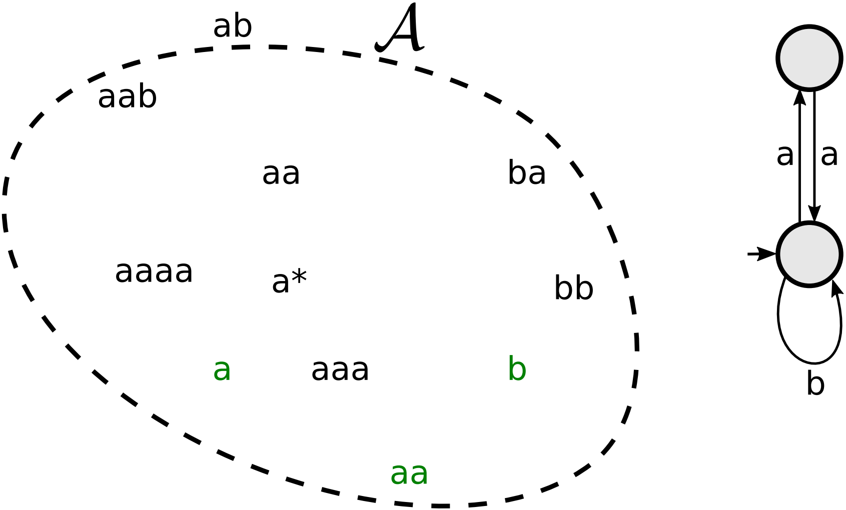


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$

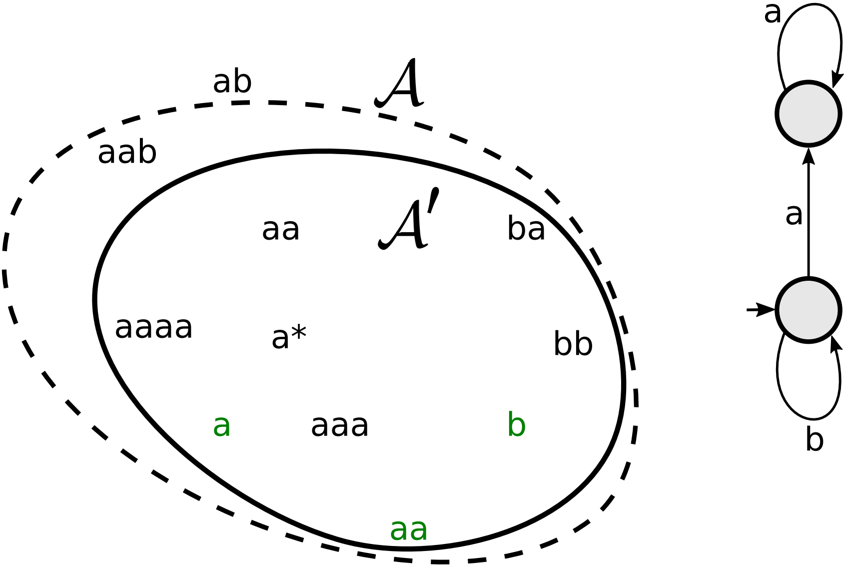
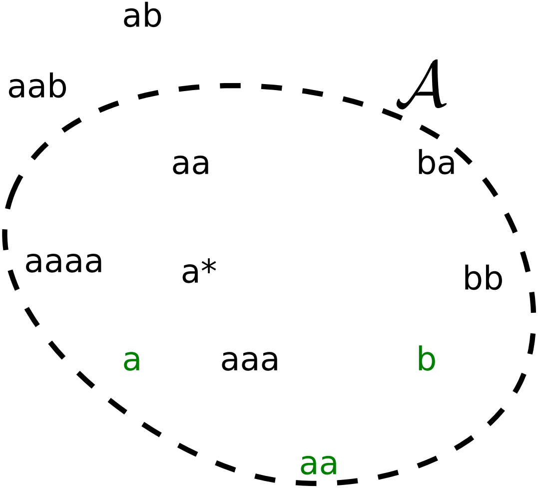


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$



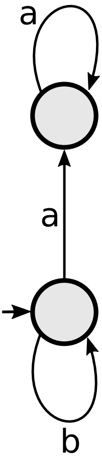
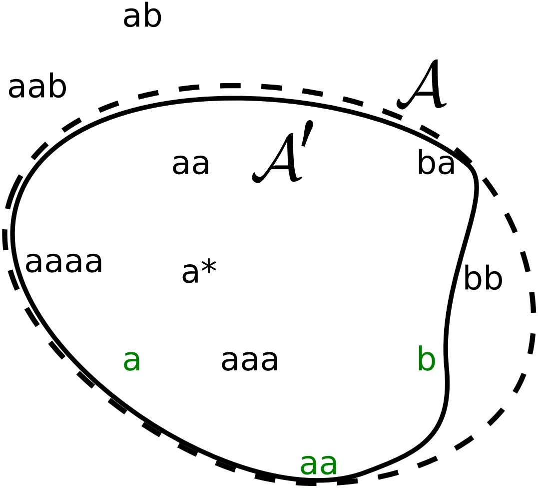


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$



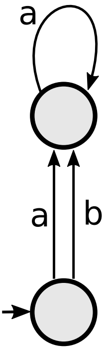
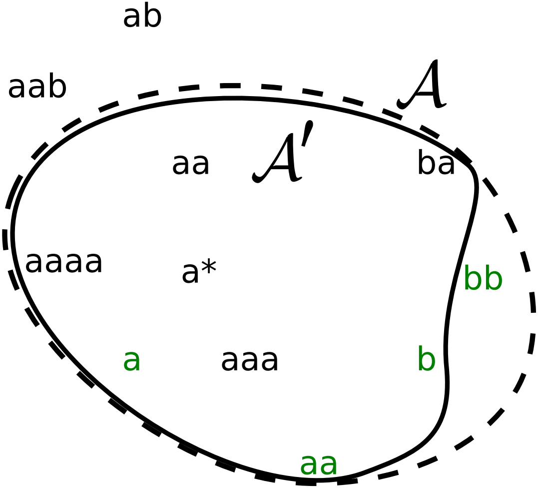


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$



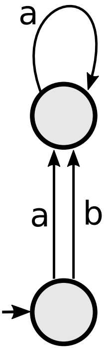


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$

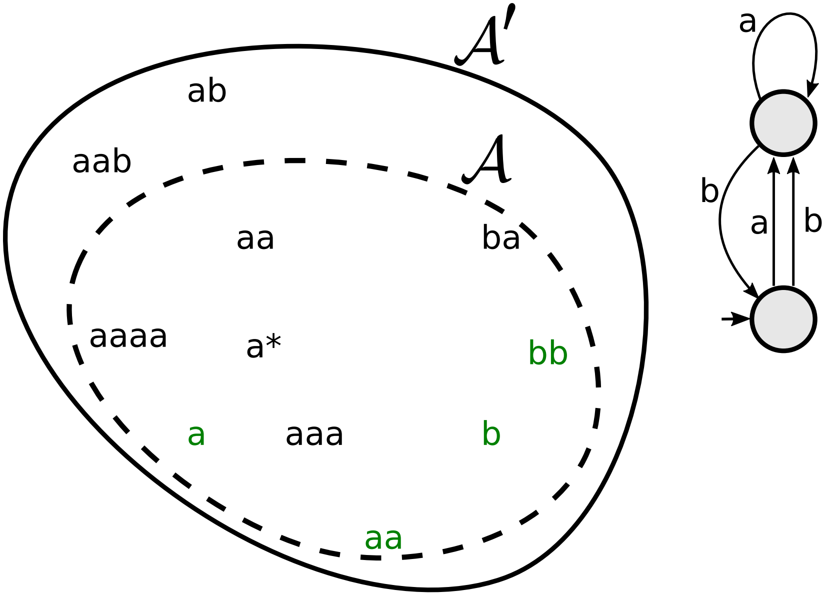
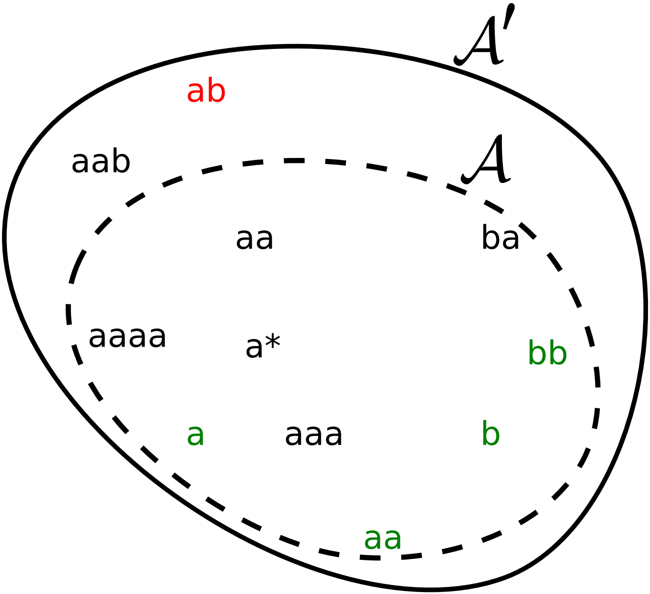


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$



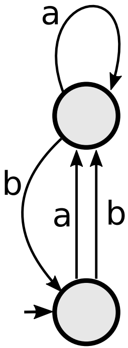
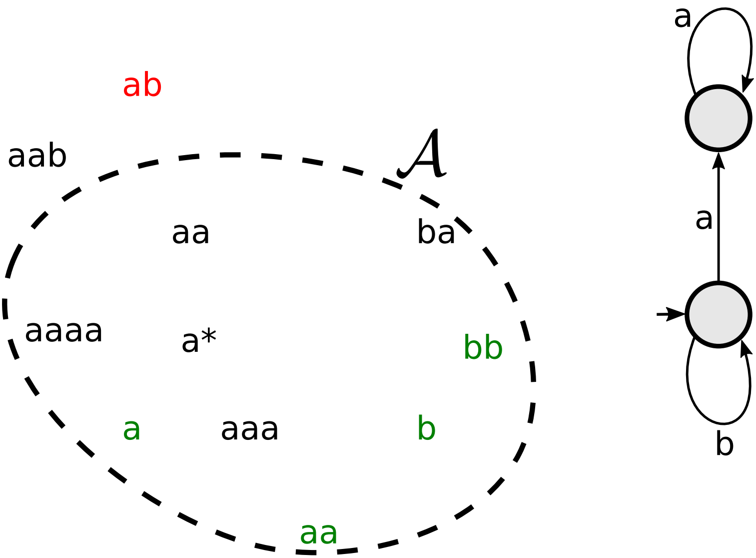


Illustration: $S_+ = \{a, aa, aaa, b, bb, bbb\}$



Input: Positive examples S_+ and an integer n
Output: A simplest n-conjecture for S_+ and negative examples S_-
Initialize C to $\emptyset$, S_- to $\emptyset$ and $\mathcal{A}$ to $\mathcal{A}_{Chaos}$
while *C is satisfiable* **do**
    Let $\mathcal{A}'$ be a DFA of a solution of C.
    if $S_+ \not\subseteq L(\mathcal{A}')$ **then**
        Let w be a shortest string in $S_+ \setminus L(\mathcal{A}')$.
        $C \leftarrow C \wedge C_w$, where C_w is clauses encoding the requirement that w must be in the conjecture.
    else
        if $L(\mathcal{A}') \subseteq L(\mathcal{A})$ **then**
            $C \leftarrow C \wedge C_{\mathcal{A}}$, where $C_{\mathcal{A}}$ is a clause to further exclude the current solution.
            if $L(\mathcal{A}') \subset L(\mathcal{A})$ **then**
                Let w be a shortest string in $L(\mathcal{A}) \setminus L(\mathcal{A}')$.
                $C \leftarrow C \wedge C_w$, where C_w is clauses encoding the requirement that w must not be in the conjecture.
                $S_- \leftarrow S_- \cup \{w\}$
                $\mathcal{A} \leftarrow \mathcal{A}'$
            end
        else
            Let w be a shortest string in $L(\mathcal{A}') \setminus L(\mathcal{A})$.
            $C \leftarrow C \wedge C_w$, where C_w is clauses encoding the requirement that w must not be in the conjecture.
            $S_- \leftarrow S_- \cup \{w\}$
        end
    end
end
return $min(\mathcal{A}), S_-$

Definition (characteristic sample)

We say that $S = (S_+, S_-)$ is a *characteristic sample* for a minimal DFA $\mathcal{A}$ if $\mathcal{A}$ is consistent with S and if for each $\mathcal{A}'$ consistent with S such that $|\mathcal{A}'| \leq |\mathcal{A}|$ we have that $\mathcal{A}'$ is isomorphic to $\mathcal{A}$

Theorem

The algorithm return a simplest n-conjecture $\mathcal{A}$ for a given S_+ and (S_+, S_-) is a characteristic sample for $\mathcal{A}$

Overview

Warning: Many simplest *n*-conjectures may exist

Remark: Knowing that there is only one solution will guarantee the best quality of the inferred model

Question: Can we check the uniqueness of the solution?

Algorithm

Input: An n-conjecture $\mathcal{A}$ and a characteristic sample (S_+, S_-) for $\mathcal{A}$
Output: Return *True* if $\mathcal{A}$ is the only simplest n-conjecture for S_+ and return a distinguishing string otherwise

Function $\texttt{CheckUniqueness}\,(\mathcal{A}, (S_+, S_-))$**:**

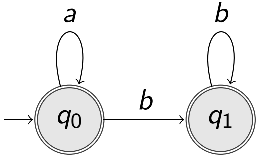
```
foreach w ∈ S₋ do
    (A′, S′₋) ← infer(S₊ ∪ {w}, |A|)
    if L(A) ⊄ L(A′) then return w ;
end
return True
```

Theorem

If there exists a single simplest n-conjecture for S_+ this algorithm determines its uniqueness, otherwise it returns a distinguishing string

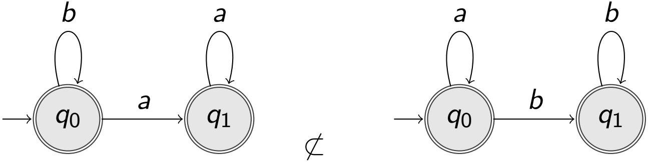
Illustration

If we consider $S_+ = \{a, aa, aaa, b, bb, bbb\} \cup \{ab\}$, the algorithm *infer* will find a second solution:



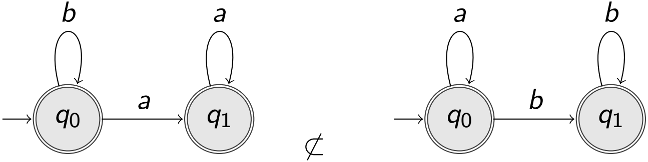
Illustration

If we consider $S_+ = \{a, aa, aaa, b, bb, bbb\} \cup \{ab\}$, the algorithm *infer* will find a second solution:



Illustration

If we consider $S_+ = \{a, aa, aaa, b, bb, bbb\} \cup \{ab\}$, the algorithm *infer* will find a second solution:



Solution is not unique and *ab* is a distinguishing string

Overview

Definition (Characteristic positive examples)

Positive examples S_+ are *characteristic positive examples* for $\mathcal{A}$ if the simplest $|\mathcal{A}|$-conjecture for S_+ is $\mathcal{A}$ and it is unique

Question: Can we find a characteristic positive examples for each DFA?

Algorithm

Input: A DFA $\mathcal{A}$
Output: Characteristic positive examples for $\mathcal{A}$
Function *GenerateCharacteristicPositiveExamples* $(\mathcal{A})$**:**

```
S+ ← ∅
while S+ is not a characteristic positive examples for A do
    Let A' be a simplest |A|-conjecture for S+ for which there exists
      w ∈ L(A) such that w ∉ L(A').
    S+ ← S+ ∪ {w}
end
return S+
```

Theorem

For each DFA $\mathcal{A}$, the algorithm GenerateCharacteristicPositiveExamples returns characteristic positive examples

Theorem

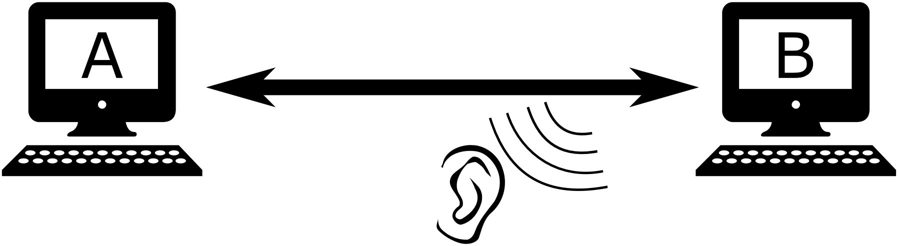
If S_+ is characteristic positive examples for $\mathcal{A}$, then each S'_+ such that $S_+ \subseteq S'_+ \subseteq L(\mathcal{A})$ is also characteristic positive examples for $\mathcal{A}$

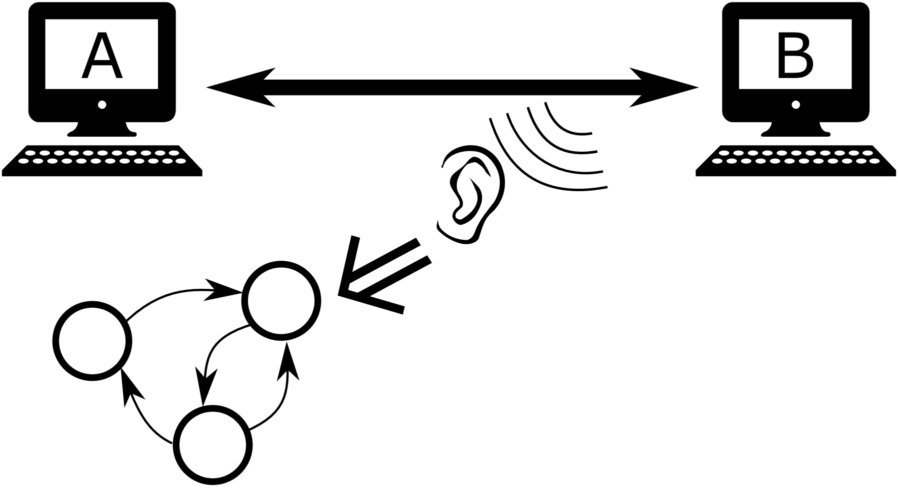
Corollary

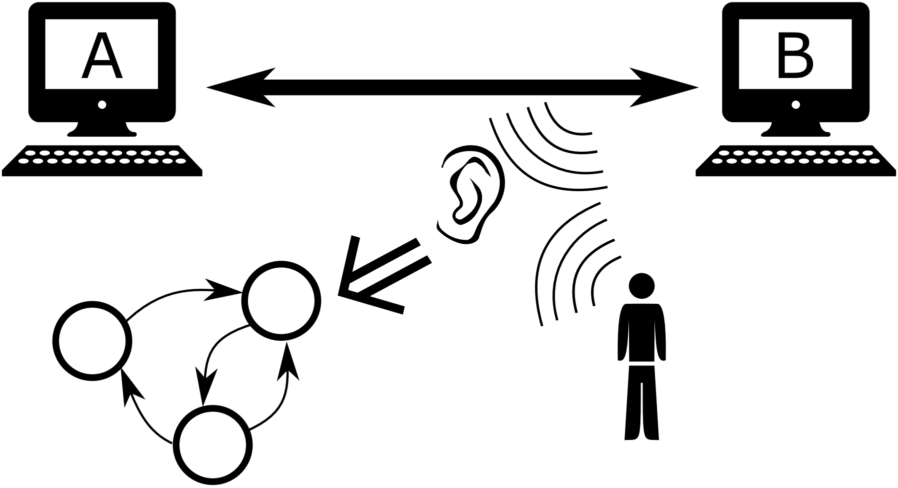
The languages generated by DFAs with n states are identifiable in the limit from positive examples by searching the simplest n-conjectures

Overview

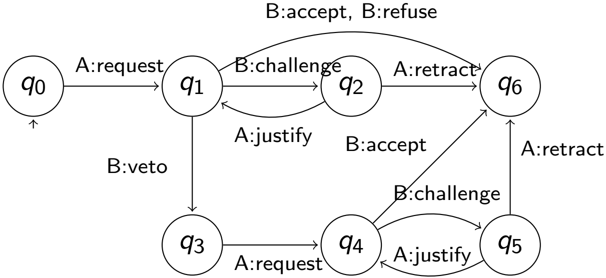

A
B


A
B


A
B

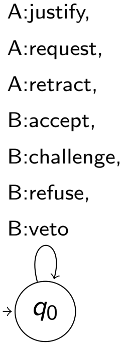

A
B

Communication protocols used:

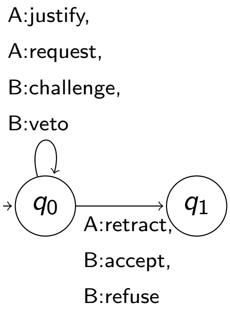


Input: 50 traces from this protocol generated with a random walk

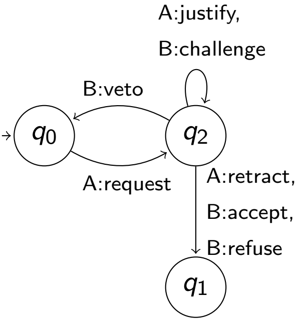
$n = 1$


A:justify,
A:request,
A:retract,
B:accept,
B:challenge,
B:refuse,
B:veto
q_0

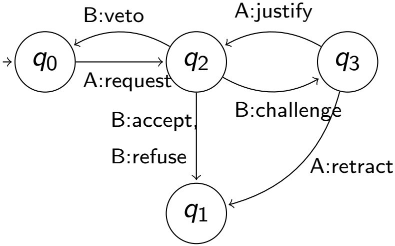
$n = 2$


A:justify,
A:request,
B:challenge,
B:veto
q_0
q_1
A:retract,
B:accept,
B:refuse

$n = 3$

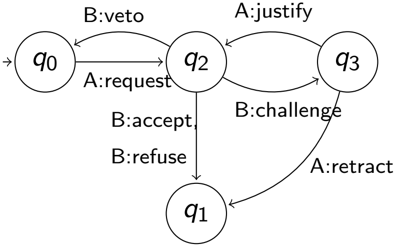

A:justify,
B:challenge
B:veto
q_0
q_2
A:request
A:retract,
B:accept,
B:refuse
q_1

$n = 4$



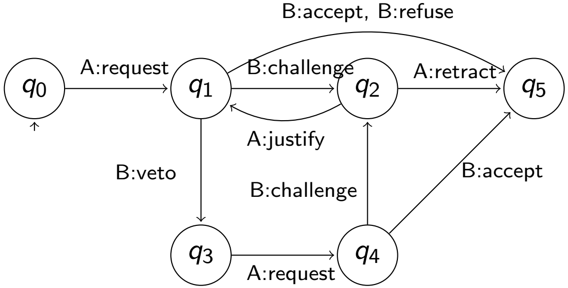
B:veto
A:justify
q_0
q_2
q_3
A:request
B:challenge
B:accept,
B:refuse
A:retract
q_1

$n = 5$

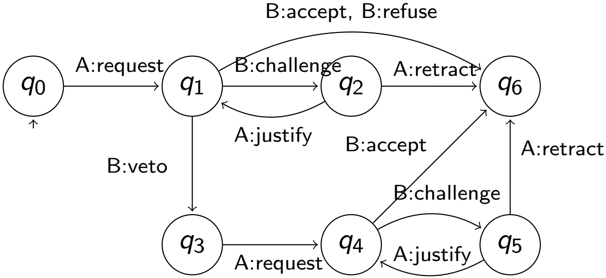


B:veto
A:justify
q_0
q_2
q_3
A:request
B:challenge
B:accept,
B:refuse
A:retract
q_1

$n = 6$


B:accept, B:refuse
A:request
B:challenge
A:retract
q_0
q_1
q_2
q_5
A:justify
B:veto
B:challenge
B:accept
q_3
q_4
A:request

$n = 7$



This solution is unique

Overview

Conclusion

- New approach to solving DFA inference problem without negative example
- Applicable in practice for small models
- Results that make sense

Perspectives

- Improving SAT formulas
- Search for heuristics
- Apply this approach to more specific models
- Link to probabilistic approaches?

Thank you