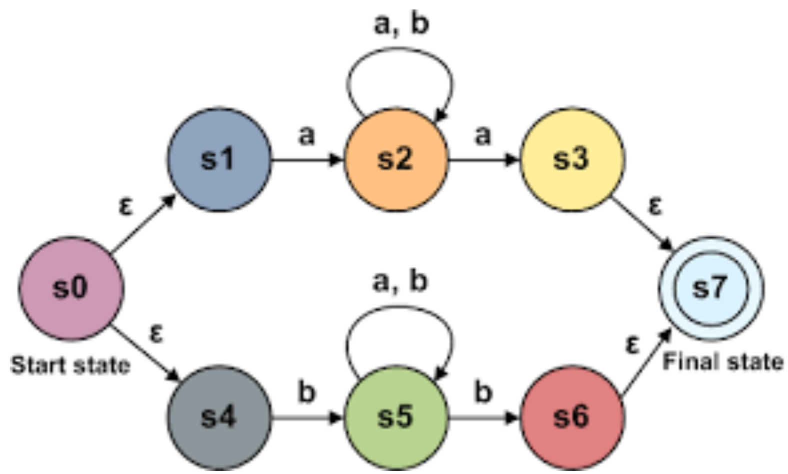
An Introduction to Automata Learning



Yu-Fang Chen
Department of Information Management
Academia Sinica

Before we start

- Have you played the game of number guessing?

- Let’s guess finite automata!

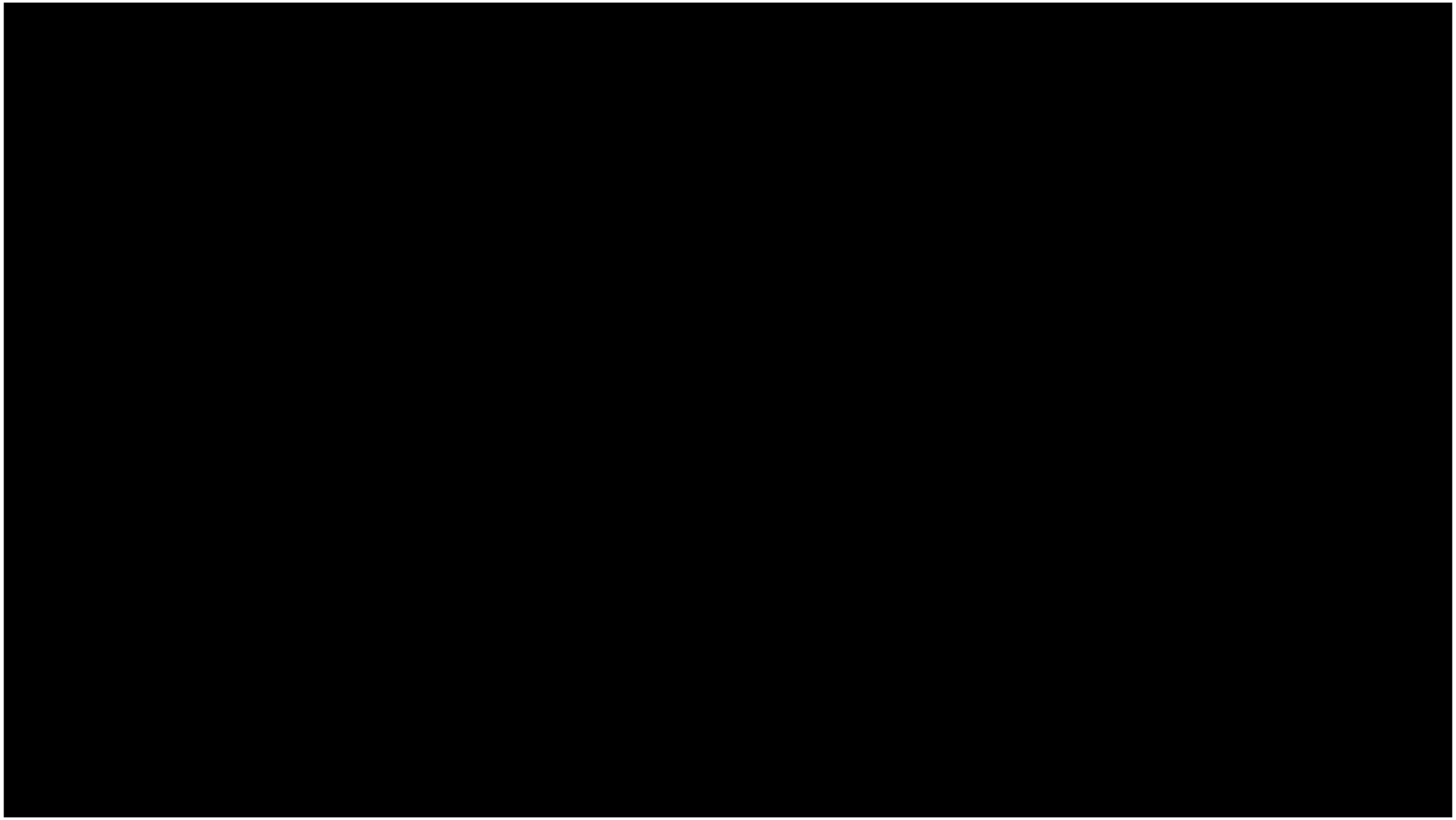
Models

- Allow predictions and analysis of the behaviors of complex systems.

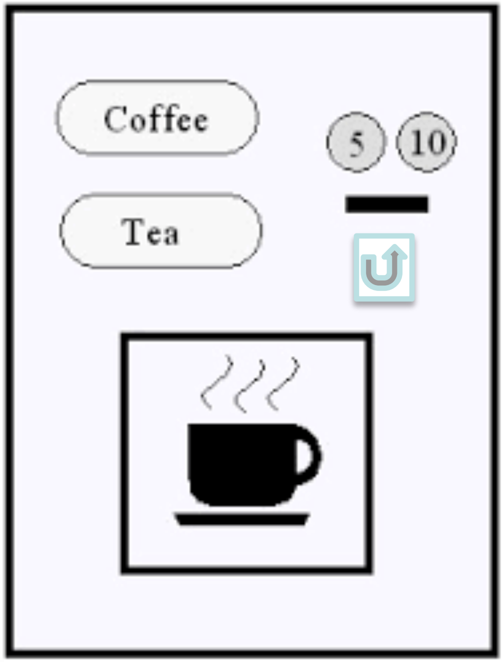
- Forms of models:
  - Formula: e.g., $E = MC^2$

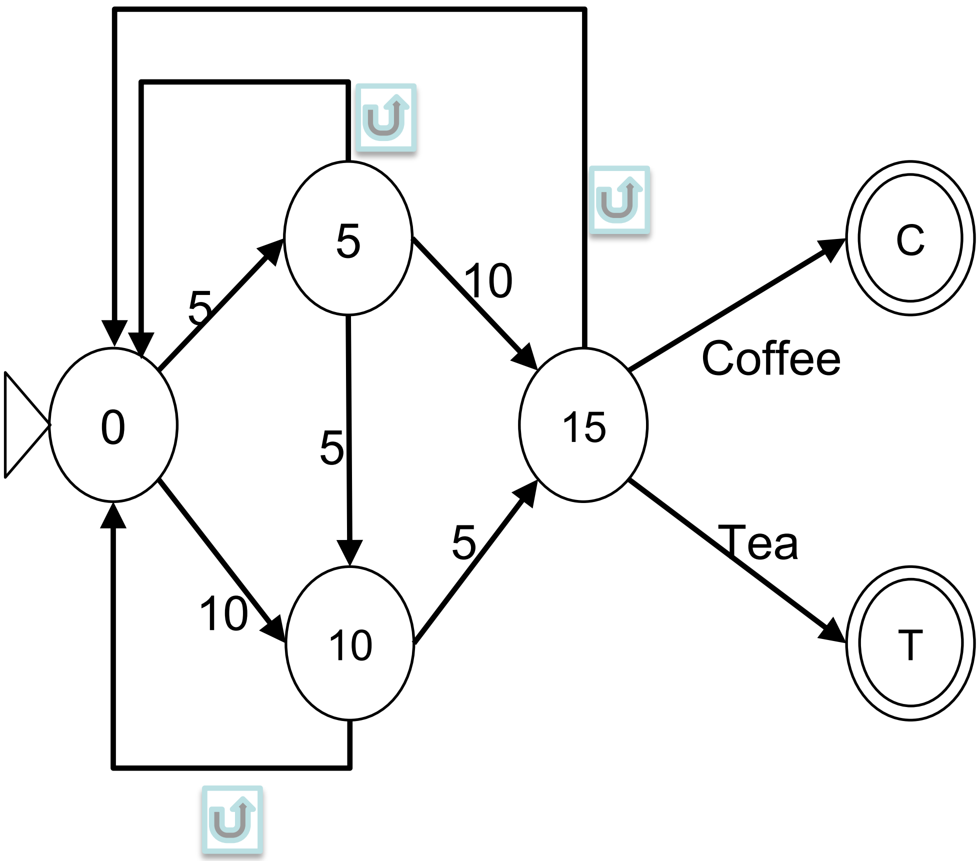
  - Finite state machine/automata.

A Video about Model Learning



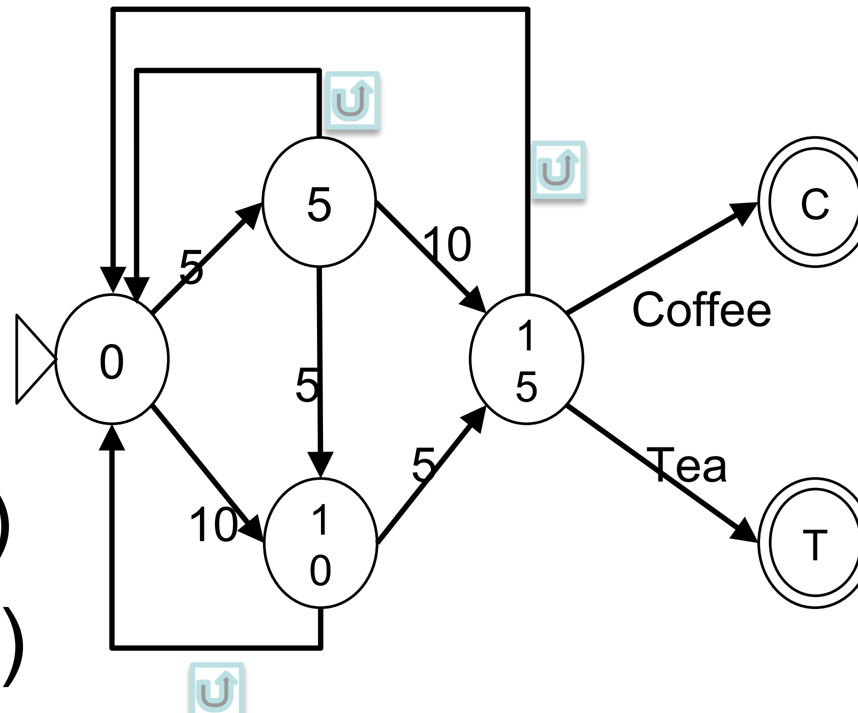
Example





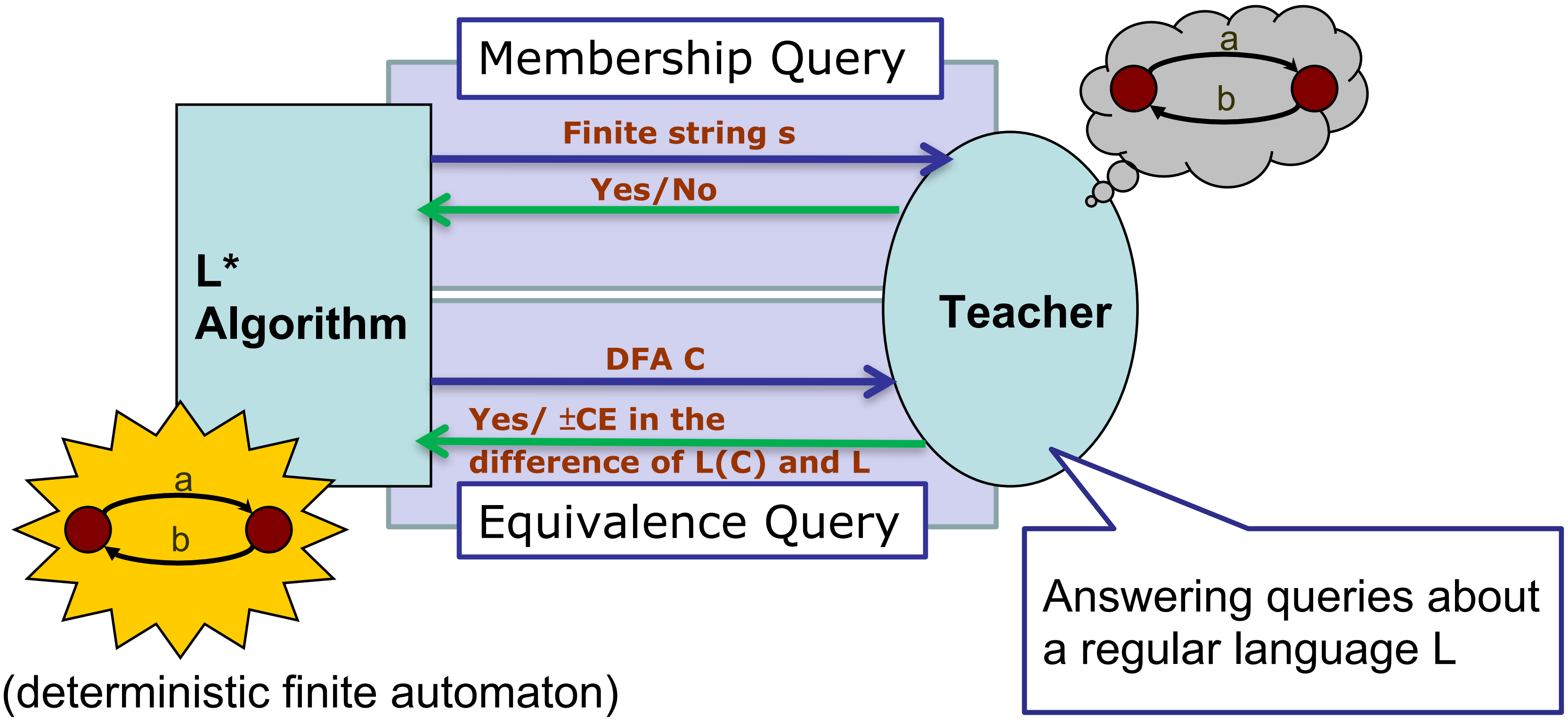
How to Obtain the Model?

- Read the manual
- Experiments
  - 10 5 Coffee (good)
  - 10 10 Coffee (bad)
  - 5 5 5 Coffee (good)
  - 5 5 10 Tea (good)



The L* Learning Algorithm

- Proposed by Dana Angluin [Info.&Comp. 1987] and improved by Rivest *et al.* [Info.&Comp. 1993]



DFA (deterministic finite automaton)

Myhill-Nerode Theorm

- Given a language L, we define a equivalence relation as follows.

$$x \equiv_L y \text{ iff } \forall z \in \Sigma^* : xz \in L \leftrightarrow xy \in L.$$

- L is regular iff $\equiv_L$ form a finite number of equivalence classes

- Each equivalence class corresponds to a state of the minimal DFA of L

- In fact, a finite subset of $\sum^*$s sufficient to distinguish all the states

Observation Table

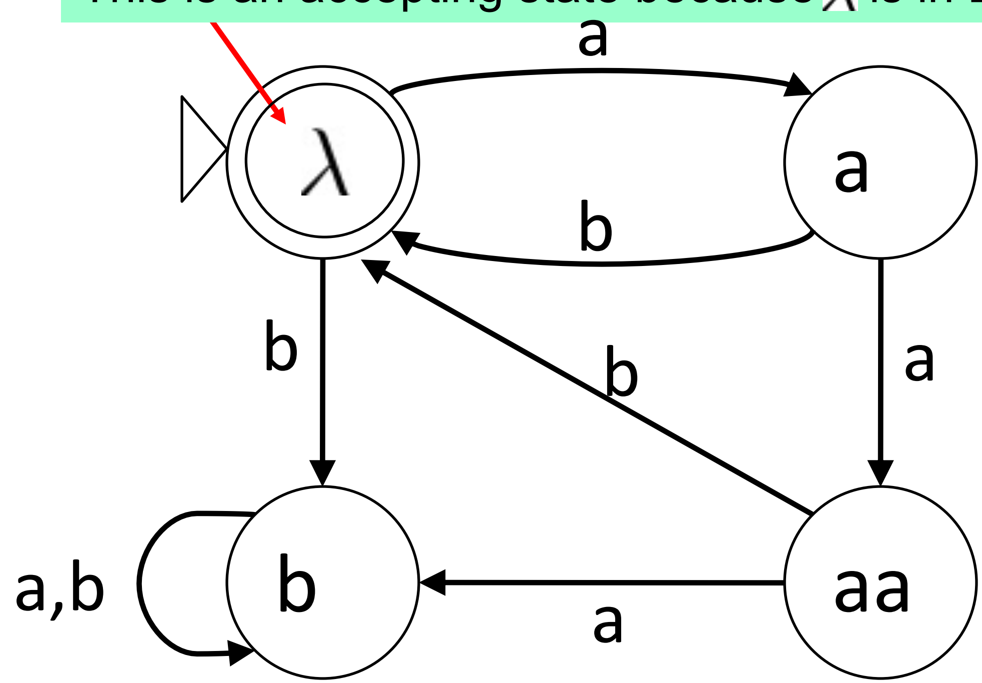
E: Distinguishing Experiments

	λ	b	ab
λ	T	F	T
a	F	T	T
b	F	F	F
aa	F	T	F
ab	T	F	T
ba	F	F	F
bb	F	F	F
aaa	F	F	F
aab	T	F	T

S States: λ, a, b, aa

SΣ Next States: ab, ba, bb, aaa, aab

This is an accepting state because λ is in L.



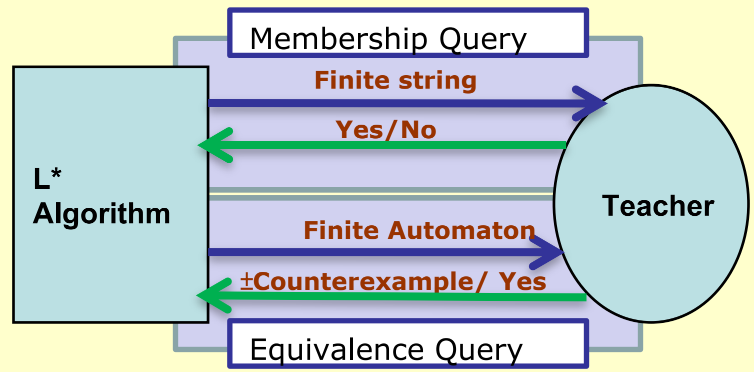
$\delta(s, a) = s'$ iff sa and s' have the same row.

Target: (ab+aab)*

L*: Initial Setting

	λ
λ	
a	
b	

A reminder:



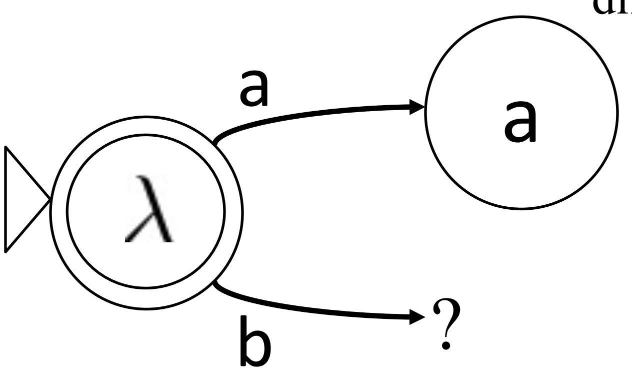
Target: (ab+aab)*

L*: Fill Up the Table by Membership Queries

	λ
λ	T
a	F
b	F

Fill up the table using **membership queries**

a represents a new equivalence class, because its **row** is different from all of those in the current S set.



Target: (ab+aab)*

L*: Table Expansion

Move a to the S set and expand the table with elements aa and ab

	λ
λ	T
a	F
b	F
aa	
ab	

Target: (ab+aab)*

L*: A Closed Table

	λ
λ	T
a	F
b	F
aa	F
ab	T

Again, fill up the table using membership queries

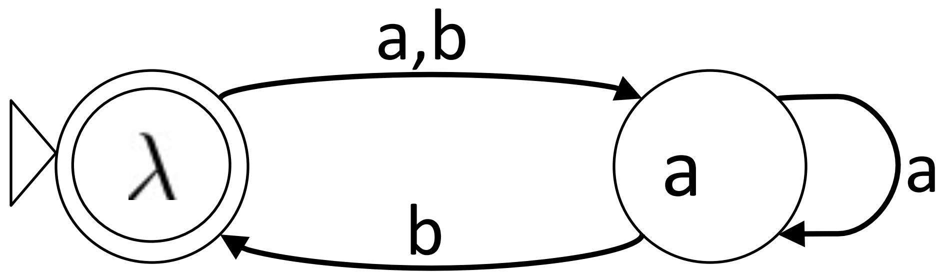
We say that the table is **closed** because every row in the SΣ set appears somewhere in the S set

Target: (ab+aab)*

L*: Making a Conjecture

	λ
λ	T
a	F
b	F
aa	F
ab	T

Construct a DFA from the learned equivalence classes



Counterexample: bb

$\delta(s, a) = s'$ iff sa and s' have the same row.

A suffix b is extracted from bb as a valid distinguishing experiment

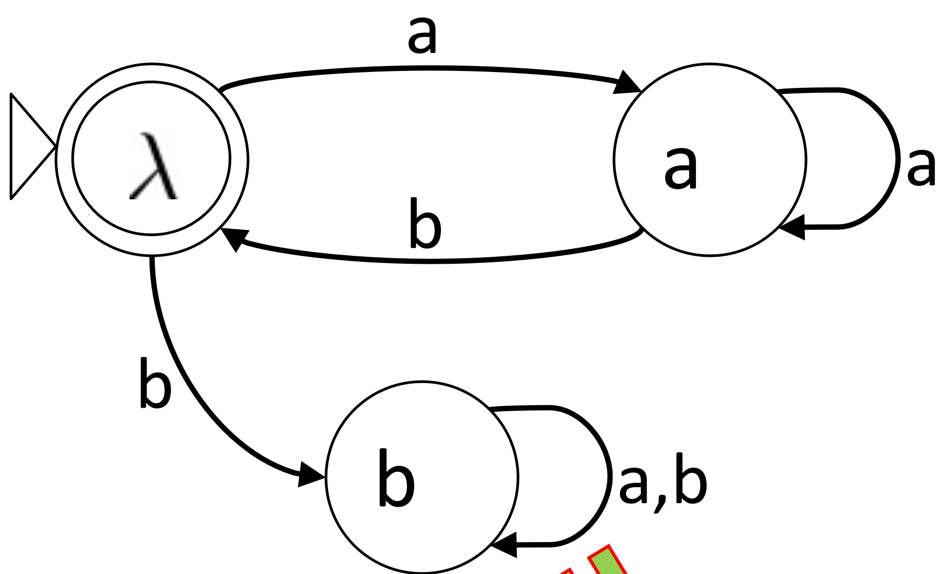
Target: (ab+aab)*

Theorem:
At least one suffix of the counterexample is a valid distinguishing experiment

L*: 2nd Iteration

Add b to the E set, fill up and expand the table following the same procedure

	λ	b
λ	T	F
a	F	T
b	F	F
aa	F	T
ab	T	F
ba	F	F
bb	F	F



Counterexample: aaab

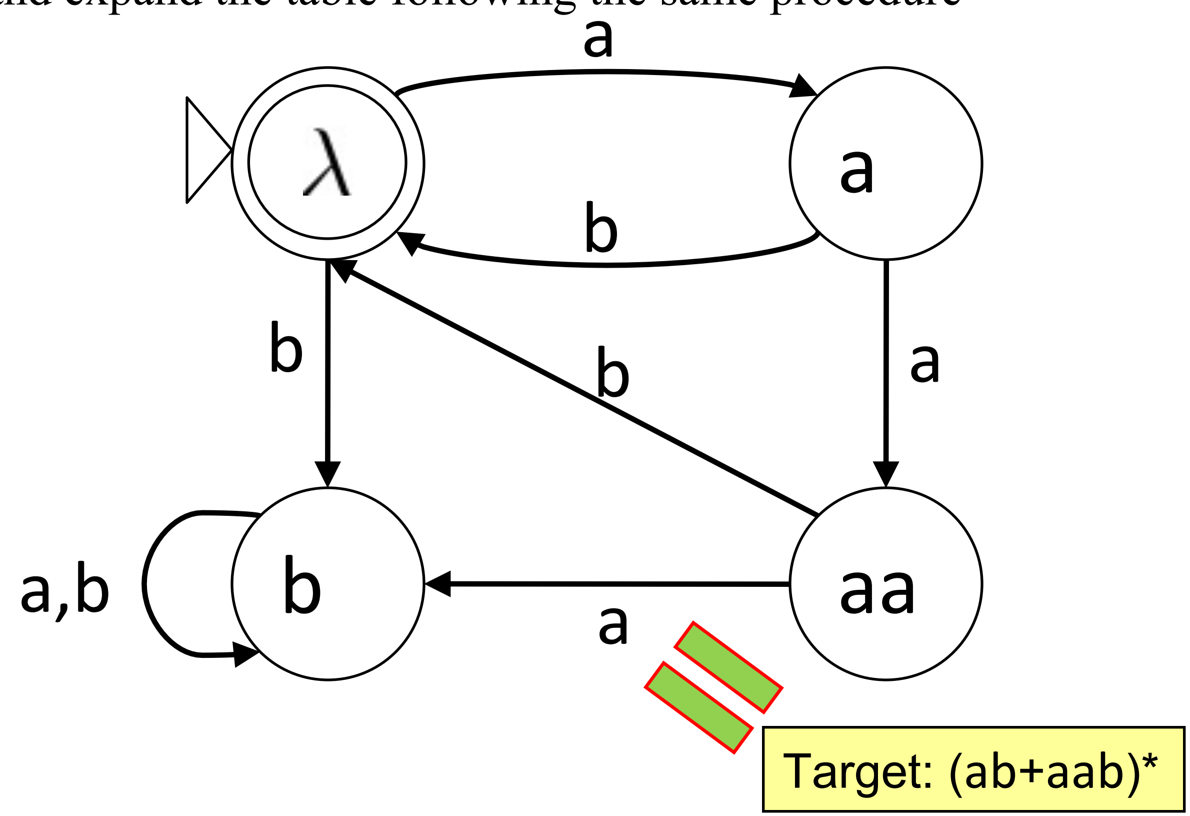
A suffix ab is extracted from aaab as a valid distinguishing experiment

Target: (ab+aab)*

L*: 3rd Iteration (Completed)

Add ab to the E set, fill up and expand the table following the same procedure

	λ	b	ab
λ	T	F	T
a	F	T	T
b	F	F	F
aa	F	T	F
ab	T	F	T
ba	F	F	F
bb	F	F	F
aaa	F	F	F
aab	T	F	T



Theorem:
The DFA produced by L* is the minimal DFA that recognizes that target language

L*: Complexity

- **Complexity:**
  - Equivalence query: at most n-1
  - Membership query: $O(|\Sigma|n^2 + n \log m)$

	λ	b	ab
λ	T	F	T
a	F	T	T
b	F	F	F
aa	F	T	F
ab	T	F	T
ba	F	F	F
bb	F	F	F
aaa	F	F	F
aab	T	F	T

Note: ***n*** is the size of the minimal DFA that recognizes L, ***m*** is the length of the longest counterexample returned from the teacher.

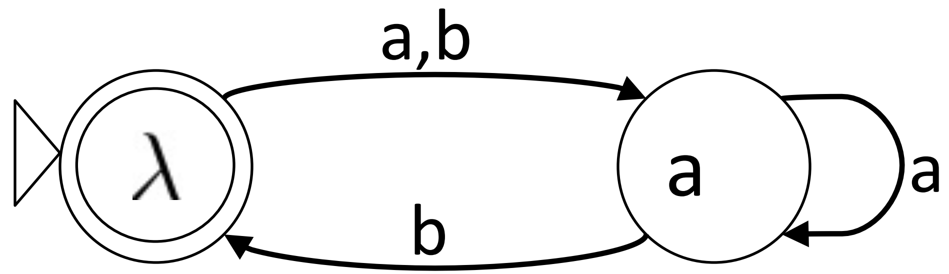
Exercise

- Let's play with it.
  - Any volunteer?

Counterexample Analysis

	λ
λ	T
a	F
b	F
aa	F
ab	T

Construct a DFA from the learned equivalence classes



Counterexample: bb

Target: (ab+aab)*

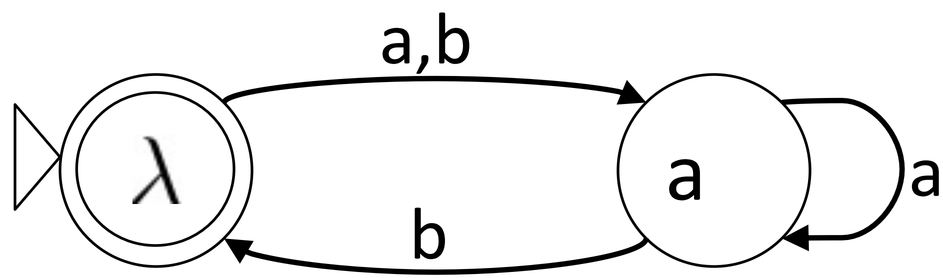
A suffix b is extracted from bb as a valid distinguishing experiment

Theorem:
At least one suffix of the counterexample is a valid distinguishing experiment

Counterexample Analysis

	λ
λ	T
a	F
b	F
aa	F
ab	T

Construct a DFA from the learned equivalence classes



Counterexample: bb

A suffix b is extracted from bb as a valid distinguishing experiment

Target: (ab+aab)*

[bb]= empty word	is in the target language
bb	is not in the target language

Exercise

- Let's play with it.
  - 2 people as a group.

A different version

- Using classification tree instead of observation table

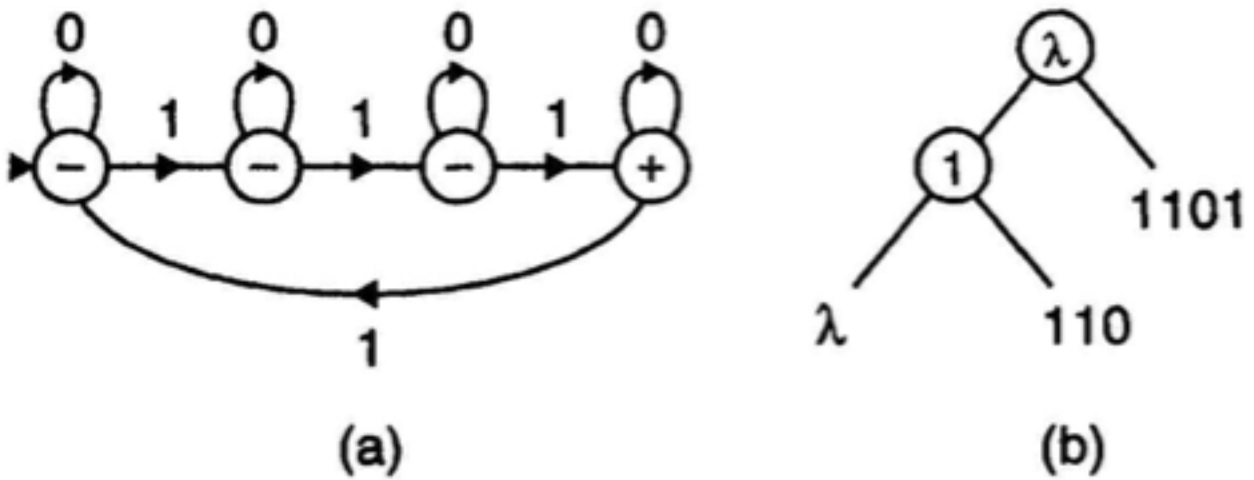
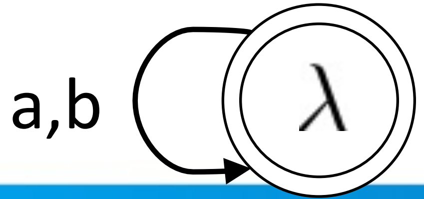


Figure 8.1: (a) *Finite automaton counting the number of* 1*'s in the input* 3 *mod* 4. (b) *A classification tree for this automaton.*

A different version

Initialization:

- Do a membership query on the string λ to determine whether the start state of M is accepting or rejecting.
- Construct a hypothesis automaton that consists simply of this single (accepting or rejecting) state with self-loops for both the 0 and 1 transitions.
- Perform an equivalence query on this automaton; let the counterexample string be γ.
- Initialize the classification tree T to have a root labeled with the distinguishing string λ and two leaves labeled with access strings λ and γ.



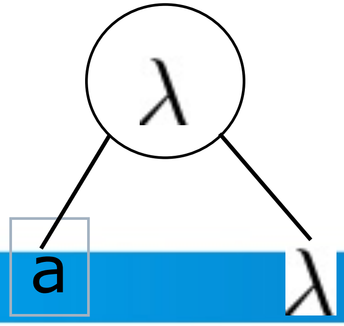
CE=a

Target: (ab+aab)*

A different version

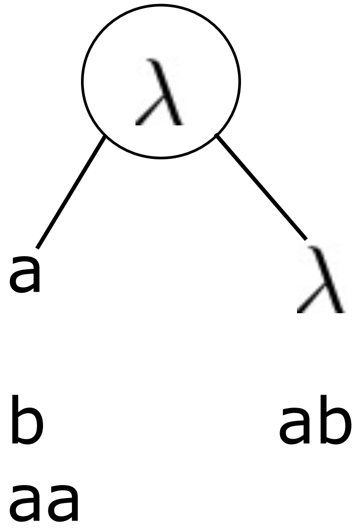
Initialization:

- Do a membership query on the string λ to determine whether the start state of M is accepting or rejecting.
- Construct a hypothesis automaton that consists simply of this single (accepting or rejecting) state with self-loops for both the 0 and 1 transitions.
- Perform an equivalence query on this automaton; let the counterexample string be γ.
- Initialize the classification tree T to have a root labeled with the distinguishing string λ and two leaves labeled with access strings λ and γ.

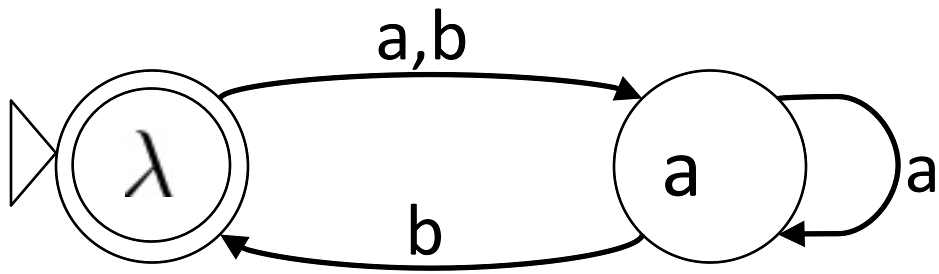


Target: (ab+aab)*

A different version



Construct a DFA from the learned equivalence classes



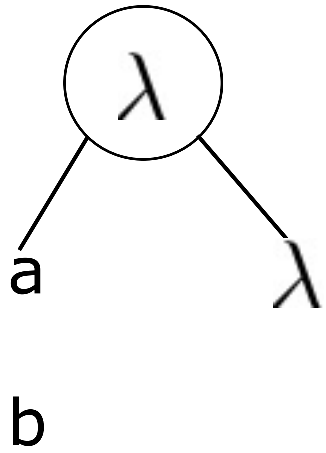
Counterexample: bb

A suffix b is extracted from bb as a valid distinguishing experiment

Target: (ab+aab)*

Theorem:
At least one suffix of the counterexample is a valid distinguishing experiment

A different version

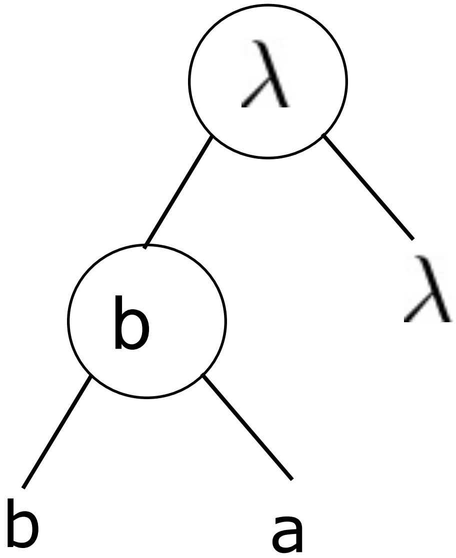


Counterexample: bb

A suffix b is extracted from bb as a valid distinguishing experiment

Target: (ab+aab)*

A different version

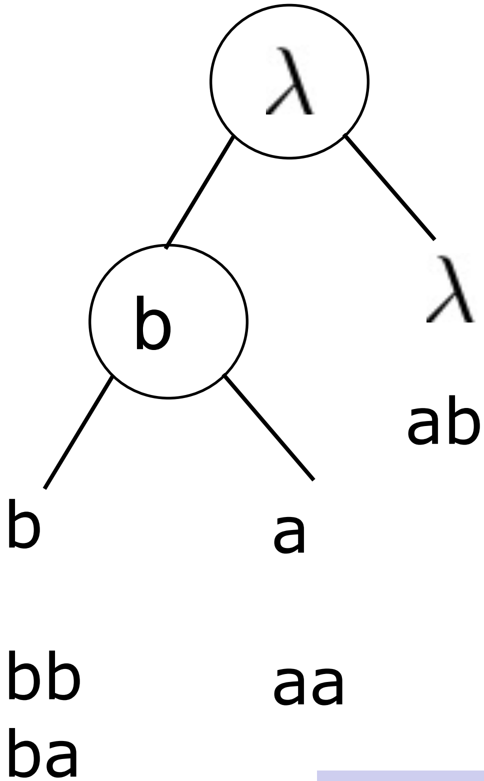


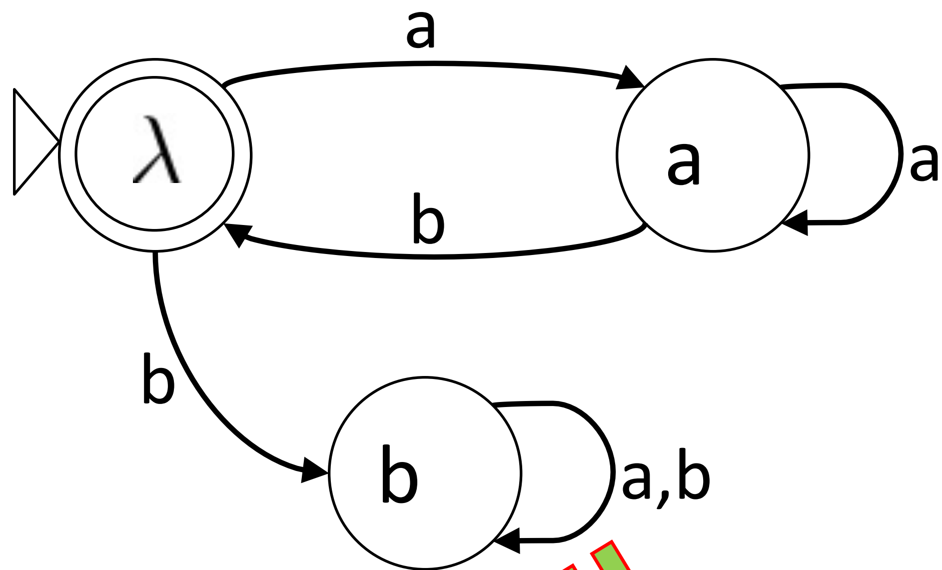
Counterexample: bb

A suffix b is extracted from bb as a valid distinguishing experiment

Target: (ab+aab)*

A different version



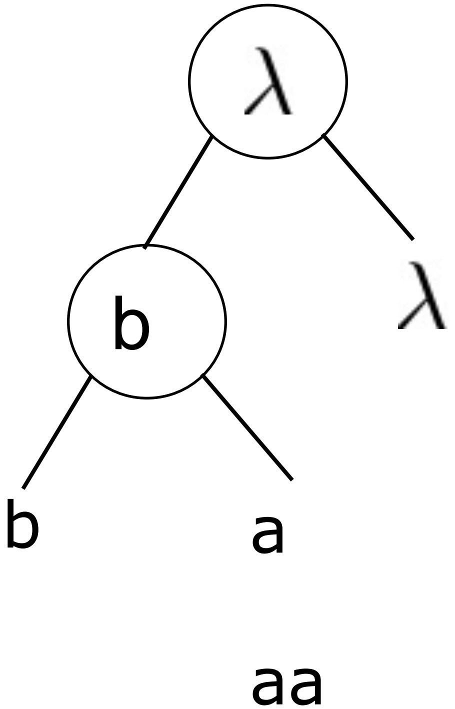


Counterexample: aaab

A suffix ab is extracted from aaab as a valid distinguishing experiment

Target: (ab+aab)*

A different version

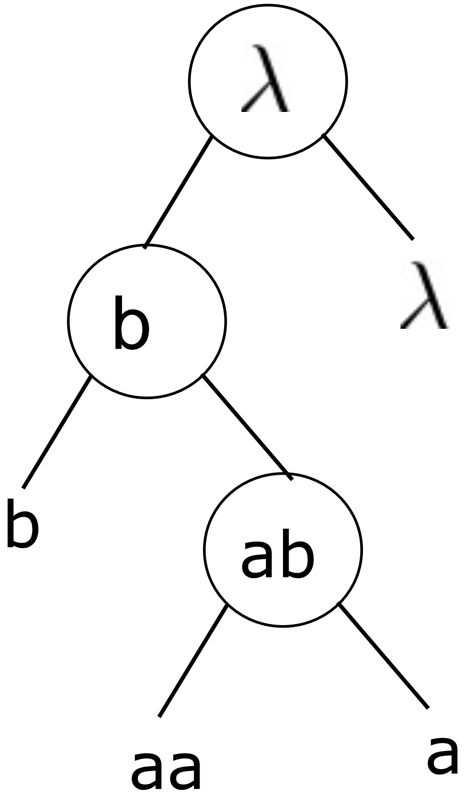


Counterexample: aaab

A suffix ab is extracted from aaab as a valid distinguishing experiment

Target: (ab+aab)*

A different version

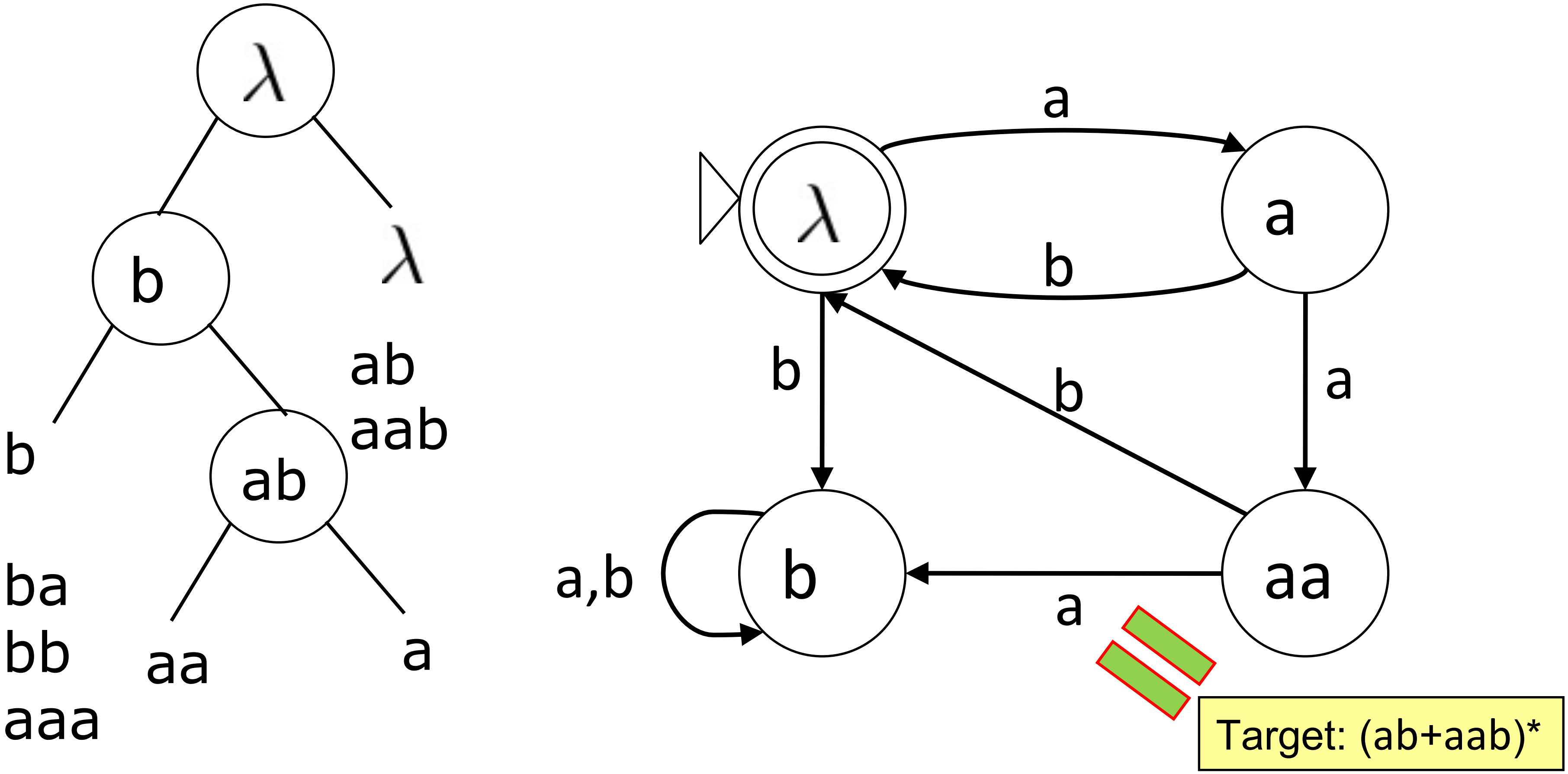


Counterexample: aaab

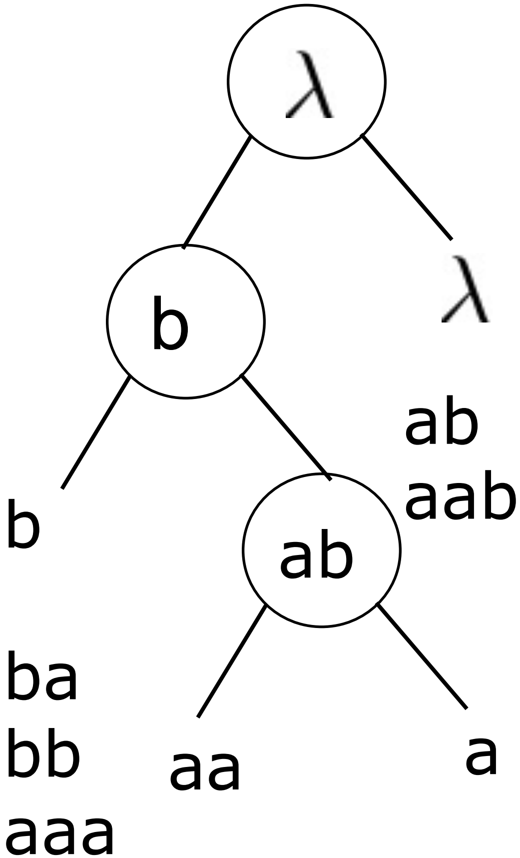
Target: (ab+aab)*

A suffix ab is extracted from aaab as a valid distinguishing experiment

A different version



Compare the two algorithms



	λ	b	ab
λ	T	F	T
a	F	T	T
b	F	F	F
aa	F	T	F
ab	T	F	T
ba	F	F	F
bb	F	F	F
aaa	F	F	F
aab	T	F	T

More Equivalence Queries

More Membership Queries

Implementations

https://learnlib.de

http://libalf.informatik.rwth-aachen.de

Applications

- Regression testing of telecommunication systems at **Siemens**
- Integration testing at **France Telecom**
- Automatic testing of an online conference service of **Springer Verlag**
- Testing requirements of a brake-by-wire system from **Volvo Technology**

Source: Frits Vaandrager, CACM, Vol. 60 No. 2, Pages 86-95

Applications: learning from gray box

- Program verification/testing:
  - Models the the sequences of events of the program under test

- Examples:
  - Decision sequence e.g., TFTTTF
  - Call sequence e.g., foo() bar() bar()
  - Label sequence

Equivalence Queries?

- PAC learning:
  - Sample according to a distribution of historical user behavior.
  - Replace equivalence query with membership queries.
- Conformance testing