Learning Regular Sets from Queries and Counterexamples

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

IISC, 28 October 2015.

Outline

Dana Angluin

Learning Regular Sets from Queries and Counterexamples, in *Information & Computation*, 1987.

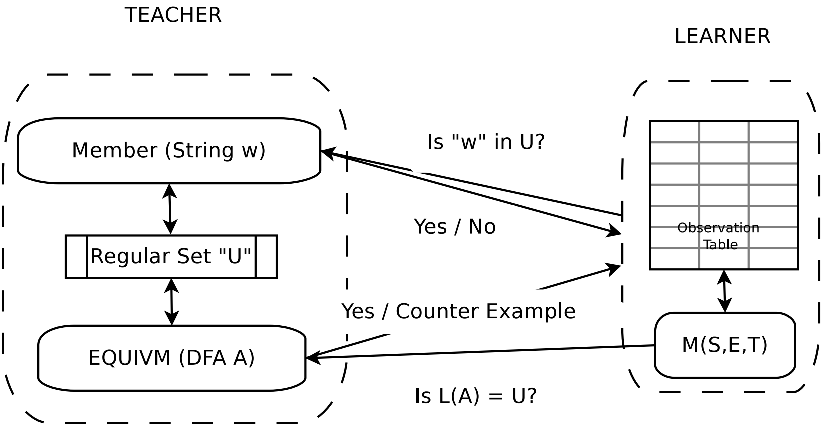
Angluin's L^* algorithm

Teacher has a regular language U in mind.
The Learner can ask two types of queries:

- Is a given string w in U? Teacher answers "Yes" or "No".
- Does a given DFA $\mathcal{A}$ accept the language U? Teacher answers "Yes" or gives a counterexample x.

Angluin's algorithm for the Learner finds the canonical DFA for U, in a number of steps polynomial in the number of states of the canonical DFA for U and the length of the longest counterexample returned by the teacher.

Working of L* Algorithm



Angluin's Algorithm by Example

Suppose the Teacher has in mind the language

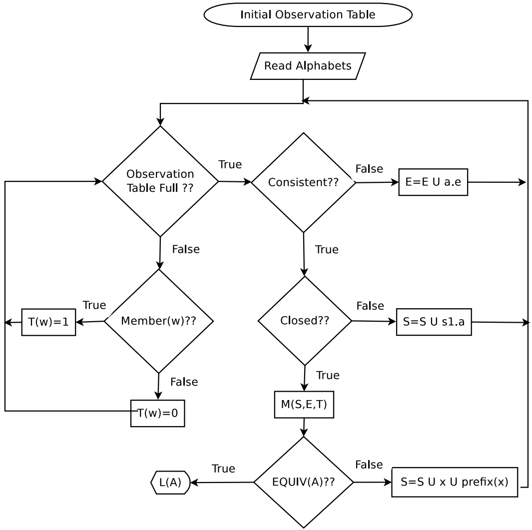
$U = \{w \in \{a, b\}^* \mid$ number of a's is even and number of b's is even$\}$

The Learner asks the Teacher if ϵ, a, and b belong to U, and obtains the following Observation Table:

		ϵ
S	ϵ	1
$S.\{a, b\}$	a	0
	b	0

The set of strings S represents the states of the automaton constructed by the Learner.
Entry (s, e) of the table represents the fact that from state s the automaton accepts/rejects the string $s \cdot e$.

Flow Chart of L* Algorithm - 0

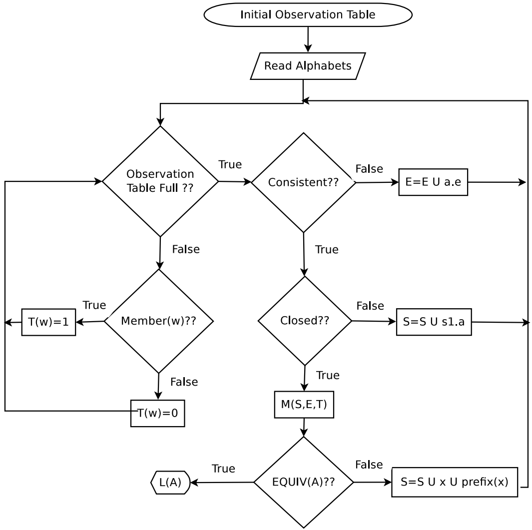


Initial OT

S

	ϵ
ϵ	?

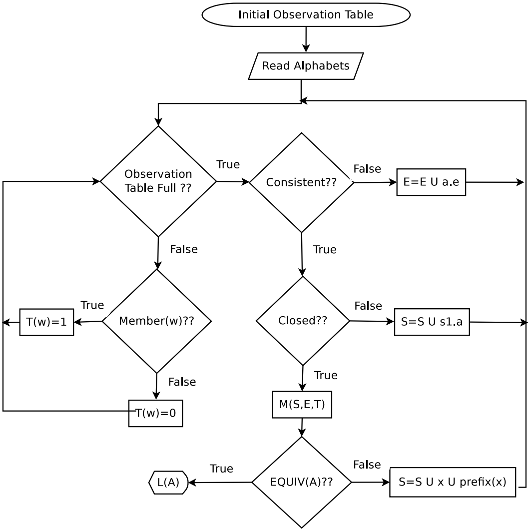
Flow Chart of L* Algorithm - 1



Observation Table

		ϵ
S	ϵ	?
$S.\{a, b\}$	a	?
	b	?

Flow Chart of L* Algorithm - 2



Observation Table

		ϵ
S	ϵ	1
$S.\{a,b\}$	a	0
	b	0

Closure

Closure of Observation Table:- An observation table is called closed provided that for each t in S.A there exist an s in S such that row(t) = row(s)

Not Closed means:-

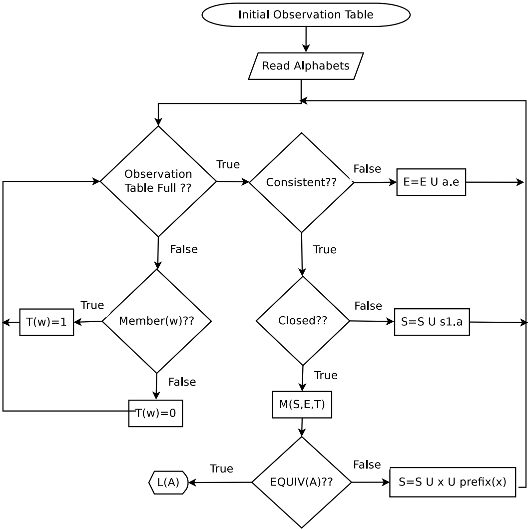
There is a row(s_1·a) in S.A is different from all s in S

To resolve this add s_1·a to S

Observation Table:

		ϵ
S	ϵ	1
$S.\{a, b\}$	a	0
	b	0

Flow Chart of L* Algorithm - 3



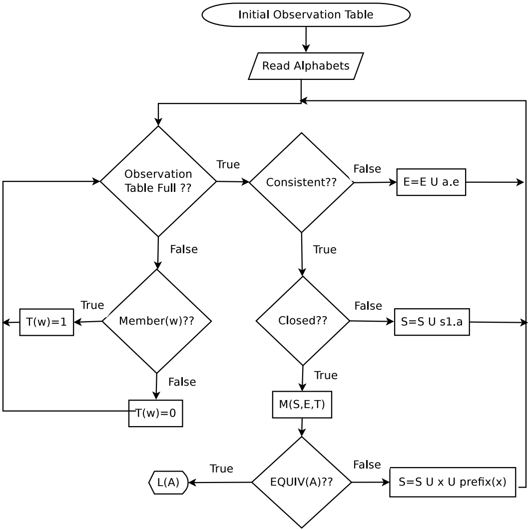
Observation Table

	ϵ
ϵ	1
a	0
b	0
aa	?
ab	?

S: rows ϵ, a

$S.\{a, b\}$: rows b, aa, ab

Flow Chart of L* Algorithm - 4



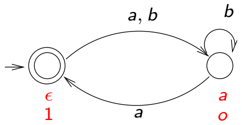
Observation Table

		ϵ
S	ϵ	1
	a	0
$S.\{a, b\}$	b	0
	aa	1
	ab	0

Constructing DFA from Observation Table: M(S,E,T)

Rules to construct Automata:

- Q = {row(s): s∈S}
- q_0 = row(ϵ)
- F = {row(s): s∈S and T(s)=1}
- δ(row(s), a) = row(s·a)



$\mathcal{A}_1$

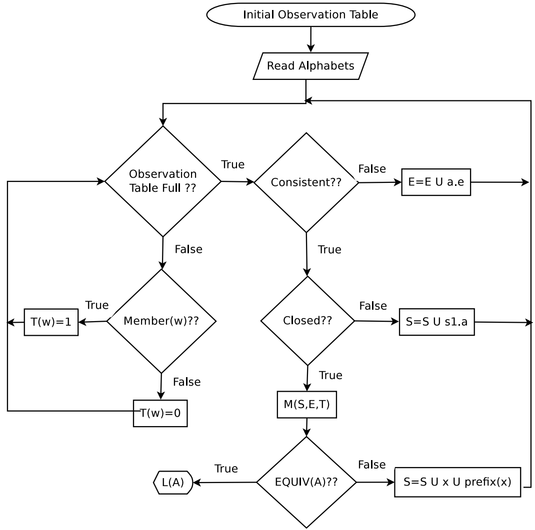
Observation Table:

		ϵ
S	ϵ	1
	a	0
$S.\{a,b\}$	b	0
	aa	1
	ab	0

- A = {Q,S,δ,Σ,F}
- F = $\{q_0\}$
- S = $\{q_0\}$
- Q = $\{q_0, q_1\}$
- Σ = { Alphabets }
- δ =

	a	b
q_0	q_1	q_1
q_1	q_0	q_1

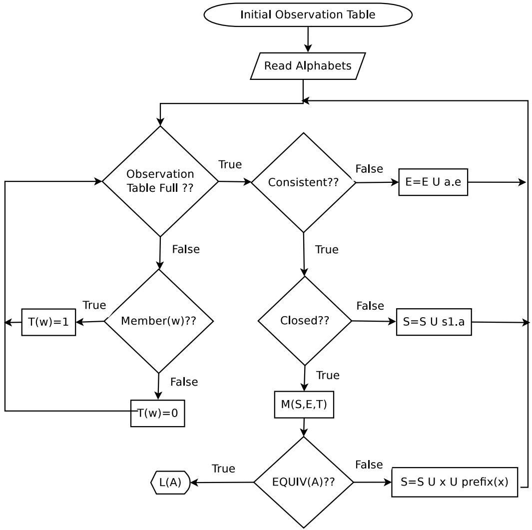
Flow Chart of L* Algorithm - 5



Observation Table

		ϵ
S	ϵ	1
	a	0
	b	0
	bb	?
$S.\{a, b\}$	aa	1
	ab	0
	ba	0
	bba	?
	bbb	?

Flow Chart of L* Algorithm - 6



Observation Table

		ϵ
	ϵ	1
S	a	0
	b	0
	bb	1
	aa	1
	ab	0
$S.\{a, b\}$	ba	0
	bba	0
	bbb	0

Consistency

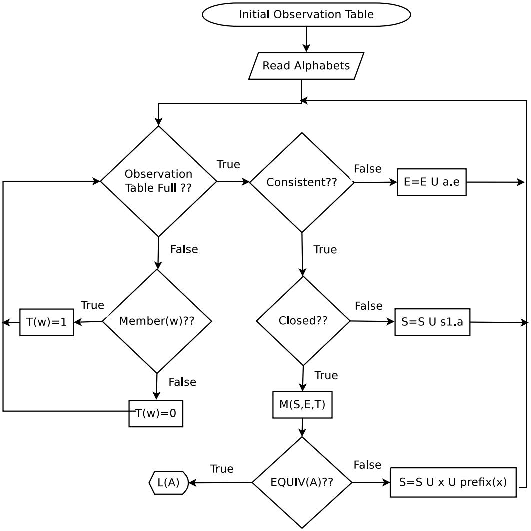
Consistency of Observation Table:-
An observation table is called consistent provided that whenever s_1 and s_2 are elements of S such that row(s_1) = row(s_2) for all a in A, row(s_1·a) = row(s_2·a)

Not Consistent means:-
There exist a certain combination of s_1 and s_2 in S, e in E, and a in A such that row(s_1) = row(s_2) but T(s_1·a·e)≠T(s_2·a·e)
To resolve this add a·e to E

Observation Table:

		ϵ
S	ϵ	1
	a	0
	b	0
	bb	1
$S.\{a,b\}$	aa	1
	ab	0
	ba	0
	bba	0
	bbb	0

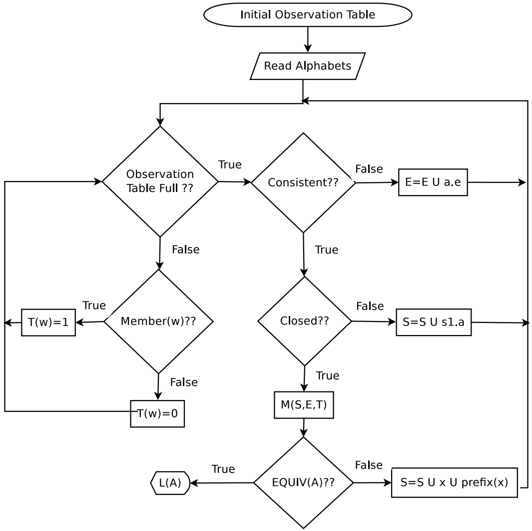
Flow Chart of L* Algorithm - 7



Observation Table

		ϵ	a
S	ϵ	1	?
	a	0	?
	b	0	?
	bb	1	?
$S.\{a,b\}$	aa	1	?
	ab	0	?
	ba	0	?
	bba	0	?
	bbb	0	?

Flow Chart of L* Algorithm - 8



Observation Table

		ϵ	a
S	ϵ	1	0
	a	0	1
	b	0	0
	bb	1	0
$S.\{a,b\}$	aa	1	0
	ab	0	0
	ba	0	0
	bba	0	1
	bbb	0	0

Angluin's Algorithm by Example

Suppose the Teacher has in mind the language

$U = \{w \in \{a, b\}^* \mid$ number of a's is even and number of b's is even$\}$

The Learner asks the Teacher if ϵ, a, and b belong to U, and obtains the following Observation Table:

		ϵ
S	ϵ	1
$S.\{a, b\}$	a	0
	b	0

The set of strings S represents the states of the automaton constructed by the Learner.
Entry (s, e) of the table represents the fact that from state s the automaton accepts/rejects the string $s \cdot e$.

Angluin's Algorithm by Example

Suppose the Teacher has in mind the language

$U = \{w \in \{a, b\}^* \mid$ number of a's is even and number of b's is even$\}$

The Learner asks the Teacher if ϵ, a, and b belong to U, and obtains the following Observation Table:

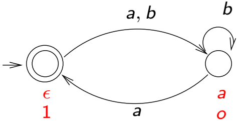
		ϵ
S	ϵ	1
$S.\{a, b\}$	a	0
	b	0

The set of strings S represents the states of the automaton constructed by the Learner.
Entry (s, e) of the table represents the fact that from state s the automaton accepts/rejects the string $s \cdot e$.
This table is not "closed" as there are no states (or "rows") corresponding to $\epsilon \cdot a$ and $\epsilon \cdot b$.

Angluin's Algorithm by Example: 2

Learner closes table by adding string a to S, and asking membership queries for aa and ab.
He now gets the observation table:

		ϵ
S	ϵ	1
	a	0
$S.\{a, b\}$	b	0
	aa	1
	ab	0



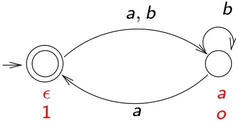
$\mathcal{A}_1$

This table is closed and consistent, and represents the DFA $\mathcal{A}_1$.

Angluin's Algorithm by Example: 2

Learner closes table by adding string *a* to *S*, and asking membership queries for *aa* and *ab*.
He now gets the observation table:

		ϵ
S	ϵ	1
	a	0
$S.\{a, b\}$	b	0
	aa	1
	ab	0



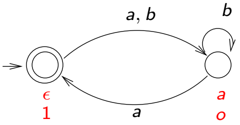
$\mathcal{A}_1$

This table is closed and consistent, and represents the DFA $\mathcal{A}_1$.
Learner now asks the Teacher if $\mathcal{A}_1$ represents the language she has in mind.

Angluin's Algorithm by Example: 2

Learner closes table by adding string a to S, and asking membership queries for aa and ab.
He now gets the observation table:

		ϵ
S	ϵ	1
	a	0
$S.\{a, b\}$	b	0
	aa	1
	ab	0



$\mathcal{A}_1$

This table is closed and consistent, and represents the DFA $\mathcal{A}_1$.
Learner now asks the Teacher if $\mathcal{A}_1$ represents the language she has in mind. Teacher replies with counterexample bb which is in U but is not accepted by $\mathcal{A}_1$.

Angluin's Algorithm by Example: 3

Learner adds *bb* and its prefixes to his set S, makes membership queries for *ba*, *bba*, and *bbb* to obtain the observation table:

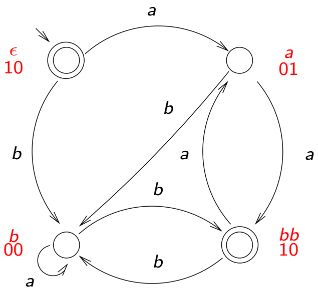
		ϵ
S	ϵ	1
	a	0
	b	0
	bb	1
$S.\{a, b\}$	aa	1
	ab	0
	ba	0
	bba	0
	bbb	0

This table is closed but not consistent. The rows for *a* and *b* are identical, but *aa* and *ba* have different rows.

Angluin's Algorithm by Example: 4

Learner adds $\epsilon \cdot a$ (that is, a) and its suffixes to the set E, and makes membership queries to obtain the observation table:

		ϵ	a
S	ϵ	1	0
	a	0	1
	b	0	0
	bb	1	0
$S.\{a, b\}$	aa	1	0
	ab	0	0
	ba	0	0
	bba	0	1
	bbb	0	0



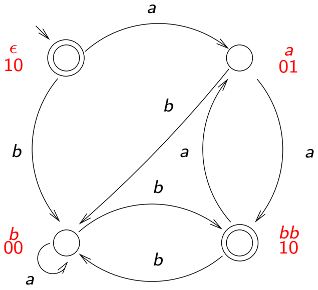
$\mathcal{A}_2$

This table is closed and consistent. So Learner conjectures the automaton $\mathcal{A}_2$.

Angluin's Algorithm by Example: 4

Learner adds $\epsilon \cdot a$ (that is, a) and its suffixes to the set E, and makes membership queries to obtain the observation table:

		ϵ	a
S	ϵ	1	0
	a	0	1
	b	0	0
	bb	1	0
$S.\{a, b\}$	aa	1	0
	ab	0	0
	ba	0	0
	bba	0	1
	bbb	0	0



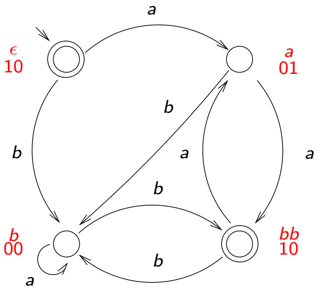
$\mathcal{A}_2$

This table is closed and consistent. So Learner conjectures the automaton $\mathcal{A}_2$. Teacher responds with counterexample abb.

Angluin's Algorithm by Example: 5

Learner adds *abb* and its prefixes to S, makes membership queries to obtain the observation table:

		ϵ	a
S	ϵ	1	0
	a	0	1
	b	0	0
	ab	0	0
	bb	1	0
	abb	0	1
$S.\{a,b\}$	aa	1	0
	ba	0	0
	aba	0	0
	bba	0	1
	bbb	0	0
	$abba$	1	0
	$abbb$	0	0



$\mathcal{A}_2$

Angluin's Algorithm by Example: 6

Learner adds b and its suffixes to E, and makes membership queries to obtain the observation table:

		ϵ	a	b
S	ϵ	1	0	0
	a	0	1	0
	b	0	0	1
	ab	0	0	0
	bb	1	0	0
	abb	0	1	0
$S.\{a,b\}$	aa	1	0	0
	ba	0	0	0
	aba	0	0	1
	bba	0	1	0
	bbb	0	0	1
	$abba$	1	0	0
	$abbb$	0	0	1

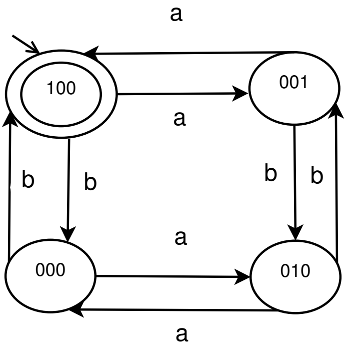


Table is closed and consistent, so Learner conjectures DFA $\mathcal{A}_3$.

Angluin's Algorithm by Example: 6

Learner adds b and its suffixes to E, and makes membership queries to obtain the observation table:

		ϵ	a	b
S	ϵ	1	0	0
	a	0	1	0
	b	0	0	1
	ab	0	0	0
	bb	1	0	0
	abb	0	1	0
$S.\{a,b\}$	aa	1	0	0
	ba	0	0	0
	aba	0	0	1
	bba	0	1	0
	bbb	0	0	1
	$abba$	1	0	0
	$abbb$	0	0	1

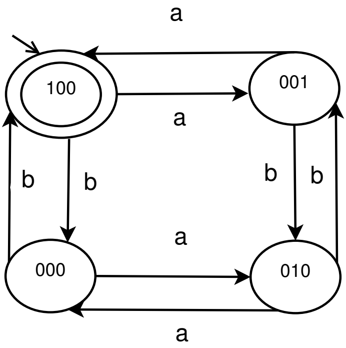


Table is closed and consistent, so Learner conjectures DFA $\mathcal{A}_3$. Teacher responds with "Yes!".

Conclusion

- Angluin's automata learning framework
- Learns the smallest automaton that satisfies given constraints.
- Does it efficiently in polynomial time in smallest automaton that satisfies the given constraints.