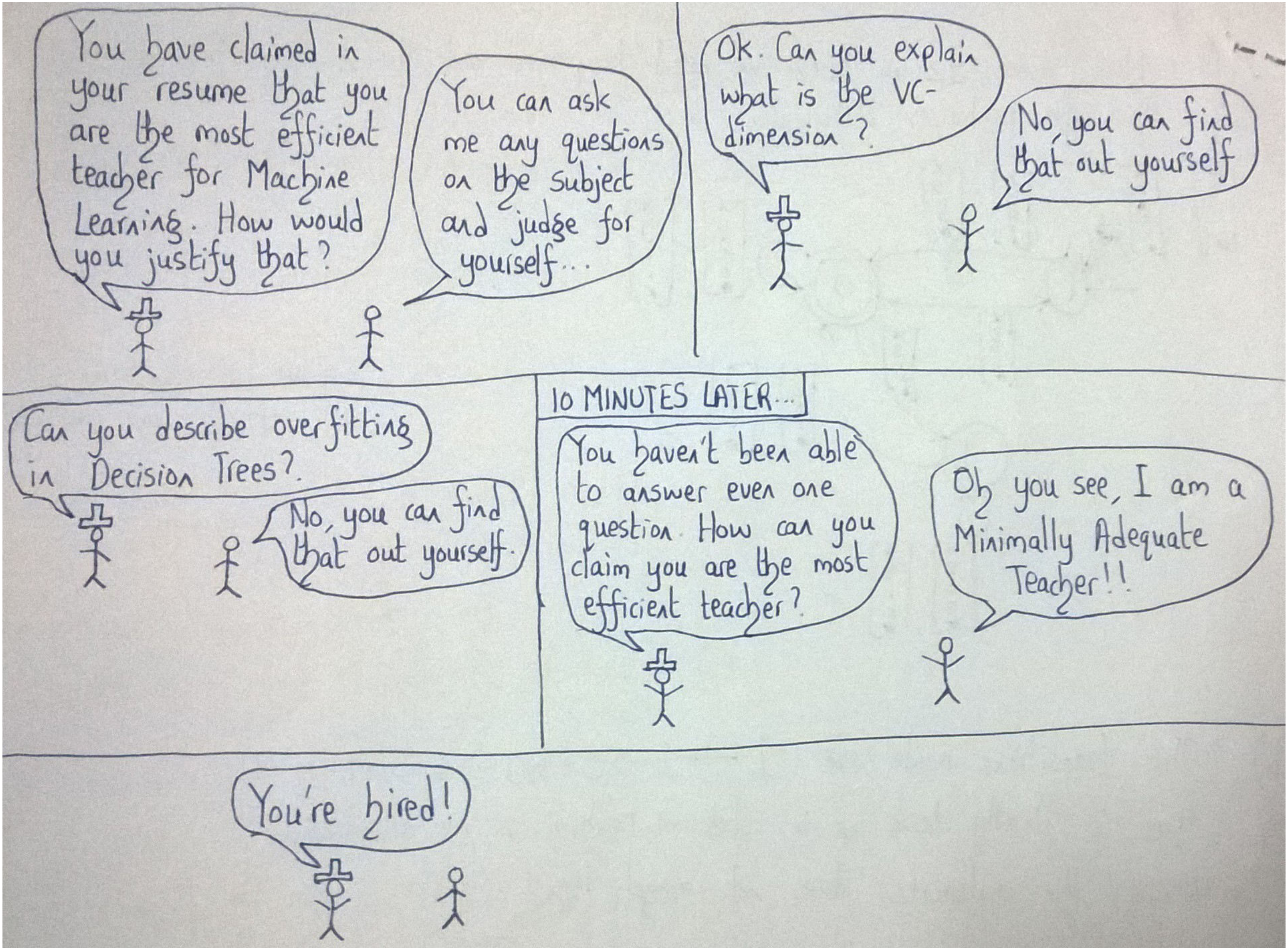
ANGLUIN'S ALGORITHM FOR LEARNING REGULAR SETS

Ullas Aparanji

Construct a DFA to accept all strings over {a,b} which have an even number of a's and an even number of b's

L* Algorithm

- Learns a DFA
- Teacher
- Oracle
- Membership queries
- Equivalence queries
- Minimally Adequate Teacher


You have claimed in your resume that you are the most efficient teacher for Machine Learning. How would you justify that?
You can ask me any questions on the subject and judge for yourself...
Ok. Can you explain what is the VC-dimension?
No, you can find that out yourself
Can you describe overfitting in Decision Trees?
No, you can find that out yourself.
10 MINUTES LATER...
You haven't been able to answer even one question. How can you claim you are the most efficient teacher?
Oh you see, I am a Minimally Adequate Teacher!!
You're hired!

NOTATIONS

- U: Unknown language to be learnt
- A: Alphabet

OBSERVATION TABLE

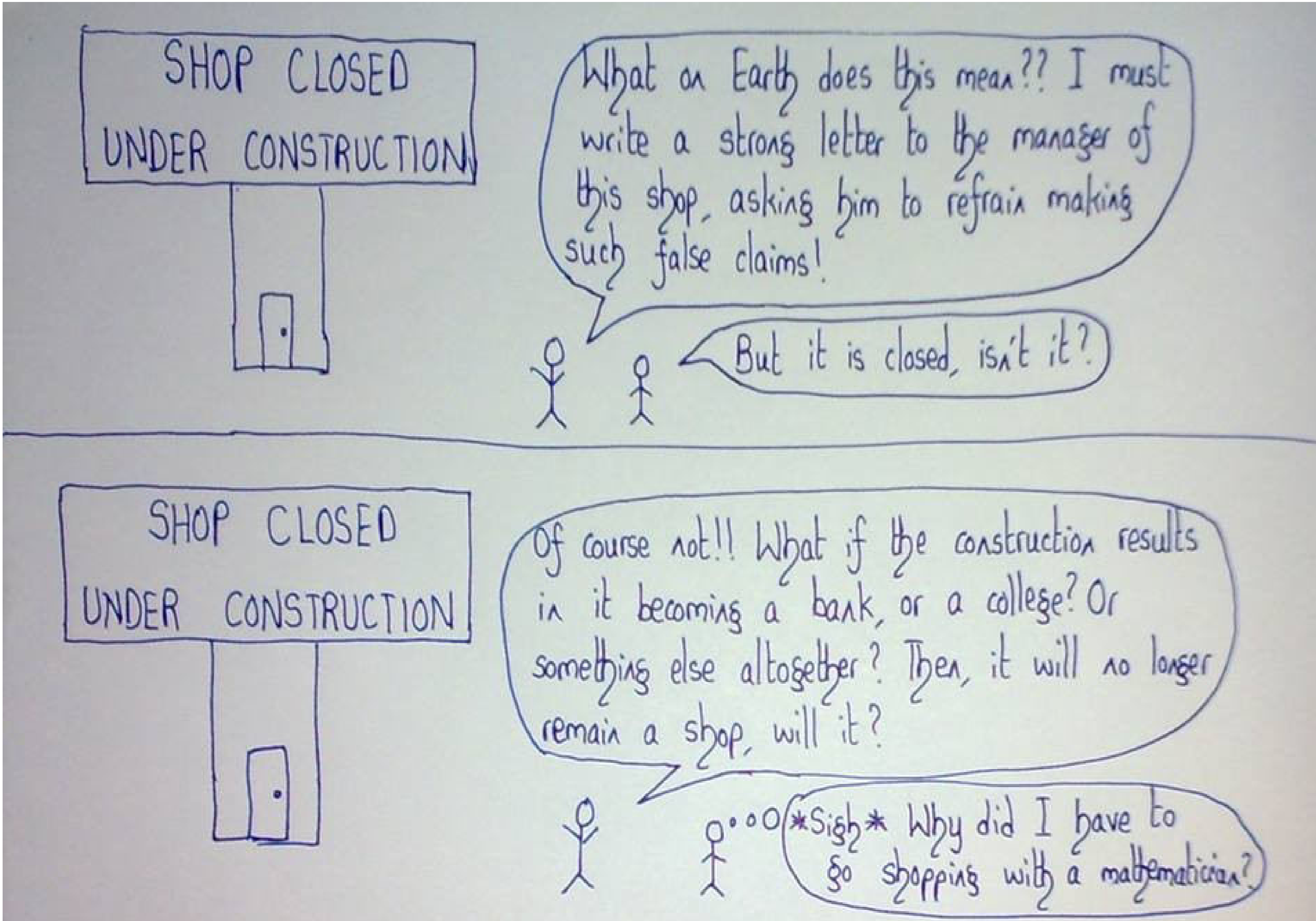
- S: Non-empty finite prefix-closed set of strings
- E: Non-empty finite suffix-closed set of strings
- T: Mapping from finite set of strings to {0,1}
- T(u) = 1 iff u belongs to U

Which of these sets are prefix-closed?

- {1110, 10, 1}
- {011, 0, λ, 11, 01}
- {110, 1, 0, λ, 11}
- {0, λ, 10, 010}

Which of these sets are suffix-closed?

- {1110, 10, 1}
- {011, 0, λ, 11, 01}
- {110, 1, 0, λ, 11}
- {0, λ, 10, 010}


SHOP CLOSED
UNDER CONSTRUCTION
What on Earth does this mean?? I must write a strong letter to the manager of this shop, asking him to refrain making such false claims!
But it is closed, isn't it?
SHOP CLOSED
UNDER CONSTRUCTION
Of course not!! What if the construction results in it becoming a bank, or a college? Or something else altogether? Then, it will no longer remain a shop, will it?
Sigh Why did I have to go shopping with a mathematician?

OBSERVATION TABLE

- Rows: Elements of S U S.A
- Columns: Elements of E
- Entry in row s and column e contains T(s.e)
- Initially S = E = { λ }
- row(s) denotes row of the table corresponding to s

What is row(a)? row(λ)?

T	λ
λ	1
a	0
b	0

TWO CRUCIAL PROPERTIES

- CLOSED: An observation table is closed if for all t belonging to S.A, there exists an s belonging to S such that row(t) = row(s)
- CONSISTENT: An observation table is consistent if whenever s1, s2 (both belonging to S) satisfy row(s1) = row(s2), then for all a belonging to A, row(s1.a) = row(s2.a)

Is this closed?

	λ	a	aa	aaa
λ	0	1	0	0
a	0	0	0	0
b	0	1	0	0

Is this closed?

	λ	a
λ	0	1
a	0	1
aa	0	0
ab	0	0
aaa	0	1
aab	0	0

Is this closed?

	λ	a
λ	0	1
a	1	1
aa	0	0
ab	0	0
aaa	0	1
aab	0	0

Is this closed?

	λ	a
λ	0	1
a	0	0
aa	0	0

ab	0	0
aaa	0	0
aab	0	0

Is this consistent?

	λ	a	aa	aaa
λ	0	1	0	0
a	0	0	0	0
b	0	1	0	0

Is this consistent?

	λ	a
λ	0	1
a	0	1
aa	0	0
ab	0	0
aaa	0	1
aab	0	0

Is this consistent?

	λ	a
λ	0	1
a	1	1
aa	0	0
ab	0	0
aaa	0	1
aab	0	0

Is this consistent?

	λ	a
λ	0	1
a	0	0
aa	0	0
ab	0	0
aaa	0	0
aab	0	0

Is this closed? Is it consistent?

T	λ
λ	1
a	0
b	0

The DFA

- Construct a DFA M(S, E, T) corresponding to closed and consistent table.
- Alphabet A
- State set Q
- Initial state q_0
- Accepting state set F
- Transition function δ

$$
\begin{aligned}
Q &= \{\mathrm{row}(s) : \; s \in S\} \\
q_0 &= \mathrm{row}(\lambda) \\
F &= \{\mathrm{row}(s) : \; s \in S, T(s) = 1\} \\
\delta(\mathrm{row}(s), a) &= \mathrm{row}(s \cdot a)
\end{aligned}
$$

Algorithm L^*

Initialize S and E to $\{\lambda\}$.
Ask membership queries for λ and a, $\forall a \in A$.
Construct the initial observation table (S, E, T).
Repeat:
    While (S, E, T) is not closed or not consistent:
        If (S, E, T) is not consistent,
            find $s_1, s_2 \in S, a \in A, e \in E$ such that $\mathrm{row}(s_1) = \mathrm{row}(s_2)$
                and $T(s_1 \cdot a \cdot e) \neq T(s_2 \cdot a \cdot e)$,
            add $a \cdot e$ to E,
            extend T to $(S \cup S \cdot A) \cdot E$ using membership queries.
        If (S, E, T) is not closed,
            find $s_1 \in S, a \in A$ such that $\mathrm{row}(s_1 \cdot a) \neq \mathrm{row}(s) \forall s \in S$,
            add $s_1 \cdot a$ to S,
            extend T to $(S \cup S \cdot A) \cdot E$ using membership queries.
    Perform an equivalence query with $M = M(S, E, T)$.
    If answer is "no" with counterexample t,
        add t and its prefixes to S,
        extend T to $(S \cup S \cdot A) \cdot E$ using membership queries.
Until answer is "yes" from equivalence query.

T	λ
λ	1
a	0
b	0

T_2	λ
λ	1
a	0
b	0
aa	1
ab	0

δ_2	a	b
q_0	q_1	q_1
q_1	q_0	q_1

Assume counterexample = bb

T_3	λ
λ	1
a	0
b	0
bb	1
aa	1
ab	0
ba	0
bba	0
bbb	0

T_4	λ	a
λ	1	0
a	0	1
b	0	0
bb	1	0
aa	1	0
ab	0	0
ba	0	0
bba	0	1
bbb	0	0

δ_4	a	b
q_0	q_1	q_2
q_1	q_0	q_2
q_2	q_2	q_0

Let counterexample = abb

T_5	λ	a
λ	1	0
a	0	1
b	0	0
bb	1	0
ab	0	0
abb	0	1
aa	1	0
ba	0	0
bba	0	1
bbb	0	0
aba	0	0
$abba$	1	0
$abbb$	0	0

T_6	λ	a	b
λ	1	0	0
a	0	1	0
b	0	0	1
bb	1	0	0
ab	0	0	0
abb	0	1	0
aa	1	0	0
ba	0	0	0
bba	0	1	0
bbb	0	0	1
aba	0	0	1
$abba$	1	0	0
$abbb$	0	0	0

δ_6	a	b
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

Algorithm L^*

Initialize S and E to $\{\lambda\}$.
Ask membership queries for λ and a, $\forall a \in A$.
Construct the initial observation table (S, E, T).
Repeat:
    While (S, E, T) is not closed or not consistent:
        If (S,E,T) is not consistent,
            find $s_1, s_2 \in S, a \in A, e \in E$ such that $\mathrm{row}(s_1) = \mathrm{row}(s_2)$
                and $T(s_1 \cdot a \cdot e) \neq T(s_2 \cdot a \cdot e)$,
            add $a \cdot e$ to E,
            extend T to $(S \cup S \cdot A) \cdot E$ using membership queries.
        If (S,E,T) is not closed,
            find $s_1 \in S, a \in A$ such that $\mathrm{row}(s_1 \cdot a) \neq \mathrm{row}(s) \forall s \in S$,
            add $s_1 \cdot a$ to S,
            extend T to $(S \cup S \cdot A) \cdot E$ using membership queries.
    Perform an equivalence query with $M = M(S, E, T)$.
    If answer is "no" with counterexample t,
        add t and its prefixes to S,
        extend T to $(S \cup S \cdot A) \cdot E$ using membership queries.
Until answer is "yes" from equivalence query.

Construct a DFA to accept all binary strings divisible by 3