

5 Automata

Wednesday, February 12, 2025

12:33 PM

Mathematical Languages

Problems: $\begin{cases} \rightarrow \text{Specification: RegEx} \\ \rightarrow \text{Recognition: Automata} \end{cases}$

Spec: $L = (+|-)?([0-9])^+ (\cdot [0-9]^+)? e (+|-)? [0-9]^+$

Rec

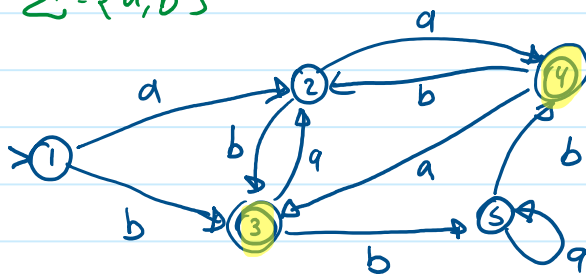
Is $27.3e2.7 \in L$? No

• Automata: a Mathematical Machine.

DEF:

- Set of states
 - initial state
 - subset of "accept" states
- Transition function
(state, symbol) $\rightarrow$ state

E.G. $\Sigma = \{a, b\}$



How to recognize:

Given a word w

- start in the initial state, and follow the transition function for each symbol in w

e.g. $abaabba$
 12324232

- If w ends in an accept state then $w \in L$
otherwise $w \notin L$

$L = \{b, aa, ab, abab, aaa, baa, bbb, bb\ aaaaaaa\ b, \dots\}$
not in $L = \{a, bb, ba,$

THEOREM

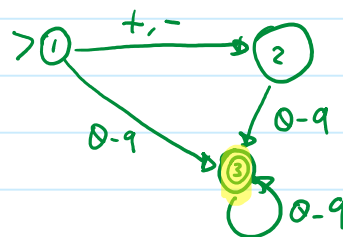
For every Language specified by RegEx
there exists an Automata that recognizes it.

E.g. Integer Literal.

$(+|-)?([0-9])^+$

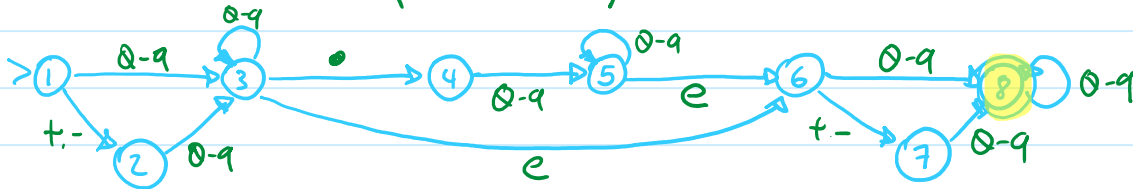
0 ✓ 00 ✓
12 ✓ +101 ✓
-17 ✓ -00 ✓

Automata



E.g. Scientific Numbers.

$(+|-)?([0-9])^+ (\cdot [0-9]^+)? e (+|-)? [0-9]^+$

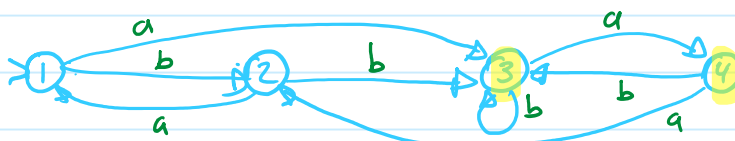


• Coding an automata:

- Loop over each character.
- at every iteration, use a **switch** statement to decide on the new state
 - each **case** of the switch corresponds to one state of the automata.
 - inside each case, consider all transitions from that state.

- inside each case, consider all transitions from that state.

E.G.



$\Sigma = \{a, b\}$ $L = \{a, bb, aa, ab, bbb, \dots\}$

Pseudocode.

```

FUNCTION recognize ( string w ) : BOOLEAN
VAR state, i, c, acc
BEGIN

```

```

    state := 1;    i := 0;    acc := False;

```

```

    WHILE i < len(w) DO

```

```

        c := w[i]

```

```

        SWITCH state OF

```

```

            CASE 1 : IF c = a THEN state := 3

```

```

                     IF c = b THEN state := 2

```

```

            CASE 2 : IF c = a THEN state := 1

```

```

                     IF c = b THEN state := 3

```

```

            CASE 3 : IF c = a THEN state := 4

```

```

                     IF c = b THEN state := 3

```

```

            CASE 4 : IF c = a THEN state := 2

```

```

                     IF c = b THEN state := 3

```

```

        END

```

```

        i := i + 1

```

```

    END

```

```

    IF state = 3 OR state = 4 THEN

```

```

        acc := True

```

```

    RETURN acc

```

```

END.

```

$w = bbaab\$$ $acc \checkmark$

state ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~3~~

We could also:

- Simplify the switch statement
- Add fail and sink states
- Return early.
- Turn switch statement into lookup table.

state \ c	a	b
1	3	2
2	1	3
3	4	3
4	2	3

- other simplifications

- This is HW #1

—○— EOF