MATH 2301 30 Sept 2020

* Recap: We defined valid regular expressions over an alphabet Σ.

  Precedence order: $* \gg$ concatenation $\gg$ |

  (In the absence of parentheses, apply * first, concat next and | last.
  Just like in an algebraic expression:
  $\wedge \gg \times \gg +$)

* We have $r \longleftrightarrow L(r) \subseteq \Sigma^*$
  the language of r

* We say $s \in \Sigma^*$ matches r if $s \in L(r)$.

* Theoretical vs practical regular expressions: a demo.
  [We saw +, •, complement as extra syntax in practical regex]
  we exclude this syntax and keep minimal syntax so it's easier to prove theorems. Many of the extra operations can be simulated with the ones we have]

* Today: Which languages can be expressed as $L(r)$ for some regex r?

  Claim: The language $L = \{0^n 1^n \mid n \in \mathbb{N}\}$
  (n 0s followed by n 1s)

  is not equal to $L(r)$ for any regex r.

  Exercise:- Try (and fail) to construct a regex for L.

  We'll try to prove this claim later.

  Def: A language L is called regular if $L = L(r)$ for some regex r.

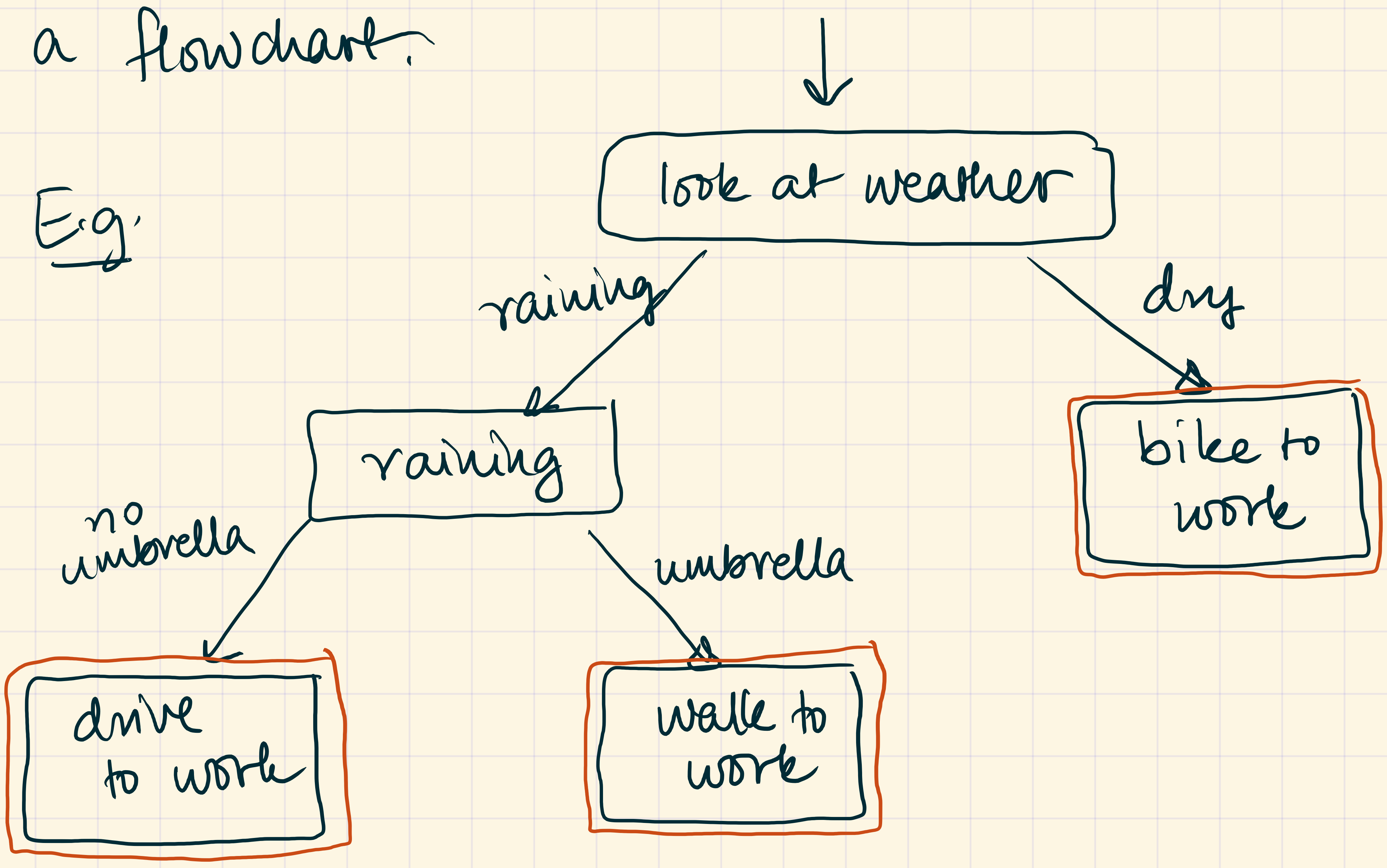
  [Not all languages will be regular.]

* (Deterministic) finite automata, or DFAs
(Finite state machines)

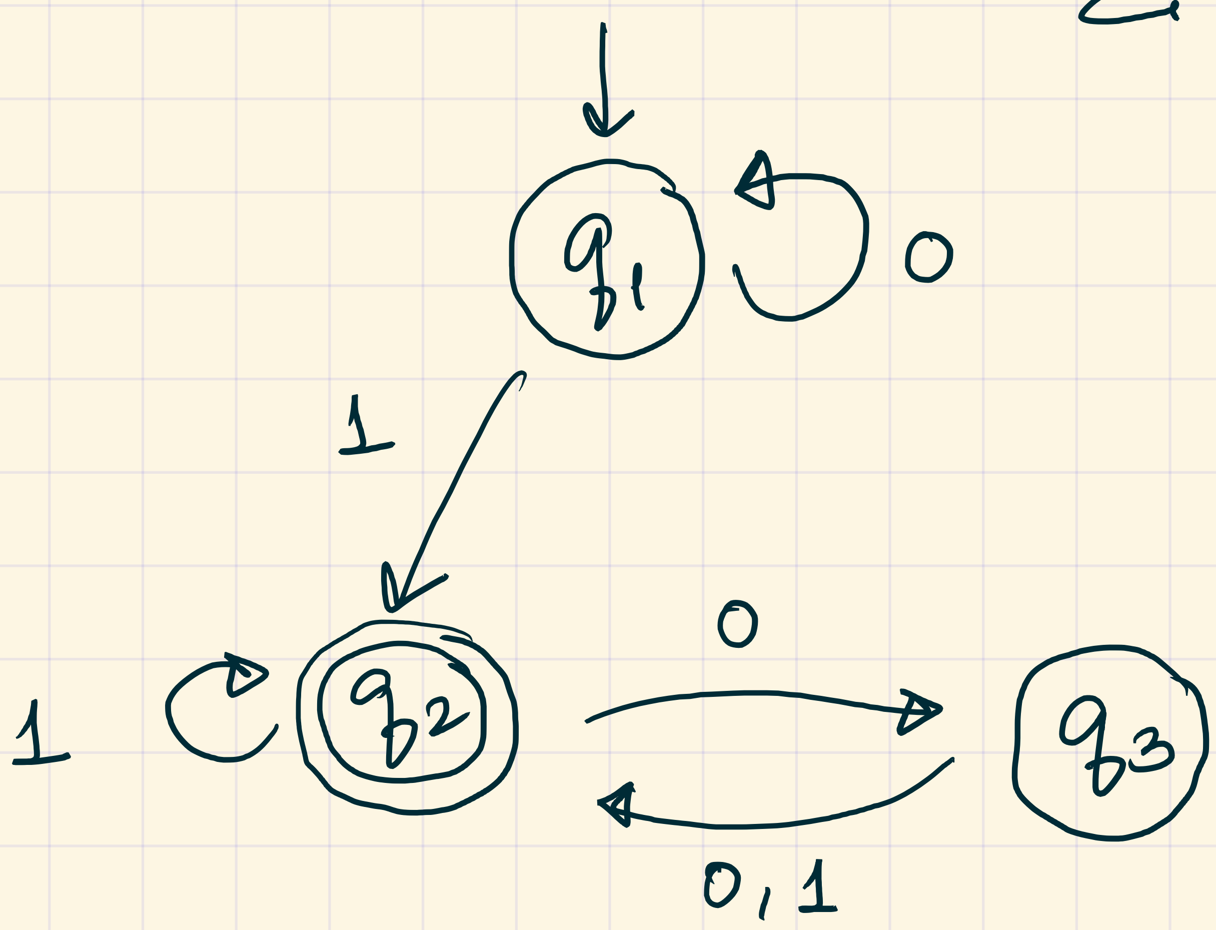
These will end up being machines / model mini computers that exactly recognise regular languages.

Informally, a DFA is a machine that simulates a flowchart.

E.g.



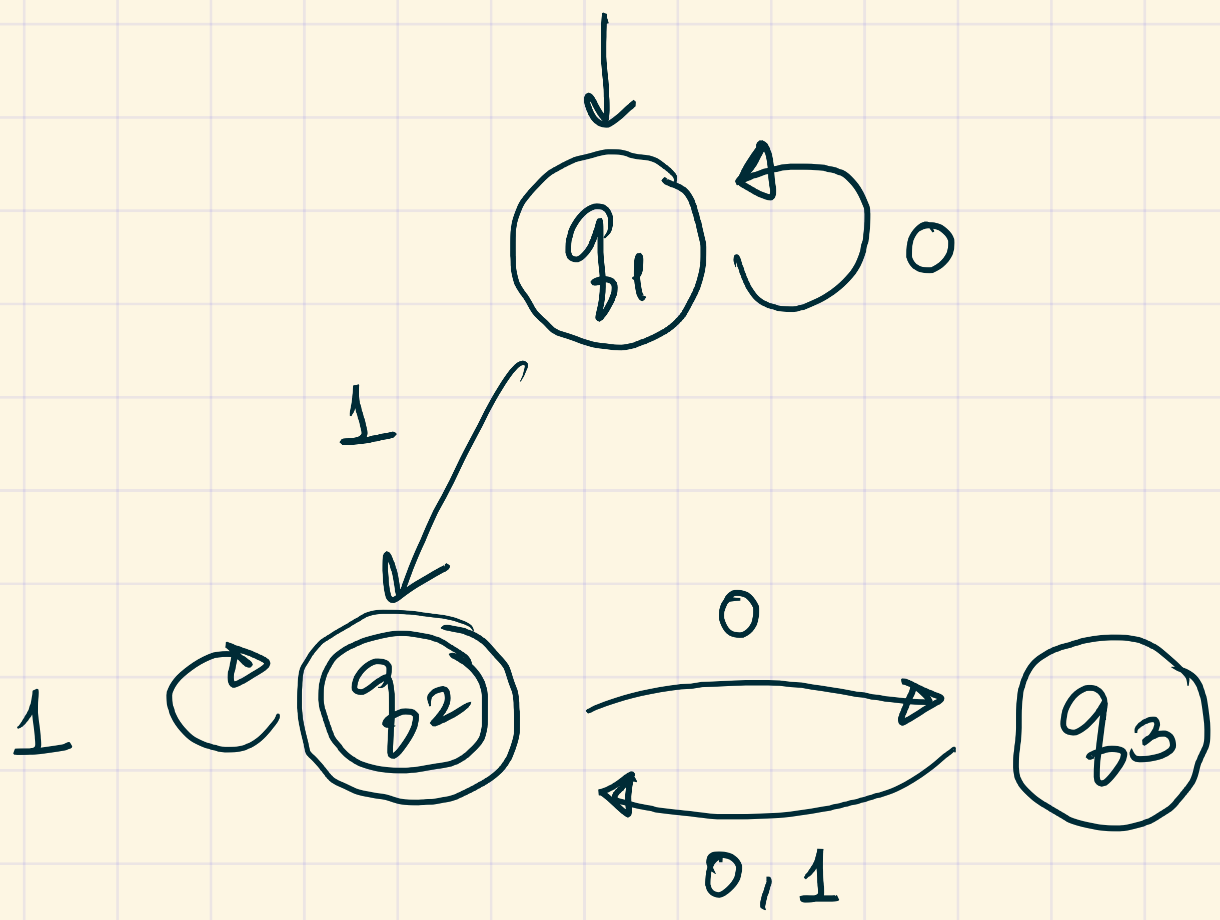
E.g.
Here is a DFA: [you need an alphabet Σ, say $\Sigma = \{0,1\}$]



States: q_1, q_2, q_3

Start state: q_1

Each state has one outgoing arrow labelled by each $a \in \Sigma$.
Every arrow except for the start arrow is labelled by some $a \in \Sigma$.



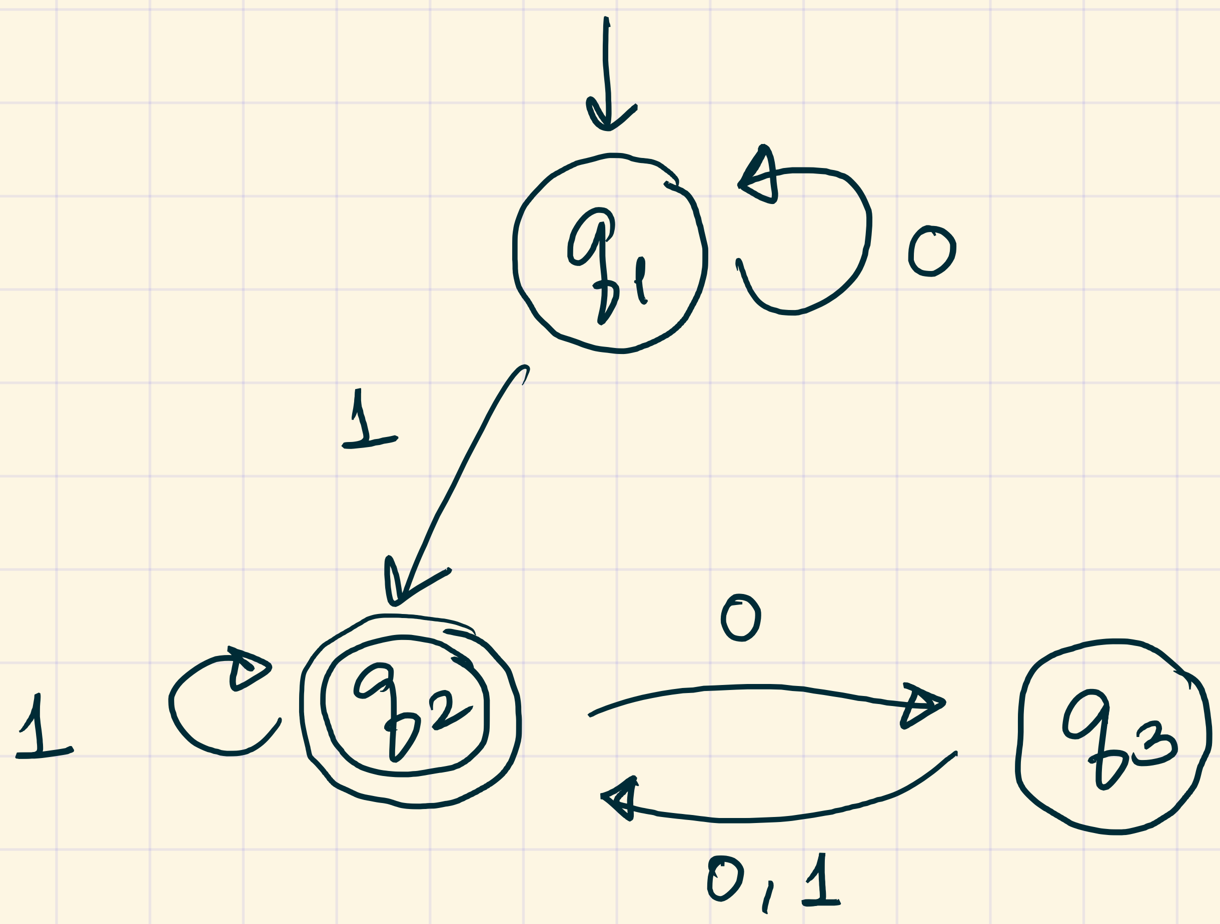
(double circled)
q_2 is an accept state

q_1, q_3 are reject states
(or fail / failure states)

Given any word $w \in \Sigma^*$ you can "run" the DFA on that word, and the DFA will either accept or reject w.

Eg. $w = 011101$

Start at q_1. "Read" the first letter, follow the corresponding arrow, continue.



Start at q_1

1. Read 0 → go to q_1
2. Read 1 → go to q_2
3. Read 1 → go to q_2
4. Read 1 → go to q_2
5. Read 0 → go to q_3
6. Read 1 → go to q_2

Since q_2 is an accept state, we accept w.

$w = 10, 010, 01110, 0, \varepsilon$ are all rejected.

Suggestion (Mikayla): $0^* 1 (1 | 00 | 01)^* = r$

then $L(r)$ is exactly the set of accepted strings of this DFA (Check!)