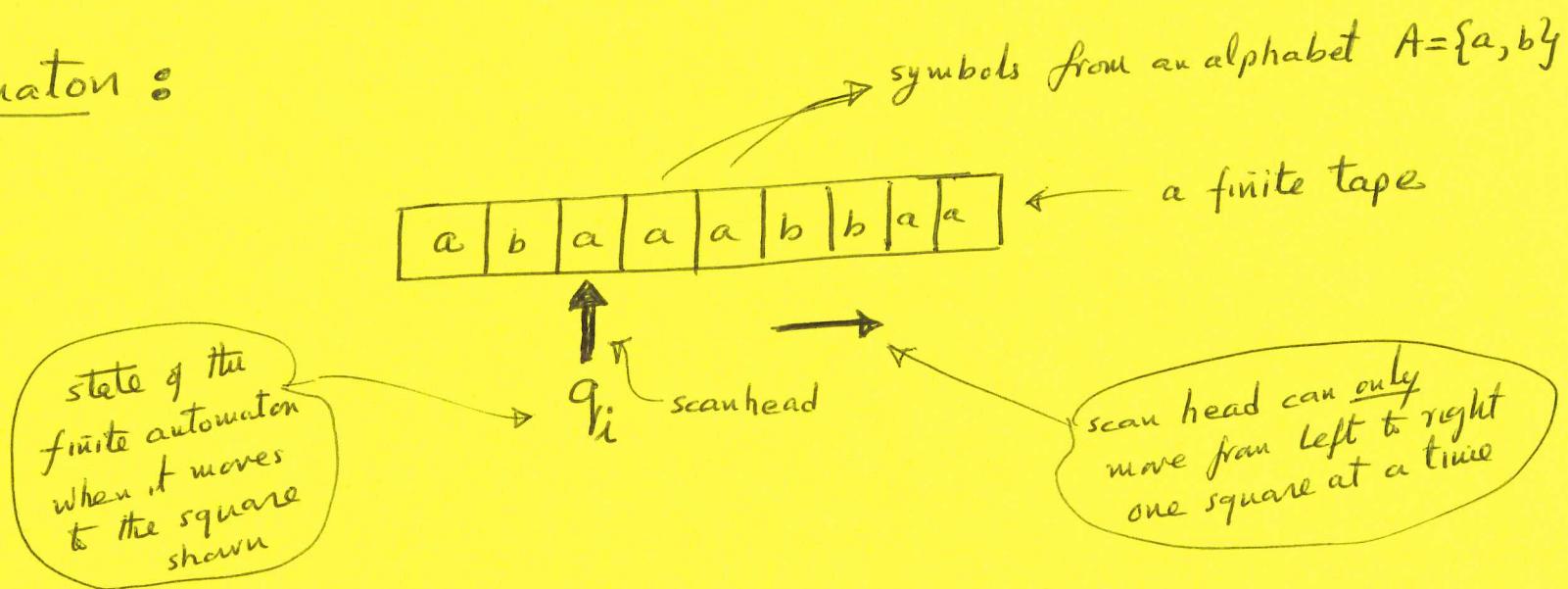


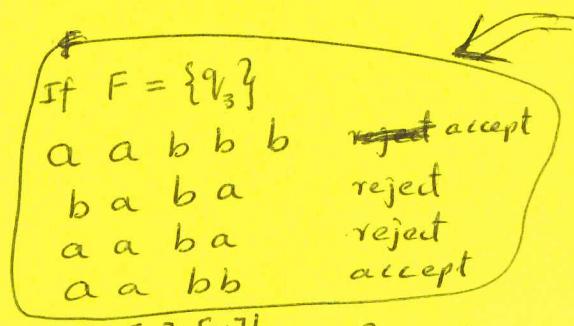
FINITE AUTOMATA and REGULAR LANGUAGES

A Finite Automaton :

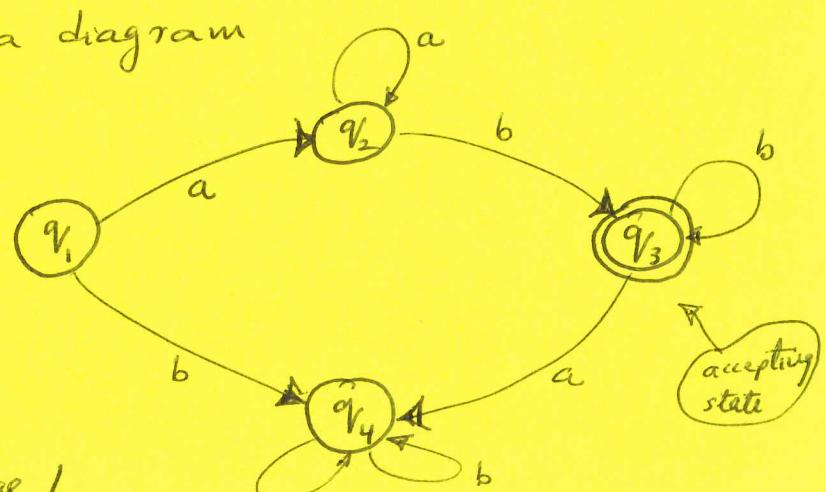


A finite automaton M consists of

- ① A tape with a finite number of squares and a scanhead that can move only from left to right one square at a time.
- ② We say M is in some state q_i as ~~M enters~~ the scanhead enters a square. Let the ~~state set~~ of states be $Q = \{q_1, q_2, \dots, q_m\}$
- ③ The symbols in the squares are supposed to come from an alphabet $A = \{s_1, s_2, \dots, s_n\}$
- ④ Scanning a symbol s_j in state q_i causes M to transition to state q_k . We say $S(q_i, s_j) = q_k$. S is called the transition function.
- ⑤ One of the states, usually q_1 , is singled out and called the initial state. M starts out in state q_1 with the scanhead looking at the first symbol.
- ⑥ We specify a set $F \subseteq Q$ of states to be the accepting or final states.
- ⑦ We say that M accepts a word $u \in A^*$ if all the symbols on the tape constitute the word u and if the last state of M is in F .
- ⑧ The transition function $S(q_i, s_j)$ may be displayed either in the form of a table or as a diagram



s	a	b
q_1	q_2	q_4
q_2	q_2	q_3
q_3	q_4	q_3
q_4	q_4	q_4

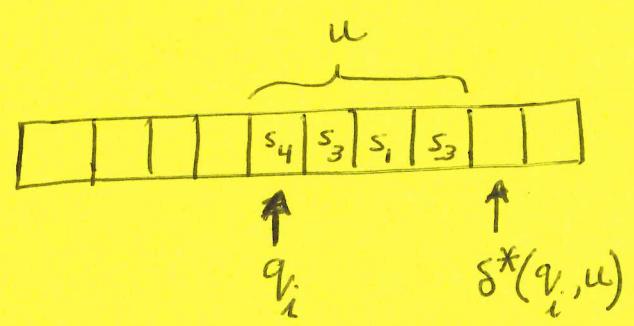


- ⑨ The set of all words is the language L accepted by a finite automaton M .

REGULAR LANGUAGE :

- Let $u \in A^*$
- We will denote by $\delta^*(q_i, u)$ the state that M enters after it has finished scanning u assuming that M was in state q_i when it started scanning u .
- A formal definition for δ^* by recursion :

$$\begin{aligned}\delta^*(q_i, 0) &= q_i \\ \delta^*(q_i, us_j) &= \delta(\delta^*(q_i, u), s_j)\end{aligned}$$
- We say a finite automaton M accepts a word u if $\delta^*(q_i, u) \in F$
- If $L(M)$ is the language accepted by M , we have $L(M) = \{u \in A^* \mid \delta^*(q_i, u) \in F\}$



A language is called regular if there exists some finite automaton that accepts it.

A finite automaton as described is also called a deterministic finite automaton (dfa).

NONDETERMINISTIC FINITE AUTOMATA

- A nondeterministic finite automaton (ndfa) permits transitions at each stage to zero, one, or more than one state.
- Whereas for a dfa, the value of $\delta(q_i, s_j)$ is a single state, for an ndfa the value will be a set of states (including the null set).
- The meaning to be associated with $\delta(q_i, s_j)$ being a set of states is that in a hardware implementation the machine could be in any of the states in the set.
- Given an ndfa with the transition function $\delta(q_i, s_j)$, we now define a function $\delta^*(q_i, u)$ where $u \in A^*$ as follows :

$$\begin{aligned}\delta^*(q_i, 0) &= \{q_i\} \\ \delta^*(q_i, us_j) &= \bigcup_{q \in \delta^*(q_i, u)} \delta(q, s_j)\end{aligned}$$

- An ndfa M with initial state q_i accepts a word $u \in A^*$ if $\delta^*(q_i, u) \cap F \neq \emptyset$

THEOREM : A language is accepted by an ndfa iff it is regular. In other words, a language is accepted by an ndfa iff it is accepted by a dfa.

PROOF : Going from dfa to ndfa, the proof is very simple. Suppose we are given a dfa that accepts a language L . Let the transition function of this dfa be $\delta(q_i, s_j) = q_k$. The transition function of the ndfa that will accept the same language is $\delta(q_i, s_j) = \{q_k\}$.

The proof is more complicated in going from ndfa to dfa. Let M be an ndfa that accepts a language L . We will now construct a dfa \tilde{M} that will accept the same language L . \tilde{M} will use the same alphabet as M . The states of the dfa \tilde{M} will be the elements of the power set of the set Q of the states of the ndfa M . That is $\tilde{Q} = \{\tilde{Q}_1, \tilde{Q}_2, \dots, \tilde{Q}_m\}$ where $m = |Q|$. We assume $\tilde{Q}_1 = \{q_i\}$. q_i will be the start state of the dfa \tilde{M} . The acceptance states of \tilde{M} is the set $\tilde{F} = \{\tilde{Q}_i \mid Q_i \cap F \neq \emptyset\}$ where F is the set of acceptance states of M .