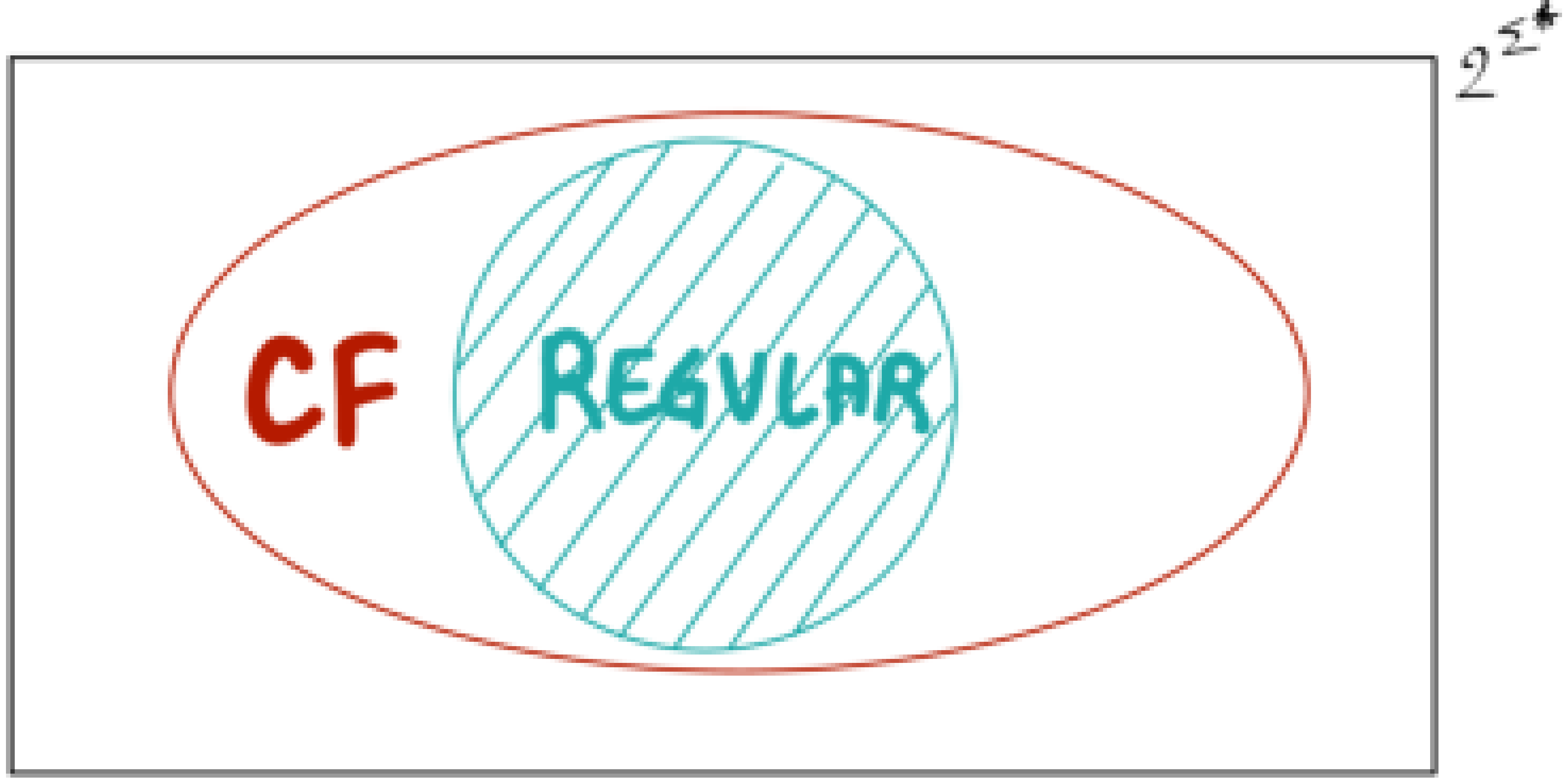
Turing Machines

So far:



We saw operational characterizations of these language classes.

Regular: DFA/NFA: Finitely many states, "one state memory"

Context-free: PDA: Finitely many states, stack (LIFO)

$\{a^n b^n a^n \mid n \geq 0\}$ is not CF.

$\{ww \mid w \in \{a,b\}^*\}$ is not CF.

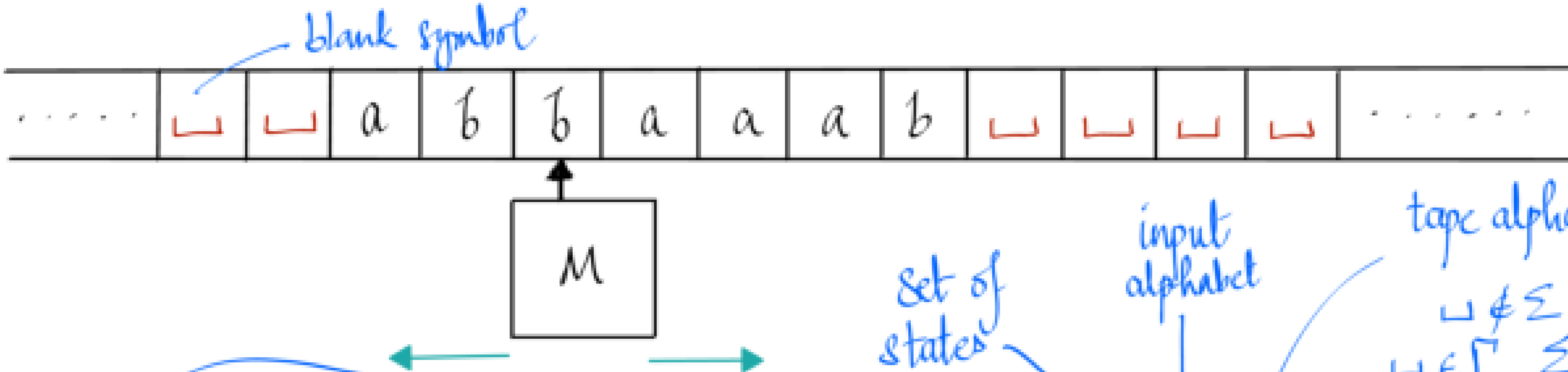
What can we add to a PDA to be able to recognize these?

A Turing machine is basically a finite-state automaton with a global tape
Each cell of the tape can contain a letter from a finite alphabet
Start out with the input string, and the rest of the tape is empty.

What can the machine do?

* Read off the tape * Write on the tape * Move the head left or right

At each step, the head scans some cell of the tape,
and the automaton is in some state, depending on which
the automaton can move to some (other) state,
the machine can write some new symbol on the tape cell, and
move the head one cell to the left or to the right



A deterministic* Turing Machine: $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$

Set of states

input alphabet

tape alphabet

$\sqcup \notin \Sigma$

$\sqcup \in \Gamma$, $\Sigma \subseteq \Gamma$.

start state

accept state

reject state

$s, t, r \in Q$.

$\delta: Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

current state

contents of current cell

new state

new contents of current cell

direction in which tape head should move

$Q \setminus \{t, r\}$

DFA, but no outgoing transitions from t and r

Input $w_1 w_2 \cdots w_n$, each $w_i \in \Sigma$

Written on n cells of the tape

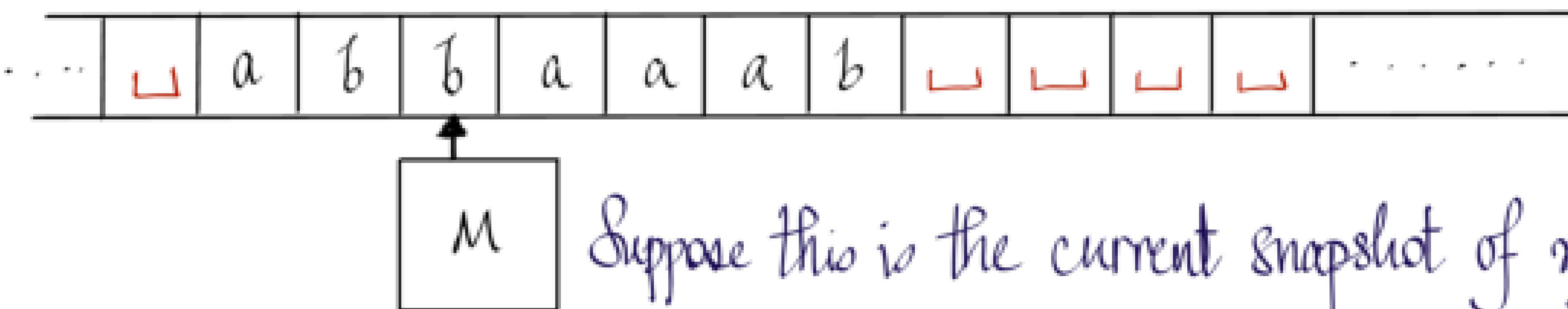
The first blank signals the end of input.

As part of its operation, the following parameters change:

- → current state
- → current tape contents
- → location of tape head

These form a configuration

$(ab, b, aaab, q)$



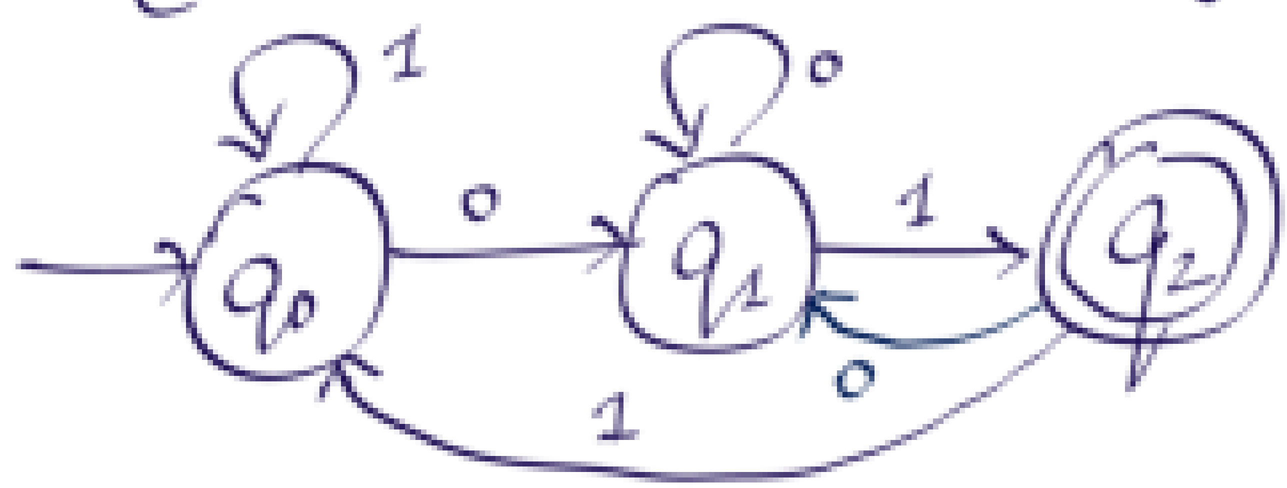
Suppose this is the current snapshot of my TM M, in state q.

We write the configuration as $abqbaaab$.

Formally, a configuration is uqv, where the machine is in state q, the current tape contents are uv, $u, v \in \Gamma^*$, and the head points to the cell containing the first letter of v.

Example:

$\mathcal{L} = \{ \omega \mid \omega \text{ ends with } 01 \} \subseteq \{0,1\}^*$



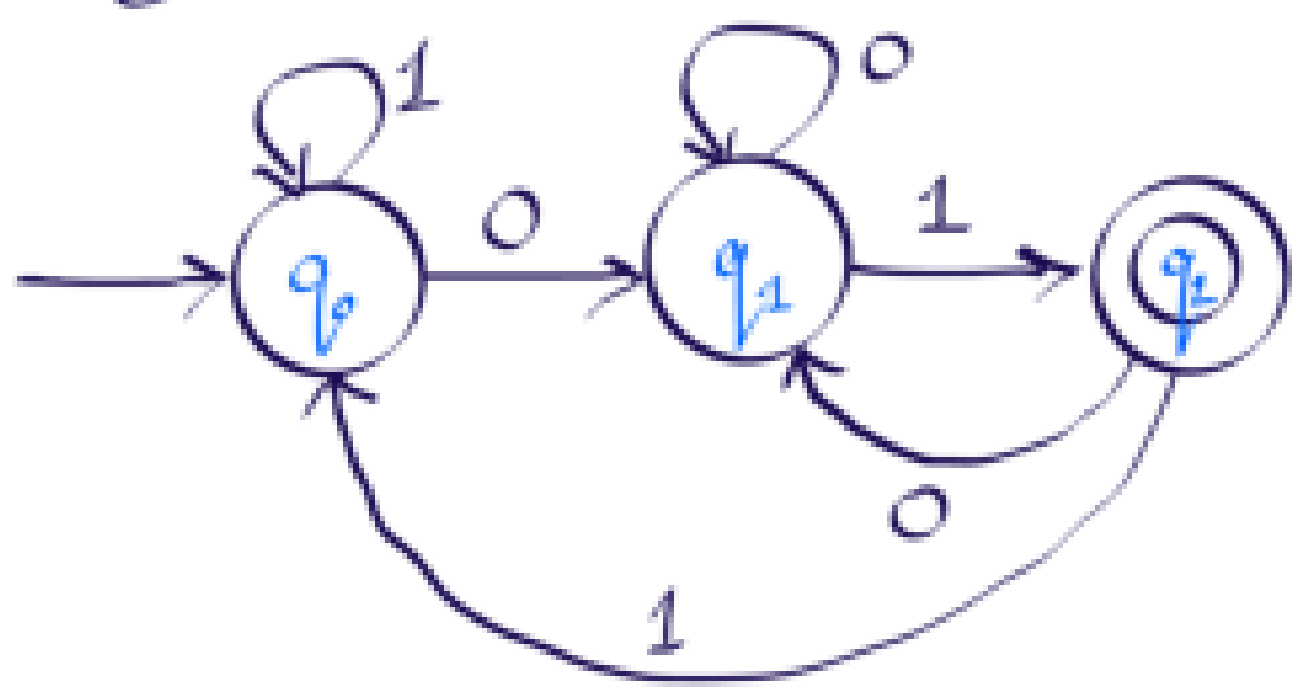
$M = (Q, \Sigma, \Gamma, \delta, s, t, r)$

$\Sigma = \{0,1\}$ $\Gamma = \{0, 1, \sqcup\}$

$s = q_0$ $Q = \{q_0, q_1, q_2, t, r\}$

Example:

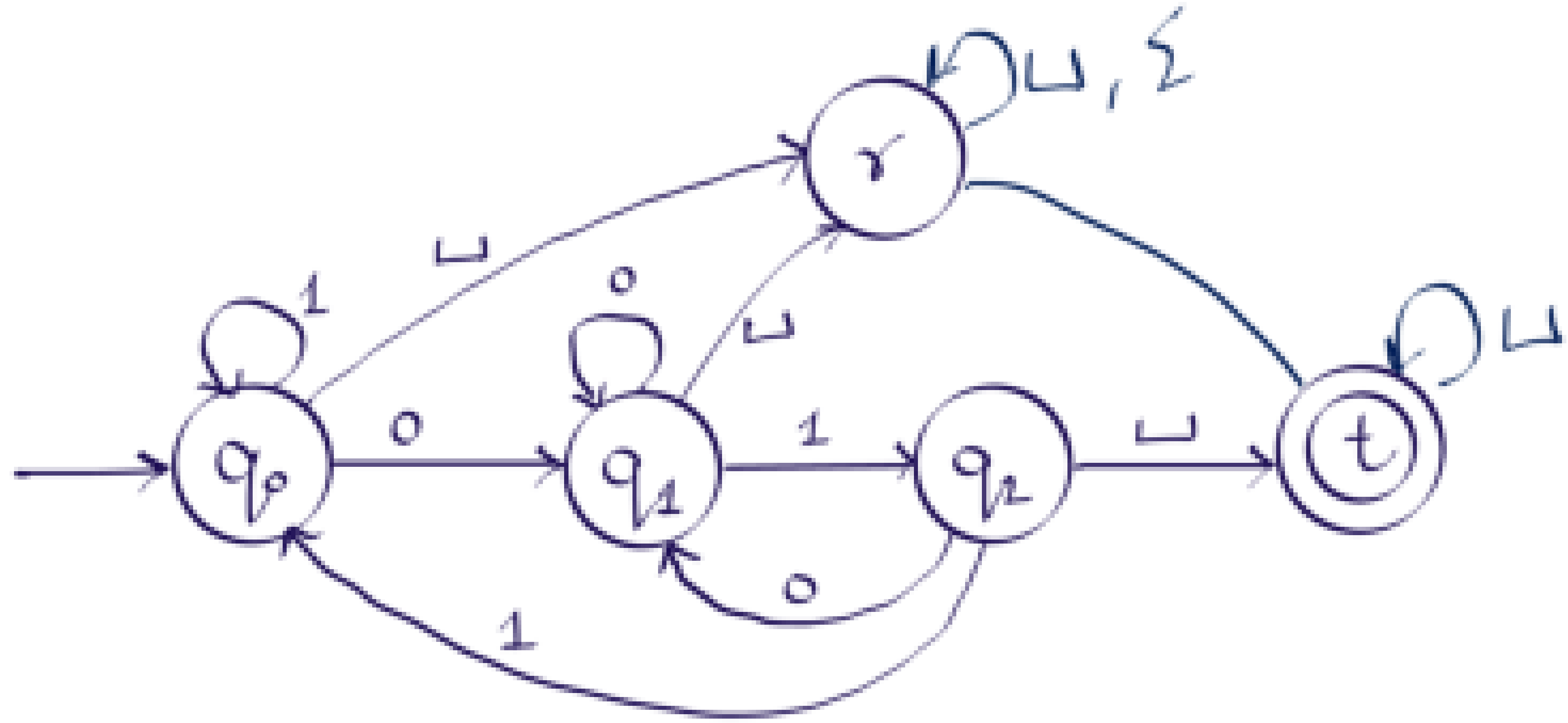
$$\mathcal{L} = \{ \omega \mid \omega \text{ ends with } 01 \} \in \{0,1\}^* \qquad \Sigma = \{0,1\}$$



$Q = \{q_0, q_1, q_2, t, r\}$

$\Gamma = \{0, 1, \sqcup\}$

$M = (Q, \Sigma, \Gamma, \delta, q_0, t, r)$

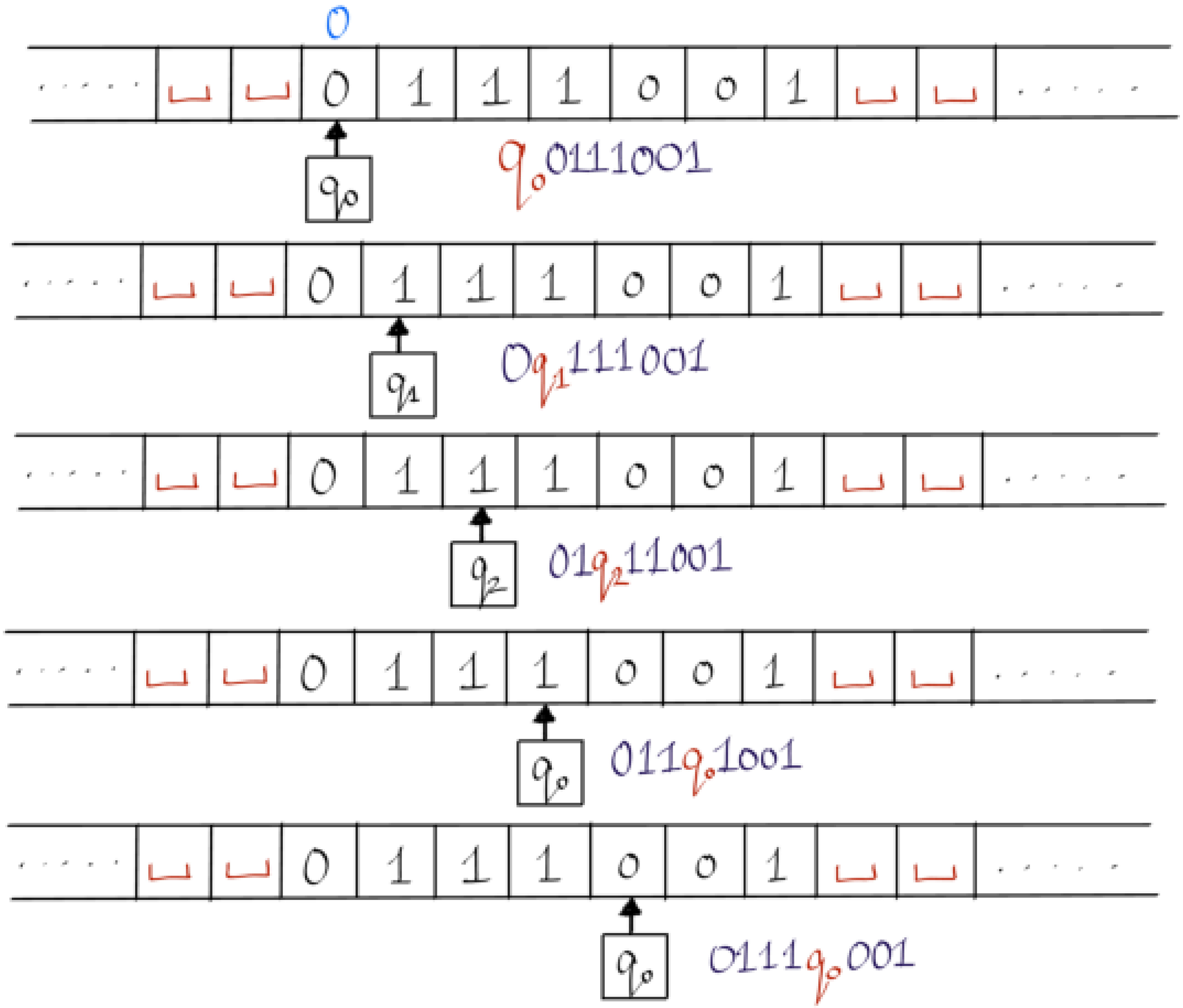


$\delta(q_0, 0) = (q_1, 0, R)$

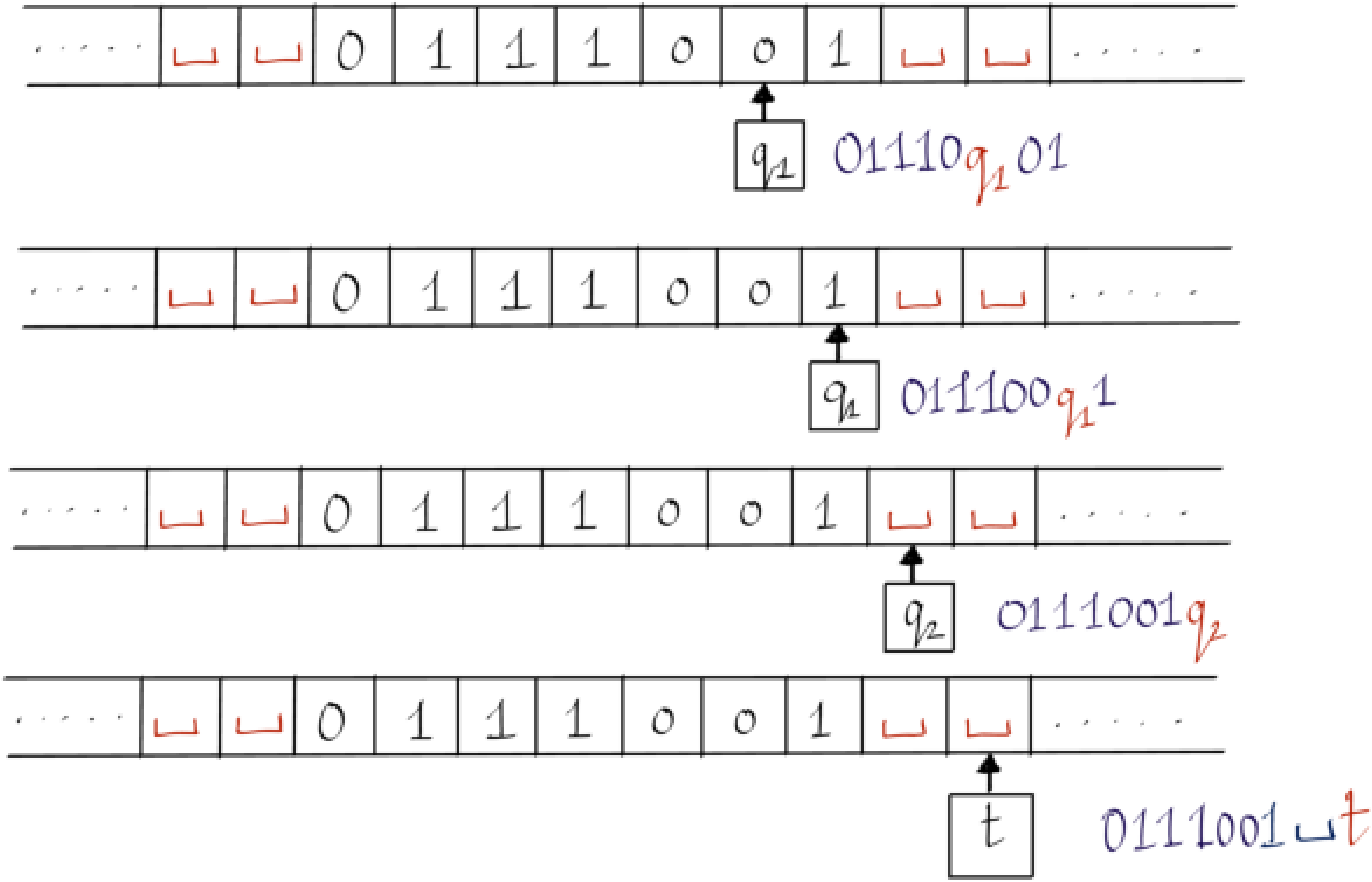
$\delta(q_0, 1) = (q_0, 1, R)$

$\delta(q_1, 0) = (q_1, 0, R)$

$\delta(q_2, \sqcup) = (t, \sqcup, R)$



0
q0 0111001
0q1 111001
01q2 11001
011q0 1001
0111q0 001


01110q1 01
011100q1 1
0111001q2
0111001␣t

Let $u, v \in \Gamma^*$, and $a, b, c \in \Gamma$. Then,

$$uaqbv \xrightarrow[M]{1} uacq'v \quad \text{iff} \quad \delta(q,b) = (q', c, R), \quad \text{and}$$

$$uaqbv \xrightarrow[M]{1} uq'acv \quad \text{iff} \quad \delta(q,b) = (q', c, L)$$

The initial configuration is of the form $s\omega$, where ω is the input.

ω is accepted by M if $s\omega \xrightarrow[M]{*} utv$, for some $u, v \in \Gamma^*$.

accepting configuration

Similarly,

ω is rejected by M if $s\omega \xrightarrow[M]{*} urv$, for some $u, v \in \Gamma^*$.

rejecting configuration

M does not proceed beyond an accepting or a rejecting configuration.

A halting configuration is either an accepting or rejecting configuration.

$$\mathcal{L}(M) = \{ w \mid s w \xrightarrow[M]{*} u t v, \text{ for some } u, v \in \Gamma^* \}$$

$\mathcal{L}$ is Turing-recognizable if there exists a TM M s.t. $\mathcal{L} = \mathcal{L}(M)$.

What is the "language of a Turing machine" $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$?

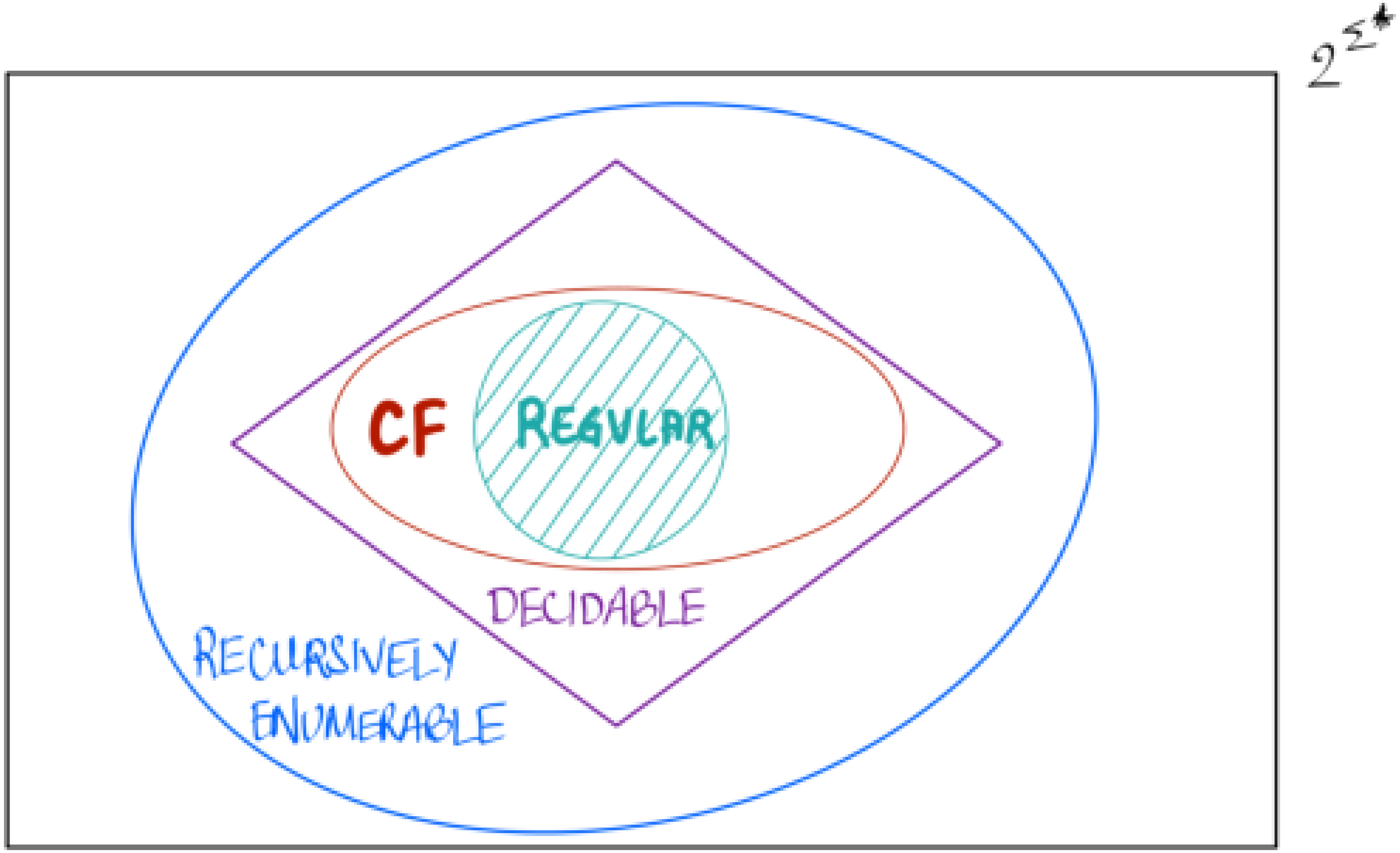
Consider $P_M = \{ \omega \mid$ there are $u, v \in \Gamma^*$ s.t. $s\omega \xrightarrow[M]{*} utv \}$, and

$N_M = \{ \omega \mid$ there are $u, v \in \Gamma^*$ s.t. $s\omega \xrightarrow[M]{*} urv \}$

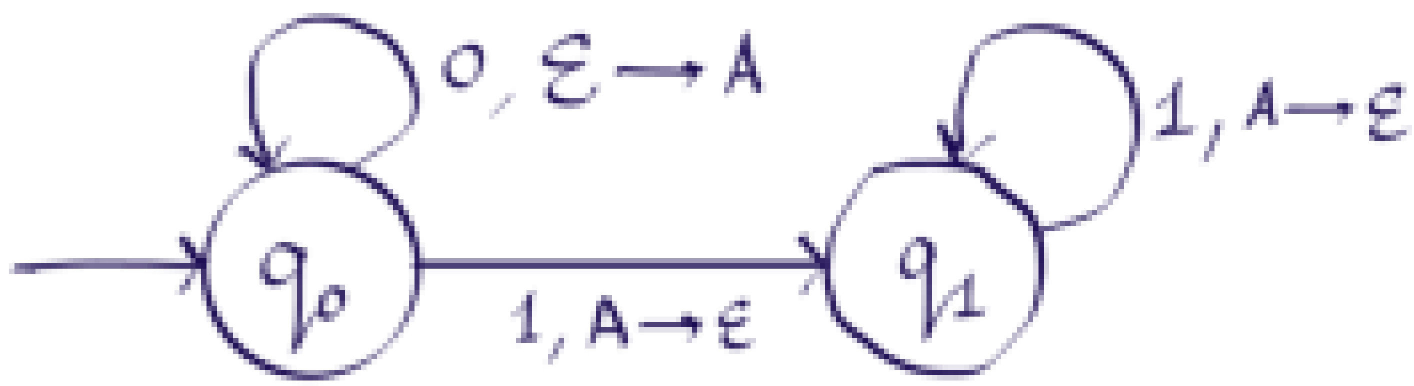
Suppose $\mathcal{L} = P_M$. Then, we say that $\mathcal{L}$ is recognized by M, and that $\mathcal{L}$ is Turing-recognizable, or recursively enumerable (r.e.)

If $\mathcal{L}$ is recognized by M, and in addition, $N_M = \{0,1\}^* \setminus P_M$, then we say that $\mathcal{L}$ is decided by M, and that $\mathcal{L}$ is decidable, or recursive

Decidability $\Rightarrow$ Turing-recognizability (but not the other way!)

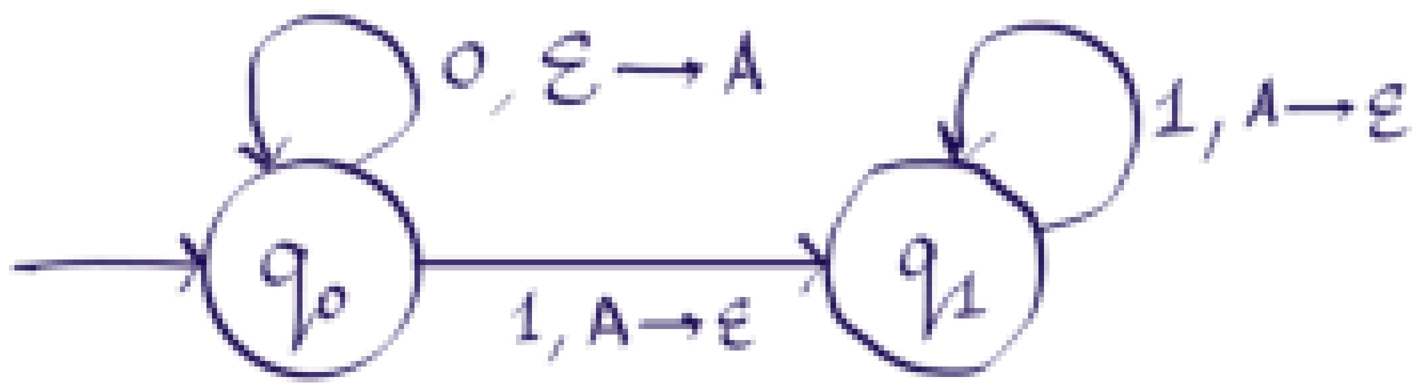

2^{Σ^*}
CF
REGULAR
DECIDABLE
RECURSIVELY
ENUMERABLE

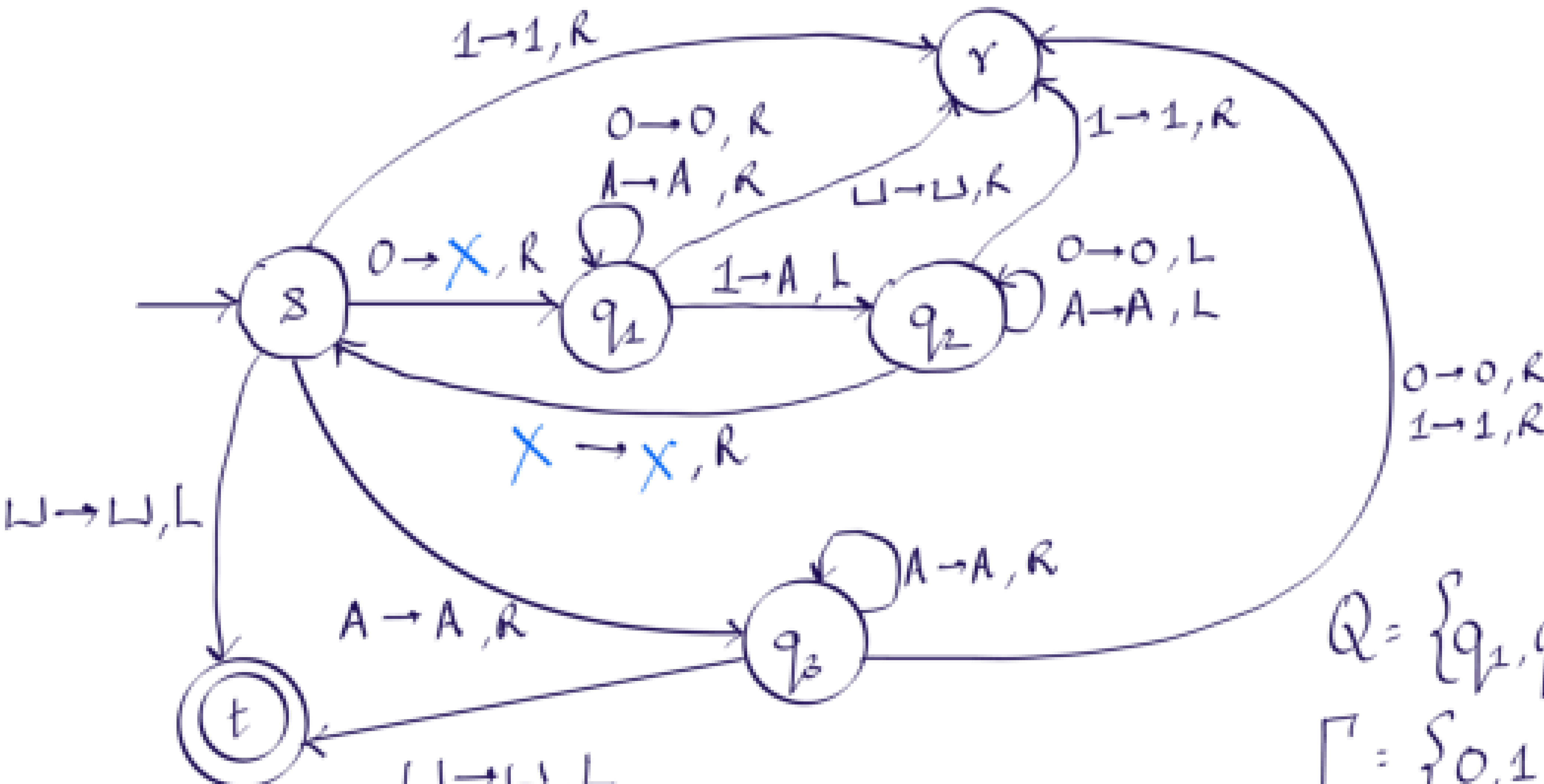
$$L = \{0^n 1^n \mid n \geq 0\}$$



$$M = (\{q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, A\}, \delta, q_0, \phi)$$

$L = \{0^n 1^n \mid n \geq 0\}$





$Q = \{q_1, q_2, q_3, s, t, r\}$

$\Gamma = \{0, 1, A, \sqcup, X\}$