Intro to Theory of Computation

CS 464

LECTURE 10

Last time:

- Pumping Lemma for CFLs
- Review of CFGs/PDAs

Today

- Review
- Turing Machines

Homework 4 due

Practice midterm 1 out

Sofya Raskhodnikova

CS 464

I-clicker problem (frequency: AC)

Consider two statements about languages R and C:

1) If R is regular and C is a CFL then $R \cap C$ is a CFL.
2) If R is not regular and C is not a CFL then $R \cap C$ is not a CFL.

A. Both (1) and (2) are true.
B. Both (1) and (2) are false.
C. (1) is true and (2) is false.
D. (1) is false and (2) is true.
E. These statements are not propositions, so they can’t be true or false.

I-clicker problem (frequency: AC)

Choose a correct statement about finite languages.

A. All finite languages are regular.

B. All finite languages are not regular.

C. Some finite languages are regular, some are not, but all of them are context-free.

D. Some finite languages are context-free, some are not.

E. None of the above statements are true.

CS 464

I-clicker problem (frequency: AC)

We can define a union operation on more than two languages.

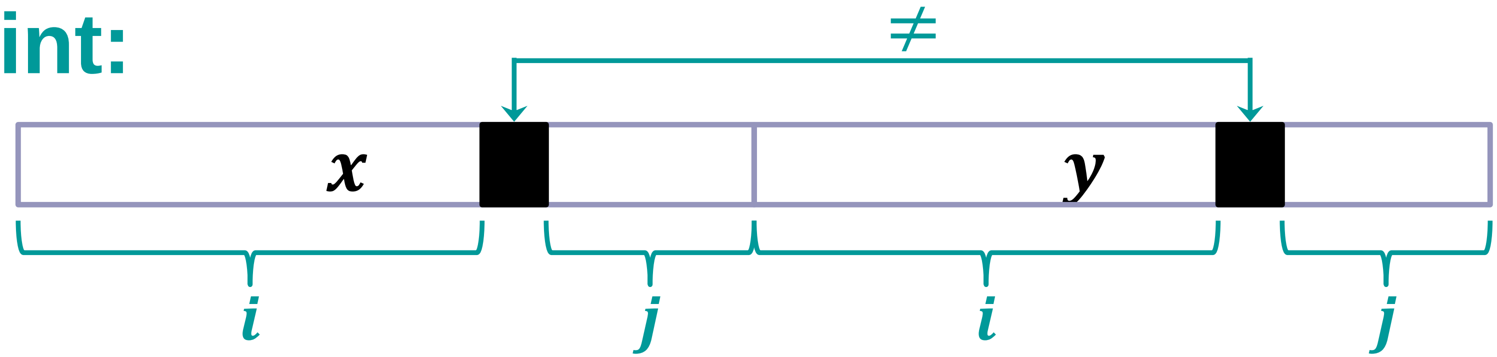
1) **Regular languages are closed under the union of 3.**
2) **Regular languages are closed under the infinite union.**

A. **Both (1) and (2) are true.**
B. **Both (1) and (2) are false.**
C. **(1) is true and (2) is false.**
D. **(1) is false and (2) is true.**
E. **These statements are not propositions, so they can’t be true or false.**

Prove the following language is CF

$\{xy \mid x, y \in \{0,1\}^* \text{ and } |x| = |y| \text{ but } x \neq y\}$.

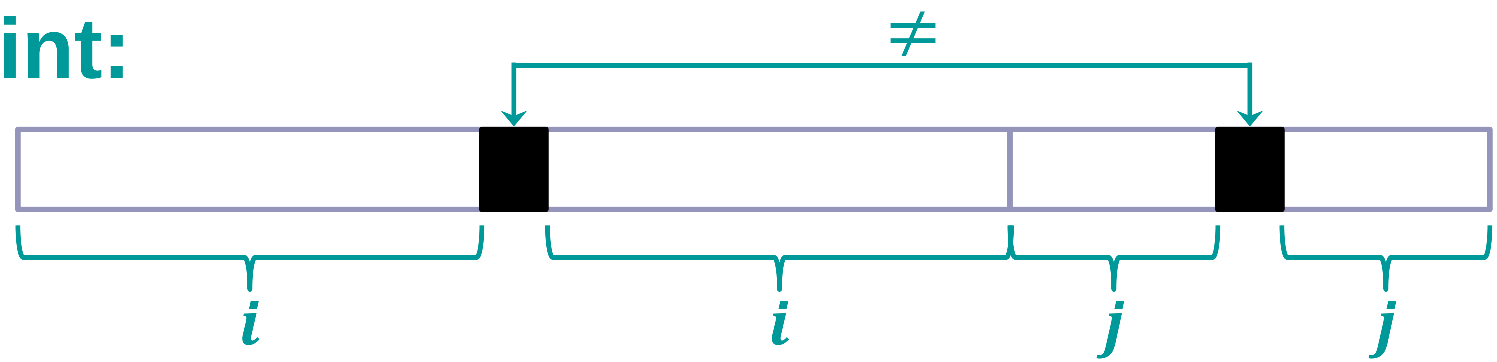
Hint:



Prove the following language is CF

$\{xy \mid x, y \in \{0,1\}^*$ **and** $|x| = |y|$ **but** $x \neq y\}$.

Hint:



CMPSC 464 so far

First model of a program: DFA / Regexp

Solvable Problems: Regular Languages

Unsolvable Problems: $\{ 0^n1^n \mid n \geq 0 \}$

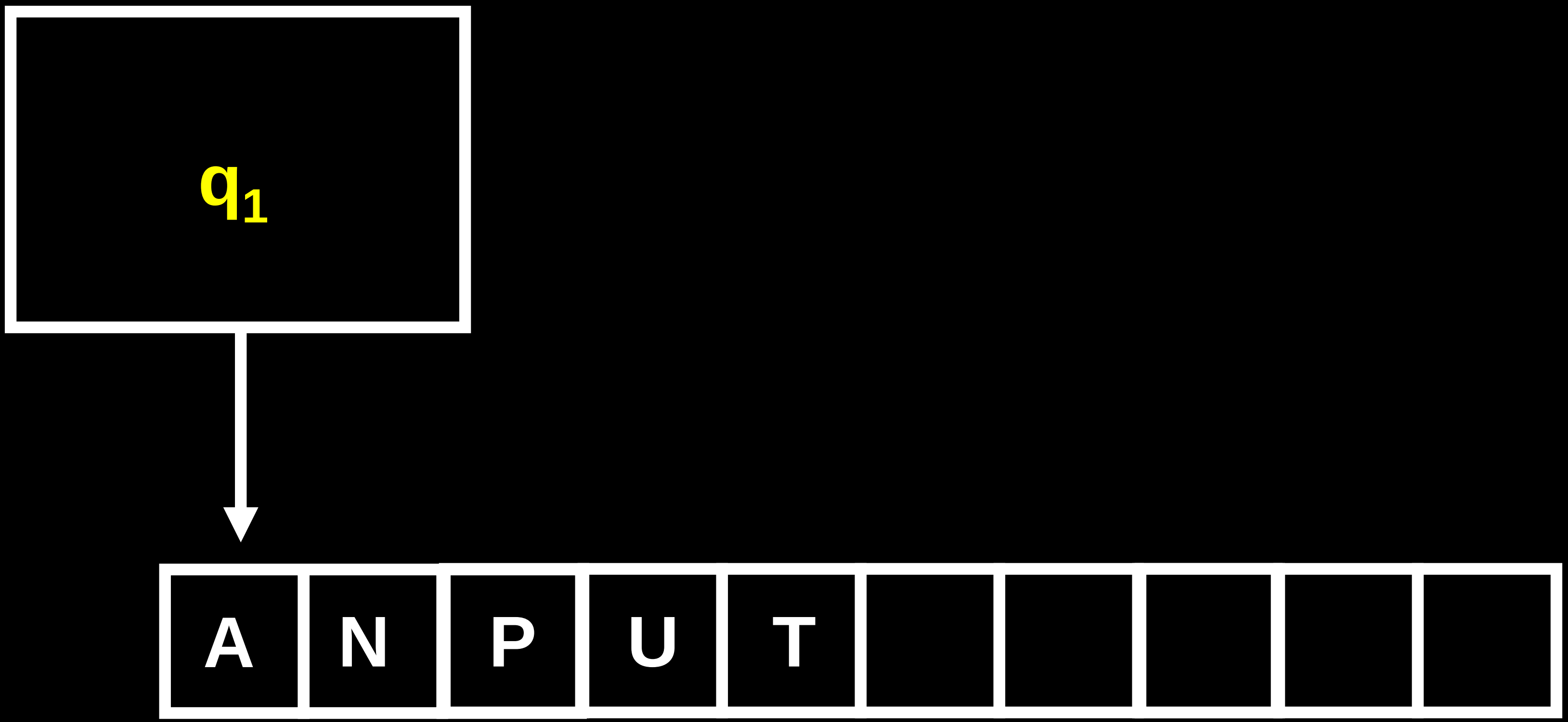
Next model of a program: PDA / CFG

Solvable Problems: Context-Free Languages

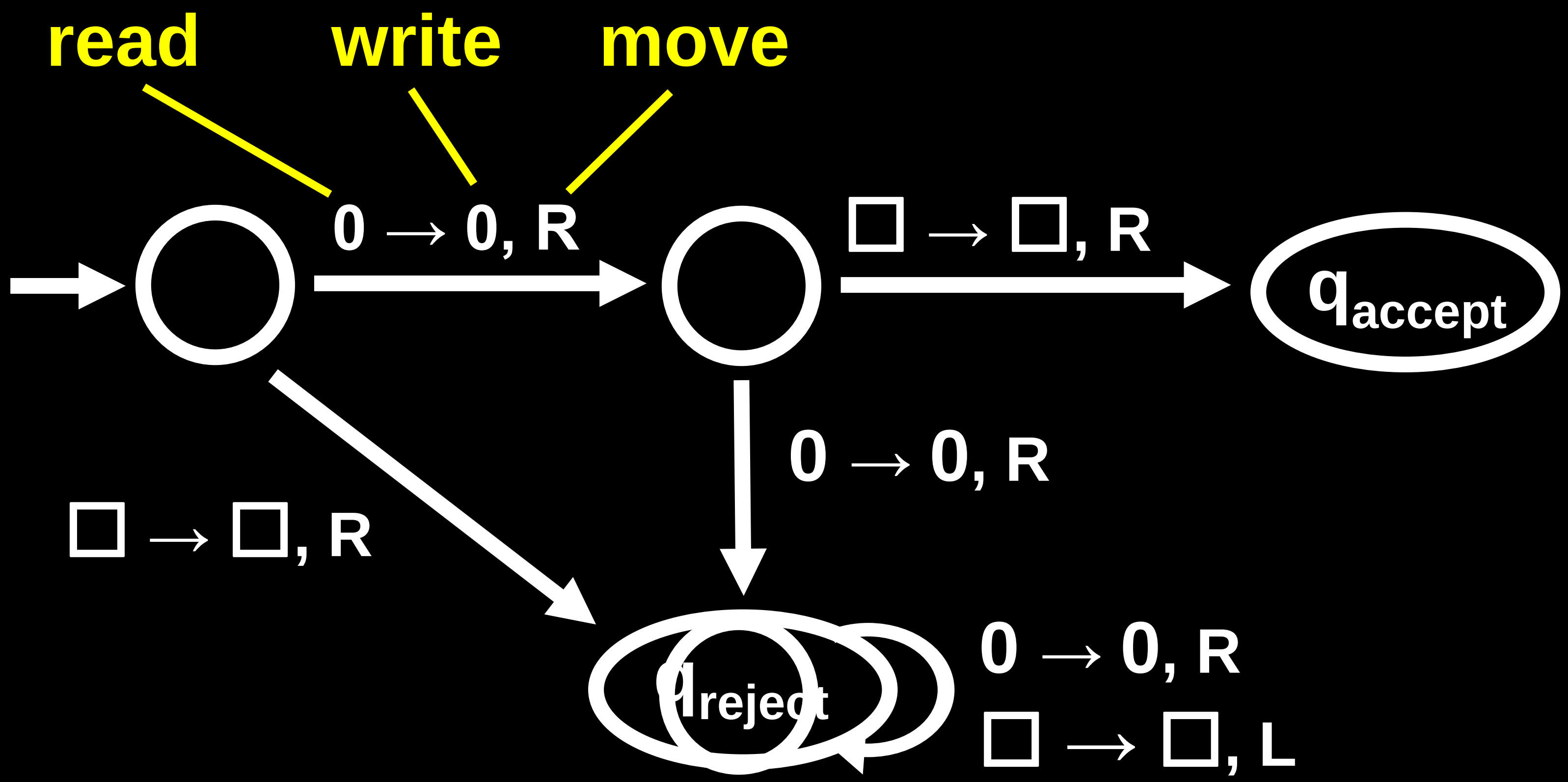
Unsolvable Problems: $\{ w\#w \mid w \in \Sigma^* \}$

```
def is_duplicate(s):
    n = len(s)/2
    if (s[n]!='#'): return False
    return s[:n] == s[n+1:]
```

TURING MACHINE (TM)



UNBOUNDED (on the right) TAPE



A TM can loop forever

TM versus PDA

TM can both write to and read from the tape

The head can move left and right

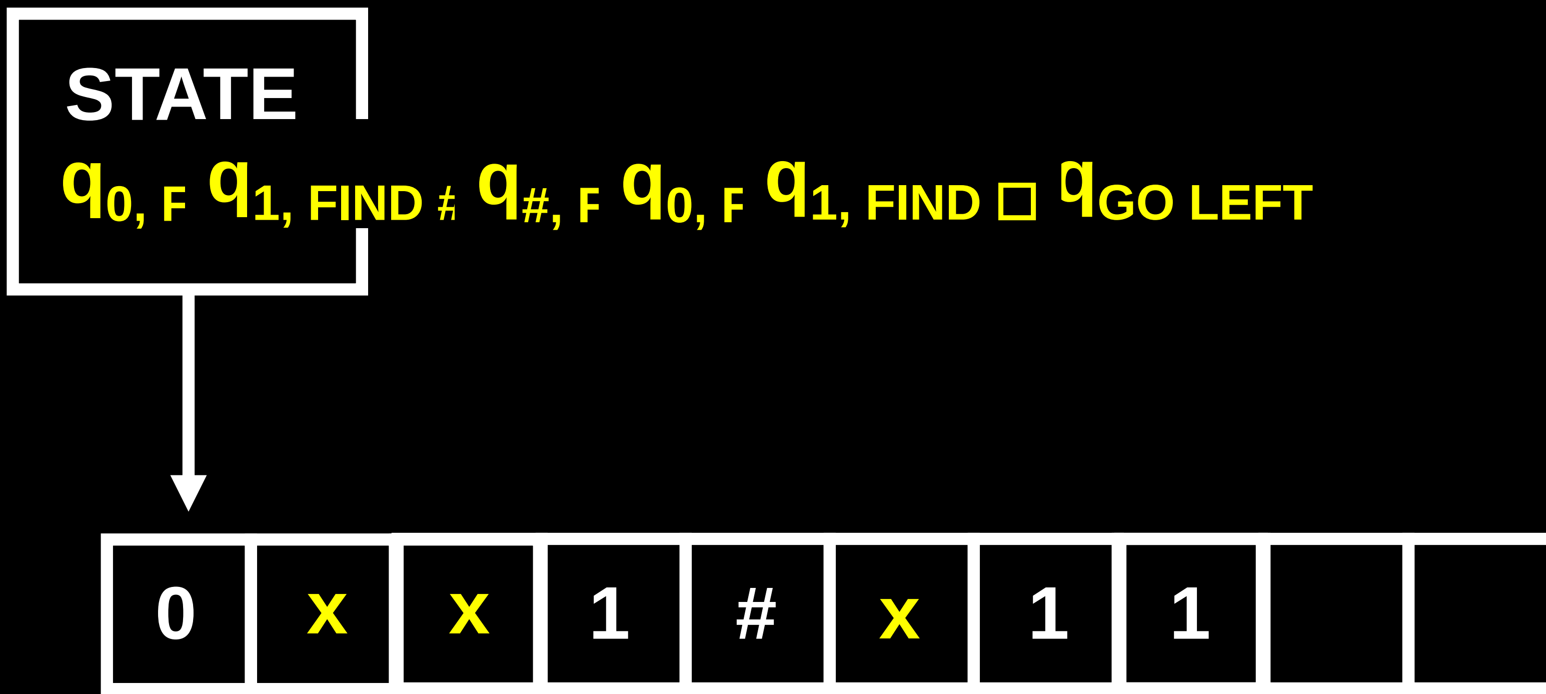
The input does not have to be read entirely

Accept and Reject take immediate effect

Infinite tape on the right, stick on the left

TM is deterministic (will consider NTMs later)

Testing membership in B = { w#w | w ∈ {0,1}* }



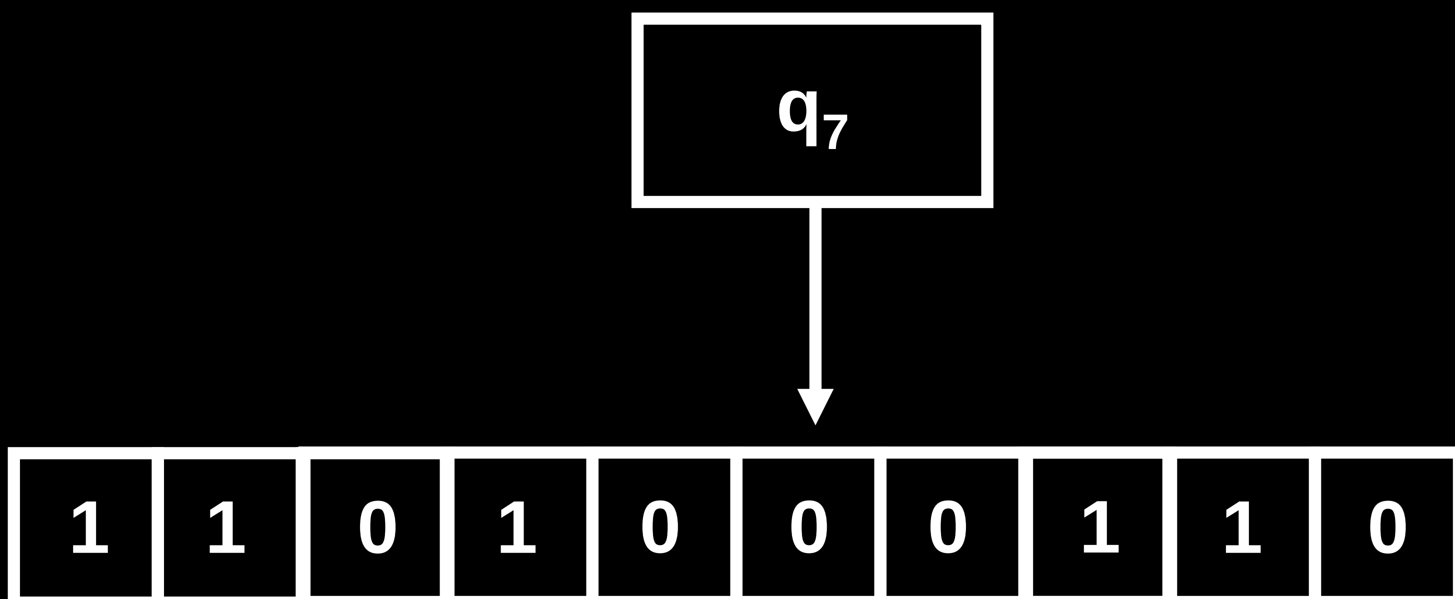
Definition of a TM

A **TM** is a 7-tuple $T = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$:

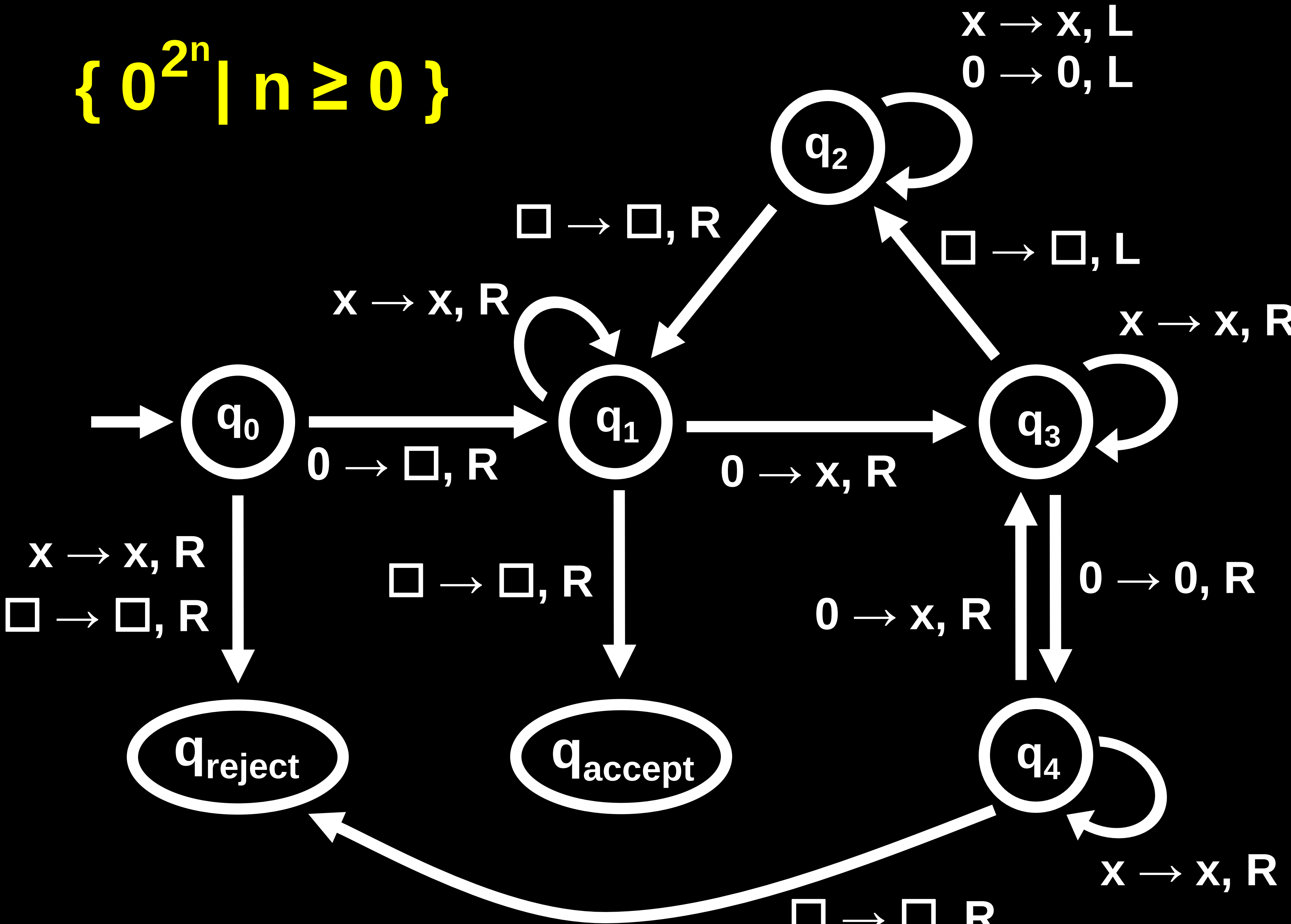
- Q is a finite set of states
- Σ is the input alphabet, where $\square \notin \Sigma$
- Γ is the tape alphabet, where $\square \in \Gamma$ and $\Sigma \subseteq \Gamma$
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$
- $q_0 \in Q$ is the start state
- $q_{accept} \in Q$ is the accept state
- $q_{reject} \in Q$ is the reject state, and $q_{reject} \neq q_{accept}$

CONFIGURATIONS

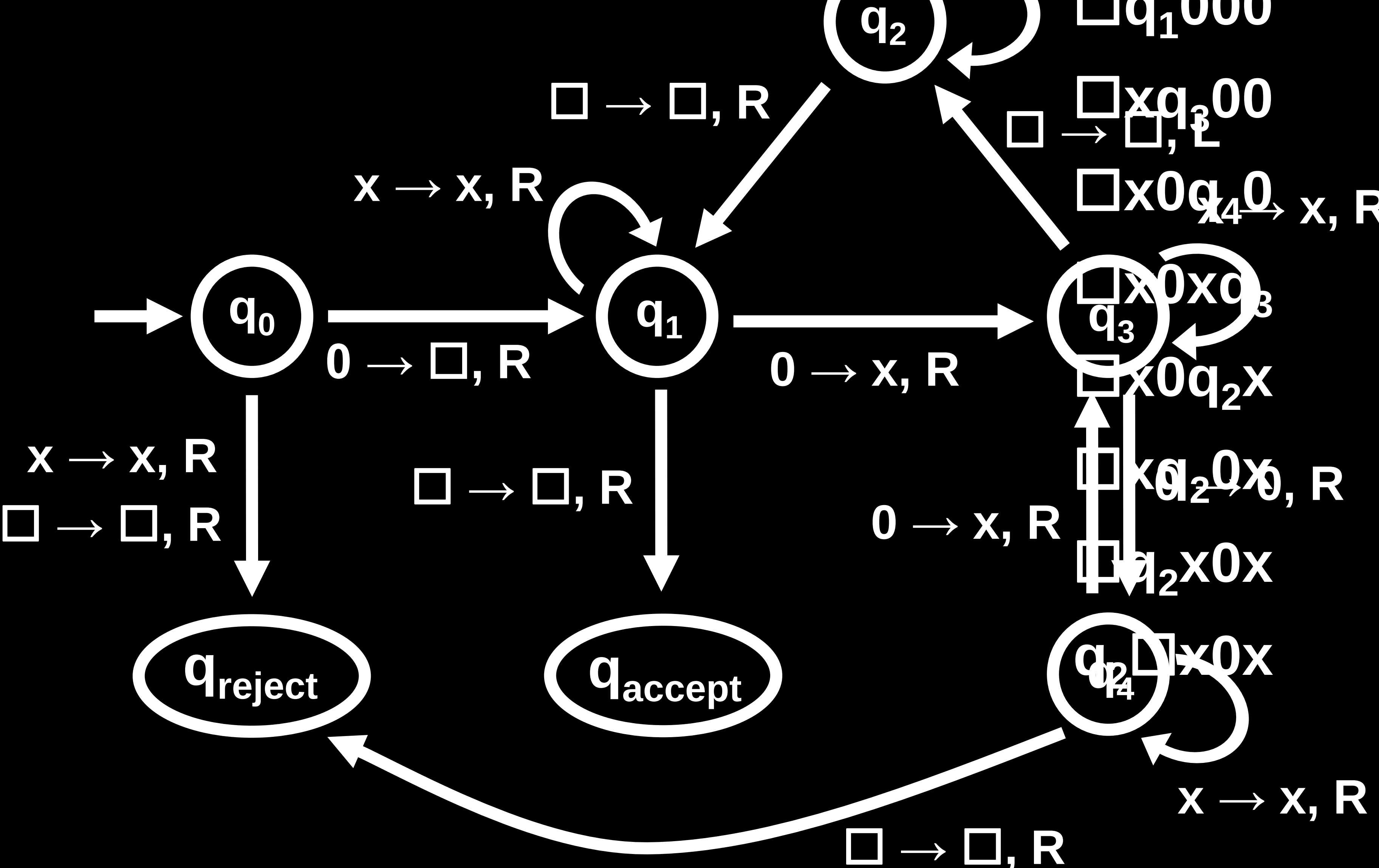
$$11010q_7 00110$$



$\{ 0^{2^n} \mid n \geq 0 \}$



$\{ 0^{2^n} \mid n \geq 0 \}$



Accepting and rejecting

A TM on input sting w may

- **either halt (enter q_{accept} or q_{reject})**
- **or never halt (loop)**

A TM is a decider if it halts on every input.