Turing Machines

final part of the course.

COMPSCI 3331

Turing Machines: Outline

- Motivation.
- Formal Definitions.
- Examples.

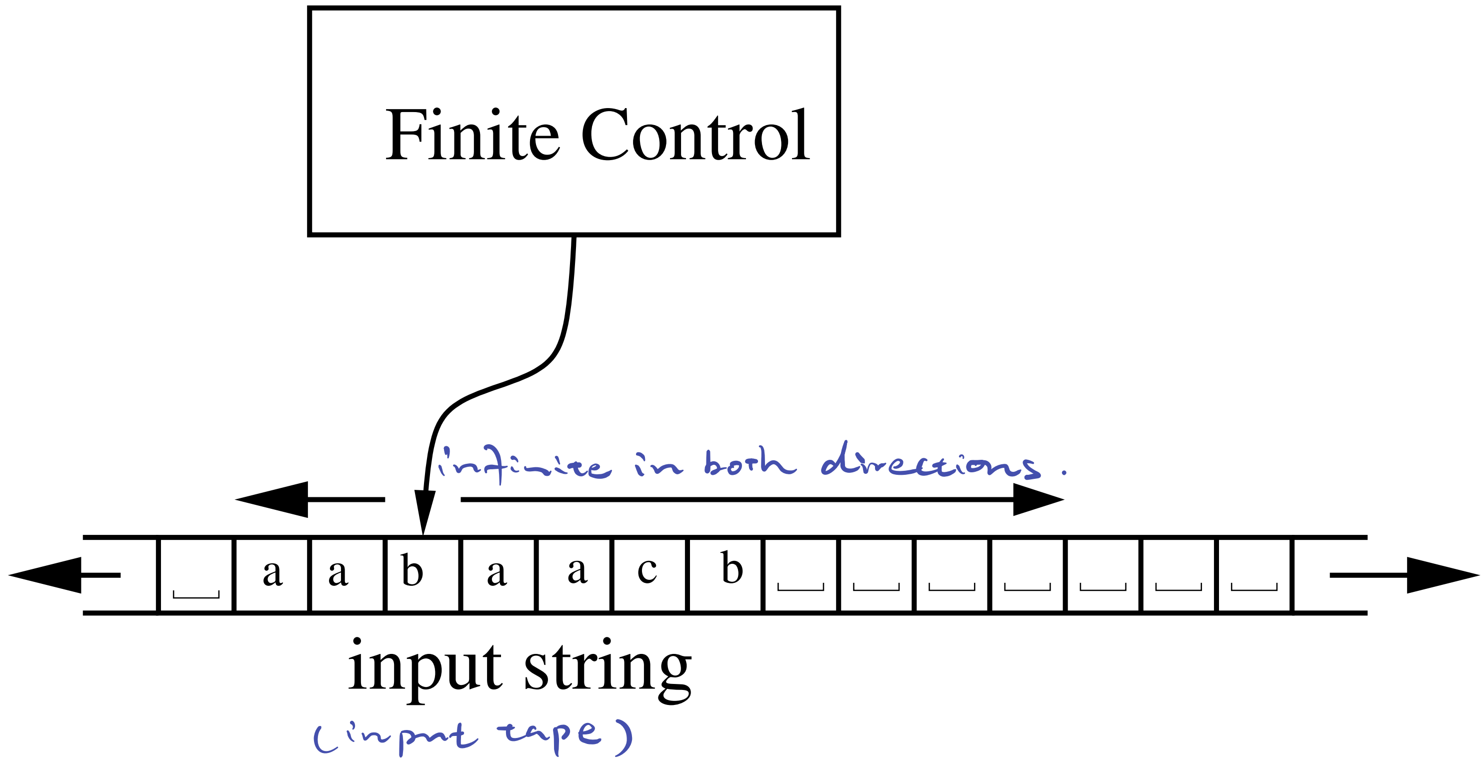
Turing Machines

- Both regular languages and CFLs can't define some languages.
- Turing machines (TMs): a formal model capable of accepting more languages.
- TMs represent our notion of **what is computable**.

Turing Machines

- Basic concept is the same: finite control, input is read sequentially (from a "tape").
- However, now the **input tape is read/write**.
  - For DFAs, NFAs, PDAs, the input tape was read-only.
- A TM can move either way on the input tape.
  - For DFAs, NFAs, PDAs, could only move to right (or stay in the same place).

Turing Machines



Alan Turing (1912–1954)

"[Any person] provided with paper, pencil, and [eraser], and subject to strict discipline, is in effect **a universal Turing Machine.**" (1948)

Turing Machines

A Turing Machine is a seven-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ where

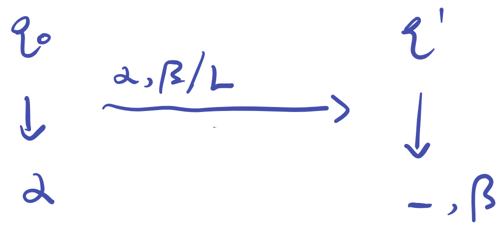
- Q is the finite set of states, set of symbols we could write at the tape at any point.
- Σ, Γ are the input and tape alphabets ($\Sigma \subseteq \Gamma$), Γ gets all input symbols.
- $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$ is the transition function .
- $q_0 \in Q$ is the start state; $F \subseteq Q$ is the set of final states.
- $B \in \Gamma - \Sigma$ is the blank symbol.

not in the input

Transition Function

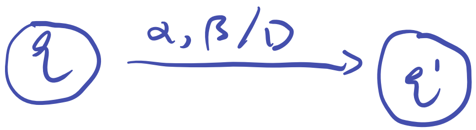
$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$$

- $\delta(q, \alpha) = (q', \beta, D)$.
- If we are in state q and currently see tape symbol $\alpha \in \Gamma$ on the tape, we
  - (a) go to state $q' \in Q$.
  - (b) rewrite α by β in the current cell of the tape.
  - (c) move the input head in direction D on the tape: L (left), R (right) or S (stationary).



Representing TMs

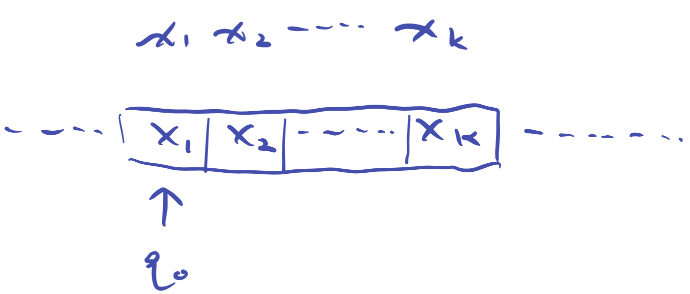
We can represent the transition $\delta(q, \alpha) = (q', \beta, D)$ as an arc:



Computation of a Turing Machine

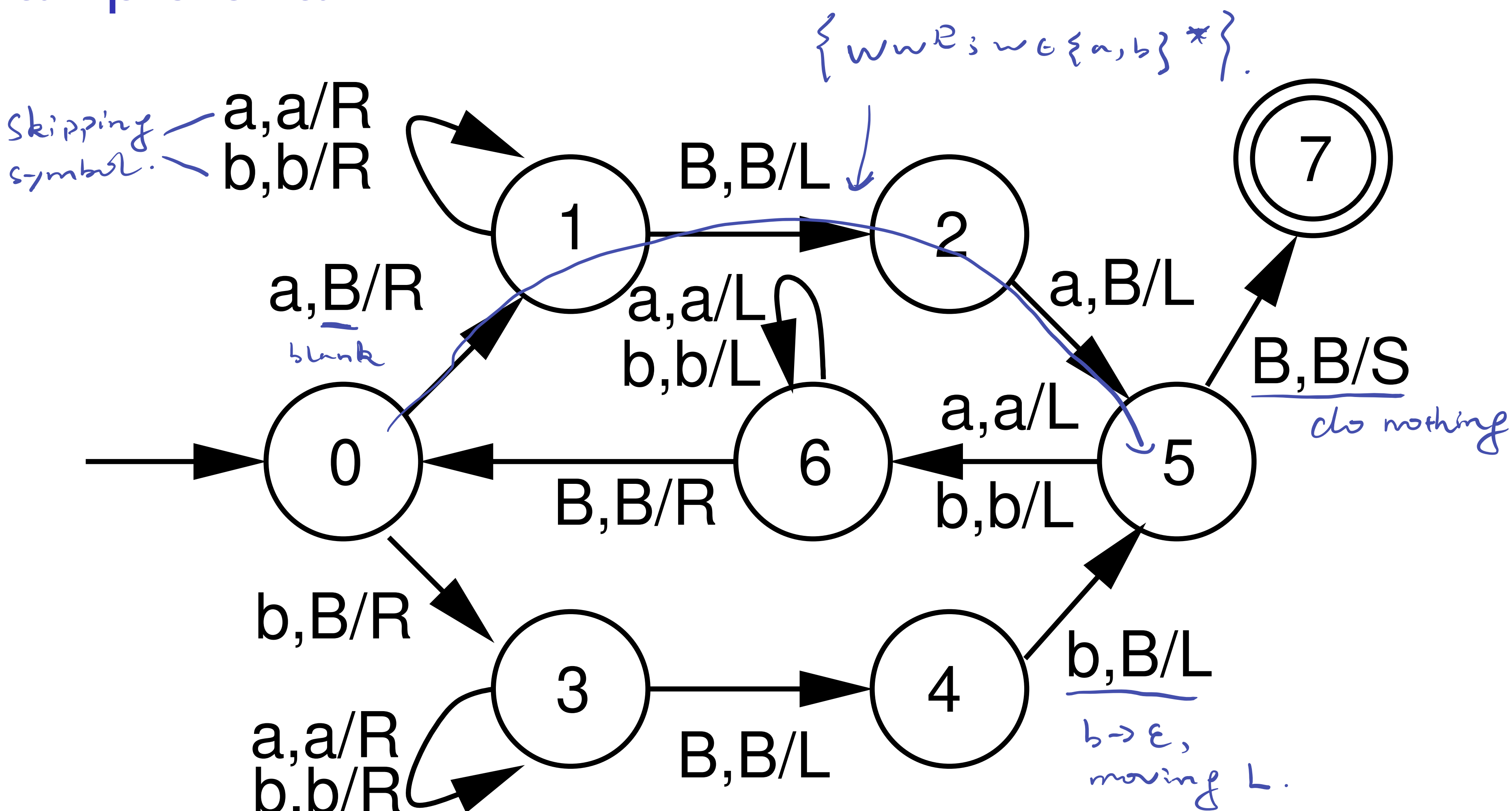
How does a TM compute?

- The input word x is initially written on the tape, and we start in state q_0. We point at the left-most symbol of x.
- Based on the current symbol on the tape and the current state, we make the move based on the transition function.
- We keep making moves as long as possible.
- We **can** move off the region occupied by x on the tape (these cells contain the blank symbol by default).
- If the TM enters an final state, the word is accepted.
- Otherwise, the string is not accepted.



In turing machine, we don't even need to move to the end of the word. The word is accepted as long as we could reach the final state.

Example of a TM



Instantaneous Description of a TM

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ be a TM. An instantaneous description (ID) of M is a word from $\Gamma^* Q \Gamma^*$.
Let $x_1 x_2 \cdots x_{i-1} q x_i x_{i+1} \cdots x_n \in \Gamma^* Q \Gamma^*$. This means that

- The non-blank symbols on the input tape from left-to-right are $x_1 x_2 x_3 \cdots x_n$.
  - (Symbols may be a blank if $i = 1$ or $i = n$.)
- The TM M's head is currently pointing at x_i.
- The TM M is currently in state q.

$$\begin{array}{c} q \\ \downarrow \\ \cdots B\,B\,B \;\; x_1 \; x_2 \cdots x_i \cdots x_n \; B\,B\,B \cdots \end{array}$$

Moves of a TM

1) update what's on tape
2) move to the direction given.

1) Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ be a TM. We denote by $\vdash_M$ the relation between IDs given by the transition function. $\delta(q, x_i) = (q', \beta, L)$. Then we have the following cases:

- If $i > 1$, then

Like "=>"

the rewritten one.

$$x_1x_2\cdots x_{i-1}qx_ix_{i+1}\cdots x_n \vdash_M x_1x_2\cdots x_{i-2}q'x_{i-1}\beta x_{i+1}\cdots x_n.$$

the original one.

pointing here

- If $i = 1$, then

$$qx_1x_2\cdots x_n \vdash_M qB\beta x_2\cdots x_n.$$

Moves of a TM

2) If $\delta(q, x_i) = (q', \beta, R)$, we have two cases:

- If $i < n$, then

$$x_1 x_2 \cdots x_{i-1} q x_i x_{i+1} \cdots x_n \vdash_M x_1 x_2 \cdots x_{i-1} \beta q' x_{i+1} \cdots x_n.$$

- If $i = n$, then

$$x_1 x_2 \cdots q x_n \vdash_M x_1 x_2 \cdots \beta \underbrace{qB}_{\text{right end}}.$$

Moves of a TM

- If $\delta(q, x_i) = (q', \beta, S)$,

$$x_1 x_2 \cdots x_{i-1} q x_i x_{i+1} \cdots x_n \vdash_M x_1 x_2 \cdots x_{i-1} q' \beta x_{i+1} \cdots x_n.$$

We denote by $\vdash_M^*$ the fact that two IDs are related by zero or more applications of $\vdash$.

Language Acceptance

The language **accepted** by a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ is defined as follows:

$$L(M) = \{w \in \Sigma^* \ : \ \exists x_1, x_2 \in \Gamma^*, q_f \in F, q_0 w \vdash_M^* x_1 q_f x_2\}.$$

Examples:

- $L = \{a^n b^{n^2} \ : \ n \geq 0\}$.
- $L = \{a^n b^n c^n \ : \ n \geq 0\}$.

  not CFL / CFG.

aaaabbbbcccc
=> AaaaBbbbcccc
=> AAAABBBBcccc finish matching a and b
=> AAAAbBBBCccc
=> AAAAbbbbCCCC finish matching b and c.

Halting and Crashing

- We say that a TM **halts** if it enters a state and has no next move. it is either accepted nor rejected.
- Informally, we say that a TM **crashes** if it enters a state that is 1) not final and then 2) has no next move (i.e., halts and rejects).
- For any TM, we can assume that when it enters a final state, it halts.
- That is, for every final state $q_f \in F$, $\delta(q_f, \alpha)$ is undefined for all $\alpha \in \Gamma$.

Some questions...

- What kinds of languages can TMs accept?
- What kinds of languages can't be accepted by a TM?
- Can every CFL be accepted by a TM?
- What about nondeterminism for TMs?