

# COL352 Lecture 23

## Contents

1	Recap	1
2	Definitions	1
3	Content	1

## 1 Recap

Completed PDAs and grammars last time.

## 2 Definitions

### Definition 1

A Deterministic Turing Machine (DTM) is a 8-tuple of the form  $(Q, \Sigma, \Gamma, \delta, q_0, \sqcup, q_{acc}, q_{rej})$  where

1.  $Q$  is a finite set of states
2.  $\Sigma$  is a finite alphabet
3.  $\Gamma \supsetneq \Sigma$  is the tape alphabet
4.  $q_0 \in Q$  is the initial state
5.  $\sqcup \in \Gamma \setminus \Sigma$  is a blank symbol (tape initially contains input  $x \in \Sigma^*$  followed by infinitely many  $\sqcup$ ).
6.  $q_{acc} \in Q$  is the accepting state,  $q_{rej} \in Q$  is the rejecting state, and  $q_{acc} \neq q_{rej}$
7.  $\delta : (Q \setminus \{q_{acc}, q_{rej}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ . Here the first input is the current state, second input is current tape content, the first component of the output is the new state, second is new tape content, and third is the direction in which we need to move.

If the machine is reading the leftmost tape cell and it makes a leftward move, the location of the head doesn't change.

## 3 Content

Turing machines – comparison with finite automaton:

1. Infinite tape
2. Can write on the tape
3. Can move to the left
4.  $q_{acc}, q_{rej}$ : no outgoing transitions and halting even if string is not read completely.

Recall that  $L = \{a^n b^n c^n \mid n \in \mathbb{N} \cup \{0\}\}$  is not context-free.

### Claim 0.1

This language can be recognized by a Turing Machine.

*Proof.* We have  $\Sigma = \{a, b, c\}, \Gamma = \{a, b, c, \sqcup, X, \triangleright\}$ .

High level idea:

1. Insert the left end mark  $\triangleright$  before the input and simultaneously check whether the input is in  $\mathcal{L}(a^*b^*c^*)$ . If not, reject (one left-to-right pass).
2. Bring the head back to the left end
3. In a left to right pass, overwrite one (the leftmost)  $a$  by a  $X$ , leftmost  $b$  by a  $X$  and left most  $c$  by a  $X$ . If no  $a, b, c$  found, then accept. If one of  $\{a, b, c\}$  found and one of  $\{a, b, c\}$  not found, reject.
4. Go to step 2.

Inserting  $\triangleright$  into the input - make states for all characters to memorize previous character, and replace the current character with the character of the current state and go to next char and so on, and end when you get to  $S_{\sqcup}$ .

	$a$	$b$	$c$	$\sqcup$	$\Delta$	$X$
$S_{\Delta}$	$S_a, \triangleright, R$	reject	reject	$r, \triangleright, R$	reject	reject
$S_a$	$S_a, a, R$	$S_b, a, R$				
$S_b$		$S_b, b, R$	$S_c, b, R$			
$S_c$			$S_c, c, R$	$r, c, R$		
$r$	$r, a, L$	$r, b, L$	$r, c, L$	$r, \sqcup, L$	$p_a, \triangleright, R$	$r, X, L$
$p_a$	$p_b, X, R$			accept		$p_a, X, R$
$p_b$	$p_b, X, R$	$p_c, X, R$				$p_b, X, R$
$p_c$		$p_c, b, R$	$r, X, R$			$p_c, X, R$

The  $S$  stuff achieves step 1, the  $r$  stuff – step 2 and the remaining step 3. Empty cell implies reject.

1.  $p_a$  : wait for an  $a$ , replace it with a  $X$ .
2.  $p_b$  : wait for a  $b$ , replace with a  $X$ .
3.  $p_c$  : wait for a  $c$ , replace with a  $X$ .

Exercise: simulate this Turing machine.  
Observe that this runs in  $O(n^2)$  time.

□

Review question: Formally define/describe a TM which recognizes  $\{ww \mid w \in \{0, 1\}^*\}$ . What about  $\{www \mid w \in \{0, 1\}^*\}$ .