COL352 Lecture 25

Contents

1 Recap 1

2 Definitions 1

3 Content 2

3.1 2-tape DTM 2

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. Σ 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. $\delta : (Q \setminus \{q_{acc}, q_{rej}\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}$. 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.

Definition 2

A DTM M accepts $x \in \Gamma^*$ if $\exists x_1, x_2 \in \Gamma^* : (\epsilon, q_0, x) \vdash_M^* (x_1, q_{acc}, x_2)$.

Definition 3

$L \in \Sigma^*$ is said to be recognised by M if $\forall x \in \Sigma^* : x \in L \iff M$ accepts x.

Definition 4

L is said to be Turing-recognisable if it is recognised by some DTM.

Definition 5

A Turing machine M is called a decider if it halts on every input $x \in \Sigma^*$, i.e.,

$$\forall x \in \Sigma^* : \exists x_1, x_2 \in \Gamma^* : (\epsilon, q_0, x) \vdash^* (x_1, q_{acc}, x_2) \text{ or } (\epsilon, q_0, x) \vdash^* (x_1, q_{rej}, x_2)$$

A language $L \in \Sigma^*$ is said to be decided by M if M is a decider and L is recognised by M.

Definition 6

$L \in \Sigma^*$ is said to be <u>decidable</u> if it is decided by some DTM M.

Definition 7

A $2-$tape DTM is $(Q, \Sigma, \Gamma, \delta, q_0, \sqcup, q_{acc}, q_{rej})$, where $Q, \Sigma, \Gamma, q_0, \sqcup, q_{acc}, q_{rej}$ are as before, and $\delta : (Q \setminus \{q_{acc}, Q_{rej}\}) \times \Gamma^2 \to Q \times \Gamma^2 times\{L, R, S\}^2$.

For now, S is used since moving both tape heads preserves parity of sum of locations, except for the borders. The language recognised or decided by 2-tape DTM is defined analogously.

Definition 8

Non-Deterministic Turing Machines

A nondeterministic Turing Machine is an 8-tuple $(Q, \Sigma, \Gamma, \Delta, q_0, \sqcup, q_{acc}, q_{rej})$, where $Q, \Sigma, \Gamma, q_0, \sqcup, q_{acc}, q_{rej}$ are the same as in the definition of DTMs.

Δ is a relation on $(Q \setminus \{q_{acc}, q_{rej}\} \times \Gamma) \times (Q \times \Gamma \times \{L, R\}$.

Definition 9

A nondeterministic Turing Machine N accepts $x \in \Sigma^*$ if there <u>exists</u> an accepting run of N on x.

Language recognized by N is the set of all x accepted by N.

Definition 10

An NTM is called a <u>nondeterministic decider</u> if $\forall x \in \Sigma^*$ <u>every</u> run of N on x halts.

A language decided by a nondetermistic decider N : set of all x accepted by N.

3 Content

Question 1

What is the set of instantaneous descriptions of a DTM?

Answer. Set of instantaneous description $= \Gamma^* \times Q \times \Gamma^*$, and a state (x_1, q, x_2) is the following: x_1 is the part of the tape that is strictly to the left of the current head, and x_2 is the remaining part (leaving out the trailing blanks).

Exercise: Formally define the "changes to" relation $\vdash_M$ on the set of instantaneous descriptions.

$\vdash_M^*$ is the reflexive transitive closure of $\vdash_M$.

Claim 0.1

$\forall L \in \Sigma^*$, L is decidable $\implies$ L is Turing-recognisable.

Proof. Definition. $\square$

Note that we will see later on that the opposite direction is not true.

3.1 2-tape DTM

We now have two tapes.
Observe that

Theorem 1

L recognised by a DTM $\iff$ L recognised by a 2-tape DTM.

L decided by a DTM $\implies$ L decided by a 2-tape DTM.

Proof. The forward implication is obvious (just don't use the second tape).

Consider the instantaneous description of the 2-tape DTM – (x_1, y_1) and (x_2, y_2) being the corresponding strings before and not before the tape head on both tapes, and let $\Box$ be the blank symbol of the 2-tape DTM.

Our single tape DTM would have the following: $\triangleright x_1 \to y_1\Box\ldots\Box\#x_2 \to y_2\Box\ldots\Box\sqcup\ldots$

Call the 2-tape one M_2 and the 1-tape one M_1.

$M_1 = (Q, \Sigma, \Gamma, \delta, q_0, \Box, q_{acc}, q_{rej})$

$M_2 = (Q', \Sigma, \Gamma \cup \{\triangleright, \to, \#, \sqcup\}, q_0', \sqcup, q_{acc}', q_{rej}')$

Behaviour of M_1

1. Preprocess phase

   To establish the invariant, we need to change $x\sqcup\ldots$ to $\triangleright \to x\# \to \sqcup\ldots$

2. Simulation phase

   We shall store (state of M_2, symbol from Γ, symbol from Γ, $L/R/S$, $L/R/S$).

   For each transition of M_2, M_1 makes three passes over the tape.

   (a) Insert $\Box$ before the $\#$ and overwrite the leftmost $\sqcup$ by $\Box$.
   (b) Read the symbols after the two $\to$s and save them in the state.
   (c) Execute the transitions of M_2 in M_1 (doesn't need a complete pass of its own).
   (d) Implement the transitions on the tape: overwrite the cells next to the $\to$s and move the $\to$s.

   For instance, if $\delta(q, a, b) = (c, d, R, L)$, then the state changes from $(q, a, b, \_, \_, \ldots)$ to $(q', c, d, R, L, \ldots)$, where the dots are for more information maintained in other passes, and also stores pass number.

$\Box$

Question 2

Suppose M_2 runs in time $T(n)$ on inputs of length n. What is the running time of M_1?

Answer. $O(T(n)^2)$, since the length of the content on the tape of M_1 after simulating t steps of M_2 is $|x| + 2t + c$ where c is a constant, so the simulation of the $(t+1)$-th step of M_2 of M_2 takes time $\approx 3(|x| + 2t + c)$, and if we sum it from $t = 1$ to $T(n)^2$.

Note 1

It might be much easier to exhibit a 2-tape Turing machine instead of a 1-tape Turing machine, and more efficient too.

Example 1

Set of palindromes over Σ: on a 1-tape DTM, the obvious algorithm takes time $\Omega(n^2)$ in the worst case – pass from char 1 to char n to char 2 and so on.

On a 2-tape DTM, copy the reverse of x on tape 2, and then compare – this is $O(n)$.

Exercise: Prove that the classes of Turing-recognisable and decidable languages are closed under union and intersection.

Answer. Just run in parallel on both tapes after copying to second tape – acceptance criteria – and/or.

Exercise: Prove that DTMs with 2-way ∞ tape can be simulated by 2-tape DTMs.

Question 3

Can we simulate an NTM by a 2-tape DTM?

Answer. Yes, we can.

Theorem 2

If L is recognised (respectively decided) by an NTM, then L is recognised (resp. decided) by a 2-tape DTM, and therefore also by a DTM.

Proof. Recognisability:

Label transitions in Δ as $T_0, T_1, \ldots, T_{m-1}$, where $m = \Delta$.

Idea: iterate over all possible strings ρ in $\{0, \ldots, m-1\}^*$, and hold ρ on tape 2.

Simulate the run given by ρ on tape 1.

1. If it terminates and accepts x, then accept.
2. If for some i, the i^{th} transition in ρ is not applicable, then break the simulation.
3. If you run out of transitions, i.e., ρ is too small, then break the simulation.

N accepts x iff there exists an accepting run of N on x. Let ρ^* be the sequence of transitions taken in the accepting run. When tape 2 contains ρ^*, then the simulation accepts x.

Conversely, if the simulation accepts, look at the content of tape 2. That must be an accepting run of N on x.

Implementation details:

1. Store $x\#\rho$ on tape 2 instead of just ρ (for each ρ, erase tape 1, copy x on tape 2, then start the simulation).
2. Observe that given any ρ, the next ρ can be computed by a single tape DTM.

$\square$