

CS 381 – HW6 SOLUTIONS

2. Construct Turing machines that compute the following languages:

(a) $\{a^{2^n} : n \in \mathbb{N}\}$

Solution a:

A Turing Machine that computes $\{a^{2^n} : n \in \mathbb{N}\}$ starts by scanning the input tape left to right to verify that all input symbols are a's. Should it encounter any character aside for 'a' or the blank symbol, the TM rejects. Upon reaching the first blank symbol, the TM replaces it with an end marker.

The TM scans left. If there is only a single 'a' on the track, the TM accepts. Otherwise, it proceeds to count the number of a's mod 2 to make sure the string is of even length, rejecting if not. The TM then proceeds to divide the string in half (see Kozen Ex 29.1 if unsure how) and erases the right half, finishing by writing an end marker at the first blank scanning from left to right. The TM repeats, starting at

Comments:

This was a fairly simple Turing Machine to construct. Make sure you know that 'counting' is not possible. However, counting mod 2 is possible. It involves two states (odd/even) whereby the TM upon reading in each new 'a', moves from one to the other.

(c) $\{w \in \{0,1\}^* : |w| \text{ is even and there exists } i \leq |w|/2 \text{ such that for all } j < i, a_j = a_{|w|/2+j} \text{ and } a_i = 1 \text{ and } a_{|w|/2+j} \text{ (where } a_i \text{ is the } i^{\text{th}} \text{ bit of } w)\}$.

Solution :

We can construct a total TM M for this language as follows:

- 1). Check if $|w|$ is even. On input w , M scans out to the first blank symbol, counting the number of symbols mod 2 to make sure w is of even length and rejecting immediately if not. It lays down a right endmarker \neg .
- 2). Mark two halves. Repeatedly scans back and forth over the input. In each pass from right to left, it marks the first unmarked 0 or 1 it sees with R . In each pass from left to right, it marks the first unmarked 0 or 1 it sees with L . It continues this until all symbols are marked.
- 3). Compare. Repeatedly scans back and forth over the marked string. In each pass from left to right, it erases the first symbol marked with L , let x be this symbol, remember x in the finite control. Keep going right until it sees the first symbol marked with R , let y be this symbol. Then compare x with y . If $x=y$ (Don't take the different marks into consideration), go back to the left and repeat this process. if $x=1, y=0$, accept. if $x=0, y=1$, reject. If we erase all the symbols, reject