Giorgi Japaridze

Theory of Computability

Measuring complexity

Section 7.1

Measuring complexity

Definition 7.1 Let **M** be a deterministic TM that halts for every input.

The ***running time*** or ***time complexity*** of **M** is the function **f: N → N**, where **f(n)** is the maximum number of steps that **M** uses on any input of length **n**.

If **f(n)** is the time complexity of **M**, we say that **M** ***runs*** in time **f(n)**, or that **M** is an ***f(n) time*** machine.

Customarily we use **n** to represent the length of the input.

If the time complexity of **M** is $f(n) = n^2+2$, at most how many steps would **M** take to accept or reject the following strings?

□

0

10

01001

What are the time complexities f(n) of:

1. The fastest machine that decides **{w | w starts with a 0}**?

2. The fastest machine that decides **{w | w ends with a 0}**?

3. The following machine **M1**, deciding the language **$\{0^k1^k \mid k \geq 0\}$**:

M1 = "On input string **w**:

1. Scan across the tape and *reject* if a **0** is found to the right of a **1**.
2. Repeat if both **0**s and **1**s remain on the tape:
3. Scan across the tape, crossing off a single **0** and a single **1**.
4. If **0**s still remain after all the **1**s have been crossed off, or if **1**s still remain after all the **0**s have been crossed off, *reject*.
   Otherwise, if neither **0**s nor **1**s remain on the tape, *accept*."

Here we are lazy to try to figure out the exact values of the constants **a**, **b**, **c** (though, apparently **b=1**). But are those constants really all that important once we know that n^2 is involved?