Miguel Alcón Doganoc
Computational Complexity
March 27, 2019

Homework 1

1 Exercise: Verifiers in L

We want to show that NP is characterized as the class of languages A for which there exists a polynomial $p(n)$ and a language B in **L** such that for every string x we have

$$x \in A \Leftrightarrow \exists y \in \{0,1\}^* \text{ such that } |y| \leq p(|x|) \text{ and} \langle x, y\rangle \in B.$$

First, we add more information to the certificate y. Specifically, we add the encoding of the verifier of x, which is the deterministic TM V. We redefine y as the concatenation of the encoding of the certificate (y_c) and the encoding of the verifier (y_v). Because $y \in \{0,1\}^*$, we have to add something to distinguish when y_c finishes and y_v starts, so $y = 1^{|y_c|}\ 0\ y_c\ y_v$.

Then, we can compute $V(x, y_c)$ using a UTM U such that $U(\langle V, \langle x, y_c\rangle\rangle) = V(x, y_c)$ with the encoding y_v.

Finally we should check that $U(\langle V, \langle x, y_c\rangle\rangle)$ computes the verification of x in polynomial time and logarithmic space, but we did not manage to achieve it.

2 Exercise: NP = NL?

We do not know how to prove it. Maybe if we would have finished with the exercise 1, we would have a hint to resolve it.

3 Exercise: Comparison in logspace

As it is said in the statement, we know that n is the length of x, and the input tape is as following:

$$\square\ x_1, x_2, ..., x_n \# y_1, y_2, ..., y_n\ \square$$

Starting from this, I describe the TM that outputs 0 if $m < n$ and 1 if $m \geq n$, with the input tape defined above. For a more understandable explanation, I have made a visual representation of both tapes (first input, second work tape) at the end of each general step. Note that the position of the head of each tape is represented as a square surrounding the symbol which is in this position. Let us start with the description.

First, we have to check if $n = 0$. If it happens, we assume that both m and n are 0, so $m = n$.

1. Scan the actual position of the head of the input tape, which is the first position starting from the left. If it is a # symbol, output 1 and halt.

$$\square\ \boxed{x_1}, x_2, ..., x_n \# y_1, y_2, ..., y_n\ \square$$

We copy y into the work tape. Because y is the binary encoding of n, we know that it is $O(\log n)$ in space.

2. Move the input tape to the right until finding the # symbol.

3. Move the input tape to the right until finding a 1.

4. Copy into the work tape from the first 1 (current position) to the $\square$ symbol (end of the input tape), from left to right.

$$\square\, x_1, x_2, ..., x_n \# y_1, y_2, ..., y_n\, \boxed{\square}$$

$$\square\, y_1, y_2, ..., y_n\, \boxed{\square}$$

Before starting to check whether $m < n$ or $m \geq n$, we have to place the head of each tape in the proper position.

5. Move the input tape to the left until finding the # symbol.
6. Move the input and the work tape simultaneously[1] to the left until finding the $\square$ symbol in the work tape.

$$\square\, x_1, x_2, ..., \boxed{x_i}, ..., x_n \# y_1, y_2, ..., y_n\, \square$$

$$\boxed{\square}\, y_1, y_2, ..., y_n\, \square$$

At this point, if we find a 1 between x_1 and x_i, with $i \in \{1, 2, ..., n\}$ or $x_i = \square$, then $m > n$.

7. Move the input tape to the left until finding a 1 or the $\square$ symbol. If it finds a 1, then it outputs 1 and halts.

$$\boxed{\square}\, x_1, x_2, ..., x_i, ..., x_n \# y_1, y_2, ..., y_n\, \square$$

$$\boxed{\square}\, y_1, y_2, ..., y_n\, \square$$

After that, if the TM haven't halted yet, we have to move the heads to the positions like at the end of step 6.

8. Move the input tape to the right until finding the # symbol.
9. Move the working tape to the right until finding the $\square$ symbol.
10. Repeat step 6.

$$\square\, x_1, x_2, ..., \boxed{x_i}, ..., x_n \# y_1, y_2, ..., y_n\, \square$$

$$\boxed{\square}\, y_1, y_2, ..., y_n\, \square$$

Now, we continue checking whether $m < n$ or $m \geq n$, but this time comparing x with y.

11. Scan both tapes and, if the TM does not halt, move the head one position to the right.
    (a) If the value of the header of the input tape is 1, and the one of the work tape is 0, then output 1 and halt.
    (b) Else if the value of the header of the input tape is 0, and the one of the work tape is 1, output 0 and halt.
    (c) Else if the value of both heads are the $\square$ symbol (end of both tapes) output 1 and halt.
    (d) Otherwise repeat step 11.

$$\square\, x_1, x_2, ..., \boxed{x_i}, ..., x_n \# y_1, y_2, ..., y_n\, \square$$

$$\square\, y_1, y_2, ..., \boxed{y_i}, ..., y_n\, \square$$

[1] First move one, then the other.

4 Exercise: Composition of logspace computable functions

Consider the function $g' : \Sigma^* \times \mathbb{N} \to \Sigma \cup \{\#\}$ that, given a string x and an integer $i \in \mathbb{N}$, returns the i-th symbol of $g(x)$, or # if $i > |g(x)|$, where # is a symbol not in Σ.

[*Function g' is computable by deterministic TMs in logarithmic space*]: Because $g(x)$ is computable by deterministic TMs in logarithmic space, then $g'(x, i)$ can compute just 1 element of the output with the same conditions. Using the same TM as $g(x)$, we can define $g'(x, i)$ as:

1. Compute all the outputs of $g(x)$ until reaching the i-th one, without writing on the output tape.
2. Compute the i-th element, output its value and halt.

[*Using g' to compute $f \circ g$*]: We can compute $(f \circ g)(x)$ applying f to x, but using $g'(x, i)$ as an input tape, i.e. as a process over x, launched before the scanning of the desired input, which is $g(x)$. We can map the operations of a conventional tape with the following operations of $g'(x, i)$, with $i \in \mathbb{N}$:

- Moving to the right represents adding 1 to i.
- Moving to the left represents subtracting 1 to i.
- Scanning the symbol of the head represents applying $g'(x, i)$.

Notice that the bounds of this 'input tape' are defined by $i \in \mathbb{N}$, and by $i > |g(x)| \Leftrightarrow$ output #. Furthermore, the starting position of the input tape's head should be $i = 0$.

Because $f(x)$ is computable by deterministic TMs in logarithmic space, and each call to g' re-uses the same space (logarithmic), we can conclude then that $(f \circ g)(x)$ is also computable by deterministic TMs in logarithmic space.