CS 181: Homework 3

Jonathan Woong

804205763

Summer 2017

Discussion 1A

Thursday 17$^{\text{th}}$ August, 2017

Problem 1. For each of the following languages

i) $\mathbf{L}_1 = \{a^p; p \text{ is a prime number}\}$

ii) $\mathbf{L}_2 = \{a^p; p \text{ is a prime number, } m \text{ is a fixed number and } m \geq p \geq 0\}$

iii) $\mathbf{L}_3 = \{a^p b^p; p \text{ is a prime number}\}$

iv) $\mathbf{L}_4 = \{a^p b^p; p \text{ is a prime number, } m \text{ is a fixed number and } m \geq p \geq 0\}$

v) $\mathbf{L}_5 = \{a^p; p$ is a prime number and p is a number of a Turing machine T_p that does not halt given the input $p\}$

find if it is:

a) a regular language

b) a context-free language

c) a recursively enumerable language

If the case (a) is true for the language $\mathbf{L}_i$, build a finite automaton A such that L(A)=$\mathbf{L}_i$.
If the case (b) is true for the language $\mathbf{L}_i$, build a PDA D and formal grammar G such that L(D)=L(G)=$\mathbf{L}_i$.
If the case (c) is true for the language $\mathbf{L}_i$, build a TM T such that L(T)=$\mathbf{L}_i$.
If the case (d) is true for the language $\mathbf{L}_i$, build an ITM M such that L(M)=$\mathbf{L}_i$.

i) $\mathbf{L}_1 = \{a^p; p \text{ is a prime number}\}$ Assume $\mathbf{L}_1$ is context free.

Let $w = a^p$ for some prime number $p \geq k$.

By pumping lemma, we can write $w = uvxyz$. We expect $uv^kxy^kz \in \mathbf{L}_1 \forall k$.

Let $v = a^q$ and $y = a^t$.

Case 1: Let $k = |uxz| = p - q - t$.

Then $|uv^kxy^kz| = k + kq + kt = k(1 + q + t)$.

$\therefore$ Since the above length is divisible by both k and $(1 + q + t)$, it must not be prime.

Case 2: Let $k = 0$.

Then $|uv^kxy^kz| = |v^2y^2| = 2p$.

$\therefore$ Since the above length is divisible by 2, it must not be prime.

Case 3: Let $k = 1$.

Then $|uv^{p+1}xy^{p+1}z| = 1 + (p + 1)q + (p + 1)t = 1 + (p + 1)(q + t) = 1 + (p - 1)(p + 1) = p^2$.

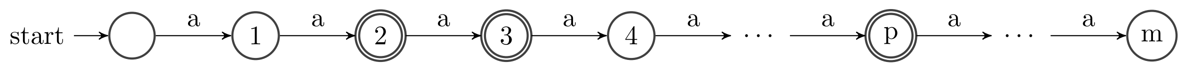
$\therefore$ Since the above length is the square of p, it must not be prime.

$\therefore$ By contradiction, $\mathbf{L}_1$ cannot be a context-free language.

ii) $\mathbf{L}_2 = \{a^p; p \text{ is a prime number, } m \text{ is a fixed number and } m \geq p \geq 0\}$

$\mathbf{L}_2$ is a regular language because we can build an NFA A such that L(A)= $\mathbf{L}_2$.

We construct the NFA with m states, where each prime numbered state is an accepting state, and each non-prime numbered state is not an accepting state. State m is an accepting state if $m = p$.



iii) $\mathbf{L}_3 = \{a^p b^p; p \text{ is a prime number}\}$

Assume $\mathbf{L}_3$ is a context-free language.

Let $w = a^p b^p$.

Since context-free languages are closed under concatenation, consider the language $\mathbf{L} = \{ww^R w\}$.

Assume $\mathbf{L}$ is context-free.

Let $ww^R w = a^p b^{2p} a^{2p} b^p = uvxyz$ where $|vxy| \leq p$, $r = a^p b^{2p}$, $s = a^{2p} b^p$.

The number of $a \in r$ is half the number of $a \in s$.

The number of $b \in r$ is twice the number of $b \in s$.

Case 1: Suppose $v = a \in r$ and $y = a \in s$.

By pumping $|uv^k xy^k z| \forall k$, the number of $a \in r$ will no longer be half the number of $a \in s$, as the quantity of both will be increased by pumping length k.

$\therefore$ This case does not satisfy the pumping lemma.

Case 2: Suppose $v = b \in r$ and $y = b \in s$.

By pumping $|uv^k xy^k z| \forall k$, the number of $b \in r$ will no longer be twice the number of $b \in s$, as the quantity of both will be increased by pumping length k.

$\therefore$ This case does not satisfy the pumping lemma.

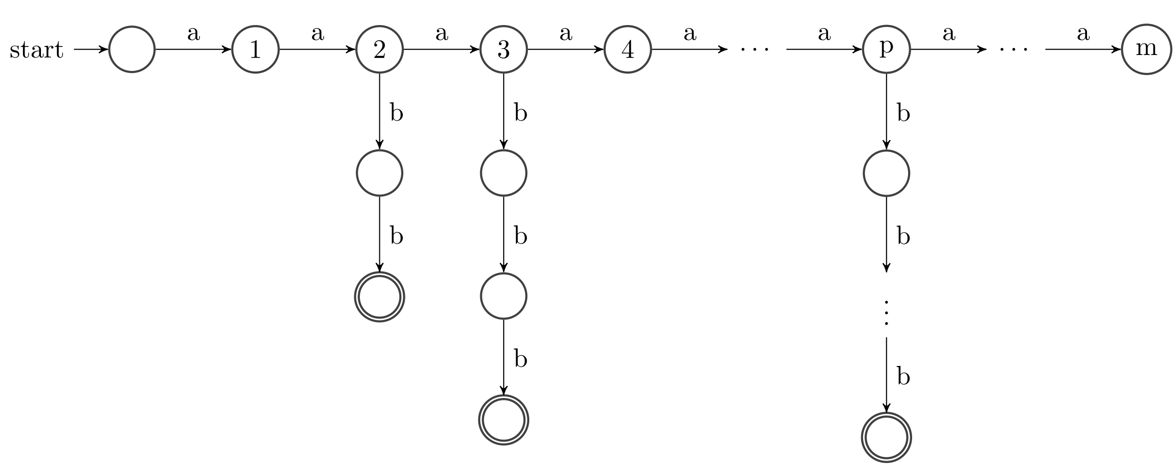
Case 3: Suppose $v = ab$ or $y = ab$.

By pumping $|uv^k xy^k z| \forall k$, the word will obtain alternating sequences of ab that do will never be accepted by $\mathbf{L}_3$. $\therefore$ This case does not satisfy the pumping lemma.

$\therefore$ By contradiction, $\mathbf{L}$ cannot be a context-free language. Since $\mathbf{L}$ is not a context-free language, $\mathbf{L}_3$ cannot be a context-free language.

iv) $\mathbf{L}_4 = \{a^p b^p; p \text{ is a prime number}, m \text{ is a fixed number and } m \geq p \geq 0\}$

$\mathbf{L}_4$ is a regular language because we can build an NFA A such that L(A)=$\mathbf{L}_4$.



v) $\mathbf{L}_5 = \{a^p; p$ is a prime number and p is a number of a Turing machine T_p that does not halt given the input $p\}$ $\mathbf{L}_5$ is a recursively enumerable language because we can build a Turing machine T such that L(T)= $\mathbf{L}_5$.

T has tape A and tape B. Tape A has the string a^p written on it, for which each a resides in one cell.

T rejects a^p if $p = 0$ or $p = 1$.

T begins by writing 2 a's from tape A onto tape B.

With both heads at the leftmost position, both pointing to the first a of each tape, they advance one cell at a time from left to right.

Once tape B reaches a blank cell, the head for tape B returns to the leftmost position, and both heads continue to advance one cell at a time.

If both tape heads reach a blank at the same time, we know that the length of a^p is divisible by the number of a's in tape B. In this case, T adds another a into tape B and repeats the process with both heads starting at the leftmost position.

When both tape heads do not reach a blank at the same time for all lengths of tape B, then the length of a^p must be prime.

vi) Let $T_1, T_2, T_3, \ldots, T_n, \ldots$ be a constructive enumeration of all Turing machines with alphabet {0,1} and one linear tape. For each of the following sets

i) $\mathbf{X}_1 = \{p; p$ is a prime number$\}$

ii) $\mathbf{X}_2 = \{px; p$ is a fixed prime number and x is a number of a Turing machine T_x that halts given the input $x\}$

iii) $\mathbf{X}_3 = \{x; x$ is a number of a Turing machine T_x that does not halt given the input $x\}$

find if it is:

a) decidable/recursive

b) recursively enumerable

c) inductively decidable

i) $\mathbf{X}_1 = \{p; p$ is a prime number$\}$

$\mathbf{X}_1$ is recursively decidable. We can construct a turing machine that implements the Euclidian Algorithm, which correctly identifies prime numbers given the set of all natural numbers.

$\mathbf{X}_1$ is also recursively enumerable, since we can construct a turing machine that enumerates over all prime numbers in the set.

ii) $\mathbf{X}_2 = \{px; p$ is a fixed prime number and x is a number of a Turing machine T_x that halts given the input $x\}$

$\mathbf{X}_2$ is recursively enumerable, since an input word px to a turing machine may not halt when x does not exactly match the number of the turing machine. For example, if the input px is given to the turing machine T_y where $x \neq y$, the turing machine T_y may never halt, making $\mathbf{X}_2$ a recursively enumerable set.

iii) $\mathbf{X}_3 = \{x; x$ is a number of a Turing machine T_x that does not halt given the input $x\}$

$\mathbf{X}_3$ is inductively decidable, since none of T_x halts when given input. We can build a machine, like in class, with an inductive turing machine M. When x is used as input to M, it enters an inner machine which outputs the pair $(x, c(T))$ where $c(T)$ is the code of the turing machine. The pair acts as input into a universal turing machine u. If x is accepted by M, u will output yes. If x is not accepted by M, u will output no. Since none of the T_x halts, u will work indefinitely.