Harvard University Extension School
Computer Science E-121

Problem Set 3

Due October 14, 2016 11:59pm

Problem set by Walter Thornton

Collaboration Statement: I collaborated with one other student, Kunal, and only the course materials.

PROBLEM 1 (2+3 points, suggested length of 1/2 a page)

(A) Prove that $|\mathbb{R}| = |\mathbb{R}\backslash[0, 1)|$.

(B) Let Σ be countably infinite. Show Σ^* is countably infinite. Now let's define Σ^∞ as the set of sequences of countably infinite length over elements from Σ. Show that Σ^∞ is uncountably infinite.

Solution.

(A) $|\mathbb{R}| = |\mathbb{R}\backslash[0, 1)|$
The cardinality of R is uncountably infinite.
The set difference $\mathbb{R}\backslash[0, 1)$ is all those members of $\mathbb{R}$ that do not appear in $[0, 1)$
Both $\mathbb{R}$ and $[0, 1)$ are uncountably infinite.
By Hilbert's Hotel Paradox, $|\mathbb{R}\backslash[0, 1)|$ is uncountably infinite and therefore is equal to the cardinality of $\mathbb{R}$

(B) Σ^* is countably infinite
Σ^* can be written as the union of all Σ^n where $n = 0, 1, 2, 3...$, the set of all strings over Σ with finite length n.
Σ^* is thus the union of a countable number of countable sets, and is therefore countably infinite. (lecture 6, slide 23)

Σ^∞ is uncountably infinite
Since Σ^∞ is defined as the set of sequences of countably infinite length over elements from Σ, it is rightfully the power set of Σ^*.
$\Sigma^\infty = P(\Sigma^*)$
By Cantor's theorem, no set can have a bijection to it's power set and the cardinality of a set is necessarily smaller than that of it's power set. Since Σ^* is countably infinite, $\mathrm{P}(\Sigma^*), or \Sigma^\infty$ is therefore uncountably infinite.

PROBLEM 2 (3+3 points, suggested length of 3/4 page)

$$L_1 = \{a^n b^m c^{n+m} : m, n \geq 0\}$$

$$L_2 = \{a^n b^{2m} a^{2n+1} : m, n \geq 0\}$$

(A) Construct a context-free grammar G_1 for L_1.

(B) Construct a context-free grammar G_2 for L_2.

Please note that there is no need to provide the full 4-tuple, a simple identification of the rules and the start variable will suffice.

Solution.

(A)
S $\rightarrow$ aSc | A
A $\rightarrow$ bAc | ϵ
where S is the start state

(B)
S $\rightarrow$ Da
D $\rightarrow$ aDaa | B
B $\rightarrow$ Bbb | ϵ
where S is the start state

PROBLEM 3 (2+2+2+2 points, suggested length of one page)

Classify the following languages over $\Sigma = \{a, b\}$ as either regular, context-free but not regular, or not context-free, and prove your answers. Proofs may use the regular and context-free pumping lemmas, closure properties, regular expressions, and informal BNF grammars (e.g., $A \rightarrow a \mid aAb$).

(A) $L = \{a^i b a^i : i \geq 0\}$
(B) $L = \{a^{i^3} : i \geq 0\}$
(C) $L = \{a^i y : i \geq 1,\ y \in \Sigma^*$, and y contains the symbol a at least i times$\}$
(D) $L = \{a^* a^i b^i a^i : i \geq 0\}$

Solution.

(A) context-free but not regular

a grammar can be constructed for $L = \{a^i b a^i : i \geq 0\}$ with the following rules:
S $\rightarrow$ aSa | b

therefore it is context free
Assume the language is regular
applying the pumping lemma, we let $w = a^p b a^p$, satisfying $w > p$

Using the form $w = xyz$, we must satisfy three conditions
1. $\mid y \mid \geq 1$
2. $\mid xy \mid \leq p$
3.for all $i \geq 0, xy^i zL$
To satisfy conditions 1. and 2., let $x = a^{p-1}$, $y = a^1$, and $z = ba^p$
when pumped in this manner, the resulting string contains more as before the bs than after, and is not it L. This is a contradiction. Therefore, the language is not regular.

(B) nonregular

This is the unary language containing strings of all a's whose length is a perfect cube.
We assume L is regular. Applying the pumping lemma, we let string $s = a^{p3}$
$\mid xyz \mid = p^3$ and thus $\mid xyz \mid = p^3$ =p$^3 + p$
But $p^3 + p < (p+1)^3$ and further $\mid xy^2z \mid > p^3$ according to condition 2. of the pumping lemma.
Therefore $\mid xy^2z \mid$ lies between two consecutive perfect cubes and hence fails the pumping lemma.
sipser pg 82 example 1.76

(C) context free but not regular

since it is regular it is also context free
a grammar can be constructed with the following rules,
S $\rightarrow$ aBEaE,
B $\rightarrow$ aBEaE $\mid \epsilon$,
E $\rightarrow$ a $\mid$ b $\mid \epsilon$

(D) not context-free

we assume it is context free.
If we let $a^* = \{\epsilon\}$, then $a^i b^i c^i$ is recognized by $L = \{a^* a^i b^i a^i : i \geq 0\}$
we set $s = a^p b^p c^p$
From Sipser pg 128, we know a string in this form arrives at a contradiction in one of two ways.
Either the number of as,bs and cs are unequal and hence not members of the language or,
v or y end up with both as and bs. Thus when pumped abab patterns emerge, and the resulting strings are not members of the language
Therefore it is not context-free

PROBLEM 4 (2+4+4 points, suggested length of 1/2 a page)

The diagram below is an example of a "railroad diagram". It turns out that they are equivalent to context-free grammars in terms of what languages they can express. Each path through the diagram, following arrows and picking up symbols as we pass through boxes or circles, determines a string given by the sequence of symbols passed through. In the example below, main diagram is at the top. The capital letters E and D we encounter are interpreted not as symbols but as references to subdiagrams (the two boxes at the bottom). To interpret what happens in the main diagram when we encounter a box labelled D, we can substitute in the box labelled D at the bottom. You might find it helpful to think of the diagram as like a C program, with a main function (at the top) and then two functions D and E, where E gets called once and D gets called several times.

You should interpret the symbols in the circles not as operators but *as symbols*, so just think of + as a letter in your alphabet. We will consider the following two railroad diagrams equivalent (just as a notational decision).

And we naturally extend the above above notational equivalence for other ranges of integers (e.g. "0-9").
(A) Give an informal description of the language described by the main diagram (at the top).
(B) Construct a context-free grammar for the language described (just the set of rules is fine, provided it is clear).
(C) Construct a regular expression for the language or prove that it is impossible to do so.

Solution.

(A) The railroad diagram gives a string that seems to be formatted as scientific notation for a calculator. Some real number, signed or unsigned, that can be followed by an e and an integer, signed or unsigned. One possible path stands out though, a number, signed or unsigned, followed by a single decimal point with nothing following.

(B)
S $\rightarrow$ +A | -A
A $\rightarrow$ T | B
T $\rightarrow$ D.DE | D.E | D.
B $\rightarrow$ D.DE | .DE | .D
E $\rightarrow$ e + D | e - D
D $\rightarrow$ $\{n \mid n \in \{0, 1, 2, 3...9\}\}$

(C) For this problem we define d $\in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Then the regular expression is
$(+ \cup - \cup \epsilon) \circ ((d^* \cup \epsilon) \circ (d^* \cup \epsilon)) \circ ((e \circ (\epsilon \cup + \cup -) \circ D^*) \cup \epsilon)$

PROBLEM 5 (4+1+1 points, suggested length of 1/2 a page + 1/3 page for challenge)

The language $L = \{ab^k a^k b^k : k \geq 0\} \cup \{a^i b^* a^* b^* : i \neq 1\}$ is *not* context-free. You may assume this without proof for the first two parts of this problem.
(A) Show that L satisfies the context-free pumping lemma.
(B) Explain why this is not a contradiction.
(C) **Challenge!** *Note: On every problem set we will provide a challenge problem, generally significantly more difficult than the other problems in the set, but worth only a few points. It is recommended that if you attempt these problems, you do so only after completing the rest of the assignment.* Prove that L is not context-free.

Solution.

(A) The language $L = \{ab^k a^k b^k : k \geq 0\} \cup \{a^i b^* a^* b^* : i \neq 1\}$ can be written as $\{a^i b^* a^* b^*$: where i$\neq 1, a \neq$and b$\neq \epsilon\}$
A string can be created $a^{p-1}ab^*a^*b^*$ and set to $s = uv^i xy^i z$ where
$u = a^{p-1}, v^i = a, x = b^*, y^i = a^*, z = b^*$. Since | vy |$> 0$ and | vxy |$\leq p$, when v^i and y^i are pumped, the resulting string is a member of the language. Therefore this does not violate the pumping lemma.

(B) The pumping lemma is useful in proving that a language is not context free because satisfying the pumping language is a necessary condition of context-free grammars. The condition is one way. If the pumping lemma were satisfied, we would have no additional information about the language being pumped, as is the case here.

(C)

PROBLEM 6 (8 points, suggested length of 3/4 a page)

Prove that if M is a PDA and there exists a natural number k such that for all w in $L(M)$ the size of the stack is at most k in every configuration of any computation of M on w, then $L(M)$ is regular. (Describe formally how to construct a finite automaton accepting $L(M)$, and then prove your construction correct.)

Solution.
If M is a PDA and there exists a natural number k such that for all w in $L(M)$ the size of the stack is at most k in every configuration of any computation of M on w, then $L(M)$ is regular.
In other words, for any PDA with a bounded stack size, there are k possible states that the machine can be in, given w input. These combinations of states can be used to construct an equivalant NFA, that accepts L(M). Therefore L(M) is regular.

We can construct $M = (Q, q0, \Sigma, \Gamma, \delta, F)$ be a PDA recognising L(M) where there is a finite number of symbols on the stack. We can construct $N = (Q, \Sigma, \delta, q0, F)$ from M so that N recognises L(M).
The states of N can be derived from M where each state of N represents a pair derived from the state of M and the state of it's stack. (q, w) where $q \in Q$ and $w \in$

If $Q_M = \{A, B, C\}$ and $\Gamma_M = \{a, b$ with a stack size of 3, plus a end of stack symbol and an empty stack, there would be 2^5 stack permutations each representing a state in N. Transitions from M would remain except traversing the new states in N upon (q, w).
This process would be akin to the subset construction of an NFA to DFA. Thus we show that any PDA M with a finite stack is equivelant to some finite state machine. Thus the language is regular.

PROBLEM 7 (0 points)

Please estimate the number of hours spent on this assignment to the nearest half hour.

Solution.
13