# CS 181 Homework 3

Sriram Balachandran

TOTAL POINTS

**55 / 65**

QUESTION 1

## 1 Question 1 12 / 20

- **- 0 pts** Correct
- **- 1 pts** confusion on state vs set
- **- 3 pts** missing/incorrect 1-2 transition specifications
- ✓ **- 8 pts correct intuition, unclear construction details**
- **- 1 pts** domain error/confusion
- **- 1 pts** minor mistake
- **- 5 pts** explanation incomplete/not detailed enough
- **- 20 pts** wrong idea, reduced the problem to concatenation
- **- 12 pts** good attempt

QUESTION 2

## Question 2 45 pts

### 2.1 2a 20 / 20

- ✓ **+ 4 pts Not totally incorrect**

Intuition

- ✓ **+ 4 pts Correct**
- **+ 2 pts** Not quite on track
- **+ 0 pts** Incorrect

Construction

- ✓ **+ 8 pts Correct**
- **+ 7 pts** Small mistake
- **+ 4 pts** Flawed / Incomplete
- **+ 0 pts** Incorrect

Correctness Proof

- ✓ **+ 4 pts Correct**
- **+ 2 pts** Flawed/Incomplete
- **+ 0 pts** Incorrect / Missing

### 2.2 2b 23 / 25

- **- 0 pts** Correct
- **- 2 pts** PDA: No marker for empty stack
- ✓ **- 2 pts Empty string not valid**
- **- 5 pts** PDA/CFG partially correct
- **- 10 pts** PDA/CFG incorrect
- **- 5 pts** No proof
- **- 20 pts** "I don't know"

1. **(20 points).** Consider a binary operation $\nabla$ defined as follows: if $A$ and $B$ are two languages, then $A\nabla B = \{xy \mid x \in A,\ y \in B, \text{ and } |x| = |y|\}$. Prove that if $A$ and $B$ are regular languages, then $A\nabla B$ is a context-free language.

   **Answer:**

   We will construct a PDA $P(Q_p, \Sigma_p, \Gamma_p, \delta_p, q_{p0}, F_p)$ that accepts $A\nabla B$ with the following intuition. Let $M(Q_m, \Sigma_m, \delta_m, q_{m0}, F_m)$ and $M(Q_n, \Sigma_n, \delta_n, q_{n0}, F_n)$ be the DFAs that accepts $A$ and $B$, respectively. We will first process the input using $M$, pushing onto the stack for every transition in $M$. In the accepting states of $M$, we will transition to $N$. Then we process the rest of the input in $N$, popping off the stack for every transition in $N$. We will only accept if we reach an accept state of $N$ with no more input left to process and have an empty stack.

$$Q_p = Q_m \cup Q_n \tag{1}$$
$$\Sigma_p = \Sigma_m \cup \Sigma_n \tag{2}$$
$$\Gamma_p = \{\$, 0\} \tag{3}$$

   Any situations not described by the transition function should transition to a rejection state.

$$\delta_p(x \in Q_m, a \in \Sigma_m, \epsilon) = (\delta_m(x, a), 0) \tag{4}$$
$$\delta_p(x \in F_m, \epsilon, A) = (q_{n0}, A) \tag{5}$$
$$\delta_p(x \in Q_n, a \in \Sigma_n, A) = (\delta_n(x, a), \epsilon) \tag{6}$$
$$q_{p0} = (q_{m0}, \$) \tag{7}$$
$$F_p = \{(q, \$)| q \in F_n\} \tag{8}$$

   To prove correctness, let us consider some word $w$ in the language that $P$ accepts. Following our construction, $w$ must be processed as follows. A \$ will be pushed onto the stack, and then P will push $n$ 0s onto the stack as it processes some $n$ characters of $w$. It will then reach an state in $F_m$, and transition to the start state of $N$. Since $P$ accepts $w$, $P$ will then read the rest of $w$ and will end by reaching a state in $F_n$ with only a \$ on the stack, and nothing left to process in $w$. Since $P$ pops a 0 off the stack every time it processes a state in $Q_m$, this must mean the second part of the string processed by $N$ is also $n$ characters in length. Both halves thus have the same length, and since the only way for a string to be accepted in $P$ is to reach an accepting state of $M$ and $N$, these strings must respectively be in A and B. Therefore, $w \in A\nabla B$. The converse applies with the same reasoning. We can write all $w \in A\nabla B$ as $xy||x| = |y|$ which would then cause $P$ to accept it. We can thus see that $L(P) = A\nabla B$. Since we were able to construct a PDA that accepts $A\nabla B$, $A\nabla B$ must be a context-free-grammar.

2. **(45 points).** This problem explores two related languages. Remember to use the ideas from part (a) in part (b).

   (a) **(20 points).** Show that the language

$$L_1 = \{x\$y \mid x, y \in \{0,1\}^* \text{ and } x \neq y\}$$

   over the alphabet $\Sigma = \{\$, 0, 1\}$ is a context-free language.

   **Answer:**

## 1 Question 1 **12 / 20**

- **- 0 pts** Correct
- **- 1 pts** confusion on state vs set
- **- 3 pts** missing/incorrect 1-2 transition specifications
- ✓ **- 8 pts correct intuition, unclear construction details**
- **- 1 pts** domain error/confusion
- **- 1 pts** minor mistake
- **- 5 pts** explanation incomplete/not detailed enough
- **- 20 pts** wrong idea, reduced the problem to concatenation
- **- 12 pts** good attempt

We can define a grammar $G$ whose language corresponds to $L_1$ as follows:

$$S \rightarrow A0C|B1C|M|\epsilon \tag{9}$$
$$M \rightarrow NMN|\$NC|CN\$ \tag{10}$$
$$A \rightarrow NAN|1C\$ \tag{11}$$
$$B \rightarrow NBN|0C\$ \tag{12}$$
$$C \rightarrow CN|\epsilon \tag{13}$$
$$N \rightarrow 0|1 \tag{14}$$

We can consider the intuition behind this grammar as follows. Let us consider $w = x\$y$ In order to determine if $x! = y$, there are 2 scenarios. The first scenario is if $x$ and $y$ aren't of the same length, then we simply need to compare the length. The second is if $x$ and $y$ are of the same length, then we should compare the $i$th character of $x$ and $y$ until we find a character difference.

Let us consider the first scenario. In order to handle this scenario, we use the symbol $M$. The $NMN$ parsing will effectively process the input string from both ends 1 character at a time. If one half of the string, $x$, is shorter than the other half, $y$, then it will then be accepted by the $\$NC|CN\$$ parsing of $M$.

Let us consider the second scenario. This is handled by the $A$ and $B$ symbols. The way that the $A$ and $B$ symbols are invoked in $S$, it is always guaranteed that there will be the same number of $N$s both before the \$ and after it. Assuming that $x$ and $y$ are of the same length $m$, this construction will create $m$ parsings, comparing the $i$th character of $x$ with the $i$th character of $y$. As long as $x$ and $y$ differ by at least one character, the grammar will accept the word.

The language accepted by $G$ will have the following properties. Firstly, it must always have a \$ symbol in it. Thus we can consider all words in the language of $G$ to be of the form $x\$y$. By the rules explained previously, we know that $|x|! = |y|$ and $x! = y$. Thus all $w \in L(G) \in L_1$.

Conversely, we can see that all $w \in L_1 \in L(G)$, since it is of the form $x\$y$ such that $x, y \in \{0,1\}^*$ and $x \neq y$. Thus $L(G) = L_1$. Since we were able to define a grammar whose language corresponds to $L_1$, $L_1$ is a context-free language.

(b) **(25 points).** Show that the language

$$L_2 = \{xy \mid x, y \in \{0,1\}^*, \; |x| = |y|, \text{ and } x \neq y\}$$

is a context-free language.

**Answer:**

We can define a grammar $G$ whose language corresponds to $L_2$ as follows:

$$S \rightarrow AB|BA|\epsilon \tag{15}$$
$$A \rightarrow NAN|1 \tag{16}$$
$$B \rightarrow NBN|0 \tag{17}$$
$$N \rightarrow 0|1 \tag{18}$$

## 2.1 2a 20 / 20

✓ + **4 pts** **Not totally incorrect**

Intuition

✓ + **4 pts** **Correct**

+ **2 pts** Not quite on track

+ **0 pts** Incorrect

Construction

✓ + **8 pts** **Correct**

+ **7 pts** Small mistake

+ **4 pts** Flawed / Incomplete

+ **0 pts** Incorrect

Correctness Proof

✓ + **4 pts** **Correct**

+ **2 pts** Flawed/Incomplete

+ **0 pts** Incorrect / Missing

We can define a grammar $G$ whose language corresponds to $L_1$ as follows:

$$S \to A0C|B1C|M|\epsilon \tag{9}$$
$$M \to NMN|\$NC|CN\$ \tag{10}$$
$$A \to NAN|1C\$ \tag{11}$$
$$B \to NBN|0C\$ \tag{12}$$
$$C \to CN|\epsilon \tag{13}$$
$$N \to 0|1 \tag{14}$$

We can consider the intuition behind this grammar as follows. Let us consider $w = x\$y$ In order to determine if $x! = y$, there are 2 scenarios. The first scenario is if $x$ and $y$ aren't of the same length, then we simply need to compare the length. The second is if $x$ and $y$ are of the same length, then we should compare the $i$th character of $x$ and $y$ until we find a character difference.

Let us consider the first scenario. In order to handle this scenario, we use the symbol $M$. The $NMN$ parsing will effectively process the input string from both ends 1 character at a time. If one half of the string, $x$, is shorter than the other half, $y$, then it will then be accepted by the $\$NC|CN\$$ parsing of $M$.

Let us consider the second scenario. This is handled by the $A$ and $B$ symbols. The way that the $A$ and $B$ symbols are invoked in $S$, it is always guaranteed that there will be the same number of $N$s both before the \$ and after it. Assuming that $x$ and $y$ are of the same length $m$, this construction will create $m$ parsings, comparing the $i$th character of $x$ with the $i$th character of $y$. As long as $x$ and $y$ differ by at least one character, the grammar will accept the word.

The language accepted by $G$ will have the following properties. Firstly, it must always have a \$ symbol in it. Thus we can consider all words in the language of $G$ to be of the form $x\$y$. By the rules explained previously, we know that $|x|! = |y|$ and $x! = y$. Thus all $w \in L(G) \in L_1$.

Conversely, we can see that all $w \in L_1 \in L(G)$, since it is of the form $x\$y$ such that $x, y \in \{0,1\}^*$ and $x \neq y$. Thus $L(G) = L_1$. Since we were able to define a grammar whose language corresponds to $L_1$, $L_1$ is a context-free language.

(b) **(25 points).** Show that the language

$$L_2 = \{xy \mid x, y \in \{0,1\}^*,\ |x| = |y|, \text{ and } x \neq y\}$$

is a context-free language.

**Answer:**

We can define a grammar $G$ whose language corresponds to $L_2$ as follows:

$$S \to AB|BA|\epsilon \tag{15}$$
$$A \to NAN|1 \tag{16}$$
$$B \to NBN|0 \tag{17}$$
$$N \to 0|1 \tag{18}$$

Let us consider the language described by $G$. By the nature of our grammar, we can first see that any word in the language will be of even length. $A$ will always give us an odd-length parse, as will $B$, and since $S$ combines these 2 symbols together, we know that 2 odd length strings will result in an even length string.

The intuition behind this grammar is to take $w$ of length $2m$, and compare every $i$th character with the $(i+m)$th character. We do this via the construction of the $A$ and $B$ symbols, which are padded by the same number of $N$ symbols on either side. This means we can always split $w$ into 2 strings of length $m$ with an equal number of $N$s in each string.

All words accepted by the grammar $G$ will be of length $2m | m \in Z, m >= 0$ and has at least one $i | 0 <= i <= m$ such that the $i$th character is not equal to the $(i+m)$th character, then $w$ is in the language of $G$. This means that $w$ could be split evenly into 2 strings $s1, s2$ of length $m$, and $s1! = s2$, which matches the construction of $L_2$. Thus $L(G)$ is a subset of $L_2$.

Conversely, since we can see that since all $w$ in $L_2$ is of the form $\{x, y \in \{0,1\}^*$, $|x| = |y|$, and $x \neq y\}$, then $L_2$ is a subset of $L(G)$. Since we were able to define a grammar whose language corresponds to $L_2$, $L_2$ is a context-free language.

## 2.2 2b **23 / 25**

- **- 0 pts** Correct
- **- 2 pts** PDA: No marker for empty stack
- ✓ **- 2 pts Empty string not valid**
- **- 5 pts** PDA/CFG partially correct
- **- 10 pts** PDA/CFG incorrect
- **- 5 pts** No proof
- **- 20 pts** "I don't know"