## 2 Theory Questions

Write the answers to these questions in your `writeup.pdf` file. You can use both typewritten or embedded photo captures of handwritten work. The former is preferred for convenience of the CS4248 TA staff.

4. **[10%]** — Subtraction Regular Expressions

   Given a string in the form of `A - B = C` where A, B, and C contains arbitrary *kmy*[non-zero] number of the character `a`. Write a regular expression that accepts all valid subtractions, and reject all invalid subtractions. Hint: you may want to learn how regular expressions can capture groups for back reference.

   For instance, the regex should accept:

   ```
   aaaa - aaa = a
   aaaaaa - aa = aaaa
   ```

   and should reject:

   ```
   aaaa - aaa = aa
   aa - aaa = a
   ```

**Explanation:** ^(a+)(a+)- \2 = \1$

The ^ and $ (or other word boundaries \b)are important as well otherwise the solution would also match cases like **aa** - **a** = **a**aaaa. Points have been deducted if this has not been handled.

Alternative solutions: ^(a*)(.+)- \2 = \1$, ^((a+)(a+) - \2 = \3)$

Solutions that handle spaces (\s) explicitly are also accepted. Other solutions also exist *but* are longer.

5. **[25%]** Regular Expression (**Language Modelling**)

A language model consists of a vocabulary $V$, and a function $p(x_1 \dots x_n)$ such that for all sentences $x_1 \dots x_n \in V^+$, $p(x_1 \dots x_n) > 0$, and in addition $\sum_{x_1 \dots x_n \in V^+} p(x_1 \dots x_n) = 1$. Here $V^+$ is the set of all sequences $x_1 \dots x_n$ such that $n \geq 1, x_i \in V$ for $i = 1 \dots (n-1)$, and $x_n$ = **STOP**.

We assume that we have a bigram language model, with

$$p(x_1, \dots x_n) = \Pi_{i=1}^{n} q(x_i | x_{i-1})$$

The parameters $q(x_i|x_{i-1})$ are estimated from a training corpus using a discounting method, with discounted counts

$$c^*(v, w) = c(v, w) - \beta$$

where $\beta = 0.5$.

We assume in this question that all words seen in any test corpus are in the vocabulary $V$, and each word in any test corpus is seen at least once in training.
There are 3 subparts to this question:

1. For any test corpus, the perplexity under the language model will be less than $\infty$. True or False? Justify.
2. For any test corpus, the perplexity under the language model will be at most $N+1$, where $N$ is the number of words in the vocabulary $V$. True or False? Justify your response.
3. Now consider a bigram language model where for every bigram ($v$, $w$) where $w \in V$ or $w = \textbf{STOP}$,
$$q(w|v) = \frac{1}{N+1}$$
where $N$ is the number of words in the vocabulary $V$.
For any test corpus, the perplexity under the language model will be equal to $N+1$. True or False? Justify your response.

**Explanation:** 1. True, since $p(x_1, \dots x_n)$ is $> 0$ for all sentences $w_1 \dots w_n$ where $n \geq 1$.

2. False. The statement can be disproved using a counter example. A correct submission by one of the students is listed below for reference-
The perplexity will **not** be at most N + 1. This can be shown using a counterexample: Assume we use the following discounting method:
$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) - \beta}{\sum_v C(w_{i-1}v)} + term2 \tag{1}$$
where $term2$ is a term to make sure the probability masses are distributed properly to get valid probability distributions. Eq. (1) could be Kneser-Ney Smoothing in which case only the first term is used for all word pairs where $C(w_{i-1}w_i) > 0$
Assume the training corpus is:
$$< s > w_1 \dots w_k STOP$$
while the test corpus *Test* is
$$< s > w_k STOP$$
This combination of training and test corpus invalidates none of the assumptions given in the assignment text. Thus we get:
$$P(w_k| < s >) = \frac{C(< s >, w_k) - \beta}{C(< s >)} = \frac{1 - 0.5}{1} = 0.5$$

$$P(STOP|w_k) = \frac{C(w_k, STOP) - \beta}{C(w_k)} = \frac{1 - 0.5}{k} = \frac{0.5}{k}$$

Thus, Perplexity can be calculated as:

$$PP(W) = \sqrt{\frac{1}{0.5} \cdot \frac{k}{0.5}} = \sqrt{4k}$$

$$= 2\sqrt{k} > 3 = N + 1 (k \geq 3)$$

We see how we can make the perplexity arbitrarily high by having a training corpus with more $w_k$ tokens. Thus it has been proved that the perplexity for any test corpus will **not** at most be N + 1 where N is the number of words in the vocabulary from the training corpus and excluding the *STOP* token. If we do not need to include <s> in the count of N, the example holds already from $k2$.

3. True, the perplexity will be N + 1. We know that

$$PP(W) = \sqrt[N]{\left(\frac{1}{q(w_i|w_{i-1})}\right)^N}$$

where $N$ = Number of words in V.
Since we are considering all $w \in V$ and $w = STOP$, $N' = N + 1$

$$PP(W) = \sqrt[N+1]{\left(\frac{1}{\left(\frac{1}{N+1}\right)}\right)^{N+1}}$$

$$= N + 1$$