University of Toronto Scarborough

CSC B36 **Final Examination** **22 August 2019**

NAME: ______________________________
(circle your last name)

STUDENT NUMBER: ____________________

Do not begin until you are told to do so. In the meantime, put your name and student number on this cover page and read the rest of this page.

Aids allowed: None. .... **Duration:** 3 hours.

There are 10 pages and each is numbered at the bottom. Make sure you have all of them.

Write legibly in the space provided. Use the backs of pages for rough work; they will not be graded.

1.	____	/ 10
2.	____	/ 20
3.	____	/ 10
4.	____	/ 10
5.	____	/ 14
6.	____	/ 6
7.	____	/ 10
Total	____	/ 80

1. **[10 marks total]** Let $S = \{3p + 5q :\ p, q \in \mathbb{N}\}$. For this question our desired result is

$$\{n \in \mathbb{N} :\ n \geq 8\} \subseteq S.$$

You will prove this result in two ways.

(a) **[1 mark]** Write an appropriate predicate on $\mathbb{N}$ for the purpose of proving our desired result.

(b) **[4 marks]** Prove our desired result with a direct proof using the Principle of Simple Induction from class. After your induction step set-up, it suffices to just state what you want to prove.

(c) **[5 marks]** Prove our desired result with a direct proof using the Principle of Complete Induction from class. After your induction step set-up, it suffices to just state what you want to prove.

2. **[20 marks]** Consider the program and its specification given on the last page of this exam.

   Use methods from class to prove this program correct with respect to its specification.

   You may make use of the following facts without proof.

   Fact 1: If $i = \text{len}(L1)$, then $nGT(L1, L2[:j]) = nGT(L1, L2)$.
   Fact 2: If $j = \text{len}(L2)$, then $nGT(L2, L1[:i]) = nGT(L2, L1)$.

[more space available on next page . . .]

[. . . additional space for question 2]

3. **[10 marks]** Let $\Sigma = \{0, 1\}$.
Consider the string and language operations given on the last page of this exam.

We define a predicate on regular expressions (over Σ) as follows.

$$P(R): \text{There are regular expressions } R_s,\ R_e,\ R_b, \text{ such that } \mathcal{L}(R_s) = \text{StartFlip}(\mathcal{L}(R)),\ \mathcal{L}(R_e) = \text{EndFlip}(\mathcal{L}(R)) \text{ and } \mathcal{L}(R_b) = \text{BothFlip}(\mathcal{L}(R)).$$

Nick worked on a structural induction proof to prove that $P(R)$ holds for every regular expression R, thereby proving that regular languages are closed under StartFlip, EndFlip and BothFlip. He wrote some rough notes for his proof, but before he had a chance to write out the final version of his proof, the Proof Stealer struck! All the most important parts of the notes were erased.
Please help Nick restore the notes for his proof by filling in the blanks below.

Base Cases:

If $R = \epsilon$, then let

$R_s =$ ________________, $R_e =$ ________________, $R_b =$ ________________.

If $R = \emptyset$, then let

$R_s =$ ________________, $R_e =$ ________________, $R_b =$ ________________.

If $R = 0$, then let

$R_s =$ ________________, $R_e =$ ________________, $R_b =$ ________________.

If $R = 1$, then let

$R_s =$ ________________, $R_e =$ ________________, $R_b =$ ________________.

Induction Step:

Assume regular expressions $S_s,\ S_e,\ S_b,\ T_s,\ T_e,\ T_b$ exist as per $P(S)$ and $P(T)$.

If $R = S + T$, then let

$R_s =$ ________________, $R_e =$ ________________, $R_b =$ ________________.

If $R = ST$, then let

$R_s =$ ________________ if $\epsilon \notin \mathcal{L}(S)$, $R_s =$ ________________ if $\epsilon \in \mathcal{L}(S)$;

$R_e =$ ________________ if $\epsilon \notin \mathcal{L}(T)$, $R_e =$ ________________ if $\epsilon \in \mathcal{L}(T)$;

$R_b =$ ________________ if $\epsilon \notin \mathcal{L}(S)$ and $\epsilon \notin \mathcal{L}(T)$,

$R_b =$ ________________ if $\epsilon \notin \mathcal{L}(S)$ and $\epsilon \in \mathcal{L}(T)$,

$R_b =$ ________________ if $\epsilon \in \mathcal{L}(S)$ and $\epsilon \notin \mathcal{L}(T)$,

$R_b =$ ________________ if $\epsilon \in \mathcal{L}(S)$ and $\epsilon \in \mathcal{L}(T)$.

If $R = S^*$, then let

$R_s =$ ________________, $R_e =$ ________________, $R_b =$ ________________.

4. **[10 marks total; 5 for each part]** Let $\Sigma = \{0, 1\}$. For $x, y \in \Sigma^*$, let $\#_y(x) = |\{(u, v) : x = uyv\}|$. Informally, $\#_y(x)$ is the number of places in x where y appears as a substring, and $\#_y(x) = 0$ means y is not a substring of x. As well, $x = x^R$ means x is a palindrome.

   (a) Let $L_{4a} = \{x \in \Sigma^* : \#_{00}(x) = 0 = \#_{11}(x) \text{ and } x = x^R\}$. Using as few states as possible, design a DFSA that accepts L_{4a}. Give both a state invariant and a state diagram.

   (b) Let $L_{4b} = \{x \in \Sigma^* : \#_{000}(x) = 0 = \#_{111}(x) \text{ and } x = x^R\}$.
   Use the Pumping Lemma to prove that L_{4b} is **not** regular.

5. **[14 marks total; 7 for each part]** Let $\Sigma = \{0,1\}$. For $x, y \in \Sigma^*$, let $\#_y(x) = |\{(u,v) : x = uyv\}|$.
Let $L_5 = \{x \in \Sigma^* : \#_{000}(x) = 0 = \#_{111}(x) \text{ and } x = x^R\}$.
L_5 is the same language as L_{4b} from question 4.

(a) Here is Nick's design for a CFG that generates L_5. It has 3 variables and 11 productions.
S generates L_5.
A generates $\{x : x \in L_5 \text{ and } x \text{ starts with } 0\}$.
B generates $\{x : x \in L_5 \text{ and } x \text{ starts with } 1\}$.
Complete Nick's CFG below according to his design by adding 11 productions.

$S \rightarrow$

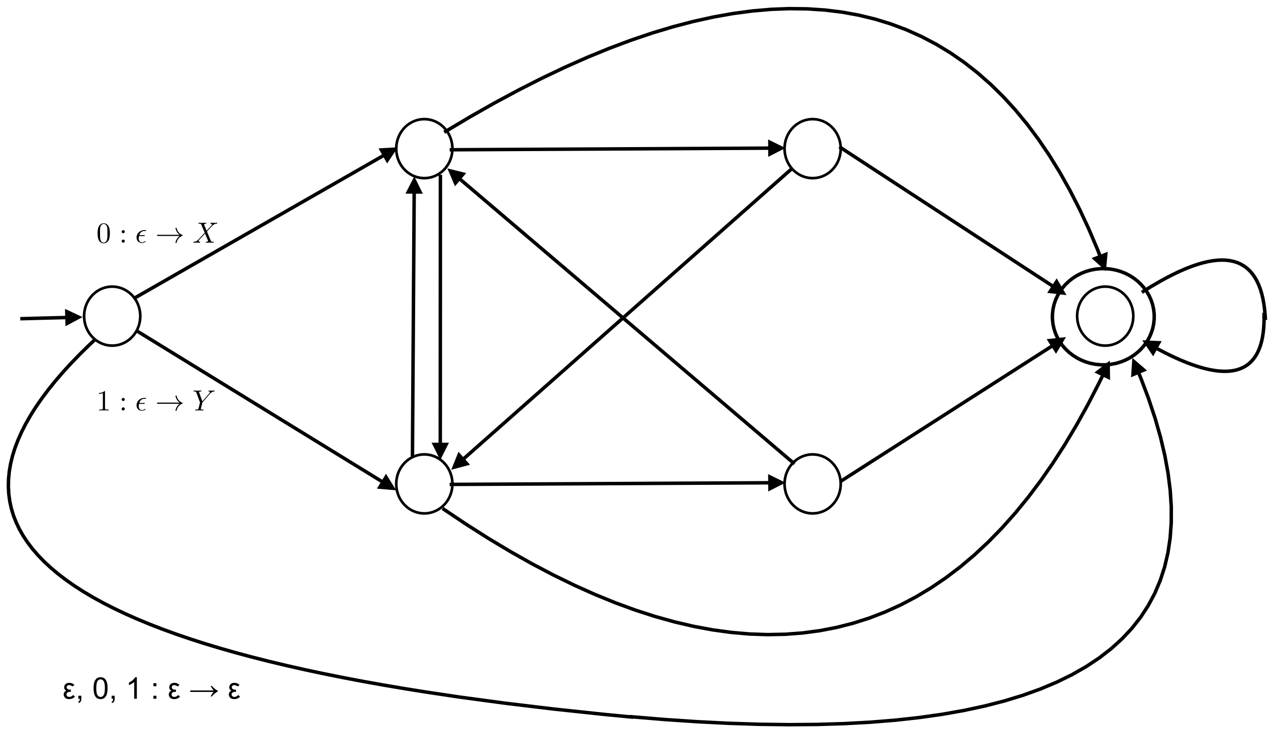
$A \rightarrow$

$B \rightarrow$

(b) Nick created a PDA that accepts L_5. Before he showed it to anyone, the Automaton Mangler struck! All transition labels, except those for transitions coming out of the initial state, were removed and thrown into a list. Here is the list of removed labels (many appear multiple times).

$$\epsilon : \epsilon \rightarrow \epsilon \quad \epsilon : \epsilon \rightarrow \epsilon$$
$$0 : \epsilon \rightarrow \epsilon \quad 0 : \epsilon \rightarrow \epsilon \quad 0 : \epsilon \rightarrow X \quad 0 : \epsilon \rightarrow X \quad 0 : \epsilon \rightarrow X \quad 0 : X \rightarrow \epsilon$$
$$1 : \epsilon \rightarrow \epsilon \quad 1 : \epsilon \rightarrow \epsilon \quad 1 : \epsilon \rightarrow Y \quad 1 : \epsilon \rightarrow Y \quad 1 : \epsilon \rightarrow Y \quad 1 : Y \rightarrow \epsilon$$

Please restore Nick's PDA below by adding labels to the 11 unlabeled transitions.
There are 14 labels, so some transitions have more than one label.



6. **[6 marks total; 3 for each part]**
Consider the ternary boolean function $f : \{0,1\}^3 \to \{0,1\}$ defined by the truth table below.

x	y	z	$f(x,y,z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

(a) Use the above truth table to give a DNF formula with variables x, y, z that represents $f(x,y,z)$.

(b) Use the above truth table to give a CNF formula with variables x, y, z that represents $f(x,y,z)$.

7. **[10 marks total; 5 for each part]**
Consider a first-order language with ternary predicate T and equality predicate $=$.

For each pair of formulas below, give an interpretation that satisfies exactly one of them.
To earn full credit, your interpretation must have the smallest domain possible.
Briefly explain your answer and clearly identify the formula that is satisfied by your interpretation.

(a) F_1: $\forall x\ \forall y\ \forall z\ \Big(T(x,y,z) \rightarrow T(z,x,y)\Big)$

F_2: $\forall x\ \forall y\ \forall z\ \Big(T(x,y,z) \rightarrow \big(T(y,x,z) \wedge T(x,z,y)\big)\Big)$

(b) F_3: $\forall x\ \forall y\ \forall z\ \forall w\ \Big(\big(T(x,y,z) \wedge T(x,y,w)\big) \rightarrow = (z,w)\Big)$

F_4: $\forall x\ \forall y\ \Big(\neg T(x,y,z)\ \vee\ \neg T(x,y,w) \vee\ = (z,w)\Big)$

You may detach this page if you find it convenient to do so.

There is no need to hand in this page, as nothing on it will be marked.

Program and its specification for question 2

```
    ▷ Precondition: $L1, L2$ are each a sorted list of distinct integers.
    ▷ Postcondition: Return $nGT(L1, L2) - nGT(L2, L1)$, where
    ▷ $nGT(A, B) = |\{(x, y) : x \in A, y \in B, x > y\}|$.
    ▷ E.g., if $L1$ is $[1, 3, 7]$ and $L2$ is $[2, 3, 5, 7]$, then
    ▷ $nGT(L1, L2) = 4$, $nGT(L2, L1) = 6$, and difL$(L1, L2)$ returns $-2$.
    difL$(L1, L2)$
        $i = 0$;   $j = 0$;   $s = 0$;   $t = 0$;
        while $i <$ len$(L1)$ and $j <$ len$(L2)$:
            if $L1[i] > L2[j]$:
                $j = j + 1$;   $s = s +$ len$(L1) - i$
            elif $L1[i] < L2[j]$:
                $i = i + 1$;   $t = t +$ len$(L2) - j$
            else:   ▷ $L1[i] == L2[j]$
                $i = i + 1$;   $j = j + 1$;   $s = s +$ len$(L1) - i$;   $t = t +$ len$(L2) - j$
        return $s - t$
```

String and language operations for question 3

Let $\Sigma = \{0, 1\}$. Consider the string operations StartFlip, EndFlip, BothFlip : $\Sigma^* \to \Sigma^*$ defined below.

$$\text{StartFlip}(x) = \begin{cases} \epsilon & \text{if } x = \epsilon; \\ x \text{ with its first bit changed} & \text{if } x \neq \epsilon. \end{cases}$$

$$\text{EndFlip}(x) = \begin{cases} \epsilon & \text{if } x = \epsilon; \\ x \text{ with its last bit changed} & \text{if } x \neq \epsilon. \end{cases}$$

$$\text{BothFlip}(x) = \begin{cases} \epsilon & \text{if } x = \epsilon; \\ x \text{ with its first bit and its last bit changed} & \text{if } x \neq \epsilon. \end{cases}$$

These string operations can be extended to operate on languages (over Σ) as follows.

$$\text{StartFlip}(L) = \{\text{StartFlip}(x) : x \neq \epsilon \text{ and } x \in L\}.$$

$$\text{EndFlip}(L) = \{\text{EndFlip}(x) : x \neq \epsilon \text{ and } x \in L\}.$$

$$\text{BothFlip}(L) = \{\text{BothFlip}(x) : x \neq \epsilon \text{ and } x \in L\}.$$