CMSC330 - Organization of Programming Languages
Spring 2023 - Exam 2

CMSC330 Course Staff
University of Maryland
Department of Computer Science

Name: ______________________________

UID: ______________________________

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination

Signature: ______________________________

Ground Rules

- You may use anything on the accompanying reference sheet anywhere on this exam
- Please write legibly. **If we cannot read your answer you will not receive credit**
- You may not leave the room or hand in your exam within the last 10 minutes of the exam
- The last page is blank and scratch work can be done there.
- If anything is unclear, ask a proctor. If you are still confused, write down your assumptions in the margin

Question	Points
Q1	10
Q2	18
Q3	10
Q4	12
Q5	15
Q6	15
Total	80

Problem 1: Language Concepts

[Total 10 pts]

An improper garbage collector can cause security vulnerabilities	**True**	**False**
Modern Languages use a combination of Reference Counting, Mark and Sweep and Stop and Copy	**True**	**False**
Lambda Calculus Expressions can be converted to Finite State Machines	**True**	**False**
The relation of FSM to Regex is bijective (1 to 1)	**True**	**False**
Eager and Lazy Evaluation will always give the same result	**True**	**False**

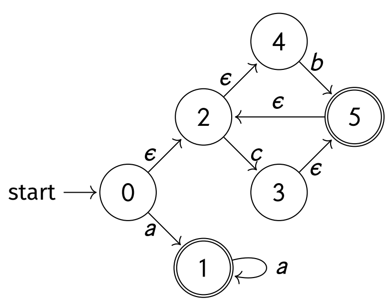
Problem 2: Finite State Machines

[Total 18 pts]

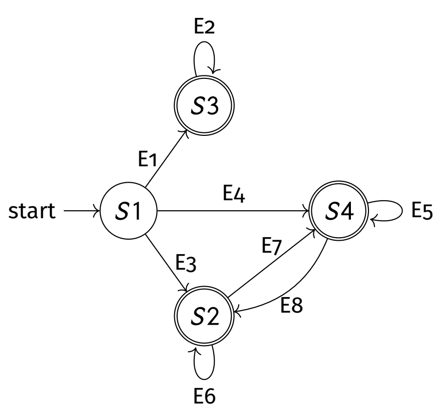
(a) Using the subset algorithm, convert the following NFA to a DFA, and fill in the blanks appropriately matching the DFA provided with the right nodes and transitions. Only the blanks will be graded. [12 pts]

NFA:

Scratch Space (if needed)



DFA:



S1: S2: S3: S4:

E1: E2: E3: E4:

E5: E6: E7: E8:

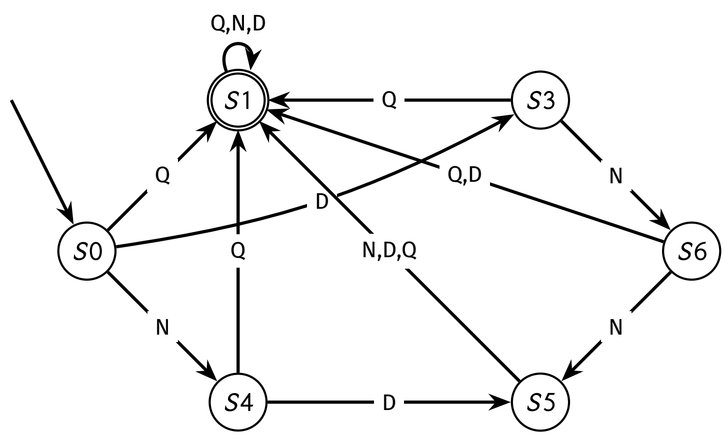
[3 pts] (b) Write a regex to describe the language of the above NFA

[3 pts] (c) Vending Machine Fun

Suppose there is a vending machine which takes in quarters (Q), dimes (D) and nickles (N). Consider the following actions you can perform when interacting with the vending machine:

Action N: Insert a Nickle   Action D: Insert a Dime   Action Q: Insert a Quarter

The price of each item is $0.25. However, the FSM for the machine was leaked and turns out you can pay less than $0.25 per item. List out the operations you want to perform to pay less than $0.25. For example, if you wanted to put in 2 quarters, followed by 1 dime, followed by 3 nickles, your answer should be `Q,Q,D,N,N,N`.



Problem 3: CFGs

[Total 10 pts]

Consider the following Grammars:

Grammar 1

$$S \rightarrow aSb \mid aaSb \mid aaaSb \mid \epsilon$$

Grammar 2

$$S \rightarrow AAASB \mid \epsilon$$
$$A \rightarrow a \mid \epsilon$$
$$B \rightarrow b$$

Grammar 3

$$S \rightarrow ASB$$
$$A \rightarrow aA \mid \epsilon$$
$$B \rightarrow bbbB \mid \epsilon$$

(a) Which of the following grammars describe strings of $a^x b^y$, $x < 3y$? Select all that apply. [2 pts]

Grammar 1   Grammar 2   Grammar 3   None

(b) Prove that Grammar 2 is ambiguous [3 pts]

(c) Draw the abstract syntax tree that would be generated by parsing the following string with the given CFG using a leftmost derivation. [5 pts]

String: "1 * 2 + 3"
CFG:
S -> M * S | M
M -> M + N | N
N -> 1 | 2 | 3 | (N), where n is any number

[Total 12 pts] **Problem 4: Operational Semantics**

Consider the following rules for LOLCODE, using OCaml as the Metalanguage:

$$\text{Rule 1: } \frac{}{\text{WIN} \longrightarrow \text{WIN}} \qquad \text{Rule 2: } \frac{}{\text{FAIL} \longrightarrow \text{FAIL}}$$

$$\text{Rule 3: } \frac{A;\ e_1 \Rightarrow v_1 \qquad A;\ e_2 \Rightarrow v_2 \qquad v_1\ <>\ v_2}{A;\ \text{DIFFRINT } e_1 \text{ AN } e_2 \Rightarrow \text{WIN}} \qquad \text{Rule 4: } \frac{A;\ e_1 \Rightarrow v_1 \qquad A;\ e_2 \Rightarrow v_2 \qquad v_1\ =\ v_2}{A;\ \text{DIFFRINT } e_1 \text{ AN } e_2 \Rightarrow \text{FAIL}}$$

$$\text{Rule 5: } \frac{A, \text{x} : v\ (\text{x})\ =\ v}{A, \text{x} : v;\ \text{x} \Rightarrow v} \qquad \text{Rule 6: } \frac{A;\ e_1 \Rightarrow v_1 \qquad A, \text{x} : v_1;\ e_2 \Rightarrow v_2}{A;\ \text{HAS A x ITZ } e_1 \ \backslash\text{n } e_2 \Rightarrow v_2}$$

$$\text{Rule 7: } \frac{A;\ e_1 \Rightarrow v_1 \qquad A;\ e_2 \Rightarrow v_2 \qquad v_3 = \text{if } v_1 > v_2 \text{ then } v_1 \text{ else } v_2}{A;\ \text{BIGGR OF } e_1 \text{AN } e_2 \Rightarrow v_3} \qquad \text{Rule 8: } \frac{}{A; n \longrightarrow n}$$

[4 pts] (a) What are the axioms in this language? Select all the apply.

Rule 1 Rule 2 Rule 3 Rule 4 Rule 5 Rule 6 Rule 7 Rule 8 none

[8 pts] (b) Complete the opsem proof for the following program:

HAS A x ITZ 7 \n DIFFRINT 2 AN (BIGGR OF 2 AN x) => WIN

$$\frac{\dfrac{}{\boxed{1}} \qquad \dfrac{\dfrac{}{\boxed{3}} \qquad \dfrac{\dfrac{}{A, x : 7; 2 \Rightarrow 2} \qquad \dfrac{\boxed{8}}{A, x : 7; x \Rightarrow 7} \qquad 7 = \boxed{7}}{A, x : 7; \boxed{4} \Rightarrow \boxed{5}} \qquad \boxed{6}}{A, x : 7;\ \text{DIFFRINT 2 AN} \boxed{2} \Rightarrow \text{WIN}}}{A; \text{HAS A x ITZ 7 } \backslash\text{n DIFFRINT 2 AN (BIGGR OF 2 AN x)} \Rightarrow \text{WIN}}$$

Blank 1: ______ Blank 2: ______ Blank 3: ______

Blank 4: ______ Blank 5: ______ Blank 6: ______

Blank 7: ______ Blank 8: ______

Problem 5: Lambda Calculus

[Total 15 pts]

For the following questions perform a single β-reduction using lazy (call by name) evaluation on the outermost expression. If you cannot reduce it, write **Beta Normal Form**. You may **not** α-convert your final answer.

(a) $(\lambda x.x\lambda y.xy)(y(\lambda x.yx))$ [2 pts]

(b) $(\lambda x.\lambda x.xx)((\lambda x.yx)((\lambda a.aa)b))$ [2 pts]

For the following questions perform a single β-reduction using Eager (call by value) evaluation on the outermost expression. If you cannot reduce it, write **Beta Normal Form**. You may **not** α-convert your final answer.

(c) $(\lambda x.x\lambda y.xy)(y(\lambda x.yx))$ [2 pts]

(d) $(\lambda x.\lambda x.xx)((\lambda x.yx)((\lambda a.aa)b))$ [2 pts]

(e) Convert the following to Beta Normal Form: $(\lambda x.(\lambda y.xa)b)(\lambda x.ax)$ [3 pts]

$\lambda x.ax$ $\quad$ $c\ d$ $\quad$ $b\ a$ $\quad$ $a\ a$ $\quad$ can't reduce $\quad$ infinite recursion $\quad$ None

Consider the following lambda calculus bindings:
true = $\lambda x.\lambda y.x$
false = $\lambda x.\lambda y.y$
if e1 then e2 else e3 = e1 e2 e3

(f) Encode the following expression: `if false then false else true` [4 pts]

[Total 15 pts] **Problem 6: Lexing, Parsing, Evaluation**

Consider the following modified Math-ew from lecture:

$$E \Rightarrow \; +E\;E \mid *\;E\;E \mid sq\;E \mid exp\;E\;E \mid and\;E\;E \mid or\;E\;E \mid N$$
$$N \Rightarrow \; 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid true \mid false$$

You may assume that the behaviour is the same as Ocaml.

[5 pts] (a) Lexing

Which of the following phrases would fail the lexing stage for the Math-ew Language? Please bubble in the circle

(A) `2 * 3 sq 2 3` (B) `4 ˆ 5` (C) `- + 1 23`

(D) `exp -2 5` (E) `5 exp 2 + 6` (F) `* 2 and true false`

(G) `and true or false false` (H) `false true` (I) `true and false or true`

[5 pts] (b) Parsing

Which of the following phrases would fail the parsing stage for the Math-ew Language? If it failed the lexing phase, **do not** mark it.

(A) `2 * 3 sq 2 3` (B) `4 ˆ 5` (C) `- + 1 23`

(D) `exp -2 5` (E) `5 exp 2 + 6` (F) `* 2 and true false`

(G) `and true or false false` (H) `false true` (I) `true and false or true`

[5 pts] (c) Evaluation

Which of the following phrases would fail the evaluator stage for the Math-ew Language? If it failed the lexing or parsing phase, **do not** mark it.

(A) `2 * 3 sq 2 3` (B) `4 ˆ 5` (C) `- + 1 23`

(D) `exp -2 5` (E) `5 exp 2 + 6` (F) `* 2 and true false`

(G) `and true or false false` (H) `false true` (I) `true and false or true`

You can use this page for scratch work: