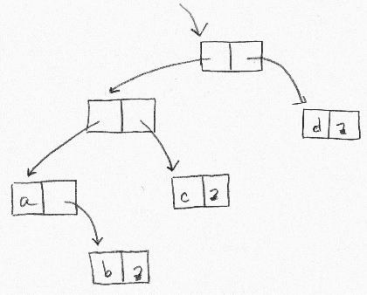
Concepts of Programming Languages, CSCI 305, Fall 2020
Exam 1, Sept. 11

Name _____

This is a closed book exam. You may not use notes, the text book, Internet, etc. If you have any questions, please call me at **406-494-7975**.

1. The following data structure in Scheme corresponds to what list? (4 pts.)



   a. (d (c) (a b))
   b. ((d) (c) (a b))
   c. ((a b) (c) (d))
   d. (((a b) c) d)
   e. None of the above

2. The programmer made a mistake in the following C code. The variable foo was declared and initialized, but later was misspelled as fob.

```
main ( ) {
        int foo=5;
        .
        .
        .
        fob++;
        .
        .
        .
}
```

   This error is most likely to be classified as a: (4 pts.)
   a. syntax error detected by the lexical analyzer
   b. syntax error detected by the parser
   c. static semantic analysis error
   d. dynamic semantic analysis error
   e. none of the above.

3. The following code snippet in C contains an error (function name begins with a digit).

```
int 5mult(int x, int  y) {
        return(x*y);
}
```

This error is most likely to be classified as a: (4 pts.)

a. syntax error detected by the lexical analyzer
b. syntax error detected by the parser
c. static semantic analysis error
d. dynamic semantic analysis error
e. compiler can't catch

4. The following code snippet in C contains an error (+-).

```
main ( ) {
        int range =  10;
        int amount = 100;
        if (range <= 0 )
                amount +-;
        else
                amount --;
}
```

This error would be classified as a: (4 pts.)

a. syntax error detected by the lexical analyzer
b. syntax error detected by the parser
c. static semantic analysis error
d. dynamic semantic analysis error
e. compiler can't catch

5. Which is the least likely to be a task of the lexical analyzer? (4 pts.)

a. Collect characters into logical groupings
b. Detect errors
c. Build symbol table
d. Remove whitespace
e. Remove comments

6. Suppose that you are designing a programming language. Give three examples of issues that you would need to decide for your programming language. That is, give three examples of possible language tradeoffs, where there are multiple ways for your programming language to achieve something, and you, as a language designer, need to decide which approach to use. (10 pts.)

- To make the language compiled, interpreted or some type of mixture using a virtual machine
- To make the language imperative/procedural, or functional, or to somehow include both. (I could also consider designing a better logical language)
- To make the type system strong, weak or somewhere in between
- Allow coercion between type systems – I could make it explicit or implicit, or somewhere in between
- Use short circuit evaluation or not, or allow both with a special syntax
- Use garbage collection or not
- Use static or dynamic scope for variables
- Use pass by value or pass by reference for procedures, or some combination
- Keep structures on the stack or in the heap
- How to indicate comments in the language
- To support automatic documentation in the language
- To require header files or not.

7. Give the definition of regular expressions which is given in the text and which we have been using in class. (10 pts.)

Regular expressions are defined recursively as:

Basis

- Any character in the alphabet is a regular expression
- The empty string, ε, is a regular expression

Induction, given regular expressions r_1 and r_2

- Concatenation: $r_1 \circ r_2$ is a regular expression (can just write $r_1 r_2$)
- Alteration: $r_1 \mid r_2$ is a regular expression
- Kleene closure: r_1 * is a regular expression

Use parenthesis to avoid ambiguity.

8. Give a regular expression that will recognize all strings of a's and b's, which do not contain the string ab. (5 pts.)

b*a*

9. Write a regular expression for integers in a language that allows integers to have exponents.

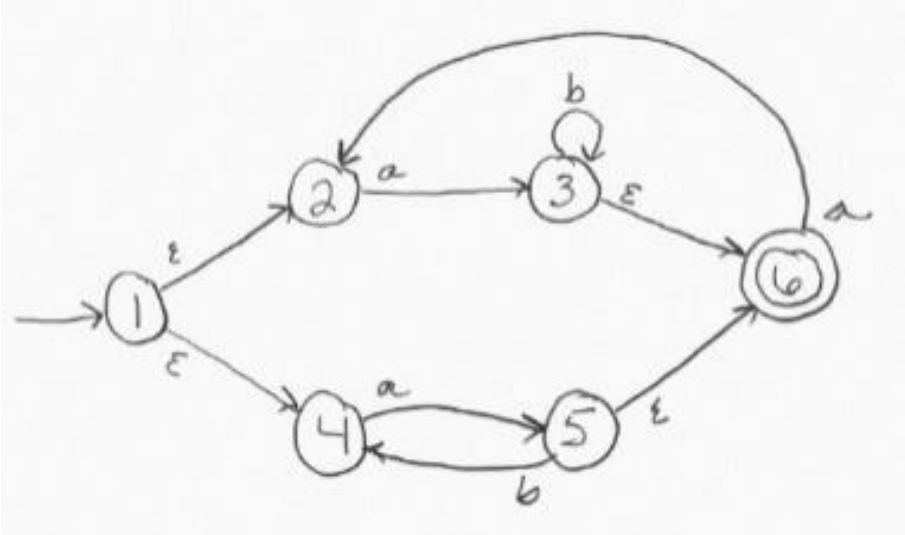
   The integers can be 0, or a possibly signed number that doesn't begin with 0. Exponents are not required, but if they are given, they are specified using either a lower-case or upper-case 'e' followed by a number. The exponent cannot be signed. A 0 exponent is allowed, but if the exponent is not zero, it cannot start with zero.

   Legal integers: 0, -12, +12, 12, 12e0, 12E10, 0E6000
   Illegal: 003, +0, 12e+0, 12e012 (10 pts.)

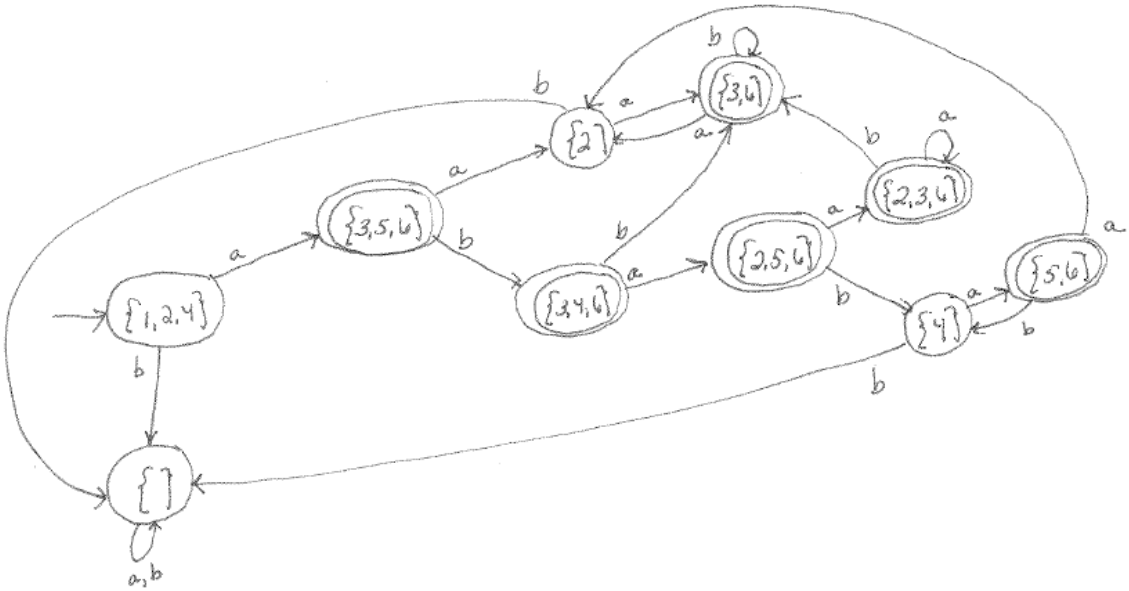
   Helpers:
   nonZ → 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
   digit → 0 | nonZ
   num → nonZ digit*
   sign → + | -
   exp → e | E

   Definition:
   integer → 0 | ((ε | sign) num (ε | exp (0 | num))

10. Using the mechanical method described in class, convert the following NFA to an equivalent DFA. (10 pts.)



Answer:



11. Define what it means for a language to have referential transparency. (5 pts.)

That an expression always evaluates to the same result in any context (so the equivalence of two expressions at any point in time implies their equivalence at all times). This means that an expression can be replaced by its value without changing the program's behavior.

See page 581 in the text for more discussion. It says that this means independent of evaluation order. It allows programmers to employ equational reasoning, in which the equivalence of two expressions at any point in time implies their equivalence at all times. This is very attractive for parallel execution since the arguments to a function can safely be evaluated in parallel.

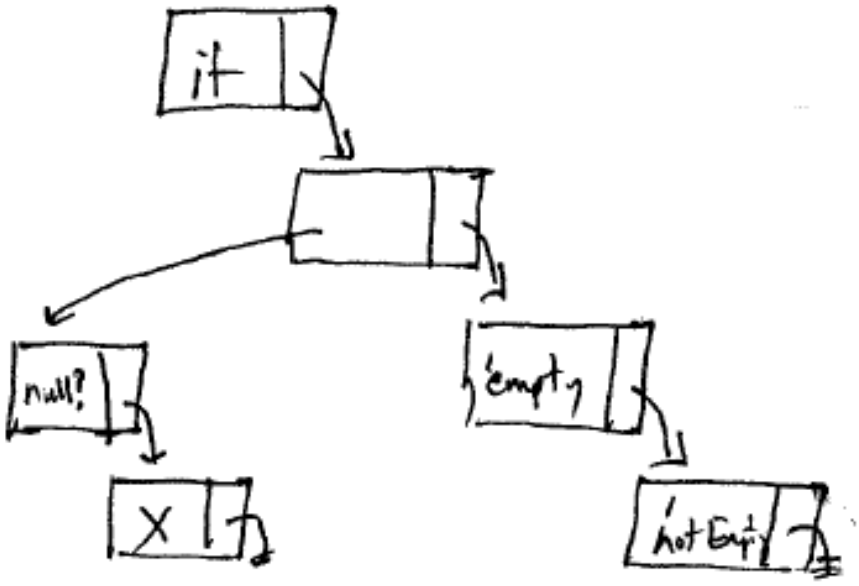
If an expression is not referentially transparent, it is referentially opaque. Two terms are referentially opaque if they cannot be substituted for each other without possibly changing the truth value of the statement.

12. Describe the difference between a lexeme and a token. (5 pts.)

Lexeme – lowest level of program
Token – category of lexemes

13. Show how the following Scheme statement would be stored within the system.
(if (null? X) 'empty 'notEmpty) (5 pts.)

Answer:



Write programs in Scheme for the following. Do not use the computer to write these programs, simply write your program onto the test page.

14. Write a Scheme function called *countZero* that returns the number of zeros in a given simple list of numbers. For example

(*countZero* '(17 5 8 0 6 0 3 0)) returns 3
(*countZero* '(0 0 0 0)) returns 4
(*countZero* '(17 5 80)) returns 0 (10 pts.)

```
(define countZero
        (lambda (myList)
                (if (null? myList)
                    0
                    (if   (= (car myList) 0)
                          (+ 1 (countZero (cdr myList)))
                          (countZero (cdr myList))
                    ); inner if
                ); outer if
        ); lambda
); define
```

Alternatively answer with an acceptable use of a helper function:

```
(define coutZeroHelper
        (lambda (number)
                (if (= number 0) 1 0)
         ); lambda
); define


(define countZero
         (lambda (myList)
                (if (null? myList)
                    0
                    (+ (coutZeroHelper (car myList))
                            (countZero (cdr myList)))
                 ); if
        ); lambda
); define
```

15. Write a Scheme function called Replace that takes two atoms and a list as parameters and replaces all occurrences of the first given atom in the list with the second given atom.

For example:

(replace 'x 'a '(a b x x c)) returns (a b a a c)

(10 pts.)

```
(define replace
  (lambda (f r myList); f - find, r - replace
  (cond
    ((null? myList) myList)
    ((equal? f (car myList))    ; Element is at the first level
         (cons r (replace f r (cdr myList))))
    (else (cons (car myList) (replace f r (cdr myList))))   ; Element not found
    );cond
    ); lambda
  ); define
```