

Name: _____

CMSC330 Summer 2009

Midterm 2: Ocaml to Objects

Instructions:

Show your work for all problems. If you need extra space, ask for a sheet of paper and label your work (i.e. which problem your work corresponds to). If I cannot read your handwriting I will have trouble giving you points. For two points extra credit, write an interesting fact in the space below.

1. Trace through the following piece of code and show the printed results using (12 pts):

a. Call by value evaluation (where assignments with = are done using object copy):

20123

b. Call by reference evaluation (where assignments with = are done using reference copy):

40143

c. Call by name evaluation (where assignments with = are done using object copy):

30125

Note that the code below is not meant to be any specific language (i.e. Java, C, etc) - assume it is a generic language and evaluate the code using the three options described.

```
void foo(int a, int b) {  
    int z = a;  
    z++;  
    a++;  
    b = b + 2;  
}
```

```
//main code  
int x = 2;  
int[] y = [0,1,2,3];  
foo(x,y[x]);
```

```
print(toS(x)+toS(y[0])+ toS(y[1])+toS(y[2])+ toS(y[3]));  
//toS() is a function that converts the integer to a string  
//the plus (+) above is string concatenation
```

2. Consider the following context free grammar:

$$\begin{array}{l} S \rightarrow SM \mid T \\ T \rightarrow M + T \mid \epsilon \mid a \\ M \rightarrow a \mid b \end{array} \qquad \begin{array}{l} S \rightarrow TS' \\ S' \rightarrow MS' \mid \epsilon \end{array}$$

- a. Is the context free grammar compatible with the top-down recursive-descent parsers we have seen in class? Circle YES or **NO**, and if no, fix the grammar so that it is (you may write/overwrite your answer above). (3 pts)
- b. Is the context free grammar ambiguous? Circle **YES** or NO and explain why. (3 pts)

$$\begin{array}{l} S \rightarrow SM \rightarrow TM \rightarrow M + T \rightarrow a + T \rightarrow a + a \\ S \rightarrow T \rightarrow M + T \rightarrow a + T \rightarrow a + a \end{array}$$

3. Consider the following context free grammar

$$\begin{array}{l} S \rightarrow aAB \mid aB \\ A \rightarrow aA \mid \epsilon \\ B \rightarrow bB \mid b \end{array} \qquad \begin{array}{l} S \rightarrow aABb \\ A \rightarrow aA \mid \epsilon \\ B \rightarrow bB \mid \epsilon \end{array}$$

(or $\begin{array}{l} S \rightarrow aLb \\ L \rightarrow AB \\ A \rightarrow aA \mid \epsilon \\ B \rightarrow bB \mid \epsilon \end{array}$)

- a. What language does the grammar recognize? Your description should be as complete as possible. (2 pts)

1 or more a-s followed by one or more b-s

- b. Is the grammar compatible with a top-down predictive parser? Circle YES or **NO**. If no, rewrite the grammar so that it is (you may write/overwrite your answer above). (3 pts)

4. Consider the following context free grammar

$$\begin{aligned} S &\rightarrow E + S \mid E \\ E &\rightarrow T * E \mid T \\ T &\rightarrow a \mid b \end{aligned}$$

- a. Which operator has higher precedence in the language that this grammar recognizes? (2 pts)

Circle one: + *

- c. Do operators in the language this grammar recognizes bind to the left or bind to the right? (2 pts)

Circle one: left **right**

5. Consider the following context free grammar (20 pts)

$$\begin{aligned} S &\rightarrow E + E \mid d \\ E &\rightarrow (E) \mid aE \mid \varepsilon \end{aligned}$$

Write a function in the language of your choice (Java, C++, Ocaml, Ruby) to parse the grammar that takes a string as input and returns nothing if the parse succeeds, or raises an exception otherwise. You may write multiple functions, and you may use the following `match` function (assume it is translated into the language of your choosing but retains the following signature):

```
String match(char c, String inputString){
    if (inputString.charAt(0) == c)
        return
        inputString.substring(1,inputString.length());
    else throw ParseException;
}
```

Answer to 5 continued on this page

```
void parseInput(String inputString){
    inputString = parseS(inputString);
    if (inputString.length() != 0)    //check for length at end
        throw ParseException;
    else
        return;
}

String parseS(String inputString){
    String lookahead = inputString.substring(0,1);
    if (lookahead == "d")
        inputString = match('d', inputString);
    else if (lookahead == '(' || lookahead == 'a' ||
             lookahead == '+'){        //need to calculate first set
                                        //correctly to include +
        inputString = parseE(inputString);
        inputString = match('+',inputString);
        inputString = parseE(inputString);
    }
    else throw ParseException;        //throw exception if not in first
                                        //set of either production

    return inputString;
}

String parseE(String inputString){
    String lookahead = inputString.substring(0,1);
    if (lookahead == '('){
        inputString = match('(',inputString);
        inputString = parseE(inputString);
        inputString = match(')',inputString);
    }
    else if (lookahead == 'a'){
        inputString = match('a',inputString);
        inputString = parseE(inputString);
    }
    return inputString;
}
```

6.

a. Can the language $a^n b^{2n}$ where $n \geq 0$ be recognized by a regular expression? A context free grammar? If yes give the RE and/or CFG that recognizes the language, otherwise explain why not. You must answer for both REs and CFGs. (4 pts)

No for RE; RE has no memory

$S \rightarrow aSbb \mid \epsilon$

b. Can the language $a^n b^n c^m d^n$ be recognized by a regular expression? A context free grammar? If yes give the RE and/or CFG that recognizes the language, otherwise explain why not. You must answer for both REs and CFGs. (4 pts)

No for RE; RE has no memory

No for CFG - CFG has only limited memory on stack to match the first pair of n things, but it will not be able to count the number of d -s after counting the number of a -s and b -s

7. Explain the difference between a weakly typed language and a strongly typed language. (4 pts)

Weakly typed does not enforce typing rules in that it allows one type to be treated as another or provides many implicit casts. Strongly typed or typesafe catches more errors at compile time and does not allow one type to be treated as another.

8. Explain the difference between static scoping and dynamic scoping in the context of a variable which is not in the current scope. That is, where does a statically scoped language look for the variable binding and where does a dynamically scoped language look for the binding? (4 pts)

Static scope follows the lexical scoping of the program to figure out the bindings of non-local variables. A dynamically scoped language looks for a non-local variable bindings by following the function calls on the run-time stack.

9. Examine the following implementation for the producer-consumer problem:

```
Lock lock = new ReentrantLock();
boolean valueReady = false;
Object value;

void produce(Object o) {
    lock.lock();
    while (valueReady);
    value = o;
    valueReady = true;
    lock.unlock();
}

Object consume() {
    lock.lock();
    while (!valueReady);
    Object o = value;
    valueReady = false;
    lock.unlock();
}
```

- a. Is there a generic program that can statically determine whether or not this code, or any code in general, will result in deadlock? Explain your answer. (2 pts)

No; this is equivalent to the halting problem (replace “deadlock” with “infinite loop”).

- b. Will this code work (that is, not result in deadlock but protect the **value** object)? If yes, explain your answer. If **no**, rewrite the code so that it will work (you can write/overwrite your answer above, using busy-waiting will receive partial credit). (8 pts)

Many options;

```
//synchronize on the value Object in both functions
void produce(Object o) {
    synchronized(value){
        value = o;
    }
}

Object consume() {
    synchronized(value){
        Object o = value;
    }
}

//use busy waiting for partial credit
void produce(Object o) {
    while (true) {
        lock.lock();
        if (!valueReady){
            value = o;
            valueReady = true;
        }
        lock.unlock();
    }
}

Object consume() {
    while (true) {
        lock.lock();
        if (valueReady){
            Object o = value;
            valueReady = false;
        }
        lock.unlock();
    }
}
```

```
//use wait and notify
Lock lock = new ReentrantLock();
Condition ready = lock.newCondition();

void produce(Object o) {
    lock.lock();
    while (valueReady){
        ready.await(); }
    value = o;
    valueReady = true;
    ready.signalAll();
    lock.unlock();
}

Object consume() {
    lock.lock();
    while (!valueReady){
        ready.await(); }
    Object o = value;
    valueReady = false;
    ready.signalAll();
    lock.unlock();
}
```

11. Examine the following type definition:

```
type familyEarningStatus =  
  NoIncome  
  | Single of int  
  | Married of familyEarningStatus * familyEarningStatus
```

- a. Write an Ocaml function called earnings that takes a `familyEarningStatus` as an argument and returns the total amount of money earned by each person in the family summed together, by examining the income associated with each `Single` person as shown in the data structure above. (8 pts)

```
let rec earnings family =  
  match family with  
    NoIncome -> 0 |  
    Single s -> s |  
    Married(m,d) -> (earnings m) + (earnings d)  
;;
```

- b. Write a function `total` that takes a list of `familyEarningStatus` as an argument and calculates the total earnings for all the families in the list. You may use your `earnings` function from the previous question as well as the `fold` and `map` functions you have seen in class/project 3. (7 pts)

```
let total list = (fold earnings 0 list) ;;
```

Extra Credit (1 point).

What has more computational power, a Turing machine with one tape, or a modern laptop with a Intel® 965PM / GM Express Chipset with 4GB Dual Channel 667MHz DDR2 SDRAM?

Turing machine has infinite memory, a computer has finite memory, so the turing machine is more powerful.