Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd

Email: [nalongsone.danddank@my.metrostate.edu](mailto:nalongsone.danddank@my.metrostate.edu)\

**Metropolitan State University**

**ICS-365 Final Exam**

**120 Points**

**Section 1. Fill-in-the-blank. 2 point each, fill in 10 of 12, Total 20 points.**

1. \_\_\_\_Symbolic Logic\_\_\_ requires 3 basic requirements. Ability to express propositions, ability to express relations between propositions, ability to describe how new propositions may be inferred.
2. \_\_\_ Unification\_\_\_ means finding values for variables in propositions that allows the matching process to succeed.
3. \_\_\_\_\_Nonprocedural Programming\_\_\_\_ do not state how a result is to be computed, but rather the form of the result
4. \_\_\_\_\_Logic Programming Languages\_ are generally considered the “do-this, then do that” languages.
5. A \_\_\_\_Horn clause\_\_ has two forms. Where a “\_Headed\_” has a single atomic proposition on left side and a “\_Headless\_ “ has an empty left side.
6. \_\_\_\_\_Virtualization\_\_\_ deals with “extending or replacing” an existing interface so as to mimic the behavior of another system.
7. The right side of a clausal form proposition is called the \_\_\_\_Antecedent\_\_\_\_\_\_\_.
8. Two access methods for implementing Dynamic Scoping are: \_Deep Access, and Shallow Access\_.
9. \_\_Contingent Statement\_\_ is a logical statement that may or may not be true.
10. \_\_\_\_Local variables\_\_\_\_ are variables defined inside subprograms.
11. \_\_\_\_Dynamic link\_\_\_\_ contains information on a subprogram while it is active such as return address, parameters and local variables.
12. When a sub goal fails, the prolog system traces its steps backward to the previous goal to

try and satisfy it. This is called \_\_\_\_\_backtracking\_\_

**Some of the answers may be found in the following vocabulary word pool:**

**clause, horn-clause, class file, logic/declarative, consequent, headed, virtualization, predicate, predicate-calculus, unification, headless, antecedent, procedural, backtracking, resolution, backtracking, consequent, proposition, local, activation record, dynamic link, recursion,** **Section 2, Multiple Choice/True False. (Select 20 of 23, 2 points each) total: 40 points.**

1. What is the correct order of the compilation process?
   1. Syntax Analysis then Lexical Analysis
   2. Lexical Analysis then Syntax Analysis
   3. Code Generation, then Syntax, then Lexical analysis
   4. None of the above
2. Imagine if I can do the following. <Register>=<Memory Location>. I can also do the following. <Memory Location> =<Register>. This can be considered:
   1. Vertical Integration
   2. Equilateral
   3. Orthogonal
   4. LRU
3. BNF grammars can capture and/or enforce some features in a language that Attribute grammars cannot.
   1. TRUE
   2. FALSE
4. In an attribute grammar, a node's synthesized attributes are based on the values associated with that node's:
   1. descendents
   2. ancestors
   3. friend class
   4. none of the above.
5. If the grammar for a language is ambiguous:
   1. Some valid expression in that language has more than one parse tree.
   2. There are zero parse trees because the grammar is ambiguous
   3. There are no valid expressions because the grammar is ambiguous.
   4. None of the above
6. What are Predicate Functions?
   1. They state the static semantic rules for a language and are associated with grammar rules.
   2. They state the grammar rules of a language and are associated with semantic rules.
   3. They are only applicable in human languages, not computer languages.
   4. There is no such thing as a predicate function.
7. Chose a language BNF or EBNF was used to define
   1. ALGOL
   2. FORTRAN
   3. 8080 Assembly
   4. Early pseudo code
8. What generally produces slower execution
   1. A purely compiled language
   2. A purely interpreted language
   3. Purse assembly/machine langague
   4. None of the above
9. What is the output of this following program?

#include <stdio.h>

int main(int argc, char \* argv[])

{

int i[2]={5,6};

int \*i\_ptr;

i\_ptr=&i[1];

(\*i\_ptr)++;

printf("%d ",\*i\_ptr);

i\_ptr--;

printf("%d \n",\*i\_ptr);

}

1. 5 6
2. 6 6
3. 6 7
4. 7 5
5. Which is a primary reason pseudo codes were invented? (circle all that apply)
   1. To simplify the difficult task of programming
   2. To make automatic fixed point arithmetic
   3. To automate indexing
   4. To completely eliminate machine language from computers
6. A variable binding is dynamic if
   1. Occurs during execution, or can change during execution
   2. Occurs before run time and is changed throughout execution
   3. Gives you a shock if you look at it.
   4. None of the above.
7. What is returned by the following scheme expression: (+ 3 ( \* 3 3))
   1. 12
   2. 15
   3. 20
   4. none of the above
8. In scheme, the function that gives a new list with a given CAR value and CDR list?
   1. BLD
   2. COM
   3. CONS
   4. None of the above
9. In scheme, the function compares data structures to determine if they have congruent structure?
   1. EQ
   2. EVEN
   3. EQV
   4. None of the above
10. What is an Associative Array?
    1. A type whose variables may store different data type values at different times.
    2. A data type in Python that can store different data types and is immutable
    3. A homogeneous aggregate of data elements in which an individual element is identified by its position in the aggregate.
    4. An unordered collection of data elements that are indexed by an equal number of values called keys.
11. General semantics of calls to a subprogram include?
    1. Parameter passing methods
    2. Save the execution status of calling program
    3. Transfer of control and arrange for the return
    4. All of the above
12. A base class contains all data and methods which are common to all objects in its inheritance hierarchy.
    1. TRUE
    2. FALSE
13. Which kind of variable generally has the longest lifetime?
    1. A heap-dynamic variable
    2. A stack-dynamic variable
    3. Both always have exactly the same lifetime
    4. Neither has any lifetime.
14. A variable in Prolog must start with an upper-case V (or an underscore)

a. TRUE

b. FALSE

1. If Prolog term A unifies with term B, and term A unifies with term C, then term B always unifies with term C.
   * 1. TRUE
     2. FALSE
2. These programming languages allow for nested subprogram: Python, C, Lua and Java?
   1. TRUE
   2. FALSE
3. Which of the following describes pass by reference?
   1. Actual value is used to initialize the corresponding formal parameter.
   2. Access path to value, usually an address, is passed to the subprogram
   3. Value is passed with its name to the subprogram.
   4. None of the above.
4. Subprograms cannot be used as parameters in any programming languages?
   1. TRUE
   2. FALSE

**Section 3. Problems. 60 Points. Pick 6. Skip problem you do not wish graded.**

**Problem 1. 10 Points.**

Consider the grammar below:

<S> → a <S> c <B> | <A>

<A> → c <A> | c

<B> → d

What are the nonterminal symbols?

<S>, <B>, <A>

What are the terminal symbols?

a, c, d

Which of the following sentences are in the language generated by this

grammar?

a. abcd

b. accd

c. acccbcc

d. acccbd

e. ccc

**Problem 2. Static and dynamic scope (10 pts)**

**K is a new programming language that is exactly like C, except it uses dynamic scoping** for determining the binding of non-local variables rather than static scoping.

Consider the program to the right.

What will it print when run when it is:

(a) Compiled with a normal C compiler?

200

100

101

200

200

101

(b) Compiled with your new K compiler?

200

200

201

201

201

100

**#include <stdio.h>**

**int i = 100;**

**void g(void)**

**{**

**printf("%d\n", i);**

**i++;**

**printf("%d\n", i);**

**}**

**int f(void)**

**{**

**int i=200;**

**printf("%d\n", i);**

**g();**

**printf("%d\n", i);**

**return(i);**

**}**

**int main(int argc,char \*\*argv)**

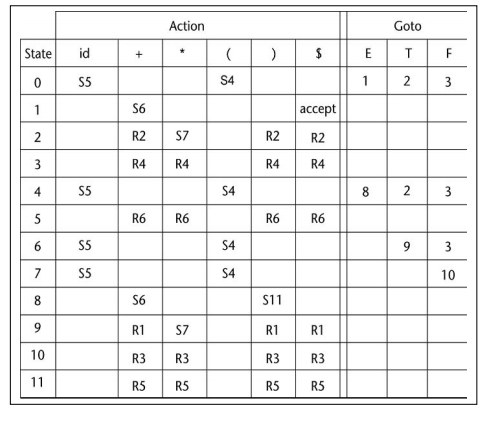
**{**

**printf("%d\n", f());**

**printf(“%d\n”, i);**

**}**

**Problem 3, LR Parser (10 points)**



Consider an LR parser for the following simple express grammar, producing the table to the right.

1: E -> E+T

2: E -> T

3: T -> T\*F

4: T -> F

5: F -> (E)

6: F -> id

1. (1pt) What does an empty cell mean?

means, the processing is wrong.

1. (1pt) What symbol marks the end of the input?

$

1. (1pt) Can this table be used to parse an input string of any length?

No.

1. (7 pts) Complete the **first Eight steps** of a Shift-Reduce table (using LR parser), parsing the input string (id+id)\*id. The first line below is the starting point (stack is zero) and not considered a step:

Stack Input Action

1. (id+id)\*id$ S4

0(4 id + id)\*id$ S5

0(4id5 + id)\*id$ R6 [0, F]

0(4F3 + id)\*id$ R4 [0, T]

0T2 + id)\*id$ R2 [0, E]

0E1 + id)\*id$ S6

0E1+6 id)\*id$ S5

0E1+6id5 )\*id$ R6 [6, F]

0E1+6F3 )\*id$ R4[6, T]

0E1+6T9 )\*id$ R1 []

**Problem 4, Scope. (10 points)** Show the stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal program. Assume problem builds and executes as described.

Calling sequence for this program for execution to reach D is:

Bigsub calls A

A calls B

B calls C

C calls D

BIGBUB is at level 1.

**procedure** Bigsub **is**

**procedure** A(Flag : Boolean) **is**

v,w : Integer

**procedure** B **is**

u,y : Integer

...

**C;**

**end**; -- of B

**B;**

...

**end**; -- of A

**procedure** C **is**

u,w,y : Integer;

**procedure** D **is**

t,z : Integer

Position 1

...

z = t\*u;

...

**end**; -- of D

...

**D;**

**end**; -- of C

**begin**

z,y : Integer

...

**A**(**true**);

...

**end**; -- of BigSub

**Problem 5, Scheme and Prolog. (10 points total)**

**Part 1. (4 points) Write a small scheme function to subtract two numbers.**

For example, if I type (minus 10 5) I should see 5

If I type (minus 5 5) I should see 0.

Solution:

(define (minus a b)

(- a b))

**Part 2.**

**A. (2 points) Write the following English statements as prolog facts:**

Charlie studies ICS365.

studies(Charlie, ICS365).

Mike studies ICS365.

studies(Mike , ICS365).

Deb studies ICS463

studies(Deb , ICS463).

Nancy studies ICS463.

studies(Nancy , ICS463).

Dorf teaches ICS365.

teaches (Dorf , ICS365).

Stein teaches ICS463.

teaches (Stein , ICS463).

**B. (4 points):**

Write the rule, in prolog, to determine who Deb’s teacher is.

teacherOfDeb(X) :- teaches(X, C), studies(Deb, C).

Write the rule, in prolog, to determine who Dorf teaches.

studentOfDorf(Y) :- studies(Y, C), teaches(Dorf, C).

**Problem 6, Associativity and Precedence. (10 points)**

**Assume the following rules of associativity and precedence for expressions:**

**Precedence:** Highest Parentheses first

^ power of

\*, /, not

+, –, mod

=, /=, <, <=, >=, >

Lowest or, xor, and

**Associativity:** Left to right

**Assume precedence at each level moves left to right.**

Show the order of evaluation of the following expressions by parenthesizing all subexpressions and placing a superscript on the right parenthesis to indicate order. For example, for the expression **a + b \* c + d**, the order of evaluation would be represented as **((a + (b \* c)1 )2 + d)3**

1. f + e \* 2 ^ d

(f + (e \* (2 ^ d)1)2)3

1. a >= b xor z or d ^17

((a >= b)2 xor (z or (d ^ 17)1)3)4

1. z \* b – c ^ d and e + f

(((z \* b)2 – (c ^ d)1)4 and (e + f)3)5

**Problem 7, Virtual Machines (10 points).**

Java and other programming languages have taken the approach of executing on a virtual machine, as opposed to compiling to native machine code. Explain one advantage and one disadvantage of the virtual machine approach. This is a 10 point question, so your answer

has to be complete and more than two sentences for each part.

Answer:

- One advantage is greater portability, like when we write code in the Window system environment, then we compile and test in window all correct and work well, but when another programmer take the code go to compile, run or test in another operating system environment like Mac OS or Linux. The code would not work correctly. So like Java programming language can avoid this situation. And it also have another advantages like greater security and resource control by building restriction into the VM, more uniform execution environment, and the byte-code programs will be smaller than machine language programs.

- One disadvantage is less efficient execution than in compiling to machine code, the uniform execution environment is likely to be the least common denominator.