

CS328, Fall 1999, Test 2

Time 75 min. Use extra paper as needed, but make sure to identify each answer.

YOU MUST RETURN THIS PAGE. NAME _____

- 1 In the semester project, make the following changes (modify the grammar). You may show the modifications separately for a) and b), or in one modified grammar - but make sure you identify which way it is.

a=10pc) allow additional global variables placed just before the first begin - these variables are optional and are listed separated by commas. For example:

```
program
int x,y,z;
begin {etc.}
```

b=25pc +5 for full specs,) we also allow functions. Each function takes any number of arguments and always returns an integer. Parameter list is C-like, and the return statement (followed by an expression) is mandatory. Functions can be called in place of any expression. Functions are defined with a block, which is the same as the begin-end block in our grammar. Functions are defined separately, as in C but before the main program. Assume we have a multi-pass compiler so that prototypes is not an issue. The following is an example program

```
int sum(int a, b) {paramater list is same as for globals, use void if none -
for only 25 pc}
begin
    int c;
    return a+b;
end;
program
int x,y;
begin
    readI(x);
    readI(y);
    z:=10+sum(x,y+1);
    writeI(z);
end.
```

2 Given the program (like Pascal, with nested functions)

```

int x, y; {global}
function A
begin
  int z; {z is in A}
  function B {B is defined inside A}
  begin
    int x; {x is in B}
    function C {C is inside B}
    begin
      int x; {x is in C}
      z:=10;
      x:=100;
      y:=1000;
      C(); {recursive call on C}
    end; {end of C}
    C();
  end; {end of B}
  B(); { A calls B}
end; {end of A}

```

a=10pc show ARs for A, B, C (show only local data, static and dynamic link)

b=5pc What is in the persistent data space?

c=10pc assume call to A() is done somewhere. Then, show that contents of the stack during the 3rd recursive call to C(). Make sure to fill out all info on the stack.

d=10pc How would the left-hand-side variables in C() be translated by the compiler (explain in words, separately for each variable).

Note: do not confuse this problem with our project - we have no ARs in our project.

3 Given the grammar, with lower case terminals and upper case nonterminals, and S the initial nonterminal

```

S -> aS | aA | bA
A -> Aa | ABb | cA
B -> bB | bdB | c

```

a=10pc remove left recursion as needed, showing the modified grammar

b=10pc left-factorize as needed what you get in a), showing the new grammar

c=10pc Is the resulting grammar LL(1)? Argue piece by piece why it is or why it is not, showing all (and only the necessary) First and Follow sets.