

Final

Name: _____

Notes

Answer all parts of all 5 questions. Put your answers on these pages. If you want to hand in scratch work, you may; make sure your name is on each page. If you need to make an assumption to clarify a problem, *write your assumption down*.

1. LR parsinga) Is LR parsing **top-down** or **bottom-up**? (circle one)

b) Consider this grammar:

- 1 Start \rightarrow Stm
- 2 Stm \rightarrow **if** E **then** Stm
- 3 Stm \rightarrow id = id
- 4 E \rightarrow id
- 5 E \rightarrow intLit

List all the **non-terminal** symbols in this grammar.

c) Show the first 5 states (numbered 0 through 4) of the characteristic finite-state machine for an LR(0) parser for the grammar in b). Please develop the states in depth-first order. Leave the **goto** field blank in items for which you have not developed successor states. Mark the **goto** field of a reducible item with **X**.

d) Is this grammar ambiguous? (circle one of: **yes no**).

2. Symbol tables

a) If you were not allowed to use hashing for symbol tables, **what data structure** would you use instead? **What is the computational complexity** of insertion in your data structure?

b) Your compiler for CSX_go uses block-structured symbol tables. The four phases of your compiler are (1) scanning, (2) parsing, (3) semantic checking, and (4) code generation. Which of these phases should invoke `closeScope()`? Your answer might include more than one phase. Circle your answer(s).

c) What information does a CSX_go compiler need to place in the symbol-table entry for a local variable? Consider all relevant phases of the compiler.

d) Consider enhancing CSX_go to allow overloaded methods that differ with respect to their signature. How would this feature change the way you store information in the symbol table?

3. Semantic checking

a) In CSX_go, what rules govern whether a right-hand side is type-compatible with the left-hand side of an assignment? Try to be complete.

b) If one branch of an `IfNode` terminates normally, but the other branch does not terminate normally, should the semantic checker mark the entire `IfNode` as terminating normally? (Circle **yes** or **no**).

c) Besides reachability/termination checking, what other **two** kinds of semantic checking are

necessary for Java?

d) List **all the semantic checks** a Java compiler needs to make on a call like this: `a.m(foo)`, where `a` is declared to be a variable of class `A`.

4. The Java Virtual Machine

a) What are the two major components of a Java class file?

c) Hand-generate Java bytecode for the following CSX_{go} code fragment. All variables are **int**. The package is called **pac**. Variables `a`, `b`, and `c` are local; variable `d` is global. Function `foo` returns an **int**. Don't worry about the exact spelling of the bytecode operations. Write down any assumptions you want about register numbers. Assume the code fragment passes all compile-time semantic checks.

```
for a >= 0 {
    a = 1 + foo(b + c * d);
    d = a;
}
```

5. Extensions

Consider modifying CSX_go to include a **switch** statement. Assume that **switch** statements contain no **default** cases.

(a) What new tokens you would need?

(b) Show all new productions you would add to your CFG. Don't include any semantic actions.

(c) What new AST nodes would you need? Give their names and describe their children.

(d) Enumerate any new semantic checking you would need.