

CSE 401 - LL Semantics, Semantics, Type Checking, & Vtables

1. Edit the following Grammars to make them LL(1). Then walk through the top down parse for the string given in the parenthesis.

Grammar 1 (“azx”)

- 0. $S ::= a B \mid a w$
- 1. $B ::= C x \mid y$
- 2. $C ::= \epsilon \mid z$

Grammar 2 (“ax”)

- 0. $S ::= a B$
- 1. $B ::= C x \mid y$
- 2. $C ::= \epsilon \mid x$

Grammar 3 (“azx”)

- 0. $S ::= S B \mid a \mid w$
- 1. $B ::= C x \mid y$
- 2. $C ::= \epsilon \mid z$

Grammar 4 (“axzw”)

- 0. $S ::= B w \mid a B$
- 1. $B ::= C w \mid x$
- 2. $C ::= B z \mid \epsilon$

2. Suppose we have the following global scope:

```
class Bar { boolean field; public int method(int i, int j); }
class Foo extends Bar { int val; public boolean whoop(int x); }
```

Now, consider the following hypothetical method definition for `Bar.method`:

```
public int method(int i, int j) {
    int r;
    boolean b;
    Foo o;
    if (this.field) {
        o = this;
        b = o.whoop(i + j);
        r = o.val;
    } else {
        r = i * j + 3;
    }
    return r;
}
```

- What variables (locals, parameters, etc.) are defined in the *local* scope in the `method` body?
- When we execute this method body, a runtime error could result. Explain how something could go wrong by giving values of the parameters and/or variables involved that would cause a runtime error.
- The method body also has type errors. Can you describe which type check(s) the compiler could use to deduce this fact?
- Does *every* possible execution of this method produce a runtime error? Can you describe any that happen to be statically correct? (Again, possible runtime values for parameters/variables would suffice.)
- Suppose that we replaced the use of `this.field` in the method body to call a boolean method that always returns false. How would this change your answers to the previous questions?