Lab Six

Alex Smith

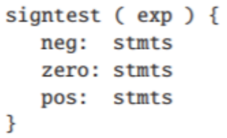
alex.smith1@Marist.edu

April 3, 2019

1 Crafting a Compiler

1.1 Exercise 9.2

Assume that we add a new kind of conditional statement to C or Java, the signtest. Its structure is:

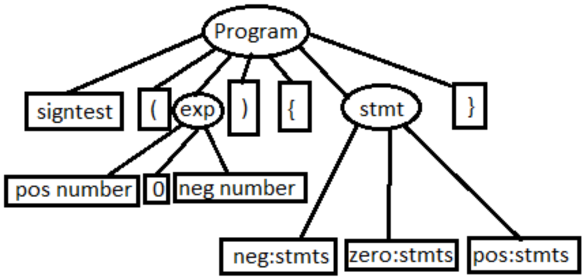


```
signtest ( exp ) {
   neg:  stmts
   zero: stmts
   pos:  stmts
}
```

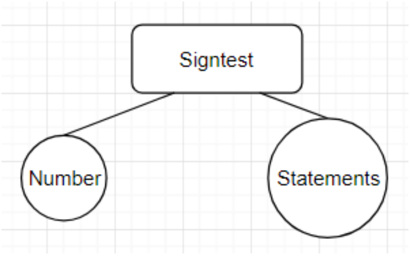
The integer expression exp is evaluated. If it is negative, the statements following neg are executed. If it is zero, the statements following zero are executed. If it is positive, the statements following pos are executed. Show the AST you would use for this construct. Revise the semantic analysis, reachability, and throws visitors for if statements (Section 9.1.2) to handle the signtest.

Left Most Derivation:
<program> → signtest (<exp>) { <stmt>}
< *exp* >⟶neg number
→pos number
→zero
< *stmt* >→neg : stmts
→zero : stmts
→pos : stmts

CST:



AST:



Semantic analysis:

```
void checkInt(int c){
boolean hasError = false;
if (c.type != int){
System.out.println("Error: expression " + c + " is not an int at " + c.lineNum);
hasError = true;
}
}
```

Reachability analysis:

```
void reachTest(int c){
boolean visitPos = false;
boolean visitZero = false;
boolean visitNeg = false;
if( c > 0){
visitPos = true;
}
else if ( c == 0) {
visitZero = true;
}
else{
visitNeg = true;
}
}
```

Throw Visitors:

```
void throwVisitor(signtestNode n){
```

```
gatherThrows(n);
/*This method is detailed in the text and is used for all keywords. The only change required is to change the
input to a Signtest. */
```