CS 241 - Week 10 Tutorial

Semantic Analysis

Spring 2015

Summary

- Error Detection

Error Detection

For each WLP4 program below, point out the error in the program and state whether it is a syntax error (i.e. something the parser would catch) or a semantic error (something semantic analysis would catch).

```
int wain(int x, int y) {
  int a = 100;
  int y = 0; // initialize y
  y = a*x;
  return y;
}
```

```
int wain(int* a, int n) {
  // loop to get the last index
  while (idx < n) {
    idx = idx + 1;
  }
  return *(a + idx);
}
```

```
int wain(int a, int b) {
  int *c = NULL;
  c = &a;
  int *d = NULL;
  d = &b;
  return (c - d);
}
```

```
int foo(int x, int y);

int wain(int x, int y) {
  int a = 0;
  int b = 1;
  x = x * foo(a,b);
  return y;
}
```

For each **C** program below, point out the error in the program and state whether it is a syntax error (i.e. something the parser would catch) or a semantic error (something semantic analysis would catch).

```
float triple(float a) {
  return a * 3.0;
}

int main() {
  int* x, y;
  int a, b;

  a = triple(4.4);
  x = &a;
  y = &b;
  b = *x;
  return *y;
}
```

```
int main() {
    double a = 2.0 * .4 / getRandom();
    int b;
    b = 2;
    return b;
}
```

- In the MIPS assembler you wrote for Assignments 3 and 4, you had to check for duplicate labels in one pass and check for missing labels in a second pass. Why is this not necessary in the WLP4 compiler?

Type Errors

Determine if each of the following WLP4 code fragments is well-typed.

```
int foo(int x, int y){
  return x + 7 * y + 1;
}
...
int a = 0;
int b = 0;
int* c = NULL;
int* d = NULL;

(1) *(d+(((c-&b)+d)-(c+(a*b))))=(c-d+*new int[d+b-c]);
(2) if(*(c+a%b)<(&a-&b)){println(&*&*c-(&b));}else{delete[]*d+&a-c;}
(3) a * foo(a,(&*c) + 1)
```