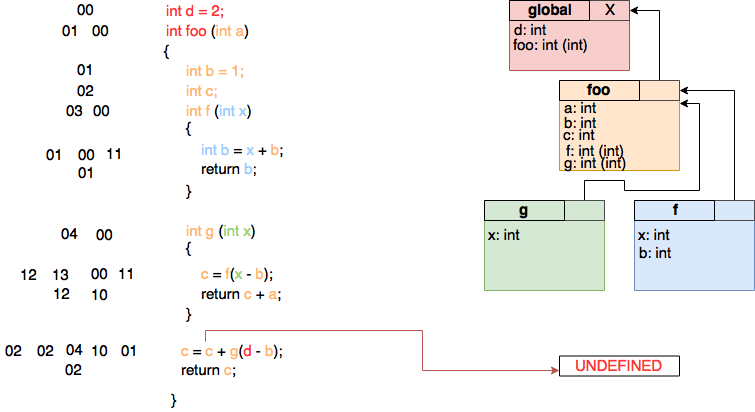
Assignment Series 4  
  
Semantic Analysis  
Andrea & Aynel

**Assignment 15: Scoping and symbol tables**

****

**Assignment 16: Lambda lifting**

**Original:**

int d = 2;

int foo ( int a)

{

int b = 1;

int c;

int f( int x) {

int b = x + b;

return b;

}

int g( int x) {

c = f( x - b);

return c + a;

}

c = c + g( d - b);

return c;

}

**New:**

int d = 2;

int f\_inner (int x, int b) {

int h = x + b;

return h;

}

int g\_inner ( int x, int b, int \*c, int a) {

\*c = f\_inner( x - b);

return \*c + a;

}

int foo (int a)

{

int b = 1;

int c;

c = c + g\_inner( d – b, b, &c, a);

return c;

}

**Assignment 17: Function overloading**[*Assume we would extend CiviC by function overloading.*

*Function overloading supports the presence of multiple functions within the same scope bearing the same name, as long as they are distinguished by different arity (number of parameters) or by different parameter types. Function calls are dispatched to the matching function definition according to the argument types inferred.*

*Describe how this extension would affect semantic analysis in the CiviC compiler in general and how you would solve the corresponding problems in detail*.]