

Lecture 19

CIS 341: COMPILERS

Announcements

- Project 5 Compiling objects in full Oat
 - Available from the course web pages
 - Updated oat.pdf fixes a few typos (mentioned on Piazza)
 - Due April 8th
- Final Exam:
 - Tuesday, April 30th noon-2:00 pm
 - Moore 216

See oat.pdf (Project 5 version)

TYPECHECKING CLASSES

A high-level tour of a variety of optimizations.

OPTIMIZATIONS

Optimizations

- The code generated by our OAT compiler so far is pretty inefficient.
 - Lots of redundant moves.
 - Lots of unnecessary arithmetic instructions.
- Consider this OAT / C program:

```
int foo(int w) {  
    int x = 3 + 5;  
    int y = x * w;  
    int z = y - 0;  
    return z * 4;  
}
```

- See example.oat, example.ll, example.s, example-opt.s, example-hand.s

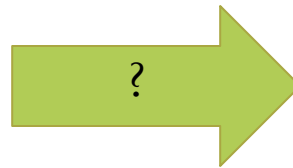
Unoptimized vs. Optimized Output

```
.globl _foo
_foo:
    pushl %ebp
    movl %esp, %ebp
    subl $64, %esp

__fresh2:
    leal -64(%ebp), %eax
    movl %eax, -48(%ebp)
    movl 8(%ebp), %eax
    movl %eax, %ecx
    movl -48(%ebp), %eax
    movl %ecx, (%eax)
    movl $3, %eax
    movl %eax, -44(%ebp)
    movl $5, %eax
    movl %eax, %ecx
    addl %ecx, -44(%ebp)
    leal -60(%ebp), %eax
    movl %eax, -40(%ebp)
    movl -44(%ebp), %eax
    movl %eax, %ecx
    movl -40(%ebp), %eax
    movl %ecx, (%eax)
    movl -40(%ebp), %eax
    movl (%eax), %ecx
    movl %ecx, -36(%ebp)
    movl -48(%ebp), %eax
    movl (%eax), %ecx
    movl %ecx, -32(%ebp)
    movl -36(%ebp), %eax
    movl %eax, -28(%ebp)
    movl -32(%ebp), %eax
    movl %eax, %ecx
    movl -28(%ebp), %eax
    imull %ecx, %eax
    movl %eax, -28(%ebp)
    leal -56(%ebp), %eax
    movl %eax, -24(%ebp)
    movl -28(%ebp), %eax
    movl %eax, %ecx
    movl -24(%ebp), %eax
    movl %ecx, (%eax)
    movl -24(%ebp), %eax
    movl (%eax), %ecx
    movl %ecx, -20(%ebp)
    movl -20(%ebp), %eax
    movl %eax, -16(%ebp)
    movl $0, %eax
    movl %eax, %ecx
    subl %ecx, -16(%ebp)
    leal -52(%ebp), %eax
    movl %eax, -12(%ebp)
    movl -16(%ebp), %eax
    movl %eax, %ecx
    movl -12(%ebp), %eax
    movl %ecx, (%eax)
    movl -12(%ebp), %eax
    movl (%eax), %ecx
    movl %ecx, -8(%ebp)
    movl -8(%ebp), %eax
    movl %eax, -4(%ebp)
    movl $4, %eax
    movl %eax, %ecx
    movl -4(%ebp), %eax
    imull %ecx, %eax
    movl %eax, -4(%ebp)
    movl -4(%ebp), %eax
    movl %ebp, %esp
    popl %ebp
    ret
```

Hand optimized code:

```
__fun__foo:
    movl -4(%esp), %eax
    shl $5, %eax
    ret
```



- Function foo may be inlined by the compiler, so it can be implemented by just one instruction!