CSE443 Compilers

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall

http://www.cse.buffalo.edu/faculty/alphonce/SP17/CSE443/index.php
https://piazza.com/class/iyn4ndqa1s3ei
Phases of a compiler

Target machine code generation

Figure 1.6, page 5 of text
# Plan for ROS

<table>
<thead>
<tr>
<th></th>
<th>M</th>
<th>W</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5/1</strong></td>
<td>Peephole optimization</td>
<td><strong>5/3</strong></td>
<td>Misc. topics (not for project)</td>
</tr>
<tr>
<td><strong>5/8</strong></td>
<td>Misc. topics (not for project)</td>
<td><strong>5/10</strong></td>
<td>Who</td>
</tr>
<tr>
<td></td>
<td>PROJECT Q's</td>
<td></td>
<td>5/12</td>
</tr>
</tbody>
</table>
Final Exam

5/19/2017, Friday

8:00AM - 11:00AM

Talbert 107
register descriptor

"For each available register, a register descriptor keeps track of the variables names whose current value is in that register." [p. 543]

address descriptor

"For each program variable, an address descriptor keeps track of the location or locations where the current value of that variable can be found." [p. 543]
getReg function: \( x = y \text{ op } z \)

1. If \( y \) is currently in a register, pick a register already containing \( y \) as \( R_y \).

2. If \( y \) is not in a register, but there is a register currently empty, pick one such register as \( R_y \). \([LD \ R_y, \ y]\)

3. If \( y \) is not in a register, and there is no empty register:

   let \( R \) be a candidate register, and suppose \( v \) is one of the variables that the register descriptor for \( R \) says is in \( R \):

   (a) If \( v \) is somewhere besides \( R \), then OK.
   (b) If \( v \) is \( x \) and not an operand, then OK.
   (c) If \( v \) is not used later, then OK.
   (d) Generate \( ST \ v, \ R \) to copy \( v \) to memory.

Repeat (a) - (d) for each \( v \) in \( R \). The score of \( R \) is the number of \( ST \) instructions generated. Choose the \( R \) with lowest score.
getReg function: \( x = y \text{ op } z \)

Need a register for the result, \( Rx \). Since a new value of \( x \) is computed, a register that holds only \( x \) is always an acceptable choice for \( Rx \).

If \( y \) is not used after \( I \), and \( Ry/Rz \) holds only \( y/z \) after being loaded, if necessary then \( Ry/Rz \) can also be used as \( Rx \).

getReg function: \( x = y \)

For copy instruction, always choose \( Rx = Ry \).
Peephole optimization

"The peephole is a small, sliding window on a program." [p. 549]

"In general, repeated passes over the target code are necessary to get the maximum benefit." [p. 550]
Peephole optimization: redundant LD/ST

LD R0, a
ST a, R0

If the ST instruction has a label, cannot remove it. (If instructions are in the same block we're OK.)
Peephole optimization: unreachable code

if E=K goto L1
goto L2
L1: ...
... ...
L2: ...
... ...

Suppose K is a constant.
if E=K goto L1
goto L2
L1: ...do something...
...
L2: ...do something...
...

Eliminate jumps over jumps

Peephole optimization: unreachable code
Peephole optimization: unreachable code

if \( E=K \) goto L1

\[ \text{L1: ...} \]

\[ \text{L2: ...} \]

goto L2

\[ \text{if } E\neq K \text{ goto L2} \]

L1: ...

\[ \text{L2: ...} \]

Eliminate jumps over jumps
Peephole optimization: unreachable code

If there are no jumps to L1, we can remove label.
Peephole optimization: unreachable code

If $E$ is set to a constant value other than $K$, then...

```plaintext
if $E = K$ goto L1
goto L2
L1: ...
...
L2: ...
```

```plaintext
if $E \neq K$ goto L2
...
...
L2: ...
```
Peephole optimization: unreachable code

- If $E = K$ goto L1
- goto L2

L1: ...
... ...
L2: ...
...

- If true goto L2

L2: ...
...

...conditional jump becomes unconditional...
Peephole optimization: unreachable code

if E=K goto L1

goto L2

L1: ...
...
L2: ...
...

goto L2
...
L2: ...
...

...and the unreachable code can be removed.
Peephole optimization: flow-of-control

goto L1
...
L1: goto L2
...
L2:
Peephole optimization: flow-of-control

goto L1
...
L1: goto L2
...
L2:

goto L2
...
L1: goto L2
...
L2:
Peephole optimization: flow-of-control

If there are no jumps to L1, and L1 is preceded by an unconditional jump...
Peephole optimization: flow-of-control

```
goto L1
...
L1: goto L2
...
L2:
```

```
goto L2
...
L2:
```

...then we can eliminate the statement labelled L1
Peephole optimization: flow-of-control

if a < b goto L1
...
L1: goto L2
...
L2:

if a < b goto L2
...
...
L2:

...similar arguments can be made for conditional jumps.
Project questions?