Global Register Allocation
drawbacks of local register allocation

• Why do we have to store all live/dirty registers at the end of a basic block?
  • Because we only consider per-basic block register allocation, and information may not match across basic blocks

• Consider the CFG

• In BB1, $x$ is mapped to $r_1$; in BB2, $x$ is mapped to $r_2$
  • What should $x$ be mapped to in BB3?
global register allocation

• To make sure that a temporary/local/global has consistent mapping across basic blocks, we want to assign that variable to a register for the entire function.
  • Isn’t this kind of like our naïve register allocation approach?
  • Key: a register might have multiple variables assigned to it.
  • All variables with the same color can be assigned to the same register in this code.

1: \( T1 = A + B \)
2: \( T2 = A + T1 \)
3: \( T3 = A + T2 \)
4: \( D = A + T3 \)
5: \( T4 = C + B \)
6: \( T5 = T4 + C \)
7: \( E = T5 + D \)
global register allocation

• Issues:
  • How do we know that two variables can be assigned to the same register?
  • How do we find the right assignment of variables to registers?
  • What do we do if we don’t have enough registers to make the assignment?

1: \[ T_1 = A + B \]
2: \[ T_2 = A + T_1 \]
3: \[ T_3 = A + T_2 \]
4: \[ D = A + T_3 \]
5: \[ T_4 = C + B \]
6: \[ T_5 = T_4 + C \]
7: \[ E = T_5 + D \]
co-locating variables

- Two variables can be assigned to the same register if they are not live at the same time
- They don’t have values we need at the same time

1: \[ T1 = A + B \]
2: \[ T2 = A + T1 \]
3: \[ T3 = A + T2 \]
4: \[ D = A + T3 \]
5: \[ T4 = C + B \]
6: \[ T5 = T4 + C \]
7: \[ E = T5 + D \]
Two variables can be assigned to the same register *if they are not live at the same time*. They don’t have values we need at the same time.

<table>
<thead>
<tr>
<th>Equation</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 = A + B</td>
<td>[A, B]</td>
</tr>
<tr>
<td>T2 = A + T1</td>
<td>[A, B, T1]</td>
</tr>
<tr>
<td>T3 = A + T2</td>
<td>[A, B, T2]</td>
</tr>
<tr>
<td>D = A + T3</td>
<td>[A, B, T3]</td>
</tr>
<tr>
<td>T4 = C + B</td>
<td>[B, C, D]</td>
</tr>
<tr>
<td>T5 = T4 + C</td>
<td>[T4, C, D]</td>
</tr>
<tr>
<td>E = T5 + D</td>
<td>[T5, D]</td>
</tr>
<tr>
<td>E</td>
<td>[E]</td>
</tr>
</tbody>
</table>
co-locating variables

• Just because you make sure that variables that are not live at the same time do not go in the same register doesn’t mean you make the right assignments

1:  \( T1 = A + B \) [A, B]
2:  \( T2 = A + T1 \) [A, B, T1]
3:  \( T3 = A + T2 \) [A, B, T2]
4:  \( D = A + T3 \) [A, B, T3]
5:  \( T4 = C + B \) [B, C, D]
6:  \( T5 = T4 + C \) [T4, C, D]
7:  \( E = T5 + D \) [T5, D]
making the right assignments

• Just because you make sure that variables that are not live at the same time do not go in the same register doesn’t mean you make the right assignments

• If we put all the temporaries in the same register, then we need an extra register for C

1: \( T1 = A + B \) \([A, B]\)
2: \( T2 = A + T1 \) \([A, B, T1]\)
3: \( T3 = A + T2 \) \([A, B, T2]\)
4: \( D = A + T3 \) \([A, B, T3]\)
5: \( T4 = C + B \) \([B, C, D]\)
6: \( T5 = T4 + C \) \([T4, C, D]\)
7: \( E = T5 + D \) \([T5, D]\)
next: graph coloring