CSc 453

Compilers and Systems Software

15 : Intermediate Code III

Department of Computer Science
University of Arizona

collberg@gmail.com

Copyright © 2009 Christian Collberg
Basic Blocks and Flow Graphs
Control Flow Graphs

- We divide the intermediate code of each procedure into basic blocks. A basic block is a piece of straight line code, i.e. there are no jumps in or out of the middle of a block.

- The basic blocks within one procedure are organized as a (control) flow graph, or CFG. A flow-graph has
  - basic blocks $B_1 \cdots B_n$ as nodes,
  - a directed edge $B_1 \rightarrow B_2$ if control can flow from $B_1$ to $B_2$.
  - Special nodes ENTER and EXIT that are the source and sink of the graph.

- Inside each basic block can be any of the IRs we’ve seen: tuples, trees, DAGs, etc.
x := a * 5
y := Z[x]
a := a + 1

if ... goto B2

if ... goto B3

if ... goto B6

goto B2

B6

ENTER

B1

B2

B3

B4

B5

Sink node

Source node

Straight line code

Basic Block

If-Statement

Loop

x := a * 5
y := Z[x]
a := a + 1

if ... goto B2

if ... goto B3

if ... goto B6

goto B2

EXIT
Source Code:

\[
\begin{align*}
X & := 20; \quad \text{WHILE } X < 10 \quad \text{DO} \\
& \quad X := X - 1; \quad A[X] := 10; \\
& \quad \text{IF } X = 4 \quad \text{THEN} \quad X := X - 2; \quad \text{ENDIF}; \\
& \quad \text{ENDDO}; \quad Y := X + 5;
\end{align*}
\]

Intermediate Code:

\[
\begin{align*}
(1) & \quad X := 20 & \quad (5) & \quad \text{if } X<>4 \quad \text{goto (7)} \\
(2) & \quad \text{if } X>=10 \quad \text{goto (8)} & \quad (6) & \quad X := X-2 \\
(3) & \quad X := X-1 & \quad (7) & \quad \text{goto (2)} \\
(4) & \quad A[X] := 10 & \quad (8) & \quad Y := X+5
\end{align*}
\]
Control Flow Graphs...

Flow Graph:

ENTER → B1

B1

X := 20;

B2

if x >= 10 goto B4

if x <> 4 goto B6

B3

X := X - 1;
A[X] := 10;
if X <> 4 goto B6

B4

Y := X + 5;

B5

X := X - 2;

B6

goto B2

B2

B4

EXIT

B6

B5

B3
Constructing Basic Blocks
Assume that the input is a list of tuples. How do we find the beginning and end of each basic block?

1. First determine a set of **leaders**, the first tuple of basic blocks:
   1. The first tuple is a leader.
   2. Tuple L is a leader if there is a tuple \( \text{if } \ldots \text{goto } L \) or \( \text{goto } L \).
   3. Tuple L is a leader if it immediately follows a tuple \( \text{if } \ldots \text{goto } B \) or \( \text{goto } B \).

2. A basic block consists of a leader and all the following tuples until the next leader.
Basic Blocks...

\[
\begin{align*}
P &:= 0; I := 1; \\
\text{REPEAT} & \\
P &:= P + I; \\
\text{IF } P > 60 \text{ THEN} & \\
P &:= 0; \\
I &:= 5 \\
\text{ENDIF}; \\
I &:= I \times 2 + 1; \\
\text{UNTIL } I > 20; \\
K &:= P \times 3
\end{align*}
\]

<table>
<thead>
<tr>
<th>Step</th>
<th>Code</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>(P := 0)</td>
<td>(\Leftarrow) Rule 1.a</td>
</tr>
<tr>
<td>(2)</td>
<td>(I := 1)</td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td>(P := P + I)</td>
<td>(\Leftarrow) Rule 1.b</td>
</tr>
<tr>
<td>(4)</td>
<td>(\text{IF } P \leq 60 \text{ GOTO (7)})</td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td>(P := 0)</td>
<td>(\Leftarrow) Rule 1.c</td>
</tr>
<tr>
<td>(6)</td>
<td>(I := 5)</td>
<td></td>
</tr>
<tr>
<td>(7)</td>
<td>(T1 := I \times 2)</td>
<td>(\Leftarrow) Rule 1.b</td>
</tr>
<tr>
<td>(8)</td>
<td>(I := T1 + 1)</td>
<td></td>
</tr>
<tr>
<td>(9)</td>
<td>(\text{IF } I \leq 20 \text{ GOTO (3)})</td>
<td></td>
</tr>
<tr>
<td>(10)</td>
<td>(K := P \times 3)</td>
<td>(\Leftarrow) Rule 1.c</td>
</tr>
</tbody>
</table>
Basic Blocks...

\[ B_1: \] [(1) \( P := 0 \); (2) \( I := 1 \)]  
\[ B_2: \] [(3) \( P := P + I \);  
\hspace{1cm} (4) IF \( P \leq 60 \) GOTO \( B_4 \)]  
\[ B_3: \] [(5) \( P := 0 \); (6) \( I := 5 \)]  
\[ B_4: \] [(7) \( T1 := I \times 2 \); (8) \( I := T1 + 1 \);  
\hspace{1cm} (9) IF \( I \leq 20 \) GOTO \( B_2 \)]  
\[ B_5: \] [(10) \( K := P \times 3 \)]
Summary
Readings and References

- Read Louden:
  - Flow Graphs 475–477

- Or, read the Dragon book:
  - Basic Blocks 528–530
  - Flow Graphs 532–534
A Control Flow Graph (CFG) is a graph whose nodes are basic blocks. There is an edge from basic block $B_1$ to $B_2$ if control can flow from $B_1$ to $B_2$.

Control flows in and out of a CFG through two special nodes ENTER and EXIT.

We construct a CFG for each procedure. This representation is used during code generation and optimization.

Java bytecode is a stack-based IR. It was never intended as an UNCOL, but people have still built compilers for Ada, Scheme and other languages that generate Java bytecode. It is painful.

Microsoft’s MSIL is the latest UNCOL attempt.
Homework
Translate the program below into quadruples. Identify beginnings and ends of basic blocks. Build the control flow graph.

PROGRAM P;
VAR X : INTEGER; Y : REAL;
BEGIN
   X := 1; Y := 5.5;
   WHILE X < 10 DO
      Y := Y + FLOAT(X);
      X := X + 1;
      IF Y > 10 THEN Y := Y * 2.2; ENDIF;
   ENDDO;
END.
Exam Question

Draw the control flow graph for the tuples.

```c
int A[5], x, i, n;
for (i = 1; i <= n; i++) {
    if (i < n) {
        x = A[i];
    } else {
        while (x > 4) {
            x = x * 2 + A[i];
        }
    }
}
x = x + 5;
```

(1) $i := 1$
(2) IF $i > n$ GOTO (14)
(3) IF $i >= n$ GOTO (6)
(4) $x := A[i]$
(5) GOTO (11)
(6) IF $x <= 4$ GOTO (11)
(7) $T1 := x * 2$
(8) $T2 := A[i]$
(9) $x := T1 + T2$
(10) GOTO (6)
(11) $x := x + 5$
(12) $i := i + 1$
(13) GOTO (2)