**Break Statements**

Compilation as tree transformation puts jumps into ‘branch’ meta instructions.

It then rewrites these meta instructions.

Alternatively, meta instruction such as ‘branch’ can be simply a function of the compiler implementation, the execution of ‘branch’ then produces the desired code.

The main idea is that compilation functions can accept parameters that denote symbolic destinations for potential jumps.

**Final Destination Parameters**

For example, an alternative to implicitly emitting instructions in the right order, we can always pass to the translation function the label to which the function should go in the end.

In such case we would have the following example translations for sequence:

[[ s1 ; s2 ]](dest) =

[[ s1 ]](freshLabel)

freshLabel: [[ s2 ]](dest)

where ‘freshLabel’ is a fresh label generated in the translation function. Note that the code for s2 here does not need to immediately follow the code for s1, there can be any gap afterwards.

The translation of assignment then needs to jump to the argument of the translation:

[[ x = e ]](dest) =

[[e]]

istore slot(x)

goto dest

**Additional Jump Destinations**

Suppose that we now wish to handle ‘break’ statements. We then add one more parameter. Normal sequencing and most other statements just pass this extra parameter along:

[[ s1 ; s2 ]](dest,brk) =

[[ s1 ]](freshLabel,brk)

freshLabel: [[ s2 ]](dest,brk)

Statements that continue normal executions use the first parameter:

[[ x = e ]](dest,brk) =

[[e]]

istore slot(x)

goto dest

In contrast, the ‘break’ statement jumps to the second label parameter:

[[ break ]](dest,brk) =

goto brk

Finally, a loop statement changes the interpretation of the second label parameter, indicating that ‘break’ inside the loop will apply to the innermost loop and not to an outer loop. Below, ‘test’ and ‘inside’ are fresh labels, and ‘branch’ is like before:

[[ while (c) s ]](dest,brk) =

test: branch(c,inside,dest) // goes to 'inside' if true, to 'dest' if false

inside: [[ s ]](dest,dest)

In the above translation rule, the change of the meaning of ‘break’ happens with

[[ s ]](dest,dest)

How would we interpret ‘continue’ statement? We can pass one more parameter. This is left as an exercise.