CS4100/5100 COMPILER DESIGN PROJECT
Code Generation in Part 4: #3 WHILE and IF/ELSE
Fall 2020

From the previous REPEAT-UNTIL discussion, recall the use of the relexpression function, which builds the comparison and jump instruction Quads. This will be applied here as well.

The While Loop

Code for the While loop consists of the following required elements:

1) A condition check
2) A conditional branch around the loop body if the condition is FALSE
3) An unconditional branch from the bottom of the loop body to the start of the condition check at the top of the loop.

Here is an approach when the **while** token is found in the <statement> switch structure:

```
... inside of statement....
else .....
  if (tokenCode == WHILE)
   {
    // declare above int saveTop, branchQuad
    GNT;                                  //move past this token
    saveTop = nextQuad; //Before generating code, save top of loop
                        // where unconditional branch will jump
    branchQuad = relexpression; //tells where branchTarget to be set
    If (tokenCode == DO)        //move past DO
     {
      GNT;
      statement;                //the loop body is processed
      AddQuad(Branch_Op, 0, 0, saveTop);//jump to top of loop
                                        //backfill the forward branch
      //Quad function for ease- set 3rd op
      Quad.setQuadOp3(branchQuad,nextQuad);//conditional jumps nextQuad
     }
    ELSE ... {Handle missing DO error}
   } //end of while structure
  else ....
```

The If/Else Statement

The If, with optional Else, is a bit more complicated to follow than the While, because there are a couple of unknown branch targets to be backfilled when they become known. The basic elements to be built are:

1) A condition check
2) A conditional branch around the IF body if the condition is FALSE
3) If an ELSE is found, add an unconditional branch around the ELSE body at the end of the IF body

Here is an approach when the **if** token is found in the <statement> switch structure:

```
        //variables needed:
        //      int branchQuad, patchElse
else .....
  if (tokenCode == IF)
   {
    GNT;                          // move past 'if'
    branchQuad = relexpression; //tells where branchTarget to be set
                                  // to jump around TRUE part
    if tokencode == THEN          //all ok, continue
     {
      GNT;                        // move past 'then'
      statement;                  //all if body quads are genned
      if tokencode == ELSE        //have to jump around to ??
       {
        GNT;                      // move past ELSE
        patchElse = nextQuad;     //save backfill quad to jump around
                                  // ELSE body, target is unknown now
        AddQuad(Branch_op, 0, 0, 0);
                                  //backfill the FALSE IF branch jump
        Quad.setQuadOp3(branchQuad,nextQuad);//conditional jump
        statement;                // gen ELSE body quads
                                  // fill in end of ELSE part
        Quad.setQuadOp3(patchElse, nextQuad);
       }
      else                        //no ELSE encountered, fix IF branch
        Quad.setQuadOp3(branchQuad, nextQuad);
     }                            //if the THEN was found
    else                          // error, no THEN
        error(...);
   }                              // end of IF statement stuff
```