

# Program Transformation

December 25, 2015

---

**Algorithm 1** analyse a module

---

```
1: function ANALYSE( $M$ )
2:   for all  $F \in M$  do
3:      $controlStructs \leftarrow getTopLevelControlStructs(F.BList)$  ▷
       ControlStruct is a conditional branch or loop
4:     for all  $S \in controlStructs$  do
5:        $analyse(S)$ 
6:     end for
7:   end for
8: end function
```

---

---

**Algorithm 2** analyse a control structure

---

```
1: function ANALYSE( $S$ )
2:   if  $isTransformable(S)$  then
3:      $transform(S)$ 
4:   else
5:      $controlStructs \leftarrow getTopLevelControlStructs(S.BList)$ 
6:     for all  $S' \in controlStructs$  do
7:        $analyse(S')$  ▷ do recursive transformation
8:     end for
9:   end if
10: end function
```

---

---

**Algorithm 3** check if a control structure is transformable

---

```

1: function ISTTRANSFORMABLE( $S$ )
2:   if  $\forall var \in def(S), var \notin OUT[S]$  then
3:     return true
4:   else
5:     return false
6:   end if
7: end function

```

---



---

**Algorithm 4** transform a control structure

---

```

1: function TRANSFORM( $S$ )
2:   for all  $B \in S$  do
3:     for all  $I \in B$  do
4:       if  $isCritical(I)$  then                                      $\triangleright$  check if dereference or
       getElementOfArray                                            $\triangleright$  check if dereference or
5:         insert  $I$  before  $S$ 
6:       end if
7:     end for
8:   end for
9:    $B \leftarrow getSuccessor(S)$ 
10:  insert unconditional branch br label B before  $S$ 
11: end function

```

---



---

**Algorithm 5** get all of the top level control structures in the basic block list

---

```

1: function GETTOPLEVELCONTROLSTRUCTS( $Blist$ )
2:    $ret \leftarrow empty\ ControlStruct\ list$ 
3:   for all  $B \in BList$  do
4:     if  $B$  is loop header then
5:        $LS \leftarrow loop\ structure\ of\ B$ 
6:       if  $LS$  has one successor then
7:         add  $LS$  to  $ret$ 
8:         skip blocks in  $LS$ 
9:       end if
10:    else if  $B$  is conditional branch block then                  $\triangleright$   $B$  is 'if' or 'switch'
        condition block
11:       $B' \leftarrow postdominator\ of\ B$ 
12:      construct a conditional structure  $CS$  using blocks between  $B$  and  $B'$ 
13:      add  $CS$  to  $ret$ 
14:      skip blocks in  $CS$ 
15:    end if
16:  end for
17:  return  $ret$ 
18: end function

```

---

---

**Algorithm 6** design of ControlStruct class

---

```
1: class ControlStruct {  
2: public:  
3: bool isTransformable();  
4:  
5:  
6: };
```

---