School of Mathematical and Computational Sciences

Indian Association for the Cultivation of Science

Compiler Construction: COM 5202

Tutorial XIII (23 April, 2025)

M. Sc Semester IV: 2024-2025 *Instructor:* Goutam Biswas

Exercise 1. *Consider the following C function:*

```
void initMat(int a[][10], int n){
     int i, j;
     for(i=0; i<n; ++i)
         for(j=0; j<n; ++j) a[i][j]=1;
}
```

Corresponding 3-address code is as follows:

```
    i=0
    goto L4
L5:
    j=0
    goto L1
L2:
(A) $1=40*i
(B) $2=4*j
(C) $3=a+$1
(D) $4=$3+$2
(E) *($4)=1
    j=j+1
L1:
    if j<n goto L2 else goto L3
L3:
       i=i+1
L4:
    if i<n goto L5 else goto L6
L6:
```

(a) *Identify the loop invarient computations from* $(A) \cdots (E)$. *Place them at appropriate places outside the loop. Do not perform any other transformation at this stage.*

(b) *Find the sequence of values in* `$1` *and* `$2`.

(c) *How do you modify the code for the computation of induction variables (a variable whose values forms an AP)* `$1` *and* `$2` *to reduce the cost of computation?*

(d) *Can the variable* `i, j` *be removed?*

(e) *How is the GCC x86-64 code?*

Exercise 2. *Consider the following C function:*

```
void evalArrIndex(int a[][10][20], int n){
     int i, j, k;

     for(i=0; i<n; ++i)
        for(j=0; j<n; ++j)
            for(k=0; k<n; ++k)
                a[2*i+5][3*j+1][4*k+3]=0;
}
```

Corresponding 3-address code is as follows:

```
    i=0
    goto L8
L7:
    j=0
    goto L5
L4:
    k=0
    goto L2
L1:
(A)    $1=2*i
(B)    $2=$1+5
(C)    $3=800*$2
(D)    $4=3*j
(E)    $5=$4+1
(F)    $6=80*$5
(G)    $7=4*k
(H)    $8=$7+3
(I)    $9=4*$8
(J)    $10=a+$3
(K)    $11=$10+$6
(L)    $12=$11+$9
(M)    *($12)=0
       k=k+1
L2:
    if k<n goto L1 else goto L3
L3:
    j=j+1
L5:
    if j<n goto L4 else goto L6
L6:
    i=i+1
L8:
    if i<n goto L7 else goto L9
L9:
    return
```

(a) *Identify the loop invarient computations from* $(A) \cdots (M)$. *Place them at appropriate places outside the loop. Do not perform any other transformation at this stage.*

(b) *Find the sequence of values in* `$1` *and* `$2`.

(c) *How do you modify the code for the computation of induction variables (a variable whose values forms an AP)* `$1` *and* `$2` *to reduce the cost of computation?*

(d) *Find the sequence of values in* `$3` *and modify its code as well.*

(e) *Eliminate dead code.*

(f) *Perform the similar transformation for the second index computation.*

(g) *Perform similar transformation for the third index computation.*

(h) *How is the 3-address code after this?*

(i) *Can we remove the variables* `i,j,k`*?*