School of Mathematical and Computational Sciences

Indian Association for the Cultivation of Science

Compiler Construction: COM 5202

Tutorial X (02 April, 2025)

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

Exercise 1.

(a) *Construct DAG for the following 3-address code sequence in a basic block.*

```
d = b * c
e = a + b
b = b * c
a = e - d
```

(b) *Identify the common subexpression and modify the code.*

(c) *Simplify the 3-address code if only* `a` *is live on exit from the block.*

(d) *Simplify the 3-address code if* `a, b, c` *are live on exit from the block.*

Exercise 2. *Consider the 2D array* `int a[5][10]` *stored in row-major order. Each data of type* `int` *takes 4-bytes.*

(a) *Write the low-level expression to compute the address of* `a[i][j]`.

(b) *Write 3-address code corresponding to the expressions.*

(c) *Rewrite the code by performing some code improvement.*

(d) *How does the address computation formula changes if the array is stored in column-major order?*

Exercise 3. *Consider the following C code and its translation to the GCC x86-64 assembly language code.*
Identify the access links and data access in the assembly language code.

```
/*
 *  nonLocalOnStk1.c : accessing non-local data on stack
 */
#include <stdio.h>

int main(){
    int m=10, n=100, p=200;
    void fA(){
         int x = 20, y=50, z=150;
         void fB(){
              int b=2, val;
              val = b+x-m+n-y+z-p;
              printf("val: %d\n", val);
         }
         fB();
    }
    fA();
    return 0;
}
```

GCC x86-64 Assembly Language Code

```
    .file    "nonLocalOnStk1.c"
    .text
    .section    .rodata
.LC0:
    .string    "val: %d\n"
    .text
    .type    fB.2388, @function
fB.2388:
.LFB2:
    .cfi_startproc
    pushq    %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    subq    $32, %rsp
    movq    %r10, %rax
    movq    %r10, -24(%rbp)
    movl    $2, -8(%rbp)
    movl    16(%rax), %ecx
    movl    -8(%rbp), %edx
    addl    %edx, %ecx
    movq    (%rax), %rdx
    movl    8(%rdx), %edx
    subl    %edx, %ecx
    movq    (%rax), %rdx
    movl    4(%rdx), %edx
    addl    %edx, %ecx
    movl    12(%rax), %edx
    subl    %edx, %ecx
    movl    8(%rax), %edx
    addl    %ecx, %edx
    movq    (%rax), %rax
    movl    (%rax), %eax
    subl    %eax, %edx
    movl    %edx, %eax
    movl    %eax, -4(%rbp)
    movl    -4(%rbp), %eax
    movl    %eax, %esi
    leaq    .LC0(%rip), %rdi
    movl    $0, %eax
    call    printf@PLT
    nop
    leave
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
```

```
.LFE2:
    .size    fB.2388, .-fB.2388
    .type    fA.2383, @function
fA.2383:
.LFB1:
    .cfi_startproc
    pushq    %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    subq    $64, %rsp
    movq    %r10, %rax
    movq    %r10, -56(%rbp)
    movq    %fs:40, %rcx
    movq    %rcx, -8(%rbp)
    xorl    %ecx, %ecx
    leaq    16(%rbp), %rdx
    movq    %rdx, -24(%rbp)
    movq    %rax, -48(%rbp)
    movl    $20, %eax
    movl    %eax, -32(%rbp)
    movl    $50, %eax
    movl    %eax, -36(%rbp)
    movl    $150, %eax
    movl    %eax, -40(%rbp)
    leaq    -48(%rbp), %rax
    movq    %rax, %r10
    movl    $0, %eax
    call    fB.2388
    nop
    movq    -8(%rbp), %rax
    xorq    %fs:40, %rax
    je    .L3
    call    __stack_chk_fail@PLT
.L3:
    leave
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE1:
    .size    fA.2383, .-fA.2383
    .globl    main
    .type    main, @function
main:
.LFB0:
    .cfi_startproc
    pushq    %rbp
    .cfi_def_cfa_offset 16
    .cfi_offset 6, -16
```

```
    movq    %rsp, %rbp
    .cfi_def_cfa_register 6
    subq    $32, %rsp
    movq    %fs:40, %rax
    movq    %rax, -8(%rbp)
    xorl    %eax, %eax
    leaq    16(%rbp), %rax
    movq    %rax, -16(%rbp)
    movl    $10, %eax
    movl    %eax, -24(%rbp)
    movl    $100, %eax
    movl    %eax, -28(%rbp)
    movl    $200, %eax
    movl    %eax, -32(%rbp)
    leaq    -32(%rbp), %rax
    movq    %rax, %r10
    movl    $0, %eax
    call    fA.2383
    movl    $0, %eax
    movq    -8(%rbp), %rdx
    xorq    %fs:40, %rdx
    je    .L6
    call    __stack_chk_fail@PLT
.L6:
    leave
    .cfi_def_cfa 7, 8
    ret
    .cfi_endproc
.LFE0:
    .size    main, .-main
    .ident    "GCC: (Ubuntu 9.4.0-1ubuntu1~18.04) 9.4.0"
    .section    .note.GNU-stack,"",@progbits
```