طراحی کامپایلرها

نیم‌سال اول ۰۳-۰۲

دانشکده‌ی مهندسی کامپیوتر
استاد: سمانه حسینمردی

تمرین چهارم
مهلت ارسال: ۱۸ دی

مسئله‌ی ۱.

به قطعه کد زیر توجه کنید. فرض کنید مقادیر i و j در ابتدا مشخص هستند و در انتهای کد فقط به مقدار متغیر x نیاز داریم.

الف

پس از مشخص کردن Basic Block ها، CFG متناظر با این کد را رسم کنید.

ب

به کمک بهینه‌سازی‌های محلی تا حد ممکن کد را ساده کنید.

```
1. a = 2 * i
2. b = a + i
3. x = a + b
4. if i > j goto 9
5. d = b + 3
6. e = b
7. b = e + 3
8. goto 14
9. c = 8
10. x = c * x
11. if x > 5 goto 5
12. a = 7
13. b = 12
14. x = x + b
```

مسئله‌ی ۲.

قطعه کد زیر را در نظر بگیرید.

الف

activation tree متناظر با این قطعه کد را رسم کنید.

ب

run-time stack را قبل و بعد از بازگشتن از تابع h رسم کنید.

```
program main()
    var a,b:int;
    procedure h(i:int)
        var c:int;
        c = i + 5;
        b = c + a;
    end h;
    procedure f(i:int)
        var d,e:int;
        procedure g(j:int)
            d = d + j;
            if (j > 1) g(j - 1);
            else a = d;
        end g;
        e = i - 1;
        g(e);
    end f;
    f(3);
    h(5);
end main;
```

مسئله‌ی ۳.

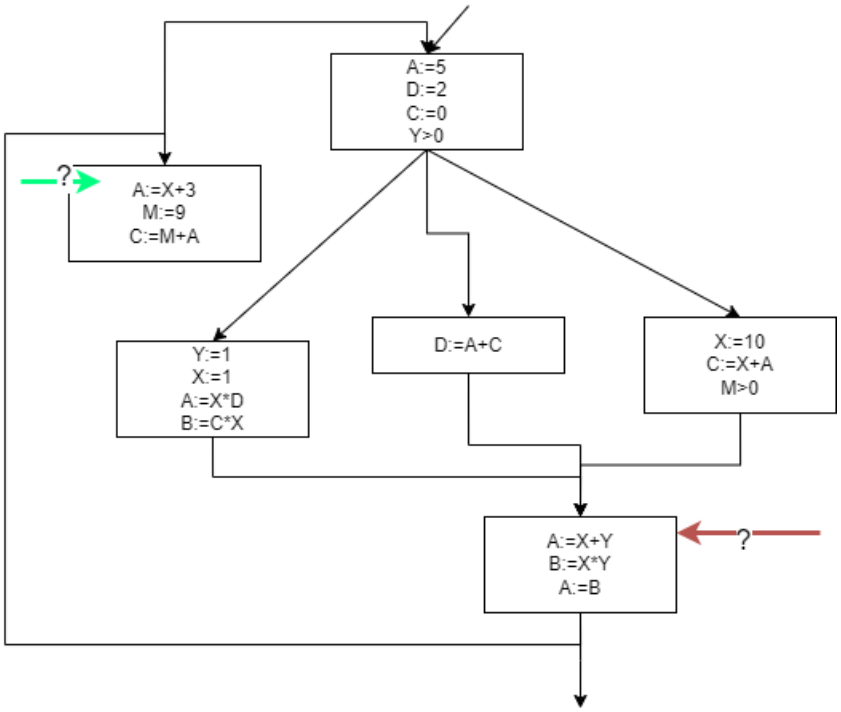
برنامه زیر را درنظر بگیرید

الف

پس از اجرای الگوریتم constant propagation تا تکمیل، dataflow information صحیح را برای A ، B و C در نقطه مشخص شده به دست آورید (فلش قرمز رنگ)

ب

پس از اجرای الگوریتم liveness analysis تا تکمیل، کدام یک از متغیرها در نقطه مشخص شده فعال هستند؟ فرض کنید همه متغیرها در هنگام خروج مرده باشند. (فلش سبز رنگ)



مسئله‌ی ۴.

فرض کنید در انتهای کد زیر، فقط e زنده است.

الف

control flow graph این کد را رسم کنید

ب

Liveness analysis را انجام دهید

ج

register inference graph را رسم کنید

د

تعداد مینیمم رجیسترهای مورد نیاز را به دست آورید

```
L0: e := 0
b := 0
d := 2
a := 9
L1: a := b + 2
c := d + 5
d := d + 10
c := d + c
e := e - c
f := b * a
b := b * b
if e < f goto L3
L2: e := e + f
goto L4
L3: e := e + 2
b := e + 2
a := e - 2
L4: d := d + 4
d := b * b
if b != e goto L1
```