

Compilers: Assignment #3

Due on Sunday, December 3, 2016

Genaft: Task 2

Mirza Hasanbasic

Indhold

Task 1	2
Task 2	4

Task 1

Det skal siges, at jeg har vedhæftet en .txt fil, så du lettere kan afprøve koden som er skrevet. Det der står i dokumentet er til det visuelle.

a)

Vi har

```
1 vtable = [a → v, b → w];
2
3 while (b != 0) && (a/b != 0)
4     if b < a then {a := a - b}
5                 else {b := b - a}
```

Hvor intermediate koden er

```
1 t_0 = v
2 t_1 = w
3 LABEL LoopStart
4 IF t_1 != 0 then NEXT0 else END (Brug rigtig syntax i tex filen !=)
5 LABEL NEXT0
6 t_2 = t_0 mod t_1
7 IF t_2 != 0 then NEXT1 else END
8 LABEL NEXT1
9 t_3 = t_1 - t_0
10 IF t_3 < 0 then NEXT2 else NEXT3
11 LABEL NEXT2
12 t_0 = t_0 - t_1
13 GOTO LoopStart
14 LABEL NEXT3
15 t_1 = t_1 - t_0
16 GOTO LoopStart
17 LABEL END
```

og MIPS koden vil være

```
1  .data
2      a: .word 8
3      b: .word 33
4  .text
5  main:
6      lw $t0, a           \# load 8
7      lw $t1, b           \# load 33
8      LoopStart:         \# LABEL
9      beq $t1, $0, END    \# Checking if t1 == 0
10     div $t0, $t1        \# dividing to get modulus
11     mfhi $t2            \# Getting the remainder, moving to $t2
12     beq $t2, $0, END    \# checking if t2 == 0
13     sub $t3, $t1, $t0   \# t3 = t1 - t0
14     bgez $t3, ELSE      \# t3 >= 0
15     sub $t0, $t0, $t1   \# first then statement a = a - b
16     j LoopStart         \# jumping to loopstart
17 ELSE:                  \# Now else statement
18     sub $t1, $t1, $t0   \# b = b - a
19     j LoopStart
20 END:
21                             \# tinyurl.com/neve79o
22     li $v0, 1           \# printer udregnet variable ud.
23     add $a0, $t0, $zero
24     syscall
25
26     li $v0, 11
27     li $a0, 10
28     syscall
29
30     li $v0, 1
31     add $a0, $t1, $zero
32     syscall
```

b)

```
1  Li t0, x
2  Li t1, y
3  Li t2, 1
4  Slt t3, t1, t0
5  Slt t4, t3, t2
```

Task 2

a)

i	succ[i]	gen[i]	kill[i]
1	2		
2	7,3	a,b	
3	4		
4	5	a	t
5	6	b	a
6	7	t	b
7	8		
8	9		z
9	10	b,a	b
10	1,11	b,z	
11			
12		a	

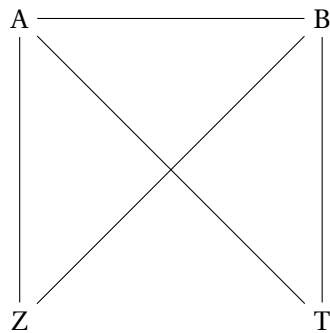
b)

FIX:

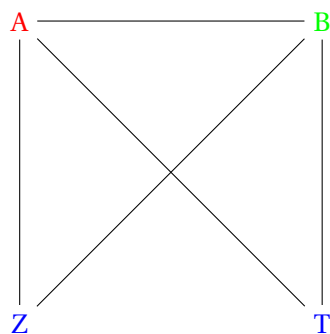
Initial			Iteration 1		Iteration 2		Iteration 3	
i	out[i]	in[i]	out[i]	in[in]	out[i]	in[in]	out[i]	in[in]
1			a,b	a,b	a,b	a,b	a,b	a,b
2			a,b	a,b	a,b	a,b	a,b	a,b
3			a,b	a,b	a,b	a,b	a,b	a,b
4			b,t	a,b	b,t	a,b	b,t	a,b
5			a,t	b,t	a,t	b,t	a,t	b,t
6			a,b	a,t	a,b	a,t	a,b	a,t
7			a,b	a,b	a,b	a,b	a,b	a,b
8			a,b,z	a,b	a,b,z	a,b	a,b,z	a,b
9			b,z,a	a,b,z	a,b,z	a,b,z	a,b,z	a,b,z
10			a	a, b,z	a,b	a,b,z	a,b	a,b,z
11			a	a	a	a	a	a
12				a		a		a

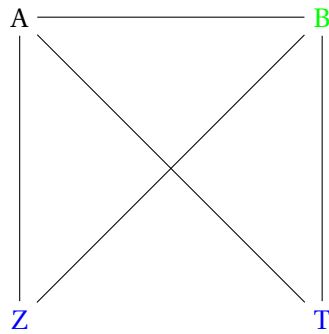
c)

i	left	interferes with
4	t	a,b
5	a	b,t
6	b	a,t
8	z	a,b
9	b	a,z

**d)****FIX:**

node	Neighbours	color
a		1
b	a	2
t	a, b	3
z	a,b	3



e)

```
1 gcd(a,b) {
2   M[addressa] := a
3   LABEL start
4   ai := M[addressa]
5   IF ai < b THEN next ELSE swap
6   LABEL swap
7   ai := M[addressa]
8   t := ai
9   ai := b
10  M[addressa] = ai
11  b := t
12  LABEL next
13  z := 0
14  ai = M[addressa]
15  b := b mod ai
16  IF b = z THEN end ELSE start
17  LABEL end
18  a := M[addressa]
19  RETURN a
20 }
```

og dette vil være med 2 registre

FIX:

