Compilers: Assignment #3

Due on Sunday, December 3, 2016

Genafl: Task 2

Mirza Hasanbasic

Indhold

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

```
vtable = [a → v, b → w];

while (b != 0) && (a/b != 0)
        if b < a then {a := a − b}
                        else {b := b − a}
```

Hvor intermediate koden er

```
t_0 = v
t_1 = w
LABEL LoopStart
IF t_1 != 0 then NEXT0 else END (Brug rigtig syntax i tex filen !=)
LABEL NEXT0
t_2 = t_0 mod t_1
IF t_2 != 0 then NEXT1 else END
LABEL NEXT1
t_3 = t_1 − t_0
IF t_3 < 0 then NEXT2 else NEXT3
LABEL NEXT2
t_0 = t_0 − t_1
GOTO LoopStart
LABE NEXT3
t_1 = t_1 − t_0
GOTO LoopStart
LABEL END
```

og MIPS koden vil være

Task 1 continued on next page. . .

```
.data
        a: .word 8
        b: .word 33
.text
main:
lw $t0, a                       \# load 8
lw $t1, b                       \# load 33
LoopStart:                      \# LABEL
beq $t1, $0, END                \# Checking if t1 == 0
div $t0, $t1                    \# dividing to get modulus
mfhi $t2                        \# Getting the remainder, moving to $t2
beq $t2, $0, END                \# checking if t2 == 0
sub $t3, $t1, $t0               \# t3 = t1 - t0
bgez $t3, ELSE                  \# t3 >= 0
sub $t0, $t0, $t1               \# first then statement a = a - b
j LoopStart                     \# jumping to loopstart
ELSE:                              \# Now else statement
sub $t1, $t1, $t0               \# b = b - a
j LoopStart
END:
                                \# tinyurl.com/neve79o
li $v0, 1                    \# printer udregnet variable ud.
add $a0, $t0, $zero
syscall

li $v0, 11
li $a0, 10
syscall

li $v0, 1
add $a0, $t1, $zero
syscall
```

b)

```
Li t0, x
Li t1, y
Li t2, 1
Slt t3, t1, t0
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)

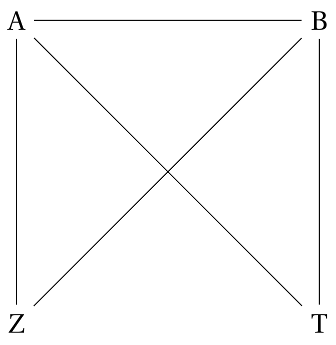
FIX:

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 ,b	a, b,z	a,b	a,b,z	a,b	a,b,z
11			a	a	a	a	a	a
12				a		a		a

Task 2 continued on next page. . .

c)

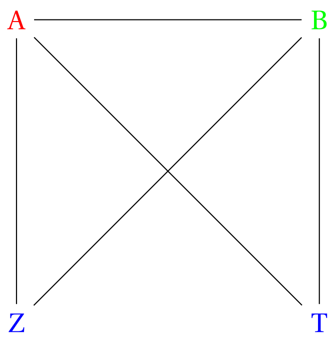
i	left	interferes with
4	t	a,b
5	a	b,t
6	b	a,t
8	z	a,b
9	b	a,z



d)

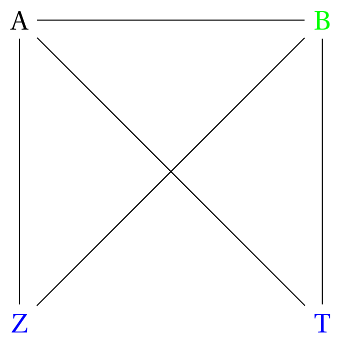
FIX:

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



Task 2 continued on next page. . .

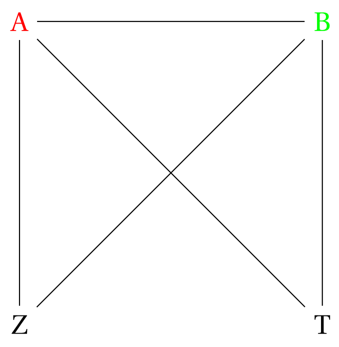
e)



```
gcd(a,b) {
    M[address_a] := a
    LABEL start
    a_i := M[address_a]
    IF a_i < b THEN next ELSE swap
    LABEL swap
    a_i := M[address_a]
    t := a_i
    a_i := b
    M[address_a] = a_i
    b := t
    LABEL next
    z := 0
    a_i = M[address_a]
    b := b mod a_i
    IF b = z THEN end ELSE start
    LABEL end
    a := M[address_a]
    RETURN a
}
```

og dette vil være med 2 registre

FIX:


A
B
Z
T