

W-assignment 1

Ola Rønning
vdl761

January 5, 2014

1 Exercise 1

Given the following program:

```

gcd(a,b) [
1: LABEL start
2: IF a < b THEN next ELSE swap
3: LABEL swap
4: t := a
5: a := b
6: b := t
7: LABEL next
8: z := 0
9: b := b mod a
10: IF b = z THEN end ELSE start
11: LABEL end
12: RETURN a
]
```

a

The $\text{gen}[i]$, $\text{kill}[i]$ and $\text{succ}[i]$ is constructed bottom up:

Table 1

Instruction i	$\text{gen}[i]$	$\text{kill}[i]$	$\text{succ}[i]$
1: LABEL start	\emptyset	\emptyset	2
2: IF a < b THEN next ELSE swap	{a,b}	\emptyset	{3,7}
3: LABEL swap	\emptyset	\emptyset	4
4: t := a	{a}	{t}	5
5: a := b	{b}	{a}	6
6: b := t	{t}	{b}	7
7: LABEL next	\emptyset	\emptyset	8
8: z := 0	\emptyset	{z}	9
9: b := b mod a	{a,b}	{b}	10
10: IF b = z THEN end ELSE start	{b,z}	\emptyset	{11,1}
11: LABEL end	\emptyset	\emptyset	12
12: RETURN a	{a}	\emptyset	\emptyset

b

To compute $\text{out}[i]$ and then $\text{in}[i]$, defined in equation 8.2, 8.1 (page 161) of ICD, I use fixed-point iteration as in the book and again construct it bottom up:

Table 2

Iteration I	Iteration 1		Iteration 2	
	$\text{out}[i]$	$\text{in}[i]$	$\text{out}[i]$	$\text{in}[i]$
1: LABEL start	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$
a,b	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$
3: LABEL swap	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$
4: $t := a$	$\{b,t\}$	$\{a,b\}$	$\{b,t\}$	$\{a,b\}$
5: $a := b$	$\{a,t\}$	$\{b,t\}$	$\{a,t\}$	$\{b,t\}$
6: $b := t$	$\{a,b\}$	$\{a,t\}$	$\{a,b\}$	$\{a,t\}$
7: LABEL next	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$
8: $z := 0$	$\{a,b,z\}$	$\{a,b\}$	$\{a,b,z\}$	$\{a,b\}$
9: $b := b \bmod a$	$\{a,b,z\}$	$\{a,b,z\}$	$\{a,b,z\}$	$\{a,b,z\}$
10: IF $b = z$ THEN end ELSE start	a	$\{a,b,z\}$	a	$\{a,b,z\}$
11: LABEL end	a	a	a	a
12: RETURN a		a		a

c

To draw the interference graph for variables a , b , t and z I use definition 8.2 (page 164) of ICD. I look up the lines i with $\text{kill}[i] \neq \emptyset$ and find the corresponding $\text{out}[i]$ for that line. The i 's that generate interference are $\{4, 5, 6, 8, 9\}$ giving the interference:

$$t \rightarrow b \quad (1)$$

$$a \rightarrow t \quad (2)$$

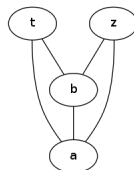
$$b \rightarrow a \quad (3)$$

$$z \rightarrow a \quad (4)$$

$$z \rightarrow b \quad (5)$$

$$b \rightarrow z \quad (6)$$

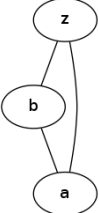
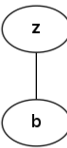

These interferences correspond to the following graph:



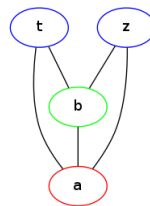
d

Using algorithm 8.3 (page 166) of ICD and three colours, red, green and blue, to colour the graph above I get the following stack and graphs: Now I pop from the stack giving t the

Table 3

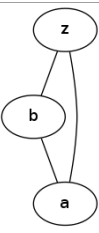


operation	stack	graph
simplify	$t \rightarrow \{a, b\}$	
simplify	$a \rightarrow \{z, b\}$ $t \rightarrow \{a, b\}$	
simplify	$z \rightarrow \{b\}$ $a \rightarrow \{z, b\}$ $t \rightarrow \{a, b\}$	
simplify	$b \rightarrow \{\}$ $z \rightarrow \{b\}$ $a \rightarrow \{z, b\}$ $t \rightarrow \{a, b\}$	

colour blue, a the colour red, z the colour Blue and b the colour green giving me the coloured graph:



e

As before I use the algorithms 8.3 from ICD, but this time with only 2 colours, red and blue:

operation	stack	graph
simplify	$t \rightarrow \{a, b\}$	
simplify	$a \rightarrow \{z, b\}$ $t \rightarrow \{a, b\}$	
simplify	$z \rightarrow \{b\}$ $a \rightarrow \{z, b\}$ $t \rightarrow \{a, b\}$	
simplify	$b \rightarrow \{\}$ $z \rightarrow \{b\}$ $a \rightarrow \{z, b\}$ $t \rightarrow \{a, b\}$	

Now I pop from the stack giving t the colour blue, a the colour red. There is no colour for b, so I spill it to memory and colour z blue. This produces the following graph:



and producing the following program:

```

gcd(a,b) [
b:=M[address_b]
1: LABEL start
b1:=M[address_b]
2: IF a < b THEN next ELSE swap
M[address_b] := b1
3: LABEL swap
4: t := a
b2:=M[address_b]
```

```
5:  a := b
M[address_b] := b2
b3:=M[address_b]
6:  b := t
M[address_b] := b3
7:  LABEL next
8:  z := 0
b4:=M[address_b]
9:  b := b mod a
M[address_b] := b4
b5:=M[address_b]
10: IF b = z THEN end ELSE start
M[address_b] := b5
11: LABEL end
12: RETURN a
    ]
```

2 Exercise 2

I expand the statments to include break and continue:

```
Stat -> Stat ; Stat
    | id := Exp
    | if Cond then { Stat }
    | if Cond then { Stat } else { Stat }
    | while Cond do { Stat }
    | repeat { Stat } until Cond
    | break
    | continue

Loop -> repeat
    | while
    |
```

I expand the translation function to include another variable: Loop, so the translation function look like: $\text{Trans}_{\text{state}}(\text{Loop}, \text{Stat}, \text{vtable}, \text{fable})$. The loop variable will be used to select which type of loop the break and/or continue is inside, and can have the form repeat, while or empty. The break and continue will then make gotos based on the type of loop based to it, all other translation functions will pass an empty value to the loop variable, and hence no changes are necessary to the function body of these.

$\text{Trans_Stat}(\text{loop}, \text{stat}, \text{vtable}, \text{fable}) = \text{case stat of}$

```
repeat Stat until Cond
    labelf = newlabel()
    labelt = newlabel()
    code1 = TransStat (repeat,Stat, vtable, ftable)
    code2 = TransCond (Cond, labelt , labelf , vtable, ftable)
    [LABEL labelf ] @ code1 @ code2 @ [LABEL labelt ]

while Cond do Stat
    labels = newlabel()
    labelt = newlabel()
    labelf = newlabel()
    code1 = TransCond (Cond, labelt , labelf , vtable, ftable)
    code2 = TransStat (while, Stat, vtable, ftable)
    [LABEL labels ] @ code1
    @ [LABEL labelt ] @ code2 @ [GOTO labels ]
    @ [LABEL labelf ]

break
case loop of
    while
        [goto label_f]
    repeat
        [goto label_t]
continue
case loop of
    while
        [goto label_s]
    repeat
        [goto label_f]
```