KØBENHAVNS UNIVERSITET

Assignment 4

(hjg708)

IPS

2. juni 2023

Task 1

$_{(11)}\text{let } x =_{(1)} u \text{ in } _{(10)}\text{let } y =_{(2)} x+x \text{ in } _{(9)}\text{let } y =_{(5)} (\text{let } x =_{(3)} foo(x) \text{ in } _{(4)}7) \text{ in } _{(8)}\text{let } t =_{(6)} x+u \text{ in } _{(7)}y*x$

n	IO	Elim	UV	OE
1	no	-	{u}	u
2	no	-	{x}	x + x
3	yes	-	{x}	foo(x)
4	no	-	{ }	7
5	yes	no	{x} ∪ ({ } \{x}) = {x}	let x = foo(x) in 7
6	no	-	{x, u}	x + u
7	no	-	{y, x}	y * x
8	no	yes	{y, x}	y * x
9	yes	no	{x} ∪ ({y, x} \{y}) = {x}	let y = (let x = foo(x) in 7) in y * x
10	yes	yes	{x}	let y = (let x = foo(x) in 7) in y * x
11	yes	no	{u} ∪ ({x} \{x}) = {u}	let x = u in (let y = (let x = foo(x) in 7) in y * x

Task 2

a)

Show succ, gen and kill sets for every instruction in the program.

i	succ(i)	gen(i)	kill(i)
1: LABEL start	2	-	-
2: IF a < b THEN next ELSE swap	3, 7	a, b	-
3: LABEL swap	4	-	-
4: t := a	5	a	t
5: a := b	6	b	a
6: b := t	7	t	b
7: LABEL next	8	-	-
8: z := 0	9	-	z
9: b := b mod a	10	a, b	b
10: IF b = z THEN end ELSE start	11, 1	b, z	-
11: LABEL end	12	-	-
12: RETURN a	-	a	-

b)

Compute in and out sets for every instruction, show the fix-point iteration.

	intial		iteration 1		iteration 2		iteration 3	
i	out(i)	in(i)	out(i)	in(i)	out(i)	in(i)	out(i)	in(i)
1: LABEL start	-	-	a, b	a, b	a, b	a, b	a, b	a, b
2: IF a < b THEN next ELSE swap	-	-	a, b	a, b	a, b	a, b	a, b	a, b
3: LABEL swap	-	-	a, b	a, b	a, b	a, b	a, b	a, b
4: t := a	-	-	b, t	a, b	b, t	a, b	b, t	a, b
5: a := b	-	-	a, t	b, t	a, t	b, t	a, t	b, t
6: b := t	-	-	a, b	a, t	a, b	a, t	a, b	a, t
7: LABEL next	-	-	a, b	a, b	a, b	a, b	a, b	a, b
8: z := 0	-	-	a, b, z	a, b	a, b, z	a, b	a, b, z	a, b
9: b := b mod a	-	-	a, b, z	a, b, z	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, b	a, b, z	a, b	a, b, z
11: LABEL end	-	-	a	a	a	a	a	a
12: RETURN a	-	-	-	a	-	a	-	a

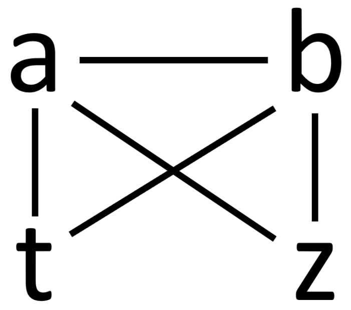
c)

Show the interference table (which documents the interference relation for each assignment instruction).

i	kill(i)	out(i)	interfere
1: LABEL start	-	a, b	-
2: IF a < b THEN next ELSE swap	-	a, b	-
3: LABEL swap	-	a, b	-
4: t := a	t	b, t	b
5: a := b	a	a, t	t
6: b := t	b	a, b	a
7: LABEL next	-	a, b	-
8: z := 0	z	a, b, z	a, b
9: b := b mod a	b	a, b, z	a, z
10: IF b = z THEN end ELSE start	-	a, b	-
11: LABEL end	-	a	-
12: RETURN a	-	-	-

d)

Draw the interference graph for a, b, t, and z



e)

Color the interference graph with 3 colors. Show the stack.

node	neigbour	color
t	-	1
b	t	2
a	b, t	3
z	a, b	1

f)

Color the interference graph with 2 colors. Select variables to spill, perform the spilling transformation and show the resulted program after spilling. You are not required to perform the whole analysis again, i.e., don't do the liveness analysis and graph coloring on the "spilled"code.

node	neigbour	color
t	-	1
b	t	2
a	b, t	spill
z	a, b	1

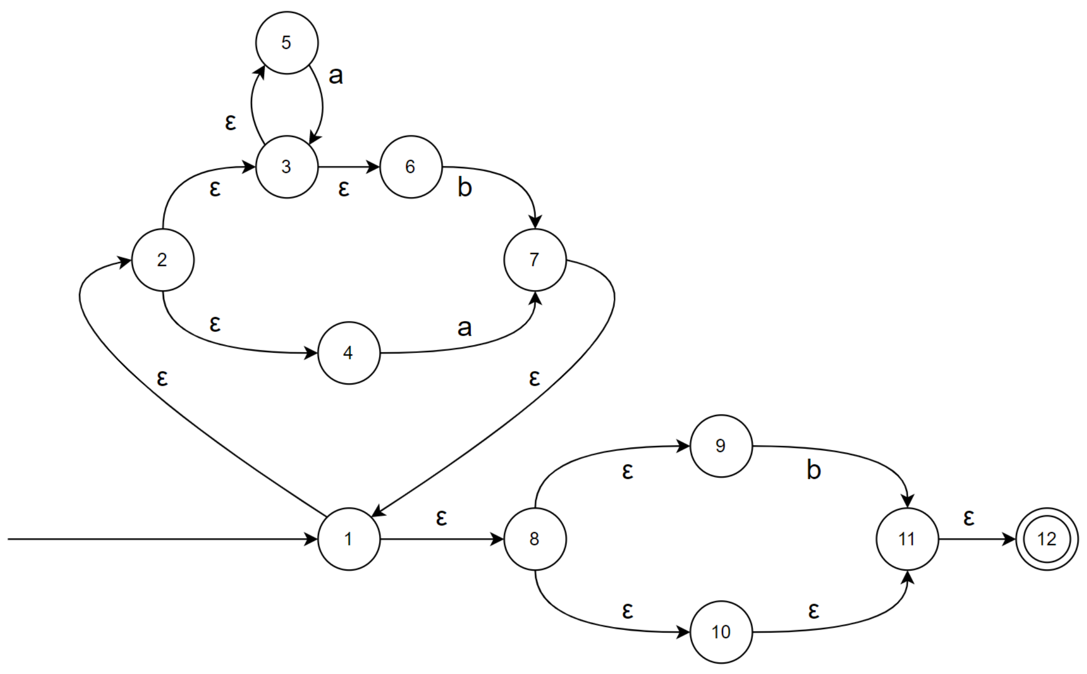
The code after spilling:

```
1:  LABEL start
    M[addr_a] := a_2
2:  IF a_2 < b THEN next ELSE swap
3:  LABEL swap
    M[addr_a] := a_4
4:  t := a_4
5:  a_5 := b
    M[addr_a] := a_5
6:  b := t
7:  LABEL next
8:  z := 0
    M[addr_a] := a_9
9:  b := b mod a_9
10: IF b = z THEN end ELSE start
11: LABEL end
    M[addr_a] := a_12
12: RETURN a_12
```

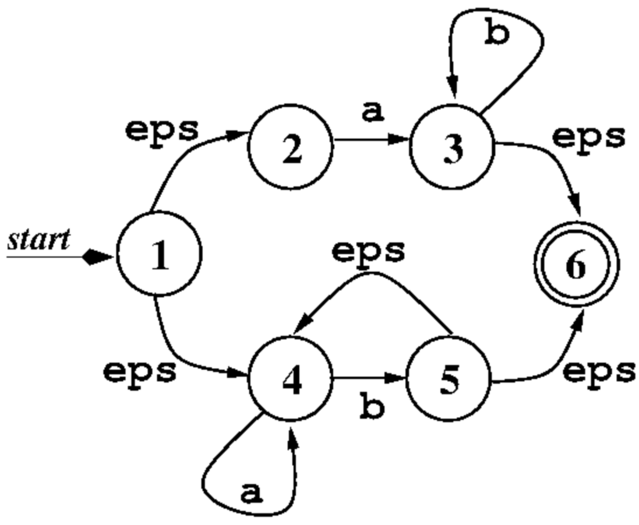
Task 3

$$(a^*b|a)^*(b|\varepsilon)$$

Using the systematic construction from Fig. 1.4 in the book, build the equivalent nondeterministic finite automaton (NFA).



Task 4



Sytematically derive the move function for all (about 10) pairs of DFA state and input character, showing how you construct each new state and transition.

Then draw the resulting DFA. Again, remember to indicate starting and accepting states, and make sure that each transition is labeled by a character from the alphabet.

The initial state in the DFA is ε-closure($\{1\}$), which is $s'_0 = \{1, 2, 4\}$.

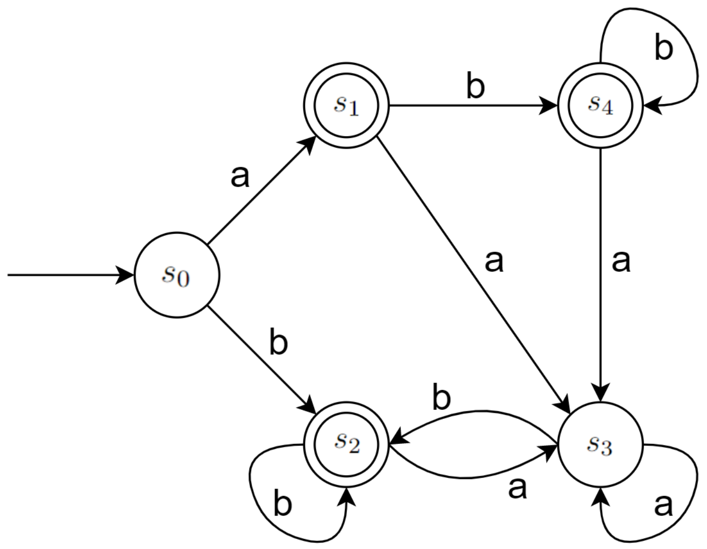
$$\begin{aligned}
move(s'_0,\ a) &= \varepsilon\text{-}closure(\{t \mid s \in \{1,2,4\}) \textit{ and } s^a t \in T\} \\
&= \varepsilon\text{-}closure(\{3,4\}) \\
&= \{3,4,6\} \\
&= s'_1 \\
move(s'_0,\ b) &= \varepsilon\text{-}closure(\{t \mid s \in \{1,2,4\}) \textit{ and } s^b t \in T\} \\
&= \varepsilon\text{-}closure(\{5\}) \\
&= \{4,5,6\} \\
&= s'_2 \\
move(s'_1,\ a) &= \varepsilon\text{-}closure(\{t \mid s \in \{3,4,6\}) \textit{ and } s^a t \in T\} \\
&= \varepsilon\text{-}closure(\{4\}) \\
&= \{4\} \\
&= s'_3 \\
move(s'_1,\ b) &= \varepsilon\text{-}closure(\{t \mid s \in \{3,4,6\}) \textit{ and } s^b t \in T\} \\
&= \varepsilon\text{-}closure(\{3,5\}) \\
&= \{3,4,5,6\} \\
&= s'_4 \\
move(s'_2,\ a) &= \varepsilon\text{-}closure(\{t \mid s \in \{4,5,6\}) \textit{ and } s^a t \in T\} \\
&= \varepsilon\text{-}closure(\{4\}) \\
&= \{4\} \\
&= s'_3 \\
move(s'_2,\ b) &= \varepsilon\text{-}closure(\{t \mid s \in \{4,5,6\}) \textit{ and } s^b t \in T\} \\
&= \varepsilon\text{-}closure(\{5\}) \\
&= \{4,5,6\} \\
&= s'_2
\end{aligned}$$

$$
\begin{aligned}
move(s'_3,\ a) &= \varepsilon\text{-}closure(\{t \mid s \in \{4\}) \ and \ s^a t \in T\} \\
&= \varepsilon\text{-}closure(\{4\}) \\
&= \{4\} \\
&= s'_3 \\
move(s'_3,\ b) &= \varepsilon\text{-}closure(\{t \mid s \in \{4\}) \ and \ s^b t \in T\} \\
&= \varepsilon\text{-}closure(\{5\}) \\
&= \{4,5,6\} \\
&= s'_2 \\
move(s'_4,\ a) &= \varepsilon\text{-}closure(\{t \mid s \in \{3,4,5,6\}) \ and \ s^a t \in T\} \\
&= \varepsilon\text{-}closure(\{4\}) \\
&= \{4\} \\
&= s'_3 \\
move(s'_4,\ b) &= \varepsilon\text{-}closure(\{t \mid s \in \{3,4,5,6\}) \ and \ s^b t \in T\} \\
&= \varepsilon\text{-}closure(\{3,5\}) \\
&= \{3,4,5,6\} \\
&= s'_4
\end{aligned}
$$

$$S' = \{s'_0, s'_1, s'_2, s'_3, s'_4\}$$

Since all states are now marked, this completes the construction of $S' = \{s'_0, s'_1, s'_2, s'_3, s'_4\}$. Where s'_1, s'_2, s'_4 contains the accepting NFA state 6.

DFA:



s_0
s_1
s_2
s_3
s_4
a
b
b
b
a
a
b
a
b
a