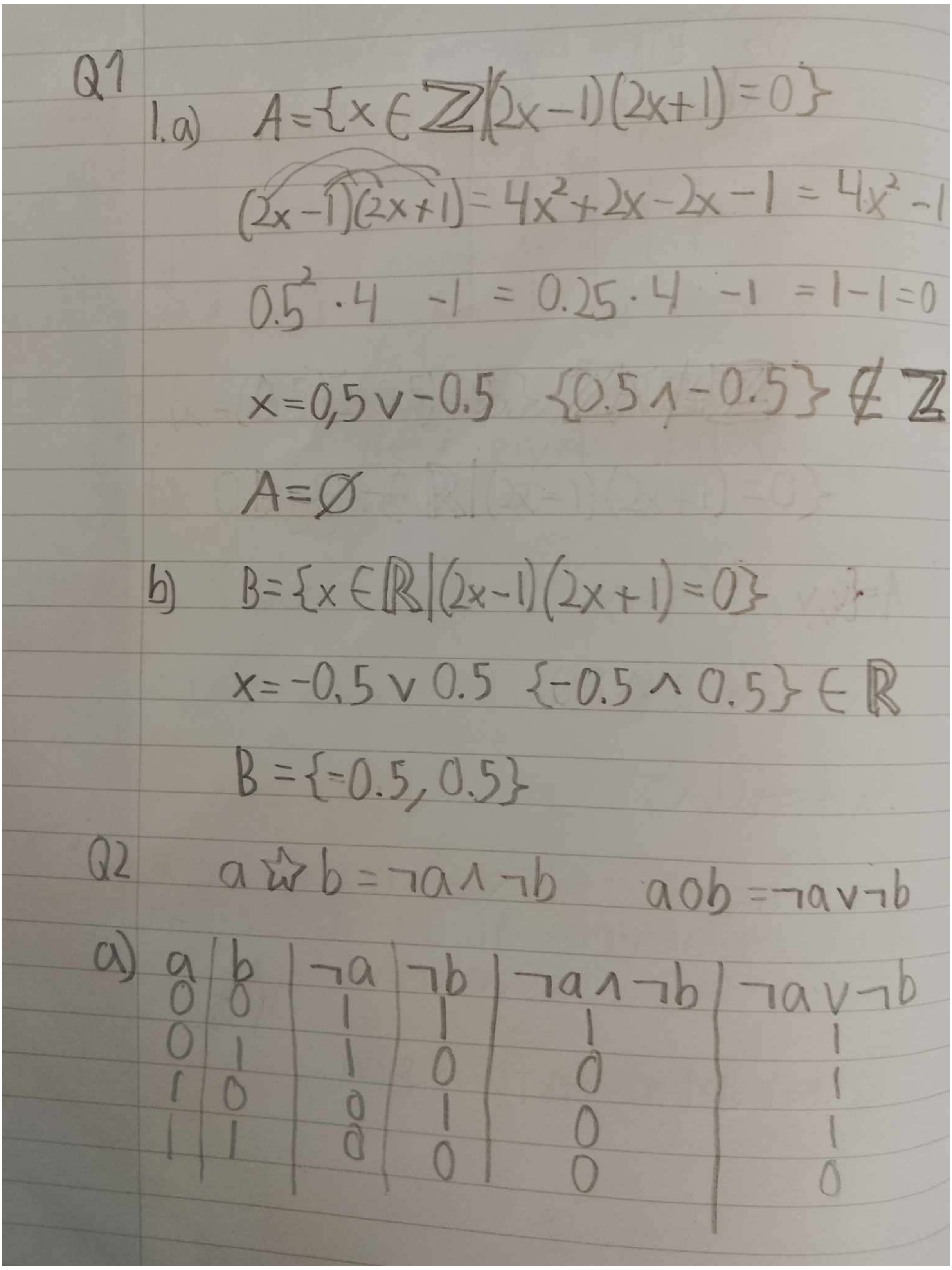
Jørgen Oltedal, Filip Olav Ulland, Johannes Tollessen Rosberg



Q1

1.a) $A=\{x\in\mathbb{Z}\mid(2x-1)(2x+1)=0\}$

$(2x-1)(2x+1)=4x^2+2x-2x-1=4x^2-1$

$0.5^2\cdot 4\ -1=0.25\cdot 4\ -1=1-1=0$

$x=0{,}5\vee -0.5\quad\{0.5\wedge -0.5\}\notin\mathbb{Z}$

$A=\emptyset$

b) $B=\{x\in\mathbb{R}\mid(2x-1)(2x+1)=0\}$

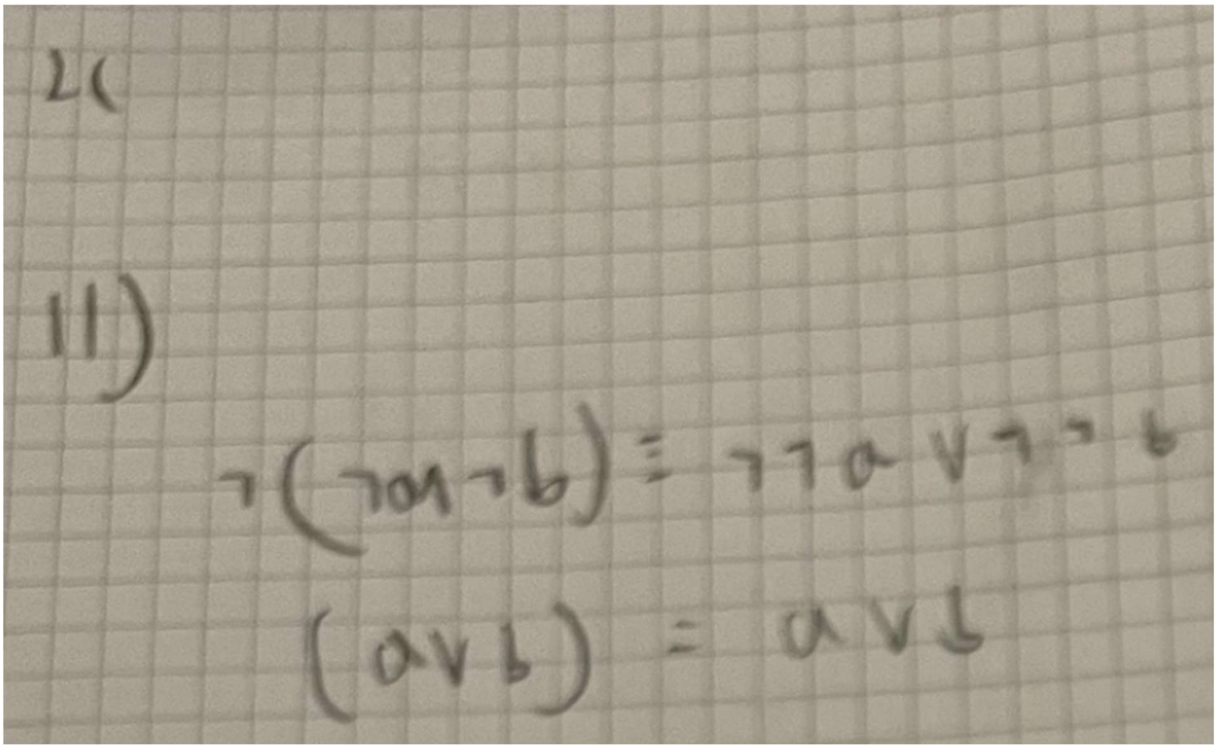
$x=-0.5\vee 0.5\quad\{-0.5\wedge 0.5\}\in\mathbb{R}$

$B=\{-0.5,0.5\}$

Q2 $a \mathbin{☆} b=\neg a\wedge\neg b\qquad a\circ b=\neg a\vee\neg b$

a)

a	b	$\neg a$	$\neg b$	$\neg a\wedge\neg b$	$\neg a\vee\neg b$
0	0	1	1	1	1
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	0

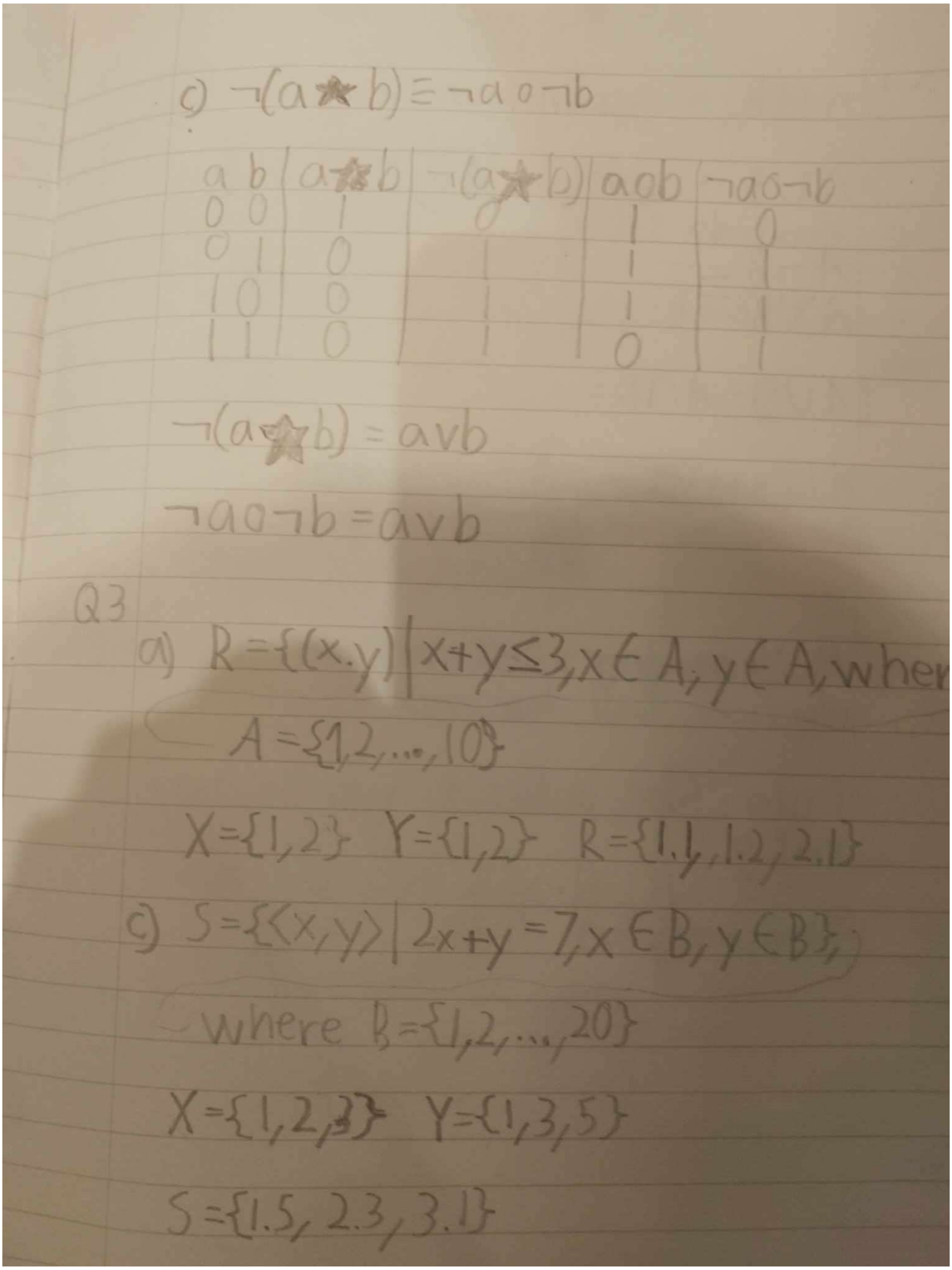


2(

11)

$\neg(\neg a \wedge \neg b) \equiv \neg\neg a \vee \neg\neg b$

$(a \vee b) = a \vee b$



c) $\neg(a \bigstar b) \equiv \neg a \circ \neg b$

a	b	$a \bigstar b$	$\neg(a \bigstar b)$	$a \circ b$	$\neg a \circ \neg b$
0	0	1	0	1	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	0	1	0	1

$\neg(a \bigstar b) = a \vee b$

$\neg a \circ \neg b = a \vee b$

Q3

a) $R = \{(x.y) \mid x+y \leq 3, x \in A, y \in A$, wher

$A = \{1, 2, \ldots, 10\}$

$X = \{1, 2\}$ $Y = \{1, 2\}$ $R = \{1.1, 1.2, 2.1\}$

c) $S = \{\langle x, y \rangle \mid 2x+y = 7, x \in B, y \in B\}$,

where $B = \{1, 2, \ldots, 20\}$

$X = \{1, 2, 3\}$ $Y = \{1, 3, 5\}$

$S = \{1.5, 2.3, 3.1\}$

Q5 $P(x)$ = all coffees $Q(x)$ = cappucino
$R(x)$ = good things

a) $\exists x((P(x) \land R(x)) \to Q(x))$ = Hvis, Det eksisterer en x, som er en god ting, så må det være en ~~god~~ cappucino.

b) $\forall x(\sim R(x) \to (P(x) \land Q(x)))$ = ~~Alle gode ting er både kaffe og cappucino~~

Alle ting som ikke er gode ~~er~~ må være kaffe og cappucino

Q6 $\exists x(R(x) \to (P(x) \land Q(x)))$

Q7 a) $\forall x \in \mathbb{Z}(\exists k \in \mathbb{Z} \mid (x \cdot k) < 0)$

b) $\forall \mathbb{Z} \mid 2a \cdot b = 2c$

```haskell
--Q2
module Main where
import Data.List (nub)

bools = [False, True]

main :: IO ()
main = do
  putStr "Q2\nnot a && not b: "
  print [[(not a && not b) | b <- bools] | a <- bools]
  putStr "not a || not b: "
  print [[(not a || not b) | b <- bools] | a <- bools]
  putStr"vi ser at 1110 og 1000 matcher\nsvarene vi fant for hånd"
```

```
runhaskell q2.hs
Q2
not a && not b: [[True,False],[False,False]]
not a || not b: [[True,True],[True,False]]
vi ser at 1110 og 1000 matcher
svarene vi fant for hånd
```

```
--Q3
module Main where
import Data.List (nub)

r :: [(Int, Int)]
r = [(x, y) | x <- a, y <- a, x + y <= 3]
  where a = [1..10]
x = nub [x | (x, y) <- r]
y = nub [y | (x, y) <- r]
firsts = [x | (x, y) <- r]

s = [(x, y) | x <- b, y <- b, 2 * x + y == 7]
  where b = [1..20]
firsts_s = nub [x | (x, y) <- s]

main :: IO ()
main = do
  putStr "Q3.a\n"
  putStr "R: "
  print r
  putStr "X: "
  print x
  putStr "Y: "
  print y
  putStr "\nQ3.b\n"
  putStr "first elements: "
  print firsts
  putStr "\nQ3.c\nS: "
  print s
  putStr "\nQ3.d\nfirst elements: "
  print firsts_s
```

```
runhaskell q3.hs

Q3.a
R: [(1,1),(1,2),(2,1)]
X: [1,2]
Y: [1,2]

Q3.b
first elements: [1,1,2]

Q3.c
S: [(1,5),(2,3),(3,1)]

Q3.d
first elements: [1,2,3]
```

```haskell
--Q4
module Main where
import Data.List (nub)


t_1 = [1..10]
t_2 = [11..20]
l_t = [(t_1!!i, t_2!!i) | i <- [0..(length t_1 - 1)]]


main :: IO ()
main = do
  putStr "Q4\ntuples of first elements: "
  print l_t
```

```
runhaskell q4.hs
Q4
tuples of first elements: [(1,11),(2,12),(3,13),(4,14),(5,15),(6,16),(7,17),(8,18),(9,19),(10,20)]
```