Jona Bako
jbako@jacobs.university.de

Homework - CS 2020 Problem Sheet #3

Problem 3.1

Prove or disprove the following two propositions:

a) $(A \cap B) \times (C \cap D) = (A \times C) \cap (B \times D)$

Let $\mathbf{(A \cap B) \times (C \cap D) = \{(a, b)\}}$

Then $\{a\} \in A \cap B$ and $\{b\} \in C \cap D$

Meaning both ($\{a\} \in A$ and $\{a\} \in B$) and ($\{b\} \in C$ and $\{b\} \in D$)

Since a is included in both A and B, and b is included in both C and D, we write:

$A \times C = \{(a, b)\}$ and $B \times D = \{(a, b)\}$

Now, $\mathbf{(A \times C) \cap (B \times D) = \{(a, b)\}}$

Therefore, the proposition is true.

---

b) $(A \cup B) \times (C \cup D) = (A \times C) \cup (B \times D)$

Let $\mathbf{(A \cup B) \times (C \cup D) = \{(x, y)\}}$

Then $\{x\} \in A \cup B$ and $\{y\} \in C \cup D$

Meaning $\{x\}$ is included in either A or B and $\{y\}$ is included in either C or D

Hence, we cannot say that the proposition is true in general, and we can simply proove that by a counter example. I.e. let $\{x\} \in A$ and $\{y\} \in D$, as well as two other values for sets B and C, i.e. $\{w\} \in B$ and $\{z\} \in C$. If we try the proposition, we will have:

$A \times C = \{(x, w)\}$ and $B \times D = \{(z, y)\}$

Now, $\mathbf{(A \times C) \cup (B \times D) = \{(x, w), (z, y)\}}$

Therefore, the proposition is false and disprooved.

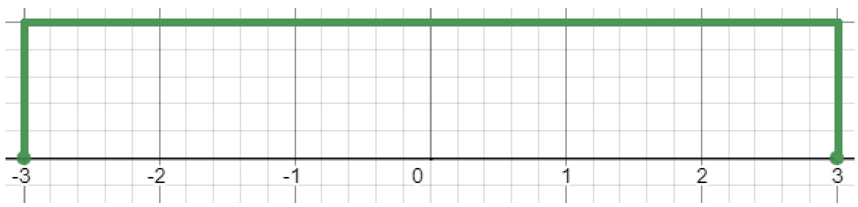
Problem 3.2

For each of the following relations, determine whether they are reflexive, symmetric, or transitive. Provide a reasoning.

a) $R = \{(a, b) | a, b \in Z \wedge |a - b| \leq 3\}$

(The absolute difference of the numbers a and b is less than or equal to 3.)

- **Reflexive:** $\forall\, a \in Z\ (a, a) \notin Z$ because a = a therefore $|a-a| = 0 \leq 3$
- **Symmetric:** $\forall\, a, b \in Z\ (a, b) \in Z$ because $| a-b | \leq 3$ means that:
  $-3 \leq a-b \leq 3$ (distance from 0 is 3)

  And if we multiply by -1 we get:
  $3 \geq b-a \geq -3$
  Therefore, for both (a, b) and (b, a) the absolute value of their difference is going to be 3.
- **Not Transitive:**
  $\forall\, a, b \in Z\ (a, b) \in R,\ |a - b| \leq 3$
  $\forall\, b, c \in Z\ (b, c) \in R,\ |b - c| \leq 3$
  This however doesn't mean that $|a - c| \leq 3$ and we can proove that by a counter example:
  i.e. a = 8, b = 5, c = 4 $|8 - 5| = 3 \leq 3$ and $|5 - 4| = 1 \leq 3$, but $|8 - 4| = 4 \nleq 3$

b) $R = \{(a, b) \mid a, b \in Z \wedge (a \bmod 10) = (b \bmod 10)\}$

(The last digit of the decimal representation of the numbers a and b is the same.)

- **Reflexive:** $\forall\, a \in Z$ we have $(a, a) \in Z$ because a mod 10 = a mod 10
- **Symmetric:** $\forall\, a \in Z\ (a, b) \in Z$ as a mod 10 = b mod 10 $\leftrightarrow$ b mod 10 = a mod 10
- **Transitive:** $\forall\, a, b \in Z\ (a, b) \in Z$, a mod 10 = b mod 10
  And $\forall\, b, c \in Z\ (b, c) \in Z$, b mod 10 = c mod 10
  Therefore, a mod 10 = c mod 10
  i.e. a = 23, b = 33, c = 43
  a%10=3, b%10=3, c%10=3

  $$\left.\begin{array}{l} a\%10 = b\%10 = 3 \text{ (TRUE)} \\ b\%10 = c\%10 = 3 \text{ (TRUE)} \end{array}\right\} \quad a\%10 = c\%10 = 3 \text{ (TRUE)}$$

Problem 3.3

Consider the two Haskell functions cnt and con defined below.
Proof by induction over s that cnt x (con s t) == (cnt x s) + (cnt x t) holds.

```
cnt :: Eq a => a -> [a] -> Int
cnt x [] = 0
cnt x (y:ys)
      | x == y = 1 + (cnt x ys)
      | otherwise = cnt x ys

con :: [a] -> [a] -> [a]
con [] ys = ys
con (x:xs) ys = x : (con xs ys)
```

Proof by Induction

To prove property **cnt x (con s t) == (cnt x s) + (cnt x t)** by induction we have:

- Base case: // Prove P([]) //

a) Consider s to be an empty string (base case s = []). Thus, we write:
On left hand side: On right hand side:
cnt x (con [] t) = cnt x t (cnt x []) + (cnt x t) = 0 + cnt x t = cnt x t
LHS = RHS -> The property is true for base case.

- Induction step: // Prove P(xs)(Induction Hypothesis) implies P(x:xs)(new variable x)

b) Assume the statement is true ∀ s.
cnt x (con s t) = (cnt x s) + (cnt x t)

c) Prove the statement is true for s with an additional element v. We assume v is added to the front of s. => cnt x (con v:s) t) = (cnt x v) + (cnt x t)
From patter matching of con we get:
con(x:xs)ys = x:con(xs:ys)
Resulting: cnt x(con(v:s)t) = cnt x(v:(con s t))
From the pattern matching of con
con x (y:ys)
x == y = 1 + (cnt x ys)
otherwise = cnt x ys
If x = v
Then, cnt x(v: (con s t)) = 1 + cnt x(con s t)
Recall cnt x(con s t) = (cnt x s) + (int x t)
0 + cnt x(con s t) = 0 + (cnt x s) + (cnt x t)
From the pattern matching of cnt, we know that
cnt(x (v:s)) = (cnt x s) = 0 t(cnt x s)
As a result 0 + (cnt x s) + (cnt x t) = cnt(x(v:s)) + (cnt x t)
In both cases, whether x is equal to v or not, we have proven that:
Cnt z(con (v:s) t) = (cnt x(v:s)) + (cnt x t)