Introduction to Computer Science - Homework 3

Rumen Mitov

September 28, 2023

Problem 3.1

Show:

$$A \cap (B \cup C) = (A \cap B) \cup (B \cap C)$$

Proof by equivalences:

L.H.S.

We know from definition:

$$\forall x \in (B \cup C) \mid x \in B \lor x \in C \tag{1}$$

Also:

$$\forall x \in (A \cap X) \mid x \in A \land x \in X \tag{2}$$

Hence, substituting X for $(B \cup C)$ in (*2*):

$$\forall x \in (A \cap (B \cup C)) \mid x \in A \land x \in (B \cup C)$$

Using what we know from (*1*):

$$\forall x \in (A \cap (B \cup C)) \mid x \in A \land (x \in B \lor x \in C)$$

R.H.S.

Using definitions (*1*) and (*2*):

$$\forall x \in ((A \cap B) \cup (A \cap C)) \mid (x \in A \wedge x \in B) \vee (x \in A \wedge x \in C)$$

We can tell from the above statement that $\forall x \in A$. Hence, the following simplification:

$$\forall x \in ((A \cap B) \cup (A \cap C)) \mid x \in A \wedge (x \in B \vee x \in C)$$

L.H.S. and *R.H.S* match.

$\therefore A \cap (B \cup C) = (A \cap B) \cup (B \cap C)$ is true. ■

Problem 3.2

a)

Prove or disprove:

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

Proof by equivalence:

Consider set **A** which contains elements $\{a_1, a_2, \ldots, a_n\}$, and set $\mathbf{B} = \{b_1, b_2, \ldots, b_m\}$.

Now let $A \cap B = \{x_1, x_2, \ldots, x_k\}$.

Similarly, set **C** which contains elements $\{c_1, c_2, \ldots, c_p\}$, and set $\mathbf{D} = \{d_1, d_2, \ldots, d_q\}$.

Now let $C \cap D = \{y_1, y_2, \ldots, y_l\}$.

L.H.S.

$$\forall x_i \in (A \cap B) \mid \forall y_i \in (C \cap D) \mid (A \cap B) \times (C \cap D) = \{(x, y), (x_2, y_2), \ldots, (x_k, y_l)\}$$

R.H.S

$$\forall a_i \in A | \forall c_i \in C \mid (A \times C) = \{(a_1, c_1), (a_1, c_2), \ldots (a_2, c_1), \ldots, (a_n, c_p)\}$$

Similarly:

$$\forall b_i \in B | \forall d_i \in D \mid (B \times D) = \{(b_1, d_1), (b_1, d_2), \ldots (b_2, d_1), \ldots, (b_m, d_q)\}$$

Finally:

$$(A \times C) \cap (B \times D) = \{(x, y), (x_2, y_2), \ldots, (x_k, y_l)\}$$

This is because $\forall x \in (A \cap B) \mid \forall y \in (C \cap D) \mid (x, y)$ are the only tuples that will belong to both $(A \times C) \wedge (B \times D)$, since x belongs to both **A** and **B** and y belongs to both **C** and **D**.

Both *L.H.S.* and *R.H.S.* match.

$\therefore (A \cap B) \times (C \cap D) = (A \times C) \cap (B \times D)$ is true. ■

b)

<u>Prove or disprove:</u>

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

<u>Proof by equivalence:</u>

Consider set **A** which contains elements $\{a_1, a_2, \ldots, a_n\}$, and set $\mathbf{B} = \{b_1, b_2, \ldots, b_m\}$.

Now let $A \cup B = \{x_1, x_2, \ldots, x_k\}$.

<u>NOTE:</u> $A \cup B$ contains all elements in **A** and all elements in **B**.

Similarly, set **C** which contains elements $\{c_1, c_2, \ldots, c_p\}$, and set $\mathbf{D} = \{d_1, d_2, \ldots, d_q\}$.

Now let $C \cup D = \{y_1, y_2, \ldots, y_l\}$.

NOTE: $C \cup D$ contains all elements in **C** and all elements in **D**.

L.H.S.

$$\forall x_i \in (A \cup B) \mid \forall y_i \in (C \cup D) \mid (A \cup B) \times (C \cup D) = \{(x, y), (x_2, y_2), \ldots, (x_k, y_l)\}$$

R.H.S

$$\forall a_i \in A | \forall c_i \in C \mid (A \times C) = \{(a_1, c_1), (a_1, c_2), \ldots (a_2, c_1), \ldots, (a_n, c_p)\}$$

Similarly:

$$\forall b_i \in B | \forall d_i \in D \mid (B \times D) = \{(b_1, d_1), (b_1, d_2), \ldots (b_2, d_1), \ldots, (b_m, d_q)\}$$

Finally:

$$(A \times C) \cup (B \times D) = \{(x, y), (x_2, y_2), \ldots, (x_k, y_l)\}$$

This is because $\forall x \in (A \cup B) \mid \forall y \in (C \cup D) \mid (x, y)$ are the in fact all the tuples that will belong to $(A \times C) \cup (B \times D)$, since x represents all the elements in **A** and all the elements in **B**. Similarly, y represents all the elements in **C** and all the elements in **D**.

Both *L.H.S.* and *R.H.S.* match.

$\therefore (A \cup B) \times (C \cup D) = (A \times C) \cup (B \times D)$ is true. $\blacksquare$

Problem 3.3

Definition 1: Relation, R, is *reflexive* $\iff \forall x \in X \mid (x, x) \in R$.

Definition 2: Relation, R, is *symmetric* $\iff \forall x, y \in X \mid (x, y) \in R \implies (y, x) \in R$.

Definition 3: Relation, R, is *transitive* $\iff$ $\forall x, y, z \in X \mid ((x, y) \in R \wedge (y, z) \in R) \implies (x, z) \in R$.

a)

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

Reflexivity:

$$(a, a) \implies \mid a - a \mid = 0 \leq 3 \implies (a, a) \in R$$

Symmetry:

$$(a, b) \implies \mid a - b \mid = \mid b - a \mid \implies ((a, b) \in R \implies (b, a) \in R)$$

Transitivity:

Let $\mid a - b \mid = d_1$ and $\mid b - c \mid = d_2$.

We assume the $\{d_1, d_2\} \in R$. However, if $d_1 + d_2 > 3$ then:

$$(a, c) \implies \mid a - c \mid > 3 \implies (a, c) \notin R$$

Hence, R is **reflexive**, **symmetric**, but **not transitive**.

b)

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

Reflexivity:

$$(a, a) \implies a \mod 10 = a \mod 10 \implies (a, a) \in R$$

Symmetry:

$$(a,b) \implies a \mod 10 = b \mod 10 \iff b \mod 10 = a \mod 10 \implies ((a,b) \in R \implies (b,a) \in R)$$

Transitivity:

Assume $\{(a,b),(b,c)\} \in R$.

Let $a = 10p \cdot b, p \in \mathbb{N}$ and $b = 10q \cdot c, q \in \mathbb{N}$.

Then:

$$a = 10p \cdot (10q \cdot c) = 100pq \cdot c, pq \in \mathbb{N}$$

Hence:

$$(a,c) \implies a \mod 10 = 100pq \cdot c = c \mod 10 \implies (a,c) \in R$$

Hence, R is **reflexive**, **symmetric**, and **transitive**.

Problem 3.4

a)

The type signature of the *zip* function is:

```
zip :: [a] -> [b] -> [(a, b)]
```

The function receives two variables. The types of the variables can be anything (as shown by the type variables **a** and **b**). The type listing shows exactly three types: variable type **a**, variable type **b** and a tuple combining the two types **a** and **b**.

NOTE: The function could take in less types if **a** and **b** were of the same type.

b)

2 + 3 is a **Num a => a**, meaning it is an expression of any numerical types (e.g. **Integers**, **Floats**), and it returns the same type of value it received.

2 + 9 ‘div‘ 3 is a **Integral a => a**, meaning it is an expression that supports only **Integers**, and it returns the same type of value it received.

2 + 9 / 3 is a **Fractional a => a**, meaning it is an expression that takes in non-**Integral** numbers (e.g. **Floats**, **Doubles**), and it returns the same type of value it received.

2 + sqrt 9 is a **Floating a => a**, meaning it is an expression that takes in either **Floats** or **Doubles**, and it returns the same type of value it receives.