JACOBS UNIVERSITY BREMEN

Assignment #4

Student name: *Ivan Kabadzhov*

Course: *Introduction to Computer Science*
Date: *October, 2018*

Problem 4.1: prefix order relations

a) Claim: $\preceq \subseteq \sum^\star \times \sum^\star$ being a relation such that $p \preceq w$ for $p, w \in \sum^\star$ if p is a prefix of w, then $\preceq$ is a partial order

Proof: *Proof by Construction*

1. reflexive: a prefix might be ϵ, so that w could be its own prefix $\Rightarrow (p, p) \in R \Leftrightarrow p \sim p$.
2. antisymmetric: assuming that p is $\epsilon \Rightarrow (p, w) \in \preceq \wedge (w, p) \in \preceq \Rightarrow (p, w) \sim (w, p)$.
3. transitive: a prefix of the prefix is also a prefix of the word.

$\therefore$ The relation is reflexive, antisymmetric and transitive so it is a partial order relation.

■

b) Claim: $\prec \subset \sum^\star \times \sum^\star$ being a relation such that $p \prec w$ for $p, w \in \sum^\star$ if p is a proper prefix of w, then $\prec$ is a strict partial order

Proof: *Proof by Construction*

1. irreflexive: a prefix cannot be ϵ, so that w could not be its own prefix $\Rightarrow (p, p) \notin R \Leftrightarrow p \nsim p$.
2. asymmetric: provided that p cannot be ϵ and $(p, w) \in \prec \Rightarrow (w, p) \notin \prec$.
3. transitive: a prefix of the prefix is also a prefix of the word.

$\therefore$ The relation is reflexive, asymmetric and transitive so it is a strictly partial order relation.

■

c) Claim: $\prec$ and $\preceq$ being order relations are not total

Proof: *Proof by Construction*

A total order relation implies that $\forall a, b \in X,\ aRb \vee bRa$. This might not be satisfied in the $\prec$ and $\preceq$ order relations.

$\blacksquare$

Problem 4.2: function composition

Given $f : A \to B \wedge B \to C$, where A, B, C are sets.

a) Claim: $g \circ f$ is bijective $\Rightarrow$ f is injective and g is surjective

Proof: *Proof by Construction*

First, I consider $g \circ f$ as being injective, which means, that every element of the codomain (C) is mapped to by at most 1 element of the domain (A). Assume that $x, y \in A$ and the question is if $f(x) = f(y)$. Therefore since $g \circ f$ is injective in our case we have $g(f(x)) = g(f(y))$. So, I have $x = y$, which means that f is injective.

Secondly, I use that $g \circ f$ is also surjective, which implies that every element of the codomain (C) is mapped to by at least 1 element of the domain (A). Let $y \in C$ and following the definition $\exists x \in A$ such that $g(f(x)) = y$. Hence, if $z = f(x) \in B$ I have $g(z) = y$. This means, that g is surjective.

$\blacksquare$

b) Find an example: $g \circ f$ is not bijective even though f is injective, and g is surjective

Example:
Let $A = \{x\}$, $B = \{x, y, z\}$, $C = \{x, y\}$. I have $f : A \to B$, which implies $f(x) = x$. Thus, f is injective, because every element of the codomain (B) is mapped to by at most 1 element of the domain (A).

For $g : B \to C$, I can say that $g(x) = x$ and $g(y) = g(z) = y$. Thus, g is surjective, because every element in C is mapped with at least 1 element of B. However, provided that $g \circ f : A \to C$, it cannot be said that every element of the codomain 'C can be matched with exactly 1 element of the domain A. In this case, $g \circ f$ is only injective ($f(g(x)) = x$).

Alternative approach for this task is to have $A = \{x, y\}$, $B = \{x, y, z\}$, $C = \{x\}$ and then $g \circ f$ would be only surjective ($f(g(x)) = f(g(y)) = x$). Again, f is going to be injective and g – surjective, following the same logic.

c) Find an example: $g \circ f$ is bijective even though f is not surjective, and g is not injective

Example:
Suppose that $A = \{x\}$, $B = \{x, y\}$, $C = \{x\}$. Then from $f : A \to B$, I have $f(x) = x$. Thus, f is not surjective, since y in B is not matched. From $g : B \to C$, I have $g(x) = g(y) = x$. Thus, g is not injective, because the element in the codomain (C) is mapped to more than one element of the domain (B). For $g \circ f = g(f)$ I have $g \circ f : A \to C$, which is $g(f(x)) = x$. Bijective from $A = \{x\}$ to $C = \{x\}$, which means that it is both injective and surjective.

Problem 4.3: suffixes and prefixes (haskell)

```haskell
suffixes :: [a] -> [[a]]
suffixes [] = [] --base case
suffixes (x:xs) = xs : suffixes xs --recursion
--recursion here is the tail of the tail of the tail...
--until the empty string is reached (the base case)

prefixes :: [a] -> [[a]]
prefixes []  =  [] --base case
prefixes (x:xs) = [] : map (x:) (prefixes xs) --recursion
--the map is the list of applying heads to each tail(as they are being recursed)
--therefore the first head is the empty string []
```