Introduction to Computer Science
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: CH-232-A
Date: 2019-10-04
Due: 2019-10-11

ICS 2019 Problem Sheet #4

Problem 4.1: *prefix order relations* (2+2+1 = 5 points)

Let Σ be a finite set (called an alphabet) and let Σ^* be the set of all words that can be created out the symbols in the alphabet Σ. (Σ^* is the Kleene closure of Σ, which includes the empty word ϵ.) A word $p \in \Sigma^*$ is called a prefix of a word $w \in \Sigma^*$ if there is a word $q \in \Sigma^*$ such that $w = pq$. A prefix p is called a proper prefix if $p \neq w$.

a) Let $\preceq \subseteq \Sigma^* \times \Sigma^*$ be a relation such that $p \preceq w$ for $p, w \in \Sigma^*$ if p is a prefix of w. Show that $\preceq$ is a partial order.

b) Let $\prec \subset \Sigma^* \times \Sigma^*$ be a relation such that for $p \prec w$ for $p, w \in \Sigma^*$ if p is a proper prefix of w. Show that $\prec$ is a strict partial order.

c) Are the two order relations $\preceq$ and $\prec$ total?

Make sure you write complete proofs for the properties of the order relations. Do not assume something is 'obvious' or 'trivial' — always reason with the definition of the order relation.

Problem 4.2: *function composition* (2+1+1 = 4 points)

Let A, B and C be sets and let $f : A \to B$ and $g : B \to C$ be two functions.

a) Prove the following statement: If $g \circ f$ is bijective, then f is injective and g is surjective.

b) Find an example demonstrating that $g \circ f$ is not bijective even though f is injective and g is surjective.

c) Find an example demonstrating that $g \circ f$ is bijective even though f is not surjective and g is not injective.

Problem 4.3: *special prime numbers (haskell)* (1 point)

A *special prime number* is a prime number that can be written as the sum of two neighboring prime numbers and 1. Two prime numbers are called neighboring if there are no other prime numbers between them. An example for a special prime number is $19 = 7 + 11 + 1$.

a) Implement a function `isSpecialPrime :: Integer -> Bool` indicating whether the argument is a special prime number or not.

```
> isSpecialPrime 19
True
> filter isSpecialPrime [2..100]
[13,19,31,37,43,53,61,79]
```

Explain how your function works and how you have tested it.

Submit your Haskell source code as a plain text file.