Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Mohammad Abdulaziz, Ph.D.

SS 2020
24. 4. 2020

Functional Data Structures

Exercise Sheet 1

Before beginning to solve the exercises, open a new theory file named `ex01.thy` and write the the following three lines at the top of this file.

theory *ex01*
imports *Main*
begin

Exercise 1.1 Calculating with natural numbers

Use the **value** command to turn Isabelle into a fancy calculator and evaluate the following natural number expressions:

"$2 + (2{::}nat)$" "$(2{::}nat) * (5 + 3)$" "$(3{::}nat) * 4 - 2 * (7 + 1)$"

Can you explain the last result?

Exercise 1.2 Natural number laws

Formulate and prove the well-known laws of commutativity and associativity for addition of natural numbers.

Exercise 1.3 Counting elements of a list

Define a function which counts the number of occurrences of a particular element in a list.

fun *count* :: "$'a\ list \Rightarrow 'a \Rightarrow nat$"

Test your definition of *count* on some examples and prove that the results are indeed correct.

Prove the following inequality (and additional lemmas, if necessary) about the relation between *count* and *length*, the function returning the length of a list.

theorem "$count\ xs\ x \le length\ xs$"

Exercise 1.4 Adding elements to the end of a list

Recall the definition of lists from the lecture. Define a function *snoc* that appends an element at the right end of a list. Do not use the existing append operator @ for lists.

fun *snoc* :: "'*a list* ⇒ '*a* ⇒ '*a list*"

Convince yourself on some test cases that your definition of *snoc* behaves as expected, for example run:

value "*snoc* [] *c*"

Also prove that your test cases are indeed correct, for instance show:

lemma "*snoc* [] $c = [c]$"

Next define a function *reverse* that reverses the order of elements in a list. (Do not use the existing function *rev* from the library.) Hint: Define the reverse of x # xs using the *snoc* function.

fun *reverse* :: "'*a list* ⇒ '*a list*"

Demonstrate that your definition is correct by running some test cases, and proving that those test cases are correct. For example:

value "*reverse* $[a, b, c]$"
lemma "*reverse* $[a, b, c] = [c, b, a]$"

Prove the following theorem. Hint: You need to find an additional lemma relating *reverse* and *snoc* to prove it.

theorem "*reverse* (*reverse* xs) = xs"

Homework 1 Maximum Value in List

Submission until Friday, 1 May, 10:00am.

Submission Instructions: Submissions are handled via https://competition.isabelle.systems/. Submit a theory file that runs in Isabelle-2020 **without errors**.

- Register an account in the system and send the tutor an e-mail with your username.
- Select the competition "FDS2020" and submit your solution following the instructions on the website.
- The system will check that your solution can be loaded in Isabelle-2020 without any errors and reports how many of the main theorems you were able to prove.
- You can upload multiple times; the last upload before the deadline is the one that will be graded.
- If you have any problems uploading, or if the submission seems to be rejected for reasons you cannot understand, please contact the tutor.
- We will be using a clone detection tool to compare solutions so please do NOT add any personal or identifying information to your homework solution theory files.

General hints:

- If you cannot prove a lemma, that you need for a subsequent proof, assume this lemma by using sorry.
- Define the functions as simply as possible. In particular, do not try to make them tail recursive by introducing extra accumulator parameters – this will complicate the proofs!
- All proofs should be straightforward, and take only a few lines.

Define a function that returns the maximal element of a list of natural numbers. The result for the empty list shall be 0.

fun *lmax* :: *"nat list* $\Rightarrow$ *nat"*

Show that the maximum is greater or equal to every element of the list.

lemma *max_greater*: *"*$x \in$ *set xs* $\Longrightarrow x \leq$ *lmax xs"*

Note: the function *set* converts a list to the set of its elements.

Prove that reversing the list does not affect its maximum. Note that we use the *reverse* function from exercise 4 here.

lemma *"lmax* (*reverse xs*) = *lmax xs"*

Hint: Induction. You may need an auxiliary lemma about *lmax* and *snoc*.