Concepts of Programming Languages

Assignment 1

March 30, 2019

Deadline for submission is 13 April 2019. All tasks will be graded equally. Solve the tasks independently, in a case of cheating the university rules will be applied. Submit one PDF file containing your full name and enrollment number to `http://e.famnit.upr.si, where you` will also receive your grade.

Exercise 1. Write a function

```
insert: ('a list, 'a list) -> int -> 'a list
```

that takes a 2-tuple of lists (`list1, list2`) and an integer i as parameters, and returns a list in which the second list is inserted into the first list at the i-th position.

Example:

```
# unija ([1;2;4;7], [2;4;7;9]) 2;;
- : int list = [1;2;2;4;7;9;4;7]
```

Exercise 2. Write a function

```
anagr: ('a list, 'a list) -> bool
```

that takes two lists (`list1, list2`) and returns `true` if the lists are anagrams. Two lists are anagrams if the elements from one may be rearranged to get the other.

Example:

```
# anagr ([1;2;4;7], [2;4;7;9]);;
- : bool = false
```

Exercise 3. Write a function

```
pairs: 'a list -> ('a * 'a) list
```

that takes a list as parameter, and collects the reverse pairs from the list into list of pairs. If the input list is of odd length, the middle element should be ignored.

Example:

```
# pairs [1;2;4;2;1;7;4];;
- : int list = [(1,4);(2,7);(4,1)]
```

Exercise 4. Write a function

```
logic: (bool * bool) -> char -> bool
```

that takes two boolean values (p,q) and a character which encodes a binary logical operator, and returns the corresponding truth value based on the truth table of the chosen operator. The encoding of the binary logical operators, given atomic statements p and q, is given as follows:

$$\begin{aligned}
\texttt{A (for and)}: &\quad p \wedge q\\
\texttt{O (for or)}: &\quad p \vee q\\
\texttt{X (for xor)}: &\quad p \oplus q\\
\texttt{I (for implication)}: &\quad p \Rightarrow q\\
\texttt{any other character}: &\quad \neg p
\end{aligned}$$

Example:

```
# logic (false, true) 'I';;
- : bool = true
```

You can find additional information about the truth tables in the following link https://en.wikipedia.org/wiki/Truth_table