Home exam 1

1. Introduction

In this assignment, we implement a set ADT. More specifically, we implement an ordered set, which allows us to iterate over the elements in the set, in sorted order.

We also implement a second application, spamfilter. This is an application that classifies mails as either spam or not spam.

The requirements given for the implementation of the set ADT are as follows:

* Adding an element to the set.
* Getting the current size of the set.
* Checking whether a specific element is contained in the set.
* Getting the union of the set and another set.
* Getting the intersection of the set and another set.
* Getting the set difference between the set and another set.
* Iterating over the elements of the set, in sorted order.

1. Technical background

In my implementation of the set ADT, I have created sets containing the list struct, so that i can easily access the list functions. For a visual representation of how this works, see Figure 1.

What I needed to know before solving this problem:

* How headerfiles communicate with the c-files via the Makefile.
* A solid understanding of how structs work, and how to manipulate the elements inside them, specifically how to manipulate and access lists via the set\_t struct.
* How linked lists work, and the definition of a set.
* Definitions needed to solve certain functions, such as union\_set().

1. Design
2. Implementation
3. Discussion
4. Evaluation
5. Conclusion
6. References