Programming Exercises 2: Sets

Submissions only via Moodle (due to 23:55, Sep. 09)

September 1, 2014

In this assignment, you will have to implement a "package" of (finite) sets of elements using the `list` data type.

```
type 'a set = 'a list;;
```

Examples:

```
s1 = ["hello"; "world"; "community"; "manager"; "stuff"; "blue"; "green"]

s2 = ["hello"; "world"; "stuff"; "blue"; "green"; "red"]

s3 = [1; 2; 3]
```

Now you will need to implement the standard operations on sets. In particular, you will need to implement the following functions

1. A distinguished set called `emptyset: 'a set`
2. A function `member` that given an element x and a set s, returns true if $x \in s$ and false otherwise. `member: 'a -> 'a set -> bool`
3. A function `subseteq` that given two sets s_1 and s_2 returns true if $s_1 \subseteq s_2$ and false otherwise. That is, every member of s_1 should be a member of s_2 if $s_1 \subseteq s_2$. `subseteq: 'a set -> 'a set -> bool`
4. A function `seteq` that given two sets s_1 and s_2 returns true if $s_1 = s_2$ and false otherwise. Two sets are equal if they have exactly the same members (though the order of writing them may differ. `seteq: 'a set -> 'a set -> bool`
5. A function `setdiff` that given two sets s_1 and s_2 (elements of the same type) returns the set difference $s_1 - s_2$. Remember that $x \in s_1 - s_2$ if $x \in s_1$ and $x \notin s_2$. `setdiff: 'a set -> 'a set -> 'a set`

   Examples:

`s1-s2 = ["community"; "manager"]`

and

`s2-s1 = ["red"]`

6. A function `union` that given two sets s_1 and s_2 (elements of the same type) returns their union $s_1 \cup s_2$. Remember that $x \in s_1 \cup s_2$ if $x \in s_1$ or $x \in s_2$. `union: 'a set -> 'a set -> 'a set`

7. A function `intersect` that given two sets s_1 and s_2 (elements of the same type) returns their intersection $s_1 \cap s_2$. Remember that $x \in s_1 \cap s_2$ if $x \in s_1$ and $x \in s_2$. `intersect: 'a set -> 'a set -> 'a set`

8. A function`powerset` that given a set s, returns its *powerset* $\mathcal{P}(s)$, i.e., the set of all its subsets. Recall that $s' \in \mathcal{P}(s)$ if $s' \subseteq s$. `powerset: 'a set -> ('a set) set`

   Example:

   `powerset s3 = [[]; [1]; [2]; [3]; [1; 2]; [1; 3]; [2; 3]; [1; 2; 3]].`

9. A function `cartesian` that given two sets s_1 and s_2 (elements of the possibly different types) returns their cartesian product $s_1 \times s_2$, which consists of pairs, the first element of which is from s_1 and the second from s_2.