```
(*************************************************************************
******** definition of very simple abstype for Sets
******** Language SML/NY
******** Author M. Bertol (c)
*************************************************************************)


abstype 'a Set = set of 'a list
with                                                                    10
    exception emptySet;

    val emptyset = set([]);                                             emptyset

    fun is_empty(set([])) = true                                        is_empty
      | is_empty(_) = false

    fun singleton(x) = set([x])                                         singleton

    fun union(set(s),set(t)) = let fun append([],l) = l                 union
                                   | append(x::l,l') = x::append(l,l')   21
                               in set(append(s,t))
                               end

    fun member(x,set(l)) = let fun list_member(_,[]) = false            member
                             | list_member(x,y::l) = if (x=y) then true
                                                     else list_member(x,l)
                           in list_member(x,l)
                           end
                                                                        30
    fun remove(x,set(l)) = let fun list_remove(x,[]) = []               remove
                             | list_remove(x,y::l) = if (x=y) then list_remove(x,l)
                                                     else y::list_remove(x,l)
                           in set(list_remove(x,l))
                           end
    fun insert(x,set(l)) = set(x::l)                                    insert

    fun singleton_split(set([])) = raise emptySet                       singleton_split
      | singleton_split(set(x::s)) = (x, remove(x,set(s)))
                                                                        40
    fun split(set([])) =  raise emptySet                                split
      | split(set(a::l)) = (set([a]),set(l))

    fun is_subset(set([]),_) = true                                     is_subset
      | is_subset(set(x::s), t) = member(x,t) andalso is_subset(set(s),t)

    fun set_eq(s,t) = is_subset(s,t) andalso is_subset(t,s)             set_eq

    fun cardinality(set([])) = 0                                        cardinality
      | cardinality(set(a::s)) = 1 + cardinality(remove(a,set(s)))      50

    fun set_to_list(set(l)) = l                                         set_to_list

    fun list_to_set(l) = set(l)                                         list_to_set

    fun mapset(f)(set(l)) = let fun maplist(f)([]) = []                 mapset
                                  | maplist(f)(h::t) = (f(h))::(maplist(f)(t))   maplist
                            in set(maplist(f)(l))
                            end
end;                                                                    60
```