Contents

1 Module `Sread_expr`

```
type  paren_type =
  | Paren
  | Brace
  | Bracket
type  expr =
  | List of (paren_type * expr list)
  | Atom of string
val string_of_expr : expr -> string
```

2 Module `Sread` : Sread: Scheme-like READ for Ocaml

Author(s): Danny Yoo (dyoo at hkn.eecs.berkeley.edu)

This library makes it relatively easy to parse out Scheme lists into OCaml data structures. For example:

Sread.read_from_string "(this is a test '(yes it is))";; - : Sread.sexpr = Sread.List [Sread.Atom "this"; Sread.Atom "is"; Sread.Atom "a"; Sread.Atom "test"; Sread.List [Sread.Atom "QUOTE"; Sread.List [Sread.Atom "yes"; Sread.Atom "it"; Sread.Atom "is"]]]

```
type  sexpr =
  | List of sexpr list
  | Atom of string
```

AST that represents the parsed list structure

```
val read_from_channel : Pervasives.in_channel -> unit -> sexpr
```

Given an in_channel, returns a function that, when called, returns the next sexpr that can be parsed from the in_channel.

```
val read_from_string : string -> sexpr
```

Given a string, returns the first sexpr that can be parsed from that string

```
val string_of_sexpr : sexpr -> string
```

Given an sexpr, returns a string representation of that sexpr.