mbScheme

1°) Introduction

Lisp is a family of programming languages invented by John McCarthy in 1958 at the Massachusetts Institute of Technology (MIT) of which Common Lisp is the best known descendant. The Scheme programming language is a programming language derived from the Lisp language, created in the 1970s at MIT by Gerald Jay Sussman and Guy L. Steele. Among the languages derived from Scheme the Racket language is one of the most important because of its specific tools. Wikipedia easily provides information pages for Lisp, Scheme and Racket on Internet at the following addresses:

https://en.wikipedia.org/wiki/Lisp_(programming_language),
https://en.wikipedia.org/wiki/Scheme_(programming_language)
https://en.wikipedia.org/wiki/Racket_(programming_language).

There are also various simple or advanced Scheme interpreters and tutorials on Internet. mbScheme is a Scheme language interpreter whose executable program is programmed in Rust language by Mr. Martin Billinger at https://github.com/mbillingr/interpreter. This Windows program, interpreter.exe, provides the native functions that can be used. mbScheme additionally includes some additional functions which are defined by the user and which provide other functions of the Scheme language.

2°) Native functions

To use native functions only, the interpreter.bat file allows this possible choice. The list of native functions is obtained with the instruction: (print-env).

The elementary entities, called atoms, which can be used are as follows.

- the boolean constants: #t and #f

- integer numbers, examples: 123, -456, 9876543210

- rational numbers, examples: 7/5, -41/101, 852369741/987456321

- real numbers, examples: 3.14, -852.258

- complex numbers, examples: 0+i, 1+2i, 1-2i, -3.4+1.2i

- characters, examples: #\A, #\a, #\space, #\+, # \", #\!

- character strings, examples: "", "Hello World !", "that's correct"

Integers and rational numbers have no limit, they are as long as needed, and they allow mathematically exact arithmetic calculations to be performed. Integers are also known as elements of rationals, rationals as elements of reals and reals as elements of complexes. Real numbers have limited precision, they are stored in 64-bit floats: the mantissa has no more than 16 decimal digits and the exponent is between -308 and +308, that is why there is also the four specific real numbers: +inf.0, -inf.0, +nan.0 and -nan.0.

There are two composite entities. The list is mainly used, examples: (1 2 3 4), ("I have" 2 "tutorials"). The vector, example: [point 1 2], is used to show a particular use.

The native functions mainly include the following.

- the predicats: boolean?, integer?, rational?, real?, complex?, char?, string?, string=?, vector?, number?, procedure?, null? (empty list), pair? (non empty list), exact? (not limited), eq?, eqv?, equal?, symbol? (a valid variable name), file?.

- the numeric operators: +, -, *, /, <, =, >.

- the list operators: car, cdr, cons.

- the converters: number->string, list->string, symbol->string, string->number, string->list, string->symbol, object->class.

- the algebric functions: min, max, numerator, denominator, quotient, remainder, round, floor, sin, cos, atan, log, make-rectangular, make-polar, magnitude, angle.

- the language words: include, define, list, make-vector, quote, lambda, if, not, and, or, cond, case, begin, apply, display, read, file-open, fdisplay, newline, file-read, file-close!, let, set!, vector-set!, vector-length, print-env, help, exit.

3°) Additional functions

The mbScheme.bat file allows direct use of both the native functions and the additional functions that are available in the mbScheme.scm file. This adds the following functions.

- the predicats: atom?, belongs?, even? odd?, point?, prime?.

- the numeric operators: %, <=, >=.

- the list operators: add, append, cadr, cddr, enclose, head, last, length, map, nth, rank, remove, replace, reverse, tail, truncate.

- the converters: rational->integer, rational->real.

- the algebric functions: abs, doble, exp, fact, fib, gcd, next-prime, pi, sign, sqrt, square.

- the language words: new-line, open, opaque-vector, vector.

- some specific procedures as examples: check-point, make-point, point-x, point-x-set!, point-y, point-y-set!, translate.