A LaTeX Package

Version 2.0

LISP on TeX

User's Guide

```
\lispinterp {
   (\print
     ((\lambda (\x)
        (\list \x (\list (\quote \quote) \x)))
      (\quote
        (\lambda (\x)
                 (\list \x (\list (\quote \quote) \x))))))
}
```

```
((\lambda (\x) (\list \x (\list (\quote \quote) \x))) (\quote (\lambda (\x) (\list \x (\list (\quote
\quote) \x)))))
```

hogehoge

LISP on TEX

Version 2.0 User's Guide

`https://bitbucket.org/hak7a3/lisp-on-tex`

Hakuta Shizuya <`hak7a3@live.jp`>

July xx, 2015

Contents

1 Introduction

LISP on TeX is a LaTeX class to run LISP programs in a document. All of it is written with TeX macros, so we do not need a special TeX engine, `\write18`, and external language systems. LISP on TeX works if you put its all style files to your `texmf` tree.

1.1 Getting Started

In order to use LISP on TeX, you should load `lisp-on-tex` package: write

```
\usepackage{lisp-on-tex}
```

on your document's preamble. Then, you can execute LISP codes by `\lispinterp`. For example, the code

```
\lispinterp {
  % define \succ function.
  (\define (\succ \n) (\+ \n :1))
  % call \succ and print the result.
  (\texprint (\succ :42))
}
```

outputs "43". As you can see in the example, you can use `%` as starting comment. The `\lispinterp` is not `\long`ed, so you CANNOT include empty lines into a LISP on TeX's program.

1.2 Class Options

LISP on TeX has options for garbage collection (GC). If you want to use GC, use `markGC` option. You can also assign heap size by `GCopt={heapsize=`n`}` where n is an integer. The default heap size is 32768. For example, the code

```
\usepackage[markGC, GCopt={heapsize=40000}]{lisp-on-tex}
```

means LISP on TeX uses GC and the heap size is 40000.

2 Objects

We define LISP on TeX's objects by using a grammatical notation like the TeXbook. In this section, ⟨foo⟩ is a non terminal symbol, `bar` is a terminal symbol, $\longrightarrow$ means "is defined to be," and | means "or". The operator $*$ is Kleene star, and $+$ is Kleene plus.

2.1 Integers

An integer is ⟨integer⟩:

$$\langle\text{integer}\rangle \longrightarrow \texttt{:}\langle\text{TeX's number}\rangle$$

where ⟨TeX's number⟩ is ⟨number⟩ in the TeXbook. For example, `:-42` means -42, `:"BEEF` means 48879, and ``:`\@`` means 64.

2.2 Strings

An string is an TeX's ⟨balanced tokens⟩ surrounded by `'`:

$$\langle\text{string}\rangle \longrightarrow \texttt{'}\langle\text{balanced tokens}\rangle\texttt{'}$$

If you want to include `'`, you should use brace; the code ``'{``}quoted \TeX{} tokens{''}'`` means "quoted TeX tokens". In ordinary Lisp interpretation, `'` is used for abbreviation of `quote`. In contrast, LISP on TeX does not support it.

2.3 CONS cells and nil

A CONS cell is ⟨cons cell⟩ and the value nil is ⟨nil⟩:

$$\begin{array}{rcl}
\langle\text{cons cell}\rangle & \longrightarrow & \langle\text{proper list}\rangle \mid \langle\text{improper list}\rangle \\
\langle\text{proper list}\rangle & \longrightarrow & \texttt{(}\langle\text{object}\rangle+\texttt{)} \\
\langle\text{improper list}\rangle & \longrightarrow & \texttt{(}\langle\text{object}\rangle+\ \texttt{.}\ \langle\text{object}\rangle+\texttt{)} \\
\langle\text{nil}\rangle & \longrightarrow & \texttt{()}
\end{array}$$

where ⟨object⟩ is a LISP on TEX's object.

2.4 Symbols

In LISP on TEX, a symbol is a control sequence. For example, `\somecs` is a symbol.

2.5 Booleans

A boolean is ⟨bool⟩;

$$\langle\text{bool}\rangle \longrightarrow \texttt{/t} \mid \texttt{/f}$$

The term `/t` means true, and `/f` means false.

2.6 -Reserved Forms