Syntax Explanation and Definition of SEN

Zirak

March 5, 2013

Contents

1 The Complete Syntax

⟨*program*⟩ → ⟨*value*⟩
| ϵ

⟨*value*⟩ → ⟨*atom*⟩
| ⟨*symbol*⟩
| ⟨*list*⟩
| ⟨*plist*⟩

⟨*atom*⟩ → '`nil`'
| '`t`'
| ⟨*anything*⟩

⟨*symbol*⟩ → '`:`' ⟨*atom*⟩

⟨*list*⟩ → '`(`' ⟨*list-values*⟩ '`)`'

⟨*list-values*⟩ → ⟨*value*⟩ ⟨*list-values*⟩
| ϵ

⟨*plist*⟩ → '`(`' ⟨*pairs*⟩ '`)`'

⟨*pairs*⟩ → ⟨*symbol*⟩ ⟨*value*⟩ ⟨*pairs*⟩
| ϵ

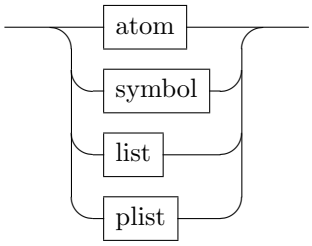
⟨*comment*⟩ → '`;`' ⟨*comment-chars*⟩

⟨*comment-chars*⟩ → ⟨*not-line-terminator*⟩ ⟨*comment-chars*⟩
| ϵ

2 Individual Components

2.1 Values

value

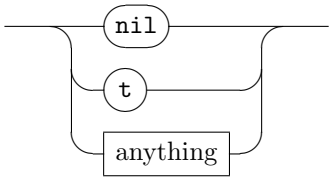


⟨*value*⟩ → ⟨*atom*⟩
| ⟨*symbol*⟩
| ⟨*list*⟩
| ⟨*plist*⟩

A *value* is any of the possible SEN structures.

2.2 Atoms

atom

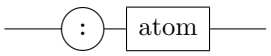


⟨*atom*⟩ → '`nil`'
| '`t`'
| ⟨*anything*⟩

An *atom* is any of the special constructs *nil* or *t*, or any combination of characters, excluding the space character and parentheses (). In addition, an *atom* may not begin with the colon, :.

2.3 Symbols

symbol

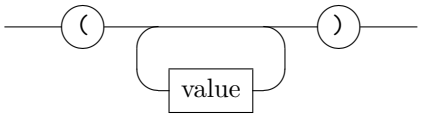


⟨*symbol*⟩ → '`:`' ⟨*atom*⟩

A *symbol* is a literal value. While an *atom* may be subject to interpretations (*t* may turn to *true* in a target language), a *symbol* will always appear as-is.

2.4 Lists

list



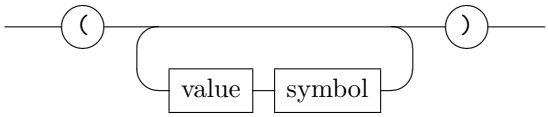
⟨*list*⟩ → '`(`' ⟨*list-values*⟩ '`)`'

⟨*list-values*⟩ → ⟨*value*⟩ ⟨*list-values*⟩
| ϵ

A *list* is one or more *value*s, separated by spaces. They do not have to be homogeneous; that is, you can mix up the value types. You may arbitrarily nest lists to easily create complex structures.

2.5 Property-Lists

plist



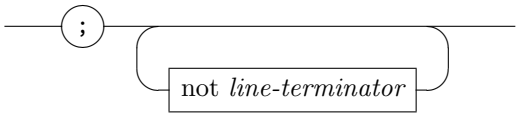
⟨*plist*⟩ → '(' ⟨*pairs*⟩ ')'

⟨*pairs*⟩ → ⟨*symbol*⟩ ⟨*value*⟩ ⟨*pairs*⟩
| ϵ

$p-lists$, or $property-lists$, can be considered a poor man's hash-table. They are made of one or more $key => value$ pairs, where the key must be a *symbol*, and the value may be any *value* allowed in the language. The *key* and *value* are separated by a space, and so are each pair.

2.6 Comments

comment



⟨*comment*⟩ → ';' ⟨*comment-chars*⟩

⟨*comment-chars*⟩ → ⟨*not-line-terminator*⟩ ⟨*comment-chars*⟩
| ϵ

A *comment* may be inserted at any point in the program. Its contents are ignored by the parser. The comment spans from the beginning of the semi-colon ; until a line-terminator is met (EOL or EOF).