Syntax Explanation and Definition of SEN

Zirak

March 5, 2013

Contents

1 The Complete Syntax

$\langle\textit{program}\rangle \to \langle\textit{value}\rangle$
$\quad\mid\ \epsilon$

$\langle\textit{value}\rangle \to \langle\textit{atom}\rangle$
$\quad\mid\ \langle\textit{symbol}\rangle$
$\quad\mid\ \langle\textit{list}\rangle$
$\quad\mid\ \langle\textit{plist}\rangle$

$\langle\textit{atom}\rangle \to$ ‘`nil`’
$\quad\mid$ ‘`t`’
$\quad\mid\ \langle\textit{anything}\rangle$

$\langle\textit{symbol}\rangle \to$ ‘`:`’ $\langle\textit{atom}\rangle$

$\langle\textit{list}\rangle \to$ ‘`(`’ $\langle\textit{list-values}\rangle$ ‘`)`’

$\langle\textit{list-values}\rangle \to \langle\textit{value}\rangle\ \langle\textit{list-values}\rangle$
$\quad\mid\ \epsilon$

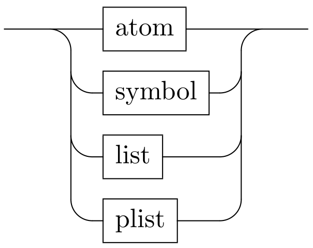
$\langle\textit{plist}\rangle \to$ ‘`(`’ $\langle\textit{pairs}\rangle$ ‘`)`’

$\langle\textit{pairs}\rangle \to \langle\textit{symbol}\rangle\ \langle\textit{value}\rangle\ \langle\textit{pairs}\rangle$
$\quad\mid\ \epsilon$

2 Individual Components

2.1 Value

value

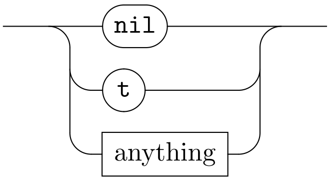


⟨*value*⟩ → ⟨*atom*⟩
  | ⟨*symbol*⟩
  | ⟨*list*⟩
  | ⟨*plist*⟩

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

2.2 Atom

atom

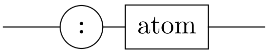


⟨*atom*⟩ → ‘`nil`’
  | ‘`t`’
  | ⟨*anything*⟩

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 Symbol

symbol

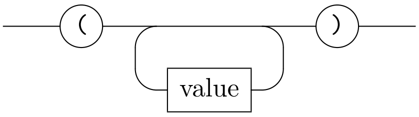


⟨*symbol*⟩ → ‘`:`’ ⟨*atom*⟩

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 List

list



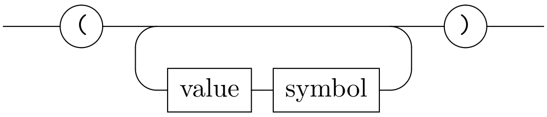
⟨*list*⟩ → ‘`(`’ ⟨*list-values*⟩ ‘`)`’

⟨*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-List

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.