YuchiKaml

Yuchiki

2018 Dec.

Contents

1 Introduction

YuchiKaml is a toy language. and YuchiKaml interpreter is an implementation of interpreter of YuchiKaml. Both are created in order to get accustomed to Sprache, a C#Parser Combinator Library. In this article, I introduce both the language and the interpreter.

2 YuchiKaml Language

YuchiKaml is a dynamic typed language with-ML like surface grammar.

2.1 Syntax

Expressions of YuchiKaml are defined by the following BNF equations:

$$
\begin{aligned}
e ::= & () \mid x \mid n \mid \text{true} \mid \text{false} \mid (e) \\
& \mid e\ e \mid !e \\
& \mid e * e \mid e / e \\
& \mid e + e \mid e - e \\
& \mid e \leq e \mid e < e \mid e \geq e \mid e > e \\
& \mid e = e \mid e \neq e \\
& \mid e \,\&\&\, e \\
& \mid e \,||\, e \\
& \mid e \triangleright e \mid e \gg e \\
& \mid \text{if } e \text{ then } e \text{ else } e \mid \text{let(rec)} x\ \tilde{a} = e \text{ in } e \mid \text{let rec } x\ a_1\ \tilde{a} = e \text{ in } e \mid \lambda x \to e
\end{aligned}
$$

The operators defined in earlier rows have stronger precedences than the operators defined in later rows. For example, $1+2*3$ is not parsed as $(1+2)*3$, but $1+(2*3)$.

In real source codes, the symbols above are notated as follows:

$\leq$	<=
$\geq$	>=
$\neq$	!=
&&	&&
$\|\|$	||
$\triangleright$	|>
$\gg$	>>
λ	\
$\rightarrow$	->

2.2 Semantics

Then we define the semantics of the expressions.

2.2.1 Value

Values of YuchiKaml is listed as below:

$$
\begin{aligned}
v(\text{value}) ::= &\ \text{VInt}\, n \mid \text{VBool}\, b \mid \text{VString}\, s \mid \text{VClos}(x, e, \Gamma) \mid \text{VBClos}\, f_b \\
\Gamma(\text{environment}) &\in \text{Var} \not\to \text{Val} \\
f_b(\text{built-in function}) &\in \text{Val} \not\to \text{Val}
\end{aligned}
$$

Here Var is the set of the variables and Val is the set of the values.

2.2.2

3 YuchiKaml Interpreter

3.1 Usage

3.2 Preprocess