Tarea 07 - El Lenguaje LETREC

Enrique Giottonini

Noviembre 09, 2022

Especificación del Lenguaje.

Sintáxis Concreta

Program	:=	**Expression**
Expression	:=	**Number**
	|	-(**Expression** **Expression**)
	|	zero? (**Expression**)
	|	if **Expression** then **Expression** else **Expression**
	|	**Identifier**
	|	let **Identifier** = **Expression** in **Expression**
	|	proc (**Identifier**) **Expression**
	|	(**Expression** **Expression**)
	|	letrec **Identifier**(**Identifier**) = **Expression** in **Expression**
Digit	:=	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
Number	:=	**Digit**
	|	**DigitNumber**
	|	**Digit** .**Digit**{**Digit**}*
Alphabetic	=	[a-zA-Z]
Identifier	:=	**Alphabetic****Identifier**
	|	**Identifier**{**Digit**}*

Sintáxis Abstracta (Notación de Racket)

Program:
- (a-program exp_1)

Expression:
- (const-exp num)
- (diff-exp exp_1 exp_2)
- (zero?-exp exp_1)
- (if-exp exp_1 exp_2 exp_3)
- (var-exp var)
- (let-exp var exp_1 body)
- (proc-exp var body)
- (call-exp rator rand)
- (letrec-exp p-name b-var p-body letrec-body)

Number: `Real`
Identifier: Versión limítada de `Symbol`

Semántica

```
value-of (const-exp n) ρ) = (num-val n)

(value-of (var-exp var) ρ) = ρ(var)

(value-of (diff-exp exp1 exp2) ρ) =
(num-val (- (expval→num (value-of exp1 ρ))
            (expval→num (value-of exp2 ρ))))

(value-of (zero?-exp exp1) ρ) =
(let ([val1 (value-of exp1 ρ)])
  (bool-val (= 0 (expval→num val1))))

(value-of (if-exp exp1 exp2 exp3) ρ) =
(if (expval→bool (value-of exp1 ρ))
  (value-of exp2 ρ)
  (value-of exp3 ρ))

(value-of (let-exp var exp1 body) ρ) =
(let ([val1 (value-of exp1 ρ)])
  (value-of body [var = val1 ]ρ))

(value-of (proc-exp var body) ρ) =
(proc-val (procedure var body ρ))
```

```
(value-of (call-exp op-exp arg-exp) ρ) =
(let ([proc (expval→proc (value-of op-exp ρ))]
      [arg (value-of arg-exp ρ)])
  (apply-procedure proc arg))
```

donde:

```
(apply-procedure (procedure var body ρ) val) =
(value-of body [var = val]ρ)
```

```
(value-of (letrec-exp p-name b-var p-body letrec-body) ρ) =
(value-of letrec-body [p-name = b-var ↦ p-body]ρ)
```

donde:

Si $\rho_1 = [\text{p-name} = \text{b-}var \mapsto \text{p-}body]\rho$, entonces:

si var = p-name
(apply-env ρ_1 var) =
(proc-val (procedure b-var p-$body$ ρ_1))
si $var \neq$ p-name .
(apply-env $\rho 1$ var) = (apply-env ρ var)