Tarea 07 - El Lenguaje LETREC

Enrique Giottonini, Miguel Navarro

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 exp1)

Expression:
- (const-exp num)
- (diff-exp exp1 exp2)
- (zero?-exp exp1)
- (if-exp exp1 exp2 exp3)
- (var-exp var)
- (let-exp var exp1 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) = (apply-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)
 = (if (equal? 0 (expval->num (value-of exp1 ρ)))
       (bool-val #t)
       (bool-val #f))

(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)
 = (value-of body (extend-env var (value-of exp1 ρ) ρ))

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

(value-of (call-exp rator rand) ρ)
 = (let ([proc (expval->proc (value-of rator ρ))]
```

```
        [arg (value-of rand ρ)])
    (apply-procedure proc arg))

(value-of (letrec-exp proc-name bound-var proc-body letrec-body) ρ)
 = (value-of letrec-body (extend-env-rec proc-name bound-var proc-body ρ))
```