Ejercicios PROC

Daniel Antonio Quihuis Herandez

12 de octubre del 2022

1 Ejercicio 3.20[*]

In PROC, procedures have only one argument, but one can get the effect of multiple argument procedures by using procedures that return other procedures. For example, one might write code like

```
let f = proc (x)
          proc (y)...
            in ((f 3) 4)
```

This trick is called Currying, and the procedure is said to be Curried. Write a Curried procedure that takes two arguments and returns their sum. You can write x + y in our language by writing (x, (0, y)).

```
let function-name =
    proc (x)
      proc(y)
        -(x, -(0, y))
```

2 Ejercicio 3.27[*]

Add a new kind of procedure called a traceproc to the language. A traceproc works exactly like a proc, except that it prints a trace message on entry and on exit.

;; Sintaxis concreta

Expression ::= trace-procedure(ID) Expression

;; Sintaxis abstracta

(trace-proc-exp var body)

;; Semantica

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