

Lenguajes de Programación

Práctica 3 (Versión preliminar)

Karla Ramírez Pulido

Manuel Soto Romero

Alejandro Hernández Mora

Alma Rocío Sánchez Salgado

Silvia Díaz Gómez

Semestre 2021-1

Facultad de Ciencias, UNAM

Fecha de inicio: 28 de octubre de 2020

Fecha de entrega: 11 de noviembre de 2020

1. Objetivos

Implementar el intérprete para el lenguaje **WAE**. Para llevar a cabo esta tarea, se debe completar el cuerpo de las funciones faltantes dentro de los archivos **grammars.rkt**, **parser.rkt** e **interp.rkt**.

La gramática del lenguaje WAE se presenta a continuación:

```
<expr> ::= <id>
| <num>
| {<op> <expr>+}
| {with {{<id> <expr>+} <expr>}
| {with* {{<id> <expr>+} <expr>}
<id> ::= a | b | c | ...
<num> ::= 1 | 2 | 3 | ...
<op> ::= + | - | * | / | modulo | expt | add1 | sub1
```

La gramática anterior se define mediante los siguientes tipos en *Racket*.

```
;; Definición del tipo Binding
(define-type Binding
  [binding (id symbol?) (value WAE?)])

;; Definición del tipo WAE
(define-type WAE
  [id (i symbol?)]
  [num (n number?)]
  [op (f procedure?) (args (listof WAE?))]
  [with (bindings (listof binding?)) (body WAE?)]
  [with* (bindings (listof binding?)) (body WAE?)])
```

2. Ejercicios

1. (3.5 pts.) Completar el cuerpo de la función (parse sexp) dentro del archivo parser.rkt el cual recibe una expresión simbólica¹, realiza el análisis sintáctico correspondiente, esto es, construye un Árbol de Sintaxis Abstracta² (ASA). Para el análisis sintáctico de las operaciones aritméticas se debe hacer un mapeo entre los símbolos de función en sintaxis concreta y las funciones de Racket.

```
;; parse: s-expression -> WAE
(define (parse sexp) ...)
```

2. (3 pts.) Completar el cuerpo de la función (subst expr sub-id value) dentro del archivo interp.rkt la cual realiza la sustitución expr[sub-id := value] correspondiente, es decir, reemplaza cada presencia de la variable sub-id en la expresión expr por otra expresión value. Recuerda cuidar el alcance de las variables en las expresiones with y with*.

```
;; subst: WAE symbol WAE -> WAE
(define (subst expr sub-id value) ...)
```

3. (3.5 pts.) Completar el cuerpo de la función (interp expr) dentro del archivo interp.rkt el cual recibe una expresión y regresa su evaluación correspondiente. Para la evaluación de expresiones with y with* es necesario usar la función subst definida en el inciso 2.

```
;; interp: WAE -> number
(define (interp expr) ...)
```

Toma en consideración:

- **Identificadores**

Se debe lanzar un error indicando que se trata de una variable libre.

```
(interp (parse 'foo)) => error: Variable libre
```

- **Números**

Al ser un valor atómico, los números se evalúan a sí mismos.

```
(interp (parse 1729)) => 1729
```

- **Operaciones aritméticas**

Se debe aplicar el operador correspondiente a la lista de operandos indicada.

```
;; Operaciones unarias
(interp (parse '{add1 18}')) => 19
(interp (parse '{sub1 35}')) => 34
```

¹Del inglés, *s-expression*. Puede ser un número, un símbolo o una lista de expresiones simbólicas.

²Del inglés, *Abstract Syntax Tree (AST)*.

```
;; Operaciones binarias
(interp (parse '{modulo 10 2})) => 0
(interp (parse '{expt 2 3})) => 8

;; Operaciones n-arias
(interp (parse '{+ 1 2 3})) => 6
(interp (parse '{- 3 2 1})) => 0
(interp (parse '{* 1 2 3})) => 6
(interp (parse '{/ 8 2 2})) => 2
```

■ Asignaciones locales simples (with)

Dada la lista de parejas de identificadores con valores de la forma (binding id value), se deben de sustituir en el cuerpo (body) correspondiente cada uno de los identificadores (id) por su valor (value).

```
(interp (parse '{with {{a 2} {b 3}} {+ a b}})) => 5
(interp (parse '{with {{a 2} {b {+ a a}}} b})) =>
error: Variable libre
```

■ Asignaciones locales anidadas (with*)

Dada la lista de parejas de identificadores con valores de la forma (binding id value), se deben de sustituir en el cuerpo (body) correspondiente cada uno de los identificadores (id) por su valor (value). En esta versión de with, se permite hacer referencia a identificadores definidos previamente (anidados).

```
(interp (parse '{with* {{a 2} {b {+ a a}}} b})) => 4
```

Se deberán respetar todas las definiciones de las funciones, es decir está prohibido cambiar el tipo que tengan, ya sea el de algún parámetro o el del resultado.

3. Requerimientos

Deberás entregar tu práctica en equipos de mínimo dos personas y máximo tres. La entrega es a través de la plataforma **Google classroom** antes de las 23:59 del día de entrega indicado, tal y como lo establecen los lineamientos de entrega. Deberás adjuntar tu solución en los archivos **parser.rkt** e **interp.rkt**. Revisa bien mayúsculas, minúsculas y espacios para el nombre del archivo que se pide.

El orden en el que aparezcan las funciones en el archivo solicitado, debe ser el orden especificado en este PDF, de lo contrario podrán penalizarse algunos puntos. Puedes utilizar funciones auxiliares, se recomienda definirla después de la función que la va a utilizar.

No dudes en aclarar tus dudas en la sesión de laboratorio y vía correo electrónico.

¡Que tengas éxito en la práctica!.