Lenguajes de Programación
Práctica 5

Semestre 2026-1

Facultad de Ciencias, UNAM

Karla Ramírez Pulido karla@ciencias.unam.mx
Alan A. Martínez López alanmartinez-012@ciencias.unam.mx
Karyme I. Azpeitia García kary_agarcia@ciencias.unam.mx

Fecha de inicio: 15 Octubre 2023
Fecha de entrega: 22 Octubre 2023

1. Objetivos

Analizar las implementaciones dadas en los archivos `grammars.rkt`, `parser.rkt`, `desugar.rkt` e `interp.rkt` para el lenguaje **CFWBAE** y realizar los siguientes ejercicios.

La gramática del lenguaje CFWBAE se presenta a continuación:

```
<expr> ::= <id>
    | <num>
    | <bool>
    | {<op> <expr>+}
    | {if <expr> <expr> <expr>}
    | {cond {<expr> <expr>+} {else <expr>}}
    | {with {{<id> <expr>}+} <expr>}
    | {with* {{<id> <expr>}+} <expr>}
    | {fun {<id>*} <expr>}
    | {<expr> <expr>*}
<id> ::= a | b | c | ...
<num> ::= 1 | 2 | 3 | ...
<bool> ::= true | false
<op> ::= + | - | * | / | modulo | expt | add1 | sub1
| < | <= | = | > | >= | not | and | or | zero?
```

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

```
;; Definici n del tipo Binding
(define-type Binding
    [binding (id symbol?) (value SCFWBAE?)])

;;Definici n del tipo condition para la definici n de cond.
(define-type Condition
    [condition (test-expr SCFWBAE?) (then-expr SCFWBAE?)]
    [else-cond (else-expr SCFWBAE?)])

;; Definici n del tipo SCFWBAE
(define-type SCFWBAE
    [idS (i symbol?)]
    [numS (n number?)]
    [boolS (b boolean?)]
    [iFS (condicion SCFWBAE?) (then SCFWBAE?) (else SCFWBAE?)]
    [opS (f procedure?) (args (listof SCFWBAE?))]
    [condS (cases (listof Condition?))]
    [withS (bindings (listof binding?)) (body SCFWBAE?)]
    [withS* (bindings (listof binding?)) (body SCFWBAE?)]
    [funS (params (listof symbol?)) (body SCFWBAE?)]
    [appS (fun SCFWBAE?) (args (listof SCFWBAE?))])
```

2. Ejercicios

2.1. Parser

1. (0.5 pts.) ¿Por qué es importante representar el programa como un árbol de sintaxis abstracta?
2. (1 pts.) Explica cómo el parser reconoce una expresión `if` y la diferencia respecto al antiguo `if0`.
3. (0.5 pts.) De acuerdo a la implementación en el archivo `parser.rkt` ¿Qué ocurre si el parser recibe una expresión con paréntesis mal colocados o una forma desconocida?

2.2. Desugar

4. (1 pts.) ¿Qué significa "desazúcar" una expresión? Da un ejemplo.
5. (1 pts.) ¿Cómo traduce cond una cascada de condiciones en una serie de if anidados? ¿Qué ocurre si no hay un else final?
6. (1 pts.) ¿Qué pasaría si olvidas aplicar desugar recursivamente dentro de las subexpresiones (por ejemplo, en el cuerpo de un with)?
7. (1 pts.) Si agregas un nuevo azúcar como `when`, ¿qué regla de traducción definirías en `desugar.rkt`?
8. (1 pts.) ¿Por qué desugar siempre debe producir una expresión válida del lenguaje base `(CFWBAE)`?

2.3. Interp

9. (1 pts.) ¿Qué papel cumple el ambiente `(DefrdSub)` en la evaluación de expresiones con variables?
10. (1 pts.) ¿Qué diferencias encuentras entre evaluar with directamente y evaluar la versión desazucarada `(como app(fun ...))`?

11. (1 pts.) De acuerdo a la implementación dada en el archivo `interp.rkt` ¿Qué estrategia de evaluación usa el intérprete: estricta o perezosa? Justifica tu respuesta.

2.4. Extra (Opcionales)

12. (1 pts.) Propón una extensión del lenguaje donde `with` acepte funciones anónimas múltiples (como parámetros simultáneos). Escribe la traducción en forma de desazúcar.

13. (0.5 pts.) Explica ¿Qué cambios serían necesarios en el `interp` si se decidiera eliminar el módulo `desugar` y manejar los azúcares directamente?

3. Requerimientos

Deberás entregar tu práctica en equipos de mínimo una persona 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.

La idea es que reutilices funciones de los ejercicios que ya se programaron anteriormente, pero ten cuidado con los cambios en la sintáxis abstracta.

¡Que tengas éxito en tu práctica!.