

Universidad Nacional Autónoma de México
Facultad de Ciencias
Lenguajes de Programación

Actividad de laboratorio 6

Karla Ramírez Pulido
karla@ciencias.unam.mx

J. Ricardo Rodríguez Abreu
ricardo_rodab@ciencias.unam.mx

Manuel Soto Romero
manu@ciencias.unam.mx

Luis A. Patlani Aguilar
ayudantefc@gmail.com

13 de septiembre de 2017

La actividad se resuelve en clase

Hora límite de entrega: 16:59:59

Semestre 2018-1

Objetivos

Entender cómo funciona un analizador semántico básico mediante la evaluación de expresiones del lenguaje WAE apoyado del algoritmo de sustitución textual.

Descripción general

La actividad consiste en definir algunos programas en WAE para poder analizar la implementación de su analizador semántico.

Archivos requeridos

El material de esta actividad de laboratorio consta de los siguientes archivos¹:

- `grammars.rkt` archivo con la definición del tipo de dato abstracto que define los árboles de sintaxis abstracta para el lenguaje WAE.
- `parser.rkt` archivo dónde se encuentra el analizador sintáctico para el lenguaje WAE.
- `interp.rkt` archivo dónde se encuentra el analizador semántico para el lenguaje WAE.
- `test-actividad6.rkt` archivo dónde se agregarán las pruebas unitarias de la actividad.

¹Los archivos pueden descargarse desde la página del curso <http://lenguajesfc.com>.

Actividad

Resolver los siguientes ejercicios de forma limpia y ordenada.

Ejercicio 6.1 Dadas las siguientes expresiones en sintaxis concreta, escribir pruebas unitarias dentro del archivo `test-actividad6.rkt` que ilustren el resultado después de evaluar las mismas. Por ejemplo, si se tiene la expresión `{+ 1 2}` se tendría que escribir la prueba `(test (interp (parse '{+ 1 2})) 3)`:

1. Identificador²:

```
foo
```

2. Número:

```
1729
```

3. Operación binaria:

```
{+ 17 {* 35 {/ 10 {- 8 2}}}}
```

4. Asignación local:

```
{with {a 2}
  {with {b 3}
    {with {c 4}
      {+ a {+ b c}}}}}
```

Ejercicio 6.2 El algoritmo de sustitución textual, visto en clase, incluye las siguientes líneas:

```
[(with bound-id named-expr body)
 (if (symbol=? bound-id sub-id)
     (with
      bound-id
      (subst named-expr sub-id val)
      body)
     (with
      bound-id
      (subst named-expr sub-id val)
      (subst bound-id sub-id val))))]
```

¿Para qué se usan estos dos casos? Escribir un programa en WAE que entre el primer caso y otro que entre en el segundo, explicar para ambos programas por qué entró en un caso y no en otro.

Los programas deben definirse y ejecutarse en `interp.rkt`. Por ejemplo, el programa `{+ 1 2}` se escribiría como: `(interp (parse '{+ 1 2}))`.

²Para escribir pruebas que se evalúen a error se puede usar la función `test/exn`.

Ejercicio 6.3 Si en la función `interp` cambiamos las siguientes líneas:

```
[(with id val body)
 (interp (subst body id val))]
```

por

```
[(with id val body)
 (interp (subst body id (num (interp val))))]
```

¿Qué cambia? ¿Cuál es la diferencia entre estos dos códigos? Escribir un programa en WAE que ejemplifique esta diferencia y explicarlo.

El programa debe definirse y ejecutarse en `interp.rkt`.

Entrega

El programa se debe entregar debidamente comentado, siguiendo las reglas de estilo mencionadas en clase y con pruebas unitarias para cada función (en caso de que se pueda).

- Si el ejercicio se termina en clase, levantar la mano para que alguno de los ayudantes de laboratorio califique presencialmente.
- Si por cuestiones de tiempo el ejercicio no se puede calificar presencialmente, enviar el código en un archivo `actividad6.tar.gz` a los correos `manu+ldp@ciencias.unam.mx` y `ayudantefc@gmail.com` con el asunto `[LDP-Actividad 6]`. Incluir el nombre de los integrantes en el cuerpo del correo y la explicación de los ejercicios 6.2 y 6.3.

Esta actividad sólo será tomada en cuenta a estudiantes que aparezcan en la lista de asistencia de la sesión de laboratorio y equivale a medio punto extra sobre la calificación final de la Práctica 3.