Lenguajes de Programación, 2023-1
Práctica 02: Expresiones aritméticas y booleanas

Manuel Soto Romero Tania Michelle Rubí Rojas Silvia Díaz Gómez

Fecha de entrega: 16 de septiembre de 2022

Objetivos

- Implementar un intérprete para un lenguaje de programación a partir de su especificación formal.
- Aprender a usar la herramienta HAPPY que será manejada durante todas las prácticas.
- Comprender las fases de generación de código ejecutable.

Especificación del lenguaje

En esta práctica agregaremos a MINILISP expresiones aritméticas y booleanas.

Sintaxis concreta

```
<expr> ::= <num>
         | <bool>
         | (<op> <expr>+)

<num> ::= ... | -1 | 0 | 1 | ...

<bool> ::= #t | #f

<op> ::= + | - | * | / | add1 | sub1 |
       | < | > | = | not | or | and
```

Sintaxis abstracta

$$\frac{\texttt{n} \in \mathbb{Z}}{\texttt{num(n)}\,ASA} \qquad \frac{\texttt{n} \in \mathbb{B}}{\texttt{boolean(b)}\,ASA} \qquad \frac{\texttt{f:String} \quad \texttt{args:}[ASA]}{\texttt{op(f,args)}\,ASA}$$

Semántica

Usaremos una semántica operacional de paso grande mediante la sintaxis abstracta del lenguaje.

Nota: En esta ocasión usaremos los símbolos … para los constructores multiparamétricos, sin embargo es importante mencionar que esta forma de especificar no es adecuada. La Práctica 3 cambiará esta notación.

Números

Los números se interpretan a la representación de números correspondiente al lenguaje anfitrión.

$$\frac{}{\texttt{num}(n) \Rightarrow \hat{n}}$$

Booleanos

Los booleanos se interpretan a la representación de booleanos correspondiente al lenguaje anfitrión.

$$\frac{}{\texttt{boolean}(b) \Rightarrow \hat{b}}$$

Operaciones multiparamétricas

Las operaciones multiparamétricas se interpretan mediante la operación correspondiente. El comportamiento de las operaciones es el habitual.

$$\frac{\texttt{elige}(f) = g \qquad args \Rightarrow_g v}{\texttt{op}(f, args) \Rightarrow v}$$

Donde:

- `elige` es una función que transforma el símbolo de operación en sintaxis abstracta en la operación correspondiente en el lenguaje anfitrión para poder resolver la operación.
- $\Rightarrow_g$ es la reducción de una lista de valores aplicando la función g de dos en dos en cada uno de los elementos.

Por ejemplo para la suma:

$$\frac{\texttt{elige}('+') = + \qquad [num(1), num(2), num(3)] \Rightarrow_+ \hat{6}}{\texttt{op}('+', [num(1), num(2), num(3)]) \Rightarrow \hat{6}}$$

Ejercicios

Adjunto a este archivo PDF se encuentran una serie de archivos con secciones que debes completar para dar vida a tu intérprete. Descárgalos y realiza lo siguiente:

Ejercicio 1: Análisis léxico y sintáctico (40 pts.)

Con el apoyo de HAPPY define el proceso de análisis léxico con el nombre `lexer` y análisis sintáctico con el nombre `parser` para el lenguaje. El archivo `Grammars.y` contiene un esqueleto con estas definiciones tu misión es completar:

1. La sección de gramáticas ubicada después de los símbolos `%%`, ahí encontrarás el siguiente comentario:
   `/* DEFINE AQUÍ TUS GRAMÁTICAS PARA EL PARSER. */`
2. La función `lexer`.

Pon especial atención a los tipos `Token` y `ASA` definidos en el archivo.

Ejercicio 2: Análisis semántico (40 pts.)

Completa la función `interp` contenida en el archivo `Interp.hs`. La función debe implementarse de acuerdo con la definición de las reglas de la semántica operacional dada en este documento.

La salida debe ser algo de tipo `Value` con el fin de homogeneizar resultados.

Ejercicio 3: Pruebas (20 pts.)

Una vez terminados los ejercicios anteriores, ejecuta el archivo `Practica02.hs` y prueba tus implementaciones. **Este archivo no se debe modificar.**

Algunas pruebas que puedes realizar son:

Expresión	**Salida esperada**
`1729`	`1729`
`#t`	`#t`
`#f`	`#f`
`(add1 (sub1 (* (+ 2 3) (- 5 2 1))))`	`10`
`(not (or (and (< 5 3) (> 10 8 2)) (= 1 2 3)))`	`#t`