Lenguajes de programación 2016-2
Práctica 1

Noé Salomón Hernández Sánchez
Albert M. Orozco Camacho
C. Moisés Vázquez Reyes

Facultad de Ciencias UNAM

En esta práctica se van a implementar la semántica estática y dinámica para el lenguaje EAB:

$$e \;=\; n \mid x \mid e + e \mid e * e \mid \;\mathsf{let}\; x = e \;\mathsf{in}\; e \;\mathsf{end} \mid \mathsf{if}\; e \;\mathsf{then}\; e \;\mathsf{else}\; e \mid \mathsf{suc}\; e \mid \mathsf{pred}\; e \mid \mathsf{iszero}\; e \mid \mathsf{true} \mid \mathsf{false}$$

la respectiva definición en Haskell es la siguiente:

```
data Asa = VNum Int
         | VBol Bool
         | Var Ident
         | Suma Asa Asa
         | Prod Asa Asa
         | Let Asa Asa Asa
         | Ifte Asa Asa Asa
         | Suc Asa
         | Pred Asa
         | Iszero Asa deriving (Show,Eq)
```

Sólo trabajaremos con *asas*, el lexer y parser de este lenguaje ya se les brinda en los archivos *LexerEAB.hs* y *ParserEAB.hs*.

1. Semántica estática

El ejercicio se encuentran dentro del archivo *EABestatica.hs*

- `vt::Ctx->Asa->Tipo`

  Realiza la verificación de tipos de una expresión.

  Los contextos de variables están representados en Haskell como sigue:

```
type Ctx = [(Ident,Tipo)]
```

donde el tipo *Ident* es un alias del tipo *String*. Los únicos tipos que nuestras expresiones pueden tener son *TBol* y *TNat*.

Ejemplos:

- `vt [(‘‘y’’,TNat)] $ Let (Var ‘‘x’’) (Var ‘‘y’’) $ Var ‘‘x’’`
  TNat
- `vt [] $ Prod (VNum 3) (Suma (Var ‘‘x’’) (VNum 5))`
  *** Exception: La variable “x” no esta declarada en el contexto

2. Semántica dinámica

Los ejercicios se encuentran dentro del archivo *EABdinamica.hs*

- `esvalor::Asa->Bool`

  Determina si una expresión es un valor.

  Ejemplos:

  - `esvalor $ VNum 7`
    True
  - `esvalor $ Suma (VNum 1) (VNum 8)`
    False

- `eval1p::Asa->Asa`

  Realiza un paso de la evaluación aplicando una regla de transición.

  Ejemplos:

  - `eval1p $ Iszero (Suma (VNum 3) (VNum 4))`
    Iszero (VNum 7)
  - `eval1p $ Prod (Suc $ VNum 4) (VNum 0)`
    Prod (VNum 5) (VNum 0)

- `evalaux::Asa->Asa`

  Evalúa una expresión mientras no sea un valor. Esta función sólamente es auxiliar de la siguiente función.

- `eval::Asa->Asa`

  Evalúa una expresión. Antes de evaluar, debe realizarse la verificación de tipos.

3. Pruebas

Para evaluar su práctica se utilizará la función *correPrueba* que se encuentra dentro de *EABdinamica.hs*.

Por ejemplo, para correr la prueba 1, hay que teclear dentro de ghci:

```
correPrueba ‘‘prueba1’’
```

y, esto generará la salida:

```
<<< CONTENIDO DEL ARCHIVO >>>
let x = 5 in 5*x end


<<< TOKENS GENERADOS POR EL LEXER >>>
[Rsv Tlet,Idn "x",Op OIgual,LNum 5,Rsv Tin,LNum 5,Op OProd,Idn "x",Rsv Tend]

<<< ASA  GENERADO POR EL PARSER >>>
Let (Var "x") (VNum 5) (Prod (VNum 5) (Var "x"))

<<< TIPO ASA >>>
TNat

<<< ASA EVALUADO >>>
VNum 25

<<< Concreta, RESULTADO >>>
25
```

- Prueba 1 = 25
- Prueba 2 = Exception: La variable “y” no esta declarada en el contexto
- Prueba 3 = 192
- Prueba 4 = True
- Prueba 5 = 16