

Compiladores 2020-1  
Facultad de Ciencias UNAM  
Práctica 2: Análisis sintáctico de un lenguaje  
fuente (Parte 2)

Lourdes del Carmen González Huesca  
Juan Alfonso Garduño Solís      Nora Hilda Hernández Luna

16 de marzo de 2022  
Fechas de entrega: 25 de marzo

## Lenguaje Fuente

Para el desarrollo de las prácticas usaremos como lenguaje fuente `minHS`. Este último, es un lenguaje de juguete basado en `Haskell`. La siguiente gramática libre del contexto define `minHS`.

```
<programa> ::= <expr>

<expr> ::= <const>
         | <var>
         | <expr> <prim> <expr>
         | if(<expr>)then{<expr>}else{<expr>}
         | fun ([<var>:<type>]*:<type>) => <expr>
         | funF (<var> [<var>:<type>]*:<type>) => <
           expr>
         | let ([<var>:<type> = <expr>]*) in <expr>
         | end
         | <expr> app <expr>
         | (<expr>)

<const> ::= <boolean> | <integer>

<boolean> ::= #t | #f

<integer> ::= <digit> | <digit><integer>

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

```
<var> ::= <car> | <car><var> | <car><digit> | <car><
    digit><var>

<car> ::= a | b | c | ... | z

<prim> ::= + | - | * | / | and | or

<type> ::= Bool | Int | Func <type> <type>
```

## Implementación

Una vez que hemos definido el *lexer* para tokenizar nuestro lenguaje fuente. Debemos crear un *parser* que pueda construir los *árboles de sintaxis abstracta* (ASA) a partir de los tokens generados. Para esta tarea Racket ofrece una función `parser` que nos permite establecer los criterios para el análisis, como definir la gramática del lenguaje, precedencia de operadores, etc. En esta práctica deberás completar la estructura que representa los ASA de miniHS y terminar de implementar la función `parser`.

## Ejercicios

1. (5 pts) Termina de implementar la estructura de miniHS mediante la instrucción `define-struct`.
2. (5 pts) Termina de implementar el `parser` para que dado un programa miniHS, devuelva el ASA correspondiente. Ejemplos en el Listing 1, hay más ejemplos en el script anexo a esta práctica.

```

Ejemplo 2:
if(#t)then{2}else{3}
ASA:
(if-exp (bool-exp #t) (num-exp 2) (num-exp 3))

Ejemplo 3:
fun ([x:Int]:Int) => x
ASA:
(fun-exp
  (typeof-exp
    (brack-exp
      (typeof-exp (var-exp 'x) (int-exp))) (int-exp))
    (var-exp 'x))

Ejemplo 7:
let ([x:Int = 1][y:Int = 2]) in x+y end
ASA:
(let-exp
  (brack-exp
    (app-t-exp
      (assign-exp
        (typeof-exp (var-exp 'x) (int-exp)) (num-exp 1))
        (assign-exp
          (typeof-exp (var-exp 'y) (int-exp)) (num-exp 2))))
    (prim-exp #<procedure:+=> (var-exp 'x) (var-exp 'y)))

Ejemplo 8:
((funF (sumita ([x:Int][y:Int]):Int) => x+y) app 2) app 4
ASA:
(app-exp
  (par-exp
    (app-exp
      (par-exp
        (fun-f-exp
          (typeof-exp (var-exp 'sumita) (brack-exp (app-t-exp
            (typeof-exp (var-exp 'x) (int-exp))
            (typeof-exp (var-exp 'y) (int-exp)))))
          (int-exp)
          (prim-exp #<procedure:+=> (var-exp 'x) (var-exp 'y))))
        (num-exp 2)))
    (num-exp 4))

```

Listing 1: Ejemplo del parser.

## Punto extra

1. (0.5 pts) Modifica la cláusula *error* para que **parser** devuelva errores más claros sobre el análisis sintáctico.

## Entrega

- La entrega será en el *classroom* del curso. Basta entregar los archivos comprimidos y el código bien documentado. Indicando los integrantes del equipo.
- Esta práctica debe realizarse en equipos de a lo más 3 personas.
- En los scripts entregados, deberán considerar incluir lo siguiente como parte de la documentación:
  - Los nombres y números de cuenta de los integrantes del equipo.
  - Descripción detallada del desarrollo de la práctica.
  - En caso de haber sido necesario, la especificación formal de los nuevos lenguajes definidos, utilizando gramáticas.
  - Comentarios, ideas o críticas sobre la práctica.
- Se recibirá la práctica hasta las 23:59:59 horas del día fijado como fecha de entrega, cualquier práctica recibida después no será tomada en cuenta.
- **Cualquier práctica total o parcialmente plagiada, será calificada automáticamente con cero y no se aceptarán más prácticas durante el semestre.**