Diseño

Estructura sintáctica

En principio, la estructura sintáctica podría seguir el estándar ISO/IEC 14977 para **EBNF** (*Extended Backus-Naur Form*), con algunas extensiones adicionales. El objetivo es que los archivos `.joy` sean legibles y fáciles de entender para los desarrolladores de lenguajes, facilitando así la descripción de gramáticas.

Las principales características definidas en éste estandar son las siguientes:

- *Definición*

  Con éste se definen nuevas reglas de producción:

  ```
  primera regla = ...
  segunda regla = ...
  ```

- *Cadena terminal*

  Con ella se definen una cadena terminal dentro de una regla de producción:

  ```
  primera regla = "a"
  seguda regla = 'b'
  ```

- *Concatenación*

  Permite la secuenciación de distintas reglas

  ```
  primera regla = "a" , "b"
  ```

  Esta regla en principio sólo podría producir la cadena “ab”

- *Alternación*

  Permite alternativas para una regla de producción.

  ```
  primera regla = "a" | "b" | "c"
  ```

  Esta regla produce o bien el caracter “a”, el caracter “b” o el caracter “c”.

- *Terminación*

  Para indicar el final de la regla de producción

  ```
  primera regla = "a" | "b" | "c" ;
  ```

- *Opción*

  Para indicar que una expresión puede no estar, o aparecer una vez.

  ```
  primera regla = ["a" | "b" | "c"] , "d" ;
  ```

  Esta regla produce las cadenas “d”, “ad”, “bd”, “cd”.

- *Repetición*

  Para indicar que una expresión puede no estar, o estar un número no determinado de veces.

  ```
  primera regla = {"a" | "b" | "c"} , "d" ;
  ```

  Esta regla produce cadenas que comienzan con una cantidad arbitraria de “a”, “b” y “c” (incluyendo cero, en cualquier orden) que terminan en “d”.

- *Agrupación*

  Para indicar que una expresión se debe considerar como una sóla.

  ```
  primera regla = (a | b) , c
  ```

  Esta regla produce las cadenas “ac” y “bc”

- *Excepción*

  Esta regla permite indicar la diferencia entre conjuntos.

  ```
  primera regla = (a | b), c
  segunda regla = primera regla - "ac"
  ```

  En este caso la segunda regla sólo puede producir la cadena “bc”

- *Secuencias especiales*

  Esta regla permitiría incluir conjuntos predefinidos en el lenguaje, por ejemplo: enteros de 32 bits, enteros de 64 bits, números de punto flotante, cadenas de texto minúsculas, cadenas de texto en mayúsculas, cadenas numéricas, símbolos especiales, etc…

  ```
  primera regla = ? int64 ?
                | ? float ?
                | primera regla + primera regla
  ```

  Esta regla permitiría sumas entre números de punto flotante y enteros de 64 bits