ISIS-1106-1 Lenguajes y Máquinas - Sección 2
Quiz Taller 1
Fecha: 13 de agosto de 2015

- Esta prueba es INDIVIDUAL.
- El intercambio de información con otro estudiante está terminantemente prohibido.
- Cualquier irregularidad con respecto a estas reglas podría ser considerada fraude.

IMPORTANTE: Soy consciente de que cualquier tipo de fraude en los exámenes es considerado como una falta grave en la Universidad. Al entregar este examen por Sicua+ doy expreso testimonio de que este trabajo fue desarrollado de acuerdo con las normas establecidas. Del mismo modo, aseguro que no participé en ningún tipo de fraude.

Nombre	Carné
Firma	Fecha

1. Contexto

El archivo .zip adjunto contiene el proyecto java del probador de parsers. Para este ejercicio, se va a modificar el parser (NuevoParser.jj) que se encuentra en el paquete newParser. Este parser reconoce expresiones aritméticas. La gramática que está implementada es la siguiente:

$$S \rightarrow E\ ;$$
$$E \rightarrow T\{(+ \mid -)T\}$$
$$T \rightarrow F\{(* \mid /)F\}$$
$$F \rightarrow [-](\texttt{num} \mid (E))$$

Identifique en NuevoParser.jj las reglas y expresiones regulares que describen cada una de estas producciones.

2. Notación científica

Se debe modificar la definición del token `CONSTANT` para que se puedan crear enteros usando la notación científica. Ejemplos de cadenas de enteros definidos en notación científica son los siguientes:

- `25e10`
- `-233E8`
- `4e10`

Tenga en cuenta que para que los números sean enteros, la parte exponencial debe ser positiva.

3. Números complejos

Se quieren agregar números complejos a la grámatica.
Se debe agregar una definición de token para números complejos (debe llamarlo `COMPLEX`).
Un número complejo es una cadena de la forma:

`PARTE_REAL , PARTE_IMAGINARIA`**`i`**

Para simplificar el taller, la parte real y la parte imaginaria solo aceptan enteros (no son expresiones) con o sin signo. la parte real y la parte imaginaria de los números complejos también pueden escribirse en notación científica.
Por ejemplo:

- `25E5,4i`
- `-233,6E2i`
- `0,-10i`

Además de la defición del token, deben modificar la regla en javaCC que corresponde al noterminal F.

$$F \rightarrow [-](\texttt{COMPLEX} \mid \texttt{num} \mid (E))$$

4. Variables y uso de arreglos

En la sección de análisis sintáctico agregue una regla para que una expresión también pueda ser una variable o un acceso a una posición de un arreglo.
Una variable es simplemente un nombre. Recuerde que los nombres de las variables deben comenzar por una letra.
Un acceso a una celda de un arreglo es un nombre seguido de unos corchetes cuadrados. El índice al que se accede dentro de un arreglo puede ser una expresión pero no puede ser un número complejo.
Los siguientes son ejemplos:

- `myarr[3]`
- `myarr[3E1-50/20]`
- `arr[33+7*otherArr[17]]`
- `myarr[lastArr[xyz+33]*5]`

Para esta también se modifica la regla para F.

$$F \rightarrow [-](V \mid \texttt{COMPLEX} \mid \texttt{num} \mid (E))$$

Usted debe definir la regla para V: variables o acceso a arreglos.
El parser resultante deberá aceptar cadenas así:

- `5E10,5i + walt[cook+simple[44/someArr[3]]]`
- `abc+xzy-0,5i*arnold[BEBACK+33]-33e4,3i`