

El objetivo de este taller es desarrollar un parser que reconozca descripciones de árboles binarios usando JavaCC.

1 Descripción de árboles binarios

Describiremos un árbol binario con la palabra `TREE` seguida de la descripción del árbol entre paréntesis. La descripción es el nombre del árbol seguido de un punto y coma seguido de la lista de nodos.

La lista de nodos se denota con la palabra `NODES` seguida por una lista de nodos entre paréntesis y separados por coma.

Cada nodo consiste en un identificador seguido de dos puntos (`:`) seguido del identificador del hijo izquierdo, luego dos puntos (`:`) y el identificador del hijo derecho. Todo esto puede o no estar seguido de dos puntos (`:`) y una lista de propiedades separadas por coma.

La lista de nodos siempre empieza con la raíz del árbol, y siempre que aparezca un nuevo nodo que no sea la raíz, su identificador debe haber aparecido previamente como hijo izquierdo o derecho de un nodo anterior. Asimismo, siempre que un identificador (distinto a `$NULL`) aparezca como hijo izquierdo o derecho de un nodo, debe existir un nodo más adelante con ese identificador.

Cada propiedad está encerrada entre `<>` y consiste de un nombre seguido de una coma seguido de un número.

Un identificador es un signo de pesos seguido de dos dígitos entre 0 y 9.

Si el árbol no tiene hijo izquierdo (o derecho) entonces el identificador es `$NULL`.

Task 1. Utilice JavaCC para describir un parser (archivo `.jj`) que reconozca las cadenas que describen un árbol binario como se explica anteriormente. Puede utilizar el proyecto `ParserTesterClean` y modificar alguno de los parsers ya existentes (por ejemplo `uniandes.teolen.parserJavaCC.newParser`). Únicamente debe subir el archivo `.jj` en su entrega.

Recuerde que adicional a la sintaxis descrita, el parser debe verificar que todo hijo izquierdo o derecho de un nodo eventualmente aparezca en la lista de nodos, y que todo nodo en la lista excepto la raíz debe aparecer como hijo izquierdo o derecho de otro nodo.

Además del enunciado, se incluye el proyecto `ParserTesterClean` y un archivo `input.txt` con un input válido que el parser debe reconocer.