



Ejercicio 6: No Terminales Extraños

## Ejercicio 6: No Terminales Extraños

### Conceptos a trabajar:

- No terminales muertos
- Producciones muertas o inalcanzables

### Proyecto:

A través de GADUN, en la carpeta raíz de la herramienta busque en la sección de “GRAMÁTICAS\_EJERCICIOS” el proyecto “*gramática NT inalcanzables.json*” y “*gramática NT muertos.json*”. Si no encuentra el directorio o el archivo por favor descargue los de REPOGADUN.

### Introducción:

Dentro de la creación de gramáticas y producciones, se debe garantizar que cada símbolo tenga un uso dentro de la estructura. Sin embargo, es común encontrarnos con casos en los que tendremos ciertos elementos de nuestra gramática que no permiten completar el análisis de esta, punto al cual toda gramática debería llegar para permitir la correcta identificación de un código fuente.

En estos casos aparecen lo que se conoce como no terminales extraños y son, en líneas menores, aquellos no terminales que no desembocan en un terminal. Dentro de estos encontramos dos tipos: **no terminales muertos** y **no terminales inalcanzables**.

- Los **no terminales muertos** son aquellos que al derivarlos, no desembocan o no conducen a un terminal.
- Los **no terminales inalcanzables** son aquellos a los que nunca se puede llegar mediante el no terminal inicial.

Cuando existen no terminales extraños se elimina toda la producción donde estos aparecen.

Para este ejercicio tendremos en cuenta los siguientes terminales y no terminales:

Terminales (nombre y elementos)		No Terminales
Var	x y	<A>
		<B>
		<C>
		<D>
		<E>

*Tabla de terminales y no terminales*

Lo siguiente será acomodarlos de tal manera que se ejemplifique la creación de no terminales extraños y cómo podemos eliminarlos. Usaremos dos archivos en particular que ya poseen los elementos descritos en la tabla, tendremos entonces los terminales indicados previamente:

	Nombre	Expresión
1	var	x,y

De la misma forma, en la sección del analizador sintáctico, ya existen los no terminales listados:

No Terminales
<A>
<B>
<C>
<D>
<E>

Adicionalmente, cada una de las gramáticas listadas en los dos archivos describe un problema de no terminales extraños que solucionaremos en la guía.

## Proceso

1. Abre el proyecto de nombre *gramática NT muertos.json* indicado para esta guía. Se trabajará primeramente con los no terminales muertos
2. Dirígete a la sección del analizador sintáctico, veremos una lista de producciones que describen una gramática que posee tres no terminales muertos, lo podemos comprobar si presionamos en el botón *Validar Producciones*, el software nos informará la existencia de este tipo de no terminales extraños y señalará algunas producciones:

Lado Izquierdo		Lado Derecho	
1	<A>	var:y <B> var:y	-
2	<A>	var:x	-
3	<B>	<C> var:y <D>	-
4	<C>	var:y	-
5	<D>	<E>	-

GADUN se encargará de señalar cada una de las producciones que contengan algún no terminal que presente el problema de ser de tipo muerto.

3. Como se trata de un problema de terminales muertos, identificamos cada uno de nuestros terminales y detallamos cuales si desembocan en un terminal. Así, vemos que las producciones <B>, <D> y <E> no llegan a un fin si las empezamos a derivar.

Es importante aclarar que <B> sigue siendo un no terminal muerto aún si posee en su definición al no terminal <C> y al terminal var:y. Esto debido a la existencia de <D> pues este no terminal no dejará que <B> termine de derivarse como tal.

4. Como ya identificamos los no terminales muertos, lo siguiente será eliminarlos en conjunto con las producciones que puedan ser problemáticas:

Lado Izquierdo		Lado Derecho	
1	<A>	var.y <B> var.y	-
2	<A>	var.x	-
3	<B>	<C> var.y	-
4	<C>	var.y	-

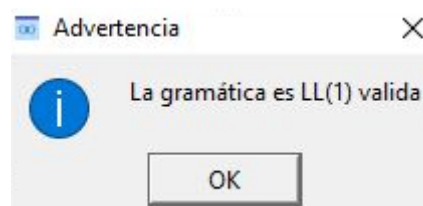
Así, hemos eliminado los no terminales <D> y <E>. No eliminamos a <B> puesto que es más sencillo simplemente editar su estructura eliminando <D>.

Adicionalmente, es necesario eliminar los terminales de la lista de no terminales para poder ejecutar nuestra gramática:

No Terminales

<A>  
<B>  
<C>

5. Al presionar otra vez el botón de *Validar Producciones*, el software nos indicará que la gramática ahora es válida.



6. Ahora trabajaremos con una gramática que posee una producción inalcanzable. Para esto abriremos el proyecto de nombre ----- indicado para esta guía.

7. Dirígete a la sección del analizador sintáctico, veremos una lista de producciones que describen una gramática que posee un no terminal inalcanzable, lo podemos comprobar si presionamos en el botón *Validar Producciones*, el software nos informará la existencia de este tipo de no terminales extraños y señalará la última producción:

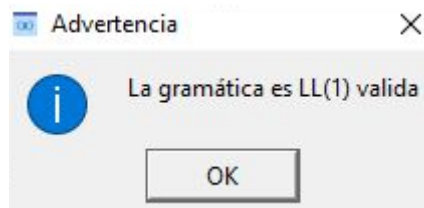
	Lado Izquierdo	Lado Derecho	
1	<A>	var:y <B> var:y	-
2	<A>	var:x	-
3	<B>	<C> var:y	-
4	<C>	var:y	-
5	<D>	<C> var:y	-

8. En general, los problemas de no terminales inalcanzables son más fáciles de ver, se tratan de aquellas producciones a las que no se puede llegar por medio de otra. En este caso podemos observar que ninguna de las producciones anteriores define una instancia de <D>, haciéndola inalcanzable.
9. El siguiente paso será eliminar la producción que contiene a <D>, como dicta la regla, también eliminaremos el no terminal <D> de la lista de no terminales:

	Lado Izquierdo	Lado Derecho	
1	<A>	var:y <B> var:y	-
2	<A>	var:x	-
3	<B>	<C> var:y	-
4	<C>	var:y	-



10. Al presionar otra vez el botón de *Validar Producciones*, el software nos indicará que la gramática ahora es válida.



### Ejercicio de refuerzo:

1. Dentro de la carpeta de "OTRAS\_GRAMÁTICAS" busque el ejercicio "gramática\_nt\_muertos2.json" que permite reconocer secuencias de la forma **1n 0m C 1m 0n** y corrija los no terminales extraños de la gramática.
2. Dentro de la carpeta de "OTRAS\_GRAMÁTICAS" busque el ejercicio "gramática\_nt\_inalcanzables2.json" que reconoce la secuencia **111 000 1010** o **111 000 0101** y corrija los no terminales extraños de la gramática.