G→ADUN

Ejercicio 3: Limitaciones del analizador léxico

Ejercicio 3: Limitaciones del analizador léxico

Conceptos a trabajar:

- Conflictos entre MEF

Proyecto:

A través de GADUN, en la carpeta raíz de la herramienta busque en la sección de *“GRAMÁTICAS_EJERCICIOS”* el proyecto *“ejercicio3_tokens_en_conflicto.json”*. Si no encuentra el directorio o el archivo por favor descargue los de Repositorio GADUN .

Introducción:

El uso de máquinas de estado finito o expresiones regulares en el reconocimiento de patrones para análisis léxico en GADUN conlleva ciertas limitaciones debido a la forma en que estos funcionan.

GADUN utiliza las expresiones regulares de manera que este busca que la sub-secuencia analizada en la secuencia ingresada o código fuente corresponda con la definición o regla de alguno de los tokens definidos, en el caso de no encontrar una coincidencia este simplemente notificará un error léxico, pero en el caso de que más de un Token reconozca la misma secuencia GADUN entrará en un estado de ambigüedad.

Es por eso que en este ejercicio observamos las limitaciones del analizador léxico en GADUN. En el ejercicio el analizador contempla:

Conjuntos	Definición
Números	Secuencia de uno o más dígitos
Letras	secuencia de uno o más caracteres alfabéticos
Variables	Secuencias alfanuméricas que empiezan por un caracter alfabetico

Tabla de conjuntos del analizador

Proceso

1. Abre el proyecto indicado para esta guía y dirígete a la sección de definición “Analizador Léxico”, para este ejercicio solo nos centraremos en esta área.

2. Dentro del proyecto encontrará que están definidos los Tokens de expresión regular correspondientes a los conjuntos de números letras y variables.

	Nombre	Expresión
1	números	d+
2	letras	l+
3	variables	l(d|l)*

   Análise las expresiones regulares de cada uno de los Tokens y el proceso de reconocimiento que llevará la herramienta de GADUN. Tome como secuencia de entrada: *“123 variable”.*

3. A continuación ejecute el editor de GADUN e ingrese la secuencia dada para analizar. Observa que el mensaje en la sección resultado será:

   “Existe un conflicto entre los token: letras variables“

   Este mensaje denota la situación que pudo analizar previamente, los tokens letras y variables pueden reconocer un margen de secuencias al mismo tiempo, esta situación puede producir ambigüedad en cuanto al análisis léxico. Esto denota una de las limitaciones del analizador léxico de GADUN y de otros analizadores en general, cuando dos o más tokens o patrones reconocen un conjunto común de secuencias esto produce ambigüedad en el reconocimiento, y este evento solamente puede solucionarse a través del ingenio del diseñador de compiladores, quien debe buscar estrategias para evitar esta ambigüedad.

   ¿Entonces porque en el ejercicio 2, variables no entra en conflicto con palabras reservadas? GADUN sigue una política de prioridades, primero realiza el reconocimiento de Tokens conjunto y después el de Tokens de

expresión regular, de esta manera, si un token Conjunto reconoce lo mismo que una expresión regular, el analizador tomará como clasificación para la secuencia reconocida el token de conjunto. Por otro lado, si dos tokens del mismo tipo sea Conjunto o Expresión regular GADUN notifica un error de ambigüedad. De esta manera GADUN permite una mayor flexibilidad en cuanto al reconocimiento.

4. Esta situación no es única y puede replicarse con otros patrones, pero en este caso el conflicto contempla el orden en que se presentan los caracteres de la secuencia, lo que quiere decir. Si una secuencia empieza con letras y continua con números o letras es una variable, pero si empieza con números el reconocimiento cambia ya que debe contemplar la secuencia continua a estos números.

   var1 → variable
   2var → número y letras
   123 → número
   letras → letras

   Como se expone en el ejercicio 1, el analizador léxico no tiene dentro de sus tareas contemplar el orden en que se presentan las secuencias, solamente el reconocerlas y transformarlas en componentes léxicos que pueden ser manipulados por el analizador sintáctico.

   Una forma de solucionar este problema es utilizar una gramática para controlar la forma en que se presentan las secuencias. Pero ya que aun no entramos en el análisis sintáctico, veremos que la forma más adecuada para solventar la situación es cambiar el patrón de uno de los tokens en conflicto, por ejemplo, cada vez que desee reconocer letras, estas deberán estar siempre entre **“”**.

   De manera que:

   var1 → variable
   “var” → letras
   123 → números

Ejercicio de refuerzo:

1. Modifique el t¡Token de letras de manera que este no entre en conflicto con el Token de variables.

2. Analise los elementos que pueden presentarse en un lenguaje de programación y qué conflictos pueden presentarse en el reconocimiento léxico de estos.