

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
 FACULTAD DE INGENIERIA  
 ESCUELA DE CIENCIAS Y SISTEMAS  
 LENGUAJES FORMALES Y DE PROGRAMACIÓN SECCIÓN A- Y SECCION B+  
 PRIMER PROYECTO

El proyecto consiste en la elaboración de un intérprete para un lenguaje de manipulación de conjuntos. Se debe leer un archivo de texto donde están programadas todas las instrucciones. Deberá realizar un programa que haga el análisis lexicográfico y sintáctico. El archivo de texto debe tener la siguiente estructura:

PROGRAMA <ID>  
 CONJUNTOS

<Definición de conjuntos >

INICIO

< Operaciones>

RESULTADOS

< Resultados >

FIN

Donde:

**DECLARACION DE CONJUNTOS:** En la sección de conjuntos se debe realizar la declaración de todas los conjuntos que se utilizaran en el programa. Debe haber al menos un conjunto declarado. La estructura de declaración es:

<Identificador> = <conjunto>

Identificador puede ser cualquier identificador definido que no sea palabra reservada.

Conjunto es:

{ <elemento1>, <elemento2>, ....., <elementoN> }

Elemento puede ser un identificador o un número entero cualquiera.

Ningún elemento puede repetirse en el mismo conjunto.

Ningún elemento o conjunto debe llamarse igual que una palabra reservada.

Letras en minúsculas y/o mayúsculas son iguales.

Un conjunto puede no tener elementos, es decir conjunto vacío.

Por ejemplo:

CONJUNTOS

A = { 1, 2, 3, 4 } B = { 1 }

CB1 = { V1, valor2, 1, 5, 3456, 3, Dia }

V = { }

**OPERACIONES:**

Se pueden realizar las siguientes operaciones de conjuntos:

Operación	Operador	descripción
UNION	+	Unión de dos conjuntos
INTERSECCION	*	Intersección de dos conjuntos
DIFERENCIA	-	Diferencia de un conjunto.
DIFERENCIA SIMETRICA	<>	Diferencia simétrica
Complemento	@	Complemento de un conjunto

La operación de complemento se realiza sobre el conjunto Universo que siempre existe con el nombre Universo y esta formado por la Unión ( + ) de todos los elementos de todos los conjuntos definidos.

Las operaciones deben asignarse a un conjunto definido. La operación de asignación es destructiva, es decir, al asignar el resultado de una expresión de conjuntos pierde sus elementos actuales y recibe como elementos los elementos resultado de la evaluación de la expresión. Los operadores no tienen presidencia y se ejecutan de izquierda a derecha, es decir, el resultado de la primera operación se ejecuta con la siguiente.

Ejemplo  
INICIO

```
V = A + B - CB1 + V <> V;  
A = A + V;  
A = A@;
```

RESULTADOS:

```
SALIDA: <NOMBRE_SALIDA>  
CB1  
B  
V
```

La salida debe hacerse a un archivo. <NOMBRE\_SALIDA> es el nombre del archivo donde se escribirán los resultados.

Luego se coloca cada conjunto que se quiera colocar en el archivo de salida. Solamente identificadores definidos. En el archivo de salida se debe colocar el conjunto y los elementos de cada conjunto. Por ejemplo el archivo de salida seria:

```
CB1 = { V1, valor2, 1, 5, 3456, 3, Dia }  
B   = { 1 }  
V   = { 2, 4 }
```

A continuación un archivo de ejemplo:

PROGRAMA EJEMPLO

CONJUNTOS

```
CONJUNTO1 = { UNO, DOS , TRES, 4 }  
C2= { }  
C3= { UNO, DOS }
```

INICIO

```
C2 = CONJUNTO1 + C3;  
C3 = CONJUNTO1 - C3;
```

RESULTADOS

```
Salida salida.txt  
CONJUNTO1  
C2  
C3
```

FIN

El archivo salida.txt tendrá:

```
CONJUNTO1 = {UNO, DOS , TRES, 4}  
C2 = {UNO, DOS , TRES, 4}  
C3= { UNO, DOS }
```

Además, del programa, la documentación del proyecto debe tener:

1. La gramática completa de todo el lenguaje definido.
2. Las expresiones regulares completas que se deben utilizar en la solución.
3. El método del árbol completo. Presentando el árbol completo, la tabla de la función siguiente, la matriz de transiciones y todo el procedimiento detallado, explicando los estados.
4. Estados de aceptación definidos.
5. Código fuente completo. Indicando separadamente donde ha aplicado el DFA a la solución.
6. Documentación de usuario y técnica completa.

**El programa debe hacerse en lenguaje Java**

Fecha de entrega: sábado 6/octubre/2007