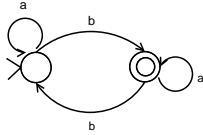


Auxiliar 2  
**CC30B**  
 Profesor: Alejandro Hevia  
 Auxiliares: Rodrigo Canovas, Fernando Krell

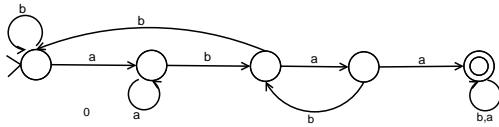
**Problema 1**

Construya la expresión regular correspondiente a cada uno de los autómatas finitos descritos a continuación.

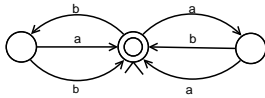
a)



b)



c) (*Propuesto*)



**Problema 2**

Demuestre que los siguientes lenguajes no son regulares.

- a)  $L = \{w \in \{a\}^* | w = a^{n^2}, n \in \mathbb{N}\}$
- b)  $L = \{w \in \{a\}^* | w = a^n \text{ con } n \in \mathbb{N} \text{ primo}\}$
- c)  $L = \{w \in \{0,1\}^* | w \text{ no contiene la misma cantidad de 1's que 0's}\}$

**Problema 3**

Sean  $p, q \in \mathbb{N}, q \geq 1$ . Una **progresión aritmética** se define como el conjunto  $\{p + qn | n \in \mathbb{N}\}$ .

- Demuestre que si  $L \subseteq \{a\}^*$  y  $\{n | a^n \in L\}$  es una progresión aritmética entonces  $L$  es regular.
- Demuestre que si  $L \subseteq \{a\}^*$  y  $\{n | a^n \in L\}$  es una unión finita de progresiones aritméticas, entonces  $L$  es regular. (*Propuesto.*)

**Problema 4**

Se sabe por materia vista en clases que  $L = \{w | w = xx^R, x \in \{a, b, c\}^*\}$  no es regular. Muestre que este lenguaje si se puede derivar de una gramática libre del contexto. Describa formalmente la gramática.

## Parte 5: Ejemplo JLEX y CUP para Tarea 3

Incluir en la variable de entorno CLASSPATH el directorio CUP que bajaron. Incluir en la variable de entorno CLASSPATH el directorio anterior al directorio JLEX (poner el directorio JLEX en el bin de java funciona).

Para compilar.

```
> java JLex.Main archivoLex.lex
> java java_cup.Main archivoCup.cup
> javac ArchivoLex.lex.java
> javac Programa.java
```

para ejecutarlo

```
> java Programa
```

Definición de la gramática en archivo Gram.cup:

```
import java_cup.runtime.*; import java.util.*; import java.io.*;
parser code{
    public static void main(String[] args) throws Exception{
        new parser(new Yylex(System.in)).parse();
    }
};

terminal SUM,MIN; terminal MUL,DIV; terminal LPAREN,RPAREN;

terminal Double NUM; terminal NL;

non terminal Double E; non terminal ROOT;

precedence left SUM,MIN;

precedence left MUL,DIV;

start with ROOT; ROOT ::=E:e NL{:System.out.print(e); :};

E ::=E:e1 SUM E:e2 {: RESULT=new
Double(e1.doubleValue()+e2.doubleValue()); :}
| E:e1 MIN E:e2 {: RESULT=new Double(e1.doubleValue()-e2.doubleValue()); :}
| E:e1 MUL E:e2 {: RESULT=new Double(e1.doubleValue()*e2.doubleValue()); :}
| E:e1 DIV E:e2 {: RESULT=new Double(e1.doubleValue()/e2.doubleValue()); :}
| LPAREN E:e RPAREN {: RESULT=e; :}
| NUM:e {:RESULT=e; :};
```