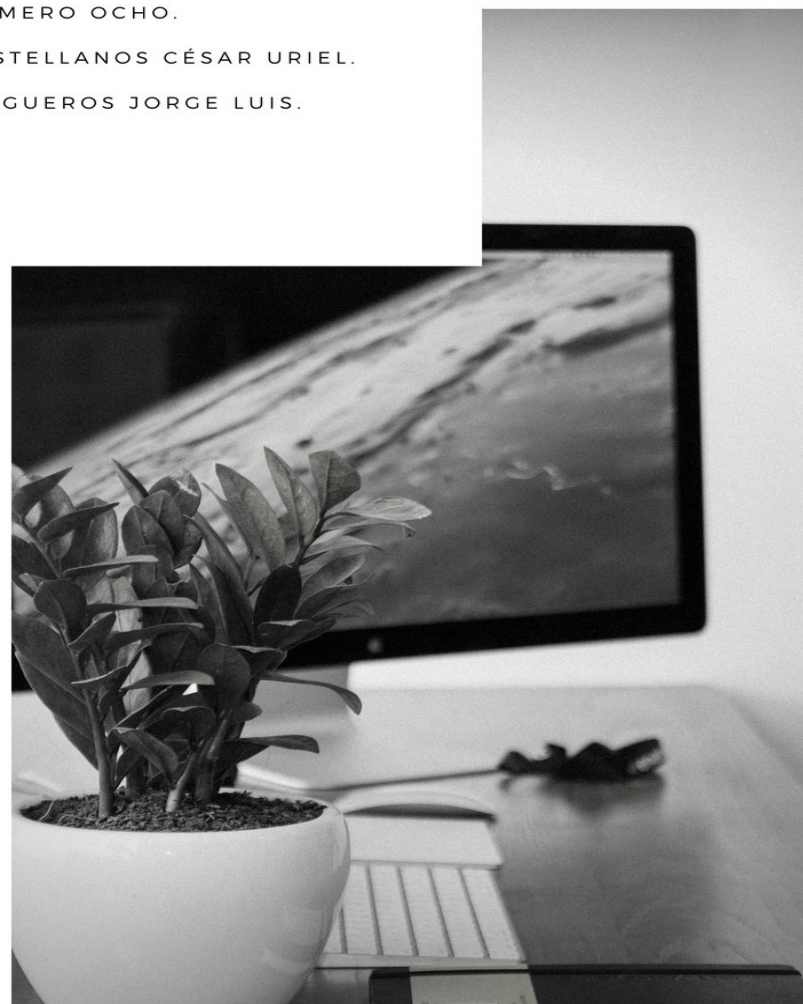




---

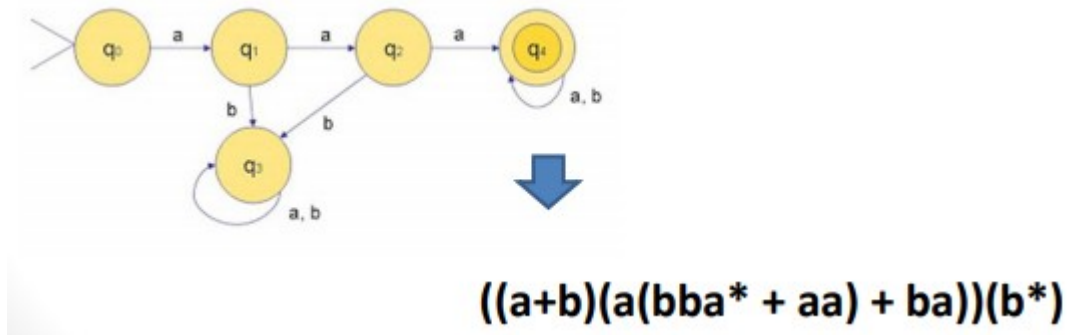
INSTITUTO POLITÉCNICO NACIONAL.  
ESCUELA SUPERIOR DE CÓMPUTO.  
REDES DE COMPUTADORAS  
PRÁCTICA NÚMERO OCHO.

ALUMNO: HERNÁNDEZ CASTELLANOS CÉSAR URIEL.  
PROFESOR: ROSAS TRIGUEROS JORGE LUIS.



Transformación de una expresión regular en un autómata finito.

- Dada una expresión regular existe un autómata finito capaz de reconocer el lenguaje que ésta define.
- Recíprocamente, dado un autómata finito, se puede expresar mediante una expresión regular del lenguaje que reconoce.



Construcción de Thompson de un AFN a partir de una expresión regular

- La construcción de Thompson construye un AFN a partir de cualquier expresión regular.
- La construcción de Thompson construye a partir de una expresión regular  $r$  un AFN que reconoce el lenguaje definido por  $r$ , esto se realiza con el objetivo de que en un algoritmo siguiente se pueda generar un AFD mínimo equivalente.
- Utiliza una notación estándar para generar el AFN

Nomenclatura de Thompson

Para la representación de una cadena vacía se utiliza el símbolo  $\epsilon$  o  $\lambda$



**Cadena Vacía**

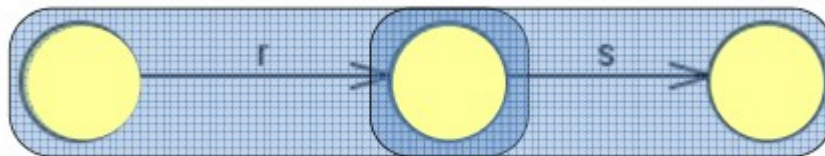
- Para representar un símbolo, se utilizan dos estados y una transición para el movimiento con el símbolo.

**r**



Para la concatenación de dos símbolos únicamente se unen.

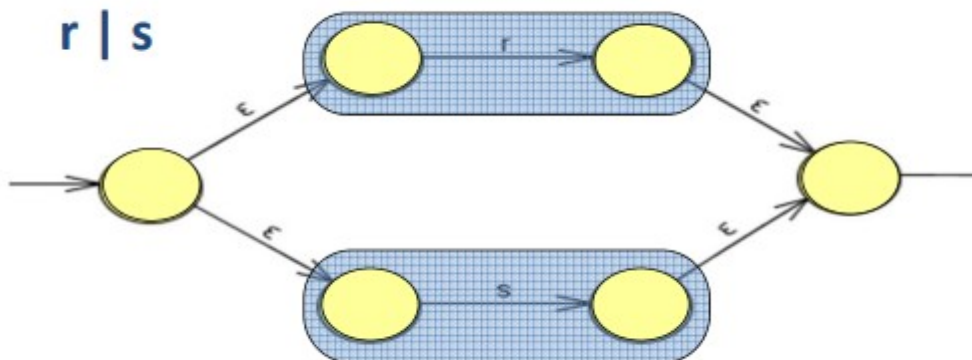
**rs**



**Concatenación de símbolo**

Para la elección de alternativas, crear transiciones  $\epsilon$  para la unión de las transiciones.

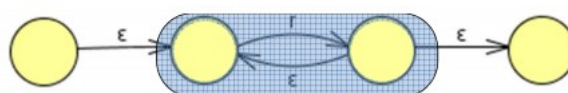
**r | s**



**Elección de alternativas**

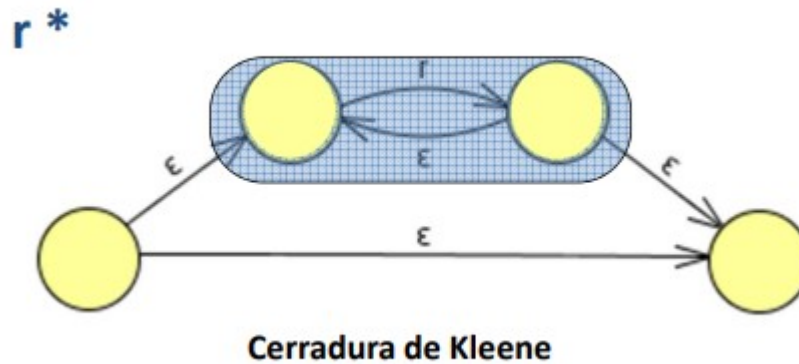
Para la cerradura positiva, se agregan transiciones  $\epsilon$  para retornar al estado previo, permitiendo agregar 1 o mas veces el símbolo

**r +**



**Cerradura positiva**

- Para la cerradura de Kleene, se agregan transiciones  $\epsilon$  para retornar a estado previo. Y otra transición  $\epsilon$  para saltar la transición con  $r$ .



Material y equipo.

El material utilizado en la práctica es el siguiente:

Herramientas de software:

- \* Oracle VM VirtualBox
- \* Ubuntu 17.04
- \* Python 3.6.2
- \* eric6 Web Browser (QtWebKit)
- \* Jflap

Herramientas de hardware:

- \* Computadora personal



## Desarrollo de la práctica

```
uriel@Uriel-PC:~/Escritorio$ python prac.py
a*b*
estado = ['estado0', 'estado1', 'estado2', 'estado3', 'estado4', 'estado5', 'estado6', 'estado7']
s = estado2
F = estado7
DELTA =
('estado0', 'a') - ['estado1']
('estado1', 'e') - ['estado0']
('estado1', 'e') - ['estado3']
('estado2', 'e') - ['estado0']
('estado2', 'e') - ['estado3']
('estado3', 'e') - ['estado6']
('estado4', 'b') - ['estado5']
('estado5', 'e') - ['estado4']
('estado5', 'e') - ['estado7']
('estado6', 'e') - ['estado4']
('estado6', 'e') - ['estado7']
uriel@Uriel-PC:~/Escritorio$
```

La práctica número ocho de teoría computacional, consistió en implementar un script que fuera capaz de convertir una expresión regular en un autómata, para esto el profesor con anterioridad nos explico el proceso de manera teórica en clase, ahora el reto era implementarlo en un computador, lo que en lo personal resultó una de las prácticas de las cuales me causaron mayor conflicto, sobre todo el como leer la expresión regular, para poder extraer de alguna manera cada uno de los elementos que lo constituyen, para posteriormente armar nuestro autómata.

## Conclusión

El algoritmo de thompson nos auxilia en obtener un autómata finito no determinista con transiciones vacías (AFND-V) a partir de una expresión regular.

Las expresiones regulares pueden ser reconocidas por un autómata finito determinista ya que los lenguajes regulares del tipo tres serán reconocidos por un autómata más restrictivo en lo que existe una equivalencia entre no determinismo y determinismo.

Con respecto a las dificultades que se tuvieron en la realización de la práctica fueron bastantes, por ello el prolongado tiempo de entrega.

## Referencias

*Algoritmo de Thompson.* (2017). *Los diccionarios y las enciclopedias sobre el Académico.* Retrieved 20 Nov 2017, from <http://www.esacademic.com/dic.nsf/eswiki/1266054>

*Teoría de Lenguajes, Gramáticas y Autómatas Para Informáticos.* (2000). [Place of publication not identified].

*Traductores e Intérpretes UCAB : Metodo de Thompson.* (2017). *Suigeneris.org.* Retrieved 20 Nov 2017, from <http://www.suigeneris.org/UCABTI/Metodo%20de%20Thompson.html>