

# TC2020 Matemáticas Computacionales

## Programación en FLEX y BISON

### TAREA 4

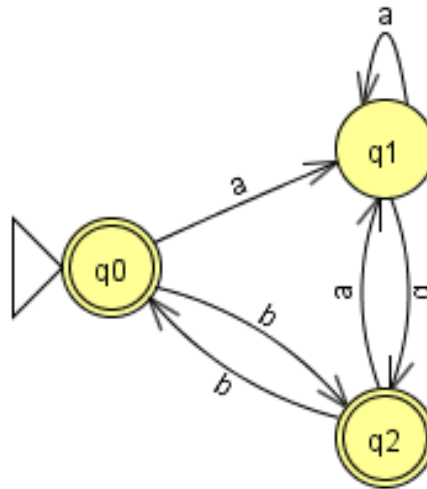
**Forma de trabajo:** En equipos de 2 integrantes

**Forma de entrega:** Subir los programas de FLEX y BISON contruidos a BLACKBOARD.

En esta asignación pondrán en práctica varios conocimientos y habilidades adquiridas durante el curso, relacionadas con el reconocimiento y descripción de lenguajes regulares y lenguajes libres de contexto, así como conceptos y herramientas requeridas en la construcción de compiladores, que es una de las aplicaciones principales de las Matemáticas Computacionales.

El objetivo de esta asignación es escribir un parser y un scanner para la conversión entre dos formatos de descripción de autómatas finitos deterministas (AFDs) en texto. La meta entonces es escribir **dos programas**: un programa en el generador de scanners FLEX y otro programa en el generador de parsers BISON. Los programas deberán utilizarse para poder **transformar los archivos XML** que genera la aplicación JFLAP al salvar un autómata finito en su **quíntuplo de la descripción formal de autómatas finitos** en notación de conjuntos.

Ejemplo de entrada y salida esperada para el siguiente autómata finito:



```
----- JFLAP -----
<?xml version="1.0" encoding="UTF-8" standalone="no"?><!--Created with
JFLAP 6.4.--><structure>#13;
  <type>fa</type>#13;
  <automaton>#13;
    <!--The list of states.-->#13;
    <state id="0" name="q0">#13;
      <x>96.0</x>#13;
      <y>122.0</y>#13;
      <initial/>#13;
      <final/>#13;
    </state>#13;
    <state id="1" name="q1">#13;
      <x>207.0</x>#13;
```

```

        <y>74.0</y>#13;
    </state>#13;
    <state id="2" name="q2">#13;
        <x>207.0</x>#13;
        <y>191.0</y>#13;
        <final/>#13;
    </state>#13;
    <!--The list of transitions.-->#13;
    <transition>#13;
        <from>1</from>#13;
        <to>1</to>#13;
        <read>a</read>#13;
    </transition>#13;
    <transition>#13;
        <from>1</from>#13;
        <to>2</to>#13;
        <read>b</read>#13;
    </transition>#13;
    <transition>#13;
        <from>0</from>#13;
        <to>1</to>#13;
        <read>a</read>#13;
    </transition>#13;
    <transition>#13;
        <from>0</from>#13;
        <to>2</to>#13;
        <read>b</read>#13;
    </transition>#13;
    <transition>#13;
        <from>2</from>#13;
        <to>1</to>#13;
        <read>a</read>#13;
    </transition>#13;
    <transition>#13;
        <from>2</from>#13;
        <to>0</to>#13;
        <read>b</read>#13;
    </transition>#13;
</automaton>#13;
</structure>

```

----- QUÍNTUPLO -----

```

ESTADOS = {q0, q1, q2}
INICIAL = q0
FINALES = {q0, q2}
TRANSICIONES = {(q1, a, q1), (q1, b, q2), (q0, a, q1), (q0, b, q2),
(q2, a, q1), (q2, b, q0)}
ALFABETO = {a, b}

```

-----

Los programas también se deben de documentar internamente mediante comentarios de C. Como mínimo espero que describan en comentarios cada estructura, variable, regla y función incluidas en los programas. Así como cada elemento no evidente o muy relevante en sus programas.

Sus fuentes deben compilar usando exactamente los comandos vistos en clase. Se supone que los archivos fuente de entrada (descripciones de los autómatas) fueron creados léxica, sintáctica y semánticamente correctos para una conversión exitosa. Para ejecutar los programas, el archivo fuente se toma de la entrada estándar (teclado) y se produce el resultado de la conversión en la salida estándar (pantalla). Sin embargo, se podrá redirigir tanto la entrada como la salida para leer un archivo de entrada y crear un archivo de salida, como en el comando siguiente:

```
>./jflap2afd.exe < afd.jff > afd.txt
```

Solamente uno de los integrantes del equipo deberá subir el trabajo a Blackboard, incluyendo el archivo compreso **M\_Tarea4.zip**, donde deberán sustituir la M por las matrículas de los integrantes del equipo, separadas mediante caracteres de subrayado (\_), que contenga los 2 programas fuente (**jflap2afd.l** y **jflap2afd.y**). Indicar tanto en el mensaje de la asignación, como dentro del documento y dentro de cada archivo fuente (como comentarios de C), el nombre de los integrantes del equipo y sus matrículas, ya que **solamente los que aparezcan tendrán derecho a calificación**.

## RECOMENDACIONES

- **No comenzar el trabajo unos cuantos días antes de la entrega.**

El dejar las cosas para lo último puede provocar problemas de organización, presión por la cercanía de la fecha de entrega, etc. A veces es necesario hacer pausas durante sesiones de trabajo para meditar una idea, tomar un poco de aire o simplemente dejar descansar al cerebro unos instantes.

- **Desarrollar la tarea en forma incremental.**

Siempre es malo desayunar todo lo de una semana en un sólo día o comerse el pastel de una sola mordida. Resuelvan primero las partes más fáciles de la tarea y a medida que esas partes queden resueltas podrán avanzar entonces a los casos más difíciles. Esto es importante pues se harán varias pruebas a la tarea, algunas fáciles, otras más difíciles, y cada prueba les hace ganar puntos. En caso de que su tarea esté incompleta es conveniente como sea que suban lo que alcanzaron a hacer, pues de lo contrario su tarea sería evaluada con 0 (cero).

- **Dividan el trabajo adecuadamente.**

Si la tarea se encarga en equipo no es para que los miembros del equipo estén pegados a la computadora al mismo tiempo tratando de resolver el problema. Es para que identifiquen de manera inteligente como pueden

dividir la tarea en sub-tareas que puedan ser resueltas en forma independiente y que luego puedan ser ensambladas para generar el sistema final.

- **Investiguen lo que hace falta para comenzar la tarea antes de empezar a resolverla.**

Muchas veces por no invertir un poco de tiempo a actividades de investigación se reinventa el hilo negro y se programan cosas para las cuales los lenguajes o herramientas computacionales nos proveen soluciones predefinidas. Por ejemplo, no es necesario que investiguen como generar salidas a archivos por medio de estatutos de I/O pues Unix permite redirigir la salida estándar a un archivo.