

Tarea 2

Instrucciones:

- Antes de comenzar revise la sección “Evaluaciones y reglas del juego” de la página web del curso <http://www.dcc.uchile.cl/~ahevia/cc30b/2009/>.
- NO se tolerarán copias o cualquier otro comportamiento de deshonestidad académica. En particular, obtener y utilizar soluciones o implementaciones desde Internet (si existiesen) o de versiones anteriores del curso se considerará copia. Ante la primera instancia de copia, el/los alumno(s) obtendrá(n) inmediatamente nota de tareas igual a 1.0 lo cual significa reprobar el curso. Además, los antecedentes se pondrán a disposición de la Escuela, arriesgando expulsión de la carrera.
- Está tarea se puede discutir en grupos de a lo más dos personas. No habrá restricciones en cuanto a la colaboración permitida.
- La tarea debe entregarse en forma digital, incluyendo todos los archivos necesarios para poder compilar su tarea. Además debe entregar de un informe breve (de a lo más tres páginas) en formato PDF. Ver al final más detalles respecto a los contenidos del informe. Se recomienda utilizar LaTeX aunque otros sistemas como MS Word u otros editores similares son aceptables. Si utiliza MS Word para escribir la tarea, NO envíe el archivo .doc. Debe convertirlo primero a formato PDF (hay diversos convertidores o *distillers* disponibles gratis en Internet). No se aceptarán tareas que hayan sido escritas a mano.
- **Importante:** Dedique tiempo a escribir su informe, no lo deje para el último minuto. Tareas que no compilan recibirán nota 1.0.
- **Fecha de Entrega:** 23 de Abril 2009, 23:59hrs. La entrega debe hacerse vía u-cursos.

El objetivo de esta tarea es escribir un programa que obtenga una expresión regular desde la entrada estándar, luego lea una palabra, y finalmente indique las posiciones dentro de la palabra donde existe una subcadena descrita por la expresión regular (ver ejemplos al final). Para esto *debe* seguir los siguientes pasos:

1. Leer una expresión regular desde la entrada estándar.
2. Convertir la expresión regular a un AFND e imprimir el autómata.
3. Convertir el AFND a AFD e imprimir el autómata.
4. Recibir una palabra desde la entrada estándar e indicar todas las posiciones dentro de la palabra donde se inicia una subcadena descrita por la expresión regular indicada. Las posiciones deben contarse desde 1.

Notación

A continuación se muestra la tabla de símbolos que se utilizarán en la expresión regular y su significado:

Símbolo u operación	Traducción
Unión	
Estrella de Kleene	*
Epsilon o palabra vacía	_ (underscore)
Conjunto Vacío	0
Paréntesis	()

El conjunto de caracteres (alfabeto) que puede aparecer en un texto o e.r. (además de los símbolos mostrados) es:

$$\{a, b, \dots, z\}$$

La operación de concatenación es implícita, esto es, ab denota la concatenación del símbolo a con el símbolo b .

Formato para describir los autómatas

Los autómatas deben imprimirse en el formato de JFLAP (ver documentación en <http://www.jflap.org/>). Por ejemplo, el archivo que describe el AFND mostrado en

<http://www.jflap.org/tutorial/fa/createfa/images/nfa.png>

está en <http://www.jflap.org/tutorial/fa/createfa/files/ex1.3a.jff>. Cualquier duda, preguntar en el foro.

Archivos a Entregar

Su implementación debe ser en C, C++, o Java (otros lenguajes son posibles, pedir autorización primero al profesor). Deben respetarse los símbolos pedidos en el enunciado y la salida del programa debe ser a través de la salida estándar.

Puede realizarse en grupos de máximo 2 personas. La tarea debe ser enviada por u-cursos y debe incluir el código de fuente y un breve informe (a lo más 3 páginas) indicando:

- Nombre de los dos miembros del grupo,
- Breve descripción del programa,
- Instrucciones de compilación y ejecución, y
- Ejemplos de uso.

Basta que la tarea sea enviada via u-cursos por uno de los integrantes del grupo.

Ejemplo

Un ejemplo de ejecución correcta es: *(notar que lo mostrado después de “AFND:” (respectivamente “AFD:”) es la descripción del AFND (resp. AFD) equivalente a la expresión regular dada como entrada al programa. Esa descripción debe estar en formato JFLAP.)*

Ingrese expresion regular?: a*b*

AFND:

```
<?xml version="1.0"?>
<structure>
  <type>fa</type>
  <state id="1">
    <x>10.0</x>
    <y>20.0</y>
    <initial/>
  </state>
  ...
  <transition>
    <from>5</from>
    <to>6</to>
    <read>b</read>
  </transition>
</structure>
```

AFD:

```
<?xml version="1.0"?>
<structure>
  ...
</structure>
```

```
Ingrese palabra a buscar?: aababbaba
  substring: a, posicion inicio: 1
  substring: aa, posicion inicio: 1
  substring: aab, posicion inicio: 1
  substring: a, posicion inicio: 2
  substring: ab, posicion inicio: 2
  substring: b, posicion inicio: 3
  substring: a, posicion inicio: 4
  substring: ab, posicion inicio: 4
  substring: abb, posicion inicio: 4
  substring: b, posicion inicio: 5
  substring: bb, posicion inicio: 5
  substring: b, posicion inicio: 6
  substring: a, posicion inicio: 7
  substring: ab, posicion inicio: 7
  substring: b, posicion inicio: 8
  substring: a, posicion inicio: 9
```

```
Ingrese palabra a buscar?: bba
  substring: b, posicion inicio: 1
  substring: bb, posicion inicio: 1
  substring: b, posicion inicio: 2
  substring: a, posicion inicio: 3
```

Notas:

- En el ejemplo anterior, la parte de la descripción del AFD es sólo un ejemplo de cómo debe imprimir el autómata. En particular, el ejemplo arriba NO muestra exactamente el AFD correspondiente a la expresión regular a^*b^* . Un ejemplo del AFD equivalente a la expresión regular del ejemplo escrito en formato JFLAP puede encontrarse en <http://www.dcc.uchile.cl/~ahevia/cc30b/2009/tarea2/ejemplo.jff>. Además, hay partes abreviadas como “...” arriba que su tarea sí debe mostrar.
- Sin embargo la parte final del ejemplo donde se ingresa una palabra y muestra las subcadenas SI es representativo de lo que su tarea debiera mostrar – los números de posiciones son correctos.
- Los autómatas no necesitan estar optimizados ni ser de alguna forma en particular, a excepción del AFND el cual no debe mostrar todos los estados posibles, sólo los alcanzables. Parte importante de la nota de su tarea es la correctitud de los autómatas generados a partir de la expresión regular.

Suerte!