

Teoría de la Computación

Trabajo Práctico

1er semestre de 2020

Consideraciones generales

Implementar los siguientes ejercicios en lenguaje *Java*. La entrega debe realizarse el día lunes 23 de junio. Puede no constar de todos los ejercicios, pero debe contener al menos un ejercicio completo. Cada ejercicio debe incluir tests unitarios con los cuales mostrar la funcionalidad. La entrega consta de una demo remota durante la clase de ese día, y el envío por mail del código hasta las 18hs de ese día. No se recibirán TPs luego de ese momento. El trabajo práctico puede realizarse de manera individual, o en grupos conformados por hasta dos personas. No es necesario entregar informe, ni decisiones tomadas, ni conclusiones, ni verso.

Ejercicios

1. (4 pts.) Sea la siguiente definición de un AFND, que debe leerse de un archivo:

```
<símboloInput>, <símboloInput>, ..., <símboloInput>
<cantEstados>
<estadoFinal>, <estadoFinal>, ..., <estadoFinal>
<estado>, <símboloInput> -> <estado>
<estado>, <símboloInput> -> <estado>
```

La primera línea lista los elementos del alfabeto de input. La segunda indica la cantidad de estados, la tercera el conjunto de estados finales, y el resto de las líneas la función de transición (los espacios extra en el archivo deben ignorarse). Ejemplo:

```
a, b, c, d
10
1, 2, 3
1, a -> 3
1, a -> 4
1, a -> 5
1, d -> 4
```

Asumimos que el estado 1 siempre será el inicial, y que los estados están numerados correlativamente de 1 a la cantidad de estados. Se pide

- a) Transformar el AFND en AFD, y determinar usando este último si el string pertenece al lenguaje.
 - b) (Opcional, 4p. extra) Dar una implementación directa *multithreaded* del AFND, donde cada thread corresponda a un camino de ejecución dentro del autómata.
2. (6 pts.) Dada una gramática G libre de contexto leída desde un archivo.
 - a) Implementar un parser *no* recursivo predictivo para la gramática G (para ellos, construir primero la tabla de parsing, y luego implementar el algoritmo visto en clase).

El formato del archivo debe ser un conjunto de líneas:

`<variable> -> <body>`

que representan las producciones de G , donde `<variable>` es un string de la forma $X_{\{i\}}$ con i un número entero. y `<body>` es un string conformado por variables como las indicamos recién y/o caracteres en minúscula, con los que denotamos los terminales de la gramática. Por ejemplo, esta sería una línea válida:

`$X_{\{4\}}$ -> $X_{\{34\}}$ a $X_{\{1\}}$ bcd`

Asumimos que $X_{\{1\}}$ es el símbolo inicial.