Agustín Fernández
Jesús Rojas

Project-Teoria-Computacion

Este es el Proyecto Final de Introducción a la Teoría de la Computación. Aquí se pueden encontrar implementados los siguientes módulos:

1. Autómata Finito Determinista (AFD)
2. Autómata Finito con Pila Determinista (AFPD)
3. Autómata Finito con Pila No Determinista (AFPN)
4. Autómata Finito con 2 Pilas (AF2P)
5. Máquina de Turing – Modelo Estándar (MT)
6. Máquina de Turing Modelo con una Cinta dividida en Pistas (MTP)
7. Máquina de Turing Modelo con Múltiples Cintas (MTMC)

🟢 El proyecto fue desarrollado en Eclipse IDE (con lo que se recomienda su uso) y es necesario tener instalado **JDK14 o superior** para poder ejecutarlo.
🟢 Es necesario tener un mínimo conocimiento teórico sobre lo anterior descrito para poder entender y usar este proyecto.
🟢 Importante no mover los archivos *.jar* ni *.bat*, así como no mover ni modificar el contenido existente en la carpeta **/src/Pruebas** (sí se pueden añadir archivos).

Ejecución

Para poder interactuar con los módulos se creó una clase (src/ProbarModulos.java) en la que se puede acceder a algunos ejemplos por módulos y se pueden usar archivos especiales para poder crear y usar los autómatas o las máquinas de Turing que usted desee.

Entonces, para **ejecutar** rápidamente, si se está usando Windows, haciendo click en el archivo **comandoCMD.bat** o vía CMD se puede usar el comando java -jar PITC-Ejecutable.jar ubicándose en el directorio del proyecto. Si todo fue bien, aparecerá un menú en el que se puede elegir, según el número, el módulo a probar.

Los menús son bastante descriptivos en lo que piden, con lo que, en ocasiones se pedirá ingresar números para elegir o decir el número de repeticiones, o cadenas de texto para ser probadas.

Procesamiento de varias cadenas

Cuando se elige la opción *Procesar una lista de cadenas, viendo el procesamiento de cada una de ellas*, se pedirá un nombre para guardar los procedimientos en un archivo de texto y quedará guardado en **/src/ProcesamientoCadenas/NombreModulo/nombreArchivo.txt**.

Si hace varias pruebas, tenga presente que los archivos de texto pueden ser sobrescritos, por lo que es buena idea usar diferentes nombres.

Agustín Fernández
Jesús Rojas

Crear y usar un nuevo Autómata o Máquina de Turing

Todos los módulos tienen la opción de, con su propio archivo especial, permitir crear y usar nuevos ejemplos de Autómatas o Máquinas de Turing. Las extensiones son:

- (AFD) .dfa
- (AFPD) .dpda
- (AFPN) .pda
- (AF2P) .msm
- (MT) .tm
- (MTP) .ttm
- (MTMC) .mttm

Para usar los archivos, estos deben estar guardados en **/src/Pruebas/NombreModuloAUsar** con su respectiva extensión y teniendo presente que no tengan el mismo nombre de algún archivo ya existente. Además, en este link se puede encontrar un archivo PDF, creado por el profesor Juan Mendivelso, en el que se recopilan y se explica a detalle cómo deben ser los formatos de los archivos en mención.

🟢 Se pueden abrir con el bloc de notas los archivos ya existentes en las distintas carpetas de **/src/Pruebas/** para ver el formato. ¡No modificarlos!

Breve explicación de la Estructura del código

Cada módulo expuesto se encuentra en un paquete (con el mismo nombre) junto a todas las dependencias y clases que necesita para su ejecución. La clase de que contiene el *main()* (de la que se partió para hacer el ejecutable) se encuentra separada en el paquete por defecto.

Ahora bien, para representar los diferentes conjuntos de estados y alfabetos se usó la estructura de datos Set. Para las transiciones están las clases "*FuncionTransicion*" en las que hay una matriz cuya dimensión depende de los parámetros de la función de transición y se usan tablas Hash y Vectores para establecer una relación eficiente entre los estados/símbolos y las posiciones de la matriz.

En los módulos que no son deterministas se usa un **Arbol** de Transiciones para ejecutar y guardar todas las posibles combinaciones que surgen de forma eficiente. En las Máquinas de Turing se usa una clase anidada en la Función de Transición, Transicion, para facilitar el acceso de los datos en la matriz de transiciones y en MTMC se usa una clase *cintaMTMC* para facilitar los movimientos entre las diferentes cintas.