

Proyecto No. 3

Instrucciones generales:

- Este proyecto es en **parejas**. Deberá inscribirse en un grupo de Canvas asociado a esta entrega.
- Cree un repositorio **privado** en GitHub.com para alojar el código correspondiente para este proyecto.
- Añada los usuarios de GitHub de su catedrático y de su(s) auxiliar(es) con rol de *collaborator*. Estos se encuentran en la sección de información del curso en Canvas.
- Coloque todas las instrucciones que considere pertinentes en un archivo denominado README.md en la raíz del repositorio.

Descripción

Este proyecto consiste en que usted desarrolle un simulador de Máquinas de Turing, con el cual usted sea capaz de introducir la descripción formal de la máquina de Turing y con ello su programa sea capaz de simular una entrada que usted configure y posteriormente su programa la simule con su implementación de la Máquina de Turing. Como resultado de la simulación, usted tendrá una configuración final sobre la cinta y deberá indicar si el resultado final sobre la cinta es aceptado o no, utilizando las descripciones instantáneas de su MT.

Usted deberá utilizar el lenguaje de programación de su elección. Se recomienda utilizar Python. A continuación, se detallan los objetivos y el funcionamiento esperado.

Objetivos

- **Generales**
 - Implementación de una simulación para Máquina de Turing.
- **Específicos**
 - **Implementación de un parsing sobre la configuración de la MT.**
 - Implementación de un parsing sobre la entrada en las cintas.
 - Diseño e implementación de la arquitectura a utilizar para las estructuras de datos que alojarán las configuraciones de su MT y permitirán una simulación utilizando las funciones de transición descritas
 - Implementación de la visualización de descripciones instantáneas en cada paso de la simulación.
 - Implementación de la lógica para aceptar inputs en las cintas.

Especificación del funcionamiento del programa

- Entrada
 - Un archivo yaml con la estructura indicada a continuación:

```
estructura-mt.yaml

---
# Lista de estados
q_states:
  q_list:
    - 'lista de estados de la MT'
    - ...
  initial: '0'
  final: '5'
# Alfabetos de la MT
alphabet:
  - 'lista de símbolos del alfabeto'
  - ...
tape_alphabet:
  - 'lista de símbolos del alfabeto de la cinta'
  - 'recuerde que incluye los símbolos de alphabet también'
# para definir el símbolo de blank, usaremos null o None, que en yaml se expresa con
una entrada vacía, como se presenta a continuación:
-
# Función de transición
delta:
# Es una lista en donde cada elemento de la lista es un objeto que guarda la
información sobre los parámetros a utilizar y el resultado al aplicar la función
- params:
    initial_state: 'estado en el que inicia'
    # Valor guardado en caché
    mem_cache_value: 'Puede ser blank o contener algo'
    tape_input: 'el input leído sobre la cinta'
  output:
    final_state: 'estado en el que termina'
    mem_cache_value: 'Puede ser blank o contener algo'
    tape_output: 'el resultado sobre la cinta'
    tape_displacement: 'desplazamiento sobre la cinta, puede ser L, R o S'
# Cadenas a ser simuladas por su MT luego de parsearla y generarla
simulation_strings:
  - aab#aab
  - ab#ab
  - abbababa#aba
```

- A continuación, un ejemplo de una MT, incompleta. En este ejemplo estamos definiendo funciones de transición: $\delta([q_0, B], a) = ([q_1, a], B, R)$ y $\delta([q_0, B], b) = ([q_1, b], B, R)$ respectivamente:

```
example_mt_1.yaml

---
q_states:
  q_list:
    - '0'
    - '1'
    - '2'
    - '3'
    - '4'
    - '5'
    initial: '0'
    final: '5'
  alphabet:
    - a
    - b
  tape_alphabet:
    - "#"
    - X
    - 
  delta:
    - params:
        initial_state: '0'
        mem_cache_value:
        tape_input: a
      output:
        final_state: '1'
        mem_cache_value: a
        tape_output:
        tape_displacement: R
    - params:
        initial_state: '0'
        mem_cache_value:
        tape_input: b
      output:
        final_state: '1'
        mem_cache_value: a
        tape_output:
        tape_displacement: R
  simulation_strings:
    - aab#aab
    - ab#ab
    - abbababa#aba
```

- Su programa solamente debe reconocer MT de una cinta, no debe implementar multi-cinta.
 - Esta configuración incluirá de una vez el listado de cadenas a simular por su MT.
 - Recuerde que el alfabeto de la cinta contiene al alfabeto de entrada (input) más otros símbolos que solo se utilizan en la cinta.
- **Procesamiento**
 - Deberá procesar este archivo y generar una MT con las configuraciones descritas en este archivo.
 - Interprete la lista de estados, estado inicial, estado final, alfabetos y funciones de transición para idear un algoritmo que pueda simular una cadena introducida en la cinta.
 - La simulación deberá de utilizar las funciones de transición que usted defina en el archivo yaml, ya que estas indican el estado en el que comienza la transición, el input leído sobre la cinta y el resultado, es decir, el nuevo estado, el input a reemplazar en la cinta y el desplazamiento sobre la cinta.
- **Salida**
 - Por cada cadena de input configurada en el archivo yaml:
 - Un listado de descripciones instantáneas (IDs) completo que muestre la simulación de la MT.
 - En caso de terminar la ejecución y la ID contiene un estado de aceptación (final) entonces debe indicar que la cadena se pudo procesar adecuadamente (aceptada).
 - De terminar la ejecución sin poder aplicar una función de transición que lleve a un estado de aceptación, debe indicar que el input es rechazado.

Evaluación

Deberá entregar un video alojado en la plataforma YouTube, no listado, y compartir el enlace al video en su entrega. En este video, deberá mostrar la ejecución de su programa de la siguiente forma:

- Ejecutar una Máquina de Turing de una cinta con un proceso reconocedor, es decir, el objetivo de esta MT debe ser reconocer alguna cadena que deba contener una forma en específico, por ejemplo, una MT que reconozca cadenas de la forma $\{a^n b^n \mid n \geq 1\}$.
- El lenguaje es de su elección y usted deberá proporcionar toda la información, como las funciones de transición, etc.
 - Para esta MT deberá mostrar la ejecución sobre 2 cadenas con longitud mayor o igual a 5 que sí sean aceptadas por la MT.
 - Para esta MT deberá mostrar la ejecución sobre 2 cadenas con longitud mayor o igual a 5 que no sean aceptadas por la MT.
- Ejecutar una segunda Máquina de Turing de una cinta con un proceso alterador, es decir, el objetivo de esta MT debe ser alterar el input sobre la cinta para cambiar su forma. Por ejemplo, darle la vuelta a una cadena.
- El único caso no aceptado es cambiar todos los elementos en la cinta por blanks.
- El lenguaje es de su elección y usted deberá proporcionar toda la información, como las funciones de transición, etc.
 - Para esta MT deberá mostrar la ejecución sobre 4 cadenas con longitud mayor o igual a 5.

- En su video debe de mostrar y explicar también la arquitectura de su programa, es decir, cómo diseñó e implementó la simulación de su MT.

Ponderación

| Característica | Ponderación |
|--|-------------|
| Ejecución MT reconocedora: <ul style="list-style-type: none">• Ejecución con 2 cadenas aceptadas.• Ejecución con 2 cadenas no aceptadas.• Dificultad de la MT propuesta. | 2 2 3 |
| Ejecución MT alteradora: <ul style="list-style-type: none">• Ejecución con 4 cadenas.• Dificultad de la MT propuesta. | 4 3 |
| Discusión sobre arquitectura del programa | 1 |
| Total | 15 puntos |