**Informe Máquina de Turing**

Aaron Sarmiento
Robert Burgos
Nelson Rincon
Sergio Heredia
Laura García
Miguel Thómas

Materia:
Sistemas complejos

Universidad Sergio Arboleda

Escuela de Ciencias Exactas e Ingeniería
Ciencias de la Computación e Inteligencia Artificial
2024

**Introducción**

La máquina de turing es un dispositivo hipotético definido por su creador como una máquina automática para la computación, desde su ideación ha sido una herramienta útil para el entendimiento de algoritmos y computadoras modernas pues la flexibilidad de la máquina de turing le permite en la teoría ser capaz de realizar cualquier cómputo o algoritmo. Comprender cómo funciona la máquina de Turing es parte del estudio de las ciencias de la computación modernas y otorga una perspectiva útil sobre el desarrollo de algoritmos para la resolución de un problema. Este informe presenta la realización de una máquina de Turing para la resolución de algunas operaciones matemáticas básicas (suma, resta, multiplicación, división, módulo, potencia y factorial ).

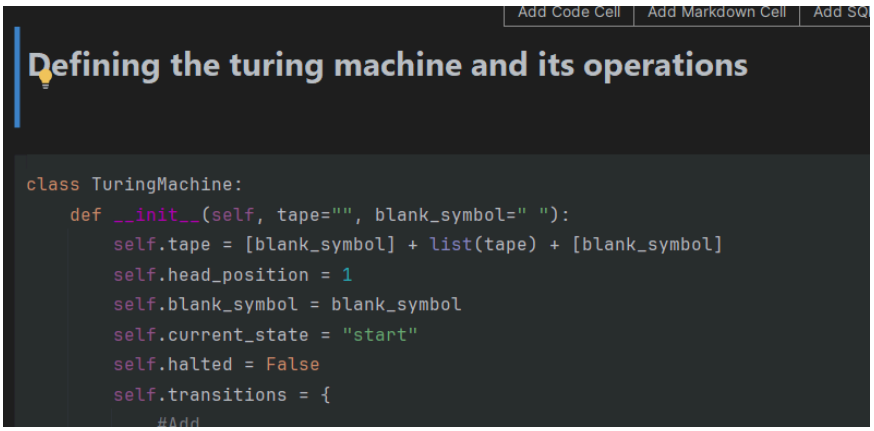
**Estructura del Código**

El código implementado para la simulación de una máquina de turing hace uso inicial de esta simplificación del concepto para proceder: La máquina de turing es un dispositivo que manipula símbolos sobre una tira de cinta infinita de acuerdo con una tabla de reglas, la manipulación sucede a través de una cabecilla que puede moverse hacia la derecha o izquierda, leer, y escribir. para esto hacemos uso de una única clase principal

- **Clase TuringMachine**

  Esta clase representa la máquina, tiene varias propiedades como

  - **Tape:** la cinta, haciendo uso de las bondades de python se usa lista de tamaño fijo que se expande a medida que es requerido.
  - **Head_position:** es la posición actual de la cabecera.
  - **Blank_symbol:** es esencialmente parte del set de reglas de la máquina, sirve de indicador de los límites superior e inferior de la cinta para la operación actual.
  - **Current_state**: almacena el estado actual de la máquina, estado según el cual accederá al set de reglas para buscar su siguiente acción.
  - **Halted:** Variable adicional pensada para reconocer si la máquina ha terminado el proceso o no.
  - **Transitions:** Es el set de reglas, estructurado a forma de diccionario donde la llave es una tupla de valor actual de la cinta y estado de la máquina y el valor devuelto es una terna de valor a escribir, dirección de movimiento y nuevo estado de la máquina.




En cuanto a los métodos propios de la clase resalta el método step, que es llamado por el método run hasta que exista confirmación de que el proceso ha terminado a través de la propiedad Halted. Este método es una traducción a código python de la definición usada de la máquina de turing: si la maquina no esta detenida busca la siguiente acción basado en el valor de la cinta en la posición actual y el estado, la siguiente acción consta de los siguientes tres componentes: valor a escribir (es posible dejar el valor que se leyó o no escribir nada) dirección del próximo movimiento y nuevo estado de la máquina.

```python
def step(self):
    if self.halted:
        raise Exception("Machine is halted")
    current_symbol = self.tape[self.head_position]
    action = self.transitions.get((self.current_state, current_symbol))
    if action:
        symbol_to_write, move_direction, next_state = action
        self.tape[self.head_position] = symbol_to_write
        if move_direction == "R":
            self.head_position += 1
        elif move_direction == "L":
            self.head_position -= 1
        if self.head_position < 0:
            self.head_position = 0
            self.tape.insert(0, self.blank_symbol)
        elif self.head_position >= len(self.tape):
            self.tape.append(self.blank_symbol)
        self.current_state = next_state
    else:
        self.halted = True
```

Con respecto a los algoritmos para la resolución de los problemas, lo que viene a ser la verdadera lógica de esta máquina de Turing se encuentra toda almacenada dentro del diccionario Transitions. Esta implementación hace uso de la lógica unitaria , donde el símbolo principal de conteo es “1” la representación decimal es igual al número de unos en la

cinta al final del proceso, es decir diez unos serían equivalentes al número 10, con esto en mente la lógica dentro del diccionario de transacciones pretende incrementar o disminuir la cantidad de unos de acuerdo a la operación que se está realizando, por ejemplo para el caso de la suma es bastante sencillo basta con eliminar el símbolo de suma y unificar la cinta de este modo “11+11” será igual a “1111” lo que en decimal es equivalente a la suma 2+2 = 4. Esta misma lógica se extiende para las demás operaciones de la siguiente manera:

```
#Subtract
("start", "-"): ("", "R", "SubtractingR"),
("SubtractingR", "1"): ("", "L", "SubtractingL"),
("SubtractingR", "") : ("", "R", "SubtractingR"),
("SubtractingL", "1"): ("", "R", "SubtractingR"),
("SubtractingL", "") : ("", "L", "SubtractingL"),
("SubtractingL", " ") : ("-", "R", "AddOne"),
("AddOne", "") : ("1", "R", "FinishedSubtracting"),
# Multiplication
("start", "*") : ("*", "L", "FindStart"),
("FindStart", "1") : ("1", "L", "FindStart"),
("FindStart", " ") : (" ", "R", "MoveStart"),
("MoveStart", "1") : (" ", "R", "Find*"),
("Find*", "1") : ("1", "R", "Find*"),
("Find*", "*") : ("*", "R", "CreateX"),
("CreateX", "1") : ("X", "R", "Prepare"),
("Prepare", "1") : ("1", "R", "Prepare"),
("Prepare", " ") : ("", "R", "Duplicate"),
("Prepare", "") : ("", "R", "Duplicate"),
("Duplicate", "1") : ("1", "R", "Duplicate"),
("Duplicate", " ") : ("1", "L", "MoveBack"),
("MoveBack", "1") : ("1", "L", "MoveBack"),
("MoveBack", "") : ("", "L", "CheckX"),
("CheckX", "1") : ("1", "L", "MoveBack"),
("CheckX", "X") : ("1", "L", "Back*"),
("MoveBack", "X") : ("X", "R", "CreateX"),
("Back*", "X") : ("1", "L", "Back*"),
("Back*", "*") : ("*", "L", "CheckStart"),
("CheckStart", "1") : ("1", "L", "FindStart"),
("CheckStart", " ") : (" ", "R", "Delete"),
("Delete", "*") : ("", "R", "Delete"),
("Delete", "1") : ("", "R", "Delete"),
("Delete", "") : ("", "R", "FinishedMultiplying"),
```

## Conclusión

Este informe ha demostrado cómo una abstracción computacional teórica puede ser realizada y manipulada dentro de un entorno de programación moderno permitiendo concluir que aunque la Máquina de Turing es una herramienta teórica poderosa y relativamente simple en su concepción, este proyecto también resalta sus limitaciones prácticas, como la eficiencia y la escalabilidad. Y, al mismo tiempo, demuestra el potencial ilimitado de la computación al sugerir que cualquier operación computable puede ser modelada si se dispone de las transiciones adecuadas.