

**Autor:** Tomasz Foryś, index:331373

**Cel i opis projektu:** Celem projektu było stworzenie działającej maszyny Turinga, składającej się z taśmy z danymi oraz głowicy działającej na niej według instrukcji.

**Podział na klasy i ich opis:**

-klasa Tape – klasa przechowująca dane na których pracuje maszyna.

-klasa Head – klasa odpowiadająca głowicy maszyny, jest w stanie się poruszać po taśmie i zmieniać swój stan.

-klasa Input – klasa odczytująca wejściowy plik tekstowy odpowiadający za początkową zawartość taśmy i głowicy.

-klasa Instructions – klasa odczytująca wejściowy plik zawierający instrukcje dla głowicy, tworzy słownik wejście : wyjście.

-klasa TextUi – klasa odpowiadająca za wyświetlanie tekstu w tekstowym interfejsie Curses.

**Pliki konfiguracyjne/Instrukcja użytkownika:**

W głównym folderze znajduje się plik konfiguracyjny „config.ini” w którym można znaleźć ścieżki do plików wejściowych i wyjściowych jak i wartość move\_cap ograniczającą długość działania maszyny.

W folderze text\_files znajdują się dwa pliki wejściowe, machine\_input.txt z początkową taśmą i głowicą oraz machine\_instructions.txt z instrukcjami dla maszyny, jak i plik wyjściowy machine\_output.txt do którego zapisywany jest wynik działania maszyny.

Instrukcje są zapisywane w formacie:

Wartość początkowa, stan początkowy, wartość wyjściowa, stan wyjściowy, kierunek przesunięcia się głowicy

Np. 0, q0, 1, q0, R lub 1, q1, 1, q0, L

Dane taśmy w machine\_input.txt są przedzielane przecinkiem i spacją, a dane głowicy są zapisywane w formacie:

Początkowa pozycja głowicy, początkowy stan głowicy

Np. 0, q0 lub 12, q3

W projekcie zapisany jest przykładowy program, który zaczyna po lewej stronie taśmy, zamienia 0 na 1 i 1 na 0 dopóki nie nadejdzie na drugą dwójkę, po czym zwraca aż do następnej dwójki, po czym znowu zwraca:

0, 1, 1, 2, 0, 0, 0, 2, 0, 0, 0, 0 -> 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1

Aby włączyć maszynę należy wejść w terminalu do folderu zawierający plik turing\_machine.py i wywołać komendę python3 turing\_machine.py. Program ma 2 dodatkowe opcje wywołania :

-i lub –instant, która od razu pokazuje końcową wersję taśmy zamiast krok po kroku

-nw lub -no\_write, która wyłącza pisanie maszyny do pliku wyjściowego.

W trybie wyświetlania krok po kroku maszyna wyświetla ostatnią zmienioną wartość na taśmie w kolorze czerwonym.

### **Część refleksyjna:**

Uznaję projekt za udaną okazję do poszerzenia wiedzy, dzięki niemu nauczyłem się używać bibliotek takich jak argparse, StringIO lub Curses które stwierdzam, że stanowiło znaczną część trudności z projektem. Po oryginalnym stworzeniu projektu, zmienie manualnego printowania na użycie interfejsu Curses wymagało całkowitej zmiany wyglądu projektu, ale poprawiło to jednak wygląd wykonywanych czynności w terminalu. Koncept maszyny Turinga jest dość prosty więc nie uważam, żebym mógł dużo do niego dodać.