

Zadanie 1A

Napisać program symulujący działanie automatu skończonego $M=(Q, S, d, q_0, F)$, gdzie $Q=\{q_0, q_1, q_2, q_3\}$, $S=\{0, 1\}$, $F=\{q_0\}$

Symulator powinien dla ciągów składających się symboli wejściowych rysować diagram przejść i zaznaczać aktualny stan przy wczytywanych kolejnych symbolach z ciągu. Po wczytaniu całego ciągu program powinien wyświetlić komunikat czy ciąg został zaakceptowany?

d	Wejścia	
Stany	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

1. Instrukcja obsługi

A. Uruchomienie programu

Aby uruchomić program należy pobrać bibliotekę pygame.

```
(base) iza@MacBook-Air-2 proj % /usr/local/bin/python3.12 -m pip
install pygame --break-system-packages
```

Po uruchomieniu programu (na przykład przez wpisanie w terminalu `python3 automat.py`) należy poczekać aż program poda komunikat o gotowości do przyjęcia ciągu wejściowego, a następnie wpisać taśmę wejściową i nacisnąć Enter.







```
(base) iza@MacBook-Air-2 desktop % python3 automat.py
pygame 2.0.3 (SDL 2.0.16, Python 3.8.8)
Hello from the pygame community. https://www.pygame.org/contribute.html
-----
Podaj ciąg wejściowy: 10110
```



Otworzy się wówczas okno z symulacją automatu.

B. Okno z symulacją

W nowym oknie pojawi się graf skierowany z czterema wierzchołkami q_0 , q_1 , q_2 i q_3 oznaczającymi cztery stany automatu skończonego. Połączone są one zgodnie z tabelą podaną w treści zadania.

Legenda

Wygląd	Cecha	Co oznacza
	Biały wierzchołek (q_1 , q_2 lub q_3)	Jeden ze stanów
	Niebieski wierzchołek (q_0)	Stan początkowy
	Wierzchołek z obwódką (q_0)	Stan końcowy (akceptujący)
	Pomarańczowy wierzchołek (q_0 , q_1 , q_2 lub q_3)	Aktualny stan
	Zielony wierzchołek (q_0)	Ostatni wierzchołek, jeżeli ciąg został zaakceptowany
	Czerwony wierzchołek (q_0 , q_1 , q_2 lub q_3)	Ostatni wierzchołek, jeżeli ciąg nie został zaakceptowany

Wygląd	Cecha	Co oznacza
	Biała strzałka (wartość 1 lub 0)	Możliwe przejście pomiędzy stanami
	Fioletowa strzałka (wartość 1 lub 0)	Aktualne przejście pomiędzy stanami

C. Wynik programu

Po wykonaniu wszystkich przejść program wypisze w terminalu kolejność przejść pomiędzy stanami.

Program zamknie okno automatycznie po ok. 6 sekundach od dojścia do ostatniego wierzchołka. Możliwe jest również zamknięcie okna przez kliknięcie znaku X w rogu.

Po zamknięciu okna program napisze w terminalu, czy ciąg został zaakceptowany czy nie.

2. Przykładowe zestawy danych

A. Przypadek akceptujący wejście

Ciągiem zaakceptowanym jest taki ciąg, który prowadzi do stanu należącego do stanów akceptujących, czyli w naszym przypadku do q_0 ($F=\{q_0\}$).

Przykładem zaakceptowanego ciągu jest 10100101.

Kolejność odwiedzanych wierzchołków:

```
q0 -> q1 -> q3 -> q2 -> q0 -> q2 -> q3 -> q1 -> q0
```

Ciąg ten wymusza przejście pomiędzy każdymi dwoma wierzchołkami pomiędzy którymi bezpośrednie przejście istnieje.

Działanie symulacji dla tego przykładu znajduje się w pliku przykład1.mov.

B. Przypadek nieakceptujący wejścia

Ciąg niezaakceptowany to taki, który nie prowadzi do stanu końcowego q_0 .

Przykładem takiego ciągu jest 1101.

Kolejność odwiedzanych wierzchołków:

```
q0 -> q1 -> q0 -> q2 -> q3
```

Po zakończeniu ciągu wejściowego, symulacja pokaże komunikat "Ciąg nie został zaakceptowany", ponieważ ostatni stan (q_3) nie jest stanem akceptującym.

Działanie symulacji dla tego przykładu znajduje się w pliku przykład2.mov.

3. Źródła

- pygame.org
- wikipedia.org
- wiedza własna
- 101computing.net