

Compilers 2021/2022

Projekt 1: wyrażenia regularne

Piotr Plebański
121309

12 września 2022

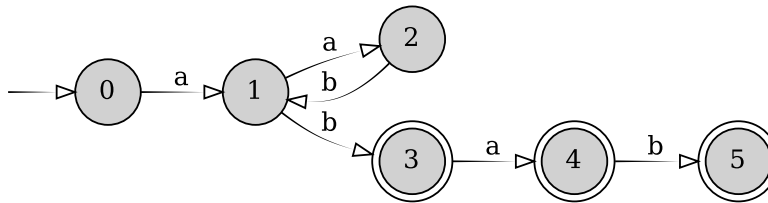
1 Wstęp

Owoce tego krótkiego projektu jest program uruchamiany z konsoli, który sprawdza, czy tekst pasuje do wyrażenia regularnego:

$$a(ab)^*b(ab|a)?$$

2 Automat skończony

Automat skończony, który generuje język zadany powyższym wyrażeniem regularnym, przedstawia rys.1.



Rysunek 1: Deterministyczny automat skończony dla wyrażenia $a(ab)^*b(ab|a)?$.

3 Tabela przejść

Dozwolone przejścia pomiędzy stanami dla danych stanów i symboli wejściowych znajdują się w tab. 1. Ostatni wiersz tabeli informuje o tym, czy stan jest akceptujący.

znak	stan					
	0	1	2	3	4	5
a	1	2		4		
b		3	1		5	
akcept.?	N	N	N	A	A	A

Tabela 1: Dozwolone przejścia dla automatu.

4 Opis projektu

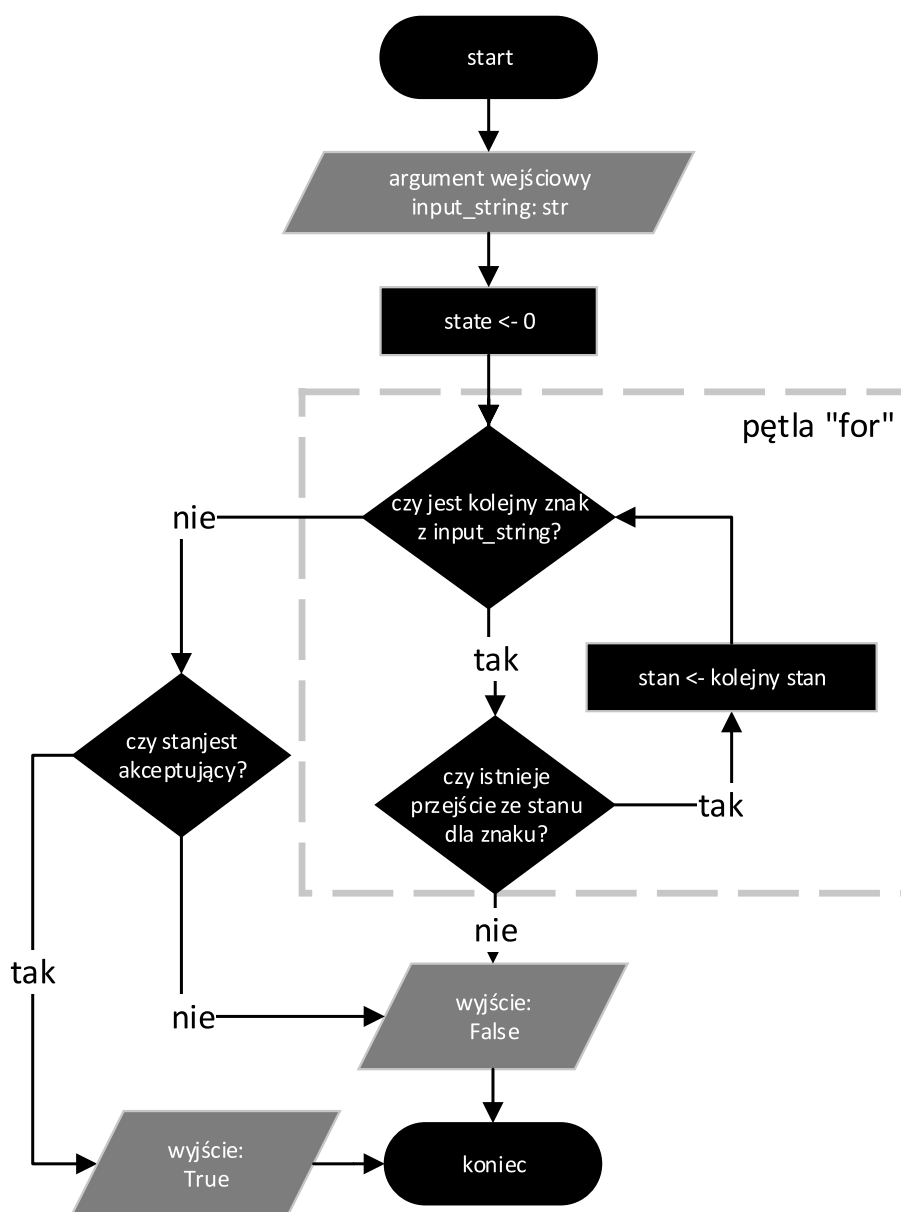
Projekt został zrealizowany w języku Python 3.8 używając jedynie pakietów z biblioteki standardowej. Projekt składa się z następujących plików:

1. compilers1_regex.py – główny plik z funkcją walidującą tekst podany na wejściu,
2. compilers1_regex.pdf – niniejsza dokumentacja,
3. README.md – krótki opis projektu i przepis na jego uruchomienie,
4. tests.py – testy jednostkowe.

5 Algorytm (plik: compilers1_regex.py)

1. Inicjalizacja słownika dozwolonych przejść (odpowiadających tablicy 1) i listy stanów końcowych.
2. Wczytanie tekstu ze standardowego wejścia.
3. Iteracyjne przechodzenie przez stany - funkcja `validate_string(input_string: str) -> bool`.
4. Wypisanie wyniku – prawda/fałsz.

Schemat działania funkcji `validate_string(input_string: str) -> bool` przedstawia rys. 2.



Rysunek 2: Algorytm sprawdzania, czy tekst podany na wejściu jest opisywany przez wyrażenie $a(ab)^*b(ab|a)^?$.