

Limbaje Formale și Automate

Tema 1

Parser șiruri de caractere

Termen de predare: 02.11.2020

Responsabili temă:
Alexandru Poboranu, Matei Popovici

Data publicării: 14.10.2020
Ultima actualizare: 14.10.2020

1. Specificații temă

1.1 Obiective

Aplicarea cunoștințelor teoretice dobândite la cursul de “Limbaje Formale și Automate” prin realizarea unui analizor de șiruri de caractere.

1.2 Cerință

Să se implementeze un parser de string-uri folosindu-se metoda Boyer-Moore. Se va da un **string1** pe baza căruia se va genera tabelul de parsare (matricea delta). Pe baza acestuia se va realiza parsarea **string2** pentru a se calcula decalajul (offset-ul).

1.3 Restricții

- caracterele folosite sunt literele mari ale alfabetului englez (de la A la Z)
 - nu aveți voie să folosiți funcții deja existente de căutare a unui substring într-un string.
- Nerespectarea acestei restricții va duce la pierderea întregului punctaj de pe temă.**

- **string1** < 40
- **string2** < 15 000 000
- fișierul principal din care se apelează funcția main se va numi “main.py” și va primi:
primul argument fișierul de intrare și al doilea argument fișierul de ieșire
Ex: python main.py ./input/test0.in ./output/test0.out

1.4 Date de intrare:

- Pe prima linie se află **string1** pe baza căruia urmează să se realizeze funcția delta
- Pe a doua linie se află **string2** în care trebuie să se caute string1 cu ajutorul funcției delta

1.5 Date de ieșire:

- Pe prima linie se vor scrie valorile de decalaj (offset) la care se găsește string1 în string2 separate de câte un spațiu

2. Exemple

2.1 Exemplul 1

Input:

```
string1 = LFA  
string2 = LFAZELFALFARLFA
```

Output:

```
0 5 8 12
```

Delta:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
e	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LF	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LFA	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Șirul vid = starea 0

L = starea 1

LF = starea 2

LFA = starea 3 (stare finală)

Explicații:

Poziția	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Litera	L	F	A	Z	E	L	F	A	L	F	A	R	L	F	A
Starea în care se trece	1	2	3	0	0	1	2	3	1	2	3	0	1	2	3
Starea finală			S-a găsit o potrivire la poziția 0					S-a găsit o potrivire la poziția 5			S-a găsit o potrivire la poziția 8				S-a găsit o potrivire la poziția 12

Se începe din starea 0

-> se găsește L la poziția 0, se trece în starea 1

-> se găsește F la poziția 1, se trece în starea 2

-> se găsește A la poziția 2, se trece în starea 3 (este stare finală => s-a găsit o potrivire la poziția 0)

-> se găsește Z la poziția 3, se trece în starea 0

-> se găsește E la poziția 4, se rămâne în starea 0

-> se găsește L la poziția 5, se trece în starea 1

-> se găsește F la poziția 6, se trece în starea 2

-> se găsește A la poziția 7, se trece în starea 3 (este stare finală => s-a găsit o potrivire la poziția 5)

-> se găsește L la poziția 8, se trece în starea 1

-> se găsește F la poziția 9, se trece în starea 2

-> se găsește A la poziția 10, se trece în starea 3 (este stare finală => s-a găsit o potrivire la poziția 8)

-> se găsește R la poziția 11, se trece în starea 0

-> se găsește L la poziția 12, se trece în starea 1

-> se găsește F la poziția 13, se trece în starea 2

-> se găsește A la poziția 14, se trece în starea 3 (este stare finală => s-a găsit o potrivire la poziția 12)

2.2 Exemplul 2

Input:

String1 = EZEZ

String2 = EZEZEZAFEEZEZZGG

Output:

0 2 9

Delta:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
e	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
EZ	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EZE	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
EZEZ	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Șirul vid = starea 0

E = starea 1

EZ = starea 2

EZE = starea 3

EZEZ = starea 4 (stare finală)

Explicații:

Poziția	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Litera	E	Z	E	Z	E	Z	A	F	E	E	Z	E	Z	Z	G	G
Starea în care se trece	1	2	3	4	3	4	0	0	1	1	2	3	4	0	0	0
Starea finală				S-a găsit o potrivire la poziția 0		S-a găsit o potrivire la poziția 2							S-a găsit o potrivire la poziția 9			

Se începe din starea 0

-> se găsește E la poziția 0, se trece în starea 1

-> se găsește Z la poziția 1, se trece în starea 2

-> se găsește E la poziția 2, se trece în starea 3

-> se găsește Z la poziția 3, se trece în starea 4 (este stare finală => s-a găsit o potrivire la poziția 0)

-> se găsește E la poziția 4, se trece în starea 3

-> se găsește Z la poziția 5, se trece în starea 4 (este stare finală => s-a găsit o potrivire la poziția 2)

-> se găsește A la poziția 6, se trece în starea 0

-> se găsește F la poziția 7, se rămâne în starea 0

-> se găsește E la poziția 8, se trece în starea 1
-> se găsește E la poziția 9, se rămâne în starea 1
-> se găsește Z la poziția 10, se trece în starea 2
-> se găsește E la poziția 11, se trece în starea 3
-> se găsește Z la poziția 12, se trece în starea 4 (este stare finală => s-a găsit o potrivire la poziția 9)
-> se găsește Z la poziția 13, se trece în starea 0
-> se găsește G la poziția 14, se rămâne în starea 0
-> se găsește G la poziția 15, se rămâne în starea 0

3. Testare

Pentru testare aveți nevoie de python3.

Pentru a rula toate testele:

python checker.py

Pentru a rula un singur test:

python checker.py test_no (unde test_no este o valoare de la 0 la 14 inclusiv)

Important! Testele automate sunt orientative, ele au rolul de a vă ghida în dezvoltarea temei și de a valida funcționalitatea. Sarcina voastră este însă *implementarea cerinței*. În urma verificării manuale, este posibil să primiți un punctaj final mai mic decât cel din teste, în cazurile în care nu ați respectat cerința.

Exemple extreme: hardcodarea rezultatelor pentru fiecare test; utilizarea unei biblioteci de parsare din python pentru rezolvarea cerinței.

4. Punctaj

Tema valorează în total **0.5 puncte** din nota finală, dintre care:

- 0.45 teste automate
- 0.05 README

În general, punctajul final ar trebui să corespundă cu punctajul primit din rularea automată a testelor. Pot exista depunctări suplimentare dacă implementarea voastră nu respectă cerința, dacă nu aveți un README clar, dacă aveți un cod foarte greu de citit, etc.

5. Trimitere

Tema va trebui încărcată pe moodle.

Trebuie să încărcăți o arhivă zip care să conțină:

- fișierul main.py care va fi invocat direct de checker
- README în care să explicați abordarea temei
- orice alte fișiere de care v-ați ajutat în dezvoltarea temei (surse de python, etc.)