Zadanie č. 2

Vo zvolenom (podľa seba) programovacom jazyku naprogramujte program, ktorý realizuje Thompsonovu konštrukciu a následne výpočet výsledného konečného automatu nad reťazcami. Váš program bude pracovať nasledovným spôsobom:

1. Prvým argumentom programu pri spustení bude textový súbor, v ktorom sú uložené elementárne regulárne výrazy a operácie medzi nimi. Tieto spolu tvoria nejaký výsledný regulárny výraz. Váš program by mal postupne konštruovať NKA podľa Thompsonovej konštrukcie (AFJ 3. týždeň prednáška/cvičenie, prípadne `https://en.wikipedia.org/wiki/Thompson%27s_construction`) podľa uvedených regulárnych výrazov a operácií medzi nimi. Formát súboru je popísaný na tretej strane zadania.

2. Druhým argumentom programu bude textový súbor, v ktorom sú zapísané reťazce, každý na novom riadku. Počet týchto reťazcov je uvedený v prvom riadku súboru. Váš program postupne pre každý reťazec pomocou výsledného konečného automatu zistí, či patrí/nepatrí do jazyka daného regulárnym výrazom z prvého súboru a vypíše na obrazovku príslušný reťazec a to, či patrí/nepatrí do daného jazyka.

Vstupy a výstupy

Elementárne regulárne výrazy a príslušné operácie sa načítajú z textového súboru predstavujúceho **prvý argument** programu pri spustení.

Druhým argumentom programu pri spustení je textový súbor obsahujúci reťazce, ktorých príslušnosť do jazyka testujeme, pričom na prvom riadku súboru je uvedený počet reťazcov a na ďalších riadkoch sú jednotlivé reťazce, každý na samostatnom riadku. Prázdny reťazec ε je reprezentovaný prázdnym riadkom. V neprázdnych reťazcoch sú použité len znaky *a,...,z,0,...,9*.

Ak by ste teda napríklad naprogramovali Váš program v jazyku Python a Váš kód by bol v súbore `zadanie2.py`, program by sa spúšťal:

```
python zadanie2.py regex.txt retazce.txt
```

kde `regex.txt` je príklad mena súboru, v ktorom sa nachádzajú elementárne regulárne výrazy s operáciami a `retazce.txt` je príklad mena súboru, v ktorom sa nachádzajú reťazce, ktoré chceme otestovať.

Zadanie môžete vypracovať v ľubovoľnom programovacom jazyku, Python je tu uvedený len ako príklad.

Implementácia algoritmov

Požaduje sa vaša samostatná práca, t.j. výsledný program musí byť výsledkom vašej samostatnej práce. Kopírovanie zdrojových kódov z internetu, prípadne od iných študentov bude hodnotené ako plagiátorstvo a v zmysle platného študijného poriadku hodnotené známkou FX. Taktiež použité algoritmy a štruktúry musia byť výsledkom vašej vlastnej implementácie, t.j. VY si musíte naprogramovať vlastnú reprezentáciu konečného automatu, postupné zreťazenie, zjednotenie a iteráciu konečných automatov a výpočet KA nad zadanými reťazcami.

To, či budete robiť akceptáciu reťazcov pre výsledný NKA z Thompsonovej konštrukcie, alebo sa rozhodnete pre jeho determinizáciu a výpočet budete robiť na DKA, nechávam na Vás.

Deadline zadania

Zdrojové kódy Vášho riešenia odovzdajte do AIS-u do príslušného miesta odovzdania. Deadline je **27. marec 2024, 23:59:59.**

Na ďalšej strane nájdete popis formátu prvého vstupného súboru.

FORMÁT PRVÉHO VSTUPNÉHO SÚBORU

Riadky sú implicitne číslované od 1, t.j. prvý riadok v súbore má číslo 1, druhý 2, atď. Na každom riadku môže byť:

- Prázdny riadok - reprezentuje regulárny výraz popisujúci prázdny reťazec ε
- Jeden symbol - reprezentuje regulárny výraz popisujúci príslušný symbol. Môžete predpokladať, že týmto symbolom je vždy alebo malé písmeno anglickej abecedy a-z, alebo číslica 0-9.
- Trojica U, i, j, kde U je písmeno veľké U (zo slova Union), i, j sú poradové čísla niektorých predchádzajúcich riadkov. Táto trojica predstavuje regulárny výraz, ktorý popisuje zjednotenie jazykov regulárnych výrazov z riadkov i a j
- Trojica C, i, j, kde C je písmeno veľké C (zo slova Concatenation), i, j sú poradové čísla niektorých predchádzajúcich riadkov. Táto trojica predstavuje regulárny výraz, ktorý popisuje zreťazenie jazykov regulárnych výrazov z riadkov i a j v poradí ij.
- Dvojica I, i, kde I je písmeno veľké I (zo slova Iteration), i je poradové čísla niektorého predchádzajúceho riadku. Táto dvojica predstavuje regulárny výraz, ktorý popisuje iteráciu jazyka regulárneho výrazu z riadku i
- Vstup bude vždy korektný, t.j. operácie I, C, U sa neodkazujú na neexistujúce riadky alebo na riadky s väčším poradovým číslom.

Príklad č. 1: Prvým argumentom by bol súbor `regex1.txt` s obsahom:

```
a
a
I,1
C,2,3
```

Druhým argumentom by bol súbor `retazce1.txt` s obsahom:

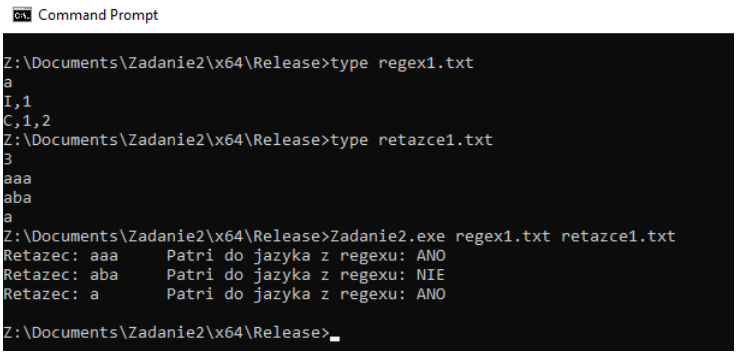
```
3
aaa
aba
a
```

K jednotlivým riadkom v prvom argumente teda prináležia nasledovné regulárne výrazy:

1. a
2. a
3. a^*
4. aa^*

Výsledný regulárny výraz (ten z posledného riadka) prvého súboru predstavuje regex aa^*, voči ktorému budeme porovnávať reťazce z druhého vstupného súboru. Následne sa z druhého vstupného súboru načítajú jednotlivé reťazce a program vypíše, či patria alebo nepatria do jazyka daného regulárnym výrazom z prvého súboru.

Príklad činnosti programu (s formátovaním výstupu sa nemusíte trápiť, dôležité je, aby bol vždy uvedený reťazec a to, či patrí/nepatrí do jazyka)):



Príklad č. 2: Prvým argumentom by bol súbor `regex2.txt` s nasledovným obsahom (prvý riadok je prázdny riadok, t.j. je tam regulárny výraz ε):

```
a
b
C,2,3
U,1,4
```

Druhým argumentom by bol súbor `retazce2.txt` s nasledovným obsahom (druhý reťazec je prázdny reťazec ε reprezentovaný prázdnym riadkom):

```
3
aaa

ab
```

K jednotlivým riadkom v prvom argumente teda prináležia nasledovné regulárne výrazy:

1. ε
2. a
3. b
4. ab
5. $\varepsilon \mid (ab)$

Výstup programu (ak sa testuje prázdny reťazec, uveďte to napríklad tak, ako na obrázku):

```
Z:\Documents\Zadanie2\x64\Release>type regex2.txt

a
b
C,2,3
U,1,4
Z:\Documents\Zadanie2\x64\Release>type retazce2.txt
3
aaa

ab
Z:\Documents\Zadanie2\x64\Release>Zadanie2.exe regex2.txt retazce2.txt
Retazec: aaa            Patri do jazyka z regexu: NIE
Retazec: (prazdny)      Patri do jazyka z regexu: ANO
Retazec: ab             Patri do jazyka z regexu: ANO

Z:\Documents\Zadanie2\x64\Release>
```

Príklad č. 3: Prvým argumentom by bol súbor `regex3.txt` s nasledovným obsahom:

```
a
b
U,1,2
I,3
```

Druhým argumentom by bol súbor `retazce3.txt` s nasledovným obsahom (prvý reťazec je prázdny reťazec ε reprezentovaný prázdnym riadkom):

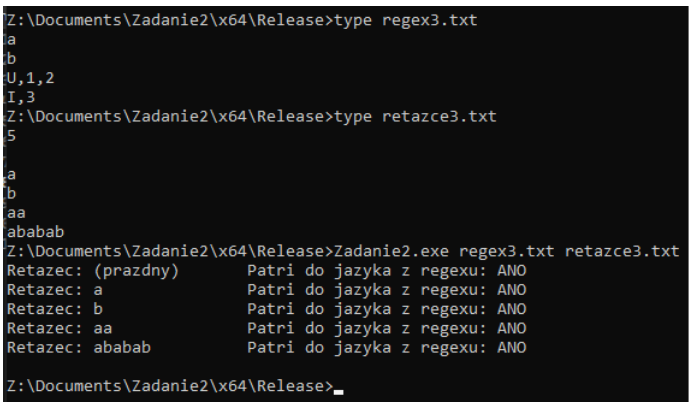
```
5

a
b
aa
ababab
```

K jednotlivým riadkom v prvom argumente teda prináležia nasledovné regulárne výrazy:

1. a
2. b
3. $a \mid b$
4. $(a \mid b)^*$

Výsledný regulárny výraz: $(a \mid b)^*$. Výstup by vyzeral napríklad nasledovne:



Príklad č. 4: Prvým argumentom by bol súbor `regex4.txt` s nasledovným obsahom (tretí riadok je prázdny riadok, t.j. je tam regex ε):

```
a
b

C,2,1
U,4,3
I,5
c
C,6,7
```

Druhým argumentom by bol súbor `retazce4.txt` s nasledovným obsahom:

```
3
c
babac
bacba
```

K jednotlivým riadkom v prvom argumente teda prináležia nasledovné regulárne výrazy:

1. a
2. b
3. ε
4. ba
5. $\varepsilon \mid ba$
6. $(\varepsilon \mid ba)^*$
7. c
8. $(\varepsilon \mid ba)^*c$

Výsledný regulárny výraz: $(\varepsilon \mid ba)^*c$. Program by pre reťazce `c,babac` vypísal, že patria do jazyka daného regexom, pre reťazec `bacba` že nepatrí do jazyka daného regexom.

Príklad č. 5: Prvým argumentom by bol súbor `regex5.txt` s nasledovným obsahom (šiesty riadok je prázdny, t.j. je tam regex ε):

```
a
c
b
C,1,2
C,4,3

I,5
U,7,6
a
I,9
b
C,10,11
C,8,12
```

Druhým argumentom by bol súbor `retazce5.txt` s nasledovným obsahom:

```
3
acbacbb
acbaaab
acbaabb
```

K jednotlivým riadkom v prvom argumente teda prináležia nasledovné regulárne výrazy:

1. a
2. c
3. b
4. ac
5. acb
6. ε
7. $(acb)^*$
8. $(acb)^* \mid \varepsilon$
9. a
10. a^*

11. b

12. a^*b

13. $((acb)^* \mid \varepsilon)a^*b$

Výsledný regulárny výraz $((acb)^* \mid \varepsilon)a^*b$. Pre reťazce `acbacbb, acbaaab` by program vypísal, že patria do jazyka daného regexom, pre reťazec `acbaabb` by program vypísal, že nepatrí do jazyka daného regexom.

Changelog:
7.3.2024 Zverejnená prvá verzia zadania.

11.3.2024 Klarifikácia zadania, kde je explicitne napísané, že máte naprogramovať Thompsonovu konštrukciu, t.j. aj s konštrukciou NKA a s prípadnou determinizáciou na DKA. Taktiež bol predĺžený deadline zadania o 3 dni z pôvodného 24.3. 23:59 na 27.3. 23:59.

18.3.2024 Upravený prvý vzorový vstup, aby tam neboli 2 inštrukcie pracujúce s tým istým predchádzajúcim riadkom.