

### Zadanie č. 3

Vo zvolenom (podľa seba) programovacím jazyku zostavte program, ktorý:

1. Dokáže vytvoriť elementárne nedeterministické konečné automaty (NKA), t.j. také, ktoré akceptujú alebo prázdny reťazec  $\varepsilon$ , alebo jeden symbol nejakej abecedy, napr.  $a$  (prázdnu množinu  $\emptyset$  ignorujte).
2. Dokáže z existujúcich NKA vytvárať nové NKA tým, že na existujúcich NKA vykoná ich zjednotenie, zretazenie, iteráciu.
3. Následne prevedie výsledný NKA na deterministický konečný automat (ďalej len DKA) a používateľ zadá nejaký reťazec nad vstupnou abecedou automatu, pre ktorý sa nad výsledným DKA uskutoční výpočet, o ktorého výsledku je používateľ informovaný (t.j. či zadaný reťazec DKA akceptuje alebo neakceptuje) (na tento bod môžete použiť Váš program vypracovaný v rámci 2. zadania)

#### *Vstupy a výstupy*

Základné NKA a príslušné operácie sa načítajú zo vstupného súboru predstavujúceho prvý argument programu.

Postupne sa na základe inštrukcií vo vstupnom súbore vytvorí nejaký výsledný NKA. Ten sa zapíše do súboru, ktorý predstavuje druhý argument programu - formát súboru nech je taký, ako bol napr. pre NKA v zadaní č. 2.

DKA ekvivalentný poslednému vytvorenému NKA sa zapíše do súboru, ktorý predstavuje tretí argument programu pri jeho spúšťaní.

Ak by sa teda Váš program volal **zadanie3.exe**, program by sa spúšťal:

**zadanie3.exe NKA.txt NKA\_vysledny.txt DKA.txt**

kde NKA.txt je meno súboru, v ktorom sa nachádzajú NKA s operáciami, NKA\_vysledny.txt je súbor, kam sa zapíše výsledný zostrojený NKA, ktorý sa ide determinizovať a DKA.txt je meno súboru, do ktorého sa uloží DKA ekvivalentný poslednému zostrojenému NKA.

Program po spustení vyzve používateľa aby zadal nejaký reťazec nad vstupnou abecedou výsledného automatu a zistí, či DKA reťazec akceptuje alebo nie - informáciu o tom vypíše na obrazovku.

## ***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 známku FX. Taktiež použité algoritmy musia byť výsledkom vašej vlastnej implementácie, t.j. napr. na konverziu NKA na DKA nepoužívajte externé knižnice, ale vlastný kód.

## ***Deadline zadania***

Zadanie odovzdajte do AIS-u do príslušného miesta odovzdania - odovzdajte zdrojový kód. Deadline je 14. apríl 2019, 23:59:59.

## **FORMÁT VSTUPNÉHO SÚBORU S OPERÁCIAMI A NKA**

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 NKA akceptujúci prázdny reťazec
- Jeden symbol - reprezentuje NKA akceptujúci príslušný symbol
- Trojica  $U, i, j$ , kde  $U$  je písmeno veľké U (zo slova Union),  $i, j$  sú poradové čísla niektorých predchádzajúcich riadkov. Táto trojica predstavuje NKA, ktorý akceptuje zjednotenie jazykov automatov 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 NKA, ktorý akceptuje zrefazenie jazykov automatov z riadkov  $i$  a  $j$  v poradí  $ij$ .
- Dvojica  $I, i$ , kde  $I$  je písmeno veľké I (zo slova Iteration),  $i$  je poradové číslo niektorého predchádzajúceho riadku. Táto dvojica predstavuje NKA, ktorý akceptuje iteráciu jazyka automatu z riadku  $i$
- Môžete predpokladať, že vstup je korektný, t.j. operácie  $I, C, U$  sa neodkazujú na neexistujúce riadky alebo na riadky s väčším poradovým číslom. Taktiež môžete predpokladať, že ak je na riadku jeden symbol, tak tento symbol je vždy jeden ASCII znak.

Príklad č. 1: Postupnosť NKA by bola daná vo vstupnom súbore:

a  
I,1  
C,1,2

Postupne sa teda vytvoria NKA, ktoré akceptujú jazyky:

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

Výsledný NKA (ten z posledného riadka) predstavuje NKA pre regulárny výraz  $aa^*$ . Následne sa vytvorí DKA k nemu ekvivalentný a program vyzve používateľa, aby zadal nejaký reťazec: Napríklad pre reťazec  $aaaa$  by program vypísal, že príslušný DKA takéto slovo **AKCEPTUJE**.

Príklad č. 2: Postupnosť NKA by bola daná vo vstupnom súbore (prvý riadok je prázdny riadok):

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

Postupne sa teda vytvoria NKA, ktoré akceptujú jazyky:

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

Výsledný NKA (ten z posledného riadka) predstavuje NKA pre regulárny výraz  $\varepsilon \mid ab$ . Následne sa vytvorí DKA k nemu ekvivalentný a program vyzve používateľa, aby zadal nejaký reťazec: Napríklad pre reťazec  $ab$  by program vypísal, že príslušný DKA takéto slovo **AKCEPTUJE**. Pre reťazec  $aa$  by program vypísal, že príslušný DKA takéto slovo **NEAKCEPTUJE**.

Príklad č. 3: Postupnosť NKA by bola daná vo vstupnom súbore:

a  
b  
U,1,2  
I,3

Postupne sa teda vytvoria NKA, ktoré akceptujú jazyky:

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

Výsledný NKA (ten z posledného riadka) predstavuje NKA pre regulárny výraz  $(a \mid b)^*$ . Následne sa vytvorí DKA k nemu ekvivalentný a program vyzve používateľa, aby zadal nejaký reťazec: Napríklad pre reťazec  $ab$  by program vypísal, že príslušný DKA takéto slovo **AKCEPTUJE**.

Príklad č. 4: Postupnosť NKA by bola daná vo vstupnom súbore (tretí riadok je prázdny):

a

b

C,2,1

U,3,4

I,5

c

C,6,7

Postupne sa teda vytvoria NKA, ktoré akceptujú jazyky:

1.  $a$

2.  $b$

3.  $\varepsilon$

4.  $ba$

5.  $\varepsilon \mid ba$

6.  $(\varepsilon \mid ba)^*$

7.  $c$

8.  $(\varepsilon \mid ba)^*c$

Výsledný NKA (ten z posledného riadka) predstavuje NKA pre regulárny výraz  $(\varepsilon \mid ba)^*c$ . Následne sa vytvorí DKA k nemu ekvivalentný a program vyzve používateľa, aby zadal nejaký reťazec: Napríklad pre reťazec  $bac$  by program vypísal, že príslušný DKA takéto slovo **AKCEPTUJE**. Reťazec  $ba$  by takýto DKA **NEAKCEPTOVAL**.

Príklad č. 5: Postupnosť NKA by bola daná vo vstupnom súbore (šiesty riadok je prázdny):

a  
c  
b  
C,1,2  
C,4,3  
  
I,5  
U,7,6  
a  
I,9  
b  
C,10,11  
C,8,12

Postupne sa teda vytvoria NKA, ktoré akceptujú jazyky:

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

Výsledný NKA (ten z posledného riadka) predstavuje NKA pre regulárny výraz  $((acb)^* \mid \varepsilon)a^*b$ . Následne sa vytvorí DKA k nemu ekvivalentný a program vyzve používateľa, aby zadal nejaký reťazec: Napríklad pre reťazec *acbaaab* by program vypísal, že príslušný DKA takéto slovo **AKCEPTUJE**. Reťazec *acbaaabbb* by takýto DKA **NEAKCEPTOVAL**.