

# Funkcionální projekt

## PLG-2-NKA

(Funkcionální a logické programování 2019/2020)

## 1 Úvod

Tento dokument popisuje požadavky na projekt **PLG-2-NKA** v předmětu FLP v akademickém roce **2019/2020**. Obecné pokyny pro všechna zadání jsou sepsány ve zvláštním dokumentu, který si nezapomeňte důkladně pročíst.

## 2 Zadání

Vytvořte program, který převádí pravé lineární gramatiky (PLG) bez jednoduchých pravidel (tj. pravidel  $A \rightarrow B$ ) na nedeterministické konečné automaty (NKA).

### 2.1 Rozhraní programu

Program se bude jmenovat **plg-2-nka** a bude jej možné spustit takto:

```
plg-2-nka volby [vstup]
```

kde

- *vstup* je jméno vstupního souboru (pokud není specifikováno, program čte standardní vstup) obsahujícího PLG ve formátu popsaném v sekci 3.
- *volby* jsou parametry ovlivňující chování programu, standardně tyto:
  - i vypíše se načtená a do vaší vnitřní reprezentace převedená PLG na stdout ve formátu popsaném v sekci 3. Nevypisujte jen načtený řetězec, tato volba ověřuje, že váš program dokáže gramatiku přecíst, převést a znovu vypsát.
  - 1 vypíše se gramatika, která je výsledkem věty 3.2 z opory předmětu TIN, na stdout ve formátu popsaném v sekci 3, s výjimkou pojmenování nových neterminálů. Ty mohou být označeny jménem skládajícím se z velkého písmene následovaného jednou nebo více číslicemi.
  - 2 vypíše se NKA přijímající stejný jazyk, jaký je generován PLG na vstupu; na stdout se vypíše ve formátu popsaném v sekci 4. Postupujte podle věty 3.6 z opory předmětu TIN. Stavů automatu označujte nezápornými celými čísly (0, 1, 2, ...).

### 3 Formát vstupu

PLG  $G = (N, \Sigma, P, S)$  na vstupu odpovídá standardní definici pravé lineární gramatiky (viz oporu předmětu TIN). Pro zjednodušení je ale abeceda  $N$  podmnožinou množiny velkých písmen  $[A-Z]$ , a abeceda  $\Sigma$  je podmnožinou množiny malých písmen  $[a-z]$ . Prázdné slovo ( $\varepsilon$ ) je reprezentováno znakem  $\#$ . Vstupní textová reprezentace PLG o  $n$  pravidlech má následující formát:

```
<seznam všech neterminálů>\n
<seznam všech terminálů>\n
<počáteční neterminál>\n
<pravidlo 1>\n
...
<pravidlo n>\n
```

Symbolsy v seznamech jsou odděleny čárkou. Například, reprezentace PLG z příkladu 2.23 opory předmětu TIN je následující:

```
A,B
a,b,c
A
A->aaB
A->ccB
B->bB
B->#
```

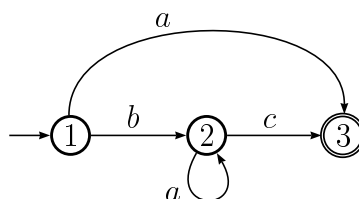
### 4 Formát výstupu

Výstupní textová reprezentace NKA o  $n$  pravidlech má následující formát:

```
<seznam všech stavů>\n
<počáteční stav>\n
<seznam koncových stavů>\n
<pravidlo 1>\n
...
<pravidlo n>\n
```

Symbolsy v seznamech jsou odděleny čárkou. Pravidlo pro přechod ze stavu 1 do stavu 2 pomocí symbolu  $x$  má tvar  $1, x, 2$ .

Například, NKA:



bude na výstupu jako:

1,2,3

1

3

1,a,3

1,b,2

2,a,2

2,c,3