

# proj1

## Zadanie

---

Naimplementujte program v jazyku swift, ktorý pre zadaný vstupný reťazec a deterministický konečný automat odsimuluje prechod stavmi nutnými pre prijatie reťazca. Pokiaľ reťazec je akceptovaný končným automatom, vypíše sa na stdout postupnosť stavov. Pokiaľ reťazec nie je akceptovaný tak program končí chybou.

## Spustenie programu

Program je možné spustiť pomocou príkazu

`swift run proj1 <vstupny_retazec> <nazov_suboru>` v domovskom priečinku programu (priečinok obsahujúci súbor *Package.swift*).

- `<vstupny_retazec>` - obsahuje jednotlivé symboly oddelené `,` (čiarkou), napr:
  - `a,b,c` obsahuje tri symboly,
  - `Prvy,Druhy,Treti,+1š_,Symbol s medzerou` obsahuje päť symbolov.
- `<nazov_suboru>` - cesta k súboru obsahujúceho reprezentáciu konečného automatu uloženú vo formáte JSON.

## JSON atribúty

- `states` - pole reťazcov obsahujúce jednotlivé stavy,
- `symbols` - pole reťazcov obsahujúce jednotlivé symboly,
- `transitions` - pole prechodov, kde každý prechod ( $pa \rightarrow q$ ) obsahuje atribúty:
  - `from` - aktualny stav  $q$ ,
  - `with` - aktualny symbol  $a$ ,
  - `to` - nový stav  $q$ ,
- `initialState` - počiatočný stav,
- `finalStates` - pole koncových stavov.

## Konečný automat č. 1

- Jazyk automatu:  $L = a^*$

```
1  {  
2    "states" : [  
3      "A"  
4    ],  
5    "symbols" : [  
6      "a"  
7    ],  
8    "transitions" : [  
9      {  
10       "with" : "a",  
11       "to" : "A",  
12       "from" : "A"  
13     }  
14   ],  
15   "initialState" : "A",  
16   "finalStates" : [  
17     "A"  
18   ]  
19 }
```

## Konečný automat č. 2

- Jazyk automatu:  $L = (ab)^+c^+$

```
1  {
2    "states" : [
3      "S",
4      "AB",
5      "C"
6    ],
7    "symbols" : [
8      "a",
9      "b",
10     "c"
11   ],
12   "transitions" : [
13     {
14       "with" : "a",
15       "to" : "AB",
16       "from" : "S"
17     },
18     {
19       "with" : "b",
20       "to" : "AB",
21       "from" : "S"
22     }, {
23       "with" : "a",
24       "to" : "AB",
25       "from" : "AB"
26     },
27     {
28       "with" : "b",
29       "to" : "AB",
30       "from" : "AB"
31     },
32     {
33       "with" : "c",
34       "to" : "C",
35       "from" : "AB"
36     },
37     {
38       "with" : "c",
39       "to" : "C",
40       "from" : "C"
41     }
42   ],
43   "initialState" : "S",
44   "finalStates" : [
45     "C"
46   ]
47 }
```

# Výstup programu

Program na stdout vypíše jednotlivé stavy, ktorými prešiel počas simulácie vstupného reťazca na konečnom automate. Za každým stavom bude symbol nového riadku LF (\n) a to vrátane posledného symbolu. Ukážkový výstup pre vstupný reťazec *aaa* a ukážkový automat č. 1 je:

```
1 | A
2 | A
3 | A
4 | A
5 |
```

Pre reťazec *abc* a automat č. 2 sa očakáva výstup:

```
1 | S
2 | AB
3 | AB
4 | C
5 |
```

## Chybové stavy

Pri detekovaní chyby sa program správne ukončí, na stderr vypíše informácie o chybe a vráti chybový kód podľa nasledujúcej tabuľky:

Typ chyby	Návratový kód
Vstupný reťacez nie je akceptovaný automatom	6
Nesprávne arumenty	11
Chyba pri práci so súborom	12
Chyba pri dekódovaní automatu	20
Automat obsahuje nedefinovaný stav	21
Automat obsahuje nedefinovaný symbol	22
Automat nie je deterministický	23
Iná chyba	99

## Kostra programu

---

Pre riešenie využite kostru programu s doporučenou štruktúrou modulov. Každý modul má pripavený samostatný priečinok v priečinku *Sources*.

## Moduly

Potrebné implementovať:

- *FiniteAutomata* - knižnica pre reprezentáciu a dekódovanie končeného automatu
- *proj1* - modul obsahujúci hlavný kód pre beh programu
- *Simulator* - knižnica pre simuláciu konečného automatu

Preimplementovaný modul:

- *MyFiniteAutomatas* - Príklady končených automatov, vstupov a výstupu použitých v testoch

## Testovanie

V kostre sú pripravené aj štyri testovacie sady, ktoré je možné spustiť pomocou príkazu `swift test` v domovskom priečinku projektu. Sady sa nachádzajú v priečinku *Tests/proj1Tests*:

- *FiniteAutomataTests* - dve sady testov pre simulátor,
- *proj1Tests* - dve sady testov pre výslednú binárku.

Doporučuje sa doplniť si aj vlastné testy.

## GitLab CI/CD

V kostre je priložený aj súbor *.gitlab-ci.yml*, ktorý automaticky spúšťa testy po každom pushnutí do repozitára na gitlabe. Pokiaľ budete využívať službu gitlab tak Vám to môže pomôcť odhaliť chyby hneď ako sa vyskytnú.

## Mac Xcode

Pre vytvorenie projektu pre Xcode *.xcodproj* na systéme MacOS je potrebné použiť príkaz `swift package generate-xcodeproj` v domovskom priečinku projektu.

## Odovzdanie

---

Odovzdáva sa *xlogin00.zip* s Vaším loginom, ktorý bude obsahovať:

- *Package.swift*
- *Sources/*
- *Tests/*

