

Проект "Детерминиран краен автомат"

Проектът работи под формата на текстов-диалогов режим. Библиотеките, които са използвани в този проект са :

- **#include<cstring>** - използвана главно за намиране на дължината на стринг.
- **#include<map>** - използвана, заради класа map, чрез който са представени преходите на автомата. map е контейнер от елементи от вида <key,value> ,където key е специфичен за всеки елемент от map.Това условие прави този клас подходящ за краен детерминиран автомат.
- **#include<set>** - използвана, заради класа set, който пази финалните състояния.
- **#include <windows.h>** - използвана , за да изведе кирилица на конзолата.
- **#include<utility>** - използвана, заради класа pair, чрез който са предадени първата част от прехода на автомата.

При стартиране на програмата се иска от потребителя да въведе автомат, който представлява обект от класа DFA, за който са реализирани следните член-данни и член-функции :

- **int number_states** - броят на състоянията на автомата.
 - **int startstate** - номерът на началното състояние
 - **set<int>finalstates** - контейнер с финалните състояния
 - **char* alphabet** - азбукта, която участва в автомата
 - **typedef pair<int,char> trans** - обект, който се състои от int и char част, като първото е номера на състоянието, а второто е буквата, която излиза от състоянието.
 - **map<trans,int> transition** - контейнер от обекти, който пази преходите на автомата.
- DFA()** - конструктор по подразбиране.
- **DFA(DFA const&)** - copy-конструктор.
 - **~DFA()** - деструктор.

- **DFA& operator=(DFA const&)** - операция за присвояване.
- **DFA(int,int)** - конструктор с 2 параметъра
- **void setfinal(int)** - функция, с която потребителя въвежда крайните състояния, като за край на се използва отрицателно число (приема се, че състоянията са положителни последователни числа).
- **void setmaps()** - функция, с която потребителят въвежда преходите като за край се използва знака равно между съответните състояния.
- **void setalphabet()** - функция, с която потребителят въвежда азбуката
- **int getNumberState() const** – функция, която връща броя на състоянията
- **int getStartstate() const** – функция, която връща номера на началното състояние
- **int returmap(const pair<int,char> sth)** – функция, която връща състоянието, към което влиза съответната буква (излизаща от съответното състояние).
- **int finalstatesize() const** – функция, която връща броя на финалните състояния.
- **int countalpha() const** – функция, която връща броя на символите в азбуката
- **char* alpha() const** – функция, която връща самата азбука.
- **bool tranfind(int,char) const** – функция, която проверява дали съществува такава двойка – състояние и буква, излизаща от него
- **bool iffinalstate(int) const** – функция, която проверява дали дадено състояние е финално.
- **void print() const** – функция, която принтира самия автомат.
- **void clearfinal()** – функция, която изтрива финалните състояния.
- **void newfinal(int x)** – функция, която добавя ново финално състояние
- **bool isAccepted(char* word)** – функция, която проверява дали подадената дума се разпознава от автомата.
- **void errorstate()** – функция, която добавя ново състояние и прави автомата тотален.
- **DFA& dopalnenie()** – функция, която прави допълнение на автомата.

- **DFA& obedinenie(DFA&)** – функция, която прави обединение на 2 автомата като стойността, която връща е пак автомат.
- **friend ostream& operator<<(ostream&, DFA const&)** – оператор, който извежда автомата.
- **DFA& se4enie (DFA& , DFA&)** – функция, която прави сечение на 2 автомата.
- **void onthescreen()** – функция, която извежда текста със съответните възможности на екрана.
- **void just_to_start()** – функция, която стартира програмата.
- **Void infolanguage()** – функция, която извежда информация за езика, който се разпознава от автомата – дали езика е краен, пълен или празен.