

PIZZO

Pierwsza lista zadań

Zadania domowe nr 0 (rozgrzewkowe)

Napisz emulator deterministycznych automatów skończonych.

Środowisko

Program będzie uruchamiany pod aktualnym stabilnym Debianem. Wszystkie ewentualne paczki, które należy doinstalować, aby skompilować lub uruchomić program, powinny być wymienione w pliku `packages.nfo` w katalogu głównym programu. W tym samym katalogu powinny być również pliki `compile.sh`, który kompiluje program¹ oraz `run.sh`, który uruchamia program.

Wejście i wyjście

Program ma wczytywać dane ze standardowego wejścia (klawiatury). Pierwsza linia danych zawiera ścieżkę do pliku, w którym znajduje się opis automatu w formacie JSON. Kolejne linie zawierają słowa nad alfabetem tego automatu. Dane kończą się znakiem końca pliku EOF.

Format pliku JSON jest następujący:

```
{
  alphabet : string[],    // alfabet (lista liter)
  states    : string[],    // lista stanów
  initial   : string,      // stan startowy
  accepting : string[],    // stany akceptujące
  transitions : {
    letter : string,      // po tej literze przechodzimy...
    from   : string,      // z tego stanu...
    to     : string,      // do tego stanu
  }[]           // lista przejść
}
```

Na przykład, to jest plik `testowy.aut`:

```
{
  alphabet : ["a", "b"],
  states    : ["q1", "q2"],
  initial   : "q1",
  accepting : ["q2"],
  transitions : [
```

¹Ten plik może być pusty, może też się składać z jednej instrukcji, n.p. `make`

```

        {letter      : "a", from      : "q1",to      : "q1"}
        {letter      : "b", from      : "q1",to      : "q2"}
        {letter      : "a", from      : "q2",to      : "q1"}
        {letter      : "b", from      : "q2",to      : "q2"}
    ]
}
```

Program powinien wypisywać na standardowe wyjście, dla każdego kolejnego słowa wejście, pojedyncze słowo „yes” jeśli należy ono do języka tego automatu, lub „no” w przeciwnym przypadku.

Przykładowe uruchomienie:

```

jmi@pizzo:~$ ./compile.sh
jmi@pizzo:~$ ./run.sh
testowy.aut
aaaa
abab
baba
bbbb
```

powinno zwrócić

```

no
yes
no
yes
```

Uwagi końcowe

Program będzie testowany wyłącznie na poprawnych danych. Automaty będą małe, ale słowa będą duże (mogą się nie mieścić w pamięci maszyny), dlatego należy wczytywać je znak po znaku. Wszystkie dane będą poprawne i nie trzeba tego sprawdzać. Wszystkie znaki używane jako litery i w nazwach stanów to będą zwykle litery alfabetu angielskiego oraz cyfry.