



به نام خدا

پروژه اول درس نظریه‌ی زبان ها و ماشین ها

مدرس درس: دکتر حسین رحمانی

دانشگاه علم و صنعت ایران

لطفا قبل از مطالعه‌ی داک به موارد زیر توجه کنید:

- 1- پروژه در قالب گروه‌های دونفره می‌باشد.
- 2- مهلت ارسال کد ها بعد از اتمام 15 روز کاری بعد از ارائه‌ی داک پروژه می‌باشد و به ازای هر روز تاخیر 10 درصد از نمره‌ی پروژه کم میشود.
- 3- در صورت یافتن کد مشابه در 2 گروه نمره‌ی هر دو گروه در پروژه صفر منظور خواهد شد.
- 4- در صورت بروز هرگونه سوال و مشکل میتوانید از حل تمرین‌های درس کمک بگیرید.
- 5- لطفا اسم توابع خواسته شده را تغییر ندهید.
- 6- کد های مربوط به پروژه را به آقای سیدنصیری از طریق تلگرام ارسال در پایان مهلت ارسال پروژه ارسال نمائید.

این پروژه مربوط به قسمت Finite Automata می‌باشد و در ابتدا به شما یک ورودی اولیه داده می‌شود که در واقع توصیف کننده‌ی یک NFA می‌باشد و فرمت ورودی اولیه به صورت زیر می‌باشد.

در ابتدا یک مجموعه به عنوان ورودی داده می‌شود که حالت های ماشین را در بر دارد و اولین عضو این مجموعه نیز حالت شروع ماشین می‌باشد.

سپس الفبای زبانی که ماشین آن را قبول می کند به شما داده می شود و در خط بعدی برنامه یک عدد صحیح نامنفی به شما داده می شود که بیانگر تعداد قوانین انتقال ماشین می باشد.

سپس در خطوط بعدی قوانین انتقال به شما داده خواهد شد و در نهایت در آخرین خط ورودی به شما مجموعه حالت های قبولی ماشین به شما داده می شود.

به عنوان مثال :

$\{q_0, q_1, q_2, q_3\}$

$\{a, b\}$

6

$q_0, q_1, a$

$q_1, q_2, b$

$q_1, q_3,$

$q_3, q_4, b$

$q_2, q_3, a$

$q_4, q_2, a$

$(q_1, q_3)$

در ورودی مقابل ماشین مورد نظر 4 حالت دارد که  $q_0$  حالت شروع می باشد و حالات  $q_1, q_3$  نیز حالت های پایان می باشند. زبان مورد پذیرش توسط این ماشین دارای الفبای  $a, b$  می باشد.

$q_0, q_1, a$  بدین معنی می باشد که ماشین اگر در حال  $q_0$  باشد و  $a$  را ببیند وارد حالت  $q_1$  می شود. در صورتی که الفبایی از زبان در این قوانین نباشد بیانگر لاندا ترنزیشن می باشد.

در ابتدا دو کلاس  $NFA, DFA$  ساخته و صفات و متدهایی که برای نوشتن توابع زیر نیاز دارید را در آنها بنویسد

(1) یک تابع به اسم `IsAcceptByNFA` نوشته که به عنوان ورودی یک رشته کاربر دریافت کند و در صورت قبول شدن رشته در ماشین مقدار `True` و درغیراین صورت مقدار `False` را برگرداند.

(2) یک تابع به اسم `FindRegex` بنویسید که از `regular expression` ماشین فوق را به عنوان خروجی در قالب یک رشته برگرداند.

(3) یک تابع به اسم `CreateEqueulvantDFA` نوشته که `DFA` معادل ماشین توصیف شده را به عنوان خروجی بازگرداند.

(4) یک تابع به اسم `IsAcceptByDFA` نوشته که دراز کاربر یک رشته به عنوان ورودی دریافت نموده و در صورت پذیرفته شدن در `DFA` مقدار `True` و درغیراین صورت مقدار `False` را برگرداند.

(5) یک تابع به اسم `MakeSimpleDFA` بنویسید که ساده شده‌ی `DFA` ورودی را به عنوان خروجی بازگرداند.

(6) یک متد نیز برای کلاس های `DFA` , `NFA` بنویسید که در صورت فراخوانی آن متد ماشین به صورت شماتیکی نمایش داده شود.

موفق و پیروز باشد