



دانشکده مهندسی کامپیوتر

دکتر رضا انتظاری ملکی

بهار ۱۴۰۰

پروژه اول

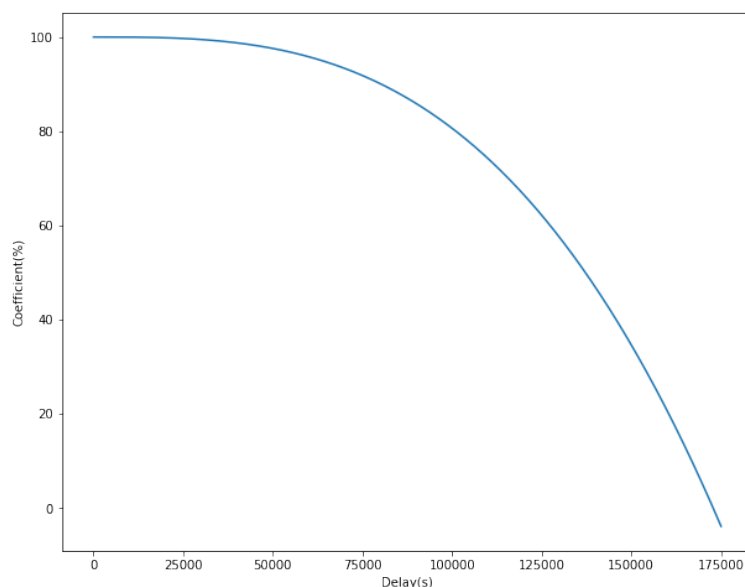
نظریه زبانها و ماشینها

طراح: گروه دستیاران آموزشی درس

تاریخ تحویل: ۳ اردیبهشت ۱۳۹۹:۵۹

قوانین

- در صورت مشاهدهی هرگونه تقلب (از گروه‌های دیگر یا از کدهای موجود در اینترنت)، نمره‌ی کل پروژه را از دست خواهید داد.
- پروژه باید به صورت انفرادی و یا در قالب گروه‌های دوفره انجام گیرد.
- تا تاریخ ۸ فروردین ۱۴۰۰ فرصت دارید تا اطلاعات اعضای گروه را در فرم موجود در [این لینک](#) وارد کنید. درغیراین‌صورت، پروژه‌ای از شما تحویل گرفته نخواهد شد. تکمیل فرم توسط یکی از اعضای گروه کافی است.
- به دلیل فشرده بودن برنامه‌ی ترم، امکان تمدید مهلت پروژه وجود ندارد.
- در صورت وجود هرگونه سوال از طریق گروه تلگرام یا تیمز مطرح کنید. (لطفا پی‌وی پیام ندهید.)
- برای پیاده‌سازی پروژه می‌توانید از هر زبان برنامه‌نویسی دلخواه خود استفاده کنید.
- پس از تحویل پروژه زمان‌بندی جهت ارائه‌ی پروژه اعلام می‌شود.
- محور افقی این نمودار مقدار تاخیر به ثانیه و محور عمودی ضریب اعمالی در نمره پروژه است.



شکل ۱: نمودار تاخیر



فهرست مطالب

۳	۱ مقدمه
۳	۲ فرمت ورودی برنامه
۴	۳ نحوه اجرای برنامه
۴	۴ توابعی که باید پیاده سازی کنید
۵	۵ نحوه ارسال



شرح پروژه

۱ مقدمه

تاکنون به طور مقدماتی با نحوه‌ی عملکرد Finite Automata آشنا شده‌اید. حال در این پروژه قرار است یک Finite Automata را به طور کامل با استفاده از دانش برنامه‌نویسی خود پیاده‌سازی کنید که تمامی ویژگی‌های خواسته‌شده را برآورده سازد.

۲ فرمت ورودی برنامه

در ابتدا به شما یک ورودی اولیه داده می‌شود که درواقع توصیف‌کننده‌ی یک NFA می‌باشد.

درخط اول ورودی، یک مجموعه وارد می‌شود که حالت‌های ماشین را دربردارد و اولین عضو این مجموعه نیز حالت شروع ماشین می‌باشد.

در خط دوم الفبای زبانی که ماشین آن را قبول می‌کند، می‌آید.

در خط سوم، مجموعه‌ی حالات پایانی ماشین موردنظر به شما داده می‌شود.

در خط چهارم، یک عدد صحیح مثبت داده می‌شود که بیانگر تعداد قوانین انتقال ماشین موردنظر است و سپس، در هریک از n خط بعدی، هریک از قوانین انتقال به شما داده خواهد شد.

به عنوان نمونه، ورودی زیر را درنظر بگیرید:

$\{q_0, q_1, q_2, q_3, q_4\}$

$\{a, b\}$

$\{q_1, q_3\}$

6

q_0, q_1, a

q_1, q_2, b

$q_1, q_3,$

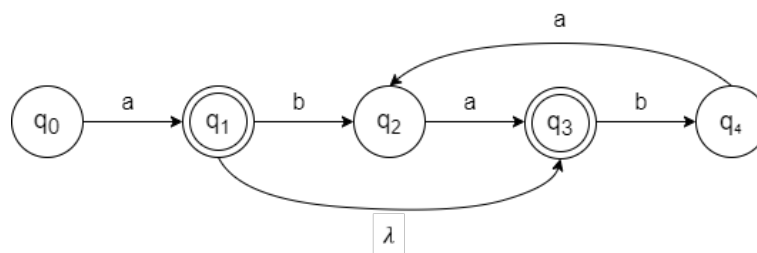
q_3, q_4, b

q_2, q_3, a

q_4, q_2, a

در ورودی بالا، ماشین موردنظر ۴ حالت دارد که q_0 حالت شروع می‌باشد و حالات q_1 و q_3 نیز حالات پایانی می‌باشند. زبان مورد پذیرش این ماشین دارای الفبای $\{a, b\}$ می‌باشد. q_0, q_1, a به این معنی است که اگر ماشین در حالت q_0 باشد و a را ببیند، وارد حالت q_1 می‌شود. در صورتی که الفبایی از زبان در این قوانین نباشد، بیانگر λ می‌باشد.

شکل زیر نمایانگر ماشین ورودی موردنظر است:



۳ نحوه‌ی اجرای برنامه

شما باید در ابتدا دو کلاس DFA و NFA را بسازید و متدهایی که برای نوشتن توابع نیاز دارید، در این کلاس‌ها پیاده‌سازی کنید. سپس در کلاس Main ورودی را از کاربر گرفته و منوی انتخاب توابع را تا زمانی که کاربر Exit وارد نکرده باشد، به کاربر نمایش دهید.

۴ توابعی که باید پیاده‌سازی کنید

توابع موردنیاز برای کلاس NFA:

(آ) یک تابع به نام `isAcceptByNFA` بنویسید که به عنوان ورودی، یک رشته را دریافت کند و در صورت پذیرفته شدن رشته در ماشین مقدار `True` و در غیر این صورت مقدار `False` را برگرداند. ✓

(ب) یک تابع به نام `createEquivalentDFA` بنویسید که DFA معادل ماشین توصیف شده را به عنوان خروجی برگرداند. ✓

(ج) یک تابع به نام `findRegExp` بنویسید که عبارت منظم (regular expression) معادل ماشین توصیف شده را به عنوان خروجی برگرداند.

(د) یک تابع به نام `showSchematicNFA` بنویسید که در صورت فراخوانی، NFA به صورت شماتیکی نمایش داده شود. ✓



توابع موردنیاز برای کلاس DFA:

- (آ) ✓ یک تابع به نام `isAcceptByDFA` بنویسید که به عنوان ورودی، یک رشته را دریافت کند و در صورت پذیرفته شدن رشته در DFA مقدار `True` و در غیر این صورت مقدار `False` را برگرداند.
- (ب) یک تابع به نام `makeSimpleDFA` بنویسید که یک DFA را به عنوان ورودی دریافت کند و آن را تا حد امکان ساده کند و DFA ساده شده را به عنوان خروجی برگرداند.
- (ج) ✓ یک تابع به نام `showSchematicDFA` بنویسید که در صورت فراخوانی، DFA به صورت شماتیکی نمایش داده شود.

۵ نحوه ارسال

تمامی فایل های موردنیاز برای اجرای پروژه را در قالب یک فایل Zip داخل کوئرا ارسال کنید. توجه کنید که هر دو عضو گروه باید فایل پروژه را آپلود کنند.

هم چنین از روز تحویل پروژه تا روز ارائه نباید به هیچ وجه کد خود را تغییر دهید. در غیر این صورت، نمره از شما کسر خواهد شد.

موفق باشید.