به نام خدا

نظریه زبان‌ها و ماشین‌ها- بهار ۱۴۰۱

تمرین شماره 10

دستیار آموزشی این مجموعه: معین کرمی

moein2000n@gmail.com

تاریخ تحویل :۲۵ خرداد

** توجه کنید که برای اثبات وجود یک ماشین تورینگ در این تمرین، نیازی نیست بیش از حد وارد جزییات طراحی ماشین شوید و تنها کافی است ثابت کنید که یک ماشین وجود دارد (با ارائه الگوریتم کلی).

1) فرض کنید D یک DFA باشد. زبان A شامل تمام پیشوند های کلمات زبان L(D) است. ثابت کنید A تصمیم پذیر است: (۱۰)

راه حل: در ابتدا مشخص می‌کنیم از کدام یک از استیت‌های D می‌توانیم به یک accept state برسیم، برای این کار ابتدا accept state ها را علامت می‌زنیم و تا وقتی که هیچ استیت جدیدی علامت نخورد، هر استیت علامت نخورده ای که به یک استیت علامت خورده راه دارد را علامت می‌زنیم.

حال DFA را روی کلمه ی x پیاده سازی می‌کنیم. فرض کنید بعد از این شبیه سازی در استیت s از D قرار بگیریم.حال اگر این استیت علامت خورد باشد یعنی می‌توانیم با اضافه کردن یک رشته به انتهای x به یک accept state برسیم پس این کلمه عضو زبان است، در غیر این صورت این کلمه عضوی از زبان نیست.

2) موارد زیر را اثبات کنید.(۱۵)

الف) زبان‌های turing-recognizable نسبت به اجتماع و اشتراک بسته اند.

ب) زبان‌های turing-decidable نسبت به اجتماع، اشتراک و مکمل بسته‌اند.

الف) اجتماع: می‌توانیم یک ماشین بسازیم که دو ماشین تورینگ را به صورت موازی روی ورودی اجرا کند و اگر هر یک accept شد، کلمه را قبول کند.

اشتراک: مانند بالا یک ماشین می‌سازیم که به صورت موازی دو ماشین را روی یک ورودی اجرا کند، اگر هر دو accept شدند، کلمه را قبول می‌کنیم و در غیر این صورت قبول نمی‌کنیم.

ب) اجتماع : چون ماشین هر دو زبان پایان پذیر است کافی است یک ماشین بسازیم که هر دو ماشین را پشت هم روی کلمه ورودی شبیه سازی کند و اگر هرکدام کلمه را قبول کرد، ما نیز کلمه را قبول می‌کنیم.

اشتراک: دقیقا مانند قسمت قبل عمل می‌کنیم ولی تنها زمانی یک کلمه را می‌پذیریم که هر دو ماشین آن را بپذیرند.

مکمل: ماشین را روی کلمه ورودی اجرا می‌کنیم ولی نتیجه را برعکس می‌کنیم.

3) الف) ثابت کنید زبان A که شامل تمام DFA هایی است که فقط رشته های palindrome را قبول می‌کنند، یک زبان تصمیم پذیر است. (۲۰)

یک ماشین تورینگ به این شکل طراحی می‌کنیم:
فرض کنید ورودی <D> به ما داده شده است، یک NFA به نام N به این صورت می‌سازیم: ابتدا D را کپی می‌کنیم و سپس تمام یال های آن را بر عکس می‌کنیم، یک استیت به نام S به آن اضافه می‌کنیم که استیت شروع است و S را با epsilon به تمام accept استیت های D وصل می‌کنیم و در نهایت همه ی استیت ها را از حالت accept خارج می‌کنیم و استیت شروع D رو عنوان اکسپت استیت در نظر می‌گیریم.
حال N قرینه ی تمام رشته هایی را می‌پذیرد که D آن ها را می‌پذیرد. در اینجا N را تبدیل به DFA می‌کنیم و نام آن را T میگذاریم.
حال اگر

$L(T) = L(D)$

باشد (چگونگی چک کردن این تساوی تدریس شده است) یعنی این DFA تنها زبان های پالیندروم را می‌پذیرد و ما این ورودی را قبول می‌کنیم و در غیر این صورت این زبان را رد می‌کنیم.

ب) زبان L را شامل تمام ماشین تورینگ هایی مانند M تعریف می‌کنیم که

$<M> \notin L(M)$

ثابت کنید این زبان تشخیص پذیر نیست.(امتیازی ۱۰)

فرض خلف کنید این زبان تشخیص پذیر باشد.
در این صورت یک ماشین تورینگ به نام T برای آن داریم که حتما halt میکند.
حال <T> را به عنوان ورودی به T می‌دهیم. ۲ حالت داریم:
۱- خودش را بپذیرد. در این صورت طبق تعریف T در لوپ افتاده و یا جواب رد داده است که این تناقض است.
۲- خودش را نپذیرد. که طبق تعریف یعنی خود را پذیرفته و این نیز تناقض است.
پس فرض خلف در هر صورت به تناقض می‌رسد و باطل شده و حکم اثبات می‌شود.

4) یک ماشین تورینگ به نام M داریم که L(M) شامل تمام ماشین های تورینگ مانند T است که

$L(T) \,!= \emptyset$

ثابت کنید L(M) تشخیص پذیر است. (۲۰)

برای اثبات این موضوع یک ماشین تورینگ به این صورت طراحی می‌کنیم:
در ابتدا فرض کنید ورودی ما T باشد و الفبای T برابر A باشد.
فرض کنید ...a1, a2, a3 دنباله ی شامل تمام رشته های ممکن از A باشد.
ماشین M را به این صورت طراحی میکنیم که به ازای ...,i =1, 2, 3 ماشین T را به اندازه ی i مرحله روی a1 تا ai اجرا می‌کند و هرگاه T یک رشته را قبول کرد، ما نیز T را می‌پذیریم.

5) ثابت کنید زبان زیر تصمیم پذیر است: (۲۰)

$A = \{<G> \mid G \text{ is a CFG over } \{0, 1\}^* \text{ and } L(G) \cap L(1^*) = \emptyset\}$

می دانیم که

L(1*) is a regular language.

در نتیجه اشتراک L(G) و L(1*) نیز یک CFL است.

حال کافی است یک CFG که زبان آن اشتراک دو زبان بالا است را بسازیم و نام آن را C بگذاریم و سپس چک کنیم آیا زبان C تهی است یا خیر، اگر تهی بود ماشین ما <G> را اکسپت می‌کند و در غیر این صورت ریجکت می‌کند.

6) زبان L را شامل <G,A> هایی در نظر می‌گیریم که G یک گرامر مستقل در متن باشد و غیر پایانه ی A در هیچ یک از اشتقاق های آن به ازای تمام کلمات زبان L(G) دیده نشود. ثابت کنید زبان L تصمیم پذیر است. (۱۵)

برای اثبات این موضوع یک ماشین تورینگ طراحی می‌کنیم که حتما به پایان برسد و در لوپ نیفتد.

الگوریتم کلی ماشین به این شکل است:

۱ - تمام پایانه ها را علامت میزنیم.

۲- اگر غیر پایانه ای وجود داشت که: ۱) علامت نخورده بود، ۲) قانونی داشت که تمام متغیر های آن اعم از پایانه ها و غیر پایانه ها علامت خورده بودند، را علامت می‌زنیم و دوباره این مرحله را تکرار می‌کنیم، در غیر این صورت به مرحله ۳ می‌رویم

۳- اگر غیر پایانه A علامت خورده بود زبان را نمی‌پذیریم، در غیر این صورت می‌پذیریم.