به نام خدا

نظریه زبان‌ها و ماشین‌ها- بهار ۱۴۰۱

تمرین شماره 10

دستیار آموزشی این مجموعه: صبا شهسواری
sabashahsavari@ut.ac.ir

تاریخ تحویل : ۷ خرداد

---

1. زبان L شامل تمام DFA هایی می‌شود که زبانشان نامتناهی است. ثابت کنید L تصمیم پذیر است.

برای تشخیص DFA های عضو این مجموعه از الگوریتم زیر استفاده می‌کنیم:

فرض کنید n تعداد state های برای یک DFA خاص باشد. در صورتی که هیچ رشته ای با طول بیشتر از n در زبان موجود نباشد، مطمئنا زبان متناهی است. برای رشته های به طول بیشتر از n از روش زیر استفاده می‌کنیم:

ادعا میکنیم که در صورتی که رشته ای با طول n یا بیشتر در زبان موجود باشد، حتما رشته ای با طول n تا 2n-1 در زبان موجود است. برای اثبات از برهان خلف استفاده میکنیم. فرض کنید m طول کوتاه ترین رشته در میان رشته‌های با طول بیشتر از n است. اگر m > 2n باشد می‌توانیم از pumping lemma استفاده کنیم و z را به صورت wxy = m بنویسیم. میدانیم wy رشته ای در زبان است. اما از آنجا که $|wx| \leq n$ است پس $m > |wy| \geq n$ که این با فرض اولیه که m کوتاهترین رشته در میان رشته های با طول بیشتر از n است در تناقض است.

بنابر این کافی است برای بررسی متناهی بودن زبان یک DFA عضویت تمام رشته های با طول بین n تا 2n-1 را در آن زبان بررسی کنیم. در صورتی که چنین رشته ای وجود نداشته باشد، زبان متناهی است و در غیر این صورت زبان نامتناهی است. چون در تعداد گام‌های محدودی می‌توان این بررسی را انجام داد پس زبان L تصمیم پذیر است.

2. زبان L شامل زوج های< G, B > است که در آن G یک گرامر مستقل از متن و B یک متغیر در آن است به طوری که B در اشتقاق حداقل یکی از رشته‌هایی که توسط G تولید می‌شود استفاده شده است. ثابت کنید این زبان تصمیم پذیر است.

گرامر 'G را به شکل زیر می‌سازیم:

تمام قواعد G که در سمت چپشان B وجود دارد را حذف می‌کنیم و در قواعدی که سمت راستشان B وجود دارد، به جای B از یک نماد جدید مانند 'B استفاده می‌کنیم.

حال اشتراک زبان 'G را با زبان $R = (B' \cup \Sigma)^* B' (B' \cup \Sigma)^*$ حساب می‌کنیم. می‌دانیم اشتراک یک زبان منظم و یک گرامر مستقل از متن یک زبان مستقل از متن است و با یک الگوریتم پایان پذیر می‌توان

اشتراک آن‌ها را حساب کرد. اگر اشتراک این دو زبان تهی باشد به این معناست که B در اشتقاق هیچ رشته G استفاده نشده پس زبان رد می‌شود و در غیر این صورت پذیرفته می‌شود.

3. عبارت منظم R زبان‌هایی را توصیف می‌کند که حداقل شامل یک رشته مانند w هستند به طوری که 111 زیررشته‌ای از w است. ثابت کنید تشخیص این که زبان R' چنین ویژگی دارد یا نه تصمیم پذیر است.

زبان عبارت منظم R' را می‌توان به شکل $L(R') \cap (\Sigma^* 111 \Sigma^*)$ نوشت. برای تصمیم گیری این زبان از الگوریتم زیر استفاده می‌کنیم:

1-یک DFA به نام A می‌سازیم که $\Sigma^* 111 \Sigma^*$ را می‌پذیرد.

۲- از اشتراک A و R یک DFA به نام B می‌سازیم. $L(B) = L(A) \cap L(R)$. (می‌دانیم می‌توان با الگوریتم پایان پذیر DFA حاصل از اشتراک دو DFA را تشکیل داد.)

3- ماشین تورینگ M که می‌تواند A را decide کند ساخته و آن را روی B اجرا می‌کنیم. اگر M بپذیرد به این معناست که $L(A) \cap L(R) = \emptyset$ بوده پس R' این ویژگی را نداشته و رد می‌شو. اگر M رد کند یعنی R' این ویژگی را داشته و پذیرفته می‌شود.

4. فرض کنید A و B دو زبان turing-recognizable باشند. تشخیص پذیر یا تشخیص ناپذیر بودن زبان A-B را ثابت کنید.

با استفاده از یک مثال نقض ثابت می کنیم A - B تشخیص پذیر نیست. فرض کنید Σ^* = A و $B = A_{TM}$ باشد.

می‌دانیم A_{TM} تصمیم پذیر نیست. A - B مکمل زبان A_{TM} می‌شود. می‌دانیم اگر یک زبان و مکملش هر دو تشخیص پذیر باشند، آن زبان حتما تصمیم پذیر است در نتیجه A - B تشخیص ناپذیر است.

5. ثابت کنید این مسئله که آیا ماشین تورینگ M فقط رشته‌هایی که Palindrome هستند را می‌پذیرد یا خیر، تصمیم ناپذیر است.

با استفاده از یک مثال نقض ثابت می‌کنیم این زبان تصمیم پذیر نیست.

فرض کنید تصمیم پذیر باشد پس یک ماشین تورینگ مانند D وجود دارد که آن را می‌پذیرد. با استفاده از این ماشین یک ماشین تورینگ N را به شکل زیر طراحی می‌سازیم:

ماشین N ورودی <M,w> که M توصیف یک ماشین تورینگ و w رشته ورودی است را دریافت می‌کند و ماشین تورینگ R را می‌سازد. ماشین R ورودی <M,w> را دریافت می‌کند و اگر w یک رشته palindrome نباشد آن را رد می‌کند و در غیر این صورت، ماشین M را روی w اجرا می‌کند اگر M پذیرفت، ماشین R نیز می‌پذیرد و در غیر این صورت رد می‌کند. پس از ساخت ماشین R، توصیف آن را به عنوان ورودی به ماشین D می‌دهد. اگر D پذیرفت، N نیز می‌پذیرد و اگر D رد کرد ماشین N نیز رد می‌کند.

چون فرض کرده بودیم D یک decider است پس حتما یا می‌پذیرد یا رد می‌کند و به همین دلیل ماشین N نیز حتما یا می‌پذیرد یا رد می‌کند(در نهایت متوقف می‌شود) . پس ماشین N یک decider برای زبان A_{TM} است. در حالی که در کلاس اثبات شده زبان A_{TM} تصمیم پذیر نیست پس فرضی که داشتیم غلط است. یعنی D نمی تواند وجود داشته باشد و این زبان تصمیم ناپذیر است.

6. تصمیم پذیری زبان‌های زیر را بررسی کنید.

الف) زبان L شامل توصیف ماشین تورینگ M و رشته w است به طوری که ماشین M در پردازش رشته w بیشتر از یک بار وارد یک state می‌شود.

این زبان تصمیم پذیر است. می‌دانیم تعداد state های یک ماشین تورینگ محدود است. اگر تعداد state های ماشین تورینگی که در ورودی داده شده N باشد، این ماشین حداکثر می‌تواند 2N-1 گام بردارد بدون آن که دو بار وارد یک state شود و در گام 2Nام (طبق اصل لانه کبوتری) حتما وارد یک state تکراری می‌شود. می‌توانیم ماشین تورینگی به شکل زیر برای decide کردن این زبان طراحی کنیم:

می‌دانیم ماشین تورینگ دونواره قدرت مشابهی با ماشین تورینگ تک‌نواره دارد پس از یک ماشین دو نواره به نام N استفاده می‌کنیم. به ازای ورودی <M,w> ، ابتدا روی نوار دوم شماره تمام state های ماشین M را دو بار می‌نویسیم سپس M را روی w اجرا می‌کنیم. هر بار که M به state شماره i رفت در نوار دوم دنبال عدد i می‌گردیم اگر پیدا شد آن را پاک کرده و ادامه می‌دهیم و اگر پیدا نشد یعنی M می‌خواهد برای بار دوم وارد آن state شود و <M,w> پذیرفته می‌شود. اگر ماشین M رشته w را پذیرفت یا رد کرد آن گاه N ورودی <M,w> رد می‌کند.

ب) زبان شامل تمام توصیف های ماشین‌های تورینگ مانند <M> که M روی هد ورودی به طول k متوقف می‌شود.

این زبان تصمیم پذیر است. می‌توانیم ماشین تورینگ پذیرنده این زبان را به شکل زیر تعریف کنیم:

ماشین تورینگ T به ازای ورودی <M>، ماشین M را روی تمام رشته‌ها به طول k اجرا می‌کند. اگر M هر یک از این رشته‌ها را رد کند یا هرگز متوقف نشود، ماشین T ورودی را رد می‌کند و اگر M تمام این رشته‌ها را بپذیرد، ماشین T ورودی را می‌پذیرد. چون تعداد رشته‌ها به طول k محدود است پس در نهایت ماشین T متقوف می‌شود و M را decide می‌کند.

7. ثابت کنید زبان‌های تشخیص پذیر تحت عمل concatenation بسته هستند.

دو زبان تشخیص پذیر A و B را در نظر بگیرید. چون A , B تشخیص پذیر هستند پس دو ماشین تورینگ مانند M_A و M_B وجود دارند که می‌توانند به ترتیب A و B را تشخیص دهند. ماشین تورینگ M_{AB} را برای تشخیص زبان AB به شکل زیر می‌سازیم:

۱- ورودی را دریافت می‌کنیم و آن را به دو زیر ورودی تقسیم می‌کنیم

۲- با استفاده از ماشین M_A ، ورودی اول را بررسی می‌کنیم. اگر قبول شود، به مرحله بعدی می‌رویم و اگر قبول نشود، M_{AB} نیز آن را رد می‌کند.

۳- با استفاده از ماشین M_B ، ورودی دوم را بررسی می‌کنیم. اگر قبول شود، به مرحله بعدی می‌رویم و اگر قبول نشود، M_{AB} نیز آن را رد می‌کند.

۴- اگر هر دو ورودی قبول شوند، ورودی را قبول می‌کنیم و در غیر این صورت رد می‌شود .