به نام خدا

نظریه زبان‌ها و ماشین‌ها- بهار 1403

تمرین شماره 11

دستیار آموزشی این مجموعه: فربد عظیم محسنی
farbodazimmohseni@gmail.com

تاریخ تحویل : 20 خرداد

- در هیچ کدام از اثبات ها به غیر از سوال 3 مجاز به استفاده از قضیه Rice نیستید.

1. یک از دانشجویان که هنوز بخش reduction را مطالعه نکرده است زبان زیر را می بیند(این زبان تمام ماشین تورینگ هایی را شامل می شود که تعداد متناهی رشته را می پذیرند):

$$FINITE_{TM} = \{< M > \mid L(M)\ is\ finite\ \}$$

و اینگونه استدلال می کند که میتوان یک ماشین تورینگ ساخت که ماشین تورینگی را به عنوان ورودی بگیرد و چک کند که آیا در stateهای آن loop وجود دارد یا نه؟ اگر وجود داشت یعنی اینکه میتوان به تعداد دلخواه loop زد و بی نهایت رشته را پذیرفت. شما به عنوان کسی که این بخش را مطالعه کرده است به سوالات زیر جواب دهید:

- الف) چرا این استدلال اشتباه است؟(5 نمره)

- ب) با استفاده از کاهش A_{TM} به $FINITE_{TM}$ ($A_{TM} \le FINITE_{TM}$) ثابت کنید $FININTE_{TM}$ تصمیم ناپذیر است(10 نمره)

- ج)حال که قسمت قبل را ثابت کردید با استفاده از کاهش $A_{TM} \le FININTE_{TM}$ نشان دهید A_{TM} تصمیم ناپذیر است(10 نمره).

الف) چون این ماشین ممکن است در loop بیفتد و در نتیجه نمی توان چنین نتیجه ای گرفت

ب) با استفاده از $A_{TM} \leq FINITE_{TM}$ اثبات می کنیم. فرض کنید $FINITE_{TM}$ تصمیم پذیر باشد. آنگاه ثابت میکنیم که در این صورت A_{TM} نیز تصمیم پذیر می شود و این تناقض دارد. ماشین R را تصمیم گیرنده $FINITE_{TM}$ در نظر گرفته و decider زبان A_{TM} را با نام S به این صورت می سازیم:

ورودی $< M, w >$ را میگیریم. ماشین تورینگ $M2$ را با ورودی x به این صورت می سازیم: به ازای هر x، ماشین M را روی w اجرا می کنیم. اگر accept شد ماشین $M2$، reject میکند و اگر reject شد ماشین $M2$، accept میکند. اینطوری اگر M روی w، accept شود $L(M2) = \Sigma*$ و زبان ما دیگر finite نیست و در غیر این صورت اگر loop بزند یا reject کند ما هیچ رشته ای را نمی پذیرم که باعث می شود زبان finite شود. حال با دادن $M2$ به R یک decider برای S ساخته ایم که این تناقض دارد پس $FINITE_{TM}$ تصمیم پذیر نیست.

ج) برای اثبات، ما ورودی ماشین finite که M است را encode کرده و به عنوان w به و به همراه خود ماشین تورینگ $FINITE_{TM}$ به A_{TM} و چون فرض شده است که A_{TM} تصمیم پذیر است میگوید که آیا finite ماشین M را می پذیرد یا نه که این عملا دارد نشان می دهد که finite هم تصمیم پذیر است، پس تناقض دارد.

2. ثابت کنید مسئله زیر undecidable است.(15 نمره)

$$Palindrome = \{< G > \mid G \text{ is a context free grammar and } L(G) \text{ has at least one palindrome}\}$$

- ★ اگر یک رشته palindrome باشد یعنی از هر دو طرف به یک شکل دیده می شود مثل abba
- ★ راهنمایی: می توانید از کاهش $PCP \leq Palindrome$ استفاده کنید

فرض کنید که Palindrome تصمیم پذیر باشد، و ورودی مسئله PCP نیز دومینو های $[\frac{a_i}{b_i}]$ باشند. هدف این است که با این دومینو ها گرامری بسازیم که در صورتی که به تصمیم گیر Palindrome داده شود عملا باعث تصمیم پذیر شدن مسئله PCP می شود که این خود دارای تناقض است.

حال گرامر را به این صورت می سازیم که در ابتدا همه a_i ها یا b_i ها را انتخاب کرده و آن ها را برعکس میکنیم. در اینجا a_i ها را و آن ها را $\overline{a_i}$ می نامیم. گرامر را به این شکل انتخاب میکنیم:

$$S -> \; b_i S \overline{a_i} \mid b_i \overline{a_i}$$

در صورتی که حتی یک palindrome برای این گرامر وجود داشته باشد، آنگاه حتما یک match برای PCP یافته ایم و چون palindrome را تصمیم پذیر فرض کردیم، PCP نیز تصمیم پذیر می شود و این تناقض دارد پس palindrome تصمیم ناپذیر است.

3. با قضیه Rice، تصمیم ناپذیر بودن کدام یک از زبان های زیر را می توان ثابت کرد و کدام یک را نمی توان(در صورت امکان اثبات کنید، در غیر این صورت دلیل بیاورید)

- الف)(10 نمره)

$$\{\langle M\rangle |\ M\ is\ a\ TM\ and\ never\ writes\ anything\ on\ a\ blank\ cell\}$$

- ب) زبان تورینگ ماشین M حداکثر 3 رشته را میپذیرد(5 نمره)

$$\{\langle M\rangle\ |\ M\ is\ a\ TM\ and\ |L(TM)|\ \ <=\ \ 3\}$$

یک زبان چه خاصیتی باید داشته باشد تا با قضیه Rice بتوان تصمیم ناپذیری آن را اثبات کرد(5 نمره)

الف) ابتدا یک ماشین تورینگ برای این زبان میسازیم، به این صورت که اولین استیت accept است و این زبان این ماشین Σ^* است همچنین هیچ کاراکتری روی نوار نمی نویسد پس زبان آن زبان خواسته شده در صورت سوال است، اسم این ماشین را M1 میگذاریم.

حال یک ماشین تورینگ دیگر برای complement این زبان می نویسیم، که در ابتدا یک کاراکتر blank روی نوار مینویسد و سپس یک کاراکتر غیر blank روی همان خانه می نویسد و سپس به accept میرود این زبان هم همه رشته ها را می پذیر ولی در زبان صورت سوال قرار ندارد. تا اینجا ویژگی nontrivial بودن وجود دارد. اسم این ماشین را M2 میگذاریم.

حال مشکل این است که این اصلا یک property برای این زبان نیست چرا که اگر یک property بود امکان نداشت که $L(M1)\ =\ L(M2)$ باشد. پس با قضیه رایس نمی توان ثابت کرد.

ب) برای اثبات این مورد می توان از قضیه رایس استفاده کرد M1 را طوری می سازیم که همه رشته ها را قبول کند و M2 را هم طوری که هیچ رشته ای را قبول نکند و این دو complement هم هستند و ویژگی Nontrivial بودن را دارند و امکان ندارد که $L(M1)\ =\ L(M2)$ چرا که یکی هیچ رشته ای را نمی پذیرد و دیگر همه رشته ها را می پذیرد. پس این یک nontrivial entity است و در نتیجه این زبان undecidable است.

در حالت کلی وقت زبان ماشین تورینگ بررسی می شود می توان از این قضیه استفاده کرد ولی اگر درباره رفتار ماشین تورینگ مثلا head یا نوار باشد از این راه نمی توان اثبات کرد.

4. ثابت کنید زبان زیر تصمیم ناپذیر است(مسئله L، ماشین تورینگ M و استیت q را دریافت می کند و می گوید که ماشین M حین اجرا آیا هرگز وارد آن استیت می شود یا نه) (10 نمره)

$$L = \{< M, q > \mid M \ never\ enters\ state\ q\}$$

با استفاده از کاهش $Empty_{TM} \leq L$ ثابت میکنیم که L تصمیم ناپذیر است. ماشین Empty یک ماشین تورینگ M میگیرد و می گوید که آیا زبان این ماشین تهی است یا خیر. حال accept استیت این ماشین را به عنوان q همراه با خود M به ماشین L می دهیم. اگر L بگوید که q استیتی است که ماشین M هرگز وارد آن نمی شود، چون q استیت accept این ماشین است عملا این زبان هیچ رشته ای را قبول نمی کند پس زبان L می تواند یک تصمیم گیرنده برای Empty باشد که این تناقض دارد.

5. زبان زیر را در نظر بگیرید(در هر دو بخش {0,1} = ∑):

$$L1 = \{< M > \mid where\ M\ is\ a\ TM\ and\ L(M) \neq \emptyset\ and\ each\ string\ in\ M's\ language\ has\ prefix\ 101\}$$

- الف) ثابت کنید زبان زیر turing recognizable **نیست**. (10 نمره)

حال زبان L2 به شکل زیر را در نظر بگیرید:

$$L2 = \{< M1, M2 > \mid where\ M1\ and\ M2\ are\ turing\ machines\ and\ L(M1) \subseteq L(M2)\}$$

- ب) با استفاده از $L1 \leq L2$ ثابت کنید L2 هم **تشخیص ناپذیر است**.(10 نمره)

الف) با استفاده از کاهش $\overline{A_{TM}}$ به L1 مسئله را اثبات میکنیم. فرض کنید L1 تشخیص پذیر باشد در آن صورت می توان ماشین M2 را به این شکل میسازیم:

اگر x که ورودی ماشین تورینگ M2 است دارای prefix 101 نباشد آن را میپذیرد و در غیر این صورت M را روی w ران می کند. دقت کنید که M و w به صورت encode شده روی نوار ماشین M2 هستند و میتواند از آن ها استفاده کند ولی ورودی ماشین M2 همان x است. حال با این توضیحات می توان نشان داد که اگر L1 رد کند یعنی ماشین تورینگ $\overline{A_{TM}}$ تشخیص پذیر می شود.

ب) ورودی L1 یعنی M را میگیریم. حال یک ماشین تورینگ M2 به این صورت می سازیم که زبان آن $101\sum^*$ باشد

در این صورت اگر M و M2 را به L2 بدهیم تشخیص میدهد که آیا M زیرمجموعه M2 است یا نه. اگر باشد یعنی M هم دارای prefix 101 است و در نتیجه L1 نیز تشخیص پذیر می شود و این با فرض تناقض دارد.

نکته ای که وجود دارد این است که M میتواند تهی باشد و از آنجایی که تهی زیرمجموعه هر زبانی است، راه حل اشتباه می شود. از بابت این اشتباه عذرخواهی میکنم، و اگر این موضوع را نوشته باشید هم نمره این بخش به شما تعلق می گیرد.

6. ثابت کنید زبان زیر تصمیم پذیر است:(10 نمره)

$$L \ = \ \{< A, B > \ | \, A \; and \; B \; are \; DFA \; and \; L(A) \ \subseteq \ L(B)\}$$

اگر $L(A) \ \subseteq \ L(B)$ آنگاه $L(A) \ \cap \ \overline{L(B)} \ = \ \emptyset$. می دانیم که زبان های regular نسبت به complement و اشتراک بسته هستند.

با استفاده از DFA های A و B، یک DFA به نام C که برابر $A \ \cap \ \overline{B}$ است میسازیم. حالا می توانیم چک کنیم که آیا زبان C تهی است یا نه. اگر تهی بود قبول میکنیم در غیر این صورت رد میکنیم.

7. تابع busy beaver به صورت $BB\colon \ \mathcal{N} \longrightarrow \mathcal{N}$ تعریف می شود. به ازای هر مقدار از k، تمام ماشین تورینگ های k-استیتی را در نظر بگیرید که با شروع از یک نوار خالی، در نهایت halt می کنند. مقدار $BB(k)$ حداکثر تعداد یک های است که پس از halt کردن بر روی نوار در بین تمام ماشین تورینگ های گفته شده باقی می ماند. اثبات کنید این تابع قابل محاسبه نمی باشد(برای تمام ماشین تورینگ های این مسئله، زبان نوار را به صورت $\Gamma \ = \ \{0, 1, \sqcup\}$ فرض کنید)(10 نمره امتیازی)

نشان می دهیم که اگر این تابع قابل محاسبه باشد، آنگاه ATM نیز decidable خواهد بود. فرض می کنیم تابع $BB(k)$ توسط ماشین F محاسبه می شود. ماشین S را برای ATM به صورت زیر می سازیم:

ماشین S:

به ازای ورودی <M,w> :

- **ماشین M_wرا روی الفبای نوار Γ به صورت زیر می سازیم:**

به ازای هر ورودی:

1. ماشین **M** را روی ورودی **w** شبیه سازی کن. در این حین تعداد استیت هایی که در حین شبیه سازی رد می شوند را ذخیره کن.

2. اگر **M** محاسباتش تمام شد یعنی **halt** کرد، به تعداد استیت هایی که در مرحله قبل برای شبیه سازی شمردیم، عدد یک روی نوار بنویس.

- با استفاده از ماشین **F** مقدار $BB(k) = b$ را بدست می آوریم. **k** در اینجا تعداد استیت های M_wاست.
- **M** را به ازای **w** به اندازه **b** قدم اجرا کن.
- در صورتی که **M** با این تعداد قدم اجرا ورودی را قبول کرد، ما نیز به **accepting** استیت میرویم. در غیر اینصورت به یک **rejecting** استیت می رویم.

در صورتی که **M** ورودی **w**را قبول کند، آنگاه M_w به تعداد قدمهای اجرای **M** روی نوار، یک مینویسد. علاوه بر آن، طبق تعریف **BB** می دانیم که مقدار **b** از تعداد یک هایی که روی نوار نوشتیم بیش تر یا برابر آن است (چون از بین تمام ماشین تورینگ هایی که به اندازه M_w استیت دارند، **b** نشان دهنده بیشترین تعداد یک های روی نوار این ماشین ها است. تعداد یک های روی نوار در انتهای محاسبات نیز قطعا از تعداد مراحل اجرای یک ماشین تورینگ بیشتر نخواهد بود). بنابراین **S** ماشین **M** را به اندازه کافی اجرا می کند تا متوجه شود که ورودی را قبول می کند و خودش نیز ورودی را قبول کند. اگر **M** ورودی را قبول نکند، **S** نیز هیچگاه قبول کردن **M** را ندیده و ورودی را رد می کند.