به نام خدا

نظریه زبان‌ها و ماشین‌ها- بهار ۱۴۰۱

تمرین شماره ۱۱

دستیار آموزشی این مجموعه: معین کرمی
moein2000n@gmail.com

تاریخ تحویل : چهارشنبه 1401/10/07

- برای حل سوالات، می‌توانید از undecidable بودن زبان‌های A_{TM} , ALL_{CFG} و زبان شامل تمام ماشین تورینگ ها با زبان تهی استفاده کنید.

1. برای زبان های زیر decidable, undecidable و یا unrecognizable بودن را مشخص کنید: (۲۰)
   a. A_{TM+} = {(T, w} | T is a Turing machine that accepts w in more than 1000 steps}

این زبان تصمیم ناپذیر است، برای اثبات این موضوع A_{TM}را به این زبان کاهش می‌دهیم. ورودی A_{TM} را به صورت (T, w) در نظر بگیرید، ماشین 'T را دقیقا مانند T میسازیم با این تفاوت که هر وقت به accept state رسیدیم، ۱۰۰۰ مرحله روی نوار حرکت می‌کنیم و سپس رشته را قبول می‌کنیم. حال (T', w) را به A_{TM+} می‌دهیم، اگر reject شود یعنی T نیز w را نمی‌پذیرد چون اگر می‌پذیرفت، حداقل ۱۰۰۰ حرکت دیگر انجام می‌داد و 'T نیز w را می‌پذیرفت و اگر accept کند واضح است که T نیز w را می‌پذیرد. پس داریم : $A_{TM} \leq_m A_{TM+}$ پس این زبان تصمیم ناپذیر است.

   b. A_{odd} = {(T, w) | T is a Turing machine that accepts w in an odd number of steps)}

این زبان نیز تصمیم ناپذیر است. برای اثبات این موضوع A_{TM}را به این زبان کاهش می‌دهیم. فرض کنید ماشین M را A_{odd} را تصمیم می‌گیرد. ورودی A_{TM} را به صورت (T, w) در نظر بگیرید، ابتدا همین ورودی را مستقیم به M می‌دهیم، اگر accept شد یعنی ماشین T کلمه w را قبول می‌کند. در غیر این صورت ماشین 'T را دقیقا مانند T میسازیم با این تفاوت که استیت‌های accept همه از حالت accept خارج و با یک یال به یک استیت جدید accept می‌روند. ماشین 'T زبانی برابر با ماشین T دارد، فقط accept کردن آن یک مرحله بیشتر از T طول می‌کشد. حال (T', w) را به M می‌دهیم، اگر accept شد یعنی T نیز w را میپذیرد، در غیر این صورت یعنی T نه در زوج و نه در فرد مرحله w را نمی‌پذیرد.

پس داریم : $A_{TM} \leq_m A_{odd}$ پس این زبان تصمیم ناپذیر است.

2. برای هر یک از زبان‌های زیر مشخص کنید آیا می‌توان با استفاده از قضیه رایس، تصمیم پذیر بودن یا نبودن آن را مشخص کرد؟ علت پاسخ خود را توضیح دهید. (۱۵)
   a. Right-T = {(T) | T is a Turing machine that never moves right 6 consecutive times on the tape}
   b. Right-T+ = {(T) | T is a Turing machine that L(T) = Right-T}

توجه کنید که نکته مهم در قضیه رایس این است که هر ویژگی nontrivial در مورد ((زبان))شناسایی شده توسط ماشین تورینگ غیر قابل تصمیم است، در نتیجه مهم ((زبان)) ماشین ترینگ است نه رفتارهای هد و ویژگی خود ماشین.

با توجه به توضیحات بالا استفاده از قضیه رایس در زبان Right-T ممکن نیست چون در این زبان رفتار ماشین تورینگ بررسی می‌شود، ولی می‌توان از این قضیه برای زبان +Right-T استفاده کرد چون در این زبان، زبان ماشین تورینگ مطرح است، نه رفتار آن.

3. تصمیم ناپذیری زبان زیر را با کاهش زبان تهی بودن زبان ماشین تورینگ به آن اثبات کنید: (۱۰)

CMP = {(T, T’) |T and T’ are Turing machines which L(T) = L(T’)}

برای اثبات تصمیم ناپذیری این زبان، زبان A_E (تمام ماشین تورینگ‌ها با زبان تهی) را به این زبان کاهش می‌دهیم.

ماشین تورینگ متناظر زبان بالا را T_C می‌نامیم. فرض کنید می‌خواهیم مشخص کنیم آیا زبان ماشین T تهی است یا خیر.

همچنین ماشین تورینگ T_{NULL} را طوری طراحی میکنیم که زبان آن تهی باشد.

حال ورودی (T, T_{NULL}) را به عنوان ورودی به T_C می‌دهیم، اگر accept شد یعنی زبان ماشین T تهی بوده و ما نیز نتیجه را اکسپت می‌کنیم، در غیر این صورت reject می‌کنیم.

در نتیجه داریم $A_E \leq_m CMP$ پس این زبان تصمیم ناپذیر است.

4. یک گرامر مستقل از متن را هنگام ((دقیق)) می‌گوییم که هیچ قانون را نتوان در آن حذف کرد. یک قانون هنگامی قابل حذف کردن است که با حذف آن زبان تولید شده توسط گرامر هیچ تغییری نکند. زبان زیر را در نظر بگیرید:

LC = {(G) | یک گرامر مستقل از متن دقیق باشد G}

آیا زبان LC تصمیم پذیر است؟ ادعای خود را اثبات کنید. (۲۵) راهنمایی: از ALL_{CFG} استفاده کنید. (زبانی که شامل تمام CFGهایی است که زبان آن‌ها شامل تمام کلمات ممکن است.)

این زبان تصمیم پذیر نیست، برای اثبات این موضوع ALL_{CFG} را به این زبان کاهش می‌دهیم.

فرض کنید ماشین تورینگ M بتواند LC را تصمیم بگیرد.

در ابتدا با استفاده از M اگر در گرامر ورودی قانونی اضافه بود آن را حذف می‌کنیم. برای اینکار هر بار میتوان یک زیر مجموعه از قواعد گرامر را حذف کرد و آن را به M داد.

گرامر جدید دقیق را 'G می‌گذاریم.

اگر قانون R غیر قابل حذف باشد، آنگاه رشته‌ای با طول کمینه هست که در اشتقاق خود از R استفاده می‌کند. این رشته را $\min(R)$ می‌نامیم.

چون 'G دقیق است برای تمامی قواعد R_i آن $\min(R_i)$ وجود دارد.

فرض کنید n بزرگتر از تمامی $\min(R_i)$ ها باشد.

به 'G قواعد زیر را اضافه میکنیم تا به ''G برسیم. فرض کنید $\Sigma = \{a_1, a_2, \ldots, a_m\}$.

$S \to vT \qquad (v \in \Sigma*, |v| = n)$

$T \to \epsilon$

$T \to a_1T \mid a_2T \mid \ \ldots \ \mid a_mT$

قوانین جدید موجب تولید $v\Sigma*$ می‌شوند. گرامر جدید ''G را به M می‌دهیم.

اگر قانونی در ''G الزامی نباشد:

- در قوانین 'G نمی‌تواند باشد، چون $|v| = n$ و رشته‌های تولید شده با رشته های $\min(R_i)$ تداخل ندارند.
- در قوانین جدید ''G است.

اگر MIN_{CFG} بپذیرد یعنی قوانین جدید ''G الزامی بوده‌اند و G اوولیه نمی‌توانسته $\Sigma*$ را پوشش بدهد.

5. فرض کنید زبان L شامل تمام ماشین تورینگ‌های دو نواره است که حداقل به ازای یک ورودی، یک کاراکتر غیر blank روی نوار دوم خود می‌نویسند. تصمیم پذیری یا تصمیم ناپذیری این زبان را مشخص کنید. (۱۵)

L = {⟨M⟩| M is a two-tape Turing machine that writes a nonblank symbol on its second tape when it is run on some input}

برای اثبات تصمیم ناپذیری زبان L، زبان A_{TM} را به آن کاهش می‌دهیم.

فرض کنید ماشین تورینگ T، زبان L را تصمیم می‌گیرد.

ورودی (M, w) را در نظر بگیید. ماشین تورینگ دو نواره 'T را به این صورت می‌سازیم:

1. ماشین 'Tبه ازای هر ورودی کلمه ی w را روی M و فقط با استفاده از نوار اول شبیه سازی می‌کند.
2. اگر M کلمه ی w را قبول کرد، 'T یک کاراکتر nonblank روی نوار دوم می‌نویسد.

حال 'T را به عنوان ورودی به T می‌دهیم.

اگر T ورودی را قبول کرد یعنی ماشین M کلمه‌ی w را قبول می‌کند و اگر رد کرد یعنی M نیز w را رد می‌کند.

پس اثبات کردیم $A_{TM} \leq_m L$ پس این زبان تصمیم ناپذیر است.

6. تصمیم پذیر بودن یا نبودن زبان زیر را مشخص کنید.(۱۵)

L = {⟨T⟩ | T is a Turing machine that never halts on a blank cell of tape}

این زبان تصمیم پذیر نیست، برای اثبات این موضوع A_{TM} را به این زبان کاهش می‌دهیم.

فرض کنید ماشین M زبان L را تصمیم می‌گیرد.

ورودی (T, w) را در نظر بگیرید، ماشین تورینگ 'T را به صورت زیر از روی این ورودی می‌سازیم:

1. اگر ورودی 'T برابر w نبود، روی یک خانه نوار blank می‌نویسد و روی همان خانه halt می‌کند.
2. اگر ورودی 'T برابر w بود، w را روی T شبیه سازی می‌کند، اگر accept شد روی یک خانه یک کاراکتر nonblank می‌نویسد و روی همان خانه halt می‌کند، در غیر این صورت روی یک خانه از نوار blank می‌نویسد و روی همان خانه هالت می‌کند.

حال 'T را به M می‌دهیم، اگر اکسپت شد یعنی T کلمه w را می‌پذیرد و اگر reject شد یعنی T کلمه w را نمی‌پذیرد.
پس اثبات کردیم $A_{TM} \leq_m L$ پس L تصمیم ناپذیر است.

7. فرض کنید مسئله‌ی PCP را به این صورت کاهش میدهیم که فقط یک نوع کاراکتر داشته باشیم (به عنوان مثال فقط کاراکتر a) آیا در این حالت نیز این مسئله باز هم undecidable خواهد بود؟(امتیازی ۱۰)

این مسئله دیگر تصمیم ناپذیر نخواهد بود. با حالت بندی زیر می‌توان جواب مسئله را پیدا کرد.

حالت ۱) بلاک A در همه ی دومینوها کوتاه تر و یا بلند تر از B باشد. در این صورت واضح است که مسئله جواب ندارد.

حالت ۲) هر دو بلاک یک دومینه هم سایز باشند: در این حالت همین تک دومینو یک جواب است.

حالت ۳)در بعضی دومینوها بلاک اول و در بقیه، بلاک دوم بلندتر است. فرض کنید در دومینه ای به نام X طول بلاک A به اندازه x حرف و در دومینه‌ای به نام Y طول بلاک B به اندازه ی y حرف طولانی تر باشد. کنار هم چیدن x تا دومینو Y و y تا دومینو X یک پاسخ است.