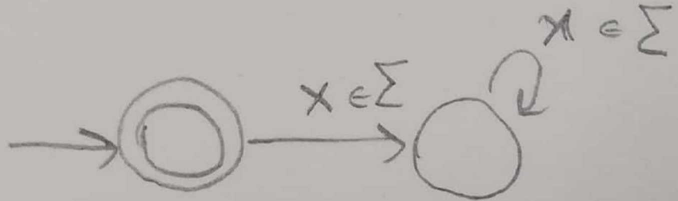
علیرضا کریمی

810101492

۱)

الف) لزوماً این حرف درست نیست که اگر loop بزنیم، می‌توانیم بی‌نهایت رشته را بپذیریم. ممکن است loop بین state هایی باشد که هیچ کدام به accepting state ختم نشوند و یا حتی آن state ها reachable نباشند.

به عنوان مثال DFA روبرو را در نظر بگیرید.



با آنکه loop داریم اما هیچگاه با رسیدن به loop، نمی‌توان بعد از آن به accepting state رسید. و در واقع این DFA تنها ε را می‌پذیرد.

۱) ب) از برهان خلف استفاده می‌کنیم و فرض می‌کنیم که $Finite_{TM}$ تصمیم‌پذیر است و از آن طریق اثبات می‌کنیم که A_{TM} نیز تصمیم‌پذیر خواهد بود (که خلاف واقع است) در واقع A_{TM} را به $Finite_{TM}$ کاهش می‌دهیم.

فرض کنید که یک ماشین M' داریم که این ماشین، به ازای هر رشته ورودی x، w را روی M، run می‌کند و در صورتی که پذیرفته شد، رشته x را نمی‌پذیرد و بالعکس. پس اگر M، w را بپذیرد، $L(M')$ تهی خواهد شد چون هیچ رشته x پذیرفته شده و در غیر اینصورت $L(M')$، شامل تمام رشته‌ها خواهد بود و تعداد رشته‌های آن نامتناهی است. حال کافی است ماشین $Finite_{TM}$، M' را گرفته (ماشین تورینگ modified شده) و تصمیم بگیرد که متناهی است یا نامتناهی. اگر تعداد رشته‌ها متناهی بود (در این حالت کلا رشته‌ای نداشتیم و $L(M')$ تهی بود پس تعداد رشته‌های آن متناهی است)، پس M، w را می‌پذیرد و بالعکس. در واقع

M accepts w if and only if $L(M') = \emptyset$ (M' is finite)

iff $\langle M, w \rangle \notin A_{TM} \Leftrightarrow L(M')$ contains all strings and is infinite

پس از آنکه $Finite_{TM}$ تصمیم‌پذیر است، می‌توان تصمیم گرفت که M، w را می‌پذیرد یا نه و A_{TM} تصمیم‌پذیر می‌شود که خلاف واقع است

پس فرض خلف باطل و حکم ثابت می‌شود که $Finite_{TM}$ تصمیم‌ناپذیر است

ج) فرض کنید A_{TM} تصمیم‌پذیر است (فرض خلف) می‌خواهیم با کاهش A_{TM} به $Finite_{TM}$ ثابت کنیم اگر A_{TM} decidable باشد $Finite_{TM}$ هم تصمیم‌پذیر شده (که خلاف واقع است). حال ماشین تورینگ R را در نظر بگیرید که قرار است نقش $Finite_{TM}$ را بازی کند، یعنی یک ماشین تورینگ M می‌گیرد و می‌گوید که $L(M)$، $Finite_{TM}$ است یا نه. حال کافی است که R را به عنوان ماشین تورینگ ورودی A_{TM} و $\langle M \rangle$ (منظور همان description انکود شده ماشین تورینگی است که به R می‌دهیم تا بفهمیم متناهی است یا نه) را به عنوان w به ماشین تورینگ A_{TM} می‌دهیم. حال اگر $\langle M \rangle$ توسط R پذیرفته شود، یعنی M، Finite است و درغیراینصورت، M، Finite نخواهد بود. درواقع با تصمیم‌پذیر بودن A_{TM}، تصمیم گرفتیم که M، متناهی است یا نه، که این خلاف واقع است چون می‌دانیم $Finite_{TM}$ تصمیم‌ناپذیر است، پس فرض خلف باطل و حکم ثابت می‌شود که A_{TM} تصمیم‌ناپذیر است.

2) خب باید PCP را به Palindrome کاهش دهیم. در واقع باید مسئله مربوط به PCP را به گونه‌ای تغییر دهیم تا به مسئله مربوط به Palindrome تبدیل شود. فرض می‌کنیم Palindrome تصمیم پذیر است (فرض خلف) و با کمک آن به این نتیجه می‌رسیم که PCP نیز باید تصمیم پذیر باشد (که خلاف واقع است). برای اینکار فرض کنید دومینوهای ما در مسئله PCP به این صورت هستند:

$x_1 x_2 \dots x_n$
$y_1 y_2 \dots y_m$

حال یک content free language درست می‌کنیم و به ازای هر دومینو، دو قانون به آن اضافه می‌کنیم (رشته بالا در سمت راست و معکوس رشته پایین در سمت چپ S قرار دارد)

$$S \to x_1 x_2 \dots x_n S y_m y_{m-1} \dots y_2 y_1 \mid x_1 x_2 \dots x_n y_m y_{m-1}$$

(در واقع به یاد داریم که CFG مربوط به Palindrome به شکل $S \to xSx \mid x \mid \varepsilon$ است و به گونه‌ای از آن الهام گرفتیم)

حال که از روی تمام دومینوها، قوانین را ساختیم، آن را به ماشین Palindrome می‌دهیم تا چک کند که حداقل یک رشته به صورت Palindrome دارد یا نه. اگر داشت یعنی دومینو PCP معادل آن وجود دارد (ممکن است این مراحل را برگردیم به عقب تا از روی قوانین، دومینوهای PCP را پیدا کنیم) و اگر نداشت، یعنی چنین دومینوهایی وجود ندارد که بالا و پایین یکی شوند و reject می‌شود. پس PCP به این گونه تصمیم پذیر خواهد شد که این خلاف واقع است پس فرض خلف باطل و حکم ثابت می‌شود که Palindrome تصمیم ناپذیر است.

3)

الف) خیر؛ چون این موضوع یک property مربوط به زبان آن ماشین تورینگ است، صرفا مربوط به عملکرد read و رفتار ماشین تورینگ است. درصورتی که ما باید دنبال یک ویژگی non-trivial مربوط به زبان باشیم که زبان آن ماشین‌ها و متمم آن هیچ کدام تهی نشوند.

ب) بله؛ می‌توان از قضیه Rice استفاده کرد و تصمیم ناپذیر خواهند بود چون زبان هایی وجود دارند که حداکثر 3 رشته را بپذیرند و متمم آنها هم حداقل ۴ رشته را دارد و هیچ کدام از این دو تهی نیستند.
در واقع باید مجموعه زبان L را بتوان به واسطه آن ویژگی غیربدیهی (non trivial) به دو زیرمجموعه ناتهی افراز کرد. ویژگی‌ها مربوط به زبان ماشین تورینگ است و ما مجموعه‌ها از زبان های صحبت می‌کنیم که برای ما قابل تشخیص هستند.

$L_1=\{a\}$
$L_2=\{a,b\}$
$L_3=\{a,b,c\}$

ادامه ۳) درواقع در مسئله ب: می توان گفت که زبان های مثل L_1 و L_2 و L_3 و ... ویژگی صفت سوال را دارند و زبان های مثل $\bar{L_1}$ ، $\bar{L_2}$ و $\bar{L_3}$ و ... ویژگی صفت سوال را ندارند. درواقع یک سری زبان ها عضو مجموعه با ویژگی P و یک سری دیگر عضو مجموعه با ویژگی $\bar{P}$ هستند و هیچ کدام از آنها تهی نیستند. اگر این شروط برقرار بود، از Rice می توان استفاده کرد و گفت که تصمیم ناپذیر است

۴) از فرض خلف استفاده می‌کنیم. فرض می‌کنیم زبان L تصمیم‌پذیر است و بعد به کمک آن مسئله $Empty_{TM}$ را حل کرده و تصمیم‌پذیر می‌شود (که خلاف واقع است). در واقع $Empty_{TM}$ را به L کاهش می‌دهیم. یعنی مسئله مربوط به $Empty_{TM}$ را (که یک ماشین تورینگ به عنوان ورودی می‌گیرد و تشخیص می‌دهد زبان آن تهی است یا نه) را به مسئله مربوط به L کاهش می‌دهیم با ایجاد تغییراتی. حال از روی ماشین تورینگ M (که به عنوان ورودی $Empty_{TM}$ داریم)، ماشین تورینگ M' را می‌سازیم. به این شکل که از تمام استیت‌های accept، یک ε-transition می‌زنیم به یک state جدید به اسم q_f و آن را accept می‌کنیم و بقیه استیت‌ها را از حالت accept خارج می‌کنیم. به این شکل تنها یک state داریم که accept است. حال ماشین تورینگ M' را به همراه q_f (استیت accept) می‌دهیم به L.

فرض کردیم L تصمیم‌پذیر است. اگر L، accept کند یعنی قطعا به q_f رسیدیم، پس قطعا رشته‌ای داریم که به accept می‌رسد و پس زبان M تهی نبوده است. اگر L ریجکت کند، یعنی هیچ وقت به q_f نرسیدیم پس هیچ رشته‌ای در M وجود ندارد. پس زبان M، تهی است. پس توانستیم تهی یا ناتهی بودن M را تصمیم بگیریم در حالی که می‌دانیم $Empty_{TM}$ تصمیم‌ناپذیر است. پس فرض خلف باطل و حکم ثابت می‌شود که L تصمیم‌ناپذیر است.

(5

الف) از کاهش دادن $\overline{A_{TM}}$ به L_1 استفاده می‌کنیم و فرض خلف می‌زنیم که L_1، recognizable است و در نتیجه $\overline{A_{TM}}$، recognizable خواهد شد (که خلاف واقع است). برای این کار باید روی ورودی شرط بگذاریم.

اگر ورودی ما به صورت $x = 101(1+0)^*$ بود accept کن. در غیر این صورت w را روی M، run کن و اگر accept شد، accept کن و بالعکس. حال این یعنی اگر w توسط M پذیرفته نشود، زبان جدید ساخته شده فقط شامل رشته با prefix 101 است و در غیر این صورت تمام رشته‌ها x را شامل می‌شود (یعنی به صورت $x = 101(1+0)^*$ است فقط)

و شامل رشته‌هایی مثل $x = 100$ و رشته‌هایی که prefix برابر 101 ندارند هم است. حال تورینگ ماشین جدید M' با Language ساخته شده جدید را به L_1 می‌دهیم. اگر L_1، accept کند، پس w روی M، run نشده و عضو $\overline{A_{TM}}$ است، در غیر این صورت، w روی M، run شده و عضو A_{TM} نیست

$\langle M, w\rangle \notin A_{TM} = \langle M, w\rangle \in \overline{A_{TM}} \iff L_1$ accept M', $L(M')$ has prefix 101

پس با recognizable بودن L_1، توانستیم $\overline{A_{TM}}$ را recognizable کنیم که خلاف واقع است. پس فرض خلف باطل و حکم ثابت می‌شود که L_1، unrecognizable است.

(راحت‌تر بود که روی $x = 101$ حالت ببندیم ولی این هم درسته احتمالاً)

ب) L_1 را به L_2 کاهش می‌دهیم و فرض می‌کنیم که L_2 تشخیص‌پذیر است (فرض خلف) و به این می‌رسیم که L_1 هم باید تشخیص‌پذیر باشد (که خلاف واقع است). می‌دانیم به عنوان ورودی یک ماشین تورینگ M_1 داریم (مربوط به L_1 است). حال ماشین تورینگ M' را می‌سازیم که تمام رشته‌ها به فرمت $101(1+0)^*$ در زبان آن است و همه را می‌پذیرد (یک DFA ساده است). حال M_1 و M' را به عنوان ورودی می‌دهیم به ماشین تورینگ مربوط به L_2. اگر accept شد که یعنی تمام رشته‌های M_1 به صورت $101(1+0)^*$ است و بالعکس. اما این کافی نیست و باید چک کنیم که زبان مربوط به M_1 تهی نباشد. یعنی فقط کافی است که یک رشته به فرمت $101(1+0)^*$ در $L(M_1)$ باشد. ادامه صفحه بعد.

ادامه 5 ب) حال تمام رشته‌ها به فرمت $(1+0)^*101$ را به ترتیب تک تک روی M_1 run می‌کنیم، در صورتی که حداقل یک رشته به فرمت $(1+0)^*101$ وجود داشته باشد، قطعاً به آن خواهیم رسید و M_1 آن را می‌پذیرد و accept می‌شود (یعنی زبان M_1 ناتهی است.) و در صورتی که وجود نداشته باشد، یا در loop گیر کردن تمام یا اگر M_1 ساده باشد، reject می‌کنیم و راحت تشخیص می‌دهیم که $(1+0)^*101$ ندارد). حال در صورتی که هم ناتهی بودن زبان M_1 برقرار بود و هم $L(M_1) \in L(M')$ بود (که هر دو تشخیص‌پذیرند، و دو زبان تشخیص‌پذیر تحت عمل اشتراک بسته‌اند،) ما نیز می‌پذیریم و در غیر این صورت یا reject می‌کنیم یا در loop ادامه دهیم

پس به این شکل می‌توانیم L_1 را تشخیص دهیم. در صورتی که می‌دانیم L_1 تشخیص‌ناپذیر بود.

پس فرض خلف باطل و حکم ثابت می‌شود که L_2 تشخیص‌ناپذیر است.

6) ابتدا DFA مربوط به $A \cap B$ را می‌سازیم و C می‌نامیم. (DFA ها تحت عمل اشتراک بسته هستند و الگوریتم ساختن DFA برای $A \cap B$ در کلاس تدریس شده) حال در صورتی که A با C ($A \cap B$) برابر باشد یعنی $A \subset B$ خواهد بود. پس کافی است برابر بودن دو DFA، A و C ($A \cap B$) را چک کنیم. در واقع مسئله چک کردن $A \subset B$ را کاهش دادیم به مسئله برابری دو DFA، A و $A \cap B$. می‌دانیم این مسئله طبق مباحثی که تدریس شد، تصمیم‌پذیر است. پس زبان L نیز تصمیم‌پذیر است

$$L = \{\langle A, B \rangle \mid A \text{ and } B \text{ are DFA and } L(A) \subseteq L(B)\}$$

(فرض خلف)

7) ببینید، از کاهش A_{TM} به BB استفاده می‌کنیم و می‌گوییم اگر BB قابل محاسبه باشد، A_{TM} تصمیم‌پذیر خواهد (که خلاف واقع است.) ببینید ماشین تورینگ M، k تا استیت دارد و می‌دانیم که اگر w را روی M، run کنیم و اجرای آن به پایان برسد و ما به تعداد مراحل اجرا (step‌های اجرا) روی tape خالی، 1 بنویسیم، این تعداد 1 از تعداد 1 های نوشته شده توسط BB(K) کمتر مساوی است. (چون M هم زیرمجموعه‌ای از ماشین‌های تورینگ halt شده k استیتی است و قطعا ماشین تورینگی شبیه آن وجود دارد که به ازای هر step، یک بنویسد روی tape و حتی ماشین تورینگی ممکن است وجود داشته باشد که تعداد step‌های بیشتری از اجرای w روی M طی کند و به ازای هر کدام از step‌ها هم 1 بنویسد. پس تعداد 1 های نوشته شده از مراحل اجرای w روی M روی tape از تعداد حداکثر 1 های نوشته شده روی tape که توسط BB(K) قابل محاسبه است، کمتر مساوی است) این یعنی اینکه اگر قرار باشد که w روی M، run شده و halt کرده و accept شود، قطعا این اتفاق در کمتر مساوی BB(K) تعداد مرحله رخ می‌دهد و در غیر اینصورت یا اصلا روی M، پذیرش نشده یا متوقف شده است و loop به وجود آمده است. با همه این تفاسیر، کافی است که ماشین تورینگ M با k استیت را حداکثر به اندازه BB(K) استیت، روی w اجرا کنیم، اگر accept شد، A_{TM} هم accept می‌کند و در غیر اینصورت هم reject می‌کنیم (چون تعداد BB(K) مرحله مشخص و محدود است، loop نداریم) پس به این ترتیب، با قابل محاسبه بودن BB(K)، A_{TM} تصمیم‌پذیر خواهد شد که این خلاف واقع است، پس فرض خلف باطل و حکم ثابت می‌شود که BB(K) غیرقابل محاسبه است.

1 دور توکن