ایمان محمدی ــ ۹۹۱۰۲۲۰۷
تمرین ۴

زمان آپلود: ۲۱ خرداد
نیم‌سال دوم ۱۴۰۱ــ۱۴۰۲

سوال ۱.

۱.۱

الف

رای طراحی یک ماشین تورینگ که این زبان را پذیرا باشد، باید توجه داشته باشیم که هر دوامل ۲ ، ۳ ، و ۵ باید بتوانند به n قسمت بشوند. زبان L شامل رشته‌هایی از 'a' است که تعداد 'a' ها بر ۲ ، ۳ ، یا ۵ بخش‌پذیر است.

در زیر یک طرح برای چنین ماشین تورینگی ارائه شده است. این ماشین تورینگ در یک حلقه زمانی برای تعدادی 'a' را بررسی می‌کند که بر ۲، ۳، یا ۵ قابل تقسیم است، و در صورت قابل تقسیم بودن، به حالت قبول می‌رود.

حالت اولیه: $q_۰$

نماد اولیه: a

نماد نهایی: ϕ

State Next	Move	Symbol New	Symbol Current	State Current
q۱	R	X	a	q۰
reject	–	Φ	Φ	q۰
q۲	R	X	a	q۱
accept	–	Φ	Φ	q۱
q۳	R	X	a	q۲
accept	–	Φ	Φ	q۲
q۴	R	X	a	q۳
accept	–	Φ	Φ	q۳
q۵	R	X	a	q۴
accept	–	Φ	Φ	q۴
q۰	R	X	a	q۵
accept	–	Φ	Φ	q۵

این، شامل کلیه ترانزیشن‌های ماشین تورینگ است.

ب

برای طراحی یک ماشین تورینگ برای این زبان، ما نیاز به یک روش دقیق و سیستماتیک داریم که اجازه دهد n تا 'a' و سپس $n^۲$ تا 'b' را بشماریم. در اینجا، ما یک روش نمادین برای این کار ارائه می‌دهیم که با استفاده از علامت‌هایی مثل X و Y ، می‌توانیم تعداد 'a' و 'b' ها را دقیقاً بشماریم. این روش به طور مشروط 'b' ها را با توان دوم تعداد 'a' ها مطابقت می‌دهد.

ماشین تورینگ به شرح زیر است: حالت اولیه: $q_۰$

نماد اولیه: a

نماد نهایی: ϕ

اینجا، ما نشان داده‌ایم که چگونه 'a' ها را با 'X' و 'b' ها را با 'Y' جایگزین کنیم، سپس باقی مانده 'a' و 'b' ها را بررسی کنیم تا مطمئن شویم که تعداد 'b' ها دقیقاً توان دوم تعداد 'a' ها است.

حالت فعلی	نماد فعلی	نماد جدید	حرکت	حالت بعدی
q۰	a	X	R	q۱
q۰	Φ	Φ	–	accept
q۱	a	a	R	q۱
q۱	b	Y	L	q۲
q۱	Φ	Φ	–	reject
q۲	a	a	L	q۲
q۲	X	X	R	q۳
q۲	Φ	Φ	–	reject
q۳	Y	Y	R	q۳
q۳	b	Y	R	q۴
q۳	Φ	Φ	–	reject
q۴	b	Y	L	q۵
q۴	Φ	Φ	L	q۵
q۵	Y	Y	L	q۵
q۵	a	a	L	q۵
q۵	X	X	R	q۰
q۵	Φ	Φ	–	reject

این ماشین تورینگ به طور ریتمیک 'a' ها و 'b' ها را بررسی می‌کند، و هر بار یک 'a' را با یک 'X' جایگزین می‌کند و سپس به دنبال تعداد مناسبی از 'b' ها (که با 'Y' جایگزین شده‌اند) می‌گردد. اگر همه‌ی 'a' و 'b' ها درست باشند، ماشین تورینگ به حالت قبول می‌رود. در غیر این صورت، اگر تعداد 'b' ها دقیقاً برابر با توان دوم تعداد 'a' ها نباشد، ماشین تورینگ به حالت رد می‌رود.

سوال ۲.

۲.۱

فرض کنیم که زبان D تصمیم پذیر است و M یک ماشین تورینگ است که D را تصمیم می‌گیرد.

برای تشخیص پذیر بودن زبان C، ما می‌توانیم یک ماشین تورینگ N را به گونه‌ای بسازیم که به ازای ورودی x، تمامی رشته‌های y را می‌سازد و بررسی می‌کند که آیا (x, y) در D هست یا نه. اگر چنین رشته‌ای وجود داشت، N x را قبول می‌کند. در غیر این صورت، N برای همیشه ادامه می‌یابد.

بنابراین، اگر D تصمیم پذیر باشد، C تشخیص پذیر است.

حال فرض کنیم که C تشخیص پذیر است و N ماشین تورینگی است که C را تشخیص می‌دهد.

برای تصمیم پذیر بودن D، ما می‌توانیم یک ماشین تورینگ M را بسازیم که ورودی‌های (x, y) را می‌گیرد و بررسی می‌کند که آیا x در C قرار دارد. اگر بله، $M(x, y)$ را قبول می‌کند. اگر نه، $M(x, y)$ را رد می‌کند.

بنابراین، اگر C تشخیص پذیر باشد، D تصمیم پذیر است.

پس، زبان C تشخیص پذیر است اگر و تنها اگر زبان D تصمیم پذیر باشد.

۲.۲

بیایید هر دو جهت این ادعا را بررسی کنیم.

اگر L تورینگ-تشخیص‌پذیر باشد، L* نیز تورینگ-تشخیص‌پذیر است. در این جهت، اگر ماشین تورینگ M وجود داشته باشد که L را تشخیص می‌دهد، می‌توانیم یک ماشین تورینگ N بسازیم که L* را تشخیص می‌دهد. N به ازای هر ورودی w، همه‌ی تقسیم‌بندی‌های ممکن w را به رشته‌های w۱، w۲، ...، wn بررسی می‌کند و سپس هر کدام از این رشته‌ها را به M ورودی می‌دهد. اگر M همه‌ی این رشته‌ها را قبول کند، N w را قبول می‌کند. اگر نه، N برای تقسیم‌بندی بعدی ادامه می‌دهد. اگر هیچ یک از تقسیم‌بندی‌ها نتواند w را به رشته‌های قابل قبول تقسیم کند، N w را رد می‌کند. بنابراین، اگر L تورینگ-تشخیص‌پذیر باشد، L* نیز تورینگ-تشخیص‌پذیر است.

اگر L* تورینگ-تشخیص‌پذیر باشد، L نیز تورینگ-تشخیص‌پذیر است. در این جهت، اگر ماشین تورینگ N وجود داشته باشد که L* را تشخیص می‌دهد، می‌توانیم یک ماشین تورینگ M بسازیم که L را تشخیص می‌دهد. M به ازای هر ورودی w، w w را به N ورودی می‌دهد. اگر N w را قبول کند و w یک رشته از L است (یعنی فقط شامل یک عنصر از L* است)، M w را قبول می‌کند. در غیر این صورت، M w را رد می‌کند. بنابراین، اگر L* تورینگ-تشخیص‌پذیر باشد، L نیز تورینگ-تشخیص‌پذیر است.

پس، زبان L تورینگ-تشخیص‌پذیر است اگر و تنها اگر زبان L* تورینگ-تشخیص‌پذیر باشد.

۲.۳

برای اینکه بتوانیم ثابت کنیم که یک زبان تصمیم‌پذیر است، باید بتوانیم یک الگوریتم یا ماشین تورینگ را ارائه دهیم که بتواند به ازای هر ورودی مشخص کند که آیا ورودی در زبان قرار دارد یا خیر.

در این مورد، زبان L شامل تمامی DFA هایی است که همه رشته‌های متعلق به زبان تعریف شده توسط آن DFA برابر با معکوس خود هستند. برای تشخیص اینکه آیا یک DFA برای یک ورودی خاص دارای این خاصیت است یا خیر، می‌توانیم الگوریتم زیر را اجرا کنیم:

برای هر وضعیت q در D، و برای هر کاراکتر a در الفبای D، اگر انتقال از q با a به q' منتهی می‌شود، و q' وضعیت پایانی است، آنگاه q باید یک وضعیت شروع باشد و همچنین برعکس.

اگر همه وضعیت‌های D این خاصیت را داشته باشند، آنگاه D را قبول کن. در غیر این صورت، آن را رد کن.

این الگوریتم در زمان چندجمله‌ای اجرا می‌شود، زیرا تنها باید برای هر وضعیت و هر کاراکتر در الفبا، یک بررسی انجام دهد. بنابراین، این الگوریتم یک الگوریتم تصمیم‌گیری برای L است، و L تصمیم‌پذیر است.

۲.۴

برای اثبات این موضوع، می‌توانیم یک الگوریتم ساده ارائه دهیم که به صورت گرافی PDA را بررسی می‌کند و تشخیص می‌دهد آیا حالت‌های غیرقابل دسترس وجود دارد یا خیر. الگوریتم به شرح زیر است:

ابتدا یک گراف G را با استفاده از PDA ایجاد می‌کنیم. در این گراف، هر حالت یک راس است و یک یال از راس i به راس j وجود دارد اگر و تنها اگر یک انتقال از حالت i به حالت j در PDA وجود داشته باشد.

از حالت اولیه PDA شروع کرده و یک جستجوی عمق اول (DFS) یا جستجوی سطح اول (BFS) را اجرا کنید تا تمام حالت‌های قابل دسترس را پیدا کنید.

در پایان، اگر هر حالتی در PDA را می‌توان با جستجویی که در مرحله ۲ انجام داده‌ایم، دست یافت، PDA هیچ حالت غیرقابل دسترسی ندارد و بنابراین ورودی را رد می‌کنیم. اگر حالتی وجود داشته باشد که نمی‌توان به آن دست یافت، PDA حداقل یک حالت غیرقابل دسترس دارد و بنابراین ورودی را می‌پذیریم.

این الگوریتم در زمان چند جمله‌ای اجرا می‌شود (زیرا برای هر یال و راس در گراف G باید تنها یک عملیات انجام دهیم)، و بنابراین زبان L تصمیم‌پذیر است.

سوال ۳.

۳.۱.۱

یک ماشین تصمیم‌گیرنده را می‌توان با یک رشته بایتی که وضعیت‌ها، الفبا، تابع انتقال، وضعیت اولیه و مجموعه وضعیت‌های پایانی را توصیف می‌کند، نشان داد. برای هر ماشین تصمیم‌گیرنده مختلف، یک رشته بایتی متمایز وجود دارد که آن را توصیف می‌کند، و برعکس، برای هر رشته بایتی مختلف، یک ماشین تصمیم‌گیرنده متمایز وجود دارد.

با این حال، تعداد رشته‌های بایتی متمایز شمارا است، چرا که هر رشته بایتی می‌تواند هر طول طبیعی را داشته باشد و برای هر بایت در رشته، ۲۵۶ انتخاب مختلف وجود دارد (از ۰ تا ۲۵۵). بنابراین، تعداد ماشین‌های تصمیم‌گیرنده نیز شمارا است.

۳.۱.۲

$$L = M : M \text{ is a Turing decider such that } M \notin L(M)$$

زبانی است که شامل تمام ماشین‌های تورینگ است که خودشان را نمی‌پذیرند. این زبان برابر با مشهورترین مثال از یک زبان ناتصمیم‌پذیر است، مشهور به مسئله رازنیک هالینگ.

در این زمینه، فرض کنید که M یک ماشین تورینگ است که L را تصمیم می‌گیرد. اکنون دو حالت ممکن است:

$M \in L.$

این بدین معناست که M خود را نمی‌پذیرد، که با فرض اینکه M یک تصمیم‌گیر برای L است تناقض دارد.

$M \notin L.$

این بدین معناست که M خود را می‌پذیرد، که باز هم با فرض اینکه M یک تصمیم‌گیر برای L است تناقض دارد.

بنابراین، هیچ ماشین تورینگی وجود ندارد که بتواند L را تصمیم بگیرد، پس L ناتصمیم‌پذیر است.

۳.۲

یکی از مهم‌ترین زبان‌هایی که این خاصیت را دارد، مسئله‌ی رازنیک هالینگ (Halting Problem) است. این مسئله به شکل زیر تعریف می‌شود:

H = M,w : M is a Turing machine that halts on input w

این زبان ناتصمیم‌پذیر است، اما هر زبان تصمیم‌پذیر دیگر به آن کاهش می‌یابد. برای نشان دادن این موضوع، فرض کنید که A یک زبان تورینگ‌ـ تصمیم‌پذیر دلخواه باشد. پس ماشین تورینگ M وجود دارد که A را تصمیم می‌گیرد. اکنون می‌توانیم تابع کاهش f را به شکل زیر تعریف کنیم:

f(x) = M,x

این تابع برای هر ورودی x ، یک زوج ماشین تورینگ و ورودی به ما می‌دهد. اکنون، x در A است اگر و تنها اگر M برای ورودی x متوقف شود. بنابراین، x در A است اگر و تنها اگر f(x) در H باشد، که این نشان‌دهنده‌ی این است که A به H کاهش می‌یابد.

اگر زبان A به زبان L کاهش یابد و L تورینگ‌ـ تصمیم‌پذیر باشد، آنگاه A نیز تورینگ‌ـ تصمیم‌پذیر است. این را می‌توان با استفاده از تابع کاهش f نشان داد: اگر M یک ماشین تورینگ باشد که L را تصمیم می‌گیرد، پس ما می‌توانیم ماشین تورینگ M' را تعریف کنیم که برای هر ورودی x ، ابتدا f(x) را محاسبه کرده و سپس M را بر روی نتیجه اجرا کند. اگر M برای f(x) متوقف شود، پس $M'x$ را می‌پذیرد. در غیر این صورت، $M'x$ را رد می‌کند. بنابراین، اگر A به L کاهش یابد و L تورینگ‌ـ تصمیم‌پذیر باشد، آنگاه A نیز تورینگ‌ـ تصمیم‌پذیر است.

۳.۳

با رسیدن به تناقض بدست می‌آوریم.

فرض کنید T تشخیص‌پذیر است. اگر اینطور باشد، می‌توانیم یک ماشین تورینگ D وجود داشته باشد که برای یک ورودی w، اگر w توصیف‌کننده یک ماشین تورینگ طلایی باشد، D w را بپذیرد. در غیر این صورت، D ، w را رد می‌کند.

حالا بیایید ماشین تورینگ D' را تعریف کنیم به این شکل که برای یک ورودی w، اگر D w را بپذیرد، D' w را رد می‌کند و برعکس.

اکنون به نظر می‌رسد D' یک ماشین تورینگ طلایی باشد، چون D' یک ورودی w را پذیرفته یا رد می‌کند بر اساس اینکه آیا D w را پذیرفته یا رد می‌کند. با این حال، اگر D' یک ماشین تورینگ طلایی باشد، آنگاه D باید D' را بپذیرد، که باعث می‌شود D' D' را رد کند. این با فرض اینکه D' یک ماشین تورینگ طلایی است، تناقض است. بنابراین، T نمی‌تواند تشخیص‌پذیر باشد.

موفق باشید.