۴۰۳۱۰۶۶۸۱

Subject: Date: محمدمهدی مرادی

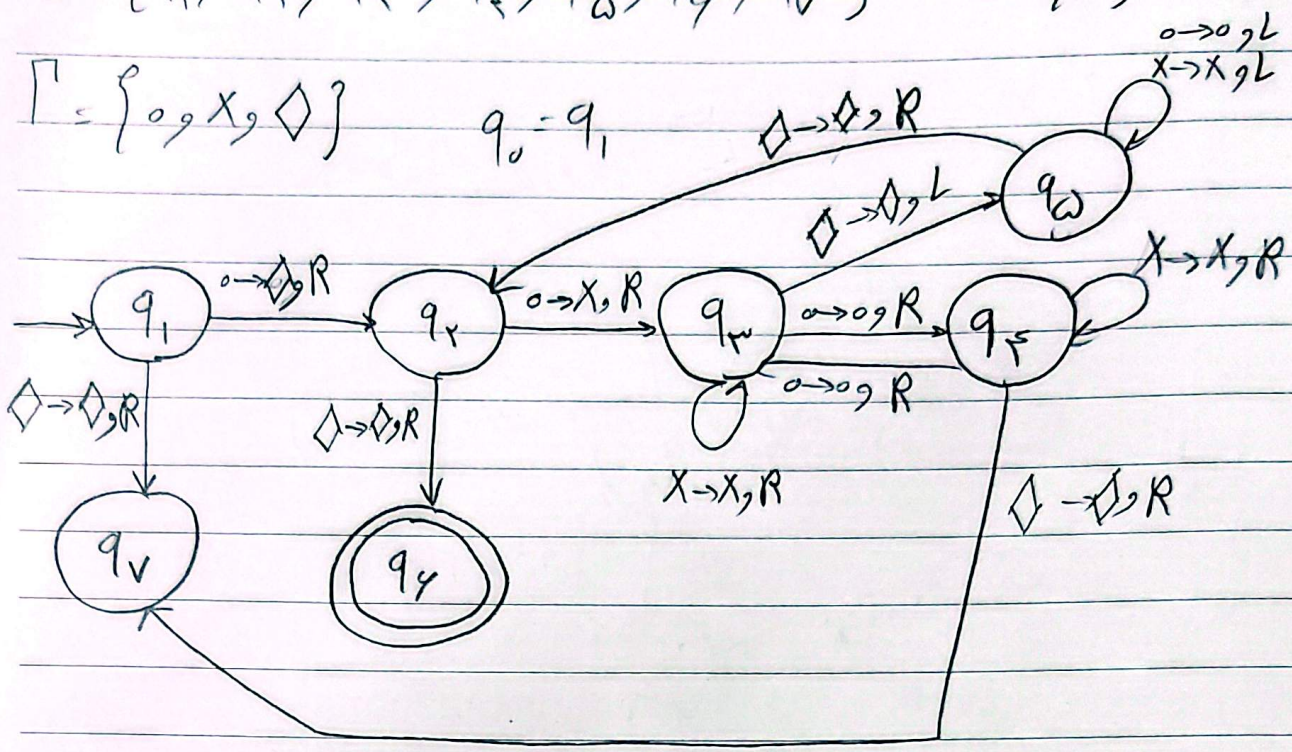
رده اول

۱- الف)

$M = (Q, \Sigma, \Gamma, \delta, q_0, \Diamond, F)$ $F = q_6$

$Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$ $\Sigma = \{0\}$

$\Gamma = \{0, X, \Diamond\}$ $q_0 = q_1$



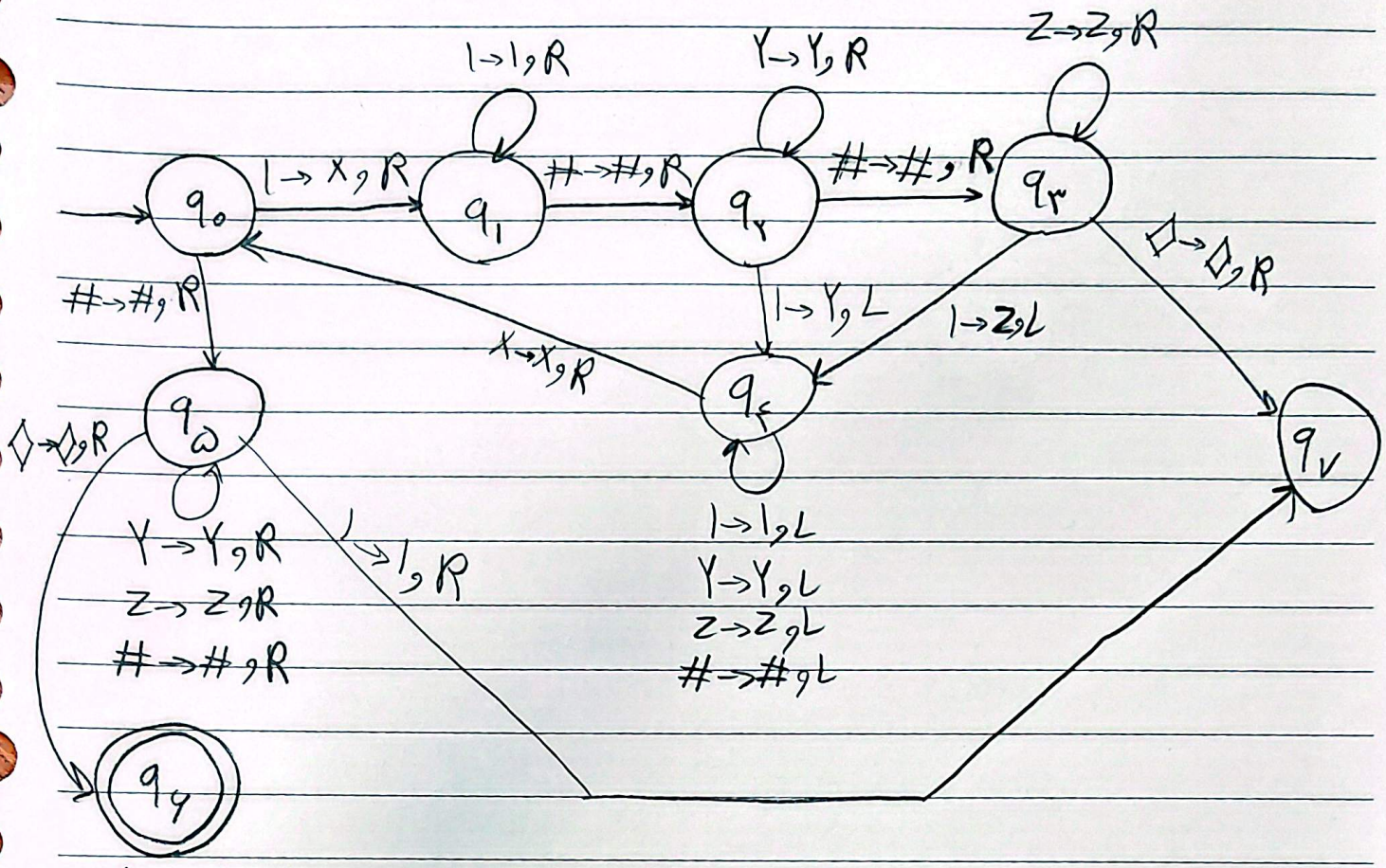
در ماشین تورینگ M ما ابتدا چک می‌کنیم که نوار خالی است (fail) سپس چک می‌کنیم آیا یک عدد صفر وجود دارد یا نه (accept) سپس یکی در میان ۰ ها را لا می‌کنیم و تا ته نوار می‌رویم. وقتی به ته نوار رسیدیم به سر برمی‌گردیم چون هر دفعه تعداد صفرها نصف شد، وقتی به سر نوار برگشتیم و $\Diamond$ بود accept می‌شود

Subject: Date:

$M = (Q, \Sigma, \Gamma, \delta, q_0, \Diamond, F)$ (۲

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$ $\Sigma = \{1, \#\}$

$\Gamma = \{1, \#, X, Y, Z, \Diamond\}$ $q_0 = q_0$ $F = q_6$



در ماشین M عملا ورودی به فرم $a\#y\#z$ هستش و سعی می‌کنیم که هر یک دونه یا از a را با یک دونه یک از y یا z خط بزنیم. اگر تمام a به X تبدیل شد و سمت راستش یک وجود نداشت accept است

DAT

Subject: Date:

ریه دوم

۵- الف) یک TM در هر لحظه ۳ مولفه دارد: state فعلی، موقعیت head و محتوای نوار پس با کمک اصل ضرب تمامی حالات را در می آوریم

$$N = \underbrace{|Q|}_{\text{تعداد حالت های ماشین متناهی TM}} \times \underbrace{k}_{\text{تعداد موقعیت head}} \times \underbrace{|\Gamma|^k}_{\text{حالات ممکن محتوای نوار}}$$

یک ماشین تصمیم گیر D را این گونه می سازیم:

D دو عدد را می شمارد: ۱- تعداد خانه های دیده شده (باید $k \geq$ باشد) $= H$

۲- تعداد مراحل اجرای M (همان TM ورودی) $= E$

اگر $k < H$ بشود به reject می رویم

اگر $N < E$ بشود یعنی حتما یکی از N حالت دوبار تکرار شده یعنی loop بی نهایت داریم پس از k خانه بیشتر استفاده نمی کند و تو loop است پس این را accept می کنیم اگر $N < E$ بشود

اگر هم M متوقف شد قبل از $k < H$ هر که accept می شود

DAT

پس این زبان تصمیم پذیر است

Subject: Date:

ب) برهان خلف: فرض می‌کنیم $EMPTY_{TM}$ تصمیم‌پذیر است

حالا می‌آییم A_{TM} را به $EMPTY_{TM}$ reduce می‌کنیم

ماشین S را برای حل A_{TM} اینگونه می‌سازیم (ورودی: $\langle M, w \rangle$)

یک TM به نام M' می‌سازد. M' رشته ورودی خودش را نادیده می‌گیرد و بجای

آن M را روی w شبیه‌سازی می‌کند اگر M رشته w را بپذیرد، M' نیز

ورودی خودش را می‌پذیرد

S، کد M' را به D (تصمیم‌گیر $EMPTY_{TM}$) می‌دهد. اگر D بگوید که تهی

است یعنی M، w را نپذیرفته و اگر بگوید تهی نیست M، w را پذیرفته

پس A_{TM} تصمیم‌پذیر است ✗ ⟸ $EMPTY_{TM}$ تصمیم‌پذیر نیست

• با استفاده از قضیه رایس هم میشد گفت چون تهی بودن ویژگی نابدیهی است

پس $EMPTY_{TM}$ تصمیم‌پذیر نیست

DAT

CamScanner

Subject: Date:

ج) برهان خلف: فرض می‌کنیم $EQUAL_{TM}$ تصمیم‌پذیر است پس

یک تصمیم‌گیر به نام D دارد. حالا زبان $EMPTY_{TM}$ را به $EQUAL_{TM}$ reduce می‌کنیم

با استفاده از D ماشین S را می‌سازیم:

S ورودی $\langle M \rangle$ را می‌گیرد تا تهی بودن آن را بررسی کند، S در داخل خود

یک TM به اسم M_{Empty} می‌سازد که وقتی ورودی می‌گیرد reject می‌کند

پس $L(M_{Empty}) = \phi$. حالا S وقت $\langle M, M_{Empty} \rangle$ را به D می‌دهد

اگر D بپذیرد یعنی $L(M) = L(M_{Empty}) = \phi$

اگر D نپذیرد یعنی $L(M) \neq \phi$

پس $EMPTY_{TM}$ تصمیم‌پذیر است ✗ $\Leftarrow$ $EQUAL_{TM}$ تصمیم‌پذیر نیست

DAT

Subject: Date:

ز) تصمیم پذیر است

فرض کنیم یک ماشین تصمیم گیرنده D داریم

D ابتدا چک می کند رشته ورودی فرصت یک DFA را دارد یا نه و اگر نداشت ← reject

برای اینکه L(A) نامتناهی باشد باید در گراف A یک loop داشته باشیم که: ۱- از حالت

start بتوانیم به آن برسیم ۲- بعد از loop به حالت accept در گراف A برسیم

پس D ابتدا تمامی مسیرهای دو شرط بالا را ندارند حذف می کند

حال D روی گراف باقی مانده یک DFS می زند و گره های دیده نشده را یک جا save

می کند و اگر دوباره به یک گره ای که از قبل دیده شده برسد پس loop داریم پس D

آن را accept می کند و چون متناهی است تعداد گره های گراف ما اگر DFS تمام

شد و loop پیدا نشد D به reject می رود

پس L تصمیم پذیر است

DAT

Subject: Date:

رده سوم

۳- برهان خلف: P تصمیم پذیر است پس یک ماشین تصمیم گیر D دارد که اگر $\langle M, G \rangle$ را به آن بدهیم منظم بودن اشتراک را در زمان متناهی تصمیم می گیرد

حال A_{TM} را به P reduce می کنیم

ماشین S را اینگونه می سازیم (ورودی $\langle M, w \rangle$):

S یک زبان غیرمنظم را مستقل از متن $0^n 1^n$ را می سازد

$$L(G) = \{0^n 1^n \mid n \geq 0\}$$

S درون خود M' را اینگونه می سازد:

هر ورودی x را نادیده می گیرد. M را روی w شبیه سازی می کند. اگر M رشته w را پذیرفت، M' نیز x را می پذیرد

$\Leftarrow$ پس اگر M رشته w را بپذیرد $\Leftarrow$ $L(M') = \Sigma^*$

$\Rightarrow L(M') \cap L(G) = \{0^n 1^n\} \Rightarrow$ نامنظم

اگر M رشته w را نپذیرد $\Leftarrow$ $L(M') = \phi$

$\Rightarrow L(M') \cap L(G) = \phi \Rightarrow$ منظم

DAT

Subject: Date:

حال S U حفت $\langle M', G \rangle$ را به D می‌دهد

اگر D بگوید انتر اکشان منظم است $\Leftarrow$ M رشته w را نپذیرفته

اگر D بگوید انتر اکشان نامنظم است $\Leftarrow$ M رشته w را پذیرفته

$\Leftarrow$ A_{TM} تصمیم پذیر شد ⁂ $\Leftarrow$ P تصمیم ناپذیر است

DAT

Subject: Date:

رده چهارم

۷- اثبات این را با کمک اسلاید Turing-Variation انجام می دهیم.

برای اینکه اثبات کنیم یک DTM یک بعدی می تواند یک DTM دوبعدی را شبیه سازی کند بدین گونه عمل می کنیم: (برعکسش بدیهی هستش. صرفا از یک بعد DTM دوبعدی استفاده می کنیم)

فرض می کنیم $|w| = k$

w_{k-1}				◊			→ رشته
$k-1$	#	0	#	k	#	0	→ مختصات → x#y

q_0 ⇐ حالت اولیه

$w = w_0 w_1 \cdots w_{k-1}$

برای هر حرکت DTM ۲ بعدی این مراحل را می رویم:

کاراکتر بالا را می خوانیم و کاراکتر جدید را می نویسیم. با توجه به جهت انتخابی (L و یا R و یا D) مختصات جدید را محاسبه می کنیم. ماشین از ابتدای نوار پایین جست جو می کند تا مختصات جدید را پیدا کند اگر پیدا نشد که head روی او نجا می رود اگر پیدا نشد به انتهای نوار می رویم و ◊ را در نوار بالا و مختصات جدید را در نوار پایین می گذاریم

DAT

(ادامه صفحه بعد)

Subject: Date:

وقتی در DTM دوبعدی متوقف می‌شویم باید تمامی کاراکترهایی را که y آنها برابر y_{final} نیست را blank کنیم. سپس تمامی کاراکترها باقی مانده را براساس x شان بدون فاصله می‌چینیم. سپس تمامی نوار دوم را پاک می‌کنیم و head به سمت راست می‌بریم تا به $\diamond$ بعد از راست‌ترین کاراکتر برسد و توقف کند

چون توانستیم این شبیه‌سازی را ارائه کنیم پس DTM یک بعدی با دو بعدی معادل است

Subject: Date:

~~رده پنجم~~

۹- الف) اول $\Leftarrow$ را ثابت می‌کنیم

چون زبان تصمیم پذیر است پس یک تصمیم گیر D وجود دارد که روی هر ورودی

halt می‌کند. تنها شمارشگر E را اینگونه می‌سازیم

E رشته‌ها را به ترتیب گفته شده به D می‌دهد اگر D پذیرفت، E آنرا چاپ می‌کند

چون E رشته‌ها را به ترتیب به D می‌دهد پس خروجی نیز ترتیب گفته شده را دارد

حالا $\Rightarrow$ را ثابت می‌کنیم

یک تنها شمارشگر E داریم که رشته‌ها را به ترتیب متعارف چاپ می‌کند

ماشین تصمیم گیر D را اینگونه می‌سازیم:

D رشته w را می‌گیرد و اگر E آنرا چاپ کرد که D به accept می‌رود

و چون به ترتیب طول چاپ می‌شوند اگر E یک رشته‌ای با طول بزرگتر از w

چاپ کرد یعنی در زبان ما نبوده پس D به reject می‌رود

چون یک تصمیم گیر ساختیم پس زبان ما تصمیم پذیر است

DAT

Subject: Date:

ب) چون تشخیص پذیر است پس یک شمارشگر E_1 داریم

شمارشگر E_2 را اینگونه می‌سازیم:

اولین رشته E_1 را چاپ می‌کند. تا وقتی که رشته‌ای با ترتیب متعارف بزرگتر از اولی از طریق E_1 چاپ نشد، E_2 کاری نمی‌کند و وقتی E_1 یک رشته با ترتیب متعارفی بزرگتر از آخرین رشته چاپ شده توسط E_2 چاپ کند، E_2 نیز آنرا چاپ می‌کند

پس E_2 یک شمارشگر با ترتیب متعارفی می‌شود و چون زبان اصلی نامتناهی است. پس E_1 تا ابد رشته جدید تولید می‌کند پس E_2 تا ابد می‌تواند دنبال رشته با ترتیب متعارفی بزرگتر از آخرین رشته چاپ شده توسط خودش بگردد

پس E_2 نامتناهی است و یک زیرمجموعه نامتناهی از زبان اصلی را جدا کرد که طبق قسمت الف می‌دانیم این زیرمجموعه نامتناهی، تصمیم پذیر است

DAT