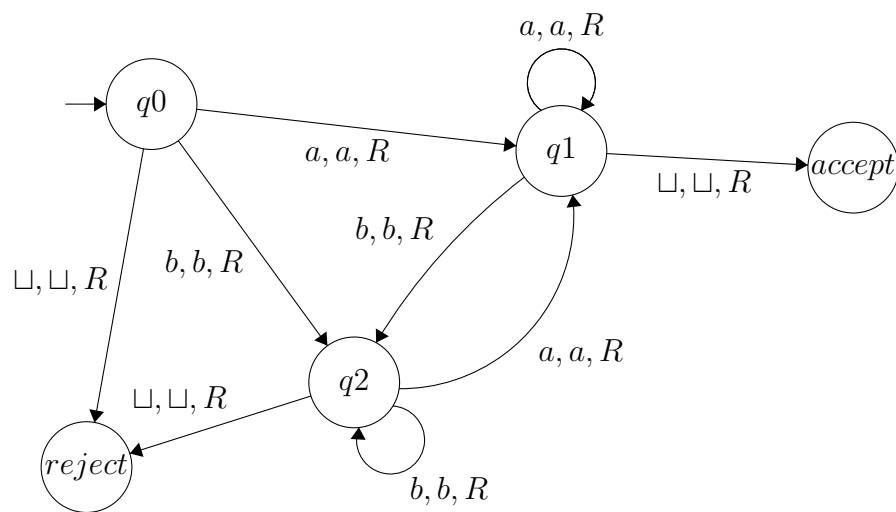


تکلیف سری چهارم

مبانی نظریه محاسبه
دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی. ترم ۴۰۳۲

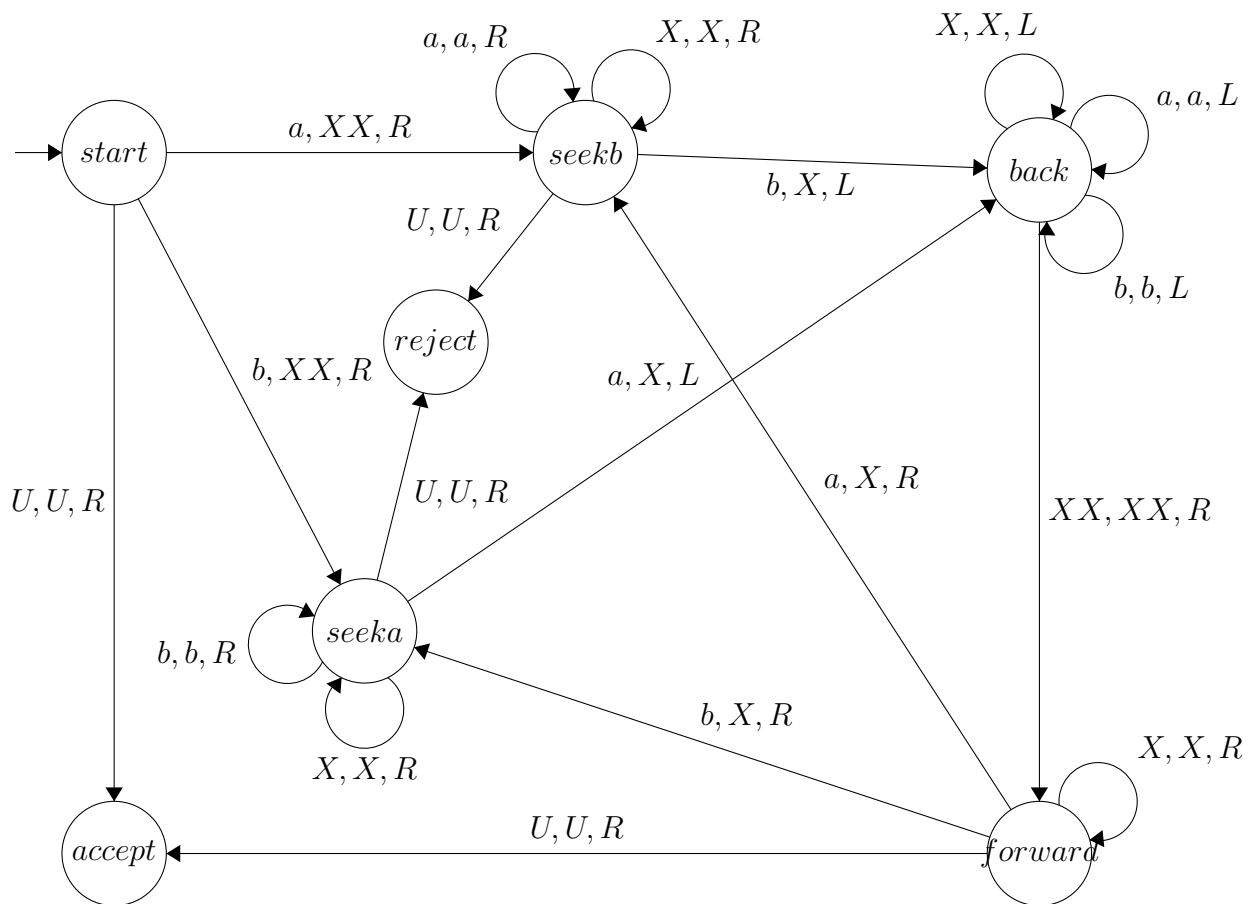
۱. برای زبانهای زیر یک ماشین تورینگ با استفاده از دیاگرام ارائه کنید.

$$A = (a + b)^* a$$



$$B = \{w \in (a + b)^* \mid n_a(w) = n_b(w)\}$$

ایده ماشین: در یک رفت و برگشت کاراکتر a با b تطبیق داده می شود و با X جایگزین می شوند (خط می خورند). همین روند ادامه پیدا می کند تا اینکه در دور جدید کاراکتر دیگری غیر از X نباشد. در این حالت ماشین به accept می رود. ماشین در حالتی به reject می رود که یک a یا b را خط زده ولی کاراکتر مخالفش را پیدا نمی کند تا با آن تطبیق دهد. برای اینکه انتهای سمت چپ مشخص شود، به جای X از کاراکتر XX استفاده شده.



۲. برای زبانهای زیر یک ماشین تورینگ ارائه کنید. توصیف سطح بالا کافی است. می‌توانید از چند نوار استفاده کنید به شرطی که تعداد نوارها $O(1)$ باشد.

$$P = \{a^n \mid n \text{ یک عدد اول است}\}$$

ایده ماشین: ماشین تقسیم بر اعداد 2 تا $n-1$ را انجام می‌دهد و هر تقسیمی باقیمانده اش صفر شد به reject می‌رود، در غیر اینصورت به accept می‌رود. از دو نوار استفاده می‌کنیم. فرض کنید می‌خواهیم تقسیم بر k را چک کنیم. فرض بر این است در نوار دوم k کاراکتر نوشته شده است. دقت کنید چون از تقسیم بر 2 شروع می‌کنیم، کافی است در پایان مرحله قبلی به انتهای نوار دوم یک کاراکتر جدید اضافه کنیم. حال چک می‌کنیم که آیا تعداد کاراکترها در نوار اول مضربی از تعداد کاراکترها در نوار دوم است یا نه. این کار را با ترفند خط زدن و رفت و برگشت روی نوار دوم انجام می‌دهیم.

$$F = \{a^n \mid n \text{ یک عدد فیبوناچی است}\}$$

ایده ماشین: به ترتیب اعداد فیبوناچی کمتر یا مساوی n را تولید می‌کنیم و اگر یک عدد فیبوناچی مساوی n یافت شد ماشین به accept می‌رود در غیر اینصورت به reject می‌رود. از سه نوار استفاده می‌کنیم. روی نوار اول ورودی نوشته شده. محتوای آن را تغییر نمی‌دهیم. ابتدا روی نوار دوم صفر کاراکتر و روی نوار سوم یک کاراکتر درج می‌کنیم. در شروع هر مرحله چک می‌کنیم آیا طول ورودی برابر با طول نوار

دوم است یا نه. اگر آری به پذیرش می‌رویم. اگر طول ورودی کمتر بود به reject می‌رویم. اگر طول ورودی بیشتر بود ادامه می‌دهیم. مجموع نوار دوم و سوم را به نوار دوم منتقل می‌کنیم. سپس محتوای نوار دوم و سوم را عوض می‌کنیم و به مرحله جدید می‌رویم.

۳. جدول زیر تابع تغییر وضعیت δ یک ماشین تورینگ را توصیف می‌کند. q_0 وضعیت شروع است. اینجا q_2 وضعیت پذیرش است. کاراکتر B اینجا کاراکتر فضای خالی است.

$$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \{q_2\}) :$$

δ	0	1	B
q_0	$\{(q_0, 1, R)\}$	$\{(q_1, 0, R)\}$	\emptyset
q_1	$\{(q_1, 0, R), (q_0, 0, L)\}$	$\{(q_1, 1, R), (q_0, 1, L)\}$	$\{(q_2, B, R)\}$
q_2	\emptyset	\emptyset	\emptyset

الف) آیا ماشین تورینگ بالا یک ماشین قطعی (معین) است؟ چرا؟

خیر. چون برای مثال در وضعیت q_1 با دیدن 0 ماشین دو گزینه برای حرکت بعدی دارد.

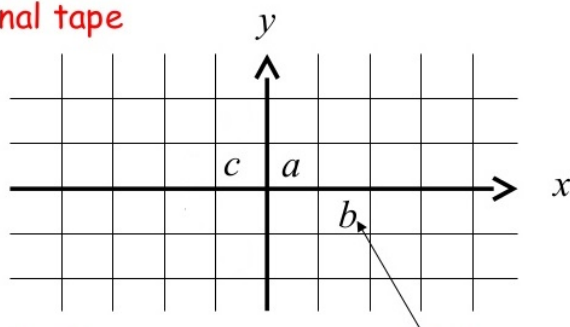
ب) کدامیک از این رشته‌ها توسط این ماشین پذیرفته می‌شود؟

01, 10, 0101, 011

ج) همه پیکربندی‌هایی که با رشته ورودی 011 قابل دسترسی است را بنویسید.

۴. ماشین تورینگ دو بعدی همانند ماشین تورینگ استاندارد است با این تفاوت که نوار آن یک گرید دو بعدی نامتناهی است و نوک خواندن و نوشتن می‌تواند در جهت‌های راست، چپ، بالا و پایین حرکت کند. توضیح دهید ماشین تورینگ استاندارد چگونه می‌تواند ماشین تورینگ دو بعدی را شبیه‌سازی کند.

2-dimensional tape



MOVES: L,R,U,D

U: up D: down

HEAD

Position: +2, -1

می‌توان این کار را با دو نوار نامتناهی از هر دو طرف شبیه‌سازی کرد. یک نوار مانند محور x هاست و نوار دیگر مانند محور y ها. نوار متناهی از هر دو طرف را هم قبلاً در کلاس توضیح داده‌ایم که چگونه با یک نوار نامتناهی یکطرفه شبیه‌سازی کرد.

۵. نشان دهید زبان $A\epsilon_{CFG}$ که معادل با همه گرامرهای مستقل از متنی است که رشته ϵ را می‌پذیرند، تصمیم پذیر است.

$$A\epsilon_{CFG} = \{\langle G \rangle \mid G \text{ یک گرامر مستقل از متن است و رشته تهی را تولید می‌کند}\}$$

گرامر G را تبدیل به فرم نرمال چامسکی می‌کنیم. اگر در گرامر حاصل، قانون $S \rightarrow \epsilon$ موجود باشد، رشته ϵ توسط گرامر G تولید می‌شود در غیر اینصورت تولید نمی‌شود.

۶. نشان دهید زبان زیر تصمیم پذیر نیست.

$$A\epsilon_{TM} = \{\langle M \rangle \mid M \text{ یک ماشین تورینگ است و رشته تهی را می‌پذیرد}\}$$

مشابه این تمرین در کلاس حل شده. جلسه ۲۵ سوال اول را ببینید.

۷. نشان دهید زبان زیر تصمیم پذیر نیست.

$$HALT' = \{\langle M \rangle \mid M \text{ روی آن متوقف می‌شود}\}$$

می‌توانیم مسئله $HALT$ را به این مسئله تقلیل دهیم. ورودی مسئله $HALT$ ماشین M و رشته w است. ماشینی M' را می‌سازیم که با گرفتن هر رشته‌ای در حلقه بینهایت بیفتد غیر از رشته w . اگر ورودی رشته w باشد، ماشین M' ماشین M را روی w اجرا می‌کند. توجه کنید ماشین M' ویژگی مورد نظر ما را دارد. با فرض اینکه الگوریتمی برای $HALT'$ وجود داشته باشد، این الگوریتم می‌تواند مسئله $HALT$ را حل کند.