تکلیف سری دوم

درس مبانی نظریه محاسبه
دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی. بهار ۱۴۰۰

۱. نشان دهید هر dfa برای زبان A حداقل k وضعیت دارد.

$$A = \{w \in \{a\}^* \mid |w| < k\}$$

اثبات مشابه جواب تمرین آخر تکلیف اول است.

۲. نشان دهید زبانهای زیر منظم نیستند.

(آ)

$$B = \{a^n b^{n-1} \mid n > 0\}$$

اثبات با لم تزریق برای زبانهای منظم. فرض کنید B منظم باشد لذا لم تزریق در مورد آن صادق است. پس عدد p وجود بطوریکه برای رشته w به طول حداقل p در زبان B، رشته w را می‌توان به سه قسمت x و y و z تقسیم کرد طوریکه گزاره‌های سه گانه لم تزریق در مورد آن برقرار باشد. برای تناقض رشته $w = a^p b^{p-1}$ را در نظر بگیرید. واضح است طول رشته حداقل p است و w عضو زبان است. چون در تقسیم بندی باید داشته باشیم $|xy| \leq p$ و $w = xyz$ پس قسمت y کاملا در قسمت a قرار میگیرد که تزریق آن باعث می‌شود رشته از زبان خارج شود.

(ب)

$$C = \{w \in \{a, b\}^* \mid w = w^R\}$$

اثبات مانند سوال قبل. اینجا رشته $w = a^p b^{2p} a^p$ را برای تناقض انتخاب می‌کنیم.

(ج)

$$D = \{w \in \{a\}^* \mid \exists n, |w| = n^2\}$$

مانند موارد بالا از لم تزریق استفاده می‌کنیم. رشته a^{p^2} را در نظر بگیرید. فرض کنید طول y برابر با q باشد. باید همه اعداد زیر مربع کامل باشند.

$$p^2 - q, p^2, p^2 + q, p^2 + 2q, \ldots$$

این امکانپذیر نیست (چرا؟)

۳. نشان دهید زبانهای منظم تحت عملگرهای زیر بسته هستند.

(آ) رشته‌هایی که از کنار هم قرار دادن تناوبی رشته‌های (بطول یکسان) دو زبان A و B بدست می‌آید.

$$PerfectShuffle(A, B) = \{w = a_1 b_1 \ldots a_k b_k \mid a_1 \ldots a_k \in A \text{ and } b_1 \ldots b_k \in B\}$$

چون A و B منظم هستند پس ماشینهای متناهی M_A و M_B برای این دو زبان وجود دارند. برای اینکه نشان دهیم زبان $PerfectShuffle(A, B)$ هم منظم است، نشان می‌دهیم با استفاده

از M_A و M_B ،می‌توان یک ماشین متناهی برای $PerfectShuffle(A,B)$ ساخت. ایده این کار ساده است. فرض کنید رشته ورودی $x = x_1x_2\ldots x_{2k-1}x_{2k}$ باشد. (دقت کنید همه رشته‌های داخل $PerfectShuffle(A,B)$ طول زوج دارند.) حرف x_1 را به ماشین M_A می‌دهیم و حرف x_2 را به ماشین M_B می‌دهیم. حرف x_3 را به M_A و حرف x_4 را به M_B می‌دهیم. به همین روال، بین ماشینها سویچ می‌کنیم و حروف را یک در میان به دو ماشین می‌دهیم. در انتها اگر هر دو ماشین در وضعیت پذیرش قرار گیرند، رشته x پذیرفته است، در غیر اینصورت پذیرفته نیست.

برای اینکه این ایده ساده را بصورت یک ماشین واقعی بیان کنیم، ماشین M_P را از روی ماشینهای M_A و M_B بدین صورت می‌سازیم. فرض کنید $M_A = (Q_A, \Sigma, \delta_A, q_0^A, F_A)$ و $M_B = (Q_B, \Sigma, \delta_B, q_0^B, F_B)$ باشد. تقریبا ایده مشابه ایده ساخت ماشینی برای $A \cap B$ است. وضعیتهای ماشین M_P بصورت $Q_A \times Q_B \times \{0,1\}$ است. وضعیت شروع $(q_0^A, q_0^B, 0)$ است. ماشین وقتی در وضعیتی مثل $(z, z', 0)$ است یعنی حرکت بعدی را ماشین A انجام می‌دهد. اگر ماشین در وضعیتی مانند $(z, z', 1)$ باشد یعنی حرکت بعدی را ماشین B انجام می‌دهد. تابع تغییر وضعیت بصورت زیر تعریف می‌شود.

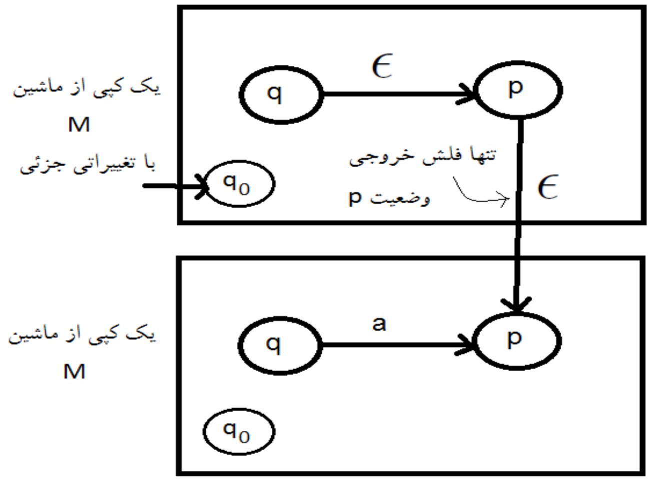
$$\delta((z,z',0),a) = (\delta_A(z,a), z', 1) \qquad \delta((z,z',1),a) = (z, \delta_B(z',a), 0)$$

وضعیت پذیرش همه وضعیتهایی مثل $(z, z', 0)$ است بطوریکه z وضعیت پذیرش ماشین M_A و z' وضعیت پذیرش ماشین M_B باشد.

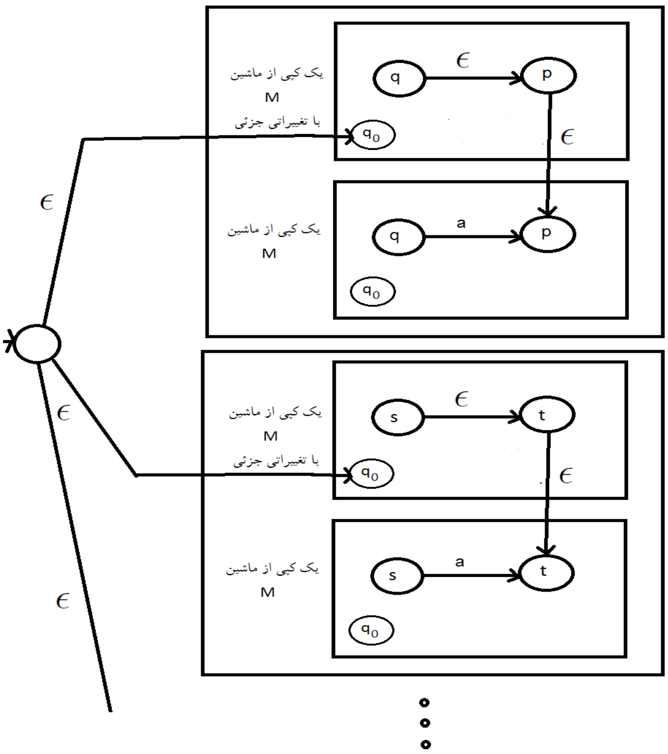
(ب) رشته‌های که از حذف یک حرف رشته‌های زبان A بدست می‌آید.

$$Drop(A) = \{w = xz \mid xyz \in A, x \in \Sigma^*, z \in \Sigma^*, y \in \Sigma\}$$

چون A منظم است ماشین متناهی M برای A وجود دارد. با اسفاده از M یک ماشین برای $Drop(A)$ می‌سازیم. یک ایده این است که برچسب فلش از وضعیت q به وضعیت p را به ϵ تغییر دهیم. اما چند مشکل وجود دارد. اول اینکه رشته $w \in A$ ممکن است چند بار از فلش q به p عبور کند. این باعث می‌شود که چندین حرف از رشته w حذف شود. ما می‌خواهیم فقط یک حرف حذف شود. برای حل این مشکل یک کپی دیگر از ماشین M اضافه می‌کنیم. مانند شکل زیر عمل میکنیم. در کپی اصلی، برچسب فلش از q به p را به ϵ تغییر می‌دهیم. همه فلشهای خروجی p را برمی‌داریم. تنها یک فلش ϵ از p به کپی دیگر ماشین M قرار می‌دهیم (به وضعیت p در کپی دیگر). نکته اینجاست. وقتی ماشین از فلش ϵ بالایی عبور کرد و وارد وضعیت p شد، از آنجا به کپی دیگر ماشین وارد می‌شود و دیگر هیچ فلش ϵ دیگری را تجربه نخواهد کرد. بدین ترتیب فقط یک حرف حذف می‌شود.



مشکل دوم اینجاست که ما باید این کار را برای همه فلشهای ماشین M انجام دهیم. پس برای هر فلش $s \to t$ یک ماشین مثل شکل بالایی ایجاد می‌کنیم و ساختار زیر را ایجاد می‌کنیم.



۴. زبان گرامرهای زیر را توصیف کنید.

$$G_1 : S \to 0S0 \mid 1S1 \mid 1S0 \mid 0S1 \mid \epsilon$$

$$(\Sigma\Sigma)^* \qquad \Sigma = \{0, 1\}$$

$$G_2 : S \to 1S0 \mid 0S1 \mid \epsilon$$

$$\{w\overline{w}^R \mid w \in \{0,1\}^*\}$$

اینجا $\overline{w}$ متمم رشته w می‌باشد.

$$G_3 : S \to 0S0 \mid 1S1 \mid \epsilon$$

$$\{ww^R \mid w \in \{0,1\}^*\}$$

۵. برای هر کدام از زبانهای زیر یک گرامر مستقل از متن ارائه کنید.

(آ)

$$C = \{w \in \{0,1\}^* \mid n_1(w) \geq 3\}$$

$$S \to 000A \mid 001A \mid 010A \mid 011A \mid 100A \mid 101A \mid 110A \mid 111A$$
$$A \to 1A \mid 0A \mid \epsilon$$

(ب) رشته‌هایی از الفبای $\Sigma = \{a,b\}$ که ab ندارند

$$S \to aA \mid bB \mid \epsilon$$
$$A \to aA \mid \epsilon$$
$$B \to bB \mid A$$

(ج) رشته‌هایی از الفبای $\{a,b\}$ که در آنها تعداد a دو برابر تعداد b است.

$$S \to SS \mid aaSb \mid bSaa \mid aSbSa \mid \epsilon$$

(د) متمم زبان $\{a^nb^n \mid n \geq 0\}$

فرض کنید زبان مورد نظر L باشد. داریم

$$L = \Sigma^* - \{a^nb^n \mid n \geq 0\}$$

$$L = \underbrace{\{a^ib^j \mid i > j\}}_{A} \cup \underbrace{\{a^ib^j \mid i < j\}}_{B} \cup \underbrace{\Sigma^* ba \Sigma^*}_{C}$$

برای هر کدام از زبانهای A و B و C بطور جداگانه یک گرامر می‌نویسیم. فرض کنید A و B و C متغیر شروع گرامرهای مورد نظر باشد. قوانین این سه گرامر را با هم ترکیب کرده و قانون $S \to A \mid B \mid C$ را اضافه می‌کنیم.