به نام خدا

نظریه زبان‌ها و ماشین‌ها- بهار ۱۴۰۳

تمرین شماره ۷

دستیار آموزشی این مجموعه: مجید فریدفر

majid.faridfar@gmail.com

تاریخ تحویل : ۲۳ اردیبهشت

---

۱) با استفاده از لم تزریق (pumping lemma)، نشان دهید که زبان‌های زیر مستقل از متن نیستند.

a) $\{w \in \{a,b\}* \mid w = a^n b^m a^n, n \geq m\}$

راه‌حل:

اگر p حداقل طول رشته برای لم تزریق باشد، رشته $w = a^p b^p a^p$ را انتخاب می‌کنیم. اگر هرکدام از v یا y شامل هم a و هم b باشند، با تزریق آنها فرم رشته ($a^* b^* a^*$) به هم می‌ریزد. پس هم v و هم y باید تنها شامل یک کاراکتر باشند. اگر در w هر یک از سه قسمت a و b و a را 1 و 2 و 3 شماره گذاری کنیم:

(v,y)	i	رشته خارج زبان
(1,1)	$i = 2$	تعداد a های اول رشته بیشتر از a های آخر رشته خواهد بود.
(2,2)	i بزرگ	با تزریق به تعداد کافی بالاخره تعداد b های وسط از a های 2 طرف بیشتر می‌شود.
(3,3)	$i = 2$	تعداد a های آخر رشته بیشتر از a های اول رشته خواهد بود.
(1,2)	$i = 2$	اگر طول v بیشتر از صفر است، با تزریق تعداد a های اول رشته بیشتر از a های آخر رشته خواهد شد.
	i بزرگ	اگر طول y بیشتر از صفر است، با تزریق به تعداد کافی بالاخره تعداد b های وسط از a های 2 طرف بیشتر می‌شود.
(2,3)	i بزرگ	اگر طول v بیشتر از صفر است، با تزریق به تعداد کافی بالاخره تعداد b های وسط از a های 2 طرف بیشتر می‌شود.
	$i = 2$	اگر طول y بیشتر از صفر است، با تزریق تعداد a های آخر رشته بیشتر از a های اول رشته خواهد شد.
(1,3)	-	از آنجایی که $\lvert vxy \rvert \leq p$ این حالت رخ نمی‌دهد.

b) $\{w \in \{a,b\}* \mid w = a^n b^m,\ n \text{ and } m \text{ are prime numbers}\}$

راه‌حل:

اگر p حداقل طول رشته لم تزریق باشد، رشته $a^n b^n$ را انتخاب میکنیم که n اولین عدد اول بزرگتر مساوی $p+2$ باشد و

$$s = uvxyz, |vy| \geq 1, |vxy| \leq p$$

اگر فرض کنیم تعداد a ها در u, v برابر j و تعداد b ها درآنها برابر k باشد، داریم:

$$1 \leq k + j \leq p$$

حال $\underline{\boldsymbol{i = (n-k)(n-j)}}$ را انتخاب میکنیم که رشته ی حاصل برابر $uv^{(n-k)(n-j)}xy^{(n-k)(n-j)}z$ خواهد بود که در نهایت به صورت زیر است:

$$a^{(n-j)+j(n-k)(n-j)}b^{(n-k)+k(n-k)(n-j)}$$

حال با توجه به شرایط لم تزریق برای j, k سه حالت داریم:

$j \neq 0, k = 0$: در این حالت رشته ی حاصل بصورت زیر خواهد بود:

$$a^{(n-j)(nj+1)}b^n$$

که در آن تعداد a ها برابر $(n-j)(nj+1)$ است و از آنجا که $1 \leq j \leq p$ و $n \geq p+2$ پس $n - j \geq 2$ خواهد بود، پس تعداد a ها اول نیست و این رشته عضو زبان نیست.

$j = 0, k \neq 0$: در این حالت رشته ی حاصل بصورت زیر خواهد بود:

$$a^n b^{(n-k)(nk+1)}$$

که در آن تعداد b ها برابر $(n-k)(nk+1)$ است و از آنجا که $1 \leq k \leq p$ و $n \geq p+2$ پس $n - k \geq 2$ خواهد بود، پس تعداد a ها اول نیست و این رشته عضو زبان نیست.

$j \neq 0, k \neq 0$: در این حالت رشته ی حاصل بصورت زیر خواهد بود:

$$a^{(n-j)(j(n-k)+1)}b^{(n-k)(k(n-j)+1)}$$

که در آن تعداد b ها برابر $(n-k)(k(n-j)+1)$ و تعدادa ها برابر $(n-j)(j(n-k)+1)$است و از آنجا که $k + j \leq p$ و $1 < k$ و$1 < j$ و $n \geq p+2$ پس تعداد a ها و b ها اول نیست و این رشته عضو زبان نیست.

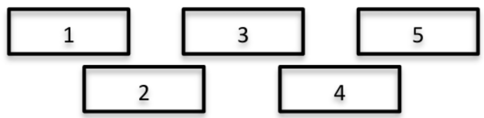
بنابراین در هیچ یک ار حالات رشته تولید شده عضو زبان نبود، پس این زبان مستقل از متن نیست.

c) $\{w \in \{a,b\}* \mid w = w^R, n_a(w) = n_b(w)\}$

راه‌حل:

– عدد p را به عنوان انتخاب حریف در نظر می‌گیریم. طول رشته ی انتخابی ما باید بیشتر از p باشد و عضو زبان L_A باشد. رشته‌ی $s = 0^p1^{2p}0^p$ را در نظر می‌گیریم که طولش $4p$ است و $4p > p$ و عضو زبان نیز هست. سپس حریف رشته انتخابی ما را به ۵ قسمت تقسیم می‌کند که $s = uvwxy$ و $|vwx| \leq p$. به خاطر وجود این شرط، عبارت vwx فقط می تواند در یکی از این ۵ قسمت رسم شده در رشته قرار بگیرد و به طور مثال امکان ندارد که همزمان شامل 0 های ابتدا و انتهای رشته شود. پس این ۵ حالت را بررسی می‌کنیم و i مناسب در هر حالت را انتخاب می‌کنیم.

w = 000 …… 00 1111…..1111 000….00



حالات ۱ و ۳ و ۵: در این حالات x و v هر دو فقط شامل 0 یا 1 هستند و با انتخاب $i = 2$ مقدار یکی از این دو را در رشته حاصل بیشتر می‌کنیم. پس با به هم خوردن شرط برابری تعداد 0 ها و 1 ها رشته حاصل عضو زبان نخواهد بود.

حالات ۲ و ۴: در این حالات x و v هر دو در یک نیمه از رشته قرار دارند و با انتخاب $i = 0$ شرط تقارن رشته را به هم می‌زنیم و رشته حاصل عضو زبان نخواهد بود. ($0^p1^{2p-m}0^{p-n} \notin L_A$ یا $0^{p-n}1^{2p-m}0^p \notin L_A$)

پس در همه حالات رشته حاصل عضو زبان نیست و طبق لم تزریق، L_A مستقل از متن نیست.

گرامر رو از ۰ و ۱ کردم a و b که یک دست تر بشه.

d) $\{w \in \{a, b\} * \mid w = w'w'w'\}$

راه‌حل:

برای عدد p، رشته‌ی $a^pba^pba^pb$ را به دیو می‌دهیم. فرض کنید دیو آن را به صورت $w=uvxyz$ می‌نویسد، به صورتی که $|vy| > 0$ و $|vxy| < p$. اگر قرار بدهیم $i=0$، رشته تبدیل می‌شود به uxz و یک سری از کاراکترها حذف می‌شوند. حالات زیر را در نظر بگیرید:

حالت ۱: v یا y شامل حرف b است.

در این حالت، از آنجا که $|vxy| < p$ و فاصله‌ی دو تا b حداقل برابر p است، پس تعداد bهایی که در v یا y قرار دارند و طی عملیات پمپ کردن حذف می‌شوند، دقیقا برابر یکی است. در نتیجه، در رشته‌ی حاصل (uxz)، دو تا حرف b وجود دارد. در نتیجه رشته‌ی حاصل به هیچ عنوان نمی‌تواند الگویی مشابه w’w’w’ داشته باشد و عضو این زبان نیست.

حالت ۲-۱: v و y فقط شامل حروف aاند و vxy در داخل یکی از بلوک‌های a^p قرار دارد.

در این حالت به رشته‌ی مشابه $a^pba^{p'}ba^pb$ یا $a^pba^pba^{p'}b$ یا $a^pba^pba^{p'}b$ تبدیل می‌شود که نمی‌تواند به فرم w’w’w’ باشد.

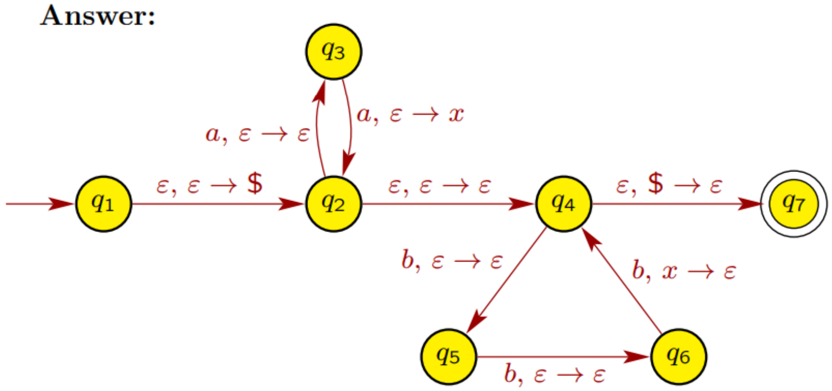
حالت ۲-۲: v و y فقط شامل حروف aاند و vxy شامل بخشی از دو بلوک متوالی a^p است.

در این حالت به رشته‌ی مشابه $a^p b a^{p'} b a^{p''} b$ یا $a^{p'} b a^{p''} b a^p b$ تبدیل می‌شود که نمی‌تواند به فرم $w'w'w'$ باشد.

۲) برای زبان‌های زیر، PDA رسم کنید.

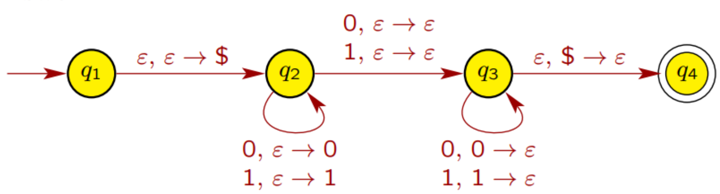
a) $\{w \in \{a, b\} * \mid w = a^{2n} b^{3n}\}$

Answer:



b) $\{w \in \{0,1\} * \mid w = w^R, \mathit{length\ of\ w\ is\ odd}\}$

Answer:



۳) گرامر S را در نظر بگیرید به طوری که تمام قواعد آن به صورت زیر است:

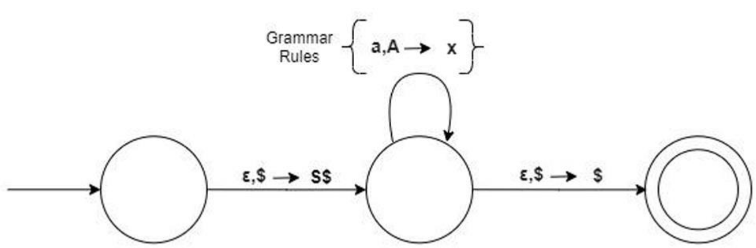
$$A \to aA_1A_2 \dots A_n$$

هم‌چنین هر زوج (A, a) حداکثر یک بار بین قواعد دیده می‌شود.

ثابت کنید برای چنین گرامری، امکان رسم DPDA وجود دارد.

راه‌حل:

طبق توضیح داده شده، گرامر S در فرم نرمال گریباخ است. پس اگر طبق روال تبدیل گرامر در فرم نرمال گریباخ به PDA، این گرامر را به PDA متناظرش تبدیل کنیم، PDA زیر به دست می‌آید (فرض کنید نام استیت‌ها از چپ به راست عبارت است از: q0 و q1 و q2):



به این صورت که به ازای هر کدام از قواعد گرامر، گذار متناظر در قسمت Grammar Rules اضافه می‌شود.

از آن جا که هر زوج (A, a) حداکثر یک بار در قواعد آمده است، پس به ازای هر a و A، دقیقا یک یال از q1 به q1 در PDA وجود دارد به صورتی که: $a, A \rightarrow X$. همچنین می‌دانیم در این PDA یالی به شکل $\varepsilon, A \rightarrow X$ یا $a, \varepsilon \rightarrow X$ ندارم. در نتیجه PDA حاصل قطعی است، چون به ازای هر حالت از نوار و رشته‌ی ورودی، دقیقا یه گذار (transition) قابل انجام است.

۴) با استفاده از خواص زبان‌های مستقل از متن نشان دهید زبان زیر مستقل از متن است:

$$\{w \in \{a, b\} * | w = a^n b^n, n \neq 5k\}$$

راه‌حل:

زبان‌های L1 و L2 را در نظر بگیرید به نحوی که:

$$L_1 = \{a^n b^n | n \geq 0\}$$

$$L_2 = \{w \in \{a, b\} * | |w| \textit{ is not a multiple of } 10\}$$

می‌دانیم زبان L1 مستقل از متن و زبان L2 منظم است (چرا؟). از آن‌جا که اشتراک یک زبان منظم و یک زبان مستقل از متن، مستقل از متن است، پس زبان گفته شده در صورت سوال هم مستقل از متن است. زیرا حاصل $L_1 \cap L_2$ است.

۵) نشان دهید اگر C زبانی مستقل از متن قطعی و R یک زبان منظم باشد، $R \cup C$، مستقل از متن قطعی است.

از آنجا که R یک زبان منظم است، پس $\bar{R}$ هم منظم است.

راه‌حل:

لم ۱: زبان‌های منظم، تحت عملیات complement بسته‌اند.

اثبات: قبلا در درس دیده‌اید.

لم۲: زبان‌های مستقل از متن قطعی، تحت عملیات complement بسته‌اند.

اثبات: کافی است در DPDA زبان، جای accepting stateها و بقیه‌ی استیت‌ها (non-accepting stateها) را عوض کنیم.

لم ۳: اشتراک یک زبان مستقل از متن قطعی و یک زبان منظم، مستقل از متن قطعی است.

اثبات: این لینک را ببینید:

https://math.stackexchange.com/questions/1541057/dcfl-are-closed-under-intersection-with-regular-languages

با استفاده از سه لم ذکر شده، مسئله را اثبات می‌کنیم. می‌دانیم:

$$C \cup R = \overline{\bar{C} \cap \bar{R}}$$

می‌دانیم $\overline{\mathrm{C}}$ یک زبان مستقل از متن قطعی (لم ۲) و $\overline{\mathrm{R}}$ یک زبان منظم است (لم ۱). در نتیجه $\overline{\mathrm{C}} \cap \overline{\mathrm{R}}$ یک زبان مستقل از متن قطعی است (لم ۳). در آخر، $\overline{\overline{\mathrm{C}} \cap \overline{\mathrm{R}}}$ یک زبان مستقل از متن قطعی است (لم ۲) و مسئله اثبات شد.

۶) ترکیب دو زبان X و Y را به این صورت تعریف می‌کنیم که برای تمام رشته‌های $\mathrm{x} \in \mathrm{X}$ و $y \in \mathrm{Y}$، رشته‌ی z ای به وجود می‌آید که:

$$\mathrm{z} = \langle \mathrm{x}, \mathrm{y} \rangle$$

- هر کاراکتر z معادل یکی از کاراکترهای x یا y است (نه هر دو)
- اگر تمام کاراکترهای z که معادل کاراکتری از x هستند را پشت سر هم قرار دهیم، x را تشکیل می‌دهند.
- اگر تمام کاراکترهای z که معادل کاراکتری از y هستند را پشت سر هم قرار دهیم، y را تشکیل می‌دهند.

برای مثال:

$$abba = < aa, bb >, abba \ \neq < abb, ba >$$

الف) نشان دهید $\langle C, R \rangle$، به طوری که R یک زبان منظم و C یک زبان مستقل از متن است، مستقل از متن می‌باشد.

راه‌حل:

فرض کنید R یک DFA به نام DR، و مجموعه‌ی استیت‌های Q دارد و C هم با یک PDA دارد به نام DC، و مجموعه‌ی استیت‌های Q' قابل نمایش است. هم‌چنین فرض کنید هم DR و هم DC هر دو دارای فقط یک استیت شروع و یک استیت پایان هستند (اگر چند استیت پایان داشتیم، می‌توانیم همه‌ی آن‌ها را به یک استیت پایان جدید ببریم). هدف این است که با تشکیل PDA ای به نام A، زبان $\langle C, R \rangle$ را نشان دهیم. سناریوی زیر را در نظر بگیرید:

شروع: برای این که رشته‌ای توسط A پذیرفته شود، باید از یکی از starting stateهای DR یا DC شروع کنیم. مثلا فرض کنید شروعمان از استیت Q'_0 در DC است. تا جایی روی این استیت‌ماشین پیشروی کرده (مثلا استیت Q'_i) و سپس باید به Q_0 در DR برویم.

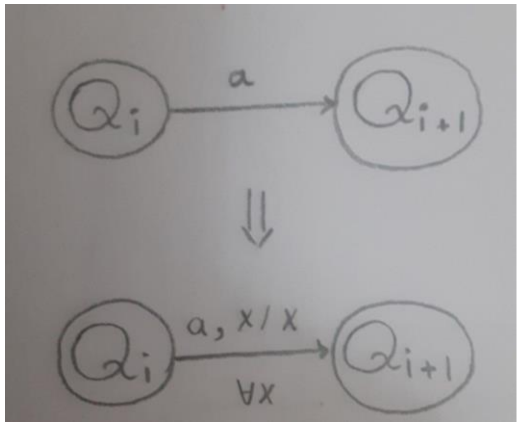
جابه‌جایی بین استیت‌ماشین‌ها: حالا ادامه‌ی پیشروی روی DR انجام می‌شود (مثلا تا استیت Q_j). حالا باید دوباره به Q'_i برگردیم و ادامه‌ی رشته را بخوانیم، و بعد از طی کردن بخش دیگری از رشته روی DC (مثلا تا استیت Q'_k) به استیت Q_j در DR برگردیم. این روند تا جایی ادامه پیدا می‌کند که به یکی از accepting stateهای DC یا DR برسیم.

پذیرفتن رشته: مثلا فرض کنید در آخر به استیت Q_f در DR رسیده‌ایم و رشته هم تمام شده. برای پذیرفته شدن کامل رشته، باید بدانیم آخرین استیتی که در DC تا آن پیشروی کرده‌ایم هم Q'_f بوده؟ به این ترتیب می‌توانیم رشته را بپذیریم.

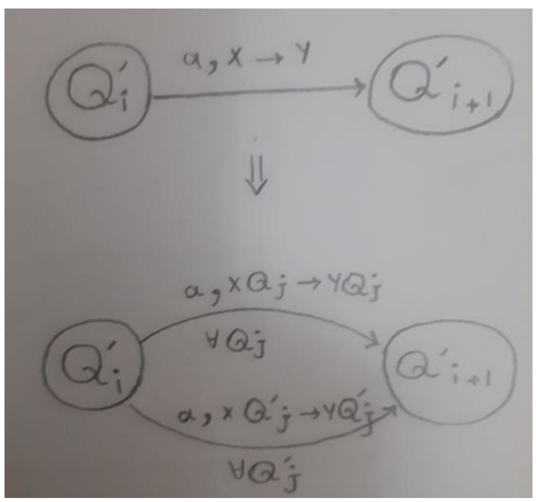
نکته‌ای که در توضیحات بالا وجود دارد این است که برای تشکیل PDA زبان $\langle C, R \rangle$ کافی است همیشه روی استک، آخرین استیتی که از استیت ماشین دیگر تا آن، پیشروی کرده‌ایم را داشته باشیم. به زبان ساده‌تر اگر الان روی DR حرکت می‌کنیم، بدانیم آخرین بار از چه استیتی از DC به اینجا آمده‌ایم، تا بتوانیم دوباره به آن برگردیم. یا اگر الان در یک accepting state روی DC هستیم و رشته‌ی ورودی تمام شده، آیا آخرین استیتی که روی DR تا آن پیشروی کرده‌ایم هم یک accepting state بوده؟ حالا طبق توضیحات زیر A را تشکیل می‌دهیم:

در ابتدا، به مجموعه حروفی که قبلا قابلیت قرار گرفتن روی استک DC را داشتند، نام تمام استیت‌های DR و DC را اضافه می‌کنیم. به عنوان مثلا حالا کاراکتر Q_i یا Q'_i هم مانند $ (کاراکتری که در ابتدای استک قرار می‌گیرد) می‌توانند روی استک پوش بشوند.

حالا DR را به شکل زیر، تبدیل به PDAای به نام ’DR می‌کنیم:



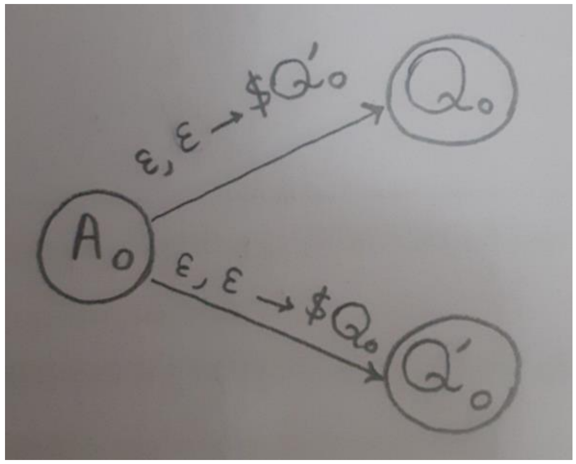
در قدم بعدی گذارهای DC را هم به این شکل تغییر داده و ’DC را تشکیل می‌دهیم:



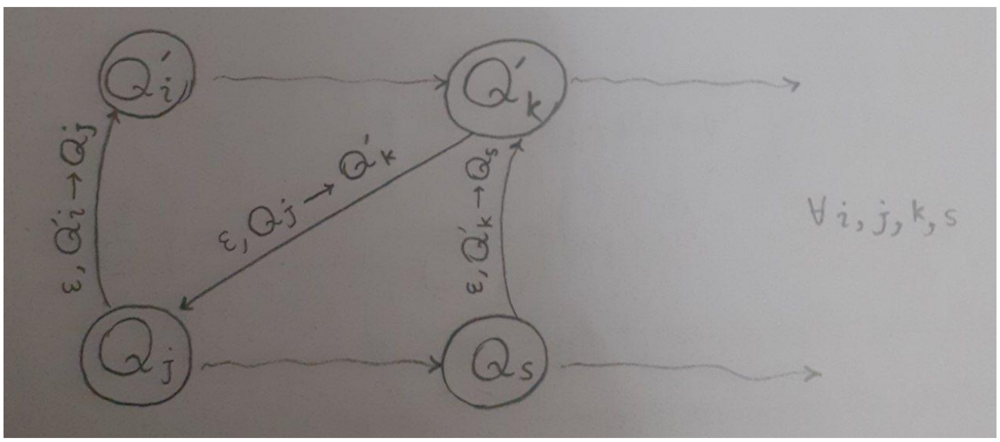
نکته‌ای که درباره‌ی ’DC و ’DR حائز اهمیت است این است که همیشه عنصری که سر استک قرار دارد، نام یکی از استیت‌های ’DC یا ’DR است (از ’DC نتیجه می‌شود) و همچنین هیچ transitionای این عنصر را تغییر نمی‌دهد (از ’DR و ’DC نتیجه می‌شود).

برای تشکیل A، ’DR و ’DC را کنار هم قرار می‌دهیم و تمام transitionهای آن‌ها را حفظ می‌کنیم. کافی است بین این دو استیت‌ماشین، گذارهای زیر را هم اضافه کنیم:

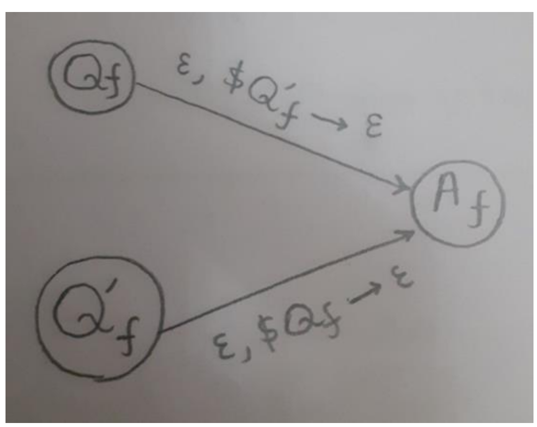
شروع:



جابه‌جایی بین استیت‌ماشین‌ها:



پذیرفتن رشته:



ب) نشان دهید $\langle C_1, C_2 \rangle$، به طوری که C1 و C2 هر دو زبان‌های مستقل از متنی باشند، لزوما مستقل از متن نیست.

راه‌حل:

زبان‌های مستقل از متن زیر را در نظر بگیرید:

$$C_1 = \mathrm{a^n b^n}$$

$$\mathrm{C_2} = c^{\mathrm{n}}\mathrm{d^n}$$

با استفاده از برهان خلف نشان می‌دهیم ترکیب این دو زبان مستقل از متن نیست. خلاف این موضوع را در نظر بگیرید. فرض کنید $\langle C_1, C_2\rangle$ مستقل از متن است. زبان زیر را در نظر بگیرید:

$$L = \langle C_1, C_2\rangle \cap \mathrm{a*c*b*d*}$$

می‌دانیم اشتراک یک زبان مستقل از متن ($\langle C_1, C_2\rangle$) و یک زبان منظم (a*c*b*d*)، مستقل از متن است (L). هم‌چنین واضح است که داریم:

$$L = \mathrm{a^n c^m b^n d^m}$$

اما به سادگی به استفاده از لم تزریق می‌توان نشان داد که این زبان مستقل از متن نیست، که این یک تناقض است و $\langle C_1, C_2\rangle$ مستقل از متن نیست.

۷) (امتیازی) عملیات No-Prefix(L) را به این صورت تعریف می‌کنیم که حاصل آن مجموعه‌ی تمام رشته‌هایی از زبان L است که هیچ پیشوندی از آن عضو این زبان نیست. ثابت کنید زبان‌های مستقل از متن نسبت به این عملیات بسته نیستند.

راه‌حل:

زبان L را در نظر بگیرید، به طوری که:

$$\mathrm{L_1} = \{\mathrm{a^n b^m c} | \mathrm{n, m} \geq 1, \mathrm{n} \neq \mathrm{m}\}$$
$$\mathrm{L_1} = \{\mathrm{a^n b^m} c^m | \mathrm{n, m} \geq 1\}$$
$$L = L_1 \cup L_2$$

پیشوندهای رشته‌های زبان $\mathrm{L_1}$ (خود رشته را در نظر نمی‌گیریم) فقط شامل a یا b هستند. درحالی که میدانیم تمام رشته‌های $\mathrm{L_1}$ و $\mathrm{L_2}$ (در نتیجه، L) حداقل یک c دارند. بنابراین تمام رشته‌های $\mathrm{L_1}$ عضو No-Prefix(L)اند.

حالا رشته‌ای به نام w از $\mathrm{L_2}$ در نظر بگیرید که عضو No-Prefix(L) نیست. می‌دانیم تعداد bها و cهای پیشوندهای رشته‌های $\mathrm{L_2}$ با هم برابر نیستند، پس هیچ‌کدام از پیشوندهای w عضو $\mathrm{L_2}$ نیست. در نتیجه این رشته حداقل یک پیشوند دارد که عضو $\mathrm{L_1}$ است، که برای آن پیشوند خاص داریم: $\mathrm{n} \neq \mathrm{m}$. در نتیجه هر رشته‌ای در $\mathrm{L_2}$ به فرم n و m برابر، عضو No-Prefix(L) است.

به صورت کلی داریم:

$$No-Prefix(\mathrm{L}) = \mathrm{L_1} \cup \{\mathrm{a^n b^n c^n} | \mathrm{n} \geq 1\}$$

می‌دانیم $\mathrm{L_1}$ و $\mathrm{L_2}$ زبان‌های مستقل از متنی هستند، در نتیجه L هم زبانی مستقل از متن است (زبان‌ها مستقل از متن نسبت به عملیات اجتماع‌گیری بسته‌اند). با استفاده از برهان خلف، اثبات می‌کنیم No-Prefix(L) مستقل از متن نیست. خلاف این موضوع

را در نظر بگیرید. می‌دانیم اشتراک یک زبان مستقل از متن و یک زبان منظم، مستقل از متن است. پس زبان زیر هم مستقل از متن است:

$$No-Prefix(\mathrm{L}) \cap (\mathrm{a}*\mathrm{b}*\mathrm{ccc}*) = \{\mathrm{a^n b^n c^n} | \mathrm{n} \geq 2\}$$

در حالی که به سادگی با استفاده از لم تزریق می‌توان نشان داد زبان حاصل مستقل از متن نیست که این یک تناقض است. در نتیجه No-Prefix(X) مستقل از متن نیست. بنابراین، زبانی مستقل از متن معرفی کردیم که No-Prefix آن مستقل از متن نیست، که اثبات می‌کند زبان‌های مستقل از متن نسبت به این عملیات بسته نیستند.