فصل اول

« زبان‌های منظم »

نظریه زبان‌ها و ماشین‌ها، چگونگی ساخت یک زبان و قواعد آن را بررسی می‌کند. منظور از زبان، شکل ظاهری و مستقل از گفتار است. نمونه این نوع زبان را می‌توان در دستگاه‌های ریاضی یافت. به طور مثال دستگاه اعداد طبیعی، دارای قواعد ساخت رشته‌های عددی است. هر عدد طبیعی با یکی از قواعد زیر تولید می‌شود:

$1: 0 \in L_N \qquad 2: x \in L_N \text{ for all } x \in \text{Variable} \qquad 3: s(x) \in L_N \text{ for all } x \in L_N \qquad 4: x + y \in L_N \text{ for all } x, y \in L_N$

برای نمونه، رشته‌های عددی $s(s(s(0)))$ یا $s(s(0)+s(s(s(0))))$ با این قواعد تولید می‌شوند. دقت کنید در اینجا ما می‌خواهیم اعداد طبیعی را با یک ساختار رشته‌ای بسازیم، از این رو رشته 0 که کمترین طول یعنی یک دارد را رشته ابتدایی گرفته و زبان خود را با 0 شروع می‌کنیم. در اینجا 0 رشته‌ای با طول یک است و مفهوم عددی صفر را ندارد. از آنجا که طبق قاعدهٔ 1 رشته $0 \in L_N$ است قاعده 3 می‌تواند روی آن اجرا شود که با سه بار تکرار این قاعده رشته $s(s(s(0)))$ ساخته می‌شود. ساخت رشته دوم نیز قاعده 4 را استفاده می‌کند و پس از آن قاعده 3 را مجدد روی $s(0)+s(s(s(0)))$ بکار می‌برد. یک زبان، پردازش‌های متنوعی روی رشته‌ها تعریف می‌کند. شاید بتوان برای یک زبان طبیعی مانند فارسی، انگلیسی، چینی و غیره نیز توصیف صوری ارائه داد. جمله در یک زبان طبیعی از تعدادی کلمه و کلمه از تعدادی حرف ساخته می‌شود. یک واحد بزرگتر مانند پاراگراف، شامل جملات متعددی است. پس هر زبان طبیعی هم برخوردار از یک مجموعه قواعد است که توسط این قواعد آن زبان تولید می‌شود. در این فصل، ابتدا مفاهیم پایه‌ای نظریه زبان‌ها و ماشین‌ها را بررسی کرده، سپس زبان‌های منظم را که ساده‌ترین نوع زبان‌های صوری هستند معرفی می‌کنیم. در پایان ماشین متناهی قطعی مورد بحث قرار می‌گیرد که می‌تواند به طور خودکار یک زبان منظم را بسازد.

مفاهیم پایه

الف ـ مجموعه‌ها: به هر بخشی از جهان، یک مجموعه گفته می‌شود. مجموعه‌ها به دو گروه تقسیم می‌شوند: قابل شمارش و غیرقابل شمارش. مجموعه A را قابل شمارش می‌گویند اگر یک تابع یک‌به‌یک مانند f از A به N (مجموعه اعداد طبیعی) وجود داشته باشد، در غیر این صورت A ناشمار (غیرقابل شمارش) است. خود مجموعه‌های قابل شمارش نیز در دو دسته متناهی و نامتناهی قرار می‌گیرند. برای مثال، مجموعه الفبای فارسی، مجموعه‌ای قابل شمارش و متناهی است؛ اعداد طبیعی (N) قابل شمارش و نامتناهی، در نهایت اعداد حقیقی (R) غیرقابل شمارش هستند.

ب ـ الفبای زبان: الفبا مجموعه متناهی و ناتهی از نمادهایی است که هیچ کدام به عناصر دیگری در الفبا تجزیه نمی‌شوند. حروف الفبا، اجزا بنیادین ساخت کلمات هستند. مجموعه الفبای یک زبان را با Σ نشان می‌دهیم. در زبان‌های کامپیوتری می‌توان الفبای ترکیبی هم داشت. این عناصر الفبا را به خاطر غیر قابل تجزیه بودن، عناصر پایانی نیز می‌نامند.

ج ـ رشته‌ها: رشته دنباله‌ای از عناصر الفبای یک زبان است. مثلاً 1001 رشته‌ای است که روی الفبای $\sum=\{0,1\}$ تعریف می‌شود. مجموعه رشته‌ها روی الفبای Σ را، با Σ^* نشان می‌دهیم. اگر Σ تک عضوی باشد Σ^* مجموعه‌ای نامتناهی شمارا و اگر Σ بیش از یک عضو داشته باشد، ناشماراست. از v ، u و w برای نام بردن عبارات و رشته‌ها استفاده می‌شود. یک رشته که بخشی از آن با یک متغیر مانند x جایگزین شود، شبه رشته نامیده می‌شود. شبه رشته با شبه جمله تفاوت دارد. به طور مثال $1001x01$ یک شبه رشته است اما متغیر u و v که به یک رشته اشاره می‌کنند را شبه جمله می‌گویند. طول رشته w با $|w|$ و معکوس آن را با w^R نشان می‌دهند. رشته تهی رشته‌ای به طول صفر است که با λ مشخص می‌گردد.

نکته ۱: با فرض متناهی بودن الفبای Σ، مجموعه‌های Σ^* و Σ^+ همواره نامتناهی است؛ زیرا طول رشته‌ها در این مجموعه‌ها محدودیتی روی Σ ندارد.

به طور کلی، یک زبان به صورت زیر مجموعه‌ای از Σ^* تعریف شده و یک رشته در زبان L را «جمله» می‌گویند. هر مجموعه از رشته‌ها، روی الفبای Σ یک زبان در نظر گرفته می‌شود.

د ـ زبان: یک زبان، با مجموعه‌ای از رشته‌ها تعریف می‌شود. برای الفبای Σ یک زبان L، زیر مجموعه‌ای از Σ^* است. مجموعه زبان‌های ممکن روی الفبای Σ با Λ نشان می‌دهیم که در اینجا $\Lambda = Power(\Sigma^*)$ می‌باشد. مجموعه زبان‌های ممکن روی الفبای ناتهی Σ یعنی Λ، مجموعه‌ای نامتناهی و غیرقابل شمارش است.

برای الفبای $\sum = \{a,b\}$ داریم: $\sum^* = \{\lambda, a, ab, aa, aab, abb, aaa, \cdots\}$

در نمونه بالا مجموعه‌ی $A = \{a, ab, bb\}$ یک زبان روی الفبای (Σ) است. از آن‌جا که تعداد رشته‌های زبان A متناهی است، به A زبان متناهی می‌گویند. اما در همین نمونه فوق مجموعه‌ی $X = \{a^n b^n \mid n \ge 0\}$ یک زبان نامتناهی روی اَلفبای Σ است.

❖ **تعریف۱: عملگرهای رشته**

عملگرهای مختلفی روی هر رشته تعریف می‌شود؛ از جمله: الحاق $(u \bullet v)$، توانی (u^n)، وارون (u^R) و بستار یا closure (u^*) که هر کدام عمل خاصی را روی رشته‌ها تعریف می‌کند.

الحاق y به x به معنای وقوع y بلافاصله پس از x است. عمل الحاق، خاصیت شرکت‌پذیری داشته اما خاصیت جابجایی ندارد.

منظور از (u^n) کنار هم قرار دادن u به تعداد n بار است و این عملگر، خاصیت پخشی ندارد یعنی $(ur)^n \ne u^n r^n$.

عملگر وارون به صورت بازگشتی به شکل زیر تعریف می‌شود:

۱ـ اگر $u = a \in \Sigma$ باشد در این صورت $u^R = a^R = a$ است.

۲ـ اگر $u = av$ که $a \in \Sigma$ و $v \in \Sigma^*$ باشد در این صورت $u^R = v^R a$ است.

برای دو رشته v و u، خواهیم داشت: $(uv)^R = v^R u^R$.

عملگر x^* عبارت است از تکرار صفر مرتبه یا بیشتر رشته یا الفبای x، که به شکل $\bigcup_{i=0}^{\infty} x^i$ بیان می‌شود. عملگر x^+ تکرار یک مرتبه یا بیشتر عبارت x است. عملگرهای * و + نیز خاصیت پخشی روی الحاق ندارند.

نکته ۲: اگر $v = uxy$ باشد آن‌گاه x یک زیر رشته از v است. اگر $u = \lambda$ باشد x پیشوندی برای v است و اگر $y = \lambda$ باشد x یک پسوند برای v محسوب می‌شود. هر شبه جمله فقط یک پیشوند دارد و هر جمله به طول n دارای $n+1$ پیشوند است. به طور مثال برای رشته v پیشوندهای λ، u و ux وجود دارند.

نکته ۳: رشته‌ای که از دو طرف یکسان نوشته شود را آئینه‌ای یا Palindrome می‌گویند؛ مانند رشته "RADAR" و "KADAK".

عملگرهای زبان

روی زبان‌های ساخته شده از الفبا Σ، عملگرهای زیادی تعریف می‌شود. این عملگرها با پردازش کلمات زبان، کلمات جدیدی را می‌سازند و در نتیجه زبان جدیدی ایجاد می‌کنند. از آنجا که هر زبان، در واقع یک مجموعه است، عملگرهای مجموعه‌ها مانند اجتماع، اشتراک، تفاضل نامتقارن و از این قبیل عملگرها نیز قابل تعریف می‌باشند. اما تمام عملگرهای زبان این گونه نیستند. برای مثال، عملگر تقسیم (/) رشته‌های یک زبان را بر اساس پسوندهایی که در زبان دیگری دارند، برش می‌زند و خارج قسمت‌ها را در زبان جدیدی می‌ریزد. نمونه‌های از این دست پردازش‌ها وجود دارد. در ادامه مهمترین و پایه‌ای‌ترین عملگرهای زبانی بررسی می‌شود.

❖ **تعریف۲: عملگرهای زبان**

مهمترین عملگرهای زبان در ادامه معرفی و بررسی می‌شوند.

الف) عملگر اجتماع و اشتراک: اگر L_1 و L_2 دو زبان روی الفبای Σ باشند، عملگر اجتماع و اشتراک روی این دو به صورت زیر است:

$$L_1 \cup L_2 = \{w \in \Sigma^* \mid w \in L_1 \ \text{or}\ \ w \in L_2\}$$

$$L_1 \cap L_2 = \{w \in \Sigma^* \mid w \in L_1 \ \text{and}\ \ w \in L_2\}$$

ب) عملگر الحاق: زبان جدید با الحاق کلمات زبان L_1 در ابتدای کلمات زبان L_2 ساخته می‌شود:

$$L_1 \bullet L_2 = \{w \in \Sigma^*,\ w = u.v \mid u \in L_1 \text{ and } v \in L_2\}$$

ج) عملگر متمم: مجموعه کلمات غیر عضو در زبان را به صورت یک زبان جدید عرضه می‌کند:

$$\overline{(L_1)} = \{w \in \Sigma^* \mid w \notin L_1\}$$

مثال ۱: زبان $L = \{aa, bb\}$ را روی الفبای $\sum = \{a,b\}$ در نظر بگیرید. زبان $\bar{L}$ را مشخص کنید؟

☑ **پاسخ:** با توجه به اینکه L دو عضوی بوده و Σ^* تعداد نا متناهی عضو دارد، لذا رشته‌هایی با طول بیش از ۲ و همین طور ab و ba و a و b و λ در متمم L وجود دارد.

$$\bar{L} = \{\lambda, a, b, ab, ba\} \bigcup \{w \in \{a,b\}^* \mid |w| > 2\}$$

د) عملگر تقسیم: نسبت راست دو زبان L_1 و L_2 یا عمل تقسیم دو زبان، شامل مجموعه پیشوندهایی از L_1 که بخش پایانی آن‌ها در L_2 قرار دارد، به صورت یک زبان جدید ساخته می‌شود:

$$L_1 / L_2 = \{w \in \Sigma^* \mid \exists v \in \Sigma^* \text{ that } wv \in L_1 \text{ and } v \in L_2\}$$

هـ) عملگر وارون: از مجموعه کلمات وارون‌سازی شده زبان جدیدی می‌سازد:

$$L_1^R = \{w \in \Sigma^* \mid w^R \in L_1\}$$

و) عملگر بستار، بعلاوه و توانی: عملگر بستار تمام کلمات ممکن از طریق بستار ترکیبی با طول مختلف را روی کلمات زبان ایجاد می‌کند. عملگر توانی این کار را تنها در طول خاص، مثلاً ۳ مرتبه الحاق تمامی کلمات یک زبان، انجام می‌دهد. عملگر بعلاوه هم فقط روی حالت صفر تعریف نمی‌شود:

(منظور از s.t. همان Such that می‌باشد)

$$L_1^* = \{w \in \Sigma^* \mid w = u_1 u_2 \dots u_n \text{ s.t. } u_i \in L_1 \text{ and } n \geq 0\}$$

زبان L_1^t به صورت $L_1^t = L_1^* - \{\lambda\}$ تعریف می‌شود. حتی زمانی که $L = \varnothing$ باشد باز هم این تعریف کارائی دارد (به مفهوم تفاضل مجموعه‌ها توجه کنید).

تعریف L_1^t به صورت $\{\omega \in \Sigma^* \mid \omega = u_1 u_2 \dots u_n \text{ s.t } u_i \in L_1 \text{ and } n \geq 1\}$ نیز عیناً معادل دیگری برای آن ارائه می‌دهد. عملگر توانی نیز به صورت زیر است.

$$L_1^n = \{w \in \Sigma^* \mid w = u_1 u_2 \dots u_n \text{ s.t. } u_i \in L_1\}$$

اگر $L = \{a^n b^n \mid n \geq 0\}$ فرض شود، زبان L^R و L^2 به این صورت بیان می‌شود:

$$L^2 = \{a^n b^n a^m b^m \mid m, n \geq 0|$$

$$L^R = \{b^n a^n \mid n \geq 0\}$$

ز) عملگر تفریق: این عملگر، اشتراک یک زبان و متمم زبانی دیگر را در زبان جدیدی می‌آورد.

$$L_1 \setminus L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ and } w \notin L_2\}$$

توجه: هر یک از علائم $-$ و $\setminus$ به عنوان نماد عملگر تفریق استفاده می‌شوند.

مثال ۲: با توجه به دو زبان $L_2 = \{0,1,11\}$ و $L_1 = \{0,01,111\}$ برای عملگرهای تقسیم و تفاضل خواهیم داشت:

☑ **پاسخ:**

$$L_1 / L_2 = \{\lambda, 0, 1, 11\} \qquad L_1 \setminus L_2 = \{01, 111\}$$

$$L_2 \setminus L_1 = \{1,\ 11\} \qquad L_2 / L_1 = \{\lambda\}$$

مثال ۳: زبان L را روی الفبای $\sum = \{a,b\}$ بیابید بطوری‌که $(\bar{L})^+ = \bar{L}^+$ باشد و سپس این موضوع را ثابت کنید؟

☑ **پاسخ:** فرض کنید $L = \phi$ باشد، در این‌صورت تساوی‌های روبه‌رو برقرار است.

$$\left.\begin{array}{l} \bar{L} = \Sigma^* \\ L^+ = \phi \end{array}\right\} \Rightarrow (\bar{L})^+ = \bar{L}^+ = \Sigma^*$$

هر یک از زبان‌های $\Sigma^+, \lambda, \Sigma^*, \{a^*\}, \{b^*\}, \{a^+\}$ و $\{b^+\}$ نیز می‌توانند پاسخ‌های دیگر این سوال باشند.

عملگرهای زبانی نقش مهمی در پردازش کلمات یک زبان دارند. ممکن است یک عملگر ویژگی‌هایی چون شرکت‌پذیری، جابه‌جایی، عضو خنثی و عضو وارون داشته باشد. در ادامه مشخص می‌کنیم، هر عملگر چه ویژگی‌ای دارد.

مثال ۴: کدام‌یک از عملگرهای زبان، ویژگی شرکت‌پذیری، جابجایی، عضو خنثی و عضو وارون دارند؟

پاسخ: **الف ــ** اگر L_1، L_2 و L_3 عضو Λ باشند برای عملگرهای اجتماع و اشتراک خاصیت شرکت‌پذیری و جابه‌جایی برقرار است. همچنین برای عضو خنثی داریم:

$L_1 \cup \varnothing = L_1$ $\varnothing$ عضو خنثی اجتماع

$L_1 \cap \Sigma^* = L_1$ Σ^* عضو خنثی اشتراک

$L_1 \cup ? = \varnothing$ تحت اجتماع هر رشته عضو وارون وجود ندارد

$L_1 \cap ? = \Sigma^*$ همینطور عضو وارون تحت اجتماع نیز وجود ندارد.

ب ــ عملگر الحاق، شرکت‌پذیر است ولی ویژگی جابه‌جایی ندارد. برای مثال اگر $L_1 = \{0,10\}$ و $L_2 = \{11,1\}$ داریم:

$$\left.\begin{aligned} L_1 \bullet L_2 &= \{011,01,1011,101\} \\ L_2 \bullet L_1 &= \{110,1110,10\} \end{aligned}\right\} \Rightarrow L_1 \bullet L_2 \neq L_2 \bullet L_1$$

همچنین عضو خنثی برای عملگر الحاق $\{\lambda\}$ است، ولی عملگر الحاق عضو وارون ندارد. $L_1 \bullet \{\lambda\} = L_1$

$L_1 \bullet ? = \{\lambda\}$

ج ــ عملگر متمم به تنهایی هیچ ویژگی ندارد. چرا که یک عملگر تک‌موضعی است و فقط یک پارامتر دارد.

د ــ عملگر تقسیم نه شرکت‌پذیر است نه خاصیت جابه‌جایی دارد. برای مثال با فرض $L_1 = \{1101,1010\}$، $L_2 = \{01,10\}$ و $L_3 = \{10,1\}$ داریم:

$$\left.\begin{aligned} L_1 / L_2 &= \{11,10\} \\ (L_1/L_2)/L_3 &= \{\lambda,1\} \\ L_2/L_3 &= \{\lambda,0\} \\ L_1/(L_2/L_3) &= \{1101,1010,101\} \end{aligned}\right\} \Rightarrow L_1/(L_2/L_3) \neq (L_1/L_2)/L_3 \Rightarrow \text{پس شرکت‌پذیر نیست}$$

همین‌طور جابجایی نیز نمی‌باشد $L_1/L_2 = \{11,10\} \neq \varnothing = L_2/L_1$

برای عملگر تقسیم زبان $\{\lambda\}$ عضو خنثی به حساب می‌آید، زیرا برای هر زبان $L \in A$ داریم $L/\{\lambda\} = L$ است. یافتن عضو وارون عملگر تقسیم، وقتی‌که L متناهی باشد، راحت است. اگر $w \in L$ که برای هر $w' \in L$ داشته باشیم $|w'| \leq |w|$ آن‌گاه $\{w\}$ یک وارون برای زبان L تحت عمل تقسیم می‌سازد، زیرا $L/\{w\} = \{\lambda\}$ است.

باید توجه داشت که ممکن است وارون‌های زیادی برای یک زبان متناهی وجود داشته باشد. وقتی L نامتناهی است وجود عضو وارون برای عملگر تقسیم قطعی نیست. برای مثال، زبان‌های L_1 و L_2 که در زیر تعریف شده‌اند، نامتناهی هستند اما L_1 وارون داشته ولی L_2 وارون ندارد. زبان $\{a\}$ برای زبان L_1 یک وارون می‌سازد که با تقسیم L_1 بر $\{a\}$ عضو همانی تقسیم، یعنی $\{\lambda\}$ ساخته می‌شود. اما برای زبان L_2 چنین عضوی وجود ندارد.

$L_1 = \{ab^n \mid n \geq 0\}$ $L_1/\{a\} = \{\lambda\}$

$L_2 = \{a^n b \mid n \geq 0\}$ $L_2/? = \{\lambda\}$

ز ــ عملگر تفریق، شرکت‌پذیر نیست؛ چرا که بنا بر تعریف داریم:

$$w \in L_1 \setminus (L_2 \setminus L_3) \Rightarrow \left.\begin{aligned} & w \in L_1 \\ & w \notin L_2 \setminus L_3 \end{aligned}\right\} \Rightarrow$$ ممکن است $w \in L_1 \cap L_3$ باشد ولی عضو L_2 نباشد

در صورتی‌که برای حالت دیگر داریم $w \in (L_1 \setminus L_2) \setminus L_3$ از این رو $w \notin L_3$ است. پس داریم: $L_1 \setminus (L_2 \setminus L_3) \neq (L_1 \setminus L_2) \setminus L_3$

همچنین ویژگی جابه‌جایی برای عملگر تفریق اصلاً برقرار نیست؛ چرا که این حالت با مفهوم تفریق، تناقض دارد. عضو همانی تفریق هم $\varnothing$ است؛ چرا که برای هر L داریم $L \setminus \varnothing = L$. از طرفی عضو وارون هر زبان L خود آن زبان است؛ چراکه $L \setminus L = \varnothing$ و به طور طبیعی هر زبان دیگر L_1 که $L \subset L_1$ نیز یک وارون برای زبان L به حساب می‌آید.

هـ ــ عملگر وارون تک‌موضعی است و یک پارامتر بیشتر ندارد لذا شرکت‌پذیری و جابه‌جایی برای آن تعریف نمی‌شود.

و ــ عملگرهای بستار، بعلاوه و توانی نیز تک موضعی هستند و به همین ترتیب این ویژگی‌ها برای آن‌ها تعریف نمی‌شود.

مثال ۵: کدام‌یک از عملگرهای تک‌موضعی متمم، وارون، بستار، بعلاوه و توانی، روی عملگرهای اجتماع، اشتراک، الحاق و تقسیم، توزیع‌پذیر است؟

توضیح: اگر $F(E(x, y)) = E(F(x), F(y))$ باشد، عملگر $F(x)$ را روی $E(x,y)$ توزیع‌پذیر می‌گویند.

☑ **پاسخ: الف ـ** بنابر قانون دمورگان، عملگر متمم روی اجتماع و اشتراک توزیع‌پذیر نیست، زیرا متمم اشتراک‌ها، اجتماع متمم‌ها است و متمم اجتماع‌ها، اشتراک متمم‌ها است. همچنین عملگر وارون، توزیع‌پذیر است اما عملگر بستار، نه روی اشتراک و نه روی اجتماع توزیع‌پذیر نیست. مثال زیر این موضوع را نشان می‌دهد:

$$L_1 = \{101, 010\}$$
$$L_2 = \{11, 00\}$$
$$\begin{cases} 10111 \in (L_1 \cup L_2)^* \\ 10111 \notin (L_1^* \cup L_2^*) \end{cases} \Rightarrow (L_1 \cup L_2)^* \neq (L_1^* \cup L_2^*)$$

و برای اشتراک فرض کنید $L_1 = \{0, 01\}$ و $L_2\{10, 00\}$ باشد لذا:

$$L_1 \cap L_2 = \varnothing \rightarrow (L_1 \cap L_2)^* = \{\lambda\}$$
$$L_1^* \cap L_2^* = \{10, 00\}^* \cap \{0, 01\}^* = \{0010, \ldots\} \neq \{\lambda\}$$

وضعیت عملگر بعلاوه و توانی روی اجتماع و اشتراک، مشابه عملگر بستار است و روی اشتراک و اجتماع توزیع‌پذیر نیست.

ب ـ عملگر متمم، روی عملگر الحاق توزیع‌پذیر نیست. برای مثال برای زبان‌های $L_1 = \{\lambda\}$ و $L_2 = \{0\}$ داریم:

$$0 \in L_1 \bullet L_2 \rightarrow 0 \notin \overline{L_1 \bullet L_2}$$
$$0 \in \overline{L_1} \text{ and } \lambda \in \overline{L_2} \Rightarrow 0 \in \overline{L_1} \bullet \overline{L_2} \Rightarrow \overline{L_1 \bullet L_2} \neq \overline{L_1} \bullet \overline{L_2}$$

ج ـ عملگر وارون نیز روی عملگر الحاق توزیع‌پذیر نیست؛ چراکه اگر $w = uv$ که $w \in L_1 \bullet L_2$ آن‌گاه $w^R = u^R v^R$ که لزوماً عضو $L_1^R \bullet L_2^R$ نیست، چرا که قسمت اول خود را از L_2^R می‌گیرد.

عملگر بستار، عملگر بعلاوه و عملگر توانی هر سه توزیع‌پذیری روی عملگر الحاق را ندارند. اگر $L_1 = \{1\}$ و $L_2 = \{0\}$ در نظر گرفته شود:

$$(L_1 \bullet L_2)^* = \{10\}^* = \{\lambda, 10, 1010, \ldots\} \neq \{1, 11, \ldots\}.\{0, 00, \ldots\}$$

د ـ عملگر تقسیم:

توزیع‌پذیری عملگر متمم روی عمل تقسیم به راحتی رد می‌شود. برای این موضوع فرض کنید $L_1 = \Sigma^+$ و $L_2 = \{\lambda\}$ باشد، در این صورت $L_1 / L_2 = \Sigma^+$. حال داریم $\overline{(L_1 / L_2)} = \{\lambda\}$ در صورتی‌که چون $\overline{L_1} = \{\lambda\}$ و $\overline{L_2} = \Sigma^+$ داریم $\overline{L_1} / \overline{L_2} = \varnothing$ و تساوی برقرار نیست.

برای اینکه نشان دهیم عملگر وارون روی عملگر تقسیم توزیع‌پذیر نیست، به مثال زیر توجه کنید:

$$\begin{cases} L_1 = \{1, 10, 100, 1010\} \\ L_2 = \{1\} \end{cases} \Rightarrow L_1 / L_2 = \{\lambda\} \Rightarrow (L_1 / L_2)^R = \{\lambda\}$$

$$\begin{cases} L_1^R = \{1, 01, 001, 0101\} \\ L_2^R = \{1\} \end{cases} \Rightarrow L_1^R / L_2^R = \{\lambda, 0, 00, 010\}$$

هـ ـ عملگرهای بستار، بعلاوه و توانی هم روی عملگر تقسیم توزیع‌پذیر نیستند:

$$\begin{cases} L_1 = \{1\} \\ L_2 = \{11\} \end{cases} \Rightarrow \frac{L_1}{L_2} = \varnothing \Rightarrow \left(\frac{L_1}{L_2}\right)^* = \{\lambda\}$$

$$\begin{cases} L_1^* = \{1, 11, 111, \ldots\} \\ L_2^* = \{11, 1111, \ldots\} \end{cases} \Rightarrow \frac{L_1^*}{L_2^*} = \{\lambda, 1, 11, \ldots\}$$

و– عملگرهای تک موضعی $\overline{()}$، $()^R$، $()^*$، $()^+$ و $()^n$ روی تفریق (\) به صورت زیر هستند. فرض کنید L_1 و L_2 همان تعریف قبلی را دارند.

$$\overline{(L_1 \setminus L_2)} = \overline{L_1} \cup (L_1 \cap L_2)$$

$$(L_1 \setminus L_2)^R = L_1^R \setminus L_2^R$$

$$(L_1 \setminus L_2)^* = L_1^* \setminus (L_1 \cap L_2)^* \cup \{\lambda\}$$

نشان دادن این موضوع برای عملگر توانی و عملگر بعلاوه مشابه عملگر بستار است، اثبات هر یک، از طریق نمودار وِن برای مجموعه‌ها قابل انجام است.

مثال ۶: **آیا عملگرهای اجتماع، اشتراک، الحاق و تقسیم روی هم توزیع‌پذیرند؟ توضیح: یک عملگر دو موضعی $F(x,y)$ روی عملگر دو موضعی $E(x,y)$ توزیع‌پذیر است اگر داشته باشیم: $F(x,E(y,z)) = E(F(x,y),F(x,z))$.**

☑ **پاسخ:**

۱– توزیع‌پذیری عملگر اجتماع روی عملگر اشتراک و برعکس: بنابر قانون دمورگان برقرار است.

۲– توزیع‌پذیری عملگر الحاق روی عملگرهای اجتماع و اشتراک: ثابت می‌شود که عمل الحاق روی اجتماع، توزیع‌پذیر بوده اما روی اشتراک توزیع‌پذیر نیست. به این صورت که اگر L_1، L_2 و L_3 سه زبان روی الفبای Σ باشند پس: $L_1 \bullet (L_2 \cup L_3) = (L_1 \bullet L_2 \cup L_1 \bullet L_3)$

$$w \in (L_1 \bullet (L_2 \cup L_3) \leftrightarrow w = uv \quad u \in L_1 \;\;\&\;\; v \in L_2 \;\; \text{or} \;\; L_3 \leftrightarrow w \in L_1 \bullet L_2 \;\text{or}\; w \in L_1 \bullet L_3$$

برای نقض توزیع‌پذیری روی اشتراک فرض کنید:

$$L_1 = \{1,10\} \quad L_2 = \{01\} \quad L_3 = \{1,0\}$$

$$L_2 \cap L_3 = \varnothing \rightarrow L_1 \bullet (L_2 \cap L_3) = L_1 \bullet \varnothing = \varnothing$$

$$\begin{cases} L_1 \bullet L_2 = \{101,1001\} \\ L_1 \bullet L_3 = \{11,10,101,100\} \end{cases} \Rightarrow L_1 \bullet L_2 \cap L_1 \bullet L_3 = \{101\} \neq \varnothing$$

پس داریم: $L_1 \bullet (L_2 \cap L_3) \neq (L_1 \bullet L_2) \cap (L_1 \bullet L_3)$

۳– توزیع‌پذیری عملگر تقسیم روی اجتماع و اشتراک نیز مشابه الحاق در حالت اجتماع، توزیع‌پذیر و در حالت اشتراک توزیع‌ناپذیراست. به این صورت که برای سه زبان L_1، L_2 و L_3 الفبای Σ داریم: $L_1 / (L_2 \cup L_3) = (L_1 / L_2) \cup (L_1 / L_3)$

اگر داشته باشیم $w \in L_1 / (L_2 \cup L_3)$ پس باید $z = wu \in L$ وجود داشته باشد که $u \in L_1$ یا $u \in L_3$ باشد. یعنی $Z \in L_1 / L_2$ یا $Z \in L_1 / L_3$ باشد. ولی برای اشتراک داریم: $L_1 / (L_2 \cap L_3) \neq (L_1 / L_2) \cap (L_1 / L_3)$

این موضوع تأیید می‌شود اگر L_1، L_2 و L_3 را به شکل زیر فرض بگیریم:

$$L_1 = \{10,101\} \quad L_2 = \{0\} \quad L_3 = \{01\}$$

آنگاه خواهیم داشت:

$$\begin{cases} L_1 / (L_2 \cap L_3) = L_1 / \varnothing = \varnothing \\ L_1 / L_2 \cap L_1 / L_3 = \{1\} \cap \{1\} = \{1\} \end{cases} \Rightarrow \varnothing \neq \{1\}$$

۴– توزیع‌پذیری عملگر تقسیم روی عملگر الحاق و برعکس:

توزیع‌پذیری تقسیم روی الحاق برقرار نیست؛ چرا که اگر فرض کنیم $L_1 = \{10,1\}$ و $L_2 = \{0\}$ و $L_3 = \{1\}$ در این‌صورت:

$$L_1 / (L_2 \bullet L_3) = \{10,1\} / \{01\} = \varnothing \neq L_1 / L_2 \;\bullet\; L_1 / L_3 = \{1\} \bullet \{\lambda\} = \{1\}$$

برای اثبات برقرار نبودن توزیع عمل الحاق روی تقسیم، می‌توان با فرض L_1، L_2 و L_3 به صورت زیر، نتیجه را اثبات کرد:

$$L_3 = \{0\}, L_2 = \{1\}, L_1 = \{10,1\}$$

$$L_1 \bullet (L_2 / L_3) = L_1 \bullet \varnothing = \varnothing$$

$$(L_1 \bullet L_2) / (L_1 \bullet L_3) = \{101,11\} / \{1001,101\} = \{\lambda\}$$

مثال ۷: طول کوچکترین رشته متعلق به مجموعه $1^*0^*\{0^i10^{2i}1 \mid i \ge 0\}01 \cap ^*\{0^i10^{2i}1 \mid i \ge 1\}$ کدام است؟

۱) 12　　۲) 18　　۳) 24　　۴) 19

پاسخ: گزینه «۴» کوچکترین رشته متعلق به این مجموعه عبارت است از $010010^4 10^8 1$ که طول آن 19 می‌باشد.

مثال ۸: طول دومین رشته کوچک متعلق به مجموعه مثال بالا کدام است؟

۱) 38　　۲) 69　　۳) 48　　۴) 60

پاسخ: گزینه «۲» دومین رشته کوچک به فرم مقابل و به طول 69 خواهد بود:

$$S_1 : \boxed{010010^4 10^8 10^{16} 10^{32} 1}$$

$$S_1 \text{ طول رشته } = 19 + 16 + 1 + 32 + 1 = 69$$

مثال ۹: آیا عملگرهای متمم، وارون، بستار، بعلاوه و توانی، خاصیت جابجایی دارند؟ توضیح: اگر $F(E(x)) = E(F(x))$ باشد دو عملگر $E(x), F(x)$ را جابه‌جاپذیر می‌گوییم.

پاسخ:

۱ـ جابجایی عملگرهای متمم و وارون:

برای اینکه نشان دهیم برای زبان L، تساوی $(\overline{L})^R = (\overline{L^R})$ برقرار است؛ بنابر تعریف مجموعه‌ها اگر $w \in (\overline{L})^R$ باشد، پس باید $w^R \in \overline{L}$ و در نتیجه $w^R \notin L$ لذا $(w^R)^R \notin L^R$ یعنی $w \notin L^R$ پس $w \in \overline{L^R}$ است. برعکس اگر $w \in \overline{L^R}$ باید $w \notin L^R$ لذا $w^R \notin L$ پس $w^R \in \overline{L}$ و همین‌طور $(w^R)^R \in \overline{L}^R$ یعنی $w \in \overline{L}^R$ است. پس زبان دو طرف با هم یکسان هستند. لذا این دو عملگر جابه‌جاپذیر هستند.

۲ـ جابجایی عملگرهای متمم و بستار (متمم و بعلاوه، متمم و توانی)

این دو جابه‌جاپذیر نیستند چراکه اگر داشته باشیم $L = \{0,1\}^*$ یعنی $L = \Sigma^*$ در این صورت داریم:

$$\begin{cases} \overline{L} = \varnothing \rightarrow \overline{L}^* = \{\lambda\} \\ L^* = \{0,1\}^* = \Sigma^* \rightarrow \overline{L^*} = \varnothing \end{cases} \Rightarrow \overline{L}^* \ne \overline{L^*}$$

با همین مثال جابه‌جاپذیر نبودن دو حالت دیگر نیز تأیید می‌شود.

۳ـ وارون و بستار

این دو عملگر برعکس بقیه جابه‌جاپذیر هستند یعنی تساوی $(L^*)^R = (L^R)^*$ برقرار است. برای اثبات برقراری این تساوی، نشان می‌دهیم هر عضو زبان سمت چپ در زبان سمت راست قرار دارد و برعکس. ابتدا طرف اول:

$$w \in (L^R)^* \Rightarrow w = v_1 v_2 \ldots v_n , v_i \in L^R \quad i \ge n \Rightarrow$$
$$v_i^R \in L \Rightarrow w^R = v_n^R \ldots v_1^R \in L^* \Rightarrow (w^R)^R \in (L^*)^R \Rightarrow w \in L^R$$

و حال برعکس:

$$w \in (L^*)^R \Rightarrow w^R \in L^* \Rightarrow w^R = (v_1 \ldots v_n)^R \in L^*$$
$$\Rightarrow v_n^R, \ldots, v_1^R \in L \Rightarrow v_1, \ldots, v_n \in L^R \Rightarrow w = v_1 \ldots v_n \in L^R \Rightarrow w^* \in (L^R)^*$$

مشابه همین روش برای اثبات جابه‌جاپذیر بودن وارون و بعلاوه و همچنین وارون و توانی انجام می‌شود و از این طریق می‌توان درستی این دو حالت را تأیید کرد.

مثال ۱۰: کدام‌یک از تساوی‌های زیر برقرار است؟

الف) $L_1/(L_2 \cap L_3) = L_1/L_2 \cup L_1/L_3$ ب) $L_1/(L_2 \bullet L_3) = L_1/L_2 \bullet L_1/L_3$ ج) $L_1 \bullet (L_2/L_3) = (L_1 \bullet L_2)/L_1 \bullet L_3$

۱) الف و ب ۲) ب وج ۳) الف و ج ۴) هیچ‌کدام.

پاسخ: گزینه «۴» برای رد تساوی عبارت الف، فرض کنید $L_1 = \{10,11\}$ و $L_2 = \{0\}$ و $L_3 = \{1\}$ باشد.

$$L_1/(L_2 \cap L_3) = \{10,11\}/\varnothing = \varnothing \neq \{1\} = \{1\} \cup \{1\} = L_1/L_2 \cup L_1/L_3$$

برای رد تساوی ب نیز با فرض بالا، عبارت روبرو را داریم:

$$L_1/(L_2 \bullet L_3) = \{10,11\}/\{01\} = \varnothing \neq \{11\} = \{1\} \cup \{1\} = L_1/L_2 \bullet L_1/L_3$$

در مثال۶ قسمت ۴ عدم تساوی ج اثبات شده است.

مثال ۱۱: اگر زبان $L_1 = \{11,00\}$ و $L_2 = \{10,01\}$ را داشته باشیم کدام تساوی زیر برقرار خواهد بود؟

۱) $\bullet^R(L_1,L_2) = \bullet(L_1^R, L_2^R)$

۲) $\bullet^R(L_1,L_2) = \bullet(L_2^R, L_1^R)$

۳) $\overline{\bullet}(L_1,L_2) = \bullet(\overline{L}_1, \overline{L}_2)$

۴) $\overline{\bullet}(L_1,L_2) = \bullet(\overline{L}_1 \setminus \overline{L}_2, \overline{L}_2 - L_1)$

پاسخ: گزینه «۲» تساوی ۱ برای هیچ دوزبانی برقرار نیست. گزینه درست، تساوی ۲ است زیرا برای هر دوزبانی، از جمله زبان‌های این سؤال برقرار است. گزینه۳و۴ تنها در زبان‌های خاصی برقرار است ولی برای هیچ‌کدام از زبان‌های این سؤال، برقرار نمی‌باشد چرا که داریم $\bullet(L_1,L_2) = \{1101,1110,0010,0001\}$ که در نتیجه $1101 \notin \overline{\bullet}(L_1,L_2)$ است اما $1101 \in \overline{L}_1 \setminus L_2$ و $\lambda = \overline{L}_2 - L_1$ می‌باشد، لذا $1101 \in \bullet(\overline{L}_1 \setminus L_2, \overline{L}_2 \setminus L_1)$ است که نشان می‌دهد تساوی ۴ برقرار نیست. همین مثال نیز نامساوی بودن تساوی ۳ را نشان می‌دهد.

مثال ۱۲: کدام‌یک از تساوی‌های زیر برقرار است؟ عملگر \ در اینجا به معنای تفریق است یعنی اگر L_1 و L_2 دو زبان روی الفبای Σ باشند. آن‌گاه:

$$\mathbf{L_1 \setminus L_2 = \{w \in \Sigma^* \mid w \in L_1 \text{ and } w \notin L_2\}}$$

۱) $L_1 \setminus (L_2 \bullet L_3) = (L_1 \setminus L_2) \bullet (L_1 \setminus L_3)$

۲) $L_1 \setminus (L_2 \bullet L_3) = (L_1 \setminus L_2)/(L_1 \setminus L_3)$

۳) $L_1 \setminus (L_2 \cap L_3) = (L_1 \setminus L_2) \cap (L_1 \setminus L_3)$

۴) $L_1 \setminus (L_2 \cup L_3) = (L_1 \setminus L_2) \cap (L_1 \setminus L_3)$

پاسخ: گزینه «۴» این سؤال که توزیع‌پذیری عملگر تقسیم روی عملگرها را بررسی کرده است. برای رد هر یک از تساوی‌ها، یک مثال نقض ارائه می‌شود. با معرفی سه زبان L_1، L_2 و L_3 و ساخت هر یک از طرفین برقرار نبودن تساوی نشان داده می‌شود.

الف)

$$L_1 = \{11,00\} \quad L_2 = \{11\} \quad L_3 = \{00\}$$
$$L_1 \setminus (L_2 \bullet L_3) = \{11,00\} \neq (L_1 \setminus L_2) \bullet (L_1 \setminus L_3) = \{0011\}$$

ب)

$$L_1 = \{101,010\} \quad L_2 = \{101\} \quad L_3 = \{010,1\}$$
$$L_1 \setminus (L_2 / L_3) = \{101\} \neq (L_1 \setminus L_2)/(L_1 \setminus L_3) = \{\lambda\}$$

ج)

$$L_1 = \{11,00\} \quad L_2 = \{11\} \quad L_3 = \{00\}$$
$$L_1 \setminus (L_2 \cap L_3) = \{11,00\} \neq (L_1 \setminus L_2) \cap (L_1 \setminus L_3) = \varnothing$$

د) این معادله برقرار است چرا که اگر داشته باشیم:

$$L_1 = \{11,00,10,01\} \quad L_2 = \{11,00\} \quad L_3 = \{10,01\}$$
$$L_1 \setminus (L_2 \cup L_3) = \varnothing = (L_1 \setminus L_2) \cap (L_1 \setminus L_3)$$

دقت شود که اگر سمت راست معادله، به جای عمل $\cap$ عمل $\cup$ را بگذاریم تساوی دیگر برقرار نیست.

گرامر

گرامر زبان، مجموعه‌ای است از قوانین که ساخت رشته‌های متعلق به زبان را کنترل می‌کند. گرامر فقط نحو (دستور، ساختار) زبان را کنترل می‌کند و روی معنی کنترلی ندارند. ساخت رشته‌های زبان توسط گرامر را، اشتقاق می‌گویند.

گرامر زبان به‌صورت $G=(\Sigma,V,R,S)$ بیان می‌شود. Σ نشانه الفبای زبان، V علامت مجموعه متغیرها، R شامل قواعد زبان است که به شکل $R\subset(V\bigcup\Sigma)^{+}\times(V\bigcup\Sigma)^{*}$ نشان داده می‌شود و S نماد آغازگر اشتقاق زبان است که $S\in V$ می‌باشد. $L(G)$ زبان تعریف شده توسط گرامر مشخص G می‌باشد. اگر دو گرامر G_1 و G_2 زبان‌های یکسان تولید نمایند آن را معادل می‌گویند، به عبارت دیگر $L(G_1)=L(G_2)$ می‌باشد.

مثال ۱۳: گرامر مقابل با شماره قوانین مشخص شده مفروض است. (علامت | به معنی "یا" می‌باشد)

(1) $E\to E+T.$

(2) $E\to T.$

(3) $T\to(E)\,|\,1\,|\,2\,|\,3.$

الف ـ مجموعه‌های Σ، R ،V و S را تعیین کنید.

ب ـ آیا رشته $(2+(3))$ متعلق به $L(G)$ است؟

☑ **پاسخ:** الف) $\Sigma=\{+,(,),1,2,3\}$ $\qquad V=\{E,T\}$ $\qquad S=\{E\}$

ب) $E\overset{2}{\Rightarrow}T\overset{3}{\Rightarrow}(E)\overset{1}{\Rightarrow}(E+T)\overset{3}{\Rightarrow}(E+(E))\overset{2}{\Rightarrow}(T+(E))\overset{3}{\Rightarrow}(2+(E))\overset{2}{\Rightarrow}(2+(T))\overset{3}{\Rightarrow}(2+(3))$

این رشته، متعلق به زبان گرامر است. از این رو: $E\overset{*}{\Rightarrow}(2+(3))$

براساس ساختار یک گرامر می‌توان تعداد مراحل اشتقاق یک جمله از گرامر داده شده را پیش از ساخت آن مشخص کرد. برای این منظور گرامرها در ساختارهای استاندارد مورد بررسی قرار می‌گیرند که در ادامه خواهد آمد.

طبقه‌بندی گرامرها

زبان‌ها بر اساس قوانین بیانگر گرامر آن زبان، به چهار گروه تقسیم می‌شوند. اگر برای زبانی حداقل یک گرامر از یک گروه وجود داشته باشد، زبان متعلق به آن گروه است. این دسته‌بندی زبان‌ها، طبقه‌بندی چامسکی نام دارد و به این صورت است:

الف ـ گرامرهای منظم (Regular) (نوع سه)

بر اساس گرامرهای باقاعده ساخته می‌شوند که قوانین گروه آن‌ها چنین است:

$A\to a\quad;A\in V,a\in\Sigma$

$A\to aA'\,;A,A'\in V,a\in\Sigma$

$S\to\lambda$ (اگر و فقط اگر $\lambda\in L$ باشد)

ماشین‌های متناهی (DFA)، پذیرنده‌ی زبان این نوع گرامرها است.

ب ـ گرامرهای مستقل از متن (Context Free): (نوع دو)

گرامرهای مستقل از متن به شکل مقابل تولید شده و ماشین‌های پشته‌ای (PDA) پذیرنده زبان این نوع گرامرها هستند.

$S\to\lambda$ (اگر و فقط اگر $\lambda\in L$)

$A\to u\ ;\ A\in V,u\in(\sum\bigcup V)^{+}$

ج ـ گرامرهای حساس به متن (Context Sensitive): (نوع یک)

$G=(\Sigma,V,R,S).$

قوانین گرامر در این زبان چنین تعریف می‌شود:

$R=\{u\to v\,|\,u\in(V\bigcup\Sigma)^{+},v\in(V\bigcup\Sigma)^{+},\mathrm{Length}(u)\le\mathrm{length}(v)\}$

زبان‌های وابسته به متن توسط ماشین کراندار خطی پذیرفته می‌شوند. این گرامرها فاقد قانون λ هستند.

د ـ گرامرهای بدون محدودیت (Unrestricted) : (نوع صفر)

گرامر زبان‌های بدون محدودیت به این‌گونه تعریف می‌شوند:

$G=(\Sigma,V,R,S).$

$R=\{u\to v\,|\,u\in(V\bigcup\Sigma)^{+},v\in(V\bigcup\Sigma)^{*})$

سمت راست این گرامر، برخلاف گرامر وابسته (حساس) به متن، می‌تواند λ هم باشد. دقت کنید که هر زبانی یک زبان بدون محدودیت است. زبان گرامرهای بدون محدودیت توسط ماشین تورینگ ساخته می‌شود.

نکته ۴: همواره رابطه‌های زیر برقرارند:

زبان منظم $\supset$ زبان مستقل از متن $\supset$ زبان حساس به متن $\supset$ زبان بدون محدودیت

مثال ۱۴: گرامر مقابل با حذف کدام قانون، وابسته به متن می‌شود؟

$S \to a \mid bSBC$
$T \to b\ TB \mid C$
$TCB \to bT$
$CT \to CB$
$B \to bB \mid C$

۱) $T \to CB$
۲) $CT \to CB$
۳) $TCB \to bT$
۴) هیچکدام

پاسخ: گزینه «۳» در گرامر وابسته به متن، هر قانون به‌شکل $u \to v$ است که باید طول u کمتر یا مساوی طول v باشد.

مثال ۱۵: برای زبان‌های L_1 روی $\sum_1 = \{a\}$ و L_2 روی $\sum_2 = \{b\}$، دو تبدیل زبانی روی $\Sigma = \{0\}$ به شکل زیر تعریف می‌گردد:

$$L_1 \oplus L_2 = \{0^{m+n} \mid a^m \in L_1 \ \text{AND} \ b^n \in L_2\}$$

$$L_1 \ominus L_2 = \{0^{m-n} \mid a^m \in L_1 \ \text{AND} \ b^n \in L_2\}$$

الف) اگر $A = \{aa\}$ و $B = \{bb\}$، $A \oplus B$ و $A \ominus B$ را مشخص کنید.

ب) اگر A و B دو زبان متناهی باشد، کاردینالیتی $A \oplus B$ و $A \ominus B$ را تعیین کنید.

پاسخ: الف ـ $A \ominus B = \{\lambda\}$ و $A \oplus B = \{0000\}$

ب ـ فرض کنید $A = \{a^{m1}, \dots, a^{m_k}\}$ و $B = \{b^{n_1}, \dots, b^{n_p}\}$ باشد به طوریکه $m_i, n_j \in \mathbb{N}$ و $1 \le i \le k$ و $1 \le j \le p$، در این صورت $|A \oplus B| \le kp$ و $|A \ominus B| \le kp$ است و این زبان‌ها نیز متناهی هستند. به طور دقیق داریم:
$|A \oplus B| \le \max(m_i) + \max(n_j)$, $|A \ominus B| \le \max(m_i) - \min(n_j)$است.

مثال ۱۶: به کمک الفبای $\sum = \{0\}$، دو زبان زیر را روی الفبای $\{a, b\}$ تعریف می‌کنیم:

$$SP(L) = \{a^m b^n \mid 0^{m+n} \in L\}$$

$$SM(L) = \{a^m b^n \mid 0^{m-n} \in L\}$$

اگر $L_1 = \{00, 0000\}$ باشد حاصل $SP(L_1)$ و $SM(L_1)$ را به دست آورید؟

پاسخ:

$$SP(L_1) = \{ab, aa, bb, aaaa, aaab, aabb, abbb, bbbb\}$$

$$SM(L_1) = \{a^i b^j \mid (i-j=2) \ \text{OR} (i-j=4)\}$$

مثال ۱۷: اگر A مجموعه پیشوند و B مجموعه پسوند و C مجموعه زیر رشته‌های یک رشته مفروض باشند، کدام رابطه زیر نادرست است؟

۱) $A \subseteq C$
۲) $A \bigcup B \subseteq C$
۳) دربرخی رشته‌ها A, B, C مساویند
۴) هیچ‌کدام

پاسخ: گزینه «۴» اجتماع مجموعه پیشوندها و پسوندها لزوماً مجموعه زیر رشته‌ها نیست. در برخی رشته‌ها مثل abc، زیر رشته $'b'$ نه پسوند است و نه پیشوند. درباره گزینه ۳، اگر $w = aaa$ فرض شود، $A = B = C$ خواهد بود. در ضمن همواره داریم :

$$\begin{cases} B \subseteq C \\ A \subseteq C \\ A \bigcup B \subseteq C \end{cases}$$

مثال ۱۸: مجموعه پیشوند زبان $L = \{ab, baa\}$ را مشخص کنید.

پاسخ:

$$\text{prefix}(L) = \{\lambda, a, ab, b, ba, baa\}$$