فصل ۱ مفاهیم پایه

در ایـــن فصل به بحث و بررســـی مفهـــوم پایه در درس نظریه خواهیم پرداخت . مجموعه‌ها از مفاهیـــم اصلی و بنیادین در ریاضیات هســـتند که کاربردهای بســـیار زیادی در علوم مهندسی دارند. در این درس مجموعه‌ها نقش اساسی را بازی می‌کنند. پس از بررسی کاربرد مجموعه‌ها در تعاریف اولیه زبان به بررسی گرامرها و ماشین‌ها خواهیم پرداخت.

اگـــر مجموعـــه را به عنوان گردایه‌ای از عناصر در نظـــر بگیریم آنگاه روابط زیر را می‌توان بر روی مجموعه‌ها تعریف کرد(S و R دو مجموعه دلخواه و U مجموعه مرجع هستند):

$$S-R=\{x \mid x\in S \text{ and } x\notin R\}$$

$$S\cup R=\{x \mid x\in S \text{ or } x\in R\}$$

$$S\cap R=\{x \mid x\in S \text{ and } x\in R\}$$

$$\overline{S}=U-S$$

روابط زیر روی مجموعه ها همواره برقرار است:

1)	$R\cup S=S\cup R$	8)	$R\cup\varnothing=\varnothing\cup R=R$
2)	$R\cap S=S\cap R$	9)	$R\cap\varnothing=\varnothing\cap R=\varnothing$
3)	$R.S\neq S.R$	10)	$(R\cup S)\cup P=R\cup(S\cup P)$
4)	$R-S\neq S-R$	11)	$(R\cap S)\cap P=R\cap(S\cap P)$
5)	$\overline{R\cap S}=\overline{R}\cup\overline{S}$	12)	$R\cup(S\cap P)=(R\cup S)\cap(R\cup P)$
6)	$\overline{R\cup S}=\overline{R}\cap\overline{S}$	13)	$R\cap(S\cup P)=(R\cap S)\cup(R\cap P)$
7)	$R\cup R=R$	14)	$(R-S)-P\neq R-(S-P)$

رو ش‌های اثبات

برای اثبات قضایا روش‌های متعددی وجود دارد. در این قسمت به بیان چند روش بسیار مهم در اثبات قضایا می‌پردازیم:

۱. اثبات مســـتقیم(direct proof): با کمک ترکیب گزاره‌های منطقی، تعاریف و قضایایی که از قبل وجود دارد حکم قضیه اثبات می‌شود. برای نمونه قضیه زیر را می‌توان با استفاده از اثبات مستقیم ثابت کرد.

قضیه ۱-۱: اگر a و b دو عدد زوج باشند آنگاه $a+b$ نیز زوج است.

اثبات: چون a و b هر دو زوج هستند لذا می‌توان آنها را به صورت $a=2x$ و $b=2y$ بیان کرد. در نتیجه $a+b=2x+2y=2(x+y)$ و لذا $a+b$ زوج است.

۲. اثبات به روش جابجایی(proof by transposition): در این روش برای اثبات گزاره $p \Rightarrow q$ از اثبات گزاره هم ارز آن یعنی $\sim q \Rightarrow \sim p$ استفاده می‌شود.

۳. اثبات به روش برهان خلف(proof by contradiction): در این روش برای اثبات درستی یک گزاره فرض می‌کنیم که آن گزاره نادرست باشد و سپس از نادرست بودن آن به تناقض خواهیم رسید. برای نمونه برای اثبات اینکه عدد $\sqrt{2}$ یک عدد گویا نیست فرض می‌کنیم که $\sqrt{2}$ گویا باشد و لذا باید $\sqrt{2}=\frac{a}{b}$ که در آن a و b فاکتور مشترک نداشته باشند. بنابر این:

$$a=b\sqrt{2} \Rightarrow a^2=2b^2$$

در نتیجه a^2 زوج است و لذا a زوج است بنابر این می‌توان آن را به صورت $a=2k$ در نظر گرفت و لذا:

$$(2k)^2=2b^2 \Rightarrow 4k^2=2b^2 \Rightarrow 2k^2=b^2$$

و بنابراین b^2 زوج بوده و لذا b زوج است بنابراین می‌توان آن را به صورت $b=2k'$ در نظر گرفت. در نتیجه عدد 2 فاکتور مشترک a و b است که این تناقض است چون فرض کرده بودیم که a و b فاکتور مشترک ندارند.

۴. اثبات به روش استقرای ریاضی(proof by mathematical induction): در این روش برای اثبات گزاره $\forall n \in N, n \geq k : P(n)$ ابتدا درستی $P(k)$ را انجام داده و سپس درستی گزاره زیر را انجام می‌دهیم.

$$\forall m \in N, m \geq k : P(m) \Rightarrow P(m+1)$$

به بیان دیگر اثبات از طریق استقرای ریاضی به معنی استفاده از قضیه زیر است:

$$\Big(P(k), (\forall m \in N, m \geq k)\big(P(m) \Rightarrow P(m+1)\big)\Big) \Rightarrow (\forall n \in N, n \geq k)P(n)$$

۵. اثبات به روش ساخت(proof by construction): معمولاً این روش در اثبات گزاره هایی کاربرد دارد که در آنها بر وجود چیزی تاکید می‌شود. در این روش برای نشان دادن وجود عددی یا الگوریتمی آن عدد یا الگوریتم ارائه می‌شود. برای نمونه به قضیه زیر دقت کنید:

قضیه ۱-۲: تعداد نامتناهی عدد گویا بین دو عدد 0 و 1 وجود دارد.

اثبات: مجموعه $\left\{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots\right\}$ دارای نامتناهی عدد گویا است که همگی در بازه $(0,1)$ قرار دارند.

۶. اثبات به روش جستجوی کامل(proof by exhaustion): در این روش حکم قضیه به تعدادی حالت متناهی تقسیم بندی می‌شود و هر قسمت جداگانه اثبات می‌شود. گاهی ممکن است تعداد حالات مختلف بسیار زیاد باشد.

۷. اثبات به روش ترکیباتی(combinatorial proof): در این روش نشان داده می‌شود که دو عبارت، هر کدام از آنها از راهی متفاوت با دیگری تعداد یک کمیت را می‌شمارند و لذا با هم برابرند. در برخی اوقات مشخص کردن تعداد عناصر یک مجموعه مانند A دارای پیچیدگی خاص خود است و لذا می‌توان به جای آن با نشان دادن وجود یک نگاشت دو سویه بین A و مجموعه دیگری مانند B ثابت کرد که تعداد عناصر دو مجموعه A و B با هم برابرند و لذا با شمردن تعداد عناصر B تعداد عناصر A را نیز مشخص کرد. برای نمونه به قضیه زیر دقت کنید:

قضیه ۱-۳: ثابت کنید تعداد درخت‌های دودویی که با n گره می‌توان ساخت برابر با $\frac{\binom{2n}{n}}{n+1}$ است.

در اثبات قضیه بالا می‌توان نشان داد که یک نگاشت دوسویه بین مجموعه چنین درخت‌هایی و مجموعه عبارات پرانتز دار خوش ساخت است که با n پرانتز باز و n پرانتز بسته می‌توان ایجاد کرد و سپس ثابت کرد که تعداد چنین عبارات پرانتز دار خوش

ساختی برابر با عدد مورد نظر است و لذا قضیه اثبات می‌شود.

به عنوان یک نمونه دیگر به قضیه زیر دقت کنید:

قضیه ۱-۴: ثابت کنید تعداد زیر درخت‌های پوشای گراف K_n (گراف کامل با n راس) برابر با n^{n-2} است.

اثبات: در اثبات این قضیه نشان داده می‌شود که یک نگاشت دوسویه بین مجموعه چنین درختهایی و مجموعه رشته‌های به طول $n-2$ ساخته شده از یک الفبای n حرفی وجود دارد. حال از آنجا که تعداد چنین رشته‌هایی برابر با n^{n-2} است لذا قضیه ثابت می‌شود.

در این درس به بحث و بررسی سه مفهوم خواهیم پرداخت:

- **زبان**
- **ماشین**
- **گرامر**

حال هر یک از مفاهیم بالا را به طور مختصر معرفی کرده و با آنها آشنا می‌شویم. در فصل‌های بعدی کتاب همین مفاهیم را به طور دقیقتر مورد بررسی قرار می‌هیم.

زبان(Language)

اگر Σ یک مجموعه متناهی از نمادها باشد و از آن به عنوان الفبا استفاده کنیم آنگاه هر دنباله متناهی از نمادهای الفبا، یک رشته (string) نامیده می‌شود. مثال:

$\Sigma=\{a,b\}$

$w_1=aab$

$w_2=bb$

تعداد نمادهای یک رشته را طول آن رشته می‌نامیم. طول رشته w با $|w|$ نشان داده می‌شود. رشته به طول صفر را با λ نشان می‌دهند. اتصال یا الحاق دو رشته w_1 و w_2 را با $w_1.w_2$ نشان می‌دهیم. مثال:

$w_1.w_2=aabbb$

بدیهی است که:

$\forall w:\lambda.w=w.\lambda=w$

اگر رشته w به طول n به صورت زیر تعریف شده باشد:

$w=a_1a_2\,..\,a_n$

آنگاه هر رشته‌ای به صورت $a_1.a_2\ldots a_i$ برای هر $0\le i\le n$ یک پیشوند(Prefix) از w و هر رشته‌ای به صورت $a_i.a_2\ldots a_n$ برای هر $0\le i\le n$ یک پسوند(Suffix) از w نامیده می‌شوند.مجموعه رشته‌هایی که پیشوند رشته w هستند را با $Prefix(w)$ و مجموعه رشته‌هایی که پسوند رشته w هستند را با $Suffix(w)$ نشان می‌دهند. همچنین معکوس رشته w که با w^R نشان داده می‌شود به صورت زیر تعریف می‌شود:

$w^R=a_na_{n-1}\ldots a_1$

همچنین اگر n یک عدد طبیعی باشد w^n به صورت زیر تعریف می‌شود:

$w^n=\underbrace{w.w.\ldots w}_{n\ \text{بار}}$

$w^0=\lambda$

اگر Σ یک الفبا باشد آنگاه مجموعه کلیه رشته‌های قابل ایجاد با نمادهای Σ را با Σ^* نشان می‌دهیم و همچنین:

$$\Sigma^+ = \Sigma^* - \{\lambda\}$$

هر زیر مجموعه‌ای از Σ^* که مجموعه‌ای از رشته‌های ایجاد شده از نمادهای Σ است را یک زبان می‌نامیم. مثال:

$$L_1 = \{\lambda, a, aab\}$$

$$L_2 = \{bb, bab\}$$

$$L_3 = \{a^n b^n \mid n \geq 0\}$$

اتصال یا الحاق دو زبان L_1 و L_2 به صورت زیر تعریف می‌شود:

$$L_1.L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$$

مثال ۱:

$$L_1 = \{\lambda, a, aab\}$$

$$L_2 = \{bb, bab\}$$

$$L_1.L_2 = \{bb, bab, abb, abab, aabbb, aabbab\}$$

معکوس زبان L که با L^R نشان داده می‌شود به صورت زیر تعریف می‌شود:

$$L^R = \{w \mid w^R \in L\}$$

مثال ۲:

در مثال قبلی می‌توان معکوس زبان را به صورت زیر محاسبه کرد:

$$L_1^R = \{\lambda, a, baa\}$$

همچنین اگر n یک عدد طبیعی باشد L^n به صورت زیر تعریف می‌شود:

$$L^n = \underbrace{L.L.\ldots L}_{\text{بار } n}$$

$$L^0 = \{\lambda\}$$

اگر L یک زبان دلخواه باشد آنگاه:

$$L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$$

$$L^+ = L^1 \cup L^2 \cup \ldots$$

به روشنی می‌توان دید که:

$$\lambda \in L \quad \Rightarrow \quad L^* = L^+$$

$$\lambda \notin L \quad \Rightarrow \quad L^* \neq L^+$$

روابطی بر روی رشته‌ها

جدول ۱-۱

1)	$(u.v)^R = v^R.u^R$	4)	$(u.v)^n \neq u^n.v^n$
2)	$(\omega^R)^R = \omega$	5)	$\left(\omega^{\overbrace{R\cdots R}^{n+1}}\right) = \begin{cases}\omega; n \text{ is even}\\ \omega; n \text{ is odd}\end{cases}$
3)	$(\omega^n)^R = (\omega^R)^n$	6)	$\left(\omega^{\overbrace{R\cdots R}^{n}}\right) = \begin{cases}\omega; n \text{ is even}\\ \omega^R; n \text{ is odd}\end{cases}$

روابطی بر روی زبان ها

جدول ۱-۲

1)	$L.\{\lambda\} = \{\lambda\}.L = L$
2)	$L.\varnothing = \varnothing.L = \varnothing$
3)	$L^+ = LL^*$
4)	$L^* = L^+ \cup \{\lambda\}$
5)	$L_1.L_2 \neq L_2.L_1$
6)	$(L^*)^* = L^*$
7)	$(L^*)^+ = (L^+)^* = L^*$
8)	$L^*.L^* = L^*$
9)	$L_1 \subseteq L_2 \Rightarrow L_1^* \subseteq L_2^*$
10)	$L_1(L_2 \cup L_3) = L_1L_2 \cup L_1L_3$
11)	$L_1(L_2 \cap L_3) \neq L_1L_2 \cap L_1L_3$
12)	$(L_1.L_2)^* \neq L_1^*.L_2^*$, $(L_1.L_2)^+ \neq L_1^+.L_2^+$
13)	$(L_1 \cup L_2)^* \neq L_1^* \cup L_2^*$, $(L_1 \cup L_2)^+ \neq L_1^+ \cup L_2^+$
14)	$(L_1 \cap L_2)^* \neq L_1^* \cap L_2^*$, $(L_1 \cap L_2)^+ \neq L_1^+ \cap L_2^+$
15)	$(L_1 - L_2)^* \neq L_1^* - L_2^*$, $(L_1 - L_2)^+ \neq L_1^+ - L_2^+$

گرامر

گرامر ابزاری برای تولید زبان محســوب می‌شـود و یک گرامر به صورت چهارتایی $G = (V, T, S, P)$ تعریف می‌شود که در آن V مجموعه متناهی و غیرتهی از متغیرها، T مجموعه متناهی و غیر تهی از نمادهای الفبا، $S \in V$ متغیر آغازین و P مجموعه‌ای از قوانین گرامر است که هر قانون به شکل $x \rightarrow y$ است که در آن

$x \in (V \cup T)^+$ و $y \in (V \cup T)^*$

معمولاً برای سادگی در نمایش، یک گرامر را فقط توسط قوانین آن نشان می‌دهند. در قوانین گرامر حروف بزرگ نشان‌دهنده متغیرها، حروف کوچک نشان‌دهنده پایانه‌ها یا نمادهای الفبا و متغیر سمت چپ قانون اول که معمولاً S است به عنوان متغیر آغازین می‌باشد. مثال زیر یک گرامر را نشان می‌دهد:

$$S \to aABb$$
$$aA \to Ba$$
$$B \to bb$$

توسط یک گرامر می‌توان رشته‌هایی ساخته شده از نمادهای الفبا را تولید کرد که مجموعه کل رشته‌های تولید شده توسط یک گرامر را زبان تولید شده توسط آن گرامر می‌نامند. برای تشخیص چگونگی تولید رشته‌ها توسط قوانین گرامر باید از متغیر آغازین S شروع کرده و توسط الگوریتم زیر به رشته‌های قابل تولید رسید. اگر uxv رشته‌ای متشکل از پایانه‌ها و متغیرها باشد و در قوانین گرامر قانونی به صورت $x \to y$ موجود باشد آنگاه می‌توان uxv را به uyv تبدیل کرد و نوشت:

$$uxv \Rightarrow uyv$$

به این عمل تبدیل، مشتق شدن uxv به uyv گفته می‌شود. حال اگر دنباله‌ای از این تبدیلات به صورت زیر وجود داشته باشد:

$$w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n$$

می‌توان برای سادگی آن را با $w_1 \overset{*}{\Rightarrow} w_n$ نشان داد که خوانده می‌شود: w_1 طی صفر یا بیشتر مرحله اشتقاق به w_n تبدیل می‌شود. به هر کدام از w_i ها در طی مراحل اشتقاق یک شکل جمله‌ای (Sentential Form) گفته می‌شود. نماد $\overset{+}{\Rightarrow}$ نشان‌دهنده یک یا بیشتر مرحله اشتقاق است. رشته‌هایی که توسط اشتقاق از روی متغیر آغازین یک گرامر تولید شده و فقط از پایانه ساخته شده است را یک رشته تولید شده توسط آن گرامر می‌نامند. حال زبان تولید شده توسط گرامر G به صورت زیر تعریف می‌شود:

$$L(G) = \left\{ w \in T^* \mid S \overset{*}{\Rightarrow} w \right\}$$

مثال۳: زبان گرامر زیر:

$$G:$$
$$S \to aSb$$
$$S \to \lambda$$

برابر با زبان زیر است:

$$L(G) = \left\{ a^n b^n \mid n \geq 0 \right\}$$

نکته: اگر دو قانون به شکل زیر داشته باشیم:

$$u \to x$$
$$u \to y$$

آنگاه برای سادگی می‌توان آن را به صورت $u \to x \mid y$ نمایش داد.

مثال۴: زبان گرامر زیر:

$G:$

$S \rightarrow aSa \mid bsb \mid \lambda$

برابر با زبان زیر است:

$$L(G) = \left\{ww^R \mid w \in \{a,b\}^*\right\}$$

مثال۵: زبان گرامر زیر:

$G:$

$S \rightarrow aSb \mid bSa \mid SS \mid \lambda$

برابر با زبان زیر است:

$$L(G) = \left\{w \in \{a,b\}^* \mid n_a(w) = n_b(w)\right\}$$

> **نکته:** $n_a(w)$ نشان‌دهنده تعداد a در رشته w است.

مثال۶: زبان گرامر زیر:

$G:$

$S \rightarrow aSb \mid SS \mid \lambda$

برابر با زبان زیر است:

$$L(G) = \left\{w \in \{a,b\}^* \mid \forall u \in \text{Prefix}(w) : n_a(u) \geq n_b(u)\right\}$$

مثال۷: زبان گرامر زیر:

$G:$

$S \rightarrow aSbS \mid \lambda$

برابر با زبان زیر است:

$$L(G) = \left\{w \in \{a,b\}^* \mid \forall u \in \text{Prefix}(w) : n_a(u) \geq n_b(u)\right\}$$

مثال۸: زبان گرامر زیر:

$G:$

$S \rightarrow AB$

$A \rightarrow aA \mid \lambda$

$B \rightarrow bB \mid \lambda$

برابر با زبان زیر است:

$$L(G) = \left\{a^m b^n \mid m,n \geq 0\right\}$$

مثال۹: زبان گرامر زیر:

$G:$

$S \rightarrow aAbc \mid \lambda$

$aA \rightarrow aaBb$

$$Bb \to bB$$
$$Bc \to Acc$$
$$bA \to Ab$$
$$A \to \lambda$$

برابر با زبان زیر است:

$$L(G) = \{a^n b^n c^n \mid n \geq 0\}$$

دسته‌بندی گرامرها

بر روی گرامرها می‌توان دسته‌بندی‌های مختلفی انجام داد که یکی از این دسته‌بندی‌ها به صورت زیر است:

۱) گرامرهای منظم(regular):

i) گرامر خطی از چپ(left linear):

$$A \to u \text{ or } A \to Bu$$
$$\begin{cases} A, B \in V \\ u \in T^* \end{cases}$$

ii) گرامر خطی از راست(right linear):

$$A \to u \text{ or } A \to uB$$
$$\begin{cases} A, B \in V \\ u \in T^* \end{cases}$$

۲) گرامرهای مستقل از متن(context free):

$$A \to x$$
$$\begin{cases} A \in V \\ x \in (V \cup T)^* \end{cases}$$

گرامرهای مستقل از متن را نیز می‌توان به صورت زیر دسته بندی کرد:

I) گرامر خطی(linear):

$$A \to u \text{ or } A \to uBv$$
$$\begin{cases} A, B \in V \\ u, v \in T^* \end{cases}$$

II) گرامر چامسکی:

$$A \to BC \text{ or } A \to a$$
$$\begin{cases} A, B, C \in V \\ a \in T \end{cases}$$

III) گرامر گریباخ:

$$A \to ax$$
$$\begin{cases} A \in V \\ a \in T \\ x \in (V \cup T)^* \end{cases}$$

IV) گرامر ساده (S-Grammar):

$$A \to ax$$
$$\begin{cases} A \in V \\ a \in T \\ x \in (V \cup T)^* \end{cases}$$

و همچنین در تمامی قوانین زوج (A, a) حداکثر یکبار وجود داشته باشد.

۳) گرامرهای حساس به متن(context sensitive):

$$x \to y$$
$$\begin{cases} x, y \in (V \cup T)^+ \\ |x| \le |y| \end{cases}$$

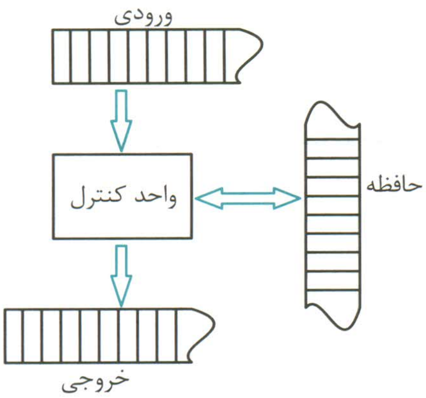
۴) گرامرهای بدون محدودیت یا نامقید(unrestricted):

$$x \to y$$
$$\begin{cases} x \in (V \cup T)^+ \\ y \in (V \cup T)^* \end{cases}$$

ماشین(Automaton)

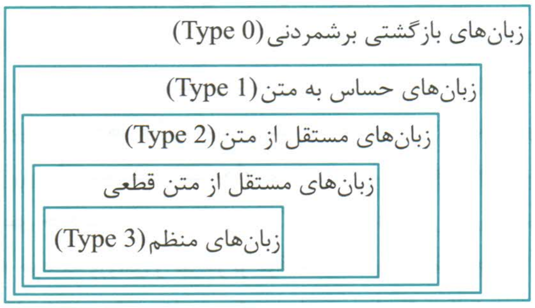
ماشین‌ها را می‌توان با دو دیدگاه نگریست. ماشین‌هایی که برای پذیرش زبان‌ها استفاده می‌شوند و آنها را پذیرنده(Accepter) می‌نامند و ماشین‌هایی که برای محاسبه استفاده می‌شوند و قادرند با خواندن ورودی آن را با محاسباتی به خروجی مورد نظر تبدیل کنند که آنها را تراگذر (Transducer) می‌نامند. ماشین‌های پذیرنده دارای خروجی دو حالته هستند که نشان دهنده موفقیت یا عدم موفقیت در پذیرش رشته ورودی می‌باشد. یک ماشین را می‌توان به صورت دیاگرام زیر در نظر گرفت. ماشین‌های مختلف با تغییراتی در همین ماشین کلی ایجاد می‌شوند. ماشین پذیرنده‌ای که دارای بخش حافظه نبوده و خروجی آن نیز فقط دارای ۲ حالت (پذیرش یا عدم پذیرش) باشد را پذیرنده متناهی(Finite Automata)، ماشین پذیرنده‌ای که حافظه آن به صورت پشته بوده و خروجی آن نیز فقط دارای ۲ حالت (پذیرش یا عدم پذیرش) باشد را ماشین پشته‌ای(Pushdown Automata)، ماشینی که دارای حافظه از دو سر محدود با قابلیت خواندن و نوشتن باشدرا ماشین کراندار خطی(Linear Bounded Automata) و ماشینی که فقط دارای حافظه‌ای نامحدود با قابلیت خواندن و نوشتن باشد را ماشین تورینگ (Turing Machine) می‌نامند. در ماشین‌های کراندار خطی و تورینگ ورودی در ابتدای کار بر روی حافظه قرار داده می‌شود و ماشین قادر است بر روی ورودی به سمت چپ یا راست حرکت کرده و در صورت نیاز حتی بر روی آن بنویسد. ماشین‌های متناهی و پشته‌ای فقط قادرند نوار ورودی را خوانده و با خواندن هر نماد ورودی به سمت انتهای ورودی حرکت می‌کنند و قادر به برگشت به عقب بر روی ورودی نیستند ولی ماشین‌های کراندار خطی و تورینگ قادرند بر روی حافظه خود که ورودی را نیز در بر گرفته است به سمت چپ یا راست حرکت کنند و لذا از حافظه خود هم به عنوان ورودی و هم به عنوان خروجی استفاده می‌کنند. ماشین‌های کراندار خطی و تورینگ را می‌توان هم به عنوان پذیرنده و هم به عنوان تراگذر به کار گرفت. هر ماشین دارای مجموعه‌ای از حالات است که این مجموعه حالات به دو دسته نهایی و غیر نهایی تقسیم‌بندی می‌شوند. شرط پذیرش یک رشته در پذیرنده‌های متناهی و پشته‌ای این است که ماشین بعد از خواندن تمام ورودی در حالت نهایی متوقف شود ولی از آنجا که در پذیرنده‌های کراندار خطی و تورینگ می‌توان بر روی نوار حافظه که ورودی نیز بر روی آن قرار گرفته به سمت چپ یا راست حرکت کرد لذا شرط پذیرش در این ماشین‌ها در این است که ماشین در حالت نهایی متوقف شود و نیازی نیست تمام ورودی خوانده شود.

ماشین‌ها از لحاظ عملکرد به دو دسته قطعی(Deterministic) و غیرقطعی(Nondeterministic) تقسیم‌بندی می‌شوند. ماشین‌های قطعی در هر لحظه با یک ورودی مشخص یک عملکرد قطعی و مشخص دارند در حالیکه ماشین‌های غیر قطعی با یک ورودی مشخص عملکردهای متفاوتی ممکن است از خود نشان دهند. معمولاً ماشین‌های قطعی برای پیاده‌سازی مناسب و ماشین‌های غیر قطعی برای طراحی مناسب محسوب می‌شوند و لذا در زبان‌هایی که طراحی آنها با ماشین قطعی دشوار است ابتدا آن را با ماشین غیرقطعی طراحی کرده و سپس پیاده‌سازی آن را با ماشین قطعی انجام می‌دهند. معمولاً وضعیت ورودی، حافظه و حالت ماشین نشان‌دهنده پیکربندی(Configuration) یک ماشین در یک لحظه خاص است.در ماشین‌های قطعی در هر لحظه ماشین فقط در یک پیکربندی قرار دارد ولی در ماشین‌های غیر قطعی ممکن است ماشین در یک لحظه در بیش از یک پیکربندی قرار داشته باشد.



هر ماشین خاص یک زبان خاص را می‌پذیرد و هر دسته از ماشین‌ها فقط قادر به پذیرش دسته خاصی از زبان‌ها هستند. ماشین‌های زیر به ترتیب دارای قدرت بیشتری می‌شوند به این معنی که دسته زبان‌های بزرگتری را پذیرش کرده و زبان هر ماشین زبان ماشین رده قبل از خود را شامل می‌شود.

- ماشین‌های متناهی قادر به پذیرش زبان‌های منظم (Regular) هستند.
- ماشین‌های پشته‌ای قطعی یا معین قادر به پذیرش زبان‌های مستقل از متن قطعی(Deterministic Context Free) هستند.
- ماشین‌های پشته‌ای غیر قطعی قادر به پذیرش زبان‌های مستقل از متن (Context Free) هستند. این مجموعه از زبان‌ها تمامی زبان‌های قابل پذیرش توسط ماشین‌های پشته‌ای قطعی و غیر قطعی را شامل می‌شود.
- ماشین‌های کراندار خطی قادر به پذیرش زبان‌های حساس به متن (Context Sensitive) هستند.
- ماشین‌های تورینگ قادر به پذیرش زبان‌های بازگشتی شمارا (Recursively Enumerable) هستند.
- ماشین‌های بالا به ترتیب دارای قدرت بیشتری می‌شوند به این معنی که دسته زبان‌های بزرگتری را پذیرش کرده و زبان هر ماشین زبان ماشین رده قبل از خود را شامل می‌شود. دیاگرام زیر این موضوع را نشان می‌دهد:



شکل ۱-۱

نکته: قدرت ماشین‌های متناهی قطعی و غیر قطعی با هم یکسان و قدرت ماشین‌های تورینگ قطعی و غیر قطعی نیز با هم یکسان هستند ولی قدرت ماشین‌های پشته‌ای قطعی کمتر از قدرت ماشین‌های پشته‌ای غیر قطعی است.

نکته: ماشین‌های کراندار خطی غیر قطعی می‌باشند.

در فصول آینده، زبان‌های فوق را به ترتیب از دسته کوچکتر شروع به بررسی می‌کنیم. برای هر دسته از زبان‌ها ماشین‌هایی که قادر به پذیرش آنها هستند، گرامرهایی که قادر به تولید آنها هستند و خصوصیات آن زبان‌ها را به طور دقیق‌تری مطرح می‌کنیم. در بررسی زبان‌ها ممکن است دسته زبان‌هایی با خصوصیات خاصی مطرح شوند که در جای خود آنها را بررسی کرده و جایگاه آنها را در این دیاگرام نسبت به بقیه دسته زبان‌ها خواهیم گفت.

نکته: زبان‌هایی هم وجود دارند که در هیچ‌کدام از دسته زبان‌های دیاگرام بالا قرار ندارند و خارج از دسته‌های فوق محسوب می‌شوند.

در فصل ۲ ماشین‌های متناهی، و الگوریتم‌های مربوط به آنها مطرح می‌شود، در فصل ۳ عبارات منظم، گرامرهای منظم و رابطه آنها با زبان‌های منظم مطرح می‌شود، در فصل ۴ خصوصیات زبان‌های منظم مورد بررسی قرار می‌گیرد، در فصل ۵ زبان‌های مستقل از متن و گرامرهای مربوط به آنها مطرح می‌شوند، در فصل ۶ الگوریتم‌های مربوط به ساده‌سازی گرامرهای مستقل از متن بیان می‌شود، در فصل ۷ ماشین‌های پشته‌ای و ارتباط آنها با زبانهای مستقل از متن بررسی می‌شود. در فصل ۸ خصوصیات زبان‌های مستقل از متن بیان می‌شوند، در فصل ۹ ماشین‌های تورینگ و نحوه عملکرد آنها مورد بررسی قرار می‌گیرند، در فصل ۱۰ انواع مختلف از ماشین‌های تورینگ مطرح می‌شود، در فصل ۱۱ دسته‌بندی زبان‌ها و ماشین‌ها بیان شده و در فصل ۱۲ نیز محدوده محاسبات الگوریتمی و انواع مسائل تصمیم‌گیری مطرح می‌شود.

نمونه سؤالات

۱. کدام‌یک از عبارات زیر صحیح است؟

i: ممکن است گاهی اوقات رابطه $(L_1 \cup L_2)^* = L_1^* \cup L_2^*$ برقرار باشد.

ii: بستار ستاره نسبت به اجتماع خاصیت پخشی ندارد بنابراین نمی‌توان گفت که

$$(L_1 \cup L_2)^* = L_1^* \cup L_2^*$$

۱) i

۲) ii

۳) i و ii

۴) هیچکدام

۲. گرامر G چه زبانی را تولید می‌کند؟

$G:$
$S \to AB \mid C$
$A \to aAb \mid \lambda$
$B \to cBd \mid \lambda$
$C \to aCd \mid D$
$D \to bDc \mid \lambda$

۱) $L = \{a^n b^m c^m d^n : n, m \geq 0\}$

۲) $L = \{a^n b^n c^m d^m : n, m \geq 0\}$

۳) $L = \{a^n b^n c^m d^m : n, m \geq 0\} \cup \{a^n b^m c^m d^n : n, m \geq 0\}$

۴) هیچ‌کدام

۳. کدام گزینه نادرست است؟

$$L_1 = \{w \mid w = xy, x, y \in \{a,b\}^*, |x| = |y|\}$$

$$L_2 = \{w \mid w = xy, x, y \in \{a,b\}^*, x = y\}$$

۱) $L_1 {L_1}^R = L_1$

۲) $(L_2 \cap L_1)^R = L_2$

۳) $(L_2 {L_2}^R) = L_2$

۴) $L_1 \cap {L_1}^R = L_1$

۴. فرض کنید:

$$L_1 = \{w \mid w \in \{a,b\}^*, |w| \bmod 2 = 0 \text{ or } |w| \bmod 3 = 0\}$$

$$L_2 = \{w \mid w \in \{a,b\}^*, w = xy, x = y^R\}$$

$$L_3=\{w \mid w\in\{a,b\}^*, w=xy, x=y\}$$

$$L_4=\{w \mid w\in\{a,b\}^*, |w| \text{ is prime number}\}$$

$$L_5=\{w \mid w\in\{a,b\}^*, |w| \text{ is not prime number}\}$$

$$L_6=\{w \mid w\in\{a,b\}^*, |w| \bmod 2=1\}$$

$$L_7=\{w \mid w\in\{a,b\}^*, |w| \bmod 2=0\}$$

$$L_8=\{w \mid w\in\{a,b\}^*, |w| \bmod 2=0 \text{ and } |w| \bmod 3=0\}$$

$$L_9=\{w \mid w\in\{a,b\}^*, \quad w=xy, |x|=|y|\}$$

کدام گزینه درست است؟

۱) $L_1=L_1^*$ و $L_2\neq L_2^*$ و $L_3=L_3^*$ و $L_5=L_5^*$ $L_4\neq L_4^*$ و $L_6\neq L_6^*$ و $L_7=L_7^*$ و $L_8\neq L_8^*$ و $L_9=L_9^*$

۲) $L_1\neq L_1^*$ و $L_2=L_2^*$ و $L_3\neq L_3^*$ و $L_5\neq L_5^*$ $L_4=L_4^*$ و $L_6=L_6^*$ و $L_7\neq L_7^*$ و $L_8=L_8^*$ و $L_9\neq L_9^*$

۳) $L_1\neq L_1^*$ و $L_2\neq L_2^*$ و $L_3\neq L_3^*$ و $L_5\neq L_5^*$ $L_4\neq L_4^*$ و $L_6\neq L_6^*$ و $L_7=L_7^*$ و $L_8=L_8^*$ و $L_9=L_9^*$

۴) $L_1\neq L_1^*$ و $L_2\neq L_2^*$ و $L_3\neq L_3^*$ و $L_5\neq L_5^*$ $L_4\neq L_4^*$ و $L_6\neq L_6^*$ و $L_7\neq L_7^*$ و $L_8\neq L_8^*$ و $L_9\neq L_9^*$

۵. گرامرزیر چه زبانی را تولید می‌کند؟

$$S\to aB \mid bA \mid \lambda$$
$$B\to b \mid bS \mid aBB$$
$$A\to a \mid aS \mid bAA$$

۱) $\{w \mid w\in\{a,b\}^*, 2n_a(w)=n_b(w)\}$

۲) $\{w \mid w\in\{a,b\}^*, n_a(w)=2n_b(w)\}$

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

۴) هیچ‌کدام

۶. گرامر زیر چه زبانی را تولید می‌کند؟

$$S\to ABS \mid ab \mid ba \mid \lambda$$
$$AB\to BA$$
$$BA\to AB$$
$$A\to a$$
$$B\to b$$

۱) $\{w \mid w\in\{a,b\}^*, 2n_a(w)=n_b(w)\}$

۲) $\{w \mid w\in\{a,b\}^*, n_a(w)=2n_b(w)\}$

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

۴) هیچ‌کدام

۷. گرامر زیر چه زبانی را تولید می‌کند؟

$$S\to aAbc \mid \lambda$$
$$A\to aAbB \mid \lambda$$
$$Bc\to cc$$
$$Bb\to bB$$

۱) $\{w \mid w\in\{a,b\}^*, 2n_a(w)=n_b(w)+n_c(w)\}$

۲) $\{w \mid w\in\{a,b\}^*, n_a(w)=2n_b(w)+n_c(w)\}$

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

۴) هیچ‌کدام

۸. گرامر زیر چه زبانی را تولید می‌کند؟

$$S \to ABC \mid SABC \mid \lambda$$
$$AB \to BA$$
$$BA \to AB$$
$$CB \to BC$$
$$BC \to CB$$
$$CA \to AC$$
$$AC \to CA$$
$$A \to a$$
$$B \to b$$
$$C \to c$$

۱) $\left\{ w \mid w \in \{a,b\}^*, 2n_a(w) = n_b(w) + n_c(w) \right\}$

۲) $\left\{ w \mid w \in \{a,b\}^*, n_a(w) = 2n_b(w) + n_c(w) \right\}$

۳) $\left\{ w \mid w \in \{a,b\}^*, n_a(w) = n_b(w) + n_c(w) \right\}$

۴) هیچکدام

۹. کدام یک از جداول زیر در رابطه با گرامرهای زیر صحیح می باشد؟

$$G_1 : \begin{cases} A \to aAB \\ B \to abA \\ B \to b \end{cases} \qquad G_2 = \begin{cases} A \to aB \\ B \to Bba \\ B \to b \end{cases} \qquad G_3 : \begin{cases} A \to aAB \\ A \to bAa \\ A \to b \end{cases}$$

$$G_4 : \begin{cases} A \to aA \\ A \to bA \\ A \to cA \end{cases} \qquad G_5 \begin{cases} A \to DE \\ D \to FC \\ F \to a \\ C \to b \end{cases} \qquad G_6 : \begin{cases} A \to aBC \\ B \to bE \end{cases}$$

$$G_7 \begin{cases} A \to aBDb \\ BD \to DB \\ BDb \to bB \end{cases}$$

۱)

نوع/گرامر	منظم	خطی	ساده	گریباخ	چامسکی	حساس به متن	بدون محدودیت
G_1	×	×	×	✓	✓	×	×
G_2	✓	×	×	×	✓	✓	✓
G_3	×	×	×	×	×	×	✓
G_4	✓	✓	×	×	×	×	×
G_5	×	×	✓	✓	×	×	×
G_6	×	×	✓	✓	×	✓	✓
G_7	×	×	×	×	×	×	✓

۲)

نوع/گرامر	منظم	خطی	ساده	گریباخ	چامسکی	حساس به متن	بدون محدودیت
G_1	×	×	×	×	×	✓	✓
G_2	×	✓	×	×	×	✓	✓
G_3	×	×	×	×	×	✓	✓
G_4	✓	✓	✓	✓	×	✓	✓
G_5	×	×	×	×	✓	✓	✓
G_6	×	×	✓	✓	×	✓	✓
G_7	×	×	×	×	×	×	✓

۳)

بدون محدودیت	حساس به متن	چامسکی	گریباخ	ساده	خطی	منظم	نوع/گرامر
✓	✓	✓	✓	✓	×	×	G_1
✓	✓	×	✓	×	✓	×	G_2
✓	✓	×	×	✓	✓	✓	G_3
✓	✓	✓	×	×	×	×	G_4
✓	✓	✓	×	✓	×	×	G_5
✓	✓	×	×	×	×	×	G_6
✓	×	×	×	×	×	×	G_7

۴) هیچ‌کدام

۱۰. زبان گرامر زیر چیست؟

$$S \rightarrow aSC \mid aS'C$$
$$S' \rightarrow aS'D \mid bS''D$$
$$S''C \rightarrow c$$
$$DC \rightarrow CD$$
$$cC \rightarrow cc$$
$$D \rightarrow d$$

۱) $L = \left\{ w \mid w = a^n b^m c^n d^m, m, n > 0 \right\}$

۲) $L = \left\{ w \mid w = a^n b^m c^m d^n, m, n > 0 \right\}$

۳) $L = \left\{ w \mid w = a^n b^n c^n d^n, n > 0 \right\}$

۴) هیچ‌کدام

۱۱. در میان گرامرهای زیر گرامرهای منظم کدامند؟

$G1: \quad S \rightarrow aSb \mid ab$

$G2: \quad S \rightarrow aA \mid ab \quad A \rightarrow Sb$

$G3: \quad S \rightarrow aA \mid ab \quad A \rightarrow bS$

$G4: \quad S \rightarrow Ab \mid ab \quad A \rightarrow Sa$

۱) G1

۲) G3 و G4

۳) همه بجز G1

۴) هر چهار گرامر

۱۲. گرامر را در نظر بگیرید:

$$G = (\{a\}, \{S, A, B, C, D, E, F\}, S, P)$$

P شامل قوانین زیر می باشد:

$S \rightarrow ABC$	$DE \rightarrow EFaa$	$aC \rightarrow Ca$
$B \rightarrow DBE \mid \lambda$	$aE \rightarrow Ea$	$FE \rightarrow EF$
$FC \rightarrow C$	$AE \rightarrow A$	$AC \rightarrow \lambda$

زبان گرامر G کدام است؟

۱) $\left\{ a^{\frac{n(n+1)}{2}} \mid n \geq 0, \text{ زوج } n \right\}$

۲) $\left\{ a^{n^2+1} \mid n \geq 0 \right\}$

۳) $\left\{ a^{n(n+1)} \mid n \geq 0 \right\}$

۴) هیچ‌کدام

۱۳. گرامر G را در نظر بگیرید:

$$G = (\{a, b, c\}, \{S, A, B, C\}, S, P)$$

شامل قوانین زیرمی باشد:

$S \to \lambda \mid S$
$S \to SABC \mid ABC$
$AB \to BA$
$C \to c$
$BA \to AB$
$CB \to BC$
$BC \to CB$
$B \to b$
$CA \to AC$
$AC \to CA$
$A \to a$

اگر $L(G)$ زبانی باشد که بوسیله گرامر فوق تعریف شده باشد، کدام یک از گزینه های زیر صحیح است؟

۱) $L(G)=\left\{x \in \{a,b,c\}^* \mid |x|_a=|x|_b \text{ or } |x|_c=|x|_c \text{ or } |x|_a=|x|_c\right\}$

۲) $L(G)=\left\{x \in \{a,b,c\}^* \mid |x|_a \geq |x|_b \geq |x|_c\right\}$

۳) $L(G)=\left\{x \in \{a^*,b^*,c^*\} \mid |x|_a=|x|_b=|x|_c\right\}$

۴) $L(G)=\left\{x \in \{a,b,c\}^* \mid |x|_a=|x|_b=|x|_c\right\}$

۱۴. فرض کنید A ، B و C زبان‌هایی هستند که بر روی یک الفبا تعریف شده اند کدام‌یک از گزاره‌های زیر درست است؟

۱) $A(B \cap C)=(AB)\cap(AC)$

۲) به ازاء هر زبان A زبان نامنظم B وجود دارد به گونه‌ای که $B \subseteq A$

۳) به ازاء هر زبان A زبان نامنظم B وجود دارد به گونه‌ای که $B \supseteq A$

۴) $A(B \cup C)=(AB)\cup(AC)$

۱۵. زبان $L=\left\{a^n b^n \mid n>0\right\}$ را می‌توان به وسیله گرامر $G=\langle V_N, V_T, S, \varphi\rangle$ تولید نمود که در آن $V_N=\{S\}$ ، $V_T=\{a,b\}$ ، S علامت شروع و φ مجموعه قواعد تولید می‌باشد، کدام یک از مجموعه قواعد زیر زبان فوق را تولید می‌نماید؟

۱) $\varphi=\{S \to bSa, S \to b, S \to a\}$

۲) $\varphi=\{S \to aSb, S \to b, S \to a\}$

۳) $\varphi=\{S \to Sa, S \to bS, S \to ab, S \to ba\}$

۴) $\varphi=\{S \to aSb, S \to bSa, S \to ab, S \to ba, S \to SS\}$

۱۶. اگر $L \subseteq \Sigma^*$ داده شده باشد، شرط $L \cap L\Sigma^+ = \varnothing$ با کدام گزاره معادل است؟

(علوم کامپیوتر ۸۶)

۱) $x, yxy \in L \Rightarrow x=\lambda$

۲) $x, xy \in L \Rightarrow y=\lambda$

۳) $y, yx^2 \in L \Rightarrow x=\lambda$

۴) $y, xyx^2 \in L \Rightarrow y=\lambda$

۱۷. گرامر وابسته به متن G مفروض است:

$G:$
$S \to S_1B$
$S_1 \to aS_1b$
$bB \to bbbB$
$aS_1b \to aa$
$B \to \lambda$

زبان گرامر G کدام است؟

(دولتی مهندسی کامپیوتر ۸۹)

۱) $\left\{a^{n+1}b^{n+k} \mid n \geq 1, k \geq 0\right\}$

۲) $\left\{a^n b^k \mid n \geq 2, k \geq 0\right\}$

۳) $\left\{a^n b^{n+2k} \mid n \geq 2, k \geq 0\right\}$

۴) $\left\{a^{n+1}b^{n+2k-1} \mid n \geq 1, k \geq 0\right\}$

۱۸. گرامر G و زبان‌های L_1 و L_2 مفروض‌اند. ارتباط $L(G)$ با L_1 و L_2 کدام است؟

(دولتی مهندسی کامپیوتر ۸۷)

$S \to Sab$
$S \to Sba$
$S \to aSb$
$S \to bSa$
$S \to abS$
$S \to baS$
$S \to \lambda$

$L_1 = \{ w \in (a+b)^* \mid$ تعداد aهای w با bهای w برابر است $\}$

$L_1 = \{ w \in (a+b)^* \mid$ تعداد abهای w با baهای w برابر است $\}$

۱) $L(G) \subset L_1$
۲) $L(G) = L_1$
۳) $L(G) = L_1 \cup L_2$
۴) $L(G) \supset L_2$

۱۹. گرامر وابسته به متن G به شرح زیر مفروض است. کدام‌یک از مجموعه رشته‌های 1 تا 4 زیر مجموعه $L(G)$ است؟

(دولتی مهندسی کامپیوتر ۸۷)

$S \to ACaB$
$Ca \to aaC$
$CB \to DB$
$CB \to E$
$aD \to Da$
$AD \to AC$
$aE \to Ea$
$aE \to a$

۱) $\{aa, aaaa\}$
۲) $\{aaa, aaaaa\}$
۳) $\{a, aaa, aaaaa\}$
۴) $\{aaaa, aaaaaa\}$

۲۰. گرامر G به شرح زیر مفروض است. $L(G)$ کدام است.

(دولتی مهندسی کامپیوتر ۸۷)

$G:$
$S \to aA$
$S \to bB$
$S \to \varepsilon$
$A \to Sa$
$A \to \varepsilon$
$B \to Sb$
$B \to \varepsilon$

۱) $(a+b)^*$
۲) $\{ w \in (a+b)^* \mid w = w^R \}$
۳) $\{ w(a+b)w^R \mid w \in (a+b)^* \}$
۴) $\{ ww^R \mid w \in (a+b)^* \}$

۲۱. اگر $\sum = \{a,b,c\}$ و $L - \sum^* = \varnothing$ باشد آن‌گاه L کدام‌یک از زبان‌های زیر می‌تواند باشد؟

(دولتی مهندسی کامپیوتر ۸۶)

$I - \Sigma^*, II - a^n b^{n^2} c^n, III - \varnothing, IV - \varepsilon$

۱) فقط I
۲) فقط II
۳) فقط I و III
۴) I، II، III و IV

حل تشریحی

۱. گزینه ۳ درست است.

بر طبق قوانین مربوط به جدول ۱-۲ رابطه مورد نظر همیشـــه برقرار نیســـت ولی در برخی اوقات نظیر زمانی که $L_1 = L_2 = \varnothing$ آنگاه رابطه مورد نظر می‌تواند برقرار باشد.

۲. گزینه ۳ درست است.

متغیر S در گرامر فوق یا می‌تواند AB را تولید کند و یا می‌تواند C را تولید کند. در حالت اول A رشـــته‌های a^nb^n را تولید کرده و B رشـــته‌های c^md^m را تولید می‌کند و در حالت دوم C شـــکل جمله‌ای a^nDd^n را تولید کرده و D نیز رشـــته‌های b^mc^m را تولید می‌کند و لذا گزینه ۳ صحیح است.

۳. گزینه ۳ درست است.

با دقت در زبان‌های L_1 و L_2 می‌توان به این نتیجه رسید که همه گزینه‌ها به جز گزینه ۳ درست هستند.

۴. گزینه ۳ درست است.

با دقت در زبان‌های فوق می‌توان به درستی گزینه ۳ پی برد.

۵. گزینه ۳ درست است.

ملاحظه می‌شود که در گرامر فوق تولید هر a حتماً یک b نیز تولید خواهد کرد و تولید هر b حتماً یک a نیز تولید خواهد کرد و همچنین هر ترکیبی از a و bها نیز قابل تولید است.

۶. گزینه ۳ درست است.

ملاحظه می‌شـــود که در گرامر فوق به دلیل وجود دو قاعده $BA \to AB$ و $AB \to BA$ هر جابه‌جایی دلخواه بر روی حروف a و b قابل ایجاد اســـت و همچنین چون تولید هر a حتماً با تولید یک b همراه اســـت و تولید هر b حتماً با تولید یک a همراه است لذا گزینه ۳ صحیح است.

۷. گزینه ۴ درست است.

از آنجا که با هر A می‌توان یک $aAbB$ تولید کرد لذا زبان مورد نظر را می‌توان به صورت زیر در نظر گرفت:

$$S \Rightarrow aAbc \overset{*}{\Rightarrow} aa^mA(bB)^m bc \overset{*}{\Rightarrow} aa^m(bB)^m bc \overset{*}{\Rightarrow} aa^mb^mbB^mc \overset{*}{\Rightarrow} aa^mb^mbc^mc$$

لذا زبان گرامر مورد نظر

$$\left\{a^n b^n c^n \mid n \geq 0\right\}$$

است و گزینه ۴ صحیح است.

۸. گزینه ۴ درست است.

زبان گرامر مورد نظر

$$\left\{w \mid w \in \{a,b\}^*, n_a(w) = n_b(w) = n_c(w)\right\}$$

است و لذا گزینه ۴ صحیح است.

۹. گزینه ۲ درست است.

بر اساس تعریف گرامرهای مورد نظر گزینه ۲ صحیح است.

۱۰. گزینه ۴ درست است.

با بررسی اشتقاق‌های ممکن به صورت زیر:

$$S \Rightarrow aSC \stackrel{*}{\Rightarrow} a^m S C^m \stackrel{*}{\Rightarrow} a^m S' C^m \stackrel{*}{\Rightarrow} a^m a^k S'' D^k C^m \stackrel{*}{\Rightarrow} a^m a^k S' D^k C^m \stackrel{*}{\Rightarrow}$$

$$a^m a^k b S'' D^{k+1} C^m \stackrel{*}{\Rightarrow} a^{m+k} b S'' C^m D^{k+1}$$

$$\stackrel{*}{\Rightarrow} a^{m+k} bcC^{m-1} D^{k+1} \stackrel{*}{\Rightarrow} a^{m+k} bc^m D^{k+1} \stackrel{*}{\Rightarrow} a^{m+k} bc^m d^{k+1}$$

می‌توان به این نتیجه رسید که گزینه ۴ صحیح است.

۱۱. گزینه ۲ درست است.

طبق تعریف گرامر منظم گزینه ب صحیح است.

۱۲. گزینه ۴ درست است.

چون گرامر رشته λ را تولید می‌کند پس گزینه ۲ نادرست است. چون گرامر رشته aa را تولید می‌کند پس گزینه ۱ نادرست است. و چون گرامر رشته aaaa را تولید می‌کند پس گزینه ۳ نادرست است. متوجه می‌شویم که گزینه ۴ صحیح است. با کمی دقت می‌توان دریافت که توسط B می‌توان به تعداد دلخواه DE تولید کرده و هر DE منجر به تولید aa می‌شود. در نتیجه رشته‌هایی به طول زوج از a قابل ساخت است.

۱۳. گزینه ۴ درست است.

با دقت در اشتقاق‌های ممکن در گرامر فوق می‌توان دریافت که گزینه ۴ صحیح است.

۱۴. گزینه ۴ درست است.

گزینه ۱ طبق بند ۱۱ جدول ۱-۲ نادرست است و گزینه ۴ نیز طبق جدول ۱-۲ صحیح است. برای نشان دادن نادرستی گزینه ۲ مثالی را در نظر بگیرید که در آن $A = \varnothing$ و برای نشان دادن نادرستی گزینه ۳ مثالی را در نظر بگیرید که در آن $A = \Sigma^*$.

۱۵. گزینه ۴ درست است.

از آنجا که زبان قابل تولید توسط گرامر د شامل زبان مورد نظر می‌باشد و هیچکدام از بقیه گرامرها این خاصیت را ندارند لذا گزینه ۴ صحیح است.

۱۶. گزینه ۲ درست است.

اگر $L \cap L\Sigma^+ = \varnothing$ شرط برقرار باشد به این معنی است که هر رشته غیر پوچ به هر کدام از جملات زبان L الحاق شود نتیجه نمی‌تواند داخل زبان L باشد. این نشان دهنده آن است که تنها رشته‌ای که به انتهای یک رشته از زبان الحاق شود و نتیجه بتواند داخل زبان باشد رشته λ است و لذا گزینه ۲ صحیح است.

۱۷. گزینه ۴ درست است.

$$S \Rightarrow S_1B \overset{*}{\Rightarrow} a^nS_1b^nB \Rightarrow a^{n-1}aab^{n-1}B \overset{*}{\Rightarrow} a^{n+1}b^{n-1}b^{2k}B \Rightarrow a^{n+1}b^{n+2k-1}$$

که در آن: $n \geq 1, k \geq 0$ لذا گزینه ۴ صحیح است.

۱۸. گزینه ۱ درست است.

$L(G)$ دارای رشته‌هایی است که تعداد a های آن با تعداد b های آن برابر است ولی شامل همه این رشته‌ها نیست برای نمونه $L(G)$ شامل رشته aabbbbaa نیست لذا گزینه ۱ صحیح است.

۱۹. گزینه ۲ درست است.

با ایجاد کوچک‌ترین اشتقاق برای گرامر مورد نظر به رشته aaa خواهیم رسید لذا گزینه ۲ صحیح است.

۲۰. گزینه ۲ درست است.

در گرامر دیده می‌شود که اگر S در شروع رشته یک a تولید کند حتماً در پایان رشته نیز a تولید می‌کند و همچنین اگر S در شروع رشته یک b تولید کند حتماً در پایان رشته نیز b تولید می‌کند و یا S با تولید یک a یا b به کار خود پایان می‌دهد.

۲۱. گزینه ۴ درست است.

از آنجاکه $L - \Sigma^*$ همیشه برابر با تهی است لذا گزینه ۴ صحیح است.

خودآزمایی

۱. گرامری برای اعداد حقیقی در زبان C بیان کنید.

۲. تراگذری طراحی کنید که اعداد باینری را به عدد مبنای شانزده معادل خود تبدیل کند.

۳. تراگذری طراحی کنید که دو عدد مبنای ۱۰ را دریافت کرده و مجموع آنها را محاسبه کند.

۴. گرامری طراحی کنید که کلیه شناسه‌های معتبر در زبان پاسکال را تولید کند. گرامر مورد نظر در کدام دسته از گرامرها قرار دارد؟