بسم الله الرحمن الرحیم

نظریه زبان‌ها و ماشین‌ها

سری کتاب‌های کمک آموزشی کارشناسی ارشد

مجموعه مهندسی کامپیوتر

مؤلفان: اسماعیل چیتگر – فائزه عباس‌پورقادری

ویراستار علمی: آزاده ضیایی

سرشناسه	: چیتگر، اسماعیل
عنوان	: نظریه زبان‌ها و ماشین‌ها
مشخصات نشر	: تهران : مشاوران صعود ماهان ، ۱۴۰۱
مشخصات ظاهری	: ۱۶۵ ص
فروست	: سری کتاب‌های کمک آموزشی کارشناسی ارشد
شابک	: ۹-۸۰۸-۴۵۸-۶۰۰-۹۷۸
وضعیت فهرست نویسی	: فیپای مختصر
یادداشت	:این مدرک در آدرس http://opac.nlai.ir قابل دسترسی است.
شناسه افزوده	: فائزه عباس‌پورقادری
شناسه افزوده	: آزاده ضیایی، ویراستار علمی
شماره کتابشناسی ملی	: ۳۷۲۶۷۷۶

نام کتاب: .... نظریه زبان‌ها و ماشین‌ها

مولفان: ....اسماعیل چیتگر، فائزه عباس‌پور قادری

مدیر تولید محتوی: ....سمیه بیگی

ویراستار علمی: .... آزاده ضیایی

ناشر: ....مشاوران صعود ماهان

نوبت و تاریخ چاپ: .... اول/ ۱۴۰۱

تیراژ: .... ۱۰۰۰ جلد

قیمت: .... ۲/۰۹۰/۰۰۰ ریال

شابک: .... ISBN: ۹-۸۰۸-۴۵۸-۶۰۰-۹۷۸

انتشـــارات مشاوران صعود ماهان: خیابان ولیعصر، بالاتر از تقـاطع مطهری،

روبروی قنادی هتل بزرگ تهـــران، جنب بانک ملی، پلاک ۲۰۵۰

تلفن: ۴- ۸۸۱۰۰۱۱۳

سخن ناشر

«ن والقلم و ما یسطرون»

کلمه نزد خدا بود و خدا آن را با قلم بر ما نازل کرد.

به پاس تشکر از چنین موهبت الهی، موسسه ماهان درصدد برآمده است تا در راستای انتقال دانش و مفاهیم با کمک اساتید مجرب و مجموعه کتب آموزشی خود برای شما داوطلبان ادامه تحصیل در مقطع کارشناسی ارشد گام موثری بردارد.

امید است تلاش‌های خدمتگزاران شما در این موسسه پایه‌گذار گام‌های بلند فردای شما باشد.

مجموعه کتاب‌های کمک آموزشی ماهان به‌منظور استفاده داوطلبان کنکور کارشناسی ارشد سراسری و آزاد تالیف شده‌اند. در این کتاب‌ها سعی کرده‌ایم با بهره‌گیری از تجربه اساتید بزرگ و کتب معتبر داوطلبان را از مطالعه کتاب‌های متعدد در هر درس بی‌نیاز کنیم.

دیگر تالیفات ماهان برای سایر دانشجویان به‌صورت ذیل می‌باشد.

- **مجموعه کتاب‌های ۸ آزمون:** شامل ۵ مرحله کنکور کارشناسی ارشد ۵ سال اخیر به همراه ۳ مرحله آزمون تالیفی ماهان همراه با پاسخ تشریحی می‌باشد که برای آشنایی با نمونه سوالات کنکور طراحی شده است. این مجموعه کتاب‌ها با توجه به تحلیل ۳ ساله اخیر کنکور و بودجه‌بندی مباحث در هر یک از دروس، اطلاعات مناسبی جهت برنامه‌ریزی درسی در اختیار دانشجو قرار می‌دهد.
- **مجموعه کتاب‌های کوچک:** شامل کلیه نکات کاربردی در گرایش‌های مختلف کنکور کارشناسی ارشد می‌باشد که برای دانشجویان جهت جمع‌بندی مباحث در ۲ ماهه آخر قبل از کنکور مفید می‌باشد.

بدین‌وسیله از مجموعه اساتید، مولفان و همکاران محترم خانواده بزرگ ماهان که در تولید و به‌روزرسانی تالیفات ماهان نقش موثری داشته‌اند، صمیمانه تقدیر و تشکر می‌نماییم.

دانشجویان عزیز و اساتید محترم می‌توانند هرگونه انتقاد و پیشنهاد درخصوص تالیفات ماهان را از طریق سایت ماهان به آدرس mahan.ac.ir با ما در میان بگذارند.

موسسه آموزش عالی آزاد ماهان

سخن مؤلف

نظریه محاسبات شامل سه بخش نظریه زبان‌ها و ماشین‌ها (تئوری آتاماتا) نظریه محاسبه‌پذیری و نظریه پیچیدگی است. بحث مطرح‌شده در این کتاب به بحث تئوری آتاماتا و مقدمه‌ای بر نظریه‌های محاسبه‌پذیری و پیچیدگی می‌پردازد که یک مرجع برای درس نظریه‌ی زبان‌ها و ماشین‌های دانشجویان مهندسی کامپیوتر و همچنین علوم کامپیوتر است. در این کتاب تلاش شده است تا تمام مباحث پر اهمیتی که در درس مهندسی نرم‌افزار بر اساس سرفصل وزارت علوم در دوره کارشناسی مطرح بوده جمع‌آوری شوند. لازم به تذکر است که منبع اصلی این جمع‌آوری کتاب نظریه زبان‌ها و آتاماتا نوشته پیتر لینز است. در این کتاب سعی شده است تمامی مطالب مهم سرفصل‌های کنکور سراسری و آزاد گنجانده شود و کاملا مباحث توضیح داده شده در هر فصل شامل نکته و مثال، تست‌های تالیفی و تست‌های سراسری و آزاد چند ساله کنکور در هر فصل می‌باشد. امید است که این کتاب برای دانشجویان و داوطلبان گرامی مفید واقع شود.

آزاده ضیایی

فهرست

فصل اول

مقدمه

- مجموعه‌ها
- توابع و رابطه‌ها
- گراف‌ها و درخت‌ها
- مفاهیم ابتدایی نظریه زبان‌ها و ماشین‌ها
- زبان‌ها
- گرامرها
- ماشین‌ها

مقدمه

قبل از ورود به بحث نظریه زبان‌ها و ماشین‌ها یک سری مفاهیم اولیه را مطرح می‌کنیم. هرچند این مطالب ساده و ابتدایی به‌نظر می‌رسند ولی دانستن دقیق آن‌ها به فهم مطالب ارائه شده در فصل‌های بعد کمک زیادی می‌کند.

مجموعه‌ها

مجموعه[1]: گروهی از اشیاء یا عناصر که بدون تکرار بوده و به‌طور کامل مشخص هستند را مجموعه گویند. همچنین ترتیب اعضای یک مجموعه اهمیتی ندارد.

- اعضای مجموعه در داخل آکولاد قرار می‌گیرند.
- برای نشان دادن تعداد اعضای مجموعه‌ای، نام آن را در داخل قدر مطلق قرار می‌دهند.

مثال: مجموعه $S=\{a,b\}$، دارای اندازه ۲ می‌باشد؛ یعنی $|S|=۲$. زیرا شامل دو عضو a و b می‌باشد.

مجموعه جهانی یا مرجع[2]: مجموعه‌ای که شامل همه اعضای ممکن بوده را مجموعه مرجع گویند و معمولاً با V یا M نشان می‌دهند.

عملگرهای روی مجموعه‌ها: اگر $S_۱$ و $S_۲$ دو مجموعه باشند، اجتماع، اشتراک، تفاضل، متمم (مکمل) و ضرب دکارتی آن‌ها به‌صورت زیر است:

اجتماع $-S_۱ \cup S_۲=\{x \mid x\in S_۱ \ \text{یا}\ x\in S_۲\}$

اشتراک $-S_۱ \cap S_۲=\{x \mid x\in S_۱ \ ,\ x\in S_۲\}$

تفاضل $-S_۱ - S_۲=\{x \mid x\in S_۱ \ ,\ x\notin S_۲\}$

متمم مجموعه S را با $\overline{S}$ نشان می‌دهند:

$$\overline{S}=\{x \mid x\in u \ ,\ x\notin S\}$$

ضرب دکارتی: ضرب دکارتی دو مجموعه $S_۱$ و $S_۲$، مجموعه همه زوج‌های مرتبی است که مولفه اول آن‌ها از مجموعه اول و مولفه دوم آن‌ها از مجموعه دوم است و به‌صورت روبه‌رو تعریف می‌شود:

$$S_۱\times S_۲=\{(x,y) \mid x\in S_۱, y\in S_۲\}$$

که طبق تعریف فوق، اگر $|S_۱|=m$ و $|S_۲|=n$، در اینصورت $|S_۱\times S_۲|=m.n$.

مثال: اگر $S_۱=\{۱,۲\}$ و $S_۲=\{a,b\}$ باشد، داریم:

$$S_۱\times S_۲=\{(۱,a),(۱,b),(۲,a),(۲,b)\}$$

1- set
2- universal set

مجموعه توانی (power set)

مجموعه همه زیرمجموعه‌های یک مجموعه است و آن را با نماد ۲^S یا $p(S)$ نمایش می‌دهند.

مثال: اگر $S=\{a,b,c\}$ آنگاه مجموعه توانی S به‌صورت زیر خواهد بود:

$$۲^S=\{\varnothing,\{a\},\{b\},\{c\}\{a,b\},\{a,c\}\{b,c\},\{a,b,c\}\}$$

افراز یک مجموعه: فرض کنید که مجموعه‌های $S_۱,S_۲,...,S_n$ زیرمجموعه‌هایی از مجموعه S باشند. اگر این زیرمجموعه‌ها در سه شرط زیر صدق کنند، آنگاه گوییم مجموعه S را به مجموعه‌های $S_۱,S_۲,...,S_n$ افراز کرده‌ایم.

۱- زیرمجموعه‌های $S_۱,S_۲,...,S_n$ دوبه‌دو مجزا باشند.

۲- $S_۱\cup S_۲\cup...\cup S_n=S$

۳- هیچ‌یک از S_iها تهی نباشند.

قانون دمورگان (Demorgan’s lows): طبق قانون دمورگان، روابط زیر را برای دو مجموعه $S_۱$ و $S_۲$ خواهیم داشت:

$$\overline{S_۱\cup S_۲}=\bar{S}_۱\cap\bar{S}_۲$$

$$\overline{S_۱\cap S_۲}=\bar{S}_۱\cup\bar{S}_۲$$

رابطه بین مجموعه‌ها: اگر هر عضو $S_۱$، عضوی از مجموعه $S_۲$ باشد، آنگاه گوییم که $S_۱$ زیرمجموعه $S_۲$ است و آن را به‌صورت روبه‌رو نشان می‌دهیم:

$$\mathbf{S_۱\subseteq S_۲}$$

اگر $S_۱\subseteq S_۲$ باشد ولی $S_۲$ شامل عضوی باشد که در $S_۱$ وجود نداشته باشد، آنگاه گوییم که $S_۱$ زیرمجموعه محض $S_۲$ است و آن را به‌صورت زیر نمایش می‌دهیم:

$$\mathbf{S_۱\subset S_۲}$$

مجموعه‌های $S_۱$ و $S_۲$ را جدا از هم گوییم هرگاه هیچ عضو مشترکی نداشته باشند؛ یعنی $S_۱\cap S_۲=\varnothing$.

مجموعه‌های متناهی و نامتناهی:

به مجموعه‌ای که شامل تعداد محدودی عضو باشد، متناهی و در غیراین صورت نامتناهی گویند.

مثال: مجموعه S که شامل اعداد طبیعی و فرد کوچک‌تر از ۱۰ می‌باشد متناهی بوده و مجموعه N که شامل اعداد طبیعی است، نامتناهی می‌باشد:

$$S=\{۱,۳,۵,۷,۹\}\quad,\quad N=\{۱,۲,۳,...\}$$

مجموعه‌های شمارش‌پذیر و شمارش‌ناپذیر:

به مجموعه‌ای که بتوان اعضای آن را شمارش کرد، مجموعه شمارش‌پذیر گویند.

حال اگر این مجموعه متناهی باشد، به آن مجموعه متناهی شمارش‌پذیر و اگر نامتناهی باشد، نامتناهی شمارش‌پذیر گویند، اما اگر نتوان اعضای این مجموعه را شمارش کرد، به این مجموعه (نامتناهی) شمارش‌ناپذیر گویند.

مثال: مجموعه اعداد طبیعی کوچک‌تر از ۱۰، یک مجموعه متناهی شمارش‌پذیر، و مجموعه اعداد طبیعی یک مجموعه نامتناهی شمارش‌پذیر می‌باشد.

همچنین مجموعه اعداد حقیقی، (نامتناهی) شمارش‌ناپذیر می‌باشد.

توجه: توضیح تکمیلی در بحث مجموعه‌های شمارش‌پذیر و شمارش‌ناپذیر، در فصل آخر کتاب آمده است.

توابع و رابطه‌ها

یک تابع قانونی است که به هر عضو از یک مجموعه، عضو منحصربه‌فردی از مجموعه‌ای دیگر را نگاشت می‌کند. اگر f بیانگر یک تابع باشد، آنگاه به مجموعه اول، دامنه و به مجموعه دوم برد گویند و آن را به‌صورت زیر می‌نویسیم.

$f: S_1 \rightarrow S_2$

و این بدان معناست که S_1 دامنه f و S_2 برد f است. اگر دامنه f همه S_1 باشد آنگاه گوییم که f یک تابع تام (total) است در غیراین‌صورت f یک تابع پاره‌ای (partial) گفته می‌شود. اغلب در نظریه زبان‌ها و ماشین‌ها دامنه و برد توابع روی اعداد صحیح مثبت تعریف می‌شود علاوه بر آن ما به بررسی رفتار این توابع وقتی که آرگمان آنها خیلی بزرگ هستند، علاقه‌مندیم. در اینگونه مواقع دانستن نرخ رشد کافیست و برای این منظور می‌توانیم از نمادهای مجانبی استفاده کنیم. فرض کنید f(n) و g(n) توابعی هستند که دامنه آنها اعداد صحیح مثبت است. اگر عدد ثابت مثبتی چون C یافت شود به‌طوری‌که برای همه nهای به قدر کافی بزرگ داشته باشیم:

$$f(n) \le Cg(n)$$

گوییم که f حداکثر رشدی برابر g دارد و می‌نویسیم:

$$f(n)=O(g(n))$$

اگر $|f(n)| \ge C|g(n)|$، آنگاه f حداقل رشدی برابر g دارد و آن را با نماد $f(n)=\Omega(g(n))$ نشان می‌دهیم. در آخر اگر ثابت‌هایی چون C_1 و C_2 وجود داشته باشند، به‌طوری‌که:

$$C_1|g(n)| \le |f(n)| \le C_2|g(n)|$$

آنگاه f و g رشدی برابر دارند و می‌نویسیم:

$$f(n)=\theta(g(n))$$

مثال: فرض کنید:

$$f(n)=4n^2+5n+6$$
$$g(n)=9n^3$$
$$h(n)=3n^2$$

آنگاه داریم:

$$f(n)=O(g(n))$$
$$g(n)=\Omega(h(n))$$
$$f(n)=\theta(h(n))$$

می‌توان یک تابع را به‌صورت مجموعه‌ای از زوج‌های مرتب نشان داد. مثلا تابع f را به‌صورت زیر نمایش دهیم:

$$f=\{(x_1,y_1),(x_2,y_2),\ldots\}$$

که در آن مؤلفه اول هر زوج مرتب، عنصری از دامنه تابع و مؤلفه دوم عنصری از بُرد تابع متناظر با عنصر اول می‌باشد. طبق تعریف یک تابع، عنصر اول همه این زوج‌های مرتب، از هم متمایزند. اگر این شرط برقرار نباشد آنگاه f تابع نمی‌باشد، بلکه یک رابطه است. بنابراین یک رابطه، حالت جامع‌تری از یک تابع می‌باشد.

از دیدگاهی دیگر، به هر زیرمجموعه از ضرب دکارتی $A \times B$، یک رابطه از مجموعه A به مجموعه B گویند.

مثال: اگر $A=\{1,2\}$ و $B=\{3,4\}$ آنگاه روابط زیر را از مجموعه A به مجموعه B داریم:

$$R_1=\{(1,3),(1,4)\}$$
$$R_2=\{(2,3)\}$$
$$R_3=\varnothing$$

نکته: یک رابطه از یک مجموعه به خودش را یک رابطه روی آن مجموعه گویند.

مثال: اگر $A=\{۱,۲,۳\}$ آنگاه رابطه‌هایی که روی مجموعه A تعریف می‌شود عبارتند از:

$$R_۱=\{(۱,۱),(۲,۲),(۳,۳)\} \quad , \quad R_۲=\{(۱,۲),(۲,۱),(۱,۳),(۳,۱)\}$$

خواص روابطی که روی یک مجموعه تعریف می‌شوند عبارتند از:

۱- **خاصیت بازتابی (reflexivity):** (رابطه R بر مجموعه A خاصیت بازتابی دارد) اگر و تنها اگر (به‌ازای تمام xهای متعلق به A، x با خودش در رابطه باشد).

R is reflexive iff $\forall x \in A$: xRx

۲- **خاصیت تقارنی (symmetry):** (رابطه R متقارن است) اگر و تنها اگر (xRy آنگاه yRx).

R is symmetric iff if xRy then yRx

۳- **خاصیت تعدی (transitivity):** (رابطه R متعدی است) اگر و تنها اگر (xRy و yRz، در این‌صورت xRz)

R is transitive iff if xRy and yRz then xRz

رابطه‌ای که هر سه خاصیت فوق را داشته باشد رابطه هم‌ارزی نامیده می‌شود.

مثال: در مثال قبل رابطه $R_۱$، بازتابی، متقارن و متعدی و رابطه $R_۲$ فقط متقارن بوده و رابطه تهی، متقارن و متعدی می‌باشد.

گراف‌ها و درخت‌ها

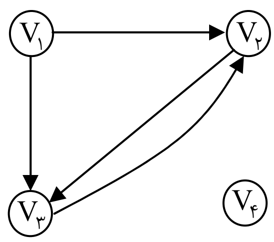
گراف ساختاری است که از دو مجموعه متناهی $V=\{v_۱,v_۲,...,v_n\}$ از رئوس و $E=\{e_۱,e_۲,...,e_n\}$ از یال‌ها تشکیل شده است. هر یال، زوجی از رئوس می‌باشند. مثلاً:

$e_i=(v_j,v_k)$

یالی از v_j به v_k است. گوییم یال e_i برای رأس v_j یک یال خروجی و برای رأس v_k یک یال ورودی است. به این ساختار یک گراف جهت‌دار گویند، زیرا هر یال جهتی را نشان می‌دهد. یک گراف ممکن است برچسب‌دار باشد. این برچسب‌ها، هم می‌تواند روی رئوس و هم می‌تواند روی یال‌ها قرار بگیرند.

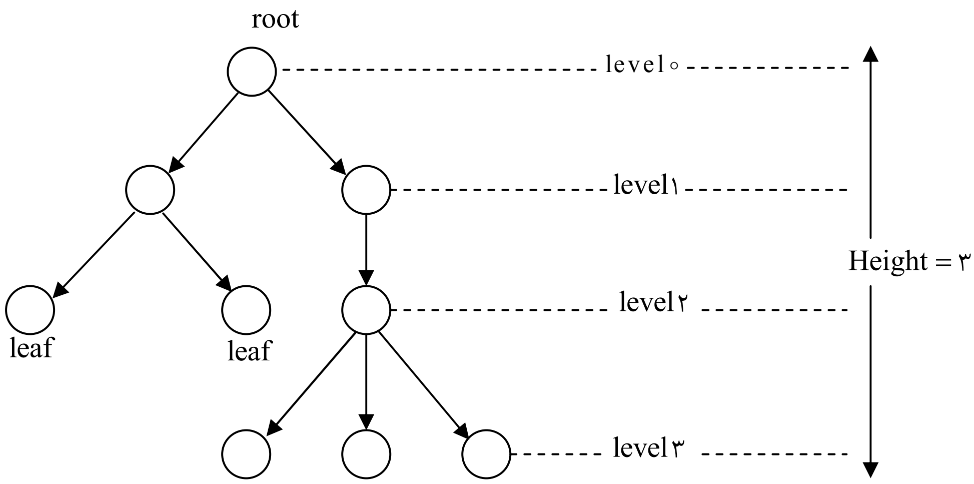
برای راحتی، گراف را به‌صورت یک دیاگرام نمایش می‌دهیم که در آن رئوس را با دایره و یال‌ها را با پیکان‌ها نمایش می‌دهیم که رئوس را به هم متصل می‌کنند.

مثال: گرافی با رئوس $\{V_۱,V_۲,V_۳,V_۴\}$ و یال‌های $\{(V_۱,V_۲),(V_۲,V_۳),(V_۳,V_۲),(V_۱,V_۳)\}$ در شکل زیر نشان داده شده است.



دنباله‌ای از یال‌ها چون $(v_i,v_j)(v_j,v_k)...(v_m,v_n)$ یک گشت (walk) از v_i به v_n نامیده می‌شود. طول گشت، تعداد یال‌هایی است که درون آن گشت وجود دارند. یک گشت که در آن هیچ یالی تکراری نباشد مسیر (path) نامیده می‌شود. یک مسیر، ساده (simple) است. هرگاه هیچ رأسی در آن تکراری نباشد. یک گشت از رأس v_i به خودش یک دور (cycle) با پایه v_i است. اگر هیچ رأسی جز پایه در یک دور تکرار نشده باشد آنگاه آن دور را ساده گویند. در آخر، به یالی از یک رأس به خودش یک طوقه (loop) گویند.

درختان نوع خاصی از گراف‌ها می‌باشند. یک درخت یک گراف جهت‌دار است که هیچ دوری ندارد. و شامل رأس خاصی به نام ریشه می‌باشد به‌طوریکه دقیقاً یک مسیر از ریشه به هر رأس دیگری وجود دارد. بر طبق این تعریف ریشه، هیچ یال ورودی ندارد و رئوسی وجود دارند که هیچ یال خروجی ندارند که به آنها برگ (leaf) گوییم. اگر یالی از v_i به v_j موجود باشد آنگاه گفته می‌شود که v_i والد v_j است. و v_j فرزند v_i می‌باشد. سطح (level) هر رأس، طول مسیری است که از ریشه به آن رأس کشیده شده است. ارتفاع درخت، بزرگ‌ترین سطح ممکن در یک درخت می‌باشد. در شکل زیر این مفاهیم نشان داده شده‌اند.



مفاهیم ابتدایی نظریه زبان‌ها و ماشین‌ها

در بحث نظریه زبان‌ها و ماشین‌ها با سه مفهوم زبان‌ها، گرامرها و ماشین‌ها سروکار داریم که با هرکدام به‌ترتیب آشنا خواهیم شد.

زبان‌ها (Languages)

الفبا (alphabet): مجموعه متناهی و غیرتهی از نمادهاست که آن‌ها را با Σ نشان می‌دهند. نماد شامل حروف، اعداد و ... می‌باشد.

رشته (string): دنباله‌ای متناهی از نمادهای الفباست و معمولاً با حروف u و v و w نشان داده می‌شود.

مثال: اگر مجموعه الفبایی Σ شامل a,b,c باشد؛ یعنی $\sum=\{a,b,c\}$ در این‌صورت رشته‌های زیر متعلق به Σ خواهند بود:

$u = aabbba$

$v = abcb$

$w = a$

طول رشته (string length): تعداد نمادهایی است که یک رشته دارد.

مثال: در مثال قبلی، برای رشته‌های w,v,u داریم:

$|u| = ۶ \quad , \quad |v| = ۴ \quad , \quad |w| = ۱$

رشته تهی (empty string): رشته‌ای است که شامل هیچ نمادی از الفبای Σ نمی‌باشد، یعنی طولش صفر است. رشته تهی را معمولاً با λ یا ε نشان می‌دهند و داریم:

$|\lambda| = ۰$

$\lambda w = w\lambda = w$

زیررشته (substring): هر زیربخش به هم پیوسته در رشته u را یک زیررشته u می‌نامیم.

مثال: اگر $w = acbb$ ، در این‌صورت تمام زیررشته‌های w عبارتند از:

مجموعه تمام زیررشته‌های w $= \{abcd, acb, cbb, ac, cb, bb, a, c, b, \lambda\}$

که abcb یک زیررشته با طول ۴ و λ یک زیررشته به طول صفر می‌باشد.

نکته: λ زیررشته تمام رشته‌ها می‌باشد.

زیررشته‌های پیشوندی و پسوندی

اگر $w = ur$ ، آنگاه زیررشته‌های u و v به‌ترتیب پیشوند و پسوند w نامیده می‌شود.

مثال: در مثال قبل، مجموعه زیررشته‌های پیشوندی و پسوندی w به‌صورت زیر خواهد بود:

مجموعه زیررشته‌های پیشوندی w $= \{acbb, acb, ac, a, \lambda\}$

مجموعه زیررشته‌های پسوندی w $= \{acbb, cbb, bb, b, \lambda\}$

توان‌های یک رشته: اگر w یک رشته باشد، آنگاه w^n یعنی nبار الحاق w با خودش. همچنین طبق تعریف $w = \lambda$. پس به‌صورت بازگشتی می‌توان توان‌های یک رشته را به‌صورت زیر تعریف کرد:

$$\begin{cases} w = \lambda \\ w^n = ww^{n-1} \end{cases}$$

مجموعه توانی الفبا: مجموعه رشته‌های به طول k بر روی مجموعه الفبای Σ را به‌صورت Σ^k تعریف می‌کنیم.

مثال: برای مجموعه الفبای $\Sigma = \{a, b, c\}$ **داریم:** $\Sigma^0 = \{\lambda\}$ ، $\Sigma^1 = \{a, b, c\}$

$$\Sigma^2 = \{aa, ab, ac, ba, bb, ca, cb, cc\}, \ldots \Sigma^3 = \{aaa, aab, \ldots\}$$

بستار ستاره‌ای (star closure): بستار ستاره‌ای حروف الفبایی Σ و مجموعه رشته‌ها به هر طول دلخواه روی مجموعه حروف الفبایی است و به‌صورت زیر تعریف می‌شود:

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

بستار مثبت (positive closure): مجموعه حروف الفبایی Σ مجموعه رشته‌هایی به هر طول دلخواه غیرصفر روی مجموعه Σ است و به‌صورت زیر تعریف می‌شود:

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

نکته: با توجه به تعاریف فوق داریم:

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

نکته: Σ همواره متناهی است. اما Σ^+ و Σ^* هیچ محدودیتی بر روی طول رشته‌ها ندارند. چون اعضای آن‌ها قابل شمارش هستند، لذا نامتناهی شمارش‌پذیر می‌باشند.

نکته: λ زیررشته همه رشته‌های موجود در Σ^* می‌باشد. (λ زیررشته خود نیز می‌باشد.)

عملگر الحاق (اتصال) (coneatenation): الحاق دو رشته u و v بر روی Σ^* را به‌صورت $u.v$ تعریف می‌کنیم. البته اغلب برای نمایش الحاق دو رشته از درج نقطه بین آن‌ها صرف‌نظر می‌کنند.

مثال: اگر $w_1 = ab$ **و** $w_2 = baa$ **در این‌صورت الحاق دو رشته** w_1 **و** w_2 **به‌صورت زیر خواهد بود.**

$$w_1.w_2 = w_1w_2 = abbaa$$

$$w_2.w_1 = w_2w_1 = baaab$$

نکته: با توجه به مثال فوق می‌بینیم که الحاق دارای خاصیت جابه‌جایی نیست. یعنی:

$$w_1w_2 \neq w_2w_1$$

البته اگر $w_1 = w_2$ باشد، در این‌صورت $w_1w_2 = w_2w_1$ خواهد بود.

نکته: اگر $u, v \in \Sigma^*$ باشد، در این‌صورت $w = uv$ هم متعلق به Σ^* خواهد بود.

نکته:

$$w\lambda = \lambda w = w$$

$$w\varnothing = \varnothing w = \varnothing$$

خواص عملگر توان: اگر n ،w بار با خودش الحاق شود، گوییم w به توان n رسیده و به‌صورت w^n نشان می‌دهیم.

مثال: اگر $\Sigma=\{a,b\}$ و $w=ab$، در این‌صورت داریم:

$$w^0=\lambda$$
$$w^1=ab$$
$$w^2=ww=abab$$
$$\vdots$$

نکته: اگر $w=xy$، در این‌صورت داریم:

$$(xy)^n \neq x^n y^n$$

البته اگر $x=y$، در این‌صورت رابطه فوق برقرار خواهد بود.

عملگر معکوس (Revers): معکوس یک رشته w را با w^R نمایش داده و به‌صورت زیر تعریف می‌کنیم:

$$w^R=\begin{cases}\lambda & w=\lambda\\ v^R z & w=zv\end{cases}$$

که $z\in\Sigma$ و $w\in\Sigma^*$

مثال: اگر $w=abbc$ داریم:

$$w^R=(\underset{z}{a}\underset{v}{\underline{bbc}})^R=(\underset{z}{\underline{bb}}\underset{v}{c})^R a=(\underset{z}{c}\underset{v}{b})^R ba$$
$$(\underset{z}{a}\underset{w}{\lambda})^R cbb=(\underset{w}{\lambda})^R cbba=cbba$$

نکته: اگر w_1 و w_2 دو رشته بر روی Σ^* باشد، آنگاه داریم:

$$(w_1^R)^R=w_1$$
$$(w_1w_2)^R=w_2^R w_1^R$$
$$(w_1^n)^R=(w_1^R)^n$$

زبان (Language): به هر زیرمجموعه‌ای از Σ^*، یک زبان روی مجموعه حروف الفبای Σ گویند، به‌عبارتی L یک زبان روی الفبای Σ است، اگر و فقط اگر $L\subseteq\Sigma^*$.

عملگرهای زبان: اجتماع، اشتراک و تفاضل:

اگر L_1 و L_2 دو زبان روی Σ^* باشند، داریم:

$$L_1\cup L_2=\{x\mid x\in L_1 \text{ یا } x\in L_2\}$$
$$L_1\cap L_2=\{x\mid x\in L_1\ ,\ x\in L_2\}$$
$$L_1-L_2=\{x\mid x\in L_1\ ,\ x\notin L_2\}$$

الحاق (اتصال): اگر L_1 و L_2 دو زبان روی Σ^* باشند، داریم:

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

نکته: دقت شود که الفبای هر دو زبان L_1 و L_2 باید یکسان باشد و نباید با یکدیگر فرق کند. مثلاً اگر $\Sigma_1=\{a,b\}$ و $\Sigma_2=\{0,1\}$ و $L_1\subseteq\Sigma_1$ و $L_2\subseteq\Sigma_2$، در این‌صورت الحاق L_1L_2، بی‌معنی خواهد بود.

مثال: برای $\Sigma=\{a,b\}$ و $L_1=\{b,ab\}$ و $L_2=\{b,\lambda\}$ و L_1L_2 و L_2L_1 به‌صورت زیر خواهد بود:

$$L_1L_2=\{ab,b,bb,abb\}$$
$$L_2L_1=\{bb,bab,b,ab\}$$

نکته: اگر L_1 و L_2 دو زبان باشند، داریم:

$$|L_1L_2|\le|L_1||L_2|$$

مثال: با توجه به مثال قبل داریم:

$$۳ = |L_۱L_۲| < |L_۱||L_۲| = ۴$$
$$۴ = |L_۲L_۱| = |L_۲||L_۱| = ۴$$

نکته: داریم:

$$L_۱(L_۲ \cup L_۳) = (L_۱L_۲) \cup (L_۱L_۳)$$
$$L_۱(L_۲ \cap L_۳) \subseteq (L_۱L_۲) \cap (L_۱L_۳)$$

عملگر توان: عمل توان بر روی زبان L به‌صورت زیر تعریف می‌شود:

$$L^۰ = \{\lambda\}$$
$$L^۱ = L$$
$$L^۲ = LL$$
$$L^۳ = L^۲L$$
$$\vdots$$
$$L^n = L^{n-۱}L$$

نکته: با توجه به این‌که زبان L، شامل مجموعه‌ای از رشته‌هاست، لذا باید به‌صورت یک مجموعه با آن رفتار کرد، در نتیجه برای زبان $L^۰$ داریم: $L^۰ = \{\lambda\}$

و درست نیست که بنویسیم $L^۰ = \lambda$.

مثال: اگر $L = \{a^nb^n \mid n \ge ۰\}$، در این‌صورت $L^۲ = \{a^nb^na^mb^m \mid n,m \ge ۰\}$ توجه شود که در این‌جا n,m از هم مستقل می‌باشند و رشته‌ای مثل $\underbrace{ab}_{L}\underbrace{aabb}_{L}$ عضوی از زبان $L^۲$ می‌باشد.

عملگرهای بستار ستاره‌ای و بستار مثبت: بستار ستاره‌ای زبان L به‌صورت زیر تعریف می‌شود:

$$L^* = L^۰ \cup L^۱ \cup \ldots = \bigcup_{i \ge ۰} L^i$$

همچنین برای بستار مثبت داریم:

$$L^+ = L^۱ \cup L^۲ \cup \ldots = \bigcup_{i \ge ۱} L^i$$

نکته: با توجه به تعریف L^* و L^+ داریم:

$$L^* = L^+ \cup \{\lambda\}$$
$$L^+ = L^*L$$
$$(L^*)^* = (L^+)^* = (L^*)^+ = (L^*)^n = (L^n)^* = L^*$$
$$(L^+)^+ = (L^+)^n = (L^n)^+ = L^+$$

مثال: اگر $L_۱ = \{a\}$ و $L_۲ = \{\lambda\}$ و $L_۳ = \varnothing$، بستار ستاره‌ای و مثبت این سه زبان عبارتند از:

$$L_۱^* = \{\lambda, a, a^۲, a^۳, \ldots\}$$
$$L_۱^+ = L_۱^*L_۱ = \{\lambda, a, a^۲, a^۳, \ldots\}\{\lambda, a, a^۲, a^۳, \ldots\} = \{a, a^۲, a^۳, \ldots\}$$
$$L_۲^* = \{\lambda\}^* = \{\lambda\}$$
$$L_۳^* = \varnothing^* = \{\lambda\}$$
$$L_۲^+ = L_۲^*L_۲ = \{\lambda\}L_۲ = L_۲$$
$$L_۳^+ = \{\lambda\}L_۳ = \varnothing$$

نکته: با توجه به مثال فوق می‌بینیم که λ در بستار ستاره‌ای همه زبان‌ها از جمله $\{\lambda\}$ و $\varnothing$ وجود دارد.

عملگر متمم (complement): متمم زبان L را با $\overline{L}$ نشان می‌دهند و به‌صورت زیر تعریف می‌کنند:

$$\overline{L}=\Sigma^{*}-L$$

نکته: $\overline{(L)^{*}}$ و $(\overline{L})^{*}$ باهم برابر نمی‌باشند. زیرا با توجه به این‌که بستار ستاره‌ای هر زبانی شامل λ است، پس بستار $\overline{L}$ یعنی $(\overline{L})^{*}$ شامل λ است. ولی با توجه به این‌که L^{*} شامل λ است، لذا متمم آن یعنی $\overline{(L^{*})}$ نمی‌تواند شامل λ باشد. یعنی $(\overline{L})^{*}\neq\overline{(L^{*})}$.

عملگر معکوس: معکوس زبان L را با L^{R} نشان داده و به‌صورت زیر تعریف می‌کنیم.

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

مثال: معکوس زبان های $L_{1}=\{\lambda,ab,abb\}$ **و** $L_{2}=\left\{a^{n}b^{n}\mid n\geq 0\right\}$ **به‌ترتیب** $L_{1}^{R}=\{\lambda,ba,bba\}$ **و** $L_{2}^{R}=\left\{b^{n}a^{n}\mid n\geq 0\right\}$ **می‌باشد.**

✓ **تست: فرض کنید:**

$$L_{1}=\left\{w\mid w\in\{a,b\}^{*},|w| \bmod 3\geq 1\right\}$$
$$L_{2}=\{a,b\}$$
$$L_{3}=\{aa,ab,ba,bb\}$$
$$L_{4}=\{aaa,aab,aba,abb,baa,bab,bba,bbb\}$$

کدام‌یک از گزینه‌های زیر صحیح است؟ **(مهندسی کامپیوتر- آزاد ۸۰)**

۱) $L_{1}=L_{4}*L_{2}\cup L_{4}*L_{4}$
۲) $L_{1}=L_{2}*L_{3}\cup L_{2}*L_{4}$
۳) $L_{1}=L_{4}*L_{2}\cup L_{4}*L_{3}$
۴) $L_{1}=L_{4}*L_{2}$

☑ **پاسخ:** گزینه «۳»

در گزینه ۳، ستاره روی L_{4} رشته‌هایی را ایجاد می‌نماید که طول آن مضرب ۳ است زیرا همان‌طور که مشاهده می‌نمایید L_4 از مجموعه‌ای از رشته‌ها دقیقا با طول ۳ ساخته شده است بنابراین L_{4}^{*} نیز رشته‌هایی با طول مضرب ۳ ایجاد خواهد کرد. الحاق این زبان با L_{2} و یا حتی L_{3} که رشته‌هایی به طول یک و دو هستند طول رشته‌ها را به‌ترتیب "مضرب ۳+۱" یا " مضرب ۳+۲" خواهد کرد که باقیمانده طول چنین رشته‌هایی بر عدد ۳ بزرگتر و مساوی ۱ خواهد شد که معادل زبان خواهد بود.

گرامرها (Grammars)

برای مطالعه دقیق زبان‌ها نیاز به یک مکانیسم برای تشریح آن‌ها داریم. در اینجا از ابزار گرامر برای این‌کار استفاده می‌کنیم.

تعریف: یک گرامر چون G، چهارتایی مرتبی به‌صورت (V,T,S,P) **است که در آن:**

V: مجموعه متناهی از متغیرها یا ناپایانه‌هاست (Variables).

T: مجموعه متناهی از ترمینال‌ها یا پایانه‌هاست (Terminals).

$S\in V$: متغیر شروع گرامر است. (Start Variable)

P: مجموعه متناهی از قوانین تولید است. (Production rules)

فرض بر این است که مجموعه‌های V و T غیرتهی و مجزا هستند.

مثال: گرامر $G=(\{S,A,B\},\{a,b\},S,P)$ که مجموعه قوانین آن به‌صورت زیر هستند را درنظر بگیرید.

$$\begin{cases} S \to AB \\ A \to aA \mid a \\ B \to bB \mid b \end{cases}$$

اشتقاق (Derivation): مراحل استنتاج یک جمله (Sentence) از زبان با استفاده از تعداد محدودی قواعد را اشتقاق گویند. مثلاً با اشتقاق زیر می‌توان رشته abb را تولید کرد.

$$S \Rightarrow AB \Rightarrow aB \Rightarrow abB \Rightarrow abb$$

تعریف: زبان تولیدشده توسط گرامر $G=(V,T,S,P)$ به‌صورت مجموعه زیر تعریف می‌شود:

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

علامت * که در بالای علامت اشتقاق در تعریف بالا آمده است، بیانگر آن است که بعد از یک یا چند مرحله اشتقاق ناپایانه S، رشته w را مشتق کند.

اگر $w \in L(G)$، آنگاه دنباله:

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n \Rightarrow w$$

اشتقاق جمله w نامیده می‌شود. رشته‌های S، $w_1, w_2, \dots, w_n$ که شامل ناپایانه و پایانه می‌باشند، فرم جمله‌ای (sentential form) نامیده می‌شوند.

مثال: زبان پذیرفته‌شده توسط گرامر زیر را بیابید:

$$S \to aSb \mid \lambda$$

حل:

برای به‌دست آوردن این زبان ابتدا چندین رشته از این زبان را به‌دست می‌آوریم:

$$S \Rightarrow \lambda$$
$$S \Rightarrow aSb \Rightarrow ab$$
$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

بدین‌ترتیب واضح است که زبان این گرامر $L=\{a^n b^n \mid n \ge 0\}$ می‌باشد.

مثال: گرامر زبان $L=\{a^n b^{n+1} \mid n \ge 0\}$ به‌صورت زیر می‌باشد:

$$\begin{cases} S \to Ab \\ A \to aAb \\ A \to \lambda \end{cases}$$

البته توجه داشته باشید که برای یک زبان می‌توان چندین گرامر نوشت که لزوماً شبیه به هم نیستند. چیزی که در این‌جا مهم است، درک این مکانیسم می‌باشد. در این مثال، زبان داده شده به‌گونه‌ای است که در رشته‌های آن ابتدا تعدادی a می‌آید و سپس تعدادی b پشت سر آن قرار می‌گیرد به‌طوری‌که تعداد bها یکی بیش‌تر از تعداد aها می‌باشد. در خط ابتدای گرامر یکی b را در انتهای رشته تولید کرده‌ایم و آن را در کنار ناپایانه A قرار داده‌ایم. ناپایانه A در خط دوم تعدادی مساوی a و b به‌صورت الگوی $a^n b^n$ تولید می‌کند. و در آخر قاعده $A \to \lambda$ را قرار داده‌ایم تا این اشتقاق تا بی‌نهایت ادامه پیدا نکند.

البته زبان‌هایی وجود دارند که تشخیص گرامر آن‌ها به این سادگی نیست و به اندکی تمرین نیاز دارند که در مثال‌های زیر نمونه‌ای از آن‌ها آورده شده‌اند.

مثال: گرامر زبان $L=\{w \mid n_a(w)=n_b(w)\}$ به‌صورت زیر است:

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

منظور از $n_a(w)$ تعداد aهای رشته w و منظور از $n_b(w)$ تعداد bهای رشته w می‌باشد.

✓ **تست: اگر** $L_1 = \{a^n b^m \mid n \geq 0, m > n\}$ **و** $L_2 = \{a^{2n} b^{2n} \mid n \geq 0\}$ **باشند کدام‌یک از گرامرهای زیر زبان** $L_1 \cap L_2$ **را تولید می‌نمایند** **(مهندسی کامپیوتر- آزاد ۸۴)**

۱) $S \to DB, D \to aDb \mid \lambda, B \to Bb \mid b$

۲) $S \to AX, X \to aX \mid \lambda, A \to aaB, B \to a$

۳) $S \to SX, X \to aXb \mid \lambda, S \to ac, C \to aD, D \to \lambda$

۴) $S \to asbb, S \to abb$

☑ **پاسخ: گزینه «۴»**

در زبان L_2 همواره تعداد bهای ظاهر شده از تعداد aها بیشتر است. بنـابراین $L_1 \cap L_2 = L_2$ و چـون $m > n$ اسـت؛ بنـابراین حالت $m = n = 0$ وجود نخواهد داشت.

✓ **تست: اگر داشته باشیم:**

$S \to abc, S \to axbc, xb \to bx, xc \to ybcc$
$by \to yb, ay \to aax, ay \to aa$

کدام‌یک از گزینه‌های ذیل صحیح است؟ **(مهندسی کامپیوتر- آزاد ۸۴)**

۱) $S \overset{*}{\Rightarrow} a^n x b^n c^n \quad n \geq 1$

۲) $S \overset{*}{\Rightarrow} a^n b^n x c^n \quad n \geq 1$

۳) $S \overset{*}{\Rightarrow} a^n b^n x c^n$

۴) هر سه مورد

☑ **پاسخ: گزینه «۴»**

ماشین‌ها (Automata)

ماشین، یک مدل انتزاعی از کامپیوترهای دیجیتالی می‌باشد. هر ماشین به یک سری ابزار ضروری برای کار کردن احتیاج دارد کـه عبارت‌اند از فایل ورودی که بتواند رشته ورودی را دریافت کند، یک مکان ذخیره‌سازی تا بتواند خروجی‌ها را در آن ذخیـره کنـد و همچنین واحد کنترل که می‌تواند متشکل از چندین حالت (state) متناهی درونی باشد، که در آن تغییر حالت براسـاس قواعـد از قبل تعیین شده‌ای امکان‌پذیر است.

در یک ماشین معین (deterministic) در هر حالت یک و تنها یک حرکت توسط تابع انتقال تعریف شده اسـت ولـی در یـک ماشین نامعین (nondeterministic) در هر حالت با الفبای ورودی خاص ممکن است چندین حرکت تعریف شده باشد و تشخیص این‌که از کدام راه حرکت کنیم بر پایه حدس و گمان است.

ماشین‌هایی که خروجی آن‌ها محدود به "yes" یا "No" هستند در اصطلاح پذیرنده (accepter) نامیده می‌شوند. با دادن یک رشته ورودی به یک پذیرنده، رشته ورودی پذیرفته یا رد خواهد شد. نوع عـام‌تـری از ماشـین‌هـا کـه مـی‌توانـد در خروجـی خـود رشته‌هایی از نمادها را تولید کند، ماشین انتقالی (transducer) نام دارد.

سوالات چهارگزینه‌ای سراسری فصل اول

۱- فرض کنید A و B و C زبان‌هایی هستند که روی یک الفبا تعریف شده‌اند. کدام‌یک از گزاره‌های زیر درست است؟
(مهندسی کامپیوتر ۷۵)

۱) $A(B\cap C)=(A.B)\cap(A.C)$

۲) به‌ازای هر زبان A، زبان نامنظم B وجود دارد به‌گونه‌ای که $B\subseteq A$.

۳) به‌ازای هر زبان A، زبان نامنظم B وجود دارد به‌گونه‌ای که $B\supseteq A$.

۴) $A.(B\cup C)=(A.B)\cup(A.C)$

۲- زبان‌های مستقل از متن L_1 و L_2 به شرح زیر مفروضند.

$$L_1=\{a^n baa^m \mid n\ge m\ge 0\}$$
$$L_2=\{a^n b^m \mid n>m\ge 0\}$$

کدام گزینه در مورد زبان $L=\{x \mid xy\in L_1 \text{ and } y\in L_2\}$ درست است؟
(مهندسی کامپیوتر ۷۹)

۱) $L=\{a^n b \mid n\ge 0\}$

۲) $L=\{a^n ba \mid n\ge 0\}$

۳) $L=\{a^n ba^{m+1} \mid n\ge m\ge 0\}$

۴) $L=\{a^n ba^m \mid n\ge m\ge 0\}$

۳- فرض کنید C,B,A زبان‌هایی هستند که در روی یک الفبا تعریف می‌شوند. کدام‌یک از گزاره‌های زیر درست است؟
(علوم کامپیوتر ۸۰)

۱) $(A\cap B)^*=(A^*)\cap(B^*)$

۲) $(A\cap B)C=(AC)\cap(BC)$

۳) $(A^*\cup B^*)C^*=(A^*C^*)\cup(B^*C^*)$

۴) $(A\cup B)^*=(A^*)\cup(B^*)$

۴- کدام‌یک از گزاره‌های زیر درست است؟
(علوم کامپیوتر ۸۰)

۱) مجموعه تمام ماشین‌های تورینگ (Turing machines) روی یک الفبا ناشمارا (uncountable) است.

۲) مجموعه همه زبان‌های نامنظم (non-regular) روی یک الفبا شمارا (countable) است.

۳) مجموعه تمام ماشین‌های تورینگ (Turing machines) روی یک الفبا شمارا (countable) است.

۴) مجموعه همه رشته‌های تعریف شده روی یک الفبا ناشمارا (uncountable) است.

۵- فرض کنید $A=\{0\}$ و $B=\{\lambda,1,10\}$ دو زبان در $\{0,1\}^*$ باشند. در مورد معادله $X=A\cup XB$ برای زبان مجهول $X\subseteq\{0,1\}^*$ کدام گزاره صحیح است؟
(علوم کامپیوتر ۸۷)

۱) $X=AB^*$ تنها جواب یکتای این معادله است.

۲) $X=B^*A$ تنها جواب یکتای معادله است.

۳) فقط برای هر زبان متناهی $C\subseteq\{0,1\}^*$، $X=B^*(A\cap C)$ یک جواب معادله است.

۴) برای هر زبان $C\subseteq\{0,1\}^*$، $X=(A\cup C)B^*$، یک جواب معادله است.

۶- برای زبان داده شده $L\subseteq L\Sigma^*$ فرض کنید $L\cap\Sigma^+=\varnothing$، کدام گزاره نادرست است؟
(علوم کامپیوتر ۸۷)

۱) L می‌تواند زبان تهی باشد.

۲) L حتماً زبانی منظم است.

۳) اگر $\lambda\in L$ آنگاه $L=\{\lambda\}$

۴) اگر $\lambda\notin L$ آنگاه برای زبان $L'=L-L\Sigma^+$ نیز داریم $L'\cap L'\Sigma^+=\varnothing$

۷- گرامر G و زبان‌های L_1 و L_2 مفروضند. ارتباط L(G) با L_1 و L_2 کدام است؟ ε نشانه رشته‌ای به طول صفر است.

(مهندسی کامپیوتر ۸۷)

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

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

$L_2 = \{w \in (a+b)^* \mid$ تعداد ab های w با ba های w برابر است$\}$

۱) $L(G) \subset L_1$

۲) $L(G) = L_1$

۳) $L(G) = L_1 \cup L_2$

۴) $L_2 \subset L(G)$

۸- گرامر وابسته به متن G به شرح زیر مفروض است. کدام‌یک از مجموعه رشته‌های ۱ تا ۴، زیرمجموعه L(G) است؟

(مهندسی کامپیوتر ۸۷)

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

۱) $\{aa, aaaa\}$

۲) $\{aaa, aaaaa\}$

۳) $\{a, aaa, aaaaa\}$

۴) $\{aaaa, aaaaaa\}$

۹- گرامر G به شرح زیر مفروض است. L(G) کدام است؟ (w^R عبارت است از w که از آخر بـه اول خوانـده شـود. و ε نشـانه رشته‌های به طول صفر است.)

(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) $\{ww^R \mid w \in (a+b)^*\}$

۱۰- گرامر G را درنظر می‌گیریم و زبان آن را L می‌نامیم. رشته‌های w_1 و w_2 با تعریف زیر را نیز درنظر می‌گیریم. کدام گـزاره صحیح است؟

(مهندسی کامپیوتر ۸۸)

$G: S \to aSD \mid bB$
$D \to dS \mid a$
$B \to bB \mid \varepsilon$
$w_1 = a^{10}ba^{7}bab^{10}d$
$w_2 = a^{10}b^{9}a^{10}d$

۱) $w_1, w_2 \in L$

۲) $w_1, w_2 \notin L$

۳) $w_2 \notin L, w_1 \in L$

۴) $w_1 \notin L, w_2 \in L$

۱۱- گرامر وابسته به متن G مفروض است:

$$G: S \rightarrow S_1 B$$
$$S_1 \rightarrow aS_1b$$
$$bB \rightarrow bbbB$$
$$aS_1b \rightarrow aa$$
$$B \rightarrow \varepsilon$$

زبان گرامر G کدام است؟ **(مهندسی کامپیوتر ۸۹)**

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

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

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

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

۱۲- گرامر G و رشته‌های w_1 و w_2 به شرح زیر مفروض‌اند:

$$S \rightarrow ac\ BdeA \mid BAB$$
$$B \rightarrow a\ Sb \mid ae \mid \varepsilon$$
$$A \rightarrow a\ Ab \mid b \mid \varepsilon$$
$$w_1 = acaaca\ \ bbdebdeb$$
$$w_2 = acaacaaeebdebbdeabb$$

کدام گزینه صحیح است؟ **(مهندسی کامپیوتر ۹۰)**

۱) $w_1 \in L(G)\ ,\ \ w_2 \notin L(G)$

۲) $w_1 \notin L(G)\ ,\ \ w_2 \in L(G)$

۳) $w_1 \in L(G)\ ,\ \ w_2 \in L(G)$

۴) $w_1 \notin L(G)\ ,\ \ w_2 \notin L(G)$

۱۳- مجموعه A را «شمارش‌پذیر» می‌نامیم اگر A متناهی یا در تناظر یک به یک با مجموعه اعداد طبیعی باشد. درغیراین‌صورت A را «ناشمارا» می‌گوییم. فرض کنید Σ یک الفبای متناهی دلخواه باشد. کدام گزینه از گزینه‌های زیر صحیح نیستند؟ **(مهندسی کامپیوتر ۹۱)**

الف) هر زبان دلخواه بر الفبای Σ شمارش‌پذیر است.

ب) مجموعه تمامی زبان‌های ممکن از الفبای Σ شمارش‌پذیر هستند.

ج) برای هر زبان دلخواه از الفبای Σ می‌توان یک گرامر صوری تولیدکننده درنظر گرفت.

د) هر زبان دلخواه از الفبای Σ که توسط یک گرامر صوری تولیدشدنی باشد بازگشتی است.

۱) ب، ج ۲) الف، ب، ج ۳) الف، ج، د ۴) ب، ج، د

۱۴- زبان L را به شکل زیر تعریف می‌کنیم:

$L = \{w \mid w \in \{a,b\}^*$ و **در هر پیشوند دلخواه از رشته w، تعداد a ها حداقل به تعداد b ها است**$\}$

کدام گرامر زیر، تولیدکننده تمامی رشته‌های زبان L خواهد بود؟ **(مهندسی کامپیوتر ۹۱)**

۱) $S \rightarrow aS \mid aSbS \mid \varepsilon$

۲) $S \rightarrow aS \mid aSb \mid \varepsilon$

۳) $S \rightarrow Sb \mid aSSb \mid a$

۴) $S \rightarrow aS \mid Sa \mid\mid aSb \mid ab\varepsilon$

پاسخنامه سوالات چهارگزینه‌ای سراسری فصل اول

۱-گزینه «۴»

☜ کوچک‌ترین زبان، زبان تهی است که منظم است، بنابراین هیچ زبانی نیست که زیرمجموعه زبان تهی باشد. بزرگ‌ترین زبان، Σ^* است که منظم است، لذا هیچ زبانی نیست که Σ^* زیرمجموعه آن باشد. در نتیجه $A = \phi$ ، مثال نقض گزینه ۲ و $A = \Sigma^*$ مثال نقض گزینه ۳ می‌باشد.

در مورد عدم صحت گزینه ۱ هم در متن درس آمده است.

۲- گزینه «۴»

☜ زبان L همان خارج‌قسمت راست L_1/L_2 می‌باشد. $w_1 = ba \in L_1$ و $w_2 = a \in L_2$ و لذا $w = w_1/w_2 = b$ در نتیجه b باید متعلق به زبان L باشد، ولی گزینه‌های ۲ و ۴، رشته b را نمی‌پذیرند.

همچنین $w'_1 = a^n ba^2 \in L_1$ و $w'_2 = a \in L_2$ و لذا $w' = w'_1 | w'_2 = a^n ba$ ، درنتیجه w' باید توسط L پذیرفته شود، اما گزینه ۱، این رشته را نمی‌پذیرد.

۳-گزینه «۳»

۴- گزینه «۳»

☜ هر مجموعه‌ای یا شماراست و یا ناشمارا، لذا یکی از دو گزینه ۱ و ۳ درست است. همچنین تعداد ماشین‌های تورینگ، نامتناهی و شمارا می‌باشد.

۵- گزینه «۴»

☜ $X = A \bigcup XB$ را به‌صورت $X \to A \mid XB$ درنظر می‌گیریم. با اشتقاق از x داریم:

$$x \mapsto xB \mapsto xBB \overset{*}{\mapsto} xB^n \mapsto AB^n$$

۶- گزینه «۲»

☜ زبان مستقل از متن $a^n b^n c^n$ ، مثال نقض گزینه ۲ می‌باشد.

با زبان مستقل از متن در فصول بعدی آشنا می‌شوید.

۷- گزینه «۲»

☜ نمی‌توانید رشته‌ای بیابید که توسط L_1 پذیرفته شود ولی توسط $L(G)$ پذیرفته نشود.

۸- گزینه «۲»

☜ برای تولید رشته aaa می‌توان از اشتقاق زیر استفاده کرد:

$$S \to ACaB \to AaaCB \to AaaE \to AaEa \to AEaa \to Eaaa \to aaa$$

و برای تولید رشته aaaaa می‌توان از اشتقاق زیر استفاده نمود:

$$S \to ACaB \to AaaCB \to AaaDB \to AaDaB \to ADaaB \to ACaaB \to AaaCaB \to AaaaaCB \to$$
$$AaaaaE \to AaaaEa \to AaaEaa \to AaEaaa \to AEaaaa \to Eaaaaa \to aaaaa$$

۹- گزینه «۲»

☜ می‌توان با جایگذاری قواعد A و B گرامر را به فرم زیر ساده کرد:

$$S \to aSa \mid bsb \mid a \mid b$$

که با توجه به آن واضح است که تمام رشته‌های پالیندروم توسط این گرامر پذیرفته می‌شوند.

۱۰- گزینه «۲»

☜ رشته تولید شده توسط گرامر داده شده هیچ‌گاه نمی‌تواند به d ختم شود، بنابراین هیچ‌کدام از رشته‌های داده شده پذیرفته نمی‌شوند.

۱۱- گزینه «۴»

☜ با توجه به اشتقاق زیر، رشته aa توسط گرامر داده شده تولید می‌شود، در حالی‌که در هیچ‌یک از زبان‌های گزینه‌های ۱ و ۳ ، رشته aa تولید نمی‌شود، بنابراین این گزینه‌ها نادرست می‌باشند.

$$S \to S_1B \to aS_1bB \to aa\varepsilon = aa$$

و با توجه به قاعده $S_1 \to aS_1b$ متوجه می‌شویم که ارتباطی بین تعداد aها و bها وجود دارد بنابراین گزینه ۲ نیز رد می‌شود.

۱۲- گزینه «۱»

☜ رشتهٔ w_1 را می‌توان توسط اشتقاق زیر پذیرفت:

$$S \to acBdeA \to acaSbdeA \to acaacBdeAbdeA \to acaacaSbdeAbdeA$$
$$\to acaacaBABbdeAbdeA \to acaacabbdebdeb$$

ولی از آنجایی‌که در گرامر داده شده یا قبل از e، d یا a قرار می‌گیرد بنابراین نمی‌توان ee را تولید نمود بنابراین w_2 نمی‌تواند عضوی از زبان گرامر داده شده باشد.

۱۳- گزینه «۴»

☜ اگر Σ مجموعه متناهی باشد، آنگاه Σ^* یک مجموعه نامتناهی ولی شماراست، و هر زبانی که زیرمجموعه‌ای از آن باشد شمارا است، ولی 2^{Σ^*} مجموعه‌ای نامتناهی و ناشماراست. از آن‌جایی‌که مجموعه زبان‌هایی که کل گرامرهای صوری می‌سازند مجموعه‌ای نامتناهی ولی شماراست. بنابراین زبانی وجود دارد که برای آن نمی‌توان گرامر صوری تولید کرد. یک گرامر صورتی می‌تواند زبان r.e. تولید کند که بازگشتی نباشد.

۱۴- گزینه «۱»

☜ گزینه‌های ۳، ۴، ε را تولید نمی‌کنند، درحالی‌که زبان L شامل ε می‌باشد.

گزینه ۲، رشته aba را تولید نمی‌کند، درحالی‌که زبان L شامل این رشته می‌باشد.

سوالات چهارگزینه‌ای آزاد فصل اول

۱- فرض کنید $\varnothing$ مجموعه تهی و λ مجموعه رشته تهی است، کدام‌یک از گزاره‌های زیر درست است؟ (سال ۷۷)

۱) $\lambda^* - \varnothing^* = \varnothing$ ۲) $\lambda^* \varnothing^* = \varnothing$ ۳) $\lambda^* - \varnothing = \varnothing$ ۴) $\varnothing^* = \varnothing$

۲- اگر A^* مجموعه تمام رشته‌های تعریف‌شده روی الفبای A باشد، ..... (سال ۷۸)

۱) A^* یک مجموعه محدود است. ۲) A^* یک مجموعه نامحدود و ناشمارا است.

۳) A^* یک مجموعه شمارا است. ۴) مجموعه تمام زیرمجموعه‌های محدود A^*، ناشمارا است.

۳- چه زبانی توسط گرامری با قوانین زیر روی $\Sigma=\{a,b\}$ تولید می‌شود؟ (مهندسی کامپیوتر ۹۰)

$S \to aA|Aa$
$S \to aA|BA|\lambda$

۱) همه رشته‌هایی که حداقل یک a دارند.
۲) همه رشته‌هایی که دقیقا یک a دارند.
۳) همه رشته‌هایی که حداکثر یک a دارند.
۴) همه رشته‌هایی که هیچ a ندارند.

پاسخنامه سوالات چهارگزینه‌ای آزاد فصل اول

۱- گزینه «۱»

$$\lambda - \lambda = \varnothing \quad , \quad \lambda^* = \varnothing^* = \{\lambda\}$$

۲- گزینه «۳»

☜ A^* یک مجموعه نامتناهی و شمارا می‌باشد. اگر مجموعه‌ای نامتناهی باشد، مجموعه تمام زیرمجموعه‌های آن ناشمارا می‌باشد.

۳- گزینه «۱»

☜ در این زبان برای رفتن از نماد شروع S به A حتما یک a دیده می‌شود و بعد از آن با استفاده از متغیر A می‌توان به هر تعداد a را داشت پس هر رشته‌ای متعلق به این گرامر حداقل یک a دارد.

فصل دوم

ماشین‌های متناهی و زبان‌های منظم

- پذیرنده متناهی معین
- تابع انتقال تعمیم‌یافته
- پذیرنده متناهی نامعین
- تفاوت‌های NFA و DFA
- الگوریتم تبدیل NFA به DFA
- کاهش حالت ماشین‌های متناهی
- عبارات منظم
- تبدیل NFA به عبارت منظم
- گرامرهای منظم
- الگوریتم تبدیل گرامر خطی راست به NFA
- الگوریتم تبدیل NFA به گرامر خطی راست
- الگوریتم تبدیل گرامر خطی چپ به NFA
- الگوریتم تبدیل NFA به گرامر خطی چپ
- تشخیص زبان‌های منظم
- طراحی آتاماتای متناهی برای زبان‌های پیمانه‌ای
- لم تزریق برای تشخیص زبان‌های منظم
- همریختی
- ماشین خارج قسمت دو زبان
- تفاضل متقارن
- نسخه‌ای دیگر از لم تزریق

ماشین‌های متناهی و زبان‌های منظم

در این نوع ماشین‌ها هیچ‌گونه حافظه موقتی وجود ندارد، بنابراین نمی‌توانند چیزی را به‌خاطر بسپارند. بلکه تنها می‌توانند اطلاعات اندکی را با تغییر حالت (state) ذخیره کنند، اما از آنجا که تعداد این حالات متناهی است، این ماشین‌ها تنها قادر به پذیرش زبان‌هایی هستند که بتوان برای تشخیص آنها از حافظه متناهی بهره برد.

پذیرنده متناهی معین (DFA: Deterministic Finite Accepter)

اولین نوع از ماشین‌هایی که در این نوشتار به بررسی آنها می‌پردازیم ماشین‌هایی هستند که از نظر نوع، پذیرنده متناهی معین هستند. تعریف دقیق این DFA ها عبارت است از:

➢ **تعریف:** DFA، یک پنج‌تایی مرتب به فرم $M = (Q,\Sigma,\delta,q_\circ,F)$ می‌باشد که در آن:

Q: مجموعه متناهی از حالات ماشین (internal state)

Σ: مجموعه الفبای ورودی (input alphabet)

$\delta : Q\times\Sigma \to Q$: تابع تام انتقال (transition function)

$q_\circ \in Q$: حالت شروع (initial state)

$F \subseteq Q$: مجموعه متناهی از حالات پایانی (Final states)

طرز کار ماشین: در لحظه شروع کار ماشین در حالت آغازین $q_\circ$ هستیم و از سمت چپ رشته ورودی شروع به خواندن می‌کنیم. در هر حرکت ماشین، یکی از نمادهای رشته ورودی را می‌خوانیم و با توجه به تابع انتقال، وارد حالت جدید می‌شویم. اگر با تمام شدن رشته ورودی، ماشین در یکی از حالات نهایی باشد، رشته پذیرفته شده و در غیراین‌صورت رشته پذیرفته نمی‌شود.

به‌عنوان مثال اگر تابع انتقال به‌صورت $\delta(q_o,a)=q_۱$ تعریف شده باشد آنگاه در حالت $q_\circ$ اگر نماد ورودی a باشد بـه حالـت $q_۱$ می‌رویم.

نکته: تابع انتقال DFA، یک تابع تام است، یعنی در هر حالت برای همه مجموعه الفبایی Σ یک و تنها یک حرکت تعریف شده است.

برای راحتی اغلب از نمایش گرافی برای نشان دادن تابع انتقال استفاده می‌کنند. به‌عنوان مثال اگر $\Sigma = \{\circ,۱\}$ مجموعه الفبای ورودی و δ تابع انتقال یک ماشین DFA با حالات $q_\circ$ و $q_۱$ با حالت شروع $q_\circ$ و حالت نهایی $q_۱$ به‌صورت زیر باشد:

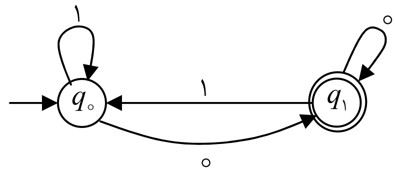
$\delta(q_\circ,۱) = q_\circ$

$\delta(q_\circ,\circ) = q_۱$

$\delta(q_۱,۱) = q_\circ$

$\delta(q_۱,\circ) = q_۱$

آنگاه این ماشین را می‌توان به‌صورت زیر نمایش داد:



در این دیاگرام حالت شروع را با یک پیکان داخلی و حالات نهایی را با دو دایره تودرتو نمایش می‌دهند.

تابع انتقال تعمیم‌یافته (Extended transition Function)

تابع انتقال تعمیم‌یافته به‌صورت $\delta^*: Q \times \Sigma^* \to Q$ تعریف می‌شود. در این تابع، پارامتر دوم به‌جای یک حرف، یک رشته می‌باشد و مقدار آن برابر حالتی است که پس از خواندن آن رشته از حالت شروع به آن می‌رسیم. به‌عنوان مثال اگر داشته باشیم.

$\delta(q_0, a) = q_1 \quad , \quad \delta(q_1, b) = q_2$

آنگاه با تابع انتقال تعمیم‌یافته می‌توان نوشت:

$$\delta^*(q_0, ab) = q_2$$

می‌توان به‌گونه‌ای رسمی‌تر تابع انتقال تعمیم‌یافته را به‌صورت بازگشتی زیر تعریف کرد.

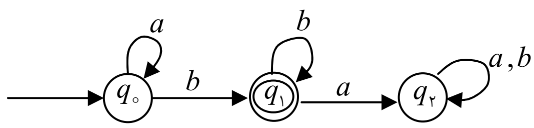
$$\begin{cases} \delta^*(q, \lambda) = q \\ \delta^*(q, aw) = \delta^*\big(\delta(q, a), w\big) \end{cases}$$

بررسی درستی این روابط با استقراء ریاضی به‌سادگی امکان‌پذیر است.

زبان پذیرفته‌شده توسط ماشین متناهی معین: مجموعه همه رشته‌هایی روی الفبای Σ است که توسط ماشین M پذیرفته می‌شود. به‌بیان دقیق‌تر داریم:

$$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$$

حالت تله (trap state): حالتی غیرنهایی است که در آن به‌ازای هر الفبای ورودی روی همان حالت دور زده و هیچ‌گاه نمی‌توان از آن خارج شد، مانند حالت q_2 در شکل زیر:

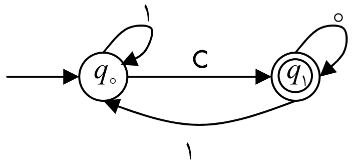


راه دیگر نشان دادن تابع انتقال استفاده از نمایش جدولی تابع می‌باشد. به‌عنوان مثال تابع انتقال دیاگرام فوق را می‌توان به‌صورت جدول زیر نشان داد.

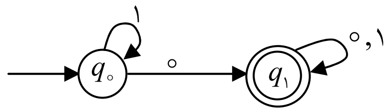
S \ Σ	a	b
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_2

مثال: برای زبان‌های زیر یک dfa رسم کنید. در این مثال فرض بر این است که $\Sigma=\{۰,۱\}$

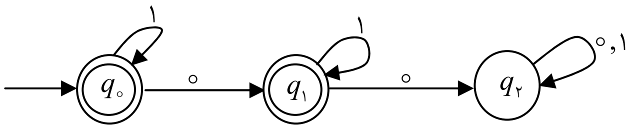
۱) $L_۱=\{w\in\Sigma^* \mid$ به صفر ختم شود $w\}$



۲) $L_۲=\{w\in\Sigma^* \mid$ شامل حداقل یک صفر باشد $w\}$

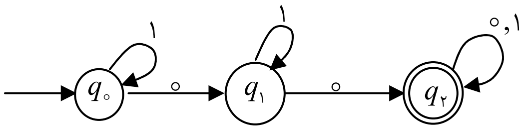


۳) $L_۳=\{w\in\Sigma^* \mid$ شامل حداکثر یک صفر باشد $w\}$

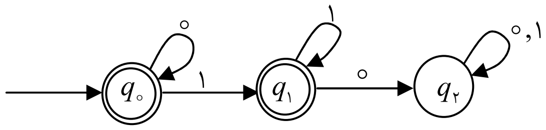


نکته: هرگاه حالت شروع، نهایی هم باشد، در این‌صورت رشته λ نیز پذیرفته خواهد شد. در این مثال $q_۲$ حالت تله است.

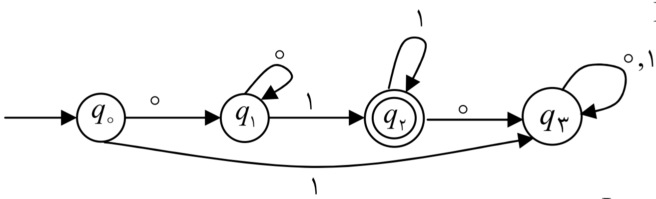
۴) $L_۴=\{w\in\Sigma^* \mid n_۰(w)\geq ۲\}$



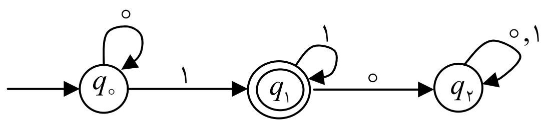
۵) $L_۵=\{۰^m۱^n \mid n,m\geq ۰\}$



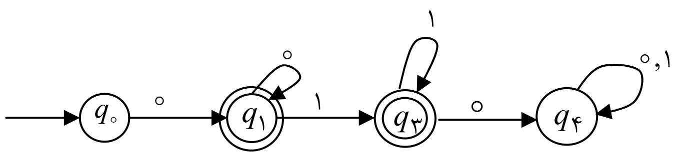
۶) $L_۶=\{۰^m۱^n \mid n,m> ۰\}$



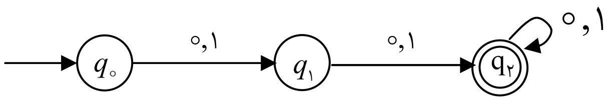
۷) $L_۷=\{۰^m۱^n \mid m\geq ۰, n> ۰,\}$



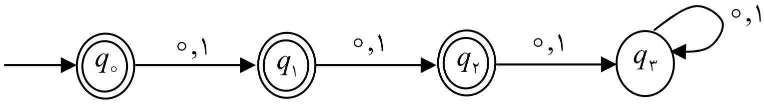
۸) $L_۸=\{۰^m۱^n \mid m> ۰, n\geq ۰\}$



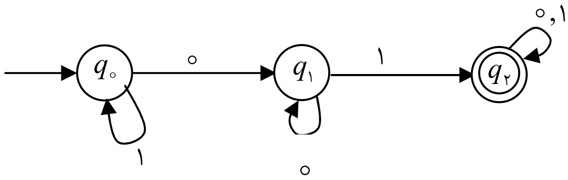
۹) $L_{9} = \{w \in \Sigma^* \mid |w| \geq 2\}$



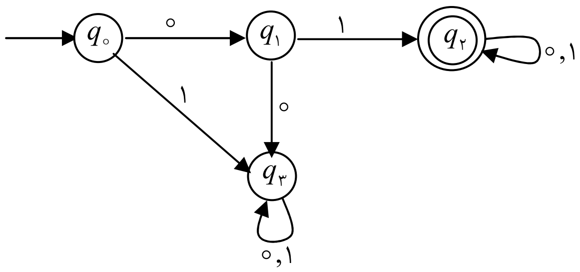
۱۰) $L_{10} = \{w \in \Sigma^* \mid |w| \leq 2\}$



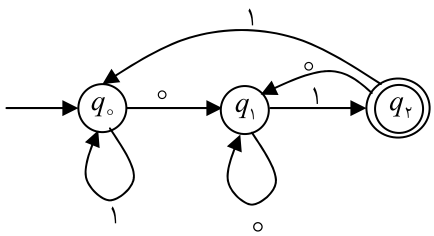
۱۱) $L_{11} = \{w \in \Sigma^*$ w شامل زیررشته ۰۱ باشد$\}$



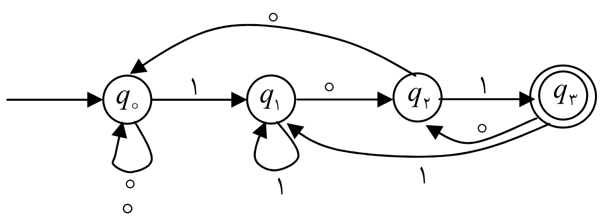
۱۲) $L_{12} = \{w \in \Sigma^* \mid$ w با ۰۱ شروع شود$\}$



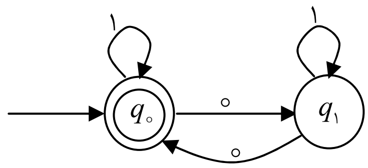
۱۳) $L_{13} = \{w \in \Sigma^* \mid$ w به ۰۱ ختم شود$\}$



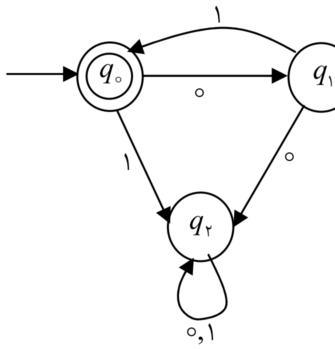
۱۴) $L_{14} = \{w \in \Sigma^* \mid$ w به ۱۰۱ ختم شود$\}$



$L_{۱۵} = \{w \in \Sigma^* \mid n_۰(w) = ۲k, k \geq ۰\}$ (۱۵

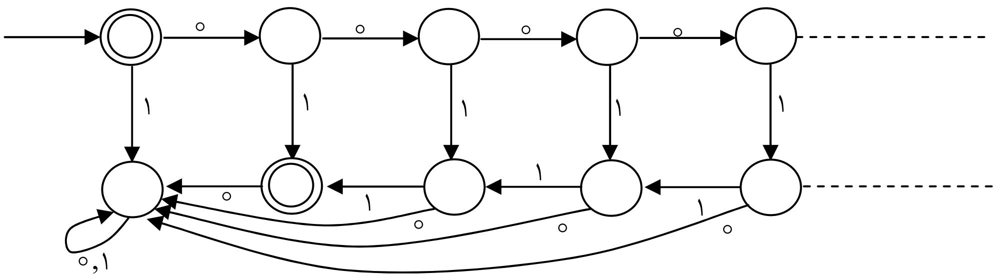


$L_{۱۶} = \{(۰۱)^n \mid n \geq ۰\}$ (۱۶



$L_{۱۷} = \{۰^n۱^n \mid n \geq ۰\}$ (۱۷

این زبان متناهی نیست، لذا نامنظم است و نمی‌توان یک DFA برای آن ایجاد کرد. ممکن است بگویید می‌توان DFA یا همان اتوماتای متناهی زیر را برای زبان $L_{۱۷}$ ارائه کرد.



در پاسخ باید بگوییم، همان‌طور که لفظ «اتوماتای متناهی» برمی‌آید، اتوماتا باید متناهی باشد، در حالی‌که DFA نامتناهی است. مشکل این راه حل این است که مقدار n می‌تواند به دلخواه بزرگ باشد. بنابراین تعداد حالات این ماشین نامتناهی خواهد بود که با متناهی بودن این ماشین در تناقض است.

$L_{۱۸} = \{۰^{۴۰۰}۱^{۴۰۰}\}$ (۱۸

به‌دلیل این‌که ۴۰۰ یک عدد ثابت است و برخلاف n در زبان $L_{۱۷}$ متغیر نمی‌باشد، لذا می‌توان dfa این زبان را رسم کرد.

زبان‌های منظم: خانواده زبان‌های پذیرفته‌شده به‌وسیله DFA، خانواده زبان‌های منظم نامیده می‌شوند. تمامی زبان‌های ارائه شده به‌جز $L_{۱۷}$ در مثال قبل منظم بودند. به‌بیان دیگر داریم:

زبان L منظم است اگر و تنها اگر پذیرنده متناهی معینی چون M وجود داشته باشد که L(M)=L.