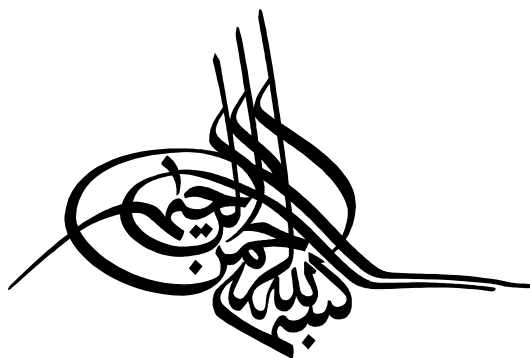




نظریه زبان‌ها و ماشین‌ها

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، ۱۴۰۴
مشخصات ظاهری	: ۱۶۵ ص
فروست	: سری کتاب‌های کمک آموزشی کارشناسی ارشد
شابک	: ۹۷۸-۶۰۰-۴۵۸-۸۰۸-۹
وضعیت فهرست نویسی	: : فنیای مختصر
یادداشت	: این مدرک در آدرس http://opac.nlai.ir قابل دسترسی است.
شناسه افزوده	: : فائزه عباس‌پور قادری
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 / ۱۴۰۴

تیراژ: ۱۰۰۰ جلد

قیمت: ۳/۸۹۰/۰۰۰ ریال

شابک: ISBN: ۹۷۸-۶۰۰-۴۵۸-۸۰۸-۹

انتشارات مشاوران صعود ماهان: خیابان ولیعصر، بالاتر از تقاطع مطهری،
 روبروی قنادی هتل بزرگ تهران، 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 پیتز لینز است. در این کتاب سعی شده است تمامی مطالب مهم سرفصل‌های کنکور سراسری و آزاد گنجانده شود و کاملاً 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

۷	فصل اول: مقدمه
۸	مجموعه‌ها
۱۰	توابع و رابطه‌ها
۱۱	گراف‌ها و درخت‌ها
۱۲	مفاهیم ابتدایی نظریه زبان‌ها و ماشین‌ها
۱۲	زبان‌ها
۱۶	گرامرها
۱۸	ماشین‌ها
۱۹	سوالات چهارگزینه‌ای سراسری و پاسخنامه فصل اول
۲۴	سوالات چهارگزینه‌ای آزاد و پاسخنامه فصل اول
۲۵	فصل دوم: ماشین‌های متناهی و زبان‌های منظم
۲۶	پذیرنده متناهی معین
۲۷	تابع انتقال تعمیم‌یافته
۳۱	پذیرنده متناهی نامعین
۳۲	تفاوت‌های DFA و NFA
۳۶	الگوریتم تبدیل DFA به NFA
۳۷	کاهش حالت ماشین‌های متناهی
۳۹	عبارات منظم
۴۳	تبدیل NFA به عبارت منظم
۴۵	گرامرهای منظم
۵۱	الگوریتم تبدیل گرامر خطی راست به NFA
۵۲	الگوریتم تبدیل NFA به گرامر خطی راست
۵۲	الگوریتم تبدیل گرامر خطی چپ به NFA
۵۳	الگوریتم تبدیل NFA به گرامر خطی چپ
۵۵	تشخیص زبان‌های منظم
۵۶	طراحی آتاماتای متناهی برای زبان‌های پیمانه‌ای
۵۸	لم تزریق برای تشخیص زبان‌های منظم
۶۱	همریختی
۶۱	ماشین خارج قسمت دو زبان
۶۳	تفاضل متقارن

نسخه‌ای دیگر از لم تزریق.....	۶۴
سوالات چهارگزینه‌ای تالیفی و پاسخنامه فصل دوم.....	۶۷
سوالات چهارگزینه‌ای سراسری و پاسخنامه فصل دوم.....	۷۱
سوالات چهارگزینه‌ای آزاد و پاسخنامه فصل دوم.....	۸۴
فصل سوم: ماشین‌های پشته‌ای و زبان‌های مستقل از متن.....	۸۵
گراف‌ها و درخت‌ها.....	۸۷
ماشین‌های پشته‌ای.....	۹۲
زبان‌های مستقل از متن.....	۹۷
گرامرهای LL(K).....	۱۰۳
ساده‌سازی گرامرهای مستقل از متن.....	۱۰۳
فرم‌های نرمال.....	۱۰۶
قاعده بازگشتی از چپ.....	۱۰۸
گرامر خطی و زبان خطی.....	۱۱۱
خواص بستاری و الگوریتم‌های تصمیم‌گیری.....	۱۱۲
سوالات چهارگزینه‌ای تالیفی و پاسخنامه فصل سوم.....	۱۱۳
سوالات چهارگزینه‌ای سراسری و پاسخنامه فصل سوم.....	۱۱۶
سوالات چهارگزینه‌ای آزاد و پاسخنامه فصل سوم.....	۱۳۲
فصل چهارم: ماشین‌های تورینگ.....	۱۳۵
ماشین تورینگ استاندارد.....	۱۳۶
زبان پذیرفته‌شده توسط ماشین تورینگ.....	۱۳۷
تز تورینگ.....	۱۳۹
اعمال تغییرات مختلف روی ماشین تورینگ استاندارد.....	۱۴۰
ماشین تورینگ جهانی.....	۱۴۳
ماشین کراندار خطی.....	۱۴۳
مجموعه‌های شمارش‌پذیر یا برشمردنی.....	۱۴۴
زبان‌های بازگشتی و بازگشتی برشمردنی.....	۱۴۵
تقسیم‌بندی چامسکی.....	۱۴۶
ارتباط بین زبان‌ها.....	۱۴۶
کاهش‌پذیری.....	۱۴۸
مسئله توقف در ماشین‌های تورینگ.....	۱۴۸
سوالات چهارگزینه‌ای تالیفی و پاسخنامه فصل چهارم.....	۱۵۱
سوالات چهارگزینه‌ای سراسری و پاسخنامه فصل چهارم.....	۱۵۲
سوالات چهارگزینه‌ای آزاد و پاسخنامه فصل چهارم.....	۱۶۲
سوالات کنکور سراسری ۹۵.....	۱۶۳
منبع.....	۱۶۵

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^۱: 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 $|S| = 2$. زیرا شامل دو عضو a و b می‌باشد.

مجموعه جهانی یا مرجع^۲: مجموعه‌ای که شامل همه اعضای ممکن بوده را مجموعه مرجع گویند و معمولاً با V یا M نشان می‌دهند. عملگرهای روی مجموعه‌ها: اگر S_1 و S_2 دو مجموعه باشند، اجتماع، اشتراک، تفاضل، متمم (مکمل) و ضرب دکارتی آن‌ها به‌صورت زیر است:

$$-S_1 \cup S_2 = \{x \mid x \in S_1 \text{ یا } x \in S_2\}$$

اجتماع

$$-S_1 \cap S_2 = \{x \mid x \in S_1, x \in S_2\}$$

اشتراک

$$-S_1 - S_2 = \{x \mid x \in S_1, x \notin S_2\}$$

تفاضل

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

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

ضرب دکارتی: ضرب دکارتی دو مجموعه S_1 و S_2 ، 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:

$$S_1 \times S_2 = \{(x, y) \mid x \in S_1, y \in S_2\}$$

که طبق تعریف فوق، اگر $|S_1| = m$ و $|S_2| = n$ ، در این‌صورت $|S_1 \times S_2| = m.n$.

☉ **مثال:** اگر $S_1 = \{1, 2\}$ و $S_2 = \{a, b\}$ باشد، داریم:

$$S_1 \times S_2 = \{(1, a), (1, b), (2, a), (2, b)\}$$

1- set

2- universal set

مجموعه توانی (power set)

مجموعه همه زیرمجموعه‌های یک مجموعه است و آن را با نماد 2^S یا $p(S)$ نمایش می‌دهند.

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

$$2^S = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

افراز یک مجموعه: فرض کنید که مجموعه‌های $S_1, S_2, \dots, S_n$ زیرمجموعه‌هایی از مجموعه S باشند. اگر این زیرمجموعه‌ها در سه شرط زیر صدق کنند، آنگاه گوییم مجموعه S را به مجموعه‌های $S_1, S_2, \dots, S_n$ افراز کرده‌ایم.

۱- زیرمجموعه‌های $S_1, S_2, \dots, S_n$ دویه‌دو مجزا باشند.

$$S_1 \cup S_2 \cup \dots \cup S_n = S \quad 2-$$

۳- هیچ یک از S_i ها تهی نباشند.

قانون دمورگان (Demorgan's laws): طبق قانون دمورگان، روابط زیر را برای دو مجموعه S_1 و S_2 خواهیم داشت:

$$\overline{S_1 \cup S_2} = \bar{S}_1 \cap \bar{S}_2$$

$$\overline{S_1 \cap S_2} = \bar{S}_1 \cup \bar{S}_2$$

رابطه بین مجموعه‌ها: اگر هر عضو S_1 ، عضوی از مجموعه S_2 باشد، آنگاه گوییم که S_1 زیرمجموعه S_2 است و آن را به صورت روبه‌رو نشان می‌دهیم:

$$S_1 \subseteq S_2$$

اگر $S_1 \subseteq S_2$ باشد ولی S_2 شامل عضوی باشد که در S_1 وجود نداشته باشد، آنگاه گوییم که S_1 زیرمجموعه محض S_2 است و آن را به صورت زیر نمایش می‌دهیم:

$$S_1 \subset S_2$$

مجموعه‌های S_1 و S_2 را جدا از هم گوییم هرگاه هیچ عضو مشترکی نداشته باشند؛ یعنی $S_1 \cap S_2 = \emptyset$.

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:

$$S = \{1, 3, 5, 7, 9\} \quad , \quad N = \{1, 2, 3, \dots\}$$

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 منحصر به فردی 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 به صورت 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 غیر این صورت 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 کافیت 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 به طوری که برای همه n های به قدر کافی بزرگ داشته باشیم:

$$f(n) \leq Cg(n)$$

گوییم که f حداکثر رشدی برابر g دارد و می‌نویسیم:

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

اگر $|f(n)| \geq C|g(n)|$ ، 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، به طوری که:

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

آنگاه f و g رشدی برابر دارند و می‌نویسیم:

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

مثال: فرض کنید:

$$f(n) = 4n^2 + 5n + 6$$

$$g(n) = 9n^2$$

$$h(n) = 3n^2$$

آنگاه داریم:

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

$$g(n) = \Omega(h(n))$$

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

می‌توان یک تابع را به صورت مجموعه‌ای از زوج‌های مرتب نشان داد. مثلاً تابع f را به صورت زیر نمایش دهیم:

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

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:

$$R_1 = \{(1, 3), (1, 4)\}$$

$$R_2 = \{(2, 3)\}$$

$$R_3 = \emptyset$$

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

مثال: اگر $A = \{1, 2, 3\}$ آنگاه رابطه‌هایی که روی مجموعه A تعریف می‌شود عبارتند از:

$$R_1 = \{(1,1), (2,2), (3,3)\} \quad , \quad R_2 = \{(1,2), (2,1), (1,3), (3,1)\}$$

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 x های متعلق به A ، x با خودش در رابطه باشد).

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

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

$$R \text{ is symmetric iff if } xRy \text{ then } yRx$$

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

$$R \text{ is transitive iff if } xRy \text{ and } yRz \text{ then } xRz$$

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

مثال: در مثال قبل رابطه R_1 ، بازتابی، متقارن و متعدی و رابطه R_2 فقط متقارن بوده و رابطه تهی، متقارن و متعدی می‌باشد.

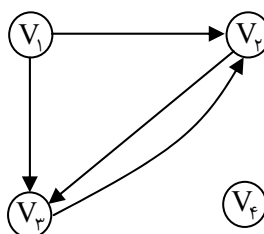
گراف‌ها و درخت‌ها

گراف ساختاری است که از دو مجموعه متناهی $V = \{v_1, v_2, \dots, v_n\}$ از رئوس و $E = \{e_1, e_2, \dots, e_n\}$ از یال‌ها تشکیل شده است. هر یال، زوجی از رئوس می‌باشند. مثلاً:

$$e_i = (v_j, v_k)$$

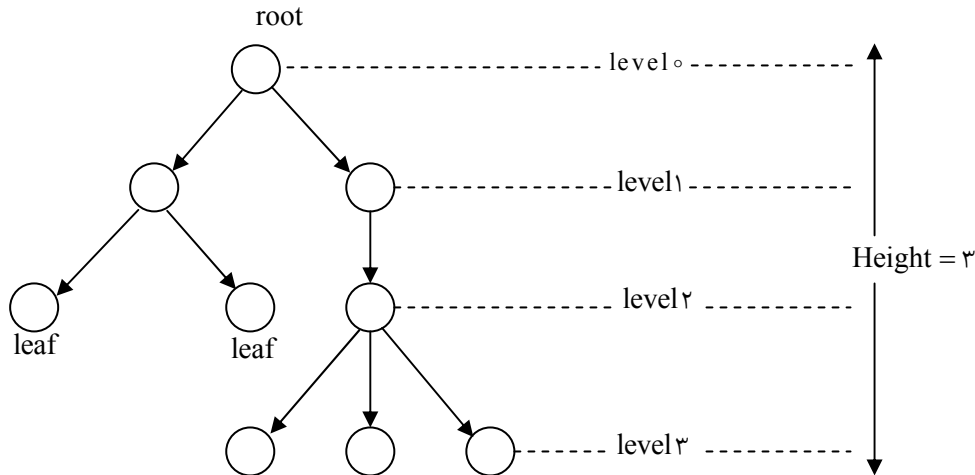
یالی از v_j به v_k است. گوییم یال e_i برای رأس v_j یک یال خروجی و برای رأس v_k یک یال ورودی است. به این ساختار یک گراف جهت‌دار گویند، زیرا هر یال جهتی را نشان می‌دهد. یک گراف ممکن است برچسب‌دار باشد. این برچسب‌ها، هم می‌تواند روی رئوس و هم می‌تواند روی یال‌ها قرار بگیرند. برای راحتی، گراف را به صورت یک دیگرام 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 $\{V_1, V_2, V_3, V_4\}$ و یال‌های $\{(V_1, V_2), (V_2, V_3), (V_3, V_2), (V_1, V_3)\}$ در شکل زیر نشان داده شده است.



دنباله‌ای از یال‌ها چون $(v_i, v_j), (v_j, v_k), \dots, (v_m, v_n)$ 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 هر کدام 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 $\Sigma = \{a, b, c\}$ در این صورت 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:

$$|u| = 6, \quad |v| = 4, \quad |w| = 1$$

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

$$|\lambda| = 0$$

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

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

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

$$W = \{abcd, acb, cbb, ac, cb, bb, a, c, b, \lambda\}$$

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 به ترتیب پیشوند و پسوند w نامیده می‌شود.

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

$$W_{\text{پیشوندی}} = \{acbb, acb, ac, a, \lambda\}$$

$$W_{\text{پسوندی}} = \{acbb, cbb, bb, b, \lambda\}$$

توان‌های یک رشته: اگر w یک رشته باشد، آنگاه w^n یعنی n بار الحاق w با خودش. همچنین طبق تعریف $w = \lambda$. پس به صورت بازگشتی می‌توان توان‌های یک رشته را به صورت 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 به صورت Σ^k تعریف می‌کنیم.

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

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

بستار ستاره‌ای (star closure): بستار ستاره‌ای حروف الفبایی Σ و مجموعه رشته‌ها به هر طول دلخواه روی مجموعه حروف

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

الفبایی است و به صورت زیر تعریف می‌شود:

بستار مثبت (positive closure): مجموعه حروف الفبایی Σ مجموعه رشته‌هایی به هر طول دلخواه غیر صفر روی مجموعه

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

Σ است و به صورت زیر تعریف می‌شود:

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

نکته: با توجه به تعاریف فوق داریم:

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) (concatenation): الحاق دو رشته u و v بر روی Σ^* را به صورت uv تعریف می‌کنیم. البته اغلب برای نمایش الحاق دو رشته از درج نقطه بین آن‌ها صرف نظر می‌کنند.

مثال: اگر $w_1 = ab$ و $w_2 = baa$ در این صورت الحاق دو رشته w_1 و w_2 به صورت زیر خواهد بود.

$$w_1.w_2 = w_2.w_1 = abbaa$$

$$w_2.w_1 = w_1.w_2 = baaab$$

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

$$w_1.w_2 \neq w_2.w_1$$

البته اگر $w_1 = w_2$ باشد، در این صورت $w_1.w_2 = w_2.w_1$ خواهد بود.

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

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

نکته:

$$w\emptyset = \emptyset w = \emptyset$$

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

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

$$\begin{aligned} w^0 &= \lambda \\ w^1 &= ab \\ w^2 &= ww = abab \\ &\vdots \end{aligned}$$

نکته: اگر $w = xy$ ، در این صورت داریم:

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

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

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

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

که $w \in \Sigma^*$ و $z \in \Sigma$

مثال: اگر $w = abbc$ داریم:

$$w^R = (\underline{a} \underline{bb} \underline{c})^R = (\underline{c} \underline{bb} \underline{a})^R = (\underline{c} \underline{b} \underline{b})^R \underline{a} = \underline{a} \underline{b} \underline{b} \underline{c}$$

$$(\underline{a} \underline{\lambda})^R \underline{c} \underline{bb} = (\underline{\lambda})^R \underline{c} \underline{bb} \underline{a} = \underline{c} \underline{bb} \underline{a}$$

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

$$\begin{aligned} (w_1^R)^R &= w_1 \\ (w_1 w_2)^R &= w_2^R w_1^R \\ (w_1^n)^R &= (w_1^R)^n \end{aligned}$$

زبان (Language): به هر زیرمجموعه‌ای از Σ^* ، یک زبان روی مجموعه حروف الفبای Σ گویند، به عبارتی 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:

$$\begin{aligned} L_1 \cup L_2 &= \{x \mid x \in L_1 \text{ یا } x \in L_2\} \\ L_1 \cap L_2 &= \{x \mid x \in L_1 \text{ و } x \in L_2\} \\ L_1 - L_2 &= \{x \mid x \in L_1, x \notin L_2\} \end{aligned}$$

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

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

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

$\Sigma_2 = \{c, d\}$ ، $L_1 \subseteq \Sigma_1$ و $L_2 \subseteq \Sigma_2$ ، در این صورت الحاق $L_1 L_2$ بی‌معنی خواهد بود.

مثال: برای $\Sigma = \{a, b\}$ و $L_1 = \{b, ab\}$ و $L_2 = \{b, \lambda\}$ و $L_3 = \{b, ab\}$ به صورت زیر خواهد بود:

$$L_1 L_2 = \{ab, b, bb, abb\}$$

$$L_2 L_1 = \{bb, bab, b, ab\}$$

$$|L_1 L_2| \leq |L_1| |L_2|$$

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

مثال: با توجه به مثال قبل داریم:

$$3 = |L_1 L_2| < |L_1| |L_2| = 4$$

$$4 = |L_2 L_1| = |L_2| |L_1| = 4$$

نکته: داریم:

$$L_1 (L_2 \cup L_3) = (L_1 L_2) \cup (L_1 L_3)$$

$$L_1 (L_2 \cap L_3) \subseteq (L_1 L_2) \cap (L_1 L_3)$$

عملگر توان: عمل توان بر روی زبان L به صورت زیر تعریف می شود:

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

$$L^1 = L$$

$$L^2 = LL$$

$$L^3 = L^2 L$$

$\vdots$

$$L^n = L^{n-1} L$$

نکته: با توجه به این که زبان L شامل مجموعه ای از رشته هاست، لذا باید به صورت یک مجموعه با آن رفتار کرد، در نتیجه برای

$$L^0 = \{\lambda\} \text{ زبان } L^0 \text{ داریم:}$$

و درست نیست که بنویسیم $L^0 = \lambda$.

مثال: اگر $L = \{a^n b^n \mid n \geq 0\}$ ، در این صورت $L^2 = \{a^n b^n a^m b^m \mid n, m \geq 0\}$ توجه شود که در این جا n, m از هم

مستقل می باشند و رشته ای مثل $\underbrace{aba}_{L_1} \underbrace{abb}_{L_2}$ عضوی از زبان L^2 می باشد.

عملگرهای بستار ستاره ای و بستار مثبت: بستار ستاره ای زبان L به صورت زیر تعریف می شود:

$$L^* = L^0 \cup L^1 \cup \dots = \bigcup_{i \geq 0} L^i$$

همچنین برای بستار مثبت داریم:

$$L^+ = L^1 \cup L^2 \cup \dots = \bigcup_{i \geq 1} L^i$$

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 $L_1 = \{a\}$ و $L_2 = \{\lambda\}$ و $L_3 = \emptyset$ ، بستار ستاره ای و مثبت این سه زبان عبارتند از:

$$L_1^* = \{\lambda, a, a^2, a^3, \dots\}$$

$$L_1^+ = L_1^* L_1 = \{\lambda, a, a^2, a^3, \dots\} \{\lambda, a, a^2, a^3, \dots\} = \{a, a^2, a^3, \dots\}$$

$$L_2^* = \{\lambda\}^* = \{\lambda\}$$

$$L_3^* = \emptyset^* = \{\lambda\}$$

$$L_1^+ = L_1^* L_1 = \{\lambda\} L_1 = L_1$$

$$L_3^+ = \{\lambda\} L_3 = \emptyset$$

نکته: با توجه به مثال فوق می‌بینیم که λ در بستر ستاره‌ای همه زبان‌ها از جمله $\{\lambda\}$ و $\emptyset$ وجود دارد. عملگر متمم (complement): متمم زبان L را با $\bar{L}$ نشان می‌دهند و به صورت زیر تعریف می‌کنند:

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

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

عملگر معکوس: معکوس زبان L را با L^R نشان داده و به صورت زیر تعریف می‌کنیم.

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

مثال: معکوس زبان‌های $L_1 = \{\lambda, ab, abb\}$ و $L_2 = \{a^n b^n \mid n \geq 0\}$ به ترتیب $L_1^R = \{\lambda, ba, bba\}$ و $L_2^R = \{b^n a^n \mid n \geq 0\}$ می‌باشد.

✓ تست: فرض کنید:

$$L_1 = \{w \mid w \in \{a, b\}^*, |w| \bmod 3 \geq 1\}$$

$$L_2 = \{a, b\}$$

$$L_3 = \{aa, ab, ba, bb\}$$

$$L_4 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

(مهندسی کامپیوتر- آزاد ۸۰)

کدام یک از گزینه‌های زیر صحیح است؟

$$L_1 = L_2 * L_3 \cup L_2 * L_4 \quad (2)$$

$$L_1 = L_4 * L_2 \cup L_4 * L_3 \quad (1)$$

$$L_1 = L_4 * L_2 \quad (4)$$

$$L_1 = L_4 * L_2 \cup L_4 * L_3 \quad (3)$$

✓ 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 دقیقاً 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 به ترتیب "مضرب ۱+۳" یا "مضرب ۲+۳" 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 این کار استفاده می‌کنیم.

تعریف: یک گرامر چون G ، چهار تایی مرتبی به صورت (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 به صورت زیر هستند را در نظر بگیرید.

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

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 $G = (V, T, S, P)$ به صورت مجموعه زیر تعریف می‌شود:

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

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.

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

اگر $w \in L(G)$ ، آنگاه دنباله:

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:

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

حل:

برای به دست آوردن این زبان ابتدا چندین رشته از این زبان را به دست می‌آوریم:

$$S \Rightarrow \lambda$$

$$S \Rightarrow aSb \Rightarrow ab$$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

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

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

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

البته توجه داشته باشید که برای یک زبان می‌توان چندین گرامر نوشت که لزوماً شبیه به هم نیستند. چیزی که در این جا مهم است، درک این مکانیسم می‌باشد. در این مثال، زبان داده شده به گونه‌ای است که در رشته‌های آن ابتدا تعدادی a می‌آید و سپس تعدادی b پشت سر آن قرار می‌گیرد به طوری که تعداد b ها یکی بیش‌تر از تعداد a ها 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 به صورت الگوی $a^n b^n$ تولید می‌کند. و در آخر قاعده $A \rightarrow \lambda$ 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 $L = \{w \mid n_a(w) = n_b(w)\}$ به صورت زیر است:

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

منظور از $n_a(w)$ تعداد a های رشته w و منظور از $n_b(w)$ تعداد b های رشته w می‌باشد.

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

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

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

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

$$(۴) S \rightarrow asbb, S \rightarrow abb$$

✓ پاسخ: گزینه «۴»

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

✓ تست: اگر داشته باشیم:

$$S \rightarrow abc, S \rightarrow axbc, xb \rightarrow bx, xc \rightarrow ybcc$$

$$by \rightarrow yb, ay \rightarrow aax, ay \rightarrow aa$$

(مهندسی کامپیوتر- آزاد ۸۴)

کدام یک از گزینه‌های ذیل صحیح است؟

$$(۱) S \Rightarrow a^n x b^n c^n \quad n \geq 1$$

$$(۲) S \Rightarrow a^n b^n x c^n \quad n \geq 1$$

$$(۳) S \Rightarrow a^n b^n x c^n$$

$$(۴) \text{ هر سه مورد}$$

✓ 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 که عبارتند از فایل ورودی که بتواند رشته ورودی را دریافت کند، یک مکان ذخیره‌سازی تا بتواند خروجی‌ها را در آن ذخیره کند و همچنین واحد کنترل که می‌تواند متشکل از چندین حالت (state) متناهی درونی باشد، که در آن تغییر حالت براساس قواعد از قبل تعیین شده‌ای امکان‌پذیر است.

در یک ماشین معین (deterministic) در هر حالت یک و تنها یک حرکت توسط تابع انتقال تعریف شده است ولی در یک 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 عام‌تری از ماشین‌ها که می‌تواند در خروجی خود 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. کدام یک از گزاره‌های زیر درست است؟
(مهندسی کامپیوتر ۷۵)

$$(1) A(B \cap C) = (A.B) \cap (A.C)$$

(۲) به ازای هر زبان A ، زبان نامنظم B وجود دارد به گونه‌ای که $B \subseteq A$.

(۳) به ازای هر زبان A ، زبان نامنظم B وجود دارد به گونه‌ای که $B \supseteq A$.

$$(4) A.(B \cup C) = (A.B) \cup (A.C)$$

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

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

$$L_2 = \{a^n b^m \mid n > m \geq 0\}$$

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

$$(1) L = \{a^n b \mid n \geq 0\}$$

$$(2) L = \{a^n b a^m \mid n \geq m \geq 0\}$$

$$(3) L = \{a^n b a^{m+1} \mid n \geq m \geq 0\}$$

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

$$(1) (A \cap B)^* = (A^*) \cap (B^*)$$

$$(2) (A \cup B)^* = (A^*) \cup (B^*)$$

$$(3) (A^* \cup B^*)^* = (A^* C^*) \cup (B^* C^*)$$

۴- کدام یک 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 $B = \{\lambda, 1, 0\}$ 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)

$$(1) X = AB^*$$

$$(2) X = B^* A$$

(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 $L \cap \Sigma^+ = \emptyset$ ، 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 $L' \cap L' \Sigma^+ = \emptyset$

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)

$$S \rightarrow Sab$$

$$S \rightarrow Sba$$

$$S \rightarrow aSb$$

$$S \rightarrow bSa$$

$$S \rightarrow abS$$

$$S \rightarrow baS$$

$$S \rightarrow \varepsilon$$

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

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

$$L(G) \subset L_1 \quad (۱)$$

$$L(G) = L_1 \quad (۲)$$

$$L(G) = L_1 \cup L_2 \quad (۳)$$

$$L_2 \subset L(G) \quad (۴)$$

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

(مهندسی کامپیوتر ۸۷)

$$G: S \rightarrow ACaB$$

$$Ca \rightarrow aaC$$

$$CB \rightarrow DB$$

$$CB \rightarrow E$$

$$aD \leftrightarrow Da$$

$$AD \rightarrow AC$$

$$aE \rightarrow Ea$$

$$AE \rightarrow a$$

$$\{aa, aaaa\} \quad (۱)$$

$$\{aaa, aaaaa\} \quad (۲)$$

$$\{a, aaa, aaaaa\} \quad (۳)$$

$$\{aaaa, aaaaaa\} \quad (۴)$$

۹- گرامر G به شرح زیر مفروض است. $L(G)$ کدام است؟ w^R عبارت است از w که از آخر به اول خوانده شود. و ε نشانه

(مهندسی کامپیوتر ۸۷)

رشته‌های به طول صفر است.

$$G: S \rightarrow aA$$

$$S \rightarrow bB$$

$$S \rightarrow \varepsilon$$

$$A \rightarrow Sa$$

$$A \rightarrow \varepsilon$$

$$B \rightarrow Sb$$

$$B \rightarrow \varepsilon$$

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

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

۱۰- گرامر G را در نظر می‌گیریم و زبان آن را L می‌نامیم. رشته‌های w_1 و w_2 با تعریف زیر را نیز در نظر می‌گیریم. کدام گزاره

(مهندسی کامپیوتر ۸۸)

صحیح است؟

$$G: S \rightarrow aSD \mid bB$$

$$D \rightarrow dS \mid a$$

$$B \rightarrow bB \mid \varepsilon$$

$$w_1 = a^1 b^a b^1 a^1 d$$

$$w_2 = a^1 b^a a^1 d$$

$$w_1 \notin L, w_2 \in L \quad (۴)$$

$$w_1 \notin L, w_2 \in L \quad (۳)$$

$$w_1, w_2 \notin L \quad (۲)$$

$$w_1, w_2 \in L \quad (۱)$$

۱۱- گرامر وابسته به متن G مفروض است:

$$\begin{aligned} G : S &\rightarrow S_1 B \\ S_1 &\rightarrow a S_1 b \\ b B &\rightarrow b b b B \\ a S_1 b &\rightarrow a a \\ B &\rightarrow \varepsilon \end{aligned}$$

(مهندسی کامپیوتر ۸۹)

زبان گرامر G کدام است؟

$$\begin{aligned} (۱) \quad \{a^{n+1}b^{n+k} \mid n \geq 1, k \geq 0\} \\ (۲) \quad \{a^n b^k \mid n \geq 2, k \geq 0\} \\ (۳) \quad \{a^n b^{n+2k} \mid n \geq 2, k \geq 0\} \\ (۴) \quad \{a^{n+1}b^{n+2k-1} \mid n \geq 1, k \geq 0\} \end{aligned}$$

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

$$\begin{aligned} S &\rightarrow ac BdeA|BAB \\ B &\rightarrow a Sb|ae|\varepsilon \\ A &\rightarrow a Ab|b|\varepsilon \\ w_1 &= acaaca \text{ } bbdebbdebb \\ w_2 &= acaaca ae bdebbdeabb \end{aligned}$$

(مهندسی کامپیوتر ۹۰)

کدام گزینه صحیح است؟

$$\begin{aligned} (۱) \quad w_1 \in L(G), \quad w_2 \notin L(G) \\ (۲) \quad w_1 \notin L(G), \quad w_2 \in L(G) \\ (۳) \quad w_1 \in L(G), \quad w_2 \in L(G) \\ (۴) \quad w_1 \notin L(G), \quad w_2 \notin L(G) \end{aligned}$$

۱۳- مجموعه A را «شمارش‌پذیر» می‌نامیم اگر A متناهی یا در تناظر یک به یک با مجموعه اعداد طبیعی باشد. در غیر این صورت A را «ناشمار» 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 در نظر گرفت.

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

$$(۱) \text{ ب، ج} \quad (۲) \text{ الف، ب، ج} \quad (۳) \text{ الف، ج، د} \quad (۴) \text{ ب، ج، د}$$

۱۴- زبان L را به شکل زیر تعریف می‌کنیم:

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

(مهندسی کامپیوتر ۹۱)

کدام گرامر زیر، تولیدکننده تمامی رشته‌های زبان L خواهد بود؟

$$\begin{aligned} (۱) \quad S &\rightarrow aS|aSb|\varepsilon \\ (۲) \quad S &\rightarrow aS|aSb|a \\ (۳) \quad S &\rightarrow aS|Sa||aSb|ab\varepsilon \\ (۴) \quad S &\rightarrow aS|aSb|\varepsilon \end{aligned}$$

پاسفنامه 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 $w' = w'_1/w'_2 = a^n ba$ در نتیجه 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»

$X = A \cup XB$ را به صورت $X \rightarrow A \mid XB$ در نظر می‌گیریم. با اشتقاق از X داریم:

$$x \mapsto xB \mapsto xBB^* \mapsto xB^n \mapsto AB^n$$

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:

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

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

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

۹- گزینه «۲»

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

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

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 حالی که در هیچ‌یک از زبان‌های گزینه‌های ۱ و ۳، رشته aa تولید نمی‌شود، بنابراین این گزینه‌ها نادرست می‌باشند.

$$S \rightarrow S_1 B \rightarrow aS_1 bB \rightarrow aa\varepsilon = aa$$

و با توجه به قاعده $S_1 \rightarrow aS_1 b$ متوجه می‌شویم که ارتباطی بین تعداد aa ها و bb وجود دارد بنابراین گزینه ۲ نیز رد می‌شود.

۱۲- گزینه «۱»

رشته w_1 را می‌توان توسط اشتقاق زیر پذیرفت:

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

ولی از آنجایی که در گرامر داده شده یا قبل از e, d, 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 Σ^* مجموعه‌ای نامتناهی و ناشماراست. از آنجایی که 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 $I.e.$ تولید کند که بازگشتی نباشد.

۱۴- گزینه «۱»

گزینه‌های ۳، ۴، ϵ را تولید نمی‌کنند، درحالی که زبان L شامل ϵ می‌باشد.
گزینه ۲، رشته aba را تولید نمی‌کند، درحالی که زبان L شامل این رشته می‌باشد.

سوالات چهارگزینه‌ای آزاد فصل اول

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

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

۲- اگر A^* مجموعه تمام رشته‌های تعریف شده 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)

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

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

پاسفنامه سوالات چهارگزینه‌ای آزاد فصل اول

۱- گزینه «۱»

$\lambda - \lambda = \emptyset$, $\lambda^* = \emptyset^* = \{\lambda\}$

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 حتماً 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 DFA و NFA
- ♦ 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 اتوماتای 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 Acceptor)

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 $M = (Q, \Sigma, \delta, q_0, F)$ می‌باشد که در آن:

Q : مجموعه متناهی از حالات ماشین (internal state)

Σ : مجموعه الفبای ورودی (input alphabet)

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

$q_0 \in Q$: 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_0 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 $\delta(q_0, a) = q_1$ تعریف شده باشد آنگاه در حالت q_0 اگر نماد ورودی a باشد به حالت

q_1 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 $\Sigma = \{0, 1\}$ مجموعه الفبای

ورودی و δ تابع انتقال یک ماشین DFA با حالات q_0 و q_1 با حالت شروع q_0 و حالت نهایی q_1 به‌صورت زیر باشد:

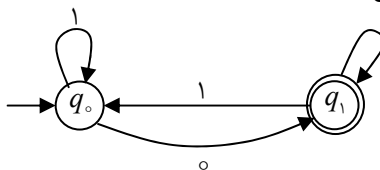
$$\delta(q_0, 1) = q_0$$

$$\delta(q_0, 0) = q_1$$

$$\delta(q_1, 1) = q_0$$

$$\delta(q_1, 0) = q_1$$

آنگاه این ماشین را می‌توان به صورت 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 به صورت $\delta^*: Q \times \Sigma^* \rightarrow Q$ تعریف می‌شود. در این تابع، پارامتر دوم به جای یک حرف، یک رشته می‌باشد و مقدار آن برابر حالتی است که پس از خواندن آن رشته از حالت شروع به آن می‌رسیم. به عنوان مثال اگر داشته باشیم.

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

آنگاه با تابع انتقال تعمیم‌یافته می‌توان نوشت:

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

می‌توان به گونه‌ای رسمی تر تابع انتقال تعمیم‌یافته را به صورت بازگشتی زیر تعریف کرد.

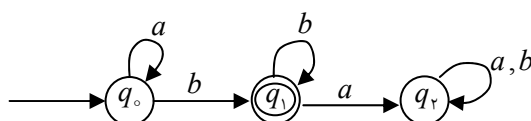
$$\begin{cases} \delta^*(q, \lambda) = q \\ \delta^*(q, aw) = \delta^*(\delta(q, a), w) \end{cases}$$

بررسی درستی این روابط با استقراء ریاضی به سادگی امکان پذیر است.

زبان پذیرفته شده توسط ماشین منتهای معین: مجموعه همه رشته‌هایی روی الفبای Σ است که توسط ماشین M پذیرفته می‌شود. به بیان دقیق تر داریم:

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

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

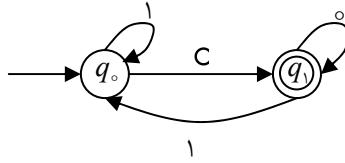


راه دیگر نشان دادن تابع انتقال استفاده از نمایش جدولی تابع می‌باشد. به عنوان مثال تابع انتقال دیاگرام فوق را می‌توان به صورت جدول زیر نشان داد.

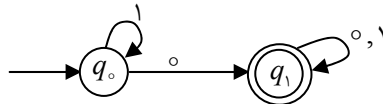
Σ	a	b
S		
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_2

مثال: برای زبان‌های زیر یک dfa رسم کنید. در این مثال فرض بر این است که $\Sigma = \{0, 1\}$

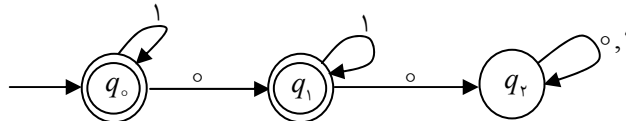
$$L_1 = \{w \in \Sigma^* \mid w \text{ به صفر ختم شود}\} \quad (۱)$$



$$L_2 = \{w \in \Sigma^* \mid w \text{ شامل حداقل یک صفر باشد}\} \quad (۲)$$

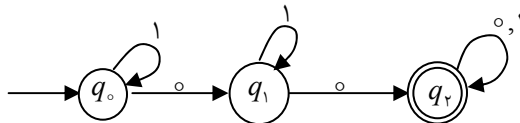


$$L_3 = \{w \in \Sigma^* \mid w \text{ شامل حداکثر یک صفر باشد}\} \quad (۳)$$

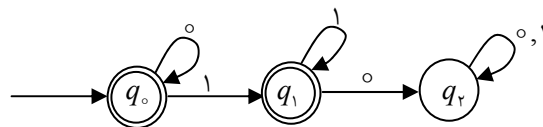


نکته: هرگاه حالت شروع، نهایی هم باشد، در این صورت رشته λ نیز پذیرفته خواهد شد. در این مثال q_2 حالت تله است.

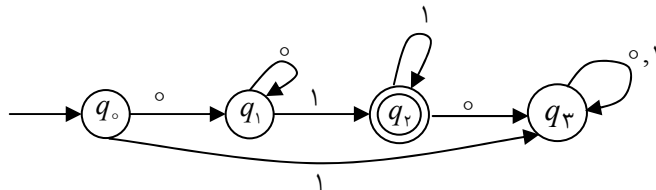
$$L_4 = \{w \in \Sigma^* \mid n_0(w) \geq 2\} \quad (۴)$$



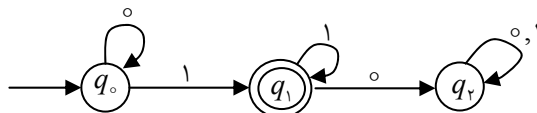
$$L_5 = \{0^m 1^n \mid n, m \geq 0\} \quad (۵)$$



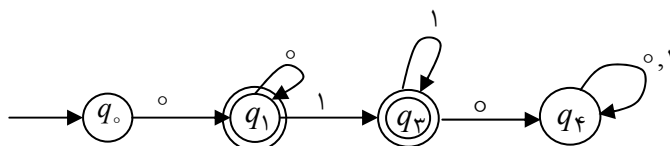
$$L_6 = \{0^m 1^n \mid n, m > 0\} \quad (۶)$$



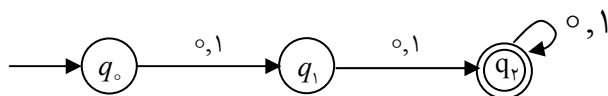
$$L_7 = \{0^m 1^n \mid m \geq 0, n > 0\} \quad (۷)$$



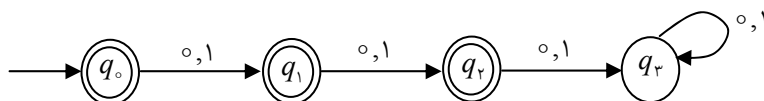
$$L_8 = \{0^m 1^n \mid m > 0, n \geq 0\} \quad (۸)$$



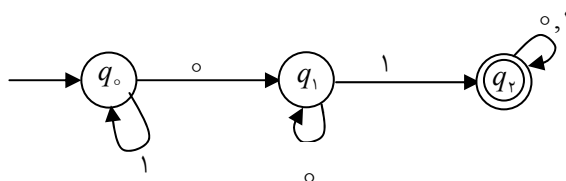
$$L_q = \{w \in \Sigma^* \mid |w| \geq 2\} \quad (۹)$$



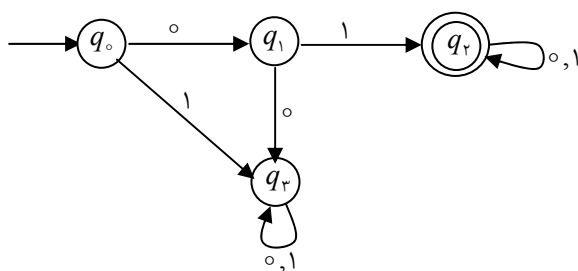
$$L_{10} = \{w \in \Sigma^* \mid |w| \leq 2\} \quad (۱۰)$$



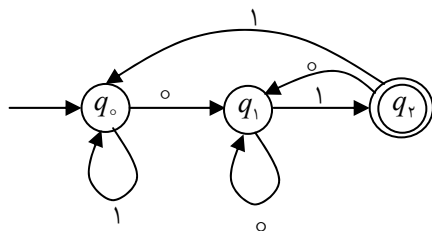
$$L_{11} = \{w \in \Sigma^* \mid w \text{ شامل زیررشته } 01 \text{ باشد}\} \quad (۱۱)$$



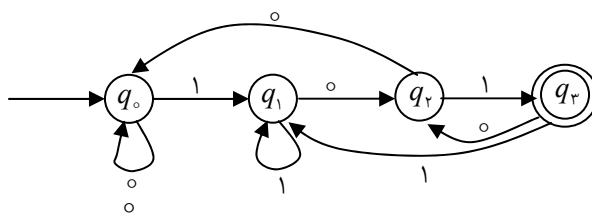
$$L_{12} = \{w \in \Sigma^* \mid w \text{ با } 01 \text{ شروع شود}\} \quad (۱۲)$$



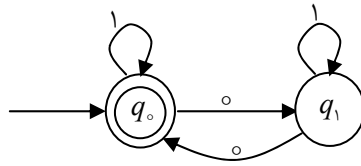
$$L_{13} = \{w \in \Sigma^* \mid w \text{ به } 01 \text{ ختم شود}\} \quad (۱۳)$$



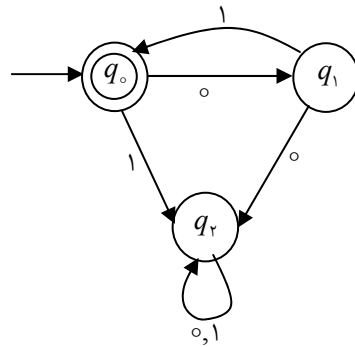
$$L_{14} = \{w \in \Sigma^* \mid w \text{ به } 101 \text{ ختم شود}\} \quad (۱۴)$$



$$L_{15} = \{w \in \Sigma^* \mid n_o(w) = rk, k \geq 0\} \quad (15)$$

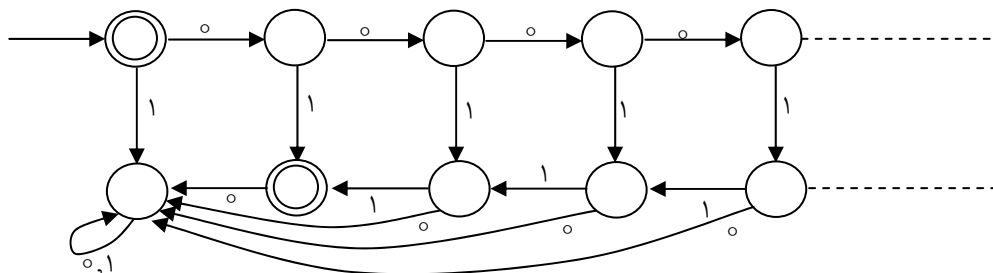


$$L_{16} = \{(o1)^n \mid n \geq 0\} \quad (16)$$



$$L_{17} = \{o^n 1^n \mid n \geq 0\} \quad (17)$$

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_{17} ارائه کرد.



در پاسخ باید بگوییم، همان‌طور که لفظ «اتوماتای متناهی» برمی‌آید، اتوماتا باید متناهی باشد، در حالی که 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.

$$L_{18} = \{o^4 o^4 o^4\} \quad (18)$$

به دلیل این که 400 یک عدد ثابت است و برخلاف n در زبان L_{17} متغیر نمی‌باشد، لذا می‌توان dfa این زبان را رسم کرد.
زبان‌های منظم: خانواده زبان‌های پذیرفته‌شده به وسیله DFA، خانواده زبان‌های منظم نامیده می‌شوند. تمامی زبان‌های ارائه شده به جز L_{17} در مثال قبل منظم بودند. به بیان دیگر داریم:
زبان L منظم است اگر و تنها اگر پذیرنده متناهی معینی چون M وجود داشته باشد که $L(M)=L$.