

نظریه زبان‌ها و ماشین‌ها

مؤلف: فرشید شیرافکن

دانشجوی دکتری بیوانفورماتیک دانشگاه تهران



ناشر: سازمان علمی آموزش فرادرس

بزرگترین پلتفرم آموزش آنلاین ایران

وب: www.faradars.org

تقدیم به:

روح پاک پدرم

- فرشید شیرافکن



سخن ناشر

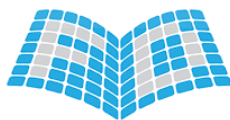
در عین تمام نقدهای وارد شده به کنکور، هنوز راه حلی عملی که در جمیع جوانب، بهتر از سبک کوتاه و چند گزینه‌ای سوالات باشد؛ ارائه نشده است. همین موضوع، کنکور را به ویژه کنکور کارشناسی ارشد به عنوان یک آزمون متمرکز و سراسری، از اهمیت دو چندانی برخوردار می‌کند.

یکی از آسیب‌های همراه با این آزمون سراسری این است که فضای رقابتی آن با ایجاد موسسات گوناگون، به سرعت از فضای یک رقابت علمی تبدیل به فضای رقابت اقتصادی می‌شود؛ به گونه‌ای که هزینه سرسام آور کلاس‌ها، دوره‌ها و منابع مرتبط با آزمون، از عهده بسیاری از دانشجویان خارج می‌شود. دانشجویانی که در عین استعداد تحصیلی بالا، در میدان رقابت مالی تحصیلی، در عین تمام شایستگی‌های خود، قدرت ادامه مسیر را از دست می‌دهند یا به نتیجه‌ای که در فضای مساوی مالی برای همه باید به آن می‌رسیدند، دست نمی‌یابند.

یکی از اهداف و آرمان‌های فرادرس به عنوان بزرگترین پروژه آموزش دانشگاهی اجرا شده بر بستر وب کشور، ایجاد دسترسی همگانی و یکسان به آموزش و دانش؛ مستقل از جغرافیا، زمان و سطح مالی دانشجویان بوده است. سیاست کاری فرادرس در راستای این آرمان، انتشار آموزش‌های ویدئویی تخصصی و دانشگاهی رایگان و یا بسیار کم هزینه، با تدریس مجرب‌ترین اساتید داخل و خارج کشور بوده است.

ما با انتشار رایگان این کتاب (به همراه نزدیک به ده کتاب رایگان دیگر) یکی از گام‌های دیگر خود را در راستای آرمان فرادرس برداشتیم. کتاب حاضر که حاصل نزدیک به دو دهه تدریس و پژوهش و تألیف مؤلف و مدرس فرادرس می‌باشد؛ در عین هزینه‌های بالای تألیف و آماده‌سازی، به جای انتشار و فروش؛ با تأمین مالی و سرمایه‌گذاری فرادرس به عنوان ناشر، به صورت کاملاً رایگان منتشر می‌شود. ما در گام‌های بعدی نیز تلاش خواهیم کرد تا هر جا بتوانیم، حتی شده یک کتاب مرجع دیگر و بیشتر را با پرداخت هزینه، آزادسازی کرده و به صورت رایگان منتشر کنیم.

مؤلفین و ناشرینی که تمایل به واگذاری حق انتشار کتاب خود به فرادرس را دارند، می‌توانند با ایمیل books@faradars.org مکاتبه نمایند. ما این کتاب‌ها را با پرداخت هزینه تألیف به مدرس و ناشر، به صورت رایگان منتشر خواهیم کرد تا همه دانشجویان مستقل از سطح مالی، به منابع مفید آزمون دسترسی داشته باشند. همچنین اگر ایده و نظری در خصوص کتاب‌های رایگان فرادرس داشته باشید، خوشحال می‌شویم که آن را با ایمیل books@faradars.org مطرح نمایید.



فرادرس

فرانتر از یک کلاس درس
www.faradars.org

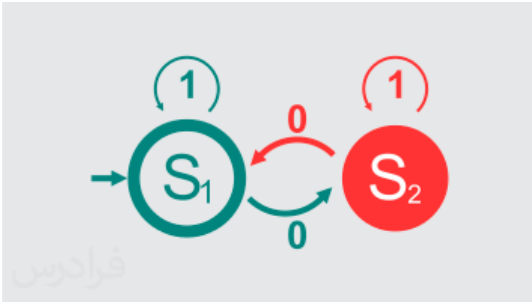
سازمان علمی آموزش فرادرس

بزرگترین پلتفرم آموزش آنلاین ایران

وب: www.faradars.org

منبع مطالعاتی تکمیلی مرتبط با این کتاب

آموزش ویدئویی نظریه زبان‌ها و ماشین‌ها



در این درس با سه موضوع "زبان، گرامر و ماشین" آشنا می‌شوید. این درس پیش‌نیاز درس طراحی کامپایلر است. با یادگیری زبان‌ها و گرامرها می‌توانید نحوه کار کامپایلر و همچنین طراحی زبان‌های برنامه‌سازی را متوجه شوید. یادگیری این درس بدون مدرس کار ساده‌ای نمی‌باشد و ما در این آموزش تجربه حداقل پانزده سال تدریس این درس را در اختیار شما گذاشته‌ایم. به امید اینکه دعای خیری برای ما شود.

مدرس: مهندس فرشید شیرافکن

مدت زمان: ۹ ساعت

faradars.org/fvsft110

جهت مشاهده آموزش ویدئویی این آموزش [کلیک](#) کنید.

درباره مدرس

مهندس فرشید شیرافکن کارشناس ارشد مهندسی کامپیوتر گرایش نرم‌افزار است و در حال حاضر دانشجوی دکتری بیوانفورماتیک دانشگاه تهران هستند. ایشان از مدرسین نمونه در زمینه ارائه و آموزش دروس دانشگاهی انتخاب شده‌اند. ایشان مشاور کنکور هستند و بیش از ۳۰ کتاب در زمینه کنکور رشته کامپیوتر تألیف نموده‌اند. ایشان در حال حاضر به عنوان یکی از برترین مدرسین فرادرس از جهت کمیت و کیفیت دروس ارائه شده، نزدیک به ۲۰ عنوان درسی را در قالب آموزش ویدئویی از طریق فرادرس منتشر کرده‌اند. این مجموعه دروس تا کنون مورد استفاده ده‌ها هزار دانشجوی سراسر کشور قرار گرفته‌اند.



مشاهده همه آموزش‌های تدریسی و تألیفی توسط مؤلف کتاب [کلیک](#) کنید.

Email: fshirafkan51@gmail.com

شماره همراه مدرس: ۰۹۱۲۱۹۷۲۰۲۸

کتاب رایگان دیگر از این مجموعه آموزشی

۱. آموزش برنامه‌نویسی سی پلاس پلاس - [کلیک کنید \(+\)](#)
۲. آموزش شیء‌گرایی در سی پلاس پلاس - [کلیک کنید \(+\)](#)
۳. آموزش پایگاه داده‌ها - [کلیک کنید \(+\)](#)
۴. آموزش ساختمان داده‌ها - [کلیک کنید \(+\)](#)
۵. آموزش سیستم عامل - [کلیک کنید \(+\)](#)
۶. آموزش ذخیره و بازیابی اطلاعات - [کلیک کنید \(+\)](#)

برای دانلود رایگان این مجموعه کتاب، به لینک زیر مراجعه کنید:

<http://faradars.org/computer-engineering-exam>



دسته بندی موضوعی آموزش های فرادرس، در ادامه آمده است:

 <p>مهندسی برق الکترونیک و رباتیک</p>	 <p>هوش مصنوعی</p>	 <p>آموزش‌های دانشگاهی و تخصصی</p>	 <p>برنامه‌نویسی</p>
<p>مهندسی برق الکترونیک و رباتیک - کلیک (+)</p>	<p>هوش مصنوعی - کلیک (+)</p>	<p>آموزش‌های دانشگاهی و تخصصی - کلیک (+)</p>	<p>برنامه‌نویسی - کلیک (+)</p>
 <p>مهارت‌های دانشگاهی</p>	 <p>مباحث مشترک</p>	 <p>دروس دانشگاهی</p>	 <p>مهندسی نرم‌افزار</p>
<p>مهارت‌های دانشگاهی - کلیک (+)</p>	<p>مباحث مشترک - کلیک (+)</p>	<p>دروس دانشگاهی - کلیک (+)</p>	<p>مهندسی نرم‌افزار - کلیک (+)</p>
 <p>آموزش‌های عمومی</p>	 <p>طراحی و توسعه وب</p>	 <p>نرم‌افزارهای عمومی</p>	 <p>نرم‌افزارهای تخصصی</p>
<p>آموزش‌های عمومی - کلیک (+)</p>	<p>طراحی و توسعه وب - کلیک (+)</p>	<p>نرم‌افزارهای عمومی - کلیک (+)</p>	<p>نرم‌افزارهای تخصصی - کلیک (+)</p>

فهرست مطالب

فصل ۱ : عبارت منظم- زبان منظم

عبارت منظم

زبان

اجتماع و اشتراک

اتصال

معکوس

مکمل

بستار

هم ریختی

تقسیم راست

زبان منظم

بسته بودن زبان‌های منظم

لم تزریق

فصل ۲ : گرامر-گرامر منظم

گرامر

انواع گرامر

زبان تولید شده توسط گرامر

گرامر منظم

فصل ۳ : اتوماتای متناهی (DFA , NFA)

انواع ماشین

ماشین‌های متناهی

پذیرنده متناهی معین (dfa)

زبان‌ها و dfa

حالت دام (تله)

مکمل dfa

پذیرنده متناهی نامعین (nfa)

هم ارزی dfa و nfa

ارتباط گرامر منظم با ماشین متناهی

کاهش تعداد حالات در ماشین‌های متناهی

نحوه تشخیص منظم بودن یک زبان

فصل ۴ : زبان و گرامر مستقل از متن

گرامر مستقل از متن

گرامر ساده

بسته بودن زبان‌های مستقل از متن

لم تزریق برای زبان‌های مستقل از متن

لم تزریق برای زبان‌های خطی

فصل ۵ : ابهام - ساده سازی گرامر - فرمهای نرمال

ابهام در گرامر و زبان

ساده سازی گرامرهای مستقل از متن

حذف متغیرها و قوانین بی فایده

حذف قوانین

حذف قوانین واحد

فرم‌های نرمال گرامر مستقل از متن

فرم نرمال چامسکی

فرم نرمال گریباخ

فصل ۶ : اتوماتای پشته ای (DPDA, NPDA)

اتوماتای پشته ای نامعین

تابع انتقال

پیگر بندی لحظه ای

اتوماتای پشته ای معین

تشخیص مستقل از متن بودن یک زبان

زبان مستقل از متن معین

ساخت اتوماتای پشته ای با استفاده از گرامر در فرم گریباخ

فصل ۷ : ماشین های تورینگ (TM)

ماشین تورینگ استاندارد

ماشین تورینگ در نقش پذیرنده زبان

ماشین تورینگ به عنوان مترجم

مدل های دیگر ماشین تورینگ

سکون دار

با نوار نیمه نامتناهی

آف لاین

با حافظه پیچیده تر

چند نواره

چند بعدی

نامعین

آتاماتای کراندار خطی (LBA)

فصل ۸ : زبان‌های بازگشتی-گرامر بدون محدودیت و حساس به متن

زبان‌های بازگشتی و بازگشتی شمارش پذیر

گرامر بدون محدودیت

گرامر حساس به متن

ارتباط بین زبان‌ها، گرامرها و ماشین‌ها

سلسله مراتب چامسکی

بررسی بسته بودن زبان‌ها تحت عملگرها

فصل ۹ : تصمیم پذیری – کاهش پذیری

زبان‌های تصمیم نا پذیر

زبان‌های تصمیم پذیر

تصمیم پذیری در زبان‌های منظم

برشمارنده-

کاهش پذیری

فصل ۱:

عبارت منظم - زبان منظم

در این کتاب، سه مفهوم اصلی یعنی زبان، گرامر (ابزار تولید زبان) و ماشین (ابزار پذیرش زبان) بررسی خواهند شد. مطالبی که بررسی می‌شوند عبارتند از:

۱- زبان و گرامر منظم و ماشین پذیرنده رشته‌های زبان منظم یعنی ماشین متناهی (FA)

۲- زبان و گرامر مستقل از متن و ماشین پذیرنده رشته‌های زبان مستقل از متن یعنی ماشین پشته‌ای (PDA)

۳- زبان و گرامر حساس به متن و ماشین تورینگ (TM) و ...

عبارت منظم

قبل از ورود به بحث زبان، عبارت منظم را بررسی می‌کنیم. عبارت منظم، ترکیبی از سمبل‌ها (از قبیل الفبا، پرانتز، عملگر بستار، عملگر الحاق) می‌باشد. عبارت منظم با انجام متوالی برخی قوانین بازگشتی روی اجزاء پایه‌ای ایجاد می‌شود.

به طور نمونه a^+b یک عبارت منظم در الفبای $\Sigma = \{a, b\}$ می‌باشد، که معرف رشته‌هایی است که با یک یا چند حرف a شروع شده و به b ختم می‌شوند. مانند ab ، aab و $aaab$.

به عنوان مثالی دیگر عبارت منظم ab^+ ، معرف رشته‌هایی است که با یک حرف a شروع شده و به یک یا چند b ختم می‌شوند. مانند ab ، abb و abb . اگر به جای بستار مثبت از بستار ستاره استفاده شده بود، یعنی عبارت منظم به صورت ab^* باشد، آنگاه، معرف رشته‌هایی است که با یک حرف a شروع شده و به صفر یا یک یا چند b ختم می‌شوند. یعنی از b می‌توان استفاده نکرد و رشته a را تولید کرد که توسط ab^+ قابل تولید نبود.

تعریف

Σ را الفبای مفروض در نظر می‌گیریم. آنگاه:

۱- \emptyset ، λ و $a \in \Sigma$ همگی عبارات منظم هستند. این عبارات را عبارات منظم پایه می‌خوانیم.

۲- اگر r_1 و r_2 عبارات منظمی باشند، آنگاه $r_1 + r_2$ ، $r_1.r_2$ ، r_1^* و (r_1) نیز عبارات منظم خواهند بود.

۳- یک رشته فقط و فقط در صورتی عبارت منظم است که با بکارگیری تعداد محدودی از قوانین بند (۲)، از عبارات منظم پایه بدست آیند.

چنانچه Σ یک الفبا باشد، آنگاه از Σ^+ برای نمایش مجموعه رشته‌های بدست آمده از الحاق چند سمبل دیگر و از Σ^* برای نمایش مجموعه رشته‌های بدست آمده از الحاق صفر یا بیشتر سمبل دیگر استفاده می‌کنیم. مجموعه Σ^* همواره حاوی λ خواهد بود. به عبارتی: $\Sigma^* = \Sigma^+ + \{\lambda\}$.

همواره Σ^+ و Σ^* نامتناهی هستند.

مثال

اگر الفبا شامل یک حرف a باشد، یعنی $\Sigma = \{a\}$ ، آنگاه:

$$\Sigma^+ = \{a, aa, aaa, \dots\} \quad \Sigma^* = \{\lambda, a, aa, aaa, \dots\}$$

چند نکته:

۱- در عبارت منظم اولویت ستاره از همه بیشتر است، سپس اولویت اتصال بیشتر بوده، و در نهایت اجتماع قرار دارد، مگر اینکه از پرانتز برای تغییر اولویت استفاده شود.

۲- تعداد سمبل‌های موجود در رشته را طول رشته می‌گویند. طول رشته w با $|w|$ نشان داده می‌شود.

۳- رشته تهی (λ)، رشته‌ای است که هیچ سمبولی ندارد. برای هر رشته w داریم: $\lambda.w = w.\lambda = w$

۴- طول رشته تهی که با λ نشان داده می‌شود، برابر صفر است. ($|\lambda| = 0$)

۵- اگر Γ یک عبارت منظم باشد، آنگاه: $r.\lambda = r$

۶- اگر Γ یک عبارت منظم باشد، آنگاه: $r + \phi = r$

مثال

آیا رشته‌های aa و $abbba$ و $abab$ را می‌توان از عبارت منظم ab^*a تولید کرد؟

حل: aa : بله - به علت استفاده از بستار ستاره برای b ، می‌توان از b استفاده نکرد.

$abbba$: بله - به علت استفاده از بستار ستاره برای b ، می‌توان چند بار از b استفاده کرد.

$abab$: خیر. ■

مثال

آیا رشته‌های aa و aba و $abbbba$ را می‌توان از عبارت منظم $a(bb)^*a$ تولید کرد؟

حل: aa : بله - از bb داخل پرانتز، استفاده نمی‌کنیم.

aba : خیر - تعداد b های تولید شده توسط عبارت منظم، باید زوج باشد.

$abbbba$: بله - از bb داخل پرانتز، دو مرتبه استفاده می‌کنیم. ■

مثال

رشته‌های تولید شده با حداکثر طول ۴، توسط عبارت منظم a^+b و a^*b عبارتند از:

$$a^+b = \{ab, aab, aaab\}, \quad a^*b = \{b, ab, aab, aaab\}$$

■

مثال

رشته‌های تولید شده با طول ۳ توسط عبارت منظم $(a+b)^+$ عبارتند از:

$$(a+b)^+ = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

■

مثال

آیا رشته $aaaabbaa$ از عبارت منظم $(ab^*aa)^+$ قابل تولید است؟

حل: بله - به علت استفاده از بستار مثبت برای کل عبارت داخل پرانتز، می‌توان چند بار از این عبارت استفاده کرد. در بار اول

aaa و در بار دوم $abbaa$ را تولید می‌کنیم. ■

مثال

نحوه تولید رشته‌های $abaa$ و $aaabaab$ و $aaabababba$ به کمک عبارت منظم $(ab + ab^*a)^+$ را مشخص کنید.

حل: عبارت داده شده دارای دو قسمت ab و ab^*a می‌باشد که به علت استفاده از بستار مثبت می‌توان چندین بار از آن استفاده کرد.

$abaa$: انتخاب ab از قسمت اول عبارت منظم و aa از قسمت دوم.

$aaabaab$: انتخاب aa از قسمت دوم، سپس aba از قسمت دوم و ab از قسمت اول.

$aaabababba$: انتخاب aa از قسمت دوم، سپس $abab$ با دو بار استفاده از قسمت اول و در نهایت $abba$ از قسمت

دوم. ■

روابط مقابل برقرارند: $(a^+)^* = (a^*)^+ = (a^*)^* = a^*$

روابط زیر برقرارند:

$$\phi^+ = \phi, \phi^* = \{\lambda\}, \lambda^* = \lambda, \lambda^* - \phi^* = \phi, \lambda - \phi^* = \phi, \lambda^* \cdot \phi^* = \lambda$$

به جای $(\alpha + \beta)^*$ هر یک از عبارت‌های زیر را می‌توان قرار داد:

$$(\alpha^* + \beta^*)^*, (\alpha + \beta^*)^*, (\alpha^* + \beta^*)^*, \alpha^*(\alpha + \beta)^*, \alpha^*(\alpha + \beta)^* \beta^*$$

$$(\alpha^* \beta^*)^*, \beta^*(\alpha \beta^*)^*, \alpha^*(\beta \alpha^*)^*, (\beta^* \alpha^*)^*$$

مثال

معادل عبارت $(a^* + b)^*(a^* + b^*)^*$ را بدست آورید.

حل: می‌توان به جای $(a^* + b)^*$ و $(a^* + b^*)^*$ از $(a + b)^*$ استفاده کرد:

$$(a + b)^*(a + b)^* = (a + b)^*$$

مثال

معادل عبارت $((a^* b^*)^*(b^* a^*)^*)^*$ را بدست آورید.

حل: می‌توان به جای $(a^* b^*)^*$ و $(b^* a^*)^*$ از $(a + b)^*$ استفاده کرد:

$$((a + b)^*(a + b)^*)^* = ((a + b)^*)^* = (a + b)^*$$

مثال

آیا رابطه $(ab)^* a = a(ba)^*$ برقرار است؟

حل: بله - تمام رشته‌های قابل تولید توسط هر کدام، به کمک دیگری نیز قابل تولید است. ■

مثال

فرض کنید a و b عبارت‌های منظم باشند. آیا رابطه زیر برقرار است؟

$$(a+b)^* a(a+b)^* = b^* a(a+b)^* = (a+b)^* ab^*$$

حل: بله - تمام رشته‌های قابل تولید توسط هر کدام، به کمک دیگری نیز قابل تولید است. ■

مثال

آیا رابطه $(a+b)^* = (a^*b + ab^*)^*$ برقرار است؟

حل: بله - تمام رشته‌های قابل تولید توسط هر کدام، به کمک دیگری نیز قابل تولید است. ■

اگر Σ شامل n عنصر باشد، آنگاه Σ^* دارای n^k رشته به طول k است. ✍

معکوس عبارت منظم

برای پیدا کردن معکوس (متمم) یک عبارت منظم، از روابط زیر استفاده می‌کنیم:

$$(\alpha^*)^R = (\alpha^R)^* \quad (\alpha + \beta)^R = \alpha^R + \beta^R \quad (\alpha\beta)^R = \beta^R \alpha^R$$

مثال

معکوس عبارت منظم $ab(c+de)$ را مشخص کنید.

$$\text{حل: } = (ed+c).ba = (c^R + (de)^R).(b.a) = (c+de)^R.(ab)^R [(ab).(c+de)]^R$$

تذکر: در واقع برای پیدا کردن معکوس یک عبارت منظم، می‌توان عبارت را از انتها به ابتدا نوشت. ■

مثال

معکوس $(a^* + b)ba^*$ را مشخص کنید.

حل: کافی است عبارت را معکوس نوشت: $(a^* + b)ba^* \Rightarrow a^*b(b + a^*)$ ■

زبان

هر مجموعه از رشته‌های روی یک الفبای Σ را یک زبان می‌گویند. عملیات قابل انجام بر روی زبان‌ها عبارتند از: اجتماع، اشتراک، تفاضل، اتصال، معکوس، مکمل (متمم)، همریختی و تقسیم راست.

مثال

مجموعه $L = \{a^n b^n : n \geq 0\}$ ، یک زبان بر روی الفبا $\Sigma = \{a, b\}$ است که شامل رشته‌هایی با تعداد برابر a و b می‌باشد. مانند: $\{\lambda, ab, aabb, aaabbb, \dots\}$ ■

اجتماع و اشتراک

از آنجا که زبان‌ها، مجموعه هستند، اجتماع و اشتراک دو زبان، به راحتی قابل تعریف است.

مثال

اجتماع و اشتراک $L_1 = \{a^n b^m : n \geq 0, m \geq 0\}$ و $L_2 = \{a^n b^n : n \geq 0\}$ را مشخص کنید.

حل: چند جمله از هر زبان را مشخص می‌کنیم:

$$L_1 = \{\lambda, ab, a^5 b^2, a^7 b^7, \dots\}, \quad L_2 = \{\lambda, ab, a^2 b^2, a^5 b^5, a^7 b^7, \dots\}$$

واضح است که زبان L_2 زیر مجموعه L_1 است. در نتیجه اشتراک آنها L_2 و اجتماع آنها L_1 است. ■

مثال

اشتراک و اجتماع دو زبان $L_1 = \{a^n b^k : n \neq k\}$ و $L_2 = \{a^n b^k : n \geq 0, k \geq 0\}$ را مشخص کنید.

حل: چند جمله از هر زبان را مشخص می‌کنیم:

$$L_1 = \{\lambda, a^5 b^2, a^7 b^4, \dots\}$$

$$L_2 = \{\lambda, ab, a^7 b^4, a^5 b^5, a^4 b^4, a^5 b^2, \dots\}$$

واضح است که زبان L_1 زیر مجموعه L_2 است ($L_1 \subseteq L_2$)، پس حاصل $L_1 \cap L_2$ همان L_1 و حاصل $L_1 \cup L_2$ همان L_2 است. ■

مثال

اشتراک L_1 و L_2 را مشخص کنید.

$$L_1 = \{a^n b^n c^m : n \geq 0, m \geq 0\}$$

$$L_2 = \{a^n b^m c^m : n \geq 0, m \geq 0\}$$

حل: چند جمله از هر زبان را مشخص می‌کنیم:

$$L_1 = \{\lambda, abc, abc^5, a^6 b^6 c^2, a^4 b^4 c^4, \dots\}$$

$$L_2 = \{\lambda, abc, a^4 b^5 c^5, a^4 b^4 c^4, a^3 b^6 c^6, \dots\}$$

بنابراین اشتراک این دو زبان، شامل جمله‌هایی است که در آنها تعداد a و b برابر است، یعنی: $L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\}$

مثال

اجتماع دو زبان زیر را مشخص کنید.

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

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

حل: اجتماع دو زبان برابر است با:

$$L_1 \cup L_2 = \{a^n b^m c^n : n \geq 0, m \geq 0\}$$

مثال

اشتراک سه زبان زیر را مشخص کنید.

$$L_1 = \{a^n b^m a^n b^p : n, m, p \geq 0\}$$

$$L_2 = \{a^n b^n a^m b^p : n, m, p \geq 0\}$$

$$L_3 = \{a^n b^m a^p b^n : n, m, p \geq 0\}$$

حل: اشتراک این سه زبان برابر است با: $L_1 \cap L_2 \cap L_3 = \{a^n b^n a^n b^n : n \geq 0\}$

■

مثال

اشتراک $L_1 = \{a^n b^m : n \geq 0, m \geq 0\}$ و $L_2 = \{b^n a^n : n \geq 0\}$ را مشخص کنید.

حل: $L_1 \cap L_2 = \{a^n : n \geq 0\} \cup \{b^m : m \geq 0\}$

اتصال

اتصال (الحاق) دو زبان به صورت زیر تعریف می‌شود:


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

مثال

تعداد اعضای اتصال $L_1 = \{10, 1\}$ و $L_2 = \{01, 011, 11\}$ را تعیین کنید.

$$L_1.L_2 = \{1001, 10011, 1011, 10111, 1111\}$$

بنابراین $L_1.L_2$ دارای پنج عضو می‌باشد. (1011 که دو بار تکرار شده را یکبار شمارش می‌کنیم.)

L^n به معنای اتصال L به تعداد n بار با خودش می‌باشد و $L^0 = \{\lambda\}$ 

مثال

اگر $L = \{a^n b^n : n \geq 0\}$ ، آنگاه L^2 برابر است با:

$$L^2 = \{a^n b^n a^k b^k : n \geq 0, k \geq 0\}$$

توجه کنید که n و k ارتباطی به هم ندارند و رشته $aabbab$ (یعنی $n=2, k=1$) در L^2 موجود است. ■

مثال

با فرض اینکه $L = \{awa : w \in \{a, b\}^*\}$ ، آنگاه L^2 برابر است با:

$$L^2 = \{aw_1 aaw_2 a : w_1, w_2 \in \{a, b\}^*\}$$

■

اتصال دو رابطه خاصیت جابجایی ندارد، یعنی $L_1 L_2 \neq L_2 L_1$ ✍

عمل اتصال نسبت به اجتماع خاصیت پخشی دارد اما نسبت به اشتراک ندارد. ✍

$$L_1(L_2 \cup L_3) = L_1 L_2 \cup L_1 L_3, \quad L_1(L_2 \cap L_3) \neq L_1 L_2 \cap L_1 L_3$$

معکوس

معکوس زبان L به صورت $L^R = \{w^R : w \in L\}$ تعریف می‌شود. روابط زیر برقرار می‌باشند:

$$\text{الف- } (L^R)^R = L \quad \text{ب- } (L^R)^n = (L^n)^R \quad \text{ج- } (L^n)^R = (L^R)^n \quad \text{د- } (L_1 L_2)^R = L_2^R L_1^R$$

مثال

معکوس $L = \{a^n b^n : n \geq 0\}$ برابر است با:

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

■

مکمل

مکمل (متمم) زبان L به صورت $\bar{L} = \Sigma^* - L$ تعریف می‌شود.

مثال

مکمل زبان $L = \{aa, bb\}$ ، را مشخص کنید. ($\Sigma = \{a, b\}$)

حل: زبان داده شده دارای دو رشته به طول ۲ است. مکمل این زبان، شامل همه رشته‌ها با طول صفر، یک، دو (به غیر از bb و aa)، سه و بزرگتر از سه است. این زبان را به صورت زیر نمایش می‌دهیم.

$$\bar{L} = \{\lambda, a, b, ab, ba\} \cup \{w \in \{a, b\}^* : |w| \geq 3\}$$

مثال

با فرض $\Sigma = \{a, b\}$ ، مکمل زبان $L = \{a^n b^n : n \in \mathbb{N}\}$ برابر است با: $\bar{L} = \{a^x b^y : x \neq y\}$

مثال

با فرض $\Sigma = \{0, 1\}$ مکمل $L = \{0^n 1^n 0^n : n \in \mathbb{N}\}$ برابر است

$$\bar{L} = \{0^n 1^m 0^k : n \neq m \neq k\}$$

بستار

بر روی زبان L دو عمل بستار ستاره‌ای و بستار مثبت قابل انجام است:

$$L^* = L^0 U L^1 U L^2 \dots = \bigcup_{i \geq 0} L^i$$

$$L^+ = L^1 U L^2 U L^3 \dots = \bigcup_{i \geq 1} L^i$$

چند نکته:

۱- عملگر بستار نسبت به هیچیک از عملگرهای "اجتماع"، اشتراک، تفاضل و الحاق" خاصیت پخشی ندارد. به عنوان نمونه: $(L_1 \cup L_2)^* \neq L_1^* \cup L_2^*$.

۲- برای هر زبان L داریم $\lambda \in L^*$ بنابراین $\lambda \notin \bar{L}^*$.

۳- هیچ زبانی وجود ندارد که در آن $\bar{L}^* = \bar{L}$.

۴- برای هر زبان L داریم: $\bar{L}^* \subset \bar{L}$.

هم ریختی

هم ریختی (homomorphism) یک نوع جایگزینی است که در آن به جای یک سمبل، از یک رشته استفاده می‌شود. با فرض اینکه Σ و Γ دو الفبا باشند، آنگاه تابع $h: \Sigma \rightarrow \Gamma^*$ هم ریختی نامیده می‌شود. تصویر هم ریختی زبان L به صورت $h(L) = \{h(w) : w \in L\}$ تعریف می‌شود.

مثال

با فرض $\Sigma = \{a, b\}$ و $\Gamma = \{a, b, c\}$ ، تابع h را بصورت $h(a) = ab$ و $h(b) = bbc$ تعریف می‌کنیم. مطلوب است $h(aba)$.

حل: کافی است در aba به جای a رشته ab و به جای b رشته bbc را قرار دهیم: $h(aba) = abbbcab$ ■

مثال

با فرض $\Sigma = \{a, b\}$ و $\Gamma = \{b, c, d\}$ ، h را به صورت $h(a) = bc$ و $h(b) = bdc$ تعریف می‌کنیم. اگر L زبان منظمی باشد که عبارت منظم $r = a^*(a + b^*)$ آن را معرفی کند، آنگاه عبارت منظمی که زبان منظم $h(L)$ را معرفی کند را مشخص کنید.

حل: کافی است در عبارت منظم داده شده، به جای a از رشته bc و به جای b از رشته bdc استفاده کرد:

$$(bc)^*(bc + (bdc)^*)$$

خانواده زبانهای منظم تحت هم ریختی بسته است. به عبارتی اگر h یک هم ریختی و L یک زبان منظم باشد، آنگاه $h(L)$ ، منظم خواهد بود.

تقسیم راست

با فرض اینکه L_1, L_2 زبان های تعریف شده بر روی یک الفبای یکسان باشند، آنگاه تقسیم راست L_1 به L_2 به صورت زیر تعریف می‌شود:

$$L_1 / L_2 = \{x : y \in L_2 \text{ برای برخی } xy \in L_1\}$$

برای تعیین L_1 / L_2 ، تمام رشته‌های موجود در L_1 با پسوندهای متعلق به L_2 را در نظر می‌گیریم. هر رشته با این فرض، پس از حذف مذکور، متعلق به L_1 / L_2 خواهد بود.

مثال

حاصل $\frac{L_1}{L_2}$ را تعیین کنید.

$$L_1 = \{001101, 0101010, 011100, 101010\}$$

$$L_2 = \{01, 10\}$$

حل: باید در رشته‌های L_1 ، پیشوندهایی را انتخاب کنیم که به 01 یا 10 ختم شوند. در زیر این پیشوندها خط کشیده شده است:

$$L_1 = \{\underline{001101}, \underline{0101010}, \underline{011100}, \underline{101010}\}$$

بنابراین خواهیم داشت:

$$L_1 / L_2 = \{0011, 01010, 1010\}$$

■

مثال

حاصل تقسیم راست L_1 / L_2 را تعیین کنید.

$$\text{الف- } L_2 = L(ab^*), \quad L_1 = L(a^*baa^*)$$

$$\text{ب- } L_2 = L(aba^*), \quad L_1 = L(a^*baa^*)$$

$$\text{پ- } L_2 = L(b^*c), \quad L_1 = L(a^*b^*c)$$

حل: باید در رشته‌های L_1 ، پیشوندهایی را انتخاب کنیم که به L_2 ختم شوند.

$$\text{الف: } \{a^*ba^*\} \quad \text{ب: } \{a^*\} \quad \text{پ: } \{a^*b^*\}$$

اگر L_1 و L_2 زبان‌های منظم باشند، آنگاه L_1 / L_2 نیز منظم خواهد بود. پس خانواده زبانهای منظم تحت تقسیم راست، بسته است.

زبان منظم

اگر بتوان برای زبانی، یک عبارت منظم نوشت، آن زبان منظم است. به ازای هر زبان منظم، یک عبارت منظم و به ازای هر عبارت منظم، یک زبان منظم وجود دارد. به عبارتی اگر L یک زبان منظم باشد، آنگاه عبارت منظمی به ازای Γ وجود خواهد داشت، بطوریکه $L=L(\Gamma)$.

برای بیان عبارت منظم یک زبان، دو شرط زیر باید برقرار باشد:

۱- هر رشته ای که عبارت منظم بیان می کند، شرط توصیف را داشته باشد.

۲- هر رشته ای که شرط توصیف را دارد، توسط عبارت منظم قابل بیان باشد.

با فرض اینکه r_1 و r_2 عبارت منظم و $L(r)$ ، زبان مربوط به عبارت منظم Γ باشد، خواهیم داشت:

$$L(r_1^*) = (L(r_1))^* \text{ الف-}$$

$$L(r_1.r_2) = L(r_1).L(r_2) \text{ ب-}$$

$$L(r_1 + r_2) = L(r_1) \cup L(r_2) \text{ ج-}$$

مثال

عبارت منظم $r = (a+b)^*(a+bb)^*$ ، در الفبای $\Sigma = \{a, b\}$ ، به چه زبانی اشاره دارد؟

حل: بخش $(a+b)^*$ به معنای هر رشته ای از a ها و b ها است. بخش $(a+bb)^*$ ، بیانگر یک a یا دو b می‌باشد. در نتیجه، $L(r)$ مجموعه تمام رشته هایی است که به یک a یا یک bb ختم می شوند.

$$L(r) = \{a, bb, aa, abb, ba, bbb, \dots\}$$



مثال

عبارت منظم $r = (aa)^*(bb)^*b$ ، در الفبای $\Sigma = \{a, b\}$ ، به چه زبانی اشاره دارد؟

حل: به مجموعه تمام رشته هایی با تعداد زوج a که قبل از تعداد فردی از b ها می‌آیند:

$$L(r) = \{a^{2n}b^{2k+1} : n \geq 0, k \geq 0\}$$



مثال

عبارت منظم $r = a^+(bb)^+$ ، در الفبای $\Sigma = \{a, b\}$ ، چه زبانی را توصیف می‌کند؟

$$L(r) = \{a^n b^{2k} : n \geq 1, k \geq 1\}$$

■

مثال

عبارت منظم $r = abbb^+(a+b)^+$ ، در الفبای $\Sigma = \{a, b\}$ ، به چه زبانی اشاره دارد؟

$$\blacksquare L = \{ab^n w : n \geq 3, w \in \{a, b\}^+\}$$

مثال

عبارت منظم زبان $L = \{a^n b^m : (n+m) \text{ is even}\}$ را بنویسید.

حل: برای اینکه مجموع تعداد a ها و b ها زوج باشد، باید تعداد هر دو زوج و یا تعداد هر دو فرد باشد.

$$r = (aa)^*(bb)^* + (aa)^* a(bb)^* b$$

■

مثال

عبارت منظم زبان $L = \{a^n b^m : (n+m) \text{ is odd}\}$ را بنویسید.

حل: برای اینکه مجموع تعداد a ها و b ها فرد باشد، باید تعداد a فرد و b زوج و یا تعداد a زوج و b فرد باشد.

$$r = (aa)^* a(bb)^* + (aa)^* (bb)^* b$$

مثال

عبارت منظمی برای $L = \{w : |w| \bmod 3 = 0\}$ ، در الفبای $\Sigma = \{a, b\}$ ، بنویسید.

حل: برای اینکه طول رشته مضرب ۳ باشد، تمام رشته‌های ممکن با طول سه را ایجاد و تکرار می‌کنیم:

$$r = ((a+b)(a+b)(a+b))^*$$

می‌توان عبارت را به صورت $(\Sigma \Sigma \Sigma)^*$ نیز نشان داد. ■

مثال

عبارت منظمی برای $L = \{w_1 b w_2 : |w_1| \bmod 2 = |w_2| \bmod 2, w_1, w_2 \in \{a\}^*\}$ در الفبای $\Sigma = \{a, b\}$ بنویسید.

حل: رشته‌های این زبان شامل یک b و تعداد زوجی a هستند:

$$r = (aa)^* b (aa)^* + a(aa)^* ba(aa)^*$$

مثال

عبارت منظمی برای زبان $L = \{a^n b^m : n \geq 4, m \leq 3\}$ ارائه دهید.

حل: مساله را به حالت‌های $m=0, 1, 2, 3$ تقسیم می‌کنیم. تعداد ۴ یا بیشتر a تولید کرده و پس از آن به تعداد لازم b قرار می‌دهیم: $r = aaaaa^* (\lambda + b + bb + bbb)$ ■

مثال

عبارت منظمی برای $L = \{vww : v, w \in \{a, b\}^+, |v| = 2\}$ بنویسید.

حل: تمام حالت‌های با ضابطه $|v| = 2$ را شمارش می‌کنیم:

$$r = aa(a+b)^* aa + ab(a+b)^* ab + ba(a+b)^* ba + bb(a+b)^* bb \quad \blacksquare$$

برای هر زبان، تعداد نامحدودی عبارت منظم وجود دارد. ✍️

مثال

عبارت‌های منظمی در الفبای $\Sigma = \{a, b\}$ برای هر یک از حالات زیر بنویسید.

الف- با aa شروع یا به bb ختم شود.

ب- با aa شروع و به bb ختم شود.

پ- به b یا ba ختم شوند.

ت- سومین نماد از راست b باشد.

ث- تعداد a بر ۳ بخش پذیر باشد.

ج- قبل یا بعد از هر a یک b باشد.

حل: منظور از x در حل این مثال، همان $(a + b)$ است.

الف: $aax^* + x^*bb$ ب: aax^*bb

پ: $x^*(b + ba)$ ت: x^*bxx

ث: $b^*(ab^*ab^*ab^*)^+$ ج: $(ab + ba + b + aba)^*$ ■

مثال

عبارت‌های منظمی در الفبای $\Sigma = \{0,1\}$ برای هر یک از حالات زیر بنویسید.

الف- شامل 0 نباشد.

ب- دقیقا دو تا 1 داشته باشد.

پ- با 1 شروع شده و شامل دو 0 متوالی نباشد.

ت- تعداد زوجی 0 داشته باشد.

ث- دارای هیچ دو 0 متوالی نباشد.

حل:

الف: 1^*

ب: $0^*10^*10^*$

پ: $(1+10)^*$

ت: $(1^*01^*01^*)^* + 1^*$

ث: $(1+01)^*(0+1^*)$ یا $(1+01)^*(0+\lambda)$

■

مثال

عبارت‌های منظمی در الفبای $\Sigma = \{0,1\}$ برای هر یک از حالات زیر بنویسید.

الف- حداقل یک عدد 1 داشته باشد.

ب- حداقل دو تا 1 داشته باشد.

پ- حداقل دو تا 1 متوالی داشته باشد.

ت- دارای زیر رشته 001 باشد.

ث- به 101 ختم شود.

ج- تعداد 0 و تعداد 1، زوج باشد.

حل: منظور از x همان $(0+1)$ است.

الف: x^*1x^* ب: $x^*1x^*1x^*$ پ: x^*11x^*

ت: x^*001x^* ث: x^*101 ج: $(00+11+(01+10)(00+11)^*(01+10))^*$

مثال

یک عبارت منظم برای هر یک از حالت‌های زیر در الفبای $\Sigma = \{a,b,c\}$ بنویسید.

الف- طول رشته زوج است. ب- طول رشته مضرب سه است.

پ- نماد شروع و پایان یکی باشد. ت- حداقل شامل یکی از سمبل‌های الفبا باشد.

حل: در حل این مثال، منظور از x همان $(a+b+c)$ است.

الف: $(xx)^*$

ب: $(xxx)^*$

پ: $a+b+c+ax^*a+bx^*b+cx^*c$

ت:

$x^*ax^*bx^*cx^* + x^*ax^*cx^*bx^* + x^*bx^*ax^*cx^* + x^*bx^*cx^*ax^* + x^*cx^*ax^*bx^* + x^*cx^*bx^*ax^*$ ■

مثال

یک عبارت منظم بنویسید که نمایش یک عدد صحیح یا اعشاری بدون علامت باشد.

حل: با فرض $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ، عبارت به صورت $D^+ + D^+.D^* + D^*.D^+$ است. مثال‌هایی از این رشته‌ها عبارتند از: 04, 3, 2.254, 12

مجموعه تمام اعداد حقیقی در زبان برنامه نویسی C، یک زبان منظم است.

مثال

آیا زبان $L = \{a^n b^k : n, k \geq 0\}$ منظم است؟

حل: بله - چون زبان L همان $a^* b^*$ است.

مثال

کدام یک از زبان‌های زیر منظم هستند؟

$$L_1 = \{uww^Rv : u, v, w \in \{a, b\}^+\}$$

$$L_2 = \{uww^Rv : u, v, w \in \{a, b\}^+, |u| \geq |v|\}$$

حل: فقط زبان L_1 منظم است. چون می‌توان یک عبارت منظم برای آن نوشت:

$$r_1 = (a + b)^+(aa + bb)(a + b)^+$$

■

مثال

کدام یک از زبان‌های زیر منظم هستند؟

$$L_1 = \{ww^R : w \in \{a\}^*\}$$

$$L_2 = \{ww^R : w \in \{a, b\}^*\}$$

حل: زبان L_1 منظم است.

■

مثال

کدام یک از زبان‌های زیر منظم هستند؟

$$L_1 = \{a^n b^m a^k : n + m + k > 5\}$$

$$L_2 = \{a^n b^m a^k : n > 5, m > 3, k \leq m\}$$

حل: فقط زبان L_1 منظم است. برای اثبات منظم بودن زبان L_1 ، کافی است آن را به حالت‌هایی مثل $m = 0, k = 0, n > 5$ تقسیم کرد. در اینصورت می‌توان برای هر یک عبارات منظمی ارائه داد. در ادامه منظم نبودن L_2 را به کمک لم تزریق نشان می‌دهیم. ■

چند نکته:

۱- این مساله که یک زبان به مقدار نامحدود حافظه نیاز دارد، الزاما به معنی نامنظم بودن آن نیست. زبان‌هایی وجود دارند که تعداد نامحدودی موقعیت دارند ولی منظم هستند. مانند:

$$L = \{w : w \text{ دارای تعداد مساوی از زیر رشته های } 01 \text{ و } 10 \text{ می باشد.}\}$$

۲- اگر L منظم باشد، آنگاه $L - \{\lambda\}$ منظم است.

۳- اگر L منظم باشد، آنگاه $L \cup \{a\}$ منظم است. (به ازای تمام $a \in \Sigma$)

۴- هر زبان متناهی، منظم است.

بسته بودن زبان‌های منظم

در حالت کلی یک مجموعه نسبت به عملی بسته است اگر اجرای آن عمل روی اعضای آن مجموعه، عضوی از همان مجموعه را نتیجه بدهد. مسئله بسته بودن یک زبان منظم روی یک عملگر، یعنی آیا زبان حاصل در اثر اعمال عملگر، باز هم منظم خواهد بود؟

✍ اگر L زبان منظمی باشد، آنگاه \bar{L} و L^* و L^R منظم خواهند بود. یعنی خانواده زبان‌های منظم تحت مکمل‌گیری، بستار ستاره‌ای و معکوس، بسته هستند.

✍ اگر L_1 و L_2 منظم باشند، آنگاه $L_1 \cup L_2$ ، $L_1 \cap L_2$ ، $L_1.L_2$ ، $L_1 - L_2$ منظم خواهند بود. یعنی خانواده زبان‌های منظم تحت اجتماع، اشتراک، الحاق و تفاضل بسته هستند.

مثال

نشان دهید خانواده زبان‌های منظم تحت تفاضل بسته هستند.

حل: تفاضل را می‌توان بر اساس اشتراک به صورت $L_1 - L_2 = L_1 \cap \overline{L_2}$ تعریف کرد. حال از آنجا که L_2 منظم است، $\overline{L_2}$ نیز منظم خواهد بود و چون زبان‌های منظم تحت اشتراک بسته هستند، $L_1 \cap \overline{L_2}$ نیز منظم خواهد بود. ■

مثال

با فرض اینکه L_1, L_2 زبان‌های منظم هستند، کدام یک از زبان‌های $L_3 = L_1 \cap L_2^R$ و $L_4 = \overline{L_1} \cap \overline{L_2}$ منظم است؟
حل: هر دو زبان منظم هستند، چون زبان‌های منظم نسبت به وارون، متمم و اشتراک بسته هستند. ■

مثال

با فرض اینکه زبان‌های L_1, L_2 روی الفبای Σ تعریف شده باشند، آنگاه کدام گزاره درست است؟

الف) اگر L_1, L_2 منظم باشند، آنگاه $L_1 \cap L_2$ منظم است.

ب) اگر L_1, L_2 منظم باشند، آنگاه $L_1 \cup L_2$ منظم است.

پ) اگر L_1, L_2 منظم باشند، آنگاه $L_1 - L_2$ منظم است.

حل: از آنجا که ϕ و Σ ، زبان‌های منظم هستند، دلیلی نادرستی گزاره‌ها عبارتند از:

الف: با فرض $L_1 = \phi$ ، آنگاه $L_1 \cap L_2 = \phi$ است و L_2 می‌تواند هر نوع زبانی باشد.


ب: با فرض $L_1 = \Sigma$ ، آنگاه $L_1 \cup L_2 = \Sigma$ است و L_2 می‌تواند هر نوع زبانی باشد.

پ: با فرض $L_1 = \phi$ ، آنگاه $L_1 - L_2 = \phi$ است و L_2 می‌تواند هر نوع زبانی باشد. ■

مثال

آیا اجتماع دو زبان نامنظم، می‌تواند منظم باشد؟

حل: بله - به طور نمونه، زبان‌های $L_1 = \{a^n b^m : n \leq m\}$ و $L_2 = \{a^n b^m : n > m\}$ هر دو نامنظم هستند، اما اجتماع آنها یعنی $L(a^* b^*)$ ، منظم است. ■

اگر L_1 و L_2 زبان‌های نامنظمی باشند، آنگاه $L_1 \cup L_2$ ممکن است منظم باشد. 

مثال

با فرض $L_i = a^i b^i$ ، آیا حاصل اجتماع نامتناهی L_i به ازای $(i = 0, 1, 2, 3, \dots)$ ، منظم است؟

حل: خیر- اجتماع این زبان‌ها برابر $L = \{a^n b^n : n \geq 0\}$ است که منظم نیست. (به ازای هر i ، L_i متناهی و در نتیجه منظم است). ■

مجموعه زبان‌های منظم، نسبت به اجتماع نامتناهی بسته نیستند.

مثال

آیا زبان $L = \{w : n_a(w) \neq n_b(w)\}$ منظم است؟

حل: خیر- می‌دانیم که زبان‌های منظم تحت مکمل‌گیری بسته هستند. بنابراین اگر L منظم باشد، باید مکمل آن هم منظم باشد. اما چون $\bar{L} = \{w : n_a(w) = n_b(w)\}$ ، منظم نیست، بنابراین L نیز منظم نمی‌باشد. ■

مثال

اگر L_1 و L_2 زبان‌های منظمی باشند، آیا $L = \{w : w \in L_1, w^R \in L_2\}$ لزوماً منظم است؟

حل: زبان L را می‌توان به صورت $L = \{w : w \in L_1, w \in L_2^R\}$ نوشت که همان $L_1 \cap L_2^R$ است. بنابراین چون زبان‌های منظم تحت معکوس و اشتراک بسته هستند، زبان $L_1 \cap L_2^R$ منظم است. ■

مثال

آیا زبان $L = \{w_1 c w_2 : w_1, w_2 \in \{a, b\}^*, w_1 \neq w_2\}$ منظم است؟

حل: فرض کنیم L منظم باشد، آنگاه متمم آن نیز منظم است. از آنجا که زبان‌های منظم تحت اشتراک بسته هستند، پس $\bar{L} \cap \{a^* c a^*\}$ باید منظم باشد. اما حاصل این اشتراک زبان $\{a^n c a^n\}$ است که منظم نیست. پس فرض خلف نادرست است و زبان L منظم نیست. ■

لم تزریق

یک زبان در صورتی منظم است که در جریان پردازش رشته‌ها، اطلاعاتی که باید در هر مرحله به خاطر آورده شوند، کاملاً محدود باشد. این گفته درست است، اما باید قبل از هر نوع استفاده‌ای به دقت اثبات شود. یکی از راه‌های اثبات نامنظم بودن یک زبان، استفاده از لم تزریق (پمپاژ) است.

اگر L یک زبان منظم نامتناهی باشد، آنگاه عدد صحیح مثبت m وجود دارد بطوریکه هر $w \in L$ با شرط $|w| \geq m$ ، را می‌توان به صورت $w = xyz$ تجزیه کرد، با فرض $|xy| \leq m$ و $|y| \geq 1$ بطوریکه $w_i = xy^i z$ به ازای تمام $i = 0, 1, 2, \dots$ عضو L باشد.

به طور خلاصه:

تمامی رشته‌های طولانی L را می‌توان به سه بخش تقسیم نمود، بطوریکه تعداد دلخواهی از تکرارهای بخش میانی، رشته دیگری را در L ایجاد کند. (می‌گوییم رشته میانی تزریق شده است.)

توسط لم پمپاژ فقط می‌توان منظم نبودن یک زبان را بررسی کرد و برای بررسی اینکه آیا یک زبان منظم هست از لم پمپاژ نمی‌توان استفاده کرد.

لم pumping یک شرط کافی برای منظم نبودن یک زبان ارائه می‌دهد.

مثال

آیا زبان $L = \{a^n b^n : n > 0\}$ منظم است؟

حل: فرض کنیم که L منظم باشد. حال رشته $w = a^n b^n$ متعلق به L را به ۳ قسمت تجزیه می‌کنیم:

$$x = a^{n-1}, y = a, z = b^n$$

حال باید به ازای تمام $i \geq 0$ ، رشته $xy^i z$ متعلق به L باشد، ولی مثلاً به ازای $i=2$ این چنین

$$\text{نیست: } a^{n-1} a^2 b^n = a^{n+1} b^n \notin L$$

بنابراین زبان داده شده منظم نیست. ■

مثال

آیا زبان $L = \{a^n b^m c^k : k = n + m\}$ منظم است؟حل: فرض کنیم که L منظم باشد. حال رشته $w = a^n b^{n+1} c^{2n+1}$ متعلق به L را به ۳ قسمت تجزیه می‌کنیم:

$$x = a^{n-1} \quad z = b^{n+1} c^{2n+1} \quad \text{و} \quad y = a$$

حال باید به ازای تمام $i \geq 0$ ، رشته $xy^i z$ ، متعلق به L باشد، ولی مثلاً به ازای $i=2$ این چنین نیست:

$$a^{n-1} a^2 b^{n+1} c^{2n+1} = a^{n+1} b^{n+1} c^{2n+1} \notin L$$

بنابراین زبان داده شده منظم نیست. چون: $(n+1) + (n+1) \neq 2n+1$ ■

مثال

آیا زبان $L = \{a^n : n \geq 2, n \text{ عددی اول است}\}$ منظم است؟حل: فرض کنیم که L منظم باشد. حال رشته $w = a^k$ با فرض اول بودن k ، متعلق به L را به ۳ قسمت تجزیه می‌کنیم:

$$x = a^{k-t-h} \quad z = a^h \quad \text{و} \quad y = a^t$$

حال باید به ازای تمام $i \geq 0$ ، رشته $xy^i z$ ، متعلق به L باشد، ولی مثلاً به ازای $i=k+1$ این چنین نیست:

$$a^{k-t-h} a^{t(k+1)} a^h = a^k a^{tk} = a^{(1+t)k} \notin L$$

مقدار $(1+t)k$ اول نیست، چون به k و $1+t$ بخش پذیر است. بنابراین زبان داده شده منظم نیست. ■

تمرین‌های فصل ۱

۱- رشته‌های تولید شده توسط عبارت منظم $(a+bb)^*(b+aa)^*$ با طول ۳ یا کمتر، را بنویسید.

۲- در الفبای $\Sigma = \{a,b\}$ ، درستی تساوی‌های زیر را بررسی کنید.

الف- $(a+b)^* = a^*(ba)^*$

ب- $(a+b)^* = a^* + b^*$

۳- در الفبای $\Sigma = \{0,1\}$ ، درستی تساوی‌های زیر را بررسی کنید.

الف- $1 + 0(0+10)^*11 = (00^*1)^*1$

ب- $0(0+10)^*11 = (0^*10)^*1$

پ- $1^*0^* + (0+1)^*01(0+1)^* = 0^*1^*$

۴- هر یک از عبارات منظم زیر در الفبای $\Sigma = \{0,1\}$ ، چه نوع رشته‌هایی را تولید می‌کنند؟

الف- $(0+1)^*(0+11)+1+\lambda$

ب- $0^*(1+1000^*)^*(\lambda+10)$

پ- $(1+01)^*(0+\lambda)$

۵- هر یک از عبارات منظم زیر، در الفبای $\Sigma = \{a,b\}$ ، چه نوع رشته‌هایی را تولید می‌کنند؟

الف- $(a+ba^*b)^*ba^*$

ب- $(b+ab+aab)^*(a+aa+\lambda)$

۶- عبارت منظم $100(000)^*1$ ، نشان دهنده چه مجموعه‌ای است؟

۷- آیا دو عبارت منظم زیر برابرند؟

$$r_1 = b^*(\lambda + ab^*ab^*)(\lambda + ab^*)$$

$$r_2 = (\lambda + b^*a)(\lambda + b^*ab^*a)b^*$$

۸- عبارت منظمی بنویسید که زبان $L = \{a^n b^{3m} c^{2k} : n \geq 1, m \geq 1, k \geq 1\}$ را توصیف کند.

۹- عبارت منظمی بنویسید که زبان $L = \{a^n b^m : n \geq 2, m \leq 3\}$ را توصیف کند.

۱۰- اگر زبان L به وسیله عبارت منظم $ba(b^*(ab)^* + a^*(bb)^*)ba$ توصیف شده باشد، وارون L را مشخص کنید.

۱۱- منظم بودن زبان‌های زیر را بررسی کنید.

الف- $\{a^n b^n (a+b)^* : n \geq 0\}$

ب- $\{a^* a^n b^n b^* : n \geq 0\}$

پ- $\{a^n b^m : n, m \geq 0\} - \{a^n b^m : n \neq m\}$

ت- $\{a^n b^m : n \geq 0, m \geq 0\} \cup \{a^n : n \geq 0\}$

ث- $\{w : n_a(w) \neq n_b(w)\} \cap \{a^* b^*\}$

۱۲- اگر L زبان نامنظم روی الفبای Σ باشد، آیا $(\Sigma^* - L)$ منظم است؟

۱۳- اگر L_1 و L_2 زبان‌های نامنظم روی الفبای Σ باشند، آیا $L_1 \cup L_2$ می‌تواند منظم باشد؟

۱۴- با توجه به زبان‌های زیر، عبارت $(L_1 \cup L_2)L_3^*$ معرف چه زبانی است؟

$$L_1 = \{a, b\}$$

$$L_2 = \{aa, ab, ba, bb\}$$

$$L_3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

۱۵- تابع $f : N \rightarrow N$ به صورت زیر می‌باشد. آیا زبان $L_f = \{0^n 1^{f(n)} : n \in N\}$ منظم است؟ ($L_f \subseteq \{0,1\}^*$)

$$f(n) = \begin{cases} 2n & n \text{ زوج} \\ 2n+1 & n \text{ فرد} \end{cases}$$

۱۶- آیا به ازای هر زبان منظم A ، زبان نامنظم B وجود دارد به گونه‌ای که $B \subseteq A$ ؟

۱۷- آیا به ازای هر زبان منظم A ، زبان نامنظم B وجود دارد به گونه‌ای که $A \subseteq B$ ؟

۱۸- اگر L_1 و L_2 زبان‌هایی منظم باشند و $L_1 \subseteq L \subseteq L_2$. آیا L منظم است؟

۱۹- آیا زبان منظم، می‌تواند ذاتا مبهم باشد؟

۲۰- لم تزیق برای منظم نبودن یک زبان شرط کافی است یا لازم؟

۲۱- نشان دهید $(L_1.L_2)^* \subset L_2^*$.

$$L_1 = \{a^n : n = 9k, k = 0, 1, 2, \dots\}$$

$$L_2 = \{a^n : n = 3k, k = 0, 1, 2, \dots\}$$

۲۲- اگر $L - \Sigma^* = \varnothing$ باشد، آنگاه L چه زبانی می‌تواند باشد؟

۲۳- اگر $L_1 = \{10, 1\}$ و $L_2 = \{011, 11\}$ باشد، حاصل $L_1.L_2$ را مشخص کنید.

۲۴- حاصل تقسیم راست L_1 / L_2 را تعیین کنید.

$$L_1 = \{0, 01, 111\}$$

$$L_2 = \{0, 1, 11\}$$

۲۵- اگر L یک زبان روی مجموعه Σ باشد، نمایشی برای $((L^R)^R)^R$ بنویسید.

۲۶- آیا اگر $s \in \text{palindrome}$ باشد، آنگاه $s^n \in \text{palindrome}$ است؟

$$\text{palindrome} = \{\lambda \text{ and all strings } s \text{ such that } \text{reverse}(s) = s\}$$

۲۷- درستی یا نادرستی گزاره‌های زیر را مشخص کنید.

الف- اگر L یک زبان منظم فاقد λ باشد، آنگاه $\text{truncate}(L)$ منظم خواهد بود.

ب- اگر L منظم باشد، $\text{chop2}(L) = \{w : vw \in L, |v| = 2\}$ هم منظم خواهد بود.

پ- اگر L منظم باشد، آنگاه $\text{even}(L) = \{\text{even}(w) : w \in L\}$ هم منظم خواهد بود.

۲۸- عبارت منظمی برای زبان $L = \{w : n_a(w) \bmod 3 > 0\}$ روی الفبای $\Sigma = \{a, b\}$ بنویسید.

۲۹- عبارت منظمی روی الفبای $\Sigma = \{a, b, c, d\}$ بنویسید که بیانگر تمام رشته‌هایی باشد که طول تمام دنباله‌های a آنها، مضرب ۳ باشد.

۳۰- عبارت منظمی روی الفبای $\Sigma = \{a, b\}$ بنویسید که بیانگر تمام رشته‌هایی باشد که شامل دنباله‌ای از a ها با طول بیشتر از ۲ نباشد.

۳۱- عبارت منظمی برای زبان $L = \{a^n b^m : n \geq 1, m \geq 1, nm \geq 3\}$ بنویسید.

۳۲- عبارت منظمی روی الفبای $\Sigma = \{a, b, c\}$ بنویسید که بیانگر تمام رشته‌هایی باشد که حداکثر دو تا a دارند.

۳۳- به کمک لم تزریق نشان دهید زبان $L = \{a^n b^p : n \leq p\}$ منظم نیست؟

۳۴- به کمک لم تزریق نشان دهید که زبان $L = \{n_a(w) = n_b(w)\}$ منظم نیست؟



حل تمرین فصل ۱

۱- رشته های تولید شده عبارتند از:

$\lambda, a, bb, ab, aaa, bbb, b, aa, aab, abb, bba$

۲- هر دو گزاره نادرست هستند:

الف: رشته b توسط سمت چپ تولید می شود اما توسط سمت راست تولید نمی شود.

ب: رشته aba توسط سمت چپ تولید می شود اما توسط سمت راست تولید نمی شود.

۳- هر سه گزاره نادرست هستند:

الف: رشته 1 توسط سمت چپ تولید می شود اما توسط سمت راست تولید نمی شود.

ب: رشته 011 توسط سمت چپ تولید می شود اما توسط سمت راست تولید نمی شود.

پ: رشته 10 توسط سمت چپ تولید می شود اما توسط سمت راست تولید نمی شود.

۴- حل هر گزاره در زیر آورده شده است:

الف: همه رشته هایی که به 01 ختم نمی شوند.

ب: تمام رشته هایی که شامل 101 نباشد.

پ: دارای یک زوج صفر متوالی نباشند.

۵- حل هر گزاره در زیر آورده شده است:

الف: دارای تعدادی فرد از کاراکتر b می باشند.

ب: همه رشته هایی که شامل سه تا a پشت سرهم نباشند.

۶- عبارت داده شده نشان دهنده مجموعه $\{ \langle 8^n + 1 \rangle_2 : n \geq 1 \}$ می باشد. به طور نمونه:

$$n = 1 \rightarrow 8^n + 1 = 9 \rightarrow 1001$$

$$n = 2 \rightarrow 8^n + 1 = 65 \rightarrow 1000001 = 100(000)1$$

$$n = 3 \rightarrow 8^n + 1 = 513 \rightarrow 1000000001 = 100(000)^2 1$$

۷- دو عبارت منظم داده شده برابرند:

$$r_1 = b^* + b^* ab^* + b^* ab^* ab^* + b^* ab^* ab^* ab^*$$

$$r_2 = b^* + b^* ab^* ab^* + b^* ab^* + b^* ab^* ab^* ab^*$$

۸- با توجه به زبان داریم:

الف- حداقل یک a باید داشت.

ب- حداقل تعداد b ها باید سه یا مضربی از سه باشد.

ج- حداقل تعداد c ها باید دو یا مضربی از دو باشد.

بنابراین عبارت منظمی که زبان L را توصیف کند برابر است با: $a^+(bbb)^+(cc)^+$

۹- جواب $aaa^*(\lambda + b + bb + bbb)$ است.

۱۰- برای بدست آوردن وارون، عبارت را از انتها به ابتدا می‌خوانیم:

$$L = ba(b^*(ab)^* + a^*(bb)^*)ba \Rightarrow L^R = ab((bb)^*a^* + (ba)^*b^*)ab$$

۱۱- بررسی گزاره‌ها:

الف: بله- از الحاق $a^n b^n$ با $(a+b)^*$ ، زبان $(a+b)^*$ حاصل می‌شود که منظم است.

ب: بله- عبارت $a^n a^n$ برابر a^{2n} و عبارت $b^n b^n$ برابر b^{2n} می‌باشد. پس زبان $a^* b^*$ بوده که منظم است.

پ: خیر- حاصل $\{a^n b^n : n \geq 0\}$ است که منظم نمی‌باشد.

ت: بله- زبان به صورت $a^* b^* + a^* = a^* b^*$ است، که منظم است.

ث: خیر- این زبان برابر $\{a^n b^m : n \neq m\}$ است که منظم نیست، چون باید نابرابری a ها و b ها چک شود.

۱۲- خیر- چون اگر L نامنظم باشد، آنگاه مکمل آن یعنی L^c حتما نامنظم است.

۱۳- بله- زبان $L_1 = \{a^n b^k : n \neq k\}$ نامنظم و زبان $L_2 = \{a^n b^k : n, k \geq 0\}$ منظم است، اما اجتماع آنها

یعنی $\{a^n b^k : n, k \geq 0\}$ ، منظم است.

مثالی دیگر: زبان $L_1 = \{a^n b^n : n \geq 0\}$ نامنظم و در نتیجه مکمل آن یعنی $L_2 = \overline{L_1}$ هم نامنظم خواهد بود. اما اجتماع آن‌ها یعنی $a^* b^*$ ، منظم است.

۱۴- با توجه به زبان‌های داده شده، می‌توان نوشت:

$$L_1 = \Sigma, L_2 = \Sigma^2, L_3 = \Sigma^3$$

از آنجا که عبارت $(\Sigma^3)^*(\Sigma \cup \Sigma^2)$ معرف مجموعه رشته‌هایی است که طول آنها مضرب ۳ نمی‌باشند، می‌توان عبارت داده شده را به صورت زیر بیان کرد:

$$L = \{w : w \in \{a.b\}^*, |w| \bmod 3 \geq 1\}$$

۱۵- خیر- تابع $f(n)$ یک به یک است، پس زبان منظم نیست.

۱۶- خیر- فرض کنید A برابر تهی باشد که زبانی منظم است. هیچ زبانی نمی‌توان پیدا کرد که زیر مجموعه تهی باشد.

۱۷- خیر- فرض کنید A برابر Σ^* باشد که زبانی منظم است. هیچ زبانی نمی‌توان پیدا کرد که Σ^* ، زیر مجموعه آن باشد.

۱۸- خیر- چون با فرض $L_1 = \phi$ و $L_2 = \Sigma^*$ ، برای زبان L چه منظم باشد و چه نامنظم، رابطه $L_1 \subseteq L \subseteq L_2$ برقرار است.

۱۹- خیر- زبان‌های منظم نمی‌توانند ذاتاً مبهم باشند.

۲۰- لم تزریق، یک شرط کافی برای منظم نبودن یک زبان است. چون اگر زبانی در لم تزریق صدق نکند، آنگاه حتماً نامنظم است.

۲۱- زبان‌های L_1 و L_2 را می‌توان به صورت زیر نوشت:

$$L_1 = \{\lambda, a^9, a^{18}, \dots\}$$

$$L_2 = \{\lambda, a^3, a^6, a^9, \dots\}$$

نتیجه الحاق این دو زبان برابر است با:

$$L_1.L_2 = \{\lambda, a^3, a^6, a^9, a^{12}, a^{15}, a^{18}, \dots\}$$

بنابراین: $(L_1.L_2)^* \subset L_2^*$

۲۲- عبارت $L - \Sigma^*$ را می‌توان به صورت $L \cap \phi$ نشان داد:

$$L - \Sigma^* = L \cap (\overline{\Sigma^*}) = L \cap \phi$$

که طبق صورت سؤال این مقدار برابر تهی است. از آنجا که اشتراک هر زبانی با زبان تهی، زبانی تهی می‌باشد، بنابراین زبان L می‌تواند هر زبانی باشد.

۲۳- نتیجه الحاق این دو زبان برابر است با:

$$L_1.L_2 = \{10,1\} \{011,11\} = \{10011,1011,101,111\}$$

۲۴- $\{\lambda, 0, 1, 11\}$

۲۵- از آنجا که روابط $(L^R)^R = L$ و $(L^R)^n = (L^n)^R$ برقرار می‌باشند، داریم:

$$((L^R)^n)^R = ((L^R)^R)^n = L^n$$

که L^n به صورت $\{w_1 w_2 \dots w_n \mid w_i \in L\}$ نمایش داده می‌شود.

۲۶- بله- رشته پالیندروم، رشته‌ای است که با معکوس‌اش یکی باشد. بنابراین اگر s یک رشته پالیندروم باشد، s^n نیز یک رشته پالیندروم است و برعکس. مثلاً $s = radar$ و $s^2 = radarradar$.

۲۷- هر سه گزاره درست هستند.

الف- زبان $truncate(L)$ ، از حذف آخرین سمبل سمت راست هر رشته از زبان L بوجود می‌آید.

ب- زبان $chop2(L)$ ، از حذف دو سمبل انتهایی سمت چپ هر رشته‌ای از زبان L بوجود می‌آید.

پ- زبان $even(w)$ ، از استخراج حروف واقع در موقعیت‌های زوج w بدست می‌آید.

$$r = (b^* ab^* ab^* ab^*)^* (a(a + \lambda)b^* ab^*) \quad -28$$

$$r = (aaa + b + c + d)^* \quad -29$$

$$r = (b^* + (a + \lambda)(a + \lambda)b)^* \cdot (a + \lambda) \cdot (a + \lambda) \quad -30$$

$$r = a^+ bbb^+ + aaa^+ b^+ + aa^+ bb^+ \quad -31$$

$$r = (b + c)^* (a + \lambda)(b + c)^* (a + \lambda)(b + c)^* \quad -32$$

۳۳- فرض کنیم که L منظم باشد. حال رشته $w = a^m b^m$ متعلق به L را به ۳ قسمت تجزیه کرد:

$$w = a^{m-t-k} a^t a^k b^m \Rightarrow x = a^{m-t-k}, y = a^t, z = a^k \cdot b^m$$

حال باید به ازای تمام $i \geq 0$ ، رشته $xy^i z$ متعلق به L باشد، ولی مثلاً به ازای $i=2$ این چنین

$$a^{m-t-k} (a^t)^2 a^k b^m = a^{m+t} b^m \notin L \text{ نیست:}$$

بنابراین زبان داده شده منظم نیست. ($t \geq 1$)

۳۴- فرض کنیم که L منظم باشد. حال رشته $w = a^m b^m$ متعلق به L را به ۳ قسمت تجزیه کرد:

$$w = a^{m-t-k} a^t a^k b^m \Rightarrow x = a^{m-t-k}, y = a^t, z = a^k \cdot b^m$$

حال باید به ازای تمام $i \geq 0$ ، رشته $xy^i z$ متعلق به L باشد، ولی مثلاً به ازای $i=0$ این چنین

$$a^{m-t-k} (a^t)^0 a^k b^m = a^{m-t} b^m \notin L \text{ نیست:}$$

بنابراین زبان داده شده منظم نیست.

فصل ۲:

گرامر - گرامر منظم

گرامر

گرامر G به صورت چهار تایی $G = (V, T, S, P)$ تعریف می‌شود که:

V : مجموعه متناهی از اشیاء به نام متغیرها

T : مجموعه متناهی از اشیاء به نام سمبل‌های پایانی (ترمینال)

S : سمبل ویژه‌ای به نام متغیر شروع ($S \in V$)

P : مجموعه متناهی از قوانین

تذکر: مجموعه‌های V و T غیر تهی و جدا از هم می‌باشند.

قوانین گرامر به شکل $x \rightarrow y$ می‌باشند که در آن $x \in (V \cup T)^+$ و $y \in (V \cup T)^*$ می‌باشد.

مثال

مجموعه V و T را در گرامر با قوانین، را مشخص کنید.

$$S \rightarrow Aab$$

$$A \rightarrow Aab$$

$$A \rightarrow a$$

حل: مجموعه V یا همان متغیرها برابر $V = \{S, A\}$ است. مجموعه T یا همان ترمینال‌ها برابر $T = \{a, b\}$

است. متغیرها با حروف بزرگ و ترمینال‌ها با حروف کوچک نمایش داده می‌شوند. ■

انواع گرامر

گرامرها را به چهار دسته تقسیم می‌شوند:

۱- منظم

گرامری که همه قواعد آن به صورت $A \rightarrow Bx \mid x$ یا $A \rightarrow xB \mid x$ باشد. ($x \in T^*$ و $A, B \in V$)

۲- مستقل از متن

گرامری که در سمت چپ کلیه قواعد آن، فقط یک متغیر باشد.

۳- حساس به متن

گرامری که تمامی قوانین آن به فرم $x \rightarrow y$ باشند که در آن x و y عضو $(V + T)^+$ باشند و $|x| \leq |y|$.

۴- بدون محدودیت

هیچ شرط و محدودیتی برای قواعد تولید ندارد. تنها محدودیت این است که λ نباید در سمت چپ قواعد تولید باشد.

مثال

نوع گرامر زیر را مشخص کنید.

1. $S \rightarrow AB$
2. $A \rightarrow aAb$
3. $bB \rightarrow bbbB$
4. $aAb \rightarrow aa$
5. $B \rightarrow \lambda$

حل: گرامر داده شده از نوع بدون محدودیت است.

این گرامر مستقل از متن نیست، چون در سمت چپ همه قواعد باید فقط یک متغیر باشد. در دو قانون ۳ و ۴ این چنین نیست.

این گرامر چون مستقل از متن نیست، پس حتما منظم نمی‌باشد.

این گرامر حساس به متن نیست، چون طول سمت چپ باید از سمت راست بیشتر باشد. در دو قانون ۴ و ۵ این چنین نیست. ■

زبان تولید شده توسط گرامر

با استفاده از گرامرها می‌توان بوسیله بکار بردن قوانین با ترکیب‌های مختلف، رشته‌های متعددی تولید کرد. مجموعه این رشته‌های پایانی، زبانی است که بوسیله گرامر تولید می‌شود.

فرض کنیم که $G = (V, T, S, P)$ یک گرامر باشد. آنگاه مجموعه $L(G) = \{w \in T^* : S \Rightarrow^* w\}$ ، زبانی است که توسط G تولید می‌شود.

تذکر: عبارت $S \Rightarrow^* w$ یعنی w از S با تعداد نامشخص (حتی صفر) مشتق می‌شود.

مثال

گرامر منظم λ $S \rightarrow aS$ ، زبان منظم a^* را تولید می‌کند. نحوه تولید رشته aa به کمک این گرامر:

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aa$$

■

مثال

گرامر منظم a $S \rightarrow aS$ ، زبان منظم a^+ را تولید می‌کند. نحوه تولید رشته aa به کمک این گرامر:

$$S \Rightarrow aS \Rightarrow aa$$

■

مثال

گرامر منظم a $S \rightarrow abS$ ، زبان منظم $(ab)^*a$ را تولید می‌کند. نحوه تولید رشته $ababa$:

$$S \Rightarrow abS \Rightarrow ababS \Rightarrow ababa$$

مثال

گرامر منظم زیر، چه زبانی را تولید می‌کند؟

$$S \rightarrow Aab$$

$$A \rightarrow Aab \mid a$$

حل: زبان منظم $a(ab)^+$ را تولید می‌کند. ■

مثال

گرامر مستقل از متن $S \rightarrow aSb \mid \lambda$ ، چه زبانی را تولید می‌کند؟

حل: زبان $\{a^n b^n : n \geq 0\}$ یعنی رشته‌های شروع شونده با a که تعداد a و b در آنها برابر است تولید می‌شود. نحوه تولید رشته $aabb$ به کمک این گرامر:

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

■

مثال

زبان گرامر مستقل از متن زیر را مشخص کنید.

$$S \rightarrow XY$$

$$X \rightarrow aXb \mid \lambda$$

$$Y \rightarrow cY \mid \lambda$$

حل: قوانین خط ۲ رشته‌هایی به فرم $a^n b^n$ و قوانین خط ۳ رشته‌هایی به فرم c^m تولید می‌کنند. در نهایت به علت وجود قانون ۱، رشته‌هایی به فرم $a^n b^n c^m$ تولید می‌شود. بنابراین:

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

نحوه تولید رشته $a^2 b^2 c$ توسط این گرامر: $(n=2, m=1)$

$$S \Rightarrow XY \Rightarrow aXbY \Rightarrow aaXbbY \Rightarrow aabbY \Rightarrow aabbcY \Rightarrow aabbc$$

■

مثال

گرامر حساس به متن زیر چه زبانی را تولید می‌کند؟

$$S \rightarrow abc \mid aAbc$$

$$Ab \rightarrow bA$$

$$Ac \rightarrow Bbcc$$

$$bB \rightarrow Bb$$

$$aB \rightarrow aa \mid aaA$$

حل: زبان حساس به متن $L = \{a^n b^n c^n : n \geq 1\}$ را تولید می‌کند. نحوه تولید رشته $a^2 b^2 c^2$:

$$S \Rightarrow aAbc \Rightarrow abAc \Rightarrow abBbcc \Rightarrow aBbbcc \Rightarrow aabbcc$$



گرامر منظم (regular)

گرامری که همه قواعد آن به صورت $A \rightarrow Bx \mid x$ باشد را خطی از چپ و گرامری که همه قواعد آن به صورت $A \rightarrow xB \mid x$ باشد را خطی از راست می‌گویند. ($x \in T^*$ و $A, B \in V$). گرامری که خطی از راست یا خطی از چپ باشد را گرامر منظم می‌گویند.

مثال

زبان منظم تولید شده توسط هر یک از گرامرهای منظم را مشخص کنید.

الف	ب
$S \rightarrow aA \mid ab$ $A \rightarrow bS$	$S \rightarrow aA \mid \lambda$ $A \rightarrow bS$

حل: الف: $(ab)^+$ ب: $(ab)^*$

مثال

زبان منظم تولید شده توسط هر یک از گرامرهای منظم را مشخص کنید.

الف	ب
$S \rightarrow aA \mid a$	$S \rightarrow aA \mid a$
$A \rightarrow aA \mid bA \mid a \mid b$	$A \rightarrow aA \mid bA \mid b$

حل: الف: $a + a(a+b)^*b$ ب: $a(a+b)^*$ ■

مثال

زبان منظم تولید شده توسط هر یک از گرامرهای منظم را مشخص کنید.

الف	ب	پ	ت
$S \rightarrow aaS \mid aA \mid bA$ $A \rightarrow bA \mid \lambda$	$S \rightarrow bS \mid aA$ $A \rightarrow bA \mid \lambda$	$S \rightarrow aS \mid aB \mid bB \mid \lambda$ $B \rightarrow bB \mid \lambda$	$S \rightarrow aS \mid aB$ $B \rightarrow bB \mid b$

حل: الف: $(aa)^*(a+b)b^*$ ب: b^*ab^* پ: a^*b^* ت: a^+b^+

مثال

زبان منظم تولید شده توسط هر یک از گرامرهای منظم را مشخص کنید.

الف	ب	پ
$S \rightarrow aaS \mid aX$ $X \rightarrow bX \mid bb$	$S \rightarrow aS \mid abX$ $X \rightarrow bX \mid \lambda$	$S \rightarrow aS \mid abS \mid \lambda$

حل: الف: $(aa)^*ab^+b$ ب: a^*abb^* پ: $(a+ab)^*$

■

مثال

زبان منظم تولید شده توسط هر یک از گرامرهای منظم را مشخص کنید.

الف	ب	پ	ت
$S \rightarrow aaS \mid abbX$ $X \rightarrow baX \mid \lambda$	$S \rightarrow aaaS \mid aX$ $X \rightarrow bbbbX \mid bbbb$	$S \rightarrow aaS \mid aaaX$ $X \rightarrow bbX \mid b$	$S \rightarrow aaS \mid aaabbX$ $X \rightarrow baX \mid \lambda$

حل:

الف: $(aa)^* abb(ba)^*$ ب: $(aaa)^* a(bbbb)^+ b$

پ: $(aa)^+ a(bb)^* b$ ت: $(aa)^+ abb(ba)^*$

مثال

گرامر منظمی برای زبان $L = \{a^n b^m : n \geq 2, m \geq 3\}$ بنویسید.

حل:

$S \rightarrow aaA$
 $A \rightarrow aA \mid bbbB$
 $B \rightarrow bB \mid \lambda$

مثال

در الفبای $\Sigma = \{a\}$ ، گرامر تولید کننده زبان $L = \{w : |w| \bmod 3 = 0\}$ را بنویسید.

حل: گرامر تولید کننده رشته‌هایی با حرف a که طول آنها مضرب ۳ باشد برابر است با:

$S \rightarrow aaaS \mid \lambda$

مثال

در الفبای $\Sigma = \{a\}$ ، گرامر تولید کننده زبان $L = \{w : |w| \bmod 3 = 2\}$ را بنویسید.

حل: $S \rightarrow aaaS \mid aa$ ■

مثال

گرامر تولید کننده زبان $L = \{w : |w| \bmod 3 > 0\}$ را بنویسید. ($\Sigma = \{a\}$)

حل: باقیمانده تقسیم صحیح هر عدد به 3، برابر 0 یا 1 یا 2 است. با توجه به شرط، حالت 0 مطرح نیست. توسط $S \rightarrow S_1$ حالتی که باقیمانده برابر 1 است و توسط $S \rightarrow S_2$ ، حالتی که باقیمانده برابر 2 است، تولید می‌شود.

$$S \rightarrow S_1 \mid S_2$$

$$S_1 \rightarrow aaaS_1 \mid a$$

$$S_2 \rightarrow aaaS_2 \mid aa$$

■

مثال

گرامر منظمی بنویسید که زبان $\{a^n b^m \mid n, m \geq 1\}$ هر دو زوج هستند را تولید کند.

حل:

$$S \rightarrow aaS \mid X$$

$$X \rightarrow bbX \mid \lambda$$

■

مثال

گرامر منظمی بنویسید که زبان $\{a^n b^m \mid n, m \geq 1\}$ هر دو فرد هستند را تولید کند.

حل:

$$S \rightarrow aaS \mid aY$$

$$Y \rightarrow bbY \mid b$$

مثال

گرامر منظمی بنویسید که زبان $\{a^n b^m : n+m \text{ زوج است}\}$ را تولید کند.

حل: برای اینکه جمع n و m زوج باشد، باید هر دو فرد یا هر دو زوج باشند. این دو حالت در مثالهای قبل بررسی شدند و کافی است با دستور $S \rightarrow S_1 \mid S_2$ ، آنها را با هم ترکیب کرد.

$$\begin{aligned} S &\rightarrow S_1 \mid S_2 \\ S_1 &\rightarrow aaS_1 \mid X \\ X &\rightarrow bbX \mid \lambda \\ S_2 &\rightarrow aaS_2 \mid aY \\ Y &\rightarrow bbY \mid b \end{aligned}$$



تمرین‌های فصل ۲

۱- نحوه تولید رشته $w = acaacabbdebde$ توسط گرامر زیر را نشان دهید.

$$S \rightarrow acBdeA \mid BAB$$

$$B \rightarrow aSb \mid ae \mid \lambda$$

$$A \rightarrow aAb \mid b \mid \lambda$$

۲- نحوه تولید رشته $w = a^5$ توسط گرامر زیر را مشخص کنید.

$$S \rightarrow ACaB$$

$$Ca \rightarrow aaC$$

$$CB \rightarrow DB$$

$$CB \rightarrow E$$

$$aD \rightarrow Da$$

$$AD \rightarrow AC$$

$$aE \rightarrow Ea$$

$$AE \rightarrow a$$

۳- نحوه تولید رشته $w = a^6$ توسط گرامر زیر را مشخص کنید.

$$S \rightarrow ABC$$

$$B \rightarrow DBE \mid \lambda$$

$$DE \rightarrow EFaa$$

$$aE \rightarrow Ea$$

$$aC \rightarrow Ca$$

$$FE \rightarrow EFa$$

$$FC \rightarrow C$$

$$AE \rightarrow A$$

$$AC \rightarrow \lambda$$

۴- نحوه تولید رشته $(((:a)),a)$ توسط گرامر زیر را مشخص کنید.

$$S \rightarrow F \mid (S,SH)$$

$$F \rightarrow ((:F)) \mid a$$

$$H \rightarrow ,SH \mid \lambda$$

۵- نحوه تولید رشته *babba* توسط گرامر زیر را مشخص کنید.

$$S \rightarrow aA \mid Aa$$

$$A \rightarrow aA \mid bA \mid \lambda$$

۶- کدام گرامر منظم می‌باشند؟

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

$$G2: S \rightarrow Ab \mid ab$$

$$G3: S \rightarrow aA \mid ab, \quad A \rightarrow Sa$$

۷- عبارت منظمی معادل با هر یک از گرامرهای زیر بنویسید.

پ	ب	الف
$S \rightarrow aA \mid bA$ $A \rightarrow aA \mid bA \mid \lambda$	$S \rightarrow aB \mid cB$ $B \rightarrow abB \mid cbB \mid acB \mid \lambda$	$S \rightarrow bS \mid aA \mid \lambda$ $A \rightarrow bA \mid aB$ $B \rightarrow bB \mid aS$

۸- عبارت منظمی معادل با هر یک از گرامرهای زیر بنویسید.

پ	ب	الف
$S \rightarrow A$ $A \rightarrow 0B \mid 0$ $B \rightarrow 1C$ $C \rightarrow 1A$	$S \rightarrow 0B \mid A$ $A \rightarrow 1A \mid S$ $B \rightarrow 1S \mid 1$	$S \rightarrow 00S \mid X$ $X \rightarrow 11X \mid \lambda$

۹- عبارت منظمی معادل با هر یک از گرامرهای زیر بنویسید.

پ	ب	الف
$S \rightarrow T \mid D$ $T \rightarrow aT \mid bD$ $D \rightarrow bD \mid aT \mid \lambda$	$S \rightarrow aaS \mid abbX$ $X \rightarrow baX \mid bbY$ $Y \rightarrow aY \mid \lambda$	$S \rightarrow bS \mid aA \mid \lambda$ $A \rightarrow bA \mid aB$ $B \rightarrow bB \mid aS$

۱۰- آیا هر گرامر خطی، گرامر منظم است؟

۱۱- یک گرامر خطی راست برای زبان $(aab^*ab)^*$ بنویسید.

پاسخ تمرین فصل ۲

۱- نحوه تولید:

$$\begin{aligned}
 S &\Rightarrow acBdeA \\
 &\Rightarrow acaSbdeA \\
 &\Rightarrow acaacBdeAbdeA \\
 &\Rightarrow acaacaSbdeAbdeA \\
 &\Rightarrow acaacaBABbdeAbdeA \\
 &\Rightarrow acaacaAbdeAbdeA \\
 &\Rightarrow acaacabbdebdde
 \end{aligned}$$
۲- اگر اشتقاق زیر را دنبال کنیم به $aaaaaa$ می‌رسیم:
$$\begin{aligned}
 S &\Rightarrow ACaB \Rightarrow AaaCB \Rightarrow AaaDB \Rightarrow AaDaB \Rightarrow ADaaB \Rightarrow ACaaB \\
 &\Rightarrow AaaCaB \Rightarrow AaaaaCB \Rightarrow AaaaaE \Rightarrow AaaaEa \Rightarrow AaaEaa \Rightarrow AaEaaa \\
 &\Rightarrow AEaaaa \Rightarrow aaaaa
 \end{aligned}$$

۳- نحوه تولید:

$$\begin{aligned}
 S &\Rightarrow ABC \Rightarrow ADBEC \Rightarrow ADDBEEC \Rightarrow ADDEEC \\
 &\Rightarrow ADEFaaEC \Rightarrow AEFaaFaaEC \Rightarrow AFaaFaaEC \\
 &\Rightarrow AFaaFaEaC \Rightarrow AFaaFEaaC \Rightarrow AFaaEFaaaC \\
 &\Rightarrow AFaEaFaaaC \Rightarrow AFEaaFaaaC \Rightarrow AEFaaaFaaaC \\
 &\Rightarrow AFaaaFaaaC \Rightarrow AFaaaFaaCa \Rightarrow AFaaaFaCaa \\
 &\Rightarrow AFaaaFCaaa \Rightarrow AFaaaCaaa \Rightarrow AFaaCaaaa \\
 &\Rightarrow AFaCaaaaa \Rightarrow AFCaaaaaa \Rightarrow ACaaaaaa \\
 &\Rightarrow aaaaaa
 \end{aligned}$$
زبان گرامر به صورت $\{a^{n(n+1)} : n \geq 0\}$ است.

۴- نحوه تولید:

$$\begin{aligned}
 S &\Rightarrow (S, SH) \Rightarrow (F, SH) \Rightarrow (((: F)), SH) \Rightarrow (((: F)), FH) \Rightarrow (((: a)), FH) \\
 &\Rightarrow (((: a)), a)
 \end{aligned}$$

۵- نحوه تولید:

$$S \Rightarrow Aa \Rightarrow bAa \Rightarrow baAa \Rightarrow babAa \Rightarrow babbAa \Rightarrow babba$$
۶- گرامر G_2 منظم است. در گرامر منظم، همه قواعد خطی راست یا خطی چپ می‌باشند.

۷- عبارت منظم هر یک از گرامرها برابرند با:

الف- $(b^* ab^* ab^* a)^* b^*$

ب- $(a+c)(ab+cb+ac)^*$

پ- $(a+b)(a+b)^*$

۸- عبارت منظم هر یک از گرامرها برابرند با:

الف- $(00)^* (11)^*$

ب- $1^* (01^+)^+$

پ- $(011)^* 0$

۹- عبارت منظم هر یک از گرامرها برابرند با:

الف- $(a+b)^2 (a+b)^+$

ب- $(aa)^* abb(ba)^* bba^*$

پ- $(a^* b + \lambda) b^* (aa^* bb^*)^*$

۱۰- خیر- گرامری منظم است که همه قواعد آن، خطی چپ یا خطی راست باشند. در گرامر خطی، ممکن است تعدادی از قواعد خطی چپ و تعدادی خطی راست باشند.

۱۱- گرامر :

$$S \rightarrow aaA \mid \lambda$$

$$A \rightarrow bA \mid abS$$

فصل ۳: اتوماتای متناهی

ماشین‌ها (اتوماتا) ابزارهایی هستند برای تشخیص رشته‌های زبان که رشته را از چپ به راست بررسی کرده و نهایتاً اعلام می‌کنند که آیا رشته متعلق به زبان هست یا نه. ماشین‌ها را می‌توان به عنوان مدل‌های ریاضی برای کامپیوترهای واقعی در نظر گرفت.

انواع ماشین

یک دسته بندی برای اتوماتا به صورت زیر است:

۱- متناهی (FA)

ماشین پذیرنده‌ای که حافظه ندارد و خروجی آن دارای دو حالت پذیرش یا عدم پذیرش است.

۲- پشته‌ای (PDA)

ماشین پذیرنده‌ای که حافظه آن به صورت پشته بوده و خروجی آن دارای دو حالت پذیرش یا عدم پذیرش است.

۳- کراندار خطی (LBA)

ماشینی دارای حافظه از دو سر محدود با قابلیت خواندن و نوشتن است.

۴- تورینگ (TM)

ماشینی دارای حافظه نامحدود با قابلیت خواندن و نوشتن است.

نکات:

۱- ماشین متناهی، قادر به پذیرش زبان منظم است.

۲- ماشین پشته‌ای، قادر به پذیرش زبان مستقل از متن است.

۳- ماشین کراندار خطی، قادر به پذیرش زبان حساس به متن است.

۴- ماشین تورینگ تشخیص دهنده، قادر به پذیرش زبان بازگشتی شمارش پذیر است.

در این فصل ماشین متناهی را بررسی می‌کنیم. ماشین‌های دیگر در فصل‌های بعدی بررسی خواهند شد.

ماشین‌های متناهی

یکی از ساده‌ترین انواع ماشین‌ها، ماشین متناهی می‌باشد که از آن در شناخت زبان‌های منظم استفاده می‌شود. اتوماتای متناهی مدل مناسبی برای کامپیوتر با محدودیت شدید حافظه است. ماشین‌های متناهی (FA)، به دو دسته معین (DFA) و نامعین (NFA) تقسیم می‌شوند.

DFA : Deterministic Finite Acceptor

NFA : Nondeterministic Finite Acceptor

پذیرنده متناهی معین (DFA)

ماشین متناهی قطعی (معین) بوسیله پنج تایی $M = (Q, \Sigma, \delta, q_0, F)$ تعریف می‌شود که در آن:

Q : مجموعه متناهی از حالات داخلی

Σ : مجموعه متناهی از علائمی به نام الفبای ورودی

δ : تابع انتقال $(Q \times \Sigma \rightarrow Q)$

q_0 : حالت شروع

F : مجموعه حالات پایانی

یک dfa، مانند همه اتوماتاها دارای حالت‌های داخلی، قوانینی برای انتقال از یک حالت به حالت دیگر، تعدادی ورودی و همچنین روشهایی برای تصمیم‌گیری هستند.

نحوه عملکرد ماشین:

اتوماتا ابتدا در حالت شروع و هد خواندن از نوار ورودی، روی آخرین سمبل از سمت چپ رشته ورودی قرار دارد. با هر یک از حرکت‌های اتوماتا، هد یک موقعیت به راست می‌رود. اگر با رسیدن به پایان رشته، اتوماتا در یکی از حالت‌های پایانی قرار داشته باشد، رشته پذیرفته می‌شود.

انتقال از یک حالت داخلی به حالت داخلی دیگر، توسط تابع انتقال انجام می‌شود. بعنوان مثال $\delta(q_i, a) = q_j$ ، یعنی اگر dfa در حالت q_i باشد و هد بر روی a باشد، آنگاه ماشین به حالت q_j تغییر حالت می‌دهد.

هد نوار در ماشین متناهی، فقط به سمت راست حرکت می‌کند.

مثال

نمودار تغییر وضعیت DFA زیر را به کمک شکل نشان دهید.

$$\Sigma = \{a, b\} \quad , \quad Q = \{q_0, q_1\} \quad , \quad F = \{q_1\}$$

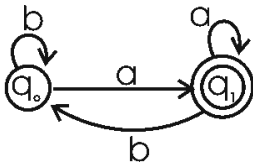
$$\delta(q_0, a) = q_1$$

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

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

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

حل: این ماشین دارای دو حالت q_0 و q_1 است که q_0 حالت شروع و q_1 حالت پایانی است. طبق تابع انتقال‌های داده شده، ماشین اگر در حالت q_0 باشد و هد a را بخواند به حالت q_1 می‌رود و اگر b را بخواند در همان حالت q_0 باقی می‌ماند. همچنین اگر ماشین در حالت q_1 باشد و هد b را بخواند به حالت q_0 رفته و اگر a را بخواند در همان حالت q_1 باقی می‌ماند. نمودار تغییر وضعیت DFA این مثال در شکل زیر نشان داده شده است:

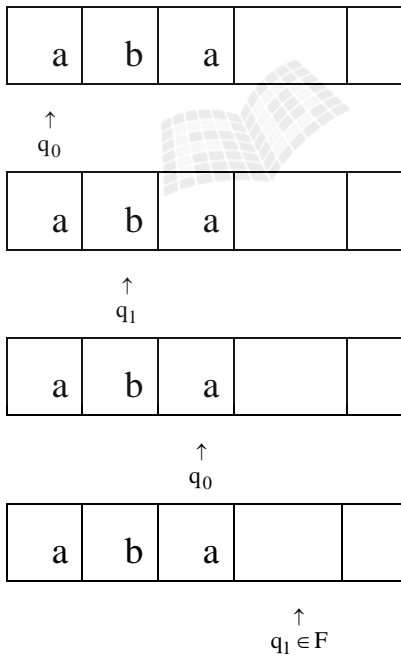


حالت نهایی به صورت دواپری دو خطی نمایش داده می‌شود. ■

مثال

آیا رشته $W=aba$ توسط ماشین مثال قبل، پذیرفته می‌شود؟

حل: بله - نمایش تعلق رشته aba به زبان ماشین:



نحوه پذیرش رشته توسط ماشین: با شروع از حالت q_0 ، ابتدا سمبل a را می‌خواند. با توجه به یال‌های گراف ماشین به حالت q_1 می‌رود. سپس، سمبل b خوانده شده و ماشین به حالت q_0 می‌رود. در نهایت سمبل a خوانده شده و ماشین به حالت q_1 می‌رود. در این لحظه، هم در پایان رشته و هم در حالت پایانی قرار داریم. بنابراین رشته aba پذیرفته می‌شود. نشان دهید ماشین رشته abb را نمی‌پذیرد. ■

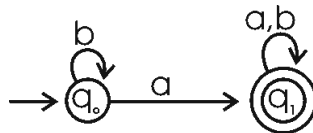
زبان‌ها و dfa

زبان مجموعه‌ای از تمام رشته‌های پذیرفته شده توسط اتومات می‌باشد. زبان پذیرفته شده توسط dfa $M = (q, \Sigma, \delta, q_0, F)$ ، مجموعه تمام رشته‌های روی Σ است که توسط M پذیرفته می‌شوند. فرم صورتی آن به صورت $L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}$ است. dfa پس از پردازش هر یک از رشته‌های Σ^* ، یا آنها را می‌پذیرد و یا رد می‌کند. عدم پذیرش به این معنی است که dfa در یکی از حالت‌های غیر پایانی متوقف شود.

مثال

عبارت منظم مربوط به DFA زیر را مشخص کنید؟

(q_0 حالت شروع و q_1 حالت پایانی است.)

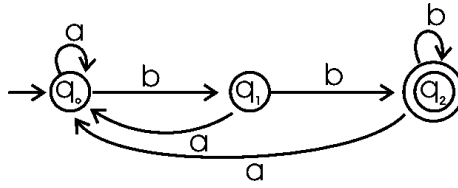


حل: $b^*a(a+b)^*$ ■

مثال

برای عبارت منظم $bb(a+b)^*$ ، یک DFA رسم کنید. $\Sigma = \{a, b\}$

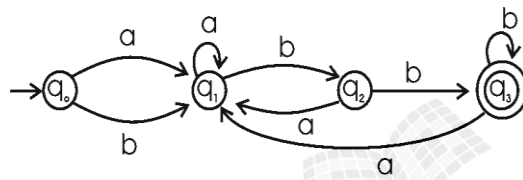
حل:



مثال

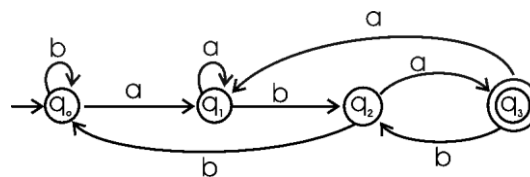
برای عبارت منظم $(a+b)^+bb$ ، یک DFA رسم کنید. $\Sigma = \{a, b\}$

حل:



مثال

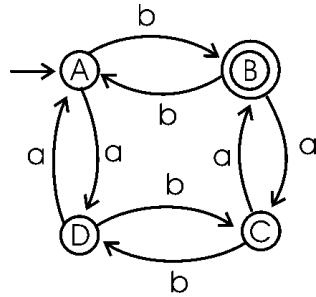
عبارت منظم مربوط به DFA زیر را مشخص کنید؟ $\Sigma = \{a, b\}$



حل: $(a+b)^*aba$ (رشته‌هایی که به aba ختم می‌شوند).

مثال

ماشین زیر چه نوع رشته‌هایی را می‌پذیرد؟ $\Sigma = \{a, b\}$

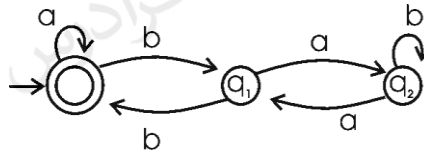


حل: رشته‌هایی با تعداد زوجی a و تعداد فردی b .

این مثال را با تغییر حالت پایانی به حالت‌های دیگر، بررسی کنید. به طور نمونه اگر C حالت پایانی بود (به جای B)، آنگاه چه زبانی را می‌پذیرفت؟ ■

مثال

عبارت منظم مربوط به DFA زیر را مشخص کنید؟ $\Sigma = \{a, b\}$

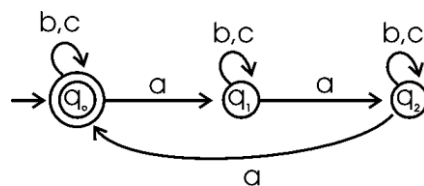


حل: $(a + b(ab^*a)^*b)^*$.

رشته λ نیز پذیرفته می‌شود، چون حالت شروع، حالت پایانی نیز می‌باشد. ■

مثال

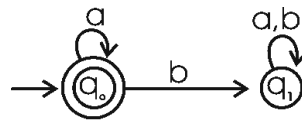
ماشین DFA زیر چه نوع رشته‌هایی را می‌پذیرد؟ $\Sigma = \{a, b, c\}$



حل: رشته‌هایی که تعداد a در آنها، مضرب 3 باشد. ■

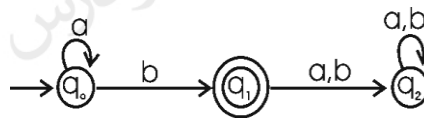
حالت دام یا تله (trap state)

می‌خواهیم ماشینی رسم کنیم که زبان a^* را روی الفبای $\Sigma = \{a, b\}$ بپذیرد. برای این کار یک DFA با یک حالت رسم می‌کنیم که هم حالت شروع و هم حالت پایانی است به طوری که یال با برچسب a از آن حالت خارج شده و به همان حالت وارد می‌شود. در این ماشین اگر قبل یا بعد از حرف a ، حرف b ظاهر شود، رشته نباید پذیرفته شود، بنابراین یک حالت دیگر به نام q_1 رسم کرده و با یالی با برچسب b به آن حالت می‌رویم که خروج از آن ممکن نمی‌باشد. در شکل زیر این DFA رسم شده که وضعیت q_1 همان وضعیت تله است:



مثال

در DFA زیر، حالت trap را مشخص کنید.

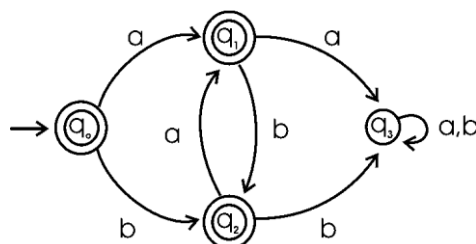


حل: حالت q_2 . (زبان ماشین a^*b است.)

مثال

DFA پذیرنده زبانی که رشته‌های آن شامل aa یا bb نباشد را رسم کنید.

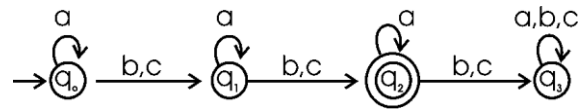
حل: حالت q_3 ، trap است.



مثال

در الفبای $\Sigma = \{a, b, c\}$ ، DFA پذیرنده زبانی که در رشته‌های آن تعداد کل **b** ها و **c** ها برابر ۲ باشد را رسم کنید.

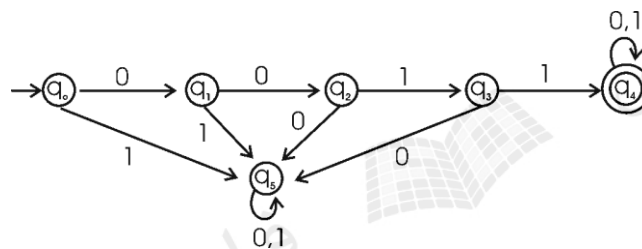
حل: حالت q_3 ، trap است.



مثال

در الفبای $\Sigma = \{0,1\}$ ، DFA پذیرنده زبانی که رشته‌های آن با زیر رشته 0011 آغاز شود را رسم کنید.

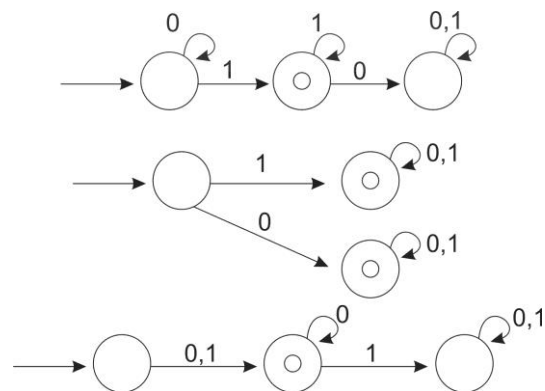
حل: حالت q_5 ، trap است.



مثال

برای هر یک از زبان‌های $1(0+1)^*$ و $1(0+1)^*$ و $(0+1)0^*$ یک DFA رسم کنید.

حل: هر dfa به ترتیب از بالا به پایین رسم شده است:



اصلاح: در شکل وسط، حالت پایینی، حالت trap است.

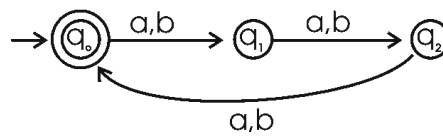
مکمل DFA

برای رسم مکمل یک DFA، کافی است که کلیه حالت‌های غیر پایانی را به حالت پایانی و کلیه حالات پایانی را به حالات غیر پایانی تبدیل کنیم. (جهت یالها و مقدار برجسب آنها تغییری نمی‌کند).

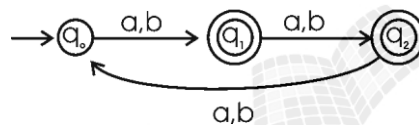
مثال

ای رسم کنید که رشته‌هایی که طول آنها مضربی از 3 نباشد را بپذیرد. ($\Sigma = \{a, b\}$)

حل: ابتدا DFA ای رسم می‌کنیم که رشته‌هایی را بپذیرد که طول آنها مضربی از 3 باشد:



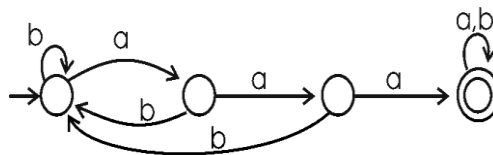
سپس DFA بالا را مکمل می‌کنیم:



مثال

ای رسم نمایید که زبان $L = \{w : \text{در آن وجود ندارد } aaa\}$ را بپذیرد. ($\Sigma = \{a, b\}$)

حل: ابتدا DFA ای رسم می‌کنیم که زبان $L = \{w : \text{زیر رشته‌ای از } w \text{ باشد } aaa\}$ را بپذیرد:



■ سپس DFA بالا را مکمل می‌کنیم.

اگر $M = (Q, \Sigma, \delta, q_0, F)$ و $\hat{M} = (Q, \Sigma, \delta, q_0, Q - F)$ دو DFA باشند، آنگاه:

$$\overline{L(M)} = L(\hat{M})$$

برای به دست آوردن مکمل یک NFA از روی نمودار حالت، باید ابتدا NFA را به DFA تبدیل

کرده و سپس مکمل DFA را بدست آورد.

پذیرنده های متناهی نامعین (NFA)

اگر به ازای هر حالت ماشین و هر نماد ورودی به صورت منحصر بفردی حالت بعدی مشخص باشد به آن معین (Deterministic) می‌گویند. در یک ماشین نامعین (Nondeterministic) در هر لحظه ممکن است چندین انتخاب مختلف موجود باشد. پذیرنده‌های متناهی نامعین (غیرقطعی)، پیچیده تر از انواع معین خود هستند. ماشین نامعین می‌تواند به ازای دریافت یک ورودی در هر حالت، به چندین حالت مختلف تغییر حالت دهد.

تعریف

یک پذیرنده متناهی نامعین (NFA) بوسیله پنج تایی $M = (Q, \Sigma, \delta, q_0, F)$ تعریف می‌شود که در آن Q و Σ و q_0 و F همانند DFA تعریف می‌شوند، ولی تابع انتقال به صورت $Q \times \Sigma \cup \{\lambda\} \rightarrow 2^Q$ تعریف می‌شود. برد این تابع مجموعه ای از حالات مجاز برای ماشین می‌باشد.

سه تفاوت عمده بین تعریف NFA و تعریف DFA وجود دارد. در NFA:

۱- محدوده تابع δ در مجموعه توانی 2^Q است. مثلاً اگر وضعیت فعلی q_0 باشد و حرف a خوانده شود، آنگاه هریک از حالت‌های q_1, q_2 می‌تواند وضعیت بعدی باشد:

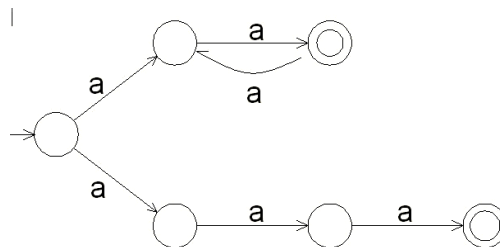
$$\delta(q_0, a) = \{q_1, q_2\}$$

۲- بعنوان ورودی قابل قبول است. یعنی NFA می‌تواند بدون استفاده از سمبل ورودی، دست به انتقال بزند. هد می‌تواند در بعضی انتقال‌ها حرکت نکند.

۳- $\delta(q_i, a)$ می‌تواند تهی باشد، یعنی هیچ انتقالی برای این وضعیت خاص تعریف نشده است.

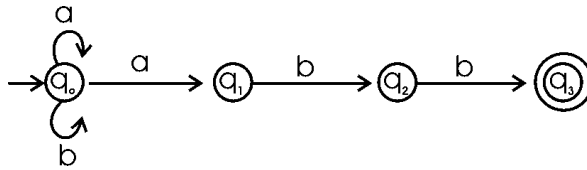
مثال

آیا ماشین زیر یک NFA می‌باشد؟



حل: بله- چون دو انتقال با برچسب a از حالت شروع دارد. اولین انتخاب منجر به پذیرش تمام رشته‌های دارای تعداد زوجی از a می‌شود و دومین انتخاب منجر به پذیرش رشته a^3 می‌شود. زبان پذیرفته شده توسط این ماشین، $L = \{a^{2n} : n \geq 1\} \cup \{a^3\}$ می‌باشد. ■

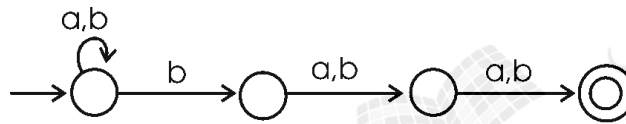
مثال

ماشین NFA زیر چه زبانی را می‌پذیرد؟ $\Sigma = \{a, b\}$ 

حل: رشته‌هایی را که به زیر رشته **abb** ختم شوند. (این ماشین **dfa** نیست. به طور نمونه، در حالت q_1 ، دریافت **a** پیش بینی نشده است یا در q_0 با دریافت **a**، دو تغییر حالت ممکن است.) ■

مثال

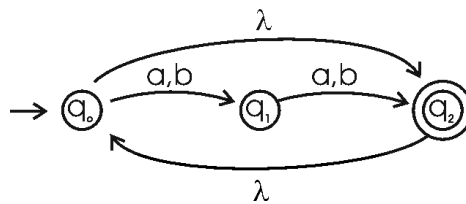
ماشین NFA زیر چه زبانی را می‌پذیرد؟



حل: رشته‌های که سومین نماد از سمت راست آنها، **b** باشد. ■

مثال

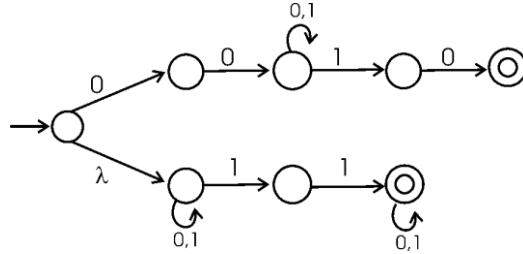
ماشین NFA زیر چه زبانی را می‌پذیرد؟



حل: ماشین داده شده، رشته‌هایی با طول زوج را می‌پذیرد. (در تعریف NFA اجازه داشتن یال با برچسب λ را داریم. به عبارتی λ بعنوان ورودی قابل قبول است. یعنی ماشین می‌تواند بدون استفاده از سمبل ورودی، دست به انتقال بزند. هد می‌تواند در بعضی انتقال‌ها حرکت نکند.) ■

مثال

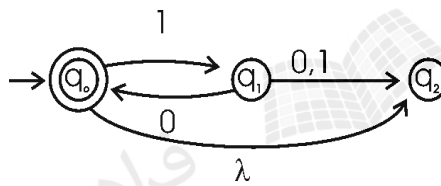
ماشین NFA زیر چه زبانی را می‌پذیرد؟ $\Sigma = \{0,1\}$



حل: رشته‌هایی که با 00 شروع و به 10 ختم شوند یا شامل زیر رشته 11 باشند.

مثال

زبان پذیرفته شده توسط اتومات شکل زیر چیست؟ $\Sigma = \{0,1\}$



حل: اتومات زبان $L = \{(10)^n : n \geq 0\}$ را می‌پذیرد.

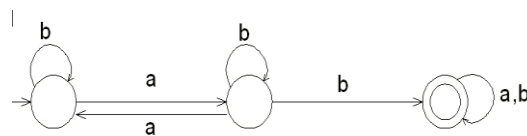
مثال

می‌دانیم که یک DFA داده شده با 4 حالت، رشته‌ای به طول 6 را می‌پذیرد. آیا تعداد اعضای زبان پذیرش شده توسط این ماشین، متناهی است؟

حل: خیر - چون طول رشته پذیرفته شده، از تعداد حالت‌های ماشین بیشتر است، بنابراین دارای حلقه می‌باشد. ماشینی که دارای حلقه است، تعداد اعضای زبان پذیرش شده آن، نامتناهی می‌باشد.

مثال

پذیرنده حالت متناهی غیر قطعی زیر چه زبانی را می‌پذیرد.



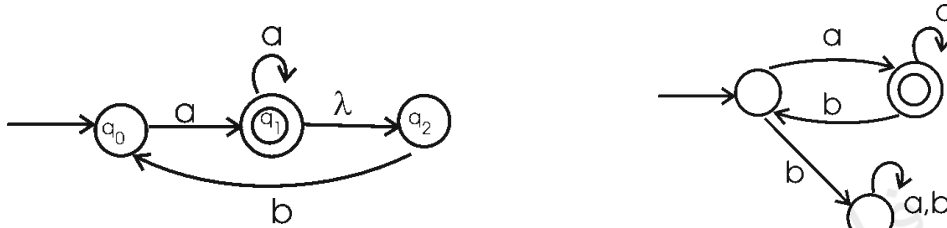
حل: $(b + ab^*a)^* ab^*b(a+b)^*$

هم‌ارزی DFA و NFA

دو ماشین متناهی را هم‌ارز می‌گویند اگر هر دو زبانی یکسان را بپذیرند. چون به ازای هر زبان معمولاً تعداد زیادی پذیرنده وجود دارد، بنابراین هر DFA یا NFA نیز تعداد زیادی پذیرنده هم‌ارز دارد.

مثال

در شکل زیر دو ماشین هم‌ارز نشان داده شده است (شکل سمت راست DFA و سمت چپ NFA است)



چند نکته:

- ۱- به ازای هر زبانی که توسط یک NFA پذیرفته می‌شود، یک DFA هم وجود دارد که آن را می‌پذیرد.
- ۲- کلاس‌های DFA و NFA ها دارای قدرت یکسان می‌باشند.
- ۳- برای هر NFA با هر تعداد دلخواه حالت پایانی، یک DFA با فقط یک حالت پایانی، هم‌ارز با آن NFA وجود دارد.
- ۴- اگر L یک زبان غیر تهی باشد بطوریکه هر w عضو L ، دارای حداقل طول n باشد، آنگاه هر DFA که L را بپذیرد، باید حداقل $n+1$ حالت داشته باشد.

ارتباط گرامر منظم با ماشین متناهی

می‌توان با داشتن انتقالات یک ماشین متناهی، گرامر منظم مربوط به زبان تولید شده توسط ماشین را مشخص کرد (و بر عکس). به طور نمونه انتقال $\delta(q_i, a) = q_j$ به قانون $q_i \rightarrow aq_j$ تبدیل می‌شود.

مثال

گرامر منظمی برای زبان $L = \{w : \text{هر دو زوج هستند} : w\}$ روی $\{a,b\}$ که DFA آن به صورت زیر است، بنویسید. (q_0 حالت شروع و پایانی است).

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

$$\delta(q_1, a) = q_0, \delta(q_1, b) = q_3$$

$$\delta(q_2, a) = q_3, \delta(q_2, b) = q_0$$

$$\delta(q_3, a) = q_2, \delta(q_3, b) = q_1$$

حل: از روی انتقالات ماشین، گرامر را می‌نویسیم:

$$q_0 \rightarrow aq_1 \mid bq_2 \mid \lambda$$

$$q_1 \rightarrow aq_0 \mid bq_3$$

$$q_2 \rightarrow aq_3 \mid bq_0$$

$$q_3 \rightarrow aq_2 \mid bq_1$$

■

مثال

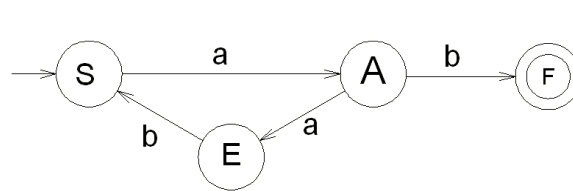
ماشین متناهی بسازید که زبان تولید شده بوسیله گرامر زیر را بپذیرد.

$$S \rightarrow aA$$

$$A \rightarrow abS \mid b$$

حل:

گراف انتقالی با سه راس S و A و F ایجاد می‌کنیم. یالی با برچسب a بین S و A ایجاد می‌نماییم. سپس راس E را به گونه‌ای ایجاد می‌کنیم که مسیری با برچسب ab بین A و S وجود داشته باشد. در نهایت یالی با برچسب b بین A و F ایجاد می‌کنیم تا ماشین شکل زیر بدست آید.



زبان تولید و پذیرفته شده توسط این گرامر، زبان منظم $L((aab)^* ab)$ خواهد بود. ■

مثال

ماشین متناهی متناظر با گرامر منظم زیر را بدست آورید.

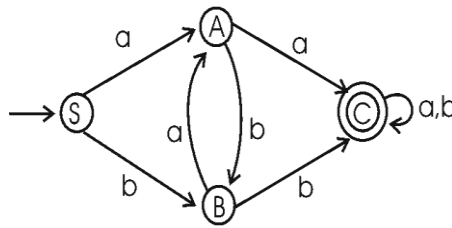
$$S \rightarrow aA \mid bB$$

$$A \rightarrow aC \mid bB \mid a$$

$$C \rightarrow aC \mid bC \mid a \mid b$$

$$B \rightarrow aA \mid bC \mid b$$

حل:



روال تبدیل NFA به DFA

۱- گراف مفروض G_D با راس $\{q_0\}$ را ایجاد کرده و آن راس را بعنوان راس شروع در نظر بگیرید.

۲- تا زمانی که همه یال‌ها در نظر گرفته نشده‌اند، مراحل زیر را تکرار کنید:

الف- هر یک از رئوس $\{q_i, q_j, \dots, q_k\}$ از G_D را در نظر بگیرید که برای $a \in \Sigma$ آن هیچ یالی خارج نشود.

ب- $\delta_N^*(q_i, a), \delta_N^*(q_j, a), \dots, \delta_N^*(q_k, a)$ را محاسبه کنید.

ج- اجتماع همه این δ_N^* ها را تشکیل دهید. در نتیجه مجموعه $\{q_1, q_m, \dots, q_n\}$ بدست می‌آید.

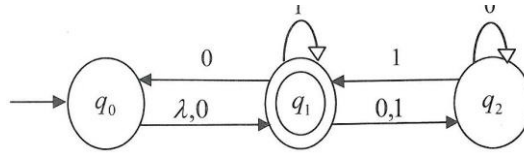
د- در صورت عدم وجود راسی با برچسب $\{q_1, q_m, \dots, q_n\}$ در G_D ، این راس را ایجاد کنید. یالی از $\{q_i, q_j, \dots, q_k\}$ به $\{q_1, q_m, \dots, q_n\}$ به نام a به G_D اضافه کنید.

۳- تمامی حالت‌های G_D که برچسب آن حاوی حداقل یک $q_f \in F_N$ باشد، بعنوان راس پایانی شناخته می‌شود.

۴- اگر nfa، λ را بپذیرد، راس $\{q_0\}$ در G_D نیز، راس پایانی می‌باشد.

مثال

NFA زیر را به DFA تبدیل کنید.



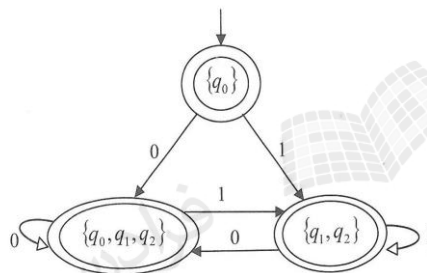
حل:

$$\delta(\{q_0\}, 0) = \{q_0, q_1, q_2\} \quad , \quad \delta(\{q_0\}, 1) = \{q_1, q_2\}$$

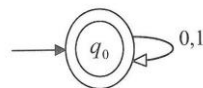
$$\delta(\{q_1, q_2\}, 0) = \{q_0, q_1, q_2\} \quad , \quad \delta(\{q_1, q_2\}, 1) = \{q_1, q_2\}$$

$$\delta(\{q_0, q_1, q_2\}, 0) = \{q_0, q_1, q_2\} \quad , \quad \delta(\{q_0, q_1, q_2\}, 1) = \{q_1, q_2\}$$

بنابراین داریم:



اما nfa حاصل λ را هم می‌پذیرد. پس q_0 هم باید وضعیت نهایی باشد در این صورت dfa حاضر همه رشته‌ها را می‌پذیرد و به طور خلاصه به شکل روبه‌رو در می‌آید:



کاهش تعداد حالات در ماشین‌های متناهی

هر dfa یک زبان منحصر بفرد را تعریف می‌کند، اما عکس این جمله صحیح نیست، یعنی برای یک زبان ممکن است چند dfa وجود داشته باشد. در عمل ممکن است از بین چند dfa که برای یک زبان وجود دارد، یکی را انتخاب کرد. معمولاً این dfa دارای حالات کمتری می‌باشد. در dfa می‌توان حالتی که دسترس پذیر نباشد را حذف کرد و بعضی از حالتها را که ادغام پذیر هستند را با هم ادغام کرد.

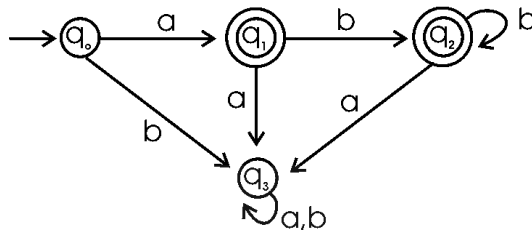
دو حالت q, p یک dfa، در صورتی ادغام پذیر هستند که به ازای هر $w \in \Sigma^*$ داشته باشیم:

$$\delta^*(q, w) \in F \Rightarrow \delta^*(p, w) \in F \quad , \quad \delta^*(q, w) \notin F \Rightarrow \delta^*(p, w) \notin F$$

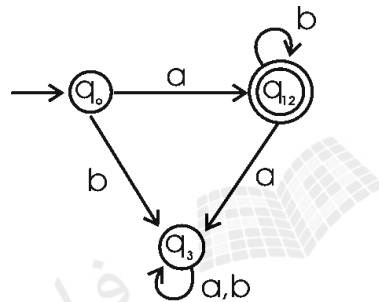
اگر p و q ادغام پذیر بوده و q و r هم ادغام پذیر باشند، آنگاه p و r ادغام پذیر هستند و در نتیجه هر سه حالت هم ادغام پذیر هستند. به عبارتی ادغام پذیر بودن یک رابطه هم ارزی است. (ادغام ناپذیر بودن، اینطور نیست).

مثال

DFA زیر را کمینه نمایید.

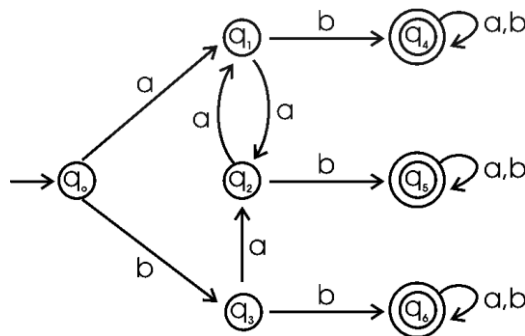


حل: حالت q_1 و q_2 ادغام پذیرند و به جای آنها حالت q_{12} را قرار می‌دهیم:



مثال

DFA زیر را کمینه نمایید.



حل: حالت های q_1 و q_2 و q_3 ادغام پذیرند و به جای آنها حالت q_{123} را قرار می‌دهیم. همچنین حالت های q_4 و q_5 و q_6 ادغام پذیرند و به جای آنها حالت q_{456} را قرار می‌دهیم.



نحوه تشخیص منظم بودن یک زبان

در فصل‌های قبل برای اینکه نشان دهیم یک زبان منظم است، عبارت منظم یا گرامر منظم برای آن ارائه دادیم. یک راه دیگر این است که بتوان برای آن یک ماشین متناهی پیدا کرد.

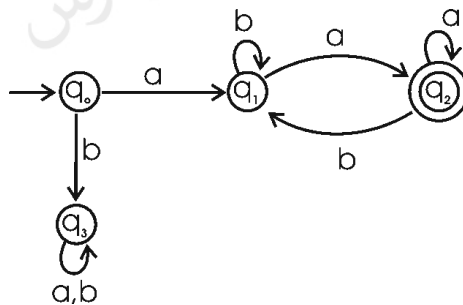
زبان L منظم است اگر و فقط اگر یک DFA مانند M وجود داشته باشد به طوری که $L = L(M)$

فرض کنیم r یک عبارت منظم باشد. آنگاه یک NFA وجود دارد که $L(r)$ را می‌پذیرد. در نتیجه $L(r)$ یک زبان منظم است.

مثال

نشان دهید که زبان $L = \{awa : w \in \{a,b\}^*\}$ منظم است.

حل: برای اثبات منظم بودن زبان داده شده، کافی است که یک DFA برای آن پیدا کنیم. زبان L رشته‌هایی را می‌پذیرد که با حرف a شروع شده و با حرف a نیز تمام می‌شود. بین این دو حرف a می‌تواند هیچ یا تعداد نامحدودی حرف a یا b ظاهر شود.

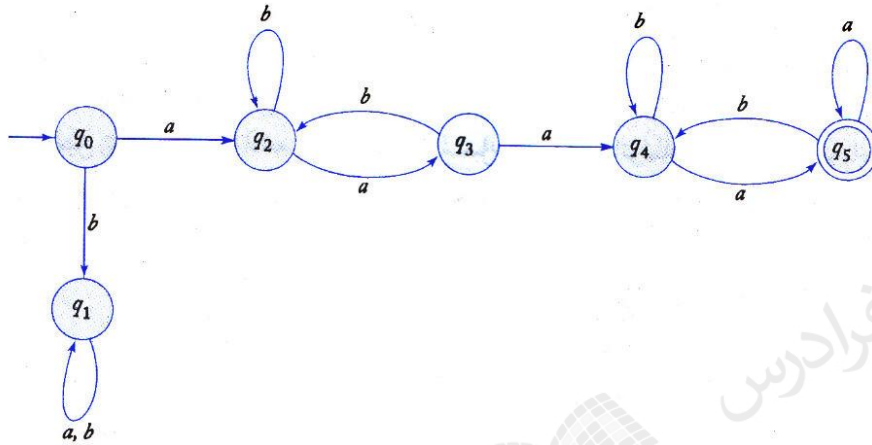


وضعیت q_3 یک تله است و اگر رشته با حرف b شروع شود، ماشین به این حالت می‌رود. ■

مثال

با فرض اینکه $L = \{awa : w \in \{a,b\}^*\}$ ، نشان دهید که زبان L^2 منظم است.

حل: زبان $L^2 = \{aw_1aaw_2a : w_1, w_2 \in \{a,b\}^*\}$ برابر L^2 است. با رسم یک DFA برای این زبان می‌توان نشان داد که L^2 نیز منظم است.

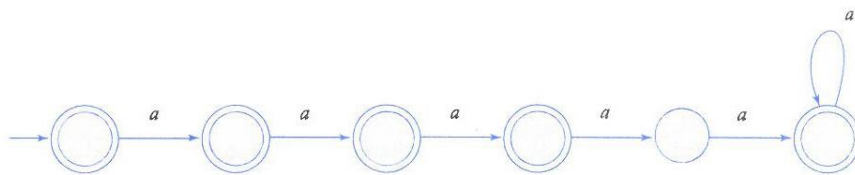


اگر زبان مفروض L منظم باشد، L^2 و L^3 و ... هم منظم خواهد بود.

مثال

نشان دهید که زبان $L = \{a^n : n \geq 0, n \neq 4\}$ منظم است؟ ($\Sigma = \{a\}$)

حل: برای اثبات منظم بودن زبان داده شده، کافی است که یک DFA برای آن پیدا کنیم:

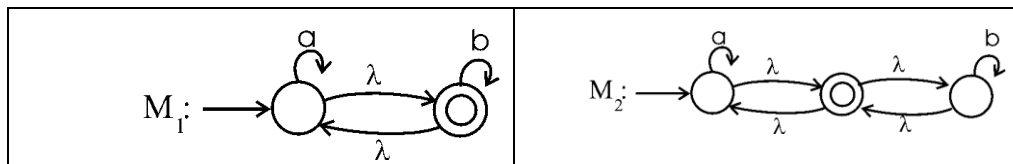


تمرین‌های فصل ۳

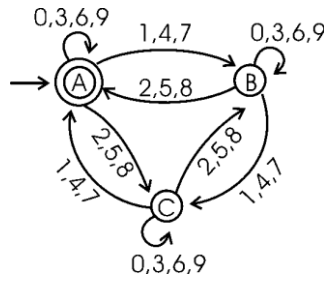
۱- عبارت منظمی معادل زبان هر یک از ماشین‌های زیر را بنویسید.

ب:	الف:
ت:	پ:
ج:	ث:

۲- چه رابطه‌ای بین $L(M_1)$ و $L(M_2)$ است؟

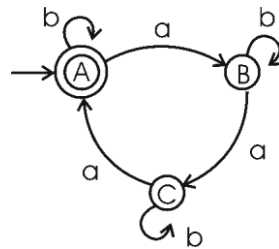


۳- چه رشته‌های توسط ماشین زیر پذیرفته می‌شوند؟

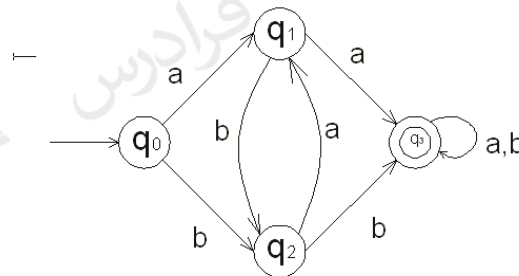


۴- یک DFA برای زبان منظم $(a^*(b+\lambda)a^*)^*$ رسم کنید.

۵- گرامر هم ارز با پذیرنده زیر را بنویسید.



۶- گرامر متناظر با ماشین زیر را بنویسید.



۷- آیا زبان $\{a^n b^m a^k : n+m+k > 3\}$ منظم است؟

۸- کدام یک از زبان‌های زیر منظم هستند؟

$$L_1 = \{a^n : n \text{ یا ضریب سه است یا ضریب پنج}\}$$

$$L_2 = \{a^n : n \text{ ضریب سه است و ضریب پنج نیست}\}$$

$$L_3 = \{a^n : n = 2^k, k \leq 2000\}$$

۹- آیا زبان زیر منظم است؟

$$L = \{w \in (0,1)^* : \text{تعداد } 0 \text{ ها و } 1 \text{ ها برابر مقدار ثابت } n \geq 0 \text{ باشد}\}$$

۱۰- آیا زبان $L = \{a^n : n = i + jk\}$ منظم است؟ (i و k ثابت و $j = 0, 1, 2, \dots$)

۱۱- آیا زبان زیر منظم است؟

A یک DFA است و در مسیر پذیرش w از چند حالت معین A عبور نمی‌شود. $L = \{w \in L(A) : \dots\}$

۱۲- آیا زبان زیر منظم است؟

$$L = \{x^n y^n : x \in (0+1)^*, y \in (0+1)^*, n \geq 0\}$$

۱۳- اگر $m(L)$ تعداد حالات DFA مینیمال متناظر با زبان $L \subseteq \{0,1\}^*$ باشد، $m(L)$ چه رابطه‌ای با $m(L^R)$ دارد؟

۱۴- آیا هر زبان منظمی، معین (قطعی) است؟

۱۵- برای $L(a\phi)$ و $L(\phi^*)$ یک NFA رسم کنید.

۱۶- ثابت کنید که تمام زبانهای متناهی، منظم هستند.

۱۷- منظور از dfa ناقص چیست؟

۱۸- کدام گزاره صحیح است؟

الف- اگر L یک زبان منظم فاقد λ باشد، یک NFA بدون انتقال λ و فقط با یک حالت پایانی وجود دارد که L را می‌پذیرد.

ب- فرض کنیم Γ یک عبارت منظم باشد. آنگاه یک NFA وجود دارد که $L(\Gamma)$ را می‌پذیرد. در نتیجه $L(\Gamma)$ یک زبان منظم است.

پ- به ازای هر NFA با چندین حالت شروع، یک NFA با دقیقاً یک حالت شروع وجود دارد که همان زبان را می‌پذیرد.

۱۹- رشته‌های تولید شده توسط گرامر زیر چه خاصیتی دارند؟

$$S \rightarrow aA \mid bB$$

$$A \rightarrow bbA \mid baB \mid aS \mid b$$

$$B \rightarrow abA \mid aaB \mid bS \mid a$$

پاسخ تمرین فصل ۳

۱- عبارت منظم معادل با هر ماشین برابر است با:

الف- $(0 + 01^*1)^*$

ب- $0^*(0+1)1^*(10^*(0+1)1^*)^*$

پ- $0 + (00 + 1 + 11)0^*0$

ت- $(0+1)(0+10+11)^*$

ث- $(a^+b^+a + b^+a^+b)(a+b)^*$

ج- $(b + ab^*a)^*ab^*b(a+b)^*$

۲- هر دو ماشین، زبان $(a+b)^*$ را می‌پذیرند.

۳- ماشین اعداد بخش پذیر بر ۳ را می‌پذیرد. مانند ۹۷۵، ۶۴۷۱، ۳۱۵

۴- زبان داده شده را می‌توان ساده کرد:

$$(a^*(b+\lambda)a^*)^* = ((a^*b+a^*)a^*)^* = (a^*ba^*+a^*)^* = (a+b)^*$$

بنابراین ماشین به صورت زیر است:



۵- گرامر به صورت زیر است:

$$A \rightarrow bA \mid aB \mid \lambda$$

$$B \rightarrow bB \mid aC$$

$$C \rightarrow bC \mid aA$$

۶- گرامر به صورت زیر است:

$$q_0 \rightarrow aq_1 \mid bq_2$$

$$q_1 \rightarrow bq_2 \mid aq_3 \mid a$$

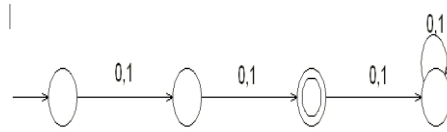
$$q_2 \rightarrow aq_1 \mid bq_3 \mid b$$

$$q_3 \rightarrow aq_3 \mid bq_3 \mid a \mid b$$

۷- بله- چون می‌توان برای آن یک NFA رسم کرد.

۸- همه زبان‌های داده شده منظم هستند، چون برای هر یک از آنها می‌توان یک DFA رسم کرد.

۹- بله- چون مقدار n ثابت است و می‌توان برای آن یک DFA طراحی کرد. به طور مثال اگر $n=2$ ، به کمک DFA زیر می‌توان رشته‌های زبان را پذیرفت:



۱۰- بله- می‌توان dfa برای آن رسم کرد. چون i و k ثابت هستند، بنابراین تعداد وضعیت‌ها مشخص و عددی ثابت و متناهی می‌باشد. ($a^{i+jk} = a^i . a^{jk}$)

۱۱- بله- چون برای آن یک DFA طراحی شده است.

۱۲- بله- چند رشته از زبان:

$$L = \{\lambda, 0^5 1^5, 1^2 0^2, (011)^2 (10)^2, (10)^3 (0)^3, \dots\}$$

۱۳- اگر ماشین DFA ای که زبان L را می‌پذیرد، به ماشینی که معکوس زبان L (یعنی L^R) را بپذیرد، تبدیل کنیم، ممکن است ماشین به NFA تبدیل شود. بنابراین برای قطعی کردن ماشین، تعداد حالات یعنی $m(L)$ به صورت توانی زیاد می‌شود. بنابراین $m(L) \leq 2^{m(L^R)}$.

۱۴- بله- چون می‌توان برای آن یک DFA رسم کرد.

۱۵- ماشین‌های NFA به صورت زیر است:

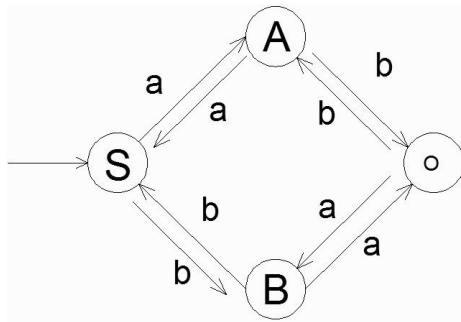


۱۶- کافی است یک NFA برای آنها طراحی کنیم. یک q_0 مشترک برای همه رشته‌های عضو وجود دارد و سپس هر رشته به صورت مجزا برای خود یک وضعیت نهایی دارد که مسیر رسیدن با این وضعیت نهایی معادل خود رشته خواهد بود.

۱۷- nfa ای که در آن هیچ انتقال λ وجود ندارد و به ازای هر $q \in Q$ و هر $a \in \Sigma$ ، $\delta(q, a)$ حاوی حداکثر یک عضو باشد را dfa ناقص می‌گویند. در این dfa، برای برخی انتقال‌ها نمی‌توان حرکتی کرد.

۱۸- همه گزاره‌ها صحیح هستند.

۱۹- شامل تعداد فردی a و تعداد فردی b هستند. البته می‌توان FA را رسم کرد و با توجه به آن زبان را تشخیص داد.



فصل ۴:

زبان و گرامر مستقل از متن

برای ساخت برنامه‌های قدرتمندتر باید تا حدی از قید محدودیت‌های موجود در گرامرهای منظم رها شویم. از زبان‌های مستقل از متن در طراحی زبان‌های برنامه‌سازی و ساخت کامپایلر استفاده می‌شود.

گرامر مستقل از متن

گرامر مفروض $G = (V, T, S, P)$ در صورتی مستقل از متن خوانده می‌شود که تمام قوانین P به فرم $A \rightarrow x$ باشند که در آن $A \in V$ و $x \in (V \cup T)^*$. به طور کلی شرط مستقل از متن بودن این است که در سمت چپ قوانین، فقط یک متغیر وجود داشته باشد.

زبان L مستقل از متن نامیده می‌شود، اگر و تنها اگر گرامر مستقل از متن G وجود داشته باشد بطوریکه $L = L(G)$.

گرامرهای منظم، مستقل از متن نیز هستند. هر زبان منظمی، یک زبان مستقل از متن نیز می‌باشد.

خانواده زبان‌های منظم یکی از زیر مجموعه‌های محض خانواده زبان‌های مستقل از متن هستند.

مثال

گرامر تولید کننده زبان $L = \{a^n b^n : n \geq 1\}$ را بنویسید.

حل: توسط این گرامر رشته‌های شروع شونده با a که تعداد a و b در آنها برابر است تولید می‌شود.

$$S \rightarrow aSb \mid ab$$

مثال

گرامر تولید کننده زبان $L = \{a^n b^n : n \geq 0\}$ را بنویسید.

حل: $S \rightarrow aSb \mid \lambda$

مثال

گرامر تولید کننده زبان $L = \{a^n b^{n+1} : n \geq 0\}$ را بنویسید.حل: $S \rightarrow aSb | b$.

روش دوم: این زبان با زبان $\{a^n b^n : n \geq 0\}$ فقط در یک b تفاوت دارد ($a^n b^n b$). بنابراین چون گرامر $\{a^n b^n : n \geq 0\}$ را با قانون $S \rightarrow aSb | \lambda$ می‌توان تولید کرد، خواهیم داشت:

$$S \rightarrow Xb$$

$$X \rightarrow aXb | \lambda$$

مثال

گرامر تولید کننده زبان $L = \{a^{n+1} b^n : n \geq 0\}$ را بنویسید.حل: $S \rightarrow aSb | a$.روش دوم: این زبان با $\{a^n b^n : n \geq 0\}$ فقط در یک a تفاوت دارد: $aa^n b^n$.

$$S \rightarrow aX$$

$$X \rightarrow aXb | \lambda$$

مثال

گرامر تولید کننده زبان $L = \{a^{n+3} b^n : n \geq 0\}$ را بنویسید.حل: $S \rightarrow aSb | aaa$.روش دوم: این مثال با $\{a^n b^n : n \geq 0\}$ فقط در سه تا a تفاوت دارد: $aaaa^n b^n$.

$$S \rightarrow aaaX$$

$$X \rightarrow aXb | \lambda$$

مثال

گرامر تولید کننده زبان $L = \{a^n b^{n-3} : n \geq 3\}$ را بنویسید.

حل: زبان را می‌توان به صورت $\{a^{k+3} b^k : k \geq 0\}$ نیز نشان داد. پس جواب، مانند مثال قبل است. ■

مثال

گرامر تولید کننده زبان $L = \{a^n b^{2n} : n \geq 0\}$ را بنویسید.حل: $S \rightarrow aSbb \mid \lambda$

روش دوم:

$$S \rightarrow aSA \mid \lambda$$

$$A \rightarrow bB$$

$$B \rightarrow b$$

مثال

گرامر تولید کننده زبان $L = \{a^{n+2} b^{3n} : n \geq 0\}$ را بنویسید.

حل: $S \rightarrow aSbbb \mid aa$ ■

مثال

گرامر تولید کننده زبان $L = \{a^n bc^{2n} : n \geq 0\}$ را بنویسید.

حل: $S \rightarrow aSc \mid b$ ■

مثال

گرامر تولید کننده زبان $L = \{a^n b^n : n \geq 0\}$ را بنویسید. (n مضربی از ۳ نیست)

حل: $S \rightarrow aaaSbbb \mid aabb \mid ab$ ■

مثال

گرامر تولید کننده زبان $L = \{a^{2n+2}b^{n+2} : n \geq 0\}$ را بنویسید.

حل:

$$S \rightarrow aXb$$

$$X \rightarrow aaXb \mid ab$$

مثال

گرامر تولید کننده زبان $L = \{a^n b c^m : n \neq m\}$ را بنویسید.

$$S \rightarrow aSc \mid aS \mid Sc \mid ab \mid bc$$

مثال

گرامر تولید کننده زبان $L = \{a^n b^n a^k b^k : n \geq 0, k \geq 0\}$ را بنویسید.

حل:

$$S \rightarrow MM$$

$$M \rightarrow aMb \mid \lambda$$



مثال

گرامر تولید کننده زبان $L = \{a^n b^n : n \geq 0\} \cup \{b^n a^n : n \geq 0\}$ را بنویسید.حل: کافی است با دستور $S \rightarrow S_1 \mid S_2$ ، گرامر $\{a^n b^n\}$ و گرامر $\{b^n a^n\}$ را با هم ترکیب کرد:

$$S \rightarrow S_1 \mid S_2$$

$$S_1 \rightarrow aS_1 b \mid \lambda$$

$$S_2 \rightarrow bS_2 a \mid \lambda$$



مثال

فرض کنید $L = \{a^n b^n : n \geq 0\}$ باشد. گرامری برای زبان L^* بنویسید.

حل:

$$S \rightarrow SA \mid \lambda$$

$$A \rightarrow aAb \mid \lambda$$



مثال

گرامری برای زبان $L = \{a^n b^k : 2n \leq k \leq 3n, n \geq 0, k \geq 0\}$ بنویسید.

حل: با $S \rightarrow aSbb \mid \lambda$ زبان $\{a^n b^{2n} : n \geq 0\}$ و با $S \rightarrow aSbbb \mid \lambda$ زبان $\{a^n b^{3n} : n \geq 0\}$ را تولید می‌کنیم:

$$S \rightarrow aSbb \mid aSbbb \mid \lambda$$



مثال

با توجه به مثال قبل، نحوه تولید رشته $a^3 b^7$ را مشخص کنید.

حل:

$$S \Rightarrow aSbb \Rightarrow aaSbbb \Rightarrow aaaSbbbbbb \Rightarrow aaabbbbbbb$$

دو بار از $S \rightarrow aSbb$ و یکبار از $S \rightarrow aSbbb$ و در آخر از $S \rightarrow \lambda$ استفاده می‌کنیم.



مثال

گرامر تولید کننده زبان $L = \{a^n b^k : k > n\}$ را بنویسید. ($n \geq 0, k \geq 0$)

حل: توسط $X \rightarrow aXb$ به تعداد مساوی a و b و به کمک $B \rightarrow bB \mid b$ ، یک یا چند b تولید می‌شود:

$$\begin{aligned} S &\rightarrow XB \\ X &\rightarrow aXb \mid \lambda \\ B &\rightarrow bB \mid b \end{aligned}$$



مثال

گرامر تولید کننده زبان $L = \{a^n b^k : n > k\}$ را بنویسید. ($n \geq 0, k \geq 0$)

حل: توسط $A \rightarrow aA \mid a$ ، یک یا چند a و توسط $X \rightarrow aXb$ به تعداد مساوی a و b تولید می‌شود:

$$\begin{aligned} S &\rightarrow AX \\ A &\rightarrow aA \mid a \\ X &\rightarrow aXb \mid \lambda \end{aligned}$$



مثال

یک گرامر مستقل از متن برای زبان $L = \{a^n b^k : n \neq k\}$ بنویسید.

حل: کافی است دو حالت $n > k$ و $k > n$ که در دو مثال قبیل بررسی شد را با هم ترکیب کرد.

$$\begin{aligned} S &\rightarrow AX \mid XB \\ X &\rightarrow aXb \mid \lambda \\ A &\rightarrow aA \mid a \\ B &\rightarrow bB \mid b \end{aligned}$$



مثال

گرامری بنویسید که زبان $L = \{a^n b^n c^k : n > 0, k > 0\}$ را تولید کند.

حل: توسط $X \rightarrow aXb \mid ab$ زبان $\{a^n b^n : n > 1\}$ و توسط $Y \rightarrow cY \mid c$ زبان $\{c^k : k > 0\}$ را تولید می‌کنیم. با دستور $S \rightarrow XY$ این دو را کنار هم قرار می‌دهیم.

$$S \rightarrow XY$$

$$X \rightarrow aXb \mid ab$$

$$Y \rightarrow cY \mid c$$



مثال

گرامری بنویسید که زبان $L = \{a^k b^n c^n : n > 0, k > 0\}$ را تولید کند.

حل: توسط قانون $X \rightarrow aX \mid a$ زبان $\{a^k : k > 0\}$ و توسط $Y \rightarrow bYc \mid bc$ زبان $\{b^n c^n : n > 0\}$ را تولید می‌کنیم. با دستور $S \rightarrow XY$ این دو را کنار هم قرار می‌دهیم.

$$S \rightarrow XY$$

$$X \rightarrow aX \mid a$$

$$Y \rightarrow bYc \mid bc$$

مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : m \leq k\}$ بنویسید.

$$(n \geq 0, m \geq 0, k \geq 0)$$

حل: توسط $A \rightarrow aA \mid \lambda$ زبان a^* و توسط دیگر قوانین $\{b^m c^k : m \leq k\}$ را می‌سازیم.

$$S \rightarrow AX$$

$$A \rightarrow aA \mid \lambda$$

$$X \rightarrow bXc \mid Y$$

$$Y \rightarrow Yc \mid \lambda$$



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : k = n + m\}$ بنویسید. ($n \geq 0, m \geq 0, k \geq 0$)

حل: می‌توان زبان را به صورت $a^n b^m c^m c^n$ نشان داد. قسمت $b^m c^m$ با X تولید می‌شود.

$$S \rightarrow aSc \mid X$$

$$X \rightarrow bXc \mid \lambda$$



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : n = k + m\}$ بنویسید. ($n \geq 0, m \geq 0, k \geq 0$)

حل: می‌توان زبان را به صورت $a^k a^m b^m c^k$ نشان داد. قسمت $a^m b^m$ با X تولید می‌شود.

$$S \rightarrow aSc \mid X$$

$$X \rightarrow aXb \mid \lambda$$



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : m = n + k\}$ بنویسید. ($n \geq 0, m \geq 0, k \geq 0$)

حل: می‌توان زبان را به فرم $a^n b^n b^k c^k$ نشان داد. قسمت $a^n b^n$ با X و قسمت $b^k c^k$ با Y تولید می‌شود.

$$S \rightarrow XY$$

$$X \rightarrow aXb \mid \lambda$$

$$Y \rightarrow bYc \mid \lambda$$



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : k = |n - m|\}$ بنویسید. ($n \geq 0, m \geq 0, k \geq 0$)

حل: شرط به معنی $k = n - m$ و $k = -(n - m)$ است. این دو حالت را می‌توان به ترتیب به صورت

$n = k + m$ و $m = n + k$ نشان داد. این حالتها در مثالهای قبل بررسی شد و باید با هم ترکیب شوند.



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : k > n + m\}$ بنویسید. ($n \geq 0, m \geq 0, k \geq 1$)

حل: کافی است زبان را به صورت $a^n b^m c^+ c^m c^n$ نشان داد. قسمت $b^m c^+ c^m$ با X تولید می‌شود.

$$S \rightarrow aSc \mid X$$

$$X \rightarrow bXc \mid cY$$

$$Y \rightarrow cY \mid \lambda$$



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : k < n + m\}$ بنویسید. ($n \geq 0, m \geq 0, k \geq 0$)

حل: زبان را به صورت $a^n (a+b)^+ b^m c^m c^n$ نشان می‌دهیم.

$$S \rightarrow aSc \mid aS \mid aX \mid bX$$

$$X \rightarrow bXc \mid bX \mid \lambda$$



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : k \neq n + m\}$ بنویسید. ($n \geq 0, m \geq 0, k \geq 0$)

حل: شرط به معنی $k > n + m$ یا $k < n + m$ است. در نتیجه این دو حالت که در مثالهای قبل بررسی شد را با هم ترکیب می‌کنیم.



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^m c^k : k = n.m\}$ بنویسید.

حل: نمی‌توان یک گرامر مستقل از متن برای این زبان نوشت. پس این زبان مستقل از متن نیست.



مثال

گرامر مستقل از متنی برای زبان $L = \{(ab)^n(cd)^n : n \geq 1\}$ بنویسید.

$$S \rightarrow aXd$$

$$X \rightarrow bSc \mid bc$$



مثال

گرامر مستقل از متنی برای زبان $L = \{(ab)^n(cde)^n : n \geq 1\}$ بنویسید.

$$S \rightarrow aXde$$

$$X \rightarrow bSc \mid bc$$



مثال

گرامر مستقل از متنی برای زبان $L = \{aa(bc)^n be(dde)^n : n \geq 0\}$ بنویسید.

$$S \rightarrow aaX$$

$$X \rightarrow bYe$$

$$Y \rightarrow cXdd \mid \lambda$$



مثال

گرامر مستقل از متنی برای زبان $L = \{ab(bbaa)^n bba(ba)^n \mid n \geq 0\}$ بنویسید.

حل: در جملات این زبان، تعداد bbaa با ba برابر است.

$$S \rightarrow abX$$

$$X \rightarrow bbYa$$

$$Y \rightarrow aaXb \mid \lambda$$



مثال

گرامر مستقل از متنی برای زبان $L = \{a^n b^k c^k d^n : n \geq 0, k \geq 1\}$ بنویسید.

$$S \rightarrow aSd \mid bXc \mid \lambda$$

$$X \rightarrow bXc \mid \lambda$$

مثال

گرامر تولید کننده زبان $L = \{w : n_a(w) = n_b(w)\}$ را بنویسید.حل: زبان تولید شده شامل رشته‌هایی با تعداد a و b های برابر است. (جملات با a یا b شروع می‌شوند).

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



مثال

گرامر تولید کننده زبان $L = \{w : n_a(w) = n_b(w) + 1\}$ را بنویسید.حل: تعداد a ها یکی بیشتر از تعداد b ها می باشد.

$$S \rightarrow XaX$$

$$X \rightarrow XX \mid aXb \mid bXa \mid \lambda$$

به طور نمونه رشته $ababa$ را تولید می کند که شامل سه تا a و دو تا b است. نحوه تولید:

$$S \Rightarrow XaX \Rightarrow aXbaX \Rightarrow abaX \Rightarrow ababXa \Rightarrow ababa$$



مثال

گرامر تولید کننده زبان $L = \{w : n_a(w) > n_b(w)\}$ را بنویسید.حل: تعداد a ها بیشتر از تعداد b ها می باشد.

$$S \rightarrow SS \mid aSb \mid bSa \mid aS \mid Sa \mid a$$



مثال

گرامر تولید کننده زبان $L = \{w : |w| \bmod 3 \geq |w| \bmod 2\}$ را بنویسید. ($\Sigma = \{a\}$)حل: طول رشته باید $6k$ ، $3k+1$ ، یا $3k+2$ باشد.

$$S \rightarrow aX \mid aaX \mid Y$$

$$X \rightarrow aaaX \mid \lambda$$

$$Y \rightarrow aaaaaaY \mid \lambda$$



مثال

گرامر تولید کننده زبان $L = \{w \in \{a,b\}^* : ww^R\}$ را بنویسید.■ حل: جمله‌های این زبان مانند **abba** است که نیمه دوم، معکوس نیمه اول است. $S \rightarrow aSa | bSb | \lambda$

مثال

گرامر تولید کننده زبان $L = \{w \in \{a,b\}^+ : ww^R\}$ را بنویسید.حل: $S \rightarrow aSa | bSb | aa | bb$

■

مثال

گرامر تولید کننده زبان $\{w \in \{a,b\}^* : w = w^R\}$ را بنویسید.حل: جمله‌های این گرامر مانند **aba** است که از هر دو طرف یکسان خوانده می‌شوند. $S \rightarrow aX | bY | \lambda$ $X \rightarrow Sa | \lambda$ $Y \rightarrow Sb | \lambda$

مثال

گرامر تولید کننده زبان $L = \{a^n ww^R b^n : n \geq 1, w \in \{a,b\}^*\}$ را بنویسید. $S \rightarrow aSb | aMb$ $M \rightarrow aMa | bMb | \lambda$

■

مثال

گرامر تولید کننده زبان $L = \{uvw^R : u, v, w \in \{a,b\}^+, |u| = |w| = 2\}$ را بنویسید. $S \rightarrow XY$ $X \rightarrow aa | bb | ab | ba$ $Y \rightarrow aYa | bYb | aXa | bXb$

■

مثال

گرامر مستقل از متن زیر چه زبانی را تولید می‌کند؟

$$S \rightarrow AB \mid \lambda$$

$$A \rightarrow 1A \mid S$$

$$B \rightarrow 0B \mid S$$

حل: زبان گرامر داده شده، زبان منظم $(1^*0^*)^+$ است. گرامر فوق یک گرامر مستقل از متن است که زبان منظم تولید می‌کند. ■

مثال

گرامر مستقل از متن زیر چه زبانی را تولید می‌کند؟

$$S \rightarrow XYZ \mid \lambda$$

$$X \rightarrow aX \mid S$$

$$Y \rightarrow bY \mid S$$

$$Z \rightarrow cZ \mid S$$

حل: زبان گرامر داده شده، زبان منظم $(a^*b^*c^*)^+$ است. گرامر فوق یک گرامر مستقل از متن است که زبان منظم تولید می‌کند. ■

گرامر منظم فقط زبان منظم تولید می‌کند، اما گرامر مستقل از متن علاوه بر زبان مستقل از متن، می‌تواند زبان منظم هم تولید کند.

گرامر ساده (S-گرامر)

گرامر مستقل از متن $G = (V, T, S, P)$ در صورتی گرامر ساده نامیده می‌شود که تمامی قوانین آن به فرم $A \rightarrow aX$ باشند که در آن $X \in V^*$ و $a \in T$ و $A \in V$ و هر زوج (A, a) حداکثر یک بار در P وجود داشته باشد.

مثال

گرامر زیر یک گرامر ساده نمی باشد، چون زوج (S, a) در دو قانون ۱ و ۳ وجود دارد.

$$S \rightarrow aS$$

$$S \rightarrow bSS$$

$$S \rightarrow aSS$$

$$S \rightarrow c$$



مثال

یک s-گرامر برای $L = \{a^n b^n : n \geq 1\}$ بنویسید.

$$S \rightarrow aA$$

$$A \rightarrow aAB \mid b$$

$$B \rightarrow b$$



اگر G یک گرامر ساده باشد، آنگاه هر رشته w عضو $L(G)$ را می توان با مجموعه عملیات های متناسب با $|w|$ تجزیه کرد.

بسیاری از ویژگی های زبان های برنامه سازی، بوسیله گرامرهای ساده قابل توصیف هستند. در کامپایلرها بیشتر از گرامرهای LL و RR استفاده می شود.

بسته بودن زبان های مستقل از متن

خانواده زبان های مستقل از متن تحت اجتماع، الحاق، بستار ستاره ای، معکوس و هم ریختی بسته است و تحت اشتراک، مکمل گیری و تفاضل بسته نیست.

مثال

نشان دهید که خانواده زبان‌های مستقل از متن، تحت اشتراک بسته نیستند.

حل: دو زبان L_1 و L_2 مستقل از متن هستند، چون برای آنها می‌توان گرامر مستقل از متن نوشت:

$$L_1 = \{a^n b^n c^m : n \geq 0, m \geq 0\}$$

$$L_2 = \{a^n b^m c^m : n \geq 0, m \geq 0\}$$

اما اشتراک این دو زبان یعنی $\{a^n b^n c^n : n \geq 0\}$ ، مستقل از متن نیست.

■

✍ اگر L_1 مستقل از متن و L_2 یک زبان منظم باشد، آنگاه $L_1 \cap L_2$ مستقل از متن است. این خاصیت، بسته بودن تحت اشتراک منظم خوانده می‌شود.

مثال

آیا زبان $L = \{a^n b^n : n \geq 0, n \neq 3\}$ مستقل از متن است؟

حل: می‌توان زبان L را به صورت $L = \{a^n b^n : n \geq 0\} \cap \overline{L_1}$ نوشت که $L_1 = \{a^3 b^3\}$ است. بنابراین زبان L از اشتراک یک زبان مستقل از متن با زبان منظم تشکیل شده که با توجه به نکته قبل، زبان L مستقل از متن است. (علت منظم بودن $\overline{L_1}$: زبان L_1 متناهی است، بنابراین منظم نیز هست. از طرفی زبان‌های منظم تحت مکمل‌گیری بسته هستند.) ■

مثال

آیا زبان $L = \{w \in \{a, b, c\}^* : n_a(w) = n_b(w) = n_c(w)\}$ ، مستقل از متن است؟

حل: خیر- اشتراک این زبان با زبان منظم $L(a^* b^* c^*)$ برابر $\{a^n b^n c^n : n \geq 0\}$ می‌باشد که می‌دانیم مستقل از متن نمی‌باشد. بنابراین L مستقل از متن نیست. (با توجه به قضیه اشتراک منظم) ■

مثال

آیا مکمل زبان $L = \{w \in \{a,b,c\}^* : n_a(w) = n_b(w) = n_c(w)\}$ ، مستقل از متن است؟

حل: بله - مکمل این زبان از اجتماع چهار حالت زیر تشکیل شده:

$$n_a(w) > n_c(w) - ۴ \quad n_a(w) < n_c(w) - ۳ \quad n_a(w) > n_b(w) - ۲ \quad n_a(w) < n_b(w) - ۱$$

قبلا نشان داده شد که تمامی این چهار حالت مستقل از متن هستند (برای آنها گرامر مستقل از متن نوشته شد). از آنجا که زبان‌های مستقل از متن تحت اجتماع بسته هستند، اجتماع این چهار حالت نیز مستقل از متن است. ■

مثال

نشان دهید زبان‌های مستقل از متن تحت متمم بسته نیستند.

حل: زبان‌های زیر و متمم آنها مستقل از متن هستند:

$$L_1 = \{a^n b^n c^m : n \geq 0, m \geq 0\} \quad L_2 = \{a^n b^m c^m : n \geq 0, m \geq 0\}$$

بنابراین $L = \overline{L_1} \cup \overline{L_2}$ مستقل از متن است. متمم زبان L برابر است با:

$$\overline{L} = \overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2$$

حاصل $L_1 \cap L_2$ برابر $\{a^n b^n c^n : n \geq 0\}$ است که مستقل از متن نیست. ■

مثال

فرض کنید $L = \{a^n b^n : n \geq 0\}$ ، باشد که زبانی مستقل از متن است. آیا زبان L^* نیز مستقل از متن است؟

حل: بله - چون می‌توان برای آن یک گرامر مستقل از متن به صورت زیر نوشت:

$$S \rightarrow SA \mid \lambda$$

$$A \rightarrow aAb \mid \lambda$$

■

برای اینکه ثابت کنیم یک زبان مستقل از متن نیست، کافی است حاصل اجتماع آن با یک زبان مستقل از متن، مستقل از متن نباشد. چون زبان‌های مستقل از متن نسبت به عمل اجتماع بسته هستند.

اگر L_1 مستقل از متن و L_2 یک زبان منظم باشد، آنگاه $L_1 - L_2$ مستقل از متن است. این خاصیت، بسته بودن تحت تفاضل منظم خوانده می‌شود.

اگر زبانی و متمم آن هر دو مستقل از متن باشند، آنگاه آن زبان لزوماً منظم نیست.

لم تزریق برای زبان‌های مستقل از متن

به کمک لم تزریق می‌توان تشخیص داد که یک زبان مستقل از متن نیست.

لم تزریق: فرض کنید L یک زبان مستقل از متن نامتناهی باشد. آنگاه عدد صحیح و مثبت m وجود دارد، بطوریکه هر w متعلق به L با فرض $|w| \geq m$ را می‌توان به صورت $w = uvxyz$ با شرایط $|vxy| \leq m$ و $|vy| \geq 1$ چنان تجزیه کرد که به ازای هر $i = 0, 1, 2, \dots$ داشته باشیم: $uv^i xy^i z \in L$

مثال

به کمک لم تزریق، نشان دهید که زبان $L = \{a^n b^n c^n : n > 0\}$ مستقل از متن نیست.

حل: فرض کنیم که L مستقل از متن باشد. حال رشته $w = a^n b^n c^n$ متعلق به L را به ۵ قسمت تجزیه کرد:

$$x = a^n, y = b, z = b^{n-2}, u = b, v = c^n$$

حال باید به ازای تمام $i \geq 0$ ، رشته $uv^i xy^i z \in L$ باشد، ولی به ازای $i=2$ این چنین

$$a^n b^i b^{n-2} b^i c^n = a^n b^2 b^{n-2} b^2 c^n = a^n b^{n+2} c^n \notin L$$

تذکر مهم: با اعمال قوانین لم تزریق روی زبان $L = \{a^n b^n : n > 0\}$ ، متوجه می‌شویم که به ازای هر مقدار i ، رشته تزریق شده در L است. از این موضوع نمی‌توان نتیجه گرفت که L مستقل از متن است و فقط می‌توان گفت که از لم تزریق نتوانستیم نتیجه‌ای بگیریم.

مثال

توسط لم تزریق می‌توان نشان داد که زبانهای زیر مستقل از متن نمی‌باشند:

$\{ww : w \in \{a,b\}^*\}$	$\{a^n b^m : n \text{ اول است یا } m \text{ اول است}\}$
$\{ww^R w : w \in \{a,b\}^*\}$	$\{a^n b^m : n \text{ اول است و } m \text{ اول نیست}\}$
$\{w : n_a(w) < n_b(w) < n_c(w)\}$	$\{a^n b^m : m, n \text{ هر دو اول هستند}\}$
$\{w : n_a(w) / n_b(w) = n_c(w)\}$	$\{a^n b^m : n = m^2\}$
$\{w : n_a(w) = n_b(w) = n_c(w)\}$	$\{a^n b^m : n \leq m^2\}$
$\{w \in \{a,b,c\}^* : n_a^2(w) + n_b^2(w) = n_c^2(w)\}$	$\{a^n b^m c^k : k = mn\}$
$\{a^n : n \text{ یک عدد اول است}\}$	$\{a^n b^n c^m : n \neq m\}$
$\{a^{n^1} : n > 0\}$	$\{a^n b^m c^k : k > n, k > m\}$
$\{a^{n^2} : n \geq 0\}$	$\{a^n b^m c^k : n < m, n \leq k \leq m\}$
$\{a^{nm} : m, n \text{ هر دو اول هستند}\}$	
$\{a^n b^m a^n b^m : n \geq 0, m \geq 0\}$	

گرامر خطی

گرامر خطی، گرامر مستقل از متنی است که در سمت راست تمام قواعد آن، حداکثر یک متغیر وجود داشته باشد. زبان مستقل از متن L در صورتی خطی خوانده می‌شود که گرامر مستقل از متن خطی G وجود داشته باشد، بطوریکه $L = L(G)$ باشد.

مثال

آیا زبان $\{a^n b^m : m \leq n \leq 2m - 1\}$ خطی است؟

حل: بله - چون می‌توان یک گرامر خطی برای آن نوشت:

$$S \rightarrow aAb \mid aaBb$$

$$A \rightarrow aAb \mid \lambda$$

$$B \rightarrow aaBb \mid aBb \mid ab \mid b$$



مثال

آیا زبان $\{a^n b^n c^m : n \geq 0, m \geq 0\}$ خطی است؟

حل: بله - چون می‌توان یک گرامر خطی برای آن نوشت:

$$S \rightarrow Sc \mid aAb \mid \lambda$$

$$A \rightarrow aAb \mid \lambda$$



مثال

زبان $\{a^n b^n : n \geq 0\}$ خطی است، چون می‌توان یک گرامر خطی برای آن نوشت. ■

مثال

نشان دهید زبان‌های خطی تحت اشتراک بسته نمی‌باشند.

حل: دو زبان زیر خطی هستند، چون می‌توان برای آنها یک گرامر خطی نوشت:

$$L_1 = \{a^n b^n c^m : n \geq 0, m \geq 0\}$$

$$L_2 = \{a^n b^m c^m : n \geq 0, m \geq 0\}$$

اشتراک این دو زبان برابر است با:

$$L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\}$$

این زبان خطی نیست. (حتی مستقل از متن هم نیست)



چند نکته:

- ۱- تمامی زبان‌های خطی، مستقل از متن هستند. اما راجع به حالت عکس، نمی‌توان با قاطعیت حرفی زد.
- ۲- خانواده زبان‌های خطی یکی از زیر مجموعه‌های مناسب خانواده زبان‌های مستقل از متن هستند.
- ۳- خانواده زبان‌های خطی، تحت اجتماع و هم‌ریختی و معکوس بسته است اما تحت اشتراک و الحاق بسته نیست.
- ۴- اگر L_1 خطی و L_2 منظم باشد، آنگاه $L_1 L_2$ یک زبان خطی است.

لم تزریق برای زبان‌های خطی:

فرض کنید L یک زبان خطی نامتناهی باشد. آنگاه عدد صحیح و مثبت m وجود دارد، بطوریکه هر w متعلق به L با فرض $|w| \geq m$ را می‌توان به صورت $w = uvxyz$ با شرایط $|uvyz| \leq m$ و $|vy| \geq 1$ چنان تجزیه کرد که به ازای هر $i = 0, 1, 2, \dots$ داشته باشیم: $uv^i xy^i z \in L$

مثال

به کمک لم تزریق می‌توان نشان داد که زبان $\{w \in \{a, b\}^* : n_a(w) = n_b(w)\}$ خطی نمی‌باشد. ■

تمرین فصل ۴

۱- گرامر G و زبان L_1 به صورت زیر مفروض است. آیا $L_1 = L(G)$ ؟

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

$$L_1 = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}$$

۲- آیا گرامرهای زیر هم ارز می‌باشند؟

$$G_1: S \rightarrow aSb \mid bSa \mid SS \mid a$$

$$G_2: S \rightarrow aSb \mid bSa \mid a$$

۳- گرامرهای زیر چه زبانی را تولید می‌کنند؟

الف	ب	پ	ت
$S \rightarrow AB$ $A \rightarrow xAy \mid \lambda$ $B \rightarrow zB \mid \lambda$	$S \rightarrow AB$ $A \rightarrow xAy \mid \lambda$ $B \rightarrow yBz \mid \lambda$	$S \rightarrow xSz \mid A$ $A \rightarrow yAz \mid \lambda$	$S \rightarrow Sz \mid Az$ $A \rightarrow xAy \mid xy$

۴- گرامرهای زیر چه زبانی را تولید می‌کنند؟

الف	ب	پ	ت
$S \rightarrow aSb \mid a \mid b$	$S \rightarrow AB$ $A \rightarrow xA \mid \lambda$ $B \rightarrow yBz \mid \lambda$	$S \rightarrow AM \mid MB$ $M \rightarrow aMb \mid \lambda$ $A \rightarrow aA \mid a$ $B \rightarrow bB \mid b$	$S \rightarrow aAb \mid bBa$ $A \rightarrow aaAb \mid ab$ $B \rightarrow bBa \mid a$

۵- گرامرهای زیر چه زبانی را تولید می‌کنند؟

الف	ب	پ	ت
$S \rightarrow aA$ $A \rightarrow bBaa$ $B \rightarrow ccAd \mid \lambda$	$S \rightarrow abA$ $A \rightarrow cdBb$ $B \rightarrow aAba \mid \lambda$	$S \rightarrow 0S1 \mid 1S0 \mid AA$ $A \rightarrow 0A \mid A1 \mid \lambda$	$S \rightarrow aSb \mid bY \mid Ya$ $Y \rightarrow aY \mid bY$

۶- گرامر زیر، چه زبانی را توصیف می‌کند؟

$$\begin{aligned} S &\rightarrow ABC \\ A &\rightarrow 0A1 \mid \lambda \\ B &\rightarrow 1B \mid 1 \\ C &\rightarrow 1C0 \mid \lambda \end{aligned}$$

۷- دو گرامر معادل با گرامر $\lambda \mid (S) \mid S \rightarrow SS$ بنویسید.

۸- یک گرامر مستقل از متن برای زبان زیر بنویسید.

$$L_1 = \{w \in \{a,b\}^* : n_a(w) \geq n_b(w) \text{ برای هر } V \text{ پیشوند } W\}$$

$$L_2 = \{w \in \{a,b\}^* : n_a(w) = n_b(w), n_a(w) \geq n_b(w) \text{ برای هر } V \text{ پیشوند } W\}$$

۹- آیا زبان زیر مستقل از متن است؟

$$L = \{a^n b^n : n \geq 0, n \text{ مضربی از } 5 \text{ نیست}\}$$

۱۰- گرامری بنویسید که همه رشته‌های دارای حداقل سه a را تولید کند. $\Sigma = \{a,b\}$

۱۱- گرامر مستقل از متنی برای مجموعه تمام عبارات منظم روی الفبای $\{a,b\}$ بنویسید.

۱۲- گرامری برای مجموعه اعداد صحیح در C بنویسید. (بدون محدودیت برای تعداد ارقام)

۱۳- کدام یک از زبانهای زیر، مستقل از متن است؟

$$\text{الف- } L = \{a^n w w^R a^n : w \in (a+b)^*, n \geq 0\}$$

$$\text{ب- } L = \{a^m c b^n : m \neq n\} \cup \{a^m d b^{2m} : m \geq 0\}$$

$$\text{پ- } L = \{a^n b^m c^n : n \geq m \geq 0\} \cup \{a^i b^j c^i : j > i \geq 0\}$$

$$\text{ت- } L = \{a^n b^n c^m : n \geq 0, m \geq 0\} \cap \{a^{2n} b^{2n} c^{2m} : n \geq 0, m \geq 0\}$$

$$\text{ث- } L = \{a^* b^* c^*\} \cap \{w : w \in (a+b+c)^*, n_a(w) = n_b(w) = n_c(w)\}$$

۱۴- گرامر مستقل از متنی برای زبان $L = \{(abc)^n (defg)^n : n \geq 1\}$ بنویسید.

۱۵- گرامر مستقل از متنی برای زبان $L = \{a(bcd)^n bce(fe)^n : n \geq 0\}$ بنویسید.

۱۶- گرامر تولید کننده هر زبان را بنویسید.

$$\text{الف- } (\Sigma = \{a\}) L = \{w : |w| \bmod 3 \neq |w| \bmod 2\}$$

$$\text{ب- } \Sigma = \{a, b\} \quad L = \{w : n_a(w) = 2n_b(w)\}$$

$$\text{پ- } \Sigma = \{a, b, c\} \quad L = \{w : n_a(w) = n_b(w) + 1\}$$

$$\text{ت- } L = \{w \in \{a, b, c\}^* : n_a(w) + n_b(w) < n_c(w)\}$$

$$\text{ث- } L = \{w \in \{a, b, c\}^* : n_a(w) + n_b(w) > n_c(w)\}$$

$$\text{ج- } L = \{a^n b^m c^k : n + 2m = k\}$$

$$\text{چ- } L = \{a^n b^m c^k : n = m \text{ or } m \leq k\}$$

$$\text{ح- } L = \{w_1 c w_2 : w_1, w_2 \in \{a, b\}^+, w_1 \neq w_2^R\}$$

۱۷- با توجه به زبان $L = \{a^m b^m : m \geq 0\}$ ، آیا $L^3 \cap L^4$ مستقل از متن است؟

۱۸- آیا $L_1 \cap L_2 \cap L_3$ ، مستقل از متن است؟

$$L_1 = \{a^p b^q a^p b^s : p, q, s \geq 0\}$$

$$L_2 = \{a^p b^p a^r b^s : p, r, s \geq 0\}$$

$$L_3 = \{a^p b^q a^r b^p : p, q, r \geq 0\}$$

۱۹- آیا زبان L ، مستقل از متن است؟

$$L = \{a^i b^j c^k : i, j, k \geq 0\} \cap \{w \in \{a, b, c\}^* : |w|_a = |w|_b\}$$

۲۰- آیا متمم زبان $L = \{0^n 1^n 0^n : n \in \mathbb{N}\}$ مستقل از متن است؟

۲۱- با توجه به زبان $L = \{(ab)^n (cd)^n : n \geq 0\}$ ، آیا L^* مستقل از متن است؟

۲۲- با فرض اینکه $L \subseteq \{1\}^*$ یک زبان باشد، در مورد نوع زبان L چه می‌توان گفت؟

۲۳- فرض کنید L_1 زبانی منظم و L_2, L_3 زبان‌های مستقل از متن باشند، آیا زبان $(L_2 \cup L_3) - L_1$ مستقل از متن است؟

۲۴- آیا زبان $L = \{a^i b^j c^k : i, j, k \geq 1, k \leq \max(i, j)\}$ مستقل از متن است؟

۲۵- زبان‌های منظم A و B را روی حروف الفبای Σ در نظر بگیرید. اگر تعریف کنیم $L = \bigcup_{n=1}^{\infty} (A^n \cap B^n)$ ،

آیا در رابطه با L چه می‌توان گفت؟

۲۶- اگر L_1 مستقل از متن و L_2 چنین نباشد، آیا $L_1 \cap L_2$ مستقل از متن است؟

۲۷- آیا زبان $L = \{a^n b^m c^m : n \geq 0, m \geq 0\}$ خطی است؟

۲۸- آیا زبان $L = \{a^m b^n c^k : m > 5, n + k = 3t\}$ ، مستقل از متن است؟ $(m, n, k, t \in N)$

۲۹- اگر زبان تولید شده توسط گرامر زیر را L بنامیم، آیا رابطه $L = L^*$ برقرار است؟

$$S \rightarrow Ab$$

$$A \rightarrow aAb \mid \lambda$$

۳۰- اگر زبان تولید شده توسط گرامر زیر را L بنامیم، آیا رابطه $L = L^*$ برقرار است؟

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

۳۱- اگر $L = \{a^n b^m : n \geq 0, n < m\}$ باشد، آنگاه گرامری برای L^2 بنویسید.

۳۲- اگر $L = \{a^n b^m : n \geq 0, n < m\}$ باشد، آنگاه گرامری برای L^* بنویسید.

۳۳- یک گرامر که مجموعه اعداد صحیح در زبان C را تولید می‌کند را بنویسید.

۳۴- گرامر زیر چه زبانی را تولید می‌کند.

$$S \rightarrow bN \mid BN \mid AM$$

$$N \rightarrow aN \mid bN \mid \lambda$$

$$B \rightarrow aBb \mid b$$

$$A \rightarrow aAb \mid a$$

$$M \rightarrow aN \mid \lambda$$

۳۵- کدام یک از زبانهای زیر، مستقل از متن است؟ (Γ و Q دو عدد صحیح و ثابت و مثبت می‌باشند).

$$\text{الف- } L = \{1^{r+mq} 0^{r^q} : m \geq 0\}$$

$$\text{ب- } L = \{1^{r+mq} 0^m : m \geq 0\}$$

۳۶- آیا زبان $L = \{a^n b^m a^p b^q : n + m \leq p + q\}$ مستقل از متن است؟

۳۷- کدام یک از زبان‌های زیر خطی و کدام یک معین می‌باشند؟

$$L_1 = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}$$

$$L_2 = \{a^n b^n : n > 0\} \cup \{a^n b^{2^n} : n > 0\}$$

۳۸- مثالی از ترکیبات زبان C بزنید، که توسط گرامر مستقل از متن، قابل توصیف نباشد.

۳۹- نشان دهید که برای هر گرامر مستقل از متن، یک گرامر معادل وجود دارد که قوانین آن به صورت $A \rightarrow aBC$ یا $A \rightarrow \lambda$ باشند.

$$A, B, C \in V$$

$$a \in \Sigma \cup \{\lambda\}$$

۴۰- گرامر تولید کننده زبان را بنویسید.

$L = \{w \in \{a,b\}^* : n_a(w) = n_b(w) \text{ و } w \text{ شامل زیر رشته } aab \text{ نیست}\}$

پاسخ تمرین فصل ۴

۱- خیر، گرامر G رشته‌هایی را تولید می‌کند که تعداد a و b در آنها برابر است و حتماً باید با a شروع شوند که این موضوع در زبان L_1 قید نشده است، بنابراین L_1 با $L(G)$ برابر نیست.

۲- خیر، چون رشته aa توسط گرامر اول تولید می‌شود، اما توسط گرامر دوم تولید نمی‌شود.

۳- زبان هر گرامر به صورت زیر است:

$$\text{الف- } L = \{x^n y^n z^m : n \geq 0, m \geq 0\}$$

قانون $A \rightarrow xAy$ رشته‌هایی به فرم $x^n y^n$ و قانون $B \rightarrow zBz$ رشته‌هایی به فرم z^m تولید می‌کنند. در نهایت به علت وجود قانون $AB \rightarrow S$ ، رشته‌هایی به فرم $x^n y^n z^m$ تولید می‌شود.

$$\text{ب- } L = \{x^n y^k z^m : k = n + m, n \geq 0, m \geq 0\}$$

قانون $A \rightarrow xAy$ رشته‌هایی به فرم $x^n y^n$ و قانون $B \rightarrow yBz$ رشته‌هایی به فرم $y^m z^m$ تولید می‌کنند. در نهایت به علت وجود قانون $AB \rightarrow S$ ، رشته‌هایی به فرم $x^n y^n y^m z^m$ تولید می‌شود.

$$\text{پ- } L = \{x^n y^m z^t : t = n + m, n \geq 0, m \geq 0\}$$

$$\text{ت- } L = \{x^n y^n z^k : n \geq 1, k \geq 1\}$$

۴- زبان هر گرامر برابر است با:

$$\text{الف- } L = \{a^n b^{n+1} : n \geq 1\} \cup \{a^{n+1} b^n : n \geq 1\}$$

$$\text{ب- } L = \{x^m y^n z^n : n \geq 0, m \geq 0\}$$

قانون $A \rightarrow xA$ رشته‌هایی به فرم x^m و قانون $B \rightarrow yBz$ رشته‌هایی به فرم $y^n z^n$ تولید می‌کنند. در نهایت به علت وجود قانون $AB \rightarrow S$ ، رشته‌هایی به فرم $x^m y^n z^n$ تولید می‌شود.

$$\text{پ- } L = \{a^n b^m : n \neq m\} \text{ (یا } a^+ a^n b^n + a^n b^n b^+)$$

ت- $L = \{a^2 a^{2k} b^k b^2 : k \geq 0\} \cup \{b^t a^{t+1} : t \geq 1\}$ رشته‌هایی که به کمک قوانین $S \rightarrow aAb, A \rightarrow aaAb | ab$ تولید می‌شود به فرم $a^2 a^{2k} b^k b^2$ می‌باشد ($k \geq 0$) و رشته‌هایی که به کمک قوانین $S \rightarrow bBa, B \rightarrow bBa | a$ تولید می‌شود به فرم $b^t a^{t+1}$ می‌باشد. ($t \geq 1$).

۵- زبان هر گرامر برابر است با:

$$L = \{a(bcc)^n baa(daa)^n : n \geq 0\} \text{ - الف}$$

$$L = \{ab(cda)^n cdb(bab)^n : n \geq 0\} \text{ - ب}$$

$$0^n AA1^n + 1^n AA0^n = 0^n (0+1)^* 1^n + 1^n (0+1)^* 0^n = (0+1)^* \text{ - پ}$$

ت- $\{a^n b^n : n \geq 0\} - \{a, b\}^*$ یعنی تمامی رشته‌ها در $\{a, b\}^*$ به غیر از $\{a^n b^n : n \geq 0\}$.

۶- قانون $\lambda \mid A \rightarrow 0A1^n$ زبان $0^n 1^n$ ، قانون $\lambda \mid B \rightarrow 1B$ زبان 1^+ و قانون $\lambda \mid C \rightarrow 1C0$ ، زبان $1^m 0^m$ را تولید می‌کنند. در نهایت به علت وجود $S \rightarrow ABC$ ، این گرامر زبان زیر را تولید می‌کند:

$$\{0^n 1^n 1^+ 1^m 0^m\} = \{0^n 1^{n+m+t} 0^m : t > 0, n \geq 0, m \geq 0\}$$

که می‌توان آن را به صورت $L = \{0^i 1^j 0^k : j > i + k\}$ نشان داد.

۷- دو گرامر زیر با گرامر $\lambda \mid S \rightarrow SS \mid (S)$ معادل هستند:

$$G1 : S \rightarrow S(S) \mid \lambda$$

$$G2 : S \rightarrow S(SS) \mid \lambda$$

۸- گرامر برای L_1 :

$$G1 : S \rightarrow aS \mid aSbS \mid \lambda$$

گرامر برای L_2 :

$$G2 : S \rightarrow aSb \mid SS \mid \lambda$$

۹- بله- چون می‌توان برای آن یک گرامر مستقل از متن به صورت زیر نوشت:

$$S \rightarrow a^5 Sb^5 \mid a^4 b^4 \mid a^3 b^3 \mid a^2 b^2 \mid ab$$

۱۰- ابتدا سه a تولید کرده و سپس تعداد دلخواهی a و b را در هر جای دلخواه از رشته اضافه می‌کنیم:

$$S \rightarrow XaXaXaX$$

$$X \rightarrow aX \mid bX \mid \lambda$$

۱۱- گرامر مورد نظر عبارت است از:

$$E \rightarrow E + E \mid E.E \mid E^* \mid (E) \mid \lambda \mid \phi \mid a \mid b$$

۱۲- گرامر مورد نظر عبارت است از:

$$S \rightarrow AB$$

$$A \rightarrow + | - | \lambda$$

$$B \rightarrow D | DB$$

$$D \rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$

۱۳- بررسی گزاره‌ها:

الف- بله، چون می‌توان یک گرامر مستقل از متن برای آن ارائه داد:

$$S \rightarrow aSa | bMb$$

$$M \rightarrow aMa | bMb | \lambda$$

ب- بله، چون می‌توان یک گرامر مستقل از متن برای آن ارائه داد:

$$S \rightarrow A | B$$

$$A \rightarrow aAb | aA | Ab | ac | cb$$

$$B \rightarrow aBbb | d$$

پ- زبان L به صورت $\{a^n b^k c^n : n, k \geq 0\}$ است که مستقل از متن می‌باشد.

ت- این زبان به صورت $L = \{a^{2n} b^{2n} c^{2m} : n \geq 0, m \geq 0\}$ است، که مستقل از متن می‌باشد.

ث- زبان L برابر $\{a^n b^n c^n : n \geq 0\}$ است، که مستقل از متن نمی‌باشد.

۱۴- این گرامر به صورت زیر است:

$$S \rightarrow abXefg$$

$$X \rightarrow cSd | cd$$

۱۵- این گرامر به صورت زیر است:

$$S \rightarrow aX$$

$$X \rightarrow bcYe$$

$$Y \rightarrow dXf | \lambda$$

۱۶- گرامر تولید کننده هر زبان را بنویسید.

الف- طول رشته نباید $6k+1$ ، $6k$ باشد.

$$S \rightarrow aaaaaaS | aa | aaa | aaaa | aaaaa$$

ب- گرامر به صورت زیر است:

$$\begin{aligned}
S &\rightarrow SS \mid \lambda \\
S &\rightarrow aabS \mid aaSb \mid aSab \mid Saab \\
S &\rightarrow abaS \mid abSa \mid aSba \mid Saba \\
S &\rightarrow baaS \mid baSa \mid bSaa \mid Sbaa
\end{aligned}$$

پ-

$$\begin{aligned}
S &\rightarrow NXNaNXN \\
X &\rightarrow aXb \mid bXa \mid XN \mid NX \mid \lambda \\
N &\rightarrow cN \mid \lambda
\end{aligned}$$

ت-

$$\begin{aligned}
S &\rightarrow aSc \mid bSc \mid cSa \mid cSb \mid SS \mid XS \mid c \\
X &\rightarrow cX \mid \lambda
\end{aligned}$$

ث-

$$\begin{aligned}
S &\rightarrow aSc \mid bSc \mid cSa \mid cSb \mid SS \mid AS \mid BS \mid a \mid b \\
A &\rightarrow aA \mid \lambda \\
B &\rightarrow bB \mid \lambda
\end{aligned}$$

ج-

$$\begin{aligned}
S &\rightarrow aSc \mid A \\
A &\rightarrow bAcc \mid \lambda
\end{aligned}$$

چ-

$$\begin{aligned}
S &\rightarrow ND \mid AMD \\
N &\rightarrow aNb \mid \lambda \\
M &\rightarrow bMc \mid \lambda \\
D &\rightarrow cD \mid \lambda \\
A &\rightarrow aA \mid \lambda
\end{aligned}$$

ح-

$$\begin{aligned}
S &\rightarrow A \mid c \\
A &\rightarrow aAa \mid bAb \mid aMb \mid bMa \mid aNc \mid bNc \mid cNa \mid cNb \\
M &\rightarrow aM \mid bM \mid cN \\
N &\rightarrow aN \mid bN \mid \lambda
\end{aligned}$$

۱۷- بله- زبان L^3 و L^4 به صورت زیر می باشند:

$$L^3 = \{a^m b^m a^n b^n a^p b^p : m, n, p \geq 0\}$$

$$L^4 = \{a^m b^m a^n b^n a^p b^p a^k b^k : m, n, p, k \geq 0\}$$

اشتراک آنها برابر L^3 است که مستقل از متن است.

۱۸- هر ۳ زبان مستقل از متن است، اما اشتراک آنها $\{a^n b^n a^n b^n : n \geq 0\}$ مستقل از متن نمی باشد.

۱۹- بله- زبان L حاصل اشتراک یک زبان مستقل از متن و یک زبان منظم است، بنابراین مستقل از متن است.

۲۰- متمم زبان داده شده یعنی $\{0^n 1^m 0^k : n \neq m \neq k\}$ مستقل از متن است. (خود زبان L مستقل از متن نیست.)

۲۱- بله- زبان L مستقل از متن می باشد و چون زبان مستقل از متن تحت بستار بسته است، بنابراین L^* نیز مستقل از متن است.

۲۲- این زبان حتما مستقل از متن نیست. مانند $L = \{1^{n^2} : n > 0\}$. البته اگر بدانیم L مستقل از متن است، آنگاه L حتما منظم خواهد بود.

۲۳- بله- زبان $L_1 - (L_2 \cup L_3)$ را به صورت $(L_2 \cup L_3) \cap \bar{L}_1$ می نویسیم. می دانیم که:

الف- زبانهای مستقل از متن نسبت به اجتماع بسته هستند، پس $L_2 \cup L_3$ مستقل از متن است.

ب- زبان منظم نسبت به عمل متمم بسته است، پس \bar{L}_1 منظم است.

ج- اشتراک یک زبان مستقل از متن با یک زبان منظم، زبانی است مستقل از متن.

در نتیجه زبان $(L_2 \cup L_3) \cap \bar{L}_1$ مستقل از متن است.

۲۴- بله- زبان L با شرط $k \leq \max(i, j)$ ، از اجتماع دو زبان با شرطهای $k \leq j$ و $k \leq i$ تشکیل شده:

$$L = \{a^i b^j c^k : i, j, k \geq 1, k \leq i\} \cup \{a^i b^j c^k : i, j, k \geq 1, k \leq j\}$$

بنابراین چون این دو زبان مستقل از متن بوده و زبان های مستقل از متن تحت اجتماع بسته هستند، زبان L نیز مستقل از متن می باشد.

۲۵- بله- چون A و B منظم هستند، آنگاه برای هر n عبارت $(A^n \cap B^n)$ منظم بوده ولی چون زبان‌های منظم تحت اجتماع نامتناهی یعنی $L = \bigcup_{n=1}^{\infty} (A^n \cap B^n)$ بسته نمی باشند، زبان L لزوماً منظم نیست ولی حتماً مستقل از متن می باشد.

۲۶- اگر L_1 مستقل از متن و L_2 منظم باشد، آنگاه $L_1 \cap L_2$ مستقل از متن است. اما در این گزینه در مورد زبان L_2 اطلاعی نداریم و در نتیجه $L_1 \cap L_2$ ممکن است مستقل از متن باشد و ممکن است نباشد.

۲۷- بله- چون می توان یک گرامر خطی برای آن نوشت:

$$S \rightarrow aS \mid bAc \mid \lambda$$

$$A \rightarrow bAc \mid \lambda$$

۲۸- با توجه به محدودیت های داده شده (تعداد a بیشتر از ۵ و مجموع تعداد b ها و c ها مضرب ۳ باشد)، زبان داده شده منظم است، و در نتیجه مستقل از متن نیز هست.

۲۹- خیر- زبان گرامر برابر $L = \{a^n b^{n+1} : n \geq 0\}$ است که شامل λ نیست. پس $L \neq L^*$.

۳۰- بله- زبان گرامر برابر $L = \{w : n_a(w) = n_b(w)\}$ است. اگر L^* را به واحدهای شامل یک L تقسیم کنیم، در هر کدام از آنها تعداد a با b برابر است. پس $L = L^*$.

۳۱- زبان L^2 برابر $L^2 = \{a^n b^m a^p b^q : n \geq 0, n < m, p \geq 0, p < q\}$ است. گرامر آن برابر است با:

$$S \rightarrow XX$$

$$X \rightarrow aXb \mid bY$$

$$Y \rightarrow bY \mid \lambda$$

۳۲- گرامر برابر است با:

$$S \rightarrow SS \mid X \mid \lambda$$

$$X \rightarrow aXb \mid bY$$

$$Y \rightarrow bY \mid \lambda$$

۳۳- گرامر برابر است با:

$$S \rightarrow IDA$$

$$I \rightarrow + \mid - \mid \lambda$$

$$D \rightarrow 0 \mid 1 \mid 2 \mid \dots \mid 9$$

$$A \rightarrow DA \mid \lambda$$

۳۴- متمم زبان $L = \{a^n b^m : n \geq 0\}$

۳۵- بررسی زبان‌ها:

الف-خیر، با فرض اینکه $r=2, q=3$ داریم: $L = \{1^{2+3m}0^6 : m \geq 0\}$. در این صورت، تعداد صفرها ثابت شده و وابستگی به تعداد 1 ها ندارد. پس زبان منظم می باشد.

ب- بله، با فرض اینکه $r=2, q=3$ داریم: $L = \{1^{2+3m}0^{2m} : m \geq 0\}$. در این صورت، 0 و 1 به یکدیگر وابسته هستند و در نتیجه زبان مستقل از متن است.

زبان را می توان به صورت $(0^r)^m 1^r (1^q)^m$ نشان داد. به علت وابستگی توانی و محدود نبودن m ، نمی توان یک گرامر منظم طراحی کرد. اما می توان یک گرامر مستقل از متن برای آن طراحی کرد.

۳۶- بله- چون می توان برای آن یک گرامر مستقل از متن نوشت:

$$S \rightarrow aSb \mid Sb \mid bAa \mid aBa \mid \lambda$$

$$A \rightarrow bAa \mid Aa \mid \lambda$$

$$B \rightarrow aBa \mid bAa \mid Ba \mid \lambda$$

۳۷- زبان مستقل از متن L_1 معین است ولی خطی نیست. زبان L_2 خطی است ولی معین نیست.

۳۸- تعریف توابع در C ، توسط گرامر مستقل از متن، قابل توصیف نیست، چون ترتیب توابع در این زبان مهم نیست.

۳۹- هر گرامر مستقل از متن را می توان به فرم نرمال چامسکی در آورد. پس ابتدا گرامر را به فرم نرمال چامسکی تبدیل کرده و سپس قواعد به فرم $A \rightarrow a$ را حذف کرده و به جای آن قواعد زیر را قرار می دهیم:

$$A \rightarrow aMN$$

$$M \rightarrow \lambda$$

$$N \rightarrow \lambda$$

قواعد دیگر نیازی به تغییر ندارند، چون قواعد به فرم $A \rightarrow BC$ در فرم مطلوب هستند. a در قواعد $A \rightarrow aBC$ می تواند λ باشد.

منتخبی از عناوین آموزشی منتشر شده بر روی فرادرس

برنامه نویسی	
مدت زمان تقریبی	عنوان آموزش
۲ ساعت	آموزش اصول و مبانی برنامه نویسی - کلیک کنید (+)
۲۳ ساعت	آموزش برنامه نویسی جاوا - کلیک کنید (+)
۲۹ ساعت	آموزش برنامه نویسی PHP - کلیک کنید (+)
۲۰ ساعت	آموزش برنامه نویسی C++ - کلیک کنید (+)
۸ ساعت	آموزش پیشرفته C++ (شی گرایبی در سی پلاس پلاس) - کلیک کنید (+)
۸ ساعت	آموزش مقدماتی برنامه نویسی سی شارپ (C#) - کلیک کنید (+)
۱۴ ساعت	مجموعه آموزش های کاربردی برنامه نویسی C# (سی شارپ) - کلیک کنید (+)
۱۴ ساعت	آموزش شی گرایبی در سی شارپ (C#) - کلیک کنید (+)
۱۸ ساعت	آموزش برنامه نویسی پایتون - مقدماتی - کلیک کنید (+)
۵ ساعت	آموزش تکمیلی برنامه نویسی پایتون - کلیک کنید (+)
۱۳ ساعت	آموزش برنامه نویسی C - کلیک کنید (+)
۱۱ ساعت	آموزش SQL Server - مقدماتی - کلیک کنید (+)
۶ ساعت	آموزش SQL Server - تکمیلی - کلیک کنید (+)
۳ ساعت	آموزش کار با دستورات پایگاه داده در SQL Server - کلیک کنید (+)
۹ ساعت	مجموعه آموزش های برنامه نویسی متلب (MATLAB) - کلیک کنید (+)
۷ ساعت	مجموعه آموزش های برنامه نویسی متلب پیشرفته - کلیک کنید (+)
۱۶ ساعت	آموزش برنامه نویسی اندروید (Android) - مقدماتی - کلیک کنید (+)
۲۰ ساعت	آموزش برنامه نویسی اندروید (Android) - تکمیلی - کلیک کنید (+)
۲۶ ساعت	آموزش برنامه نویسی اندروید (Android) - پیشرفته - کلیک کنید (+)
۳ ساعت	آموزش پروژه محور برنامه نویسی دلفی (Delphi) - کلیک کنید (+)
۴ ساعت	آموزش زبان برنامه نویسی فرترن (Fortran) - مقدماتی - کلیک کنید (+)
۲ ساعت	آموزش مروری زبان برنامه نویسی FORTRAN با مثال های کاربردی - کلیک کنید (+)
۱۰ ساعت	آموزش پروژه محور اینترنت اشیا (IoT) - کنترل لوازم منزل با شبکه و پیامک و ماژول های SIM و ESP8266 - کلیک کنید (+)
۵ ساعت	آموزش برنامه نویسی Swift (سوئیفت) برای برنامه نویسی iOS - کلیک کنید (+)
۶ ساعت	آموزش برنامه نویسی پاسکال (Pascal) - کلیک کنید (+)
۱۶ ساعت	آموزش برنامه نویسی برای دانش آموزان با اسمال بیسیک (Small Basic) - کلیک کنید (+)
۶ ساعت	آموزش برنامه نویسی تصویری به کودکان با زبان اسکرچ (Scratch) - کلیک کنید (+)
۱ ساعت	آموزش محافظت از کدهای نرم افزاری با SmartAssembly (اسمارت اسمبلی) - کلیک کنید (+)

برنامه نویسی (ادامه از صفحه قبل)	
مدت زمان تقریبی	عنوان آموزش
۱۰ ساعت	آموزش مبانی برنامه نویسی (الگوریتم و فلوچارت) با رویکرد حل مسأله - کلیک کنید (+)
۱۰ ساعت	آموزش پروژه محور Python (پایتون) - ساخت نرم افزار برای Windows و Linux - کلیک کنید (+)
۵ ساعت	آموزش برنامه نویسی آردوینو (Arduino) با پایتون (Python) - کلیک کنید (+)
۲ ساعت	آموزش ساخت اپلیکیشن اندروید (Android) با PHP و MySQL - کلیک کنید (+)
۳ ساعت	آموزش مقدماتی زبان برنامه نویسی کاتلین (Kotlin) برای توسعه اندروید (Android) - کلیک کنید (+)
۱ ساعت	آموزش داکر (Docker) برای توسعه دهندگان - کلیک کنید (+)
۱ ساعت	آموزش توسعه وب با زبان برنامه نویسی Go - کلیک کنید (+)
۱۳ ساعت	آموزش ویژوال بیسیک دات نت (Visual Basic.NET) - مقدماتی - کلیک کنید (+)
۱۶ ساعت	آموزش ویژوال بیسیک دات نت (Visual Basic.NET) - تکمیلی - کلیک کنید (+)
۱ ساعت	آموزش تاریخچه گیت (Git) و مقدمات آن - کلیک کنید (+)
۱ ساعت	آموزش گیت (Git) برای مدیریت نسخه توزیع شده - کلیک کنید (+)
۵ ساعت	آموزش زبان برنامه نویسی AWK در لینوکس - پردازش و آنالیز فایل های متنی - کلیک کنید (+)
۷ ساعت	آموزش برنامه نویسی R و نرم افزار RStudio - کلیک کنید (+)
۵ ساعت	آموزش تکمیلی برنامه نویسی R و نرم افزار RStudio - کلیک کنید (+)
۱ ساعت	آموزش مقدماتی ساخت ربات تلگرام با PHP - کلیک کنید (+)
۹ ساعت	آموزش کتابخانه قالب استاندارد (STL) در C++ برای ساده نویسی برنامه ها - کلیک کنید (+)
۴ ساعت	آموزش برنامه نویسی با روش سه لایه به زبان VB.Net - کلیک کنید (+)
۴ ساعت	آموزش بازی سازی در پایتون با کتابخانه Pygame - کلیک کنید (+)
۵ ساعت	آموزش پایتون گرافیکی (رابط های گرافیکی پایتون) - کلیک کنید (+)
۱ ساعت	آموزش ساخت ربات تلگرام با پایتون (Python) - کلیک کنید (+)
۷ ساعت	مجموعه آموزش های طراحی رابط های گرافیکی (GUI) در متلب
۱۴ ساعت	آموزش تکمیلی طراحی رابط گرافیکی کاربر (GUI) با متلب (MATLAB) - کلیک کنید (+)
۱ ساعت	آموزش ساخت ربات تلگرام با متلب (MATLAB) - کلیک کنید (+)
۱ ساعت	آموزش ساخت ربات تلگرام با جاوا (Java) - کلیک کنید (+)
۲ ساعت	آموزش پایگاه داده ها در جاوا - کلیک کنید (+)
۳ ساعت	آموزش کار با فایل ها در جاوا با پکیج Java/IO (به همراه پروژه دفترچه یادداشت) - کلیک کنید (+)
۱ ساعت	آموزش پروژه محور جاوا - تولید QR code و ایجاد و اسکن بارکد (Barcode) - کلیک کنید (+)

برنامه نویسی (ادامه از صفحه قبل)	
مدت زمان تقریبی	عنوان آموزش
۶ ساعت	آموزش پروژه محور جاوا (Java) - طراحی و ساخت شبکه اجتماعی - کلیک کنید (+)
۴ ساعت	آموزش طراحی رابط کاربری مدرن در .NET. با کامپوننت های جانوس (Janus) - کلیک کنید (+)
۲ ساعت	آموزش ساخت بازی ساده در ویژوال بیسیک - کلیک کنید (+)
۱۰ ساعت	آموزش پیاده سازی سیستم انبارداری و فروش با ویژوال بیسیک دات نت (VB.NET) - مقدماتی - کلیک کنید (+)
۱۲ ساعت	آموزش پیاده سازی سیستم انبارداری و فروش با ویژوال بیسیک دات نت (VB.NET) - تکمیلی - کلیک کنید (+)
۴ ساعت	آموزش مقدماتی پیاده سازی مسائل بهینه سازی در پایتون (Python) - کلیک کنید (+)
۱۲ ساعت	آموزش الگوهای طراحی (Design Patterns) در پایتون (Python) - کلیک کنید (+)
۲ ساعت	آموزش رایگان نکات و ترفندهای متلب - کلیک کنید (+)
۱۴ ساعت	مجموعه آموزش های برنامه نویسی متلب برای علوم و مهندسی - کلیک کنید (+)
۶ ساعت	آموزش فریم ورک Spring در جاوا - کلیک کنید (+)
۱۸ ساعت	آموزش ORM هایبرنیت (Hibernate) جاوا (Java) - کلیک کنید (+)
۲ ساعت	آموزش آشنایی با LINQ to SQL در C# - کلیک کنید (+)
۱ ساعت	آموزش برنامه نویسی تحت شبکه با سی شارپ در قالب پروژه - کلیک کنید (+)
۴ ساعت	آموزش برنامه نویسی با روش سه لایه به زبان سی شارپ - کلیک کنید (+)
۳ ساعت	آموزش Cryptography در دات نت - کلیک کنید (+)
۴ ساعت	آموزش قفل نرم افزاری در C# از طریق رجیستری - کلیک کنید (+)
۱۹ ساعت	آموزش پروژه محور سی شارپ (C#) - سیستم مدیریت آموزشگاه - کلیک کنید (+)
۱۴ ساعت	آموزش پروژه محور سی شارپ (C#) - سیستم مدیریت حسابداری و انبارداری - کلیک کنید (+)
۷ ساعت	آموزش برنامه نویسی سورس کدهای الگوریتمی در سی شارپ - کلیک کنید (+)
۱۰ ساعت	آموزش پروژه محور سی شارپ (C#) - پیاده سازی سیستم هتلداری - کلیک کنید (+)
۴ ساعت	آموزش ساخت سیستم آزمون گیر در سی شارپ - کلیک کنید (+)
۳ ساعت	آموزش ارسال و دریافت پیامک با مودم در سی شارپ (C#) - کلیک کنید (+)
۱۱ ساعت	آموزش پروژه محور سی شارپ (C#) - پیاده سازی سیستم مدیریت آتلیه - کلیک کنید (+)
۷ ساعت	آموزش پروژه محور سی شارپ (C#) - پیاده سازی سیستم نوبت دهی - کلیک کنید (+)
۹ ساعت	آموزش پروژه محور سی شارپ (C#) - پیاده سازی سیستم صدور بلیط - کلیک کنید (+)
۳ ساعت	آموزش رسم نمودار در سی شارپ (C#) - کلیک کنید (+)
۷ ساعت	آموزش پروژه محور سی شارپ (C#) - سیستم مدیریت بانک - کلیک کنید (+)
۵ ساعت	آموزش ساخت بازی با سی شارپ (C#) - کلیک کنید (+)
۲ ساعت	آموزش Metaprogramming (برنامه نویسی متا) در پایتون - کلیک کنید (+)

برنامه نویسی (ادامه از صفحه قبل)	
مدت زمان تقریبی	عنوان آموزش
۱۲ ساعت	آموزش کاربردی Entity Framework در سی شارپ (C#) - کلیک کنید (+)
۶ ساعت	آموزش پروژه محور سی شارپ (C#) - پیاده سازی سیستم مدیریت کارمندان - کلیک کنید (+)
۶ ساعت	آموزش پروژه محور سی شارپ (C#) - پیاده سازی سیستم مدیریت چک های بانکی - کلیک کنید (+)
۴ ساعت	آموزش پروژه محور سی شارپ (C#) - شبیه سازی دستگاه عابر بانک - کلیک کنید (+)
۲ ساعت	آموزش گزارش گیری با کریستال ریپورت و استیمول سافت (Stimulsoft) در سی شارپ (C#) - کلیک کنید (+)
۲ ساعت	آموزش پایگاه داده اس کیو لایت (SQLite) در سی شارپ (C#) - کلیک کنید (+)
۶ ساعت	آموزش پروژه محور سی شارپ (C#) - پیاده سازی نرم افزار حسابداری شخصی - کلیک کنید (+)
۱۵ ساعت	آموزش پروژه محور سی شارپ (C#) - پیاده سازی سیستم رستوران آنلاین - کلیک کنید (+)
۳ ساعت	آموزش کار با بانک اطلاعاتی اکسس (Access) در سی شارپ (C#) - کلیک کنید (+)
۲ ساعت	آموزش کار با فایل ها و دایرکتوری در C# (سی شارپ) - کلیک کنید (+)
۳ ساعت	آموزش پروژه محور سی شارپ C# - پیاده سازی نرم افزار FeedReader (فیدریدر) - کلیک کنید (+)
۴ ساعت	آموزش کار با دستورات گرافیکی در سی شارپ (C#) - کلیک کنید (+)
۳ ساعت	آموزش مقدماتی WPF در سی شارپ (C#) برای ساخت و اجرای اینترفیس - کلیک کنید (+)
۴ ساعت	آموزش WPF در سی شارپ برای ساخت و اجرای اینترفیس - پیشرفته - کلیک کنید (+)
۲ ساعت	آموزش WPF در سی شارپ (C#) برای ساخت و اجرای اینترفیس - تکمیلی - کلیک کنید (+)
۱ ساعت	آموزش ساخت ربات تلگرام با سی شارپ (C#) - کلیک کنید (+)
۱۲ ساعت	آموزش پروژه محور پایگاه داده LocalDB در سی شارپ (C#) - سیستم مدیریت کارمندان - کلیک کنید (+)
۵ ساعت	آموزش کتابخانه Spire.Doc در سی شارپ (C#) برای ایجاد و ویرایش فایل Word - کلیک کنید (+)
۱ ساعت	آموزش پروژه محور سی شارپ (C#) - ساخت برنامه دانلودگر - کلیک کنید (+)

علوم کامپیوتر	
مدت زمان تقریبی	عنوان آموزش
۱۰ ساعت	آموزش ساختمان داده ها - کلیک کنید (+)
۸ ساعت	آموزش پیشرفته ساختمان داده (همراه با حل نمونه سوالات کنکور ارشد و دکتری) - کلیک کنید (+)
۲۰ ساعت	آموزش ساختمان داده ها (مرور - تست کنکور ارشد) - کلیک کنید (+)
۱۱ ساعت	آموزش سیستم های عامل - کلیک کنید (+)
۱۲ ساعت	آموزش سیستم عامل (مرور - تست کنکور ارشد) - کلیک کنید (+)
۸ ساعت	آموزش پایگاه داده ها - کلیک کنید (+)
۸ ساعت	آموزش پایگاه داده ها (مرور - تست کنکور ارشد) - کلیک کنید (+)
۱۰ ساعت	آموزش طراحی و پیاده سازی زبان های برنامه سازی - کلیک کنید (+)
۹ ساعت	آموزش طراحی و پیاده سازی زبان های برنامه سازی (مرور - تست کنکور ارشد) - کلیک کنید (+)
۱۵ ساعت	آموزش طراحی الگوریتم - کلیک کنید (+)
۸ ساعت	آموزش طراحی الگوریتم به همراه حل مثال های عملی - کلیک کنید (+)
۴ ساعت	آموزش روش های حل روابط بازگشتی - کلیک کنید (+)
۲ ساعت	آموزش رابطه های بازگشتی در طراحی الگوریتم و ساختمان گسسته (مرور - تست کنکور ارشد) - کلیک کنید (+)
۱۵ ساعت	آموزش طراحی الگوریتم (مرور - تست کنکور ارشد) - کلیک کنید (+)
۲ ساعت	آموزش روش تقسیم و حل در طراحی الگوریتم (مرور - تست کنکور کارشناسی ارشد) - کلیک کنید (+)
۱۸ ساعت	آموزش مبانی منطق و نظریه مجموعه ها - کلیک کنید (+)
۱۵ ساعت	آموزش مدار منطقی (مرور - تست کنکور ارشد) - کلیک کنید (+)
۱۲ ساعت	آموزش مبانی الکترونیک دیجیتال - بخش اول - کلیک کنید (+)
۱۱ ساعت	آموزش مبانی الکترونیک دیجیتال - بخش دوم - کلیک کنید (+)
۲۰ ساعت	آموزش معماری کامپیوتر با رویکرد حل مساله - مقدماتی - کلیک کنید (+)
۱۴ ساعت	آموزش معماری کامپیوتر با رویکرد حل مساله - پیشرفته - کلیک کنید (+)
۹ ساعت	آموزش نظریه زبان ها و ماشین ها - کلیک کنید (+)
۸ ساعت	آموزش نظریه زبان ها و ماشین (مرور - تست کنکور ارشد) - کلیک کنید (+)
۳ ساعت	آموزش شبیه سازی و تست نظریه زبان ها و ماشین در JFLAP - کلیک کنید (+)
۴ ساعت	آموزش یادگیری عمیق در پزشکی - کلیک کنید (+)
۱۶ ساعت	آموزش ساختمان گسسته با رویکرد حل مساله - کلیک کنید (+)
۱۲ ساعت	آموزش ساختمان گسسته (مرور و حل تست های کنکور کارشناسی ارشد) - کلیک کنید (+)
۱۳ ساعت	آموزش آمار و احتمال مهندسی - کلیک کنید (+)
۱۴ ساعت	آموزش آمار و احتمال مهندسی (حل تمرین و تست کنکور ارشد) - کلیک کنید (+)

علوم کامپیوتر (ادامه از صفحه قبل)	
مدت زمان تقریبی	عنوان آموزش
۱۹ ساعت	آموزش شبکه‌های کامپیوتری ۱ - کلیک کنید (+)
۲۳ ساعت	آموزش شبکه‌های کامپیوتری ۲ - همراه با حل مسأله - کلیک کنید (+)
۱ ساعت	آموزش شبیه‌سازی شبکه‌های مبتنی بر سیسکو (Cisco) در نرم افزار Boson NetSim - کلیک کنید (+)
۱۵ ساعت	آموزش طراحی کامپایلر - کلیک کنید (+)
۱۵ ساعت	آموزش طراحی کامپایلر (مرور و حل تست‌های کنکور کارشناسی ارشد) - کلیک کنید (+)
۸ ساعت	آموزش ذخیره و بازیابی اطلاعات - کلیک کنید (+)
۲۳ ساعت	آموزش ساختمان داده‌ها همراه با پیاده‌سازی در C++ - کلیک کنید (+)
۹ ساعت	آموزش انتقال داده‌ها در شبکه‌های کامپیوتری و اینترنت به همراه حل مسائل - کلیک کنید (+)
۵ ساعت	آموزش مقدماتی Hadoop (هدوپ) برای تجزیه و تحلیل کلان داده - کلیک کنید (+)
۲۲ ساعت	آموزش نظریه صف (Queueing theory) - کلیک کنید (+)
۴ ساعت	آموزش گرافیک کامپیوتری - کلیک کنید (+)
۱۶ ساعت	آموزش گرافیک کامپیوتری با OpenGL - کلیک کنید (+)
۱۲ ساعت	آموزش مهندسی نرم افزار ۱ - کلیک کنید (+)
۱۰ ساعت	آموزش زبان تخصصی مهندسی کامپیوتر - کلیک کنید (+)
۱ ساعت	آموزش مروری بر پیچیدگی محاسبات (Computational Complexity) - کلیک کنید (+)
۱ ساعت	آموزش تبدیل فلوچارت به کد با Flowgorithm - کلیک کنید (+)
۱ ساعت	آموزش نرم افزار RAPTOR برای ترسیم فلوچارت - کلیک کنید (+)
۲ ساعت	آموزش حل سوالات کنکور کارشناسی ارشد کامپیوتر - ۹۴ - کلیک کنید (+)
۱۷ ساعت	آموزش مرور و حل تست‌های کنکور کارشناسی ارشد معماری کامپیوتر - کلیک کنید (+)
۱۰ ساعت	آموزش برنامه نویسی اسمبلی - کلیک کنید (+)
۱۵ ساعت	آموزش نظریه گراف و کاربردها - کلیک کنید (+)
۱۵ ساعت	آموزش ترکیبیات و کاربردها - کلیک کنید (+)
۸ ساعت	آموزش نظریه بازی‌ها و کاربرد آن در اقتصاد - کلیک کنید (+)
۴ ساعت	آموزش مدل سازی UML با نرم افزار Rational Rose - کلیک کنید (+)
۱۳ ساعت	آموزش الگوریتم موازی و پردازش موازی - کلیک کنید (+)
۲۰ ساعت	آموزش ترکیب اطلاعات (Information Fusion) - کلیک کنید (+)
۳ ساعت	آموزش نهان نگاری دیجیتال (Digital watermarking) - کلیک کنید (+)
۳ ساعت	آموزش نهان نگاری و پیاده‌سازی آن در سی شارپ (C#) - کلیک کنید (+)
۱ ساعت	آموزش واترمارکینگ در نهان نگاری با سی شارپ (C#) - کلیک کنید (+)
۱ ساعت	آموزش شبکه‌های پیچیده پویا (Complex Dynamical Networks) - کلیک کنید (+)

علوم کامپیوتر (ادامه از صفحه قبل)	
مدت زمان تقریبی	عنوان آموزش
۳ ساعت	آموزش تخمین تلاش لازم برای توسعه نرم افزارها با متلب - کلیک کنید (+)
۴ ساعت	آموزش توسعه نرم افزار با متد ICONIX و زبان مدل سازی UML - کلیک کنید (+)
۱ ساعت	آموزش نگارش پروپوزال، پایان نامه و ارائه - ویژه پردازش تصویر و بینایی کامپیوتر - کلیک کنید (+)

هوش مصنوعی	
مدت زمان تقریبی	عنوان آموزش
۴ ساعت	آموزش هوش مصنوعی - کلیک کنید (+)
۱۹ ساعت	آموزش هوش مصنوعی - تکمیلی - کلیک کنید (+)
۲۴ ساعت	آموزش هوش مصنوعی توزیع شده - کلیک کنید (+)
۱۵ ساعت	آموزش هوش مصنوعی (مرور و حل تست کنکور ارشد) - کلیک کنید (+)
۲۸ ساعت	مجموعه آموزش های شبکه های عصبی مصنوعی در متلب - کلیک کنید (+)
۱۰ ساعت	مجموعه آموزش های کاربردی شبکه های عصبی مصنوعی - کلیک کنید (+)
۲ ساعت	آموزش طراحی سیستم های فازی عصبی یا ANFIS با استفاده از الگوریتم های فراابتکاری و تکاملی - کلیک کنید (+)
۵ ساعت	آموزش کاربردی مکترونیک - محاسبات پایه طراحی درب های خودکار - کلیک کنید (+)
۴ ساعت	آموزش شبکه عصبی GMDH به همراه پیاده سازی عملی در متلب - کلیک کنید (+)
۳ ساعت	آموزش شبکه های عصبی گازی به همراه پیاده سازی عملی در متلب - کلیک کنید (+)
۳ ساعت	آموزش طبقه بندی و بازشناسی الگو با شبکه های عصبی LVQ در متلب - کلیک کنید (+)
۳ ساعت	آموزش یادگیری عمیق (Deep learning) - کلیک کنید (+)
۳ ساعت	آموزش برنامه نویسی یادگیری عمیق با پایتون (TensorFlow و Keras) - کلیک کنید (+)
۲۱ ساعت	مجموعه آموزش های سیستم های فازی در متلب - کلیک کنید (+)
۱۴ ساعت	مجموعه آموزش های تئوری و عملی الگوریتم ژنتیک - کلیک کنید (+)
۱ ساعت	آموزش پیاده سازی ترکیب الگوریتم ژنتیک و PSO در متلب - کلیک کنید (+)
۱۹ ساعت	مجموعه آموزش های بهینه سازی چند هدفه در متلب - کلیک کنید (+)
۲ ساعت	آموزش حل مساله فروشنده دوره گرد با استفاده از الگوریتم ژنتیک - کلیک کنید (+)
۲ ساعت	آموزش انتخاب ویژگی یا Feature Selection - کلیک کنید (+)
۴ ساعت	آموزش انتخاب ویژگی با استفاده از الگوریتم های فراابتکاری و تکاملی - کلیک کنید (+)
۲ ساعت	آموزش تخمین خطای طبقه بندی یا Classifier Error Estimation - کلیک کنید (+)
۱ ساعت	آموزش کاهش تعداد رنگ تصاویر با استفاده از روش های خوشه بندی هوشمند - کلیک کنید (+)

مهندسی برق	
مدت زمان تقریبی	عنوان آموزش
۱۱ ساعت	آموزش الکترونیک ۱ - کلیک کنید (+)
۲۵ ساعت	آموزش حل تمرین الکترونیک ۱ - کلیک کنید (+)
۱۱ ساعت	آموزش الکترونیک ۱ (مرور و حل تست) - کلیک کنید (+)
۸ ساعت	آموزش الکترونیک ۲ - کلیک کنید (+)
۲۵ ساعت	آموزش حل تمرین الکترونیک ۲ - کلیک کنید (+)
۱۲ ساعت	آموزش الکترونیک ۲ (مرور و حل تست) - کلیک کنید (+)
۱۵ ساعت	آموزش الکترونیک ۳ - کلیک کنید (+)
۱۲ ساعت	آموزش الکترونیک ۳ (مرور و حل مساله) - کلیک کنید (+)
۱۴ ساعت	آموزش مبانی مهندسی برق ۱ - کلیک کنید (+)
۱۴ ساعت	آموزش مبانی مهندسی برق ۱ (مرور و حل مساله) - کلیک کنید (+)
۱۶ ساعت	آموزش مبانی مهندسی برق ۲ - کلیک کنید (+)
۲۴ ساعت	آموزش مدارهای الکتریکی ۱ - کلیک کنید (+)
۱۷ ساعت	آموزش حل تمرین مدارهای الکتریکی ۱ - کلیک کنید (+)
۱۴ ساعت	آموزش مدارهای الکتریکی ۱ (مرور و حل تست) - کلیک کنید (+)
۱۱ ساعت	آموزش مدارهای الکتریکی ۲ - کلیک کنید (+)
۱۸ ساعت	آموزش مدارهای الکتریکی ۲ (حل تمرین) - کلیک کنید (+)
۱۳ ساعت	آموزش مدارهای الکتریکی ۲ (مرور و حل تست) - کلیک کنید (+)
۹ ساعت	آموزش هندسه فینسلر (Finsler Geometry) - کلیک کنید (+)
۱۶ ساعت	آموزش زبان تخصصی مهندسی برق (همراه با حل نمونه سوالات کنکور ارشد) - کلیک کنید (+)
۲ ساعت	آموزش مروری ماشین‌های الکتریکی ۱ - کلیک کنید (+)
۱۴ ساعت	آموزش ماشین‌های الکتریکی ۱ - کلیک کنید (+)
۱۶ ساعت	آموزش ماشین‌های الکتریکی ۱ (مرور و حل تست) - کلیک کنید (+)
۱۷ ساعت	آموزش ماشین‌های الکتریکی ۲ - کلیک کنید (+)
۱۳ ساعت	آموزش ماشین‌های الکتریکی ۲ (مرور و حل تست کنکور ارشد) - کلیک کنید (+)
۱۱ ساعت	آموزش ماشین‌های الکتریکی ۳ - کلیک کنید (+)
۹ ساعت	آموزش مدارهای مخبراتی - کلیک کنید (+)
۹ ساعت	آموزش ماشین‌های الکتریکی مخصوص - کلیک کنید (+)
۱۴ ساعت	آموزش کنترل موتورهای الکتریکی صنعتی ۱ - کلیک کنید (+)
۱۷ ساعت	آموزش کنترل موتورهای الکتریکی صنعتی ۲ - کلیک کنید (+)
۱۵ ساعت	آموزش کنترل صنعتی - حل مساله - کلیک کنید (+)

مهندسی برق (ادامه از صفحه قبل)	
مدت زمان تقریبی	عنوان آموزش
۹ ساعت	مجموعه آموزش‌های مدارهای منطقی (طراحی دیجیتال) - کلیک کنید (+)
۱۲ ساعت	آموزش مبانی الکترونیک دیجیتال - بخش اول - کلیک کنید (+)
۱۱ ساعت	آموزش مبانی الکترونیک دیجیتال - بخش دوم - کلیک کنید (+)
۲۰ ساعت	آموزش الکترومغناطیس مهندسی - کلیک کنید (+)
۱۸ ساعت	آموزش الکترومغناطیس مهندسی (مرور و حل تست کنکور ارشد) - کلیک کنید (+)
۹ ساعت	آموزش الکترومغناطیس ۱ - کلیک کنید (+)
۳ ساعت	آموزش مقدماتی شبکه‌های هوشمند (Smart grid) انرژی الکتریکی - کلیک کنید (+)
۵ ساعت	آموزش کاربردی میکاترونیک - محاسبات پایه طراحی درب‌های خودکار - کلیک کنید (+)
۲ ساعت	آموزش کامسول مولتی فیزیکس - COMSOL Multiphysics (مباحث منتخب) - کلیک کنید (+)
۴ ساعت	آموزش نرم افزار COMSOL Multiphysics برای پدیده‌های الکترومغناطیس - کلیک کنید (+)
۸ ساعت	آموزش شبیه سازی مدولاسیون دیجیتال در متلب (MATLAB) - کلیک کنید (+)
۴ ساعت	آموزش نرم افزار سیلواکو (SILVACO) برای شبیه سازی افزاره‌های نیمه رسانا - کلیک کنید (+)

ابزار مدیریت وبسایت	
مدت زمان تقریبی	عنوان آموزش
۱ ساعت	آموزش کار با cPanel - کلیک کنید (+)
۱ ساعت	آموزش کار با کنترل پنل plesk برای ارائه خدمات وب - کلیک کنید (+)
۱ ساعت	آموزش مدیریت هاست با دایرکت ادمین (DirectAdmin) - کلیک کنید (+)
۳ ساعت	آموزش جوملا (Joomla) برای راه اندازی و مدیریت وب سایت - کلیک کنید (+)
۱ ساعت	آموزش نرم افزار Mobirise برای ساخت وب سایت بدون کدنویسی - کلیک کنید (+)
۸ ساعت	آموزش مقدماتی Oracle APEX برای تولید و توسعه نرم افزارهای تحت وب - کلیک کنید (+)
۱۱ ساعت	آموزش وردپرس (WordPress) - مقدماتی - کلیک کنید (+)
۶ ساعت	مجموعه آموزش‌های راه اندازی سایت و کار با وردپرس - تکمیلی - کلیک کنید (+)
۳ ساعت	آموزش گوگل آنالیتیکس (Google Analytics) برای تحلیل آمار وب سایت - کلیک کنید (+)
۲ ساعت	آموزش آنالیز و بهینه سازی سایت با GTmetrix - کلیک کنید (+)
۱۰ ساعت	آموزش تست نفوذ در وب و راه‌های مقابله با آن - کلیک کنید (+)

طراحی و گرافیک	
مدت زمان تقریبی	عنوان آموزش
۱۲ ساعت	آموزش فتوشاپ (Photoshop) - مقدماتی - کلیک کنید (+)
۱۴ ساعت	مجموعه آموزش‌های کاربردی طراحی و گرافیک با نرم افزار کورل (CorelDRAW) - کلیک کنید (+)
۱۰ ساعت	آموزش کورل پیشرفته - کلیک کنید (+)
۱۰ ساعت	آموزش نرم افزار طراحی گرافیکی Adobe Illustrator - کلیک کنید (+)
۵ ساعت	آموزش نرم افزار طراحی گرافیکی Adobe Illustrator - تکمیلی - کلیک کنید (+)
۱ ساعت	آموزش مقایسه نرم افزارهای گرافیکی Photoshop, CorelDRAW, Illustrator - کلیک کنید (+)
۲ ساعت	آموزش مقدماتی طراحی کاراکتر در Adobe Illustrator (کاراکتر پلیس) - کلیک کنید (+)
۸ ساعت	آموزش نرم افزار Blender برای طراحی و مدل سازی سه بعدی - کلیک کنید (+)
۱۴ ساعت	آموزش مدل سازی کاراکتر در بلندر - ایده پردازی تا ریگ - کلیک کنید (+)
۱ ساعت	آموزش طراحی فلت (Flat design) با کورل (CorelDRAW) - کلیک کنید (+)
۳ ساعت	آموزش فتوشاپ برای استفاده در صنعت چاپ و تبلیغات - کلیک کنید (+)
۳ ساعت	آموزش تایپوگرافی (Typography) - سایه زدن حروف در فتوشاپ (Photoshop) - کلیک کنید (+)
۱ ساعت	آموزش پروژه محور فتوشاپ (Photoshop) - ساخت ماسک (Mask) فانتزی - کلیک کنید (+)
۲ ساعت	آموزش آشنایی با پنل Adjustments در فتوشاپ - کلیک کنید (+)
۳ ساعت	آموزش ساخت جلوه های ویژه در فتوشاپ - کلیک کنید (+)
۶ ساعت	آموزش Adobe Lightroom (ادوبی لایت روم) برای ویرایش و روتوش تصاویر - کلیک کنید (+)
۲ ساعت	آموزش Adobe Dimension برای ایجاد تصاویر سه بعدی محصولات و تبلیغات - کلیک کنید (+)
۲ ساعت	آموزش Adobe Bridge (ادوبی بریج) برای مدیریت فایل های چندرسانه ای - کلیک کنید (+)
۴ ساعت	آموزش ایندیزاین (InDesign) برای چاپ و صفحه آرایی کتاب و مجلات - کلیک کنید (+)
۲۲ ساعت	آموزش نرم افزار SketchBook Pro (اسکچ بوک) برای طراحی آناتومی حیوانات - کلیک کنید (+)
۳ ساعت	آموزش طراحی مفهومی (Conceptual) خودرو با SketchBook - کلیک کنید (+)
۳ ساعت	آموزش نقاشی سه بعدی با نرم افزار Paint 3D ویندوز - کلیک کنید (+)
۳ ساعت	آموزش مبانی هنرهای تجسمی ۱ - کلیک کنید (+)
۱ ساعت	آموزش اصول طراحی نشانه (Sign Design) - کلیک کنید (+)
۹ ساعت	آموزش نقاشی دیجیتال با Photoshop و Mischieff - کلیک کنید (+)
۲ ساعت	آموزش طراحی لوگو با Adobe Illustrator - کلیک کنید (+)
۳ ساعت	آموزش طراحی محیط بازی های دو بعدی با نرم افزار Adobe Illustrator و Photoshop - کلیک کنید (+)

طراحی و گرافیک (ادامه از صفحه قبل)	
مدت زمان تقریبی	عنوان آموزش
۱ ساعت	آموزش طراحی لوگو - آشنایی با مفاهیم و ایده پردازی (همراه با مثال عملی) - کلیک کنید (+)
۲ ساعت	آموزش طراحی برجسب و کاور CD و DVD - کلیک کنید (+)
۵ ساعت	آموزش ابزار سه بعدی در فتوشاپ (Photoshop) - کلیک کنید (+)
۲ ساعت	آموزش فتوشاپ سه بعدی - طراحی و متحرک سازی - کلیک کنید (+)
۲ ساعت	آموزش موتور رندر آرنولد (Arnold Renderer) برای ایجاد جلوه های ویژه در DS Max ^۳ - کلیک کنید (+)
۵ ساعت	آموزش کاربرد فتوشاپ در معماری - کلیک کنید (+)

روباتیک	
مدت زمان تقریبی	عنوان آموزش
۲ ساعت	آموزش رباتیک و ربات های سری - کلیک کنید (+)
۹ ساعت	آموزش برنامه نویسی آردوینو (Arduino) با محوریت پروژه های رباتیک - مقدماتی - کلیک کنید (+)
۷ ساعت	آموزش برنامه نویسی آردوینو (Arduino) - تکمیلی - کلیک کنید (+)
۳ ساعت	آموزش سینماتیک مستقیم و معکوس روبات ها - کلیک کنید (+)
۵ ساعت	آموزش مبانی ربات های برنامه پذیر - کلیک کنید (+)
۱۶ ساعت	آموزش شبیه سازی و کنترل ربات با اندروید (Android) - کلیک کنید (+)
۱۳ ساعت	آموزش مکترونیک کاربردی ۱ - کلیک کنید (+)
۷ ساعت	آموزش مکترونیک کاربردی ۲ - کلیک کنید (+)
۵ ساعت	آموزش کاربردی مکترونیک - محاسبات پایه طراحی درب های خودکار - کلیک کنید (+)
۱۹ ساعت	آموزش برد آردوینو (Arduino) با انجام پروژه های عملی - کلیک کنید (+)
۵ ساعت	آموزش پروژه محور آردوینو با سی شارپ و اندروید - کنترل موتور پله ای - کلیک کنید (+)
۵ ساعت	آموزش پیاده سازی سیستم های کنترلی با آردوینو - کلیک کنید (+)

فصل ۵:

ابهام - ساده سازی گرامر - فرم‌های نرمال

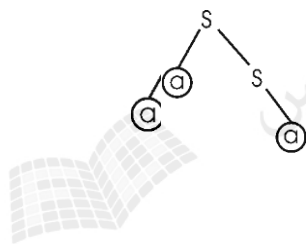
ابهام در گرامر و زبان

گرامر مستقل از متن G در صورتی مبهم خوانده می‌شود که یک رشته $w \in L(G)$ وجود داشته باشد که حداقل دو درخت اشتقاق مجزا داشته باشد. به بیان دیگر، ابهام به طور ضمنی به معنای وجود دو یا چند اشتقاق چپ‌ترین یا راست‌ترین، نیز می‌باشد.

مثال

آیا گرامر $S \rightarrow aS \mid aa \mid a$ مبهم است؟

حل: بله - چون به طور نمونه برای تولید رشته aa ، دو درخت اشتقاق وجود دارد:



مثال

آیا گرامر زیر مبهم است؟

$$S \rightarrow AB \mid aaB$$

$$A \rightarrow a \mid Aa$$

$$B \rightarrow b$$

حل: بله - چون به طور نمونه برای تولید رشته abb ، دو اشتقاق چپ وجود دارد:

$$1: S \Rightarrow aaB \Rightarrow aab$$

$$2: S \Rightarrow AB \Rightarrow AaB \Rightarrow aaB \Rightarrow aab$$

مثال

آیا گرامر $\lambda \mid aSbS \mid bSaS \mid S \rightarrow aSbS$ مبهم است؟حل: بله - چون به طور نمونه برای تولید رشته $abab$ ، دو اشتقاق چپ وجود دارد:

$$1: S \Rightarrow aSbS \Rightarrow abSaSbS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow abab$$

$$2: S \Rightarrow aSbS \Rightarrow abS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow abab$$



مثال

آیا گرامر $\lambda \mid SS \mid aSb \mid S \rightarrow aSb$ مبهم است؟حل: بله - چون به طور نمونه برای تولید رشته ab ، دو اشتقاق چپ وجود دارد:

$$1: S \Rightarrow aSb \Rightarrow ab$$

$$2: S \Rightarrow SS \Rightarrow aSbS \Rightarrow abS \Rightarrow ab$$

مثال

آیا زبان تولید شده توسط گرامر مبهم $\lambda \mid SS \mid aSb \mid S \rightarrow aSb$ ، مبهم است؟

حل: خیر - چون می‌توان یک گرامر غیر مبهم نظیر این گرامر نوشت:

$$S \rightarrow A \mid \lambda$$

$$A \rightarrow aAb \mid ab \mid AA$$



مثال

آیا گرامر زیر مبهم است؟

$$S \rightarrow aB \mid A$$

$$A \rightarrow aA \mid \lambda$$


$$B \rightarrow bB \mid a$$

حل: بله - چون به طور نمونه برای تولید رشته aa ، دو اشتقاق چپ وجود دارد:

$$1: S \Rightarrow A \Rightarrow aA \Rightarrow aaA \Rightarrow aa$$

$$2: S \Rightarrow aB \Rightarrow aa$$

بنابراین گرامر منظم نیز می‌تواند مبهم باشد.

خانواده زبان‌های مستقل از متن غیر مبهم، تحت اجتماع بسته نمی‌باشند. 

مثال

نشان دهید زبان $L = \{a^n b^n c^m : n, m \geq 0\} \cup \{a^n b^m c^m : n, m \geq 0\}$ ، یک زبان مستقل از متن ذاتا مبهم است. (این زبان را می‌توان به صورت $L = \{0^i 1^j 2^k : i = j \text{ or } j = k\}$ نیز نشان داد.)

حل: گرامر این زبان به صورت زیر است. زبان $\{a^n b^n c^m : n, m \geq 0\}$ توسط $S \rightarrow S_1$ و زبان $\{a^n b^m c^m : n, m \geq 0\}$ توسط $S \rightarrow S_2$ تولید می‌شود.

$$S \rightarrow S_1 \mid S_2$$

$$S_1 \rightarrow S_1 c \mid A$$


$$A \rightarrow aAb \mid \lambda$$


$$S_2 \rightarrow aS_2 \mid B$$


$$B \rightarrow bBc \mid \lambda$$

این گرامر مبهم است، چون برای رشته $a^n b^n c^n$ دو اشتقاق مجزا وجود دارد.

تذکره: هر دو زبان $\{a^n b^n c^m : n, m \geq 0\}$ و $\{a^n b^m c^m : n, m \geq 0\}$ مستقل از متن غیر مبهم هستند. ■

اگر L مستقل از متن بوده و هر گرامر تولید کننده L ، مبهم باشد، آنگاه این زبان ذاتا مبهم است. 

زبان‌های منظم نمی‌توانند ذاتا مبهم باشند. 

تمام S -گرامرها، فاقد ابهام هستند. 

ساده سازی گرامرهای مستقل از متن

در تعریف گرامرهای مستقل از متن، هیچ محدودیتی برای سمت راست قانون در نظر گرفته نشده است که این آزادی در برخی استدلال‌ها، مشکل ایجاد می‌کند. در بسیاری موارد بهتر است محدودیت شدیدی قائل شویم.

حذف متغیرها و قوانین بی فایده

یک متغیر مفید است اگر و تنها اگر در حداقل یک اشتقاق حضور داشته باشد. عوامل غیرمفید بودن یک متغیر عبارتند از:

۱- قابل دسترس نبودن از طریق متغیر شروع گرامر

۲- ناتوانی در اشتقاق یک رشته پایانی

قانونی که شامل یک متغیر بی فایده باشد، قانون بی فایده نامیده می شود.

گرامر $G = (V, T, S, P)$ را یک گرامر مستقل از متن فرض کنید. آنگاه گرامر هم ارز $\hat{G} = (\hat{V}, \hat{T}, S, \hat{P})$ وجود دارد که شامل هیچ متغیر یا قانون بی فایده نمی باشد.

مثال

در گرامر زیر متغیر B بی فایده می باشد، چون از طریق متغیر شروع یعنی S، قابل دسترس نمی باشد:

$$S \rightarrow A$$

$$A \rightarrow aA \mid \lambda$$

$$B \rightarrow bA$$

پس قانون $B \rightarrow bA$ بی فایده است و می توان آن را حذف کرد، بدون اینکه تغییری در زبان ایجاد کند. ■

مثال

در گرامر زیر، کدام متغیر بی فایده است؟

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

$$A \rightarrow aA$$

حل: متغیر A بی فایده است، چون نمی تواند رشته ای از پایانی ها را تولید کند. بنابراین قانون

$S \rightarrow A$ و $A \rightarrow aA$ را می توان حذف کرد، بدون اینکه تغییری در زبان ایجاد شود. ■

مثال

در گرامر زیر کدام متغیرها بی فایده هستند.

$$S \rightarrow aS \mid A \mid C$$

$$A \rightarrow a$$

$$B \rightarrow aa$$

$$C \rightarrow aCb$$

حل: متغیر C بی فایده است، چون یک رشته پایانی را تولید نمی کند. همچنین متغیر B بی فایده است، چون از متغیر شروع قابل دستیابی نمی باشد. بنابراین می توان متغیرهای C و B و قوانین مربوط به آنها را حذف کرد. گرامر نهایی به صورت زیر می باشد:

$$S \rightarrow aS \mid A$$

$$A \rightarrow a$$



مثال

گرامر زیر را ساده کنید.

$$S \rightarrow aAb \mid bBa \mid bCa$$

$$A \rightarrow aaAb \mid ab$$

$$B \rightarrow bBa \mid a$$

$$C \rightarrow aC \mid bC$$

حل: در گرامر داده شده، متغیر C، غیر مفید است و قاعده هایی که از C استفاده می کنند را می توان از گرامر حذف کرد. بنابراین گرامر به صورت زیر می باشد:

$$S \rightarrow aAb \mid bBa$$

$$A \rightarrow aaAb \mid ab$$

$$B \rightarrow bBa \mid a$$



مثال

در گرامر زیر متغیرهای بی فایده را حذف کنید.

$$S \rightarrow AC \mid BS \mid B$$

$$A \rightarrow aA \mid aF$$

$$B \rightarrow CF \mid b$$

$$C \rightarrow cC \mid D$$

$$D \rightarrow aD \mid BD \mid C$$

$$E \rightarrow aA \mid BSA$$

$$F \rightarrow bB \mid b$$

حل: ابتدا متغیرهایی را که به رشته‌ای از الفبا نمی‌رسند یعنی C و D را حذف می‌کنیم:

$$S \rightarrow BS \mid B$$

$$A \rightarrow aA \mid aF$$

$$B \rightarrow b$$

$$E \rightarrow aA \mid BSA$$

$$F \rightarrow bB \mid b$$

سپس متغیرهایی را که نمی‌توان از S به آنها رسید، یعنی A و E و F را حذف می‌کنیم:

$$S \rightarrow BS \mid B$$

$$B \rightarrow b$$

حذف قوانین λ

هر قانونی از یک گرامر مستقل از متن به فرم $A \rightarrow \lambda$ را قانون λ می‌گویند. این قوانین در بعضی مواقع نامطلوب می‌باشند.

هر متغیر A که اشتقاق $\lambda \xRightarrow{*} A$ برای آن امکان پذیر باشد را متغیر میرا می‌نامند.

برخی گرامرها زبانهایی را تولید می‌کنند که هر چند فاقد λ هستند، تعدادی متغیر میرا یا قانون λ در آنها وجود دارند. در این موارد، می‌توان قوانین λ را حذف کرد.

به ازای هر گرامر مستقل از متن، یک گرامر هم‌ارز فاقد قانون λ ، وجود دارد.

مثال

گرامر زیر، زبان $\{a^n b^n : n \geq 1\}$ را تولید می‌کند. این زبان فاقد λ می‌باشد.

$$S \rightarrow aAb$$

$$A \rightarrow aAb \mid \lambda$$

برای حذف قانون $A \rightarrow \lambda$ ، دو قانون جدید که با جایگزینی λ در A های سمت راست بدست آمده‌اند را به گرامر اضافه می‌کنیم:

$$S \rightarrow aAb \mid ab$$

$$A \rightarrow aAb \mid ab$$

■

مثال

قوانین λ را در گرامر زیر حذف کنید.

$$S \rightarrow AB$$

$$A \rightarrow aA \mid \lambda$$

$$B \rightarrow bB \mid \lambda$$

حل: ابتدا $A \rightarrow \lambda$ را حذف می‌کنیم:

$$S \rightarrow AB \mid B$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

سپس $B \rightarrow \lambda$ را حذف می‌کنیم:

$$S \rightarrow AB \mid A \mid B \mid \lambda$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow bB \mid b$$

تذکر مهم: نیازی به حذف قانون $S \rightarrow \lambda$ نمی‌باشد، چون S ، متغیر شروع است. ■

مثال

در گرامر زیر، قوانین λ را حذف کنید؟

$$S \rightarrow ABaC$$

$$A \rightarrow BC$$

$$B \rightarrow b \mid \lambda$$

$$C \rightarrow D \mid \lambda$$

$$D \rightarrow d$$

حل: ابتدا $B \rightarrow \lambda$ را حذف می‌کنیم:

$$S \rightarrow ABaC \mid AaC$$

$$A \rightarrow BC \mid C$$

$$B \rightarrow b$$

$$C \rightarrow D \mid \lambda$$

$$D \rightarrow d$$

سپس $C \rightarrow \lambda$ را حذف می‌کنیم:

$$S \rightarrow ABaC \mid AaC \mid ABa \mid Aa$$

$$A \rightarrow BC \mid C \mid B \mid \lambda$$

$$B \rightarrow b$$

$$C \rightarrow D$$

$$D \rightarrow d$$

در نهایت $A \rightarrow \lambda$ را حذف می‌کنیم:

$$S \rightarrow ABaC \mid AaC \mid ABa \mid Aa \mid BaC \mid aC \mid Ba \mid a$$

$$A \rightarrow BC \mid B \mid C$$

$$B \rightarrow b$$

$$C \rightarrow D$$

$$D \rightarrow d$$

■

حذف قوانین واحد

هر قانونی از یک گرامر مستقل از متن به فرم $A \rightarrow B$ که در آن $A, B \in V$ ، قانون واحد یا یک نامیده می‌شود. این قوانین گاهی اوقات نامطلوب هستند و باید حذف شوند.

مثال

از گرامر زیر قوانین واحد را حذف کنید.

$$S \rightarrow Aa \mid B$$

$$B \rightarrow A \mid bb$$

$$A \rightarrow bc \mid a \mid B$$

حل:

$$S \rightarrow Aa \mid bc \mid a \mid bb$$

$$B \rightarrow bc \mid a \mid bb$$

$$A \rightarrow bc \mid a \mid bb$$

توجه کنید که در اثر حذف قوانین واحد، متغیر B و قوانین مربوط به آن بی‌فایده شده‌اند. ■

ممکن است حذف قوانین λ ، باعث تولید قوانین واحد شود که قبلاً وجود نداشته‌اند. ✍

زبان L را یک زبان مستقل از متن فاقد λ فرض کنید. آنگاه یک گرامر مستقل از متن وجود خواهد داشت که L را تولید کرده و فاقد هرگونه قانون بی‌فایده، قانون λ و قانون واحد باشد. ✍

مثال

در گرامر زیر قوانین λ را حذف کنید.

$$S \rightarrow aA$$

$$A \rightarrow BB$$

$$B \rightarrow aBb \mid \lambda$$

حل: با حذف $\lambda \rightarrow B$ گرامر زیر بدست می‌آید:

$$\begin{aligned} S &\rightarrow aA \mid a \\ A &\rightarrow BB \mid B \\ B &\rightarrow aBb \mid ab \end{aligned}$$

در این گرامر قانون واحد $A \rightarrow B$ تولید شده که قبلا وجود نداشت. ■

مثال

در گرامر زیر تمامی قوانین λ ، واحد و بی‌فایده را حذف کنید.

$$\begin{aligned} S &\rightarrow aA \mid aBB \\ A &\rightarrow aaA \mid \lambda \\ B &\rightarrow bC \mid bbC \\ C &\rightarrow B \end{aligned}$$

حل: ابتدا قانون $\lambda \rightarrow A$ را حذف می‌کنیم:

$$\begin{aligned} S &\rightarrow aA \mid aBB \mid a \\ A &\rightarrow aaA \mid aa \\ B &\rightarrow bC \mid bbC \\ C &\rightarrow B \end{aligned}$$

سپس قانون واحد $B \rightarrow C$ را حذف می‌کنیم:

$$\begin{aligned} S &\rightarrow aA \mid aBB \mid a \\ A &\rightarrow aaA \mid aa \\ B &\rightarrow bC \mid bbC \\ C &\rightarrow bC \mid bbC \end{aligned}$$

نهایتا اینکه B و C بی‌فایده هستند، بنابراین داریم:

$$\begin{aligned} S &\rightarrow aA \mid a \\ A &\rightarrow aaA \mid aa \end{aligned}$$

زبان تولید شده بوسیله این گرامر $L((aa)^*a)$ می‌باشد. ■

فرم‌های نرمال گرامر مستقل از متن

برای گرامر مستقل از متن، دو فرم نرمال چامسکی و گریباخ وجود دارد.

فرم نرمال چامسکی

یک گرامر مستقل از متن در صورتی در فرم نرمال چامسکی قرار دارد که تمام قوانین آن به یکی از دو فرم $A \rightarrow BC$ و یا $A \rightarrow a$ باشد. که در آن A, B, C عضو V بوده و a عضو T است.

مثال

گرامر زیر در فرم نرمال چامسکی قرار دارد:

$$S \rightarrow AS \mid BS \mid a$$

$$A \rightarrow SA \mid a$$

$$B \rightarrow SB \mid b$$



مثال

گرامر زیر را به فرم نرمال چامسکی تبدیل کنید.

$$S \rightarrow ABa$$

$$A \rightarrow aab$$

$$B \rightarrow Ac$$

حل:

قدم اول : معرفی متغیرهای جدید X, Y, Z	قدم دوم : نرمال کردن قانون اول و سوم
$S \rightarrow ABX$	$S \rightarrow AT$
$X \rightarrow a$	$T \rightarrow BX$
$A \rightarrow XXY$	$A \rightarrow XF$
$Y \rightarrow b$	$F \rightarrow XY$
$B \rightarrow AZ$	$B \rightarrow AZ$
$Z \rightarrow c$	$X \rightarrow a, Y \rightarrow b, Z \rightarrow c$

مثال

گرامر زیر را به فرم نرمال چامسکی تبدیل کنید.

$$S \rightarrow aAB$$

$$A \rightarrow aA \mid b$$

$$B \rightarrow bB \mid b$$

حل:

قدم اول	قدم دوم
$S \rightarrow TAB$	$S \rightarrow TK$
$T \rightarrow a$	$K \rightarrow AB$
$A \rightarrow TA \mid b$	$T \rightarrow a$
$B \rightarrow FB \mid b$	$A \rightarrow TA \mid b$
$F \rightarrow b$	$B \rightarrow FB \mid b$
	$F \rightarrow b$

مثال

گرامر زیر را به فرم نرمال چامسکی تبدیل کنید.

$$S \rightarrow aAbB \mid ab$$

$$A \rightarrow ABS \mid a$$

$$B \rightarrow bb$$

حل:

قدم اول	قدم دوم	قدم سوم
$S \rightarrow TAFB \mid TF$	$S \rightarrow TK \mid TF$	$S \rightarrow TK \mid TF$
$A \rightarrow ABS \mid a$	$K \rightarrow AFB$	$K \rightarrow AX$
$B \rightarrow FF$	$A \rightarrow AU \mid a$	$X \rightarrow FB$
$T \rightarrow a$	$U \rightarrow BS$	$A \rightarrow AU \mid a$
$F \rightarrow b$	$B \rightarrow FF$	$U \rightarrow BS$
	$T \rightarrow a$	$B \rightarrow FF$
	$F \rightarrow b$	$T \rightarrow a$
		$F \rightarrow b$

مثال

گرامر زیر را به فرم نرمال چامسکی تبدیل کنید.

$$\begin{aligned} S &\rightarrow abAB \\ A &\rightarrow bAB \mid \lambda \\ B &\rightarrow BAa \mid A \mid \lambda \end{aligned}$$

حل: ابتدا قواعد λ را حذف می‌کنیم:

حذف $A \rightarrow \lambda$	حذف $B \rightarrow \lambda$
$S \rightarrow abAB \mid abB$	$S \rightarrow abAB \mid abB \mid abA \mid ab$
$A \rightarrow bAB \mid bB$	$A \rightarrow bAB \mid bB \mid bA \mid b$
$B \rightarrow BAa \mid A \mid Ba \mid \lambda$	$B \rightarrow BAa \mid A \mid Ba \mid Aa \mid a$

سپس قاعده $B \rightarrow A$ را حذف می‌کنیم:

$$\begin{aligned} S &\rightarrow abAB \mid abB \mid abA \mid ab \\ A &\rightarrow bAB \mid bB \mid bA \mid b \\ B &\rightarrow BAa \mid Ba \mid Aa \mid a \mid bAB \mid bB \mid bA \mid b \end{aligned}$$

در نهایت گرامر را به فرم نرمال چامسکی در می‌آوریم:

قدم اول: معرفی متغیرهای جدید Y, X	قدم دوم: معرفی متغیرهای P, T, K, N, M جدید
$\begin{aligned} S &\rightarrow XYAB \mid XYB \mid XYA \mid XY \\ A &\rightarrow YAB \mid YB \mid YA \mid b \\ B &\rightarrow BAX \mid BX \mid AX \mid a \mid YAB \mid YB \mid YA \mid b \\ X &\rightarrow a \\ Y &\rightarrow b \end{aligned}$	$\begin{aligned} S &\rightarrow XM \mid XN \mid XK \mid XY \\ M &\rightarrow YT \\ T &\rightarrow AB \\ N &\rightarrow YB \\ A &\rightarrow YT \mid YB \mid YA \mid b \\ B &\rightarrow BP \mid BX \mid AX \mid a \mid YT \mid YB \mid YA \mid b \\ P &\rightarrow AX \\ X &\rightarrow a \\ Y &\rightarrow b \end{aligned}$

در فرم نرمال چامسکی، تولید رشته‌ای به طول n دارای اشتقاقی با طول $2n-1$ می‌باشد. ($n \geq 1$) اشتقاق از $A \rightarrow a$ و $n-1$ اشتقاق از $A \rightarrow BC$

فرض کنید G یک گرامر مستقل از متن (CFG) در فرم نرمال چامسکی با b متغیر باشد. در این صورت اگر G بتواند رشته‌ای را با تعداد گام‌های اشتقاق بیشتر از b تولید کند، آنگاه $L(G)$ نامحدود است.

فرم نرمال گریباخ

یک گرامر مستقل از متن در فرم نرمال گریباخ است هرگاه تمام قوانین آن به فرم $A \rightarrow aX$ باشند، که در آن، $x \in V^*$, $a \in T$.

مثال

گرامر $S \rightarrow abSb \mid aa$ را به فرم نرمال گریباخ تبدیل کنید.

حل: متغیرهای جدید A و B را معرفی می‌کنیم که مترادف با a و b هستند:

$$S \rightarrow aBSB \mid aA$$

$$A \rightarrow a$$

$$B \rightarrow b$$



مثال

گرامر $S \rightarrow ab \mid aS \mid aaS$ را به فرم نرمال گریباخ تبدیل کنید.

$$S \rightarrow aX \mid aS \mid aYS$$

$$X \rightarrow b$$

$$Y \rightarrow a$$



مثال

گرامر زیر را به فرم نرمال گریباخ تبدیل کنید.

$$S \rightarrow AB$$

$$A \rightarrow aA \mid bB \mid b$$

$$B \rightarrow b$$

حل: قاعده $S \rightarrow AB$ با تعریف فرم نرمال گریباخ مغایر است. بنابراین به جای A در این قانون از قوانین جایگزین آن استفاده می‌کنیم:

$$S \rightarrow aAB \mid bBB \mid bB$$

$$A \rightarrow aA \mid bB \mid b$$

$$B \rightarrow b$$

به ازای هر گرامر مستقل از متن در صورتیکه شامل ϵ نباشد، یک گرامر معادل به فرم نرمال گریباخ وجود دارد.

در فرم نرمال گریباخ، برای تولید رشته‌ای به طول n ، به اشتقاقی با طول n نیاز است. چون در هر مرحله یکی از نمادهای رشته ایجاد می‌شود.

مثال

گرامر زیر را به فرم نرمال گریباخ تبدیل کنید.

$$S \rightarrow ABb \mid a$$

$$A \rightarrow aaA \mid B$$

$$B \rightarrow bAb$$

حل: ابتدا قاعده $A \rightarrow B$ را حذف می‌کنیم:

$$S \rightarrow ABb \mid a$$

$$A \rightarrow aaA \mid bAb$$

$$B \rightarrow bAb$$

سپس متغیر A در $B \rightarrow bAb$ را جایگزین می‌کنیم:

$$S \rightarrow aaABb \mid bAbBb \mid a$$

$$A \rightarrow aaA \mid bAb$$

$$B \rightarrow bAb$$

سپس به فرم نرمال گریباخ تبدیل می‌کنیم:

$$S \rightarrow aXABY \mid bAYBY \mid a$$

$$A \rightarrow aXA \mid bAY$$

$$B \rightarrow bAY$$

$$X \rightarrow a$$

$$Y \rightarrow b$$



تمرین فصل ۵

۱- درخت اشتقاق را تعریف کنید.

۲- با توجه به گرامر زیر، درخت اشتقاق برای تولید رشته $abbbb$ ، را رسم کنید.

$$S \rightarrow aAB$$

$$A \rightarrow bBb$$

$$B \rightarrow A \mid \lambda$$

۳- اگر G یک گرامر مستقل از متن باشد، آنگاه هر $w \in L(G)$ اشتقاق چپ دارد یا یک اشتقاق راست؟

۴- ترتیب حذف قواعد تولید λ ، بی فایده و واحد را از یک گرامر مستقل از متن که زبان آن فاقد λ است را مشخص کنید.

۵- از گرامر زیر، قوانین نامطلوب (λ ، واحد و بی فایده) را حذف کنید.

$$S \rightarrow aA \mid aBB$$

$$A \rightarrow aaA \mid \lambda$$

$$B \rightarrow bB \mid bbC$$

$$C \rightarrow B$$

۶- آیا حذف قوانین بی فایده لزوماً موجب تولید گرامرهای کمینه می‌شود؟

۷- قوانین بی فایده را از گرامر زیر حذف کنید.

$$S \rightarrow aA \mid a \mid B \mid C$$

$$A \rightarrow aB \mid \lambda$$

$$B \rightarrow Aa$$

$$C \rightarrow cCD$$

$$D \rightarrow ddd$$

۸- قوانین λ و بی فایده را از گرامر زیر حذف کنید.

$$S \rightarrow AaB \mid aaB$$

$$A \rightarrow \lambda$$

$$B \rightarrow aaA \mid \lambda$$

۹- گرامر زیر را به فرم نرمال چامسکی تبدیل کنید.

$$S \rightarrow aSb \mid ab$$

۱۰- گرامر زیر را به فرم نرمال چامسکی تبدیل کنید.

$$S \rightarrow aSaA \mid A$$

$$A \rightarrow abA \mid b$$

۱۱- گرامر $S \rightarrow aSb \mid ab$ را به فرم نرمال گریباخ تبدیل کنید.

۱۲- گرامر $S \rightarrow aSb \mid aS \mid aaS$ را به فرم نرمال گریباخ تبدیل کنید.

۱۳- توسط الگوریتم CYK که توسط کوک، یانگر و کسامی ابداع شد، چه موضوعی را می‌توان تصمیم‌گیری کرد؟ پیچیدگی این الگوریتم را مشخص کنید.

۱۴- اگر G گرامر مستقل از متنی باشد که هر یک از قوانین آن به فرم $A \rightarrow v$ بوده و $|v| > 1$ ، آنگاه محدوده ارتفاع درخت اشتقاق به ازای تمامی $w \in L(G)$ را مشخص کنید.

۱۵- فرض کنید $G=(V,T,S,P)$ یک گرامر مستقل از متن (CFG) بدون هیچ قانون λ یا قانون واحدی باشد. بعلاوه، فرض کنید K حداکثر تعداد سمبل‌های موجود در طرف راست قوانین P باشد. گرامر معادل در فرم نرمال چامسکی، حداکثر دارای چه تعداد قانون تولید است؟

۱۶- گرامر زیر را با BNF نمایش دهید.

$$E \rightarrow T \mid E + T$$

$$T \rightarrow F \mid T * E$$

حل تمرین فصل ۵

۱- فرض کنید $G = (V, T, S, P)$ یک گرامر مستقل از متن باشد. یک درخت مرتب، درخت اشتقاقی برای G خواهد بود اگر و تنها اگر خواص زیر را داشته باشد:

الف- ریشه دارای نام S است.

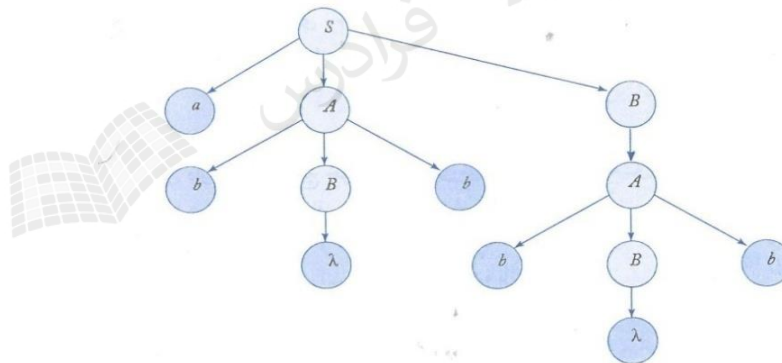
ب- هر یک از برگها دارای نامی از $T \cup \{\lambda\}$ باشد.

پ- هر یک از گره‌های میانی دارای نامی از V است.

ت- اگر یکی از گره‌ها دارای نام $A \in V$ بوده و فرزندان آن هم از چپ به راست به صورت a_1, a_2, \dots, a_n نامگذاری شوند، آنگاه P باید قانونی به فرم $A \rightarrow a_1 a_2 \dots a_n$ داشته باشد.

ث- برگ‌های دارای نام λ ، هیچ خواهر و برادری ندارند.

۲- درخت اشتقاقی تولید رشته $abbbb$:



در تمامی درختهای اشتقاقی، با شروع از ریشه که با متغیر شروع گرامر نامگذاری می‌شود و خاتمه یافتن به برگ‌هایی که پایانی‌ها و یا λ هستند، درخت تکمیل می‌شود.

۳- اگر G یک گرامر مستقل از متن باشد، آنگاه هر $w \in L(G)$ یک اشتقاق چپ و یک اشتقاق راست دارد.

۴- ترتیب حذف برابر است با:

الف- قواعد تولید λ ب- قواعد واحد ج- قواعد بی فایده

۵- مراحل کار به صورت زیر است:

۱- حذف قوانین λ	۲- حذف قوانین واحد	۳- حذف قواعد بی فایده
$S \rightarrow aA \mid a \mid aBB$ $A \rightarrow aaA \mid aa$ $B \rightarrow bB \mid bbC$ $C \rightarrow B$	$S \rightarrow aA \mid a \mid aBB$ $A \rightarrow aaA \mid aa$ $B \rightarrow bB \mid bbC$ $C \rightarrow bB \mid bbC$	$S \rightarrow aA \mid a$ $A \rightarrow aaA \mid aa$

۶- خیر- به طور نمونه در گرامر $S \rightarrow aA; A \rightarrow a$ هیچ قانون بی فایده، قانون واحد یا قانون λ به چشم نمی خورد. اما چون $S \rightarrow aa$ یکی از گرامرهای متناظر است، گرامر فوق کمینه نمی باشد.

مثالی دیگر: گرامر زیر را در نظر بگیرید:

$$S \rightarrow aAb \mid ab$$

$$A \rightarrow aAb \mid ab$$

این گرامر زبان $L = \{a^n b^n : n \geq 1\}$ را تولید می کند. اگر روال حذف قوانین بی فایده را روی این گرامر اجرا کنیم، هیچ یک از قواعد حذف نمی شود. از طرفی گرامر زیر که کمینه است، همین زبان را تولید می کند:

$$S \rightarrow aSb \mid ab$$

۷- متغیر D و قوانین متناظر با آن بی فایده است.

$$S \rightarrow aA \mid a \mid B$$

$$A \rightarrow aB \mid \lambda$$

$$B \rightarrow Aa$$

۸- گرامر بعد از حذف قوانین λ و بی فایده به صورت زیر در می آید:

$$S \rightarrow aaB \mid aB \mid aa \mid a$$

$$B \rightarrow aa$$

۹- به جای a و b از متغیرهای M و N استفاده می کنیم:

$$S \rightarrow MAN \mid MN$$

$$M \rightarrow a$$

$$N \rightarrow b$$

به جای AN از X استفاده می‌کنیم:

$$S \rightarrow MX \mid MN$$

$$X \rightarrow AN$$

$$M \rightarrow a$$

$$N \rightarrow b$$

۱۰- ابتدا قانون یکه $S \rightarrow A$ را حذف می‌کنیم:

$$S \rightarrow aSaA \mid abA \mid b$$

$$A \rightarrow abA \mid b$$

سپس گرامر را به فرم نرمال چامسکی در می‌آوریم:

$S \rightarrow MSMA \mid MNA \mid b$ $A \rightarrow MNA \mid b$ $M \rightarrow a$ $N \rightarrow b$	$S \rightarrow MX \mid MY \mid b$ $X \rightarrow SMA$ $Y \rightarrow NA$ $A \rightarrow MY \mid b$ $M \rightarrow a$ $N \rightarrow b$	$S \rightarrow MX \mid MY \mid b$ $X \rightarrow SK$ $K \rightarrow MA$ $Y \rightarrow NA$ $A \rightarrow MY \mid b$ $M \rightarrow a$ $N \rightarrow b$
---	--	--

۱۱- فرم نرمال گریباخ:

$$S \rightarrow aSX \mid aX$$

$$X \rightarrow b$$

۱۲- فرم نرمال گریباخ:

$$S \rightarrow aSX \mid aS \mid aYS$$

$$X \rightarrow b$$

$$Y \rightarrow a$$

۱۳- می‌توان عضویت یا عدم عضویت یک رشته در زبانی که بوسیله گرامری که در فرم نرمال چامسکی تولید می‌شود را تصمیم‌گیری کرد. پیچیدگی این الگوریتم $|w|^3$ می‌باشد. این الگوریتم مساله را به بخش‌های کوچکتر تقسیم می‌کند. روش کار در کتاب لینز آورده شده است.

۱۴- ارتفاع در محدوده $\log_{|v|}^{|w|} \leq h \leq \frac{|w|-1}{|v|-1}$ است.

۱۵- حداکثر تعداد قانون تولید برابر است با: $(k-1)|P|+|T|$

۱۶- از فرمی به نام BNF برای تعریف گرامرها استفاده می‌شود.

$\langle \text{expression} \rangle ::= \langle \text{term} \rangle \mid \langle \text{expression} \rangle + \langle \text{term} \rangle$

$\langle \text{term} \rangle ::= \langle \text{factor} \rangle \mid \langle \text{term} \rangle * \langle \text{factor} \rangle$



فصل ۶:

اتوماتای پشته‌ای

ماشین پشته‌ای (PDA (Push Down Automaton) علاوه بر اجزاء ماشین متناهی (FA)، دارای یک پشته نیز می‌باشد. این ماشین، علاوه بر خواندن نوار ورودی از چپ به راست، قادر به نوشتن هر تعداد نماد درون یک پشته (push) و بازیابی آنها (pop) می‌باشد. ماشین‌های پشته‌ای پذیرنده زبانهای مستقل از متن می‌باشند. ماشین پشته‌ای از ماشین متناهی قویتر است، یعنی زبان‌های بیشتری را می‌پذیرد. اتوماتای پشته‌ای بر دو نوع معین (DPDA) و نامعین (NPDA) است.

اتوماتای پشته‌ای نامعین

پذیرنده پشته‌ای نامعین بوسیله هفت تایی $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ تعریف می‌شود:

Q : مجموعه متناهی از حالت‌های داخلی واحد کنترل

Σ : الفبای ورودی

Γ : الفبای پشته

δ : تابع انتقال

q_0 : حالت شروع واحد کنترل

z : سمبل ته پشته

F : مجموعه حالات پایانی

تعریف : زبان مورد پذیرش توسط یک npda، مجموعه تمام رشته‌هایی است که npda با رسیدن به انتهای آن رشته‌ها، می‌تواند در حالت پایانی قرار گیرد.

تابع انتقال

قانون $\delta(q_i, a, A) = (q_j, B)$ ، یعنی اگر ماشین در وضعیت q_i باشد و a مقابل هد نوار خوان باشد و A بالای پشته باشد، آنگاه ماشین به حالت q_j تغییر وضعیت داده و A را از پشته pop کرده و B را push می‌کند. تابع انتقال در ماشین پشته‌ای نامعین به صورت $Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow Q \times \Gamma^*$ بیان می‌شود.

یعنی این تابع ۳ ورودی و دو خروجی دارد. ورودی‌های آن به ترتیب برابرند با: حالت جاری ماشین، سمبل خوانده شده از نوار یا λ و سمبل بالای پشته. خروجی‌های نیز عبارتند از: حالت بعدی ماشین و سمبلی که در بالای پشته push می‌شود.

تفسیر چند قانون نمونه :

۱- قانون $\delta(q_0, a, 0) = \{(q_1, 1)\}$ یعنی ماشین در وضعیت q_0 با خواندن a روی نوار به وضعیت q_1 رفته و 0 از بالای پشته pop شده و 1 در پشته push می‌شود. یعنی به جای 0 مقدار 1 قرار می‌دهیم.

۲- قانون $\delta(q_0, a, 0) = \{(q_1, 10)\}$ یعنی ماشین در وضعیت q_0 با خواندن a روی نوار به وضعیت q_1 رفته و 0 بالای پشته با 10 جایگزین می‌شود. در واقع 1 در بالای پشته push می‌شود.

۳- قانون $\delta(q_0, a, 0) = \{(q_1, \lambda)\}$ یعنی ماشین در وضعیت q_0 با خواندن a روی نوار به وضعیت q_1 رفته و 0 با λ جایگزین می‌شود. یعنی 0 از بالای پشته pop می‌شود.

مثال

نحوه عملکرد $\delta(q_0, a, z) = \{(q_1, 1z)\}$ را تشریح کنید.

حل: هرگاه واحد کنترل در حالت q_0 قرار گیرد و سمبل ورودی خوانده شده a و سمبل واقع در بالای پشته z باشد، آنگاه واحد کنترل به حالت q_1 رفته و 1 در پشته push می‌شود. ■

مثال

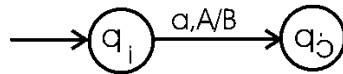
نحوه عملکرد $\delta(q_0, a, 0) = \{(q_1, 1), (q_2, \lambda)\}$ را تشریح کنید.

حل: هرگاه واحد کنترل در حالت q_0 قرار گیرد و سمبل ورودی خوانده شده a و سمبل واقع در بالای پشته 0 باشد، آنگاه یکی از دو حالت زیر اتفاق می‌افتد:

۱- واحد کنترل به حالت q_1 رفته و 0 از پشته pop شده و 1 به جای آن push می‌شود.

۲- واحد کنترل به حالت q_2 رفته و سمبل 0 از بالای پشته حذف می‌شود. ■

گراف انتقال برای $\delta(q_i, a, A) = (q_j, B)$ به صورت زیر می‌باشد:



مثال

ماشین npda با انتقالات زیر چه زبانی را می‌پذیرد؟

$$\Sigma = \{a, b\}, \Gamma = \{0, 1\}$$

$$\delta(q_0, a, z) = \{(q_1, 1z), (q_f, z)\}$$

$$\delta(q_0, \lambda, z) = \{(q_f, z)\}$$

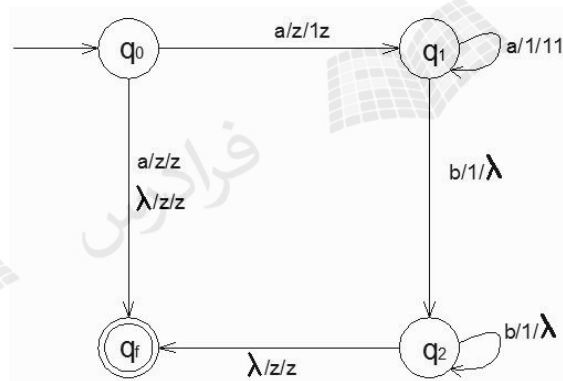
$$\delta(q_1, a, 1) = \{(q_1, 11)\}$$

$$\delta(q_1, b, 1) = \{(q_2, \lambda)\}$$

$$\delta(q_2, b, 1) = \{(q_2, \lambda)\}$$

$$\delta(q_2, \lambda, z) = \{(q_f, z)\}$$

حل: شکل ماشین به صورت زیر است:



در صورتی npda به حالت نهایی می‌رود که رشته ورودی جزء زبان $L = \{a^n b^n : n \geq 0\} \cup \{a\}$ باشد. با ذکر دو مثال این ماشین را بررسی می‌کنیم:

۱- اگر ورودی aabb باشد، با خواندن اولین a به q_1 رفته و عدد 1 در پشته push می‌کند. سپس با خواندن a بعدی، عدد 1 دیگری در پشته push کرده و در q_1 باقی می‌ماند. با خواندن b، عدد 1 بالای پشته pop شده و به حالت q_2 می‌رود. با خواندن b بعدی، عدد 1 دیگری نیز pop شده و در q_2 باقی می‌ماند. در نهایت چون رشته ورودی تمام شده و پشته خالی است، به حالت نهایی q_f می‌رود.

بنابراین توسط انتقال $\delta(q_1, a, 1) = \{(q_1, 11)\}$ پس از خواندن یک سمبل a از ورودی، یک سمبل 1 به پشته اضافه می‌شود و در انتقال $\delta(q_2, b, 1) = \{(q_2, \lambda)\}$ ، به محض برخورد با یک b از ورودی، یک سمبل 1 از پشته حذف می‌کند. به کمک این دو انتقال، تعداد a ها شمارش شده و با تعداد b ها تطبیق داده می‌شود.

۲- اگر ورودی a باشد، ماشین از q_0 به حالت نهایی q_f می‌رود.

علت نامعین بودن این ماشین این است که در حالت q_0 ، با خواندن a ، ماشین هم می‌تواند به q_1 برود و هم به q_3 .

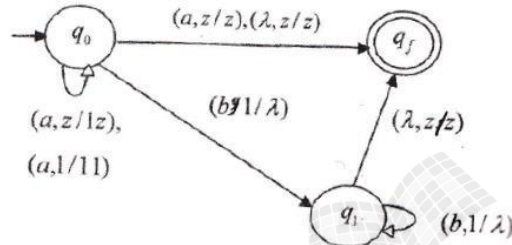
توجه: در یکی از شرایط زیر ورودی پذیرفته نمی‌شود:

۱- اگر زمانی که هنوز تعدادی b در ورودی مانده باشد و پشته خالی باشد.

۲- زمانی که b ها تمام شده و پشته خالی نباشد.

۳- در ورودی بعد از b ها، نماد a ظاهر شود.

این ماشین را با ۳ حالت به صورت زیر نیز می‌توان ساخت.



مثال

ماشین npda با انتقالات زیر چه زبانی را می‌پذیرد؟

$$\delta(q_0, a, z) = \{(q_0, 0z)\}$$

$$\delta(q_0, b, z) = \{(q_0, 1z)\}$$

$$\delta(q_0, a, 1) = \{(q_0, \lambda)\}$$

$$\delta(q_0, b, 0) = \{(q_0, \lambda)\}$$

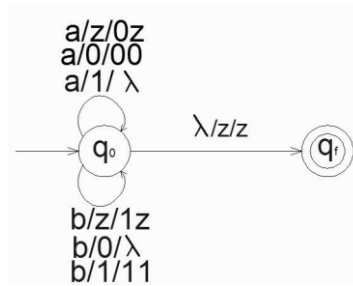
$$\delta(q_0, a, 0) = \{(q_0, 00)\}$$

$$\delta(q_0, b, 1) = \{(q_0, 11)\}$$

$$\delta(q_0, \lambda, z) = \{(q_f, z)\}$$

حل:

این ماشین زبان $L = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}$ را می‌پذیرد. یعنی رشته‌هایی که تعداد a ها با تعداد b ها در آن برابر است ولی فرقی ندارد با a شروع شود یا با b . به طور نمونه اگر رشته ورودی ba باشد، ماشین با خواندن b ، 1 در پشته $push$ می‌کند. با خواندن a ، مقدر 1 بالای پشته را pop می‌کند. در این حالت چون ورودی تمام شده و بالای پشته Z است، به حالت نهایی می‌رود. شکل این ماشین به صورت زیر است:



برای جواب دادن به این نوع سؤال‌ها، نیازی به رسم شکل نیست و برای فهم بهتر شکل رسم شد.

مثال

ای npda بسازید که زبان $L = \{a^n b^{2n} : n \geq 0\}$ را روی الفبای $\Sigma = \{a,b\}$ بپذیرد؟

حل: با دیدن هر a ، دو تا 1 در پشته $push$ کرده و با دیدن هر b ، یک عدد 1 از پشته pop می‌کنیم.

$$\delta(q_0, a, z) = \{(q_1, 11z)\}$$

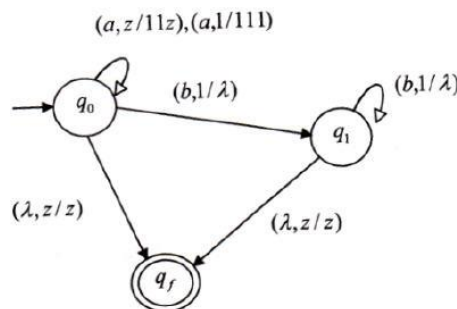
$$\delta(q_0, a, 1) = \{(q_1, 111)\}$$

$$\delta(q_0, \lambda, z) = \{(q_f, z)\}$$

$$\delta(q_0, b, 1) = \{(q_1, \lambda)\}$$

$$\delta(q_1, b, 1) = \{(q_1, \lambda)\}$$

$$\delta(q_1, \lambda, z) = \{(q_f, z)\}$$



■

مثال

ماشین npda با انتقالات زیر چه زبانی را می‌پذیرد؟

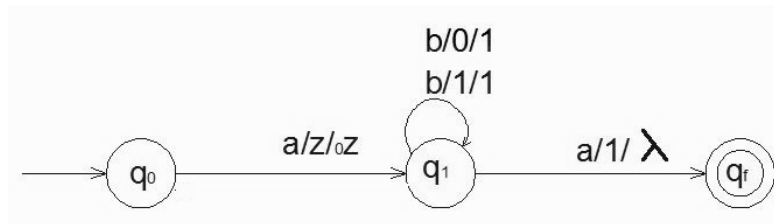
$$\delta(q_0, a, z) = \{(q_1, 0z)\}$$

$$\delta(q_1, b, 0) = \{(q_1, 1)\}$$

$$\delta(q_1, b, 1) = \{(q_1, 1)\}$$

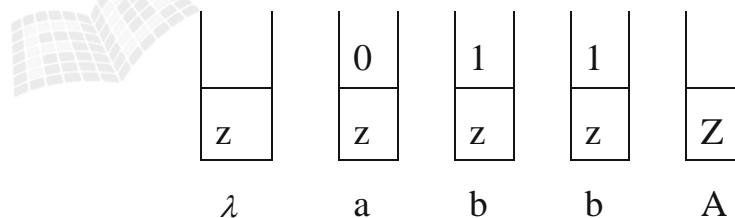
$$\delta(q_1, a, 1) = \{(q_f, \lambda)\}$$

حل: شکل ماشین به صورت زیر است:

زبان مورد نظر ab^+a می‌باشد.

در این ماشین نیازی به تطبیق دادن تعداد a با b نیست و فقط باید توالی چک شود. یعنی باید ابتدا یک a ، سپس یک یا چند b و در نهایت یک a بی‌آید.

به طور نمونه اگر رشته ورودی $abba$ باشد، با خواندن اولین a ، عدد 0 را $push$ کرده و با خواندن b ، عدد 0 را pop و 1 را $push$ می‌کند. با خواندن b بعدی تغییری در پشته نمی‌افتد. در نهایت با خواندن a ، عدد 1 بالای پشته pop شده و به حالت نهایی می‌رود:



مثال

ماشین npda با انتقالات زیر چه زبانی را می‌پذیرد؟

$$\delta(q_0, a, z) = \{(q_0, 0z)\} \quad \delta(q_0, a, 0) = \{(q_0, 00)\} \quad \delta(q_0, a, 1) = \{(q_0, 01)\}$$

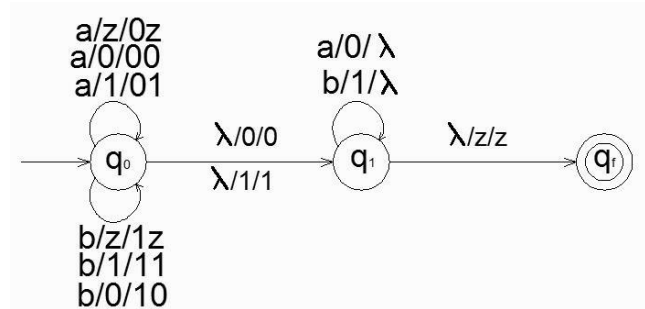
$$\delta(q_0, b, z) = \{(q_0, 1z)\} \quad \delta(q_0, b, 0) = \{(q_0, 10)\} \quad \delta(q_0, b, 1) = \{(q_0, 11)\}$$

$$\delta(q_1, a, 0) = \{(q_1, \lambda)\} \quad \delta(q_1, b, 1) = \{(q_1, \lambda)\} \quad \delta(q_1, \lambda, z) = \{(q_f, z)\}$$

$$\delta(q_0, \lambda, 0) = \{(q_1, 0)\} \quad \delta(q_0, \lambda, 1) = \{(q_1, 1)\}$$

حل:

شکل این ماشین به صورت زیر است:



این ماشین زبان $L = \{ww^R : w \in \{a,b\}^+\}$ را می‌پذیرد. رشته $abba$ نمونه‌ای از رشته‌های این زبان است. این رشته از دو بخش W و w^R تشکیل شده است. برای پی بردن به نحوه عملکرد این ماشین تصور کنید که در وسط رشته، λ قرار دارد. (مثلاً $ab\lambda ba$).

توسط قوانین $\delta(q_0, \lambda, 0) = \{(q_1, 0)\}$ ، $\delta(q_0, \lambda, 1) = \{(q_1, 1)\}$ می‌توان بدون خواندن ورودی و تغییر در پشته، فقط از حالت q_0 به q_1 رفت. از این ویژگی برای تشخیص وسط رشته استفاده شده است.

ماشین مورد نظر با دیدن a در ورودی، عدد 0 و با دیدن b عدد 1 را در پشته **push** می‌کند. نکته این ماشین در این است که نباید تا انتهای رشته ورودی این عمل را انجام داد و باید در وسط رشته، از حالت q_0 به q_1 رفت. در حالت q_1 با دیدن a ، 0 بالای پشته **pop** می‌شود و با دیدن b ، 1 بالای پشته **pop** می‌شود. بعد از پایان ورودی، در صورتی که پشته خالی باشد، از q_1 به q_f رفته و رشته پذیرفته می‌شود. ■

پیکربندی لحظه‌ای

پیکربندی لحظه‌ای یک اتومات پشته‌ای، سه تایی (q, w, u) است که در آن، q حالت جاری واحد کنترل، w بخش خوانده نشده رشته ورودی و u محتویات پشته است (چپ‌ترین سمبل بیانگر بالای پشته است). انتقال از یک پیکربندی به پیکربندی دیگر را بوسیله \mapsto نمایش می‌دهند.

به طور نمونه دنباله تغییرات پیکربندی برای پذیرش رشته $abba$ در ماشین پذیرنده زبان $L = \{ww^R : w \in \{a,b\}^+\}$ به صورت زیر است:

$$(q_0, abba, z) \mapsto (q_0, bba, 0z) \mapsto (q_0, ba, 10z) \mapsto (q_1, ba, 10z) \mapsto (q_1, a, 0z) \\ \mapsto (q_1, \lambda, z) \mapsto (q_f, z).$$

توجه شود که در حرکت سوم از ویژگی نامعین بودن اتومات برای تشخیص وسط رشته استفاده شد و بدون خواندن ورودی و تغییر در پشته، فقط از حالت q_1 به حالت q_f رفت.

اتوماتای پشته‌ای معین

اتومات پشته‌ای $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ معین گفته می‌شود اگر هم در تعریف npda صدق کند و همچنین دارای محدودیت‌هایی به این شرح باشد که به ازای هر $a \in \Sigma \cup \{\lambda\}$, $b \in \Gamma$:

۱- $\delta(q, a, b)$ حداکثر یک عضو داشته باشد.

۲- اگر $\delta(q, \lambda, b)$ تهی نباشد، آنگاه $\delta(q, c, b)$ باید به ازای هر $c \in \Sigma$ تهی باشد.

اولین شرط فوق صرفاً مستلزم آن است که به ازای هر سمبل ورودی مفروض و هر عنصر بالای پشته، حداکثر یک حرکت قابل انجام باشد. بر اساس شرط دوم، چنانچه در یکی از پیکربندی‌های مفروض به یک انتقال λ برخورد کنیم، آنگاه هیچ حرکتی برای جلو بردن و مصرف ورودی امکان پذیر نمی‌باشد. به طور مثال اگر قانون $\delta(q_0, \lambda, z) = \{(q_f, z)\}$ داشته باشیم، آنگاه نباید قانونی مانند $\delta(q_0, a, z) = \{(q_0, 0z)\}$ وجود داشته باشد.

هیچ هم ارزی بین اتوماتای پشته‌ای معین و نامعین وجود ندارد. (بالعکس اتوماتای متناهی)

هر PDA قطعی، می‌تواند transition λ - داشته باشد. (بدون دریافت ورودی، بتواند تغییر حالت دهد).

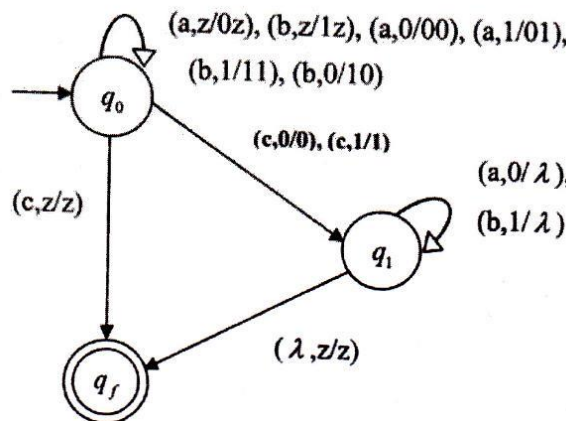
زبان مستقل از متن معین

زبان L یک زبان مستقل از متن معین (قطعی) نامیده می‌شود اگر و تنها اگر یک dpda به نام M وجود داشته باشد که $L = L(M)$

مثال

آیا زبان مستقل از متن $L = \{wcw^R : w \in \{a, b\}^*\}$ ، معین است؟

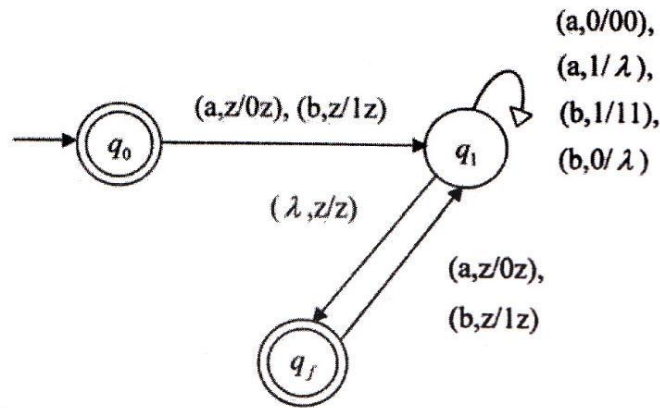
حل: بله- چون می‌توان یک dpda برای آن ساخت.



مثال

آیا زبان مستقل از متن $L = \{w \in \{a,b\}^* : n_a(w) = n_b(w)\}$ ، معین است؟

حل: بله- چون می‌توان برای آن یک dpda طراحی کرد:



مثال

آیا زبان مستقل از متن $L = \{a^n b^n : n \geq 0\} \cup \{a^n b^{2n} : n \geq 0\}$ ، معین است؟

حل: خیر- چون باید یک یا دو b را با هر یک از a ها تطابق دهد. اطلاعات موجود در ابتدای رشته چنان نیست که بتوان به کمک آن انتخاب خود را قطعی کرد. ■

مثال

آیا زبان مستقل از متن $L = \{ww^R : w \in \{a,b\}^*\}$ ، معین است؟

حل- خیر- چون وسط رشته باید با آزمایش و خطا حدس زده شود. ■

مثال

آیا زبان مستقل از متن $L = \{a^n b^n : n \geq 1\} \cup \{a\}$ ، معین است؟

حل: بله- چون می توان برای آن یک dpda طراحی کرد:

$$\delta(q_0, a, z) = \{(q_3, 1z)\}$$

$$\delta(q_3, a, 1) = \{(q_1, 11)\}$$

$$\delta(q_1, a, 1) = \{(q_1, 11)\}$$

$$\delta(q_1, b, 1) = \{(q_1, \lambda)\}$$

$$\delta(q_1, \lambda, z) = \{(q_2, z)\}$$

که q_2 و q_3 حالت‌های پایانی هستند.

در نگاه اول ممکن است فکر کنید که این زبان نامعین است. اما چون توانستیم یک dpda برای آن بسازیم، این زبان معین است. اگر اولین سمبل ورودی یک a باشد، ماشین به یکی از حالات پایانی رفته و در صورت وجود سمبل‌های بعدی، از این حالت خارج شده و $a^n b^n$ را می پذیرد. ■

چند نکته:

۱- خانواده زبان‌های مستقل از متن معین، تحت اجتماع، اشتراک و معکوس بسته نیستند.

۲- زبان‌های مستقل از متن معین، تحت عمل مکمل‌گیری بسته می باشند.

۳- خانواده زبان‌های مستقل از متن معین، تحت تفاضل منظم بسته است.

۴- اجتماع یک زبان مستقل از متن معین با یک زبان منظم، مستقل از متن معین می باشد.

چند نکته:

۱- تمامی زبان‌های منظم، مستقل از متن معین می باشند. چون هر زبان منظمی را می توان با یک dfa پذیرفت و چنین dfa یک dpda با پشته استفاده نشده است.

۲- زبان‌های مستقل از متن معین، هیچگاه ذاتا مبهم نیستند.

۳- به ازای هر npda به نام M ، یک npda به نام \widehat{M} حداکثر ۳ حالتی وجود دارد، بطوریکه:

$$L(M) = L(\widehat{M})$$

البته می توان تعداد حالت‌ها را به ۲ حالت کاهش داد. (با قرار دادن یک سمبل در پشته)

ساخت اتوماتای پشته‌ای با استفاده از گرامر در فرم گریباخ

می‌توان با داشتن گرامر در فرم گریباخ برای یک زبان مستقل از متن، ماشین پشته‌ای برای آن طراحی کرد.

مثال

pda ای بسازید که زبان تولید شده بوسیله گرامر $aSbb \mid a$ را پذیرش کند.

حل: ابتدا گرامر را به فرم گریباخ در می‌آوریم:

$$S \rightarrow aSA \mid a$$

$$A \rightarrow bB$$

$$B \rightarrow b$$

مراحل تبدیل گرامر به ماشین:

الف- قرار دادن متغیر شروع روی Z:

$$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$$

ب- شبیه‌سازی قوانین:

$$S \rightarrow aSA \mid a : \delta(q_1, a, S) = \{(q_1, SA), (q_1, \lambda)\}$$

$$A \rightarrow bB : \delta(q_1, b, A) = \{(q_1, B)\}$$

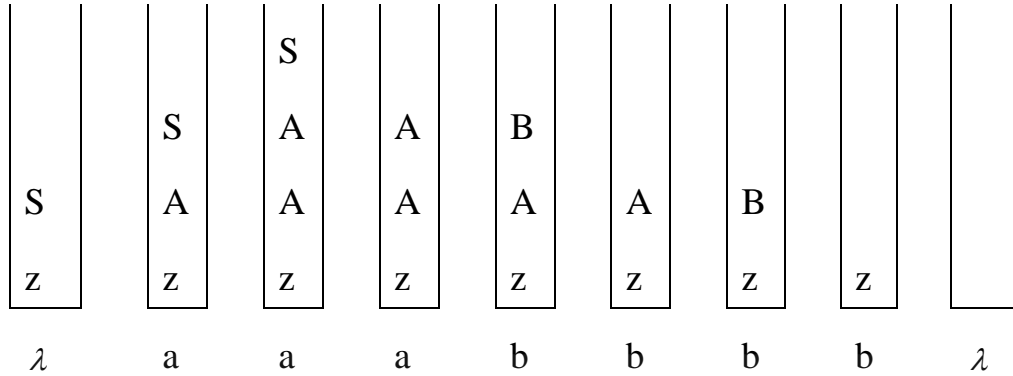
$$B \rightarrow b : \delta(q_1, b, B) = \{(q_1, \lambda)\}$$

ج- با وقوع Z در ته پشته، کار اشتقاق تمام شده و ماشین بوسیله دستور زیر در حالت پایانی قرار می‌گیرد:

$$\delta(q_1, \lambda, z) = \{(q_f, \lambda)\}$$

این ماشین پشته‌ای زبان $L = \{a^{n+1}b^{2n} : n \geq 0\}$ را می‌پذیرد.

نحوه پذیرش رشته a^3b^4 توسط ماشین:



مثال

یک npda برای زبان $L = \{a^n b^{n+1} : n \geq 0\}$ پیدا کنید.

حل: گرامر آن به فرم $S \rightarrow aSb \mid b$ است. فرم گریباخ این گرامر برابر است با:

$$S \rightarrow aSb \mid b$$

$$B \rightarrow b$$

پس یک ماشین پشته ای با سه حالت q_0 و q_1 و q_f می سازیم. در صورتی می توان حالت q_1 را حذف کرد که از یک سمبل ویژه پشته مانند z_1 برای نشانه گذاری آن استفاده کرد:

$$\delta(q_0, \lambda, z) = \{(q_0, Sz_1)\}$$

$$\delta(q_0, a, S) = \{(q_0, SB)\}$$

$$\delta(q_0, b, S) = \{(q_0, \lambda)\}$$

$$\delta(q_0, b, B) = \{(q_0, \lambda)\}$$

$$\delta(q_0, \lambda, z_1) = \{(q_f, \lambda)\}.$$



مثال

یک npda ای طراحی کنید که زبان تولید شده توسط گرامر $S \rightarrow aSbb \mid aab$ را بپذیرد.

حل: ابتدا گرامر $S \rightarrow aSbb \mid aab$ را به فرم گریباخ در می‌آوریم:

$$S \rightarrow aSA \mid aD$$

$$A \rightarrow bB$$

$$B \rightarrow b$$

$$D \rightarrow aB$$

سپس npda آن را طراحی می‌کنیم:

$$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$$

$$\delta(q_1, a, S) = \{(q_1, SA)\}$$

$$\delta(q_1, a, S) = \{(q_1, D)\}$$

$$\delta(q_1, a, D) = \{(q_1, B)\}$$

$$\delta(q_1, b, A) = \{(q_1, B)\}$$

$$\delta(q_1, b, B) = \{(q_1, \lambda)\}$$

$$\delta(q_1, \lambda, z) = \{(q_f, z)\}$$



تشخیص مستقل از متن بودن یک زبان

یکی از روش‌های تشخیص مستقل از متن بودن یک زبان، امکان طراحی یک ماشین پشته‌ای برای آن زبان است. قبلاً دیدیم که روش دیگر این است که بتوان برای آن زبان، یک گرامر مستقل از متن نوشت.

مثال

آیا $L = \{a^n b^m c^n d^m : m, n \geq 0\}$ مستقل از متن است؟

حل: خیر - ترتیب قرار گرفتن نمادها طوری است که نمی‌توان توسط یک پشته، تساوی تعداد a ها با تعداد c ها و همچنین تعداد b ها با تعداد d ها را کنترل کرد. ■

مثال

آیا $L = \{a^n b^n c^m d^m : m \geq n \geq 0\}$ مستقل از متن است؟

حل: خیر- می توان تساوی تعداد نمادهای a با b و تعداد c با d را چک کرد. اما با همان پشته دیگر نمی توان شرط $m \geq n$ را نیز چک کرد. ■

مثال

آیا زبان $\{a^{3n}b^{2n}a^{5n} : n \in N\}$ مستقل از متن است؟

حل: خیر- بعد از تطبیق تعداد a ها با b ها ، مقدار n فراموش شده و نمی توان تعداد a در a^{5n} را چک کرد. ■

گرامر LL

گرامرها در تمام جنبه‌های نحو زبان‌های برنامه سازی موفق نمی باشند. با تعمیم این گرامرها بطوری که قابلیت تجزیه را از دست بدهند، به گرامر LL می‌رسیم. L اول به معنای پویا از چپ به راست و L دوم به معنای ساخت اشتقاق های چپ ترین است. در این گرامرها، با نگاه به بخش محدودی از ورودی شامل سمبل خوانده شده بعلاوه تعداد متناهی از سمبل های بعد از آن، به راحتی قانون مناسب را می توان تعیین کرد.

مثال

گرامر $S \rightarrow aSb \mid ab$ یک $LL(2)$ است، چون با در اختیار داشتن سمبل جاری خوانده شده و پیش بینی سمبل بعدی، می توان قانون درست را انتخاب کرد. اگر سمبل خوانده شده a و سمبل بعدی b باشد، از قانون $S \rightarrow ab$ و در غیر اینصورت از $S \rightarrow aSb$ باید استفاده کرد. ■

مثال

یک گرامر برای زبان $L = \{a^n b^m c^{n+m} : n \geq 0, m \geq 0\}$ بسازید.

حل:

$$S \rightarrow aSc \mid X \mid \lambda$$

$$X \rightarrow bXc \mid \lambda$$

این گرامر $LL(1)$ است. چون اگر سمبل جاری خوانده شده a باشد، از $S \rightarrow aSc$ ، اگر b باشد از $S \rightarrow X$ و اگر c باشد از $S \rightarrow \lambda$ باید استفاده کرد. ■

مثال

یک گرامر برای زبان $L(a^*ba) \cup L(abb^*)$ بسازید.

حل:

$$\begin{aligned} S &\rightarrow S_1 \mid S_2 \\ S_1 &\rightarrow aS_1 \mid ba \\ S_2 &\rightarrow abbB \\ B &\rightarrow bB \mid \lambda \end{aligned}$$

این گرامر $LL(3)$ است. چون اگر سه سمبل اول، aaa یا aab یا aba باشد، آنگاه رشته در a^*ba و اگر abb باشد، رشته در abb^* وجود دارد. ■

چند نکته:

- ۱- گرامری که ساده باشد، حتما LL نیز می باشد.
- ۲- گرامر LL ، مبهم نمی باشد.
- ۳- زبان گرامری $LL(k)$ ، مستقل از متن معین است.
- ۴- یک گرامر به فرم نرمال گریباخ، در صورتی که LL باشد، الزاما $LL(1)$ خواهد بود.

تمرین فصل ۶

۱- کدام زبان مستقل از متن نیست؟

$$L_1 = \{a^n b^j a^k a^m : n + j \leq k + m\}$$

$$L_2 = \{a^n b^j a^k b^m : n \leq k, j \leq m\}$$

۲- آیا زبان $L = \{a^i b^j c^k d^t a^i b^k c^j d^i : i, j, k, t \geq 0\}$ مستقل از متن است؟

۳- آیا زبان $L = \{a^n b^m c^{m+3} d^p e^{n+p} : n, m, p \geq 0\}$ مستقل از متن است؟

۴- کدام جمله صحیح است؟

الف) اگر یک PDA هیچگاه عمل POP کردن را انجام ندهد آنگاه زبانی که می پذیرد منظم است.

ب) مجموعه زبان‌های پذیرفته شده توسط NPDA از DPDA بیشتر است.

۵- در یک اتومات پوش دان (Push Down) طول پشته حداکثر 6 است. زبانهایی که این اتومات می تواند بپذیرد، در کدام مجموعه زبان قرار می گیرد؟

۶- یک PDA با n حالت برای محاسبه کلیه ورودی‌ها حداکثر از 2^n سلول از حافظه پشته خود استفاده می کند. آیا برای زبان این ماشین یک گرامر منظم وجود دارد؟

۷- آیا هر زبان مستقل از متن، دارای یک PDA متناظر با پشته متناهی است؟

۸- اتوماتون یک PDA، فاقد انتقال بلادرنگ (λ -transition) است. آیا زمان محاسبه هر ورودی، متناهی است؟

۹- آیا یک PDA دلخواه در هر مسیر محاسبه برای ورودی w، فقط از مقدار متناهی از حافظه پشته خود استفاده می کند؟

۱۰- هر زبان مستقل از متن می تواند با یک PDA با حداکثر چند حالت توصیف شود؟

۱۱- آیا زبان $L = \{w \in \{0,1\}^* : n_0(w) = f(n_1(w))\}$ مستقل از متن است؟

$$f(x) = \begin{cases} 3 & x \text{ is odd} \\ x+1 & x \text{ is even} \end{cases}$$

۱۲- آیا زبان $L = \{w \in \{0,1\}^* : |w|_0 = f(|w|_1)\}$ که در آن $f(n) = n \bmod 3$ است، مستقل از متن است؟
($|w|_i$ تعداد حروف i در کلمه w باشد)

۱۳- زبان $L = \{a^n b^n c^n : n \geq 0\}$ مفروض است. آیا زبان $h(L)$ مستقل از متن است؟

$h(L)$:

$$h(a) = a,$$

$$h(b) = bb,$$

$$h(c) = a$$

۱۴- کدام یک از زبان‌های زیر مستقل از متن می‌باشند؟

$$\{a^n b^n c^m d^m : m \geq 0, n \geq 0\}$$

$$\{a^m b^k c^m : m, k \geq 0\}$$

$$\{a^n b^{2m} c^m : n, m \geq 0\}$$

$$\{a^m b^m c^n d^n : n, m \geq 0\}$$

$$\{a^n b^m : n = 3m\}$$

$$\{a^n b^m c^n : m, n \geq 0\}$$

$$\{a^n b^m c^m d^n : m, n \geq 0\}$$

۱۵- برای کدام یک از زبان‌های زیر، می‌توان یک ماشین پشته‌ای طراحی کرد؟ $x, y \in \{a, b\}^*$

$$L_1 = \{xy : x \neq y\}$$

$$L_2 = \{xy : |x| \neq |y|\}$$

$$L_3 = \{xy : |x| = |y|, x \neq y\}$$

$$L_4 = \{xy : x = y^R\}$$

$$L_5 = \{xy : x = y\}$$

۱۶- نشان دهید که تمامی زبان‌های منظم، مستقل از متن معین هستند.

۱۷- نشان دهید که زبان‌های مستقل از متن معین، هیچگاه ذاتا مبهم نیستند.

۱۸- یک زبان مستقل از متن معین ارائه دهید که معکوس آن نامعین باشد.

۱۹- اگر در یک npda، در هر حرکت، طول پشته حداکثر به اندازه یک سمبل افزایش یابد، آنگاه

دستور $\delta(q_i, a, b) = \{(q_j, cde)\}$ را چگونه باید پیاده‌سازی کرد؟

۲۰- برای هر یک از زبان‌های زیر، ماشین پشته‌ای طراحی کنید.

$$L_1 = L(aa^+b)$$

$$L_2 = L(ab^+aba^*)$$

۲۱- برای هر یک از زبان‌های زیر، ماشین پشته‌ای طراحی کنید.

$$L_1 = \{a^n b^m c^{n+m} : n \geq 0, m \geq 0\}$$

$$L_2 = \{a^n b^{n+m} c^m : n \geq 0, m \geq 0\}$$

$$L_3 = \{a^3 b^n c^n : n \geq 0\}$$

$$L_4 = \{a^n b^m : n \geq 0, n \neq m\}$$

۲۲- برای هر یک از زبان‌های زیر، ماشین پشته‌ای طراحی کنید.

$$L_1 = \{w \in \{a, b\}^* : n_a(w) = n_b(w) + 1\}$$

$$L_2 = \{w \in \{a, b, c\}^* : n_a(w) + n_b(w) = n_c(w)\}$$

$$L_3 = \{w \in \{a, b\}^* : wcw^R\}$$

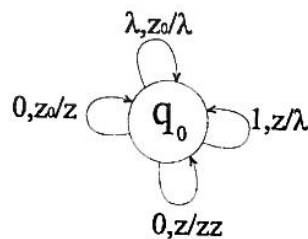
$$L_4 = \{w \in \{a, b\}^* : 2n_a(w) \leq n_b(w) \leq 3n_a(w)\}$$

۲۳- برای زبان زیر، ماشین پشته‌ای طراحی کنید.

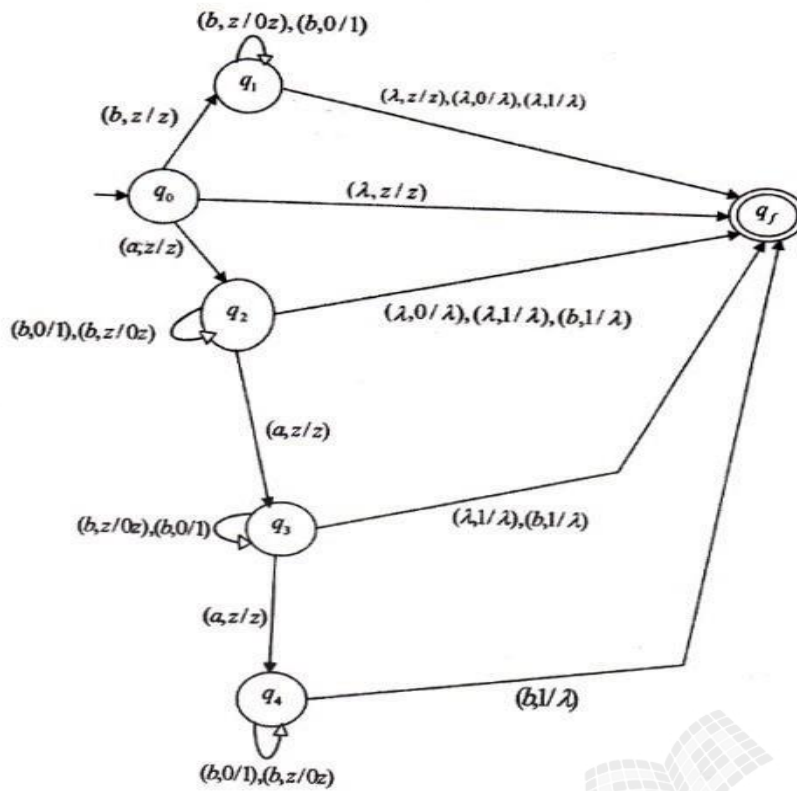
$$L = \{ab(ab)^n b(ab)^n : n \geq 0\}$$

۲۴- فرض کنید $L = \{a^n b^{2n} : n \geq 0\}$ باشد. یک ماشین پشته‌ای برای L^* طراحی کنید.

۲۵- اتومات پوش دان $(Q, \Sigma, \Gamma, \delta, q_0, z_0)$ با $Q = \{q_0\}, \Sigma = \{0, 1\}, \Gamma = \{z_0, z\}$ مطابق شکل زیر مفروض است. زبان این اتومات را مشخص کنید.



۲۶- ماشین پشته‌ای زیر، چه زبانی را می‌پذیرد؟



حل تمرین فصل ۶

۱- زبان L_2 مستقل از متن نمی باشد، چون رشته های آن را نمی توان توسط یک پشته بررسی کرد.

۲- بله- زیرا می توان جملات آن را به کمک یک پشته، بررسی کرد.

۳- زبان داده شده را می توان به صورت زیر نوشت:

$$L = \{a^n b^m c^m c^3 d^p e^p e^n : n, m, p \geq 0\}$$

می توان جملات این زبان را به کمک یک پشته، بررسی کرد. پس مشتقل از متن است.

۴- هر دو گزاره صحیح هستند.

۵- چون طول پشته محدود است، ماشین پشته‌ای مانند ماشین متناهی عمل کرده و فقط زبان های منظم را می پذیرد.

۶- بله- ماشین PDA ای که برای محاسبه ورودی ها حداکثر از تعدادی مشخص از خانه های پشته (مثلاً 2^n)، استفاده کند، معادل DFA است. بنابراین برای زبان این ماشین، یک گرامر منظم وجود دارد.

۷- خیر- هر زبان منظم دارای یک PDA متناظر با پشته متناهی است.

۸- بله- چون اگر انتقال λ نداشته باشد، به ازای هر الفبای رشته، یک انتقال انجام شده و چون طول رشته متناهی است، بنابراین حتما در زمان متناهی پیمایش رشته به پایان می رسد.

۹- خیر- چون ممکن است در حلقه بی افتد و از مقدار نامتناهی از حافظه پشته خود استفاده کند.

۱۰- بله- برای هر زبان مستقل از متن، یک PDA با حداکثر دو حالت وجود دارد.

۱۱- بله- زبان L را می توان به صورت زیر نوشت:

$$L = \{w \in \{0,1\}^* : n_0(w) = 3 \text{ or } n_0(w) = n_1(w) + 1\}$$

تعداد یکها، هر عددی می تواند باشد ولی تعداد صفرها برابر 3 یا یک واحد بیشتر از تعداد یکها است. بنابراین چون می توان رشته های این زبان را به کمک یک پشته پذیرفت، زبان مستقل از متن است.

۱۲- از آنجا که باقیمانده تقسیم هر عددی بر 3، برابر یکی از اعداد 0 تا 2 می باشد، داریم:

$$L = \{w \in \{0,1\}^* : |w|_0 = 0 \text{ or } |w|_0 = 1 \text{ or } |w|_0 = 2\}$$

در این زبان تعداد یکها، هر عددی می‌تواند باشد ولی تعداد صفرها باید یکی از اعداد 0، 1 و یا 2 باشد. بنابراین چون می‌توان رشته‌های این زبان را به کمک یک پشته پذیرفت، زبان مستقل از متن است.

۱۳- خیر- برای مشخص کردن زبان مربوط به همومورفیسم، کافی است در $\{a^n b^n c^n\}$ به جای a همان a ، به جای b مقدار bb و به جای c مقدار a را قرار داد. در این صورت خواهیم داشت:

$$a^n (bb)^n a^n = a^n b^{2n} a^n$$

این زبان مستقل از متن نیست، چون علاوه بر کنترل تعداد نمادهای a ، b در $a^n b^{2n}$ ، باید تعداد a ، b در $b^{2n} a^n$ نیز چک شود که با یک پشته ممکن نیست.

۱۴- همه زبان‌های داده شده مستقل از متن هستند، چون می‌توان برای آنها یک ماشین پشته‌ای رسم کرد.

۱۵- برای همه به غیر از L_5 .

۱۶- هر زبان منظمی را می‌توان با یک dfa پذیرفت. چنین dfa ای یک dpda با پشته استفاده نشده است.

۱۷- به ازای هر زبان مستقل از متن معین، یک dpda وجود دارد. اگر این dpda به گرامر تبدیل شود، گرامر به دست آمده، غیر مبهم خواهد بود.

۱۸- معکوس زبان مستقل از متن معین زیر، نامعین است:

$$L = \{w_1 c w_2, w_1 = w_2^R, w_1, w_2 \in \{a, b, c\}^*\}$$

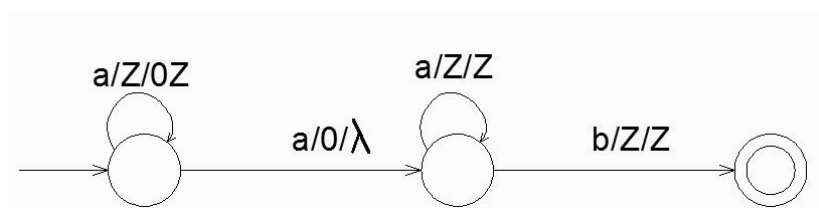
۱۹- از دو دستور زیر استفاده می‌کنیم:

$$\delta(q_i, a, b) = \{(q_k, de)\}$$

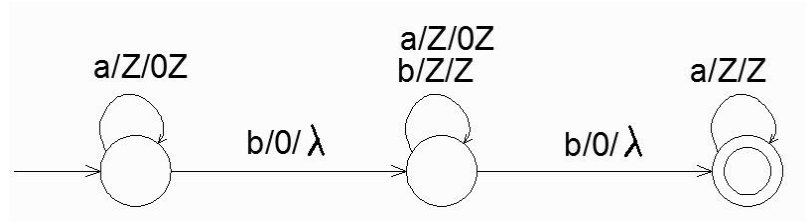
$$\delta(q_k, \lambda, d) = \{(q_j, cd)\}$$

۲۰- هر یک از ماشین‌ها در زیر نشان داده شده است:

ماشین زبان $L_1 = L(aa^+b)$:

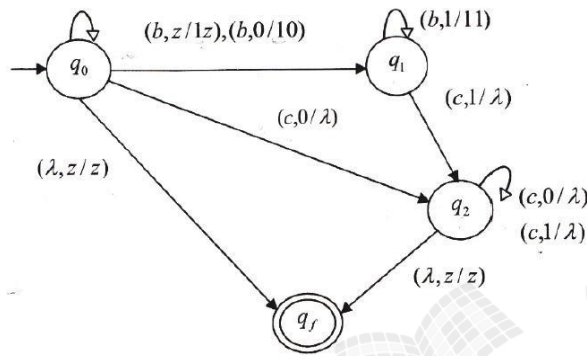


ماشین زبان $L_2 = L(ab^+aba^*)$:

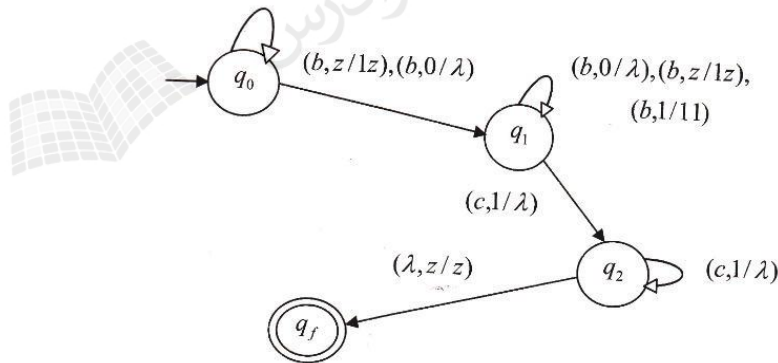


۲۱- هر یک از ماشین‌ها در زیر نشان داده شده است:

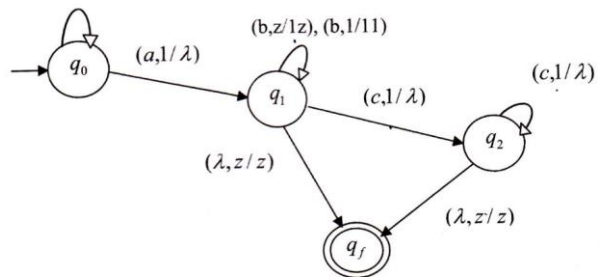
ماشین زبان $L_1 = \{a^n b^m c^{n+m} : n \geq 0, m \geq 0\}$:



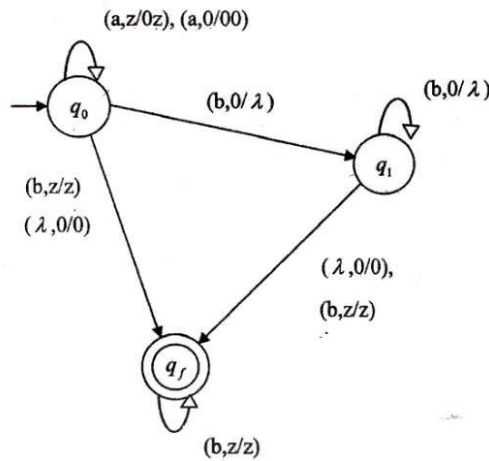
ماشین زبان $L_2 = \{a^n b^{n+m} c^m : n \geq 0, m \geq 0\}$:



ماشین زبان $L_3 = \{a^3 b^n c^n : n \geq 0\}$:

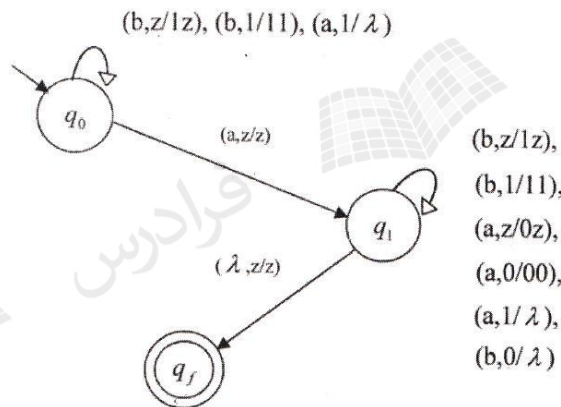


ماشین زبان $L_4 = \{a^n b^m : n \geq 0, n \neq m\}$:

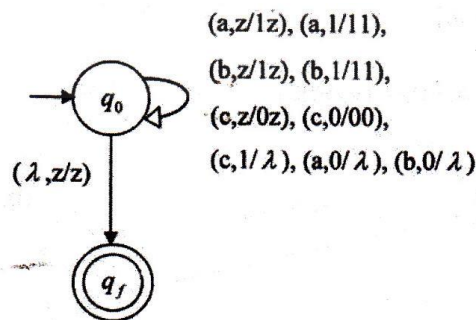


۲۲- هر یک از ماشین ها در زیر نشان داده شده است:

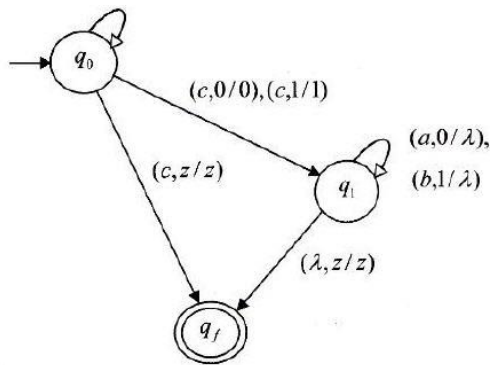
ماشین زبان $L_1 = \{w \in \{a,b\}^* : n_a(w) = n_b(w) + 1\}$:



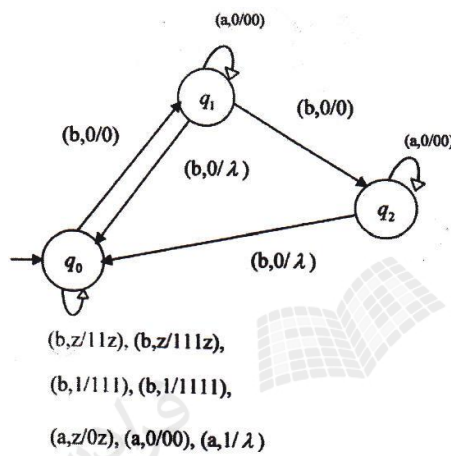
ماشین زبان $L_2 = \{w \in \{a,b,c\}^* : n_a(w) + n_b(w) = n_c(w)\}$:



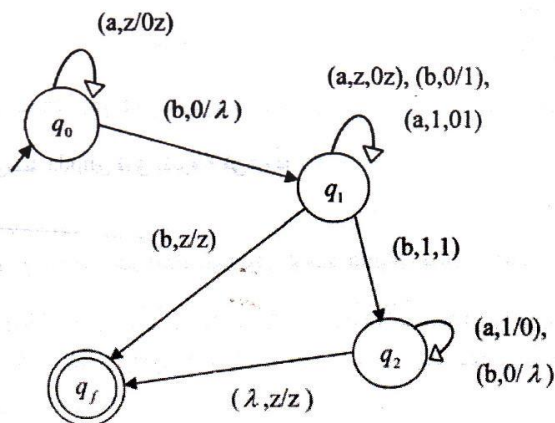
ماشین زبان $L_3 = \{w \in \{a,b\}^* : wcw^R\}$



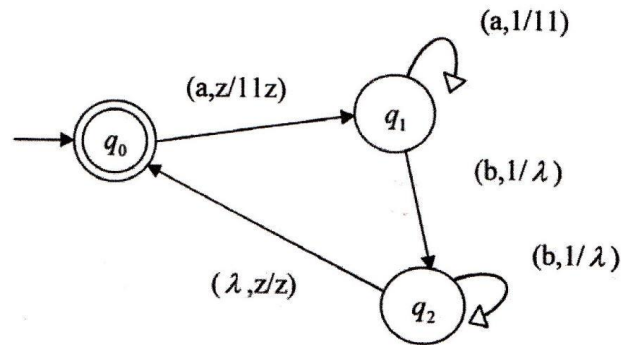
ماشین زبان $L_4 = \{w \in \{a,b\}^* : 2n_a(w) \leq n_b(w) \leq 3n_a(w)\}$



۲۳- ماشین به صورت زیر است:



۲۴- ماشین پشته‌ای به صورت زیر است:



۲۵- زبان ماشین داده شده برابر است با:

$$\{w \in (0+1)^* : n_0(w) = n_1(w) \text{ and } n_0(v) > n_1(v) \text{ for any prefix } v, w = vr, r \in \Sigma^+\}$$

۲۶- ماشین داده شده زبان $L = \{a^n b^m : n \leq m \leq 3\}$ را می‌پذیرد. حالت‌های ممکن:

الف- اگر $n=0$ باشد، آنگاه m برابر 0 تا 3 می‌تواند باشد.

ب- اگر $n=1$ باشد، آنگاه m برابر 1 تا 3 می‌تواند باشد.

پ- اگر $n=2$ باشد، آنگاه m برابر 2 تا 3 می‌تواند باشد.

ت- اگر $n=3$ باشد، آنگاه m برابر 3 می‌تواند باشد.

فصل ۷: ماشین تورینگ

ماشین تورینگ یک کامپیوتر ساده دارای واحد پردازشی با حافظه محدود و نواری با ظرفیت نامحدود است. دستوراتی که این ماشین می‌تواند انجام دهد بسیار محدود است. ماشین تورینگ کاملترین مدل برای یک ماشین محاسبه‌گر می‌باشد و از ماشین‌های متناهی و پشت‌های کاملتر است. ماشین تورینگ پذیرنده زبانهای منظم، مستقل از متن، وابسته به متن و بدون محدودیت است.

ماشین تورینگ استاندارد

ماشین تورینگ M یک هفت‌تایی است که به صورت $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ تعریف می‌شود، که در آن:

Q : مجموعه حالات داخلی

Σ : الفبای ورودی

Γ : الفبای نوار

δ : تابع انتقال

B : سمبل خالی (متعلق به Γ است و در Σ نیست).

q_0 : حالت شروع

F : مجموعه حالت‌های پایانی

نوار ماشین تورینگ به سلول‌هایی تقسیم شده و هر سلول قادر به نگهداری فقط یک سمبل است. به این نوار یک هد خواندن-نوشتن متصل است که می‌تواند به سمت راست یا چپ نوار حرکت کرده و در هر حرکت فقط یک سمبل را بخواند و بنویسد.

تابع انتقال δ در تورینگ، به صورت $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ تعریف می‌شود. یعنی دو ورودی و ۳ خروجی دارد. به عنوان نمونه، وضعیت $\delta(q_0, a) = (q_1, b, R)$ یعنی ماشین در وضعیت q_0 بوده و هد a را می‌بیند. ماشین به وضعیت q_1 رفته و a با b جایگزین می‌شود و هد به سمت راست می‌رود.

مثال

عملکرد ماشین تورینگ زیر چیست؟

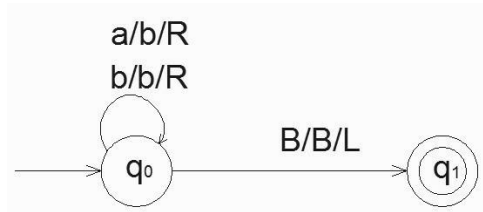
$$Q = \{q_0, q_1\}, \quad \Sigma = \{a, b\}, \quad \Gamma = \{a, b, B\}, \quad F = \{q_1\}$$

$$\delta(q_0, a) = (q_0, b, R)$$

$$\delta(q_0, b) = (q_0, b, R)$$

$$\delta(q_0, B) = (q_1, B, L)$$

حل: نمایش این ماشین به صورت زیر است:



در این ماشین اگر سمبل a ، زیر هد قرار بگیرد، یک b جایگزین آن شده و هد به سمت راست حرکت می‌کند و ماشین در حالت q_0 باقی می‌ماند. اگر b زیر هد قرار بگیرد، محتوای نوار هیچ تغییری نکرده و هد به راست می‌رود. ماشین با رسیدن به اولین سلول خالی (B)، یک سلول به عقب بر می‌گردد و در حالت نهایی q_1 متوقف می‌شود. با فرض اینکه رشته aa بر روی نوار باشد، عملکرد این ماشین را می‌توان به صورت زیر نمایش داد: $q_0aa \mapsto bq_0a \mapsto bbq_0B \mapsto bq_1b$

ماشین تورینگ در نقش پذیرنده زبان

زبانی که ماشین تورینگ M می‌پذیرد به صورت زیر تعریف می‌شود:

$$L(M) = \{w \in \Sigma^+ : q_f \in F, x_1, x_2 \in \Gamma^*, q_0 w \xrightarrow{*} x_1 q_f x_2\}$$

بر اساس این تعریف ورودی w روی نوار نوشته شده و در هر یک از طرفین آن از سمبل فضای خالی استفاده می‌شود.اگر w عضو $L(M)$ نباشد، یکی از دو حالت زیر اتفاق می‌افتد:

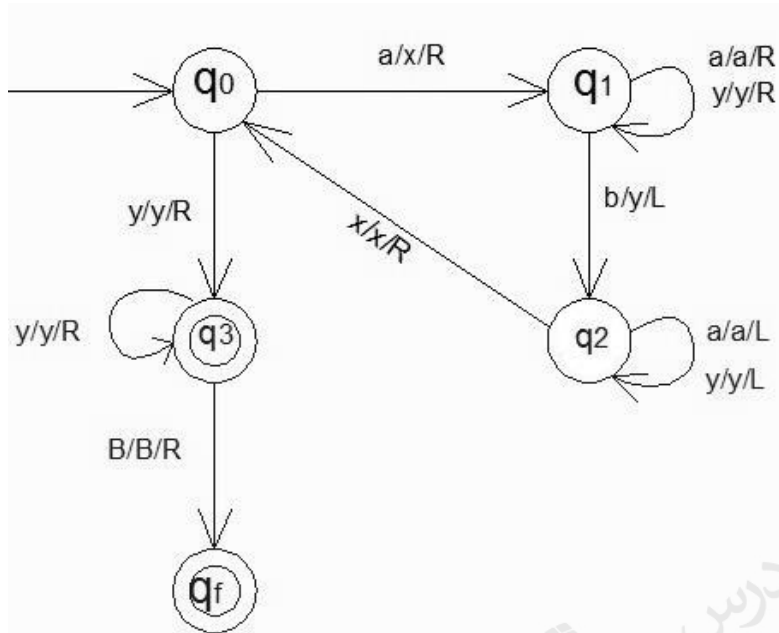
۱- ماشین در یک حالت غیر پایانی متوقف می‌شود.

۲- ماشین به یک حلقه بی‌نهایت وارد شده و هرگز متوقف نمی‌شود.

بنابراین، هر رشته‌ای که باعث توقف M نشود، عضو $L(M)$ نمی‌باشد.

مثال

ماشین تورینگ زیر چه زبانی را می‌پذیرد؟



حل:

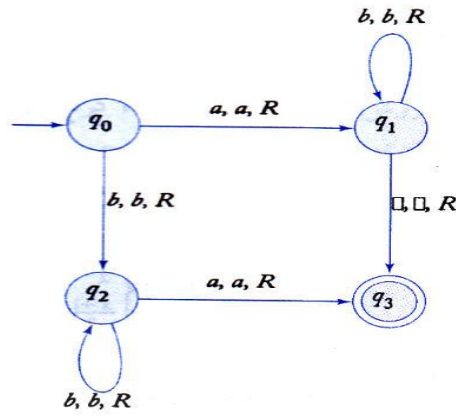
این ماشین زبان $\{a^n b^n : n \geq 1\}$ را می‌پذیرد. در این ماشین با خواندن یک a به جای آن X قرار داده می‌شود و a های بعدی را رد کرده تا به یک b برسد، سپس آن را با y جایگزین کرده و دوباره به سمت چپ‌ترین a رفته و آنرا با X جایگزین می‌کند و مجدداً چپ‌ترین b را با y جایگزین می‌کند و این حرکت پاندولی را ادامه می‌دهد و تک تک a ها را با b نظیر آن تطبیق می‌دهد. اگر a یا b ای باقی نمانده بود، آنگاه رشته حتماً عضو زبان خواهد بود.

بعنوان مثال، ورودی $aabb$ پیکربندی های متوالی زیر را ایجاد می‌کند:

$$\begin{aligned}
 q_0 aabb &\mapsto xq_1 abb \mapsto xaq_1 bb \mapsto xq_2 ayb \mapsto q_2 xayb \mapsto xq_0 ayb \mapsto xxq_1 yb \\
 &\mapsto xxyq_1 b \mapsto xxq_2 yy \mapsto xq_2 xyy \mapsto xxq_0 yy \mapsto xxyq_3 y \mapsto xxyyq_3 B \mapsto xxyyBq_f B.
 \end{aligned}$$

مثال

یک ماشین تورینگ با گراف انتقال زیر، چه زبانی را می‌پذیرد؟ (فاصله خالی با نماد مربع نشان داده شده است.)



حل: ماشین داده شده زبان زیر را می‌پذیرد:

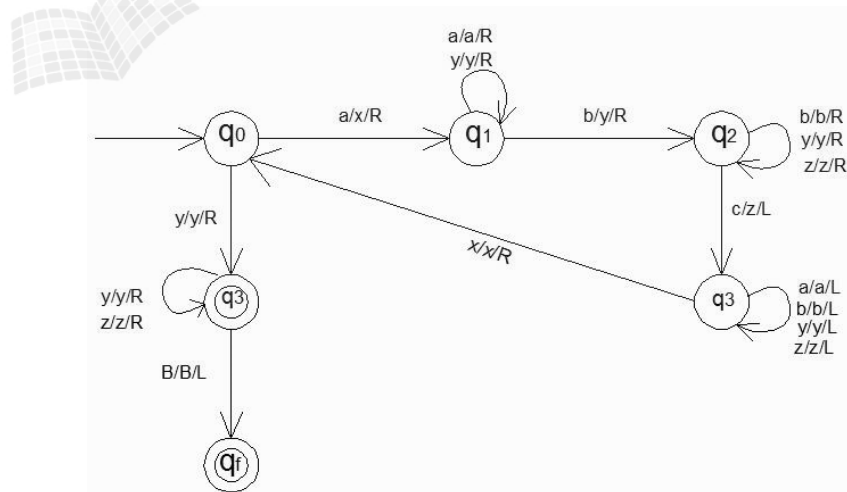
$$L = \{ab^n : n \geq 0\} \cup \{b^na : n \geq 1\}$$



مثال

ماشین تورینگ برای پذیرش زبان $L = \{a^n b^n c^n : n \geq 1\}$ طراحی کنید.

حل: حرف a را با x، b را با y و c را با z علامت‌گذاری می‌کنیم.



قبلا دیدیم که زبان $L = \{a^n b^n c^n : n \geq 1\}$ مستقل از متن نیست و توسط ماشین پشته‌ای پذیرش نمی‌شود. اما این زبان

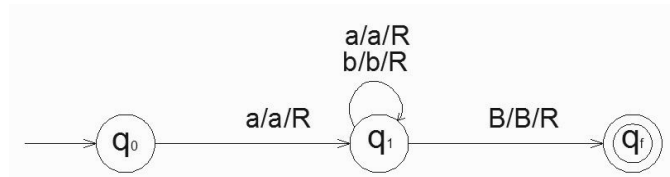
توسط تورینگ پذیرش شد. پس ماشین تورینگ از ماشین پشته‌ای قویتر می‌باشد. ■

مثال

ماشین تورینگ زیر چه زبانی را می‌پذیرد؟

$$\begin{aligned}\delta(q_0, a) &= (q_1, a, R) \\ \delta(q_1, a) &= (q_1, a, R) \\ \delta(q_1, b) &= (q_1, b, R) \\ \delta(q_1, B) &= (q_f, B, R)\end{aligned}$$

حل: شکل ماشین به صورت زیر است:



این ماشین زبان $L(a(a+b)^*)$ روی الفبای $\{a,b\}$ را می‌پذیرد. پس ماشین تورینگ می‌تواند زبان منظم را بپذیرد. ■

مثال

ماشین تورینگ زیر چه زبانی را می‌پذیرد؟

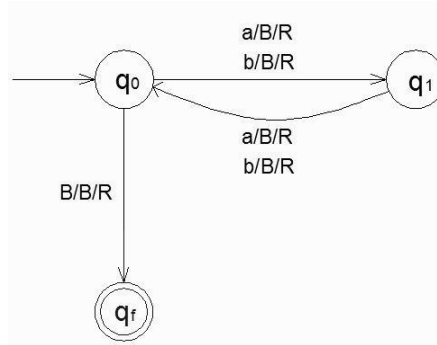
$$\begin{aligned}\delta(q_0, a) &= (q_1, a, R) \\ \delta(q_1, b) &= (q_2, b, R) \\ \delta(q_2, a) &= (q_2, a, R) \\ \delta(q_2, b) &= (q_f, b, R)\end{aligned}$$

حل: این ماشین زبان $L = L(aba^*b)$ روی الفبای $\{a,b\}$ را می‌پذیرد. ■

مثال

یک ماشین تورینگ طراحی کنید که زبان $L = \{w \mid |w| \text{ زوج است}\}$ را روی الفبای $\{a, b\}$ پذیرش کند.

حل:



مثال

یک ماشین تورینگ طراحی کنید که زبان $L = \{a^i b^j c^k : k = i \times j, i, j, k \geq 1\}$ را تصمیم‌گیری کند.

حل: کارهای انجام شده روی رشته ورودی توسط ماشین:

۱- ابتدا رشته ورودی را از چپ به راست پویش کرده تا مطمئن شود که رشته ورودی عضو $a^* b^* c^*$ است و اگر اینطور نبود به حالت عدم پذیرش می‌رود.

۲- هد به انتهای سمت چپ برگردانده می‌شود.

۳- یکی از a ها را علامت زده و آنقدر به راست می‌رود تا به یک نماد b برسد. سپس b ها را با c ها تطبیق می‌دهد، یعنی با علامت زدن یک b ، یک c را نیز علامت می‌زند. این کار را تا تمام شدن b ها ادامه می‌دهد.

۴- تمام b های علامت گذاری شده دوباره به نماد b برگردانده می‌شوند. تا زمانی که a باقی مانده باشد، مرحله سوم تکرار می‌شود. اگر تمام a ها علامت گذاری شده باشند، بررسی می‌شود که تمام c ها نیز علامت گذاری شده باشند. در این صورت رشته پذیرفته شده و در غیر این صورت رشته رد می‌شود.

این زبان را نمی‌توان با ماشین پشته‌ای پذیرفت. ■

برای همه ماشین‌های تورینگ، یک ماشین دیگر با تنها یک حالت پایانی وجود دارد که همان زبان را می‌پذیرد. در واقع اگر چند حالت پایانی داشته باشیم، یک حالت پایانی جدید معرفی می‌کنیم و برای همه $a \in \Gamma$ و $q \in F$ انتقالات $\delta(q, a) = (q_f, a, R)$ را انجام می‌دهیم.

ماشین تورینگ به عنوان مترجم

ماشین تورینگ علاوه بر دارا بودن خاصیت پذیرش زبان‌ها، یک مدل ساده انتزاعی برای کامپیوترهای رقمی می‌باشد. در واقع تمامی توابع ریاضی معمولی توسط ماشین تورینگ، محاسبه پذیر بوده و میزان پیچیدگی آنها، تاثیری بر این امر نخواهد داشت.

تابع f محاسبه پذیر توسط تورینگ گفته می‌شود، اگر ماشین تورینگ مفروض M وجود داشته باشد که برای همه w های موجود در دامنه تابع، داشته باشیم: $q_0 w \xrightarrow{*} q_f f(w)$

مثال

ماشین تورینگی برای محاسبه جمع دو عدد صحیح مثبت X و Y طراحی کنید.

قرارداد ۱: هر عدد صحیح مثبت X بوسیله $w(x) \in \{1\}^+$ نمایش داده می‌شود، بطوریکه $|w(x)| = x$. مثلاً عدد ۳ به صورت ۱۱۱ و یا عدد ۲ به صورت ۱۱ نمایش داده می‌شود.

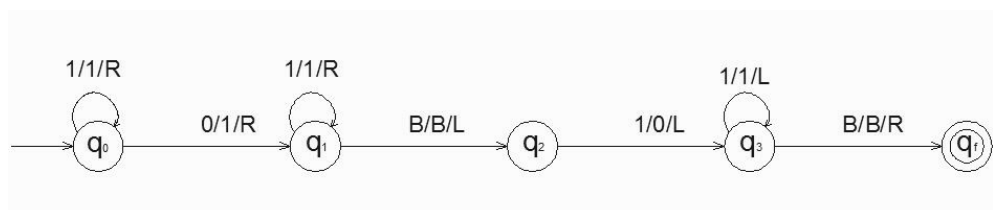
قرارداد ۲: اعداد بوسیله یک ۰ از هم جدا شده و هد روی چپ ترین علامت عدد X می‌باشد.

قرارداد ۳: بعد از محاسبه، $w(x+y)$ روی نوار قرار گرفته و فقط یک ۰ بعد از آن مشاهده می‌شود و هد در انتهای سمت چپ نتیجه قرار خواهد داشت.

بنابراین در واقع می‌خواهیم ماشین تورینگی را برای انجام محاسبه زیر طراحی کنیم:

$$q_0 w(x) 0 w(y) \xrightarrow{*} q_f w(x+y) 0$$

حل: تنها کاری که باید انجام شود، انتقال صفر بین دو عدد، به انتهای سمت راست عدد Y می‌باشد.



بررسی جمع ۱۱۱ با ۱۱:

$$q_0 111011 \mapsto 1q_0 11011 \mapsto 11q_0 1011 \mapsto 111q_0 011 \mapsto 1111q_1 11 \mapsto 11111q_1 1$$

$$\mapsto 111111q_1 B \mapsto 111111q_2 1 \mapsto 11111q_3 10 \mapsto \dots \mapsto q_3 B111110 \mapsto Bq_f 111110. \blacksquare$$

تذکر:

کار با دنباله سمبل‌های یکتایی ۱، هر چند در محاسبات مشکل به نظر می‌رسد، ولی در برنامه ریزی ماشین‌های تورینگ بسیار سودمند است. برنامه‌های حاصل از این روش در مقایسه با روشهای دیگر کوتاهتر و ساده‌تر هستند.

مثال

ماشین تورینگ طراحی کنید که رشته‌هایی از 1 را کپی کند. مثلاً با دادن 11 خروجی 1111 حاصل شود.

(یعنی برای هر $w \in \{1\}^+$ مقدار $q_0 w | - q_f w w^*$ را محاسبه کند).

حل:

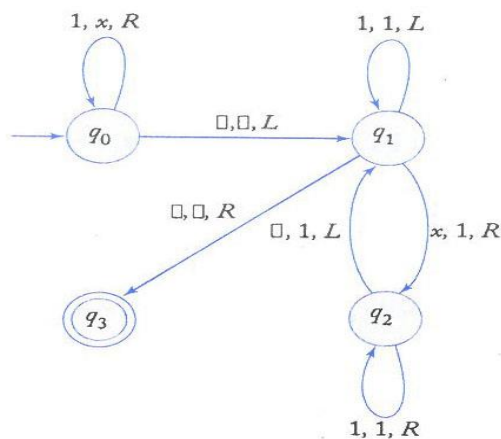
مراحل کار:

۱- هر یک از 1ها را با یک X جایگزین می‌کنیم.

۲- راست‌ترین X را پیدا کرده و آنرا با 1 جایگزین می‌کنیم.

۳- به اولین سمبل خالی سمت راست فعلی رفته و در آنجا یک 1 ایجاد می‌کنیم.

۴- مراحل ۲ و ۳ را آنقدر تکرار می‌کنیم تا هیچ X دیگری وجود نداشته باشد.

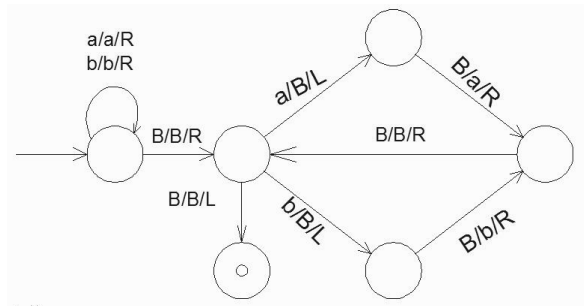


فاصله خالی با نماد مربع نشان داده شده است. ■

مثال

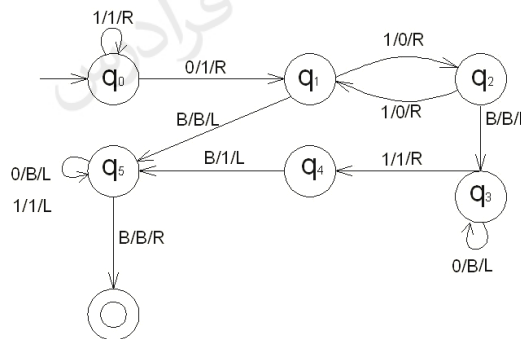
در الفبای $\{a,b\}$ ، ماشین تورینگ برای اتصال دو رشته به یکدیگر طراحی کنید. $(q_0 w_1 B w_2^* | - q_f w_1 w_2)$

حل: بین دو رشته یک بلانک قرار دارد. ابتدا تمام حروف رشته اول را خوانده تا به B برسیم، سپس بعد از خواندن B ، اگر در ابتدای رشته دوم حرف a بود آن را به B تبدیل کرده و به سمت چپ برگشته و B را به a تغییر می‌دهیم و اگر b بود آن را B کرده و به سمت چپ برگشته و B را به b تغییر می‌دهیم. سپس به سمت راست حرکت کرده و این عملیات را تا رسیدن به B در q_2 ادامه می‌دهیم.



مثال

خروجی ماشین تورینگ زیر با ورودی $\{w = x0y : x = 1^*, y = 1^*\}$ با فرض $x = 11, y = 1111$ را مشخص کنید.



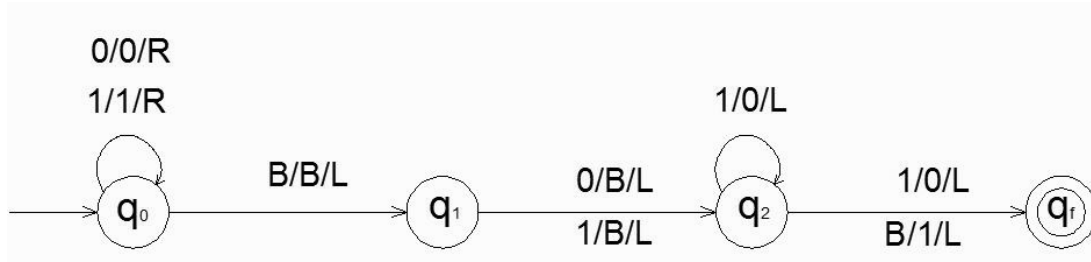
حل:

ورودی $w = 1101111$ ، یعنی x برابر 2 و y برابر 4 است که طبق تعریف با یک صفر از هم جدا شده‌اند. هد از روی همه یکها عبور کرده تا به صفر برسد (چرخه در q_0). سپس ماشین به حالت q_1 رفته و صفر بین x و y به یک تبدیل می‌شود. با حرکت بین دو حالت q_1 و q_2 ، یکهای y به صفر تبدیل شده و چون تعداد یکهای y زوج است، در پایان به q_1 می‌رسد. هد به علامت فاصله خالی پایان رشته برسد و به چپ حرکت کرده و ماشین به حالت q_5 می‌رود. $(w = 1110000)$ در این حالت، تمام صفرها به سمت ابتدای رشته به فاصله خالی تبدیل شده تا به یک برسد. $(w = 111)$. از یکها نیز به سمت ابتدای رشته عبور کرده تا به فاصله خالی اول رشته برسد. در این حالت ماشین به حالت پایانی می‌رود. رشته در این حالت برابر 111 یعنی 3 است. این ماشین تابع $f(x, y) = x + 1 + (y \bmod 2)$ را محاسبه می‌کند:

$$f(2,4) = 2 + 1 + (4 \bmod 2) = 3 + 0 = 3$$

مثال

خروجی ماشین تورینگ زیر با ورودی 1101 و 111 را مشخص کنید.



اصلاح: یال از q_2 به q_f با برچسب $1/0/L$ باید $0/1/L$ شود.

حل:

الف- ورودی 1101 (عدد دهدهی ۱۳): در حالت q_0 از تمام یکها و صفرهای اول رشته به سمت راست عبور می‌کنیم. با رسیدن به blank آخر رشته، یک واحد به عقب بر می‌گردیم. هم‌اکنون در q_1 هستیم و هد روی 1 آخر رشته قرار دارد. در این حالت به جای 1 انتهای رشته، blank قرار داده و به چپ می‌رویم. هم‌اکنون در q_2 هستیم و رشته روی نوار 110 است. در این حالت 0 آخر رشته به 1 تبدیل شده و به حالت پایانی می‌رسیم. پس خروجی 111 است.

ب- ورودی 111 (عدد دهدهی ۷): در حالت q_0 از تمام یکها و صفرهای اول رشته به سمت راست عبور می‌کنیم. با رسیدن به blank آخر رشته، یک واحد به عقب بر می‌گردیم. هم‌اکنون در q_1 هستیم. در این حالت به جای 1 انتهای رشته، blank قرار داده و به چپ می‌رویم. هم‌اکنون در q_2 هستیم و رشته برابر 11 است. در این حالت تمامی 1 های رشته از آخر به اول به 0 تبدیل می‌شود. سپس با رسیدن به blank اول رشته، این blank به 1 تبدیل شده و به حالت پایانی می‌رویم. پس خروجی 100 است.

توضیح در رابطه با عملکرد ماشین:

این ماشین تابع $F(x) = x \text{ div } 2 + 1$ را محاسبه می‌کند. x یک عدد در مبنای دو است. این تابع، ابتدا مقدار x را نصف کرده و سپس یک واحد به آن اضافه می‌کند. عمل نصف کردن با حذف بیت کم ارزش (شیفت به راست) انجام می‌شود. (انتقال از q_1 به q_2).

عمل یک واحد اضافه کردن به صورت زیر است:

الف: اگر رشته با صفر تمام شود، فقط آن صفر به یک تبدیل می‌شود.

ب: اگر رشته با یک تمام شود، از آخر به اول رشته حرکت کرده و تمام یکها را صفر کرده و اولین صفر را یک می‌کنیم.

در حالتی که بعد از حذف بیت کم ارزش، بقیه بیتها یک باشند، همه یکها به صفر تبدیل شده و به blank اول رشته می‌رسیم، آن blank به یک تبدیل می‌شود. ■

مدل‌های دیگر ماشین تورینگ

در این بخش نشان می‌دهیم که ماشین‌های تورینگ استاندارد، معادل با مدل‌های پیچیده‌تر هستند. طبق تز تورینگ، پیچیده کردن ماشین‌های تورینگ استاندارد از طریق تجهیز آنها به ابزار ذخیره سازی پیچیده تر، تاثیری بر قدرت ماشین ندارد. چون هر نوع محاسبه‌ای که با این ماشین‌های جدید قابل انجام باشد، مدلی از محاسبه مکانیکی محسوب شده و به همین علت بوسیله یک مدل استاندارد هم قابل انجام است.

تز تورینگ: این فرضیه می‌گوید که هر نوع محاسبه‌ای که بطور مکانیکی قابل انجام باشد، با ماشین تورینگ هم قابل انجام است.

ویژگی‌های ماشین تورینگ استاندارد:

- ۱- نامحدود بودن نوار ماشین از دو طرف (ممکن بودن حرکت به راست یا چپ به هر تعداد)
- ۲- معین بودن (به ازای هر پیکربندی فقط یک حرکت تعریف می‌شود).
- ۳- عدم وجود هیچ فایله ورودی خاصی و همچنین عدم وجود هیچ وسیله خروجی خاصی.

مدل‌های دیگر ماشین‌های تورینگ عبارتند از:

- ۱- اعمال تغییرات جزئی در تعریف ماشین تورینگ (سکون دار- با نوار نیمه متناهی - آف لاین)
- ۲- ماشین‌های تورینگ با حافظه پیچیده تر (چند نواره و چند بعدی)
- ۳- ماشین‌های تورینگ نامعین
- ۴- ماشین تورینگ عمومی
- ۵- اتوماتای کراندار خطی

ماشین‌های تورینگ سکون دار

هد در این نوع ماشین‌ها می‌تواند پس از بازنویسی محتوای سلول، در جای خود باقی بماند و به جلو یا عقب حرکت نکند. گنجاندن این انتخاب جدید برای حرکت هد، قدرت ماشین را افزایش نمی‌دهد. دسته ماشین‌های تورینگ سکون دار، هم ارز با دسته ماشین‌های تورینگ استاندارد می‌باشند.

ماشین‌های تورینگ با نوار نیمه نامتناهی

نوار در این ماشین فقط از یک طرف نامحدود است. وقتی هد در انتها قرار می‌گیرد، حرکت به چپ مجاز نیست. این محدودیت هیچ تاثیری بر قدرت ماشین نمی‌گذارد.

ماشین‌های تورینگ آف لاین

ماشین تورینگ **offline**، علاوه بر نوار شامل یک فایل ورودی فقط خواندنی نیز می‌باشد. در این نوع ماشین‌ها، تمامی حرکت‌ها توسط موارد زیر تصمیم‌گیری می‌شود:

الف- حالت درونی

ب- سمبلی که در حال حاضر از فایل ورودی خوانده می‌شود.

ج- آنچه که بوسیله هد خواندن-نوشتن مشاهده می‌شود.

ماشین‌های تورینگ با حافظه پیچیده تر

می‌توان ابزار ذخیره سازی ماشین تورینگ استاندارد را پیچیده تر کرد، اما این عمل قدرت ماشین را افزایش نمی‌دهد. با ذکر دو مثال (چند نواره و چند بعدی)، این موضوع را نشان می‌دهیم.

ماشین‌های تورینگ چند نواره

ماشین تورینگ با چند نوار که هر نوار، دارای هد خواندن-نوشتن می‌باشد که به طور مستقل کنترل می‌شود.

در یک ماشین دو نواره، تابع انتقال $\delta(q_0, a, e) = (q_1, x, y, L, R)$ ، به این معنی است که ماشین در حالت q_0 بوده و اولین هد یک a و دومین هد یک e را می‌بیند. سپس a روی اولین نوار با X جایگزین شده و هد به سمت چپ حرکت خواهد کرد. در عین حال، e روی نوار دوم به Y تغییر یافته و هد به سمت راست حرکت می‌کند. پس از این کار واحد کنترل به q_1 تغییر حالت می‌دهد.

مثال

به کمک ماشین تورینگ دو نواره پذیرش زبان $\{a^n b^n\}$ بسیار ساده تر می شود. در ابتدا رشته $a^n b^n$ روی نوار اول قرار دارد. سپس تمامی a ها را از نوار اول خوانده و به نوار دوم کپی می کنیم. با رسیدن به اولین b روی نوار اول، آنها را با a های نوار دوم تطبیق می دهیم و به سادگی تعیین می کنیم آیا تعداد a ها و b ها برابر هستند یا خیر. بنابراین بدون نیاز به جابجایی متوالی هد به راست و چپ، این عمل انجام شد. ■

ماشین های تورینگ چند بعدی

در این ماشین‌ها، نوار به صورت نامتناهی در بیش از یک بعد گسترش یافته است. در ماشین تورینگ دو بعدی، هد علاوه بر حرکت به چپ و راست، می تواند به بالا و پایین نیز حرکت کند.

ماشین های تورینگ نامعین

ماشین تورینگ نامعین مشابه تورینگ معین است با این تفاوت که دارای تغییر وضعیت‌های متفاوتی می باشد، یعنی در هر لحظه از محاسبات، ماشین می تواند یکی از چندین انتخاب را دنبال کند. به عبارتی برد تابع انتقال آن، مجموعه تمام انتقالات ممکن است که بوسیله ماشین انتخاب می شوند.

تابع δ به صورت $Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$ تعریف می شود. تابع δ در ماشین تورینگ معین به صورت $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ می باشد.

مثال

ماشین تورینگ با انتقالاتی به فرم زیر نامعین است:

$$\delta(q_0, a) = \{(q_1, b, R), (q_2, c, L)\}$$

■

چند نکته:

- ۱- دسته ماشین های تورینگ معین و دسته ماشین های تورینگ نامعین هم ارز هستند.
- ۲- یک ماشین تورینگ نامعین به هیچ وجه قدرتمندتر از نوع معین خود نیست. در واقع نامعین بودن به قدرت ماشین تورینگ اضافه نمی کند.
- ۳- هر ماشین تورینگ نامعین را می توان بوسیله یک ماشین تورینگ معین شبیه سازی کرد.
- ۴- اگر M_N یک ماشین تورینگ نامعین باشد، آنگاه یک ماشین تورینگ معین مانند M_D وجود دارد، به طوری که: $L(M_N) = L(M_D)$

چند نکته:

۱- وقتی که بیش از یک حرکت ممکن باشد، ماشین به تعداد لازم کپی از خودش تهیه می‌کند و به هر کدام کاری را واگذار می‌کند.

۲- یک آتاماتای پشته‌ای، مانند یک ماشین تورینگ نامعین است که نوار آن به صورت پشته استفاده می‌شود.

۳- دسته اتوماتای دو پشته‌ای، هم ارز با دسته ماشین‌های تورینگ است.

ماشین تورینگ عمومی

ماشین تورینگ عمومی M_{II} ، اتوماتی است که با در اختیار داشتن توصیف هر ماشین تورینگ M بعنوان ورودی و رشته w ، قادر به شبیه‌سازی محاسبه M روی w می‌باشد.

آتاماتای کراندار خطی (LBA)

یک اتومات کراندار خطی، یک ماشین تورینگ نامعین است، ولی با این محدودیت که مقدار نواری که می‌تواند استفاده کند تابعی از ورودی است. همچنین قسمت قابل استفاده نوار به سلول‌هایی که حاوی ورودی است محدود می‌باشند و برای حفاظت از این محدوده از دو علامت گروه $[,]$ استفاده می‌شود.

(LBA : Linear Bounded Automata)

مثال

زبان $\{a^n b^n c^n : n \geq 1\}$ توسط یک LBA پذیرفته می‌شود. چون محاسباتی که برای پذیرش این زبان نیاز است، احتیاجی به فضای خارج از ورودی اولیه ندارد. ■

چند نکته:

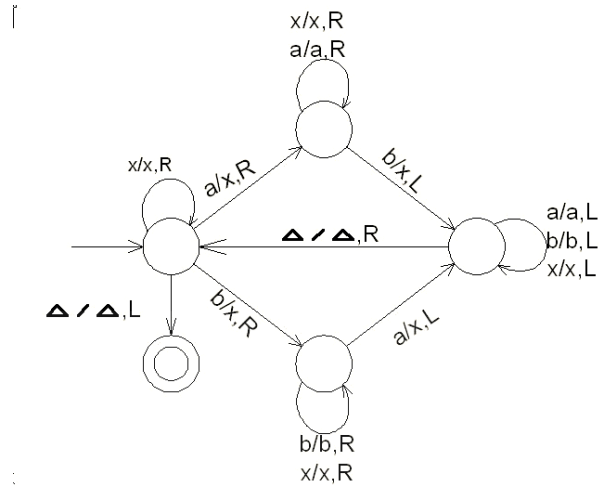
۱- هر زبان پذیرفته شده بوسیله یک اتومات پشته‌ای، بوسیله یک اتومات کراندار خطی هم پذیرفته می‌شود، اما زبان‌هایی وجود دارند که بوسیله اتوماتای کراندار خطی پذیرفته می‌شوند، اما هیچ اتوماتای پشته‌ای به ازای آن وجود ندارد. پس ماشین کراندار خطی، قویتر از ماشین پشته‌ای است.

۲- هر زبان مستقل از متن فاقد λ را می‌توان بوسیله یک LBA پذیرفت.

۳- هیچ روشی برای اثبات تناظر LBA های معین با نسخه نامعین خود وجود ندارد.

تمرین فصل ۷

۱- ماشین تورینگ زیر چه زبانی را می‌پذیرد؟ (Δ نماد blank ماشین تورینگ است)



۲- در ماشین تورینگ M با دستورات زیر اگر محتوی نوار برابر رشته $aaabbb$ باشد، پس از دقیقا 6 حرکت، محتوای چه خواهد بود؟

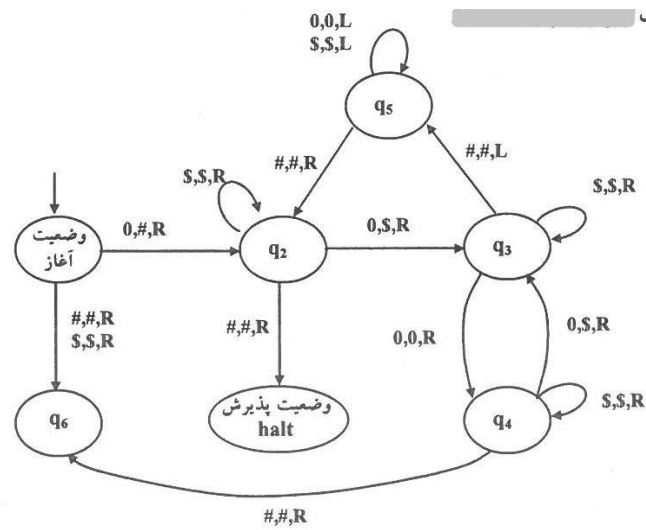
$$\begin{aligned} \delta(q_0, a) &= (q_1, X, R) & \delta(q_0, B) &= (q_f, B, R) \\ \delta(q_1, a) &= (q_1, a, R) & \delta(q_1, b) &= (q_2, Y, L) \\ \delta(q_1, Y) &= (q_1, Y, R) & \delta(q_1, B) &= (q_f, B, R) \\ \delta(q_2, a) &= (q_2, a, L) & \delta(q_2, X) &= (q_1, X, R) & \delta(q_2, Y) &= (q_2, Y, L) \end{aligned}$$

۳- زبان ماشین تورینگ با قواعد زیر کدام است؟

$$\begin{aligned} \delta(q_0, a) &= (q_1, x, R) & \delta(q_0, y) &= (q_3, y, R) \\ \delta(q_1, a) &= (q_1, a, R) & \delta(q_1, y) &= (q_1, y, R) & \delta(q_1, B) &= (q_2, y, L) \\ \delta(q_2, a) &= (q_2, a, L) & \delta(q_2, y) &= (q_2, y, L) \\ \delta(q_3, x) &= (q_0, x, R) & \delta(q_3, y) &= (q_3, y, R) & \delta(q_3, B) &= (q_f, B, R) \end{aligned}$$

۴- آیا ماشین تورینگ زیر، 000000 را می‌پذیرد؟ (نماد \$ یکی از حروف نوار ماشین است).

(نماد # نشان دهنده خانه خالی)



۵- کدام یک از زبان‌های زیر توسط ماشین تورینگ فقط خواندنی پذیرفته نمی‌شوند؟

الف- { حاصلضرب تعداد صفرهای X در تعداد یک‌های آن بر 7 بخش پذیر باشد : $x \in \{0,1\}^*$ }

ب- { تعداد صفرهای X منهای تعداد یک‌های X برابر 2 باشد : $x \in \{0,1\}^*$ }

۶- یک ماشین تورینگ در حداکثر ۶۴ مرحله به پیکربندی نهایی می‌رسد. آیا زبان این ماشین با یک عبارت منظم قابل توصیف است؟

۷- آیا تعریف الگوریتم توسط چرچ با تعریف الگوریتم توسط تورینگ معادل است؟

۸- آیا می‌توان ماشین تورینگ ساخت که خروجی نداشته باشد و به ازاء هیچ ورودی نیز در حلقه نیافتد، یعنی به ازای هر ورودی متوقف شود؟

۹- آیا با استفاده از نمادها و حالات اضافی می‌توان برای هر ماشین تورینگ قطعی، یک ماشین تورینگ قطعی معادل ساخت که سرعت محاسبه آن برای هر ورودی حداقل دو برابر ماشین اولیه باشد؟

۱۰- یک ماشین تورینگ آف لاین با چه ویژگی‌هایی، متناظر با یک ماشین متناهی است؟

۱۱- یک ماشین تورینگ تک نواره که قادر به نوشتن در محلی از نوار که شامل رشته ورودی است نباشد، چه نوع زبانی را می‌پذیرد؟

۱۲- یک ماشین تورینگ با امکان باقی ماندن در محل قبلی به جای حرکت به سمت چپ، چگونه ماشین است؟

۱۳- ماشین تورینگ چند شیاره را تعریف کنید.

۱۴- یک ماشین تورینگ یکبار مصرف (write-once TM)، چه نوع ماشینی است؟

۱۵- یک ماشین تورینگ با امکان بازنشاندن به سمت چپ (TM with left reset)، چه نوع ماشینی است؟

۱۶- آیا زبان $L = \{x^i y^j z^{j+2} w^k v^{i+k} : i, j, k \geq 0\}$ توسط LBA پذیرفته می‌شود؟

۱۷- ماشین تورینگ پذیرنده زبان $L = \{0^n 1^n : n \geq 0, n \notin \{2, 5\}\}$ را طراحی کنید.

۱۸- یک ماشین تورینگ طراحی کنید که زبان $L = \{0^{2^n} : n \geq 0\}$ را تصمیم‌گیری کند.

۱۹- ایده کار در ماشین تورینگی که وسط رشته‌ای با طول زوج را پیدا می‌کند، چگونه است؟

پاسخ تمرین فصل ۷

۱- ماشین رشته‌هایی را در الفبای $\Sigma = \{a, b, x\}$ می‌پذیرد که تعداد a ها با b ها برابر باشند. مانند رشته axb یا $bxabb$.

$$L = \{w \in (a + b + x)^* : n_a(w) = n_b(w)\}$$

۲- در زیر 6 حرکت آورده شده و در انتها $XaaYbb$ روی نوار خواهد بود:

$$q_0aaabbb \mapsto Xq_1aabbb \mapsto Xaq_1abbb \mapsto Xaaq_1bbb \mapsto Xaq_2aYbb$$

$$\mapsto Xq_2aaYbb \mapsto q_2XaaYbb$$

۳- ماشین داده شده رشته‌هایی را می‌پذیرد که تعداد a ها و تعداد b ها در آنها برابر است و با حرف a نیز شروع می‌شود. به عبارتی زبان این ماشین $\{a^n b^n : n \geq 1\}$ می‌باشد.

۴- خیر- عملکرد ماشین:

$$\begin{aligned} q_0 000000 &\mapsto q_2 00000 \mapsto q_3 0000 \mapsto q_4 000 \mapsto q_3 00 \\ &\mapsto q_4 0 \mapsto q_3 \# \mapsto q_5 \# \mapsto q_5 0 \# \mapsto q_5 \$ \# \\ &\mapsto q_5 0 \$ \# \mapsto q_5 \$ 0 \$ \# \mapsto q_5 \$ \$ 0 \$ \# \mapsto q_2 \$ 0 \$ \$ \# \\ &\mapsto q_2 0 \$ \$ \# \mapsto q_3 \$ 0 \$ \# \mapsto q_3 \$ \$ 0 \$ \# \mapsto q_4 \$ \$ \$ 0 \$ \# \\ &\mapsto q_4 \$ \$ \$ 0 \$ \# \mapsto q_6 \end{aligned}$$

ماشین در حالت q_6 متوقف شد و رشته ورودی پذیرفته نشد.

تذکر: ماشین رشته‌هایی را می‌پذیرد که تعداد صفرهای آن، توانی از ۲ باشد.

۵- در یک ماشین تورینگ فقط خواندنی، هد فقط می‌تواند بخواند و توانایی نوشتن یا پاک کردن خانه‌های نوار حافظه را ندارد. این ماشین مانند یک ماشین متناهی است و بنابراین فقط زبانهای منظم را پذیرش می‌کند. پس زبان گزاره b که مستقل از متن است را نمی‌پذیرد.

۶- بله- زبان ماشین تورینگ که در تعداد مراحل متناهی به پیکربندی نهایی می‌رسد، با یک عبارت منظم قابل توصیف است.

۷- بله- چرچ برای تعریف الگوریتم، از ریاضیات لاندن و تورینگ از ماشین استفاده کرد. طبق نظریه چرچ-تورینگ این دو تعریف معادل هستند.

۸- بله

۹- بله- به طور نمونه هر بار با خواندن دو نماد ورودی، تغییر حالت دهد.

۱۰- ویژگی‌هایی که یک ماشین تورینگ آف لاین را با یک ماشین متناهی متناظر می‌کند:

الف- ورودی فقط یک مرتبه قابل خواندن باشد.

ب- ورودی از چپ به راست حرکت کند.

پ- ورودی قابل بازنویسی نباشد.

ت- حداکثر فقط از n سلول اضافی نوار، بعنوان فضای کاری استفاده کند. (n برای تمام ورودی‌ها ثابت است).

۱۱- این ماشین قادر به تشخیص زبانهای مستقل از متن می‌باشد.

۱۲- مشابه تورینگ معمولی است ولی تابع انتقال آن به صورت $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, S\}$ می‌باشد. در هر لحظه ماشین می‌تواند هد را به سمت راست حرکت داده یا در محل قبلی باقی بماند. این نوع ماشین معادل ماشین تورینگ معمولی نیست. (منظور از S ، باقی ماندن در محل قبلی است).

۱۳- اگر در ماشین تورینگ، هر یک از سلول‌های نوار به چند بخش به نام شیار تقسیم شود به ماشین حاصل، ماشین تورینگ چند شیاری می‌گویند. علائم نوار این ماشین، چندتایی‌های الفبای ساده‌تر دیگری می‌باشند. این تغییر در تعریف ماشین تورینگ، آن را گسترش نمی‌دهد، چون فقط لازم است، Γ را الفبایی فرض کنیم که هر یک از سمبل‌های آن از چندین بخش تشکیل شده است.

۱۴- یک ماشین تورینگ تک نواره است که هر خانه از نوار را فقط یکبار می‌تواند تغییر دهد. این نوع ماشین، معادل یک ماشین تورینگ استاندارد می‌باشد.

۱۵- مشابه یک ماشین تورینگ معمولی است با این تفاوت که تابع انتقال به صورت زیر است:

$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, reset\}$$

تابع انتقال $\delta(q_i, a) = (q_j, b, reset)$ یعنی هر گاه ماشین در حالت q_i بوده و نماد a را بخواند، پس از نوشتن نماد b به جای a ، به ابتدای سمت چپ نوار بازگشته و به حالت q_j می‌رود. این ماشین نمی‌تواند یک نماد به چپ برود. این ماشین کلاس زبان‌های تشخیص پذیر تورینگ را تشخیص می‌دهد.

۱۶- زبان L را می‌توان به صورت $x^i y^j z^j z^2 w^k v^k v^i$ نوشت که زبانی مستقل از متن معین است. بنابراین توسط DPDA قابل پذیرش است. در نتیجه توسط LBA که از PDA قویتر است، نیز قابل پذیرش است.

۱۷- برای طراحی ماشین تورینگ پذیرنده این زبان، از ۴ ماشین تورینگ استفاده می‌کنیم:

ماشین اول: پذیرنده زبان $\{0^n 1^n : n \geq 0\}$

ماشین دوم: پذیرنده $0^2 1^2$

ماشین سوم: پذیرنده $0^5 1^5$

ماشین چهارم: این ماشین رشته ورودی را به ۳ ماشین بالا داده و اگر ماشین اول آن را پذیرفته و دو ماشین دیگر آنرا نپذیرند، آنگاه این ماشین رشته را می‌پذیرد. اگر یکی از ماشین‌های دوم یا سوم رشته را بپذیرند، آنگاه رشته ورودی را رد (reject) می‌کند.

۱۸- رشته‌های زبان داده شده، رشته‌هایی از صفر هستند که طول آنها توان ۲ باشد. به طور نمونه رشته‌های:

0,00,0000,00000000,...

کارهای انجام شده روی رشته ورودی توسط ماشین:

۱- از سمت چپ به راست روی نوار حرکت می‌کند تا از تمام صفرها عبور کند و صفرها را به صورت یک در میان علامت می‌گذارد.

۲- اگر در مرحله ۱ فقط بیش از یک عدد صفر وجود دارد، به مرحله پذیرش می‌رود.

۳- اگر در مرحله ۱ بیش از یک عدد صفر وجود دارد و تعداد صفرها فرد باشد، به مرحله عدم پذیرش می‌رود.

۴- هد به انتهای سمت چپ نوار بازگردانده می‌شود.

۵- به گام ۱ باز می‌گردد.

هر بار که گام اول اجرا می‌شود، تعداد صفرها نصف می‌شود. وقتی که در گام اول از روی تمام صفرها عبور می‌کند، مشخص می‌شود که تعداد صفرها فرد یا زوج بوده است. اگر تعداد صفرها فرد و بزرگتر از یک باشد، تعداد صفرها در رشته اولیه نمی‌تواند توان ۲ باشد. بنابراین ماشین رشته ورودی را نمی‌پذیرد. ولی اگر فقط یک عدد صفر باقی مانده باشد، نشان دهنده این است که تعداد صفرها در رشته ورودی توان ۲ است و رشته پذیرفته می‌شود.

۱۹- ایده کار به این صورت است که یک سمبل c در هر یک از دو سر رشته اضافه می‌کنیم. مثلاً اگر رشته ورودی $aaaa$ باشد، خواهیم داشت: $caaaac$. حال در سمت چپ به جای ca از ac و در سمت راست به جای ac از ca استفاده می‌کنیم. این کار را ادامه داده تا c ها به هم برسند. در نهایت یک c را حذف می‌کنیم. سمبل c باقی مانده، وسط رشته را نشان می‌دهد.



فصل ۸:

زبان‌های بازگشتی - گرامر بدون محدودیت و حساس به متن

خانواده زبان‌های مرتبط با ماشین‌های تورینگ بسیار گسترده‌اند، چون ماشین‌های تورینگ قادر به انجام انواع محاسبات الگوریتمی هستند. البته زبانی وجود دارد که توسط هیچ ماشین تورینگی پذیرفته نشود چون تعداد زبان‌ها، بیشتر از ماشین‌های تورینگ است.

زبان‌های بازگشتی (RE) و بازگشتی شمارش پذیر (REC)

زبان بازگشتی شمارش پذیر (برشمردنی): زبانی که ماشین تورینگی وجود داشته باشد که آنرا پذیرش کند.

زبان بازگشتی: زبانی که ماشین تورینگی باشد که آنرا پذیرفته و روی هر w موجود در Σ^+ ، در یک حالت پایانی یا غیر پایانی، متوقف شود.

بین زبانی که یک ماشین تورینگ پذیرنده به ازای آن وجود دارد و زبانی که یک الگوریتم عضویت به ازای آن وجود دارد، تفاوت هست. وجود یک ماشین تورینگ پذیرنده، به معنای وجود الگوریتم عضویت مربوطه نیست، چون ماشین‌های تورینگ لزوماً برای ورودی که آنرا نمی‌پذیرند، توقف نمی‌کنند و ممکن است در حلقه بی‌افتند.

هر زبانی که بوسیله یک روش الگوریتمی مستقیم قابل توصیف باشد، بوسیله یک ماشین تورینگ هم قابل پذیرش بوده و بنابراین بازگشتی شمارش پذیر است.

زبان‌های بازگشتی زیر مجموعه زبان‌های بازگشتی شمارش پذیر می‌باشند. به عبارتی هر زبان بازگشتی، بازگشتی شمارش پذیر نیز می‌باشد. یک زبان بازگشتی شمارش پذیر وجود دارد که بازگشتی نیست.

هر زبان بازگشتی می‌تواند زبان یک ماشین تورینگ غیر قطعی باشد.

مثال

زبان‌های زیر، هم بازگشتی هستند و هم بازگشتی شمارش پذیر :

$$L = \{a^n b^n c^{2n} : n \geq 0\}$$

$$L = \{a^n b^n c^n d^n : n > 0\}$$

$$L = \{a^n b^m c^n d^m : n \geq 1, m \geq 1\}$$

$$L = \{ww : w \in \{a, b\}^+\}$$

$$L = \{w \in \{a, b, c\}^* : n_a(w) = n_b(w) = n_c(w)\}$$

$$L = \{w \in \{a, b, c\}^* : n_a(w) = n_b(w) \leq n_c(w)\}$$



مثال

چرا زبان بازگشتی شمارش پذیر، را زبان تشخیص پذیر نیز می نامند؟

حل: یک تشخیص دهنده، الگوریتمی است که در صورت تعلق یک رشته به یک زبان می تواند تعلق آن را مشخص کند. اگر رشته متعلق به زبان نباشد، ماشین یا در حالت غیر پایانی متوقف شده و یا در حلقه بی‌نهایت (Loop) می افتد. بنابراین زبان بازگشتی شمارش پذیر، را زبان تشخیص پذیر نیز می نامند. ■

مثال

چرا زبان بازگشتی را زبان تصمیم پذیر نیز می نامند؟

حل: یک تصمیم گیرنده برای یک زبان، الگوریتمی است که مشخص می کند آیا رشته w متعلق به زبان هست یا خیر و در هر صورت ماشین نهایتاً متوقف می شود. بنابراین زبان بازگشتی را زبان تصمیم پذیر (decidable) نیز می نامند. ■

ویژگی‌های زبان‌های بازگشتی و بازگشتی شمارش پذیر

- ۱- یک زبان بازگشتی خواهد بود اگر و تنها اگر یک الگوریتم عضویت به ازای آن وجود داشته باشد.
- ۲- اگر L بازگشتی باشد، لزوماً L^+ هم بازگشتی است.
- ۳- مکمل هر زبان بازگشتی، بازگشتی است.
- ۴- مکمل یک زبان مستقل از متن، حتماً بازگشتی است.
- ۵- اشتراک یک زبان بازگشتی شمارش پذیر با زبان بازگشتی، لزوماً غیر بازگشتی نیست.
- ۶- اشتراک یک زبان بازگشتی شمارش ناپذیر با زبان بازگشتی، لزوماً بازگشتی شمارش پذیر نیست.
- ۷- اشتراک زبان بازگشتی شمارش ناپذیر با بازگشتی شمارش پذیر، لزوماً بازگشتی شمارش پذیر نیست.
- ۸- اگر L یک زبان متناهی باشد، آنگاه L^+ بازگشتی شمارش پذیر است.
- ۹- اگر L یک زبان مستقل از متن باشد، آنگاه L^+ بازگشتی شمارش پذیر است.
- ۱۰- اگر L_1 بازگشتی و L_2 بازگشتی شمارش پذیر باشد، آنگاه $L_2 - L_1$ لزوماً بازگشتی شمارش پذیر است.
- ۱۱- اگر L و \bar{L} هر دو بازگشتی شمارش پذیر باشند، آنگاه هر دو بازگشتی هستند.
- ۱۲- اگر L بازگشتی باشد، آنگاه \bar{L} هم بازگشتی است و در نتیجه هر دو بازگشتی شمارش پذیر هستند.
- ۱۳- اگر یک زبان بازگشتی شمارش پذیر نباشد، مکمل آن بازگشتی نیست.
- ۱۴- مکمل زبان بازگشتی شمارش پذیر، لزوماً بازگشتی شمارش پذیر نیست.
- ۱۵- یک زبان بازگشتی شمارش پذیر وجود دارد که مکمل آن بازگشتی شمارش پذیر نیست.
- ۱۶- یک زبان تصمیم پذیر است اگر و تنها اگر خود زبان و مکمل آن هر دو تشخیص پذیر تورینگ باشند.
- ۱۷- فرض کنید L چنان باشد که یک ماشین تورینگ وجود دارد که اعضای L را در ترتیب مناسب شمارش می‌کند. در اینصورت، L بازگشتی است.
- ۱۸- مجموعه تمام زبان‌هایی که بازگشتی شمارش پذیر نیستند، قابل شمارش نمی‌باشند.

نکاتی در رابطه با شمارش پذیر بودن

- ۱- اگر S یک مجموعه شمارش پذیر نامتناهی باشد، آنگاه مجموعه توانی آن، (2^S) شمارش پذیر نیست.
- ۲- مجموعه اعداد حقیقی R ، ناشمارا می باشد، چون تناظری بین R و N موجود نیست.
- ۳- مجموعه تمامی رشته‌ها روی الفبا Σ و Σ^* ، شمارش پذیر هستند.
- ۴- مجموعه تمامی دنباله‌های نامحدود، شمارش پذیر نیستند.
- ۵- تمامی زبان‌های سلسله مراتب چامسکی، شمارش پذیر هستند.
- ۶- مجموعه تمام ماشین‌های تورینگ، شمارش پذیر هستند.
- ۷- تمامی ماشین‌های منظم، پشته‌ای، LBA و تورینگ، نامتناهی ولی شمارش پذیر هستند.
- ۸- ضرب دکارتی تعدادی مجموعه شمارش پذیر، باز هم شمارش پذیر است.

گرامر بدون محدودیت

گرامر مفروض $G = (V, T, S, P)$ بدون محدودیت خوانده می شود، اگر تمامی قوانین آن به فرم $u \rightarrow v$ باشند که در آن u ، عضو $(V \cup T)^+$ و v عضو $(V \cup T)^*$ می باشد. در گرامرهای بدون محدودیت، اساساً هیچ شرط و محدودیتی برای قواعد تولید قائل نمی شویم. بعلاوه هر تعداد غیر پایانی و پایانی را می توان با هر ترتیبی در طرفین راست و چپ قرار داد. فقط λ نمی تواند در سمت چپ قواعد تولید رخ دهد. این گرامر ها بسیار قدرتمندتر از گرامرهای منظم و مستقل از متن هستند.

گرامرهای بدون محدودیت، متناظر با بزرگترین خانواده زبان‌ها بوده و بوسیله ابزار مکانیکی قابل تشخیص می باشند.

مثال

زبان تولید شده توسط گرامر بدون محدودیت زیر را تعیین کنید.

$$S \rightarrow aSBC \mid \lambda$$

$$aB \rightarrow ab$$

$$bC \rightarrow bc$$

$$bB \rightarrow bb$$

$$cC \rightarrow cc$$

$$cB \rightarrow Bc$$

حل: گرامر رشته‌هایی را تولید می‌کند که تعداد a ها و b ها و c ها با یکدیگر برابر باشند و ابتدا a ها، سپس b ها و در نهایت c ها ظاهر شوند. بنابراین داریم: $L = \{a^n b^n c^n : n \geq 0\}$ ■

مثال

زبان گرامر بدون محدودیت زیر را مشخص کنید.

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow aAb \\ bB &\rightarrow bbbB \\ aAb &\rightarrow aa \\ B &\rightarrow \lambda \end{aligned}$$

حل: یک مسیر اشتقاق را طی می‌کنیم:


$$S \Rightarrow AB \Rightarrow aAbB \Rightarrow aaAbbbB \Rightarrow a^n Ab^n B \Rightarrow a^{n+1} b^{n-1} B \Rightarrow a^{n+1} b^{n+1} B \Rightarrow \dots$$


بر این اساس می‌توان ادعا کرد که گرامر فوق‌قادر به مشتق‌سازی زبان زیر است:

$$L = \{a^{n+1} b^{n+2k-1} : n \geq 1, k \geq 0\}$$

■

گرامرهای بدون محدودیت، صرفاً خانواده زبان‌های بازگشتی شمارش‌پذیر را ایجاد می‌کنند. 

هر زبانی که بوسیله یک گرامر بدون محدودیت ایجاد شود، بازگشتی شمارش‌پذیر است. 

برای هر زبان بازگشتی شمارش‌پذیر L ، یک گرامر بدون محدودیت G وجود دارد، بطوریکه 

$$L=L(G).$$

گرامر حساس به متن

گرامر مفروض $G = (V, T, S, P)$ حساس به متن (وابسته به متن) خوانده می‌شود، اگر تمامی قوانین آن به فرم $x \rightarrow y$ باشند که در آن x و y عضو $(V \cup T)^+$ باشند و $|x| \leq |y|$.

طبق تعریف بالا، قاعده $x \rightarrow \lambda$ غیر مجاز است. بنابراین گرامرهای حساس به متن هرگز قادر به تولید زبانهای دارای رشته تهی نمی‌باشند.

تعریف: زبان مفروض L حساس به متن خوانده می‌شود، اگر گرامر حساس به متن G وجود داشته باشد، بطوریکه $L = L(G) \cup \{\lambda\}$ یا $L = L(G)$.

مثال

یک گرامر حساس به متن برای زبان $L = \{a^n b^n c^n : n \geq 1\}$ پیدا کنید.

$$S \rightarrow abc \mid aAbc$$

$$Ab \rightarrow bA$$


$$Ac \rightarrow Bbcc$$


$$bB \rightarrow Bb$$

$$aB \rightarrow aa \mid aaA$$

تذکر: چون زبان حساس به متن $\{a^n b^n c^n\}$ ، مستقل از متن نیست، می‌توان نتیجه گرفت که خانواده زبانهای مستقل از متن، یکی از زیر مجموعه‌های خانواده زبان‌های حساس به متن است. ■

تمامی زبان‌های حساس به متن، بازگشتی هستند. 

یک زبان بازگشتی وجود دارد که حساس به متن نمی‌باشد. 

هر زبان حساس به متنی بوسیله یک ماشین تورینگ پذیرفته شده و بنابراین، بازگشتی شمارش پذیر محسوب می‌شود. 

مثال

یک گرامر حساس به متن برای زبان $L = \{a^n b^n c^n d^n : n > 0\}$ پیدا کنید.

حل:

$$S \rightarrow aSBCD \mid abcd$$

$$cB \rightarrow Bc$$

$$dB \rightarrow Bd$$

$$dC \rightarrow Cd$$

$$bB \rightarrow bb$$

$$cC \rightarrow cc$$

$$dD \rightarrow dd$$

■

مثال

یک گرامر حساس به متن برای زبان $L = \{a^n b^m c^n d^m : n \geq 1, m \geq 1\}$ پیدا کنید.

حل:

$$S \rightarrow aAcD \mid aBcD$$

$$A \rightarrow aAc \mid aBc$$

$$Bc \rightarrow cB$$

$$Bb \rightarrow bB$$


$$BD \rightarrow Ed$$


$$cE \rightarrow Ec$$


$$bE \rightarrow Eb$$

$$aE \rightarrow ab$$



به ازای هر زبان L حساس به متن دارای λ ، یک اتومات کراندار خطی M وجود دارد بطوریکه $L=L(M)$. 


اگر زبان L بوسیله یک اتومات کراندار خطی مفروض به نام M پذیرفته شود، آنگاه یک گرامر حساس به متن وجود دارد که L را تولید می کند. 

اتوماتای کراندار خطی عملاً ضعیف‌تر از ماشین‌های تورینگ بوده و فقط قادر به پذیرش یکی از زیر مجموعه‌های مناسب زبان‌های بازگشتی می باشند. 

ارتباط بین زبان‌ها، گرامرها و ماشین‌ها

جدول زیر ارتباط بین زبان‌ها، گرامرها و ماشین‌ها را نشان می‌دهد:

گرامر	زبان	ماشین
منظم	منظم	متناهی (FA)
	خطی	
	مستقل از متن معین	پشته‌ای معین (DPDA)
مستقل از متن	مستقل از متن	پشته‌ای (PDA)
	حساس به متن	کراندار خطی (LBA)
	بازگشتی	تورینگ تصمیم‌گیرنده
بدون محدودیت	بازگشتی شمارش‌پذیر	تورینگ تشخیص‌دهنده

ماشین‌های تورینگ تشخیص‌دهنده، قدرت بیشتری از ماشین‌های تورینگ تصمیم‌گیرنده دارند. 

سلسله مراتب چامسکی

نوام چامسکی زبان‌ها را در چهار گروه، از نوع صفر تا نوع سه، دسته‌بندی کرد.

زبان‌های نوع صفر: شامل زبان‌های بازگشتی شمارش‌پذیر، می‌باشند.

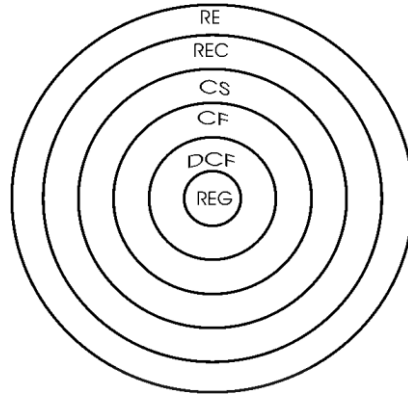
زبان‌های نوع یک: شامل زبان‌های حساس به متن می‌باشند.

زبان‌های نوع دو: شامل زبان‌های مستقل از متن می‌باشند.

زبان‌های نوع سه: شامل زبان‌های منظم می‌باشند.

هر یک از خانواده‌های زبان‌های نوع i ، یکی از زیر مجموعه‌های مناسب خانواده نوع $i-1$ محسوب می‌شوند.

نمودار زیر این رابطه را مشخص می‌کند:



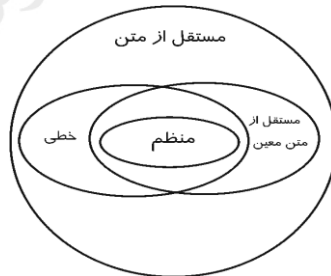
ارتباط بین زبان‌ها را می‌توان به کمک رابطه زیر نمایش داد:

$$REG \subseteq DCF \subseteq CF \subseteq CS \subseteq REC \subseteq RE$$

بازگشتی شمارش پذیر \subseteq بازگشتی \subseteq حساس به متن \subseteq مستقل از متن \subseteq مستقل از متن معین \subseteq منظم

هر زبان منظمی، زبانی مستقل از متن است. (چون زبان‌های منظم حالت خاصی از زبانهای مستقل از متن می‌باشند).

در شکل زیر جایگاه زبان‌های خطی و مستقل از متن معین (قطعی) نشان داده شده است.



با توجه به شکل می‌توان گفت که:

الف: تمامی زبان‌های خطی، مستقل از متن نیز هستند.

ب: زبان خطی وجود دارد که مستقل از متن معین نیست. مانند $L = \{a^n b^n\} \cup \{a^n b^{2n}\}$

پ: زبان مستقل از متن معینی وجود دارد که خطی نیست. مانند: $L = \{w : n_a(w) = n_b(w)\}$

بررسی بسته بودن زبان‌ها تحت عملگرها

جدول زیر خواص بسته بودن شش نوع زبان را تحت عملگرهای مختلف نشان می‌دهد:

بازگشتی شمارش پذیر	بازگشتی	حساس به متن	مستقل از متن	مستقل از متن معین	منظم	
✓	✓	✓	✓	-	✓	الحاق
✓	✓	✓	✓	-	✓	اجتماع
✓	✓	✓	-	-	✓	اشتراک
-	✓	✓	-	✓	✓	مکمل
✓	✓	✓	✓	-	✓	معکوس
✓	✓	✓	✓	-	✓	بستار ستاره
✓	-	-	✓	-	✓	همریختی

همه انواع زبان‌های جدول بالا، تحت اشتراک منظم و تحت همریختی معکوس بسته هستند.

تمرین‌های فصل ۸

۱- چه زبانی توسط گرامر حساس به متن زیر تولید می‌شود؟

$$S \rightarrow aSBD \mid abD$$

$$bB \rightarrow bb$$

$$bD \rightarrow bc$$

$$DB \rightarrow BD$$

$$cD \rightarrow cc$$

$$S \rightarrow AB$$

$$A \rightarrow aAb$$

$$bB \rightarrow bbbB$$

$$aAb \rightarrow aa$$

$$B \rightarrow \lambda$$

۲- چه زبانی توسط گرامر بدون محدودیت زیر تولید می‌شود؟

۳- چه زبانی توسط گرامر بدون محدودیت زیر تولید می‌شود؟

$$S \rightarrow aSC \mid aAC$$

$$A \rightarrow bAD \mid bBD$$

$$BC \rightarrow c$$

$$DC \rightarrow CD$$

$$cC \rightarrow cc$$

$$D \rightarrow d$$

۴- چه زبانی توسط گرامر بدون محدودیت زیر تولید می‌شود؟

$$S \rightarrow ABaC$$

$$Ba \rightarrow aaB$$

$$BC \rightarrow DC \mid E$$

$$aD \rightarrow Da$$

$$AD \rightarrow AB$$

$$aE \rightarrow Ea$$

$$AE \rightarrow \lambda$$

۵- آیا زبان $L = \{a^n b^m c^n d^m : n, m \geq 0\}$ بازگشتی است؟

۶- با فرض اینکه L_1 و L_2 زبان‌های بازگشتی شمارش پذیر باشند، آیا $L_1 \cup L_2$ بازگشتی شمارش پذیر است؟

۷- اگر $\langle M \rangle$ یک کودینگ از ماشین تورینگ باشد، آیا زبان L ، تصمیم پذیر است؟

$$L = \{ \langle M, w \rangle : \text{می پذیرد} \}$$

۸- در ماشین تورینگ، اگر نوار از یک سمت محدود شود، آیا کلاس زبانهای مشخص شده با کلاس زبانهای بازگشتی شمارش پذیر، تفاوت خواهد کرد؟

۹- اگر تعداد نمادهای ماشین تورینگ محدود شود، آیا کلاس زبانهای مشخص شده با کلاس زبانهای بازگشتی شمارش پذیر تفاوت خواهد کرد؟

۱۰- اگر کلاس زبانهای شمارش پذیر بازگشتی، کلاس زبانهای پذیرفته شده توسط ماشینهای تورینگ قطعی یک نواره باشد، آیا محدود کردن حرکت هد ماشین فقط به سمت راست، منجر به تغییر این کلاس می شود؟

۱۱- کدام گزاره درست است؟

الف- برای هر زبان بازگشتی شمارش پذیر، ماشین تورینگ قطعی وجود دارد که برای هر ورودی داخل زبان در زمان متنهایی متوقف می شود.

ب- برای زبان بازگشتی شمارش پذیر، ممکن است ماشین با یک رشته خارج از زبان متوقف نشود.

۱۲- اگر A و B دو مجموعه تصمیم پذیر (decidable) باشند، آیا $A \cap B^c$ لزوماً تصمیم پذیر است؟

حل تمرین فصل ۸

۱- چند اشتقاق نمونه به صورت زیر است:

$$S \Rightarrow abD \Rightarrow abc$$

$$S \Rightarrow aSBD \Rightarrow aabDBD \Rightarrow aabBDD \Rightarrow aabbDD \Rightarrow aabbcD \Rightarrow aabbc$$

بنابراین زبان گرامر $\{a^n b^n c^n : n \geq 1\}$ است.

۲- زبان تولید شده برابر است با:

$$L = \{a^{n+2} b^{n+2k} : n \geq 0, k \geq 0\}$$

۳- یک اشتقاق نمونه به صورت زیر است:

$$\begin{aligned} S &\Rightarrow aSC \Rightarrow aaACC \Rightarrow aabBDCC \Rightarrow aabBCDC \Rightarrow aabcDC \\ &\Rightarrow aabcCD \Rightarrow aabccD \Rightarrow aabccd \end{aligned}$$

بنابراین رشته $a^2 b c^2 d$ تولید شد. زبان گرامر $\{a^n b^m c^n d^m : n, m > 0\}$ است.

۴- با تولید چند جمله گرامر می توان حدس زد که زبان گرامر $\{a^{2^n} : n \geq 1\}$ است.

$$S \Rightarrow ABaC \Rightarrow AaaBC \Rightarrow AaaE \Rightarrow AaEa \Rightarrow AEaa \Rightarrow aa$$

۵- بله - چون این زبان را با LBA می توان پذیرفت.

۶- بله- زبان‌های بازگشتی شمارش پذیر، تحت بستار ستاره ای بسته هستند، پس زبان L_1^* بازگشتی شمارش پذیر است. همچنین زبان‌های بازگشتی شمارش پذیر، تحت اجتماع بسته هستند، پس زبان $L_2 \cup L_1^*$ بازگشتی شمارش پذیر است. و در نهایت زبان‌های بازگشتی شمارش پذیر، تحت الحاق بسته هستند، پس زبان $(L_2 \cup L_1^*)$ بازگشتی شمارش پذیر است.

۷- بله- چون ماشین تورینگ بعد از چند گام مشخص، رشته W را می پذیرد، پس زبان L، تصمیم پذیر است.

۸- خیر- تحمیل این شرط باعث نمی شود که قدرت تورینگ کم شود.

۹- بله- محدود کردن تعداد نمادهای ماشین، ممکن است قدرت ماشین را کم کند.

۱۰- بله- اگر در یک ماشین تورینگ، هد نتواند به سمت چپ حرکت کند، قدرت ماشین تغییر خواهد کرد.

۱۱- هر دو گزاره درست هستند.

۱۲- بله- مکمل هر مجموعه تصمیم پذیر، تصمیم پذیر است، بنابراین B^c تصمیم پذیر است و چون مجموعه های تصمیم پذیر تحت اشتراک بسته هستند، بنابراین $A \cap B^c$ تصمیم پذیر است.



فصل ۹:

تصمیم پذیری - کاهش پذیری

زبان‌هایی وجود دارند که نمی‌توان تعلق یا عدم تعلق یک رشته به آن زبان‌ها را تعیین کرد، یعنی دارای الگوریتم عضویت نمی‌باشند و تصمیم ناپذیرند. در این فصل به چند نمونه از این زبان‌ها اشاره می‌شود.

زبان‌های تصمیم ناپذیر

با فرض اینکه M و M_1 و M_2 یک ماشین تورینگ باشند، زبان‌های زیر تصمیم پذیر (Undecidable) نیستند.

$$L = \{ \langle M, w \rangle : w \text{ را می پذیرد.} \}$$

$$L = \{ \langle M, w \rangle : w \text{ متوقف شود.} \}$$

$$L = \{ \langle M \rangle : L(M) \text{ دو رشته به طول یکسان دارد.} \}$$

$$L = \{ \langle M \rangle : L(M) \text{ یک زبان منظم است.} \}$$

$$L = \{ \langle M \rangle : L(M) = \phi \}$$

$$L = \{ \langle M \rangle : L(M) \text{ متناهی است.} \}$$

$$L = \{ \langle M \rangle : L(M) \text{ بیش از 10 عضو دارد.} \}$$

$$L = \{ \langle M \rangle : \text{هر رشته به طول 10 را می پذیرد.} \}$$

$$L = \{ \langle M \rangle : L(M) = \Sigma^* \}$$

$$L = \{ \langle M \rangle : \text{ماشین رشته } w^R \text{ را می پذیرد اگر رشته } w \text{ را بپذیرد.} \}$$

$$L = \{ \langle M, w \rangle : L(M) \text{ رشته ای به طول } k \text{ دارد.} \}$$

$$L = \{ \langle M_1, M_2 \rangle : L(M_1) = L(M_2) \}$$

$$L = \{ \langle M_1, M_2 \rangle : L(M_1) \subseteq L(M_2) \}$$

با فرض اینکه G و G_1 و G_2 یک گرامر مستقل از متن باشند، زبان‌های زیر تصمیم پذیر نیستند:

$$L = \{ \langle G \rangle : G \text{ مبهم است} \}$$

$$L = \{ \langle G \rangle : L(G) = \Sigma^* \}$$

$$L = \{ \langle G_1, G_2 \rangle : L(G_1) = L(G_2) \}$$

$$L = \{ \langle G_1, G_2 \rangle : L(G_1) \subseteq L(G_2) \}$$

$$L = \{ \langle G_1, G_2 \rangle : L(G_1) \cap L(G_2) = \emptyset \}$$

زبان‌های زیر تصمیم پذیر نیستند:

$$L = \{ \langle G_1, G_2 \rangle : L(G_1) \cap L(G_2) = \emptyset, \text{ گرامر بدون محدودیت و } G_2, \text{ گرامر منظم است} \}$$

$$L = \{ \langle G \rangle : L(G) = L(G)^R, \text{ یک گرامر بدون محدودیت است} \}$$

$$L = \{ \langle G \rangle : L(G) = \emptyset, \text{ یک گرامر بدون محدودیت است} \}$$

$$L = \{ \langle M \rangle : L(M) = \Sigma^*, \text{ یک ماشین پشته ای است} \}$$

$$L = \{ \langle M \rangle : L(M) = \emptyset, \text{ یک LBA است} \}$$

زبان‌های تصمیم پذیر (بازگشتی)

با فرض اینکه M و M_1 و M_2 یک dfa باشند، زبان‌های زیر تصمیم پذیر هستند:

$$L = \{ \langle M \rangle : L(M) = \emptyset \}$$

$$L = \{ \langle M \rangle : \text{ماشین } \Sigma^* \text{ را تشخیص می‌دهد} \}$$

$$L = \{ \langle M \rangle : \text{هیچ رشته‌ای که تعداد فرد یک داشته باشد را نمی‌پذیرد} \}$$

$$L = \{ \langle M, w \rangle : \text{M رشته ورودی } w \text{ را می‌پذیرد} \}$$

$$L = \{ \langle M, w \rangle : \text{اگر M رشته } w \text{ را بپذیرد، آنگاه } w^R \text{ را نیز می‌پذیرد} \}$$

$$L = \{ \langle M_1, M_2 \rangle : L(M_1) = L(M_2) \}$$

با فرض اینکه G یک گرامر مستقل از متن باشد، زبان‌های زیر تصمیم پذیر هستند:

$$L = \{ \langle G \rangle : L(G) = \emptyset \}$$

$$L = \{ \langle G \rangle : \text{گرامر } G, \text{ رشته } w \text{ را تولید می کند.} \}$$

$$L = \{ \langle G \rangle : |L(G)| < \infty \text{ یعنی } L(G) \text{ متناهی است,} \}$$

$$L = \{ \langle G \rangle : \text{رشته } w \in L(G) \text{ با طول کمتر از } n \text{ وجود دارد.} \}$$

$$L = \{ \langle G, w \rangle : \text{گرامر } G, \text{ رشته } w \text{ را تولید می کند.} \}$$

زبان‌های زیر تصمیم پذیر (بازگشتی) هستند:

$$L = \{ \langle G \rangle : L(G) = \Sigma^* \text{ و } G \text{ یک گرامر منظم است} \}$$

$$L = \{ \langle R, w \rangle : \text{عبارت منظمی است که رشته ورودی } w \text{ را تولید می کند.} \}$$

$$L = \{ \langle R, S \rangle : L(R) \subseteq L(S) \text{ و } R \text{ و } S \text{ دو عبارت منظم هستند} \}$$

$$L = \{ \langle M, w \rangle : \text{یک NFA است که رشته ورودی } w \text{ را می پذیرد.} \}$$

$$L = \{ \langle M, w \rangle : \text{یک ماشین تورینگ بوده و پس از 10 گام رشته ورودی } w \text{ را می پذیرد.} \}$$

$$L = \{ \langle M, w \rangle : \text{یک LBA بوده که رشته ورودی } w \text{ را می پذیرد.} \}$$

$$L = \{ \langle M \rangle : \text{یک اتوماتای پشته ای است که دارای حالت بی فایده است.} \}$$

$$L = \{ \langle M \rangle : \text{یک ماشین تورینگ است که دارای حالت بی فایده است.} \}$$

حالت بی فایده در یک ماشین، حالتی است که به ازای هیچ رشته ورودی، ماشین به آن حالت وارد نمی شود.

تصمیم‌پذیری در زبان‌های منظم

با فرض اینکه L و L_1 و L_2 زبان منظم باشند، الگوریتم‌های وجود دارد که تعیین کند که:

۱- آیا $\lambda \in L$ ؟

۲- آیا $w^R \in L$ ؟

۳- آیا $L = L^*$ ؟

۴- تعیین تپی بودن L .

۵- تعیین متناهی یا نامتناهی بودن L .

۶- تعیین وجود یا عدم وجود w در L ($w \in \Sigma^*$).

۷- تعیین $L_1 = L_2$.

۸- آیا $w \in L_1 - L_2$ ؟

۹- آیا $L_1 \subseteq L_2$ ؟

۱۰- آیا $L_1 = L_1 L_2$ ؟

۱۱- آیا $L = L_1 L_2$ ؟

کاهش پذیری

کاهش دادن (reduction)، روشی برای تبدیل یک مسئله به مسئله دیگری است به طوری که حل مسئله دوم به حل مسئله اولیه کمک کند. به چند مثال توجه کنید:

۱- مسئله اندازه گیری مساحت یک مستطیل به مسئله اندازه گیری طول و عرض آن کاهش می یابد.

۲- مسئله حل یک سیستم معادلات خطی به معکوس کردن یک ماتریس کاهش می یابد.

۳- مسئله پیدا کردن مسیر در شهر، به مسئله خرید نقشه شهر کاهش می یابد.

در نظریه محاسبه پذیری، اگر A به B کاهش پذیر بوده و B تصمیم پذیر باشد، آنگاه A نیز تصمیم پذیر خواهد بود. همچنین اگر A تصمیم پذیر نبوده و به B کاهش یابد، آنگاه B نیز تصمیم پذیر نخواهد بود. برای نشان دادن اینکه یک مسئله تصمیم ناپذیر است، نشان می دهیم که یک مسئله تصمیم ناپذیر به این مسئله کاهش یافته است.

زبان A ، نگاشت کاهش پذیر به زبان B می باشد و به صورت $A \leq_m B$ نوشته می شود، اگر یک تابع محاسباتی $f: \Sigma^* \rightarrow \Sigma^*$ موجود باشد که برای هر رشته $w: w \in A \Rightarrow f(w) \in B$.

یک نگاشت کاهش پذیر از A به B ، روشی برای تبدیل سئوالات مربوط به عضویت A به سئوالات مربوط به عضویت B می باشد.

✍ اگر مسئله ای، نگاشت کاهش به مسئله دیگری که قبلا حل شده داشته باشد، می توانیم مسئله اول را نیز حل شده فرض کنیم.

چند نکته:

۱- رابطه \leq_m یک رابطه متعددی (تراپا) است.

۲- اگر $A \leq_m B$ بوده و B تصمیم پذیر باشد، آنگاه A نیز تصمیم پذیر است.

۳- اگر $A \leq_m B$ بوده و A تصمیم ناپذیر باشد، آنگاه B نیز تصمیم ناپذیر است.

۳-اگر $A \leq_m B$ بوده و B تشخیص پذیر تورینگ باشد، آنگاه A نیز تشخیص پذیر تورینگ است.

۴- اگر $A \leq_m B$ بوده و A تشخیص پذیر تورینگ نباشد، آنگاه B تشخیص پذیر تورینگ نخواهد بود.

۵- اگر A تشخیص پذیر تورینگ باشد و $A \leq_m \bar{A}$ باشد، آنگاه A تصمیم پذیر است.

تعریف: زبان A را کاهش پذیر تورینگ (Turing Reducible) به زبان B گویند و با $A \leq_T B$ نشان می دهند، اگر نسبت به B تصمیم پذیر باشد. اگر $A \leq_T B$ بوده و B تصمیم پذیر باشد، آنگاه A نیز تصمیم پذیر است.

✍ برای هر دو زبان دلخواه A و B ، زبانی مانند C وجود دارد بطوریکه: $B \leq_T C$ و $A \leq_T C$

تمرین‌های فصل ۹

۱- کدام یک از زبان‌های زیر تصمیم‌پذیر هستند؟

$$L_1 = \{ \langle M \rangle : L(M) \text{ متناهی است.} \}$$

$$L_2 = \{ \langle M, w \rangle : M \text{ یک ماشین تورینگ بوده و رشته ورودی } w \text{ را می‌پذیرد.} \}$$

$$L_3 = \{ \langle M_1, M_2 \rangle : L(M_1) = L(M_2) \text{ است.} \}$$

$$L_4 = \{ \langle M, w \rangle : M \text{ یک ماشین تورینگ بوده و پس از 10 گام رشته ورودی } w \text{ را می‌پذیرد.} \}$$

۲- کدام یک از زبان‌های زیر تصمیم‌پذیر هستند؟

$$L_1 = \{ \langle G \rangle : L(G) = \phi \text{ است.} \}$$

$$L_2 = \{ \langle M \rangle : L(M) = \Sigma^* \text{ است.} \}$$

$$L_3 = \{ \langle G \rangle : L(G) = \Sigma^* \text{ است.} \}$$

$$L_4 = \{ \langle G_1, G_2 \rangle : L(G_1) \cap L(G_2) = \phi \text{ است.} \}$$

۳- در جدول زیر، مسائلی که تصمیم‌پذیر هستند، را مشخص کنید.

متناهی بودن L	$L = \Sigma^*$	$L = \phi$	$w \in L$	
				منظم
				مستقل از متن
				حساس به متن
				بازگشتی
				بازگشتی شمارش‌پذیر

۴- دو گزاره زیر چه رابطه‌ای با هم دارند؟

الف- ابهام در گرامرهای مستقل از متن یک مساله تصمیم ناپذیر است.

ب- حداقل یک مساله تصمیم ناپذیر قابل کاهش (Reducible) به مساله ابهام در گرامرهای مستقل از متن وجود دارد.

۵- برشمارنده (enumerator) را تعریف کنید.

۶- کدام گزاره(ها) درست است؟

الف- یک زبان تشخیص پذیر تورینگ است اگر و تنها اگر برشمارنده‌ای برای برشمردن آن موجود باشد.

ب- یک زبان تصمیم پذیر است اگر و تنها اگر یک برشمارنده برای برشمردن آن زبان با ترتیب فرهنگ لغت (lexicographic) موجود باشد.

۷- کدام گزینه نادرست است؟

(۱) اگر $A \leq_m B$ بوده و B تصمیم پذیر باشد، آنگاه A نیز تصمیم پذیر است.

(۲) اگر $A \leq_m B$ بوده و A تصمیم ناپذیر باشد، آنگاه B نیز تصمیم ناپذیر است.

(۳) اگر $A \leq_m B$ بوده و B تشخیص پذیر تورینگ باشد، آنگاه A نیز تشخیص پذیر تورینگ است.

(۴) اگر $A \leq_m B$ بوده و A تشخیص پذیر تورینگ باشد، آنگاه B تشخیص پذیر تورینگ نخواهد بود.

۸- برای کدام یک از گزاره‌های زیر، الگوریتم وجود دارد؟

الف- تعیین مستقل از متن بودن $L_1 \cap L_2$ (L_1 مستقل از متن و L_2 منظم است)

ب- وجود عضو مشترک در میان یک زبان منظم و یک زبان مستقل از متن.

۹- برای کدام یک از گزاره‌های زیر، الگوریتم وجود دارد؟

الف- آیا $L = \text{tail}(L)$ ؟

ب- تشخیص زبان‌های منظم دو طرفه.

ج- آیا L حاوی رشته w ای است که \widehat{w} زیر رشته‌ای از آن باشد؟

۱۰- اگر L یک زبان منظم باشد، برای کدام یک از گزاره‌های زیر، الگوریتم وجود دارد؟

الف- آیا L حاوی رشته‌ای با طول زوج است یا خیر؟

ب- آیا رابطه $|L| \geq 5$ در هر زبان منظم برقرار است یا خیر؟

ج- آیا L حاوی تعداد متناهی رشته با طول زوج است یا خیر؟

۱۱- اگر L_1 یک زبان مستقل از متن و L_2 یک زبان منظم باشند، نشان دهید الگوریتمی برای تعیین اینکه آیا L_1 و L_2 عنصر مشترکی دارند یا خیر، وجود دارد.

۱۲- نشان دهید که الگوریتمی وجود دارد که مشخص کند که آیا زبان تولید شده توسط یک گرامر مستقل از متن، شامل کلماتی با طول کمتر از n هست؟



پاسخ تمرین‌های فصل ۹

۱- زبان L_4 ، تصمیم پذیر است.

۲- زبان L_1 ، تصمیم پذیر است.

۳- در جدول زیر، مسائلی که تصمیم پذیر هستند، با یک تیک مشخص شده اند:

متناهی بودن L	$L = \Sigma^*$	$L = \phi$	$w \in L$	
✓	✓	✓	✓	منظم
✓		✓	✓	مستقل از متن
		✓	✓	حساس به متن
			✓	بازگشتی
				بازگشتی شمارش پذیر

به طور نمونه اگر L یک زبان مستقل از متن باشد، مسئله $L = \Sigma^*$ تصمیم پذیر نیست، چون برای تست این موضوع باید عضویت همه رشته های عضو Σ^* در L را بررسی کرد که چون تعداد آنها متناهی نمی باشد، این عمل شدنی نیست.

۴- دو گزاره معادل اند.

۵- برشمارنده یک ماشین تورینگ با یک چاپگر است. یک برشمارنده با یک نوار خالی شروع می کند. زبان برشمرده شده، مجموعه تمامی رشته هایی است که در نهایت چاپ می شود. البته ممکن است بعضی رشته ها را تکراری چاپ کند. اگر برشمارنده متوقف نشود، ممکن است لیستی شامل بی نهایت رشته چاپ کند.

۶- هر دو گزاره درست است.

۷- گزینه ۴ نادرست است. درست آن به صورت زیر است:

اگر $A \leq_m B$ بوده و A تشخیص پذیر تورینگ نباشد، آنگاه B تشخیص پذیر تورینگ نخواهد بود.

۸- برای هر دو گزاره الگوریتم وجود دارد.

۹- برای هر سه گزاره الگوریتم وجود دارد. در رابطه با هر گزاره باید بدانیم که:

الف- تابع $tail(L)$ به صورت $tail(L) = \{v : uv \in L, u, v \in \Sigma^*\}$ تعریف می‌شود.

ب- یک زبان در صورتی دو طرفه نامیده می‌شود که $L = L^R$

ج- L حاوی رشته w ای است که \widehat{w} زیر رشته ای از آن باشد، یعنی: $w = u\widehat{w}v$. $(u, v \in \Sigma^*)$

۱۰- برای هر سه گزاره الگوریتم وجود دارد.

۱۱- زبان $L_1 \cap L_2$ را تشکیل می‌دهیم و گرامر مستقل از متن G را به آن نسبت می‌دهیم. حال تعیین می‌کنیم که آیا این زبان تهی است یا خیر.

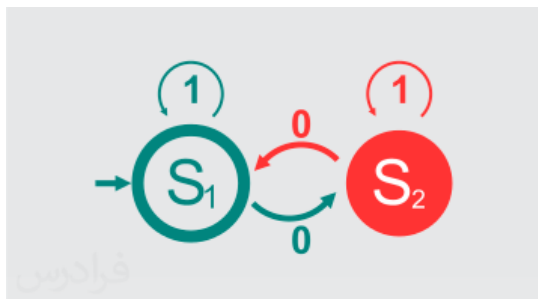
۱۲- گرامر را به فرم نرمال چامسکی در می‌آوریم. عضویت تمام رشته‌هایی با طول کمتر از n را با الگوریتم CYK، بررسی می‌کنیم.

دسته بندی موضوعی آموزش های فرادرس، در ادامه آمده است:

 <p>مهندسی برق الکترونیک و رباتیک</p>	 <p>هوش مصنوعی</p>	 <p>آموزش‌های دانشگاهی و تخصصی</p>	 <p>برنامه‌نویسی</p>
<p>مهندسی برق الکترونیک و رباتیک - کلیک (+)</p>	<p>هوش مصنوعی - کلیک (+)</p>	<p>آموزش‌های دانشگاهی و تخصصی - کلیک (+)</p>	<p>برنامه‌نویسی - کلیک (+)</p>
 <p>مهارت‌های دانشگاهی</p>	 <p>مباحث مشترک</p>	 <p>دروس دانشگاهی</p>	 <p>مهندسی نرم‌افزار</p>
<p>مهارت‌های دانشگاهی - کلیک (+)</p>	<p>مباحث مشترک - کلیک (+)</p>	<p>دروس دانشگاهی - کلیک (+)</p>	<p>مهندسی نرم‌افزار - کلیک (+)</p>
 <p>آموزش‌های عمومی</p>	 <p>طراحی و توسعه وب</p>	 <p>نرم‌افزارهای عمومی</p>	 <p>نرم‌افزارهای تخصصی</p>
<p>آموزش‌های عمومی - کلیک (+)</p>	<p>طراحی و توسعه وب - کلیک (+)</p>	<p>نرم‌افزارهای عمومی - کلیک (+)</p>	<p>نرم‌افزارهای تخصصی - کلیک (+)</p>

منبع مطالعاتی تکمیلی مرتبط با این کتاب

آموزش ویدئویی نظریه زبان‌ها و ماشین‌ها



در این درس با سه موضوع "زبان، گرامر و ماشین" آشنا می‌شوید. این درس پیش‌نیاز درس طراحی کامپایلر است. با یادگیری زبان‌ها و گرامرها می‌توانید نحوه کار کامپایلر و همچنین طراحی زبان‌های برنامه‌سازی را متوجه شوید. یادگیری این درس بدون مدرس کار ساده‌ای نمی‌باشد و ما در این آموزش تجربه حداقل پانزده سال تدریس این درس را در اختیار شما گذاشته‌ایم. به امید اینکه دعای خیری برای ما شود.

مدرس: مهندس فرشید شیرافکن
مدت زمان: ۹ ساعت

faradars.org/fvsft110

جهت مشاهده آموزش ویدئویی این آموزش [کلیک](#) کنید.