مقدمه

درس نظریه زبانها را می‌توان یکی از مفهومی ترین دروس رشته مهندسی نرم افزار در نظـر گرفـت. تـست‌هـای ایـن درس را بـه ندرت می‌توان با روشها و فرمول‌های تستی حل نمود و برای بالا بردن توانایی‌هـای خـود در تـست زنـی، تنهـا راه درک مفـاهیم و تمرین مستمر است. با بالا بردن توانایی‌های خود، در بسیاری از سوالات خواهید دید نیازی به حل تشریحی و کامل سـوال ندارید. البته همانطور که گفته شد، این به معنی یک فرمول تستی نمی باشد. ویژگی منحـصر بفـرد دیگـر ایـن درس، در ایـن اسـت کـه مطالب درس بطور بسیار زیادی وابسته به هم بوده و در بسیاری از تست‌ها شما نیاز به اطلاع از تمامی مباحـث درس داریـد. درس با معرفی کلی گرامرها، زبان‌ها، ماشین‌ها و ارتباط کلی آنها با یکدیگر شروع می‌شود. در فصول بعدی ابتدا زبان‌های منظم بررسـی شده و بصورت کامل با آنها آشنا می‌شوید. سپس زبانهای مستقل از متن مورد بررسـی قـرار مـی‌گیرنـد. در نهایـت زبانهـای نـوع یک(وابسته به متن)، نوع صفر(زبانهای بدون محدودیت و ماشینهای تورینگ) و انواع آنها بررسی خواهنـد شـد. بـرای موفقیـت در تست‌های این درس ترتیب مشخص شده را تقریبا باید حفظ کنید؛ زیرا همانطور که بحث شد، مقایسه ویژگی‌های این زبان‌هـا بـر روی یک شاخص خاص از تست‌های معمول در این درس است.

در این جزوه سعی شده در کنار مطالب مفهومی که داوطلب تا این لحظه مطالعه نموده است، قدرت حل تمـرین و تـست زنـی وی افزایش داده شود. بررسی مرحله به مرحله روند حل تمرین ها و نیز حل تشریحی اکثر تست‌های مهم از آزمونهای سـالهای اخیـر، کمک شایانی به شما داوطلب عزیز در حل تست‌های آزمون سراسری خواهد کرد. همچنین در کنـار ایـن مطالـب، برخـی نکـات کنکوری، بدون اثبات آنها، نیز در این جزوه قرار داده شده تا راهنمای شما در درک مسائل سطح بالاتر باشد. پیـشنهاد مـی‌شـود که مسائل اینچنین را تحلیل کرده و سعی نمایید که اثبات این مطالب را درک نمایید. هرچند نیاز آنچنانی به این اثبـات نخواهیـد داشت ولی این مسئله گستره دید شما را در این درس به خوبی افزایش می‌دهد.

امید است جزوه حاضر در کنار مطالب مطالعه شده قبلی، سطح علمی و قدرت تست زنی داوطلبین را افزایش دهد.

به یاد داشته باشید: موفقیت از آن شماست اگر بخواهید.

با آرزوی موفقیت و پیروزی

حمید حاج سید جوادی

حمید طاهرپور

در جدول ذیل دروس به سرفصلهای مهم آن طبقه بندی شده و مشخص شده است که در هر سال از هر مبحث چند تست سوال شده است و دانشجوی محترم می تواند زمان باقیمانده تا کنکور را با توجه به اهمیت مباحث مدیریت نماید.

رشته: مهندسی کامپیوتر		درس: نظریه زبان‌ها و ماشین‌ها					
ردیف	مبحث	۱۳۸۵	۱۳۸۶	۱۳۸۷	۱۳۸۸	۱۳۸۹	مجموع ۵ سال	نسبت از کل
		تعداد تست	تعداد تست	تعداد تست	تعداد تست	تعداد تست		
1	مفاهیم اولیه زبان ها و ماشین ها، برابری زبان و گرامر	0	2	2	1	0	5	17%
2	ماشین های متناهی (DFA,NFA)	1	0	2	1	1	5	17%
3	زبانها، عبارات و گرامرهای منظم و خطی	1	0	0	1	0	2	7%
4	خصوصیات زبانهای منظم	2	0	1	0	1	4	13%
5	زبان ها و گرامرهای مستقل از متن	0	1	0	0	1	2	7%
6	ساده سازی گرامرهای مستقل از متن	0	0	0	0	1	1	3%
7	ماشین پشته ای	0	1	0	1	0	2	7%
8	خصوصیات زبانهای مستقل از متن	0	1	0	0	0	1	3%
9	ماشین تورینگ، انواع مدلها و ویژگیهای آنها	1	1	0	1	0	3	10%
10	زبانهای نوع صفر و یک	0	0	1	0	0	1	3%
11	ویژگیهای زبانهای نوع صفر و یک و محدودیتهای محاسباتی	1	0	0	1	2	4	13%
جمع		**6**	**6**	**6**	**6**	**6**	**30**	**100%**

در درس نظریه زبانها و ماشینها هدف اصلی تقسیم بندی انواع مسائلی است که در دنیای پیرامون ما وجود دارند. ما ابتدا برای هر مسئله، یک تعریف رسمی ارائه داده و این تعریف را به عنوان یک زبان در نظر می‌گیریم. برای هر زبان نیز با سه مفهوم اساسی سروکار داریم. گرامر زبان، ماشین پذیرنده زبان و طریقه نمایش رشته های زبان. در نهایت چهار گروه اصلی را معرفی کرده و تمامی زبانها(مسائل پیرامون) را در یکی از این چهار گروه دسته بندی خواهیم کرد. برای هر کدام از این گروه ها باید مفاهیم معرفی شده فوق را تعریف و ارائه دهیم.

این چهار دسته عبارتند از:

۱. **زبانهای منظم– نوع سه(گرامر های منظم، ماشین متنهای، عبارت های منظم)**

این زبانها در حل مسائلی کاربرد دارند که برای حل آنها نیازی به حافظه نامتناهی نداشته باشیم و تنها ابزار ما در تشخیص زبان، یک حافظه متناهی است که در قالب یکسری حالت(با تعداد مشخص و متناهی) نمایش داده میشود.

۲. **زبانهای مستقل از متن– نوع دو (گرامر مستقل از متن، ماشین پشته ای)**

با استفاده از این زبانها، ما قدرت بیشتری را خواهیم داشت و مسائلی را میتوانیم تعریف و حل کنیم که در آنها احتیاج به یک حافظه نامتناهی از نوع پشته باشد.

۳. **زبانهای حساس به متن– نوع یک(گرامر حساس به متن، ماشین خطی کراندار)**

اگر در حل مسئله ای، احتیاج به یک حافظه نا متنهاهی و غیر پشته ای باشد، به نحوی که میزان حافظه مورد نیاز را با توجه به اندازه ورودی، بتوان از ابتدای کار محدود نمود، آنگاه می توان مسئله را با یک گرامر حساس به متن تعریف کرد. برای پذیرش یک رشته در زبان نیز می توان از یک ماشین خطی کراندار استفاده نمود.

۴. **زبانهای بی قید و شرط– نوع صفر(گرامر بی قید و شرط، ماشین تورینگ)**

خواهیم دید که با این ابزار هر مسئله قابل حلی را میتوان تعریف و شناسایی نمود. پس برای هر مسئه ای در دنیای پیرامون، اگر الگوریتمی وجود داشته باشد، میتوان یک ماشین تورینگ متناظر با آنرا ساخت. هر چند با تعاریف مقدماتی که ما در این دروس مشاهده کرده ایم، ممکن است این ادعا غیر قابل باور بنظر بیاید، ولی امکان پذیر و قابل اجراست.

معرفی زبانهای منظم

برای تعریف زبانهای منظم، گرامر های منظم را داریم و برای بررسی عضویت یک رشته در اینچنین زبانی، از ماشینهای حالت متناهی و برای نمایش زبان از ابزار عبارت منظم استفاده می‌کنیم. همانطور که می‌دانید، این سه ابزار معادل یکدیگرند و وجود هر یک شرط لازم و کافی برای منظم بودن یک زبان می باشد. این سه ابزار قابل تبدیل به هم بود و برای اینکار الگوریتم داریم. پس با داشتن هر یک، دو ابزار دیگر را می‌توانیم بسازیم.

یادداشت:

....
....
....
....

-زبان a^nb^n یک زبان منظم نیست. پس قاعدتا نمی‌توانیم برای آن یک گرامر منظم بسازیم. به دلیل اینکه گرامر های منظم خطی از راست و یا خطی از چپ هستند، پس قاعدتا نمی‌توانند از میانه رشته شد داشته باشند. پس واضح است تعداد یکسان الفا را در دو سمت رشته نمی‌توانند بسازند.

$A \rightarrow aA$

$A \rightarrow Aa$

$\rightarrow$ $A \rightarrow aAb$ غ ق ق

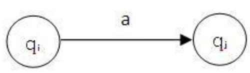
از دیدگاه دیگر، در یک ماشین با تعداد حالات متناهی و بدون استفاده از هیچ حافظه ای، نمی‌توان تعداد برابر a و b را در دو طرف رشته، در حالت کلی و طول بینهایت، بررسی و تضمین نمود. شما می‌توانید در یک ماشین با بکارگیری تعداد حالات زیادی، برابری a وb را تا یک تعداد معین بررسی کنید ولی برای تعداد نامتناهی این کار ممکن نیست. پس همانطور که انتظار داشتیم، برای این زبان نمی‌توان ماشین حالت متناهی نوشت.

از طرفی، طبق تعریف عبارات منظم، می‌دانیم که با تعریف یکسری عبارات پایه‌ای زبان و با استفاده از عملگرهایی مانند اجتماع (عمل جمع)، اشتراک، الحاق، بستارهای ستاره و جمع، می‌توان عبارات منظم دیگر را ساخت. پس می‌بینید زبان a^nb^n از این قانون پیروی نمی‌کند. پس باز هم مطابق انتظار، نمی‌توان برای این زبان عبارت منظم داشت.

طبق مطالب فوق، می‌بینیم که در واقع قدرت این سه ابزار، در زبان‌های منظم برابر بوده و به همین دلیل هم معادل همدیگرند. در زبانهای منظم، معمولا مستقیما با گرامر های منظم سروکار نداریم. در عوض تست های فراوانی بر روی عبارات منظم و یا ماشینهای حالت متناهی داشته ایم.

ماشین حالت متناهی

همانطور که گفته شد، ماشین ها یکی از سه ابزار اصلی ما در این درس هستند. از این ابزار برای بررسی پذیرش یک رشته در زبان استفاده می‌شود. اگر این ماشین را معادل یک مسئله در دنیای خارج فرض کنیم، آنگاه ماشین بررسی می‌کند که (رشته) ورودی ما یکی از پاسخ-های مسئله می‌باشد یا خیر؟ ماشین حالت متناهی از یکسری حالت با تعداد متناهی تشکیل شده است که بین این حالت ها یکسری یال جهتدار قرار گرفته اند. بر روی هر یک از این یالها، یکی از حروف الفبای مجاز در زبان قرار دارد. به عنوان مثال، شکل زیر به معنی این است که در حالت i ام با دیدن کاراکتر a در رشته ورودی، به حالت jام حرکت می‌کنیم و بر روی رشته نیز یک کاراکتر به جلو حرکت نماییم.



بر اساس الفبایی که بر روی یالها قرار میگیرند، ماشینهای متناهی به دو دسته کلی تقسیم میشوند:

ماشین حالت متناهی قطعی و غیر قطعی

یادداشت:

....

....

....

....

در ماشین های حالت متناهی DFA (Definite Finit-state Machine)، در هر حالت، با دیدن هر کاراکتر ورودی، بصورت قطعی حرکت میکنیم. یعنی در تمام حالات، به ازای هر کاراکتر ورودی فقط یک یال خروجی وجود دارد. یا به عبارت دیگر در هر حالت، با یک کاراکتر مشخص، دو حرکت مجاز نداریم.

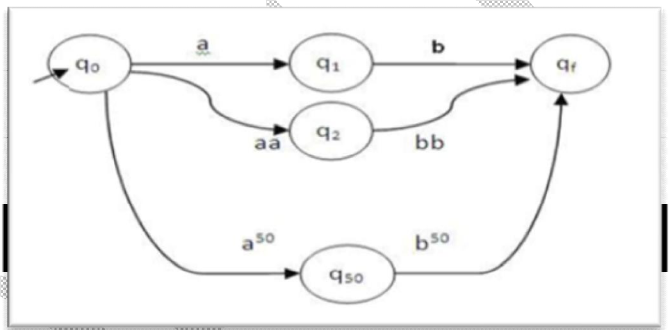
در ماشینهای ماشین حالت متنهای غیر قطعی NFA (Non-definite Finit-state Machine)، محدودیت فوق را نداریم. با هر ورودی، به هر تعداد دلخواه میتوانیم حرکت تعریف کنیم. همانطور که مشخص است، dfa خود یک nfa می باشد و از طرفی برای تبدیل nfa به dfa نیز الگوریتم داریم. پس با این حساب dfa و nfa معادل یکدیگرند.

نکاتی در مورد طراحی ماشین متناهی

همانطور که می‌دانید، هر چند برای منظم بودن یک زبان ما باید برای آن یک ماشین متناهی(تعداد حالات متناهی است) بسازیم؛ ولی الزاما زبان نباید حتما متناهی باشد. آنچه واضح است، یک زبان متناهی با هر شرایطی، اگرتعداد رشته هایش متناهی باشد، منظم است. زیرا با ماشین متناهی می‌توان تمام رشته های آنرا ساخت.

مثال: زبان زیر یک زبان منظم است زیرا برای آن یک ماشین متناهی داریم.

$\{a^n b^n \mid n<50 \}$



همچنین زبان $\{ a^n b^m c^{n+2m} \mid n<100 , m<40 \}$ نیز یک زبان منظم است. زیرا متناهی است.

مدل کردن مسئله برای طراحی ماشین متناهی

همانطور که می‌دانید، زبان‌های منظم بسیاری داریم که تعداد رشته هایشان نامتناهی هستند که برای آنها می‌توان ماشین متناهی رسم نمود. در این گزینه ها dfa را داریم. از طرفی روال تبدیل nfa به dfa روال سخت و زمانبری است. پس بهتر است که از ابتدا سعی نماییم که مستقیما dfa مربوط به زبان را بسازیم. به این منظور ابتدا سعی می‌کنیم که خود را به سرعت از حالت شروع به حالت نهایی برسانیم(با پیمایش کوچکترین حرکات در گرامر یا پیمایش کوچکترین رشته های ممکن در عبارت منظم). سپس سعی می‌کنیم در حالت نهایی بمانیم مگر اینکه به محدودیت های مشخص شده در مسئله برسیم.

یادداشت:

....

....

....

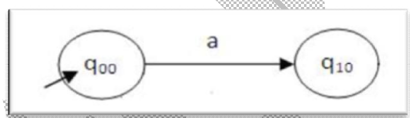
....

برای طراحی سریع dfa یا nfa ، سعی کنید که حالت‌های موجود در ماشین را با یک روال معین و با معنی نامگذاری کنید تا نام هر حالت برای شما معنای مشخصی داشته باشد.این روش نامگذاری حالت، شما را بسیار در رسم سریعتر ماشین کمک میکند. از طرفی با این روش از بسیاری اشتباهات در رسم dfa پیشگیری خواهد شد.

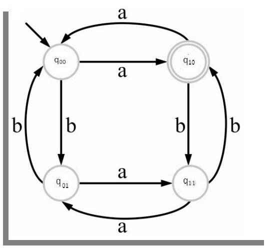
به عنوان مثال فرض کنید هدف ما رسم dfa برای برای شناسایی رشته‌هایی از الفبای {a,b} است که در آنها تعداد a ها فرد و تعداد b ها زوج باشد. در مرحله اول مسئله را در یک قالب محدود مدل کنیم. یعنی باقیمانده تعداد a ها و b های ملاقات شده در رشته پیمایش شده را بر عدد ۲ محاسبه کنیم. باقیمانده ۱ نشانگر فرد بودن تعداد کاراکترها و باقیمانده ۰ نشانگر زوج بودن تعداد کاراکتر دارد. پس با این حساب چهار حالت زیر را خواهیم داشت:

وضعیت قبول	تعداد b	تعداد a
×	فرد	فرد
√	زوج	فرد
×	فرد	زوج
×	زوج	زوج

همانطور که مشاهده می‌شود، از این چهار حالت فقط یکی مورد نظر ما است. ما برای این مسئله یک dfa با همین چهار حالت را در نظر گرفته و آنرا به صورت q_{ij} نامگذاری می‌کنیم. i به معنی باقیمانده تعداد a ها و j نشانگر باقیمانده تعداد b ها بر عدد ۲ است. به عبارت دیگر i=0 به معنی تعداد زوج رویت شده از کاراکتر a و i=1 تعداد فرد از کاراکتر a را نشان می‌دهد. با این تفاصیل، در q_{00} که نشانگر محلی از رشته است که هم تعداد aها و هم تعداد bها زوج است، با دیدن کاراکتر a به q_{10} میرویم که نشانگر تعداد فرد از کاراکتر a و همان تعداد زوج قبلی از کاراکتر b است.



پس با این حساب که حرکت با هر یک از ورودی‌ها اینچنین تاثیری بر تعداد فرد و یا زوج a ها (و یا b ها) خواهد داشت، ماشین متناهی این زبان بصورت زیر خواهد بود:



فرض کنید که هدف طراحی یک dfa باشد برای پذیرش زبانی که در تمام رشته هایش حداقل یک زیر رشته aa داشته باشیم. به این صورت عمل میکنیم که ابتدا حالت شروع را می‌سازیم. چون در ابتدا کار، شرط مورد نظر ما برآورده نشده است(aa هنوز دیده نشده است) پس حالت شروع را حالت نهایی درنظر نمی‌گیریم. این حالت را q_0 نامگذاری کرده، این معنی که تعداد کاراکترهای a در انتهای رشته ای که شروع شده(یا تا این مرحله پیمایش شده است) برابر ۰ کاراکتر بوده است.

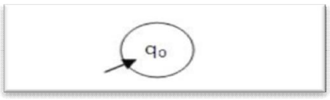
یادداشت:

....

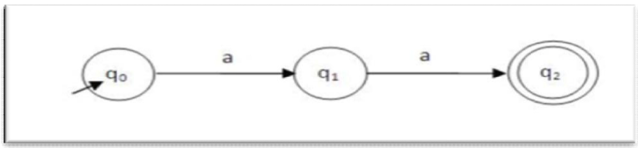
....

....

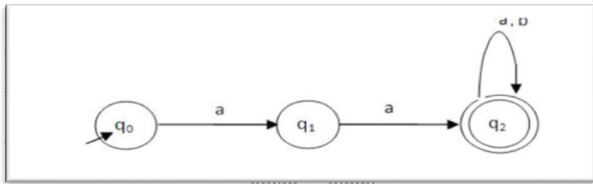
....



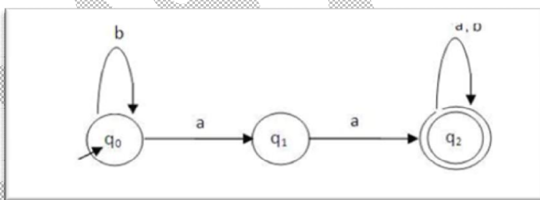
همانطور که گفتیم، در مرحله اول باید به سرعت خود را به حالت نهایی برسانیم. پس ابتدا با aa شروع می‌کنیم. پس در حالت شروع با دیدن اولین کاراکتر a، به حالت q_1 میرویم. اندیس ۱ در اینجا به این معنی است که در انتهای رشته پیمایش شده، ۱ کاراکتر a وجود داشته است. از q_1 با دیدن یک a دیگر، به q_2 میرویم. در این حالت، طبق تعریف، دو کاراکتر a(زیر رشته aa) را دیده ایم. پس این حالت محدودیت مورد نظر ارضا شده است و این حالت را یک حالت نهایی فرض میکنیم.



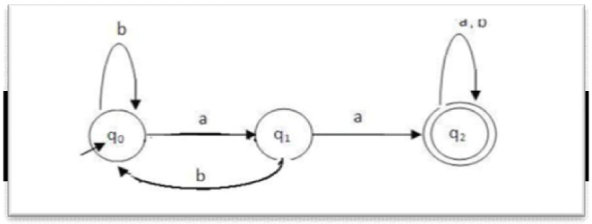
حال که خود را به حالت نهایی رساندیم، باید سعی در ماندن در این حالت کنیم. چون هیچ محدودیت دیگری در مسئله تعریف نشده است، پس نگران این موضوع نیستیم. یعنی در q_2 با دیدن هر ورودی، بر روی همین حالت خواهیم ماند(حلقه بر روی حالت نهایی).



در مرحله بعد، چون هدف رسم dfa بوده، باید حرکتهای مربوط به تمام ورودی ها را در هر حالت مشخص نماییم. از q_0 شروع میکنیم. در این حالت حرکتی برای ورودی b نداریم و باید این حرکت را تعریف کنیم. چون در این حالت با دیدن هر b تعداد a های انتهای رشته پیمایش شده تغییری نمیکند(در واقع همان ۰ باقی میماند) پس در همین حالت خواهیم ماند.



همانطور که گفتیم، حالت q_1 یعنی حالتی که تا این مرحله از کار در انتهای رشته پیمایش شده فقط یک عدد a مشاهده شده است. پس اگر یک کاراکتر b در انتهای رشته اضافه شود، یعنی رشته به b ختم میشود. پس تعداد a های انتهای رشته برابر صفر شده و طبق تعریفی که داشتیم، این موضوع به معنی حرکت به حالت q_0 است. با این تفاصیل dfa نهایی به صورت شکل زیر در می آید:



یادداشت:

....

....

....

....

حال مثال سخت تری را بررسی می‌کنیم. در این مسئله جدید رشته‌هایی را عضو زبان می‌گیریم که حتما زیر رشته aa را شامل شوند و در ضمن تمام رشته ها شامل زیر رشته bb نباشند. برای حل این مسئله، حالت ها را با دو اندیس تعریف می‌کنیم و هر اندیس وظیفه شمارش یکی از کاراکتر ها را دارد.

q_{ij} : حالتی که در انتهای رشته i کاراکتر a و یا j کاراکتر b وجود دارد. ادامه حل این مسئله را در قالب تست سال ۸۹ که در ادامه آمده است، مشاهده می‌کنید.

یادداشت:

....

....

....

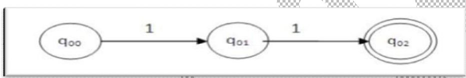
....

تست) L زبانی است با الفبای $\Sigma=\{0 , 1\}$ به قسمی که کلیه رشته های L دارای حداقل یک زیر رشته ۱۱ و فاقد زیر رشته ۰۰ هستند. کوچکترین آتاماتایی که این زبان را شناسایی کند دارای چند وضعیت است؟ (وضعیت شناسایی همان Final State است.)

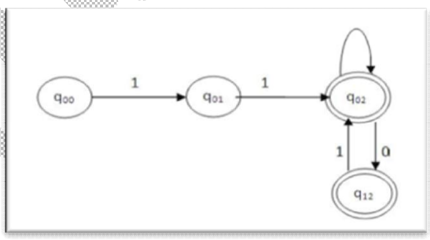
(سراسری ۸۹)

۱) ۵ وضعیت که دو وضعیت آن از نوع شناسایی است
۲) ۶ وضعیت که دو وضعیت آن از نوع شناسایی است
۳) ۵وضعیت که سه وضعیت آن از نوع شناسایی است
۴) ۶وضعیت که سه وضعیت آن از نوع شناسایی است

حل) برای حل این تست، باید دو شرط مختلف را با هم ارضا کنیم. برای بررسی شرط اول، تعداد ۱ های پشت سرهم را میشماریم تا حداقل به یک زیر رشته ۱۱ برسیم. در عین حال باید مراقب باشیم که هیچ زیرر رشته ۰۰ در زبان پذیرفته نشود. طبق روال قبلی، حالات را با دو اندیس، یکی برای شمارش ۰ ها و دیگری برای شمارش ۱ ها نامگذاری می‌کنیم. در گام اول از حالت شروع(q_{00}) باید به سرعت خود را به حالت نهایی برسانیم.(شکل زیر)



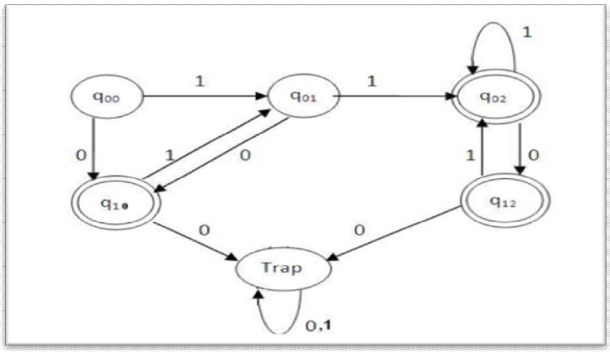
حال باید سعی در باقی ماندن در این وضعیت نماییم. مشخص است که در حالت پایانی بادیدن هر تعداد ۱، هیچ محدودیتی از بین نمی‌رود. پس با مشاهده ورودی ۱، در همین وضعیت می‌مانیم. اما با دیدن ورودی ۰، چون امکان بوجود آمدن زیر رشته ۰۰ پیش می‌آید، پس حالت را تغییر داده و به حالتی می‌رویم که در آن مشخص باشد ما شرط اول را ارضا کرده ایم ولی با اینحال، در انتهای رشته یک کاراکتر ۰ وجود دارد. در این حالت جدید چون شرط دوم نقض نشده است، پس همچنان در یک حالت نهایی هستیم. در این حالت بادیدن کاراکتر ۱، چون در انتهای رشته دیگر ۰ نداریم، پس به حالت قبلی رفته و با دیدن کاراکتر ۰، چون در رشته زیر رشته ۰۰ پدیدار می‌شود، به حالت غیر قابل قبول(Trap) می‌رویم.



مرحله نهایی، تکمیل ماشین با تمامی ورودی ها است. برای حالت شروع و نیز حالت q_{01} بادیدن ورودی صفر، چون هنوز زیر رشته ۱۱ را نداریم و از طرفی در انتهای رشته یک کاراکتر ۰ پدیدار شده است، پس به وضعیت جدیدی می‌رویم تا در صورت مشاهده یک کاراکتر ۰ دیگر در ادامه رشته، متوجه نقض شدن محدودیت مورد نظر شویم و به حالت Trap برویم. در این حالت جدید q_{10}، اگر کاراکتر ۱ را ببینیم، به وضعیت q_{01} می‌رویم. زیرا در انتهای رشته یک کاراکتر ۱ داریم و در صورت مشاهده یک کاراکتر ۱ دیگر باید به وضعیت نهایی برویم. در نهایت شکل زیر ماشین متنهای مورد نظر را نشان می-دهد:

یادداشت:

....
....
....
....



تست) عبارت منظم R و گرامرهای G_1، G_2 و G_3 با تعریف زیرمفروض‌اند. اگر زبان R را L بنامیم و L_1، L_2 و L_3 به ترتیب زبان گرامرهای مذکور باشند. کدام گزاره صحیح است؟

(سراسری ۸۸)

$$R = ((aa|b)^*b)^*a$$

G_1:	G_2:	G_3:
$S \to bS\|aA\|aC$	$S \to bS\|aA\|aC$	$S \to bS\|Aa\|C$
$A \to aS$	$A \to Sa$	$A \to aS$
$C \to \lambda$	$C \to \lambda$	$C \to a$

۱) $L = L_1 = L_2 = L_3$ ۲) $L_1 \neq L_3, L = L_3$

۳) $L_2 \neq L, L = L_1 = L_3$ ۴) $L_3 \neq L_2, L = L_1 = L_2$

حل) در این تست تمام گزینه ها نادرست هستند.

اما روال کلی کار به اینصورت است که از رشته تهی(λ) شروع میکنیم. هر سه گرامر و همچنین عبارت مورد نظر این رشته را تولید نمی‌کنند. پس این رشته کمکی به ما نمی‌کند. با توجه به عبارت داخل پرانتز در عبارت منظم، می‌بینیم که در ابتدای تمام رشته ها، ما همواره aab را داریم و امکان بوجود آمدن ba نیست. با مقایسه گرامر ها، گرامر دوم و سوم این مسئله را نقض کرده و هر دو گرامر ممکن است در رشته هایشان، ab را تولید کنند.

همچنین با بررسی رشته aaa، میبینیم که هر سه گرامر توانایی تولید این عبارت را دارند، در صورتیکه در عبارت منظم داده شده، همواره قبل از آخرین a حداقل یک کاراکتر b وجود دارد.

یادداشت:

....

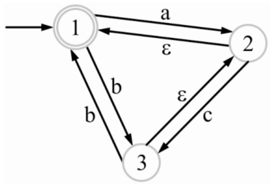
....

....

....

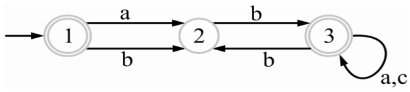
تست) **ماشین متناهی M به شکل زیر مفروض است. گزاره صحیح کدام است؟**

(سراسری ۸۷)



۱) $L(M) = \left(a^* \mid (b \mid ac)^* (b \mid \varepsilon)\right)^*$

۲) ماشین قطعی زیر معادل M است:



۳) $L(M) = \left\{ w \in (a \mid b \mid c)^* \mid \text{با } c \text{ شروع نمی‌شود } w \right\}$

۴) زبان گرامر مقابل همان $L(M)$ است:

$$S \to aS \mid bS \mid acS \mid bA \mid acA$$
$$A \to cA \mid b \mid \varepsilon$$

حل) گزینه ۳ درست است.

ازآنجاکه در ماشین M فقط این محدودیت وجود دارد که حرف c نمی‌تواند شروع‌کننده رشته باشد و در غیر از این مورد هر سه حرف a، b یا c می‌توانند به هر ترتیبی واقع شوند، بنابراین گزینه ۱ و ۲ نادرست هستند زیرا در گزینه ۱ حرف c فقط باید بعد از a بیاید و در ماشین گزینه ۲ نیز محدودیت‌های دیگری نیز برقرار است. در گرامر گزینه ۴ نیز حرف a نمی‌تواند پایان‌دهنده رشته باشد؛ بنابراین گزینه ۳ درست است.

تست) اتومات متناهی M و زبان L_1 تا L_4 مفروض‌اند. رابطه $L(M)$ با L_1 تا L_4 کدام است؟

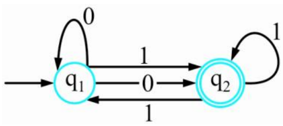
(سراسری ۸۷)

$$L_1 = (0+1)(0+1)^*$$

$$L_2 = \left(0+(0+1)1^*1\right)^*(0+1)1^*$$

$$L_3 = 0^*(0+1)1^*\left(10^*+(0+1)1^*\right)^*$$

$$L_4 = (0+110)(0+1)^*$$



۱) $L(M) = L_2 = L_3 = L_4$

۲) $L(M) = L_1 = L_2 = L_3$

۳) $L(M) = L_2 = L_3$

۴) $L(M) = L_4$

یادداشت:

....

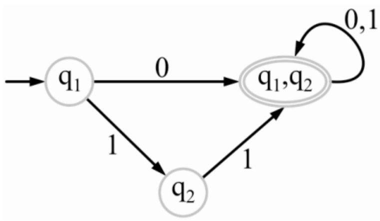
....

....

....

حل) **گزینه ۳ درست است.**

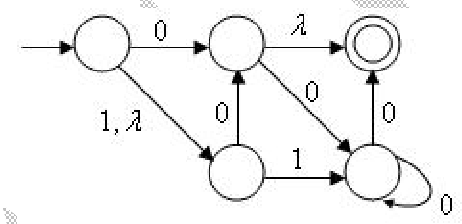
با تبدیل ماشین مورد نظر به یک ماشین قطعی به ماشین زیر خواهیم رسید:



مشاهده می‌شود که ماشین مورد نظر فقط رشته‌هایی را که با 10 شروع می‌شوند نمی‌پذیرد. ازآنجاکه زبان L_1 با هر حرفی می‌تواند شروع شود، گزینه ۲ نادرست است. زبان L_4 نمی‌تواند شامل رشته‌ای باشد که با 111 شروع می‌شود؛ بنابراین گزینه‌های ۱ و ۴ نیز نادرست هستند. با کمی بررسی می‌توان دریافت که گزینه ۳ درست است.

تست) عبارت منظم متناظر با زبان اتوماتون زیرکدام است؟

(علوم کامپیوتر ۸۴)



۱) $0+(00+1+11)0^*$

۲) $0+(00+1+11)^*0^*$

۳) $(00+1+11)0^*$

۴) د) $0+(00+1+11)00^*$

حل) گزینه ۴ درست است.

رشته "۱ " در این ماشین پذیرفته نمی‌شود. پس سه گزینه اول غلط هستند.

متمم گیری از ماشین متناهی

برای متمم کردن یک زبان منظم فقط می‌توان از DFA استفاده کرد. برای این کار حالت نهایی را به غیرنهایی تبدیل می‌کنیم و بلعکس(حتی حالت Trap). دقت کنید که برای متمم کردن زبان منظم از NFA نمی‌توان استفاده کرد.

دلیل: شکل زیر یک nfa برای زبان a*ba* است.

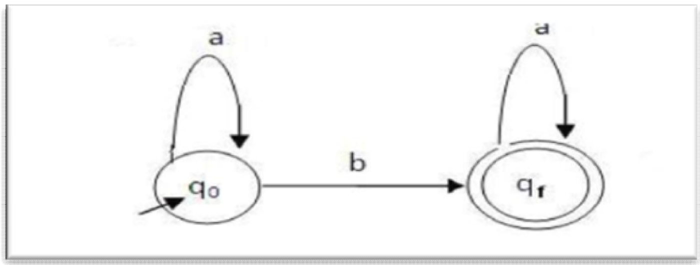
یادداشت:

....

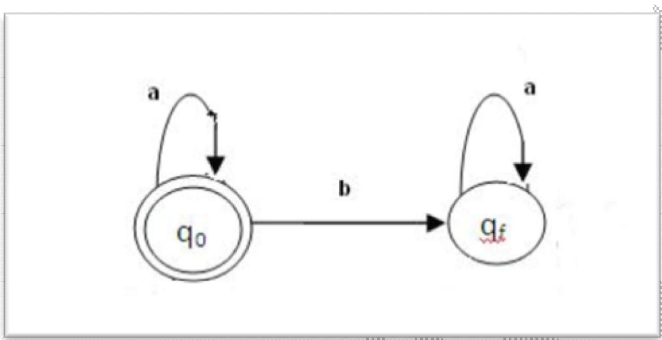
....

....

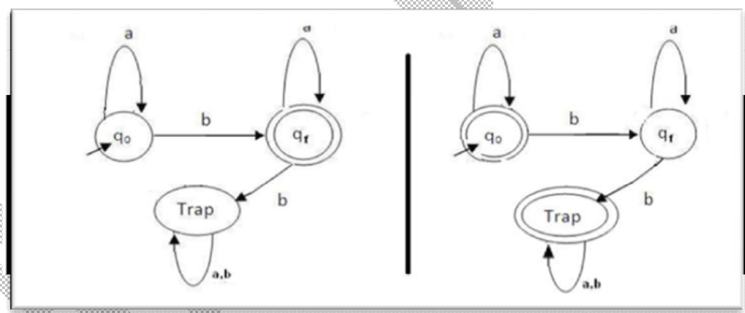
....



طبق روال متمم گیری به شکل زیر خواهیم رسید.



کاملا مشخص است که این ماشین، مکمل زبان فوق نیست. زیرا به عنوان مثال عبارت bb را نمیپذیرد. دلیل این مساله در این است که حالت Trap را مشخص نکرده ایم. به عبارت دیگر در nfa بالا، به ازای تمام ورودی ها، دستور انتقال مشخصی را نداریم. در شکل زیر dfa مربوط به زبان و همچنین مکمل آنرا می‌بینید.



تست) **اگر $M=(Q,Q_0,\Sigma,F,\delta)$ یک اتومات متناهی باشد تعریف می‌کنیم: $\overline{M}=(Q,Q_0,\Sigma,Q-F,\delta)$ همچنین $d(M)$ اتومات قطعی معادل M خواهد بود. اگر M_1 و M_2 دو اتومات متناهی باشند M_1+M_2 اتومات متناهی است که زبان آن اجتماع زبان‌های M_1 و M_2 است. فرض کنید G_1 و G_2 دو گرامر منظم باشند که زبان آن‌ها به ترتیب معادل زبان‌های M_1 و M_2 هستند. کدام عبارت زیر درست است؟**

(سراسری ۸۸)

۱) $L(G_1)-L(G_2)=L(\overline{\overline{M_1}+M_2})$

۲) $L(G_1)-L(G_2)=L(\overline{d(\overline{d(M_1)}+M_2)})$

۳) $L(G_1)-L(G_2)=L(\overline{(d(\overline{M_1})+d(M_2)})$

۴) $L(G_1)-L(G_2)=L((\overline{\overline{d(M_1)}+\overline{d(M_2)}}))$

یادداشت:

....

....

....

....

حل) گزینه ۲ درست است.

نکته‌ی این سوال در این است که برای متمم کردن یک زبان منظم فقط از DFA می‌توان استفاده کرد و از NFA نمی‌توان برای این کار استفاده کرد. بنابراین در این سوال هر جا که بخواهیم متمم زبانی را داشته باشیم باید از d(M) استفاده کنیم.

بنابراین داریم:

$$L(G_1)-L(G_2)=L(G_1)\cap\overline{L(G_2)}=\overline{\overline{L(G_1)}\cup L(G_2)}=\overline{\overline{d(M_1)}\cup M_2}=\overline{d(\overline{d(M_1)}\cup M_2)}$$

بنابراین گزینه ۲ درست می‌باشد.

تست) **اگر** $|w|_i$ **تعداد حرف** i **در کلمه** w **باشد، آن‌گاه کدام گزاره در مورد زبـان** $L=\left\{w\in\{0,1\}^*\,\big|\,|w|_0=f\left(|w|_1\right)\right\}$ **کـه در آن** $f(n)=n \bmod 5$ **است، صحیح‌تر است؟**

(علوم کامپیوتر ۸۵)

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

۱) مستقل از متن است

۲) متناهی است.

۳)) مستقل از متن نیست

۴) وابسته به متن است ولی مستقل از متن نیست

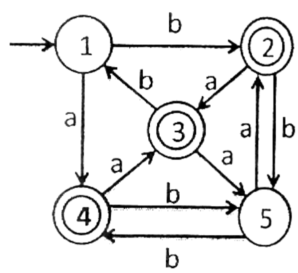
حل) **گزینه ۱ درست است.**

در زبان مورد نظر باید تعداد 0 های موجود در رشته برابر با باقی‌مانده تعداد 1های رشته بر عدد 5 باشد. چنین زبانی را می‌توان با یک ماشین DFA با 5 حالت پذیرفت، پس زبان مورد نظر منظم است و گزینه ۱ درست است. ازآنجاکه تعداد رشته‌های زبان مورد نظر نامتناهی است گزینه ۲ نادرست است. نیز چون زبان مورد نظر منظم است مستقل از متن نیز است؛ بنابراین گزینه‌های ۳ و ۴ نادرست هستند.

بدست آوردن آتاماتای کمینه

تست) **اتومات متناهی زیر را در نظر می‌گیریم. اتومات کمینه مربوطه دارای چند حالت خواهد بود؟**

(سراسری ۸۵)



یادداشت:

....

....

....

....

۱) ۳   ۲) ۲

۳) ۵   ۴) ۴

حل) گزینه ۱ درست است.

برای بدست آوردن آتاماتای بهینه یک DFA بیشتر از روش ادغام‌پذیری و روال کاهش استفاده می‌شود ولی یک روش مشابه با این روش K هم ارزی می‌باشد در این روش منظور از K هم ارزی این است که آیا دو حالت با تمام k نماد الفبا به یک حالت می‌رود یا خیر اگه به یک حالت برود می‌گوییم این دو حالت با هم k هم ارز است.

K=0 صفر هم ارزی $\{\{q_1,q_2,q_3\},\{q_4,q_5\}\} \leftarrow$ حالت نهایی و غیر نهایی

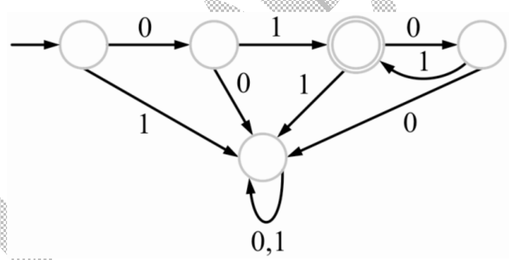
K=1 یک هم ارزی $\{\{q_1\},\{q_2,q_3\},\{q_4,q_5\}\} \leftarrow$ هم‌ارزی با تمام رشته‌هایی به طول یک

K = 2 دو هم ارزی $\{\underbrace{\{q_1\}}_{1},\underbrace{\{q_2,q_3\}}_{2},\underbrace{\{q_4,q_5\}}_{3}\} \leftarrow$ هم ارزی با تمام رشته‌هایی به طول دو

از آنجایی که دو مرحله متوالی هم ارزی با هم برابر شدند دیگر احتیاج به ادامه k هم ارزی نیست و در آخرین مجموعه ۳ مجموعه حالت داریم بنابراین آتاماتای بهینه سه حالت دارد.

تست) تعداد حالات اتوماتون مینیمال برای اتوماتون زیر برابر است با:

(علوم کامپیوتر ۸۴)

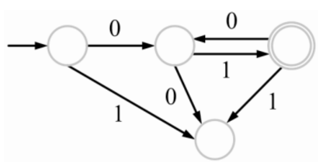


۱) ۲   ۲) ۳

۳) ۴   ۴) ۵

حل) گزینه ۳ درست است.

ماشین مینیمال برای ماشین مورد نظر به صورت زیر است:



بنابراین گزینه ۳ درست است.

یادداشت:

....

....

....

....

تست) گر $\Sigma = \{0,1,...,n\}$ و $L = \{w \mid \exists k \geq 0, |w| = 2k\} \subseteq \Sigma^*$ یک زبان با حروف Σ باشد، آن‌گاه تعداد حالات اتوماتون قطعی متناهی (DFA) مینیمال متناظر با زبان L ....است.

(علوم کامپیوتر ۸۴)

۱) n

۲) 2n-1

۳) 2n

۴) عددی ثابت و مستقل از n

حل) گزینه ۴ درست است.
زبان مورد نظر رشته‌هایی را نشان می‌دهد که دارای طول زوج هستند؛ بنابراین ماشین متناهی مینیمال آن دارای دو حالت باید باشد که بتواند رشته‌های با طول زوج و فرد را از هم متمایز کند و گزینه ۴ درست است.

عبارات منظم

اگر Σ الفبای داده شده باشد آنوقت

۱. $\varnothing$ و λ و $a \in \Sigma$ عبارات منظم هستند و به آنها عبارات منظم ابتدایی می‌گوییم.

۲. اگر r_1 و r_2 عبارات منظم باشند $r_1 + r_2$, $r_1.r_2$, r_1^* و (r_1) نیز عبارت منظم می‌باشند.

۳. رشته‌ای عبارت منظم است اگر و فقط اگر بتوان آن را از عبارات منظم ابتدائی با اعمال دفعات متناهی از قانون ۲ تولید کرد.

نکات مربوط به عبارات منظم:

نکته: عبارت منظم $\varnothing^*$ برابر λ است.

نکته: به ازای هر زبان منظم می‌توان تعداد نامتناهی عبارت منظم در نظر گرفت ولی برعکس آن درست نیست.

نکته: اگر r_1 و r_2 دو عبارت منظم باشند در این صورت روابط زیر برقرار است:

$$(r_1^*)^* = r_1^*$$
$$r_1^*(r_1 + r_2)^* = (r_1 + r_2)^*$$
$$(r_1 + r_2)^* = (r_1^* r_2^*)^*$$
$$(r_1 + r_2)^* = (r_1^* + r_2^*)$$

یادداشت:

....

....

....

....

تست) اگر زبان L بوسیله عبارت منظم $ba(b^*(ab)^* + a^*(bb)^*)ba$ توصیف شده باشد، وارون L (یعنی L^R) به وسیله کدام یک از عبارات منظم زیر بیان می‌گردد؟

(علوم کامپیوتر ۸۳)

۱) $ab((ab)^*b^* + (bb)^*a^*)ab$

۲) $ab(a^*(ba)^* + b^* + (aa)^*)ab$

۳) $ab((ba)^*b^* + (bb)^*a^*)ab$

۴) $ab(b^*(ab)^* + a^*(bb)^*)ab$

حل) گزینه ۲ درست است.

عبارت را از انتها به ابتدا می‌نویسیم در معکوس کردن دقت کنید که اگر بین دو عبارت + باشد تغییری ایجاد نمی‌شود. زیرا + به معنای یا می‌باشد. بنابراین گزینه ۲ درست است.

تست) اگر زبان **L** بوسیله عبارت منظم $ba(b^*(ab)^* + a^*(bb)^*)ba$ توصیف شده باشد، وارون **L** (یعنی $\mathbf{L^R}$) به وسیله کدام یک از عبارات منظم زیر بیان می‌گردد؟

(علوم کامپیوتر ۸۳)

۱) $ab((ab)^*b^* + (bb)^*a^*)ab$

۲) $ab(a^*(ba)^* + b^* + (aa)^*)ab$

۳) $ab((ba)^*b^* + (bb)^*a^*)ab$

۴) $ab(b^*(ab)^* + a^*(bb)^*)ab$

حل) گزینه ۲ درست است.

عبارت را از انتها به ابتدا می‌نویسیم در معکوس کردن دقت کنید که اگر بین دو عبارت + باشد تغییری ایجاد نمی‌شود. زیرا + به معنای یا می‌باشد. بنابراین گزینه ۲ درست است.

تست) **زبان‌های زیر با $\alpha, \gamma \in \Sigma^*$ و $\beta \in \Sigma^+$ مفروضند. کدام یک از گزینه‌های زیر درست است؟**

(سراسری ۸۷)

$L_1 = \{\alpha^i(\alpha\beta)^j(\gamma a)^i \;\; j \geq 0, i \geq 0\}$

$L_2 = \{\alpha^i(\alpha\beta)^j(\gamma a)^{i+j} \;\; j \geq 0, i \geq 1\}$

$L_3 = \{\alpha^i(\alpha\beta)^j(\gamma a)^i \;\; j \geq 1, i \geq 1\}$

۱) L_3 و L_1 هر دو منظم هستند

۲) L_1 منظم و L_3 نامنظم است

۳) L_1 منظم و L_2 نامنظم است

۴) L_3, L_2, L_1 همگی نامنظم هستند.

یادداشت:

....

....

....

....

حل) گزینه ۱ درست است.

در این سوال باید دقت کرد که اینکه α و γ و β چه مقادیری می‌توانند داشته باشند. با در نظر گرفتن این مقادیر می‌توان گفت زبان L_1 منظم است زیرا می‌توان فرض کرد $i=0$ و $j=1$ و $\beta=\lambda$ در این صورت این زبان فقط شامل α می‌شود که این عبارت به علت اینکه $\alpha \in \Sigma^*$ است شامل تمامی رشته‌هایی تولیدی بوسیله الفبا می‌شود که بزرگ‌ترین مجموعه رشته برای L_1 می‌باشد بنابراین می‌توان گفت $L_1=\Sigma^*$ بنابراین این زبان منظم است. در سایر زبان‌ها نیز می‌توان چنین حالت‌هایی را برقرار کرد و ثابت کرد زبان‌های $L_2=L_3=\Sigma^*$ بنابراین تمامی زبان‌ها منظم می‌باشند. بنابراین در گزینه فقط گزینه ۱ می‌تواند درست باشد.

تست) کدام عبارت منظم، زبان زیر را توصیف می‌کند

(علوم کامپیوتر ۸۵)

$$L=\left\{a^m b^{3n} c^{2k} \mid m\ge 1, n\ge 1, k\ge 1\right\}$$

۱) $a^*(bbb)^*(cc)^*$

۲) $a^*b^*b^*b^*c^*c^*$

۳) $aa^*(bbb)^*bbb(cc)^*cc$

۴) $aa^*bbbb^*b^*b^*ccc^*c^*$

گزینه ۳ درست است.

حل) زبان مورد نظر دارای تعداد بزرگ‌تر از صفر a و سپس تعداد بزرگ‌تر از صفر و مضرب سه b و سپس تعداد بزرگ‌تر از صفر و زوج c است و تنها گزینه ۳ مناسب است.

معکوس کردن ماشین متناهی

اگر L منظم باشد آنگاه L^R نیز منظم است زیرا برای L یک DFA وجود دارد حال با معکوس کردن جهت یال‌ها و تبدیل حالات نهایی به شروع و بالعکس می‌توان معکوس زبان را بدست آورد.

برابری زبان و گرامر

تست‌های زیادی وجود دارند که در آنها برابری گرامری زبان مدنظر است. یعنی گرامر خاصی معرفی شده و در گزینه‌ها زبانهای مختلف و یا عبارات متفاوتی آورده می‌شوند. هدف تشخیص اینست که گرامر مربوط به کدام زبان است. برای اینکار می‌توان به ساختار گرامر دقت کرد و سپس با استفاده از آن زبان مورد نظر را تشخیص داد. همچنین بالعکس این موضوع نیز صادق است.

یادداشت:

....

....

....

....

از جمله گرامرها و زبان‌هایی که در تشخیص گرامرها و زبان‌ها به ما کمک می‌کند می‌توان به موارد زیر اشاره کرد:

$$L_1 = \{a^n b^n \mid n \ge 0\}$$
$$G_1 : S \to aSb \mid \lambda$$

این زبان بیان می‌کند که به ازای هر a که از ابتدا می‌آید باید یک b از انتها بیاید.

$$L_2 = \{n_a(w) = n_b(w) \mid w \in \{a,b\}^*\}$$
$$G_2 : S \to aSb \mid bSa \mid SS \mid \lambda$$

این زبان بیان می‌کند که به ازای هر a که یک b باید بیاید این زبان برخلاف زبان قبل هم با a و هم با b باید شروع شود. همچنین برای اینکه هر ترکیب دلخواهی از آن‌ها را داشته باشیم از SS استفاده شده است.

$$L_3 = \{WW^R \mid w \in \{a,b\}^*\}$$
$$G_3 : S \to aSa \mid bSb \mid \lambda$$

این زبان بیان می‌کند که به ازای هر a از ابتدا یک a از انتها و به همین صورت برای bها خواهد بود. بنابراین این زبان را آیینه‌ای می‌گویند.

این سه گرامر و زبان‌های آن‌ها کمک خوبی در تشخیص گرامر و زبان در سوالات دارد برای مثال به سوال زیر دقت کنید.

تست) **با در نظر گرفتن گرامرها و زبان‌های زیر گزینه درست را انتخاب کنید.**

(سراسری ۸۰)

$G_1 : S \to aSb \mid SS \mid \lambda$

$G_2 : S \to aSa \mid bSb \mid \lambda$

$G_3 : S \to Ab, G_1 : A \to aAa \mid b$

$L_1 = \{w \in \{a,b\}^* \mid n_a(w) = n_b(w)\}$

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

$L_3 = \{ww \mid w \in L(a^* b)\}$

۱) $L_3 = L(G_2), L_1 = L(G_1)$

۲) $L_2 = L(G_1), L_3 = L(G_2)$

۳) $L_2 = L(G_3), L_3 = L(G_2)$

۴) $L_3 = L(G_2), L_2 = L(G_3), L_1 = L(G_1)$

حل) گزینه ۳ درست است.

$L_1 \ne L(G_1)$ زیرا گرامر G_1 با b هیچگاه شروع نمی‌شود. بنابراین گزینه ۱ و ۴ نادرست است.

$L_3 = L(G_2)$ زیرا در گرامر خاصیت آیینه‌ای دیده می‌شود.

$L_2 \ne L(G_3)$ زیرا به ازای هر a یک a خواهیم داشت همچنین یک b در وسط و یک b در انتها به رشته‌های این زبان اضافه می‌شود.

نکته: گرامرهایی که قوانین آن‌ها به صورت زیر است را گرامرهای بدون قید گویند.

$$u \to v \qquad u, v \in (V+T)^+$$

یادداشت:

....

....

....

....

نکته: گرامرهایی که قوانین آن‌ها به صورت زیر است را گرامرهای حساس به متن گویند.

$$u \to v \quad ,|u| \le |v| \quad u, v \in (V+T)^+$$

نکته: گرامرهایی را که سمت چپ قوانین آنها یک متغیر وجود داشته باشد مستقل از متن می‌گویند.

نکته: گرامرهای مستقل از متنی را که سمت راست قوانین آنها تنها یک متغیر وجود دارد گرامرهای خطی می‌گویند.

نکته: گرامرهایی خطی که متغیر سمت راست قوانین آنها در راست‌ترین مکان (خطی راست) یا چپ‌ترین مکان (خطی چپ) قرار داشته باشد.

نکته: در سؤالات آزمون مربوط به گرامرها می‌توان از روش حذفی استفاده کرد. برای مثال به سوال زیر دقت کنید:

تست) **زبان گرامر G کدام است؟**

(سراسری ۸۶)

$G: S \to aAb \mid bBa \mid bCa$
$A \to aaAb \mid ab$
$B \to bBa \mid a$
$C \to aC \mid bC$

۱) $a^{2k+2}b^{k+1} \cup b^+a^+ \quad k \ge 0$

۲) $a^{2k}b^k \cup (ba)^*a \quad k \ge 1$

۳) $a^{k+1}b^k \cup b^l a^l \quad l \ge 1, k \ge 2$

۴) $a^2a^{2k}b^kb^2 \cup b^l a^{l+1} \quad k \ge 0, l \ge 1$

حل) گزینه ۴ درست است.

برای حل این سوال از تکنیک تستی استفاده می‌کنیم. یکی از تکنیک‌هایی که در این سوالات کاربرد دارد این است که ببینیم کوچکترین رشته‌ای که گرامر مورد نظر تولید می‌کند را پیدا کنیم و سپس با گزینه‌ها چک کنیم.

کوچکترین رشته‌ای که بوسیله این گرامر تولید می‌شود برابر bba می‌باشد. حال گزینه‌ها را چک می‌کنیم.

کوچکترین رشته در گزینه ۱: ba ← نادرست است

کوچکترین رشته در گزینه ۲: a ← نادرست است

کوچکترین رشته در گزینه ۳: ba ← نادرست است

کوچکترین رشته در گزینه ۴: bba ← درست است.

یادداشت:

....
....
....
....

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

(سراسری ۸۷)

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

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

حل) گزینه ۲ درست است.

کوچکترین رشته‌ای که می‌توان بوسیله گرامر بالا تولید کرد به صورت زیر می‌باشد.

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

که با این رشته گزینه ۱ و ۳ نادرست است. با ادامه اشتقاق گرامر می‌توان دید که aaaaa نیز می‌تواند بوسیله این گرامر تولید شود.

تست) **گرامر زیر چه زبانی را تولید می‌نماید؟**

(سراسری ۸۴)

$G: S \to S_1B$
$S_1 \to aS_1b$
$bB \to bbbB$
$as_1b \to aa$
$B \to \lambda$

۱) $L(G) = \{a^{n+2}b^{3n} \mid n \geq 0\}$
۲) $L(G) = \{a^nb^{n+2k} \mid n \geq 2, k \geq 0\}$
۳) $L(G) = \{a^{n+2}b^{n+2k} \mid n \geq 0, k \geq 0\}$
۴) $L(G) = \{a^{n+1}b^{n+k} \mid n \geq 1, k \geq 0\}$

یادداشت:

....

....

....

....

حل) گزینه ۳ درست است.

برای بست آوردن جواب این سوال ابتدا از کوچکترین رشته عضو گرامر و زبان‌ها شروع می‌کنیم.

کوچکترین رشته عضو گرامر به صورت زیر ایجاد می‌شود.

$$S \to S_1B \to aS_1bB \to aaB \to aa$$

حال گزینه‌ها را چک می‌کنیم.

کوچکترین رشته عضو زبان گزینه ۲: a^2b^2 بنابراین این گزینه نادرست است.

کوچکترین رشته عضو زبان گزینه ۴: a^2b بنابراین این گزینه نادرست است.

سپس با ادامه اشتقاق داریم

$$S \to S_1B \to aS_1bB \to aS_1bbbB \to aabbB \to aabb$$

این جمله در زبان گزینه ۱ نمی‌باشد. بنابراین گزینه ۳ درست است.

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

(علوم کامپیوتر ۸۶)

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

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

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

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

حل) گزینه ۲ درست است.

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

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

(سراسری ۸۶)

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

۱) فقط I

۲) فقط II)

۳) فقط I و III

۴) I، II، III و IV

یادداشت:

....

....

....

....

گزینه ۴ درست است.

حل

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

خواص بستاری زبانهای منظم

خواص بستاری به این معنی است که حاصل اجرای عملگرهای متفاوت بر روی زبانهاب مختلف، از چه نوعی خواهد بود. یعنی مثلا اگر عملگر اجتماع را بر روی دو زبان منظم اجرا کنیم، نتیجه از چه نوع و گونه ای از زبانها خواهد بود؟ آیا نتیجه همواره منظم است یا خیر؟ دقت کنید که اگر بخواهیم اثبات کنیم که نتیجه یک عملگر همواره از یک نوع خاص است، باید برای آن یک الگوریتم کلی معرفی کنیم و در صورتی که بخواهیم نشان دهیم که خروجی الزاما از یک نوع خاص نیست، می توانیم با یک مثال نقض این کار را انجام دهیم. به عنوان نمونه، برای مسئله فوق، گوییم زبانهای منظم تحت عملگر اجتماع بسته هستند. یعنی همواره اجتماع هر دو زبان منظم، منظم خواهد بود. همانطور که گفته شد، برای اثبات این مسئله باید الگوریتم ارائه داد. الگوریتم پیشنهادی به این صورت است که چون هر دو زبان ورودی منظم هستند، پس برای آنها ماشین متناهی داریم. از طرفی برای اجتماع دو ماشین متناهی نیز الگوریتم داریم. پس میتوانیم یک ماشین متناهی بسازیم که اجتماع دو زبان را پذیرش کند. حال چون برای اجتماع دو زبان ماشین متناهی داریم و از طرفی گفته شد که معادل هر ماشین متناهی یک زبان منظم داریم، پس حاصل اجتماع هر دو زبان منظم، همواره یک زبان منظم است.

نکته: بطور کلی زبانهای منظم تحت تمامی عملگر های متعارف بسته هستند، مگر اجتماع و اشتراک نامتناهی

برای اثبات اینکه زبانهای منظم تحت عملگر اجتماع نامتناهی بسته نیستند، کافی است که با یک مثال نقض این مسئله را نشان دهیم. زبانهای زیر را در نظر بگیرید:

$$
\begin{array}{lcl}
L0 & = & \lambda \\
L1 & = & \{ab\} \\
L2 & = & \{aabb\} \\
\vdots & \vdots & \vdots
\end{array}
$$

اجتماع بینهایت زبان فوق برابر زبان a^nb^n خواهد بود که می‌دانیم یک زبان غیر منظم(مستقل از متن) است. پس با این مثال نقض، مسئله فوق را رد می‌کنیم.

نکته: زبانهای نامنظم، تحت عملگر متمم گیری بسته هستند.

نکته:

یعنی همواره متمم یک زبان غیر منظم، غیر منظم خواهد بود. دلیل این امر نیز واضح است. با استفاده از برهان خلف، فرض میکنیم که متمم یک زبان نامنظم، منظم باشد. اگر متمم یک زبان منظم باشد، برای آن ماشین متناهی داریم. پس طبق روال متمم گیری از ماشین متناهی، خواهیم توانست که برای خود زبان نیز یک ماشین متناهی تعریف نماییم. پس زبان مورد نظر نیز منظم خواهد بود. که این مسئله نقیض فرض ما خواهد بود.

یادداشت:

....

....

....

....

تست) کدام گزاره نادرست است؟

(علوم کامپیوتر ۸۶)

۱) اشتراک دو زبان منظم روی یک مجموعه الفبای مشخص حتماً منظم است.

۲) هر زبان نامنظم زیرمجموعه یک زبان منظم است.

۳) هر زبان ناتهی حتماً شامل یک زبان ناتهی و منظم است.

۴) اجتماع تعداد دلخواهی از زبان‌های منظم حتماً منظم است.

حل) گزینه ۴ درست است.

می‌دانیم که زبان‌های منظم تحت اجتماع نامتناهی بسته نیستند؛ بنابراین گزینه ۴ درست است. ازآنجاکه زبان‌های منظم تحت اشتراک بسته هستند، جمله ۱ درست است. ازآنجاکه همه زبان‌ها زیرمجموعه $\sum^*$ هستند، جمله ۲ درست است. ازآنجاکه هر زبان ناتهی حداقل دارای یک رشته است و آن رشته به‌تنهایی چون متناهی است پس منظم است، جمله ۳ درست است.

نکته: برای مثال نقض در تست های مربوط به خواص بستاری زبانهای منظم، می‌توان از زبانهای Φ (تهی) ، زبان $\sum^*$ و زبان شامل رشته تهی $L=\{\lambda\}$ را در نظر داشته باشید. دقت کنید که هر سه این زبانها، منظم هستند.

نکته: زبانهای نامنظم، تحت عملگرهای اجتماع، اشتراک، الحاق و بستار ستاره ای بسته نیستند.

برای نشان دادن این موضوع، باید نشان دهیم که حاصل اجرای هر یک از این عملگر ها بر روی دو زبان نامنظم، ممکن است منظم باشد. برای مثال زبانهای $L1=\{a^nb^n \mid n>0\}$ و $L2=\{b^na^n \mid n>0\}$ را درنظر بگیرید. اشتراک این دو زبان، زبان تهی است که یک زبان منظم می‌باشد. پس اشتراک زبانهای نامنظم، الزاما نامنظم نیست و این زبانها تحت این عملگر بسته نیستند.

نکته: گر L1 منظم و L1 U L2 نیز منظم باشد، آیا L2 الزاما منظم است؟

پاسخ خیر است. به عنوان مثال زبان $L1=\sum^*$ و L2 را هر زبانی نا منظمی مانند a^nb^n در نظر بگیرید. حاصل اجتماع این دو زبان نیز برابر خود L1 بوده و منظم است، در حالیکه زبان L2 یک زبان نامنظم بود.

نکته: اگر L1 منظم و $L1 \cap L2$ نیز منظم باشد، آیا L2 الزاما منظم است؟

پاسخ خیر است. به عنوان مثال زبان $L1=\Phi$ و L2 را هر زبانی نا منظمی مانند a^nb^n در نظر بگیرید. حاصل اشتراک این دو زبان نیز برابر خود L1 بوده و منظم است، در حالیکه زبان L2 یک زبان نامنظم بود.

نکته: اگر L1 منظم و L2.L1 نیز منظم باشد، آیا L2 الزاما منظم است؟

یادداشت:

....

....

....

....

پاسخ خیر است. به عنوان مثال زبان L1=Φ و L2 را هر زبانی نا منظمی در نظر بگیرید. حاصل الحاق این دو زبان نیز برابر خود L1 بوده و منظم است، در حالیکه زبان L2 یک زبان نامنظم بود.

خانواده زبان‌های منظم تحت عمل

Head (L)	$\left\{x \mid xy \in L_1, y \in \Sigma^*\right\}$
Shuffle(L_1,L_2)	$L1 = \left\{w_1, w_2, ...\right\}$
Minus(n)	حذف n عنصر از چپ (برای رشته‌هایی با طول کمتر از n بدون تغییر)
nor(L_1,L_2)	$\left\{w : w \notin L_1, w \notin L_2\right\}$
Third	$\text{Third}\left(q_1 q_2 q_3 ... q_n\right) = q_3 q_6 ...$
exchange	$\text{exchange}\left(q_1 q_2 ... q_{n-1} q_n\right) = q_n q_2 ... q_{n-1} q_1$
even(L)	$\text{even}\left(q_1 q_2 q_3 ... q_n\right) = q_2 q_4 ...$
truncate	اگر تابعtruncate، سمت راست‌ترین عنصر رشته را حذف کند و L منظم
Shift(L)	$\text{shift}\left(q_1 q_2 ... q_n\right) = q_n q_1 q_2 ... q_{n-1}$

بسته است.

تصمیم پذیری و مسائل تصمیم پذیر روی زبانهای منظم

اگر برای حل مسئله‌ای الگوریتم وجود داشته باشد یا به عبارت دیگر اگر در حل مسئله‌ای بتوانیم جواب بله یا خیر بگوییم آن مسئله تصمیم‌پذیر است. برای زبانهای منظم، مسئله تعلق رشته به زبان، یک مسئله تصمیم پذیر است. زیرا میتوانیم ماشین متناهی قطعی زبان مورد را رسم کرده و سپس طبق رشته ورودی این ماشین را پیمایش نماییم. در صورتی که در انتهای پیمایش رشته، ماشین در یکی از حالات نهایی و قابل قبول خود باشد، گوییم رشته متعلق به زبان است و در غیر این صورت پاسخ خیر خواهد بود.

مسائل زیر همگی تصمیم‌پذیر هستند:

☑ تعلق یک رشته مانند w به یک زبان منظم تصمیم‌پذیر است.

☑ تهی، متناهی، نامتناهی بودن زبان‌های منظم تصمیم‌پذیر است.

☑ تساوی زبان‌های منظم یک مسئله تصمیم‌پذیر است.

تست) کدام‌یک از زبان‌های زیر نامنظم هستند؟

(سراسری ۸۵)

۱) $\{a^n b^n (a+b)^* \mid n \geq 0\}$

۲) $\{b^* a^n b^n a^* \mid n \geq 0\}$

۳) $\{a^* a^n b^n b^* \mid n \geq 0\}$

۴) هر سه نامنظم هستند

یادداشت:

....

....

....

....

حل) گزینه ۲ درست است.

مشابه این سوال در سوال‌های قبل دیده شد در این سوال نیز باید دقت کنیم که وجود $a^n b^n$ الزاما باعث نامنظم بودن یک زبان نمی‌شود در این سوال گزینه ۱ منظم است زیرا می‌توان فرض کرد n=0 در این صورت زبان این گزینه به $(a+b)^*$ تبدیل می‌شود که بزرگترین مجموعه رشته را دارد و منظم است. گزینه ۳ نیز منظم است زیرا وجود a^* و b^* باعث شده زبان این گزینه به a^*b^* تبدیل شود که یک زبان منظم است. تنها زبانی که منظم نیست زبان گزینه ۲ می‌باشد.

تست) کدام یک از زبان‌های زیر منظم است؟

(سراسری ۸۴)

$L_1 = \{x^n y^n \mid x \in (0+1)^*, y \in (0+1)^*, n \geq)\}$

$L_2 = \{w \in L(A) \mid$ است و در مسیر پذیرش w از چند حالت معین A عبور نمی‌شود. A یک DFA$\}$

$L_3 = \{w \in (0+1)^* \mid$ تعداد **0** ها با **1** ها مساوی و برابر مقدار ثابت $n \geq 0$ باشد.$\}$

۱) L_3, L_1 ۲) L_3, L_2

۳) L_3, L_2, L_1 ۴) هیچکدام منظم نیستند

حل) گزینه ۳ درست است.

زبان L_1 با فرض n=1 و $x = \lambda$ برابر است با $L_1 = y$ و از آنجا که $y \in (0+1)^*$ بنابراین داریم $L_1 = (0+1)^*$ یک زبان منظم است.

در زبان L_2 اگر در پذیرش رشته‌ها از چند حالت در DFA نگذریم باز هم برای آن رشته می‌توان یک DFA طراحی کرد بنابراین این زبان منظم است.

در زبان L_3 از آنجا که تعداد صفرها و یک‌ها برابر یک مقدار ثابت است در تمامی حالات (چه صفرها و یک‌ها تعدادشان برابر باشد یا نباشد) این زبان منظم خواهد بود زیرا در این حالت زبان متناهی است

لم Pumping برای زبان‌های منظم

اگر L یک زبان منظم و نامتناهی باشد یک عدد صحیح مثبت مانند m وجود دارد بطوریکه هر رشته $w \in L$ با شرط $|w| \geq m$ می‌تواند به صورت زیر تجزیه شود.

$$w = xyz$$
$$|xy| \leq m$$
$$|y| \geq 1$$
$$w_i = wy^i z$$

یادداشت:

....

....

....

....

بطوریکه به ازای تمامی $i=0,1,2,\dots$ رشته w_i در L باشد.

به عبارت دیگر رشته‌های به طول کافی را می‌توان به سه قسمت تجزیه کرد بطوریکه تعداد دلخواه تکرار در قسمت وسط، یک رشته دیگر در L را بدهد این عمل تکرار رشته وسط را دمش می‌گوییم اگر برعکس این کار انجام شود این عمل را مکش می‌گوییم.

نکته: لم Pumping برای نشان دادن اینکه برخی زبان‌ها منظم نیستند مورد استفاده قرار می‌گیرد و همواره با برهان خلف انجام می‌شود. از این لم نمی‌توان برای اثبات منظم بودن زبان‌ها استفاده کرد.

نکته: ثابت Pumping در زبان‌های منظم، تعداد حالات DFA می‌باشد.

تست) فرض کنید $\sum$ یک الفبا و L یک زبان بر روی $\sum$ و N مجموعه اعداد طبیعی باشد. L یک زبان قابل تعریف توسط ماشین متناهی نیست اگر:

(سراسری ۷۶)

۱) $(\exists n \in N)\ (\forall x \in L \text{ such that } |x| \geq n)\ (\exists u,v,w \in \Sigma^*) \text{ such that } x=uvw, |uv| \leq n, |v| \geq 1 \text{ and } (\forall i \in N)\ (uv^iw \in L)$

۲) $(\forall n \in N)\ (\exists x \in L \text{ such that } |x| \geq n)\ (\forall u,v,w \in \Sigma^*) \text{ such that } x=uvw, |uv| \leq n, |v| \geq 1 \text{ and } (\exists i \in N)\ (uv^iw \notin L)$

۳) $(\forall n \in N)\ (\exists x \in L \text{ such that } |x| \geq n)\ (\forall u,v,w \in \Sigma^*) \text{ such that } x=uvw, |uv| \leq n, |v| \geq 1 \text{ and } (\exists i \in N)\ (uv^iw \in L)$

۴) $(\exists n \in N)\ (\forall x \in L \text{ such that } |x| \geq n)\ (\exists u,v,w \in \Sigma^*) \text{ such that } x=uvw, |uv| \leq n, |v| \geq 1 \text{ and } (\forall i \in N)\ (uv^iw \notin L)$

حل) طبق تعریف لم Pumping گزینه ۲ صحیح است.

تعریف زبانهای مستقل از متن

اگر در حل مسئله ای به یک حافظه نامتناهی با ویژگی پشته (آخرین عنصر وارد شده همواره اول خارج میشود) احتیاج داشته باشیم، برای تعریف مسئله از یک زبان مستقل از متن استفاده کرده و برای تعیین پذیرش یک رشته در اینگونه زبانها(تعیین اینکه راه حل ارائه شده، پاسخ مسئله است یا خیر؟) از ماشین پشته ای(PushDown Automata-PDA) استفاده میکنیم. ویژگی اصلی این زبانها، با توجه به نوع تعریف گرامر و توانایی های حافظه پشته ای، در این است که می توانند تعداد نامتناهی را، پشت سر هم، شمارش کرده و یا معکوس یک رشته را محاسبه نماید.

با فرض الفبای $\sum=\{a,b\}$ و $n>0$ مستقل از متن بودن زبانهای زیر را بررسی میکنیم:

زبان a^nb^n : از نظر گرامری، رشته های این زبان را با دستور زیر میتوان ساخت:

$S \rightarrow aSb \mid ab$

یادداشت:

....

....

....

....

که این گرامر یک گرامر مستقل از متن است. چون در اولین قانون تولید این گرامر پس از غیر پایانه S، اجازه داده ایم که پایانه دیگر حضور داشته باشد، پس تعداد مساوی a و b در دو طرف غیر پایانه تضمین میشود.(کاری که در گرامرهای منظم نمیتوان انجام داد). از نظر ماشین پشته ای نیز، چون ما یک حافظه نامتناهی از نوع پشته داریم، پس میتوانیم با دیدن هر کاراکتر a در ابتدای رشته، علامتی را داخل پشته اضافه(push) کنیم. سپس با رسیدن به bها در رشته ورودی، ضمن تغییر حالت، شروع به خالی کردن پشته میکنیم. با این حساب، در صورتی که با خالی شدن پشته، رشته ورودی ما نیز به انتها برسد، نشاندهنده اینست که تعداد a ها با تعداد bهایی که پس از آنها آمده اند برابر بوده و در نهایت رشته ورودی به عنوان یک رشته مجاز در زبان، پذیرش میشود.

زبان a^nb^{2n} : از نظر گرامری، رشته های این زبان را با دستور زیر میتوان ساخت:

$S \rightarrow aSbb \mid abb$

در ماشین پشته ای نیز مانند مثال قبلی، باید ترتیب b ها بعد از aها رعایت شود و تعداد b ها نیز دو برابر تعداد aها باشد.

اجازه دهید که در این مثال ساده، نکته بسیار مهمی را تشریح کنیم. دقت کنید که طبق مطالب عنوان شده، پشته توانایی عمل ضرب را ندارد. پس چگونه باید 2 ، یعنی دو برابر بودن تعداد b را تشخیص دهد؟

در اینجا به نکته ای که در زبانهای منظم نیز داشتیم، بر میگردیم. یعنی سعی کنیم که مسئله داده شده را به صورتی مدل کنیم تا با ابزاری که در اختیار داریم، بتوانیم آنرا حل کنیم.

خب این تغییر چه میتواند باشد؟ قرار است که مسئله را طوری تغییر دهیم که تعداد مساوی پشت سر هم داشته باشیم. پس راه حل به اینصورت خواهد بود که a^nb^{2n} را به صورت $a^n(bb)^n$ در نظر بگیریم. یعنی مانند مثال قبلی به تعداد a علامت در پشته قرار داده ولی با مشاهده دو کاراکتر b پشت سر هم، یک علامت را از پشته خارج کنیم. یا اینکه با دیدن هر کاراکتر a دو علامت در پشته قرار دهیم و سپس با دیدن b ها یک علامت را خارج کنیم. در این صورت با خالی شدن همزمان پشته و رسیدن به انتهای رشته، پذیرش با موفقیت انجام میشود.

نکته : پس دقت کنید که پشته در کل قادر به انجام عمیات ضرب نیست و در مثال فوق، ما مسئله را به نحوی تغییر دادیم که پشته توانایی حل مسئله را داشته باشد.

زبان $a^nb^nc^n$: همانطور که مشخص است با قوانینی مجاز در گرامرهای مستقل از متن، نمیتوان برابری سه مقدار مختلف را تضمین کرد. مثلا گرامر زیر را در نظر بگیرید. آیا این گرامر معادل زبان مورد نظر ما است؟

$$S \rightarrow a\,S_1\,b\,S_2\,c$$
$$S_1 \rightarrow a\,S_1\,b \mid ab$$
$$S_2 \rightarrow b\,S_2\,c \mid bc$$

پاسخ سوال فوق، خیر است. این گرامر در واقع زبان $a^nb^nb^{m-1}c^n$ که همان زبان $a^nb^{n+m-1}c^n$ است را میسازد. دقت کنید که با جدا سازی روال تولید از S به S_1 و S_2 در واقع تضمین تعداد برابر aو b وc را از بین می‌برد. قوانین تولید S_1 تعداد برابر a,b و قانون تولید S_2، تعداد برابر b , c را تضمین می‌کنند. با نوشتن چند رشته از این گرامر، دلیل اینکه در عبارت مربوط به این زبان یک واحد از مجموع n وm کم کرده‌ایم را بررسی کنید.

یادداشت:

....

....

....

....

حال همین مسئله را از دید یک ماشین پشته ای بررسی میکنیم. فرض کنید که با مشاهده هر a یک کاراکتر در پشته قرار دهیم. با رسیدن به b ها، باید شروع به خارج کردن علائم از پشته نماییم. اگر در رشته ورودی به کاراکتر c برسیم و همزمان پشته نیز خالی شود، آنگاه برابری a و b را تشخیص داده ایم. اما مشکل از اینجا شروع میشود. یعنی ما با خالی کردن پشته، عملا عدد n را ازدست داده ایم و به هیچ عنوان نمی‌توانیم برابری تعداد c ها را با a و b تشخیص دهیم. پس زبان $a^nb^nc^n$ نمی تواند یک زبان مستقل از متن باشد.

نکته: ایده زیر را در طراحی ماشین پشته ای برای مسئله اخیر در نظر بگیرید.

"با دیدن هر کاراکتر a، دو کاراکتر در پشته قرار دهیم. سپس با دیدن کاراکتر های b و پس از آن کاراکتر های c، پشته را خالی کنیم."

این مسئله در واقع تضمین می‌کند که تعداد a ها در رشته، برابر نصف مجموع کاراکترهای b وc باشد. (دلیل نصف بودن را با نوشتن چند جمله از این زبان بیابید). ولی تعداد برابر a و b و c را تضمین نمی‌کند.

زبان $L=\{w\,c\,w^R \mid w \in \{a,b\}^*\}$ یک زبان مستقل از متن است. از نظر گرامری رشد زبان از دو طرف علامت C به ترتیب عکس خواهد بود. پس گرامر به صورت زیر خواهد بود:

$S \rightarrow aSa \mid bSb \mid c$

از نظر ماشین پشته ای نیز، از ابتدای رشته با دیدن هر کاراکتر علامت متناظر با آنرا در پشته قرار می‌دهیم. این کار را تا زمان رسیدن به علامت c ادامه میدهیم. با رسیدن به کاراکتر c، شروع به خارج نمودن علائم از پشته می‌کنیم. با خارج کردن هر کاراکتر از پشته، باید بررسی نماییم که کارکتر خارج شده با کاراکتری از رشته که روی آن هستیم، برابر باشد. در صورت برابر نبودن هر کاراکتر، شرط w و w^R نقض شده و پذیرش انجام نمی‌شود. اما در صورت برابری تمامی کارکتر ها، با رسیدن به انتهای رشته و خالی شدن همزمان پشته، می دانیم که w^R را مشاهده کرده ایم. پس رشته پذیرش می‌شود.

تست) زبان‌های L_1 و L_2 مفروض‌اند، کدام عبارت صحیح است؟

(سراسری ۸۹)

$$L_1\left\{w_1w_2 \mid w_1,w_2 \in (a+b)^*, |w_1|=|w_2|, w_1^R \neq w_2\right\}$$

$$L_2=\left\{a^n ww^R b^n \mid w \in (a+b)^*\right\}$$

۱) L_2 مستقل از متن و L_1 مستقل از متن نیست.

۲) L_1,L_2 مستقل از متن هستند.

۳) L_1 مستقل از متن و L_2 مستقل از متن نیست.

۴) هیچ‌یک از L_1 و L_2 مستقل از متن نیست.

یادداشت:

....

....

....

....

حل) گزینه ۲ درست است.
در مورد زبان اول، همانطور که برابری یک عبارت و معکوسش را با یک پشته می‌توان بررسی نمود، عکس این موضوع را نیز می‌توان با پشته نشان داد. زیرا در حالت اول در صورتی که با رسیدن به انتهای رشته، تمام علائم خارج شده از پشته برابر با علائم وارد شده به آن باشند، نشاندهنده $W\ W^R$ است و در غیر اینصورت، نشانگر عدم برابری دو زیر رشته مشاهده شده. برابری طول نیز دقیقا معادل زوج بودن طول رشته است. پس با پشته می‌توان زبان اول را شناسایی کرد. پس زبان مستقل از متن است. در مورد زبان دوم نیز که کاملا مشخص است، زبان از نوع مستقل از متن است.

ابهام در گرامرها و زبان‌های مستقل از متن

نکته (اشتقاق چپ و راست): در یک اشتقاق اگر در هر قدم متغیر سمت چپ شکل جمله‌ای جایگزین شود آن را اشتقاق چپ می‌گوییم و اگر در هر قدم راست‌ترین متغیر جایگزین شود آنرا اشتقاق راست می‌گوییم.

نکته (گرامر گنگ یا مبهم): گرامر مستقل از متن G را در صورتی گنگ می‌گوییم که یک $W\in L(G)$ وجود داشته باشد که حداقل دو اشتقاق چپ متفاوت یا دو اشتقاق راست متفاوت بتوان برای آن بدست آورد.

نکته (زبان ذاتاً گنگ یا ذاتاً مبهم): اگر L یک زبان مستقل از متن باشد که تمام گرامرهایی که L را تولید می‌کند گنگ باشد زبان را ذاتاً گنگ می‌گوییم.

مثال: زبان $L=\left\{a^nb^nc^m\right\}\cup\left\{a^nb^mc^m\right\}$ **به فرض مثبت بودن n و m، یک زبان مستقل از متن ذاتاً گنگ می‌باشد.**

نکته: برای تشخیص اینکه یک گرامر غیر گنگ است و همچنین برای تشخیص غیر گنگ بودن زبان الگوریتمی وجود ندارد. در نتیجه این دو مسئله تصمیم‌پذیر نمی‌باشد.

نکته (گرامر ساده): گرامر مستقل از متن $G=(V,T,S,P)$ را گنگ می‌گوییم در صورتی که تمامی قواعد آن به صورت زیر باشد.

$$A\rightarrow ax$$

در اینجا $A\in V$ و $a\in T$ و $x\in V^*$ است و هر زوج (A,a) حداکثر یکبار در P واقع شود. به گرامر ساده، S-grammer نیز می‌گویند.

نکته: هر گرامری که ساده باشد، غیر گنگ می‌باشد.

نکته: زبان‌های منظم ذاتاً گنگ نیستند. زیرا می‌توان برای آن‌ها یک گرامر ساده نوشت.

یادداشت:
....
....
....
....

نکته: برای اثبات اینکه یک زبان ذاتاً گنگ نمی‌باشد کافیست که برای آن زبان یک گرامر غیرگنگ بیابیم.

پویش و عضویت

با داشتن یک رشته w در L (G) می‌توان آن را به یک روش کاملاً واضع پویش نمود. برای این کار تمامی اشتقاق‌های چپ را می‌سازیم و ملاحظه می‌کنیم که آیا یکی از آنها با w منطبق می‌شود یا خیر.

نکته: به ازای هر گرامر مستقل از متن الگوریتمی وجود دارد که هر رشته $w \in L(G)$ را در تعداد مراحلی که متناسب با $|w|3$ است پویش کند.

ساده‌سازی گرامرهای مستقل از متن و فرم‌های نرمال

برای آسان کردن کار با گرامرهای مستقل از متن آنها را ساده می‌کنیم و سپس می‌توان گرامرها را به صورت نرمال درآوریم.

نکته: با انجام هر یک از مراحل ساده‌سازی گرامر زبان پذیرفته شده توسط آن گرامر تغییری نخواهد کرد.

نکته: در ساده‌سازی گرامرها حذف قوانین λ تولید ممکن است باعث ایجاد قوانین یکه بشود.

نکته: در ساده‌سازی گرامرها ترتیب حذف قوانین به صورت زیر می‌باشد:

۱. حذف قوانین λ

۲. حذف قوانین یکه

۳. حذف قوانین نامفید

نکته: پیچیدگی گرامر را به صورت زیر تعریف می‌کنند:

$$\text{Complexity (C)} = \sum_{A \to v \in P} \{1 + |v|\}$$

نکته: حذف قوانین بی‌فایده همیشه پیچیدگی را کاهش می‌دهد. اما حذف قانون λ تولید پیچیدگی را کاهش نمی‌دهد

فرم نرمال چامسکی

گرامری در فرم نرمال چامسکی است که تمام قوانین آن به صورت زیر باشد:

$$A \to BC$$

$$A \to a$$

بطوریکه $A, B, C \in V, a \in T$

یادداشت:

....

....

....

....

فرم نرمال گریباخ

گرامری در فرم نرمال گریباخ است که تمام قوانین آن به صورت زیر باشد

$$A \to ax$$

بطوریکه $x \in V^*, a \in T$ باشد.

نکته: با دقت به تعریف بالاو تعریف گرامر ساده می‌بینیم که فرم نرمال گریباخ عام‌تر از گرامرهای ساده می‌باشد یا به عبارت دیگر گرامرهای ساده زیر مجموعه‌ای از گرامرها به فرم نرمال گریباخ می‌باشند. اما برعکس آن صادق نمی‌باشد.

نکته: تعداد مراحل اشتقاق در یک گرامر به فرم نرمال چامسکی برای رشته W برابر $2|w|-1$ می‌باشد.

نکته: تعداد مراحل اشتقاق در یک گرامر به فرم نرمال گریباخ برای رشته W برابر $|w|$ می‌باشد.

تست) فرض کنید **G** یک گرامر مستقل از متن به فرم نرمال چاسکی باشد که رشته مبهم **W** را تولید می‌کند. در این صورت کدام گزینه درست است؟

(سراسری ۸۱)

۱) حداکثر تعداد مراحل در اشتقاق‌های چپ برای W برابر $2|w|$ است

۲) تعداد مراحل در اشتقاق‌های چپ برای W ثابت است.

۳) حداقل تعداد مراحل در اشتقاق‌های چپ برای W برابر $2|w|^2$ است.

۴) تعداد مراحل در دو اشتقاق چپ متفاوت برای W ممکن است متفاوت باشد

حل) گزینه ۲ صحیح است.

بنا به نکات گفته شده تعداد اشتقاق در این نوع گرامر ثابت است.

تست) ماشینی که با دریافت یک گرامر دلخواه چامسکی و یک رشته دلخواه **W** از واژه‌های زبان گرامر، تعیین می‌کند که آیا **W** به زبان گرامر تعلق دارد یا خیر مفروض است. بهترین عملکرد زمانی ممکن برای این ماشین بر حسب (طول رشته **W**) کدام است؟

(سراسری ۸۳)

$L_1 = \{x^n y^n \mid x \in (0+1)^*, y \in (0+1)^*, n \geq)\}$

$L_2 = \{w \in L(A) \mid$ **است و در مسیر پذیرش** w **از چند حالت معین** A **عبور نمی‌شود.** A **یک** DFA$\}$

$L_3 = \{w \in (0+1)^* \mid$ **تعداد** 0 **ها با** 1 **ها مساوی و برابر مقدار ثابت** $n \geq 0$ **باشد.**$\}$

یادداشت:

....

....

....

....

۱) $o(|w|^2)$ ۲) $o(|w|^3)$

۳) $o(2^{|w|})$ ۴) $o(\log|w|)$

حل) گزینه ۲ درست است.

از آنجا که قواعد گرامر در فرم نرمال چامسکی به صورت زیر است

$A \to BC$

$A \to a$

به همین خاطر برای رسیدن به یک رشته $|w|$ باید به اندازه $o(|w|^3)$ زمان احتیاج داریم. (الگوریتم CYK)

تست) گرامر $G:(\{S,X,Y\},\{a,b\},S,R)$ را با مجموعه قوانین زیر در نظر بگیرید:

(علوم کامپیوتر ۸۷)

$R:$
$S \to XY$
$S \to a$
$X \to YS \mid b$
$Y \to XS \mid b$

کدام گزاره نادرست است؟

۱) گرامر G یک گرامر مستقل از متن است.

۲) گرامر G به فرم نرمال چامسکی است

۳) baba توسط گرامر G تولید می‌شود.

۴) baba فقط به یک روش اشتقاق از روی قوانین گرامر G تولید می‌شود.

حل) **گزینه ۴ درست است.**

از آنجا که baba دارای یک اشتقاق چپ گرا و یک اشتقاق راست گرای متفاوت از یکدیگر است لذا گزینه ۴ صحیح است.

آتاماتاهای پشته‌ای (PDA)

این آتاماتاها دارای حافظه پشته می‌باشد بنابراین قدرت این ماشین‌ها از ماشین‌های متناهی بیشتر است.

این آتاماتاها به دو دسته تقسیم می‌شوند:

۱. آتاماتاهای پشته‌ای غیرقطعی (NPDA)

یادداشت:

....

....

....

....

۲. آتاماتاهای پشته‌ای قطعی (DPDA)

قدرت ماشین اول برابر PDA ها می‌باشد. یعنی می‌توان تمام زبان‌های مستقل از متن را با آن‌ها پیاده‌سازی کرد ولی قدرت ماشین دوم محدودتر می‌باشد و فقط زبان‌هایی به نام زبان‌های مستقل از متن قطعی را می‌توان با آن‌ها پیاده‌سازی کرد.

$$L(DPDA) \subset L(NPDA)$$

نکته: تعریف معین و نامعین بودن ماشین‌های پشته‌ای با ماشین‌های متناهی فرق می‌کند در اینجا حضور تغییر وضعیت‌های λ به معنای عدم قطعیت نمی‌باشد. همچنین بر خلاف ماشین‌های متناهی بعضی تغییر وضعیت‌های ماشین پشته‌ای معین ممکن است تهی باشد (یعنی تعریف نشده باشد) اما در ماشین‌های متناهی به ازای تمامی ورودی‌ها باید حالت تغییر وضعیت مشخص باشد و نمی‌تواند تهی باشد. در اینجا تنها شرطی که برای معین بودن وجود دارد این است که در همه حالات فقط یک حرکت امکان‌پذیر است.

نکته: هر زبان منظم یک زبان مستقل از متن معین است.

نکته: اگر L_1 یک زبان مستقل از متن معین و L_2 یک زبان منظم باشد، آنگاه $L_1 \cap L_2$ و $L_1 \cup L_2$ یک زبان مستقل از متن معین است.

نکته: معکوس یک زبان مستقل از متن معین، الزاماً معین نمی‌باشد.

نکته: یک زبان مستقل از متن معین هیچ گاه ذاتاً گنگ نیست.

نکته: برای هر NPDA، یک NPDA معادل وجود دارد که در آن پشته در حالت پذیرش خالی باشد.

نکته: تنها برای زبان‌های مستقل از متن معینی که هیچ رشته عضو زبان پیشوند رشته دیگر نباشد، یک DPDA معادل وجود دارد که در آن پشته در حالت پذیرش خالی باشد.

برای مثال برای زبان a^* یک DPDA با خالی شدن پشته در حالت پذیرش وجود ندارد.

نکته: برای هر زبان مستقل از متن یک NPDA با حداکثر سه حالت وجود دارد.

نکته: زبان‌های مستقل از متن قطعی، تحت عمل اشتراک و اجتماع بسته نیستند.

نکته: زبان‌های مستقل از متن قطعی، تحت همریختی بسته نیستند.

نکته: زبان‌های مستقل از متن قطعی، تحت عمل اشتراک و اجتماع با زبان‌های منظم بسته هستند.

یادداشت:

....

....

....

....

تست) کدام‌یک از عبارات زیر درست است؟

i: اگر L یک زبان مستقل از متن باشد، آن‌گاه یک ماشین پشته‌ای قطعی (DPDA) به نام M وجود دارد به طوری که $L = L(M)$.

ii: ماشین پشته‌ای که در هر حرکت خود محتوای پشته را فقط یک حرف افزایش و کاهش دهد، قادر است بخشی از زبان های مستقل از متن را بپذیرد (نه تمامی آن‌ها را).

۱) فقط ii
۲) فقط i
۳) i , ii
۴) هیچکدام

گزینه ۴ درست است.

حل) در جمله i برای زبان‌های مستقل از متن قطعی ماشین پشته‌ای قطعی وجود دارد ولی برای زبان‌های مستقل از متن غیر قطعی ماشین پشته‌ای قطعی موجود نیست؛ بنابراین i نادرست است. برای فوق زبان‌های مستقل از متن می‌توان ماشین پشته‌ای با خصوصیات مطرح‌شده در ii ایجاد کرد؛ بنابراین ii نیز نادرست است پس گزینه ۴ درست است.

تست) گزاره‌های زیر را در نظر بگیرید.

(سراسری ۸۰)

الف) زبان یک ماشین متناهی قطعی **(DFA)** یک زبان مستقل از متن قطعی است.

ب) زبان یک ماشین متناهی غیرقطعی **(NFA)** یک زبان مستقل از متن قطعی است.

ج) زبان یک ماشین پوش‌دان **(Push Down)** قطعی یک زبان مستقل از متن قطعی است

۱) فقط (ب) درست است
۲) فقط (ج) درست است
۳) فقط (الف) و (ج) درست است
۴) (الف) و (ب) و (ج) درست است

حل) گزینه ۴ درست است.

زبان ماشین متناهی قطعی و غیرقطعی زبان‌های منظم می‌باشند که زیر مجموعه‌ای از زبان‌های مستقل از متن قطعی هستند بنابراین گزاره‌های الف و ب درست است. گزاره ج نیز واضح است که صحیح می‌باشد.

یادداشت:

....
....
....
....

تست) $L=\left\{a^m cb^n : m\neq n\right\}\cup\left\{a^m d b^{2m} : m\geq 0\right\}$، کدام گزینه نادرست است؟

(سراسری ۸۶)

۱) هر هومرفیسم L با یک PDA معین شناسائی می‌شود
۲) یک گرامر غیرمبهم برای زبان L موجود است
۳) یک PDA نامعین برای شناسائی L موجود است
۴) همه موارد

حل) گزینه ۱ درست است.

گزینه ۱ نادرست است زیرا اگر هومرفیسمی به صورت $c\to\lambda$ و $d\to\lambda$ داشته باشیم زبان حاصل دیگر مستقل از متن معین نمی‌تواند باشد. در مورد گزینه ۲ می‌توان با استفاده از گرامر زیر ثابت کرد که درست است.

$S\to A|D$

$A\to aAb|Bc|cC$

$B\to aB|a$

$C\to bc|b$

$D\to aDbb|d$

در مورد گزینه ۳ نیز می‌توان گفت درست است زیرا زبان L یک زبان مستقل از متن می‌باشد بنابراین برای آن حتماً می‌توان یک NPDA طراحی کرد.

تست) $L=\left\{a^m cb^n \mid m\neq n\right\}\cup\left\{a^m db^{2m} \mid m\geq 0\right\}$ کدام گزینه نادرست است؟

(سراسری ۸۶)

۱) هر همومورفیسم L با یک PDA معین شناسایی می‌شود
۲) یک گرامر غیر مبهم برای زبان L موجود است.
۳) یک PDA نامعین برای شناسایی L موجود است.
۴) همه موارد

گزینه ۱ درست است.

حل) از آنجا که زبان L یک زبان مستقل از متن نامعین است و هر همومورفیسم آن نیز ممکن است مستقل از متن نامعین باشد لذا گزینه ۱ صحیح است.

یادداشت:

....

....

....

....

تست) مجموعه‌های زیر را در نظر بگیرید:

(سراسری ۸۵)

L (PDA): مجموعه زبان‌هایی که برای آنها PDA (Pushdown Automata) وجود دارد.

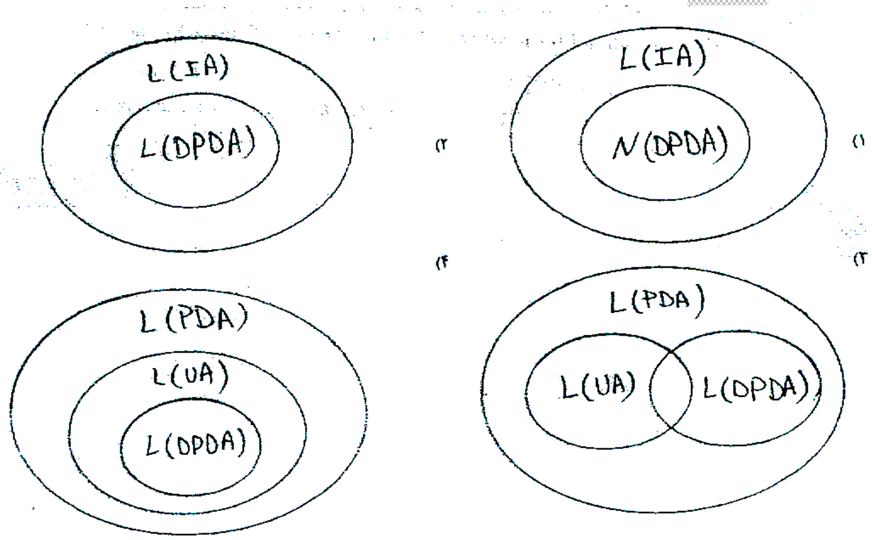
L (DPDA): مجموعه زبان‌هایی که برای آنها DPDA (Deterministic PDA) وجود دارد.

N (DPDA): مجموعه زبان‌هایی که برای آنها DPDA وجود درد و با خالی شدن پشته پذیرفته می‌شوند.

L (UA): مجموعه زبان‌های مستقل از متن غیرمبهم (unambiguous context free)

L (IA): مجموعه زبان‌های مستقل از متن ذاتاً مبهم (Inherently Ambiguous)

کدام یک از نمودارهای مجموعه‌ای زیر درست است؟



حل) گزینه ۴ درست است.

زبان‌های مستقل از متن معین هیچگاه ذاتاً مبهم نمی‌توانند باشند بنابراین گزینه ۱ و ۲ نادرست است. همچنین می‌توان ثابت کرد که زبان‌های مستقل از متن غیرمبهمی وجود دارد که مستقل از متن معین نمی‌باشند برای این مطلب به مثال زیر توجه کنید

$L=\{a^n b^n \mid n>0\} \cup \{a^n b^{2n} \mid n>0\}$

برای این زبان یک گرامر به صورت زیر می‌توان گرامر غیرمبهم زیر را در نظر گرفت

$S \to A \mid B$

$A \to a\,Ab \mid ab$

$B \to aBbb \mid abb$

یادداشت:

....

....

....

....

اما برای این زبان نمی‌توان یک DPDA طراحی کرد بنابراین مستقل از متن قطعی نمی‌باشد.

بنا به این مطلب زبان‌های مستقل از متن معین حتماً زیر مجموعه‌ای از زبان‌های غیرمبهم می‌باشند ولی بلعکس آن برقرار نمی‌باشد.

تست) برای کدام یک از گروه زبان‌های زیر **DPA** قطعی **(Deterministic Push Down Automata)** که در حالت خالی شدن **Stack** می‌پذیرد وجود دارد؟

(سراسری ۸۴)

۱) تمام زبان‌های مستقل از متن قطعی

۲) تمام زبان‌های منظم محدود (یعنی تعداد رشته‌های زبان محدود است.)

۳) تمام زبان‌های مستقل از متنی که هیچ رشته‌ای از زبان پیشوند رشتهٔ دیگری از زبان نباشد

۴) تمام زبان‌های منظمی که هیچ رشته‌ای از زبان پیشوند رشتهٔ دیگری از زبان نباشد

حل) گزینه ۳ درست است.

زبان‌هایی برای آن‌ها DPDA با خالی شدن پشته در هنگام پذیرش وجود دارد که هیچ رشته‌ای از زبان پیشوند رشته دیگری نباشد زیرا در این زبان‌ها اگر این حالت وجود داشته باشد پشته آتاماتا بعد از یک رشته، خالی می‌شود و دیگر با این پشته که عنصر انتهای پشته در آن نیست نمی‌توان رشته بعدی را بررسی کرد.

تست) زبان‌های منظم L_1، L_2، L_3 و L_4 مفروض‌اند:

(سراسری ۸۸)

$= L(a*)$

$= L\left((a+b)^*\right)$

$=\{w \in (a+b)^* \mid$ تعداد b های w زوج باشد $\}$

$=\{w \in (a+b)^* \mid$ تعداد b های w زوج و تعداد a های آن فرد باشد $\}$

برای چند زبان از این ۴ زبان می‌توان ماشین پشته‌ای با حداکثر ۲ حالت ساخت؟

۱) ۱ ۲) ۲

۳) ۳ ۴) ۴

حل) **گزینه ۴ درست است.**

ازآنجاکه همه زبان‌های مذکور منظم هستند، مستقل از متن نیز هستند. همچنین می‌دانیم که برای هر زبان مستقل از متن یک PDA با حداکثر دو حالت وجود دارد؛ بنابراین گزینه ۴ درست است.

یادداشت:

....

....

....

....

لم Pumping برای زبان‌های مستقل از متن

اگر L یک زبان مستقل از متن نامتناهی باشد آنوقت یک عدد صحیح مثبت مانند m وجود دارد بطوریکه هر رشته $w \in L$ با شرط $|w| \ge m$ می‌تواند به صورت زیر تجزیه شود.

$$w = uvxyz$$

$$|vxy| \le m$$

$$|vy| \ge 1$$

$$w_i = uv^i xy^i z$$

بطوریکه به ازای تمامی $i = 0,1,2,...$ رشته w_i در L باشد.

نکته: لم مکش / دمش برای نشان دادن اینکه برخی زبان ها مستقل از متن نیستند مورد استفاده قرار می‌گیرد و همواره با برهان خلف انجام می‌شود. از این لم نمی‌توان برای اثبات مستقل از متن بودن زبان‌ها استفاده کرد.

نکته: ثابت Pumping در زبان‌های مستقل از متن، تعداد متغیرهای گرامر آن زبان‌ها می‌باشد.

نکته: زبان‌های زیر همگی مستقل از متن نیستند، این مطلب را با لم مکش و دمش می‌توان اثبات کرد.

$L = \left\{a^{n!} \middle| n \ge 0\right\}$

$L = \left\{a^n b^j \middle| n = j^2\right\}$

$L = \{a^n \mid$ عدد اول است n$\}$

$L = \{a^{n^2} \mid n \ge 0\}$

خواص بستاری زبان‌های مستقل از متن

نکته: زبان‌های مستقل از متن بر روی الحاق و اجتماع و بستار ستاره بسته هستند.

نکته: زبان‌های مستقل از متن بر روی متمم و اشتراک بسته نیستند. بنابراین این زبان‌ها بر روی تفاضل نیز بسته نیستند.

نکته: زبان‌های مستقل از متن بر روی اشتراک و اجتماع با یک زبان منظم بسته هستند.

نکته: زبان‌های مستقل از متن تحت وارون و هومومورفیسم بسته هستند.

نکته: زبان‌های مستقل از متن معینی که خطی نباشند و بلعکس وجود دارند.

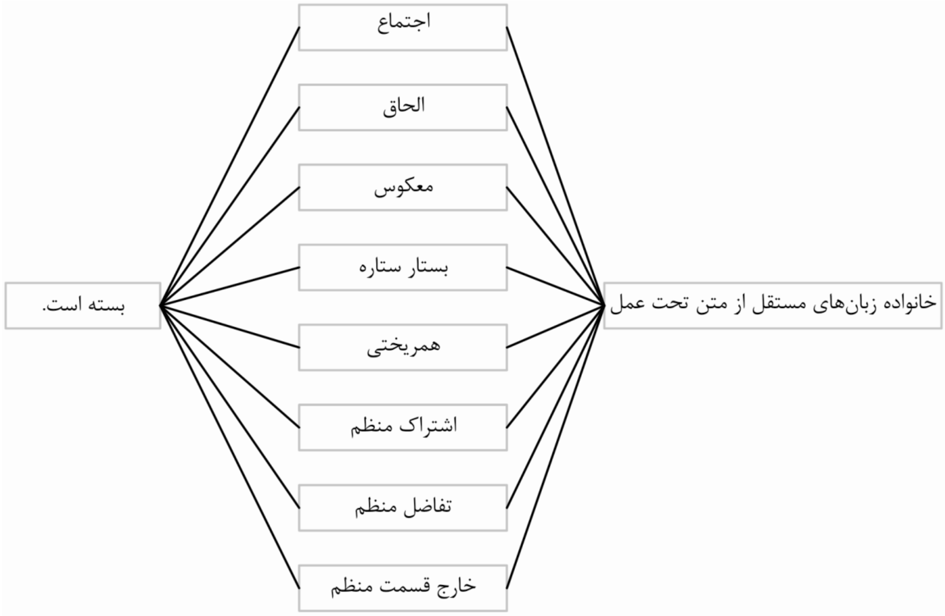
یادداشت:

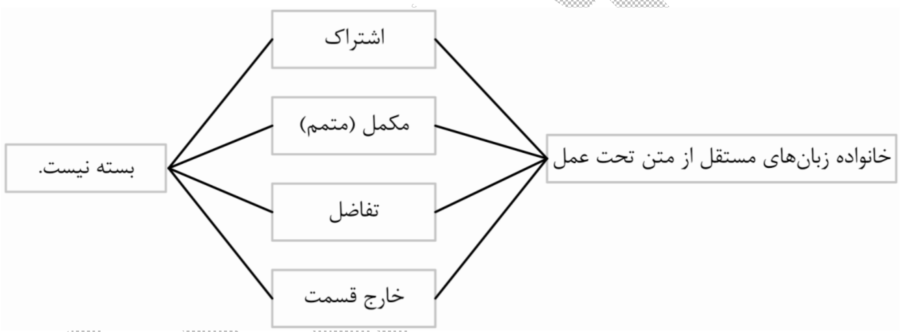
....

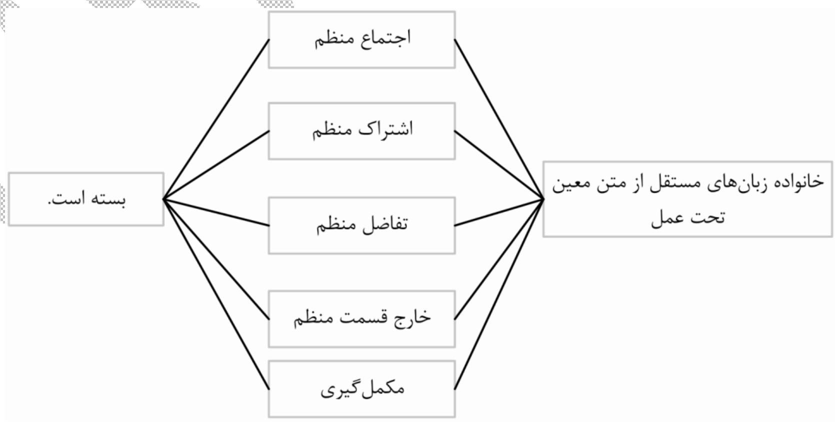
....

....

....


اجتماع
الحاق
معکوس
بستار ستاره
همریختی
اشتراک منظم
تفاضل منظم
خارج قسمت منظم
بسته است.
خانواده زبان‌های مستقل از متن تحت عمل


اشتراک
مکمل (متمم)
تفاضل
خارج قسمت
بسته نیست.
خانواده زبان‌های مستقل از متن تحت عمل


اجتماع منظم
اشتراک منظم
تفاضل منظم
خارج قسمت منظم
مکمل‌گیری
بسته است.
خانواده زبان‌های مستقل از متن معین تحت عمل

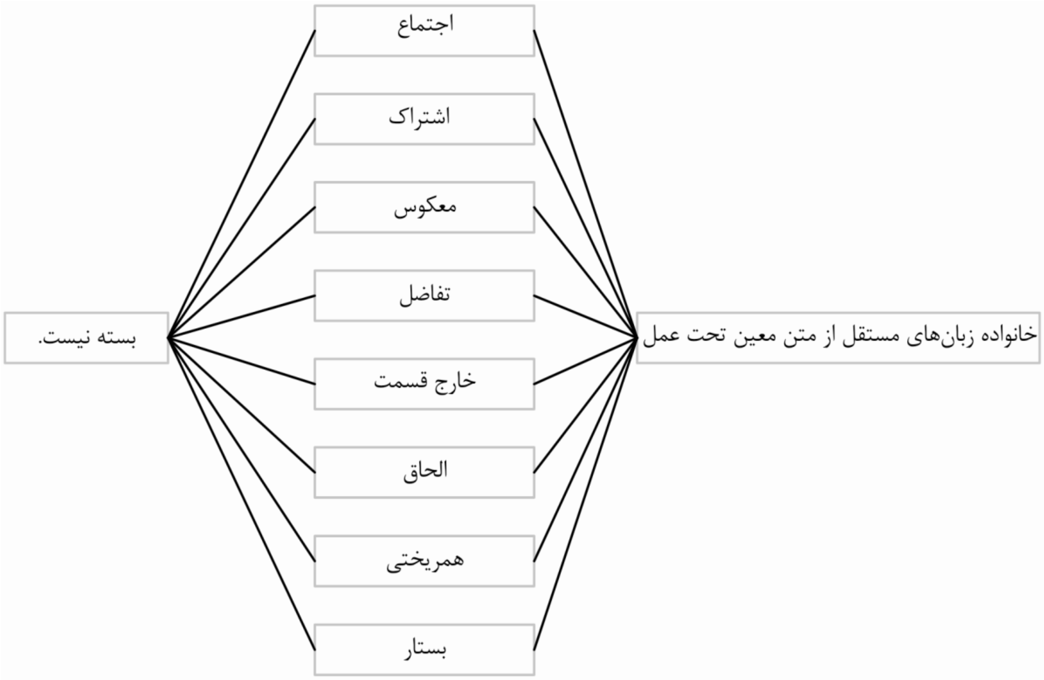
یادداشت:

....

....

....

....



تصمیم‌پذیری در زبان‌های مستقل از متن

مسائل زیر در مورد زبانهای مستقل از متن تصمیم‌پذیرند:

۱. وجود رشته‌ای به طول دقیقاً n در زبان‌های مستقل از متن

۲. وجود عضو مشترک بین یک زبان منظم و یک زبان مستقل از متن

۳. تهی بودن یا متناهی بودن زبان‌های مستقل از متن

۴. عضویت یک رشته در یک زبان مستقل از متن

مسائل زیر تصمیم‌پذیر نیستند:

۱. تشخیص مبهم بودن یک گرامر مستقل از متن

۲. تشخیص این که برای یک زبان مستقل از متن آیا $L=\Sigma^*$

۳. تساوی دو زبان مستقل از متن

یادداشت:

....

....

....

....

تست) **عمل بر زدن روی زبان‌های L_1 و L_2 به شرح زیر تعریف می‌شود:**

$$S(L_1,L_2)=\{(wv)^* \mid w\in L_1, v\in L_2\}$$

کدام گزاره صحیح است؟

(سراسری ۸۴)

۱) زبان‌های مستقل از متن تحت عمل بر زدن (S) بسته نیستند.

۲) زبان‌های مستقل از متن تحت عمل بر زدن (S) بسته هستند

۳) زبان‌های مستقل از متن تحت عمل بر زدن (S) بسته نیستند ولی زبان‌های منظم تحت آن عمل بسته هستند

۴) زبان‌های منظم تحت عمل بر زدن (S) بسته نیستند ولی زبان‌های مستقل از متن تحت آن عمل بسته هستند

حل) گزینه ۲ درست است.

عمل برزدن در حقیقت از ترکیب دو عمل الحاق و بستار ستاره ایجاد شده که هم زبان‌های منظم و هم زبان‌های مستقل از متن تحت این دو عمل بسته هستند بنابراین گزینه ۲ صحیح است)

تست) کدام یک از زبانهای زیر مستقل از متن است؟

(سراسری ۸۳)

۱) $L=\{a^{n^2} : n=3k\}$

۲) $L=\{a^{2^n} : n=3k)$

۳) $L=\{a^n : n\geq 100$ یا n عدد اول$\}$

۴) هیچکدام

حل) گزینه ۳ صحیح است.

زبان‌های گزینه ۱ و ۲ زبان‌های مستقل از متن نیستند. در مورد زبان گزینه ۳ به می‌توان آن را به صورت اجتماع دو زبان به صورت زیر نوشت.

$L_1=\{a^n \mid n\geq 100\}$

$L_2=\{a^n \mid$ n عدد اول است$\}$

$L=L_1 U L_2$

متمم این زبان به صورت زیر می‌شود

$\overline{L}=\{a^n \mid n\geq 100\}\cap\{a^n \mid$ n عدد اول نیست$\}$

این زبان، یک زبان متناهی می‌باشد و منظم است از آنجا که زبان‌های منظم تحت متمم بسته هستند بنابراین می‌توان گفت که L نیز منظم است بنابراین L حتما مستقل از متن نیز می‌باشد.

یادداشت:

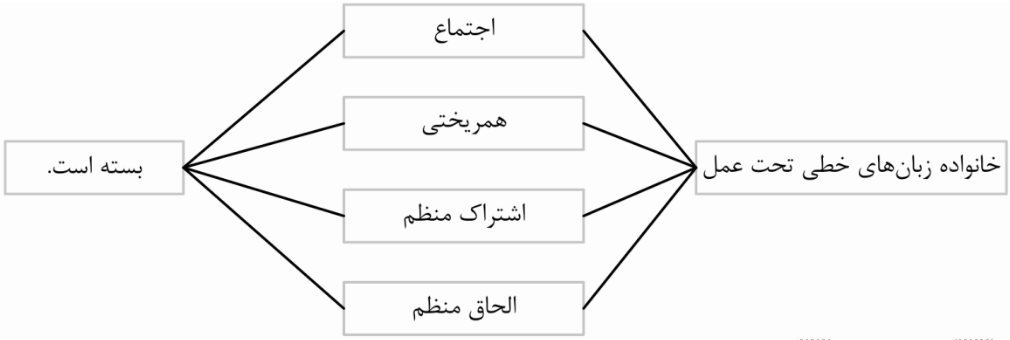
....

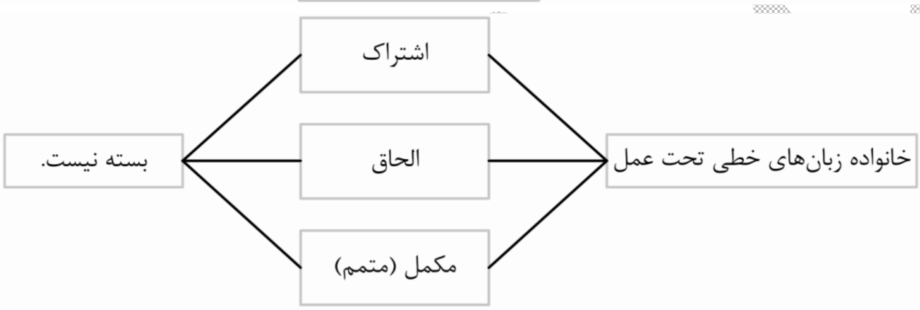
....

....

....

خواص بستاری زبان‌های خطی





ماشین تورینگ

قبل از بررسی زبانهای نوع اول زبانهای نوع صفر یا زبانهای بی قید و شرط را بررسی می‌کنیم. ماشین معادل این زبانها، ماشین تورینگ نام دارد و گوییم که این ماشینها معادل الگوریتم هستند. یعنی برای هر مسئله‌ای در دنیا که راه حل داشته باشد، می توان یک ماشین تورینگ طراحی نمود. در واقع ماشین تورینگ توانایی حل هر مسئله ای را دارد. به همین دلیل است که با هر دستکاری در عملکرد ماشین تورینگ، نمی‌توان توانایی های آنرا افزایش داد. مثلا با اضافه کردن یک نوار ورودی اضافه، جدا سازی ورودی و خروجی و یا ... تنها ممکن است بدلیل افزایش منابع سرعت کار را افزایش داد ولی تواناییهای ماشین را نمی‌توان افزایش داد. البته ممکن است گاهی در ماشین تورینگ استاندارد، دستکاری‌هایی انجام دهیم که محدود کننده باشند. در این گونه موارد باید با بررسی این محدودیت های اعمال شده، تاثیر آنها را کشف کرده و کارایی ماشین جدید را بسنجیم.

یادداشت:

....
....
....
....

تست) **ماشین تورینگ تک‌نواره $M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$ مفروض است. در این ماشین Q مجموعه حالات کنترل، Σ الفبای زبان، Γ الفبای قابل درج روی نوار، δ تابع تبدیل، $q_0 \in Q$ حالت اولیه ماشین، $B \in \Gamma - \Sigma$ علامت مخصوص جاهای خالی روی نوار و $F \subseteq Q$ مجموعه حالات نهایی M است. در تعریف این ماشین $\Gamma = \Sigma \cup \{B\}$ و δ فقط شامل حرکت به راست بوده و محتوای نوار را تغییر نمی‌دهد. زبان‌های قابل پذیرش توسط M**

الف) از نوع منظم هستند.

ب) از نوع حساس به متن هستند و از نوع مستقل از متن نیستند.

ج) از نوع بدون محدودیت هستند و از نوع حساس به متن نیستند.

د) از نوع مستقل از متن هستند و از نوع منظم نیستند.

گزینه ۱ درست است.

حل) ماشین تورینگی که فقط قادر به حرکت به سمت راست باشد دارای قدرتی معادل با ماشین متناهی است؛ بنابراین گزینه ۱ درست است.

نکته: ماشین‌های تورینگ جهت محاسبه یا پذیرش می‌تواند مورد استفاده قرار بگیرد.

نکته: قدرت ماشین‌های دو پشته‌ای با ماشین‌های تورینگ برابر است.

نکته: قدرت انواع ماشین‌های تورینگ با ماشین تورینگ استاندارد برابر است

زبان‌های حساس به متن

زبان‌هایی را توسط گرامرهای حساس به متن پیاده‌سازی می‌شوند را زبان حساس به متن گویند.

نکته: زبان‌های حساس به متن را می‌توان با آتاماتای کراندار خطی (LBA) پیاده‌سازی کرد.

نکته: زبان‌های مستقل از متن زیر مجموعه زبان‌های حساس به متن هستند.

یادداشت:

....

....

....

....

تست) ماشین تورینگ M با دستورات حرکت زیر مفروض است که در آن q_0 حالت شروع، q_f حالت پایانی و B علامت خانه‌های خالی دو طرف نوار است. منظور از $\delta(q,a)=(P,X,R)$ این است که اگر M در حالت q و سر آن مقابل حرف a روی نوار باشد، آن‌گاه به حالت P رفته، a را با X عوض کرده و سر را به اندازه یک خانه به راست می‌برد (اگر به‌جای **R**، **L** باشد، آن‌گاه به چپ می‌رود) اگر در شروع کار M (یعنی حالت q_0 و سر در ابتدای ورودی روی نوار) محتوی نوار برابر رشتهٔ aaabbb باشد، پس از دقیقاً 11 حرکت δ محتوی نوار کدام است؟

(سراسری ۸۸)

۱) XaaYYb
۲) XXaYYb.
۳) XXaYbb
۴) XXXYYY

حل) **گزینه ۱ درست است.**

با پیمایش مرحله به مرحله قوانین، باتوجه به محل هد ماشین و رشته ورودی، پس از ۱۱ حرکت عبارت فوق بر روی نوار ماشین خواهد بود

تست) برای تشخیص زبان $L=\{a^nb^n \mid n\geq 0\}$ یک ماشین تورینگ ساخته‌ایم. حداقل هزینه تشخیص $w\in L$ با این ماشین تورینگ در چه حدی است؟

(سراسری ۸۶)

۱) $O(n)$
۲) $O(n^2)$.
۳) $O(n^3)$
۴) $O(2^n)$

حل) **گزینه ۲ درست است.**

برای تشخیص زبان L باید به ازای دیدن هر a به ابتدای حروف b رفته و دوباره به ابتدای رشته بازگشت؛ بنابراین درمجموع به اندازه $O(n^2)$ باید زمان صرف کرد؛ پس گزینه ۲ درست است.

یادداشت:

....
....
....
....

تست) فرض کنید:

$$L_1 = \left\{a^p b^q a^p b^s \mid p,q,s \ge 0\right\}$$
$$L_2 = \left\{a^p b^p a^r b^s \mid p,r,s \ge 0\right\}$$
$$L_3 = \left\{a^p b^q a^r b^p \mid p,q,r \ge 0\right\}$$

کدام‌یک از گزینه‌های زیر درست است؟

الف L_1 **یک زبان مستقل از متن** (Context-Free) **است.**

ب L_3 یک زبان مستقل از متن نیست.

ج L_2 **یک زبان مستقل از متن نیست**

د $L_1 \cap L_2 \cap L_3$ **یک زبان مستقل از متن است.**

گزینه ۲ درست است.

حل) امکان استفاده از انتقال بلادرنگ تغییری در قدرت ماشین تورینگ ایجاد نمی‌کند؛ پس گزینه ۱ نادرست است. داشتن دو نوار یک‌طرفه معادل با داشتن یک نوار دوطرفه است؛ ازاین‌رو گزینه ۳ نادرست است. محدود شدن حروف الفبا به 0 و 1 نیز تأثیری در قدرت ماشین نخواهد داشت ولی محدودیت در جهت حرکت قدرت ماشین تورینگ را به حد ماشین متناهی کاهش می‌دهد؛ پس گزینه ۲ درست است.

همانطور که گفتیم، اگر برای حل مسئله ای راه حل داشته باشیم، آنگاه برای آن مسئله ماشین تورینگ داریم. در دنیای واقعی، ما برای برخی مسائل می‌توانیم پاسخ صریح ارائه دهیم. یعنی راه حل ما همواره پاسخ قبول و یا رد را به ما خواهد داد. به عبارت دیگر در صورت عضویت یک رشته در زبان مورد نظر، ماشین همواره پاسخ yes و در غیر اینصورت همواره پاسخ no را به ما می‌دهد. در برخی مسائل-(زبانها)، قضیه کمی پیچیده تر است. یعنی ماشین در صورت پذیرش رشته در زبان، پاسخ yes را به ما می‌دهد ولی روی تمامی رشته‌های غیر عضو، پاسخ قطعی به ما نمیدهد. یعنی ممکن است زمان بسیاری بگذرد ولی ماشین هنوز پاسخ yes را به ما نداده باشد. این حالت را نمی توان پاسخ no در نظر گرفت زیرا ممکن است ماشین برای تعیین عضویت به زمان بیشتری نیاز داشته باشد. اصطلاحا گوییم که ماشین تورینگ بر روی برخی رشته هایش توقف نمی‌کند. پس طبق مطالب فوق مشخص است که حتی نمی‌توان یک بازه زمانی را برای توقف ماشین معرفی کرد.

اگر یک ماشین تورینگ(یک زبان بی قید و شرط – نوع صفر) بر روی تمام رشته های ورودی پاسخ قطعی yes / no را به ما بدهد، یعنی بر روی تمام رشته هایش متوقف شود، گوییم این زبان یک زبان بازگشتی(Recursive - REC) است.

یادداشت:

....

....

....

....

اما اگر زبان بی قید و شرط، بر روی یکسری از رشته ها پاسخ قطعی yes را بدهد ولی روی تمام رشته های ورودی توقف نداشته باشد، گوییم این زبان شمارش پذیر بازگشتی(Recursively Enumerable - RE) است.

برای درک این دو تعریف، این سوال را درنظر بگیرید:

"در تهران تا ۲۴ ساعت آینده برف می بارد"

این مسئله، یک مسئله بازگشتی است. زیرا برای دادن پاسخ صریح yes / no به آن الگوریتم داریم. الگوریتم کار به این صورت است:

"تا فردا ساعت ۱۲ صبر کن، در صورتی که برف بارید، پاسخ yes را بده و در صورتی که تا پایان مهلت برف نبارید، پاسخ no را بده."

حال مسئله دوم را به این صورت در نظر بگیرید: "روزی از آسمان یک کیسه پر از پول در داخل حیاط منزل شما می افتد"

برای بررسی این مسئله چه راه حلی وجود دارد؟ یک راه حل اینست که دائما حیاط منزلتان را بررسی کنید تا در صورت مشاهده کیسه، پاسخ yes را بدهید. اما در صورتی که کیسه ای را نبینید، آیا می‌توان اصل قضیه را زیر سوال برد؟ پاسخ خیر است. دلیل آن هم ذات سوال و مسئله است. به صورت مسئله دقت کنید؛ چون گفته است که روزی پس شما با گذشت زمانی مشخص، اگر اتفاقی نیفتاد، نمیتوانید موضوع را نفی کنید. زیرا ممکن است اتفاق چند روز دیگر پیش بیاید. پس هیچگاه نمی‌توانید به این مسئله پاسخ no را با قطعیت بدهید. اما اگر صورت سوال به این صورت بود که : "تا سال آینده..." یا " تا ۱۰۰ سال آینده ... "؛ آنگاه برای این مسئله، مانند مسئله قبل می-توانستیم پاسخ yes یا no را بصورت قطعی بدهیم.

پس به صورت خلاصه باید گفت که اگر برای یک مسئله خاص همواره بتوان پاسخ yes و یا no را در یک زمان مشخص و متناهی داد(یا به عبارت دیگر ماشین تورینگ آن مسئله، بر روی تمام رشته های ورودی متوقف شود و پاسخ قطعی پذیرش و یا عدم پذیرش را بدهد) گوییم زبان مورد نظر بازگشتی است. در غیر این صورت، اگر فقط پاسخ yes را بدهد(yes را تضمین کند بدون زمان مشخص) ولی در حالات رد ممکن باشد پاسخ مشخص ندهد، زبان را شمارش پذیر بازگشتی می نامیم.

روال بر شمارش

یک الگوریتم جهت شمارش عناصر یک زبان را روال برشمارش گویند.

نکته: هر زبانی که دارای روال برشمارش باشد، یک زبان بازگشتی برشمردنی است.

نکته: زبان‌های بازگشتی زیر مجموعه زبان‌های بازگشتی برشمردنی هستند.

نکته: زبان‌های بازگشتی بر روی متمم بسته هستند.

نکته: زبان‌های بازگشتی برشمردنی بر روی متمم بسته نیستند.

نکته: زبان‌های بازگشتی بر روی الحاق و اجتماع و اشتراک بسته هستند.

یادداشت:

....

....

....

....

نکته: زبان‌های بازگشتی برشمردنی بر روی الحاق و اجتماع و اشتراک بسته هستند.

مسئله توقف

مسئله توقف در ماشین‌های تورینگ به این معنی است که آیا می‌توان ماشین تورینگی طراحی کرد که یک ماشین تورینگ و یک رشته دلخواه را به عنوان ورودی دریافت کند و تشخیص دهد که آیا این ماشین بر روی این رشته توقف می‌کند یا خیر.

نکته: مسئله توقف در ماشین‌های تورینگ، یک مسئله تصمیم‌پذیر نمی‌باشد.

تصمیم پذیری در زبانهای بازگشتی و بازگشتی برشمردنی

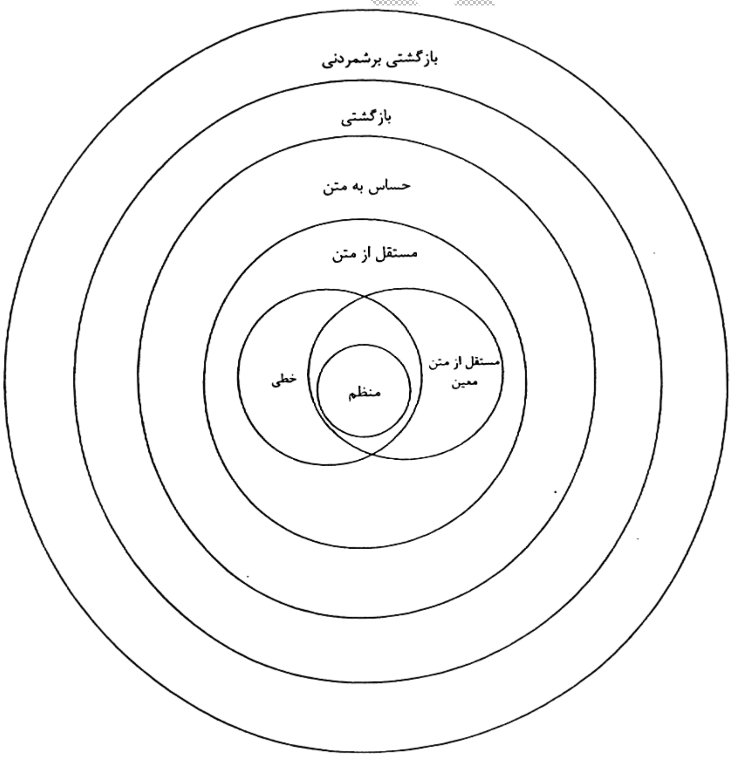
مسائل زیر تصمیم‌پذیر نیستند:

۱. تهی بودن یک گرامر مقید

۲. برقراری رابطه زیر مجموعه بین دو زبان تولید شده توسط ماشین تورینگ

طبقه‌بندی زبان‌ها

نمودار زیر رابطه بین زبان‌های مختلف را نشان می‌دهد. این نمودار گسترش یافته نمودار چامسکی می‌باشد.



یادداشت:

....

....

....

....

خواص الگوریتمیک خانواده زبان‌های بازگشتی (Rec)

- اگر L یک زبان بازگشتی باشد، آن‌گاه برای آن الگوریتم عضویت وجوددارد.
- اگر L یک زبان بازگشتی باشد آن‌گاه الگوریتمی که مشخص کند آیا زبان L ، تهی است یا خیروجودندارد.

خواص الگوریتمیک خانواده زبان‌های بازگشتی‌برشمردنی (RE)

- اگر L یک زبان بازگشتی برشمردنی باشد، آن‌گاه برای‌آن الگوریتم عضویت وجودندارد.(برای زبان‌های غیربازگشتی الگوریتم عضویت وجودندارد.)
- اگر G یک‌گرامر بی‌قید وشرط (بدون محدودیت) باشد، آن‌گاه الگوریتمی که مشخص‌کند آیا زبان $L(G)$ و معکوس آن $L(G)^R$ باهم برابراست یاخیر، وجودندارد.

تست) زبان $L=\left\{0^n1^{f(n)} \mid n\in\mathbb{N}\right\}\subseteq\{0,1\}^*$ را برای تابع $f:\mathbb{N}\to\mathbb{N}$ در نظر بگیرید. کدام گزاره نادرست است؟

(علوم کامپیوتر ۸۷)

الف اگر f یک‌به‌یک باشد، آن‌گاه L منظم نیست.

ب برای برخی توابع f پوشا L می‌تواند مستقل از متن باشد

ج زبان L می‌تواند محاسبه‌پذیر نباشد.

د زبان L حداقل وابسته به متن است.

گزینه ۴ درست است.

حل) ازآنجاکه ممکن است تابع f محاسبه‌پذیر نباشد، برای آن هیچ ماشین تورینگی وجود ندارد و بنابراین گزینه ۴ درستاست

تست) کدام گزاره درست است؟

(علوم کامپیوتر ۸۷)

الف برای هر زبان $L\subseteq\{0,1\}^*$ یک اتوماتون قطعی وجود دارد که زبان آن با L برابر است

ب برای هر زبان $L\subseteq\{0,1\}^*$ یک اتوماتون قطعی متناهی وجود دارد که زبان آن با L برابر است.

ج برای هر زبان $L\subseteq\{0,1\}^*$ یک اتوماتون غیر قطعی متناهی وجود دارد که زبان آن با L برابر است.

د برای هر زبان نامتناهی $L\subseteq\{0,1\}^*$ یک اتوماتون غیر قطعی متناهی وجود دارد که زبان آن با L برابر است.

یادداشت:

....

....

....

....

حل) **همه گزینه‌ها نادرست است.**

ازآنجاکه زبان‌هایی وجود دارند که با هیچ ماشین مکانیکی قابل پذیرش نیستند، همه گزینه‌ها نادرست هستند

تست) زبانی **L** مجموعه تمامی زوج‌های مرتب < **M, W** > است که در آن **M** که یک ماشین تورینگ و **W** یک رشته است به طوری که ماشین **M** بر ورودی **W** متوقف نمی‌شود. کدام یک از جملات زیر صحیح است؟

(سراسری ۸۸)

الف **L** بازگشتی است.

ب **L** به طور بازگشتی شمار است.

ج **L** بازگشتی نیست.

د **L** به طور بازگشتی شمارا نیست.

۱) ب ۴) الف و ب

۲) ب و ج ۳) ج و د

حل) گزینه ۴ درست است.

برای جواب به این سوال باید دقت کرد زبان‌های بازگشتی بر روی تمام رشته‌ها توقف می‌کنند چه رشته‌هایی که عضو زبان ماشین است که رشته‌هایی که عضو زبان آن نیست. ولی در مورد زبان‌های بازگشتی برشمردنی فقط ماشین تورینگ معادل با آن بر روی رشته‌های عضو زبان توقف می‌کند و بر روی سایر رشته‌ها ممکن است توقف کند و ممکن است در حلقه بینهایت بیافتد.

بنا به مطالب بالا این زبان بازگشتی نیست زیرا بر روی w توقف نمی‌کند. از طرفی این زبان بازگشتی برشمردنی نیز نیست زیرا این زبان‌ها نمی‌توانند رشته‌هایی که بر روی آن‌ها توقف می‌کنند را تشخیص دهند زیرا اگر این گونه بود مسئله توقف در ماشین‌های تورینگ تصمیم‌پذیر بود. بنابراین گزینه ۴ درست است

تست) برای تشخیص زبان $L=\left\{a^n b^n \mid n \geq 0\right\}$ یک ماشین تورینگ ساخته‌ایم. حداقل هزینه تشخیص $w \in L$ با این ماشین تورینگ در چه حدّی است؟

(سراسری ۸۶)

۱) O (n) ۲) O(n2)

۳) O(n2) ۴) O (2n)

یادداشت:

....

....

....

....

حل) گزینه ۲ درست است.

اگر ماشین تورینگ مورد نظر را از نوع استاندارد در نظر بگیریم آنگاه برای پیاده سازی این زبان به $2n^2$ بار حرکت نیاز داریم. دلیل این موضوع نیز به این خاطر است که به ازای هر a باید n بار به جلو حرکت کنیم و یک b را به y تبدیل کنیم و دوباره n بار برگردیم و a بعدی را بررسی کنیم.

تست) زبان **L** با تعریف زیر مفروض است. کدام یک از گزاره‌ها غلط است؟

$L=\{x^i v^j\ z^{j+2}\ w^k\ v^{i+k}\ |i,j,k\geq 0\}$

(سراسری ۸۶)

۱) یک اتاماتای پشته‌ای غیرقطعی مثل A وجود دارد به قسمی که L= L(A)

۲) رشته‌های L توسط یک اتاماتای قطعی کراندار (Linear Bounded Automata) قابل شناسایی هستند

۳) زبان L از نوع مستقل از متن معین (DCFL) نمی‌باشد.

۴) زبان L از نوع بازگشتی شمارش‌پذیر است

یادداشت:

....

....

....

....

حل) گزینه ۳ درست است.

گزینه‌ها را بررسی می‌کنیم:

گزینه ۱ درست است: این زبان یک زبان مستقل از متن می‌باشد، بنابراین برای آن آتاماتای پشته‌ای غیرقطعی وجود دارد.

گزینه ۲ درست است: زیرا این زبان مستقل از متن است بنابراین حتماً حساس به متن نیز می‌باشد و برای آن می‌توان LBA در نظر گرفت.

گزینه ۳ نادرست است: زیرا این زبان مستقل از متن قطعی است.

گزینه ۴ درست است: زیرا زبان‌های مستقل از متن زیر مجموعه زبان‌های بازگشتی می‌باشند.

تست) زبان L مجموعه تمامی زوج‌های مرتب $\langle M, w \rangle$ است که در آن M یک ماشین تورینگ و w یک رشته است به طوری که ماشین M بر ورودی w متوقف نمی‌شود. کدام‌یک از جملات زیر صحیح است؟

(سراسری ۸۸)

i : L بازگشتی است.

ii : L به طور بازگشتی شمارا است.

iii : L بازگشتی نیست.

iv : L به طور بازگشتی شمارا نیست.

الف) i , ii

ب) ii

ج) ii , iii

د) iii و iv

حل) گزینه ۴ درست است.

زبان مورد نظر تشخیص‌پذیر نیست به این معنی که ماشین تورینگی وجود ندارد که بتواند با دریافت رشته‌های آن، آن‌ها را تشخیص داده و متوقف شود؛ بنابراین زبان مورد نظر بازگشتی برشمردنی نیست پس گزینه ۴ درست است.

تست) مجموعه زبان‌های بازگشتی **(Recursive)** را **R** و مجموعه زبان‌های بازگشتی شمارش‌پذیر **(Recursively Enumerable)** را **RE** می‌نامیم. زبان **L** مفروض است. در کدام یک از حالت‌های زیر یک ماشین تورینگ که برای تمام رشته‌های **L** به حالت توقف برسد وجود دارد؟

سراسری ۸۹)

الف) $\overline{L} \notin RE, L \in RE$

ب) $\overline{L} \notin R, L \in RE$

ج) $\overline{L} \in RE, L \in RE$

د) هیچکدام

یادداشت:

....

....

....

....

گزینه ۳ درست است.

حل) دقت کنید که منظور از زبانهایی که "بر روی تمامی رشته‌های ورودی متوقف شوند" اینست که زبان مورد نظر یک زبان بازگشتی است. از طرفی می‌دانیم که اگر یک زبان و متمم آن، هر دو بازگشتی شمارش پذیر باشند، آنگاه آن زبان یک زبان بازگشتی است. دلیل این مسئله اینست که چون خود زبان بازگشتی شمارش‌پذیر است، پس رشته های عضوش را میشناسد و از طرفی چون متمم آن نیز شمارش‌پذیر بازگشتی است، پس ما می‌توانیم رشته هایی که جزو زبان نیستند را نیز می‌توانیم شناسایی کنیم. پس هم رشته های عضو و هم رشته های غیر عضو آن قابل شناسایی هستند. یعنی در مورد این زبان ما پاسخ **Yes** و **No** را می‌توانیم مشخص کنیم. پس زبان مورد نظر بازگشتی است.

تست) گرامر زیر را در نظر بگیرید. همه گزاره های زیر صحیح اند به جز**:**

(علوم کامپیوتر ۸۵)

$$
\begin{aligned}
&G:\\
&S \to XY\\
&X \to 0X\\
&X \to \lambda\\
&Y \to 1Y\\
&Y \to \lambda
\end{aligned}
$$

۱) $L(G)$ مستقل از متن است.

۲) $w \in L(G)$ تصمیم پذیر نیست.

۳) $L(G)$ منظم است.

۴) $L(G)$ اشتراک دو زبان مستقل از متن است.

حل) **گزینه ۲ درست است.**

از آنجا که گرامر مورد نظر مستقل از متن است لذا زبان آن نیز مستقل از متن است و جمله ۱ درست است. زبان گرامر مورد نظر برابر با 0^*1^* است و لذا منظم بوده و جمله ۳ درست است. هر زبان را می‌توان اشتراک خودش با خودش در نظر گرفت و لذا جمله ۴ نیز درست است. از آنجا که زبان‌های منظم دارای الگوریتم عضویت هستند لذا جمله ۲ نادرست بوده و بنابراین گزینه ۲ صحیح است.

یادداشت:

....

....

....

....

دسته بندی یکسری زبان

ردیف	زبان	منظم	مستقل از متن
۱	$L=\left\{w_1cw_2 : w_1,w_2 \in \{a,b\}^*, w_1 \neq w_2\right\}$	–	–
۲	$L=\left\{w_1cw_2 : w_1,w_2 \in \{a,b\}^*, w = w_2\right\}$	–	–
۳	$L=\left\{uww^Rv : u,v \in \{a,b\}^+, \lvert u\rvert \geq \lvert v\rvert\right\}$	–	–
۴	$L=\left\{a^nb^n : n\geq 0\right\}\cup\left\{a^nb^{2n} : n\geq 0\right\}\cup\left\{a^nb^nc^n : n\geq 0\right\}$	–	–
۵	$L=\left\{a^nb^nc^n : n\geq 0\right\}$	–	–
۶	$L=\left\{ww : w\in\{a,b\}^*\right\}$	–	–
۷	$L=\left\{a^{n!} : n\geq 0\right\}$	–	–
۸	$L=\left\{a^nb^j : n=j^2\right\}$	–	–
۹	$L=\{a^n \mid$ عدد اول است n $\}$	–	–
۱۰	$L=\left\{a^{n^2} : n\geq 0\right\}$	–	–
۱۱	$L=\left\{a^nb^j : n\geq j^2\right\}$	–	–
۱۲	$L=\left\{a^nb^j : n\geq (j-1)^3\right\}$	–	–
۱۳	$L=\left\{a^nb^jc^k : k=nj\right\}$	–	–
۱۴	$L=\left\{a^nb^jc^k : k>n, k>j\right\}$	–	–
۱۵	$L=\left\{a^nb^jc^k : n<j, n\leq k\leq j\right\}$	–	–
۱۶	$L=\left\{w\in\{a,b,c\}^* : n_a(w)<n_b(w)<n_c(w)\right\}$	–	–
۱۷	$L=\left\{w\in\{a,b,c\}^* : n_a(w)/n_b(w)=n_c(w)\right\}$	–	–

یادداشت:

....

....

....

....

۱۸	$L=\{a^nb^ja^nb^j : n \ge 0, j \ge 0\}$	–	–
۱۹	$L=\{a^nb^ja^kb^l : n \le k, j \le l\}$	–	–
۲۰	$L=\{a^nb^nc^j : n \le j\}$	–	–
۲۱	$L=\{a^{nm} \mid$ m و n اعداد اول هستند $\}$	–	–
۲۲	$L=\{a^nb^nc^m : n,m \ge 0\} \cap \{a^nb^mc^m \; n,m \ge 0\}$	–	–
۲۳	$L=\{w \in \{a,b,c\}^* : n_a(w)=n_b(w)=n_c(w)\}$	–	–

یادداشت:

....

....

....

....

ردیف	زبان	منظم	مستقل از متن
۲۴	$L=\left\{w\in\{a,b,c\}^*: n_a(w)=n_b(w)=n_c(w)\right\}\cap(abc)^*$	–	–
۲۵	$L=\left\{a^nb^na^nb^n : n\geq 0\right\}$	–	–
۲۶	$L=\left\{w_1w_2 : w_1\neq w_2, \|w_1\|=\|w_2\|\right\}$	–	–
۲۷	$L=\left\{a^{m'>m} : m, m'>0\right\}$	–	–
۲۸	$L=\left\{xww^Ry : x,y,w\in\{a,b\}^+, \|x\|\geq\|y\|\right\}$	–	–
۲۹	$L=\left\{w_1\subset w_2 : w_1,w_2\in\{a,b\}^*\right\}$	–	–
۳۰	$L=\left\{a^nb^nc^{2n} : n\geq 1\right\}$	–	–
۳۱	$L=\left\{a^nb^{2n}a^n : n\geq 1\right\}$	–	–
۳۲	$L=\left\{a^nb^n : n\geq 0\right\}$	–	✓
۳۳	$L=\left\{ww^R : w\in\Sigma^*\right\}$	–	✓
۳۴	$L=\left\{ab(bbaa)^n bba(ba)^n : n\geq 0\right\}$	–	✓
۳۵	$L=\left\{a^nb^l : n\neq l\right\}$	–	✓
۳۶	$L=\left\{a^{2n}b^m : n,m\geq 0\right\}$	✓	✓
۳۷	$L=\left\{ab^{2n} : n\geq 1\right\}$	✓	✓
۳۸	$L=\left\{a^nb^m : n\leq(m+3)\right\}$	–	✓
۳۹	$L=\left\{a^nb^m : n\neq m-1\right\}$	–	✓
۴۰	$L=\left\{a^nb^m : n\neq 2m\right\}$	–	✓
۴۱	$L=\left\{a^nb^m : 2n\leq m\leq 3n\right\}$	–	✓
۴۲	$L=\left\{w\in\{a,b\}^* : n_a(w)\neq n_b(w)\right\}$	–	✓
۴۳	$L=\left\{a^nb^mc^k : n=m \text{ or } m\leq k, n,m\geq 0\right\}$	–	✓
۴۴	$L=\left\{a^nb^mc^k : n=m \text{ or } m\neq k, n,m\geq 0\right\}$	–	✓
۴۵	$L=\left\{a^nb^mc^k : k=n+m, n,m\geq 0\right\}$	–	✓
۴۶	$L=\left\{a^nb^mc^k : n+2m=k\right\}$	–	✓

یادداشت:

....

....

....

....

ردیف	زبان	منظم	مستقل از متن
۴۷	$L=\{a^n b^m c^k : k \mid m-n\}$	–	✓
۴۸	$L=\{w\in\{a,b,c\}^* : n_a(w)+n_b(w)\neq n_c(w)\}$	–	✓
۴۹	$L=\{a^n b^m c^k : k\neq n+m\}$	–	✓
۵۰	$L=\{a^n ww^R b^n : w\Sigma^*, n\geq 1\}$	–	✓
۵۱	$L=\{a^n b^m : n\leq m+3\}$	–	✓
۵۲	$L=\{uvwv^R : u,v,w\in\{a,b\}^+, \lvert u\rvert=\lvert v\rvert=2\}$	–	✓
۵۳	$L=\{w_1cw_2 : w_1,w_2\in\{a,b\}^+, w_1\neq w_2^R\}$	–	✓
۵۴	$L=\{a^n b^n c^m : n,m\geq 0\}\cup\{a^n b^m c^m : n.m\geq 0\}$	–	✓
۵۵	$L=\{w\in\{a,b\}^* : n_a(w)=n_b(w)\}$	–	✓
۵۶	$L=\{a^n b^{2n} : n\geq 0\}$	–	✓
۵۷	$L=\{wcw^R : w\in\{a,b\}^*\}$	–	✓
۵۸	$L=\{a^n b^m c^{n+m} : n\geq 0, m\geq 0\}$	–	✓
۵۹	$L=\{w\in\{a,b\}^* : 2n_a(w)\leq n_b(w)\leq 3n_a(w)\}$	–	✓
۶۰	$L=\{w_1cw_2 : w_1,w_2\in\{a,b\}^*, w_1\neq w_2^R\}.\{a^*\}$	–	✓
۶۱	$L=\{ab(ab)^n b(ab)^n : n\geq 0\}$	–	✓
۶۲	$L=\{a^{n+1} b^{2n} : n\geq 0\}$	–	✓
۶۳	$L=\{a^n b^{n+1} : n\geq 0\}$	–	✓
۶۴	$L=\{a^n b^n : n\geq 0\}\cup\{a^n b^{2n} : n\geq 0\}$	–	✓
۶۵	$L=\{a^n b^n : n\geq 0\}\cup\{a\}$	–	✓
۶۶	$L=\{a^n ww^R a^n : n\geq 0, w\in\{a,b\}^*\}$	–	✓
۶۷	$L=\{a^n b^j a^j b^n : n\geq 0, j\geq 0\}$	–	✓
۶۸	$L=\{a^n b^j a^k b^l : n+j\leq k+1\}$	–	✓
۶۹	$L=\{a^n b^n a^m b^m : n\geq 0, m\geq 0\}$	–	✓

یادداشت:

....

....

....

....

ردیف	زبان	منظم	مستقل از متن
۷۰	$L=\{a^n b^j ; j \le n \le 2j-1\}$	–	✓
۷۱	$L=\{a^n b^n : n \ge 0, n \ne 100\}$	–	✓
۷۲	$L=\{w \in \{a,b\}^* : n_a(w)=n_b(w),$ w شامل زیر رشته aab نیست $\}$	–	✓
۷۳	$L=\{a^n b^n \mid$ n مضربی از 5 نیست $\}$	–	✓
۷۴	$L=\{xa^n b^n : n \ge 0\} \cup \{ya^n b^n : n \ge 0\}$	–	✓
۷۵	$L=\{a^n b^n : n \ge 0\} \cup \{a^n c^{2n} : n \ge 0\}$	–	✓
۷۶	$L=\{a^n b^n : n \ge 0\} \cup \{b^{2n} : n \ge 0\}$	–	✓
۷۷	$L=\{wcw^R v : w, v \in \{a,b\}^*\}$	–	✓
۷۸	$L=\{wcw^R a^n : w \in \{a,b\}^*, n \ge 0\}$	–	✓
۷۹	$L=\{a^n b^m : n \ne 2m\}$	–	✓
۸۰	$L=\{a^i b^j c^i c^j c^k d^k : i, j, k \ge 0\}$	–	✓
۸۱	$L=\{a^n : n=k^2,$ برای برخی از $k>0\}$	–	–
۸۲	$L=\{a^n : n=2^k,$ برای برخی از $k>0\}$	–	–
۸۳	$L=\{a^n b^k c^l \mid$ $\frac{n}{l}$ یک عدد صحیح است $\}$	–	–
۸۴	$L=\{a^n b^l a^k : n>5, l>3, k>1\}$	–	✓
۸۵	$L=\{a^n b^l \mid$ $n+1$ عدد اول است $\}$	–	–
۸۶	$L=\{abab^n : n \ge 0\} \cup \{aba^n : n \ge 0\}$	✓	✓
۸۷	$L=\{a^n : n \ge 0\} \cup \{b^n a : n \ge 1\}$	✓	✓
۸۸	$L=\{awa : w \in \{a,b\}^*\}$	✓	✓
۸۹	$L=\{ab^5 wb^4 : w \in \{a,b\}^*\}$	✓	✓
۹۰	$L=\{a^n b^m : n \ge 1, m \ge 1, nm \ge 3\}$	✓	✓
۹۱	$L=\{ab^n w : n \ge 3, w \in \{a,b\}^+\}$	✓	✓
۹۲	$L=\{a^{2n} b^{2m+1} : n, m \ge 0\}$	✓	✓

یادداشت:

....

....

....

....

ردیف	زبان	منظم	مستقل از متن
۹۳	$L=\{vwv : v,w\in\{a,b\}^*, \lvert v\rvert=2\}$	✓	✓
۹۴	$L=\{a^nb^m \mid$ n+m زوج است $\}$	✓	✓
۹۵	$L=\{w_1bw_2 : \lvert w_1\rvert \bmod 2=\lvert w_2\rvert \bmod 2, w_1,w_2\in\{a\}^*\}$	✓	✓
۹۶	$L=\{w\in\{a,b\}^* : n_a(w), n_b(w)$ هر دو زوج‌اند $\}$	✓	✓
۹۷	$L=\{a^n : n=i+jk, i,k \text{ fixed}, j=0,1,2,\ldots\}$	✓	✓
۹۸	$L=\{a^nb^la^k : n+l+k>5\}$	✓	✓
۹۹	$L=\{uww^Rv : u,v,w\in\{a,b\}^+\}$	✓	✓

ردیف	زبان مستقل از متن	قطعی	غیر قطعی
۱	$L=\{a^nb^n : n\ge 0\}$	✓	✓
۲	$L=\{ww^R : w\in\Sigma^*\}$	–	✓
۳	$L=\{a^nb^m : 2n\le m\le 3n\}$	–	✓
۴	$L=\{w\in\{a,b\}^* : n_a(w)\ne n_b(w)\}$	✓	✓
۵	$L=\{a^nb^mc^k : n=m \text{ or } m\le k, n,m\ge 0\}$	–	✓
۶	$L=\{a^nb^mc^k : n=m \text{ or } m\ne k, n,m\ge 0\}$	–	✓
۷	$L=\{w_1cw_2 : w_1,w_2\in\{a,b\}^+, w_1\ne w_2^R\}$	✓	✓
۸	$L=\{w\in\{a,b\}^* : n_a(w)=n(w)=n_b(w)\}$	✓	✓
۹	$L=\{a^nb^{2n} : n\ge 0\}$	✓	✓
۱۰	$L=\{wcw^R : w\in\{a,b\}^*\}$	✓	✓
۱۱	$L=\{w\in\{a,b\}^* : 2n_a(w)\le n_b(w)\le 3n_a(w)\}$	–	✓

یادداشت:

....

....

....

....

۱۲	$L=\left\{w_1cw_2 : w_1,w_2\in\{a,b\}^*, w_1\neq w_2^R\right\}.\left\{a^a\right\}$	✓	✓
۱۳	$L=\left\{a^nb^n : n\geq 0\right\}\cup\left\{a^nc^{2n} : n\geq 0\right\}$	–	✓
۱۴	$L=\left\{a^nb^n : n\geq 0\right\}\cup\{a\}$	✓	✓
۱۵	$L=\left\{a^nb^mc^k : n=m \mid m=k\right\}$	–	✓
۱۶	$L=\left\{a^nb^na^mb^m : n,m\geq 0\right\}$	✓	–
۱۷	$L=\left\{xa^nb^n : n\geq 0\right\}\cup\left\{ya^nb^b : n\geq 0\right\}$	✓	✓
۱۸	$L=\left\{a^nb^n : n\geq 0\right\}\cup\left\{a^nc^{2n} : n\geq 0\right\}$	✓	✓
۱۹	$L=\left\{a^nb^n : n\geq 0\right\}\cup\left\{b^{2n} : n\geq 0\right\}$	✓	✓
۲۰	$L=\left\{wcw^Rv : w,v\in\{a,b\}^*\right\}$	✓	✓

ردیف	زبان	منظم	مستقل از متن	حساس به متن	ماشین پذیرنده
۱	$L=\left\{a^nb^nc^n \mid n\geq 1\right\}$	–	–	✓	کراندار خطی (LBA)
۲	$L=\left\{a^{n!} \mid n\geq 1\right\}$	–	–	✓	کراندار خطی (LBA)
۳	$L=\left\{a^{n^2} \mid n=m, m\geq 1\right\}$	–	–	✓	کراندار خطی (LBA)
۴	$L=\left\{a^n \mid\right.$ n عدد اول است $\left.\right\}$	–	–	✓	کراندار خطی (LBA)
۵	$L=\left\{a^n \mid\right.$ n عدد اول نیست $\left.\right\}$	–	–	✓	کراندار خطی (LBA)
۶	$L=\left\{ww \mid w\in\{a,b\}^+\right\}$	–	–	✓	کراندار خطی (LBA)
۷	$L=\left\{w^n \mid w\in\{a,b\}^+, n\geq 1\right\}$	–	–	✓	کراندار خطی (LBA)

یادداشت:

....

....

....

....