به نام خدا

مقدمه ای بر
نظریهٔ زبان‌ها و ماشین‌ها

مولفان :

وحید خطیبی بردسیری

(عضو هیئت علمی دانشگاه آزاد اسلامی واحد بردسیر)

بهروز صادقی

(عضو هیئت علمی دانشگاه پیام نور) واحد تایباد

سمیه عسکرپور

(مدرس دانشگاه فنی و حرفه ای دختران کرمان)

منیره اسفندیاری

(مدرس دانشگاه پیام نور واحد تایباد)

انتشارات ارسطو

(چاپ و نشر ایران)

۱۳۹٤

شابک : ۹ – ۱۲ – ۷۹۴۰ – ۶۰۰ – ۹۷۸
شماره کتابشناسی ملی : ۳۸۳۵۱۱۲
عنوان و نام پدیدآور : مقدمه ای بر نظریه زبان ها و ماشین ها
مشخصات نشر : مشهد: ارسطو، ۱۳۹۴.
مشخصات ظاهری : ۴۰۳ ص.: مصور ،جدول ،نمودار
یادداشت : فهرستنویسی کامل این اثر در نشانی: http://opac.nlai.ir قابل دسترسی است.
سرشناسه : خطیبی بردسیری ، وحید ، ۱۳۶۰ –
سرشناسه : صادقی ، بهروز، ۱۳۶۱–
سرشناسه : عسکرپور ، سمیه ، ۱۳۶۰ –
سرشناسه : اسفندیاری ، منیره ، ۱۳۶۵ –
وضعیت فهرست نویسی : فیپای مختصر

نام کتاب :مقدمه ای بر نظریهٔ زبان ها و ماشین ها
مولفان : وحید خطیبی بردسیری - بهروز صادقی – سمیه عسکرپور – منیره اسفندیاری
ویراستاران : فخرالسادات فانیان (مدرس دانشگاه آزاد اسلامی واحد کرمان)
فرزانه فاموری (عضو هیئت علمی دانشگاه آزاد اسلامی واحد فسا)
ناشر : ارسطو
صفحه آرایی ، تنظیم و طرح جلد : پروانه مهاجر
تیراژ : ۱۰۰۰
نوبت چاپ : اول - ۱۳۹۴
چاپ : مهتاب
قیمت : ۲۹۰۰۰ تومان
شابک : ۹-۱۲-۷۹۴۰-۶۰۰-۹۷۸
تلفن های مرکز پخش : ۵۰۹۶۱۴۵ ــ ۳۵۰۹۶۱۴۶ - ۰۵۱
www.chaponashr.ir

فهرست

عنوان **صفحه**

پیش گفتار ۱۰

فصل اول : مقدمه ای بر نظریه محاسبات ۱۱

۱-۱ الفبا و رشته ها Alphabet and Strings ۱۱

۱-۲ عملیات بر روی رشته ها ۱۲

۱-۳ زبانLanguage ۱۴

۱-۳-۱ عملگرهای قابل تعریف بر روی یک زبان ۱۵

۱-۴ گرامر (Grammer) ۱۹

۱-۵ اشتقاق ۱۹

فصل دوم : ماشین آتاماتا ۲۱

۲-۱ ماشین‌های آتاماتا (Automata) ۲۱

۲-۲ پذیرنده Acceptor ۲۲

۲-۳ ماشین متناهی (Finite Automata) ۲۴

۲-۳-۱ ماشین های متناهی قطعی (Deterministic Finite Automata) ۲۴

۲-۳-۲ یک دسته بندی غیر رسمی از ماشین های آتاماتای متناهی قطعی ۲۶

۲-۴ ماشین‌های آتاماتای متناهی غیرقطعی (None Deterministic Finite Automata) ۳۵

۲-۵ تفاوت‌های ماشین آتاماتای متناهی قطعی و غیر قطعی ۳۵
۲-۶ تبدیل ماشین آتاماتای متناهی غیرقطعی به قطعی ۳۹
فصل سوم: عبارات منظم، گرامرهای منظم، زبانهای منظم ۴۳
۳-۱ عبارات منظم، گرامرهای منظم، زبانهای منظم ۴۳
۳-۲ عبارات منظمRegular Expressions ۴۳
۳-۲ معادل بودن عبارات منظم وماشین های آتاماتای محدود ۴۶
۳-۲-۱ تبدیل عبارت منظم به ماشین آتاماتای متناهی غیرقطعی ۴۶
۳-۳ تبدیل ماشین آتاماتای قطعی به عبارت منظم ۴۷
۳-۴ گرامر های منظم ۵۲
۳-۴-۱ گرامر خطی راست ۵۲
۳-۴-۲ گرامر خطی چپ ۵۲
۳-۴-۳ گرامر خطی ۵۳
۳-۵ نحوه تبدیل ماشین آتاماتای قطعی به گرامر منظم ۵۴
۳-۶ تبدیل گرامر به ماشین ۵۵
۳-۷ فرایند کمینه سازی ماشین آتاماتای متناهی قطعی ۵۶
۳-۸- ترکیب وضعیتهای معادل در ماشین آتاماتای متناهی قطعی ۵۶
فصل چهارم : ویژگیهای زبان های منظم ۵۹
۴-۱ لم تزریق ۵۹
۴-۲ خواص بستاری زبان های منظم ۶۰
۴-۳ خواص تصمیم پذیری زبان‌های منظم ۶۳
فصل پنجم : زبانهای مستقل از متنContext Free ۶۹

۵-۱ گرامرهای مستقل از متن ۶۹
۵-۲ تجزیه کننده ها، درخت اشتقاق، ابهام ۷۵
۵-۲-۱ تجزیه کننده بالا به پایین ۷۵
۵-۳ ابهام ۷۷
فصل ششم : ساده سازی گرامر های مستقل از متن و فرم‌های نرمال ۸۰
۶-۱ حذف قواعد بی فایده ۸۰
۶-۲ حذف قوانین λ ۸۱
۶-۳ حذف قوانین یکتا ۸۲
۶-۴ حذف قوانین بازگشتی چپ ۸۲
۶-۵ دو فرم نرمال مهم ۸۳
۶-۵-۱ فرمال نرمال چامسکی ۸۳
۶-۵-۲ فرم نرمال گریباخ ۸۴
۶-۵-۳ الگوریتم CYK ۸۵
فصل هفتم : آتاماتای پشته‌ای (Push Down Automata) ۸۶
۷-۱ ماشین‌های پشته‌ای ۸۶
۷-۲ ماشین‌های پشته‌ای نامعین None Deterministic Push Down Automata ۹۰
فصل هشتم : خواص زبانهای مستقل از متن ۱۱۴
۸-۱ لم تزریق ۱۱۴
۸-۱-۱ لم تزریقی برای زبانهای خطی ۱۱۵
۸-۲ خواص بسته بودن زبانهای مستقل از متن: ۱۱۶
فصل نهم : ماشین تورینگ Turing Machine ۱۲۳

۹-۱ ماشین تورینگ ۱۲۳
۹-۱-۱ تعریف ماشین تورینگ ۱۲۴
۹-۱-۳ وضعیت ماشین ۱۲۵
۹-۲ ماشین تورینگ به عنوان پذیرنده ۱۲۶
۹-۳ ماشین تورینگ به عنوان تراگذر ۱۲۸
۹-۴ تز تورینگ ۱۳۲
۹-۵ ترکیب ماشین های تورینگ ۱۳۲
فصل دهم : انواع دیگر ماشین های تورینگ ۱۳۴
۱۰-۱ انواع دیگر ماشین تورینگ ۱۳۴
۱۰-۲ انواع ماشین تورینگ ۱۳۵
۱۰-۲-۱ ماشین تورینگ با گزینه ایستادن ۱۳۵
۱۰-۲-۲ ماشین تورینگ چندشیاره: ۱۳۶
۱۰-۲-۳ ماشین تورینگ با نوار نیمه محدود (از یک سر محدود) ۱۳۶
۱۰-۲-۴ ماشین تورینگ چند نواره ۱۳۶
۱۰-۲-۵ ماشین تورینگ چند بعدی ۱۳۷
۱۰-۲-۶ ماشین تورینگ برون خط ۱۳۷
۱۰-۲-۷ ماشین تورینگ غیرقطعی ۱۳۸
۱۰-۲-۸ ماشین تورینگ عمومی ۱۴۰
۱۰-۲-۹ کدینگ کردن ماشین تورینگ ۱۴۰
۱۰-۲-۱۰ ماشین تورینگ عمومی ۱۴۱
فصل یازدهم : دسته بندی زبان‌های صوری و آتاماتا ۱۴۵

۱-۱۱ زبان‌های برگشتی شمارش پذیرش ۱۴۵
۲-۱۱ زبانهای بازگشتی ۱۴۵
۳-۱۱ زبان‌هایی که بازگشتی برشمردنی نیستند ۱۴۷
۴-۱۱ مسائلی که توسط ماشین تورینگ قابل حل نیستند ۱۵۰
۵-۱۱ گرامرهای نامقید (Unrestricted Grammars) ۱۵۳
۶-۱۱ گرامرهای حساس به متن ۱۵۴
۷-۱۱ گرامرهای حساس به متن، زبانهای حساس به متن و گرامرهای نامقید ۱۵۵
۸-۱۱ آتاماتای خطی محدود ۱۵۶
۹-۱۱ تقسیم بندی چامسکی ۱۵۷
۱۰-۱۱ ویژگیهای بستاری ۱۵۸
۱۱-۱۱ پیچیدگی محاسباتی ۱۶۰
نمونه سوالات تستی و پاسخنامه تشریحی دانشگاه پیام نور ۱۶۳
نیم سال اول ۸۶-۸۵ ۱۶۳
سوالات تشریحی ۱۶۶
پاسخ سوالات تشریحی ۱۶۷
نیم سال دوم ۸۶-۸۵ ۱۶۸
سوالات تشریحی ۱۷۳
پاسخ سوالات تشریحی ۱۷۳
نیم سال اول ۸۷-۸۶ ۱۷۵
سوالات تشریحی ۱۸۱
پاسخ سوالات تشریحی ۱۸۱

نیم سال دوم ۸۶ ..... ۱۸۴
سوالات تشریحی ۱۹۱
پاسخ سوالات تشریحی ۱۹۲
نیم سال اول ۸۷ ..... ۱۹۳
سوالات تشریحی ۲۰۵
پاسخ سوالات تشریحی ۲۰۶
نیمسال دوم ۸۸-۸۷ ..... ۲۰۸
تابستان ۸۸ ..... ۲۱۹
سئوالات تشریحی ۲۲۹
پاسخ تشریحی ۲۳۰
نیم سال اول ۸۹-۸۸ ..... ۲۳۲
سوالات تشریحی ۲۴۳
پاسخ تشریحی ۲۴۵
نیمسال دوم ۸۹-۸۸ ..... ۲۵۰
نیمسال اول ۹۰-۸۹ ..... ۲۶۲
سوالات تشریحی ۲۷۲
پاسخ سوالات تشریحی ۲۷۳
نیمسال دوم ۹۰-۸۹ ..... ۲۷۵
سوالات تشریحی ۲۸۷
پاسخ سوالات تشریحی ۲۸۸
نیمسال اول ۹۱-۹۰ ..... ۲۹۰

نیمسال دوم ۹۱-۹۰ ..... ۳۰۷
تابستان۹۱ ..... ۳۲۲
نیمسال اول۹۲-۹۱ ..... ۳۳۰
نیمسال دوم ۹۲-۹۱ ..... ۳۴۱
تابستان ۹۲ ..... ۳۵۲
نیم سال اول ۹۳-۹۲ ..... ۳۶۵
نیمسال دوم ۹۳-۹۲ ..... ۳۷۹
تابستان ۹۳ ..... ۳۹۳
پاسخ سوالات تستی ۴۰۶

پیش گفتار

رشته کامپیوتر یک زمینه مطالعاتی نظری و کاربردی می باشد که البته جنبه های کاربردی آن در جوامع امروزی ، بسیار فراتر از جنبه هاب نظری می باشد که در این زمینه وجود دارد.شاید برای دانشجویان رشته های مهندسی کاپیوتر این سئوال مطرح باشد در شرایطی که کاربردهای عملی کامپیوتر ها به مراتب روز به روز بیشتر میگردد چرا بایستی مطالعه نظری این علم مهندسی را نیز فرابگیریم. در پاسخ به اینگونه سئوالات باید یادآور شد که تمامی کاربردهای متنوع کامپیوتری امروزی، مبتنی بر یک سریه نظریه ها و مفاهیم اساسی می باشندکه فرا گیری آنها تنها راه رسیدن به کاربردهای این چنینی برای طراحان و برنامه نویسان سیستم های کامپیوتری میباشد.. البته باید عنوان داشت که مفاهیم مطرح شده در نظریه‌ها، اکثرا در طراحی مدارات الکترونیکی، زبانهای برنامه نویسی، کامپایلرها و مفسرها، سیستم های تشخیص الگو، پردازش زبانهای طبیعی و ... نقش عمده ای دارند. از طرفی دیگر، مسائل مرتبط با نظریه های کامپیوتری عموما جنبه ریاضی و البته سرگرمی داشته و می‌توانند به صورت حل معماها درکاربردهای طبیعی نیز بیان گردند.

هدف از درس نظریه زبان ها و ماشین ها ، ارائه مدلهای مجردی از ماشین هاو کامپیوترها می باشد. در این درس مفاهیم سه گانه زبان، گرامر و آتاماتا مورد بررسی تکمیلی قرار گرفته و زبانهای صوری مختلف از حوزه های گوناگون ساختار زبانی ، نوع گرامر به کاررفته در آنها و نیز توانمندی‌های ماشین های قابل تعریف در آنها تبیین میگردند. یک زبان صوری ، انتزاعی از مشخصه‌های کلی و عمومی زبانهای برنامه نویسی می باشد. هرچند بسیاری از زبانهای مورد مطالعه در این کتاب به مراتب ساده تر از زبانهای برنامه نویسی هستند اما از همان ویژگیهای مشترک برخوردار هستند و بامطالعه این زبانهای صوری است که می‌توان مسائل مختلف مربوط به زبان‌های برنامه نویسی را آموخت و نسبت به ارائه راه حل های جدید برای آنها اقدام نمود.

فصل اول : مقدمه ای بر نظریه محاسبات

۱-۱ الفبا و رشته ها Alphabet and Strings

الفبا مجموعه‌ای متناهی از نمادها یا سمبل‌ها می‌باشد که آن را با Σ نشان می‌دهیم. مانند:

$\Sigma = \{0, 1\}$ یا $\Sigma = \{a, b, c\}$

رشته: یک رشته روی الفبا به صورت دنباله ای متناهی از نمادهای آن الفبا می باشد.

مانند رشته ۰۱۰۱۰ که روی الفبای $\Sigma = \{0, 1\}$ تعریف شده است.

طول رشته ها برابر است با تعداد نمادهای موجود در آن رشته که آن را با $|w|$ نشان می‌دهیم مثلاً اگر $w_0 = 010$ باشد $|w_0| = 3$

رشته به طول صفر را با λ نشان می دهیم. یعنی $|\lambda| = 0$

نکته: $\lambda \notin \Sigma$ ولی اگر $\lambda \in \Sigma$ باشد باید برای رشته به طول صفر یا همان رشته تهی از یک نماد دیگر استفاده کرد.

۲-۱ عملیات بر روی رشته ها

الحاق یا اتصال (Concatination)

می‌توان دو رشته u وv را کنار هم قرار داد و رشته Z=uv را ایجاد نمود. طول رشته z برابر با مجموع طول رشته های u وv می باشد.

ترتیب قرار گرفتن رشته ها مهم می باشد.

نکته: U*V ≡ UV

نکته: UV ≠ VU

نکته: w=λ⁎w=w*λ

نکته: w^n یعنی الحاق رشته w با خودش به تعداد n بار مثلاً اگر w= a b

$$w^0 = \lambda, \qquad w^1 = w = ab, \qquad w^2 = w * w = abab,$$

$$w^3 = w * w^2 = ababab$$

معکوس رشته: معکوس یا وارون رشته W را با W^R نمایش می‌دهیم و به صورت زیر تعریف می‌کنیم:

$$|w| = 0 => w^R = \lambda$$

$$|w| = n + 1\ , n \geq 0 => W = u * a\ , U \in \Sigma^*, a \in \Sigma => W^R = a * u^R$$

نکته: عملگر وارون نسبت به عمل الحاق دارای خاصیت پخشی است.

$$(UV)^R = V^R * U^R$$

$$(W^R)^R = W \qquad (w^n)^R = (w^R)^n$$

زیر رشته : رشته Vرا یک زیر رشته از رشته W گوییم بطوریکه دنباله ای متوالی از حروف W باشد. مثلاً اگر W=abbaa باشد رشته های λ ,a ,... ,a, a b , a می توانند زیر رشته هایی از آن باشند.

پیشوند: زیر رشته هایی از W که از سمت چپ ترین حروف آن شروع شوند را پیشوند آن رشته گوییم مانند: زیر رشته های...,abb, ab ,a ,λ

پسوند : زیر رشته هایی از w که به سمت راست ترین حرف یک رشته ختم شوند را پسوند آن رشته گوییم مانند: زیر رشته های ...,bbaa , b a a ,a a ,a, λ

نکته : اگر در یک رشته ، به ترتیب زیررشته، پیشوند و پسوند آن را با SUB, PRE, SUF نمایش دهیم خواهیم داشت:

$S \cup F \subseteq S \cup B$

$PREF \subseteq S \cup B$

$S \cup F \cup PREF \subseteq S \cup B$

بستار ستاره ای (Σ^*): اگر Σ یک الفبا باشد. Σ^* مجموعه کلیه رشته هایی است که می‌توان با Σ تولید نمود. این مجموعه که شامل رشته ای بطول صفر (λ) نیز می باشد بر خلاف Σ، همیشه نامتناهی ولی شمارا است.

نکته: طبق تعریف داریم: $\Sigma^+ = \Sigma^* - \{\lambda\}$

$$\Sigma = \{a, b\} \rightarrow \Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \dots\} \rightarrow \Sigma^+ = \{\Sigma^* - \{\lambda\}\}$$

نکته : $\emptyset$ یک زبان تهی است ولی زبان $L=\{\lambda\}$ شامل یک رشته تهی می‌باشد پس

$$l_1 = \emptyset \quad \text{و} \quad l_2 = \{\lambda\} \rightarrow l_1 \neq l_2$$

نکته: عملگر الحاق نسبت به اجتماع خاصیت پخش‌پذیری دارد اما نسبت به اشتراک خاصیت پخش‌پذیری ندارد. مثلاً اگر سه رشته U,V, W را داشته باشیم.

$u^*(v \cup w) = u^*v \cup u^*w$

$u^*(v \cap w) \neq u^*v \cap u^*w$

نکته: مجموعه Σ^+ نیز مانند Σ^*، نامتناهی اما شمارا است.

۱-۳ زبان Language

اگر Σ یک الفبا باشد آنگاه هر زیر مجموعه ای از Σ^* را یک زبان روی آن الفبا گوییم و آن را با L نمایش می دهیم.

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

$l_1 = \{a, b, aa, bb\}$, $l_2 = \{\text{رشته ها یی به طول زوج}\}$

$l_3 = \{a^n b | n \geq 4\}$, $l_3 = \{a^{n+1} b^m | n \geq 0, m = n\}$

پس یک زبان می تواند مانند l_1 متناهی باشد، یا می تواند مانند l_2 نامتناهی باشد.

یعنی مجموعه زیر مجموعه های یک مجموعه نامتناهی (در اینجا Σ^*) می تواند متناهی یا نامتناهی باشد.

۱-۳-۱ عملگرهای قابل تعریف بر روی یک زبان

۱- **الحاق** دو زبان l_1, l_2 که به صورت زیر تعریف می گردند:

$$l = l_1 * l_2 = \{w | w = xy, x \in l_1 , y \in l_2\}$$

مثلاً اگر $l_1 = \{a, ba\}$ و $l_2 = \{a, b, bb\}$ باشد داریم:

$$l_3 = l_1 * l_2 = \{aa, ab, abb, baa, bab, babb\}$$

$$l_4 = l_2 * l_1 = \{aa, aba, ba, bba, bba, bbba\} \rightarrow l_1 * l_2 \neq l_2 * l_1$$

نکته:

$$\begin{cases} l_1 * l_2 \neq l_2 * l_1 \\ , \\ |l_1 * l_2| \leq |l_1| * |l_2| \end{cases}$$

نکته: در عمل الحاق { λ}=L عضو خنثی می باشد و∅=Lعضو صفراست. یعنی

$$\forall L \subseteq \Sigma^* : \mathrm{L} * \{\lambda\} = \{\lambda\} * \mathrm{L} = \mathrm{L} \qquad \forall \mathrm{L} \subseteq [^*: L\emptyset = \emptyset L = \emptyset$$

نکته:

$$l_1(l_2 \cup l_3) = l_1 l_2 \cup l_1 l_3$$

$$l_1(l_2 \cap l_3) \neq l_1 l_2 \cap l_1 l_3$$

۲- **توان**: به صورت زیر تعریف می گردد:

$$\forall L \subseteq \Sigma^* : \forall \mathrm{i} \in N : l^0 = \{\lambda\}, l^{i+1} = l^i * l$$

۳- **متمم**: به صورت زیر تعریف میگردد:

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

نکته: اگر زبانی متناهی باشد متمم آن قطعاً نامتناهی خواهد بود و اگر زبان L نامتناهی باشد متمم آن می تواند متناهی ویا نامتناهی باشد.مثلاً اگر زبانی به طول رشته های زوج را در نظر بگیریم متمم آن نامتناهی است و اگر زبانی باطول رشته های بیش از ۳را در نظر بگیریم؛ متمم آن قطعاً متناهی خواهد بود.

۴- **بستار ستاره ای** یا (Star closure, Kleene star)

مجموعه تمام توان های از صفر تا بینهایت زبان L که آنرا با l^* نمایش میدهیم.

$$L^* = l^0 \cup l^1 \cup l^2 \cup \ldots = \bigcup_{i=0}^{\infty} l^i \qquad \text{که} \qquad l^0 = \{\lambda\}$$

نکته: λ عضو ثابت بستار کلیه زبانها می باشد:

$$\forall L \subseteq \Sigma^*: \lambda \in L^*$$

λ در بستار زبان تهی نیز موجود است :

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

۵- **بستار تراگذری**(Transitive closure) زبان L را با l^+ نمایش داده و عبارتست از مجموعه کلیه رشته هایی که با الحاق یک یا چند رشته از L بوجود می آیند:

$$L^+ = l^1 \cup l^2 \cup l^3 \cup \ldots = \bigcup_{i=1}^{\infty} l^i$$

نکته : اگر الفبای خود زبان L شامل λ باشد؛ آنگاه L^+ نیز شامل λ خواهد بود یعنی $l^+ = l^*$ می‌شود.

ولی در غیر این صورت L^+ ؛ شامل λ نخواهد بود. اما L^* همیشه شاملλ می باشد.

نکته : روابط زیر همواره برقرار هستند:

$$\begin{cases} \emptyset^+ = \emptyset \\ l^+ = ll^* \\ l^* = l^+ \cup \{\lambda\} \end{cases}$$

$$\begin{cases} (l^*)^* = l^* \\ (l^*)^+ = (l^+)^* = l^* \\ l^*l^* = l^* \end{cases}$$

$$(\overline{l})^* \neq \overline{(l^*)} \qquad \forall\, l_1, l_2: l_1 \subseteq l_2 \rightarrow l_1^* \subseteq l_2^*$$

$$(l_1 l_2)^* \neq l_1^* l_2^* \quad , \quad (l_1 \cup l_2)^* \neq l_1^* \cup l_2^*$$

$$(l_1 \cap l_2)^* \neq l_1^* \cap l_2^* \quad , \quad (l_1 - l_2)^* \neq l_1^* - l_2^*$$

۶- **اجتماع** :

$$l_1 \cup l_2 = \{w: w \in l_1 \; or \; w \in l_2\}$$

۷- **اشتراک** :

$$l_1 \cap l_2 = \{w: w \in l_1 \; And \; w \in l_2\}$$

۸- **تفاضل** :

$$l_1 - l_2 = \{w: w \in l_1 \; And \; w \notin l_2\}$$

۹- **معکوس** :

$$l^R = \{w^R: w \in l\}$$

نکته: دو زبان l_1, l_2 با هم مساویند اگر و فقط اگر زیر مجموعه یکدیگر باشند :

$$if \; (l_1 \subseteq l_2) \; and (l_2 \subseteq l_1) then \; l_1 = l_2$$

نسبت راست دو زبان (l_1/l_2):

$$\forall l_1, l_2 \subseteq \Sigma^*: l_1/l_2 = \{x \in \Sigma^* | xy \in l_1, y \in l_2\}$$

نکته: اگر زبان $\bar{l}$ شامل λ باشد ،$l/\bar{l}$ شامل کلیه رشته های زبان Lخواهد بود.

$$\forall\, l,\bar{l} \subseteq \Sigma^*, \lambda \in \bar{l} \rightarrow l \subseteq l/\bar{l}$$

۱-۴ گرامر (Grammer)

یگ گرامر یک چهار گانه بصورت$G = (V, S, P, T)$ میباشد بطوری که در آن V الفبای گرامر است و مجموعه ای از پایانه ها و غیر پایانه هاست. Tیک مجموعه محدود از ترمینال ها یا همان الفبای زبان می باشد، به طوری که $S \subset V$ و T متغیر شروع بوده و $P,S \in V$ نیز مجموعه مولد ها یا همان قوانین تولیدی می باشد . P رابطه ایست متناهی که به صورت زیر تعریف می گردد:

$$p: (V \cup T)^+ \rightarrow (V \cup T)^*$$

نکته: یک گرامر بصورت زیر تعریف می گردد:

$$L(G): \left\{ w \in l^* \middle| s \overset{*}{\Rightarrow} w \right\}$$

۱-۵ اشتقاق

به این معنی می باشد که از متغیر شروع گرامر (S) در طی چند مرحله (صفر یا بیشتر) جایگزینی قوانین، به یک رشته از زبان آن گرامر برسیم.

نکته: چنانچه رشته $u\ x\ z$ را داشته باشیم که $x \in (V \cup T)^*$ و نیز $u, z \in (V \cup T)^*$ و قانون به صورت $x \rightarrow y$ داشته باشیم آنگاه : $UXZ \overset{*}{\Rightarrow} UYZ$ را یک اشتقاق گوییم.

اشتقاق بر دو نوع است :

۱- **سمت چپ ترین اشتقاق**: اگر در هر مرحله، سمت چپ ترین متغیر را در عبارتها جایگزین کنیم گوییم اشتقاق چپ داریم.

۲- **سمت راست ترین اشتقاق**: اگر در هر مرحله سمت راست ترین متغیر را در عبارتها جایگزین کنیم گوییم اشتقاق راست داریم .

نکته : رابطه $\Rightarrow$ (رابطه اشتقاق) فاقد خواص بازتابی وتراگذری است اما بستار تراگذاری این رابطه که با نماد $\overset{+}{\Rightarrow}$ نشان داده می شود ،دارای خاصیت ترا گذاری می باشد ،ونیز بستار بازتابی وتراگذاری رابطه اشتقاق که با نماد $\overset{*}{\Rightarrow}$ نشان داده می شود دارای خواص بازتابی وتراگذاری می باشد.

- درخت تجزیه یا درخت اشتقاق (Derivation Tree or Parse Tree):

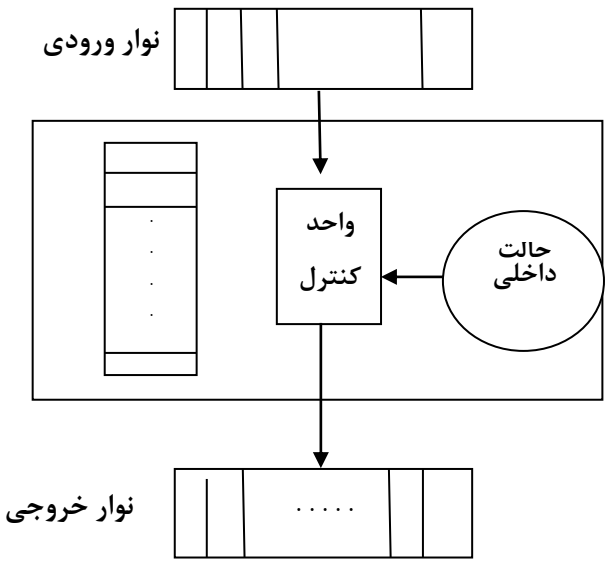
جهت نمایش مراحل اشتقاق در گرامرهای مستقل از متن ،می توان از درخت تجزیه استفاده کرد.

به عبارتی درخت تجزیه روشی مناسب جهت تشخیص توانایی ایجاد رشته توسط یک گرامر است.

فصل دوم : ماشین آتاماتا

۲-۱ ماشین‌های آتاماتا (Automata)

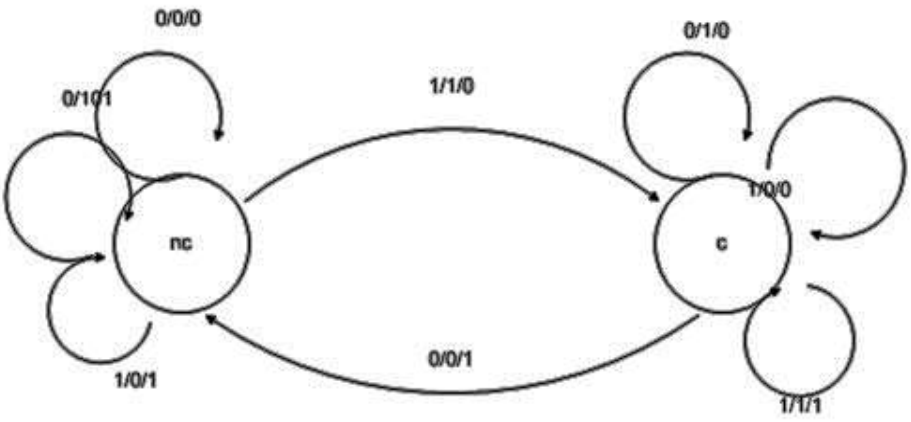
یک آتاماتا، ماشینی است که دارای یک واحد کنترل بوده ودر هر لحظه در یک حالت داخلی قرار دارد.این ماشین می تواند در هر لحظه یک ورودی را خوانده وپس از پردازش، یک خروجی تولید نماید وحالت داخلی خود را نیز تغییر دهد.



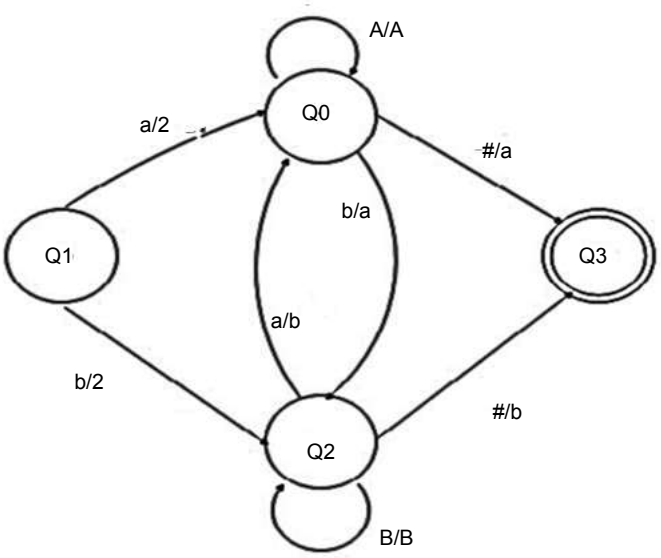
۲-۲ پذیرنده Acceptor

- ماشین آتاماتایی است که خروجی آن "بله" یا "خیر" می باشد.
- این ماشین رشته ای را به عنوان ورودی دریافت کرده و تعیین می کند آیا به زبان آن ماشین تعلق دارد یا خیر؟
- یک پذیرنده فقط نوار ورودی داشته ونوار خروجی ندارد.

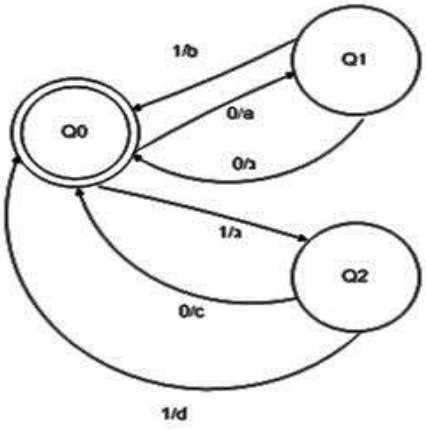
ماشین تراگذری طراحی کنید که دو عدد دودویی را باهم جمع نماید:



- یک تراگذر طراحی کنید که رشته ورودی روی $(\Sigma = \{a, b\})$ را که به # ختم می‌شود را دریافت و آن را با یک تاخیر در خروجی چاپ نماید.



- یک متن از حروف a ,b ,c ,d تشکیل شده است. تراگذاری طراحی کنید که این متن کد شده به رشته ای از ۱ و ۰ را دریافت و رمزگشایی نماید.



a	0	0
b	0	1
c	1	0
d	1	1

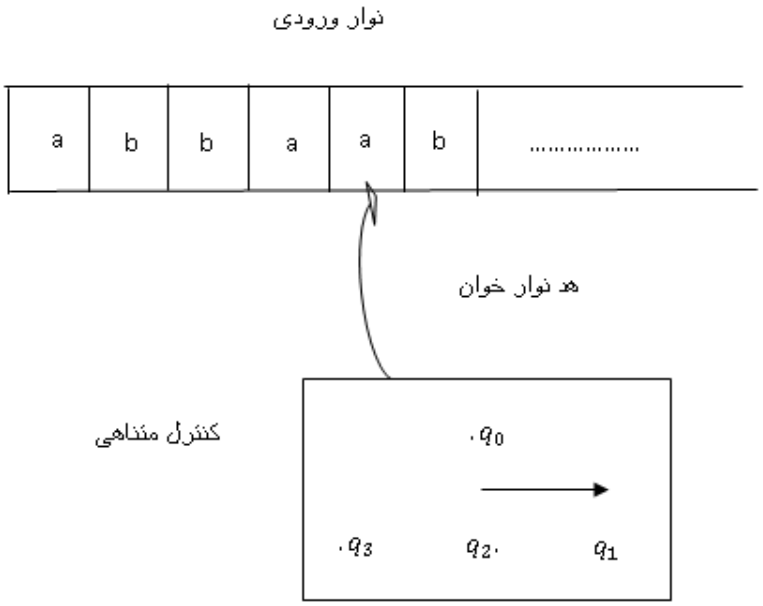
۲-۳ ماشین متناهی (Finite Automata)

- ماشین های متناهی (FA) در حقیقت ماشین های پذیرنده زبان های منظم می باشند.
- چنانچه برای یک زبان بتوان یک ماشین متناهی تعریف نمود گوئیم آن زبان منظم است.
- ماشین متناهی به دو دسته اصلی ۱- **ماشین متناهی قطعی** (Deterministic) ، ۲- **ماشین های متناهی غیر قطعی** (None Deterministic) تقسیم می شوند.

۲-۳-۱ ماشین های متناهی قطعی (Deterministic Finite Automata)

یک ماشین DFA از سه بخش اصلی تشکیل شده است که مطابق شکل عبارتند از:

نوار ورودی ،هد نوار خوان و کنترل متناهی



نکته: در این ماشین نوار ورودی از سمت چپ متناهی واز سمت راست نامتناهی است

این ماشین بصورت زیر تعریف می گردد:

$M = (\Sigma, Q, F, q_0, S)$

Σ←الفبای ورودی

Q←مجموعه حالات

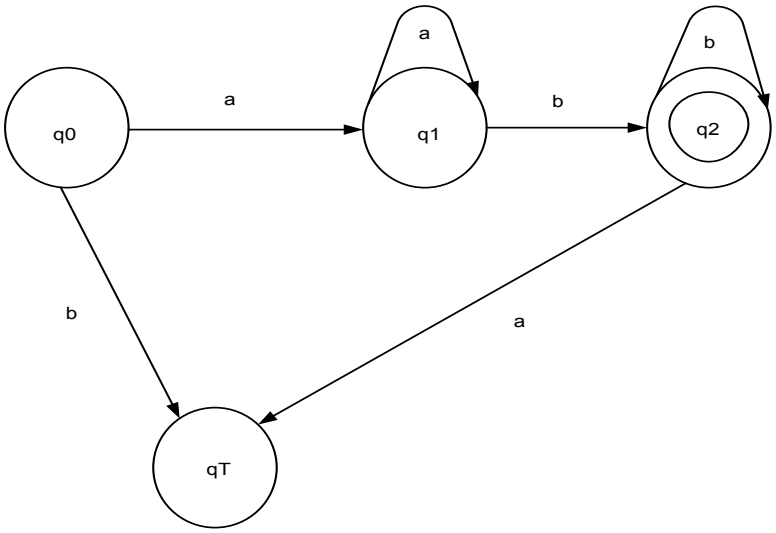
F←مجموعه حالات پذیرش$(F \subseteq Q)$

$q_0 \in Q$← حالت شروع

S←تابع انتقال $Q * \Sigma \to Q$

مثال : برای زبان $L = \{a^n b^m | n, m \geq 1\}$ ماشین DFA زیر تعریف می گردد:

$\Sigma = \{a, b\}, Q = \{q_0, q_1, q_2\}$



در این مثال q_0 حالت شروع است و حالت q_2 نیز حالت نهایی و پذیرش می باشد که با q_f نشان داده شده است. حالت دام یا Trap نیز که با q_t آمده است حالتی است که اگر ماشین وارد آن شود دیگر به هیچ حالت دیگری نرفته و در آن گیرمی افتد. این حالت نباید جزء حالات پذیرش باشد.

زبان پذیرفته شده توسط ماشین M عبارتست از:

$$l(M) = \{w \in \Sigma^* | \sigma^*(q_0, w) \in F\}$$

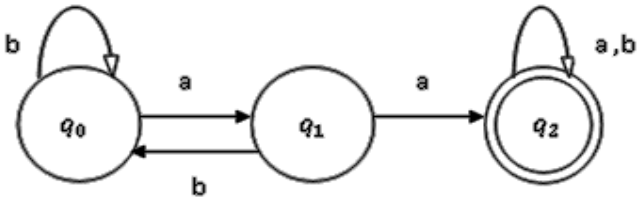
که در آن :

$$\sigma^*\left(q_0, aw\right) = \sigma^*\left(\sigma\left(\hat{q_0}, a\right), w\right), \sigma^*: Q * \Sigma^* \to Q$$

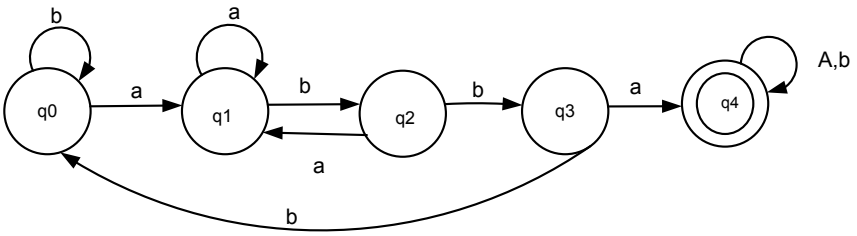
۲-۳-۲ یک دسته بندی غیر رسمی از ماشین های آتاماتای متناهی قطعی

ماشین هایی که دارای زیر رشته خاصی هستند: با فرض $(\Sigma = \{a, b\})$

ماشینی طراحی کنید که رشته هایی را بپذیرد که دارای زیر رشته a a باشند.



ماشینی طراحی کنید که رشته هایی را بپذیرد که دارای زیر رشته a b b a باشد



ماشینی طراحی کنید که رشته هایی را بپذیرند که زیر رشته خاصی را نداشته باشند.

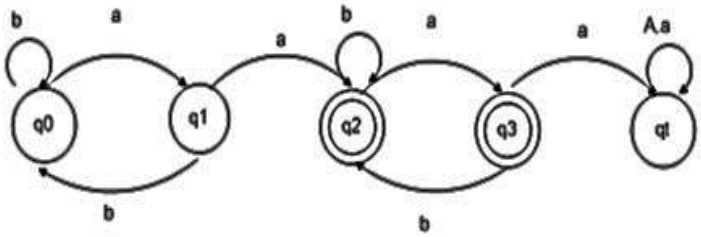
در اینجا می توان حالت قبلی را رسم کرد و سپس جای حالات پذیرش و غیر پذیرش را عوض نمود.

قضیه: چنانچه ماشین $M = (\Sigma, Q, F, q_0, \delta)$ زبان L را بپذیرد، آنگاه ماشین $\overline{M}$

$\overline{M} = (\Sigma, Q, Q - F, q_0, \delta)$ است ، زبان $\overline{L}$ (متمم L) را می پذیرد.

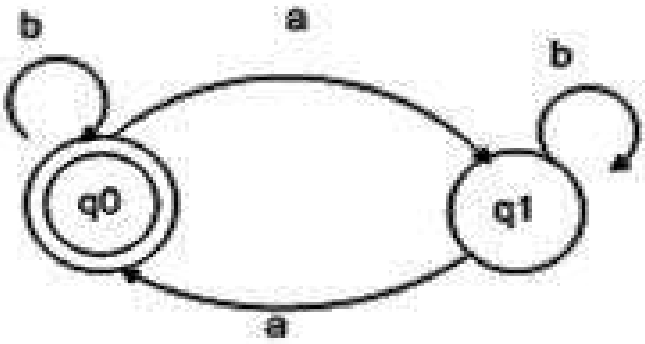
ماشین هایی که رشته هایی را بپذیرند که از یک رشته خاص , فقط یک نمونه داشته باشند.

ماشینی طراحی کنید که رشته هایی را بپذیرد که فقط دارای یک زیر رشته aa باشند

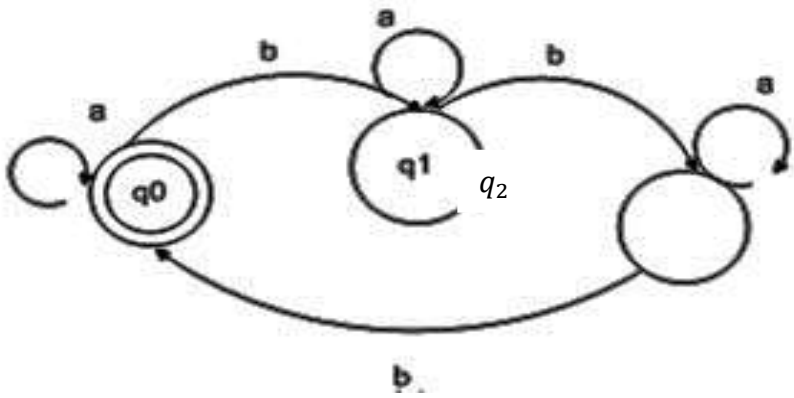


ماشین هایی که رشته هایی را بپذیرند که تعداد عناصر رشته های خاصی بر عدد معینی , بخش پذیر باشد.

ماشینی طراحی کنید که رشته هایی را بپذید که تعداد a ها زوج باشد.

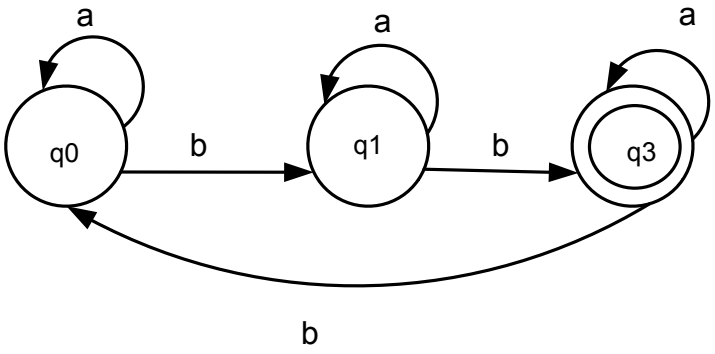


ماشینی طراحی کنید که رشته هایی را بپذیرد که تعداد bها بر۳ بخش پذیر باشد.

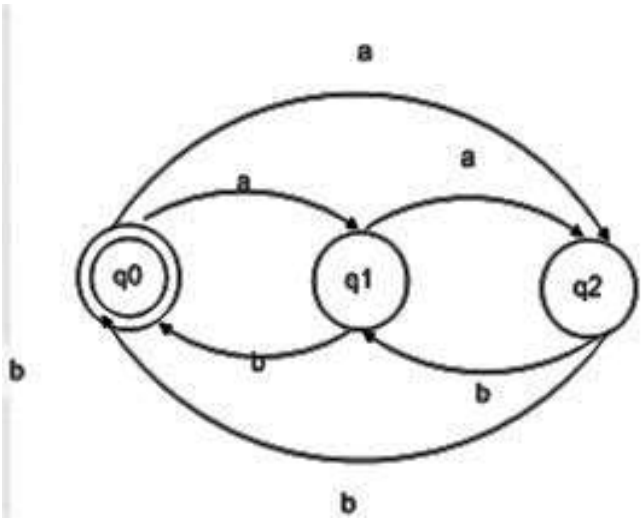


ماشینی مناسب برای هر یک از عبارات زیر تعریف و طراحی کنید

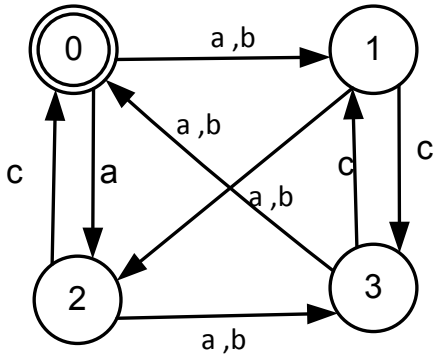
$$l_1 = \{w \in \Sigma^* = \{a,b\}^* | n_b(w)\ mod\ 3 > 1\}$$



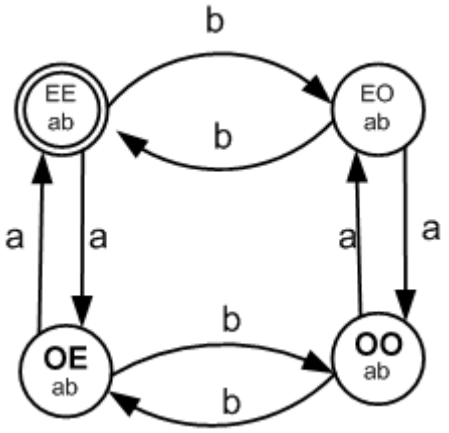
$$l_2 = \{w \in \Sigma^* = \{a,b\}^* | (n_a(w) - n_b(w)\ mod\ 3 = 0)\}$$



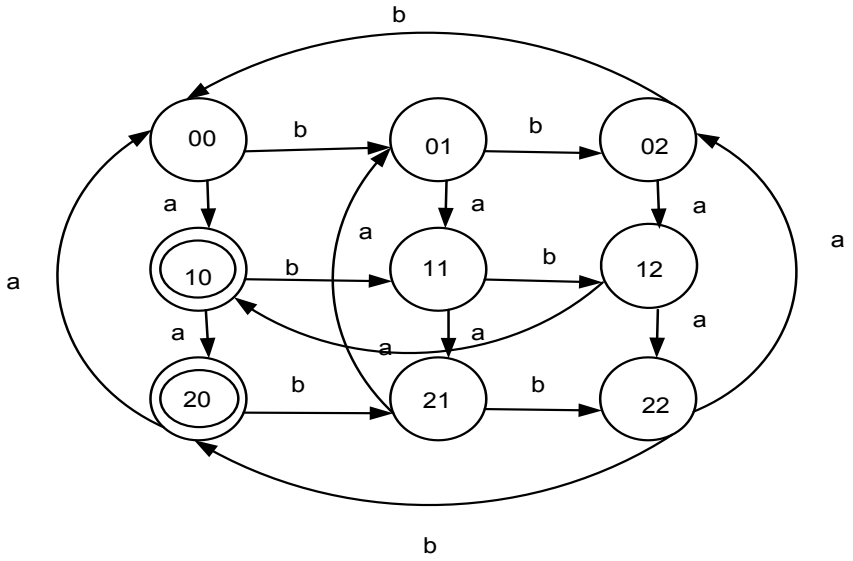
$$L_3 = \{wc\ \Sigma^* = \{a,b\}^* \big| (n_a(w) + n_b(w) + 2n_c(w))\ mod\ 4 = 0\}$$



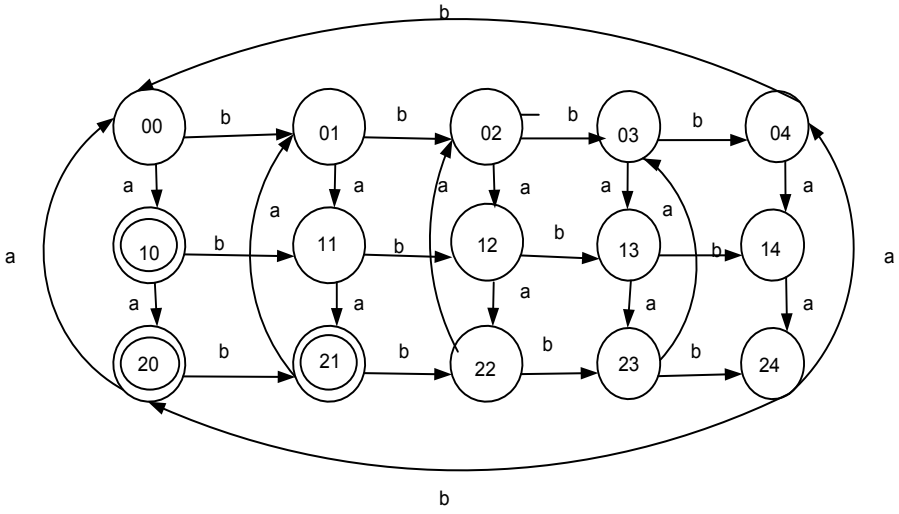
$$L_4 = \{wc \mid \Sigma^* = \{a,b\}^* \big| \text{تعداد } b\text{ها فرد ، تعداد } a\text{ها زوج}\}$$



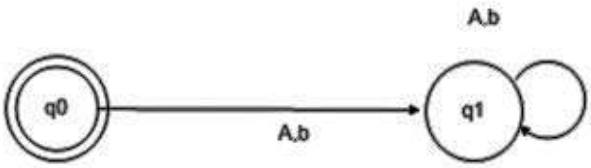
$$l_5 = \{wc\Sigma^* = \{a,b\}^* | (n_a(w)\ mod\ 3 >)\ (n_b(w)\ mod\ 3)\}$$



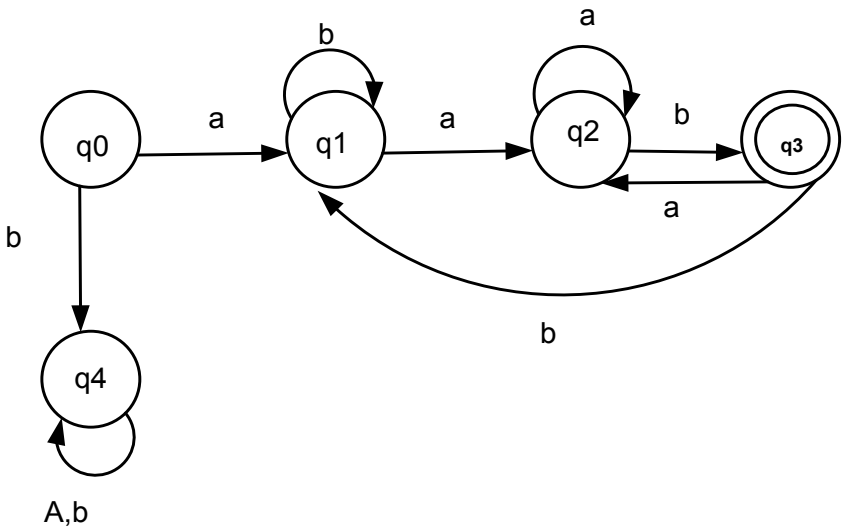
$$l_5 = \{wc\Sigma^* = \{a,b\}^* | (n_a(w)\ mod\ 3 >)\ (n_b(w)\ mod\ 5)\}$$



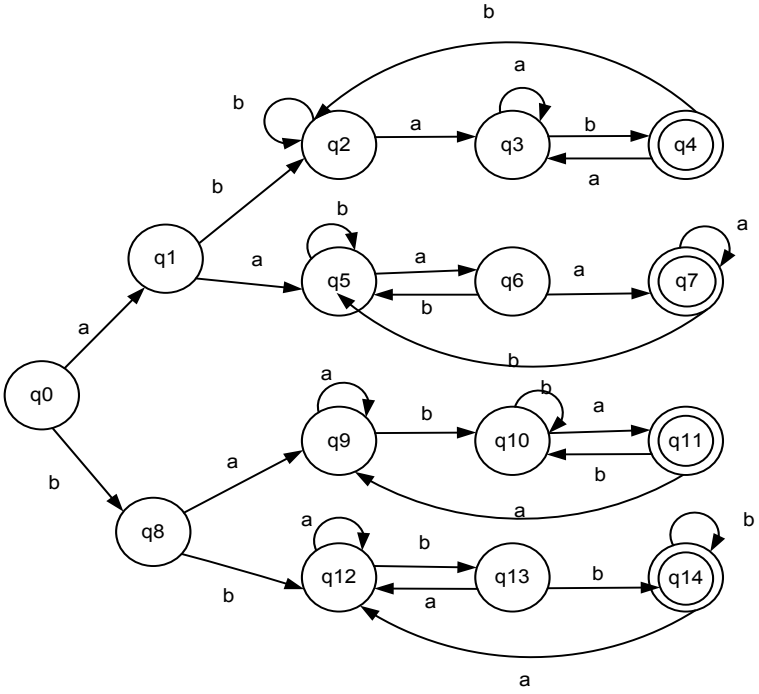
برای زبان تهی یک DFA رسم کنید.



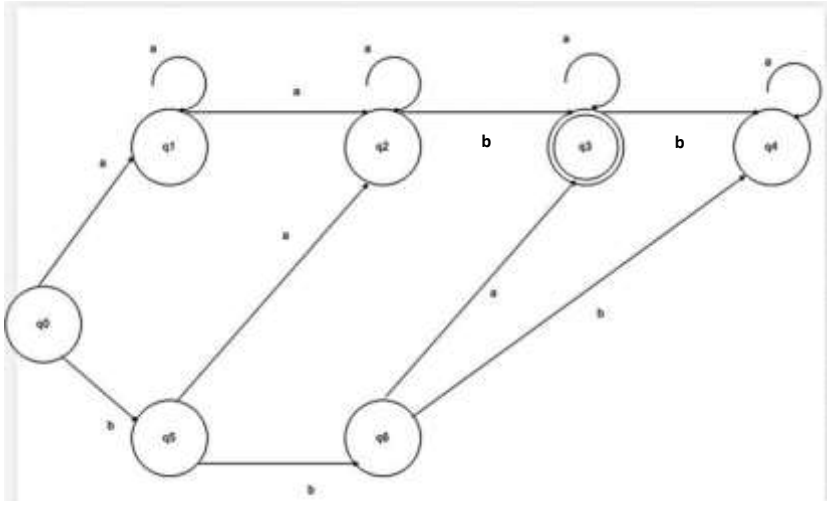
اگر $l = \{awab | w \in \Sigma^* = \{a, b\}^*\}$ یک DFA برای آن رسم کنید.



برای $l = \{uwu|w,u \in \{a,b\}^*, |u| = 2\}$ یک DFA رسم کنید

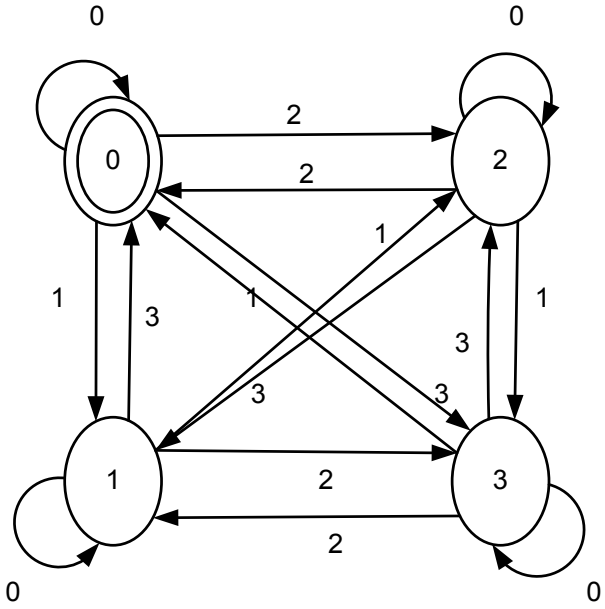


ماشینی طراحی که رشته هایی را بپذیرد که حداقل یک a و دقیقاً دو عدد b داشته باشند.



رشته هایی را ماشین بپذیرد که الفبای آن ارقام بوده و مجموع ارقام رشته بر عدد خاصی بخش پذیر باشد.

با الفبای Σ={0,1,2,3,} ماشینی بسازید که مجموع ارقام رشته های پذیرفته شده در آن بر ۴ بخش پذیر باشد.



نکته: اگر زبان L که دارای یک DFA معادل نا محدود باشد، قطعاً DFA آن دارای سیکل میباشد بطوریکه از یکی از گره های موجود در سیکل ،مسیری به یکی از حالات نهایی وجود خواهد داشت. البته از حالت شروع نیز حتماً مسیری به یکی از گره های درون سیکل وجود دارد.

۲-۴ ماشین‌های آتاماتای متناهی غیرقطعی (None Deterministic Finite Automata)

NFA مانند DFA است اما تابع آن بصورت زیر تعریف می گردد:

$$M = (\Sigma, Q, q_0, F, \delta) \rightarrow \delta: Q * (\Sigma U\{\lambda\}) \rightarrow 2^Q$$

اگر که در آن داریم :

2^Q مجموعه زیر مجموعه های Q می باشد.

۲-۵ تفاوت‌های ماشین آتاماتای متناهی قطعی و غیر قطعی

۱) در یک انتقال می توان هیچ حرفی را پردازش نکرد.

$$\delta(q_i, \lambda) = \{q_j\}$$

۲) می توان تحت σ و از یک حالت بایک ورودی به چندین حالت مختلف رفت .

$$\delta(q_i, a) = \{q_j, q_k\}$$

۳) در NFA می توان از یک حالت با یک ورودی، هیچ انتقالی نداشته باشیم.

$$\delta(q_i, a) = \emptyset$$

نکته: در NFA رشته هایی پذیرفته می شوند که حداقل یک مسیر برای رسیدن به حالت پذیرش موجود می باشد.

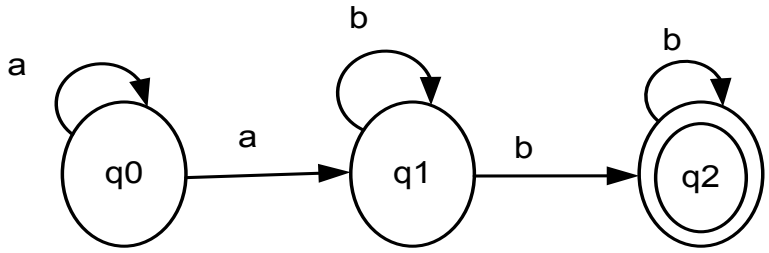
زبان پذیرفته شده توسط ماشین NFA مانند M عبارتست از :

$$L(M) = \{W|\delta^*(q_0, W)N\,F \neq \emptyset\}$$

در NFA هنگامی رشته ای پذیرفته می شود که پس از پایان پردازش رشته، به حالت پذیرش برسیم. اگر وسط پردازش رشته توقف(HALT) کنیم (یعنی به ازای آن ورودی از حالت مورد نظر هیچ انتقالی نداشته باشیم) حتی اگر در حالت پذیرش نیز باشیم ،آن رشته پذیرش نمی شود به این وضعیت اصطلاحاً پیکربندی مرده یا Dead Configuration گویند.

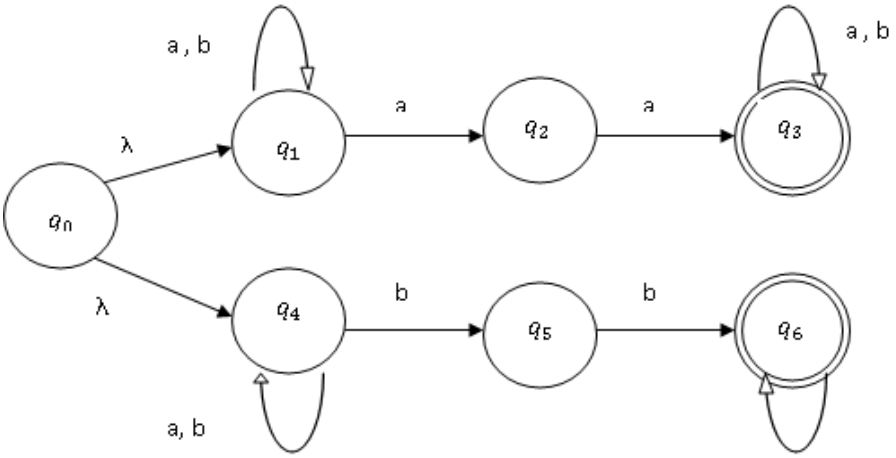
نکته : ماشین های امروزی ، همگی DFA هستند بنابراین جهت ساخت یک ماشین NFA بایستی از تکنیک "Back Tracking" استفاده کرد.

ماشین NFA معادل زبان $L = \{a^n, b^m|N, M \geq 1\}$ برابر است با:

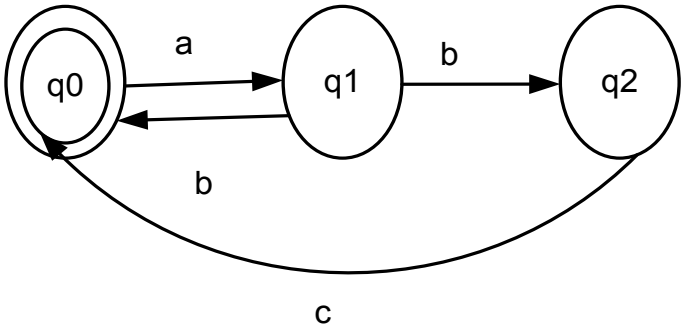


ماشین *NFA* معادل زبان زیر را طراحی کنید.

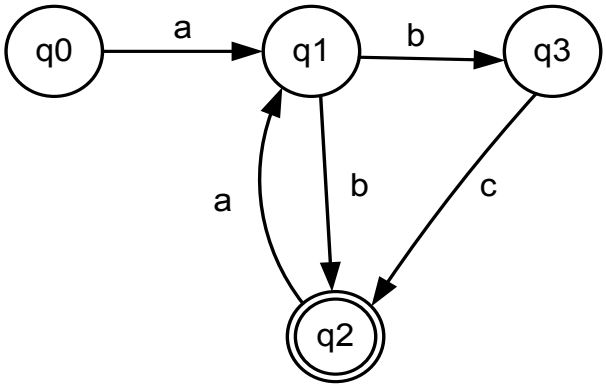
زبان حاوی رشته های است که دارای زیر رشته *a a* یا*bb* باشند.



ماشین NFA با ۳ حالت طراحی کنید که رشته های زبان $\{ab, abc\}^*$ را بپذیرد.



ماشین *NFA* با ۴ حالت طراحی کنید که رشته های زبان $\{ab, abc\}^+$ را بپذیرد.



نکته: اگر NFA دارای چندین حالت شروع باشد می توان با اضافه کردن یک حالت شروع به آن و عمل (λ-Transition) از آن حالت به حالت شروع به NFA با حالت معمولی رسد.

تساوی دو ماشین (هم ارزی): دو ماشین متناهی M_2 و M_1 را زمانی هم ارز یا معادل (equivalent) گوییم که زبان تولید شده توسط آنها باهم یکسان باشد یعنی

$$L(M_1) = L(M_2)$$

قضیه: ماشین‌های NFA , DFA با یکدیگر معادل هستند یعنی الگوریتمی وجود دارد که می‌تواند یک NFA را به DFA معادل تبدیل کند. یعنی DFA ای که همان NFA موردنظر را پذیرش نماید.

نکته: هر DFA یکNFA است اما هر NFA یک DFA نیست بلکه قابل تبدیل به DFA می‌باشد.

نکته: از نکات بالا قابل درک است که DFA،NFA دارای قدرت یکسانی هستند.

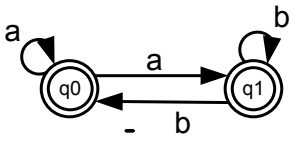
نکته: زمان لازم برای پذیرش یا عدم پذیرش رشته W با طول n توسط NFA متناسب با $O(n^2)$ است این در حالی است که در DFA این زمان متناسب با O(n) می باشد.

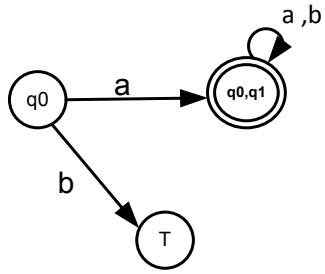
پس DFAها به مراتب بسیار بهینه تر از NFAها می باشند.

نکته: اگر در NFAتعداد حالات Kباشد در DFAمعادل آن ،این تعداد حداکثر 2^K خواهد بود.

۲-۶ تبدیل ماشین آتاماتای متناهی غیرقطعی به قطعی

ماشین NFA زیر را به DFA معادل آن تبدیل کنید.

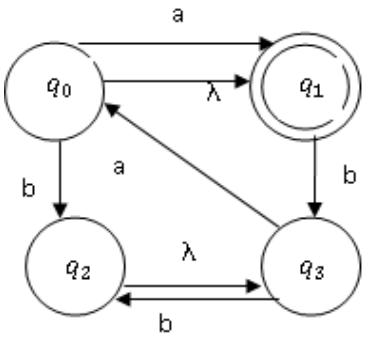


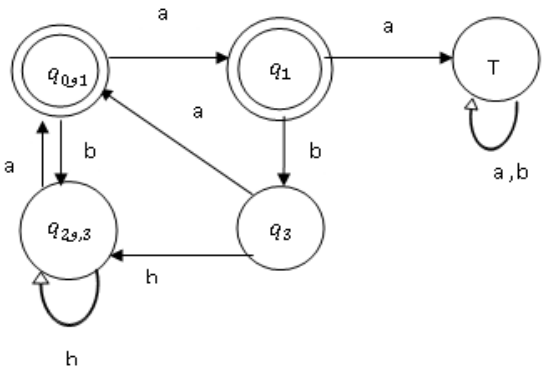


حالت	a	b
Q۰	(q۰,q۱)	ø
Q۱	ø	(q0,q1)

حالت	a	b
q۰	(q۰,q۱)	ø
q۱	ø	(q0,q1)
q۰,	(q۰,q۱)	(q۰,q۱)

❖ DFA معادل با NFA زیر طراحی کنید.





حالات	a	b
Q_0	(q_0,q_1)	q_2
Q_1	Ø	q_3
Q_2	q_0	(q_2,q_3)
Q_3	q_0	(q_2,q_3)

حالت	a	b
q_0	(q_0,q_1)	q_2
q_1	Ø	q3
q_2	q_0	(q_2,q_3)
q_3	q_0	(q_2,q_3)
(q_0,q_2)	q_1	(q_2,q_3)
(q_2,q_3)	q_0	(q_2,q_3)