# فصل ششم

# نظریه محاسبه

# (Theory of Computation)

## ۶-۱: مقدمات

در این فصل تلاش بر این است که تا اندازه‌ای موضوع نظریه محاسبه را بیان نماییم در واقع سعی بر این خواهد بود که به سوالات زیر پاسخ داده شود.

۱. خواص ریاضی نرم افزار و سخت افزارهای کامپیوتر چیست؟
۲. محاسبه و الگوریتم چیست؟ آیا می‌توان تعاریف دقیقی برای این مفاهیم ارائه کرد؟
۳. محدودیت‌های کامپیوتر چیست؟ آیا هر چیزی محاسبه پذیر است؟

### ۶-۱-۱: هدف از نظریه محاسبه

هدف تدوین مدل‌های ریاضی رسمی است که دنیای واقعی کامپیوترها را نمایان سازند. این زمینه از تحقیق توسط ریاضیدانان و اهل منطق در دهه ۱۹۳۰ شروع شد، زمانی که آنها تلاش می‌کردند تا معنی محاسبات (computation) را درک نمایند.

سوال اصلی این بود که آیا همه مسائل ریاضی را می‌توان با روش‌های سیستماتیک حل کرد؟ همین تحقیقات بود که در آن زمان به پیدایش کامپیوتر منجر شد. این روزها، نظریه محاسبه می‌تواند به سه زمینه زیر تقسیم گردد:

الف) نظریه پیچیدگی (Complexity theory)

ب) نظریه محاسبه پذیری (Computability theory)

ج) نظریه اتوماتا (Automata theory)

### ۶-۱-۲: نظریه پیچیدگی

سوال اصلی در این زمینه عبارت است از؛ چه عواملی باعث می‌شوند که محاسبه پذیری برخی مسائل سخت و برخی دیگر آسان انجام شود؟

در بیان ساده و غیر رسمی، مسأله‌ای را آسان گوییم هرگاه به راحتی حل پذیر باشد. مثلاً منظم کردن اعداد تا ۱۰۰۰۰۰۰، پیدا کردن نام در یک فهرست تلفن، پیدا کردن سریع‌ترین راه برای رانندگی از نقطه A تا B (از شهری به شهر دیگر). از طرف دیگر مسأله را سخت گوییم هرگاه، راه‌حل کارا و ساده‌ای نداشته باشد و یا اصلاً ندانیم که چنین راه‌حلی برایش وجود دارد. به عنوان نمونه تجزیه یک عدد صحیح ۳۰۰ رقمی به عوامل اول.

موضوع اصلی- دسته‌بندی مسائل بر طبق میزان سختی آنها. ارایه یک اثبات دقیق از این گونه مسائل که سخت به نظر می‌آیند واقعاً سخت است.

### ۶-۱-۳: نظریه محاسبه پذیری

در سال‌های ۱۹۳۰ به بعد، Gödel ،Turing و Church کشف کردند که برخی از مسائل پایه‌ای ریاضی با کامپیوتر قابل حل نیست (این موضوع ممکن است قدری عجیب باشد، زیرا کامپیوترها در حدود سال ۱۹۴۰ اختراع شده‌اند). مثالی از این نوع عبارتست از؛ آیا یک گزاره دلخواه ریاضی درست و یا نادرست است؟

برای وارد شدن در این موارد به تعاریف رسمی از مفاهیم زیر نیاز داریم.

کامپیوتر، الگوریتم و محاسبه

مدل‌های نظری که برای درک حل پذیری و یا حل ناپذیری مسائل مورد نیاز بود منجر به توسعه کامپیوترهای واقعی گردید.

موضوع اصلی در این زمینه عبارتست از: دسته بندی مسائل از نقطه نظر حل پذیری و حل ناپذیری.

### ۶-۱-۴: نظریه اتوماتا

نظریه اتوماتا با تعاریف و خواص انواع مختلف مدل‌های محاسباتی سر و کار دارد؛ نظیر:

- اتوماتای متناهی؛ که در پردازش متن، کامپایلرها و طراحی سخت افزار بکار می‌روند.

- گرامرهای مستقل از متن (Context- Free Grammars): این‌ها برای تعریف زبان‌های برنامه نویسی و هوش مصنوعی به کار می‌روند.
- ماشین تورینگ (Turing Machine): این ماشین، یک مدل ساده محض از یک کامپیوتر واقعی نظیر PC، ارایه می‌کند.

موضوع اصلی: آیا همه این مدل‌ها دارای توانایی یکسان‌اند؟ یا برخی از آنها قادرند مسائل بیشتری را حل کنند.

## ۶-۲: روش‌های اثبات کردن

در ریاضیات، یک قضیه گزاره‌ای است که درست می‌باشد. یک اثبات دنباله‌ای از گزاره‌های ریاضی است که با شکل‌گیری خاص نشان می‌دهند یک قضیه درست است. سوال اصلی این است که چگونه برای اثبات اقدام کنیم؟ این سوال شبیه سوالی است که در زمان حل مسأله‌ای مطرح می‌کنیم. روشن است که پاسخ این سوالات آسان نیست وگرنه همه مسائل حل‌پذیر بودند. اگرچه راه مشخصی برای رسیدن به اثبات در حالت کلی وجود ندارد، اما راه‌هایی را می‌تواند بیان کرد که اثبات کردن را آسان‌تر پیگیری می‌کند. در این جا برخی موارد را بیان می‌کنیم.

۱. قضیه‌ای را که می‌خواهید اثبات کنید خوب بخوانید و درک نمایید. شاید در اغلب موارد این کار از همه سخت‌تر باشد.
۲. گاهی در داخل قضیه مورد اثبات قضایای دیگری وجود دارند که باید به خوبی تفکیک شوند.
۳. تلاش کنید چند حالت ساده و خاص از قضیه را پیدا کنید.
۴. زمانی که به اثبات رسیدید آن را با دقت بنویسید. گاهی در زمان نوشتن به اشتباهاتی برخورد می‌کنیم که پیش از آن مورد توجه نبوده‌اند.
۵. پیدا کردن اثبات زمان لازم دارد، باید صبور بود، فکر کرد، دقیق نوشت، و تلاش نمود که هرچه بیشتر به قطعیت رسید.

### ۶-۲-۱: اثبات‌های مستقیم (Direct Proofs)

همان طوری که از کلمه مستقیم انتظار داریم، اثبات قضیه در این روش باید سرراست باشد. مثال زیر را بررسی می‌کنیم.

**قضیه:** اگر $n$ یک عدد صحیح، مثبت و فرد باشد، آنگاه $n^۲$ نیز چنین است.

**اثبات:** چون $n$ فرد است پس $n = ۲k + ۱$ که $k \geq ۰$ آنگاه داریم:

$$n^۲ = (۲k+۱)^۲ = ۴k^۲ + ۴k + ۱ = ۲(۲k^۲ + ۲k) + ۱$$

حال چون $۲(۲k^۲ + ۲k)$ زوج است پس $n^۲$ فرد است.

## ۶-۲-۲: اثبات‌های ساختاری (Constructive Proofs)

در این روش نه تنها وجود یک شئی معین نشان داده می‌شود، بلکه روش پیدا کردن آن نیز ارایه می‌گردد. به عنوان مثال:

**قضیه:** شئی با خاصیت $P$ وجود دارد.

**اثبات:** شئی را نشان می‌دهیم که در خاصیت $P$ صدق کند.

## ۶-۲-۳: اثبات‌های غیر ساختاری (Non Constructive Proofs)

در این روش نشان داده می‌شود که شئی معین وجود دارد، اما روش تولید آن را بررسی نمی‌کند. به عنوان مثال:

**قضیه:** اعداد اصم (گنگ) $x$ و $y$ وجود دارند، به طوری که $x^y$ گویا است.

**اثبات:** دو حالت ممکن را بررسی می‌کنیم.

حالت ۱: $\sqrt{۲}^{\sqrt{۲}} \in Q$

در این حالت فرض می‌کنیم $x = y = \sqrt{۲}$. بعداً ثابت می‌کنیم که $\sqrt{۲}$ اصم است.

حالت ۲: $\sqrt{۲}^{\sqrt{۲}} \notin Q$

در این حالت فرض می‌کنیم $x = \sqrt{۲}^{\sqrt{۲}}$ و $y = \sqrt{۲}$. پس:

$$x^y = (\sqrt{۲}^{\sqrt{۲}})^{\sqrt{۲}} = \sqrt{۲}^{۲} = ۲$$

یعنی ادعای قضیه درست است.

توجه کنیم که این روش قضیه را اثبات کرده است، اما روش تولید زوج اعداد اصم x و y را ارائه نمی‌دهد.

### ۶-۲-۴: اثبات‌ها با تناقض (Proofs by Contradiction)

در این روش فرض می‌کنیم می‌خواهیم درستی گزاره S را ثابت کنیم. با استفاده از تناقض فرض می‌کنیم S~ درست باشد، یعنی نقیض S را فرض می‌گیریم سپس با تدابیری که در پیش می‌گیریم نادرست بودن S~ را نشان می‌دهیم، پس S درست خواهد بود. به عنوان مثال:

**قضیه:** ثابت کنید $\sqrt{۲}$ یک عدد اصم است، یعنی $\sqrt{۲}$ نمی‌تواند به صورت خارج قسمت دو عدد صحیح m و n نوشته شود (گویا نیست).

**اثبات:** قضیه را با روش تناقض اثبات می‌کنیم. یعنی فرض می‌کنیم $\sqrt{۲}$ گویا است، پس می‌توان نوشت $\sqrt{۲} = m/n$ که $m \geq ۱$ , $n \geq ۱$ و عامل مشترک ندارند. آنگاه اعمال ساده ریاضی را به صورت زیر انجام می‌دهیم:

$۲n^۲ = m^۲$ پس $m^۲$ زوج است و لذا m نیز زوج است و فرض می‌کنیم $m = ۲k$ که k عدد صحیح مثبتی است. در نتیجه $۲n^۲ = m^۲ = ۴k^۲$ و $n^۲ = ۲k^۲$ که زوج بودن n را نشان می‌دهد. پس m و n هر دو زوج‌اند و این اختلاف فرض است، زیرا m و n عامل مشترک نداشته‌اند. در نتیجه $\sqrt{۲}$ اصم است.

### ۶-۲-۵: اثبات‌ها به استقراء

استقراء یکی از قوی‌ترین و مهم‌ترین روش‌های اثبات قضایا است. بدین ترتیب بیان می‌شود که به ازای هر عدد صحیح n، فرض می‌کنیم P(n) گزاره‌ای وابسته به n باشد. می‌خواهیم ثابت کنیم P(n) به ازای همه اعداد صحیح مثبت n درست است. روش بدین صورت انجام می‌گیرد:

گام پایه‌ای: ثابت می‌کنیم که $P(۱)$ درست است.

گام استقرایی: ثابت می‌کنیم به ازای هر $n \geq ۱$، گزاره زیر برقرار است. اگر P(n) درست باشد، آنگاه $P(n+۱)$ نیز درست است. P(n) را فرض استقراء گوییم. به عنوان مثال:

**قضیه:** ثابت کنید به ازای هر عدد صحیح و مثبت n داریم:

$$۱+۲+۳+...+n=\frac{n(n+۱)}{۲}$$

**اثبات:** فرض می‌کنیم $n=۱$، بدیهی است تساوی درست است.

حال فرض می‌کنیم $n \geq ۱$ و قضیه برای n درست باشد، یعنی

$$۱+۲+۳+...+n=\frac{n(n+۱)}{۲}$$

باید ثابت کنیم،

$$۱+۲+۳+...+(n+۱)=\frac{(n+۱)(n+۲)}{۲}$$

اثبات را به صورت زیر انجام می‌دهم:

$$\begin{aligned}۱+۲+۳+...+(n+۱) &= ۱+۲+۳+...+n+(n+۱)\\ &= \frac{n(n+۱)}{۲}+(n+۱)\\ &= \frac{(n+۱)(n+۲)}{۲}\end{aligned}$$

یادآوری می‌کنیم که موضوع استقراء در فصل سوم به طور کامل و دقیق توضیح داده شده است و در اینجا به دلیل بیان روش‌های مختلف اثبات بار دیگر و به طور خلاصه بیان شد.

## ۶-۳: اتوماتای متناهی و زبان‌های منظم (با قاعده)

## (Finite Automata and Regular Languages)

در این فصل رده‌ای از زبان‌ها را مورد تجزیه و تحلیل قرار می‌دهیم که آنها را زبان‌های منظم نامند. این زبان‌ها با کامپیوترهای با حافظه کمتر پردازش می‌شوند. موضوع را با یک مثال شروع می‌کنیم.

### مثال ۱: کنترل در واژه عوارض

می‌خواهیم ماشینی (کامپیوتری) طراحی کنیم که عمل کنترل عوارض را انجام دهد.

زمانی که یک خودرو به دروازه عوارضی می‌رسد دروازه بسته است. دروازه زمانی باز می‌شود که راننده مبلغ ۲۵ (واحد پول) پرداخت نماید. فرض می‌کنیم تنها سه سکه ۵، ۱۰، ۲۵ داشته باشیم و همچنین فرض می‌کنیم که پول‌های اضافی پرداخت شده بازگشت داده نمی‌شود.

پس از رسیدن خودرو به دروازه، راننده تعدادی سکه وارد ماشین می‌کند. این ماشین در هر لحظه باید تصمیم بگیرد که دروازه را باز کند یا خیر، یعنی آیا راننده مبلغ ۲۵ (یا بیشتر) را وارد کرده است یا نه. ماشین برای تصمیم گیری در هر یک از شش حالت زیر قرار می‌گیرد.

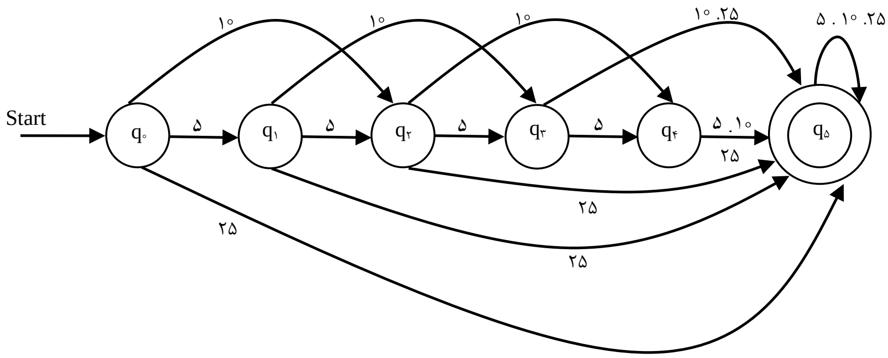
- ماشین در حالت $q_۰$ است، اگر هیچ پولی دریافت نکرده باشد.
- ماشین در حالت $q_۱$ است، اگر دقیقاً مبلغ ۵ دریافت کرده باشد.

- ماشین در حالت $q_۲$ است، اگر دقیقاً مبلغ ۱۰ دریافت کرده باشد.
- ماشین در حالت $q_۳$ است، اگر دقیقاً مبلغ ۱۵ دریافت کرده باشد.
- ماشین در حالت $q_۴$ است، اگر دقیقاً مبلغ ۲۰ دریافت کرده باشد.
- ماشین در حالت $q_۵$ است، اگر دقیقاً مبلغ ۲۵ دریافت کرده باشد.

در آغاز یعنی وقتی که خودرو به دروازه می‌رسد، ماشین در حالت $q_۰$ است. فرض کنیم که راننده مجموعه سکه‌های (۱۰، ۵، ۵، ۱۰) را داشته باشد.

- ماشین پس از دریافت اولین سکه ۱۰، از حالت $q_۰$ به حالت $q_۲$ انتقال می‌یابد.
- ماشین پس از دریافت اولین سکه ۵، از حالت $q_۲$ به حالت $q_۳$ انتقال می‌یابد.
- ماشین پس از دریافت دومین سکه ۵، از حالت $q_۳$ به حالت $q_۴$ انتقال می‌یابد.
- ماشین پس از دریافت دومین سکه ۱۰، از حالت $q_۴$ به حالت $q_۵$ انتقال می‌یابد.

در این لحظه دروازه باز می‌شود (بخاطر داشته باشید که سکه‌های اضافی پس داده نمی‌شود). شکل زیر همه رفتارهای ماشین را در ازای همه حالات نشان می‌دهد. حالت $q_۵$ با دو دایره تودرتو نمایانده شده است، زیرا حالت خاص است (پذیرش). به محض اینکه ماشین به این حالت می‌رسد دروازه باز می‌شود.



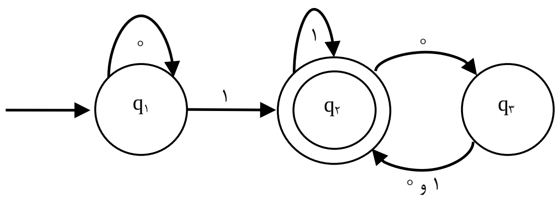
مشاهده می‌شود که ماشین (کامپیوتر) تنها باید بخاطر داشته باشد که در هر لحظه در کدام حالت قرار دارد. پس تنها به یک مقدار کمی حافظه نیاز دارد. باید بتواند میان هر یک از شش حالت تمایز قایل شود، بنابراین حافظه‌ای به اندازه

$$[\log ۶] = ۳\text{bits}$$

کافی است.

### ۶-۳-۱: اتوماتای متناهی معین (Deterministic finite automata)

به مثالی دیگر توجه می‌کنیم؛ نمودار زیر را در نظر می‌گیریم:




گوییم $q_۱$ حالت شروع و $q_۲$ حالت پذیرش است. رشته ۱۱۰۱ را به عنوان ورودی در نظر می‌گیریم. این رشته به صورت زیر پردازش می‌شود.

- در آغاز ماشین در حالت شروع $q_۱$ است.
- ماشین پس از خواندن اولین ۱، از حالت $q_۱$ به حالت $q_۲$ می‌رود.
- ماشین پس از خواندن دومین ۱، از حالت $q_۲$ به حالت $q_۲$ می‌رود (در واقع انتقالی انجام نمی‌گیرد).
- ماشین پس از خواندن اولین ۰، از حالت $q_۲$ به حالت $q_۳$ می‌رود.
- ماشین پس از خواندن سومین ۱، از حالت $q_۳$ به حالت $q_۲$ می‌رود.

پس از پردازش همه رشته ۱۱۰۱ ماشین در حالت $q_۲$، که حالت پذیرش است، قرار می‌گیرد. گوییم رشته ۱۱۰۱ توسط ماشین پذیرفته شده است.

حال رشته ۰۱۰۱۰۱۰ را در نظر می‌گیریم. پس از خواندن این رشته از چپ به راست (با شروع از حالت $q_۱$)، ماشین در حالت $q_۳$ قرار می‌گیرد. چون $q_۳$ حالت پذیرش نیست، گوییم ماشین رشته ۰۱۰۱۰۱۰ را رد می‌کند (نمی‌پذیرد). امید است توجه کرده باشید که این ماشین هر رشته دودویی را که به ۱ ختم می‌شود، می‌پذیرد. در واقع ماشین رشته‌های بیشتری را می‌پذیرد، به صورت زیر:

- هر رشته‌ای که دارای تعداد زوج ۰ باشد که پس از سمت راست‌ترین ۱ می‌آید.
- هر رشته دودویی دیگر توسط این ماشین رد می‌شود. چنین رشته‌هایی عبارتند از رشته تهی، رشته‌ی شامل فقط ۰ها و رشته‌هایی که تعداد ۰های آنها که بعد از سمت راست‌ترین ۱ قرار می‌گیرد، فرد است.

حال تعریف رسمی اتوماتای متناهی را بیان می‌کنیم:

**تعریف ۶-۳-۲:** یک اتوماتای متناهی عبارتست از یک ۵ تایی $M=(Q,\Sigma,\delta,q,F)$ که در آن:

۱. $Q$ یک مجموعه متناهی است، عناصرش را حالت‌ها می‌نامند.
۲. $\Sigma$ یک مجموعه متناهی است که آن را الفبا (alphbet) نامند، عناصرش را نمادها گویند.
۳. $\delta: Q\times\Sigma\rightarrow Q$ تابعی است که آن را تابع انتقال نامند.
۴. $q$ عنصری از $Q$ است، آن را حالت شروع نامند.
۵. $F$ یک زیر مجموعه از $Q$ است، عناصرش را حالت‌های پذیرش نامند.

می‌توان به تابع انتقال به عنوان یک برنامه‌ی ماشین متناهی $M=(Q,\Sigma,\delta,q,F)$ نگاه کرد. این تابع می‌گوید که $M$ در یک گام چه عملی انجام می‌دهد.

- فرض کنیم $r$ یک حالتی از $Q$ و $a$ نمادی از $\Sigma$ باشد. اگر اتوماتای متناهی $M$ در حالت $r$ باشد و $a$ را بخواند، آنگاه از $r$ به حالت $\delta(r,a)$ انتقال می‌یابد (در حقیقت $\delta(r,a)$ می‌تواند با $r$ مساوی باشد).

کامپیوتری را که برای دروازه عوارضی طراحی کردیم یک اتوماتای متناهی است. برای آن مثال داریم $Q=\{q_۰,q_۱,q_۲,q_۳,q_۴,q_۵\}$ و $\Sigma=(۵,۱۰,۲۵)$، حالت شروع $q_۰$ است و $F=\{q_۵\}$ و $\delta$ به صورت جدول زیر عمل می‌کند.

| | ۵ | ۱۰ | ۲۵ |
|---|---|---|---|
| $q_۰$ | $q_۱$ | $q_۲$ | $q_۵$ |
| $q_۱$ | $q_۲$ | $q_۳$ | $q_۵$ |
| $q_۲$ | $q_۳$ | $q_۴$ | $q_۵$ |
| $q_۳$ | $q_۴$ | $q_۵$ | $q_۵$ |
| $q_۴$ | $q_۵$ | $q_۵$ | $q_۵$ |
| $q_۵$ | $q_۵$ | $q_۵$ | $q_۵$ |

مثالی که در آغاز این بخش بیان کردیم نیز یک اتوماتای معین است. برای آن داریم $F=\{q_۲\}$ و حالت شروع: $q_۲$، $\Sigma=\{۰,۱\}$، $Q=\{q_۱,q_۲,q_۳\}$، $\delta$ با جدول زیر توصیف می‌شود.

| | ۰ | ۱ |
|---|---|---|
| $q_۱$ | $q_۱$ | $q_۲$ |
| $q_۲$ | $q_۳$ | $q_۲$ |
| $q_۳$ | $q_۲$ | $q_۲$ |

فرض کنیم M یک اتوماتای متناهی باشد. زبان M را با L(M) نشان می‌دهیم و گوییم L(M) عبارتست از همه رشته‌های دودویی که توسط M پذیرفته می‌شود و می‌نویسیم:

$$L(M)=\{w \mid \text{شامل حداقل یک ۱ است و به تعداد زوج ۰ها ختم می‌گردد } w\}$$

حال تعریف رسمی زبان یک اتوماتای متناهی را بیان می‌کنیم.

**تعریف ۶-۳-۳:** فرض کنیم $M=(Q,\Sigma,\delta,q,F)$ یک اتوماتای متناهی و $w=w_۱w_۲...w_n$ رشته‌ای از $\Sigma$ باشد. دنباله $r_۰,r_۱,...,r_n$ از حالت‌ها را به صورت زیر تعریف می‌کنیم:

- $r_۰=q,$
- $r_{i+۱}=\delta(r_i,w_{i+۱}),\quad \text{for}\quad i=۰,۱,...,n-۱$

۱. اگر $r_n \in F$، آنگاه گوییم M رشته w را می‌پذیرد.
۲. اگر $r_n \notin F$، آنگاه گوییم M رشته w را رد می‌کند.

در این تعریف، w ممکن است رشته تهی باشد، آن را با $\varepsilon$ نشان می‌دهیم و طول آن صفر است، بنابراین در تعریف بالا $n = 0$. در این حالت دنباله $r_0, r_1, ..., r_n$ از حالت‌ها طول یک دارد و شامل تنها حالت $r_0 = q$ است. رشته تهی M پذیرفته می‌شود اگر و تنها اگر حالت شروع q به F تعلق داشته باشد.

**تعریف ۶-۳-۴:** فرض کنیم $M = (Q, \Sigma, \delta, q, F)$ یک اتوماتای متناهی باشد. زبان L(M) که توسط M پذیرفته می‌شود به این صورت تعریف می‌شود، مجموعه‌ای از همه رشته‌های پذیرفته شده توسط M.

$$L(M) = \{w \mid \text{رشته‌ای بر روی } \Sigma \text{ است و } M \text{ آن را می‌پذیرد } w\}$$

**تعریف ۶-۳-۵:** یک زبان A را منظم گوییم هرگاه یک اتوماتای متناهی M موجود باشد طوری که $A = L(M)$.

این بخش را با بیان یک تعریف معادل از زبان پذیرفته شده بوسیله اتوماتای متناهی، به پایان می‌بریم.

فرض کنیم $M = (Q, \Sigma, \delta, q, F)$ یک اتوماتای متناهی باشد. تابع انتقال $\delta : Q \times \Sigma \to Q$ می‌گوید که وقتی M در حالت $r \in Q$ قرار دارد و نماد $a \in \Sigma$ را می‌خواند، از حالت r به حالت $\delta(r, a)$ انتقال می‌یابد.

فرض کنیم $\Sigma^*$ مجموعه‌ی همه رشته‌های روی الفبای $\Sigma$ را نمایش دهد ($\Sigma^*$ شامل رشته تهی $\varepsilon$ است).

تابع $\delta$ را به صورت تابع $\bar{\delta} : Q \times \Sigma^* \to Q$ گسترش می‌دهیم و به صورت زیر تعریف می‌کنیم.

به ازای هر حالت $r \in Q$ و برای هر رشته w از $\Sigma$ داریم:

$$\bar{\delta}(r,w)=\begin{cases} r \quad \text{if} \quad w=\varepsilon, \\ \delta(\bar{\delta}(r,v)a) \quad \text{if} \quad w=va, \end{cases}$$

$v$ یک رشته و $a\in\Sigma$ است.

معنی تابع $\bar{\delta}$ چیست؟ فرض کنیم $r$ یک حالت از $Q$ و $w$ رشته‌ای از $\Sigma$ باشد، آنگاه:

- $\bar{\delta}(r,w)$ حالتی است که $M$ به آن می‌رسد، وقتی که از حالت $r$ شروع می‌کند و رشته $w$ را از چپ به راست می‌خواند و با استفاده از $\delta$ از حالتی به حالت دیگر می‌رود. پس می‌نویسیم:

$$L(M)=\{w \mid \bar{\delta}(q,w)\in F \ , \ \text{یک رشته بر روی } \Sigma \ w\}$$

### ۶-۳-۶: چند مثال از یک اتوماتای متناهی

#### مثال اول

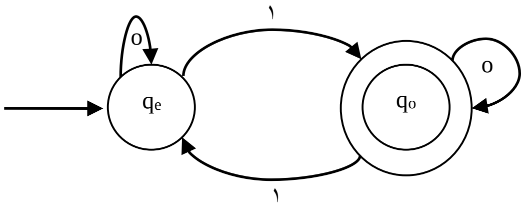
فرض کنیم $\{w$ یک رشته دودویی شامل تعداد فرد ۱ها است $A=\{w \mid$. گوییم این زبان $A$ منظم است. برای اثبات این موضوع، باید یک اتوماتای متناهی $M$ بسازیم که $A=L(M)$. این $M$ را چگونه می‌سازیم؟ ایده اول: اتوماتای متناهی رشته ورودی $w$ را از چپ به راست می‌خواند و تعداد ۱هایی را که دیده است در نظر می‌گیرد. پس از خواندن همه رشته $w$، بررسی می‌کند که آیا تعداد ۱ها فرد است که در این صورت $w$ پذیرفته شده است یا تعداد ۱ها زوج است، که در این حال $w$ پذیرفته نشده است. با این روش، اتوماتای متناهی نیاز به یک حالت برای هر عدد صحیح $i\geq 0$ دارد که بیان کند تعداد ۱هایی خوانده شده تا حالا برابر $i$ است. بنابراین برای طراحی یک اتوماتای متناهی که این روال را انجام دهد، به تعداد نامتناهی حالت نیاز داریم. اما، می‌دانیم که تعریف اتوماتای متناهی به تعداد حالت‌های متناهی نیاز دارد. راه بهتر و درست‌تر این است که با ردیابی مشخص کند که تعداد ۱های خوانده شده تا این لحظه، زوج است یا فرد. این راه اتوماتای متناهی زیر را می‌سازد.

- مجموعه‌ی حالت‌ها $Q=\{q_e,q_o\}$ است. اگر اتوماتای متناهی در حالت $q_e$ باشد، آنگاه تعداد زوج ۱ها را خوانده است و اگر در حالت $q_o$ باشد، آنگاه تعداد ۱های خوانده شده فرد است.
- الفبا عبارتست از $\Sigma=\{0,1\}$.

- حالت شروع برابر $q_e$ است، زیرا در لحظه شروع تعداد ۱های خوانده شده توسط اتوماتا مساوی صفر است و صفر یک عدد زوج می‌باشد.
- مجموعه‌ی حالت‌های پذیرش $F = \{q_o\}$ است.
- تابع انتقال $\delta$ با جدول زیر بیان می‌شود:

| | ۰ | ۱ |
|---|---|---|
| $q_e$ | $q_e$ | $q_o$ |
| $q_o$ | $q_o$ | $q_e$ |

اتوماتای متناهی $M = (Q, \Sigma, \delta, q_e, F)$ را همچنین می‌توان با نمودار حالت‌ها به صورت زیر توصیف کرد. پیکان‌هایی که از دوایر بیرون آمده‌اند و به حالت $q_e$ وارد شده‌اند، بیان می‌کنند که $q_e$ حالت شروع است؛ حالتی که با دو دایره تودرتو مشخص شده بیانگر حالت پذیرش می‌باشد.




پس یک اتوماتای متناهی $M$ درست کردیم که زبان $A$ را می‌پذیرد، لذا $A$ زبان منظم است.

**مثال دوم**

زبان $A$ عبارتست از

$$A = \{w \mid \text{یک رشته دودویی است و ۱۰۱ را به عنوان زیر رشته در بردارد } w\}$$

بار دیگر گوییم $A$ یک زبان منظم است. به بیان دیگر نشان می‌دهیم که یک اتوماتای متناهی $M$ وجود دارد که $A$ را پذیرش می‌کند، یعنی $A = L(M)$.

این ماشین در زمان خواندن رشته ورودی از چپ به راست به ترتیب زیر عمل می‌کند:

- از همه oها عبور می‌کند و در حالت شروع قرار می‌گیرد.
- با دیدن اولین ۱ به حالت "شاید دو نماد بعدی ۰۱ باشد" می‌رود.
  - اگر نماد بعدی ۱ باشد در حالت "شاید دو نماد بعدی ۰۱ باشد" باقی می‌ماند.
  - از طرف دیگر، اگر نماد بعدی ۰ باشد به حالت "شاید نماد بعدی ۱ باشد" می‌رود.
    - ❖ اگر نماد بعدی ۱ باشد، به حالت پذیرش می‌رود (اما خواندن رشته را تا انتها ادامه می‌دهد).
    - ❖ از طرف دیگر، اگر نماد بعدی ۰ باشد به حالت شروع می‌رود و ۰ها را رد می‌کند تا دوباره ۱ را بخواند.

با بیان چهار حالت زیر، پردازش بدیهی خواهد بود.

- $q_1$: اتوماتای M در این حالت قرار می‌گیرد اگر آخرین نماد خوانده شده ۱ باشد، اما زیر رشته‌ی ۱۰۱ هنوز خوانده نشده است.
- $q_{10}$: M در این حالت قرار دارد اگر دو نماد آخر خوانده شده ۱۰ باشد، اما زیر رشته‌ی ۱۰۱ هنوز خوانده نشده است.
- $q_{101}$: M در این حالت قرار دارد اگر زیر رشته ۱۰۱ در رشته‌ی ورودی خوانده شده باشد.
- $q$: در سایر حالات، M در این حالت قرار می‌گیرد.

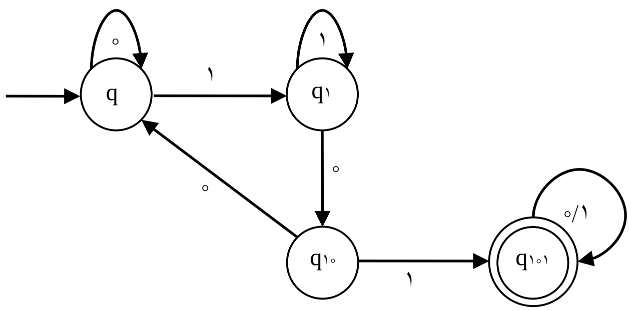
حال توصیف رسمی اتوماتای M را که زبان A را می‌پذیرد به صورت زیر داریم:

- $Q=\{q, q_1, q_{10}, q_{101}\}$ :
- $\Sigma=\{0,1\}$ :
- : حالت شروع $q$ است.
- : مجموعه پذیرش $F=\{q_{101}\}$ است و
- : تابع انتقال $\delta$ با جدول زیر بیان می‌شود.

| | ۰ | ۱ |
|---|---|---|
| $q$ | $q$ | $q_1$ |

| | | |
|---|---|---|
| $q_{۱}$ | $q_{۱۰}$ | $q_{۱}$ |
| $q_{۱۰}$ | $q$ | $q_{۱۰۱}$ |
| $q_{۱۰۱}$ | $q_{۱۰۱}$ | $q_{۱۰۱}$ |

شکل زیر نمودار اتوماتای متناهی $M$ را نشان می‌دهد.




### مثال سوم

اتوماتاهای متناهی که تاکنون دیدیم دقیقاً یک حالت پذیرش داشتند. در این بخش مثالی از یک اتوماتای متناهی می‌بینیم که حالت‌های پذیرش بیشتری دارند. زبان $A$ را چنین در نظر می‌گیریم:

$$A = \{w \in \{۰,۱\}^* \mid \text{w یک ۱ را در مکان سوم از راست دارد}\}$$

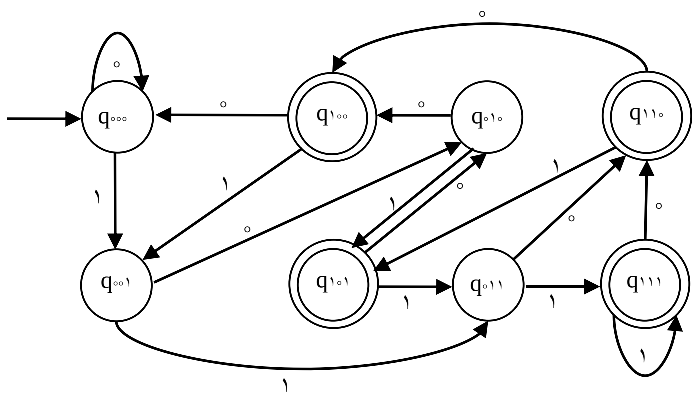
که در آن $\{۰,۱\}^*$ عبارتست از مجموعه‌ی همه رشته‌های دودویی است و شامل رشته تهی $\varepsilon$ نیز می‌باشد. گوییم $A$ یک زبان منظم است. پس باید یک اتوماتای متناهی $M$ درست کنیم که $A = L(M)$.

در ابتدا مشکل و شاید غیر ممکن به نظر آید که بتوانیم یک اتوماتای متناهی بسازیم که بتواند درک کند که به نماد سوم از راست رسیده است، اما خواهیم دید که می‌توان آن را درست کرد. ایده اصلی این است که بتواند بخاطر داشته باشد که سه نماد آخر خوانده شده‌اند. با این ترتیب اتوماتای متناهی هشت حالت $q_{ijk}$ دارد، که $i$، $j$ و $k$ در مجموعه $\{۰,۱\}$ حرکت می‌کنند. اگر اتوماتا در حالت $q_{ijk}$ باشد آنگاه موارد زیر را داریم:

- اگر $M$ حداقل سه نماد را خوانده باشد، آنگاه آخرین سه نماد خوانده شده $ijk$ هستند.

- اگر M تنها دو نماد را خوانده باشد، آنگاه این نمادها jk هستند و به علاوه $i = ۰$.
- اگر M تنها یک نماد را خوانده باشد، آنگاه این نماد k است و به علاوه $i = j = ۰$.
- اگر M هیچ نمادی را نخوانده باشد، آنگاه $i = j = k = ۰$.

حالت شروع $q_{۰۰۰}$ است و مجموعه‌ی پذیرش‌ها عبارتست از $\{q_{۱۰۰}, q_{۱۱۰}, q_{۱۰۱}, q_{۱۱۱}\}$. تابع انتقال اتوماتای M در نمودار زیر نشان داده شده است.




## ۶-۴: اعمال منظم (با قاعده) Regular Operations

در این بخش سه عمل را بر روی زبان‌ها تعریف می‌کنیم. بعد از این موضوع بسته بودن تحت این اعمال را بیان خواهیم کرد.

فرض کنیم A و B دو زبان بر روی یک مجموعه الفبا باشند.

۱. اجتماع A و B عبارت است از: $A \cup B = \{w \mid w \in A \ \text{ or } \ w \in B\}$

۲. اتصال A و B عبارت است از: $AB = \{ww' \mid w \in A \ \text{ and } \ w' \in B\}$

به بیان دیگر AB مجموعه‌ای از همه رشته‌هایی است که با یک رشته دلخواه w در A و یک رشته دلخواه 'w در B و چسباندن آنها به یکدیگر ساخته می‌شود. (توجه داریم که w در سمت چپ 'w است).

۳. ستاره A عبارتست از:

$$A^* = \{u_1u_2...u_k \mid k \geq 0 \text{ and } u_i \in A \ , \ i = 1, 2, ..., k\}$$

می‌توان گفت $A^*$ با هر تعداد متناهی از رشته‌های A و چسباندن آنها به یکدیگر بدست می‌آید. توجه شود که $k = 0$ نیز مجاز است و متناظر با رشته تهی $\varepsilon$ می‌باشد. پس $\varepsilon \in A^*$. برای مثال فرض کنیم $A = \{0, 01\}$ و $B = \{1, 10\}$، آنگاه

$$A \bigcup B = \{0, 01, 1, 10\},$$
$$AB = \{01, 010, 011, 0110\},$$
$$A^* = \{\varepsilon, 0, 01, 00, 001, 010, 0101, 000, 0001, 00101, ...\}$$

به عنوان مثالی دیگر، اگر $\Sigma = \{0, 1\}$، آنگاه $\Sigma^*$ مجموعه‌ی همه رشته‌های دودویی است (شامل رشته تهی).

مشاهده می‌شود که یک رشته همواره دارای یک طول متناهی است.

پیش از ادامه بحث یک تعریف دیگر (و معادل) از ستاره زبان A ارایه می‌کنیم. تعریف می‌کنیم $A^0 = \{\varepsilon\}$ و برای $k \geq 1$، $A^k = AA^{k-1}$، یعنی $A^k$ اتصال دو زبان A و $A^{k-1}$ است. پس داریم $A^* = \bigcup_{k=0}^{\infty} A^k$.

**قضیه ۶-۴-۱:** مجموعه‌ی زبان‌های منظم تحت عمل اجتماع بسته است، یعنی اگر A و B زبان‌های منظم بر روی الفبای $\Sigma$ باشند، آنگاه $A \bigcup B$ نیز زبان منظم است.

**اثبات:** چون A و B زبان‌های منظم هستند، پس اتوماتاهای متناهی $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ و $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ وجود دارند که به ترتیب A و B را پذیرش می‌کنند. برای اثبات این که $A \bigcup B$ منظم است باید یک اتوماتای متناهی M درست کنیم که $A \bigcup B$ را بپذیرد. به عبارت دیگر M باید این خاصیت را داشته باشد که برای هر $w \in \Sigma^*$

"M، w را پذیرش می‌کند اگر و تنها اگر $M_1$ یا $M_2$، w را بپذیرند"

در نگاه اول، می‌توانیم فکر کنیم که M مراحل زیر را می‌تواند انجام دهد:

- با شروع از $q_1$ که حالت شروع $M_1$ است، $M$، $M_1$ را برای $w$ بررسی می‌کند.
- اگر بعد از خواندن $w$، $M_1$ در یکی از حالات $F_1$ باشد، آنگاه $w \in A$. پس $w \in A \bigcup B$ و در نتیجه $M$، $w$ را پذیرش می‌کند.
- از طرف دیگر، اگر پس از خواندن $w$، $M_1$ در حالتی قرار گیرد که در $F_1$ نیست، آنگاه $w \notin A$ و $M$، $M_2$ را برای $w$ بررسی می‌نماید و این کار را با حالت شروع $q_2$ از $M_2$ آغاز می‌کند. اگر پس از خواندن $w$، $M_2$ در حالتی از $F_2$ قرار گیرد آنگاه $w \in B$، در نتیجه $w \in A \bigcup B$ و نتیجه می‌شود که $M$، $w$ را پذیرش می‌کند. در غیر این صورت $w \notin A \bigcup B$ و $M$، $w$ را رد می‌نماید.

اما این روش عملی نیست، زیرا اتوماتای متناهی $M$ رشته ورودی $w$ را تنها یک بار می‌خواند. روش درست این است که $M_1$ و $M_2$ هم زمان بررسی شوند.

حال مجموعه‌ی $Q$ از حالت‌های $M$ را به صورت حاصلضرب کارتزین $Q_1 \times Q_2$ تعریف می‌کنیم. اگر $M$ در حالت $(r_1, r_2)$ قرار گیرد، نتیجه می‌شود که:

- اگر $M_1$ رشته ورودی را تا این لحظه خوانده باشد، آنگاه در حالت $r_1$ قرار دارد و
- اگر $M_2$ رشته ورودی را تا این لحظه خوانده باشد، آنگاه در حالت $r_2$ قرار می‌گیرد.

این به اتوماتای متناهی $M = (Q, \Sigma, \delta, q, F)$ منجر می‌شود که در آن،

- $Q = Q_1 \times Q_2 = \{(r_1, r_2) \mid r_1 \in Q_1 \text{ and } r_2 \in Q_2\}$، مشاهده شود که $|Q| = |Q_1| \times |Q_2|$ که متناهی است.
- $\Sigma$ الفبای $A$ و $B$ است (یادآوری می‌کنیم که فرض کردیم $A$ و $B$ زبان‌هایی بر روی یک الفبا هستند).
- حالت شروع $q$ از $M$ عبارتست از $q = (q_1, q_2)$.
- $F$ مجموعه‌ی حالت‌های پذیرش $M$ برابر است با

$$F = \{(r_1, r_2) \mid r_1 \in F_1 \text{ or } r_2 \in F_2\} = (F_1 \times Q_2) \bigcup (Q_1 \times F_2)$$

- تابع انتقال $\delta : Q \times \Sigma \to Q$ با تساوی زیر بیان می‌شود:

$$\delta((r_1,r_2),a)=(\delta_1(r_1,a),\delta_2(r_2,a))$$

به ازای هر $r_1 \in Q_1$ , $r_2 \in Q_2$ , $a \in \Sigma$.

برای پایان دادن اثبات، باید نشان دهیم که اتوماتای متناهی M در واقع زبان $A \cup B$ را می‌پذیرد. اگرچه این مطلب به طور ذاتی از بحث بالا روشن می‌باشد.

ساده‌ترین راه برای ارایه یک تعریف رسمی این است که از تابع گسترش یافته انتقال $\overline{\delta}_1$ و $\overline{\delta}_2$ استفاده کنیم. (این توابع گسترش یافته را بعداً تعریف می‌کنیم).

در ادامه یادآور می‌شویم که باید ثابت کنیم:

"M، w را می‌پذیرد اگر و تنها اگر $M_1$، w را بپذیرد و یا $M_2$، w را پذیرش نماید."

یعنی،

"M، w را می‌پذیرد اگر و تنها اگر $\overline{\delta}_1(q_1,w) \in F_1$ یا $\overline{\delta}_2(q_2,w) \in F_2$ "

با استفاده از تابع انتقال گسترش یافته $\overline{\delta}$ از تابع انتقال $\delta$ از M داریم:

$$\overline{\delta}((q_1,q_2),w) \in F \Leftrightarrow \overline{\delta}_1(q_1,w) \in F_1 \ \text{ or } \ \overline{\delta}_2(q_2,w) \in F_2 \qquad (۱-۶)$$

با بکارگیری تعریف تابع انتقال گسترش یافته می‌توان دید که:

$$\overline{\delta}((q_1,q_2),w)=(\overline{\delta}_1(q_1,w),\overline{\delta}_2(q_2,w))$$

این تساوی بیان می‌کند که رابطه (۴,۱) درست است و در واقع M زبان $A \cup B$ را می‌پذیرد. حال ببینیم درباره بستار زبان‌های منظم تحت اعمال اتصال و ستاره چه می‌توان گفت؟ نتیجه این است که زبان‌های منظم تحت این اعمال بسته‌اند. اما چگونه می‌شود این موضوع را ثابت کرد؟

فرض کنیم A و B دو زبان با قاعده باشند و $M_1$ و $M_2$ دو اتوماتای متناهی باشند که به ترتیب A و B را می‌پذیرند. چگونه یک اتوماتای متناهی M درست کنیم که AB را پذیرش نماید؟ فرض کنیم u یک رشته ورودی باشد، حال M باید تصمیم بگیرد که آیا u می‌تواند به دو رشته w و w' شکسته شود یا خیر (یعنی بتوانیم بنویسیم $u = ww'$)، به طوری که $w \in A$ و $w' \in B$. به بیان ساده M باید

تصمیم بگیرد که آیا u می‌تواند به دو زیر رشته شکسته شود یا خیر؟ به طوری که رشته اول توسط $M_1$ و رشته دوم توسط $M_2$ پذیرفته شوند. مشکل از این جا ناشی می‌شود که M باید با یک بار اسکن کردن رشته u، تصمیم لازم را بگیرد. اگر $u \in AB$، آنگاه M باید بتواند تصمیم بگیرد که با این یک بار اسکن کردن، u را در کجا به دو زیر رشته بشکند. مشابهاً اگر $u \notin AB$ آنگاه M با همین یک بار اسکن کردن باید نتیجه بگیرد که u نمی‌تواند به دو زیر رشته شکسته شود، به طوری که اولین زیر رشته در A و دومین آن در B قرار گیرد.

به نظر می‌آید اگر A زبان منظم باشد، اثبات این که $A^*$ یک زبان منظم است از اثبات بسته بودن اتصال هم مشکل‌تر خواهد بود. برای اثبات این موارد به یک اتوماتای متناهی نیاز داریم که وقتی یک رشته u به عنوان ورودی در نظر گرفته می‌شود، بتواند تصمیم بگیرد که آیا u می‌تواند به دو زیر رشته شکسته شود، به طوری که هر زیر رشته در A قرار گیرد. مسأله این است که اگر $u \in A^*$، اتوماتای متناهی باید تعیین کند که رشته u به چند تا زیر رشته و در کجا باید شکسته شود؟ یادآور می‌شویم که این امر باید با یک بار اسکن کردن u نتیجه بگیرد.

همان‌طوری که پیش از این تذکر داده شد، اگر A و B زبان‌های منظم باشند، آنگاه هر دوی AB و $A^*$ نیز منظم هستند. برای اثبات این ادعاها نیاز به یک اتوماتای متناهی از نوع کلی وجود دارد.

اوتوماتاهایی را که تا این جا دیدیم از نوع معین (Deterministic) هستند. یعنی:

- اگر اتوماتای متناهی M در حالت r قرار گیرد و اگر نماد a را بخواند، آنگاه M از حالت r به حالت یکتای تعریف شده با $\delta(r,a)$ انتقال می‌یابد. از این جا به بعد این اتوماتای متناهی را یک اتوماتای متناهی معین یا (DFA) که معادل واژه "Deterministic finite automata" است می‌نامیم. سپس مفهوم اتوماتای متناهی نامعین یا (NFA) که معادل واژه "Nondeterministic finite automata" است، را بیان می‌کنیم. برای این نوع اتوماتا حالت‌های صفر یا حالت‌های ممکن بسیاری وجود دارد تا به آنها انتقال پیدا کند. در نگاه اول اتوماتای متناهی نامعین به نظر می‌آید که از نوع DFA قوی‌تر باشد، به هر حال ثابت می‌شود که قدرت آنها مشابه یکدیگرند، با استفاده از این واقعیت اثبات این که رده زبان‌های منظم تحت اعمال اتصال و ستاره بسته است، آسان‌تر خواهد بود.

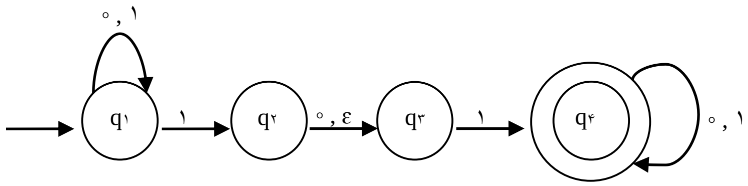
## ۶-۵: اتوماتای متناهی نامعین (Nondeterministic finite automata)

موضوع را با سه مثال شروع می‌کنیم. این مثال‌ها تفاوت بین این نوع اتوماتا را با اتوماتای معین نشان می‌دهد. پس از ارایه این مثال‌ها یک تعریف رسمی از اتوماتای متناهی نامعین بیان می‌کنیم.

### ۶-۵-۱: چند مثال

#### مثال اول

نمودار زیر را در نظر می‌گیریم:




سه تفاوت با اتوماتای متناهی که تا حال دیده‌ایم مشاهده می‌شود. اول: اگر اتوماتا در حالت شروع $q_1$ باشد و نماد ۱ را بخواند، آنگاه دو انتخاب دارد: یا در حالت شروع $q_1$ می‌ماند و یا به حالت شروع $q_2$ می‌رود. دوم: اگر اتوماتا در حالت $q_2$ باشد، آنگاه بدون خواندن یک نماد می‌تواند به حالت $q_3$ انتقال یابد. این امر با پیکانی نشان داده شده است که رشته تهی $\varepsilon$ را به عنوان برچسب دارد. سوم: اگر اتوماتا در حالت $q_3$ باشد و نماد ۰ را بخواند، آنگاه نمی‌تواند ادامه دهد. حال ببینیم این اتوماتا زمانی که رشته ورودی ۰۱۰۱۱۰ را دریافت می‌کند، چه کاری می‌تواند انجام دهد. در آغاز اتوماتا در حالت شروع $q_1$ است.

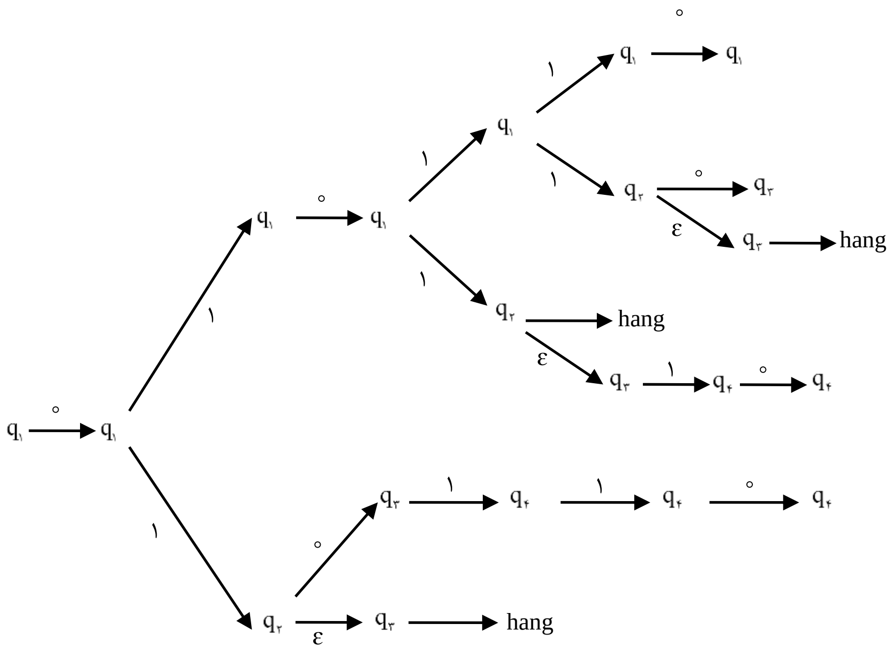
- چون اولین نماد در رشته ورودی ۰ است، اتوماتا پس از خواندن این نماد در حالت $q_1$ می‌ماند.
- نماد دوم ۱ است و اتوماتا می‌تواند یا در حالت $q_1$ بماند و یا به حالت $q_2$ انتقال یابد.

- اگر اتوماتا در حالت $q_1$ بماند، آنگاه پس از خواندن نماد سوم باز هم در این حالت باقی می‌ماند.
- اگر اتوماتا به حالت $q_2$ منتقل شود، آنگاه باز هم دو انتخاب دارد:

❖ یا نماد سوم ورودی را می‌خواند که ۰ است و به حالت $q_3$ می‌رود،

❖ و یا بدون خواندن نماد سوم به حالت $q_3$ می‌رود.

اگر به همین ترتیب ادامه دهیم، آنگاه می‌بینیم که برای رشته ورودی ۰۱۰۱۱۰ هفت محاسبه ممکن وجود دارد. همه این محاسبات در نمودار زیر دیده می‌شوند.




حال پایین‌ترین مسیر را در نمودار بالا در نظر می‌گیریم.

- وقتی نماد اول را می‌خواند، اتوماتا در حالت $q_1$ می‌ماند.
- وقتی نماد دوم را می‌خواند، اتوماتا در حالت $q_2$ می‌ماند.
- اوتوماتا نماد سوم را نمی‌خواند ( به بیان دیگر، رشته تهی $\varepsilon$ را می‌خواند) و به حالت $q_3$ می‌رود.

در این لحظه اوتوماتا نمی‌تواند ادامه دهد. نماد سوم ۰ است، اما مسیری وجود ندارد که از $q_3$

بیرون رود و با برچسب ۰ مشخص شده باشد و مسیری دیده نمی‌شود که از $q_۳$ بیرون رود و برچسب $\varepsilon$ داشته باشد. در نتیجه محاسبات در این نقطه متوقف می‌شود یا اتوماتا هنگ می‌کند.

از این نمودار می‌توان دید که از هفت محاسبه ممکن، دقیقاً دو تا به حالت پذیرش $q_۴$ رفته است، (پس از این که همه رشته‌ی ۰۱۰۱۱۰ خوانده شده است).

گوییم اتوماتا رشته‌ی ۰۱۰۱۱۰ را می‌پذیرد، زیرا حداقل یک محاسبه وجود دارد که به حالت پذیرش می‌رسد.

حال رشته ۰۱۰ را در نظر می‌گیریم. در این حالت سه محاسبه ممکن وجود دارد:

۱. $q_۱ \xrightarrow{۰} q_۱ \xrightarrow{۱} q_۱ \xrightarrow{۰} q_۱$

۲. $q_۱ \xrightarrow{۰} q_۱ \xrightarrow{۱} q_۲ \xrightarrow{۰} q_۳$

۳. $q_۱ \xrightarrow{۰} q_۱ \xrightarrow{۱} q_۲ \xrightarrow{\varepsilon} q_۳ \longrightarrow$ هنگ

هیچ کدام از این محاسبات به حالت پذیرش منتهی نمی‌شوند (پس از خواندن همه رشته‌ی ۰۱۰) پس گوییم اتوماتا رشته‌ی ۰۱۰ را نمی‌پذیرد یا رد می‌کند.

نمودار بالا مثالی از یک اتوماتای متناهی نامعین (NFA) است. به طور غیر رسمی یک NFA یک رشته را می‌پذیرد، اگر حداقل یک مسیر در نمودار موجود باشد که (i) در حالت آغازی شروع نماید، (ii) پیش از خواندن همه رشته هنگ نکند، و (iii) به یک حالت پذیرش برسد.

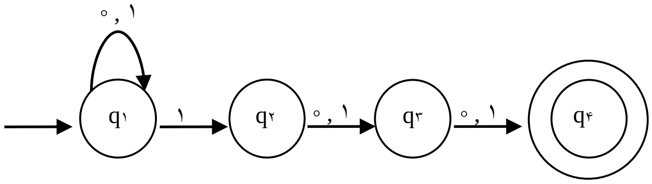
یک رشته‌ای که برای آن (i)، (ii) و (iii) برقرار نباشد، بوسیله NFA رد می‌شود. NFA ای که در بالا ارایه شد، همه رشته‌های دودویی که زیر رشته‌های ۱۰۱ یا ۱۱ را در خود داشته باشند، پذیرش می‌کند؛ رشته‌های دیگر رد می‌شوند.

### مثال دوم

زبان A را به صورت زیر در نظر می‌گیریم:

$$A = \{w \in \{۰,۱\}^* \mid \text{نماد ۱ را در مکان سوم از راست دارا است } w\}$$

نمودار حالتی زیر یک NFA را تعریف می‌کند که همه رشته‌های موجود در A را می‌پذیرد و رشته‌های دیگر را که در A نیستند رد می‌کند.




این NFA چنین عمل می‌کند.

اگر در حالت شروع $q_1$ باشد و نماد ۱ را بخواند، آنگاه یا در حالت $q_1$ می‌ماند و یا حدس می‌زند که این نماد، سومین نماد از راست در رشته ورودی است. در این حال NFA به حالت $q_2$ انتقال می‌یابد و سپس بررسی می‌کند که آیا در رشته ورودی دقیقاً دو نماد باقی مانده است؟ اگر بیش از دو نماد باقی مانده باشد، آنگاه NFA پس از خواندن دو نماد بعدی هنگ می‌کند (در حالت $q_4$).

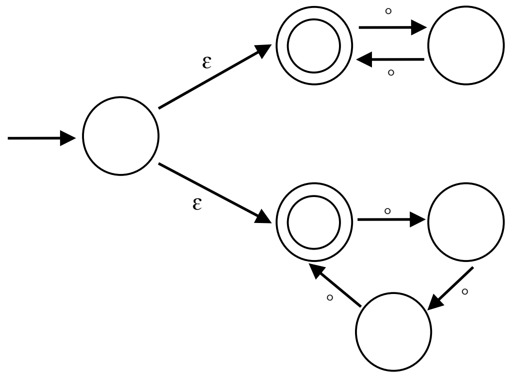
ملاحظه کنید که این روند حدس زدن چگونه به کار می‌رود:

اتوماتا می‌تواند رشته ورودی را تنها یکبار، از چپ به راست، بخواند. از این روی نمی‌داند که چه زمانی به نماد سوم از راست می‌رسد. وقتی NFA یک ۱ را می‌خواند، می‌تواند حدس بزند که این نماد سوم از راست است. پس از دریافت این حدس، تحقیق می‌کند که آیا این حدس درست بود یا نه.

در مثال سوم که پیش از این مطرح گردید، یک DFA برای همین زبان A دیدیم. حال مشاهده می‌شود که NFA یک ساختار ساده‌تر از DFA دارد.

**مثال سوم**

نموادر حالتی زیر را در نظر می‌گیریم. این نمودار یک NFA را با الفبای $\{0\}$ نشان می‌دهد.




این NFA زبان $A=\{o^K \mid K\equiv o \mod ۲ \text{ or } K=o \mod ۳\}$ را می‌پذیرد. در این زبان $o^K$ رشته‌ای شامل K تا o است. (اگر $K=\circ$، آنگاه $o^K=\varepsilon$).

مشاهده کنید که A اجتماع دو زبان زیر است:

$$A_۱=\{o^K \mid K\equiv o \mod ۲\} \text{ و } A_۲=\{o^K \mid K\equiv o \mod ۳\}$$

این NFA اساساً شامل دو DFA است. یکی $A_۱$ را می‌پذیرد و دیگری $A_۲$ را پذیرش می‌کند. فرض کنیم W یک رشته ورودی باشد، NFA باید تصمیم بگیرد که آیا $W\in A$ این تصمیم معادل این است که آیا $W\in A_۱$ یا $W\in A_۲$ و یا خیر. NFA این تصمیم را به صورت زیر اتخاذ می‌کند:

در شروع حدس می‌زند آیا (i) بررسی کند $W\in A_۱$ یا خیر (یعنی، طول W زوج است)، یا (ii) به ببیند که $W\in A_۲$ یا خیر (یعنی، طول W مضربی از ۳ است).

پس از اتخاذ تصمیم، تحقیق می‌کند که آن حدس درست بوده است یا خیر. اگر $W\in A$، آنگاه راهی وجود دارد که بوسیله آن نشان دهد حدس درست بوده است و W واقعاً عضوی از A است (با رسیدن به یک حالت پذیرش). اگر $W\notin A$، آنگاه مهم نیست که کدام حدس زده شده است و NFA هرگز به یک حالت پذیرش نمی‌رسد.

### ۶-۵-۲: تعریف اتوماتای متناهی نامعین

اتوماتای متناهی نامعین نیز تعریف مشابهی با اتوماتای متناهی معین دارد. این دو اتوماتا هر دو دارای تعدادی حالت هستند، یک الفبای ورودی دارند، و دارای یک تابع انتقال، یک حالت شروع و تعدادی حالت

پذیرش می‌باشند. اما آنها در نوع تابع انتقال با هم متفاوت هستند. در DFA تابع انتقال با یک حالت شروع و یک نماد ورودی به حالت بعدی می‌رود، اما در NFA تابع انتقال حالت شروع و نماد ورودی یا نماد تهی را می‌گیرد و مجموعه‌ی حالت‌های ممکن بعدی را تولید می‌نماید.

حال با توجه با ایده‌ای که از مثال‌های قبل بدست آوردیم. یک تعریف رسمی از این اتوماتاها ارایه می‌کنیم. برای هر الفبای $\Sigma$ مجموعه $\Sigma_\varepsilon$ را به صورت $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ تعریف می‌نماییم. با استفاده از مفهوم مجموعه‌ی توانی، برای هر مجموعه‌ی Q، مجموعه‌ی توانی Q را با P(Q) نشان می‌دهیم و می‌نویسیم.

$$P(Q) = \{R \mid R \subseteq Q\}$$

**تعریف ۶-۵-۳:** یک اتوماتای متناهی نامعین (NFA) عبارتست از ۵ تایی $M = \{Q, \Sigma, \delta, q, F\}$ که در آن:

۱. Q یک مجموعه‌ی متناهی است و عناصرش را حالت‌ها نامند.
۲. $\Sigma$ یک مجموعه‌ی متناهی است که الفبا نام دارد و عناصرش را نمادها گویند.
۳. تابع انتقال $\delta$ عبارتست از $\delta : Q \times \Sigma_\varepsilon \to P(Q)$
۴. q عضوی از Q است و حالت شروع نام دارد.
۵. F یک زیرمجموعه از Q است و عناصرش را حالت‌های پذیرش نامند.

مانند DFA ها، تابع انتقال $\delta$ را به عنوان برنامه اتوماتای متناهی $M = \{Q, \Sigma, \delta, q, F\}$ در نظر می‌گیریم.

- فرض کنیم $r \in Q$ و $a \in \Sigma_\varepsilon$. آنگاه $\delta(r, a)$ یک زیرمجموعه از Q (احتمالاً تهی) می‌باشد. اگر NFA ی M در حالت شروع r باشد و a را بخواند (که a ممکن است رشته تهی باشد)، آنگاه M می‌تواند از حالت r به هر حالت $\delta(r, a)$ انتقال یابد. اگر $\delta(r, a) = \phi$، آنگاه M ادامه نمی‌دهد و محاسبات به حالت هنگ در می‌آید. مثال ۶:۱-۴ یک NFA است که در آن $Q = \{q_1, q_2, q_3, q_4\}$ و $\Sigma = \{0, 1\}$. حالت شروع $q_1$ است و مجموعه حالت‌های پذیرش $F = \{q_4\}$ است. تابع انتقال $\delta$ با جدول زیر بیان می‌گردد.

| | ۰ | ۱ | $\varepsilon$ |
|---|---|---|---|
| $q_1$ | $\{q_1\}$ | $\{q_1, q_2\}$ | Q |
| $q_2$ | $\{q_3\}$ | Q | $\{q_3\}$ |
| $q_3$ | Q | $\{q_4\}$ | Q |
| $q_4$ | $\{q_4\}$ | $\{q_4\}$ | Q |

**تعریف ۶-۵-۴:** فرض کنیم $M=\{Q,\Sigma,\delta,q,F\}$ یک NFA باشد و $W\in\Sigma^*$. گوییم M، W را می‌پذیرد، اگر W بتواند به صورت $W=y_1y_2...y_m$ نوشته شود. که در آن $y_i\in\Sigma_\varepsilon$ و $1\le i\le m$، و دنباله‌ی $r_0, r_1, ..., r_m$ از حالت‌های Q طوری موجود باشد که

- $r_0=q$،
- $r_{i+1}\in\delta(r_i, y_{i+1})$، که $0\le i\le m-1$ و
- $r_m\in F$.

در غیر این صورت گوییم M، رشته‌ی W را رد می‌کند. در مثال اول این بخش، NFA رشته‌ی ۰۱۱۰۰ را می‌پذیرد. این موضوع می‌تواند به صورت زیر دیده شود:

- $W=01\varepsilon100=y_1y_2y_3y_4y_5y_6$ و
- $r_0=q_1$، $r_1=q_1$، $r_2=q_2$، $r_3=q_3$، $r_4=q_4$، $r_5=q_4$، $r_6=q_4$.

**تعریف ۶-۵-۵:** فرض کنیم $M=\{Q,\Sigma,\delta,q,F\}$ یک NFA باشد. زبان $L(M)$ که توسط M پذیرفته شده است عبارتست از:

$$L(M)=\{W\in\Sigma^* \mid \text{M، W را می‌پذیرد}\}$$

### ۶-۵-۶: بسته بودن تحت اعمال منظم

**قضیه ۶-۵-۷:** مجموعه‌ی زبان‌های منظم تحت عمل اجتماع بسته است. یعنی اگر $A_1$ و $A_2$ زبان‌های منظم بر روی الفبای $\Sigma$ باشند، آنگاه $A_1\cup A_2$ نیز یک زبان منظم است.

**اثبات:** چون $A_1$ منظم است پس یک NFA مانند $M_1=\{Q_1,\Sigma,\delta_1,q_1,F_1\}$ وجود دارد به طوری که $A_1=L(M_1)$. مشابهاً یک NFA ی $M_2=\{Q_2,\Sigma,\delta_2,q_2,F_2\}$ موجود است که $A_2=L(M_2)$. می‌توانیم فرض کنیم $Q_1\cap Q_2=Q$، زیرا در غیر این صورت نام‌های جدیدی به حالت‌های $Q_1$ و $Q_2$ می‌دهیم. از این دو NFA ها یک NFA مانند $M=\{Q,\Sigma,\delta,q_0,F\}$ می‌سازیم. به طوری که $L(M)=A_1\cup A_2$. شکل ۶-۱ این ساختار را نشان می‌دهد. این NFA به صورت زیر تعریف می‌شود:

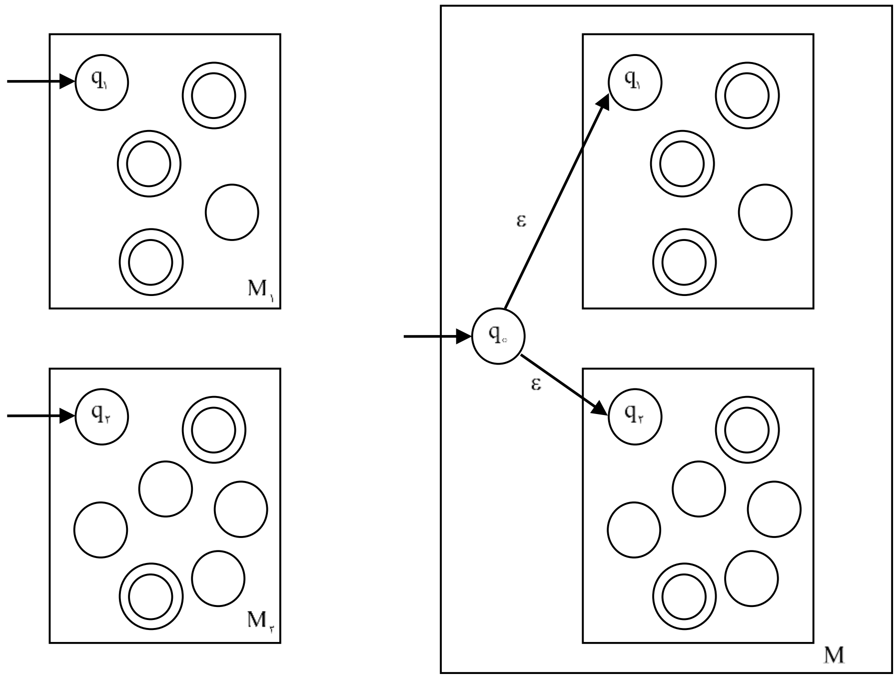
۱. $Q=\{q_0\}\cup Q_1\cup Q_2$، که در آن $q_0$ یک حالت جدید است.

۲. $q_0$ حالت شروع M است.

۳. $F=F_1\cup F_2$ (حالت‌های پذیرش).

۴. $\delta: Q\times\Sigma_\varepsilon\to P(Q)$ به صورت زیر تعریف می‌شود. به ازای هر $r\in Q$ و هر $a\in\Sigma_\varepsilon$،

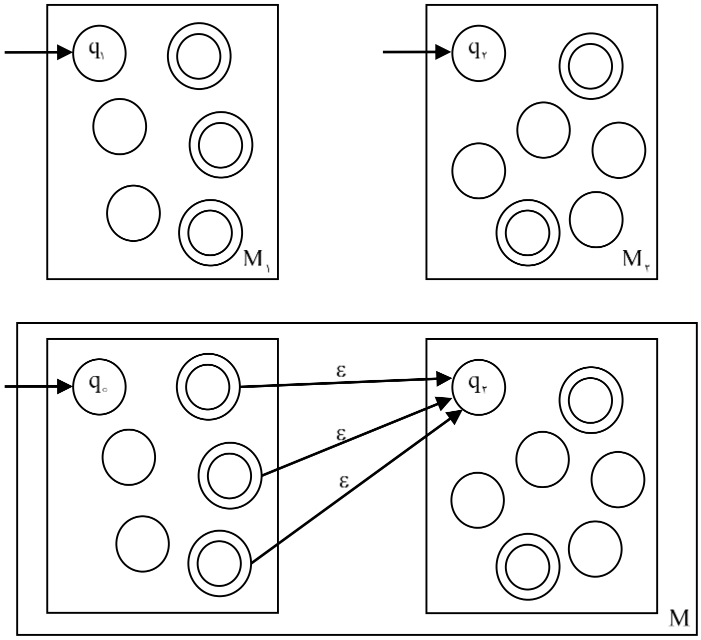
$$\delta(r,a)=\begin{cases}\delta_1(r,a) & \text{if} \quad r\in Q_1,\\ \delta_2(r,a) & \text{if} \quad r\in Q_2,\\ \{q_1,q_2\} & \text{if} \quad r=q_0, a=\varepsilon,\\ \phi & \text{if} \quad r=q_0, a\notin\varepsilon.\end{cases}$$




**شکل ۶-۱:** اتوماتای متناهی نامعین M زبان $L(M_1) \cup L(M_2)$ را می‌پذیرد.

**قضیه ۶-۵-۸:** مجموعه‌ی زبان‌های منظم تحت عمل اتصال بسته است، یعنی اگر $A_1$ و $A_2$ زبان‌های منظم بر روی الفبای $\Sigma$ باشند، آنگاه $A_1A_2$ نیز زبان منظم است.

**اثبات:** فرض کنیم $M_1 = \{Q_1, \Sigma, \delta_1, q_1, F_1\}$ یک NFA باشد به طوری که $A_1 = L(M_1)$. مشابهاً $M_2 = \{Q_2, \Sigma, \delta_2, q_2, F_2\}$ نیز یک NFA در نظر گرفته شود، به طوری که $A_2 = L(M_2)$. مانند اثبات قضیه قبل فرض می‌کنیم $Q_1 \cap Q_2 = \phi$. حال یک NFA مانند $M = \{Q, \Sigma, \delta, q_0, F\}$ طوری می‌سازیم که $L(M) = A_1 A_2$. روش ساختن در شکل ۶-۲ نشان داده شده است.




**شکل ۶-۲:** اتوماتای متناهی نامعین M زبان $L(M_1)L(M_2)$ را می‌پذیرد.

این NFA به صورت زیر تعریف می‌شود:

۱. $Q = Q_1 \cup Q_2$.

۲. $q_0 = q_1$.

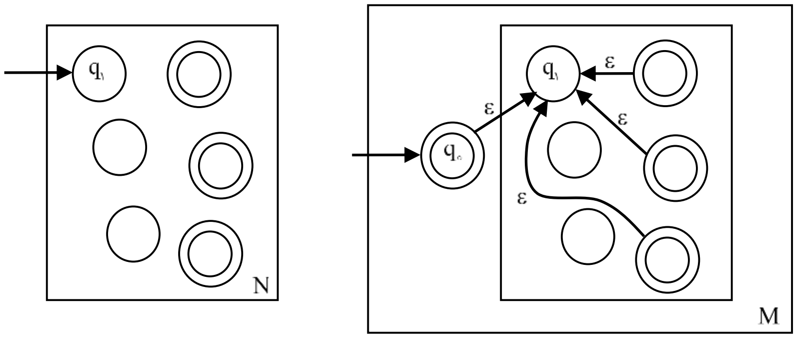
۳. $F = F_2$.

۴. تابع انتقال $\delta : Q \times \Sigma_\varepsilon \to P(Q)$ به صورت زیر تعریف می‌شود: به ازای هر $r \in Q$ و هر $a \in \Sigma_\varepsilon$.

$$\delta(r,a) = \begin{cases} \delta_1(r,a) & \text{if} \quad r \in Q_1 \quad \text{and} \quad r \notin F_1, \\ \delta_1(r,a) & \text{if} \quad r \in F_1 \quad \text{and} \quad a \neq \varepsilon, \\ \delta_1(r,a) \cup \{q_2\} & \text{if} \quad r \in F_1 \quad \text{and} \quad a = \varepsilon, \\ \delta_2(r,a) & \text{if} \quad r \in Q_2. \end{cases}$$

**قضیه ۶-۵-۹:** مجموعه‌ی زبان‌های منظم تحت عمل ستاره بسته است، یعنی اگر $A$ یک زبان منظم باشد، آنگاه $A^*$ نیز زبان منظم است.

**اثبات:** فرض کنیم $\Sigma$ الفبای $A$ و $N=(Q_1,\Sigma_1,\delta_1,q_1,F_1)$ یک NFA باشند، به طوری که $A=L(N)$. یک NFA ی $M=\{Q,\Sigma,\delta,q_0,F\}$ می‌سازیم که $L(M)=A^*$. روش ساختن در شکل ۶-۳ بیان شده است.




**شکل ۶-۳:** اتوماتای متناهی نامعین $M$ زبان $\left(L(N)\right)^*$ را می‌پذیرد.

این NFA به صورت زیر تعریف می‌شود:

۱. $Q=\{q_0\}\cup Q_1$ که $q_0$ حالت جدید است.

۲. $q_0$ حالت شروع $M$ است.

۳. $F=\{q_0\}\cup F_1$ (چون $\varepsilon\in A^*$، پس $q_0$ یک حالت پذیرش است).

۴. تابع انتقال $\delta:Q\times\Sigma_\varepsilon\to P(Q)$ به صورت زیر تعریف می‌شود: به ازای هر $r\in Q$ و هر $a\in\Sigma_\varepsilon$،

$$\delta(r,a)=\begin{cases}\delta_1(r,a) & \text{if} \quad r\in Q_1 \quad \text{and} \quad r\notin F_1,\\ \delta_1(r,a) & \text{if} \quad r\in F_1 \quad \text{and} \quad a\neq \varepsilon,\\ \delta_1(r,a)\bigcup\{q_1\} & \text{if} \quad r\in F_1 \quad \text{and} \quad a=\varepsilon,\\ \{q_1\} & \text{if} \quad r=q_0 \quad \text{and} \quad a=\varepsilon,\\ \phi & \text{if} \quad r=q_0 \quad \text{and} \quad a\neq\varepsilon.\end{cases}$$

**قضیه ۶-۵-۱۰:** مجموعه زبان‌های منظم منظم تحت اعمال مکمل و اشتراک بسته‌اند.

۱. اگر $A$ یک زبان منظم بر روی الفبای $\Sigma$ باشد، آنگاه مکمل آن یعنی $\overline{A}=\{W\in\Sigma^* \mid W\notin A\}$ نیز یک زبان منظم است.
۲. اگر $A_1$ و $A_2$ زبان‌های منظم بر روی الفبای $\Sigma$ باشند، آنگاه اشتراک آنها یعنی $A_1\bigcap A_2=\{W\in\Sigma^* \mid W\in A_1 \text{ and } W\in A_2\}$ نیز زبان منظم است.

## ۶-۶: عبارات منظم

در این بخش عبارات منظم را بیان می‌کنیم. این عبارات برای توصیف زبان‌ها به کار می‌روند. خواهیم دید که رده زبان‌هایی که با عبارات منظم توصیف می‌شوند بر رده زبان‌های منظم منطبق می‌باشند.

پیش از بیان رسمی مفاهیم و تعاریف مربوط چند مثال ارایه می‌کنیم.

عبارت $(0\bigcup 1)01^*$ را در نظر می‌گیریم. زبانی که با این عبارت توصیف می‌شود، مجموعه‌ی همه رشته‌های دودویی است که:

۱. با ۱ یا ۰ شروع می‌شود (به دلیل وجود $(1\bigcup 0)$)،
۲. برای آن نماد دوم ۰ است (به دلیل وجود ۰) و
۳. به صفر یا ۱ های بیشتری ختم می‌شود (به دلیل وجود $1^*$).

پس زبان توصیف شده با این عبارت به صورت زیر می‌تواند باشد:

$$\{00, 001, 0011, 00111, \ldots, 10, 101, 1011, 10111, \ldots\}$$

حال چند مثال دیگر ارایه می‌شود (در همه این مثال‌ها الفبا مجموعه‌ی $\{0,1\}$ است).

- زبان $\{W$ شامل دقیقاً دو $0$ است $|\,W\}$ با عبارت زیر بیان می‌شود:

$$1^*01^*01^*$$

- زبان $\{W$ شامل حداقل دو $0$ است $|\,W\}$ با عبارت زیر توصیف می‌شود:

$$(0\cup1)^*0(0\cup1)^*0(0\cup1)^*$$

- زبان {رشته‌ی ۱۰۱۱ یک زیر رشته از $W$ است $|\,W\}$ با عبارت زیر بیان می‌شود:

$$(0\cup1)^*1011(0\cup1)^*$$

- زبان {طول $W$ عدد زوج است $|\,W\}$ با عبارت زیر توصیف می‌گردد:

$$\big((0\cup1)(0\cup1)\big)^*$$

- زبان {طول $W$ عدد فرد است $|\,W\}$ با عبارت زیر بیان می‌گردد:

$$(0\cup1)\big((0\cup1)(0\cup1)\big)^*$$

- زبان $\{0,1011\}$ عبارتست از:

$$1011\cup0$$

- زبان {نمادهای اول و آخر $W$ مساوی‌اند $|\,W\}$ چنین توصیف می‌شود:

$$0(0\cup1)^*0\cup1(0\cup1)^*1\cup0\cup1$$

یادآوری می‌کنیم که نتیجه یک عبارت منظم یک زبان می‌باشد. همچنین در نظر داشته باشید که عبارات منظم روش قدرتمندی برای توصیف الگوهای کامپیوتری می‌باشد. در سیستم عامل UNIX، زبان‌-های برنامه نویسی، و ویراستارهای متنی، همگی از عبارات منظم استفاده می‌شود.

نکته دیگر که باید به آن توجه کنیم این است که تقدم اعمال در عبارات منظم به ترتیب ستاره، اتصال و در نهایت اجتماع می‌باشد، مگر از پرانتزها برای تغییر تقدم‌ها استفاده شود.

حال یک تعریف رسمی از عبارت منظم ارایه می‌دهیم:

**تعریف ۶-۶-۱:** فرض کنیم $\Sigma$ یک الفبای غیر تهی باشد.

۱. $\varepsilon$ یک عبارت منظم است.
۲. $\phi$ یک عبارت منظم است.
۳. به ازای هر $a \in \Sigma$، $a$ یک عبارت منظم است.
۴. اگر $R_۱$ و $R_۲$ عبارات منظم باشند، آنگاه $R_۱ \cup R_۲$ یک عبارت منظم است.
۵. اگر $R_۱$ و $R_۲$ عبارات منظم باشند، آنگاه $R_۱R_۲$ یک عبارت منظم است.
۶. اگر $R$ یک عبارت منظم باشد، آنگاه $R^*$ نیز عبارت منظم است.

حال مثالی ارایه می‌کنیم. گوییم عبارت $(\circ\cup۱)^*۱\circ۱(\circ\cup۱)^*$ یک عبارت منظم است، الفبا مجموعه $\{\circ,۱\}$ است. بدیهی است باید نشان دهیم که این عبارت در تعریف بالا صدق می‌کند. این کار را چنین انجام می‌دهیم:

- بنابر ۳، $\circ$ عبارت منظم است.
- بنابر ۳، ۱ عبارت منظم است.
- چون $\circ$ و ۱ عبارات منظم‌اند، بنابر ۴، پس $\circ\cup۱$ نیز عبارت منظم است.
- چون بنابر ۶، $\circ\cup۱$ منظم است پس $(\circ\cup۱)^*$ نیز منظم است.
- بنابر ۵، چون ۱ و $\circ$ منظم‌اند، پس $۱\circ$ نیز عبارت منظم است.
- بنابر ۵، چون $۱\circ$ و ۱ منظم‌اند، پس $۱\circ۱$ نیز عبارت منظم است.
- بنابر ۵، چون $(\circ\cup۱)^*$ و $۱\circ۱$ عبارات منظم‌اند، پس $(\circ\cup۱)^*۱\circ۱$ نیز یک عبارت منظم است.
- بنابر ۵، چون $(\circ\cup۱)^*۱\circ۱$ و $(\circ\cup۱)^*$ عبارات منظم‌اند، پس $(\circ\cup۱)^*۱\circ۱(\circ\cup۱)^*$ نیز عبارت منظم است.

حال زبانی که با یک عبارت منظم توصیف می‌شود به صورت زیر تعریف می‌گردد:

**تعریف ۶-۶-۲:** فرض کنیم $\Sigma$ یک الفبای غیر تهی باشد.

۱. عبارت منظم $\varepsilon$، زبان $\{\varepsilon\}$ را بیان می‌کند.
۲. عبارت منظم $\phi$، زبان $\phi$ را بیان می‌کند.
۳. به ازای هر $a \in \Sigma$، عبارت منظم a، زبان $\{a\}$ را بیان می‌کند.
۴. اگر $R_1$ و $R_2$ عبارات منظم باشند و فرض کنیم $L_1$ و $L_2$، به ترتیب زبان‌های توصیف شده به وسیله آنها باشد، آنگاه عبارت منظم $R_1 \cup R_2$ زبان $L_1 \cup L_2$ را توصیف می‌کند.
۵. اگر $R_1$ و $R_2$ عبارات منظم باشند و فرض کنیم $L_1$ و $L_2$، به ترتیب زبان‌های توصیف شده به وسیله آنها باشد، آنگاه عبارت منظم $R_1R_2$ زبان $L_1 L_2$ را بیان می‌کند.
۶. اگر R یک عبارت منظم و L زبان توصیف شده بوسیله آن باشد. آنگاه عبارت منظم $R^*$ زبان $L^*$ را بیان می‌کند.

چند مثال را در این جا در نظر می‌گیریم.

- عبارت منظم $(0 \cup \varepsilon)(1 \cup \varepsilon)$ زبان $\{01, 0, 1, \varepsilon\}$ را توصیف می‌کند.
- عبارت منظم $0 \cup \varepsilon$ زبان $\{0, \varepsilon\}$ را توصیف می‌نماید، در حالی که عبارت منظم $1^*$ زبان $\{\varepsilon, 1, 11, 111, \ldots\}$ را توصیف می‌کند.

در نتیجه، عبارت منظم $(0 \cup \varepsilon)1^*$ زبان $\{0, 01, 011, \ldots, \varepsilon, 1, 11, 111, \ldots\}$ را بیان می‌کند. توجه کنید که این زبان با عبارت $01^* \cup 1^*$ نیز توصیف می‌شود.

- عبارت منظم $1^*\phi$ ، زبان تهی، یعنی زبان $\phi$، را توصیف می‌کند.
- عبارت منظم $\phi^*$ زبان $\{\varepsilon\}$ را بیان می‌کند.

**تعریف ۶-۶-۳:** فرض کنیم $R_1$ و $R_2$ عبارات منظم و $L_1$ و $L_2$، به ترتیب زبان‌های توصیف شده به وسیله آنها باشد، آنگاه اگر $L_1 = L_2$ (یعنی $R_1$ و $R_2$ با یک زبان بیان می‌شوند)، آنگاه می‌نویسیم $R_1 = R_2$.

از این رو اگرچه $(0\cup\varepsilon)1^*$ و $01^*\cup 1^*$، عبارات منظم متفاوتی هستند، می‌نویسیم $(0\cup\varepsilon)1^* = 01^*\cup 1^*$، زیرا یک زبان را توصیف می‌کنند.

**قضیه ۶-۶-۴:** فرض کنیم $R_1$، $R_2$ و $R_3$ عبارات منظم باشند، آنگاه، اتحادهای زیر برقرار است.

۱. $R_1\phi=\phi R_1=\phi$

۲. $R_1\varepsilon=\varepsilon R_1=R_1$

۳. $R_1\cup\phi=\phi\cup R_1=R_1$

۴. $R_1\cup R_1=R_1$

۵. $R_1\cup R_2=R_2\cup R_1$

۶. $R_1(R_2\cup R_3)=R_1R_2\cup R_1R_3$

۷. $(R_1\cup R_2)R_3=R_1R_3\cup R_2R_3$

۸. $R_1(R_2R_3)=(R_1R_2)R_3$

۹. $\phi^*=\varepsilon$

۱۰. $\varepsilon^*=\varepsilon$

۱۱. $(\varepsilon\cup R_1)^*=R_1^*$

۱۲. $(\varepsilon\cup R_1)(\varepsilon\cup R_1)^*=R_1^*$

۱۳. $R_1^*(\varepsilon\cup R_1)=(\varepsilon\cup R_1)R_1^*=R_1^*$

۱۴. $R_1^*R_2\cup R_2=R_1^*R_2$

۱۵. $R_1(R_2R_1)^*=(R_1R_2)^*R_1$

۱۶. $(R_1\cup R_2)^*=(R_1^*R_2)^*R_1^*=(R_2^*R_1)^*R_2^*$

اثبات این اتحادها ساده و قدری خسته کننده است، لذا از آنها صرف نظر می‌کنیم. اما پیشنهاد می‌کنیم که خواننده به شکلی خود را متقاعد کند که این اتحادها معانی مناسبی دارند. برای مثال تساوی زیر را در نظر می‌گیریم.

$$(0\cup\varepsilon)1^*=01^*\cup 1^*$$

می‌توانیم درستی این تساوی را با کمک برخی از اتحادهای بالا نشان دهیم.

(بنابر ۷) $$(\circ \cup \varepsilon)۱^* = \circ ۱^* \cup \varepsilon ۱^*$$

(بنابر ۲) $$= \circ ۱^* \cup ۱^*$$

## ۶-۷: لم تزریق و زبان‌های نامنظم

## (The pumping lemma and nonregular languages)

در بخش‌های پیشین دیدیم که رده زبان‌های منظم تحت اعمال مختلفی بسته هستند و این زبان‌ها با اتوماتاهای متناهی، معین یا نامعین، توصیف می‌گردند. این خواص ما را در تولید روش‌هایی یاری نمودند که بوسیله آنها می‌توانیم نشان دهیم یک زبان منظم است.

در این بخش روشی را نشان می‌دهیم که می‌تواند برای اثبات نامنظم بودن برخی زبان‌ها به کار رود.

در حقیقت خاصیتی را بیان می‌کنیم که همه زبان‌های منظم باید دارا باشند و این خاصیت را، لم تزریق گوییم. آنگاه هر زبانی که در این خاصیت صدق نکند، باید نامنظم باشد.

لم تزریق بیان می‌کند که هر رشته‌ی به اندازه کافی طولانی، در یک زبان منظم می‌تواند تزریق شود، یعنی در این رشته بخشی وجود دارد که می‌تواند هر تعداد دفعه تکرار شود، پس رشته‌های حاصل همه در زبان قرار می‌گیرند.

در واقع بر اساس لم تزریق می‌توان نتیجه گرفت که هر رشته‌ی متعلق به زبان منظم L را در صورتی که به اندازه کافی طولانی باشد، می‌تواند به سه بخش تقسیم کرد، به طوری که تکرار بخش میانی به هر میزان رشته‌ی متعلق به زبان L را تولید نماید.

**قضیه ۶-۷-۱: لم تزریق برای زبان‌های منظم**

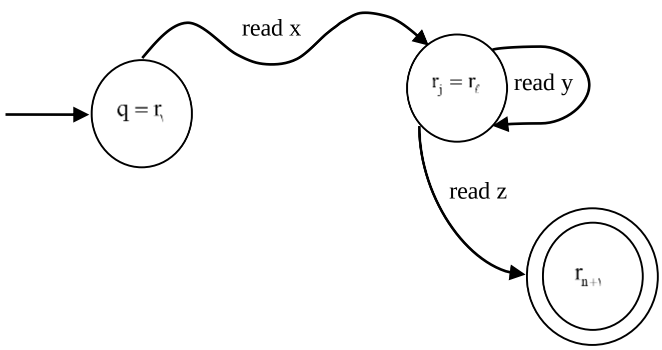
فرض کنیم $A$ یک زبان منظم باشد، آنگاه عدد صحیحی مانند $P \geq ۱$ وجود دارد که آن را طول تزریق گویند، به طوری که موارد زیر برقرار باشند. هر رشته‌ی $S \in A$، با $|S| \geq P$ می‌تواند به صورت $S = xyz$ نوشته شود، به طوری که:

۱. $y \notin \varepsilon$ (یعنی $|y| \geq ۱$)،
۲. $|xy| \leq P$،
۳. به ازای هر $i \geq ۰$، $x y^i z \in A$.

در حقیقت لم تزریق بیان می‌کند که با جایگزین کردن بخش $y$ در $S$ بوسیله صفر یا تعداد زیادی $y$های دیگر، رشته‌ی حاصل باز هم در زبان $A$ قرار دارد.

**اثبات:** فرض کنیم $\Sigma$ الفبای $A$ باشد. چون $A$ یک زبان منظم است، پس یک DFA مانند $M = \{Q, \Sigma, \delta, q, F\}$ وجود دارد که $A$ را می‌پذیرد. تعداد حالت‌های موجود در $Q$ را با $P$ تعریف می‌کنیم. فرض کنیم $S = S_۱ S_۲, \ldots, S_n$ یک رشته‌ی دلخواه در $A$ باشد و $n \geq P$. تعریف می‌کنیم $r_۱ = q$، $r_۲ = \delta(r_۱, s_۱)$، $r_۳ = \delta(r_۲, s_۲)$، ... و $r_{n+۱} = \delta(r_n, s_n)$. بنابراین زمانی که $M$ رشته‌ی $S$ را از چپ به راست می‌خواند، حالت‌های $r_۱, r_۲, \ldots, r_{n+۱}$ را می‌بیند. چون $S$ رشته‌ای در $A$ است، پس $r_{n+۱}$ به $F$ تعلق دارد.

حال اولین $P + ۱$ رشته‌ی $r_۱, r_۲, \ldots, r_{P+۱}$ را در این دنباله در نظر می‌گیریم. چون تعداد حالت‌های $M$ برابر $P$ است، اصل لانه کبوتری می‌گوید که باید یک حالت وجود داشته باشد که در این دنباله دو بار رخ داده است. یعنی، اندیس‌های $j$ و $\ell$ طوری وجود دارند که $۱ \leq j \leq \ell \leq P + ۱$ و $r_j = r_\ell$.



حال x، y و z را به صورت‌های زیر تعریف می‌کنیم:

$$z = S_\ell ... S_n \ , \ y = S_j ... S_{\ell-1} \ , \ x = S_1 S_2 ... S_{j-1}$$

چون $j < \ell$، داریم $y \neq \varepsilon$، اثبات اولین مورد در قضیه. از آنجا که $\ell \leq p+1$، داریم $|xy| = \ell - 1 \leq p$، و این اثبات مورد دوم در قضیه است. برای دیدن اثبات قسمت سوم قضیه، یادآور می‌شویم که رشته $S = xyz$ بوسیله M پذیرفته می‌شود. زمانی که x را می‌خواند M از حالت q به حالت $r_j$ منتقل می‌شود. زمانی که y را می‌خواند، از حالت $r_j$ به حالت $r_\ell = r_j$ می‌رود، یعنی پس از خواندن y، M بار دیگر در حالت $r_j$ است. در لحظه‌ای که z را می‌خواند از حالت $r_j$ به حالت پذیرش $r_{n+1}$ انتقال می‌یابد. پس زیررشته y می‌تواند با هر تعداد $i \geq 0$ تکرار شود و رشته متناظر $xy^iz$ هنوز با M پذیرفته می‌گردد و این نتیجه می‌دهد که $xy^iz \in A$، به ازای هر $i \geq 0$.

### ۶-۷-۲: کاربردهایی از لم تزریق

برای استفاده از لم تزریق برای اثبات نامنظم بودن زبان A، فرض می‌کنیم که این زبان منظم باشد و سعی می‌کنیم به تناقض برسیم. سپس از لم تزریق استفاده می‌کنیم تا تضمین کند که طول تزریق P موجود بوده و هر رشته‌ای که طول حداقل P را در A داشته باشد، می‌تواند تزریق شود. سپس رشته‌ای همانند S در A می‌یابیم که طول آن بزرگتر یا مساوی P بوده، ولی نتوان آن را تزریق نمود. در نهایت نشان می‌دهیم با هر روشی که S را به x، y و z تجزیه کنیم (البته با رعایت شرایط تزریق)، برای هر روش تجزیه می‌توان i ای پیدا کرد که $xy^iz$ عضو A نباشد. در هر حال وجود S در مورد زبان منظم A یک تناقض می‌باشد. پس A نامنظم است.

### ۶-۷-۳: چند مثال

**مثال اول**

حال $x$، $y$ و $z$ را به صورت‌های زیر تعریف می‌کنیم:

$$z = S_{\ell}...S_n \ , \ y = S_j...S_{\ell-1} \ , \ x = S_1S_2...S_{j-1}$$

چون $j < \ell$، داریم $y \neq \varepsilon$، اثبات اولین مورد در قضیه. از آنجا که $\ell \leq p+1$، داریم $|xy| = \ell - 1 \leq p$، و این اثبات مورد دوم در قضیه است. برای دیدن اثبات قسمت سوم قضیه، یادآور می‌شویم که رشته $S = xyz$ بوسیله $M$ پذیرفته می‌شود. زمانی که $x$ را می‌خواند $M$ از حالت $q$ به حالت $r_j$ منتقل می‌شود. زمانی که $y$ را می‌خواند، از حالت $r_j$ به حالت $r_{\ell} = r_j$ می‌رود، یعنی پس از خواندن $y$، $M$ بار دیگر در حالت $r_j$ است. در لحظه‌ای که $z$ را می‌خواند از حالت $r_j$ به حالت پذیرش $r_{n+1}$ انتقال می‌یابد. پس زیررشته $y$ می‌تواند با هر تعداد $i \geq 0$ تکرار شود و رشته متناظر $xy^iz$ هنوز با $M$ پذیرفته می‌گردد و این نتیجه می‌دهد که $xy^iz \in A$، به ازای هر $i \geq 0$.

### ۶-۷-۲: کاربردهایی از لم تزریق

برای استفاده از لم تزریق برای اثبات نامنظم بودن زبان $A$، فرض می‌کنیم که این زبان منظم باشد و سعی می‌کنیم به تناقض برسیم. سپس از لم تزریق استفاده می‌کنیم تا تضمین کند که طول تزریق $P$ موجود بوده و هر رشته‌ای که طول حداقل $P$ را در $A$ داشته باشد، می‌تواند تزریق شود. سپس رشته‌ای همانند $S$ در $A$ می‌یابیم که طول آن بزرگتر یا مساوی $P$ بوده، ولی نتوان آن را تزریق نمود. در نهایت نشان می‌دهیم با هر روشی که $S$ را به $x$، $y$ و $z$ تجزیه کنیم (البته با رعایت شرایط تزریق)، برای هر روش تجزیه می‌توان $i$ ای پیدا کرد که $xy^iz$ عضو $A$ نباشد. در هر حال وجود $S$ در مورد زبان منظم $A$ یک تناقض می‌باشد. پس $A$ نامنظم است.

### ۶-۷-۳: چند مثال

**مثال اول**

زبان $A=\{0^n1^n \mid n\geq 0\}$ را در نظر می‌گیریم. با تناقض نشان می‌دهیم که A زبان منظم نیست. فرض کنیم A یک زبان منظم باشد. طول تزریق را $P\geq 1$ در نظر می‌گیریم. رشته‌ی $S=0^P1^P$ را بررسی می‌کنیم. بدیهی است $S\in A$ و $|S|=2P\geq P$. بنابراین با توجه به لم تزریق، S را می‌توان به صورت $S=xyz$ نوشت که $y\neq\varepsilon$، $|xy|\leq P$ و $xy^iz\in A$ به ازای هر $i\geq 0$.

مشاهده می‌شود که چون $|xy|\leq P$، پس رشته‌ی y تنها شامل ۰ها است. به علاوه چون $y\neq\varepsilon$ ، y شامل حداقل یک صفر می‌باشد. اما حال با مشکل مواجه هستیم، بدین ترتیب که هیچ یک از رشته‌های $xy^0z=xz\ ,\ xy^2z=xyyz\ ,\ xy^3z=xyyyz,\ldots$ در A قرار ندارند. به هر حال با کمک لم تزریق، همه این رشته‌ها باید در A قرار گیرند. از این رو با یک تناقض روبه‌رو شده‌ایم و نتیجه می‌گیریم که A یک زبان منظم نیست.

**مثال دوم**

زبان $A=\{w\in\{0,1\}^* \mid$ تعداد ۰ها با تعداد ۱ها در w برابرند$\}$ را در نظر می‌گیریم. نشان می‌دهیم A زبان منظم نیست.

فرض کنیم A یک زبان منظم باشد. اگر $P\geq 1$ طول تزریق داده شده در لم باشد، آنگاه رشته‌ی $S=0^P1^P$ را بررسی می‌کنیم. بدیهی است $S\in A$ و $|S|=2P\geq P$. از لم تزریق، $S=xyz$ که $y\neq\varepsilon$ و $|xy|\leq P$ و $xy^iz\in A$ به ازای هر $i\geq 0$. چون $|xy|\leq P$، رشته‌ی y تنها ۰ها را دارد. از آنجا که $y\neq\varepsilon$، y حداقل یک ۰ دارد. پس رشته‌ی $xy^2z=xyyz$ شامل تعداد ۰های بیشتر از تعداد ۱ها است، و این بیان می‌کند که این رشته در A نیست. اما از لم تزریق چنین برمی‌آید که این رشته باید در A باشد. پس به تناقض رسیده‌ایم و لذا A منظم نیست.

**مثال سوم**

زبان $A=\{ww \mid w\in\{0,1\}^*\}$ را در نظر می‌گیریم و ثابت می‌کنیم که A زبان نامنظم است. فرض کنیم A زبان منظم باشد. طول تزریق عبارتست از $P\geq 1$. حال رشته‌ی $S=0^P10^P1$ را در نظر می‌گیریم. بدیهی است $S\in A$ و $|S|=2P+2\geq P$. با کمک لم تزریق $S=xyz$ که $y\not\in\varepsilon$ و $|xy|\leq P$ و $xy^iz\in A$ به ازای هر $i\geq 0$. چون $|xy|\leq P$، رشته‌ی S تنها 0ها را دارد از آنجا که $y\not\in\varepsilon$، y حداقل یک 0 دارد. بنابراین رشته‌ی $xy^2z=xyyz$ در A نیست. اما با توجه به لم تزریق این رشته در A است و این تناقض می‌باشد، لذا A زبان منظم نیست.

اگر $S=0^{2P}$ اختیار شود که رشته‌ای در A است با طول حداقل P، به تناقض نمی‌رسیم. دلیل آن این است که رشته‌ی y می‌تواند طول زوج داشته باشد. پس $0^{2P}$ برای نشان دادن این که A منظم نیست، رشته‌ی نادرستی است. با انتخاب $S=0^P10^P1$ به یک تناقض می‌رسیم، پس این رشته برای اثبات نامنظم بودن A انتخاب درستی است.

**مثال چهارم**

زبان $A=\{0^m1^n \mid m>n\geq 0\}$ را در نظر می‌گیریم و با تناقض ثابت می‌کنیم که A زبان نامنظم است. فرض می‌کنیم A زبان منظم باشد و $P\geq 1$ طول تزریق در نظر گرفته شود. رشته‌ی $S=0^{P+1}1^P$ را انتخاب می‌کنیم. بدیهی است $S\in A$ و $|S|=2P+1\geq P$. با استفاده از لم تزریق $S=xyz$ که $y\not\in\varepsilon$، $|xy|\leq P$ و $xy^iz\in A$، به ازای هر $i\geq 0$.

چون $|xy|\leq P$، رشته‌ی y تنها شامل 0ها است. از آنجا که $y\not\in\varepsilon$، y حداقل یک 0 دارد. حال رشته‌ی $xy^0z=xz$ را در نظر می گیریم. تعداد ۱ها در این رشته برابر P است، در حالی که تعداد 0ها حداکثر برابر P می‌باشد. در نتیجه رشته‌ی $xy^0z$ در A نیست. اما با توجه به لم تزریق این رشته در A قرار دارد. این تناقض ثابت می‌کند که A یک زبان منظم نیست.

**مثال پنجم**

رشته‌ی $A=\{۱^{n^۲} \mid n \geq ۰\}$ را مورد نظر قرار می‌دهیم و با تناقض ثابت می‌کنیم که A یک زبان منظم نیست. فرض کنیم A یک زبان منظم باشد و طول تزریق $P \geq ۱$. رشته‌ی $S=۱^{P^۲}$ را بررسی می‌کنیم.
بدیهی است $S \in A$ و $|S|=P^۲ \geq P$. می‌نویسیم $S=xyz$ که $y \not\in \varepsilon$ و $|xy| \leq P$ و $xy^iz \in A$
به ازای هر $i \geq ۰$ داریم $|S|=|xyz|=P^۲$ و $|xy^۲z|=|xyyz|=|xyz|+|y|=P^۲+|y|$. چون $|xy| \leq P$، پس $|y| \leq P$. از آنجا که $y \not\in \varepsilon$ داریم $|y| \geq ۱$. لذا نتیجه می‌شود:

$$P^۲ < |xy^۲z| \leq P^۲+P < (P+۱)^۲$$

بنابراین طول رشته‌ی $xy^۲z$ اکیداً بین دو مربع متوالی قرار گرفته است. پس این طول مربع یک عدد نمی‌تواند باشد و $xy^۲z$ در A قرار نمی‌گیرد، و از آنجا که بنابر لم تزریق باید در A قرار داشته باشد پس اثبات تمام است و A نامنظم است.

**مثال ششم**

زبان A را به صورت $A=\{۱^n \mid$ یک عدد اول است $n\}$ در نظر می‌گیریم و ثابت می‌کنیم که A یک زبان منظم نیست. فرض کنیم A منظم باشد و طول تزریق $P \geq ۱$. فرض کنیم $n \geq P$ یک عدد اول باشد، و رشته $S=۱^n$ را در نظر می‌گیریم. بدیهی است $S \in A$ و $|S|=n \geq P$. حال می‌نویسیم $S=xyz$ که $y \not\in \varepsilon$ و $|xy| \leq P$ و $xy^iz \in A$ به ازای هر $i \geq ۰$.

اگر K یک عدد صحیح باشد به طوری که $y=۱^K$. چون $y \not\in \varepsilon$ داریم $K \geq ۱$. حال برای هر $i \geq ۰$، عدد $n+(i-۱)K$ یک عدد اول است، زیرا $xy^iz=۱^{n+(i-۱)K} \in A$. برای $i=n+۱$ داریم:

$$n+(i-۱)K=n+nK=n(۱+K)$$

که یک عدد اول نیست، زیرا $n \geq ۲$ و $۱+K \geq ۲$. این تناقض ثابت می‌کند A منظم نیست.