$$1+2+3+...+(n+1) = 1+2+3+...+n+(n+1)$$
$$= \frac{n(n+1)}{2} + (n+1)$$
$$= \frac{(n+1)(n+2)}{2}$$

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.

6-3: 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 1: 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 25 (واحد پول) پرداخت نماید. فرض می‌کنیم تنها سه سکه 5، 10، 25 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 25 (یا بیشتر) را وارد کرده است یا نه. ماشین برای تصمیم گیری در هر یک از شش حالت زیر قرار می‌گیرد.

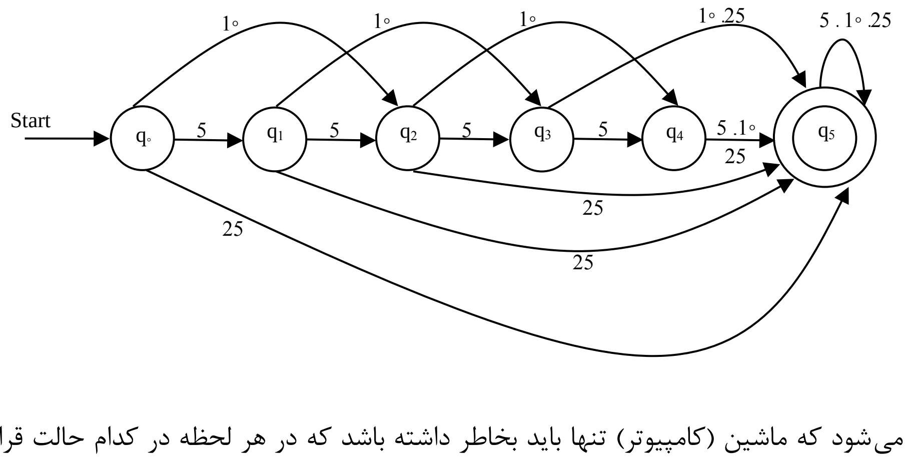
- ماشین در حالت q_0 است، اگر هیچ پولی دریافت نکرده باشد.
- ماشین در حالت q_1 است، اگر دقیقاً مبلغ 5 دریافت کرده باشد.

- ماشین در حالت q_2 است، اگر دقیقاً مبلغ 10 دریافت کرده باشد.
- ماشین در حالت q_3 است، اگر دقیقاً مبلغ 15 دریافت کرده باشد.
- ماشین در حالت q_4 است، اگر دقیقاً مبلغ 20 دریافت کرده باشد.
- ماشین در حالت q_5 است، اگر دقیقاً مبلغ 25 دریافت کرده باشد.

در آغاز یعنی وقتی که خودرو به دروازه می‌رسد، ماشین در حالت q_0 است. فرض کنیم که راننده مجموعه سکه‌های (10، 5، 5، 10) را داشته باشد.

- ماشین پس از دریافت اولین سکه 10، از حالت q_0 به حالت q_2 انتقال می‌یابد.
- ماشین پس از دریافت اولین سکه 5، از حالت q_2 به حالت q_3 انتقال می‌یابد.
- ماشین پس از دریافت دومین سکه 5، از حالت q_3 به حالت q_4 انتقال می‌یابد.
- ماشین پس از دریافت دومین سکه 10، از حالت q_4 به حالت q_5 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_5 با دو دایره تودرتو نمایانده شده است، زیرا حالت خاص است (پذیرش). به محض اینکه ماشین به این حالت می‌رسد دروازه باز می‌شود.



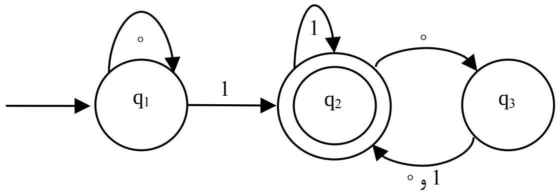
مشاهده می‌شود که ماشین (کامپیوتر) تنها باید بخاطر داشته باشد که در هر لحظه در کدام حالت قرار دارد. پس تنها به یک مقدار کمی حافظه نیاز دارد. باید بتواند میان هر یک از شش حالت تمایز قایل شود، بنابراین حافظه‌ای به اندازه

$$[\log ۶]=۳\mathrm{bits}$$

کافی است.

6-3-1: 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 1، از حالت $q_۱$ به حالت $q_۲$ می‌رود.
- ماشین پس از خواندن دومین 1، 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 1، 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 1 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 1 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 1 قرار می‌گیرد، فرد است.

حال تعریف رسمی اتوماتای متناهی را بیان می‌کنیم:

تعریف 6-3-2: یک اتوماتای متناهی عبارتست از یک 5 تایی $M=(Q,\Sigma,\delta,q,F)$ که در آن:

1. Q یک مجموعه متناهی است، عناصرش را حالت‌ها می‌نامند.
2. Σ یک مجموعه متناهی است که آن را الفبا (alphbet) نامند، عناصرش را نمادها گویند.
3. $\delta: Q\times\Sigma \to Q$ تابعی است که آن را تابع انتقال نامند.
4. q عنصری از Q است، آن را حالت شروع نامند.
5. 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 Σ 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 $Q=\{q_0,q_1,q_2,q_3,q_4,q_5\}$ و $\Sigma=(5,10,25)$، حالت شروع q_0 است و $F=\{q_5\}$ و δ به صورت جدول زیر عمل می‌کند.

	5	۱۰	25
q_0	q_1	q_2	q_5
q_1	q_2	q_3	q_5
q_2	q_3	q_4	q_5

q_3	q_4	q_5	q_5
q_4	q_5	q_5	q_5
q_5	q_5	q_5	q_5

مثالی که در آغاز این بخش بیان کردیم نیز یک اتوماتای معین است. برای آن داریم $F=\{q_2\}$ و حالت شروع: q_2، $\Sigma=\{0,1\}$، $Q=\{q_1,q_2,q_3\}$، δ با جدول زیر توصیف می‌شود.

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

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:

$$L(M)=\{w \mid \text{شامل حداقل یک } 1 \text{ است و به تعداد زوج } 0 \text{ها ختم می‌گردد } w\}$$

حال تعریف رسمی زبان یک اتوماتای متناهی را بیان می‌کنیم.

تعریف 6-3-3: فرض کنیم $M=(Q,\Sigma,\delta,q,F)$ یک اتوماتای متناهی و $w=w_1w_2...w_n$ رشته‌ای از Σ باشد. دنباله $r_n,...,r_1,r_0$ از حالت‌ها را به صورت زیر تعریف می‌کنیم:

- $r_0=q,$
- $r_{i+1}=\delta(r_i,w_{i+1}), \quad \text{for} \quad i=0,1,...,n-1$

1. اگر $r_n\in F$، آنگاه گوییم M رشته w را می‌پذیرد.
2. 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 ε نشان می‌دهیم و طول آن صفر است، بنابراین در تعریف بالا $n=0$. در این حالت دنباله $r_0, r_1, \ldots, r_n$ 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 6-3-4: 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.

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

تعریف6-3-5: 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 Σ^* مجموعه‌ی همه رشته‌های روی الفبای Σ را نمایش دهد (Σ^* شامل رشته تهی ε است).

تابع δ 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 Σ 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 $\overline{\delta}$ چیست؟ فرض کنیم r یک حالت از Q و w رشته‌ای از Σ باشد، آنگاه:

- $\overline{\delta}(r, w)$ 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 δ از حالتی به حالت دیگر می‌رود. پس می‌نویسیم:

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

6-3-6: چند مثال از یک اتوماتای متناهی

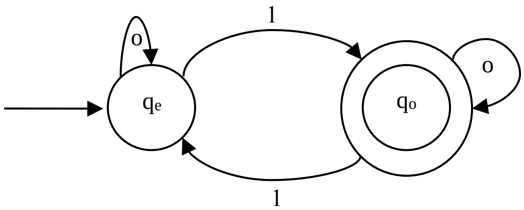
مثال اول

فرض کنیم $A = \{w \mid$ یک رشته دودویی شامل تعداد فرد 1ها است $w\}$. گوییم این زبان A منظم است. برای اثبات این موضوع، باید یک اتوماتای متناهی M بسازیم که $A = L(M)$. این M را چگونه می-سازیم؟ ایده اول: اتوماتای متناهی رشته ورودی w را از چپ به راست می‌خواند و تعداد 1هایی را که دیده است در نظر می‌گیرد. پس از خواندن همه رشته w، بررسی می‌کند که آیا تعداد 1ها فرد است که در این صورت w پذیرفته شده است یا تعداد 1ها 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 1هایی 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 1های 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 1ها را خوانده است و اگر در حالت q_o باشد، آنگاه تعداد 1های خوانده شده فرد است.
- الفبا عبارتست از $\Sigma = \{0, 1\}$.
- حالت شروع برابر q_e است، زیرا در لحظه شروع تعداد 1های خوانده شده توسط اتوماتا مساوی صفر است و صفر یک عدد زوج می‌باشد.
- مجموعه‌ی حالت‌های پذیرش $F = \{q_o\}$ است.
- تابع انتقال δ با جدول زیر بیان می‌شود:

	0	1
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 0ها عبور می‌کند و در حالت شروع قرار می‌گیرد.
- با دیدن اولین 1 به حالت ”شاید دو نماد بعدی ۰۱ باشد“ می‌رود.
- اگر نماد بعدی 1 باشد در حالت ”شاید دو نماد بعدی ۰۱ باشد“ باقی می‌ماند.

- از طرف دیگر، اگر نماد بعدی ۰ باشد به حالت "شاید نماد بعدی 1 باشد" می‌رود.

❖ اگر نماد بعدی 1 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 1 را بخواند.

با بیان چهار حالت زیر، پردازش بدیهی خواهد بود.

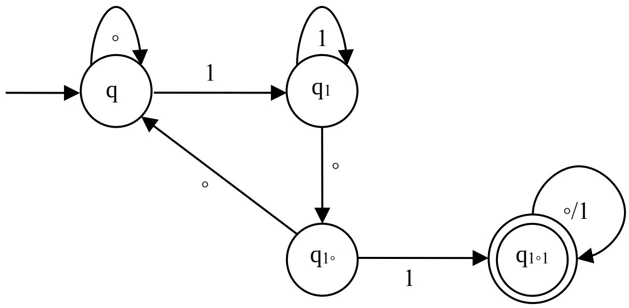
- $q_{۱}$: اتوماتای M در این حالت قرار می‌گیرد اگر آخرین نماد خوانده شده 1 باشد، اما زیر رشته‌ی ۱۰۱ هنوز خوانده نشده است.
- $q_{۱۰}$: M در این حالت قرار دارد اگر دو نماد آخر خوانده شده ۱۰ باشد، اما زیر رشته‌ی ۱۰۱ هنوز خوانده نشده است.
- $q_{۱۰۱}$: M در این حالت قرار دارد اگر زیر رشته ۱۰۱ در رشته‌ی ورودی 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:

- $Q = \{q, q_{۱}, q_{۱۰}, q_{۱۰۱}\}$:
- $\Sigma = \{۰,۱\}$:
- : حالت شروع q است.
- : مجموعه پذیرش $F = \{q_{۱۰۱}\}$ است و
- : تابع انتقال δ با جدول زیر بیان می‌شود.

	۰	1
q	q	$q_{۱}$
$q_{۱}$	$q_{۱۰}$	$q_{۱}$
$q_{۱۰}$	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:

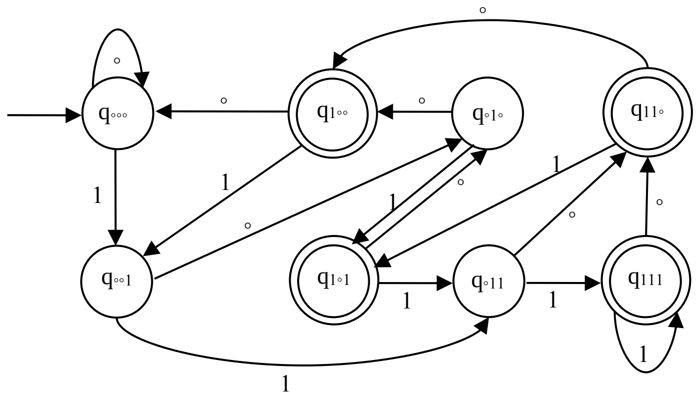
$$A = \{w \in \{0,1\}^* \mid \text{یک } 1 \text{ را در مکان سوم از راست دارد } w\}$$

که در آن $\{0,1\}^*$ عبارتست از مجموعه‌ی همه رشته‌های دودویی است و شامل رشته تهی ε 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 $\{0,1\}$ 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 $i = 0$.
- اگر M تنها یک نماد را خوانده باشد، آنگاه این نماد k است و به علاوه $i = j = 0$.
- اگر M هیچ نمادی را نخوانده باشد، آنگاه $i = j = k = 0$.

حالت شروع q_{000} است و مجموعه‌ی پذیرش‌ها عبارتست از $\{q_{100}, q_{110}, q_{101}, q_{111}\}$. تابع انتقال اتوماتای M در نمودار زیر نشان داده شده است.



6-4: 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.

1. اجتماع A و B عبارت است از: $A \cup B = \{w \mid w \in A \ \text{ or } \ w \in B\}$
2. 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).

3. ستاره A عبارتست از:

$$A^* = \{u_1 u_2 \dots u_k \mid k \geq 0 \ \text{ and } \ u_i \in A \ , \ i = 1, 2, \dots, k\}$$

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 ε می‌باشد. پس $\varepsilon \in A^*$. برای مثال فرض کنیم $A = \{0, 01\}$ و $B = \{1, 10\}$، آنگاه

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

به عنوان مثالی دیگر، اگر $\Sigma = \{0,1\}$، آنگاه Σ^* 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 $A^\circ = \{\varepsilon\}$ و برای $k \geq 1$، $A^k = AA^{k-1}$، یعنی A^k اتصال دو زبان A و A^{k-1} است. پس داریم $A^* = \bigcup\limits_{k=0}^{\infty} A^k$.

قضیه 6-4-1: مجموعه‌ی زبان‌های منظم تحت عمل اجتماع بسته است، یعنی اگر A و B زبان‌های منظم بر روی الفبای Σ 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 $w \in A \cup B$ و نتیجه می‌شود که M، w را پذیرش می‌کند. در غیر این صورت $w \notin A \cup B$ 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.
- Σ 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

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

- 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 δ از M داریم:

$$\overline{\delta}((q_1, q_2), w) \in F \Leftrightarrow \overline{\delta}_1(q_1, w) \in F_1 \ \text{ or } \ \overline{\delta}_2(q_2, w) \in F_2 \qquad (6\text{-}1)$$

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 (4.1) درست است و در واقع M زبان $A \bigcup B$ 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 بشکنند. 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.

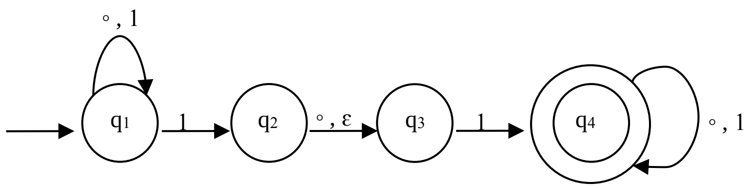
6-5: اتوماتای متناهی نامعین (Nondeterministic finite automata)

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.

6-5-1: چند مثال

مثال اول

نمودار زیر را در نظر می‌گیریم:

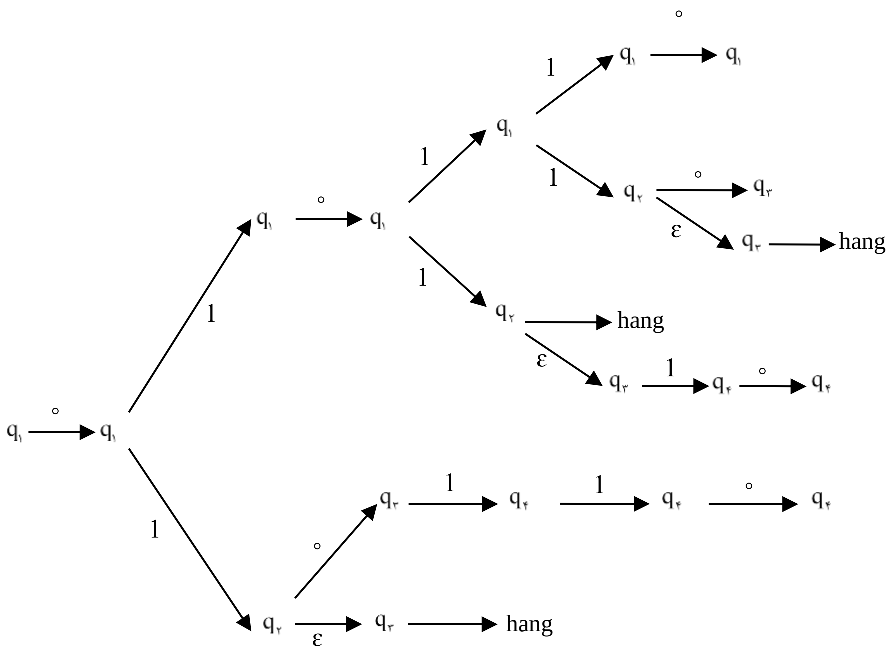


سه تفاوت با اتوماتای متناهی که تا حال دیده‌ایم مشاهده می‌شود. اول: اگر اتوماتا در حالت شروع q_1 باشد و نماد 1 را بخواند، آنگاه دو انتخاب دارد: یا در حالت شروع q_1 می‌ماند و یا به حالت شروع q_2 می‌رود. دوم: اگر اتوماتا در حالت q_2 باشد، آنگاه بدون خواندن یک نماد می‌تواند به حالت q_3 انتقال یابد. این امر با پیکانی نشان داده شده است که رشته تهی ε 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 1 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 $q_۱$ می‌ماند.
- وقتی نماد دوم را می‌خواند، اتوماتا در حالت $q_۲$ می‌ماند.
- اوتوماتا نماد سوم را نمی‌خواند (به بیان دیگر، رشته تهی ε را می‌خواند) و به حالت $q_۳$ می‌رود. در این لحظه اوتوماتا نمی‌تواند ادامه دهد. نماد سوم ۰ است، اما مسیری وجود ندارد که از $q_۳$ بیرون رود و با برچسب ۰ مشخص شده باشد و مسیری دیده نمی‌شود که از $q_۳$ بیرون رود و برچسب ε 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:

1. $q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_1 \xrightarrow{0} q_1$
2. $q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3$
3. $q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{\varepsilon} q_3 \longrightarrow$ 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.

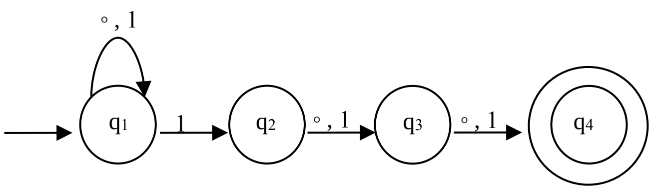
یک رشته‌ای که برای آن (i)، (ii) و (iii) برقرار نباشد، بوسیله NFA رد می‌شود. NFA ای که در بالا ارایه شد، همه رشته‌های دودویی که زیر رشته‌های ۱۰۱ یا 11 را در خود داشته باشند، پذیرش می‌کند؛ رشته‌های دیگر رد می‌شوند.

مثال دوم

زبان A را به صورت زیر در نظر می‌گیریم:

$$A = \{w \in \{0,1\}^* \mid \text{w نماد 1 را در مکان سوم از راست دارا است}\}$$

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 1 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).

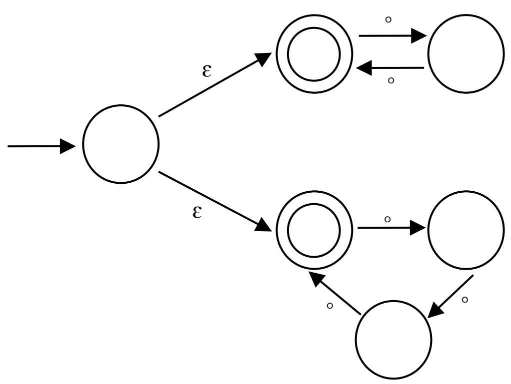
ملاحظه کنید که این روند حدس زدن چگونه به کار می‌رود:

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 1 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 3 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.

6-5-2: 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 Σ مجموعه Σ_ε 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 می-نویسیم.

$$P(Q)=\{R\,|\,R\subseteq Q\}$$

تعریف 6-5-3: یک اتوماتای متناهی نامعین (NFA) عبارتست از 5 تایی $M=\{Q,\Sigma,\delta,q,F\}$ که در آن:

1. Q یک مجموعه‌ی متناهی است و عناصرش را حالت‌ها نامند.
2. Σ یک مجموعه‌ی متناهی است که الفبا نام دارد و عناصرش را نمادها گویند.
3. تابع انتقال δ عبارتست از $\delta:Q\times\Sigma_{\varepsilon}\rightarrow P(Q)$
4. q عضوی از Q است و حالت شروع نام دارد.
5. F یک زیرمجموعه از Q است و عناصرش را حالت‌های پذیرش نامند.

مانند DFA ها، تابع انتقال δ 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 4-6:1 یک NFA است که در آن $Q=\{q_{۱},q_{۲},q_{۳},q_{۴}\}$ و $\Sigma=\{۰,۱\}$. حالت شروع $q_{۱}$ است و مجموعه حالت‌های پذیرش $F=\{q_{۴}\}$ است. تابع انتقال δ با جدول زیر بیان می‌گردد.

	۰	1	ε
$q_{۱}$	$\{q_{۱}\}$	$\{q_{۱},q_{۲}\}$	Q
$q_{۲}$	$\{q_{۳}\}$	Q	$\{q_{۳}\}$
$q_{۳}$	Q	$\{q_{۴}\}$	Q
$q_{۴}$	$\{q_{۴}\}$	$\{q_{۴}\}$	Q

تعریف 6-5-4: 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 $r_m,...,r_1,r_0$ 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 01100 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 6-5-5: فرض کنیم $M=\{Q,\Sigma,\delta,q,F\}$ یک NFA باشد. زبان $L(M)$ که توسط M پذیرفته شده است عبارتست از:

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

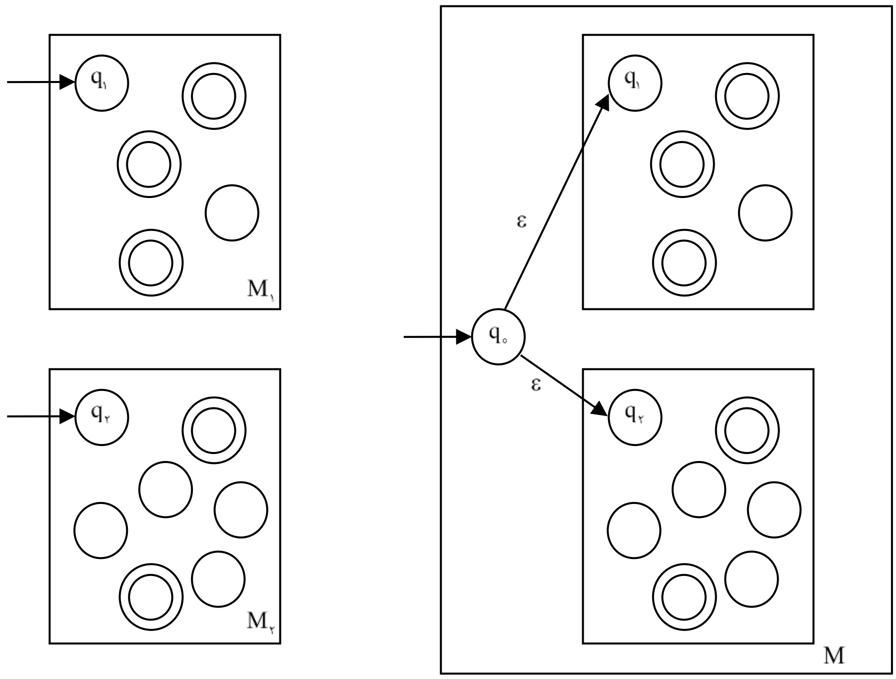
6-5-6: بسته بودن تحت اعمال منظم

قضیه 6-5-7: مجموعه‌ی زبان‌های منظم تحت عمل اجتماع بسته است. یعنی اگر A_1 و A_2 زبان‌های منظم بر روی الفبای Σ 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 6-1 این ساختار را نشان می‌دهد. این NFA به صورت زیر تعریف می‌شود:

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$، که در آن q_0 یک حالت جدید است.
2. q_0 حالت شروع M است.
3. $F = F_1 \cup F_2$ (حالت‌های پذیرش).
4. $\delta : Q \times \Sigma_\varepsilon \to P(Q)$ به صورت زیر تعریف می‌شود. به ازای هر $r \in Q$ و هر $a \in \Sigma_\varepsilon$،

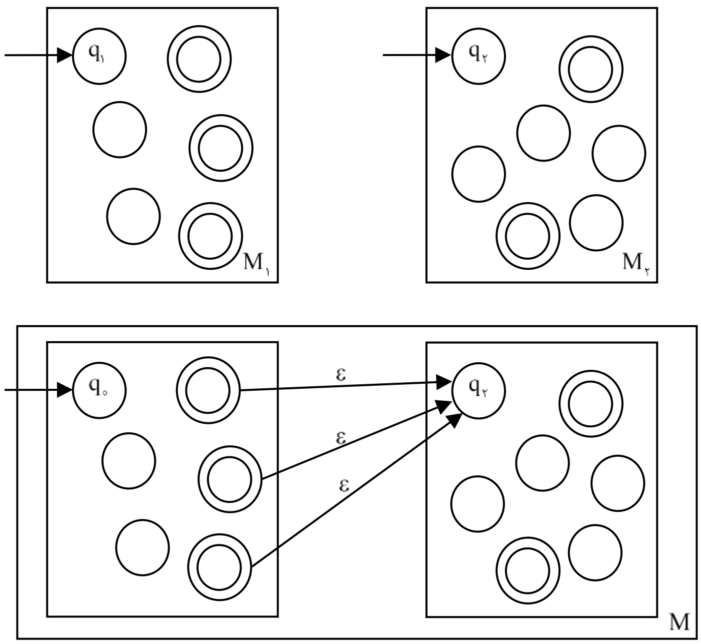
$$\delta(r,a) = \begin{cases} \delta_1(r,a) & \text{if} \quad r \in Q_1, \\ \delta_2(r,a) & \text{if} \quad r \in Q_2, \\ \{q_1, q_2\} & \text{if} \quad r = q_0, a = \varepsilon, \\ \phi & \text{if} \quad r = q_0, a \neq \varepsilon. \end{cases}$$



شکل 6-1: اتوماتای متناهی نامعین M زبان $L(M_1) \cup L(M_2)$ را می‌پذیرد.

قضیه 6-5-8: مجموعه‌ی زبان‌های منظم تحت عمل اتصال بسته است، یعنی اگر A_1 و A_2 زبان‌های منظم بر روی الفبای Σ 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 $L(M)=A_1A_2$. روش ساختن در شکل 6-2 نشان داده شده است.



شکل 6-2: اتوماتای متناهی نامعین M زبان $L(M_1)L(M_2)$ را می‌پذیرد.

این NFA به صورت زیر تعریف می‌شود:

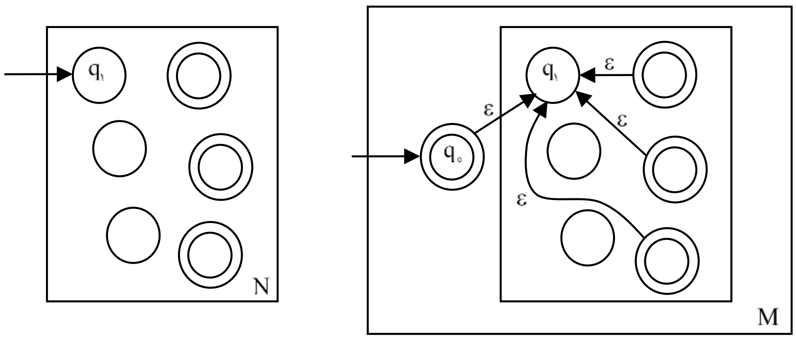
1. $Q=Q_1\cup Q_2$.

2. $q_0 = q_1$؛
3. $F = F_2$؛
4. تابع انتقال $\delta : Q \times \Sigma_\varepsilon \rightarrow P(Q)$ 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 6-5-9: مجموعه‌ی زبان‌های منظم تحت عمل ستاره بسته است، یعنی اگر A یک زبان منظم باشد، آنگاه A^* نیز زبان منظم است.

اثبات: فرض کنیم Σ 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 6-3 بیان شده است.



شکل 6-3: اتوماتای متناهی نامعین M زبان $(L(N))^*$ را می‌پذیرد.

این NFA به صورت زیر تعریف می‌شود:

1. $Q=\{q_0\}\cup Q_1$ که q_0 حالت جدید است.
2. q_0 حالت شروع M است.
3. $F=\{q_0\}\cup F_1$ (چون $\varepsilon\in A^*$، پس q_0 یک حالت پذیرش است).
4. تابع انتقال $\delta: Q\times\Sigma_\varepsilon \to P(Q)$ به صورت زیر تعریف می‌شود: به ازای هر $r\in Q$ و هر $a\in\Sigma_\varepsilon$،

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

قضیه 6-5-10: مجموعه زبان‌های منظم منظم تحت اعمال مکمل و اشتراک بسته‌اند.

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

6-6: 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 $(0\cup 1)01^*$ را در نظر می‌گیریم. زبانی که با این عبارت توصیف می‌شود، مجموعه‌ی همه رشته‌های دودویی است که:

1. با 1 یا ۰ شروع می‌شود (به دلیل وجود $(۰\cup۱)$)،
2. برای آن نماد دوم ۰ است (به دلیل وجود ۰) و
3. به صفر یا 1 های بیشتری ختم می‌شود (به دلیل وجود ۱^*).

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

$$\{۰۰,۰۰۱,۰۰۱۱,۰۰۱۱۱,...,۱۰,۱۰۱,۱۰۱۱,۱۰۱۱۱,...\}$$

حال چند مثال دیگر ارایه می‌شود (در همه این مثال‌ها الفبا مجموعه‌ی $\{۰,۱\}$ است).

- زبان $\{W \mid$ شامل دقیقاً دو ۰ است $W\}$ با عبارت زیر بیان می‌شود:

$$۱^*۰۱^*۰۱^*$$

- زبان $\{W \mid$ شامل حداقل دو ۰ است $W\}$ با عبارت زیر توصیف می‌شود:

$$(۰\cup۱)^*۰(۰\cup۱)^*۰(۰\cup۱)^*$$

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

$$(۰\cup۱)^*۱۰۱۱(۰\cup۱)^*$$

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

$$\left((۰\cup۱)(۰\cup۱)\right)^*$$

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

$$(۰\cup۱)\left((۰\cup۱)(۰\cup۱)\right)^*$$

- زبان $\{۱۰۱۱,۰\}$ عبارتست از:

$$۱۰۱۱\cup۰$$

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

$$۰(۰\cup۱)^*۰\cup۱(۰\cup۱)^*۱\cup۰\cup۱$$

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 6-6-1: فرض کنیم Σ یک الفبای غیر تهی باشد.

1. ε یک عبارت منظم است.
2. ϕ یک عبارت منظم است.
3. به ازای هر $a \in \Sigma$، a یک عبارت منظم است.
4. اگر R_1 و R_2 عبارات منظم باشند، آنگاه $R_1 \cup R_2$ یک عبارت منظم است.
5. اگر R_1 و R_2 عبارات منظم باشند، آنگاه R_1R_2 یک عبارت منظم است.
6. اگر R یک عبارت منظم باشد، آنگاه R^* نیز عبارت منظم است.

حال مثالی ارایه می‌کنیم. گوییم عبارت $(0\cup 1)^*101(0\cup 1)^*$ یک عبارت منظم است، الفبا مجموعه $\{0,1\}$ است. بدیهی است باید نشان دهیم که این عبارت در تعریف بالا صدق می‌کند. این کار را چنین انجام می‌دهیم:

- بنابر 3، 0 عبارت منظم است.
- بنابر 3، 1 عبارت منظم است.
- چون 0 و 1 عبارات منظم‌اند، بنابر 4، پس $0\cup 1$ نیز عبارت منظم است.
- چون بنابر 6، $0\cup 1$ منظم است پس $(0\cup 1)^*$ نیز منظم است.
- بنابر 5، چون 1 و 0 منظم‌اند، پس 10 نیز عبارت منظم است.
- بنابر 5، چون 10 و 1 منظم‌اند، پس 101 نیز عبارت منظم است.
- بنابر 5، چون $(0\cup 1)^*$ و 101 عبارات منظم‌اند، پس $(0\cup 1)^*101$ نیز یک عبارت منظم است.

- بنابر 5، چون $101(0\cup 1)^*(0\cup 1)$ و $(0\cup 1)^*$ عبارات منظم‌اند، پس $(0\cup 1)^*101(0\cup 1)^*(0\cup 1)$ نیز عبارت منظم است.

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

تعریف 6-6-2: فرض کنیم Σ یک الفبای غیر تهی باشد.

1. عبارت منظم ε، زبان $\{\varepsilon\}$ را بیان می‌کند.
2. عبارت منظم ϕ، زبان ϕ را بیان می‌کند.
3. به ازای هر $a\in\Sigma$، عبارت منظم a، زبان $\{a\}$ را بیان می‌کند.
4. 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.
5. 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_1L_2 را بیان می‌کند.
6. 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 $\{\varepsilon,1,11,111,...\}$ را توصیف می‌کند.

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

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

تعریف 6-6-3: 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 6-6-4: فرض کنیم R_1، R_2 و R_3 عبارات منظم باشند، آنگاه، اتحادهای زیر برقرار است.

1. $R_1\phi = \phi R_1 = \phi$
2. $R_1\varepsilon = \varepsilon R_1 = R_1$
3. $R_1 \cup \phi = \phi \cup R_1 = R_1$
4. $R_1 \cup R_1 = R_1$
5. $R_1 \cup R_2 = R_2 \cup R_1$
6. $R_1(R_2 \cup R_3) = R_1R_2 \cup R_1R_3$
7. $(R_1 \cup R_2)R_3 = R_1R_3 \cup R_2R_3$
8. $R_1(R_2R_3) = (R_1R_2)R_3$
9. $\phi^* = \varepsilon$
10. $\varepsilon^* = \varepsilon$
11. $(\varepsilon \cup R_1)^* = R_1^*$
12. $(\varepsilon \cup R_1)(\varepsilon \cup R_1)^* = R_1^*$
13. $R_1^*(\varepsilon \cup R_1) = (\varepsilon \cup R_1)R_1^* = R_1^*$
14. $R_1^*R_2 \cup R_2 = R_1^*R_2$
15. $R_1(R_2R_1)^* = (R_1R_2)^*R_1$
16. $(R_1 \cup R_2)^* = (R_1^*R_2)^*R_1^* = (R_2^*R_1)^*R_2^*$

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.

$$(0 \cup \varepsilon)1^* = 01^* \cup \varepsilon 1^* \qquad \text{(بنابر 7)}$$

$$= 01^* \cup 1^* \qquad \text{(بنابر 2)}$$

6-7: 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 6-7-1: لم تزریق برای زبان‌های منظم

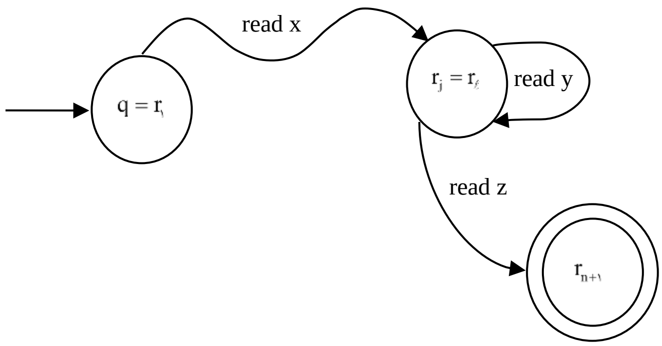
فرض کنیم A یک زبان منظم باشد، آنگاه عدد صحیحی مانند $P \geq ١$ وجود دارد که آن را طول تزریق گویند، به طوری که موارد زیر برقرار باشند. هر رشته‌ی $S \in A$، با $|S| \geq P$ می‌تواند به صورت $S = xyz$ نوشته شود، به طوری که:

1. $y \not\in \varepsilon$ (یعنی $|y| \geq ١$)،
2. $|xy| \leq P$،
3. به ازای هر $i \geq ٠$، $x y^{i} z \in A$.

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

اثبات: فرض کنیم Σ 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 $S = S_١ S_٢, ..., S_n$ یک رشته‌ی دلخواه در A باشد و $n \geq P$. تعریف می‌کنیم $r_١ = q$، $r_٢ = \delta(r_١, s_١)$، $r_٣ = \delta(r_٢, s_٢)$، ... و $r_{n+١} = \delta(r_n, s_n)$. بنابراین زمانی که M رشته‌ی S را از چپ به راست می‌خواند، حالت‌های $r_{n+١}, ..., r_٢, r_١$ را می‌بیند. چون S رشته‌ای در A است، پس $r_{n+١}$ به F تعلق دارد.

حال اولین $P+١$ رشته‌ی $r_{P+١}, ..., r_٢, r_١$ 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 ℓ طوری وجود دارند که $١ \leq j \leq \ell \leq P+١$ و $r_j = r_\ell$.



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

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

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 $i \geq 0$.

6-7-2: 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.

6-7-3: 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 0ها 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 0ها با تعداد 1ها 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 0ها را دارد. از

آنجا که $y \neq \varepsilon$، y حداقل یک ۰ دارد. پس رشته‌ی $xy^2z = xyyz$ شامل تعداد ۰های بیشتر از تعداد 1ها 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 $S = 0^P 1 0^P 1$ را در نظر می‌گیریم. بدیهی است $S \in A$ و $|S| = 2P + 2 \geq P$. با کمک لم تزریق $S = xyz$ که $y \notin \varepsilon$ و $|xy| \leq P$ و $xy^iz \in A$ به ازای هر $i \geq 0$. چون $|xy| \leq P$، رشته‌ی S تنها ۰ها را دارد از آنجا که $y \notin \varepsilon$، y حداقل یک ۰ 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 $S = 0^P 1 0^P 1$ به یک تناقض می‌رسیم، پس این رشته برای اثبات نامنظم بودن A انتخاب درستی است.

مثال چهارم

زبان $A = \{0^m 1^n \mid m > n \geq 0\}$ 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 $y \notin \varepsilon$، $|xy| \leq P$ و $xy^iz \in A$، به ازای هر $i \geq 0$.

چون $|xy| \le P$، رشته‌ی y تنها شامل ۰ها است. از آنجا که $y \not\in \varepsilon$، y حداقل یک ۰ دارد. حال رشته‌ی $xy^{\circ}z = xz$ را در نظر می گیریم. تعداد 1ها در این رشته برابر P است، در حالی که تعداد ۰ها حداکثر برابر P می‌باشد. در نتیجه رشته‌ی $xy^{\circ}z$ 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 $A = \{۱^{n^۲} \mid n \ge ۰\}$ 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 $P \ge ۱$. رشته‌ی $S = ۱^{P^۲}$ را بررسی می‌کنیم. بدیهی است $S \in A$ و $|S| = P^۲ \ge P$. می‌نویسیم $S = xyz$ که $y \not\in \varepsilon$ و $|xy| \le P$ و $xy^{i}z \in A$ به ازای هر $i \ge ۰$ داریم $|S| = |xyz| = P^۲$ و $|xy^۲z| = |xyyz| = |xyz| + |y| = P^۲ + |y|$. چون $|xy| \le P$، پس $|y| \le P$. از آنجا که $y \not\in \varepsilon$ داریم $|y| \ge ۱$. لذا نتیجه می‌شود:

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

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 $A = \{۱^n \mid n$ یک عدد اول است$\}$ در نظر می‌گیریم و ثابت می‌کنیم که A یک زبان منظم نیست. فرض کنیم A منظم باشد و طول تزریق $P \ge ۱$. فرض کنیم $n \ge P$ یک عدد اول باشد، و

رشته $S=1^n$ را در نظر می‌گیریم. بدیهی است $S\in A$ و $|S|=n\geq P$. حال می‌نویسیم $S=xyz$ که $y\notin\varepsilon$ و $|xy|\leq P$ و $xy^iz\in A$ به ازای هر $i\geq 0$.

اگر K یک عدد صحیح باشد به طوری که $y=1^K$. چون $y\notin\varepsilon$ داریم $K\geq 1$. حال برای هر $i\geq 0$، عدد $n+(i-1)K$ یک عدد اول است، زیرا $xy^iz=1^{n+(i-1)K}\in A$. برای $i=n+1$ داریم:

$$n+(i-1)K=n+nK=n(1+K)$$

که یک عدد اول نیست، زیرا $n\geq 2$ و $1+K\geq 2$. این تناقض ثابت می‌کند A منظم نیست.

قضیه 6-8: قضیه هیگمن (Higman's theorem)

فرض کنیم Σ یک الفبای متناهی باشد، برای هر دو رشته‌ی x و y در Σ^* گوییم، x یک زیر دنباله از y است، اگر x بتواند با حذف صفر و یا نمادهای بیشتر از y بدست آید. برای مثال، ۱۰۱۱۰ یک زیر دنباله از ۰۰۱۰۰۱۰۱۰۱۰۱۰۰۰۱ است. برای هر زبان $L\subseteq\Sigma^*$، تعریف می‌کنیم:

$SUBSEQ(L)=\{x\,|$ یک $y\in L$ وجود داشته باشد، به طوری که x یک زیر دنباله از y باشد$\}$

یعنی $SUBSEQ(L)$ زبانی است که شامل همه زیر دنباله‌های رشته‌های موجود در L است. در 1952، هیگمن (Higman) قضیه زیر را بیان و اثبات نمود.

قضیه 6-8-1: Higman

برای الفبای متناهی Σ و هر زبان $L\subseteq\Sigma^*$، زبان $SUBSEQ(L)$ منظم است.

قضیه 6-8-2: قضیه دیکسون (Dickson's theorem)

در اثبات قضیه هیگمن، از قضیه‌ای استفاده می‌شود که در 1913 توسط دیکسون مطرح گردید.

فرض کنیم $n \in N$، برای هر دو نقطه‌ی $p=(p_1,p_2,...,p_n)$ و $q=(q_1,q_2,...,q_n)$ در N^n ، گوییم q بر p غالب است، اگر $p_i \le q_i$، برای هر $1 \le i \le n$.

قضیه 6-8-3: دیکسون (Dickson)

فرض کنیم $S \subseteq N^n$، و M مجموعه‌ای شامل همه عناصر S باشد که مینیمال هستند، (بر اساس تعریف غالب پذیری)، آنگاه

$M=\{q \in S|$ ای در $S \setminus \{q\}$ وجود ندارد به طوری که q بر p غالب باشد $p\}$

(تذکر: $A \setminus B = \{x \mid x \in A, x \notin B\}$)

پس مجموعه M متناهی است.

اثبات: این قضیه را به استقراء ثابت می‌کنیم. اگر $n=1$، آنگاه $M=\phi$ (اگر $S=\phi$) یا M دقیقاً شامل یک عنصر است (اگر $S \ne \phi$). بنابراین اگر $n=1$ قضیه برقرار است. فرض کنیم $n \ge 2$ و قضیه برای همه زیر مجموعه‌های N^{n-1} برقرار باشد. فرض کنیم S یک زیرمجموعه از N^n، و مجموعه‌ی M اعضای مینیمال در S باشند. اگر $S=\phi$، آنگاه $M=\phi$ و در نتیجه M متناهی است. فرض کنیم $S \ne \phi$: عنصر دلخواه q از M را مشخص می‌کنیم. اگر $p \in M \setminus \{q\}$، آنگاه q بر p غالب نیست. در نتیجه، اندیسی مانند i طوری وجود دارد که $p_i \le q_i - 1$. نتیجه می‌شود:

$$M \setminus \{q\} \subseteq \bigcup_{i=1}^{n} (N^{i-1} \times [1, q_i - 1] \times N^{n-i})$$

برای همه‌ی i و k ها که $1 \le i \le n$ و $1 \le k \le q_i - 1$، تعریف می‌کنیم:

$$S_{ik} = \{p \in S \mid p_i = k\},$$
$$M_{ik} = \{p \in M \mid p_i = k\}$$

آنگاه

$$M \setminus \{q\} = \bigcup_{i=1}^{n} \bigcup_{k=1}^{q_i - 1} M_{ik} \qquad (6\text{-}4)$$

لم 6-8-4: M_{ik} یک زیر مجموعه از مجموعه‌ی همه عناصر S_{ik} است که بر حسب رابطه غالب‌پذیری، مینیمال هستند.

اثبات: فرض کنیم p عضوی از M_{ik} باشد و p در S_{ik} مینیمال نباشد. آنگاه عضوی مانند r در S_{ik} وجود دارد که $r \neq p$ و p بر r غالب است. چون p و r هر دو اعضای S هستند، پس $p \notin M$ که این تناقض است. چون S_{ik} یک زیرمجموعه از N^{n-1} است، از فرض استقراء نتیجه می‌شود که S_{ik} شامل تعداد متناهی عناصر مینیمال است. این موضوع به همراه لم 6-8-4 نشان می‌دهد که M_{ik} یک مجموعه‌ی متناهی است. پس بنابر (6-4)، $M \setminus \{q\}$ اجتماعی از تعداد متناهی از مجموعه‌های متناهی است. در نتیجه مجموعه‌ی M متناهی است.

6-8-5: اثبات قضیه هیگمن

اثبات قضیه 6-8-1 را برای حالتی که $\Sigma = \{0,1\}$ ارایه می‌کنیم. اگر $L = \phi$ یا $SUBSEQ(L) = \{0,1\}^*$، آنگاه $SUBSEQ(L)$ به طور بدیهی یک زبان منظم است. از این روی، می‌توانیم فرض کنیم که L تهی نیست و $SUBSEQ(L)$ یک زیرمجموعه‌ی محض $\{0,1\}^*$ است.

یک رشته z به طول حداقل 2 را در مکمل $\overline{SUBSEQ(L)}$ از زبان $SUBSEQ(L)$ بر می‌داریم. مشاهده می‌کنیم که این امر امکان پذیر است، زیرا $\overline{SUBSEQ(L)}$ یک زبان نامتناهی است. عناصر 0 ها، 1ها را در z وارد می‌کنیم، به طوری که در رشته‌ی حاصل z'، 0ها و 1ها متناوب باشند.

برای مثال، اگر $z = 0011101011$، آنگاه $z' = 0101010101010101$. فرض کنیم $n = |z'| - 1$ که $|z'|$ یعنی طول z'، آنگاه $n \geq |z| - 1 \geq 1$.

$A(0,1)$– تناوب یک رشته‌ی دودویی x عبارتست از هر رخداد از 01 یا 10 در x است. برای مثال، رشته‌ی 1101001 شامل چهار تناوب $(0,1)$ است. تعریف می‌کنیم

$$A = \{x \in \{0,1\}^* \mid \text{حداکثر } n \text{ تا } (0,1) \text{ تناوبی دارد } x\}$$

لم 6-8-6: $\mathrm{SUBSEQ(L)} \subseteq \mathrm{A}$

اثبات: اگر $\mathrm{x} \in \mathrm{SUBSEQ(A)}$ و فرض کنیم $\mathrm{x} \notin \mathrm{A}$، آنگاه x حداقل تعداد $|\mathrm{z'}| = \mathrm{n}+۱$، (۰,۱) تناوبی دارد و در نتیجه 'z یک زیر دنباله از x است. به خصوص z یک زیر دنباله از x است. چون $\mathrm{x} \in \mathrm{SUBSEQ(L)}$، نتیجه می‌شود که $\mathrm{z} \in \mathrm{SUBSEQ(L)}$ که تناقض است.

لم 7-8-6: $\overline{\mathrm{SUBSEQ(L)}} = (\mathrm{A} \cap \overline{\mathrm{SUBSEQ(L)}}) \cup \overline{\mathrm{A}}$

اثبات: از لم 6-8-6 نتیجه می‌شود.

لم 8-8-6: زبان $\overline{\mathrm{A}}$ منظم است.

اثبات: مکمل $\overline{\mathrm{A}}$ از A زبانی است که شامل همه رشته‌های دودویی است که حداقل تعداد n+۱، (۰,۱) تناوبی دارد. اگر برای مثال، $\mathrm{n} = ۳$، آنگاه $\overline{\mathrm{A}}$ با عبارت منظم زیر توصیف می‌شود:

$$(۰۰^* ۱۱^* ۰۰^* ۱۱^* ۰(۰ \cup ۱)^*) \cup (۱۱^* ۰۰^* ۱۱^* ۰۰^* ۱(۰ \cup ۱)^*)$$

این ما را متقاعد می‌کند که ادعای مطرح شده به ازای هر مقدار n درست است.

6-9: تمرینات فصل ششم

1. برای هریک از زبان‌های زیر، یک DFA بسازید که این زبان‌ها را بپذیرد. در همه حالات الفبا عبارت است از $\{0,1\}$
   1. $\{w \mid$ طول w بر 3 تقسیم‌پذیر باشد$\}$
   2. $\{w \mid$ ۱۱۰ یک زیر رشته از w نباشد$\}$
   3. $\{w \mid$ w شامل حداقل 5 تا 1 باشد$\}$
   4. $\{w \mid$ w شامل رشته‌ی ۱۰۱۱ باشد$\}$
   5. $\{w \mid$ w شامل حداقل دو تا 1 و حداکثر دو تا ۰ باشد$\}$
   6. $\{w \mid$ w شامل تعداد فردی از 1ها یا دقیقاً دو تا ۰ باشد$\}$
   7. $\{w \mid$ w با 1 شروع و به ۰ ختم شود$\}$
   8. $\{w \mid$ هر مکان فرد در w عدد 1 را داشته باشد$\}$
   9. $\{w \mid$ w حداقل طولی برابر 3 داشته و سومین نماد آن ۰ باشد$\}$
   10. $\{\varepsilon, 0\}$

2. برای هر یک از زبان‌های زیر یک NFA بسازید، با تعداد حالت‌های مشخص شده که زبان را بپذیرد. الفبا در همه حالات $\{0,1\}$ است.
   1. زبان $\{w \mid$ w به ۱۰ ختم شود$\}$، با سه حالت.
   2. زبان $\{w \mid$ w زیر رشته‌ی ۱۰۱۱ را داشته باشد$\}$، با پنج حالت.
   3. زبان $\{w \mid$ w شامل تعداد فردی از 1ها یا دقیقاً دو تا ۰$\}$، با شش حالت.

3. برای هر یک از زبان‌های زیر یک NFA بسازید، که زبان‌ها را بپذیرد. الفبا در همه حالات $\{0,1\}$ است.
   1. $\{w \mid$ w شامل زیر رشته‌ی ۱۱۰۰۱ باشد$\}$
   2. $\{w \mid$ طول w حداقل دو و به ۱۰ ختم نشود$\}$
   3. $\{w \mid$ w با 1 شروع شود یا به ۰ ختم گردد$\}$

4. فرض کنیم A و B دو زبان منظم بر روی Σ باشند. ثابت کنید که تفاضل آنها یعنی $A \setminus B = \{w \mid w \in A \text{ and } w \notin B\}$ یک زبان منظم است.

5. برای هر یک از عبارات منظم زیر، دو رشته ارایه کنید که اعضای زبان‌های بیان شده باشند و دو رشته نشان دهید که نباشند. الفبا $\Sigma = \{a, b\}$.
   1. $a(ba)^* b$
   2. $(a \cup b)^* a(a \cup b)^* b(a \cup b)^* a(a \cup b)^*$
   3. $(a \cup ba \cup bb)(a \cup b)^*$

6. عبارت منظم هر یک از زبان‌های زیر را بیان کنید. الفبا $\{0,1\}$ است.
   1. $\{w \mid$ حداقل دارای سه تا 1 باشد $w\}$
   2. $\{w \mid$ شامل حداقل دو تا 1 و حداکثر یک 0 باشد $w\}$
   3. $\{w \mid$ شامل تعداد زوج 0 ها و دقیقاً دو تا 1 باشد $w\}$
   4. $\{w \mid$ شامل دقیقاً دو تا 0 و حداقل دو تا 1 باشد $w\}$
   5. $\{w \mid$ شامل تعداد زوج 0 ها باشد و هر 0 به دنبال حداقل یک 1 بیاید $w\}$
   6. $\{w \mid$ هر مکان فرد در w عدد 1 باشد$\}$

7. با استفاده از لم تزریق ثابت کنید که زبان‌های زیر منظم نیستند.
   1. $\{a^n b^m c^{n+m} \mid n \geq 0, m \geq 0\}$
   2. $\{a^n b^n c^{2n} \mid n \geq 0\}$
   3. $\{a^n b^m c^n \mid n \geq 0, m \geq 0\}$
   4. $\{a^{2^n} \mid n \geq 0\}$ (توجه: a^{2^n} رشته‌ای شامل 2^n تا از aها است)
   5. $\{a^n b^m c^k \mid n \geq 0, m \geq 0, k \geq 0, n^2 + m^2 = k^2\}$
   6. $\{avu \mid u \in \{0, b\}^*, u \neq \varepsilon, v \in \{a, b\}^*\}$

8. فرض کنید A یک زبان شامل تعداد متناهی رشته باشد.

1. ثابت کنید A یک زبان منظم است.
2. فرض کنید n ماکزیمم طول هر رشته در A باشد. ثابت کنید هر DFA که A را پذیرش کند حداقل $n+1$ حالت دارد (راهنمایی: از طول تزریق ارایه شده در اثبات قضیه لم تزریق استفاده کنید).

9. فرض کنید L یک زبان منظم و M یک DFA باشد که این زبان را می‌پذیرد و فرض کنید P تعداد حالت‌های M باشد، آنگاه ثابت کنید $L \neq \phi$ اگر و تنها اگر L دارای یک رشته به طول کمتر از P باشد.

10. فرض کنید L یک زبان منظم و M یک DFA باشد که این زبان را می‌پذیرد و فرض کنید P تعداد حالت‌های M باشد، آنگاه ثابت کنید L یک زبان نامتناهی است اگر و تنها اگر L شامل یک رشته مانند w با شرایط زیر باشد.

$$P \leq |w| \leq 2P-1$$