به نام خدا

# نظریهٔ زبان‌ها و ماشین‌ها

آرش شفیعی

# زبان‌های منظم

## مقدمه

- پیشتر، از پذیرنده‌های متناهی برای شناسایی زبان‌های منظم استفاده کردیم.
- در این قسمت از عبارت‌های منظم و گرامرهای منظم برای توصیف زبان‌های منظم استفاده می‌کنیم.

# زبان‌های منظم

- یکی از روش‌های توصیف زبان‌های منظم، استفاده از عبارت‌های منظم [1] است.
- یک عبارت منظم تشکیل شده است از رشته‌هایی بر روی الفبای $\Sigma$، پرانتزهای باز و بسته، و عملگرهای مثبت (+) و نقطه (.) و ستاره(*).
- برای مثال، زبان $\{a\}$ را با عبارت منظم $a$ نشان می‌دهیم.
- عملگر مثبت (+) در عبارت‌های منظم به معنی اجتماع دو زبان (دو مجموعه) است. بنابراین زبان $\{a, b, c\}$ با عبارت منظم $a + b + c$ نمایش داده می‌شود.
- از عملگر نقطه (.) برای الحاق دو عبارت منظم و از عملگر ستاره (*) برای بستار-ستاره بر روی یک عبارت منظم استفاده می‌شود.
- عبارت $(a + (b \cdot c))^*$ به معنی بستار ستاره روی زبان $\{a\} \cup \{bc\}$ است، یعنی $L^*$ جایی که $L = \{a, bc\}$. بنابراین زبان مورد نظر برابر است با $\{\lambda, a, bc, aa, abc, bca, bcbc, aaa, aabc, \cdots\}$.

---

[1] regular expressions

# زبان‌های منظم

- فرض کنید $\Sigma$ یک الفبا باشد. آنگاه

۱. $\emptyset$، $\lambda$ و $a \in \Sigma$ عبارت‌های منظم هستند که به آنها عبارت‌های منظم اولیه [1] می‌گوییم.

۲. اگر $r_1$ و $r_2$ دو عبارت منظم باشند، آنگاه $r_1 + r_2$، $r_1 \cdot r_1$، $r_1^*$ و $(r_1)$ نیز عبارت‌های منظم هستند.

۳. یک عبارت شامل عملگرها و نمادهای الفبا را یک عبارت منظم می‌گوییم اگر و فقط اگر بتوان آن را از عبارت‌های منظم اولیه با اعمال تعداد محدودی از قوانین مذکور در مرحلهٔ ۲ به دست آورد.

[1] primitive regular expressions

## زبان‌های منظم

- اگر $\Sigma = \{a, b, c\}$ آنگاه رشتهٔ $(a + b \cdot c)^* \cdot (c + \emptyset)$ یک عبارت منظم است،
چرا که می‌توان آن را با اعمال چند قانون پی در پی از عبارت‌های منظم اولیه به دست آورد.
- ولی عبارت $(a + b+)$ یک عبارت منظم نیست.
- اگر $r$ یک عبارت منظم باشد، آنگاه $L(r)$ نشان دهندهٔ زبان مربوط به عبارت منظم $r$ است.

# زبان‌های منظم

- زبان $L(r)$ که توسط عبارت منظم $r$ تعیین شده است، با استفاده از قوانین زیر تعریف می‌شود:

۱. $\emptyset$ یک عبارت منظم است که نشان‌دهندهٔ زبان تهی $\{\}$ است.

۲. $\lambda$ یک عبارت منظم است که نشان‌دهندهٔ زبان $\{\lambda\}$ است.

۳. به ازای هر $a \in \Sigma$، $a$ یک عبارت منظم است که نشان‌دهندهٔ زبان $\{a\}$ است.

- اگر $r_1$ و $r_2$ دو عبارت منظم باشند، آنگاه

۴. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$

۵. $L(r_1 \cdot r_2) = L(r_1)L(r_2)$

۶. $L((r_1)) = L(r_1)$

۷. $L(r_1^*) = (L(r_1))^*$

# زبان‌های منظم

- همچنین می‌توانیم از بستار مثبت روی یک عبارت منظم استفاده کنیم:
$$r^{+} = rr^{*}$$

- برای الحاق و اجتماع و بستار–ستاره روابط زیر برقرار است:
$$r + \lambda = r + \lambda$$
$$r\lambda = r$$
$$\lambda^{*} = \lambda$$
$$r + \emptyset = r$$
$$r\emptyset = \emptyset$$
$\emptyset^{*} = \lambda$ (این رابطه یک قرارداد است مانند قرارداد ۱ $=$ ۰°)

# زبان‌های منظم

- زبان $L(a^* \cdot (a+b))$ را با استفاده از مجموعه‌ها نشان دهید.

- زبان $L(a^* \cdot (a+b))$ را با استفاده از مجموعه‌ها نشان دهید.

$$\begin{aligned} L(a^* \cdot (a+b)) &= L(a^*)L((a+b)) \\ &= (L(a))^*(L(a) \cup L(b)) \\ &= \{\lambda, a, aa, \cdots\} \cdot \{a, b\} \\ &= \{a, aa, \cdots, b, ab, aab, \cdots\} \\ &= \{a^n : n \geq ۱\} \cup \{a^n b : n \geq ۰\} \end{aligned}$$

# زبان‌های منظم

- زبان $L(a \cdot b + c)$ را با استفاده از مجموعه‌ها نشان دهید.

# زبان‌های منظم

- زبان $\mathrm{L}(\mathrm{a}\cdot\mathrm{b}+\mathrm{c})$ را با استفاده از مجموعه‌ها نشان دهید.
- بسته به تقدم عملگرها می‌توانیم این زبان را با $\{\mathrm{ab},\mathrm{c}\}$ یا $\{\mathrm{ab},\mathrm{ac}\}$ نشان دهیم.
- در تعریف عبارات منظم از تقدم صحبت نکردیم. با استفاده از تعریف داده شده عبارات منظم برای رفع ابهام یا باید کاملا پرانتزگذاری شده باشند، و یا اینکه عملگرهای سمت چپ تقدم بیشتری داشته باشند.
- برای سادگی، ما در اینجا تقدمی مانند عملیات ریاضی در نظر می‌گیریم. یعنی بستار ستاره (که مانند توان است) نسبت به الحاق ( که مانند ضرب است) تقدم بیشتری دارد و همچنین الحاق نسبت به اجتماع (که مانند جمع است) تقدم بیشتری دارد.
- همچنین برای سادگی برای الحاق عملگر نقطه (.) را حذف می‌کنیم و می‌نویسیم ab به جای $\mathrm{a}\cdot\mathrm{b}$ .

# زبان‌های منظم

- زبان متناظر با عبارت منظم $(a+b)^*(a+bb)$ را با استفاده از مجموعه‌ها نشان دهید.

# زبان‌های منظم

- زبان متناظر با عبارت منظم $(a+b)^*(a+bb)$ را با استفاده از مجموعه‌ها نشان دهید.
- قسمت اول یعنی $(a+b)^*$ نشان دهندهٔ همه رشته‌ها بر روی الفبای $a$ و $b$ است.
- این رشته‌ها در پایان به $a$ یا $bb$ الحاق می‌شوند.
- بنابراین زبان مورد نظر زبانی است از همهٔ رشته‌هایی که با $a$ یا $bb$ خاتمه پیدا می‌کنند.
- $\{wa : w \in \{a,b\}^*\} \cup \{wbb : w \in \{a,b\}^*\} = \{a, bb, aa, abb, ba, bbb, \cdots\}$

## زبان‌های منظم

- زبان متناظر با عبارت منظم $r = (aa)^*(bb)^*b$ را با استفاده از مجموعه‌ها نشان دهید.

# زبان‌های منظم

- زبان متناظر با عبارت منظم $r = (aa)^*(bb)^*b$ را با استفاده از مجموعه‌ها نشان دهید.
- $L(r) = \{a^{۲n}b^{۲m+۱} : n \geq ۰, m \geq ۰\}$

# زبان‌های منظم

- بر روی الفبای $\Sigma = \{0, 1\}$ عبارت منظم r را پیدا کنید به طوری که
- $L(r) = \{w \in \Sigma^* :$ حداقل دو صفر پی در پی داشته باشد w$\}$

# زبان‌های منظم

- بر روی الفبای $\Sigma = \{0, 1\}$ عبارت منظم $r$ را پیدا کنید به طوری که
- $L(r) = \{w \in \Sigma^* :$ حداقل دو صفر پی در پی داشته باشد $w\}$
- $r = (0 + 1)^* 00 (0 + 1)^*$

# زبان‌های منظم

- عبارت منظم $r$ را پیدا کنید به طوری که
- $L(r) = \{w \in \{0, 1\}^* :$ دو صفر پی در پی ندارد $w\}$

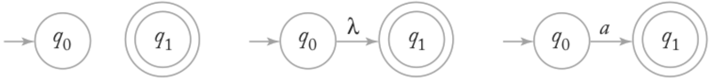
- عبارت منظم $r$ را پیدا کنید به طوری که
- $L(r) = \{w \in \{۰, ۱\}^* : w \text{ دو صفر پی در پی ندارد}\}$
- $r = (۱ + ۰۱)^*(۰ + \lambda)$

# عبارت‌ها و زبان‌های منظم

- فرض کنید r یک عبارت منظم باشد. آنگاه یک ماشین متناهی غیرقطعی وجود دارد که زبان $L(r)$ را می‌پذیرد. در اینصورت $L(r)$ یک زبان منظم است.

# عبارت‌ها و زبان‌های منظم

- فرض کنید r یک عبارت منظم باشد. آنگاه یک ماشین متناهی غیرقطعی وجود دارد که زبان $L(r)$ را می‌پذیرد. در اینصورت $L(r)$ یک زبان منظم است.
- از برهان با ساخت استفاده می‌کنیم یعنی روشی ارائه می‌کنیم که از هر عبارت منظم بتوان یک ماشین غیرقطعی به دست آورد.
- با ماشین‌هایی شروع می‌کنیم که عبارت‌های منظم اولیهٔ $\emptyset$، $\lambda$ و $a \in \Sigma$ را می‌پذیرند.
- این پذیرنده‌ها در زیر نشان داده شده‌اند.

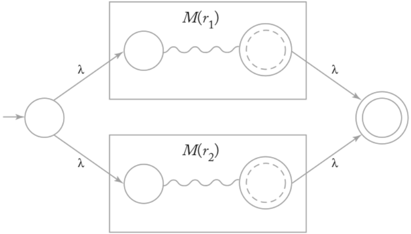


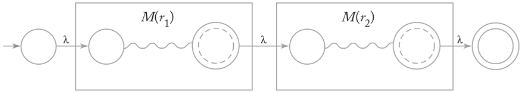
# عبارت‌ها و زبان‌های منظم

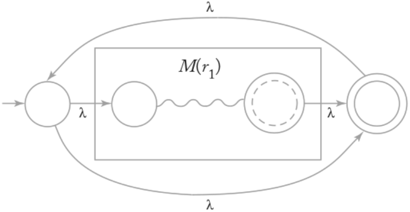
- اکنون فرض کنید ماشین‌های $M(r_1)$ و $M(r_2)$ زبان‌های منظمی را می‌پذیرند که با عبارت‌های منظم $r_1$ و $r_2$ نشان داده می‌شوند. برای سادگی در روند اثبات، هر ماشین ان‌اف‌ای را با معادل آن که تنها یک حالت پایانی دارد نشان می‌دهیم (با اتصال حالت‌های پایانی توسط رشته تهی به یک حالت پایانی واحد، این کار همیشه ممکن است).
- اکنون ماشین‌هایی می‌سازیم که زبان‌هایی را می‌پذیرند که با عبارت‌های منظم $r_1 + r_2$، $r_1 r_2$ و $r_1^*$ نمایش داده می‌شوند.

# عبارت‌ها و زبان‌های منظم

- سه ماشین متناهی غیرقطعی برای پذیرفتن زبان‌های متناظر سه عبارت منظم $r_1 + r_2$، $r_1r_2$ و $r_1^*$ به صورت زیر ساخته می‌شوند.



# عبارت‌ها و زبان‌های منظم

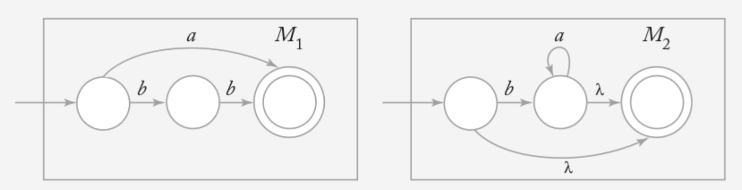
- بنابراین برای هر عبارت منظم پیچیده‌ای می‌توانیم از ترکیب کردن این مراحل استفاده کنیم و ماشین متناهی غیرقطعی را که زبان متناظر با آن عبارت منظم را می‌پذیرد را بسازیم.

# عبارت‌ها و زبان‌های منظم

- یک ان‌اف‌ای طراحی کنید که زبان $L(r)$ را بپذیرد، به طوری که $r = (a + bb)^*(ba^* + \lambda)$

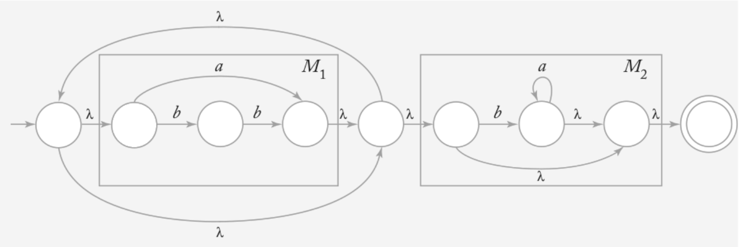
# عبارت‌ها و زبان‌های منظم

- یک ان‌اف‌ای طراحی کنید که زبان $L(r)$ را بپذیرد، به طوری که $r = (a + bb)^*(ba^* + \lambda)$
- ابتدا ماشین‌هایی طراحی می‌کنیم که عبارت‌های $(a + bb)$ و $(ba^* + \lambda)$ را بپذیرند.



## عبارت‌ها و زبان‌های منظم

- یک ان‌اف‌ای طراحی کنید که زبان $L(r)$ را بپذیرد، به طوری که $r = (a + bb)^*(ba^* + \lambda)$
- ابتدا ماشین‌هایی طراحی می‌کنیم که عبارت‌های $(a + bb)$ و $(ba^* + \lambda)$ را بپذیرند.
- سپس با قوانین بستار و الحاق دو ماشین طراحی شده را با هم ترکیب می‌کنیم.



# عبارت‌ها و زبان‌های منظم

- پس تا اینجا توانستیم برای هر عبارت منظم یک ماشین متناهی طراحی کنیم و از آنجایی که هر ماشین متناهی یک زبان منظم را می‌پذیرد، بنابراین برای هر عبارت منظم یک زبان منظم وجود دارد.

- آیا برای هر زبان منظم نیز یک عبارت منظم وجود دارد؟

# عبارت‌ها و زبان‌های منظم

- از آنجایی که هر زبان منظم را می‌توان توسط یک ماشین متناهی نشان داد و هر ماشین متناهی یک گراف گذار دارد، پس برای به دست آوردن عبارت منظم متناظر آن کافی است همهٔ گشت‌ها از رأس آغازی به تمام حالات پایانی را با عبارات منظم وصف کرد.
- برای انجام این کار در اینجا از گراف گذار تعمیم‌یافته [1] استفاده می‌کنیم.
- ابتدا گراف گذار تعمیم‌یافته را تعریف می‌کنیم و سپس روشی برای تبدیل یک ماشین غیرقطعی به یک گراف گذار تعمیم‌یافته ارائه می‌کنیم.
- سپس روشی برای یافتن عبارت منظم متناظر با گراف گذار تعمیم‌یافته ارائه می‌کنیم.
- پس برای هر زبان منظم یک ماشین غیرقطعی وجود دارد و برای هر ماشین غیرقطعی یک گراف گذار تعمیم‌یافته و برای هر گراف گذار تعمیم‌یافته یک عبارت منظم.

[1] generalized transition graph (GTG) or generalized nondeterministic finite automaton (GNFA)

# عبارت‌ها و زبان‌های منظم

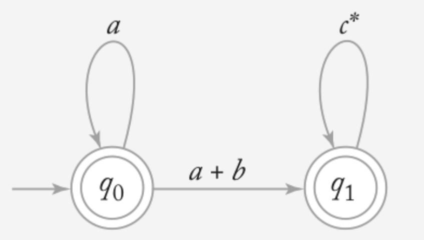
- گراف گذار تعمیم‌یافته یک گراف گذار است که برچسب یال‌های آن عبارت‌های منظم هستند.
- بنابراین برچسب هر گشت از حالت آغازی به یک حالت پایانی الحاق همهٔ عبارات منظم آن گشت است.
- تمام رشته‌هایی که توسط آن عبارات منظم به دست می‌آیند، عضوی از زبانی هستند که توسط گراف گذار شناسایی می‌شوند.
- مجموعهٔ همهٔ عبارات منظمی که در این گراف گذار پذیرفته می‌شوند، زبان متناظر با آن را می‌سازند.
- بنابراین هر زبانی که توسط یک گراف تعمیم یافته شناسایی می‌شود باید منظم باشد، زیرا جملات آن توسط عبارات منظم به دست می‌آیند.

# عبارت‌ها و زبان‌های منظم

- یک گراف گذار تعمیم‌یافته با دو حالت برای زبان منظم $L(a^* + a^*(a+b)c^*)$ بیابید.

# عبارت‌ها و زبان‌های منظم

- یک گراف گذار تعمیم‌یافته با دو حالت برای زبان منظم $L(a^* + a^*(a+b)c^*)$ بیابید.
- می‌توانیم به جای حلقه با برچسب a از برچسب $a^*$ نیز استفاده کنیم.

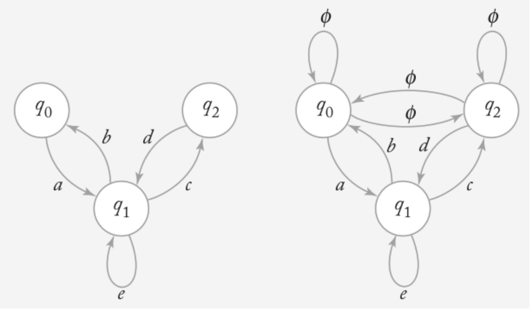


# عبارت‌ها و زبان‌های منظم

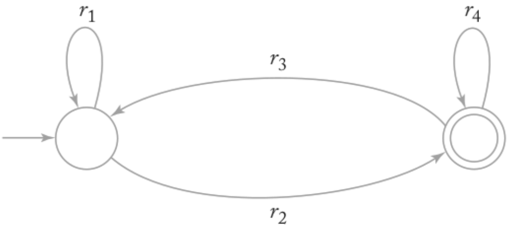
- هر گراف گذار یک پذیرندهٔ متناهی غیرقطعی را می‌توانیم به یک گراف گذار تعمیم یافته (gtg) تبدیل کنیم.
- هر یالی که در nfa با نماد a نشان داده شده است را می‌توانیم در gtg با عبارت منظم a نشان دهیم.
- هر یالی که در nfa با نماد $a, b, \cdots$ نشان داده شده است را می‌توانیم در gtg با عبارت منظم $a + b + \cdots$ نشان دهیم.
- بنابراین برای هر زبان منظمی یک گراف تعمیم یافته وجود دارد.

- یک گراف گذار تعمیم‌یافتهٔ کامل گرافی است که همهٔ یال‌های آن حاضر باشند. بنابراین یال‌هایی که وجود ندارند را با عبارت منظم $\emptyset$ نشان می‌دهیم.
- بنابراین یک gtg کامل با $n$ رأس تعداد $n^۲$ یال دارد.

- گراف گذار تعمیم یافتهٔ سمت چپ با معادل کامل آن در سمت راست نشان داده شده است.

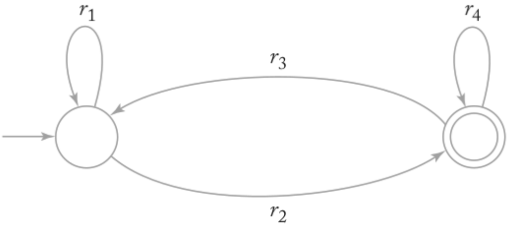


- گراف گذار تعمیم یافتهٔ کامل زیر دو حالت دارد. عبارت منظمی را که این ماشین می‌پذیرد، تعیین کنید.



# عبارت‌ها و زبان‌های منظم

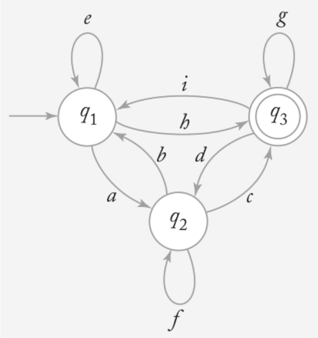
- گراف گذار تعمیم یافتهٔ کامل زیر دو حالت دارد.
- عبارت منظمی را که این ماشین می‌پذیرد، برابر است با : $r = r_1^* r_2 (r_4 + r_3 r_1^* r_2)^*$



# عبارت‌ها و زبان‌های منظم

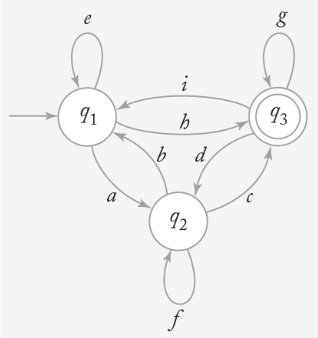
- وقتی یک gtg کامل بیشتر از دو حالت دارد، می‌توانیم در هر مرحله یک حالت آن را حذف کنیم تا در نهایت به یک gtg کامل با دو حالت برسیم.

- گراف گذار تعمیم‌یافته کامل زیر را در نظر بگیرید. چگونه می‌توانیم حالت $q_2$ را حذف کنیم؟



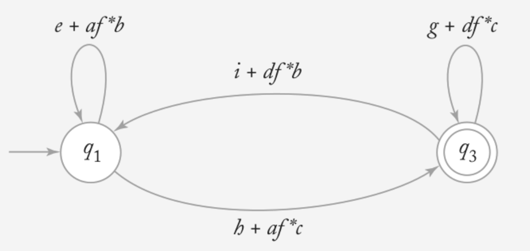
# عبارت‌ها و زبان‌های منظم

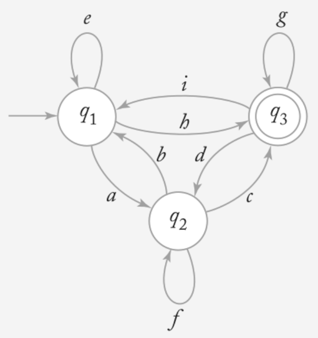
- گراف گذار تعمیم‌یافته کامل زیر را در نظر بگیرید. چگونه می‌توانیم حالت $q_2$ را حذف کنیم؟
- باید همهٔ مسیرهایی که برای گذار از $q_2$ عبور می‌کنند محاسبه کنیم تا بتوانیم $q_2$ را حذف کنیم.



# عبارت‌ها و زبان‌های منظم

- گراف گذار تعمیم‌یافته کامل سمت چپ، کاهش یافتهٔ gtg کامل سمت راست است.



# عبارت‌ها و زبان‌های منظم

- پس به ازای هر gtg کامل می‌توانیم یک حالت در هر گام حذف کنیم تا به یک gtg کامل با دو حالت برسیم و عبارت منظم مربوط به آن گراف را محاسبه کنیم.

روند تبدیل ماشین متناهی غیرقطعی به عبارت منظم

۱. یک nfa را که از حالت‌های $q_۰, q_۱, \cdots, q_n$ تشکیل شده است و حالت پایانی آن متفاوت از حالت آغازی آن است، در نظر بگیرید.

۲. این nfa را به یک gtg کامل تبدیل کنید. فرض کنید برچسب یالی که از حالت $q_i$ به $q_j$ می‌رود برابر است با $r_{ij}$.

۳. اگر این gtg کامل فقط دو حالت داشته باشد و حالت $q_i$ حالت آغازی و حالت $q_j$ حالت پایانی آن باشد، آنگاه عبارت منظم آن برابر است با: $r = r_{ii}^* r_{ij} (r_{jj} + r_{ji} r_{ii}^* r_{ij})^*$

روند تبدیل ماشین متناهی غیرقطعی به عبارت منظم

۴. اگر gtg کامل سه حالت داشته باشد، حالت اولیه $q_i$، حالت پایانی $q_j$ و حالت سوم $q_k$ باشد، یال‌هایی با برچسب‌های $r_{pq} + r_{pk}r_{kk}^*r_{kq}$ به ازای $p = i, j$ و $q = i, j$ بسازید و حالت $q_k$ و یال‌های قبلی را حذف کنید.

۵. اگر gtg کامل چهار حالت یا بیشتر دارد، حالت $q_k$ را برای حذف کردن در نظر بگیرید. قوانین مرحله ۴ را برای هر زوج از حالت‌های $(q_i, q_j), i \neq k, j \neq k$ در نظر بگیرید. در هر مرحله برای ساده‌سازی عبارات می‌توانید قوانین زیر را به کار ببرید: $r + \emptyset = r$، $r\emptyset = \emptyset$ و $\emptyset^* = \lambda$ .

۶. گام‌های ۳ تا ۵ را تکرار کنید تا عبارت منظم متناظر با گراف گذار نظر به دست بیاید.

# عبارت‌ها و زبان‌های منظم

- عبارت منظمی برای این زبان بیابید:
$L = \{w \in \{a, b\}^*$ : تعداد نمادهای a زوج و تعداد نمادهای b فرد است $\}$

# عبارت‌ها و زبان‌های منظم

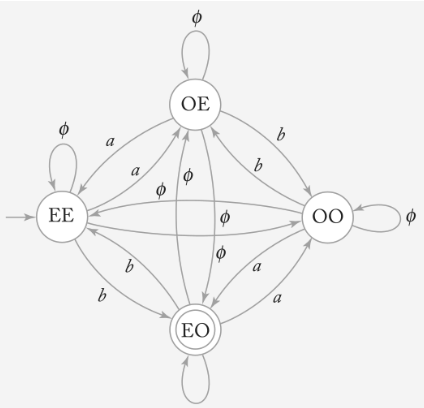
- عبارت منظمی برای این زبان بیابید:
$L = \{w \in \{a, b\}^* :$ تعداد نمادهای a زوج و تعداد نمادهای b فرد است $\}$

- ابتدا یک ماشین غیرقطعی با چهار حالت می‌سازیم که طوری که حالت EE حالتی است که در آن تعدادی زوج از a و b خوانده شده است، حالت OE حالتی است که در آن تعدادی فرد از a و تعدادی زوج از b خوانده شده، در حالت EO تعدادی زوج از a و تعدادی فرد از b خوانده شده، و در حالت OO تعدادی فرد از a و b خوانده شده است.

# عبارت‌ها و زبان‌های منظم

- عبارت منظمی برای این زبان بیابید:
$L = \{w \in \{a, b\}^*$ : تعداد نمادهای a زوج و تعداد نمادهای b فرد است $\}$

- گراف گذار کامل زیر را طراحی می‌کنیم.

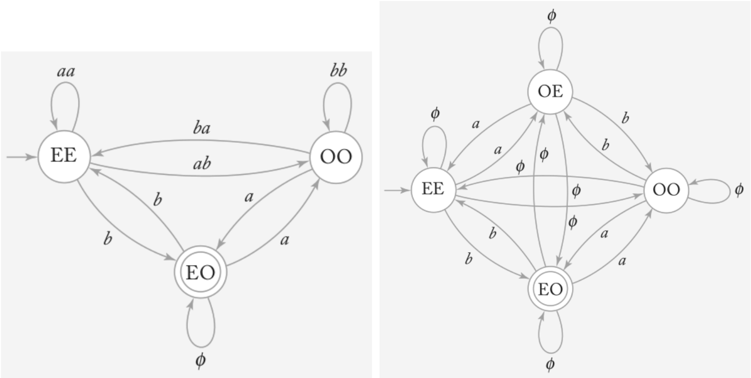


# عبارت‌ها و زبان‌های منظم

- عبارت منظمی برای این زبان بیابید:

$L = \{w \in \{a,b\}^*$ : تعداد نمادهای a زوج و تعداد نمادهای b فرد است $\}$

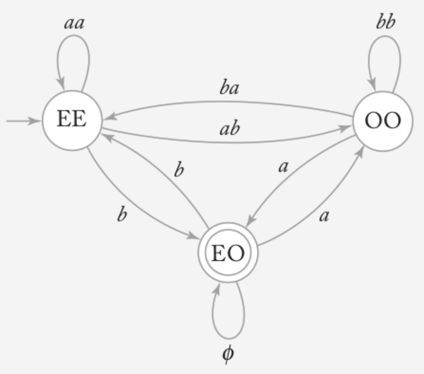
- یکی از حالات gtg کامل را حذف می‌کنیم.

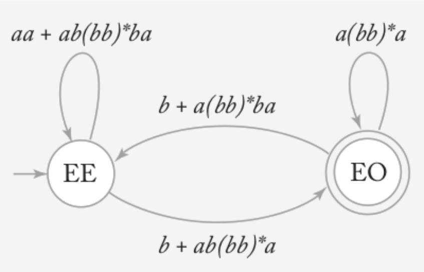


# عبارت‌ها و زبان‌های منظم

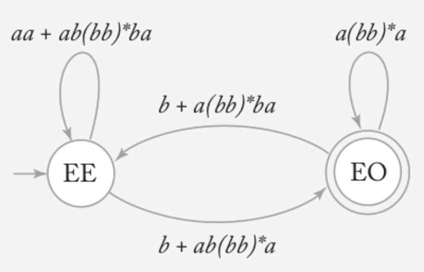
- عبارت منظمی برای این زبان بیابید:
$\mathrm{L} = \{\mathrm{w} \in \{\mathrm{a}, \mathrm{b}\}^*$ : تعداد نمادهای a زوج و تعداد نمادهای b فرد است $\}$

- یکی دیگر از حالات gtg کامل را حذف می‌کنیم.



- عبارت منظمی برای این زبان بیابید:
$L = \{w \in \{a, b\}^*$ : تعداد نمادهای a زوج و تعداد نمادهای b فرد است $\}$



# عبارت‌ها و زبان‌های منظم

- استفاده از روش تبدیل nfa به gtg کامل به طوری دستی بسیار خسته کننده است و عبارتی که در پایان به دست می‌آید بسیار پیچیده است و در عمل نمی‌توان از آن استفاده کرد.
- تنها دلیل ارائه این روش، ارائه روشی رسمی برای اثبات قضایا است.

# عبارت‌ها و زبان‌های منظم

- قضیه: فرض کنید $L$ یک زبان منظم باشد. یک عبارت منظم $r$ وجود دارد به طوری که $L = L(r)$

- اثبات: اگر $L$ یک زبان منظم باشد، یک nfa برای آن وجود دارد. با استفاده از روند تبدیل ماشین متناهی غیرقطعی به عبارت منظم، یک عبارت منظم برای آن می‌سازیم که همان عبارت منظم مورد نظر است.

# گرامرهای منظم

- زبان‌های منظم را علاوه بر نمایش با استفاده از مجموعه‌ها و عبارات منظم، می‌توان با گرامرهای منظم نیز نمایش داد.
- یک گرامر $G = (V, T, S, P)$ راست‌خطی [1] گفته می‌شود، اگر همهٔ قوانین تولید آن به صورت $A \to xB$ یا $A \to x$ باشد، به طوری که $A, B \in V$ و $x \in T^*$
- یک گرامر $G = (V, T, S, P)$ چپ‌خطی [2] گفته می‌شود، اگر همهٔ قوانین تولید آن به صورت $A \to Bx$ یا $A \to x$ باشد.
- یک گرامر منظم [3] گرامری است که راست‌خطی و یا چپ‌خطی باشد.

---
[1] right-linear
[2] left-linear
[3] regular grammar

# گرامرهای منظم

- گرامر $G_1 = (\{S\}, \{a, b\}, S, P_1)$ با قوانین $P_1$ به صورت $S \rightarrow abS|a$ یک گرامر راست‌خطی است.
- عبارت منظمی برای زبان متناظر با این گرامر پیدا کنید.

# گرامرهای منظم

- گرامر $G_1 = (\{S\}, \{a, b\}, S, P_1)$ با قوانین $P_1$ به صورت $S \rightarrow abS|a$ یک گرامر راست‌خطی است.
- عبارت منظمی برای زبان متناظر با این گرامر پیدا کنید.
- $r = (ab)^*a$

# گرامرهای منظم

- گرامر $G_2 = (\{S, S_1, S_2\}, \{a, b\}, S, P_2)$ با قوانین $P_2$ به صورت $S \to S_1ab$، $S_1 \to S_1ab|S_2$، $S_2 \to a$ یک گرامر چپ‌خطی است.
- عبارت منظمی برای زبان متناظر با این گرامر پیدا کنید.

# گرامرهای منظم

- گرامر $G_2 = (\{S, S_1, S_2\}, \{a, b\}, S, P_2)$ با قوانین $P_2$ به صورت $S \rightarrow S_1ab$، $S_1 \rightarrow S_1ab|S_2$، $S_2 \rightarrow a$ یک گرامر چپ‌خطی است.
- عبارت منظمی برای زبان متناظر با این گرامر پیدا کنید.
- $r = aab(ab)^*$

# گرامرهای منظم

- گرامر $G = (\{S, A, B\}, \{a, b\}, S, P)$ با قوانین $S \to A$، $A \to aB|\lambda$، و $B \to Aa$ یک گرامر منظم نیست.
- گرچه برخی از قوانین راست‌خطی و برخی چپ‌خطی هستند، ولی گرامر نه چپ‌خطی است و نه راست‌خطی.
- این گرامر نمونه‌ای از یک گرامر خطی [1] است.
- گرامر خطی حداکثر یک متغیر در سمت راست هر قانون دارد و مکان آن متغیر بدون اهمیت است.
- هر گرامر منظم یک گرامر خطی نیز هست، ولی هر گرامر خطی منظم نیست.
- نشان می‌دهیم که گرامرهای منظم روش دیگری برای بیان زبان‌های منظم هستند.

[1] linear grammar

# گرامرهای منظم

- اکنون نشان می‌دهیم که زبان‌های تولید شده توسط گرامرهای منظم، منظم هستند.
- ابتدا نشان می‌دهیم یک گرامر راست‌خطی، زبانی منظم تولید می‌کند.
- برای این کار، یک ان‌اف‌ای طراحی می‌کنیم که اشتقاق‌های یک گرامر راست‌خطی را شبیه‌سازی کند.
- در یک گرامر راست‌خطی، اشتقاق‌ها بدین صورت هستند : $ab \cdots cD \Rightarrow ab \cdots cdE$ که توسط قانون $D \to dE$ به وجود آمده است.
- ماشین غیرقطعی می‌تواند این گام را شبیه‌سازی کند، به طوری که در حالت $D$ با خواندن نماد $d$ به حالت $E$ می‌رود.
- پس به طور کلی، متغیر گرامر را معادل یک حالت در نظر می‌گیریم و نمادهای پایانی (پیشوند صورت جمله‌ای) را نماد خوانده شده از ورودی بر روی یال در نظر می‌گیریم.

# گرامرهای منظم

- قضیه: فرض کنید $G = (V, T, S, P)$ یک گرامر راست‌خطی باشد. آنگاه $L(G)$ یک زبان منظم است.
- اثبات: فرض می‌کنیم $V = \{V_۰, V_۱, \cdots\}$ به طوری که $S = V_۰$ باشد و قوانین تولید به صورت $V_۰ \rightarrow v_۱ V_i, V_i \rightarrow v_۲ V_j, \cdots V_n \rightarrow v_l, \cdots$ داشته باشیم.
- اگر $w$ یک جمله در زبان $L(G)$ باشد، در یک اشتقاق خواهیم داشت:

$$V_۰ \Rightarrow v_۱ V_i \Rightarrow v_۱ v_۲ V_j \overset{*}{\Rightarrow} v_۱ v_۲ \cdots v_k V_n \Rightarrow v_۱ v_۲ \cdots v_k v_l = w$$

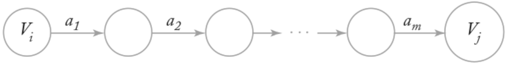
- ماشینی طراحی می‌کنیم که در ابتدا در حالت $V_۰$ قرار دارد. به ازای هر متغیر $V_i$ یک حالت غیرپایانی با نام $V_i$ در ماشین در نظر می‌گیریم.

# گرامرهای منظم

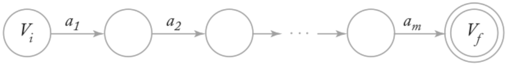
- به ازای هر قانون تولید $V_i \to a_1a_2\cdots a_mV_j$ ماشین گذارهایی برای اتصال $V_i$ به $V_j$ دارد. تابع گذار تعمیم‌یافته $\delta^*$ به ازای این قانون اینگونه تعریف می‌شود: $\delta^*(V_i, a_1a_2\cdots a_m) = V_j$

- برای هر قانون تولید $V_i \to a_1a_2\cdots a_m$ گذارهای ماشین به صورتی است که $\delta^*(V_i, a_1a_2\cdots a_m) = V_f$ به طوری که $V_f$ یک حالت پایانی است.

- حالت‌های میانی را نیز برای شبیه‌سازی تابع گذار تعمیم‌یافته به صورت زیر می‌سازیم.




$V_i \to a_1a_2\ldots a_mV_j$




$V_i \to a_1a_2\ldots a_m$

- پس فرض کردیم $w \in L(G)$ و یک ان‌اف‌ای معادل گرامر $G$ طراحی کردیم به طوری که مسیری از $V_0$ به $V_i$ با برچسب $v_1$ ، مسیری از $V_i$ به $V_j$ با برچسب $v_2$ و الی آخر وجود دارد. پس $V_f \in \delta^*(V_0, w)$ و بنابراین $w$ توسط ماشین طراحی شده پذیرفته می‌شود. پس هر رشتهٔ تولید شده توسط این گرامر توسط یک ماشین متناهی غیرقطعی پذیرفته می‌شود.

- از طرف دیگر، فرض کنید رشتهٔ $w$ توسط ماشین طراحی شده پذیرفته می‌شود. برای پذیرفتن $w$ باید مسیری از حالت‌ها به صورت $V_0, V_i, \cdots, V_f$ وجود داشته باشد که برچسب آن به صورت $v_1, v_2, \cdots$ باشد. بنابراین $w$ باید بدین صورت باشد: $w = v_1 v_2 \cdots v_k v_l$ و بنابراین اشتقاق $V_0 \Rightarrow v_1 V_i \Rightarrow v_1 v_2 V_j \overset{*}{\Rightarrow} v_1 v_2 \cdots v_k V_k \Rightarrow v_1 v_2 \cdots v_k v_l$ امکان پذیر است. بنابراین $w$ در $L(G)$ است. پس هر رشتهٔ پذیرفته شده توسط این ماشین غیرقطعی، توسط گرامر $G$ نیز تولید می‌شود.

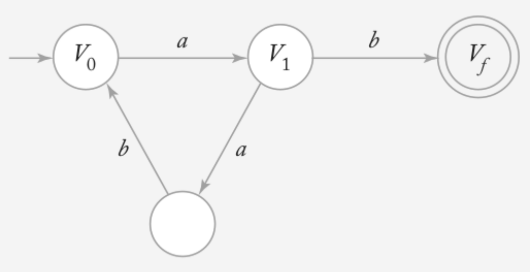
- پس برای $L(G)$ یک ان‌اف‌ای معادل آن طراحی کردیم و بنابراین $L(G)$ منظم است.

# گرامرهای منظم

- یک ماشین متناهی طراحی کنید که زبان تولید شده توسط گرامر G با قوانین $V_0 \to aV_1 \quad , \quad V_1 \to abV_0|b$ را بپذیرد.

# گرامرهای منظم

- ماشین متناهی زیر زبان تولید شده توسط گرامر G با قوانین $V_۰ \rightarrow aV_۱ \quad , \quad V_۱ \rightarrow abV_۰|b$ را می‌پذیرد.
- این ماشین زبان منظم $L((aab)^*ab)$ را می‌پذیرد.



# گرامرهای منظم

- برای اینکه نشان دهیم هر زبان منظم می‌تواند توسط یک گرامر راست‌خطی تولید شود، از ماشین متناهی زبان منظم شروع می‌کنیم و نشان می‌دهیم که گذار از حالت‌های ماشین متناظر با اشتقاق توسط قوانین تولید یک گرامر راست‌خطی است.
- حالت‌های ماشین همان متغیرها در قوانین تولید هستند و برچسب روی یال‌ها همان پایانه‌ها در قوانین.
- می‌توانیم همانند قبل اثبات رسمی نیز ارائه کنیم.

# گرامرهای منظم

- یک گرامر راست‌خطی برای زبان منظم $L(aab^*a)$ بسازید.

# گرامرهای منظم

- یک گرامر راست‌خطی برای زبان منظم $L(aab^*a)$ بسازید.

| | |
|---|---|
| $\delta(q_0, a) = \{q_1\}$ | $q_0 \longrightarrow aq_1$ |
| $\delta(q_1, a) = \{q_2\}$ | $q_1 \longrightarrow aq_2$ |
| $\delta(q_2, b) = \{q_2\}$ | $q_2 \longrightarrow bq_2$ |
| $\delta(q_2, a) = \{q_f\}$ | $q_2 \longrightarrow aq_f$ |
| $q_f \in F$ | $q_f \longrightarrow \lambda$ |

# گرامرهای منظم

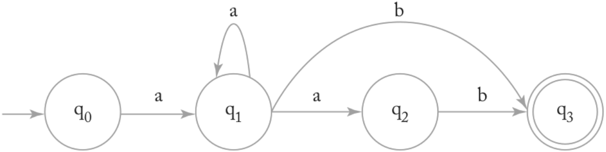
- زبان $L$ منظم است اگر و فقط اگر گرامر منظم $G$ برای آن وجود داشته باشد به طوری که $L = L(G)$ .
- در این بخش الگوریتم‌هایی برای تبدیل عبارت‌های منظم و گرامرهای منظم به ماشین‌های متناهی و بالعکس ارائه کردیم.
- پس زبان‌های منظم را می‌توانیم توسط ماشین‌های متناهی قطعی و غیرقطعی، عبارت‌های منظم و گرامرهای منظم وصف کنیم.

# مثال

- یک ماشین متناهی غیرقطعی برای زبان $L(aa^*(ab+b))$ طراحی کنید.

# مثال

- یک ماشین متناهی غیرقطعی برای زبان $L(aa^*(ab+b))$ طراحی کنید.



# مثال

- یک عبارت منظم برای زبان $L = \{a^n b^m : n \geq ۳, m \bmod ۲ = ۱\}$ پیدا کنید.

# مثال

- یک عبارت منظم برای زبان $L = \{a^n b^m : n \geq ۳, m \bmod ۲ = ۱\}$ پیدا کنید.
- $aaaa^*(bb)^*b$

# مثال

- یک عبارت منظم برای زبان $L = \{a^n b^m : n \geq ۳, m \leq ۴\}$ پیدا کنید.

# مثال

- یک عبارت منظم برای زبان $L = \{a^n b^m : n \geq ۳, m \leq ۴\}$ پیدا کنید.
- $aaaa^*(\lambda + b + bb + bbb + bbbb)$

# مثال

- یک عبارت منظم برای زبان $L = \{vwv : v, w \in \{a, b\}^*, |v| = ۲\}$ پیدا کنید.

# مثال

- یک عبارت منظم برای زبان $L = \{vwv : v, w \in \{a, b\}^*, |v| = ۲\}$ پیدا کنید.
- $aa(a+b)^*aa + ab(a+b)^*ab + ba(a+b)^*ba + bb(a+b)^*bb$

# مثال

- یک عبارت منظم برای زبان $L = \{uwv : u, v, w \in \{a, b\}^*, |u| = |v| = ۲\}$ پیدا کنید.

# مثال

- یک عبارت منظم برای زبان $L = \{uwv : u, v, w \in \{a, b\}^*, |u| = |v| = ۲\}$ پیدا کنید.
- $(aa + ab + ba + bb)(a + b)^*(aa + ab + ba + bb)$
- $(a + b)(a + b)(a + b)^*(a + b)(a + b)$

# مثال

- به ازای $\Sigma = \{a, b, c\}$، یک عبارت منظم برای زبانی بیابید که جملات آن فقط دو $a$ دارند.

# مثال

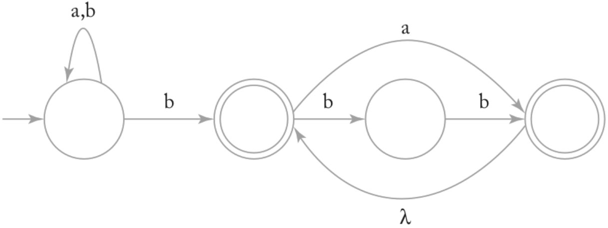
- به ازای $\Sigma = \{a, b, c\}$، یک عبارت منظم برای زبانی بیابید که جملات آن فقط دو $a$ دارند.
- $(b + c)^*a(b + c)^*a(b + c)^*$

# مثال

- یک ماشین متناهی غیرقطعی برای زبان $L((a+b)^*b(a+bb)^*)$ طراحی کنید.

# مثال

- یک ماشین متناهی غیرقطعی برای زبان $L((a+b)^*b(a+bb)^*)$ طراحی کنید.

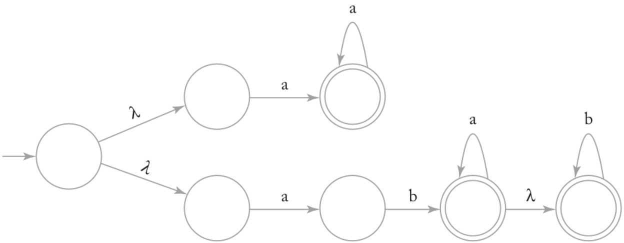


# مثال

- یک ماشین متناهی برای زبان $L(aa^* + aba^*b^*)$ طراحی کنید.

# مثال

- یک ماشین متناهی برای زبان $L(aa^* + aba^*b^*)$ طراحی کنید.

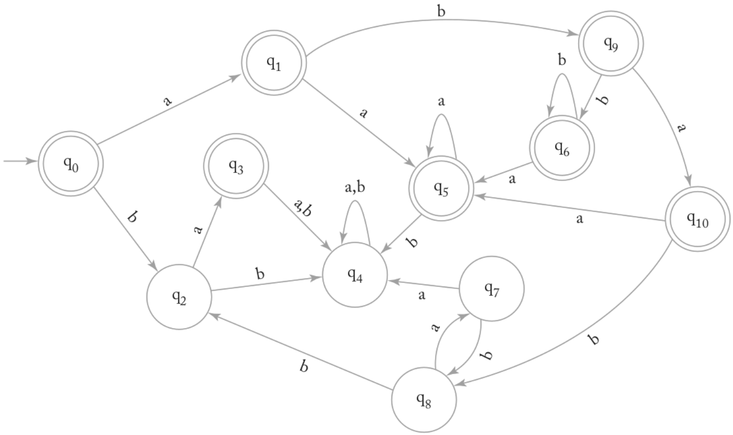


# مثال

- یک ماشین متناهی قطعی برای زبان $L = L(ab^*a^*) \cup L((ab)^*ba)$

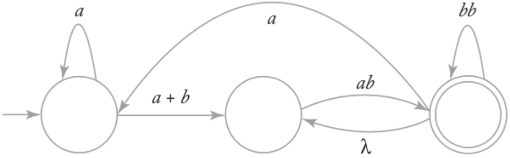
# مثال

- یک ماشین متناهی قطعی برای زبان $L = L(ab^*a^*) \cup L((ab)^*ba)$



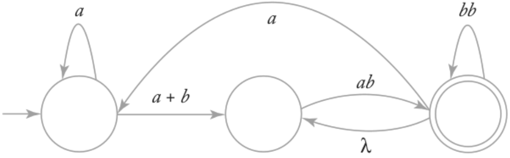
# مثال

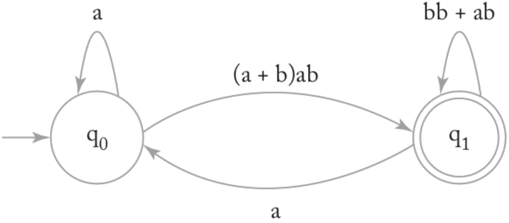
- گراف گذار تعمیم یافته زیر را به گرافی با دو حالت تبدیل کنید و عبارت منظم معادل آن را بنویسید.



## مثال

- گراف گذار تعمیم یافته زیر را به گرافی با دو حالت تبدیل کنید و عبارت منظم معادل آن را بنویسید.
- $r = a^*(a + b)ab(bb + ab + aa^*(a + b)ab)^*$



# مثال

- به ازای $\Sigma = \{a, b\}$، عبارت منظمی برای زبانی بیابید طول جملات آن زوج باشد.

# مثال

- به ازای $\Sigma = \{a, b\}$، عبارت منظمی برای زبانی بیابید طول جملات آن زوج باشد.
- $(aa + bb + ab + ba)^*$
- $((a + b)(a + b))^*$

# مثال

- به ازای $\Sigma = \{a, b\}$، عبارت منظمی برای زبانی بیابید طول جملات آن بزرگتر یا مساوی ۳ باشد.

# مثال

- به ازای $\Sigma = \{a, b\}$، عبارت منظمی برای زبانی بیابید طول جملات آن بزرگتر یا مساوی ۳ باشد.
- $(a + b)(a + b)(a + b)(a + b)^*$

# مثال

- به ازای $\Sigma = \{a, b\}$، عبارت منظمی برای زبانی بیابید تعداد نمادهای $a$ در آن فرد باشد.

# مثال

- به ازای $\Sigma = \{a, b\}$، عبارت منظمی برای زبانی بیابید تعداد نمادهای a در آن فرد باشد.
- $b^*ab^*(ab^*ab^*)^*$
- $b^*(ab^*ab^*)^*ab^*$
- $b^*a(b + ab^*a)^*$
- $(b + ab^*a)^*ab^*$

# مثال

- به ازای $\Sigma = \{۰, ۱\}$، عبارت منظمی برای همهٔ اعداد دودویی مساوی یا بیشتر از ۳۲ (۱۰۰۰۰۰) بیابید.

# مثال

- به ازای $\Sigma = \{۰, ۱\}$، عبارت منظمی برای همهٔ اعداد دودویی مساوی یا بیشتر از ۳۲ (۱۰۰۰۰۰) بیابید.
- $۱(۰ + ۱)^۵(۰ + ۱)^*$

# مثال

- به ازای $\Sigma = \{a, b\}$، عبارت منظمی برای زبانی بیابید که ابتدا و انتهای همهٔ جملات آن یکسان باشد.

# مثال

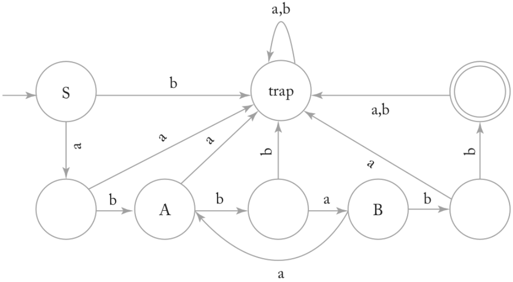
- به ازای $\Sigma = \{a, b\}$، عبارت منظمی برای زبانی بیابید که ابتدا و انتهای همهٔ جملات آن یکسان باشد.
- $a(a + b)^*a + b(a + b)^*b + a + b$

# مثال

- یک ماشین متناهی قطعی برای گرامری با قوانین زیر بیابید. عبارت منظم و گرامر چپ خطی متناظر آن چیست؟ $S \rightarrow abA \quad , \quad A \rightarrow baB \quad , \quad B \rightarrow aA|bb$

## مثال

- یک ماشین متناهی قطعی برای گرامری با قوانین زیر بیابید. عبارت منظم و گرامر چپ خطی متناظر آن چیست؟ $S \rightarrow abA \quad , \quad A \rightarrow baB \quad , \quad B \rightarrow aA|bb$
- $L(G) = L(abba(aba)^*bb)$
- $S \rightarrow Abb \quad , \quad A \rightarrow Aaba|B \quad , \quad B \rightarrow abba$



# مثال

- یک گرامر راست‌خطی و یک گرامر چپ خطی برای زبان $L((aaab^*ab)^*)$ بیابید.

# مثال

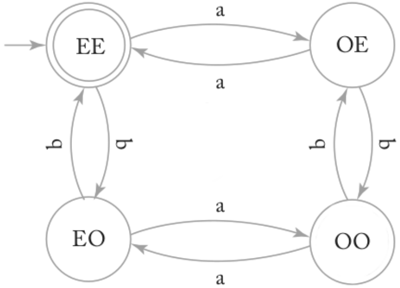
- یک گرامر راست‌خطی و یک گرامر چپ خطی برای زبان $L((aaab^*ab)^*)$ بیابید.
- $S \rightarrow aaaA|\lambda \quad , \quad A \rightarrow bA|B \quad , \quad B \rightarrow ab|abS$
- $S \rightarrow Aab|\lambda \quad , \quad A \rightarrow Ab|B \quad , \quad B \rightarrow Saaa$

## مثال

- یک گرامر منظم برای زبان $L = \{w \in \{a, b\}^* : n_a(w) \bmod ۲ = ۰, n_b(w) \bmod ۲ = ۰\}$ بیابید.

# مثال

- یک گرامر منظم برای زبان $L = \{w \in \{a,b\}^* : n_a(w) \bmod ۲ = ۰, n_b(w) \bmod ۲ = ۰\}$ بیابید.
- از ماشین متناهی این زبان استفاده می‌کنیم.
- $EE \to aOE|bEO|\lambda \ , \ OE \to aEE|bOO \ , \ EO \to aOO|bEE \ , \ OO \to aEO|bOE$



# ویژگی‌های بستاری زبان‌های منظم

- اگر مجموعهٔ همهٔ زبان‌های منظم را در نظر بگیریم، آنگاه می‌خواهیم بررسی کنیم آیا این مجموعه در برابر عملگرهای مختلف مانند الحاق و اجتماع و اشتراک بسته است یا خیر.
- اگر یک مجموعه بر روی یک عملگر بسته باشد، نتیجه اعمال آن عملگر بر روی اعضای آن مجموعه عضوی از همان مجموعه است.
- پس اگر مجموعه‌ای از زبان‌های منظم داشته باشیم و زبان‌های منظم در برابر یک عملگر بسته باشند، آنگاه اعمال آن عملگر بر روی آن زبان‌ها زبانی منظم ایجاد می‌کند.
- در اینجا ویژگی‌های بستاری [1] زبان‌های منظم را بررسی می‌کنیم.

[1] closure properties

# ویژگی‌های بستاری زبان‌های منظم

- اگر دو زبان $L_1$ و $L_2$ منظم باشند، زبان‌های $L_1 \cup L_2$، $L_1 \cap L_2$، $L_1L_2$، $\overline{L_1}$، $L_1^*$ نیز منظم هستند.
- می‌گوییم خانوادهٔ زبان‌های منظم بر روی عملگرهای اجتماع، اشتراک، الحاق، متمم، و بستار-ستاره بسته است.

# ویژگی‌های بستاری زبان‌های منظم

- اگر دو زبان $L_1$ و $L_2$ منظم باشند، زبان‌های $L_1 \cup L_2$، $L_1L_2$، و $L_1^*$ نیز منظم هستند.
- اثبات: اگر دو زبان $L_1$ و $L_2$ منظم باشند، دو عبارت منظم $r_1$ و $r_2$ برای آنها وجود دارد که آن دو زبان را وصف می‌کند.
- پیشتر نشان دادیم اگر $r_1$ و $r_2$ منظم باشند، $r_1 + r_2$، $r_1r_2$، $r_1^*$ نیز منظم هستند که زبان‌های $L_1 \cup L_2$، $L_1L_2$، $L_1^*$ را وصف می‌کنند.

# ویژگی‌های بستاری زبان‌های منظم

- اگر زبان $L_1$ منظم باشند، زبان $\overline{L_1}$ نیز منظم است.
- اثبات: ماشین متناهی قطعی متناظر با $L_1$ را در نظر می‌گیریم. متمم زبان در یک ماشین متناهی قطعی، ماشینی است که در آن جای حالت‌های پایانی و غیر پایانی عوض شده باشد. پس ماشینی برای متمم آن وجود دارد و بنابراین متمم یک زبان منظم نیز منظم است.

# ویژگی‌های بستاری زبان‌های منظم

- اگر دو زبان $L_1$ و $L_2$ منظم باشند، زبان $L_1 \cap L_2$، نیز منظم است.
- اثبات: می‌توانیم از قوانین دمورگان استفاده کنیم. $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$. از آنجایی که زبان‌های منظم بر روی متمم و اجتماع بسته هستند، بر روی اشتراک نیز بسته‌اند.

# ویژگی‌های بستاری زبان‌های منظم

- برای اثبات بسته بودن زبان‌های منظم بر روی اشتراک همچنین می‌توانیم ماشینی بسازیم که هر یک از حالات آن متناظر با یک حالت از ماشین اول و یک حالت از ماشین دوم است.
- به عبارت دیگر فرض کنید $L_1 = L(M_1)$ و $L_2 = L(M_2)$ به طوری که $M_1 = (Q, \Sigma, \delta_1, q_0, F_1)$ و $M_2 = (P, \Sigma, \delta_2, p_0, F_2)$. ماشینی به صورت $\widehat{M} = (\widehat{Q}, \Sigma, \widehat{\delta}, (p_0, q_0), \widehat{F})$ که حالت‌های آن به صورت $\widehat{Q} = Q \times P$ شامل همهٔ حالت‌های $(q_i, p_j)$ باشد به طوری که $\widehat{\delta}((q_i, p_j), a) = (q_k, q_l)$ وقتی $\delta_1(q_i, a) = q_k$ و $\delta_2(p_j, a) = q_l$ . حالت‌های پایانی $\widehat{F}$ حالت‌های $(q_i, p_i)$ هستند به طوری که $q_i \in F_1$ و $p_j \in F_2$ . سپس نشان می‌دهیم هر رشته $w \in L_1 \cap L_2$ توسط ماشین $\widehat{M}$ پذیرفته می‌شود و بنابراین اشتراک دو زبان منظم است.

# ویژگی‌های بستاری زبان‌های منظم

- نشان دهید اگر $L_1$ و $L_2$ دو زبان منظم باشند، آنگاه $L_1 - L_2$ نیز منظم است.

# ویژگی‌های بستاری زبان‌های منظم

- نشان دهید اگر $L_1$ و $L_2$ دو زبان منظم باشند، آنگاه $L_1 - L_2$ نیز منظم است.
- می‌نویسیم $L_1 - L_2 = L_1 \cap \overline{L_2}$
- از آنجایی که متمم یک زبان منظم و اشتراک دو زبان منظم، یک زبان منظم است، پس تفاضل دو زبان منظم نیز یک زبان منظم است.

# ویژگی‌های بستاری زبان‌های منظم

- خانوادهٔ زبان‌های منظم بر روی عملگر معکوس بسته است.

- برای اثبات می‌توانیم یک ان‌اف‌ای برای زبان L بسازیم. سپس حالت پایانی را تبدیل به حالت آغازی و حالت آغازی را تبدیل به حالت پایانی می‌کنیم و جهت یال‌های گذار را برعکس (از مقصد به مبدأ) می‌کنیم. این ماشین معکوس زبان L را می‌پذیرد.

- اگر ان‌اف‌ای بیشتر از یک حالت پایانی داشت، همهٔ حالت‌های پایانی را به یک حالت پایانی با گذار توسط نماد تهی متصل می‌کنیم.

# محدودیت زبان‌های منظم

- زبان‌هایی وجود دارند که منظم نیستند و نمی‌توان برای آنها ماشین متناهی پیدا کرد.
- در اینجا برای اینکه نشان دهیم یک زبان منظم نیست از لم تزریق [1] (لم پمپاژ) استفاده می‌کنیم.

[1] pumping lemma

# محدودیت زبان‌های منظم

- زبان‌های منظم توسط ماشین‌های متناهی شناسایی می‌شوند.
- ماشین‌های متناهی حافظهٔ محدود دارند، چرا که فقط می‌توانند به یاد بیاورند در چه حالتی هستند و حافظهٔ جانبی ندارند.
- به طور شهودی می‌توانیم حدس بزنیم که زبان‌هایی که برای شناسایی به ماشینی با حافظهٔ بیشتر نیاز دارند نمی‌توانند منظم باشند.
- این حدس را بیشتر بررسی می‌کنیم.

# محدودیت زبان‌های منظم

- اصل لانه کبوتری : اگر $n$ شیء (کبوتر) را در $m$ جعبه (لانه کبوتر) قرار دهیم، و $n > m$ باشد، آنگاه حداقل یکی از جعبه‌ها باید بیشتر از یک شیء را شامل شود.

# محدودیت زبان‌های منظم

- آیا زبان $L = \{a^n b^n : n \geq ۰\}$ منظم است؟

# محدودیت زبان‌های منظم

- آیا زبان $L = \{a^n b^n : n \geq ۰\}$ منظم است؟
- با برهان خلف و با استفاده از اصل لانه کبوتری نشان می‌دهیم که این زبان منظم نیست.
- فرض کنید $L$ یک زبان منظم باشد. آنگاه باید یک ماشین متناهی قطعی به صورت $M = (Q, \{a, b\}, \delta, q_۰, F)$ برای آن وجود داشته باشد.
- حال $\delta^*(q_۰, a^i)$ را به ازای $i = ۱, ۲, ۳, \cdots$ در نظر بگیرید. از آنجایی که تعداد نامحدودی $a$ می‌تواند وجود داشته باشد ولی تعداد حالت‌های ماشین محدود است، طبق اصل لانه کبوتری حالتی به نام $q$ وجود دارد به طوری که $\delta^*(q_۰, a^n) = q$ و $\delta^*(q_۰, a^m) = q$ به ازای $n \neq m$ .
- از آنجایی که ماشین رشتهٔ $a^n b^n$ را می‌پذیرد، باید داشته باشیم $\delta^*(q, b^n) = q_f \in F$
- پس داریم: $\delta^*(q_۰, a^m b^n) = \delta^*(\delta^*(q_۰, a^m), b^n) = \delta^*(q, b^n) = q_f$
- این نتیجه با فرض اولیه تناقض دارد چراکه گفتیم ماشین فقط رشته‌های $a^n b^n$ را می‌پذیرد، پس $L$ نمی‌تواند منظم باشد.

# محدودیت زبان‌های منظم

- برای پذیرفتن رشتهٔ $a^n b^n$ یک ماشین متناهی باید قادر باشد بین پیشوندهای $a^n$ و $a^m$ تفاوت قائل شود.
- اما از آنجایی که فقط تعداد محدودی حالت داخلی وجود دارد، به ازای برخی $n$ و $m$ ها، ماشین متناهی نمی‌تواند تفاوتی قائل شود، بنابراین این زبان منظم نیست.

# لم تزریق

- برای اثبات اینکه یک زبان منظم نیست، از لم تزریق استفاده می‌کنیم.
- در لم تزریق به طور شهودی از این خاصیت استفاده می‌کنیم: برای یک زبان نامحدود، در یک گراف گذار با n حالت، در هر گشتی در گراف که طول آن بیشتر از n باشد، از برخی از حالت‌ها چندین بار عبور می‌کنیم. بنابراین گراف گذار باید دارای دور باشد.
- زبان‌های محدود منظم هستند چرا که می‌توان یک ماشین غیرقطعی برای پذیرفتن همهٔ جملات آنها طراحی کرد.

# لم تزریق

- قضیه: فرض کنید $L$ یک زبان منظم نامحدود باشد. آنگاه، یک عدد صحیح مثبت $m$ وجود دارد، به طوری که هر رشتهٔ $w \in L$ با طول $|w| \geq m$ می‌تواند به صورت $w = xyz$ به سه قسمت تقسیم شود به طوری که $|xy| \leq m$، $|y| \geq ۱$ و همچنین $w_i = xy^iz \in L$، به ازای همهٔ مقادیر $i = ۰, ۱, ۲, \cdots$.
- به عبارت دیگر هر رشته‌ای در زبان منظم $L$ که به اندازهٔ کافی طولانی باشد، می‌تواند به سه قسمت تقسیم شود به طوری که از تکرار قسمت میانی، یک رشتهٔ دیگر در زبان $L$ به دست می‌آید. می‌گوییم قسمت میانی پمپاژ (تزریق) می‌شود.

# لم تزریق

- اثبات لم : اگر $L$ یک زبان منظم باشد، یک دی‌اف‌ای برای آن وجود دارد که آن را می‌پذیرد. فرض کنید حالت‌های این ماشین $q_0, q_1, \cdots, q_n$ باشد.
- نشان می‌دهیم که اگر $m$ را برابر با $n+1$ بگیریم، می‌توانیم رشتهٔ $xyz$ را با شرایط گفته شده پیدا کنیم.
- حال رشتهٔ $w \in L$ را در نظر بگیرید به طوری که $|w| \geq m = n + 1$ . از آنجایی که زبان $L$ نامحدود است، این کار را می‌توانیم همیشه انجام دهیم.

# لم تزریق

- دنباله‌ای از حالت‌ها را در نظر بگیرید که ماشین برای پذیرفتن $w$ باید از آنها بگذرد : $q_0, q_i, q_j, \cdots, q_f$
- از آنجایی که این دنباله $|w| + 1$ حالت را در بر می‌گیرد و ماشین تنها $n + 1$ حالت دارد، بنابراین حداقل یکی از حالت‌ها باید تکرار شده باشد. پس دنباله باید بدین شکل باشد: $q_0, q_i, q_j, \cdots, q_r, \cdots, q_r, \cdots q_f$
- بنابراین باید زیررشته‌هایی وجود داشته باشند به طوری که $\delta^*(q_0, x) = q_r$، $\delta^*(q_r, y) = q_r$، و $\delta^*(q_r, z) = q_f$ به طوری که $|xy| \leq n + 1 = m$ و $|y| \geq 1$ .
- همچنین داریم $\delta^*(q_0, xz) = q_f$، $\delta^*(q_0, xy^2z) = q_f$، $\delta^*(q_0, xy^3z) = q_f$ و الی‌آخر.

- با استفاده از لم تزریق نشان دهید که زبان $L = \{a^n b^n : n \geq 0\}$ منظم نیست.

## لم تزریق

- با استفاده از لم تزریق نشان دهید که زبان $L = \{a^n b^n : n \geq 0\}$ منظم نیست.
- فرض کنید زبان $L$ منظم باشد. پس لم تزریق باید برای آن برقرار باشد. مقدار $m$ هرچه باشد می‌توانیم به ازای آن $w = a^m b^m$ را در نظر بگیریم که طول آن بزرگتر از $m$ است.
- پس در رشتهٔ $xyz = a^m b^m$ زیررشتهٔ $y$ فقط از $a$ تشکیل شده است. فرض کنید طول رشتهٔ $y$ برابر باشد با $k$.
- پس داریم: $w_i = xy^i z = a^{m-k}(a^k)^i b^m$
- به ازای $i = 0$ به دست می‌آوریم $w_0 = xy^0 z = a^{m-k} b^m$ که این رشته عضو زبان $L$ نیست. پس فرض اولیه نادرست است و $L$ منظم نیست.

## لم تزریق

- نشان دهید زبان $L = \{ww^R : w \in \Sigma^*\}$ منظم نیست.

# لم تزریق

- نشان دهید زبان $L = \{ww^R : w \in \Sigma^*\}$ منظم نیست.
- فرض کنیم که زبان منظم است. پس باید در لم تزریق صدق کند و یک $m$ وجود داشته باشد که به ازای هر جملهٔ $w$ در $L$ شرایط لم تزریق برقرار باشد.
- اگر $m$ هر مقداری داشته باشد، می‌توانیم یکی از جملات زبان را به صورت $w = a^m b^m b^m a^m$ در نظر بگیریم.
- از آنجایی که $|xy| \leq m$ پس $y$ تنها از نمادهای $a$ تشکیل شده است. فرض کنیم طول $y$ برابر با $k$ باشد.
- برای رشتهٔ $w_i = xy^iz = a^{m-k}(a^k)^i b^m b^m a^m$ اگر در نظر بگیریم $i = ۰$ آنگاه رشته‌ای داریم به صورت $a^{m-k} b^m b^m a^k$ که در زبان $L$ نیست.
- پس فرض اولیه نادرست است و $L$ منظم نیست.

## لم تزریق

- فرض کنید $\Sigma = \{a, b\}$. با استفاده از لم تزریق نشان دهید $L = \{w \in \Sigma^* : n_a(w) < n_b(w)\}$ منظم نیست.

# لم تزریق

- فرض کنید $\Sigma = \{a, b\}$. با استفاده از لم تزریق نشان دهید $L = \{w \in \Sigma^* : n_a(w) < n_b(w)\}$ منظم نیست.
- به ازای $m$ داده شده، جملهٔ $w = a^m b^{m+۱}$ را در نظر می‌گیریم.
- در اینصورت $y = a^k$ به ازای $۱ \leq k \leq m$
- بنابراین $w_i = xy^iz = a^{m-k}(a^k)^i b^{m+۱}$
- اکنون به ازای $i = ۲$ داریم $w_۲ = a^{m+k}b^{m+۱}$ که در $L$ نیست و $L$ منظم نیست.

# لم تزریق

- نشان دهید زبان $L = \{a^n : \text{مربع کامل است } n\}$ منظم نیست.

# لم تزریق

- نشان دهید زبان $L = \{a^n : \text{n مربع کامل است}\}$ منظم نیست.
- به ازای $m$ داده شده، جملهٔ $w = a^{m^۲}$ را انتخاب می‌کنیم.
- اگر $w = xyz$ آنگاه $y = a^k$ به ازای $۱ \leq k \leq m$.
- در اینصورت داریم $w_۰ = a^{m^۲-k}$ اما با توجه به اینکه $m^۲ - k > (m - ۱)^۲$ بنابراین $w_۰$ در زبان $L$ نیست و زبان $L$ منظم نیست.

# لم تزریق

- نشان دهید زبان $L = \{a^n b^l : n \neq l\}$ منظم نیست.

# لم تزریق

- نشان دهید زبان $L = \{a^n b^l : n \neq l\}$ منظم نیست.
- فرض می‌کنیم زبان L منظم باشد، پس لم تزریق باید برای آن برقرار باشد.
- به ازای هر m با در نظر گرفتن جملهٔ $a^{m+1}b^m$ نمی‌توانیم اثبات کنیم که زبان منظم نیست. گرچه با در نظر گرفتن $y = a$ و انتخاب $i = 0$ می‌توانیم جملهٔ $a^m b^m$ را تولید کنیم که در زبان L نیست، ولی اگر y به گونه‌ای دیگر (مثلا $y = a^2$) در نظر گرفته شود نمی‌توانیم به مطلوب دست پیدا کنیم.
- بنابراین برای اثبات غیرمنظم بودن این زبان با استفاده از لم تزریق باید جملهٔ دیگری را به ازای m داده شده در نظر بگیریم.

# لم تزریق

- نشان دهید زبان $L = \{a^n b^l : n \neq l\}$ منظم نیست.
- فرض می‌کنیم زبان $L$ منظم باشد، پس لم تزریق باید برای آن برقرار باشد.
- به ازای هر $m$ جملهٔ $w = a^{m!} b^{(m+1)!}$ را در نظر بگیرید.
- از آنجایی که $y = a^k$ است، داریم $w = xyz = a^{m!-k} a^k b^{(m+1)!}$ و $w_i = xy^iz = a^{m!-k} (a^k)^i b^{(m+1)!}$ بنابراین: $w_i = a^{m!+k(i-1)} b^{(m+1)!}$
- حال برای $k$ هر مقداری در نظر گرفته شود، همیشه می‌توانیم در نظر بگیریم: $i = 1 + \frac{mm!}{k}$ که به دست می‌دهد: $w_i = a^{m!+mm!} b^{(m+1)!} = a^{(m+1)!} b^{(m+1)!}$
- پس در هر صورت $w_i$ در زبان $L$ نیست و بنابراین طبق لم تزریق فرض اولیه مبنی بر منظم بودن زبان نادرست بوده است.

## لم تزریق

- لم تزریق یک شرط لازم است و یک شرط کافی نیست، بدین معنی که اگر یک زبان منظم باشد، شرایط گفته شده در لم برقرار است ولی اگر شرایط گفته شده برقرار بود، زبان الزاما منظم نیست.
- بنابراین از لم تزریق برای اثبات منظم نبودن زبان استفاده می‌کنیم و نه اثبات منظم بودن.
- لم تزریق می‌گوید اگر زبانی منظم باشد یک $m$ **وجود دارد** به طوری که به ازای **هر** رشتهٔ $w$ با طول بیشتر از $m$ رشته را **می‌توان** با شرایط گفته شده به سه قسمت $x$ و $y$ و $z$ تقسیم کرد به طوری که اگر قسمت میانی یعنی $y$ به ازای **هر** $i$ تکرار شود (به هر مقداری پمپاژ شود) رشتهٔ به دست آمده در زبان مورد نظر قرار می‌گیرد.
- پس برای اثبات نامنظم بودن زبان از طریق برهان خلف، فرض می‌کنیم زبانی منظم باشد. پس باید طبق لم تزریق یک $m$ **وجود داشته باشد**. **یکی** از رشته‌هایی که طول آن بیشتر از $m$ است را در نظر می‌گیریم. نشان می‌دهیم که **نمی‌توان** آن رشته را به سه قسمت تقسیم کرد، به طوری که از پمپاژ قسمت میانی، رشته‌ای در آن زبان به دست آید. به عبارت دیگر **به هر نحوی** رشته را تقسیم کنیم، یک مقدار $i$ (تعداد تکرار معین از قسمت میانی) **وجود دارد** که به ازای آن لم تزریق **برقرار نباشد**. پس فرض اولیه نادرست بوده و آن زبان نامنظم است.

# مثال

- با استفاده از ویژگی‌های بستاری زبان‌های منظم بر روی عملگرها، نشان دهید زبان $L_1 = \{a^n b^l : n \neq l\}$ منظم نیست.

# مثال

- با استفاده از ویژگی‌های بستاری زبان‌های منظم بر روی عملگرها، نشان دهید زبان $L_۱ = \{a^n b^l : n \neq l\}$ منظم نیست.
- فرض می‌کنیم زبان $L_۱$ یک زبان منظم است، آنگاه زبان $\overline{L_۱} = \Sigma^* - L_۱$ نیز طبق ویژگی‌های بستاری منظم است.
- از طرفی می‌دانیم زبان $L_۲ = \{a^n b^l : n, l \geq ۰\}$ یک زبان منظم است زیرا یک پذیرندهٔ متناهی قطعی برای آن وجود دارد.
- طبق ویژگی‌های بستاری $L_۳ = \overline{L_۱} \cap L_۲$ باید منظم باشد.
- از آنجایی که $L_۳ = \{a^n b^n : n \geq ۰\}$ است و قبلا ثابت کردیم زبان $L_۳$ منظم نیست، پس فرض اولیه نادرست بوده، و $L_۱$ منظم نیست.