۴ خصوصیات زبان‌های منظم

Properties of Regular languages

◄ آیا همه‌ی زبان‌های رسمی، منظم هستند؟

◄ آیا هر زبانی با یک آتوماتون متناهی (هرچند پیچیده) قابل پذیرش است؟

◄ پاسخ پرسش‌های فوق منفی است.

خصوصیات زبان:

- **خواص بستاری (Closure Properties)**
  آیا حاصل یک عملیات بر روی زبان‌های منظم، زبانی منظم است؟
- **پرسش‌های تصمیم‌گیری (Deciding Questions)**
  آیا می‌توان در مورد خواص یک زبان (متناهی بودن، تساوی، ...) اظهار نظر کرد؟

۴-۱ خواص بستاری زبان‌های منظم

۴-۱-۱ بستار برای اعمال متداول زبان‌ها

قضیه

اگر زبان‌های L_1 و L_2 منظم باشد، آن‌گاه

۱) $L_1 \cup L_2$ منظم است.

۲) $L_1 \cap L_2$ منظم است.

۳) $L_1 L_2$ منظم است.

۴) $\bar{L_1}$ منظم است.

۵) L_1^* منظم است.

۶) L_1^R منظم است.

۷) $L_1 - L_2$ منظم است.

◄ **اثبات.**

برای خلاصه‌تر شدن روند اثبات، نمادگذاری زیر را استفاده می‌کنیم:

$REG(\Sigma)$: خانواده‌ی تمام زبان‌های منظم روی الفبای Σ،

$REGEX(\Sigma)$: مجموعه‌ی تمام عبارت‌های منظم روی الفبای Σ،
$FA(\Sigma)$: مجموعه‌ی تمام آتوماتای متناهی روی الفبای Σ.
$DFA(\Sigma)$: مجموعه‌ی تمام آتوماتای متناهی قطعی روی الفبای Σ.
(۵،۳،۱)

$$\begin{cases} L_1 \in REG(\Sigma) \Rightarrow \exists r_1 \in REGEX(\Sigma)(L(r_1) = L_1) \\ L_2 \in REG(\Sigma) \Rightarrow \exists r_2 \in REGEX(\Sigma)(L(r_2) = L_2) \end{cases} \Rightarrow$$

$$\begin{aligned} r_1 r_2 \in REGEX(\Sigma) &\Rightarrow L(r_1 r_2) \in REG(\Sigma) \\ &\Rightarrow L(r_1)L(r_2) = L_1 L_2 \in REG(\Sigma), \\ r_1 + r_2 \in REGEX(\Sigma) &\Rightarrow L(r_1 + r_2) \in REG(\Sigma) \\ &\Rightarrow L(r_1) \cup L(r_2) = L_1 \cup L_2 \in REG(\Sigma), \\ r_1^* \in REGEX(\Sigma) &\Rightarrow L(r_1^*) \in REG(\Sigma) \\ &\Rightarrow [L(r_1)]^* = L_1^* \in REG(\Sigma). \end{aligned}$$

(۲)

$$\begin{cases} L_1 \in REG(\Sigma) \Rightarrow \exists M_1 \in FA(\Sigma)(M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1) \wedge L(M_1) = L_1)) \\ L_2 \in REG(\Sigma) \Rightarrow \exists M_2 \in FA(\Sigma)(M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2) \wedge L(M_2) = L_2)) \end{cases}$$

آتوماتون $M' = (Q', \Sigma, \delta', q'_0, F')$ را به گونه‌ای می‌سازیم که

$$\begin{cases} Q' = Q_1 \times Q_2 \\ \forall a \in \Sigma \delta'((q,p),a) = (\delta_1(q,a), \delta_2(p,a)) \\ q'_0 = (q_0^1, q_0^2) \\ F' = \{(q,p) \in Q' : q \in F_1 \wedge p \in F_2\} \end{cases}$$

در این صورت داریم

$$\begin{aligned} L(M') &= \{w \in \Sigma^* : \delta'^*(q'_0, w) \in F'\} \\ &= \{w \in \Sigma^* : \delta'^*((q_0^1, q_0^2), w) \in F'\} \\ &= \{w \in \Sigma^* : (\delta_1^*(q_0^1, w), \delta_2^*(q_0^2, w)) \in F'\} \\ &= \{w \in \Sigma^* : \delta_1^*(q_0^1, w) \in F_1 \wedge \delta_2^*(q_0^2, w) \in F_2\} \\ &= \{w \in \Sigma^* : \delta_1^*(q_0^1, w) \in F_1\} \cap \{w \in \Sigma^* : \delta_2^*(q_0^2, w) \in F_2\} \\ &= L(M_1) \cap L(M_2) \\ &= L_1 \cap L_2. \end{aligned}$$

بنابراین

$$\exists M' \in FA(\Sigma)(L(M') = L_1 \cap L_2) \Rightarrow L_1 \cap L_2 \in REG(\Sigma)$$

(۴)

$$L_1 \in REG(\Sigma) \Rightarrow \exists M_1 \in DFA(\Sigma)(M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1) \wedge L(M_1) = L_1))$$

آتوماتون $M' = (Q', \Sigma, \delta', q_0', F')$ را به گونه‌ای می‌سازیم که

$$\begin{cases} Q' = Q_1 \\ \delta' = \delta_1 \\ q_0' = q_0^1 \\ F' = Q - F_1 \end{cases}$$

در این صورت داریم

$$\begin{aligned} L(M') &= \{w \in \Sigma^* : \delta'^*(q_0', w) \in F'\} \\ &= \{w \in \Sigma^* : \delta'^*(q_0', w) \in Q - F_1\} \\ &= \{w \in \Sigma^* : \delta_1^*(q_0, w) \notin F_1\} \\ &= \overline{L(M_1)} \\ &= \overline{L_1} \end{aligned}$$

بنابراین

$$\exists M' \in DFA(\Sigma)(L(M') = \overline{L_1}) \Rightarrow \overline{L_1} \in REG(\Sigma)$$

(۶)

$$L_1 \in REG(\Sigma) \Rightarrow \exists M_1 \in FA(\Sigma)(M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1) \wedge L(M_1) = L_1))$$

با توجه به اینکه می‌توان M_1 را به یک آتوماتون معادل تبدیل کرد که فقط یک حالت نهایی q_f^1 را داشته باشد، یعنی $F = \{q_f^1\}$، آتوماتون $M' = (Q', \Sigma, \delta', q_0', F')$ را به گونه‌ای می‌سازیم که

$$\begin{cases} Q' = Q_1 \\ \forall a \in \Sigma \ q \in \delta'(p, a) \Leftrightarrow p \in \delta_1(q, a) \\ q_0' = q_f^1 \\ F' = \{q_0^1\} \end{cases}$$

در این صورت داریم

$$\begin{array}{rcll} w \in L(M_1) & \Leftrightarrow & \delta_1^*(q_0^1, w) = \{q_f^1\} & \\ & \Leftrightarrow & \delta'^*(q_f^1, w^R) = \{q_0^1\} & \text{(اثبات به استقرا)} \\ & \Leftrightarrow & \delta'^*(q_0', w^R) = F & \\ & \Leftrightarrow & \delta'^*(q_0', w^R) \subseteq F & \\ & \Leftrightarrow & w^R \in L(M') & \end{array}$$

بنابراین

$$\exists M' \in DFA(\Sigma)(L(M') = L_1^R) \Rightarrow L_1^R \in REG(\Sigma)$$

(۷)

$$\begin{cases} L_1 \in REG(\Sigma) \Rightarrow \exists M_1 \in FA(\Sigma)(M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1) \wedge L(M_1) = L_1)) \\ L_2 \in REG(\Sigma) \Rightarrow \exists M_2 \in FA(\Sigma)(M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2) \wedge L(M_2) = L_2)) \end{cases}$$

آتوماتون $M' = (Q', \Sigma, \delta', q_0', F')$ را به گونه‌ای می‌سازیم که

$$\begin{cases} Q' = Q_1 \times Q_2 \\ \forall a \in \Sigma \delta'((q,p),a) = (\delta_1(q,a), \delta_2(p,a)) \\ q_0' = (q_0^1, q_0^2) \\ F' = \{(q,p) \in Q' : q \in F_1 \wedge p \notin F_2\} \end{cases}$$

در این صورت داریم

$$\begin{array}{rcl} L(M') & = & \{w \in \Sigma^* : \delta'^*(q_0', w) \in F'\} \\ & = & \{w \in \Sigma^* : \delta'^*((q_0^1, q_0^2), w) \in F'\} \\ & = & \{w \in \Sigma^* : (\delta_1^*(q_0^1, w), \delta_2^*(q_0^2, w)) \in F'\} \\ & = & \{w \in \Sigma^* : \delta_1^*(q_0^1, w) \in F_1 \wedge \delta_2^*(q_0^2, w) \notin F_2\} \\ & = & \{w \in \Sigma^* : \delta_1^*(q_0^1, w) \in F_1\} - \{w \in \Sigma^* : \delta_2^*(q_0^2, w) \in F_2\} \\ & = & L(M_1) - L(M_2) \\ & = & L_1 - L_2. \end{array}$$

بنابراین

$$\exists M' \in FA(\Sigma)(L(M') = L_1 - L_2) \Rightarrow L_1 - L_2 \in REG(\Sigma)$$

۲-۱-۴ چند عمل بر روی زبان‌ها

همریختی

تعریف **همریختی (homomorphism)** فرض کنید Σ و Γ دو الفبا باشند. تابع

$$h : \Sigma \to \Gamma^*$$

را یک همریختی می‌گوییم. یک همریختی، یک جایگزینی است که در آن یک حرف با یک رشته جایگزین می‌شود.
می‌توان دامنه‌ی h را به رشته‌ها توسعه داد:

$$h(a_۱ a_۲ \dots a_n) = h(a_۱)h(a_۲)\dots h(a_n) \qquad , a_i \in \Sigma, ۱ \leq i \leq n$$

تعریف **تصویر همریختی یک زبان (homomorphic image)** اگر $L \subseteq \Sigma^*$ و $h : \Sigma^* \to \Gamma^*$ یک همریختی باشد، آن‌گاه تصویر همریختی L به صورت زیر تعریف می‌شود:

$$h(L) = \{h(w) : w \in L\}$$

خواص همریختی اگر $L_۱$ و $L_۲$ دو زبان روی الفبای Σ باشند:

$$h(L_1 L_2) = h(L_1)h(L_2)$$
$$h(L_1 \cup L_2) = h(L_1) \cup h(L_2)$$
$$h(L_1 \cap L_2) \subseteq h(L_1) \cap h(L_2)$$

قضیه خانواده‌ی زبان‌های منظم تحت هر همریختی دلخواه بسته است.

◄ **اثبات.**

فرض می‌کنیم که L یک زبان منظم باشد، در این صورت یک گرامر خطی از راست با قواعدی به شکل

$$A \to a \qquad \text{یا} \qquad A \to aB$$

وجود دارد که L را تولید می‌کند. اگر h یک همریختی دلخواه باشد، با اعمال آن روی پایانه‌های سمت راست قواعد گرامر فوق به قواعدی به شکل

$$A \to h(a) \qquad \text{یا} \qquad A \to h(a)B$$

می‌رسیم که یک گرامر خطی از راست را ایجاد می‌کند. بسادگی می‌توان نشان داد که گرامر حاصل $h(L)$ را تولید می‌کند. پس $h(L)$ نیز منظم خواهد بود.

اگر L یک زبان منظم و h یک همریختی دلخواه باشد، $h(L)$ نیز منظم است.

خارج قسمت راست

تعریف

خارج قسمت راست دو زبان (right quotient) فرض کنید که $L_۱, L_۲ \subseteq \Sigma^*$. در این صورت، خارج قسمت راست این دو زبان به صورت زیر تعریف می‌شود:

$$L_۱/L_۲ = \{x : xy \in L_۱, \text{ for some } y \in L_۲\}$$

برای به دست آوردن خارج قسمت راست $L_۱$ به $L_۲$ تمامی رشته‌های $L_۱$ که دارای پسوندی متعلق به $L_۲$ هستند را در نظر می‌گیریم. هر یک از این رشته‌ها پس از حذف پسوند، متعلق به $L_۱/L_۲$ است.

مثال

اگر $L_۱ = L(a^*baa^*)$ و $L_۲ = L(ab^*)$ باشد، در این صورت $L_۱/L_۲ = L(a^*ba^*)$ خواهد بود.

خواص خارج قسمت راست اگر $L_۱$ و $L_۲$ دو زبان روی الفبای Σ باشند:

$L_1/L_2 \subseteq L_1/\Sigma^*$

$\lambda \in L_2 \Rightarrow L_1 \subseteq L_1/L_2$

قضیه

خانواده‌ی زبان‌های منظم تحت عمل خارج قسمت راست نسبت به یک زبان منظم دیگر بسته است.

◄ اثبات.

اگر $L_۱$ منظم باشد، داریم:

$$L_۱ \in REG(\Sigma) \Rightarrow \exists M \in DFA(\Sigma)(L_۱ = L(M) \wedge M = (Q, \Sigma, \delta, q_۰, F))$$

به ازای i از ۱ تا $|Q|$، آتوماتون قطعی M_i را از روی M می‌سازیم به گونه‌ای که حالت شروع آن با q_i جایگزین شده باشد:

$$M_i = (Q, \Sigma, \delta, q_i, F)$$

حال بررسی می‌کنیم که آیا حداقل یک y در $L(M_i)$ وجود دارد که در $L_۲$ باشد:

$$\delta^*(q_i, y) = q_f \in F,\ q_i \in Q,\ y \in L_۲$$

که در این صورت q_i را به مجموعه‌ی $\hat{F}$ می‌افزاییم:

$$\textbf{if } L_۲ \cap L(M_i) \neq \varnothing \textbf{ then } q_i \in \hat{F}$$

حال آتوماتون قطعی $\hat{M} = (Q, \Sigma, \delta, q_۰, \hat{F})$ را می‌سازیم. نشان می‌دهیم که $L(\hat{M}) = L_۱/L_۲$ است:

$$\begin{aligned}
x \in L_۱/L_۲ &\Rightarrow \exists y \in L_۲(xy \in L_۱)\\
&\Rightarrow \delta^*(q_۰, xy) \in F\\
&\Rightarrow \exists q \in Q(\delta^*(q_۰, x) = q \wedge \delta^*(q, y) \in F)\\
&\Rightarrow q \in \hat{F} \quad (\text{مطابق تعریف})\\
&\Rightarrow x \in L(\hat{M}).\\
x \in L(\hat{M}) &\Rightarrow \delta^*(q_۰, x) = q \wedge q \in \hat{F}\\
&\Rightarrow \exists y \in L_۲(\delta^*(q, y) \in F \wedge xy \in L_۱)\\
&\Rightarrow x \in L_۱/L_۲.
\end{aligned}$$

بنابراین $L(\hat{M}) = L_۱/L_۲$ است، پس

$$\exists \hat{M} \in DFA(\Sigma)(L(\hat{M}) = L_۱/L_۲) \Rightarrow L_۱/L_۲ \in REG(\Sigma)$$

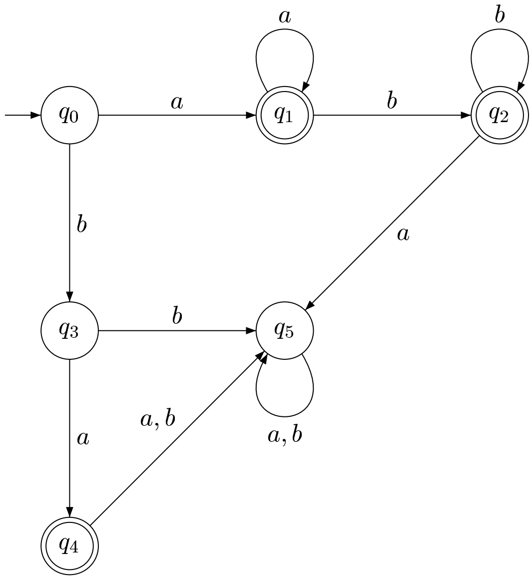
■

اگر $L_۱$ و $L_۲$ زبان‌های منظم باشند، آن‌گاه $L_۱/L_۲$ نیز منظم است.

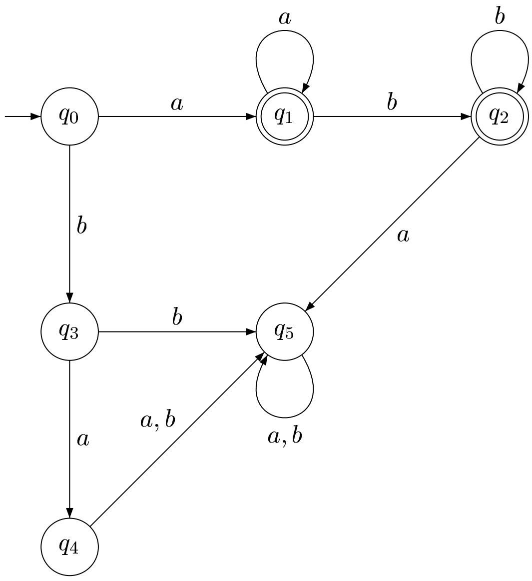
الگوریتم ایجاد DFAی $L_۱/L_۲$ از روی DFAی $L_۱$. چون DFAی مربوط به $L_۱/L_۲$ باید هر پیشوندی از رشته‌های $L_۱$ را بپذیرد که پسوند آن در $L_۲$ باشد،
به ازای هر حالت $q \in Q$ به عنوان حالت شروع تعیین می‌کنیم که آیا گشتی با برچسب v به یک حالت نهایی وجود دارد که در آن $v \in L_۲$ باشد.
اگر بود، هر رشته‌ی x که $\delta^*(q_۰, x) = q$ باشد در $L_۱/L_۲$ قرار دارد. بنابراین باید DFAی $L_۱$ را به گونه‌ای تغییر می‌دهیم q حالت نهایی باشد.

مثال

اگر $L_1 = \{a^n b^m : n \geq 1, m \geq 0\} \cup \{ba\}$ و $L_2 = \{b^m : m \geq 1\}$ باشد، می‌خواهیم L_1/L_2 را محاسبه کنیم. DFAی L_1 به شکل زیر است:



که با اعمال الگوریتم ارایه شده حالت q_4 در مجموعه‌ی حالات نهایی L_1/L_2 قرار نخواهد داشت:



و در نتیجه:

$$L_1/L_2 = \{a^n b^m : n \geq 1, m \geq 0\}$$

خانواده‌ی زبان‌های منظم تحت اعمال زیر بسته است:

الحاق	$L_1 L_2$
اجتماع	$L_1 \cup L_2$
اشتراک	$L_1 \cap L_2$
تفاضل	$L_1 - L_2$
تفاضل متقارن	$L_1 \Theta L_2 = (L_1 - L_2) \cup (L_2 - L_1)$
بستار ستاره‌ای	L_1^*
مکمل	$\bar{L}$
معکوس	L_1^R
همریختی	$h(L)$
خارج قسمت راست	$L_1 / L_2 = \{x : xy \in L_1, \text{ for some } y \in L_2\}$
خارج قسمت چپ	$L_1 \backslash L_2 = \{y : xy \in L_2, \text{ for some } x \in L_1\}$
سر (head)	$head(L_1) = L_1 / \Sigma^*$
ته (tail)	$tail(L_1) = \Sigma^* \backslash L_1$

◄ **تذکر** خانواده‌ی زبان‌های منظم تحت عمل اجتماع و اشتراک نامتناهی بسته نیست.

مثال

زبان $L_i = \{a^i b^i\}$ برای هر i ثابت و معلوم یک زبان منظم است. اما اجتماع نامتناهی زبان‌های منظم L_i

$$L = \bigcup_{i=0}^{\infty} \{a^i b^i\} = \{a^n b^n : n \geq 0\}$$

یک زبان نامنظم را ایجاد می‌کند.

۴-۲ الگوریتم‌های تصمیم‌گیری برای زبان‌های منظم

فرض می‌کنیم که زبان‌های منظم مورد بحث در یک بازنمایی استاندارد نمایش داده شده‌اند.

قضیه

وجود الگوریتم عضویت برای زبان‌های منظم اگر L یک زبان منظم باشد، الگوریتمی وجود دارد که به کمک آن می‌توان تعیین کرد آیا $w \in L$ است یا خیر.

◄ **اثبات.**

زبان L را به صورت DFA نمایش می‌دهیم. در این صورت

$$\begin{cases} \delta^*(q_0, w) \in F \Rightarrow w \in L \\ \delta^*(q_0, w) \notin F \Rightarrow w \notin L \end{cases}$$

بنابراین وجود الگوریتم عضویت واضح است.

■

قضیه

وجود الگوریتم برای تعیین متناهی بودن زبان‌های منظم اگر L یک زبان منظم باشد، الگوریتمی وجود دارد که به کمک آن می‌توان تعیین کرد آیا L تهی، متناهی یا نامتناهی است یا خیر.

◄ **اثبات.**

زبان L را به صورت DFA نمایش می‌دهیم. در این صورت

- اگر مسیر ساده‌ای از q_0 به حداقل یکی از رأس‌های نهایی وجود داشته باشد، $L \neq \varnothing$ است.
- رأس‌هایی که پایه‌ی یک دور در گراف گذر حالت هستند را می‌یابیم.
  اگر یکی از این رأس‌ها روی مسیری از رأس مبدأ به یک رأس نهایی باشد، زبان نامتناهی است.

بنابراین وجود الگوریتم‌های تهی بودن و متناهی بودن واضح است.

■

قضیه

وجود الگوریتم برای تعیین تساوی زبان‌های منظم اگر L_1 و L_2 دو زبان منظم باشند، الگوریتمی وجود دارد که به کمک آن می‌توان تعیین کرد آیا $L_1 = L_2$ است یا خیر.

◄ **اثبات.**

زبان L را به صورت زیر تعریف می‌کنیم:

$$L = L_1 \Theta L_2 = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) \in REG(\Sigma)$$

حال L را به صورت DFA نمایش می‌دهیم و از الگوریتم تهی بودن زبان‌های منظم استفاده می‌کنیم:

$$L = \varnothing \Leftrightarrow L_۱ = L_۲$$

بنابراین وجود الگوریتم تساوی واضح است.

مسایل تصمیم‌پذیر برای زبان‌های منظم:

۱) آیا w در زبان منظم L قرار دارد؟

۲) آیا زبان منظم L تهی است؟

۳) آیا زبان منظم L متناهی است؟

۴) آیا زبان‌های منظم $L_۱$ و $L_۲$ با هم مساوی‌اند؟

۴-۳ شناسایی زبان‌های نامنظم

۴-۳-۱ استفاده از اصل لانه کبوتر

اصل لانه‌ی کبوتر، یک راه حل دقیق برای بیان این جمله به دست می‌دهد که «یک آتوماتون متناهی حافظه‌ی محدود دارد».

مثال

نشان دهید که زبان $L = \{a^n b^n : n \geq 0\}$ است، منظم نیست.

این اثبات با برهان خلف انجام می‌شود. فرض می‌کنیم که L منظم باشد ($\Sigma = \{a, b\}$):

$$L \in REG(\Sigma) \Rightarrow \exists M \in DFA(\Sigma)(M = (Q, \Sigma, \delta, q_0, F) \wedge L(M) = L))$$

حال به $\delta^*(q_0, a^i)$ نگاه می‌کنیم ($i = 1, 2, 3, \ldots$). چون تعداد iها نامحدود است اما $|Q|$ محدود است، پس طبق اصل لانه کبوتری باید حالتی مانند q وجود داشته باشد که

$$\begin{cases} \delta^*(q_0, a^n) = q \ \wedge \ \delta^*(q_0, a^m) = q \ \wedge \ (m \neq n) \\ a^n b^n \in L(M) \Rightarrow \delta^*(q, b^n) = q_f, \ q_f \in F \end{cases} \Rightarrow$$

$$\delta^*(q_0, a^m b^n) = \delta^*(\delta^*(a_0, a^m), b^n) = \delta^*(q, b^n) = q_f$$

اما این خلاف فرض اولیه‌ی $m = n$ است. بنابراین فرض خلف نادرست است و L نمی‌تواند منظم باشد.

۴-۳-۲ لم تزریق

قضیه

لم تزریق اگر L یک زبان منظم نامتناهی باشد، آنگاه عدد صحیح مثبت m وجود دارد به طوری که هر رشته‌ی $w \in L$ با شرط $|w| \geq m$ می‌تواند به صورت

$$w = xyz$$

تجزیه شود که در آن

$$|xy| \leq m, \qquad |y| \geq 1$$

است و هر رشته‌ی w_i به صورت

$$w_i = xy^i z, \qquad i = 0, 1, 2, \ldots$$

در L باشد.

کلیه‌ی رشته‌های به اندازه‌ی کافی بزرگ در L را می‌توان به گونه‌ای به سه قسمت تجزیه کرد که تعداد دلخواهی تکرار رشته‌ی میانی، رشته‌ی دیگری در L را به دست دهد. در این صورت می‌گوییم رشته‌ی میانی **تزریق** شده است.

تذکراتی در مورد استفاده از لم تزریق

- لم تزریق برای نشان دادن منظم نبودن یک زبان استفاده می‌شود، نه منظم بودن آن.
- لم تزریق بر این پایه استوار است که در یک گراف گذر حالت با n راس هرگشت به طول n یا بیشتر باید حداقل یک راس را تکرار کند، یعنی دارای دور است.
- گرچه از وجود m و تجزیه‌ی xyz می‌توانیم مطمئن باشیم، اما نمی‌دانیم که مقدار آنها چیست و نمی‌توانیم به دلیل اینکه لم تزریق به ازای مقادیر خاصی از آنها نقض می‌شود، بگوییم که به تناقض رسیده‌ایم.
- لم تزریق به ازای هر $w \in L$ و i صادق است، پس اگر لم تزریق حتی به ازای یک i یا w نقض شود، زبان مورد نظر نمی‌تواند منظم باشد.

بکارگیری لم تزریق برای اثبات نامنظم بودن یک زبان. استفاده از لم تزریق در یک بحث رقابتی مناسب‌تر است: هدف ما در بردن بازی این است که نشان دهیم L منظم نیست.

	حریف	خودمان
۱	حریف m را انتخاب می‌کند (ثابت نامشخص)	
۲		بر اساس m رشته‌ی $w \in L$ را با شرط $\|w\| \geq m$ انتخاب می‌کنیم.
۳	حریف تجزیه‌ی $w = xyz$ را با شرط $\|xy\| \leq m$ و $\|y\| \geq ۱$ انجام می‌دهد.*	
۴		i را به گونه‌ای انتخاب می‌کنیم که $w_i \notin L$

* فرض می‌کنیم حریف این تجزیه را به گونه‌ای انجام می‌دهد که سخت‌ترین حالت را برای بردن بازی توسط ما ایجاد کند

◄ **نکته** زبانی منظم است که یک ماشین بتواند با پردازش‌هایی که نیاز به حافظه‌ی نامحدود ندارد آن را بپذیرد.

مثال

نشان دهید که زبان های زیر نامنظم هستند.

۱) $\{a^n b^n : n \geq ०\}$

	حریف		خودمان
۱	$m \geq 0$	۲	$w = a^m b^m \in L,$ $\|w\| = 2m \geq m$
۳	$w = xyz = \underbrace{a^{m-k-l}}_{x}\underbrace{a^k}_{y}\underbrace{a^l b^m}_{z},$ $k \geq 1$ $w_i = xy^iz = a^{m-k-l}(a^k)^i a^l b^m$	۴	$i = 0 \Rightarrow w_0 = a^{m-k}b^m \notin L$

۲) $\{ww^R : w \in \{a,b\}^*\}$

$$\overbrace{aa\ldots a}^{m}\overbrace{bb\ldots b}^{m}\overbrace{bb\ldots b}^{m}\overbrace{aa\ldots a}^{m}$$

$$x \quad y \qquad\qquad z$$

خودمان	حریف
	۱ $m \geq 0$
۲ $w = a^m b^m b^m a^m \in L,$ $\lvert w\rvert = 4m \geq m$	
	۳ $w = xyz =$ $\underbrace{a^{m-k-l}}_{x}\underbrace{a^k}_{y}\underbrace{a^l b^m b^m a^m}_{z},$ $k \geq 1$ $w_i = xy^iz =$ $a^{m-k-l}(a^k)^i a^l b^m b^m a^m$
۴ $i = 0 \Rightarrow$ $w_0 = a^{m-k} b^{2m} a^m \notin L$	

۳) $\{a^{n!} : n \geq 0\}$

خودمان	حریف
	۱ $m \geq 0$
۲ $w = a^{m!}$ $\lvert w\rvert = m! \geq m$	
	۳ $w = xyz =$ $\underbrace{a^k}_{x}\underbrace{a^l}_{y}\underbrace{a^{m!-k-l}}_{z},$ $k \geq 1,\ k + l \leq m$ $w_i = xy^iz =$ $a^k(a^l)^i a^{m!-k-l}$
۴ $i = 0 \Rightarrow$ $w_0 = a^{m!-k} \notin L$	

زیرا $m! - k = j!$ برای هیچ j صحیحی ممکن نیست؛ زیرا برای $m > 2$ و $k \leq m$ داریم:

$$m! - k > (m-1)!$$

۴) $L = \{a^n b^k c^{n+k} : n \geq 0, k \geq 0\}$

همریختی h را به صورت زیر تعریف می‌کنیم:

$$\begin{cases} h(a) = h(b) = a \\ h(c) = b \end{cases}$$

داریم:

$$\begin{aligned} h(L) &= \{a^n a^k b^{n+k} : n, k \geq 0\} \\ &= \{a^{n+k} b^{n+k} : n, k \geq 0\} \\ &= \{a^m b^m : m \geq 0\} \notin REG(\{a,b\}) \end{aligned}$$

چون $h(L)$ منظم نیست، پس L هم نمی‌تواند منظم باشد:

$$h(L) \notin REG \Rightarrow L \notin REG$$

۵) $L = \{a^n b^m : n \neq m\}$

زبان L' را به صورت زیر تعریف می‌کنیم:

$$L' = \overline{L} \cap L(a^*b^*) = \{a^n b^n : n \geq ۰\}, \qquad \Sigma = \{a, b\}$$

برهان خلف: فرض می‌کنیم L منظم باشد، در این صورت

$$\begin{cases} L \in REG(\Sigma) \Rightarrow \overline{L} \in REG(\Sigma) \\ L(a^*b^*) \in REG(\Sigma) \end{cases} \Rightarrow \overline{L} \cap L(a^*b^*) \in REG(\Sigma) \Rightarrow L' \in REG(\Sigma)$$

اما می‌دانیم که L' منظم نیست، پس فرض خلف باطل بوده و L نمی‌تواند منظم باشد.

▲

❋ اثبات لم تزریق

اثبات لم تزریق با برهان خلف انجام می‌شود. فرض می‌کنیم که L یک زبان منظم باشد، در این صورت DFAی M با مجموعه‌ی حالات $Q = \{q_۰, q_۱, \ldots, q_n\}$ برای آن وجود خواهد داشت:

$$L \in REG(\Sigma) \Rightarrow \exists M \in DFA(\Sigma)(M = (Q, \Sigma, \delta, q_۰, F) \wedge L = L(M))$$

رشته‌ی $w \in L$ را به گونه‌ای انتخاب می‌کنیم که $|w| \geq m = |Q| + ۱ = n + ۱$ باشد.
(چون L نامتناهی فرض شده است، این کار ممکن است.)
دنباله‌ی حالاتی که آتوماتون حین پردازش w از آنها می‌گذرد را در نظر می‌گیریم:

$$q_۰, q_i, q_j, \ldots, q_f$$

چون این دنباله دقیقاً دارای $|w| + ۱$ عنصر است،
طبق اصل لانه‌ی کبوتر حداقل یکی از این عناصر باید تکرار شود
و این تکرار نباید پس از حرکت nام رخ دهد.
پس شکل این دنباله می‌شود:

$$q_۰, q_i, q_j, \ldots, \underbrace{q_r, q_r, \ldots, q_r}, \ldots, q_f$$

در نتیجه باید زیررشته‌های x، y و z از w به شکلی موجود باشد که

$$\begin{cases} \delta^*(q_۰, x) = q_r \\ \delta^*(q_r, y) = q_r \\ \delta^*(q_r, z) = q_f \end{cases}$$

که در آن $m = n + ۱ \geq |xy|$ و $|y| \geq ۱$. از اینجا خواهیم داشت:

$$\begin{array}{rcl}\delta^*(q_۰, xz) & = & q_f \\ \delta^*(q_۰, xy^۲z) & = & q_f \\ \delta^*(q_۰, xy^۳z) & = & q_f \\ & \vdots & \end{array}$$

و اثبات کامل می‌شود.