به نام خدا

نظریهٔ زبان‌ها و ماشین‌ها

آرش شفیعی

ماشین‌های پشته‌ای

ماشین‌های پشته‌ای

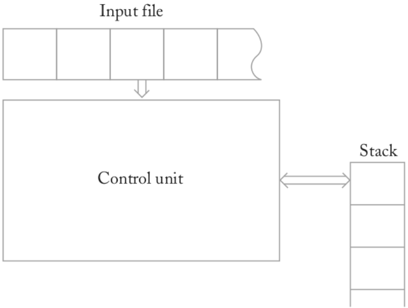
- علاوه بر گرامرهای مستقل از متن، ماشین‌های پشته‌ای [1] نیز برای پذیرش زبان‌های مستقل از متن به کار می‌روند.
- ماشین‌های پشته‌ای محدودیت ماشین‌های متناهی را که کمبود حافظهٔ آنها بود، با اضافه کردن یک پشته [2] نامحدود رفع می‌کنند.
- برای مثال برای پذیرش زبان $L = \{ww^R : w \in \Sigma^*\}$ نیاز به ذخیرهٔ نمادهای رشتهٔ w داریم. از آنجایی که w نامحدود است، نیاز به یک حافظهٔ نامحدود داریم.
- ماشین‌های پشته‌ای نیز به دو صورت قطعی و غیرقطعی هستند. ماشین‌های پشته‌ای غیرقطعی زبان‌های مستقل از متن را پذیرش می‌کنند اما ماشین‌های پشته‌ای قطعی تنها زیرمجموعه‌ای از این زبان‌ها را می‌پذیرند.

---

[1] pushdown automata
[2] stack

ماشین‌های پشته‌ای

- شمای کلی ماشین پشته‌ای در زیر آمده است. در هر حرکت، ماشین یک نماد از ورودی و یک نماد از روی پشته می‌خواند. با توجه به نمادهای خوانده شده، ماشین حالت خود را تغییر می‌دهد و بر روی پشته می‌نویسد.



ماشین‌های پشته‌ای غیرقطعی

- پذیرندهٔ پشته‌ای غیرقطعی [1] (npda) به صورت یک هفت‌تایی $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ تعریف می‌شود، به طوری که :
- Q مجموعه‌ای است متناهی از حالات داخلی واحد کنترل
- Σ الفبای ورودی است
- Γ مجموعه‌ای متناهی از نمادهاست به نام الفبای پشته [2]
- $\delta : (Q \times (\Sigma \cup \{\lambda\}) \times \Gamma) \rightarrow$ (زیرمجموعه‌ای متناهی از $Q \times \Gamma^*$) تابع گذار است.
- $q_0 \in Q$ حالت آغازی واحد کنترل است
- $z \in \Gamma$ نماد آغازی پشته [3] است.
- $F \subseteq Q$ مجموعه‌ای از حالات پایانی است.

[1] nondeterministic pushdown accepter (npda)
[2] stack alphabet
[3] stack start symbol

ماشین‌های پشته‌ای غیرقطعی

- طبق تعریف ماشین پشته‌ای غیرقطعی، در هر حرکت با توجه به نماد ورودی، آخرین نماد بر روی پشته و حالت فعلی، حالت بعدی انتخاب می‌شود و یک رشته بر روی پشته نوشته می‌شود.
- نماد خوانده شده از ورودی می‌تواند تهی باشد که این یک گذار تهی [1] است.
- همچنین برای یک گذار، پشته نمی‌تواند خالی باشد.
- از آنجایی که این ماشین غیرقطعی است، پس در هر گذار، اگر چندین انتخاب وجود داشته باشد، یک کپی از ماشین به ازای هر انتخاب به اجرا ادامه می‌دهد و اگر یکی از کپی‌های ماشین، با اتمام خواندن رشته از ورودی، در یک حالت پایانی متوقف شد، رشته پذیرفته می‌شود.

[1] λ–transition

ماشین‌های پشته‌ای غیرقطعی

- فرض کنید یک قانون گذار در یک ماشین پشته‌ای غیرقطعی به صورت $\delta(q_1, a, b) = \{(q_2, cd), (q_3, \lambda)\}$ تعریف شده باشد.
- در صورتی که ماشین در حالت q_1 باشد و نماد a از ورودی خوانده شود و نماد b در بالای پشته باشد، آنگاه دو گذار ممکن خواهد بود.
- یا ماشین به حالت q_2 می‌رود و رشتهٔ cd در پشته نوشته می‌شود (نوشتن یک رشته نماد به نماد از راست به چپ رشته صورت می‌گیرد).
- یا ماشین به حالت q_3 می‌رود و چیزی در پشته نوشته نمی‌شود (در این صورت نماد b از بالای پشته حذف می‌شود).

ماشین‌های پشته‌ای غیرقطعی

- ماشین پشته‌ای غیرقطعی با $Q = \{q_۰, q_۱, q_۲, q_۳\}$، $\Sigma = \{a, b\}$، $\Gamma = \{۰, ۱\}$، $z = ۰$، $F = \{q_۳\}$ و حالت اولیهٔ $q_۰$ و تابع گذار به صورت زیر را در نظر بگیرید.
- این ماشین چه زبانی را شناسایی می‌کند؟

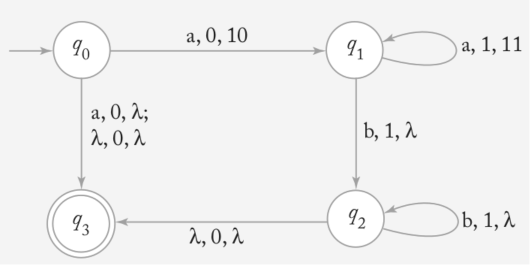
$$\delta(q_۰, a, ۰) = \{(q_۱, ۱۰), (q_۳, \lambda)\}$$
$$\delta(q_۰, \lambda, ۰) = \{(q_۳, \lambda)\}$$
$$\delta(q_۱, a, ۱) = \{(q_۱, ۱۱)\}$$
$$\delta(q_۱, b, ۱) = \{(q_۲, \lambda)\}$$
$$\delta(q_۲, b, ۱) = \{(q_۲, \lambda)\}$$
$$\delta(q_۲, \lambda, ۰) = \{(q_۳, \lambda)\}$$

ماشین‌های پشته‌ای غیرقطعی

- دقت کنید که برخی از گذارها در این ماشین تعریف نشده‌اند، مثلا $\delta(q_0, b, 0)$. در صورتی که ماشین در این پیکربندی قرار بگیرد، به بن‌بست می‌خورد (به پیکربندی مرده می‌رود).
- در این ماشین در حالت q_1 با خواندن نماد a نماد ۱ به پشته اضافه می‌شود و با خواندن نماد b ماشین به حالت q_2 می‌رود و با خواندن نمادهای b متوالی در این حالت نماد ۱ از پشته حذف می‌شود.
- این ماشین تعداد نمادهای a و b را شمارش می‌کند و اگر تعداد این نمادها برابر باشد به حالت پایانی می‌رود.
- این ماشین زبان $L = \{a^n b^n : n \geq 0\} \cup \{a\}$ را می‌پذیرد.

ماشین‌های پشته‌ای غیرقطعی

- ماشین‌های پشته‌ای را می‌توانیم به صورت یک گراف گذار نیز نمایش دهیم.
- در اینصورت برچسب روی یال‌ها به صورت یک سه‌تایی a, b, c است که در آن a نماد خوانده شده از ورودی، b نماد برداشته شده از پشته، و c رشتهٔ نوشته شده بر روی پشته است.



ماشین‌های پشته‌ای غیرقطعی

- در هر لحظه، یک ماشین پشته‌ای در حالت q قرار دارد، رشتهٔ w از ورودی هنوز خوانده نشده است، و رشتهٔ u در پشته قرار دارد (نماد سمت چپ در رشتهٔ u در بالای پشته قرار دارد).
- این سه مقدار را به صورت یک سه‌تایی (q, w, u) نشان می‌دهیم و آن را توصیف لحظه‌ای [1] ماشین پشته‌ای می‌خوانیم.
- یک حرکت در ماشین پشته‌ای را با نماد $\vdash$ نشان می‌دهیم.
- بنابراین داریم $(q_2, w, yx) \vdash (q_1, aw, bx)$ اگر و تنها اگر $(q_2, y) \in \delta(q_1, a, b)$ باشد.

[1] instantaneous description

ماشین‌های پشته‌ای غیرقطعی

- اگر یک حرکت شامل چندین گام باشد آن را با $\overset{*}{\vdash}$ نشان می‌دهیم.
- بنابراین عبارت $(q_1, w_1, x_1) \overset{*}{\vdash} (q_2, w_2, x_2)$ بدین معناست که در حرکتی با چند گام ماشین می‌تواند از پیکربندی اول به پیکربندی دوم برسد.
- وقتی چندین ماشین را به طور همزمان بررسی می‌کنیم، می‌نویسیم $\vdash_M$ بدین معنا که حرکت در ماشین M مد نظر است.

ماشین‌های پشته‌ای غیرقطعی

- فرض کنید $M = (Q, \Sigma, \Gamma, \delta, q_\circ, z, F)$ یک ماشین پشته‌ای غیرقطعی باشد.
- زبانی که توسط M پذیرفته می‌شود $L(M)$ بدین صورت تعریف می‌شود:
$$L(M) = \{w \in \Sigma^* : (q_\circ, w, z) \overset{*}{\vdash}_M (p, \lambda, u), p \in F, u \in \Gamma^*\}$$
- به عبارت دیگر زبانی که توسط ماشین M پذیرفته می‌شود مجموعهٔ همهٔ رشته‌هایی است که ماشین M را در پایان رشته در یک حالت پایانی قرار دهد. محتوای پشتهٔ u در پایان خواندن رشته بی‌اهمیت است.

ماشین‌های پشته‌ای غیرقطعی

- یک npda برای زبان L طراحی کنید: $L = \{w \in \{a, b\}^* : n_a(w) = n_b(w)\}$

ماشین‌های پشته‌ای غیرقطعی

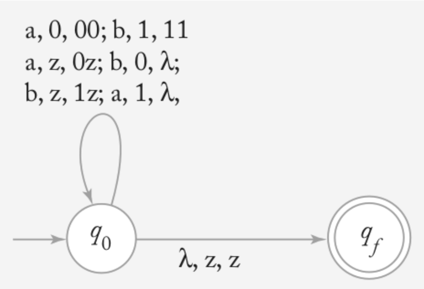
- یک npda برای زبان L طراحی کنید: $L = \{w \in \{a, b\}^* : n_a(w) = n_b(w)\}$
- یک راه حل ابتدایی: می‌توانیم با خواندن a یک نماد مانند نماد صفر به پشته اضافه کنیم و با خواندن b یک نماد صفر از بالای پشته حذف کنیم.
- مشکل این راه حل این است که اگر در ابتدا تعدادی b مشاهده کنیم پشته خالی می‌شود و ماشین متوقف می‌شود.

ماشین‌های پشته‌ای غیرقطعی

- یک npda برای زبان L طراحی کنید: $L = \{w \in \{a, b\}^* : n_a(w) = n_b(w)\}$
- یک راه حل دیگر: می‌توانیم با خواندن b در صورتی که پشته خالی بود نماد ۱ را به ازای اعداد منفی به پشته اضافه کنیم.
- بنابراین با خواندن a اگر در محدودهٔ اعداد مثبت قرار داشتیم (نماد صفر در بالای پشته دیدیم)، نماد صفر را به پشته اضافه می‌کنیم. و اگر در محدودهٔ اعداد منفی قرار داشتیم (نماد ۱ در بالای پشته دیدیم)، نماد ۱ را از پشته حذف می‌کنیم.
- همچنین با خواندن b اگر در محدودهٔ اعداد مثبت قرار داشتیم (نماد صفر در بالای پشته دیدیم)، نماد صفر را از پشته حذف می‌کنیم. و اگر در محدودهٔ اعداد منفی قرار داشتیم (نماد ۱ در بالای پشته دیدیم)، نماد ۱ را به پشته اضافه می‌کنیم.

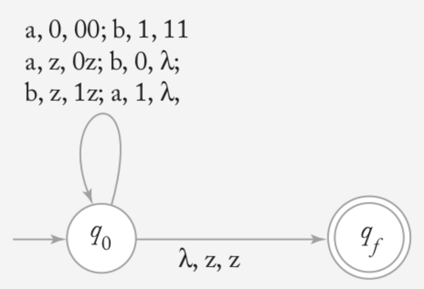
ماشین‌های پشته‌ای غیرقطعی

- یک npda برای زبان L طراحی کنید: $L = \{w \in \{a, b\}^* : n_a(w) = n_b(w)\}$



ماشین‌های پشته‌ای غیرقطعی

- برای مثال برای رشتهٔ baab داریم:
- $(q_0, baab, z) \vdash (q_0, aab, 1z) \vdash (q_0, ab, z) \vdash (q_0, b, 0z) \vdash (q_0, \lambda, z) \vdash (q_f, \lambda, z)$
- بنابراین رشته پذیرفته می‌شود.



ماشین‌های پشته‌ای غیرقطعی

- یک npda برای زبان L طراحی کنید: $L = \{ww^R : w \in \{a, b\}^+\}$

ماشین‌های پشته‌ای غیرقطعی

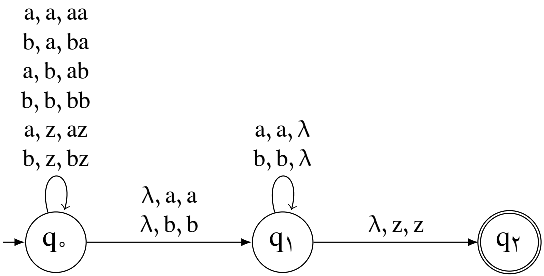
- یک npda برای زبان L طراحی کنید: $L = \{ww^R : w \in \{a, b\}^+\}$
- به ازای خواندن هر نماد از ورودی، آن نماد را در پشته ذخیره می‌کنیم. بعد از خواندن w برای بررسی w^R به ازای خواندن هر نماد باید آن نماد را با نماد بالای پشته مقایسه کنیم و در صورتی که دو نماد برابر بودند، نماد بالای پشته را حذف کنیم. اگر در پایان خواندن رشته به پشتهٔ خالی رسیدیم، رشته پذیرفته می‌شود.
- تنها مشکل این راه حل این است که نمی‌دانیم در کدام لحظه به وسط رشته رسیده‌ایم. ولی از آنجایی که ماشین غیرقطعی است، همهٔ مسیرهای ممکن برای رسیدن به یک حالت پایانی بررسی می‌شوند.

ماشین‌های پشته‌ای غیرقطعی

- یک npda برای زبان L طراحی کنید: $L = \{ww^R : w \in \{a, b\}^+\}$
- ماشین $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ با $Q = \{q_0, q_1, q_2\}$، $\Sigma = \{a, b\}$، $\Gamma = \{a, b, z\}$، $F = \{q_2\}$ را در نظر می‌گیریم.
- برای خواندن قسمت w گذارهای زیر را تعریف می‌کنیم: $\delta(q_0, a, a) = \{(q_0, aa)\}$، $\delta(q_0, b, a) = \{(q_0, ba)\}$، $\delta(q_0, a, b) = \{(q_0, ab)\}$، $\delta(q_0, b, b) = \{(q_0, bb)\}$، $\delta(q_0, a, z) = \{(q_0, az)\}$، $\delta(q_0, b, z) = \{(q_0, bz)\}$
- برای حدس زدن وسط رشته و گذار از q_0 به q_1 گذارهای زیر را تعریف می‌کنیم: $\delta(q_0, \lambda, a) = \{(q_1, a)\}$، $\delta(q_0, \lambda, b) = \{(q_1, b)\}$
- برای مقایسه w^R در برابر محتوای پشته گذارهای زیر را تعریف می‌کنیم: $\delta(q_1, a, a) = \{(q_1, \lambda)\}$، $\delta(q_1, b, b) = \{(q_1, \lambda)\}$
- در نهایت برای پذیرفتن رشته گذار زیر را تعریف می‌کنیم: $\delta(q_1, \lambda, z) = \{(q_2, z)\}$

ماشین‌های پشته‌ای غیرقطعی

- یک npda برای زبان L طراحی کنید: $L = \{ww^R : w \in \{a, b\}^+\}$



ماشین‌های پشته‌ای غیرقطعی

- دنبالهٔ حرکت‌ها برای پذیرفتن رشتهٔ abba چنین است:
$(q_0, abba, z) \vdash (q_0, bba, az) \vdash (q_0, ba, baz) \vdash$
$(q_1, ba, baz) \vdash (q_1, a, az) \vdash (q_1, \lambda, z) \vdash (q_2, z)$
- در وسط رشته یعنی در جایی که توصیف لحظه‌ای ماشین (q_0, ba, baz) است، ماشین دو انتخاب برای حرکت خود دارد.
- یکی از انتخاب‌ها استفاده از گذار $\delta(q_0, b, b) = \{(q_0, bb)\}$ است که منجر به حرکت $(q_0, ba, baz) \vdash (q_0, a, bbaz)$ می‌شود.
- انتخاب دوم استفاده از گذار $\delta(q_0, \lambda, b) = \{(q_1, b)\}$ است. این انتخاب منجر به پذیرفتن رشته می‌شود بنابراین ماشین این گذار را انتخاب می‌کند.

مثال

- یک ماشین پشته‌ای برای زبان $L = \{a^n b^{3n} : n \geq 0\}$ طراحی کنید.

مثال

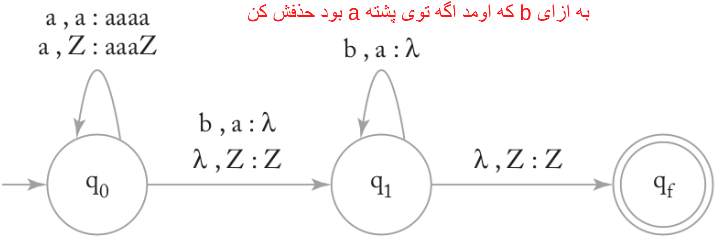
نکته: ماشین پشته ای عملا توانایی شمارش رو به ما می رسونه

نکته: در هر مرحله یک ورودی می بینیم

- یک ماشین پشته‌ای برای زبان $L = \{a^n b^{3n} : n \geq 0\}$ طراحی کنید.

استیت q0 میگه توی پشته ما یا اصلا هیچی نیست یا قبلا a بوده

به ازای b که اومد اگه توی پشته a بود حذفش کن



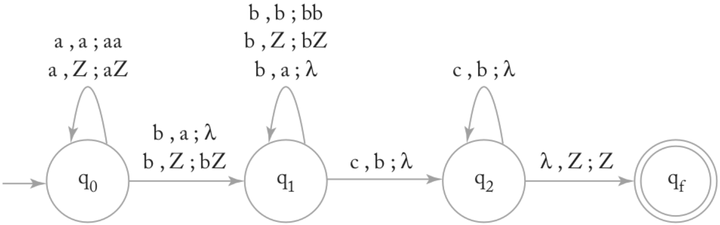
یعنی به ازای هر a بتونیم بریم به q1
ما اول کار نمیدونیم بعد از چندتا a می ریم توی استیت بعدی واسه همین توی دوتا استیت q0 , q1 همزمان هستیم یعنی همونی که میگه لاندا و z که این یعنی هیچی نیاد و توی پشتمون هم چیزی نباشه و دوباره توی پشته چیزی نباشه یعنی همزمان هم توی q0 هستیم و هم توی q1 یعنی ممکنه اول کار هیچ a نبینیم

مثال

- یک ماشین پشته‌ای برای زبان $L = \{a^n b^{n+m} c^m : n \geq ۰, m \geq ۱\}$ طراحی کنید.

مثال

- یک ماشین پشته‌ای برای زبان $L = \{a^n b^{n+m} c^m : n \geq ۰, m \geq ۱\}$ طراحی کنید.



ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- نشان می‌دهیم که برای هر زبان مستقل از متن یک ماشین پشته‌ای وجود دارد که آن را می‌پذیرد.
- برای سادگی اثبات فرض می‌کنیم که گرامر مستقل از متن به فرم گریباخ تبدیل شده است.
- به طور خلاصه، ماشین پشته‌ای، هر قانون گرامر به صورت $A \rightarrow ax$ را بدین گونه شبیه‌سازی می‌کند که با خواندن نماد پایانی سمت راست قانون $(a \in T)$ از ورودی و خواندن متغیر سمت چپ قانون $(A \in V)$ از پشته، متغیرهای سمت راست قانون $(x \in V^*)$ را در پشته ذخیره می‌کند.

ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- یک ماشین پشته‌ای طراحی کنید که زبان تولید شده توسط یک گرامر با قوانین تولید $S \rightarrow aSbb|a$ را می‌پذیرد.

ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- یک ماشین پشته‌ای طراحی کنید که زبان تولید شده توسط یک گرامر با قوانین تولید $S \to aSbb|a$ را می‌پذیرد.
- ابتدا گرامر را به فرم نرمال گریباخ تبدیل می‌کنیم، بنابراین قوانین $S \to aSA|a \ , \ A \to bB \ , \ B \to b$ را به دست می‌آوریم.
- ماشین پشته‌ای معادل آن سه حالت دارد: $\{q_0, q_1, q_2\}$ به طوری که حالت q_0 حالت آغازی و حالت q_2 یک حالت پایانی است.
- در حالت اولیه متغیر آغازی را به پشته اضافه می‌کنیم و به حالت q_1 گذار می‌کنیم:
$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$

ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- قانون $S \rightarrow aSA|a$ را بدین صورت شبیه‌سازی می‌کنیم که با خواندن a از ورودی و خواندن S از پشته یا SA را به پشته اضافه می‌کنیم و یا به پشته چیزی اضافه نمی‌کنیم، بنابراین داریم:
$$\delta(q_1, a, S) = \{(q_1, SA), (q_1, \lambda)\}$$
- همچنین برای قوانین دیگر گرامر داریم: $\delta(q_1, b, A) = \{(q_1, B)\}$ ، $\delta(q_1, b, B) = \{(q_1, \lambda)\}$
- اگر در ورودی نمادی باقی نماند و پشته خالی شود می‌توانیم گذار به حالت پایانی را انجام دهیم:
$$\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$$
- این الگوریتم را به حالت کلی تعمیم می‌دهیم.

ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- برای هر زبان مستقل از متن L یک ماشین پشته‌ای غیرقطعی M وجود دارد به طوری که $L = L(M)$.
- اگر L یک زبان مستقل از متن بدون رشتهٔ تهی باشد، آنگاه یک گرامر مستقل از متن در فرم نرمال گریباخ برای آن وجود دارد.
- فرض کنید این گرامر $G = (V, T, S, P)$ باشد.
- می‌توانیم یک ماشین پشته‌ای طراحی کنیم که اشتقاق‌های چپ این گرامر را شبیه‌سازی کند.
- ماشین پشته‌ای $M = (\{q_۰, q_۱, q_f\}, T, V \cup \{z\}, \delta, q_۰, z, \{q_f\})$ را در نظر بگیرید، به طوری که $z \not\in V$.
- در این ماشین الفبای ورودی برابر با مجموعهٔ نمادهای پایانی گرامر G و الفبای پشته مجموعهٔ متغیرهای گرامر است.

ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- تابع گذار را به ازای مقادیر اولیه تعریف می‌کنیم: $\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$

- بنابراین در حرکت اول، متغیر آغازی S را به پشته اضافه می‌کنیم. از نماد z برای تشخیص دادن خالی شدن پشته و پایان فرایند اشتقاق استفاده می‌کنیم.

- همچنین به ازای هر یک از قوانین $A \rightarrow au$ در گرامر، تابع گذار را بدین صورت تعریف می‌کنیم: $(q_1, u) \in \delta(q_1, a, A)$.

- برای شبیه‌سازی هر اشتقاق که A را به au تبدیل می‌کند، در ماشین a را از ورودی خوانده و متغیر A را از پشته حذف و متغیرهای u را به پشته اضافه می‌کنیم.

- در پایان در صورتی که رشته به پایان برسد، و پشته خالی از متغیرهای گرامر شود، به حالت پایانی گذار می‌کنیم: $\delta(q_1, \lambda, z) = \{(q_f, z)\}$

ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- می‌توان نشان داد که برای هر اشتقاق در فرایند اشتقاق یک گرامر، یک حرکت متناظر در یک ماشین پشته‌ای غیرقطعی وجود دارد، و بنابراین هر جملهٔ w که از گرامر G به دست می‌آید را می‌توان توسط ماشین پشته‌ای متناظر آن پذیرفت.
- همینطور به ازای هر حرکت در ماشین پشته‌ای غیر قطعی M ، یک اشتقاق در گرامر G که ماشین M با قوانین تولید آن ساخته شده است وجود دارد، و بنابراین هر جملهٔ w که توسط ماشین M پذیرفته می‌شود را می‌توان توسط گرامر G به دست آورد.

- برای گرامر $S \to aA \quad , \quad A \to aABC|bB|a \quad , \quad B \to b \quad , \quad C \to c$ یک ماشین پشته‌ای بسازید.

ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- برای گرامر $S \to aA \quad , \quad A \to aABC|bB|a \quad , \quad B \to b \quad , \quad C \to c$ یک ماشین پشته‌ای بسازید.
- برای تابع گذار (گذار از حالت آغازی و گذار به حالت پایانی) چنین تعریف می‌کنیم:

$$\delta(q_0, \lambda, z) = \{(q_1, Sz)\}$$
$$\delta(q_1, \lambda, z) = \{(q_f, z)\}$$

- سپس برای قوانین تولید تابع گذار را چنین تعریف می‌کنیم:

$$\delta(q_1, a, S) = \{(q_1, A)\}$$
$$\delta(q_1, a, A) = \{(q_1, ABC), (q_1, \lambda)\}$$
$$\delta(q_1, b, A) = \{(q_1, B)\}$$
$$\delta(q_1, b, B) = \{(q_1, \lambda)\}$$
$$\delta(q_1, c, C) = \{(q_1, \lambda)\}$$

- برای پذیرفتن رشتهٔ aaabc این ماشین حرکت‌های زیر را انجام می‌دهد:
$$(q_0, aaabc, z) \vdash (q_1, aaabc, Sz) \vdash (q_1, aabc, Az) \vdash (q_1, abc, ABCz)$$
$$\vdash (q_1, bc, BCz) \vdash (q_1, c, Cz) \vdash (q_1, \lambda, z) \vdash (q_f, \lambda, z)$$

- این جمله با استفاده از فرایند اشتقاق زیر مشتق می‌شود:
$$S \Rightarrow aA \Rightarrow aaABC \Rightarrow aaaBC \Rightarrow aaabC \Rightarrow aaabc$$

ماشین‌های پشته‌ای و زبان‌های مستقل از متن

- همچنین روشی برای تبدیل یک ماشین پشته‌ای به یک گرامر مستقل از متن وجود دارد که در اینجا به آن نمی‌پردازیم.
- پس برای هر گرامر مستقل از متن یک ماشین پشته‌ای و برای هر ماشین پشته‌ای یک گرامر مستقل از متن وجود دارد.

ماشین‌های پشته‌ای قطعی

- یک پذیرندهٔ پشته‌ای قطعی[1] (dpda) بر خلاف پذیرندهٔ پشته‌ای غیرقطعی هیچ‌گاه حق انتخاب ندارد.
- ماشین پشته‌ای $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$ قطعی گفته می‌شود اگر تابع گذار آن نسبت به ماشین پشته‌ای غیرقطعی محدودیت‌های زیر را داشته باشد:

۱. $\delta(q, a, b)$ حداکثر یک عضو داشته باشد.

۲. اگر $\delta(q, \lambda, b)$ تهی نباشد، آنگاه $\delta(q, c, b)$ باید به ازای همهٔ مقادیر $c \in \Sigma$ تهی باشد.

- به طوری که $q \in Q, a \in \Sigma \cup \{\lambda\}, b \in \Gamma$

[1] deterministic pushdown accepter (dpda)

ماشین‌های پشته‌ای قطعی

- اولین محدودیت باعث می‌شود در هر حالت با خواندن هر یک از نمادهای الفبا و خواندن هر یک از نمادهای پشته ماشین فقط بتواند به حداکثر یک حالت برود. گرچه ماشین ممکن است به بن‌بست نیز برخورد کند.

- دومین محدودیت باعث می‌شود وقتی گذار تهی برای یک پیکربندی امکان‌پذیر است، هیچ گذار دیگری برای آن پیکربندی با خواندن هیچ نماد دیگری امکان‌پذیر نباشد.

- پس گرچه گذار تهی نیز وجود دارد و ماشین ممکن است به بن‌بست برخورد کند، اما در هر پیکربندی فقط یک گذار ممکن وجود دارد.

- زبان L یک زبان مستقل از متن قطعی [1] گفته می‌شود اگر توسط یک ماشین پشته‌ای قطعی M پذیرفته شود به طوری که $L = L(M)$.

[1] deterministic context-free language

ماشین‌های پشته‌ای قطعی

ماشین های پشته ای همیشه سه حالت دارند؟؟؟

- زبان $L = \{a^n b^n : n \geq 0\}$ یک زبان مستقل از متن قطعی است.
- زیرا ماشین پشته‌ای قطعی $M = (\{q_0, q_1, q_2\}, a, b, 0, 1, \delta, q_0, 0, \{q_0\})$ با گذارهای زیر وجود دارد که آن را می‌پذیرد.

$$\delta(q_0, a, 0) = \{(q_1, 10)\}$$
$$\delta(q_1, a, 1) = \{(q_1, 11)\}$$
$$\delta(q_1, b, 1) = \{(q_2, \lambda)\}$$
$$\delta(q_2, b, 1) = \{(q_2, \lambda)\}$$
$$\delta(q_2, \lambda, 0) = \{(q_0, \lambda)\}$$

ماشین‌های پشته‌ای قطعی

- زبان $\{ww^R : w \in \Sigma^*\}$ یک زبان مستقل از متن است ولی یک زبان مستقل از متن قطعی نیست، زیرا هیچ ماشین مستقل از متن قطعی برای آن وجود ندارد.
- دلیل آن این است که برای تشخیص وسط رشته به عدم قطعیت نیاز داریم.
- زبان‌های مستقل از متن قطعی با گرامرهای مستقل از متن قطعی تولید می‌شوند و اهمیت این گرامرها در این است که تجزیه را در زمان چندجمله‌ای $O(n)$ به ازای جملات با طول n انجام می‌دهند.
- در فرایند تجزیه جملات با استفاده از گرامر مستقل از متن قطعی (همانند گرامر ساده)، همیشه برای یک اشتقاق تنها یک انتخاب وجود دارد.

لم تزریق برای زبان‌های مستقل از متن

- از لم تزریق برای زبان‌های مستقل از متن استفاده می‌کنیم برای اینکه نشان دهیم که یک زبان مستقل از متن نیست.
- نشان می‌دهیم که اگر یک زبان مستقل از متن باشد، آنگاه می‌توان هر رشتهٔ به اندازهٔ کافی طولانی از آن زبان را به پنج قسمت تقسیم کرد به طوری که از تکرار (پمپاژ) قسمت دوم و چهارم رشته‌ای به دست آید که در همان زبان است.

لم تزریق برای زبان‌های مستقل از متن

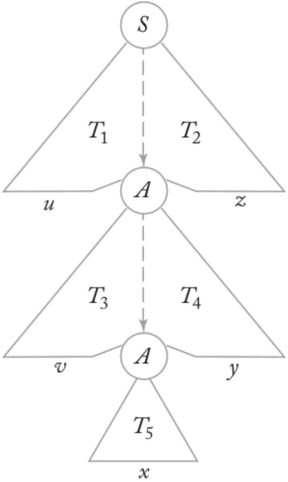
- فرض کنید L یک زبان نامحدود مستقل از متن باشد. آنگاه یک عدد صحیح مثبت m وجود دارد به طوری که هر جملهٔ $w \in L$ با طول $|w| \geq m$ می‌تواند به پنج قسمت تقسیم شود $w = uvxyz$ به طوری که $|vxy| \leq m$، $|vy| \geq ۱$ و همچنین $uv^ixy^iz \in L$ به ازای همهٔ مقادیر $i = ۰, ۱, ۲, \cdots$

لم تزریق برای زبان‌های مستقل از متن

- فرض کنید L یک زبان نامحدود مستقل از متن باشد. آنگاه یک عدد صحیح مثبت m وجود دارد به طوری که هر جملهٔ $w \in L$ با طول $|w| \geq m$ می‌تواند به پنج قسمت تقسیم شود $w = uvxyz$ به طوری که $|vxy| \leq m$، $|vy| \geq ۱$ و همچنین $uv^ixy^iz \in L$ به ازای همهٔ مقادیر $i = ۰, ۱, ۲, \cdots$
- از آنجایی که L نامحدود است، به ازای جملات طولانی، فرایندهای اشتقاق بسیار طولانی می‌تواند وجود داشته باشد که ارتفاع درخت اشتقاق آنها نیز بسیار زیاد است.
- حال یکی از این درخت‌های اشتقاق مرتفع را به همراه یک مسیر طولانی از ریشه تا یکی از برگ‌ها (به طوری که طول مسیر از تعداد متغیرهای گرامر بیشتر است) را در نظر بگیرید.
- از آنجایی که تعداد متغیرهای این گرامر محدود است، برخی از متغیرها باید در این مسیر تکرار شده باشند.

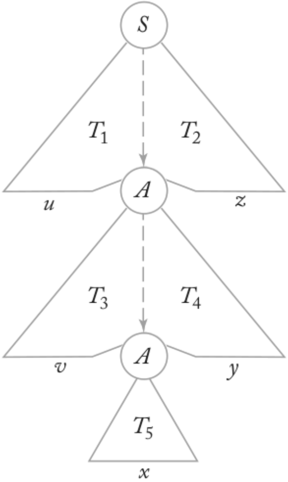
لم تزریق برای زبان‌های مستقل از متن

یک درخت اشتقاق مرتفع با تکرار متغیر A در زیر نشان داده شده است.



لم تزریق برای زبان‌های مستقل از متن

فرایند اشتقاق برای محصول این درخت بدین صورت است: $S \overset{*}{\Rightarrow} uAz \overset{*}{\Rightarrow} uvAyz \overset{*}{\Rightarrow} uvxyz$



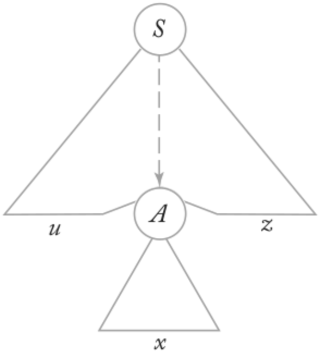
لم تزریق برای زبان‌های مستقل از متن

- پس فرایند اشتقاق برای جملهٔ uvxyz که محصول این درخت اشتقاق مرتفع است بدین صورت است:
$S \overset{*}{\Rightarrow} uAz \overset{*}{\Rightarrow} uvAyz \overset{*}{\Rightarrow} uvxyz$ به طوری که u،v،x،y،z شامل تنها نمادهای پایانی هستند.

- از آنجایی که دو اشتقاق $A \overset{*}{\Rightarrow} vAy$ و $A \overset{*}{\Rightarrow} x$ در فرایند اشتقاق این گرامر امکان‌پذیر هستند، لذا می‌توان در اولین بار مشاهدهٔ متغیر A در فرایند اشتقاق، به جای $A \overset{*}{\Rightarrow} vAy$ از $A \overset{*}{\Rightarrow} x$ استفاده کرد و جملهٔ uxy را به دست آورد. همچنین در هر بار مشاهدهٔ متغیر A می‌توان برای i بار از اشتقاق $A \overset{*}{\Rightarrow} vAy$ استفاده کرد و جملهٔ uv^ixy^iz را به دست آورد.

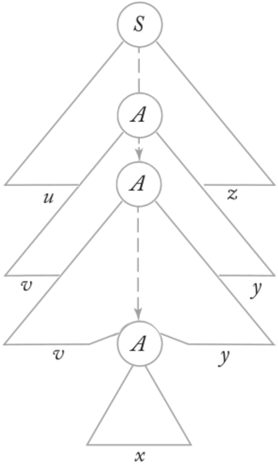
لم تزریق برای زبان‌های مستقل از متن

می‌توان در اولین بار مشاهدهٔ متغیر A در فرایند اشتقاق، به جای $A \overset{*}{\Rightarrow} vAy$ از $A \overset{*}{\Rightarrow} x$ استفاده کرد و جملهٔ uxy را به دست آورد.



لم تزریق برای زبان‌های مستقل از متن

همچنین در هر بار مشاهدهٔ متغیر A می‌توان برای i بار از اشتقاق $\mathrm{A} \overset{*}{\Rightarrow} \mathrm{vAy}$ استفاده کرد و جملهٔ $\mathrm{uv^ixy^iz}$ را به دست آورد.



لم تزریق برای زبان‌های مستقل از متن

- حال می‌خواهیم مقدار m را پیدا کنیم.
- باید تعیین کنیم برای رشته‌هایی با چه طولی حداقل یک متغیر در درخت اشتقاق تکرار می‌شود.
- در حالت کلی برای هر گرامر مستقل از متن داده شده، مقدار دقیق m به قوانین تولید گرامر بستگی دارد، اما می‌توانیم با استفاده از گرامر مستقل از متن در فرم نرمال چامسکی یک تقریب بالا برای مقدار m پیدا کنیم.
  از آنجایی که درخت اشتقاق در فرم نرمال چامسکی یک درخت دودویی است، استدلال بر روی این درخت ساده‌تر است.

لم تزریق برای زبان‌های مستقل از متن

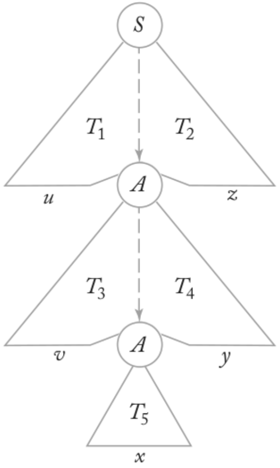
- فرض کنید گرامر G را به فرم نرمال چامسکی تبدیل می‌کنیم. در اینصورت درخت اشتقاق یک درخت دودویی است.

- اگر ارتفاع این درخت برابر باشد با تعداد متغیرهای گرامر، یعنی $|V|$، آنگاه حداقل یک مسیر از ریشه تا برگ با $|V| + ۱$ رأس وجود دارد، ولی از آنجایی که رأس آخر، یعنی برگ درخت، یک نماد پایانی است، بنابراین تعداد $|V|$ متغیر در یکی از مسیرها وجود دارد. طول جملات چنین درختی حداکثر برابر است با $۲^{|V|-۱}$ است (دقت کنید که در سطح آخر درخت یعنی جایی که متغیرها به نمادهای پایانی تبدیل می‌شوند، هر رأس تنها یک فرزند دارد).

- حال فرض کنید $m = ۲^{|V|}$. در اینصورت ارتفاع درخت اشتقاق برای جملاتی با طول برابر یا بیشتر از m باید حداقل $|V| + ۱$ باشد و بنابراین حداقل $|V| + ۲$ رأس در یکی از مسیرهای آن وجود دارد. آخرین رأس در این مسیر یک نماد پایانی است، پس حداقل $|V| + ۱$ متغیر در این مسیر وجود دارد و طبق اصل لانه کبوتری حداقل یکی از متغیرها در این مسیر تکرار شده است.

لم تزریق برای زبان‌های مستقل از متن

- از آنجایی که در این مسیر حداقل یک متغیر تکرار شده است، می توانیم رشته را طبق شکل زیر به پنج قسمت uvxyz تقسیم کنیم.



لم تزریق برای زبان‌های مستقل از متن

- همانطور که اشاره شد، از آنجایی که دو اشتقاق $A \overset{*}{\Rightarrow} vAy$ و $A \overset{*}{\Rightarrow} x$ در فرایند اشتقاق این گرامر امکان‌پذیر هستند:

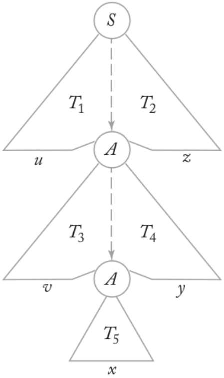
۱. می‌توان در اولین بار مشاهدهٔ متغیر A در فرایند اشتقاق، به جای $A \overset{*}{\Rightarrow} vAy$ از $A \overset{*}{\Rightarrow} x$ استفاده کرد و جملهٔ uxy را به دست آورد.

۲. همچنین در هر بار مشاهدهٔ متغیر A می‌توان برای i بار از اشتقاق $A \overset{*}{\Rightarrow} vAy$ استفاده کرد و جملهٔ uv^ixy^iz را به دست آورد.

لم تزریق برای زبان‌های مستقل از متن

- حال باید اطمینان حاصل کنیم که زیر رشته‌های v و y به طور همزمان نمی‌توانند تهی باشند.

- از آنجایی که G در فرم نرمال چامسکی است، لذا هیچ قانون تهی و یکه‌ای در آن وجود ندارد و بنابراین v و y نمی‌توانند به طور همزمان تهی باشند، بنابراین داریم ۱ $\geq |vy|$.



لم تزریق برای زبان‌های مستقل از متن

- حال فرض کنیم متغیر A پایین‌ترین متغیر تکرار شونده در مسیری است که در نظر گرفته‌ایم و در مسیر ریشه تا برگ‌ها هیچ متغیر تکرار شونده‌ای پایین‌تر از A وجود ندارد.
- به عبارت دیگر، می‌توانیم فرض کنیم که در زیردرخت T_5 هیچ متغیری تکرار نشده است. اگر متغیری در این زیر درخت تکرار شده بود می‌توانستیم آن متغیر را به جای متغیر A به عنوان متغیر تکرار شونده در نظر بگیریم.
- همچنین به طور مشابه می‌توانیم فرض کنیم که در زیر درخت‌های T_3 و T_4 هیچ متغیری تکرار نشده است.
- از آنجایی که در این زیردرخت‌ها هیچ متغیری تکرار نشده است، پس حداکثر طول رشتهٔ vxy را می‌توان با محاسبهٔ بلندترین طول مسیر ممکن از برگ‌ها تا دومین تکرار متغیر A محاسبه کرد. این مسیر حداکثر $|V| + 1$ متغیر دارد، و آخرین رأس در مسیر نماد پایانی است پس ارتفاع آن $|V| + 1$ است، پس طول vxy حداکثر برابر است با $m = 2^{|V|}$. بنابراین خواهیم داشت $|vxy| \leq m$.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{a^n b^n c^n : n \geq 0\}$ مستقل از متن نیست.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{a^n b^n c^n : n \geq ۱\}$ مستقل از متن نیست.
- فرض می‌کنیم زبان L مستقل از متن باشد، پس لم تزریق باید برای آن برقرار باشد.
- به ازای m داده شده، جملهٔ $a^m b^m c^m$ را در نظر می‌گیریم.
- اگر زیر رشتهٔ vxy به نحوی انتخاب شود که فقط شامل نمادهای a باشد، آنگاه با پمپاژ کردن رشته به هر مقداری، رشتهٔ $a^k b^m c^m$ با $k \neq m$ را به دست می‌آوریم که در L نیست.
- اگر زیررشتهٔ vxy به نحوی انتخاب شود که شامل تعدادی مساوی a و b باشد، آنگاه با پمپاژ کردن، رشتهٔ $a^k b^k c^m$ با $k \neq m$ به دست می‌آید که آن هم در زبان L نیست.
- بدین علت که طول $|vxy| \leq m$ پس نمی‌توانیم رشتهٔ vxy را طوری انتخاب کنیم که شامل نمادهای a و b و c شود.
- پس در هر صورت رشتهٔ به دست آمده بعد از پمپاژ در زبان L نیست و بنابراین فرض اولیه مبنی بر مستقل از متن بودن زبان نادرست بوده است و زبان L مستقل از متن نیست.

لم تزریق برای زبان‌های مستقل از متن

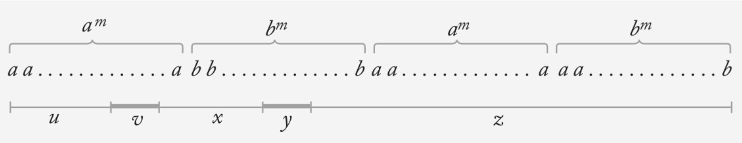
- حال فرض کنید می‌خواهیم با لم تزریق ثابت کنیم که زبان $L = \{a^n b^n\}$ مستقل از متن نیست (گرچه این زبان مستقل از متن است).
- فرض کنیم L مستقل از متن باشد، پس لم تزریق باید برای آن برقرار باشد. اگر رشته $a^m b^m$ را در نظر بگیریم زیر رشته vxy را می‌توان به نحوی انتخاب کرد که با پمپاژ کردن آن رشته‌ای به دست می‌آید که در زبان L است. می‌توان این زیر رشته را به طوری انتخاب کنیم که در آن تعداد a و b برابر باشد. در اینصورت همیشه با پمپاژ کردن آن رشته‌ای به دست می‌آید که در زبان L است. پس به تناقض نمی‌رسیم.
- گرچه به تناقض نمی‌رسیم ولی نمی‌توانیم نشان دهیم که L مستقل از متن است، زیرا لم تزریق یک شرط لازم است و کافی نیست. باید از روش دیگری استفاده کنیم تا نشان دهیم L مستقل از متن است، برای مثال یک گرامر مستقل از متن برای آن پیدا کنیم.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{ww : w \in \{a, b\}^*\}$ مستقل از متن نیست.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{ww : w \in \{a, b\}^*\}$ مستقل از متن نیست.
- فرض کنیم L مستقل از متن باشد، آنگاه باید لم تزریق برای آن برقرار باشد. به ازای m داده شده رشتهٔ $a^m b^m a^m b^m$ را در نظر می‌گیریم.
- زیر رشتهٔ vxy به هر نحوی که انتخاب شود، رشتهٔ به دست آمده با در نظر گرفتن $i = ۰$ در زبان L نیست.
- برای مثال اگر vxy را به صورت زیر انتخاب کنیم، با در نظر گرفتن $i = ۰$ داریم: $a^k b^j a^m b^m$ به طوری که $k < m, j < m$



- نشان دهید زبان $L = \{a^{n!} : n > ۲\}$ مستقل از متن نیست.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{a^{n!} : n > ۲\}$ مستقل از متن نیست.
- فرض می‌کنیم زبان L مستقل از متن باشد، پس لم تزریق باید برای آن برقرار باشد و می‌توان رشتهٔ $a^{m!}$ را به پنج قسمت $uvxyz$ تقسیم کرد.
- به ازای هر تقسیم بندی خواهیم داشت: $v = a^p$ و $y = a^q$
- در این صورت به ازای $i = ۰$ طول رشتهٔ uxz برابر است با $m! - (p + q)$
- این رشته در زبان L است تنها اگر $m! - (p + q) = j!$
- اما از آنجایی که $p + q \leq m$ بنابراین پس $m! - (p + q) > (m - ۱)!$ ، زیرا $m! - (m - ۱)! = (m - ۱)!(m - ۱) > m \geq p + q$.
- بنابراین فرض اولیه نادرست بوده و زبان L نمی‌تواند مستقل از متن باشد.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{a^n b^j : n = j^2\}$ مستقل از متن نیست.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{a^n b^j : n = j^۲\}$ مستقل از متن نیست.
- به ازای m داده شده، رشتهٔ $a^{m^۲} b^m$ را در نظر می‌گیریم.
- اگر زیررشتهٔ vxy با $|vy| = k$ به طور کامل در نمادهای a باشد، آنگاه بدیهی است که به ازای $i = ۰$ داریم $m^۲ - k < m^۲$ و اگر این زیررشته به طور کامل در نمادهای b باشد، به ازای $i = ۰$ داریم $m^۲ > (m - k)^۲$ و بنابراین رشتهٔ پمپاژ شده در زبان L نیست.
- حال اگر رشتهٔ v شامل تعداد $k_۱$ نماد a و رشتهٔ y شامل تعداد $k_۲$ نماد b باشد، آنگاه به ازای $i = ۰$ تعداد $m^۲ - k_۱$ نماد a و تعداد $m - k_۲$ نماد b به دست می‌آید. اما داریم:
  $(m - k_۲)^۲ \le (m - ۱)^۲ = m^۲ - ۲m + ۱ < m^۲ - k_۱$ زیرا می‌دانیم $k_۱ < m$ و $k_۲ \ge ۱$
- پس رشتهٔ پمپاژ شده در L نیست و L نمی‌تواند مستقل از متن باشد.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{a^i b^j c^k : k = \max(i, j)\}$ مستقل از متن نیست.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $\{a^ib^jc^k : k = \max(i, j)\}$ $L =$ مستقل از متن نیست.
- فرض می‌کنیم زبان L مستقل از متن است و به ازای m داده شده، رشتهٔ $a^mb^mc^m$ را در نظر می‌گیریم.
- چند حالت برای رشتهٔ vxy وجود دارد. یا این رشته حاوی نماد c است که در این صورت با در نظر گرفتن $i = 0$ تعداد a بیشتر از c می‌شود و رشته‌ای به دست می‌آید که در L نیست. یا رشتهٔ vxy حاوی نماد c نیست که در این صورت با در نظر گرفتن $i = 2$ رشته‌ای به دست می‌آید که تعداد نمادهای a یا نمادهای b یا تعداد هر دو نماد a و b در آن بیشتر از نماد c است و رشتهٔ به دست آمده در L نیست.
- پس فرض اولیه مبنی بر مستقل از متن بودن زبان نادرست بوده است.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{w \in \{a, b, c\}^* : n_a(w) < n_b(w) \wedge n_a(w) < n_c(w)\}$ مستقل از متن نیست.

لم تزریق برای زبان‌های مستقل از متن

- نشان دهید زبان $L = \{w \in \{a,b,c\}^* : n_a(w) < n_b(w) \wedge n_a(w) < n_c(w)\}$ مستقل از متن نیست.
- فرض می‌کنیم زبان L مستقل از متن است و به ازای m داده شده، رشتهٔ $a^m b^{m+1} c^{m+1}$ را در نظر می‌گیریم.
- چند حالت برای رشتهٔ vxy وجود دارد. یا این رشته حاوی نماد a است که در این صورت با در نظر گرفتن $i = 2$ تعداد a بیشتر از c می‌شود و رشته‌ای به دست می‌آید که در L نیست. یا رشتهٔ vxy حاوی نماد a نیست که در این صورت با در نظر گرفتن $i = 0$ رشته‌ای به دست می‌آید که تعداد نمادهای b یا نمادهای c یا تعداد هر دو نماد b و c در آن کمتر از نماد a است و رشتهٔ به دست آمده در L نیست.
- پس فرض اولیه مبنی بر مستقل از متن بودن زبان نادرست بوده است.

ویژگی‌های بستاری زبان‌های مستقل از متن

- مجموعهٔ زبان‌های مستقل از متن بر روی عملگرهای اجتماع، الحاق، و بستار–ستاره بسته، اما بر روی اشتراک و متمم بسته نیست.

ویژگی‌های بستاری زبان‌های مستقل از متن

- مجموعهٔ زبان‌های مستقل از متن بر روی عملگرهای اجتماع، الحاق، و بستار-ستاره بسته است.
- فرض کنید L_1 و L_2 دو زبان مستقل از متن باشند که به ترتیب با گرامرهای $G_1 = (V_1, T_1, S_1, P_1)$ و $G_2 = (V_2, T_2, S_2, P_2)$ تولید می‌شوند. فرض می‌کنیم V_1 و V_2 دو مجموعهٔ مجزا هستند.
- زبان $L(G_3)$ که توسط گرامر $G_3 = (V_1 \cup V_2 \cup \{S_3\}, T_1 \cup T_2, S_3, P_3)$ تولید می‌شود را در نظر بگیرید.
- فرض می‌کنیم S_3 عضو V_1 و V_2 نیست و قوانین تولید P_3 را به صورت $P_3 = P_1 \cup P_2 \cup \{S_3 \to S_1 | S_2\}$ در نظر می‌گیریم.

ویژگی‌های بستاری زبان‌های مستقل از متن

- می‌توانیم نشان دهیم که $L(G_3) = L_1 \cup L_2$.
- به ازای هر $w \in L_1$ می‌توانیم فرایند اشتقاق $S_3 \Rightarrow S_1 \overset{*}{\Rightarrow} w$ را بنویسیم. همین استدلال را در مورد $w \in L_2$ نیز به کار می‌بریم.
- همچنین به ازای هر $w \in L(G_3)$ اولین مرحله در فرایند اشتقاق $S_3 \Rightarrow S_1$ یا $S_3 \Rightarrow S_2$ است. بدین ترتیب w یا در L_1 است و یا در L_2 .
- پس گرامری پیدا کردیم برای اجتماع دو زبان مستقل از متن و بنابراین اجتماع دو زبان مستقل از متن یک زبان مستقل از متن است.

ویژگی‌های بستاری زبان‌های مستقل از متن

- با استدلالی مشابه، گرامر G_4 را با قوانین تولید $\{S_4 \rightarrow S_1S_2\}$ $P_4 = P_1 \cup P_2 \cup$ و نشان می‌دهیم $L(G_4) = L_1L_2$.
- پس گرامری پیدا کردیم برای الحاق دو زبان مستقل از متن و بنابراین الحاق دو زبان مستقل از متن یک زبان مستقل از متن است.
- در نهایت با استدلالی مشابه استدلال اجتماع، گرامر G_5 را با قوانین تولید $\{S_5 \rightarrow S_1S_5|\lambda\}$ $P_5 = P_1 \cup$ و نشان می‌دهیم $L(G_5) = L_1^*$.
- پس گرامری پیدا کردیم برای بستار-ستاره بر روی یک زبان مستقل از متن و بنابراین بستار-ستارهٔ یک زبان مستقل از متن، یک زبان مستقل از متن است.

ویژگی‌های بستاری زبان‌های مستقل از متن

- مجموعهٔ زبان‌های مستقل از متن بر روی اشتراک و متمم بسته نیست.
- اثبات: از برهان خلف و یک مثال نقض استفاده می‌کنیم، یعنی فرض می‌کنیم مجموعهٔ زبان‌های مستقل از متن بر روی اشتراک بسته باشد. آنگاه دو زبان مستقل از متن را انتخاب می‌کنیم و نشان می‌دهیم که اشتراک آنها زبانی است که مستقل از متن نیست. پس فرض اولیه نادرست بوده و اشتراک دو زبان مستقل از متن همیشه یک زبان مستقل از متن نیست.
- دو زبان $L_1 = \{a^n b^n c^m : n, m \geq 0\}$ و $L_2 = \{a^n b^m c^m : n, m \geq 0\}$ را در نظر می‌گیریم که هر دو مستقل از متن هستند زیرا یک گرامر مستقل از متن برای هر یک از آنها وجود دارد.
- می‌دانیم $L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\}$ که توسط لم تزریق نشان دادیم مستقل از متن نیست. پس اشتراک دو زبان مستقل از متن مستقل از متن نیست.

ویژگی‌های بستاری زبان‌های مستقل از متن

- برای اثبات بسته نبودن زبان‌های مستقل از متن بر روی متمم از قانون دمورگان استفاده می‌کنیم:
$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

- اگر زبان‌های مستقل از متن بر روی متمم بسته بودند، آنگاه سمت راست عبارت بالا همیشه یک زبان مستقل از متن به دست می‌داد. اما نشان دادیم که سمت چپ عبارت بالا، یعنی اشتراک دو زبان مستقل از متن می‌تواند مستقل از متن نباشد.

- پس مجموعهٔ زبان‌های مستقل از متن بر روی عملگر متمم بسته نیست.

ویژگی بستاری

- فرض کنید L_1 یک زبان مستقل از متن و L_2 یک زبان منظم باشد. آنگاه $L_1 \cap L_2$ یک زبان مستقل از متن است.

ویژگی بستاری

- فرض کنید L_1 یک زبان مستقل از متن و L_2 یک زبان منظم باشد. آنگاه $L_1 \cap L_2$ یک زبان مستقل از متن است.
- فرض کنید $M_1 = (Q, \Sigma, \Gamma, \delta_1, q_0, z, F_1)$ یک ماشین پشته‌ای غیرقطعی باشد که زبان L_1 را می‌پذیرد و $M_2 = (P, \Sigma, \delta_2, p_0, F_2)$ یک ماشین متناهی قطعی باشد که زبان L_2 را می‌پذیرد.
- یک ماشین پشته‌ای $\widehat{M} = (\widehat{Q}, \Sigma, \Gamma, \widehat{\delta}, \widehat{q_0}, z, \widehat{F})$ می‌سازیم که رشته‌هایی را می‌پذیرد که دو ماشین M_1 و M_2 می‌پذیرند.
- وقتی نمادی از ورودی خوانده می‌شود، ماشین $\widehat{M}$ حرکت‌های هر دو ماشین M_1 و M_2 را شبیه‌سازی می‌کند.

ویژگی بستاری

- برای این کار ماشین $\widehat{M}$ را به طوری طراحی می‌کنیم که $\widehat{Q} = Q \times P$ و $\widehat{q_0} = (q_0, p_0)$ و $\widehat{F} = F_1 \times F_2$.

- همچنین $\widehat{\delta}$ را طوری طراحی می‌کنیم که $((q_k, p_l), x) \in \widehat{\delta}((q_i, p_j), a, b)$ اگر و تنها اگر $(q_k, x) \in \delta_1(q_i, a, b)$ و همچنین $\delta_2(p_j, a) = p_l$.

- در صورتی که در ماشین پشته‌ای غیرقطعی گذار تهی وجود داشت و داشتیم $a = \lambda$، آنگاه قرار می‌دهیم $p_j = p_l$.

- به عبارت دیگر هر حالت در ماشین $\widehat{M}$ به نام (q_i, p_j) نمایندهٔ حالت‌های ماشین پشته‌ای M_1 و ماشین متناهی قطعی M_2 است و گذار برای آن حالت به ازای نمادهای الفبا برای هر دو ماشین محاسبه می‌شود.

ویژگی بستاری

- می‌توان نشان داد که $((q_0, p_0), w, z) \vdash^*_{\widehat{M}} ((q_r, p_s), \lambda, x)$ به ازای $q_r \in F_1$ و $p_s \in F_2$ اگر و تنها اگر $(q_0, w, z) \vdash^*_{\widehat{M}} (q_r, \lambda, x)$ و $\delta^*(p_0, w) = p_s$
- بنابراین یک رشته توسط ماشین $\widehat{M}$ پذیرفته می‌شود اگر و تنها اگر توسط ماشین M_1 و ماشین M_2 پذیرفته شود یا به عبارت دیگر آن رشته در $L(M_1) \cap L(M_2) = L_1 \cap L_2$ باشد.
- می‌گوییم زبان‌های مستقل از متن بر روی اشتراک زبان‌های منظم بسته‌اند.

ویژگی بستاری

- نشان دهید زبان $L = \{a^n b^n : n \geq ۰, n \neq ۱۰۰\}$ مستقل از متن است.

ویژگی بستاری

- نشان دهید زبان $L = \{a^n b^n : n \geq ۰, n \neq ۱۰۰\}$ مستقل از متن است.
- فرض کنیم $L_۱ = \{a^{۱۰۰} b^{۱۰۰}\}$. می‌دانیم $L_۱$ منظم است و بنابراین $\overline{L_۱}$ نیز منظم است.
- زبان $L_۲ = \{a^n b^n : n \geq ۰\}$ نیز مستقل از متن است زیرا یک گرامر مستقل از متن برای آن وجود دارد.
- بنابراین $L = L_۲ \cap \overline{L_۱}$ نیز طبق ویژگی بستاری، مستقل از متن است.

ویژگی بستاری

- نشان دهید زبان $L = \{w \in \{a, b, c\}^* : n_a(w) = n_b(w) = n_c(w)\}$ مستقل از متن نیست.

ویژگی بستاری

- نشان دهید زبان $L = \{w \in \{a,b,c\}^* : n_a(w) = n_b(w) = n_c(w)\}$ مستقل از متن نیست.
- فرض کنیم زبان L یک زبان مستقل از متن باشد. آنگاه زبان $L \cap L(a^*b^*c^*) = L_2 = \{a^n b^n c^n : n \geq 0\}$ نیز طبق ویژگی بستاری باید مستقل از متن باشد.
- اما با استفاده از لم تزریق برای زبان‌های مستقل از متن نشان دادیم که زبان L_2 مستقل از متن نیست.
- بنابراین فرض اولیه نادرست بوده و زبان L نمی‌تواند مستقل از متن باشد.