نظریهٔ زبان‌ها و خودکاره‌ها
دستور مستقل از متن

محسن هوشمند
دانشکده تکنولوژی اطلاعات و علم رایانه
دانشگاه تحصیلات تکمیلی علوم پایه زنجان

قضیه کلین(نحوه تشخیص منظم بودن یک زبان)

زبانی منظم است اگر آن را بتوان با خمم نشان داد.

زبانی منظم است اگر آن را بتوان با خمن نشان داد.

زبانی منظم است اگر آن را بتوان با عبارت منظم نشان داد.

زبانی منظم است اگر آن را بتوان با دستور منظم نشان داد.

تعریف دستور

دستور به صورت چهارتائی $G = (V,\Sigma,S,R)$ نمایش داده می‌شود که

V مجموعه متناهی از اشیاء به نام متغیرهاست.

Σ مجموعه متناهی الفباست (در کتاب لینز پایانه خوانده می‌شوند).

$S \in V$ متغیر آغاز است.

R مجموعه متناهی از قوانین تولید (یا قوانین جانشینی) است (در کتاب لینز با P نمایش داده می‌شود).

مجموعه‌های V و Σ ناتهی و جدا از هم هستند.

•

در حالت کلی تمامی قوانین دستور به صورت

$$x \to y$$

است که $x \in (V \cup \Sigma)^+$ و $y \in (V \cup \Sigma)^*$

اشتقاق: رشته قوانین جانشینی که منجر به رشته‌ای خواهد شد.

انواع دستور

۱- منظم: قوانینی به صورت $A \to xB|x$ یا $A \to Bx|x$ دارد.

۲- مستقل از متن: در سمت چپ تمامی قوانین آن، صرفا یک متغیر باشد.

۳- حساس به متن: قوانین آن $x \to y$ است که $x,y \in (V \cup \Sigma)^+$ و $|x| \leq |y|$.

۴- بدون محدودیت: هیچ محدویتی روی قوانین تولید اعمال نشود مگر اینکه نباید ε در سمت چپ قوانین باشد.

دستور مستقل از متن

Context free language

زبان متن‌آزاد

زبان بافتار آزاد

زبان مستقل‌ازمتن

$$a^n b^n, n \geq 0$$

مهم‌ترین موضوع در نظریه زبان‌ها

مورد استفاده در زبان‌های برنامه‌سازی

دستور مستقل از متن

دستور منظم

$$A \rightarrow Bx|x$$

یا

$$A \rightarrow xB|x$$

دستور قوی‌تر

- کاهش محدودیت و قیود

دستور مستقل از متن

تعریف- دستور $G = (V,\Sigma,S,R)$ مستقل از متن است اگر قوانین آن به صورت $A \to x$ باشد،
به‌طوری‌که $A \in V$ و $x \in (V \cup \Sigma)^*$

زبان مستقل از متن است اگر و فقط اگر دستور مستقل از متن G موجود باشد به طوری که L=L(G)

هر دستور منظم، دستور مستقل از متن است.

مثال

$S \rightarrow aSb|ab$

$S \Longrightarrow ab$

$S \Longrightarrow aSb \Longrightarrow aaSbb \Longrightarrow aaabbb$

$L = \{a^n b^n, n \geq 1\}$

مثال

$a^n b^n; n \geq 0?$

$S \rightarrow aSb|\epsilon$

مثال

$\{a^n b^{n+1}; n \geq 0\}$?

1- $S \rightarrow aSb|b$

2- $a^n b^n b$

$S \rightarrow Ab$

$A \rightarrow aAb|\epsilon$

مثال

$\{a^{n+1}b^n; n \geq 0\}$?

1- $S \to aSb|a$

2- aa^nb^n

$S \to aA$

$A \to aAb|\epsilon$

مثال

$\{a^n b^n a^k b^k; n, k \geq 0\}$?

$S \rightarrow AB$

$A \rightarrow aAb|\epsilon$

$B \rightarrow aBb|\epsilon$

مثال

$a^n b^m; n \neq m$

دو حالت

n>m: $a^{n-m} a^m b^m$

$S \rightarrow AS_1$

$A \rightarrow aA|a$

$S_1 \rightarrow aS_1 b|\epsilon$

n<m:

$S \rightarrow S_1 B$

$B \rightarrow bB|b$

مثال

$\{a^n b^{2n}; n \geq 0\}$?

$S \rightarrow aSbb|\epsilon$

مثال

$\{a^n b^n; n \geq 0\} \cup \{b^n a^n; n \geq 0\}$?

$S \to S_1 | S_2$

$S_1 \to a S_1 b | \epsilon$

$S_2 \to b S_2 a | \epsilon$

مثال

$\{a^n b^m c^k; m = n + k; n, k \geq 0\} = \{a^n b^n b^k c^k\}$

$S \to AB$

$A \to aAb|\epsilon$

$B \to bBc|\epsilon$

مثال

$\{a^n b^m c^k; k > n + m; n, m \geq 0\} = \{a^n b^m c^l c^m c^k\}$

$S \rightarrow aSc|A$

$A \rightarrow bAc|cY$

$Y \rightarrow cY|\epsilon$

مثال

$\{a^n b^m c^k; k = n.m; n, m \geq 0\}$?

مثال

$\{aa(bc)^n be(dde)^n; n \geq 0\}$?

$S \rightarrow aaA$

$A \rightarrow bBe$

$B \rightarrow cAdd|\epsilon$

$S \rightarrow aaA$
$A \rightarrow bcAdde|be$

مثال

دستور مولد $\{ww^R : w \in \{a,b\}^*\}$

$S \to aSa|bSb|\epsilon$

$S \to aA|bB|\epsilon$
$A \to Sa$
$B \to Sb$

مثال

دستور مولد $\{w: \#_a(w) = \#_b(w)\}$

$S \rightarrow SS|aSb|bSa|\epsilon$

مثال

دستور روبرو چه زبانی تولید می‌کند

$S \rightarrow AB|\epsilon$

$A \rightarrow 1A|\epsilon$

$B \rightarrow 0B|\epsilon$

دستور خطی و غیرخطی

دستور مام که در سمت راست قواعد تولید آن، حداکثر یک متغیر موجود باشد.

دستور مام $\{a^n b^n c^m;\ n \geq 0, m \geq 0\}$ خطی است؟

$S \rightarrow Sc|aAb|\epsilon$

$A \rightarrow aAb|\epsilon$

دستورهای خطی، مام هستند

- دستور مام $A \rightarrow x$ به‌طوری‌که $A \in V$ و $\mathrm{x} \in (V \cup \Sigma)^*$
- دستور خطی $\mathrm{x} \in \Sigma^* V \Sigma^*$

اشتقاق چپ و راست

دستور مستقل از متنی که خطی نباشد (در سمت راست بیش از یک متغیر باشد).

جانشینی متغیرها در این وضعیت اختیاری است.

$1 - \ S \rightarrow AB$

$2 - \ A \rightarrow aaA|\epsilon$

$3 - \ B \rightarrow Bb|\epsilon$

$S \Longrightarrow AB \Longrightarrow aaAB \Longrightarrow aaB \Longrightarrow aab$

$S \Longrightarrow AB \Longrightarrow ABb \Longrightarrow Ab \Longrightarrow aaAb \Longrightarrow aab$

بهتر است ترتیب خاصی در نظر بگیریم.

تعریف

- اشتقاق چپ: اگر در هر مرحله سمت چپ‌ترین متغیر جانشین بپذیرد
- اشتقاق راست: اگر در هر مرحله سمت راست‌ترین متغیر جانشین بپذیرد

درخت اشتقاق

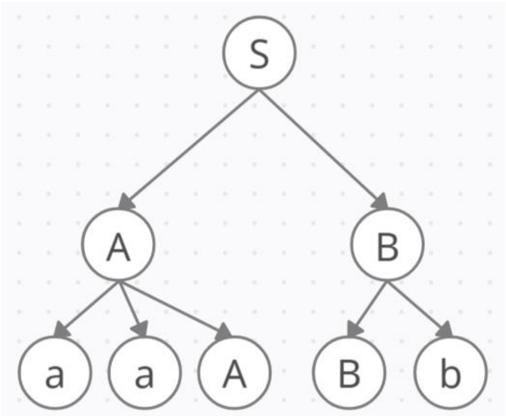
روش دیگر نمایش اشتقاق

$x \to y$ هر رأسْ سمت چپ قانون تولیدی است و فرزندان آن رأسْ سمت راست آن قانون هستند.

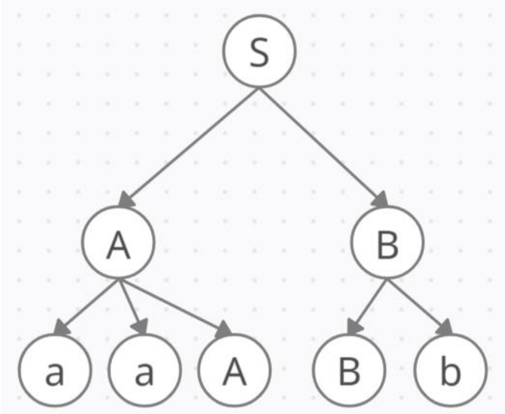
$1 - \ S \to AB$

$2 - \ A \to aaA|\epsilon$

$3 - \ B \to Bb|\epsilon$



درخت اشتقاق



دستور $G = (V,\Sigma,S,R)$ مستقل از متن است. درخت اشتقاق درخت مرتبی است با ویژگی‌های

۱- S ریشه است

۲- هر برگ دارای مقداری از $\{\epsilon\}$ U Σ است.

۳- رأس‌های غیربرگ مقداری از V دارند

۴- اگر رأس $A \in V$ و فرزندان آن از چپ به راست $\{a_1,a_2, \dots ,a_n\}$ باشند، آن‌گاه R دارای قانون $A \to a_1 a_2, \dots a_n$ است.

۵- رأس ϵ هم‌زاد ندارد.

درخت اشتقاق جزئی

دارای ویژگی‌های ۳ و ۴ و ۵ اما ویژگی ۱ لزوما برای آن صدق نکند و ویژگی ۲ به صورت زیر

۲- هر برگ دارای مقداری از $V \cup \Sigma \cup \{\epsilon\}$ است.

$1 - \ S \rightarrow aAB$

$2 - \ A \rightarrow bBb$

$3 - \ B \rightarrow A|\epsilon$

قضیه

دستور $G = (V,\Sigma,S,R)$ مستقل از متن است. آن‌گاه

۱- هر $w \in L(G)$ درخت اشتقاق دارد

۲- محصول هر درخت اشتقاق در $L(G)$ است.

۳- اگر درختی اشتقاق جزیی G با ریشهٔ S باشد، آن‌گاه جمله‌ای از G است.

قضیه

$A \rightarrow v$ و $|v| = k > 1$ آن‌گاه درخت اشتقاق $w \in L(G)$ ارتفاع h دارد به طوری که

$$\log_k |w| \le h \le \frac{|w| - 1}{k - 1}$$

ابهام و تجزیه

Ambiguity and parsing

وقتی بخواهیم بدانیم رشته‌ای را L(G) می‌پذیرد یا نه.

الگوریتم عضویت: بررسی می‌کند رشته‌ای در L(G) است یا نه.

تجزیه: یافتن دنباله‌ای از قوانین که w از آن‌ها مشتق شده است.

تجزیه و عضویت به طور مرتب و مرحله به مرحله تمامی اشتقاق‌های ممکن (چپ) را بدست آوریم و بررسی شود که w با یکی از آن‌ها منطبق است یا خیر (با شروع از ریشه S و افزایش طول)

ابهام و تجزیه

$$S \rightarrow SS|aSb|bSa|\epsilon$$

$$w = aabb$$

دور اول

$$S \Longrightarrow SS$$
$$S \Longrightarrow aSb$$
$$S \Longrightarrow bSa$$
$$S \Longrightarrow \epsilon$$

دور دوم

$$S \Longrightarrow SS \Longrightarrow SSS$$
$$S \Longrightarrow SS \Longrightarrow aSbS$$
$$S \Longrightarrow SS \Longrightarrow bSaS$$
$$S \Longrightarrow SS \Longrightarrow S$$

$$S \Longrightarrow aSb \Longrightarrow aSSb$$
$$S \Longrightarrow aSb \Longrightarrow aaSbb$$
$$S \Longrightarrow aSb \Longrightarrow abSab$$
$$S \Longrightarrow aSb \Longrightarrow ab$$

دور بعد

$$S \Longrightarrow aSb \Longrightarrow aaSbb \Longrightarrow aabb$$

ابهام و تجزیه

مشکلات

طولانی و اتلاف هزینه (ناکارا) اگر w عضو L(G) نباشد ممکن است متوقف نشود

حل: دستکاری قوانین تهی $A \to \epsilon$

اگر شکل دو قانون $A \to \epsilon$ و $A \to B$ را نداشته باشیم، مشکلات کمتری داریم.

ابهام و تجزیه

$S \rightarrow SS|aSb|bSa|\epsilon$

ختم تجزیهٔ جستجو کامل پس از حداکثر $|w|$ دور

- یا رسیدن به رشته یا $w \notin L(G)$

قضیه

دستور $G = (V,\Sigma,S,R)$ مستقل از متن است و دو قانون $A \to \epsilon$ و $A \to B$ را ندارد، آن‌گاه روش تجزیه جستجو کامل را می‌توان به گونه‌ای نوشت که به ازای هر $w \in \Sigma^*$، تجزیه یا w را تولید می‌کند و یا می‌گوید تجزیه ممکن نیست و $w \notin L(G)$.

اثبات

در هر قدم در اشتقاق طول و پایانه‌های هر جمله یک واحد افزایش پیدا می‌کند. نه طول و نه تعداد نشانه‌های الفبا نمی‌توانند بیش از طول w باشند. پس اشتقاق نمی‌تواند بیشتر از $2|w|$ دور باشد، یا در این تعداد دورها تجزیه w را می‌یابد یا $w \notin L(G)$.

←نظری و نه کاربردی: تولید بسیار زیادی از جمله

دستور ساده

S-Grammar

دستور $G = (V,\Sigma,S,R)$ مستقل از متن با قواعد $A \rightarrow ax$ و $a \in \Sigma$ و $A \in \mathrm{V}$ و $x \in V^*$

و (A,a) حداکثر یک بار در R ظاهر شود

مثال $S \rightarrow aS|bSS|c$

مثال $S \rightarrow aS|bSS|aSS|c$ ✖

دستور ساده

اگر دستور ساده

- هر رشته با O(|w|) تجزیه‌پذیر
- $w = a_1 a_2 \dots a_n$
  - به دلیل وجود یک قاعده که S در سمت چپ و شروع با a_1 در سمت راست
    - $S \Longrightarrow a_1 A_1 \dots A_m$
  - جانشینی A_1
    - صرفا یک انتخاب
      - $S \Longrightarrow a_1 a_2 B_1 \dots B_k \dots A_m$
    - پس تولید یک حرف در هر مرحله
      - کل مراحل پردازش برابر $|w|$

ابهام

چند درخت اشتقاق w را تولید می‌کند.

تعریف

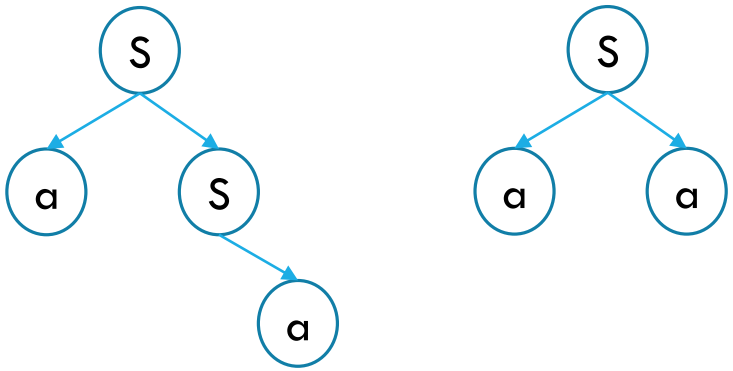
دستور مام مبهم است اگر رشته‌ای دارای حداقل دو درخت اشتقاق متفاوت باشد.

به طور ضمنی به معنای وجود دو یا چند اشتقاق چپ و یا اشتقاق راست.

ابهام

$S \to aS|aa|a$

رشتهٔ aa

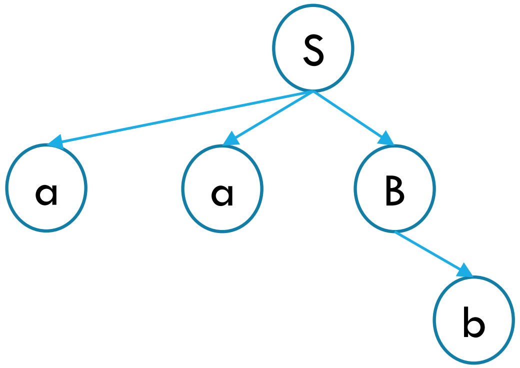


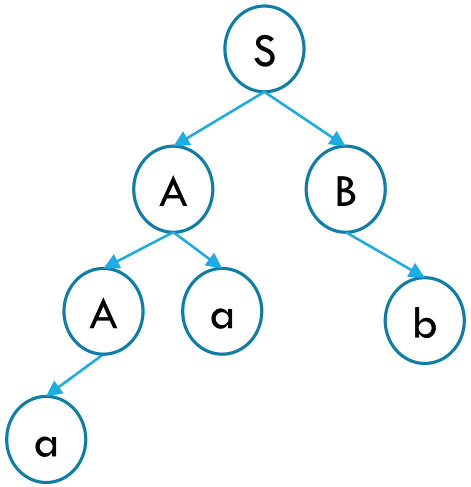
ابهام

$S \rightarrow AB|aaB$

$A \rightarrow a|Aa$

$B \rightarrow b$





$S \Longrightarrow aaB \Longrightarrow aab$

$S \Longrightarrow AB \Longrightarrow AaB \Longrightarrow aaB \Longrightarrow aab$

ابهام

دستور منظم چطور؟

$$S \rightarrow aB|A$$

$$A \rightarrow \epsilon|aA$$

$$B \rightarrow bB|a$$

رشتهٔ aa

$$S \Longrightarrow aB \Longrightarrow aa$$

$$S \Longrightarrow A \Longrightarrow aA \Longrightarrow aaA \Longrightarrow aa$$

پس دستور منظم نیز می‌تواند مبهم باشد.

ولی ذاتا مبهم نیست.

دستورهای مبهم

امکان اشتقاق جمله‌ای با دو درخت تجزیهٔ متفاوت

مثال- جملهٔ id := id +id + id

$S \rightarrow id \ := E$

$E \rightarrow id$

$E \rightarrow num$

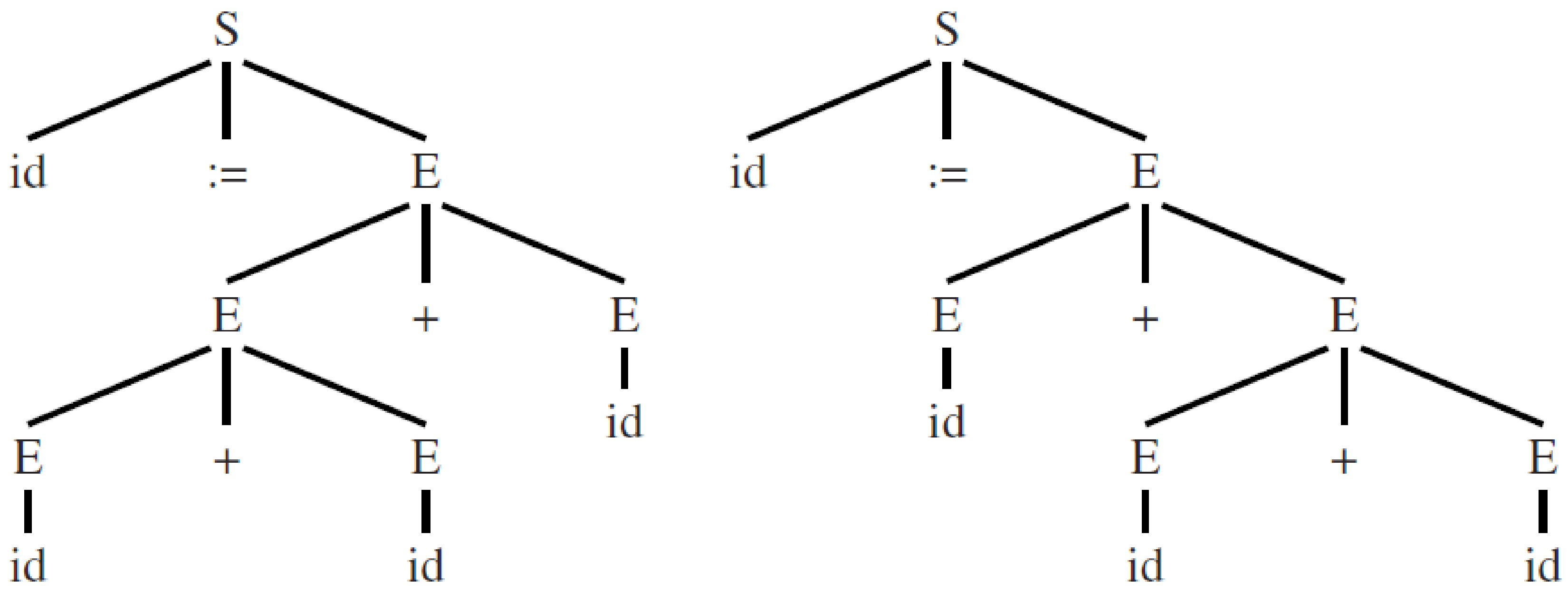
$E \rightarrow E * E$

$E \rightarrow E \,/\, E$

$E \rightarrow E + E$

$E \rightarrow E - E$

$E \rightarrow (E)$



دستورهای مبهم – مثال ۲

دستوری مبهم با دو درخت تجزیهٔ متفاوت برای جملهٔ ۱-۲-۳

۱ – ۲ – ۳

$E \rightarrow id$

$E \rightarrow num$

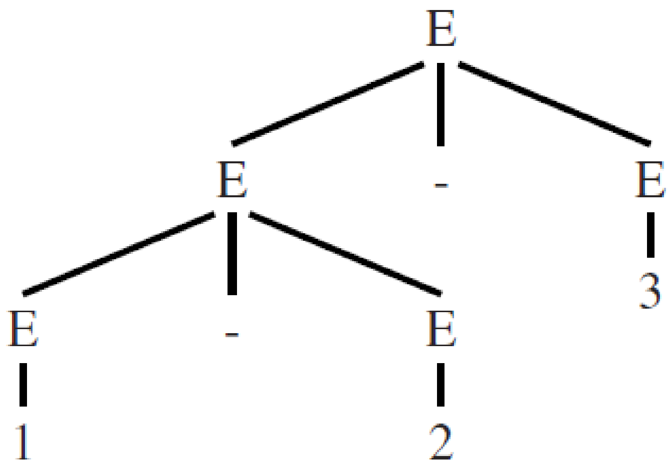
$E \rightarrow E * E$

$E \rightarrow E / E$

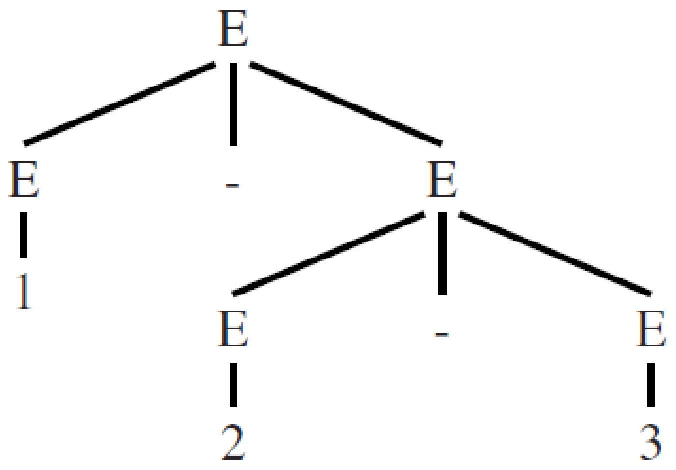
$E \rightarrow E + E$

$E \rightarrow E - E$

$E \rightarrow (E)$



(۱ – ۲) – ۳ = –۴



۱ – (۲ – ۳) = ۲

دستورهای مبهم – مثال ۳

دستوری مبهم با دو درخت تجزیهٔ متفاوت برای جملهٔ

۱ + ۲ * ۳

$$E \rightarrow id$$

$$E \rightarrow num$$

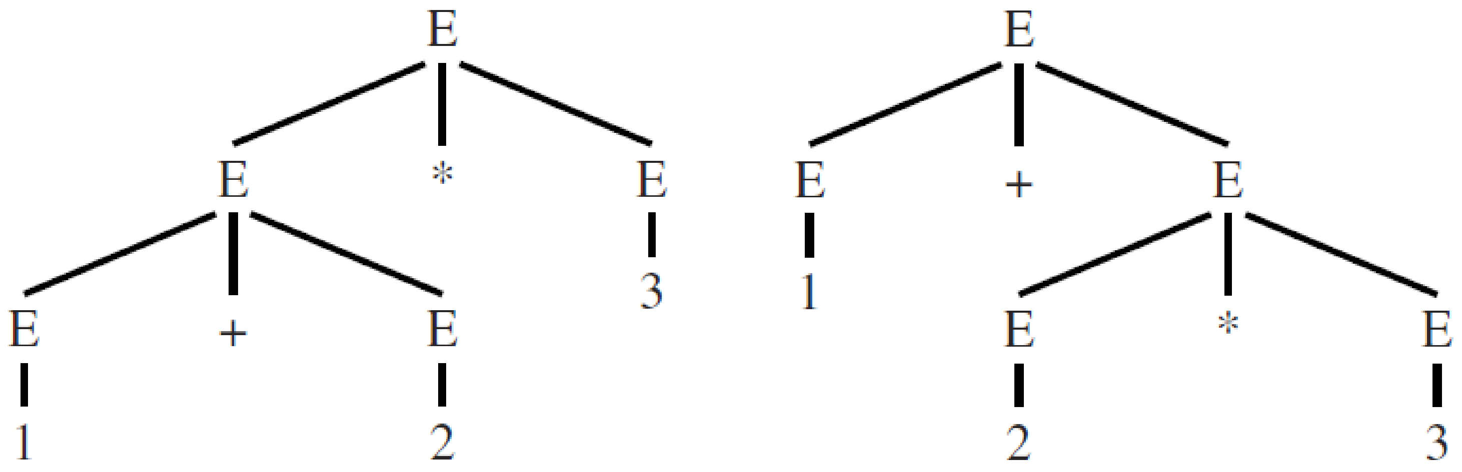
$$E \rightarrow E * E$$

$$E \rightarrow E \,/\, E$$

$$E \rightarrow E + E$$

$$E \rightarrow E - E$$

$$E \rightarrow (E)$$



(۱ + ۲) * ۳ = ۹

۱ + (۲ * ۳) = ۷

ابهام

راه حل

- اعطای اولویت
  - خارج حیطهٔ دستور
- پس دستور بهتر است طوری نوشته شود که هر رشته یک اشتقاق داشته باشد

تعریف زبان ذاتا مبهم

- اگر زبانی مام باشد و هر دستور مولد آن مبهم باشد.

$$S \rightarrow aB|A$$

$$A \rightarrow \epsilon|Aa$$

$$B \rightarrow bB|a$$

مثال

زبان ذاتا مبهم است $\{a^n b^m c^k; m = n$یا$m = k\}$?

$S \rightarrow S_1 | S_2 aB | A$

$S_1 \rightarrow S_1 c | A$

$A \rightarrow \epsilon | aAb$

$S_1 \rightarrow aS_2 | B$

$B \rightarrow bB | a$

رشتهٔ $a^2 b^2 c^2$

$S \Longrightarrow S_1 \Longrightarrow S_1 c \Longrightarrow S_1 cc \Longrightarrow Acc \Longrightarrow aAbcc \Longrightarrow aaAbbcc \Longrightarrow aabbcc$

$S \Longrightarrow S_2 \Longrightarrow aS_2 \Longrightarrow aaS_2 \Longrightarrow aaB \Longrightarrow aabBc \Longrightarrow aabbBcc \Longrightarrow aabbcc$

رفع ابهام دستور

$$E \rightarrow id$$
$$E \rightarrow num$$
$$E \rightarrow E * E$$
$$E \rightarrow E / E$$
$$E \rightarrow E + E$$
$$E \rightarrow E - E$$
$$E \rightarrow (E)$$

$$E \rightarrow E + T$$
$$E \rightarrow E - T$$
$$E \rightarrow T$$
$$T \rightarrow T * F$$
$$T \rightarrow T / F$$
$$T \rightarrow F$$
$$F \rightarrow id$$
$$F \rightarrow num$$
$$F \rightarrow (E)$$

تغییرات

- جهت
  - اعمال اولویت‌ها
  - اعمال شرکت‌پذیری چپ عملگرها
- شامل
  - افزودن متغیرهای T و F
    - E مخفف Expression
    - T مخفف Term
    - F مخفف Factor
  - قرارداد

T-ها نمایشگر ضرب‌پذیرها (تقسیم)

F-ها نمایشگر جمع‌پذیرها (تفریق)

رفع ابهام دستور - ادامه

دو دستور متناظر
- تولید یکسانی از جملات

دستور جدید بر عکس دستور قبلی
- هر جمله دارای دقیقا «یک» درخت تجزیه

$$
\begin{aligned}
E &\rightarrow id \\
E &\rightarrow num \\
E &\rightarrow E * E \\
E &\rightarrow E \,/\, E \\
E &\rightarrow E + E \\
E &\rightarrow E - E \\
E &\rightarrow (E)
\end{aligned}
\quad \Longrightarrow \quad
\begin{aligned}
E &\rightarrow E + T \\
E &\rightarrow E - T \\
E &\rightarrow T \\
T &\rightarrow T * F \\
T &\rightarrow T \,/\, F \\
T &\rightarrow F \\
F &\rightarrow id \\
F &\rightarrow num \\
F &\rightarrow (E)
\end{aligned}
$$

رفع ابهام دستور - ادامه

$E \rightarrow E + T$

$E \rightarrow E - T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow T / F$

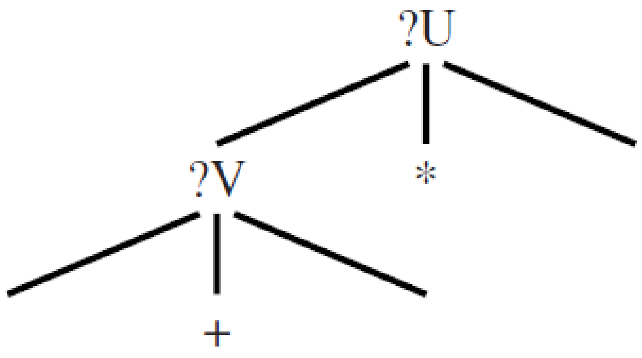
$T \rightarrow F$

$F \rightarrow id$

$F \rightarrow num$

$F \rightarrow (E)$

عدم‌امکان تولید درخت تجزیهٔ زیر از دستور مقابل



دو عملگر با اولویت متفاوت

- عملگر با اولویت بالاتر در سطح پائین‌ترِ درخت

به چه معنی؟

- «اعمال و کد کردن قانون» در «ساختار»

رفع ابهام دستور - ادامه

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

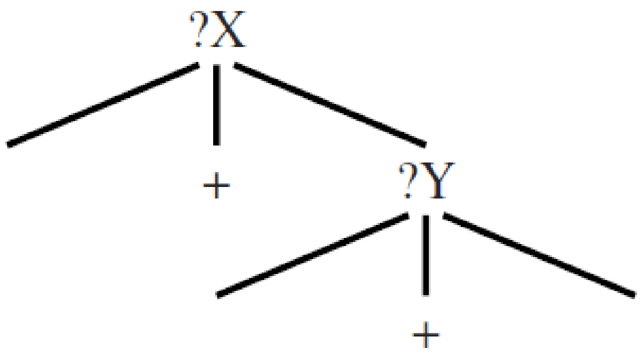
$$T \rightarrow F$$

$$F \rightarrow id$$

$$F \rightarrow num$$

$$F \rightarrow (E)$$

عدم‌امکان تولید درخت تجزیهٔ زیر از دستور مقابل



دو عملگر با اولویت یکسان

- شرکت‌پذیری چپ عملگرهای با اولویت یکسان ذخیره شده در در سطح پائین‌ترِ درخت

به چه معنی؟

- باز هم «اعمال و کد کردن قانون» در «ساختار»

در صورت ایجاد شرکت‌پذیری راست

- تغییر به $T \rightarrow F * T$

منابع

[سیپسر]

[لینز]

[ببرسبز]

[اژدرها]

“