

کامپیایر  
تجزیه گر

نحو، دستور مستقل از متن، اشتقاق، ابهام و رفع آن

محسن هوشمند

دانشکده تکنولوژی اطلاعات و علم رایانه

دانشگاه تحصیلات تکمیلی علوم پایه زنجان

**Syn.tax:** the way in which words are put together to form phrases, clauses, or sentences. *Webster's Dictionary*

**نحو:** (زبان‌شناسی) مجموعه‌ی قواعدی که درباره‌ی آرایش کلمات برای ساخت جمله یا عبارت بحث می‌کند.

فرهنگ فارسی عمید

# تجزیه گر

ورودی: رشته کلمات دسته‌بندی شده با پویشگر  
تایید جمله معتبر از رشته ورودی  
به وسیله اشتقاق مطابق دستور زبان برنامه‌نویسی

رشته حروف زبان سطح بالا

lexical analysis/scanner  
تحلیل لغوی

رشته توکن

syntax analysis/parser  
تحلیل نحو

درخت نحو

semantic analysis/ elaboration  
تحلیل معنا

درخت نحو

intermediate code generator  
مولد کد میانی

نمایش میانی

machine independent code optimizer  
بهینه‌ساز کد میانی

نمایش میانی

code generator  
مولد کد

کد ماشین مقصد

machine dependent code optimizer  
بهینه‌ساز کد وابسته به ماشین

کد ماشین مقصد

# مقدمه

مرحله دوم در پیشا

کار با نتیجه حاصل از پویشگر

هر واژه با دسته نحوی

اشتقاق ساختار نحوی برنامه

اهمیت سرعت

▪ متناسب با اندازه برنامه

▪ در توازن با عملکرد

# مقدمه

بیشتر مبتنی بر ابزارهای خودکار تولید تجزیه‌گر

کار اصلی: اعتبار نحوی برنامه ورودی

نیاز

- سازوکار صوری جهت مشخص‌سازی جمله معتبر
- روش سیستماتیک معین کردن عضویت در زبان

دستور مستقل از متن

# مقدمه

## تجزیه‌گر

با داشتن رشته‌ای از کلمات و دستوری

سعی بر اثبات سازنده اشتقاق رشته کلمات از دستور

انواع الگوریتم‌ها جهت تعیین عضویت در مستقل از متن

- تجزیه بالابه‌پائین

- نزول بازگشتی

- چچ(۱)

- تجزیه پائین به بالا

- دسته چر(۱)

# عبارت منظم

$[a - z] ([a - z] | [0 - 9])^* ([+-\times\div] [a - z] ([a - z] | [0 - 9])^*)^*$

# چرا عبارت منظم مناسب نیست؟

عدم توانایی نمایش اولویت و تقدم

استفاده از پرانتز جهت نمایش اولویت

عبارت منظم ناتوان در نمایش پرانتز متقارن [؟]

- هر ساختار جفتی چون آغاز و پایان

- {}

- Begin ... end

- Then else

ناتوان در شمارش

- به دلیل داشتن مجموعه متناهی از حالت‌ها

خمم

- مناسب برای ریزنحو ولی نامناسب برای نحو

- ناتوان در شمارش

# بیان نحو

1.  $expr \rightarrow term \mid expr \text{ add\_op } term$
2.  $term \rightarrow factor \mid term \text{ mult\_op } factor$
3.  $factor \rightarrow id \mid number \mid - factor \mid ( expr )$
4.  $add\_op \rightarrow + \mid -$
5.  $mult\_op \rightarrow * \mid /$

# دستورهای مستقل از متن (مام)

اجازه به تعریف انطباق و درستی پرانتزگذاری

تعریف بازگشتی

شامل

- متغیرها، یا غیرپایانه‌ها
- پایانه‌ها یا الفباء
- الفباء دستور مستقل از متن همان تکه‌ها هستند

نمایش گر چگونگی تولید رشته‌های خوش-نحو از الفباء

قوانین تولید

*SheepNoise* → *SheepNoise* baa  
| baa

# دستورهای مستقل از متن (مام)

چهارتائی  $(V, \Sigma, S, R)$

$V$  - مجموعه متناهی متغیرها

$\Sigma$  - مجموعه متناهی الفباء (در اینجا تکه‌ها)

$S$  - عضوی از متغیرها و متغیر شروع

$R$  - قواعد تولید  $V \rightarrow (V \cup \Sigma)^*$

*symbol*  $\rightarrow$  *symbol symbol*  $\dots$  *symbol*

علامت در سمت چپ صرفاً تک متغیر

علامت در سمت راست: ترکیبی از متغیرها و حروف الفباء

# دستورهای مستقل از متن (مام)

چهارتائی  $(V, \Sigma, S, R)$

$V$  - مجموعه متناهی متغیرها

$\Sigma$  - مجموعه متناهی الفباء (در اینجا تکه‌ها)

$S$  - عضوی از متغیرها و متغیر شروع

$R$  - قواعد تولید  $V \rightarrow (V \cup \Sigma)^*$

$Start \rightarrow \begin{array}{l} Paren \\ | \\ Bracket \end{array}$

$Paren \rightarrow \begin{array}{l} \underline{( Bracket )} \\ | \\ \underline{( )} \end{array}$

$Bracket \rightarrow \begin{array}{l} \underline{[ Paren ]} \\ | \\ \underline{[ ]} \end{array}$

## مثال -

- 1  $S \rightarrow S ; S$
- 2  $S \rightarrow \text{id} := E$
- 3  $S \rightarrow \text{print} ( L )$

- 4  $E \rightarrow \text{id}$
- 5  $E \rightarrow \text{num}$
- 6  $E \rightarrow E + E$
- 7  $E \rightarrow ( S , E )$

- 8  $L \rightarrow E$
- 9  $L \rightarrow L , E$

$$V = \{S E L\}$$

$$\Sigma = \{\text{id print num} , + ( ) := ;\}$$

$S$

$R?$

## مثال - ادامه

1  $S \rightarrow S ; S$   
2  $S \rightarrow \text{id} := E$   
3  $S \rightarrow \text{print} ( L )$

4  $E \rightarrow \text{id}$   
5  $E \rightarrow \text{num}$   
6  $E \rightarrow E + E$   
7  $E \rightarrow ( S , E )$

8  $L \rightarrow E$   
9  $L \rightarrow L , E$

جمله‌ای از دستور بالا

$\text{id} := \text{num}; \text{id} := \text{id} + (\text{id} := \text{num} + \text{num}, \text{id})$

از متن مبدا مثل

$a := 7;$

$b := c + (d := 5 + 6, d)$

نوع-تکه‌ها:  $\text{id}, \text{num}, :=$

نام‌ها  $a, b, c, d$  و اعداد  $7, 5, 6$

▪ «مقادیر معنایی» متناظر با تکه‌ها

# اشتقاق

جهت نمایش درستی جمله‌ای مبتنی بر دستوری

▪ اجرای «اشتقاق»

▪ بیان دیگر، نمایش گر چگونگی تولید رشته‌های خوش-نحو از تکه‌ها

شروع از «متغیر آغاز»

جانشینی هر «متغیر» واقع در «سمت چپ قاعده تولیدی» با «سمت راست قاعده تولید» مذکور

▪ تکرار این کار

▪ روش‌مند

انواع اشتقاق «یک» جمله

▪ اشتقاق چپ‌ترین

▪ اشتقاق راست‌ترین

# اشتقاق - ادامه

بیان دیگر

- انتخاب قانونی با متغیر شروع
- جاگذاری آن با مقدار سمت چپ قانون
- ادامه جاگذاری‌ها تا رسیدن به جمله موردنظر

اشتقاق چپ‌ترین (چپ)

- همیشه بسط چپ‌ترین متغیر در سمت راست قاعده تولید

اشتقاق راست‌ترین (راست)

- همیشه بسط راست‌ترین متغیر در سمت راست قاعده تولید

# اشتقاق - مثال ۱

S

S ; S

S ; id := E

id := E ; id := E

id := num ; id := E

id := num ; id := E + E

id := num ; id := E + (S , E )

id := num ; id := id + (S , E )

id := num ; id := id + (id := E , E )

id := num ; id := id + (id := E + E , E )

id := num ; id := id + (id := E + E , id )

id := num ; id := id + (id := num + E , id )

id := num ; id := id + (id := num + num , id )

چه نوع اشتقاقی؟  
▪ هیچکدام!

# اشتقاق - مثال ۱ - ادامه

اشتقاق چپ‌ترین

$S$

$S$  ;  $S$

id :=  $E$  ;  $S$

id := num ;  $S$

id := num ; id :=  $E$

id := num ; id :=  $E$  +  $E$

⋮

## اشتقاق - مثال ۲

اشتقاق راست

1.	$Expr$	$\rightarrow$	$Expr Op Number$
2.		$ $	$Number$
3.	$Op$	$\rightarrow$	$+$
4.		$ $	$-$
5.		$ $	$\times$
6.		$ $	$\div$

$Number + Number.$

<i>Rule</i>	<i>Sentential Form</i>
	$Expr$
1	$Expr Op Number$
3	$Expr + Number$
2	$Number + Number$

## اشتقاق - مثال ۳

$Expr$	$\rightarrow$	$Expr Op Number$
		$Number$
$Op$	$\rightarrow$	$+$
		$-$
		$\times$
		$\div$

$Number + Number \times Number.$

<i>Rule</i>	<i>Sentential Form</i>
	$Expr$
1	$Expr Op Number$
5	$Expr \times Number$
1	$Expr Op Number \times Number$
3	$Expr + Number \times Number$
2	$Number + Number \times Number$

## اشتقاق - مثال ۴

“slope \* x + intercept”:

$$\begin{aligned} \text{expr} &\implies \text{expr op } \underline{\text{expr}} \\ &\implies \text{expr } \underline{\text{op}} \text{ id} \\ &\implies \underline{\text{expr}} + \text{id} \\ &\implies \text{expr op } \underline{\text{expr}} + \text{id} \\ &\implies \text{expr } \underline{\text{op}} \text{ id} + \text{id} \\ &\implies \underline{\text{expr}} * \text{id} + \text{id} \\ &\implies \text{id} * \text{id} + \text{id} \\ &\quad (\text{slope}) \quad (\text{x}) \quad (\text{intercept}) \end{aligned}$$

# صورت بکوس-نائور

طراحی جهت تعریف الگول-۶۰

عدم اجازه بستار کلین

پرانتزگذاری عبارت منظم

$\langle \text{SheepNoise} \rangle ::= \langle \text{SheepNoise} \rangle \underline{\text{baa}}$   
| baa

پیشگامان بکوس، نائر، چامسکی

▪ تقریبا همزمان

▪ پیشرفت علم، و همزمانی اکتشافات و ابداعات

# اشتقاق و درخت تجزیه

## اشتقاق

- مجموعه جانشینی‌ها که رشته‌ای از الفبا را از متغیر شروع موجب می‌شود

## شکل جمله‌ای

- هر رشته از علائم در طول اشتقاق

## شکل جمله‌ای نهائی

- صرفاً شامل الفباء

## درخت تجزیه

- نمایشی تصویری اشتقاق

- ریشه: متغیر شروع دستور

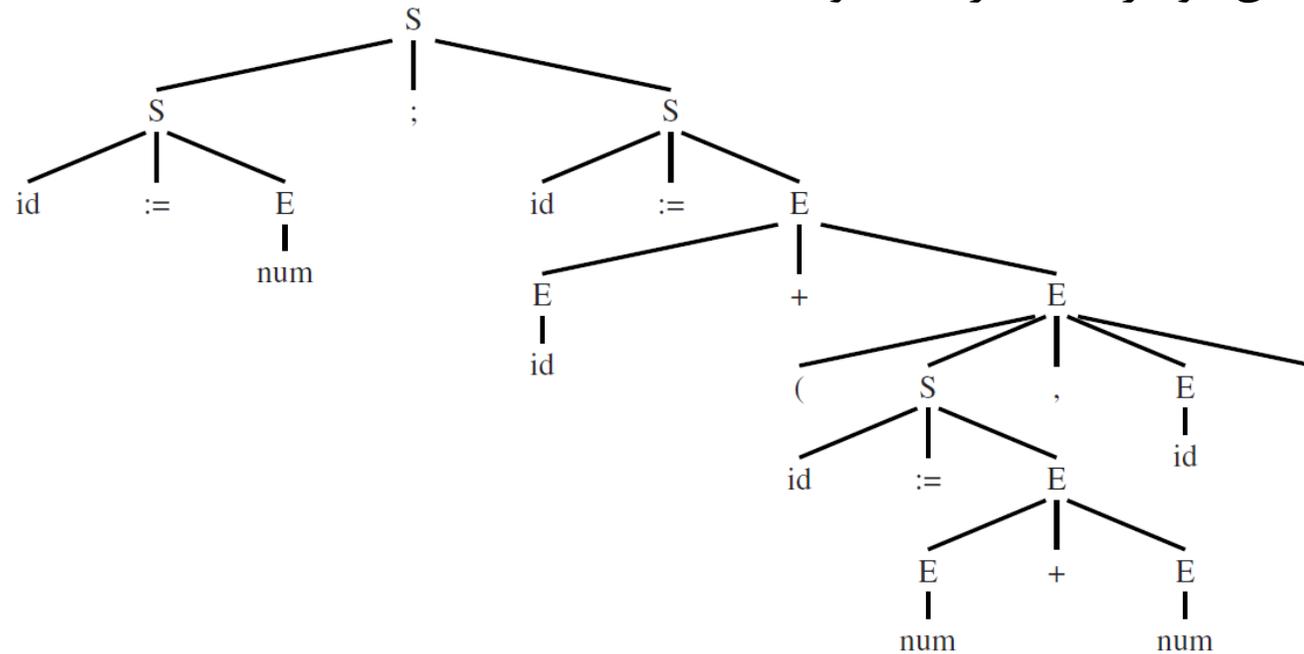
- برگ‌ها نتایج آن

- هر راس داخلی و فرزندانش نمایشگر یک قانون تولید

# درخت تجزیه

حاصل از اتصال هر متغیر به علائمی که در پی اشتقاقی از آن مشتق شده‌اند.

امکان یکی بودن اشتقاق دو درخت تجزیه متفاوت



# دستورهای مبهم

امکان اشتقاق جمله‌ای با دو درخت تجزیه متفاوت

مثال - جمله  $id := id + id + id$

$E \rightarrow id$

$E \rightarrow num$

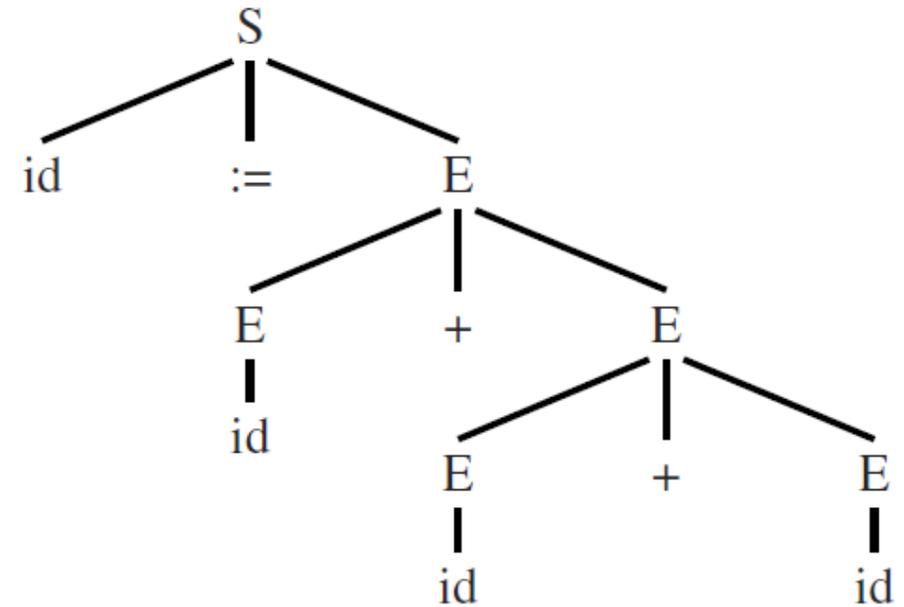
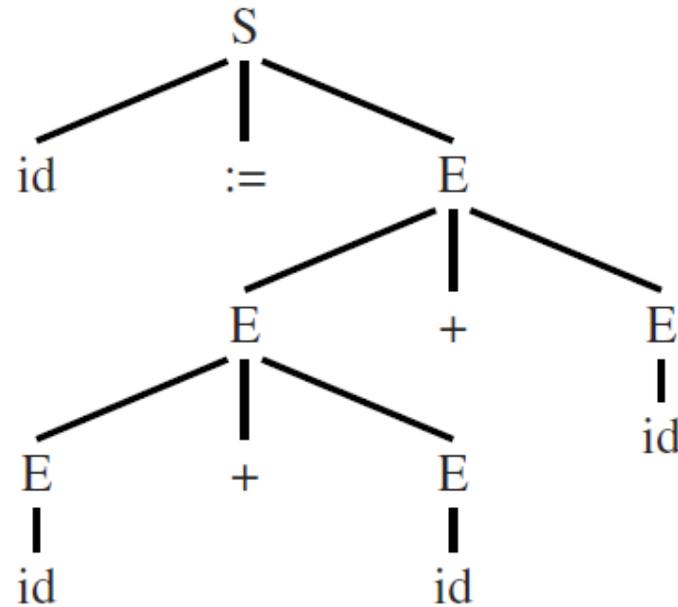
$E \rightarrow E * E$

$E \rightarrow E / E$

$E \rightarrow E + E$

$E \rightarrow E - E$

$E \rightarrow ( E )$



## دستورهای مبهم - مثال ۲

دستوری مبهم با دو درخت تجزیه متفاوت برای جمله ۱-۲-۳

$$E \rightarrow id$$

$$E \rightarrow num$$

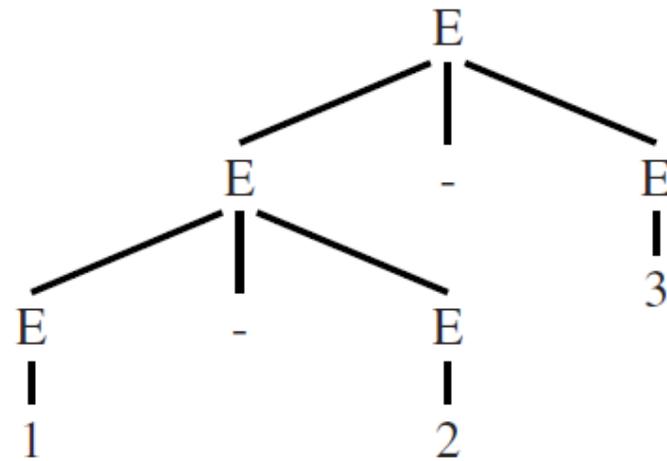
$$E \rightarrow E * E$$

$$E \rightarrow E / E$$

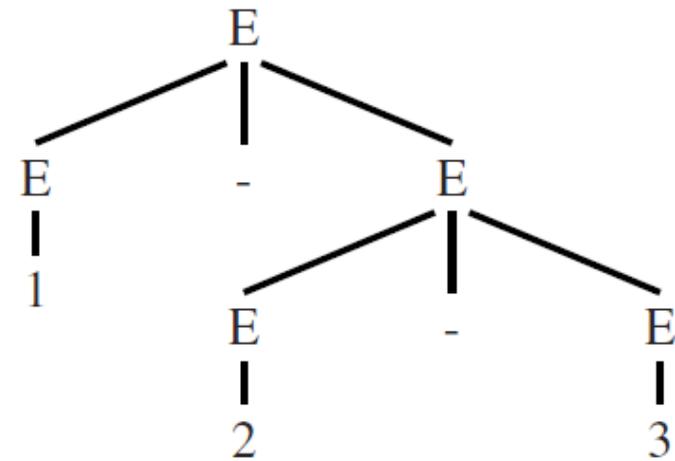
$$E \rightarrow E + E$$

$$E \rightarrow E - E$$

$$E \rightarrow ( E )$$



$$(1-2)-3 = -4$$



$$1-(2-3) = 2$$

# دستورهای مبهم - مثال ۳

دستوری مبهم با دو درخت تجزیه متفاوت برای جمله

$$1+2*3$$

$$E \rightarrow id$$

$$E \rightarrow num$$

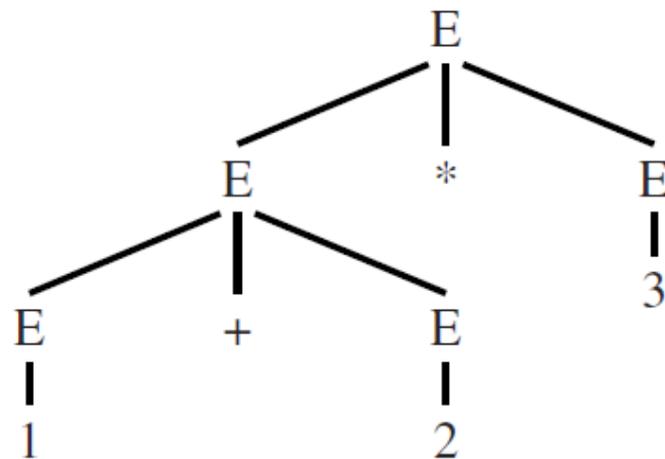
$$E \rightarrow E * E$$

$$E \rightarrow E / E$$

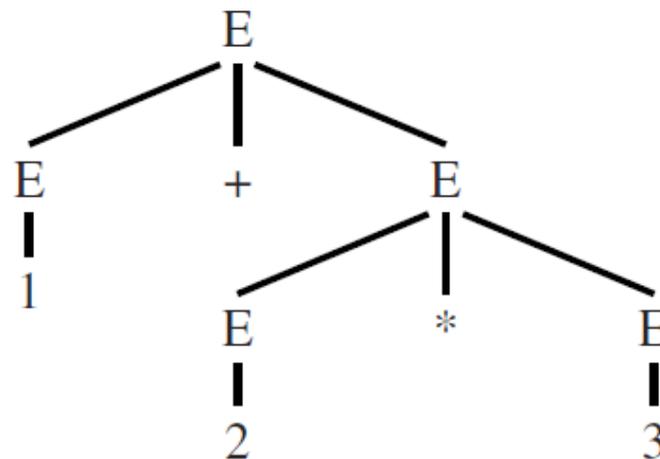
$$E \rightarrow E + E$$

$$E \rightarrow E - E$$

$$E \rightarrow ( E )$$



$$(1+2)*3=9$$

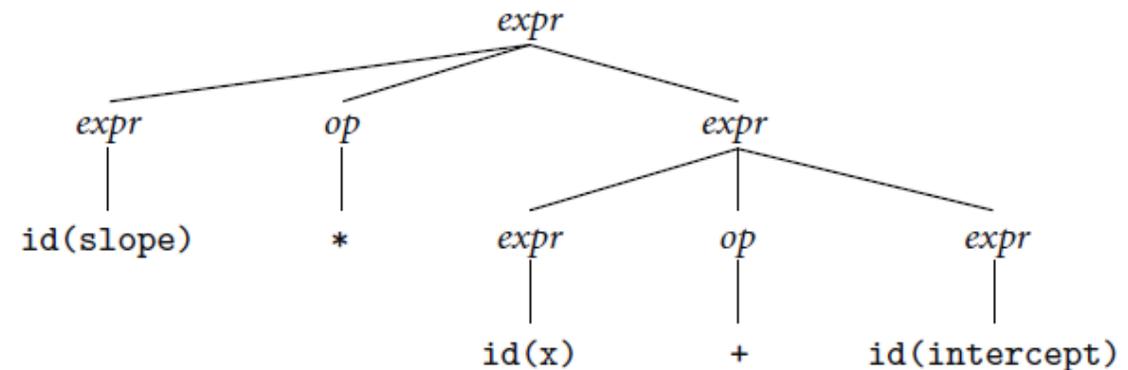
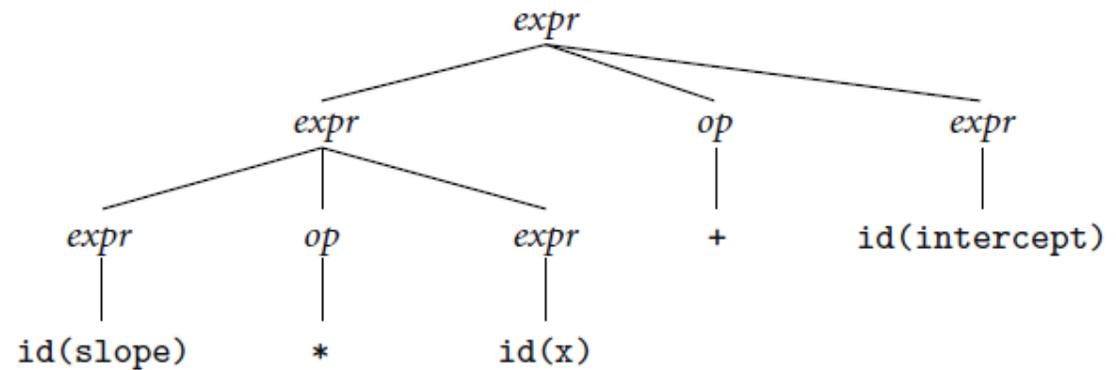


$$1+(2*3)=7$$

# دستورهای مبهم - مثال ۴

“slope \* x + intercept”:

$expr \Rightarrow expr \ op \ expr$   
 $\Rightarrow expr \ op \ id$   
 $\Rightarrow \underline{expr} \ + \ id$   
 $\Rightarrow expr \ op \ \underline{expr} \ + \ id$   
 $\Rightarrow expr \ \underline{op} \ id \ + \ id$   
 $\Rightarrow \underline{expr} \ * \ id \ + \ id$   
 $\Rightarrow \quad id \quad * \ id \ + \quad id$   
 $\quad (slope) \quad (x) \quad (intercept)$

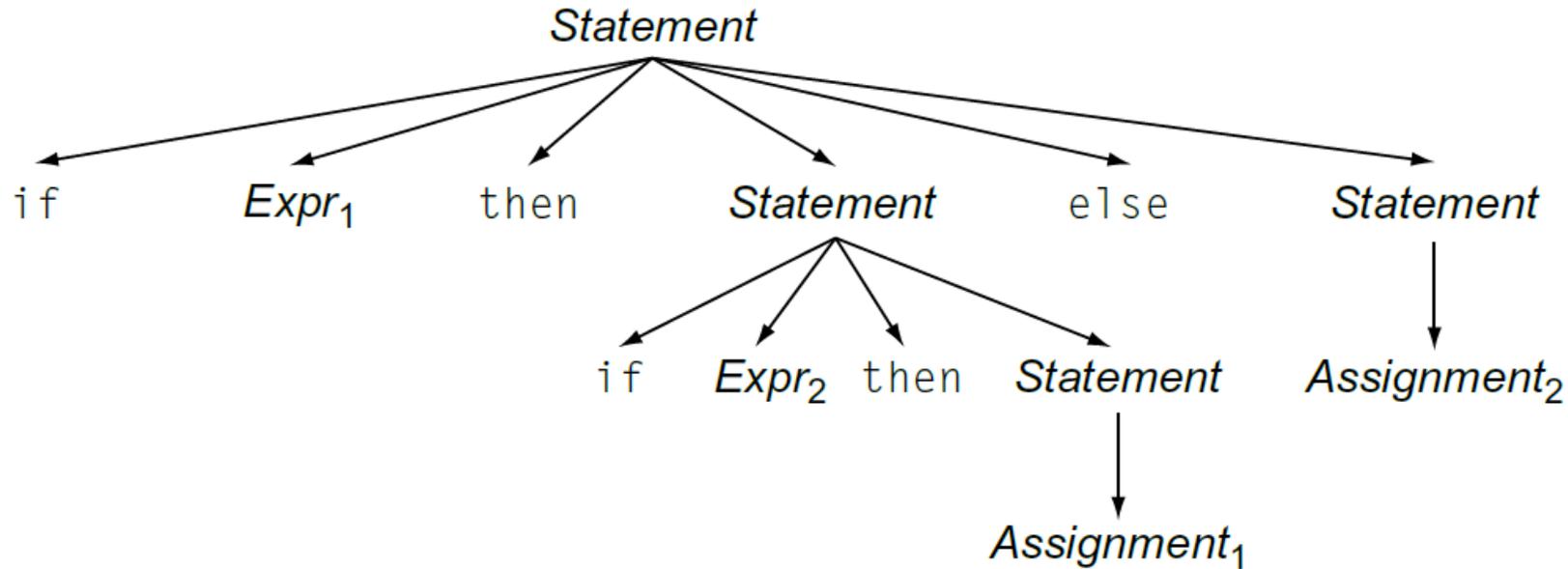
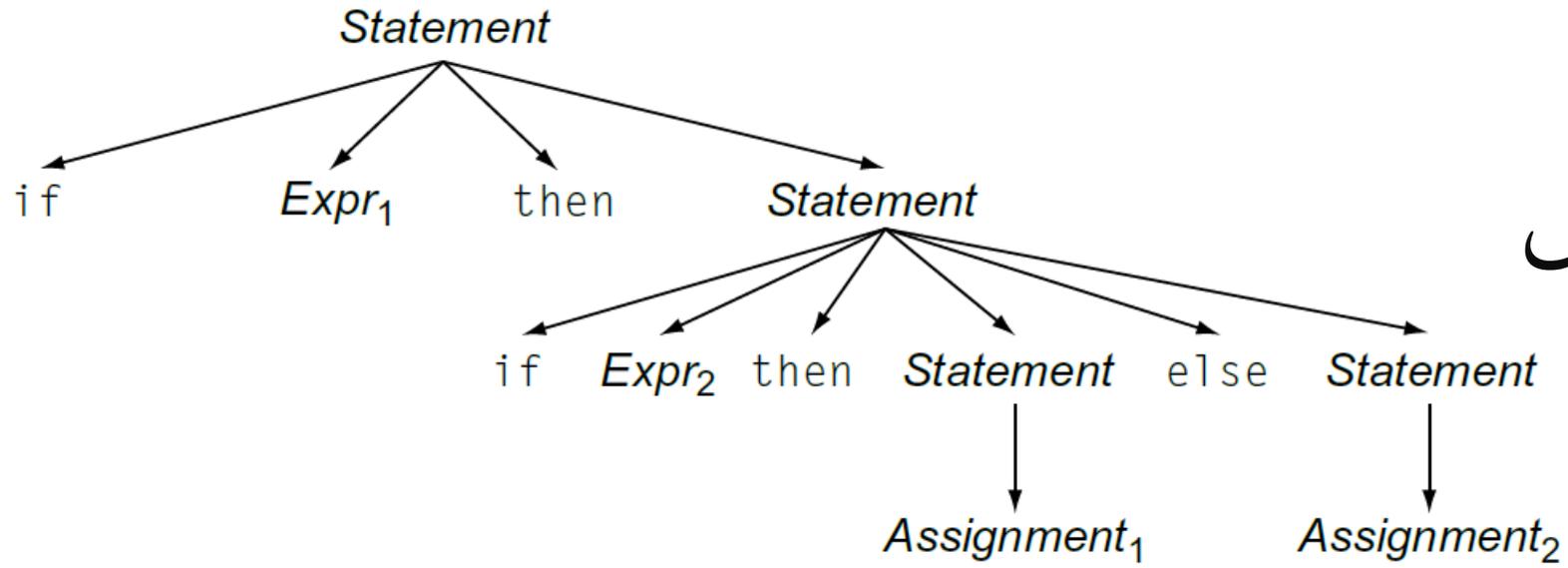


مثال

$S \rightarrow iEtSeS|iEtS|A$

جمله  $iEtSeS$

مثال



# دستورهای مبهم - ادامه

## ابهام

- دستور دارای بیش از یک درخت تجزیه برای چند رشته الفباء
- مشکل ساز هنگام ساخت تجزیه گر
- هنگام کامپایل
- عدم وجود الگوریتمی برای تشخیص ابهام

بی نهایت دستور مستقل از متن برای هر زبان مستقل از متن

- بسیاری از زبان ها دارای دستور غیر مبهم
- سعی بر استفاده از دستورهای غیر مبهم
- امکان تبدیل دستور مبهم به غیر مبهم
- دوری از علامت های بی فایده
- متغیرهایی که منجر به تولید رشته ای نشوند
- یا حرف الفبائی که در هیچ رشته ای نیاید

هنگام طراحی دستور مربوط به زبان برنامه نویسی

- به دنبال دستور منعکس کننده ساختار داخلی برنامه

# دلیل [های] ابهام دستورها

مثال -

$$E \rightarrow id$$

$$1 + 2 * 3 = \begin{cases} (1+2)*3 = 9 \\ 1+(2*3) = 7 \end{cases}$$

$$E \rightarrow num$$

$$1 - 2 - 3 = \begin{cases} (1-2)-3 = -4 \\ 1-(2-3) = 2 \end{cases}$$

$$E \rightarrow E * E$$

$$E \rightarrow E / E$$

$$E \rightarrow E + E$$

$$E \rightarrow E - E$$

$$E \rightarrow ( E )$$

دلیل [ها]؟

اولویت بالاتر ضرب نسبت به جمع

دلیل دیگر؟

شرکت پذیری از چپ (شرکت پذیری چپ) هر عملگر با اولویت یکسان

ترجیح  $(1-2)-3$  به جای  $1-(2-3)$

# رفع ابهام دستور

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)

$$E \rightarrow id$$

$$E \rightarrow num$$

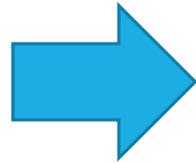
$$E \rightarrow E * E$$

$$E \rightarrow E / E$$

$$E \rightarrow E + E$$

$$E \rightarrow E - E$$

$$E \rightarrow ( E )$$



$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

$$T \rightarrow F$$

$$F \rightarrow id$$

$$F \rightarrow num$$

$$F \rightarrow ( E )$$

# رفع ابهام دستور - ادامه

$E \rightarrow id$

$E \rightarrow num$

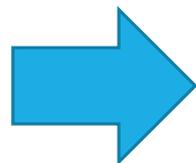
$E \rightarrow E * E$

$E \rightarrow E / E$

$E \rightarrow E + E$

$E \rightarrow E - E$

$E \rightarrow ( E )$



$E \rightarrow E + T$

$E \rightarrow E - T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow T / F$

$T \rightarrow F$

$F \rightarrow id$

$F \rightarrow num$

$F \rightarrow ( E )$

دو دستور متناظر

▪ تولید یکسانی از جملات

دستور جدید بر عکس دستور قبلی

▪ هر جمله دارای دقیقاً «یک» درخت تجزیه

# رفع ابهام دستور - /د/امه

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

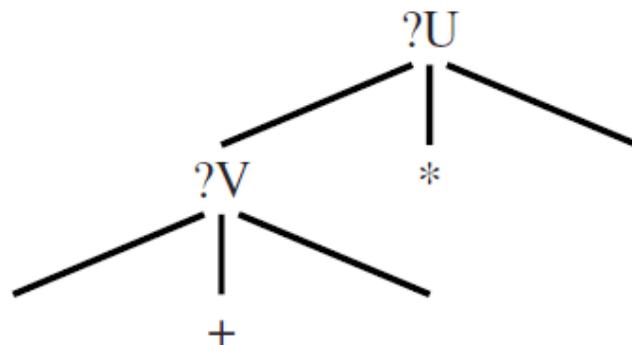
$$T \rightarrow F$$

$$F \rightarrow id$$

$$F \rightarrow num$$

$$F \rightarrow ( E )$$

عدم امکان تولید درخت تجزیه زیر از دستور مقابل



دو عملگر با اولویت متفاوت

▪ عملگر با اولویت بالاتر در سطح پایین‌تر درخت

به چه معنی؟

▪ «اعمال و کد کردن قانون» در «ساختار»

# رفع ابهام دستور - /د/امه

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

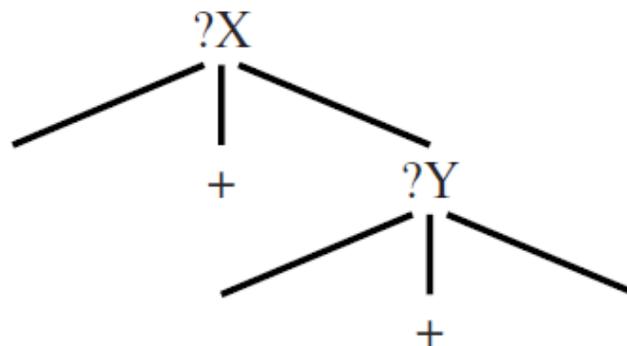
$$T \rightarrow F$$

$$F \rightarrow id$$

$$F \rightarrow num$$

$$F \rightarrow ( E )$$

عدم امکان تولید درخت تجزیه زیر از دستور مقابل



دو عملگر با اولویت یکسان  
▪ شرکت پذیری چپ عملگرهای با اولویت یکسان ذخیره شده در سطح پایین تر درخت

به چه معنی؟  
▪ باز هم «اعمال و کد کردن قانون» در «ساختار»

در صورت ایجاد شرکت پذیری راست  
▪ تغییر به  $T \rightarrow F * T$

# رفع ابهام دستور - /د/امه

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

$$T \rightarrow F$$

$$F \rightarrow id$$

$$F \rightarrow num$$

$$F \rightarrow ( E )$$

معمولا حذف ابهام با تبدیل دستور

دستور مبهم معمولا موجب مشکل در نوشتن برنامه و فهم آن

دلیل تعریف متغیرها مشخص شد - بحث ترجمه یا استفاده یا ابداع اصطلاحی متفاوت در زبانی

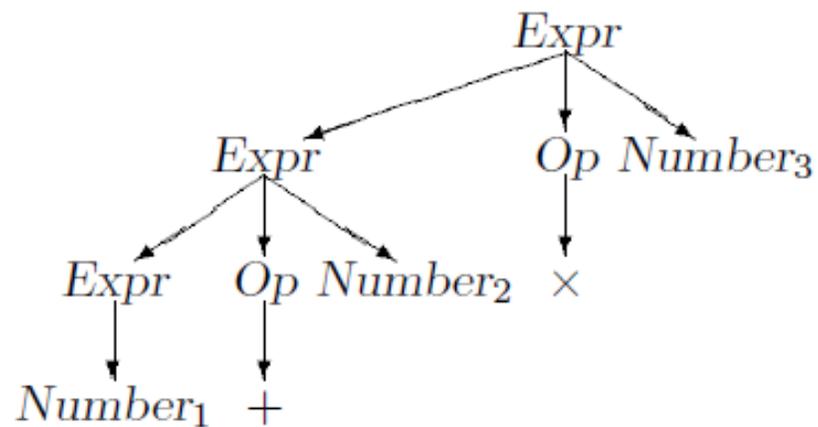
▪ معنا در متن

▪ توجه به دلیل افزودن و بکارگیری اصطلاحی جدید

# معنا در ساختار

Number + Number × Number

1.  $Expr \rightarrow Expr Op Number$
2.  $\quad \quad \quad | \quad Number$
3.  $Op \rightarrow +$
4.  $\quad \quad \quad | \quad -$
5.  $\quad \quad \quad | \quad \times$
6.  $\quad \quad \quad | \quad \div$



$(Number_1 + Number_2) \times Number_3$

$Number_1 + (Number_2 \times Number_3)$

# معنا در ساختار

1.	$Expr$	$\rightarrow$	$Expr Op Number$
2.			$Number$
3.	$Op$	$\rightarrow$	$+$
4.			$-$
5.			$\times$
6.			$\div$

نیاز به تعریف اولویت

بازسازی ساختار جهت جاسازی اولویت

در این مثال دو سطح اولویت

▪ سطح پائین تر جهت  $+$  و  $-$

▪ سطح بالاتر جهت  $*$  و  $/$

1.	$Expr$	$\rightarrow$	$Expr + Term$
2.			$Expr - Term$
3.			$Term$
4.	$Term$	$\rightarrow$	$Term \times Number$
5.			$Term \div Number$
6.			$Number$

افزودن متغیری دیگر به ازای هر سطح اولویت

▪  $Expr$  اولویت پائین تر

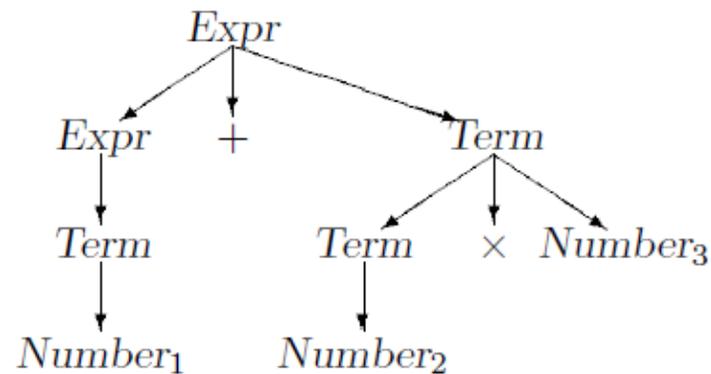
▪  $Term$  اولویت بالاتر

# معنا در ساختار

1.  $Expr \rightarrow Expr + Term$
2.  $Expr \rightarrow Expr - Term$
3.  $Expr \rightarrow Term$
4.  $Term \rightarrow Term \times Number$
5.  $Term \rightarrow Term \div Number$
6.  $Term \rightarrow Number$

Number + Number × Number

Rule	Sentential Form
	$Expr$
1	$Expr + Term$
4	$Expr + Term \times Number_3$
6	$Expr + Number_2 \times Number_3$
3	$Term + Number_2 \times Number_3$
6	$Number_1 + Number_2 \times Number_3$

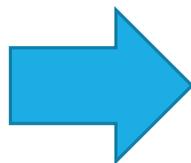


# معنا در ساختار

افزودن پرانتز اختیاری با بالاترین اولویت چگونه؟

تعریف متغیر دیگر و دوباره نویسی قواعد

1.		<i>Expr</i>	→	<i>Expr</i> + <i>Term</i>
2.				<i>Expr</i> - <i>Term</i>
3.				<i>Term</i>
4.		<i>Term</i>	→	<i>Term</i> × <b>Number</b>
5.				<i>Term</i> ÷ <b>Number</b>
6.				<b>Number</b>



1.		<i>Expr</i>	→	<i>Expr</i> + <i>Term</i>
2.				<i>Expr</i> - <i>Term</i>
3.				<i>Term</i>
4.		<i>Term</i>	→	<i>Term</i> × <i>Factor</i>
5.				<i>Term</i> ÷ <i>Factor</i>
6.				<i>Factor</i>
7.		<i>Factor</i>	→	( <i>Expr</i> )
8.				<b>Number</b>
9.				<b>Identifier</b>

# نشان پایان-فایل

$$S \rightarrow E \$$$

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

$$E \rightarrow T$$

$$T \rightarrow T * F$$

$$T \rightarrow T / F$$

$$T \rightarrow F$$

$$F \rightarrow id$$

$$F \rightarrow num$$

$$F \rightarrow ( E )$$

جهت مشخص سازی پایان فایل

- نیاز به تعریف علامتی جدید که در متغیرها یا حروف الفباء دستور بکار نرفته است
- مثلا # یا \$

S متغیر آغاز

جهت نمایش اینکه \$ پس از عبارت-S می آید

- استفاده از دستور افزوده

- شروع با متغیر آغاز جدید S' و قاعده تولید  $S' \rightarrow S$
- مثال- در دستور روبرو، E متغیر آغاز

# منابع

[بیرسبز]

[اژدرها]

[کوپر]

[فیشر]