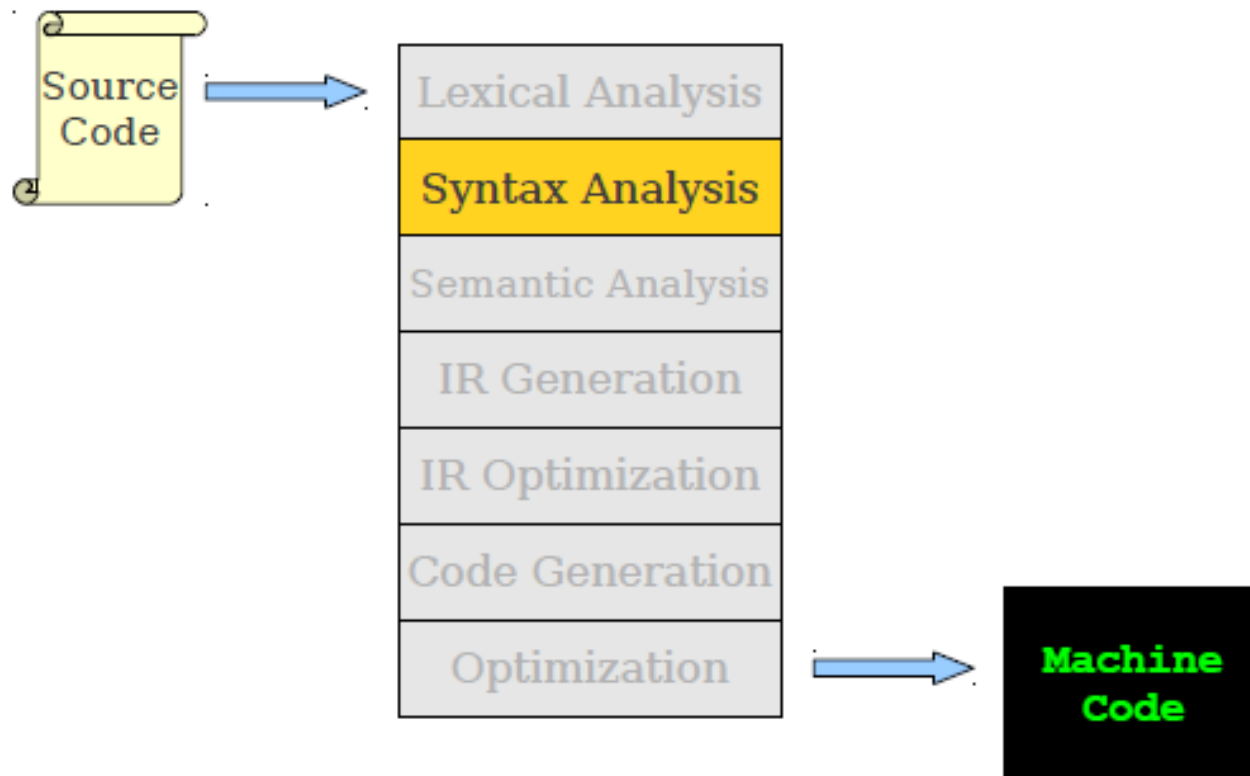
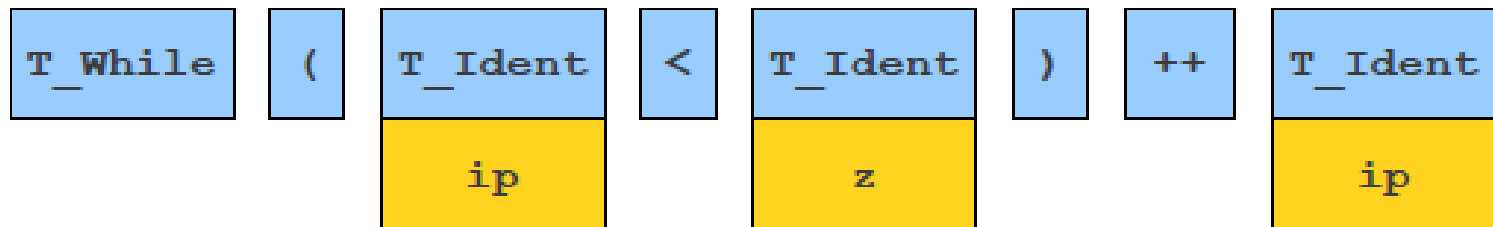
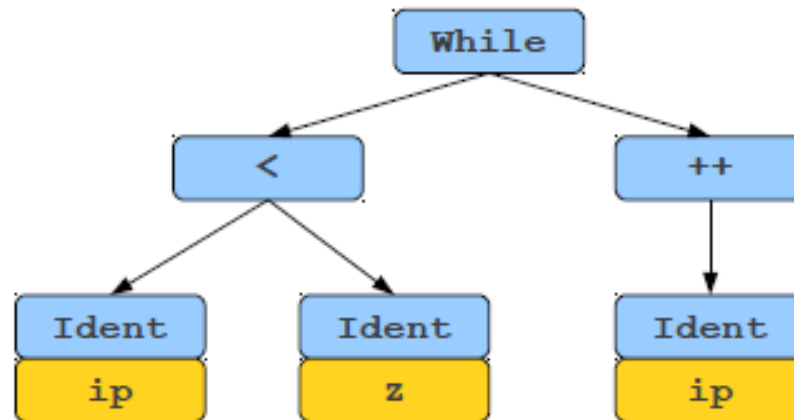


# طراحی کامپایلرها

تحلیل نحوی (قسمت اول)





w	h	i	l	e		(	i	p		<		z	)	\n	\t	+	+	i	p	;
---	---	---	---	---	--	---	---	---	--	---	--	---	---	----	----	---	---	---	---	---

```

while (ip < z)
    ++ip;
  
```



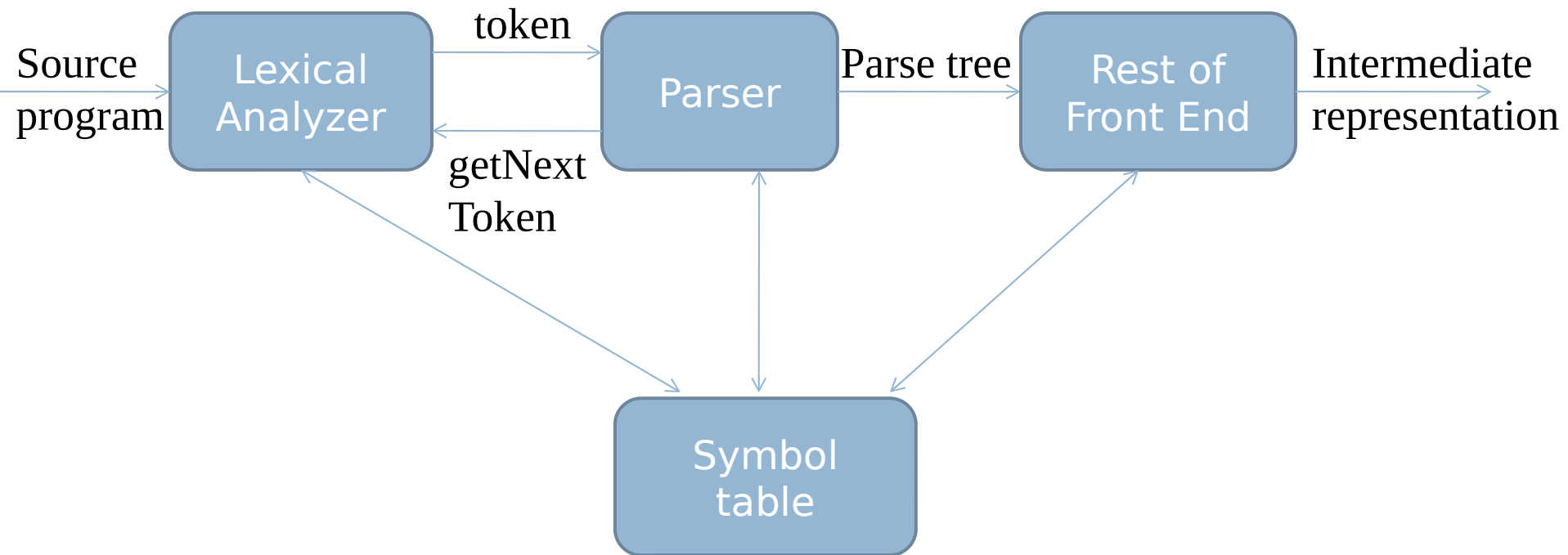
T_Do	[	T_For	]	=	T_New	T_IntConst
						0

d	o	[	f	o	r	]		=		n	e	w		0	;
---	---	---	---	---	---	---	--	---	--	---	---	---	--	---	---

do[for] = new 0;

# نقش پارسر

5



# استفاده از گرامرها

6

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid \mathbf{id}$$
$$E \rightarrow TE'$$
$$E' \rightarrow +TE' \mid \varepsilon$$
$$T \rightarrow FT'$$
$$T' \rightarrow *FT' \mid \varepsilon$$
$$F \rightarrow (E) \mid \mathbf{id}$$

# پوشش خطا

7

## □ خطاهای متداول برنامه سازی

- لغوی
- نحوی
- معنایی
- منطقی

## □ اهداف پوشش خطا

- وجود خطا واضح و صحیح گزارش شود.
- هر خطا سریعتر پوشش داده شود تا خطاهای بعدی تشخیص داده شوند.
- کمترین سربار به پردازش برنامه های صحیح اعمال شود.

# استراتژیهای پوشش خطا

8

## Panic mode recovery □

- نمادهای ورودی یکی یکی دور ریخته شوند تا به یکی از کاراکترهای همگام سازی مشخص شده برسد.

## پوشش سطح عبارت □

- پیشوندی از ورودی باقیمانده با رشته ای جایگزین می شود که اجازه دهد پارسر به کارش ادامه دهد.

## مولدهای خطا □

- به گرامر، مولدهایی اضافه شود که ساختارهای خطا را تولید کنند.

## • تصحیح سراسری

- انتخاب دنباله مینیمم تغییرات برای به دست آوردن اصلاح با کمترین هزینه



# Context free grammars

9

- Terminals پایانه
- Nonterminals غیر پایانه
- Start symbol نماد شروع
- Productions مولدها، قولین

expression  $\rightarrow$  expression + term

expression  $\rightarrow$  expression – term

expression  $\rightarrow$  term

term  $\rightarrow$  term \* factor

term  $\rightarrow$  term / factor

term  $\rightarrow$  factor

factor  $\rightarrow$  (expression)

factor  $\rightarrow$  **id**

# اشتقاق (Derivation)

10

□ قوانین برای تولید رشته استفاده می شوند.

## □ Rightmost and leftmost derivations

- $E \rightarrow E + E \mid E * E \mid -E \mid (E) \mid \mathbf{id}$

- Derivations for  **$-(\mathbf{id}+\mathbf{id})$**

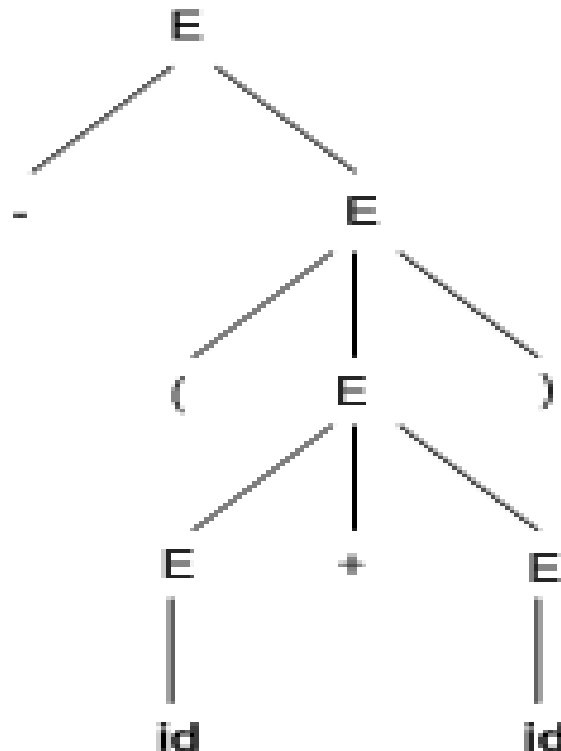
- $E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(E+E) \Rightarrow -(\mathbf{id}+E) \Rightarrow -(\mathbf{id}+\mathbf{id})$

# Parse trees (درخت تجزیه)

11

□ **-(id+id)**

■  $E \Rightarrow -E \Rightarrow -(E) \Rightarrow -(E+E) \Rightarrow -(\mathbf{id}+E) \Rightarrow -(\mathbf{id}+\mathbf{id})$



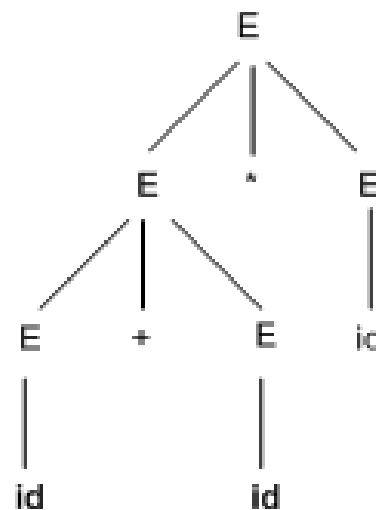
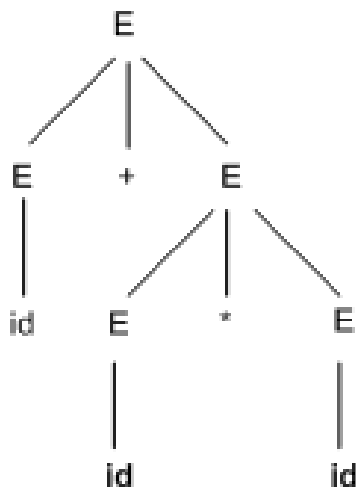
# Ambiguity (ابهام)

12

□ برای برخی رشته ها بیش از یک درخت تجزیه وجود دارد.

□  $E \rightarrow E + E \mid E * E \mid -E \mid (E) \mid \mathbf{id}$

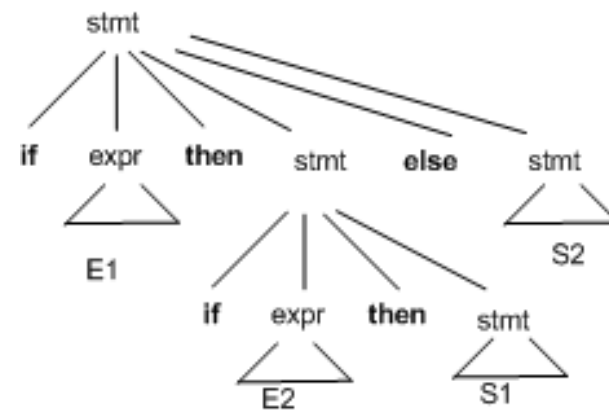
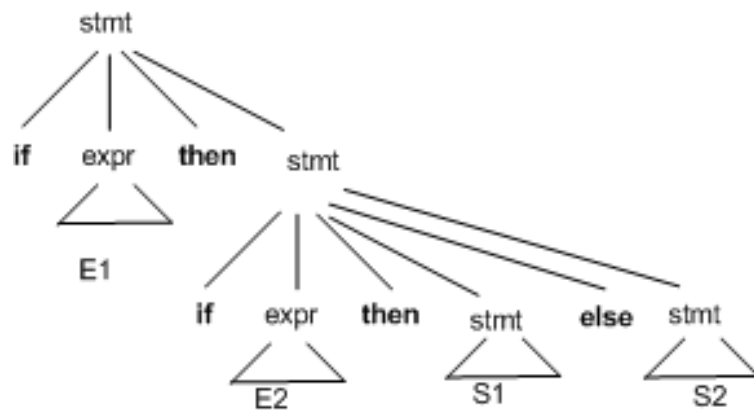
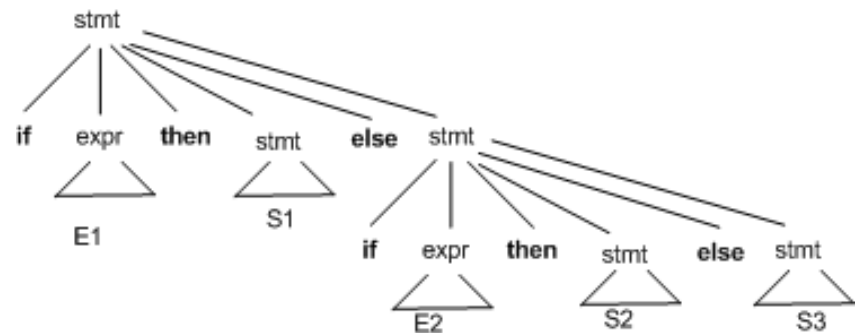
□ Example:  $\mathbf{id + id * id}$



# رفع ابهام

13

stmt  $\rightarrow$  **if** expr **then** stmt  
| **if** expr **then** stmt **else** stmt  
| **other**



# رفع ابهام (ادامه)

14

stmt  $\rightarrow$  matched\_stmt  
| open\_stmt

matched\_stmt  $\rightarrow$  If expr **then** matched\_stmt **else** matched\_stmt  
| other

open\_stmt  $\rightarrow$  If expr **then** stmt  
| If expr **then** matched\_stmt **else** open\_stmt

# رفع بازگشتی چپ

15

□ گرامر بازگشتی چپ است اگر غیرپایانه  $A$  وجود داشته باشد که یک اشتقاق به شکل  $A \Rightarrow A\alpha$  داشته باشد.

□ برای رفع بازگشتی چپ:

- For a rule like:
  - $A \rightarrow A\alpha|\beta$
- We may replace it with
  - $A \rightarrow \beta A'$
  - $A' \rightarrow \alpha A' \mid \epsilon$

# فاکتورگیری چپ

16

□ گرامر زیر را در نظر بگیرید:

- $\text{Stmt} \rightarrow \text{if expr then stmt else stmt}$
- $\text{Stmt} \rightarrow \text{if expr then stmt}$
- اگر ورودی if دیده شود، پارسر تشخیص نمیدهد کدام قانون را استفاده کند.

□ برای فاکتورگیری چپ:

□ قانون  $A \rightarrow \alpha\beta1 \mid \alpha\beta2$  با قوانین زیر جایگزین می شود:

- $A \rightarrow \alpha A'$
- $A' \rightarrow \beta1 \mid \beta2$



# Left factoring (cont.)

17

## □ Algorithm

- For each non-terminal  $A$ , find the longest prefix  $\alpha$  common to two or more of its alternatives. If  $\alpha \neq \epsilon$ , then replace all of  $A$ -productions  $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \dots \mid \alpha\beta_n \mid \gamma$  by
  - $A \rightarrow \alpha A' \mid \gamma$
  - $A' \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$

## □ Example:

- $S \rightarrow l E t S \mid i E t S e S \mid a$
- $E \rightarrow b$