

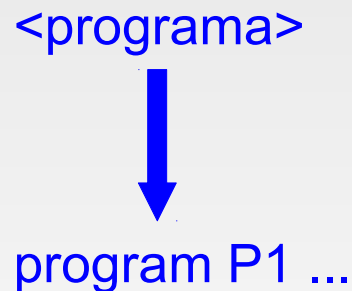
Construção de Compiladores

Análise Sintática Descendente – parte1

Profa. Helena Caseli
helenacaseli@dc.ufscar.br

Análise Sintática Descendente

- Como é feita?
 - A análise é feita da raiz para as folhas
 - Parte-se do símbolo inicial da gramática e, por meio de derivações, chega-se à sequência de *tokens* retornada pelo analisador léxico



- Expansão
- Derivação à esquerda

Análise Sintática Descendente

- Analisadores sintáticos descendentes (ASD) – 2 tipos
 - Analisador com retrocesso
 - Testa diferentes possibilidades de análise sintática da entrada, retrocedendo se alguma possibilidade falhar
 - Mais poderoso e mais lento (tempo exponencial)
 - Analisador preditivo
 - Tenta prever a construção seguinte na cadeia de entrada com base em um ou mais *tokens* de verificação à frente
 - ➔ **Gramática LL(1)**

ASD com Retrocesso

- Como é feita?
 - A partir da raiz, a árvore é expandida aplicando-se as produções adequadas
 - Se a aplicação de uma produção falha, tenta-se outra a partir do ponto de falha
 - O processo termina quando a cadeia é reconhecida ou quando as regras se esgotaram e a cadeia não foi reconhecida

ASD com Retrocesso

- Exemplo

$$\begin{aligned} \langle E \rangle &::= \langle T \rangle + \langle E \rangle \mid \langle T \rangle \\ \langle T \rangle &::= \langle F \rangle * \langle T \rangle \mid \langle F \rangle \\ \langle F \rangle &::= a \mid b \mid (\langle E \rangle) \end{aligned}$$

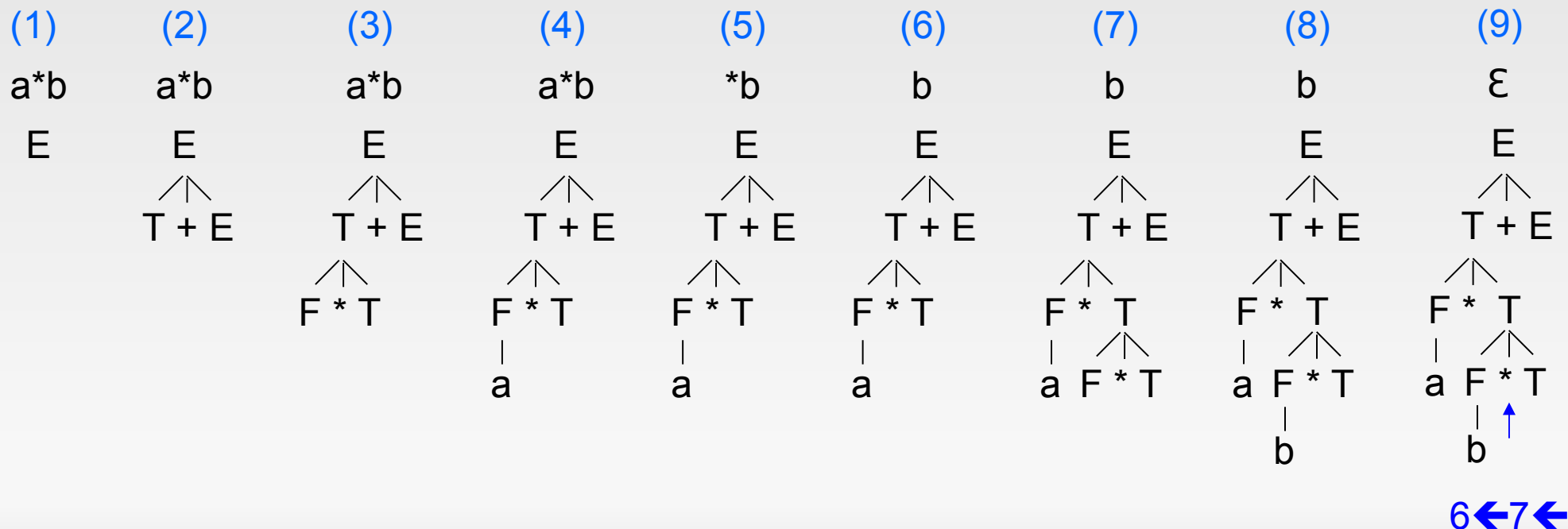
Reconhecer a cadeia a^*b

ASD com Retrocesso

Exemplo

$\langle E \rangle ::= \langle T \rangle + \langle E \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle F \rangle * \langle T \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= a \mid b \mid (\langle E \rangle)$

Reconhecer a cadeia a^*b



ASD com Retrocesso

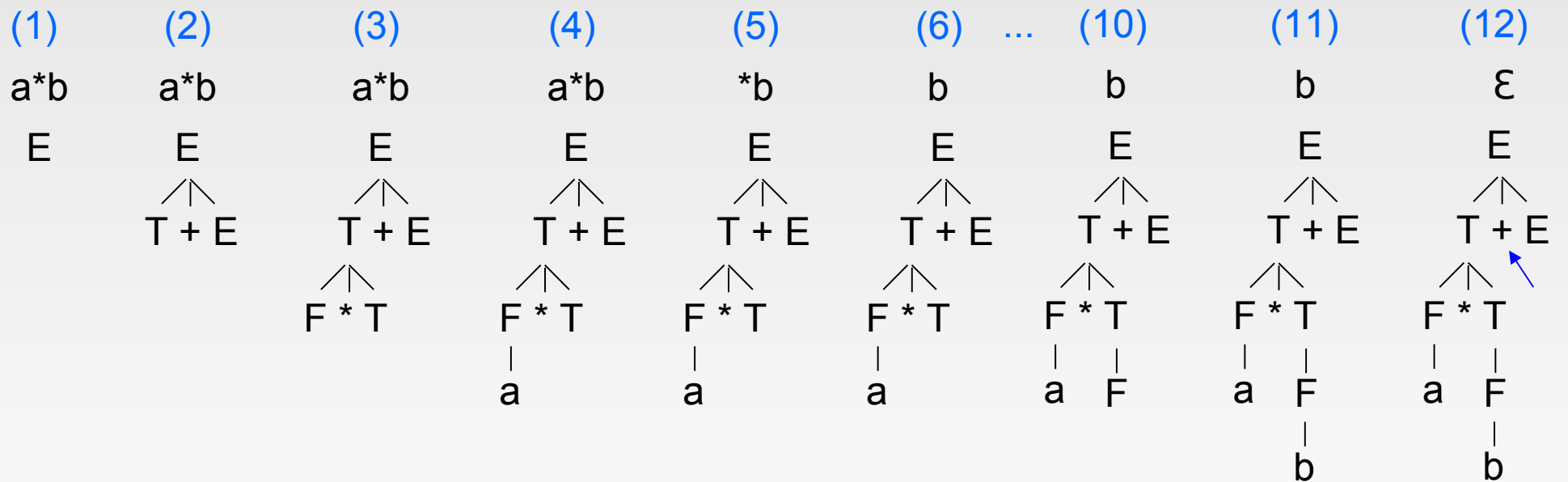
Exemplo

$\langle E \rangle ::= \langle T \rangle + \langle E \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle F \rangle * \langle T \rangle \mid \langle F \rangle$

$\langle F \rangle ::= a \mid b \mid (\langle E \rangle)$

Reconhecer a cadeia a^*b



2 ← 3 ← 6 ← 10 ←

ASD com Retrocesso

- Exemplo

$$\begin{aligned} \langle E \rangle &::= \langle T \rangle + \langle E \rangle \mid \langle T \rangle \\ \langle T \rangle &::= \langle F \rangle * \langle T \rangle \mid \langle F \rangle \\ \langle F \rangle &::= a \mid b \mid (\langle E \rangle) \end{aligned}$$

Reconhecer a cadeia a^*b

(1)	(2)	...	(13)	(14)	(15)
a^*b	a^*b		a^*b	a^*b	*b
E	E		E	E	E
	$\swarrow \downarrow \searrow$		$\swarrow \downarrow \searrow$	$\swarrow \downarrow \searrow$	$\swarrow \downarrow \searrow$
	T + E		T + E	T + E	T + E
			F	F	F
				a	a

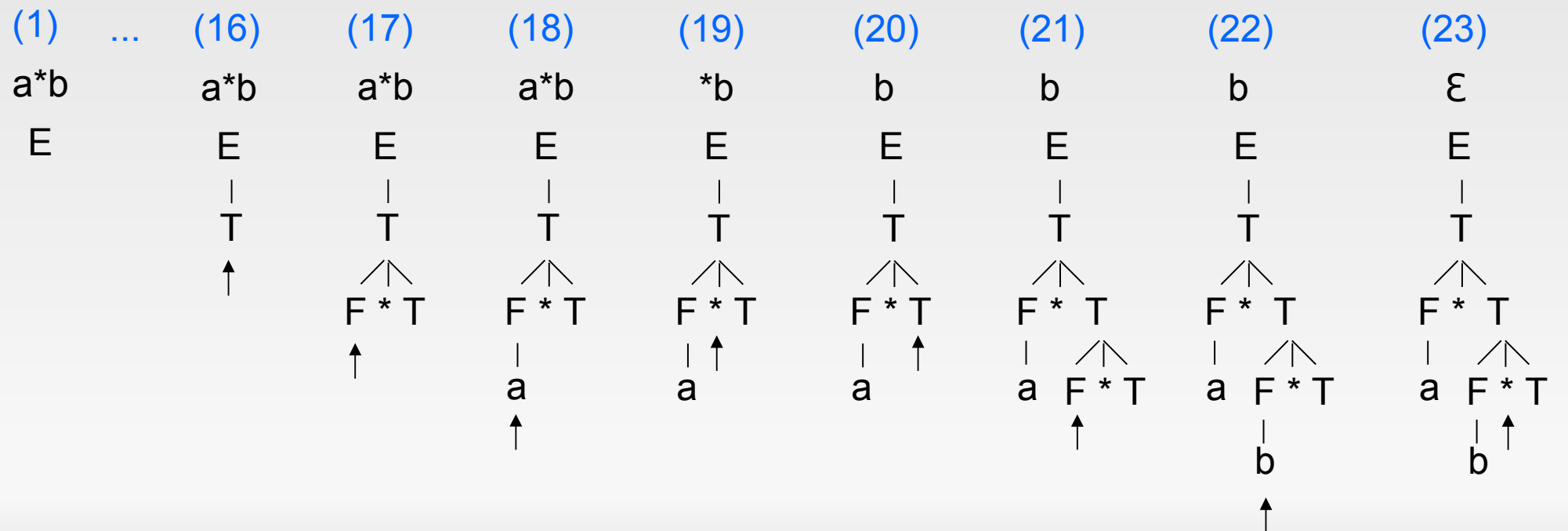
1 ← 2 ← 13 ←

ASD com Retrocesso

Exemplo

$\langle E \rangle ::= \langle T \rangle + \langle E \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle F \rangle * \langle T \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= a \mid b \mid (\langle E \rangle)$

Reconhecer a cadeia a^*b

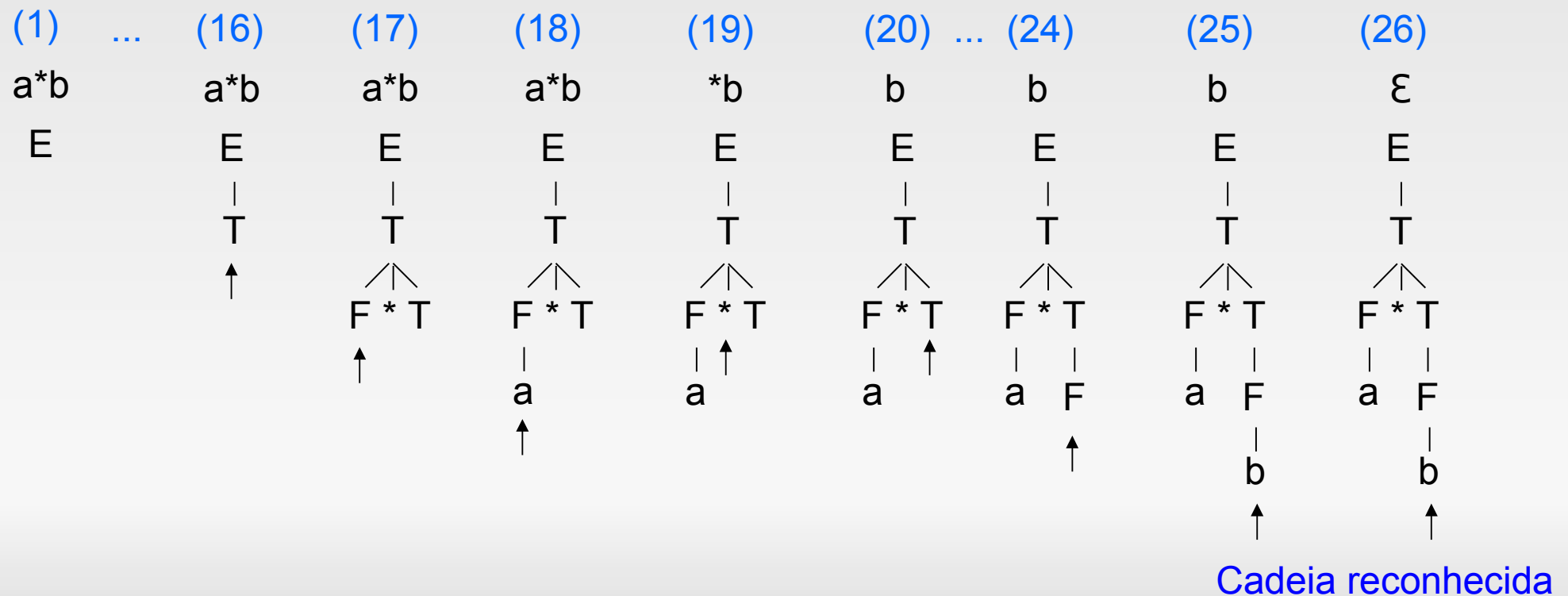


ASD com Retrocesso

Exemplo

$\langle E \rangle ::= \langle T \rangle + \langle E \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle F \rangle * \langle T \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= a \mid b \mid (\langle E \rangle)$

Reconhecer a cadeia a^*b



ASD com Retrocesso

- Análise sintática descendente com retrocesso
 - Vantagem
 - É de fácil implementação
 - Desvantagens
 - Apresenta tempo de execução exponencial
 - É ineficiente
 - Gera a necessidade de desfazer ações semânticas
 - Tem dificuldade em indicar com precisão o local do erro
- ➔ Assim, a análise sintática descendente só é eficiente quando são eliminados os retrocessos
 - ➔ Sabe-se de antemão qual regra aplicar
 - ➔ **Análise sintática descendente preditiva**

ASD Preditiva

- Análise sintática descendente preditiva – 2 tipos
 - Recursiva
 - É bastante versátil e é o método mais adequado para um analisador sintático escrito manualmente
 - Não recursiva (ou tabular)
 - Embora não seja muito utilizada na prática, é útil como estudo de um esquema simples com uma pilha explícita
 - É útil, também, para formalizar alguns problemas que aparecem na análise descendente recursiva
- ➔ **Gramática LL(1)**

ASD Preditiva

- Gramática LL(1)
 - *Left to right, Leftmost derivation*
 - Dado 1 único símbolo da cadeia de entrada, sabe-se qual regra da gramática aplicar
 - Características
 - Não é ambígua
 - Não é recursiva à esquerda
 - Está fatorada à esquerda
 - ➔ Para um não-terminal, não possui regras cujo lado direito começa com o mesmo terminal
 - Técnicas de transformação

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle)$
 $| \text{número}$
 $\langle \text{op} \rangle ::= + \mid - \mid *$

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle)$
 $| \text{número}$
 $\langle \text{op} \rangle ::= + \mid - \mid *$

➤ A gramática é ambígua ?

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 | ($\langle \text{exp} \rangle$)
 | número
 $\langle \text{op} \rangle ::= + \mid - \mid *$



$\langle \text{exp} \rangle ::= \textcolor{blue}{\langle \text{exp} \rangle} \langle \text{soma} \rangle \langle \text{termo} \rangle$
 | $\langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \textcolor{blue}{\langle \text{termo} \rangle} \langle \text{mult} \rangle \langle \text{fator} \rangle$
 | $\langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Ambiguidade removida aplicando-se regras de

- precedência de operadores
- [associatividade à esquerda](#)

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle$
 $| \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle$
 $| \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

➤ A gramática é recursiva à esquerda?

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle$
 $| \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle$
 $| \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Transforma regras do tipo
 $\langle A \rangle ::= \langle A \rangle \alpha \mid \beta$
em

$\langle A \rangle ::= \beta \langle A' \rangle$
 $\langle A' \rangle ::= \alpha \langle A' \rangle \mid \varepsilon$

ANTES

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle$
 $| \langle \text{termo} \rangle$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle$
 $| \langle \text{fator} \rangle$

DEPOIS

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$
 $\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{termo} \rangle \langle \text{exp}' \rangle \mid \varepsilon$
 $\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$
 $\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{fator} \rangle \langle \text{termo}' \rangle \mid \varepsilon$

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

<exp> ::= **<exp>** <soma> <termo>
 | <termo>

<soma> ::= + | -

<termo> ::= **<termo>** <mult> <fator>
 | <fator>

<mult> ::= *

<fator> ::= (<exp>) | número



<exp> ::= <termo> **<exp'>**

<exp'> ::= <soma> **<termo>** **<exp'>** | ϵ

<soma> ::= + | -

<termo> ::= <fator> **<termo'>**

<termo'> ::= <mult> **<fator>** **<termo'>** | ϵ

<mult> ::= *

<fator> ::= (<exp>) | número

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

<exp> ::= **<exp>** <soma> <termo>
 | <termo>

<soma> ::= + | -

<termo> ::= **<termo>** <mult> <fator>
 | <fator>

<mult> ::= *

<fator> ::= (<exp>) | número



<exp> ::= **<termo>** <exp'>

<exp'> ::= <soma> <exp> | ϵ

<soma> ::= + | -

<termo> ::= **<fator>** <termo'>

<termo'> ::= <mult> <termo> | ϵ

<mult> ::= *

<fator> ::= (<exp>) | número

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$
 $\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{exp} \rangle \mid \varepsilon$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$
 $\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{termo} \rangle \mid \varepsilon$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

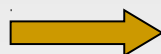
➤ A gramática está fatorada à esquerda?

ASD Preditiva

- Gramática LL(1)
 - Exemplo
 - A gramática seguinte é LL(1)? Se não é, transforme-a

Gramática original

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{op} \rangle \langle \text{exp} \rangle$
 $| (\langle \text{exp} \rangle)$
 $| \text{número}$
 $\langle \text{op} \rangle ::= + \mid - \mid *$



Gramática LL(1)

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$
 $\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{exp} \rangle \mid \varepsilon$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$
 $\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{termo} \rangle \mid \varepsilon$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= := \langle E \rangle$
 $\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= := \langle E \rangle$
 $\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$

➤ A gramática é ambígua ?

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= := \langle E \rangle$
 $\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$

➤ A gramática é recursiva à esquerda?

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= \langle E \rangle$
 $\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$

Transforma regras do tipo
 $\langle A \rangle ::= \langle A \rangle \alpha \mid \beta$
em

$\langle A \rangle ::= \beta \langle A' \rangle$
 $\langle A' \rangle ::= \alpha \langle A' \rangle \mid \varepsilon$

ANTES

$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$

DEPOIS

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle T \rangle \langle E' \rangle \mid \varepsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle F \rangle \langle T' \rangle \mid \varepsilon$

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= \langle E \rangle$
 $\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$

Transforma regras do tipo
 $\langle A \rangle ::= \langle A \rangle \alpha \mid \beta$
em

$\langle A \rangle ::= \beta \langle A' \rangle$
 $\langle A' \rangle ::= \alpha \langle A' \rangle \mid \varepsilon$

ANTES	DEPOIS
$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$	$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
	$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$
$\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$	$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
	$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= := \langle E \rangle$
 $\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$
 $\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$



$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= := \langle E \rangle$
 $\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle E \rangle \mid \epsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle T \rangle \mid \epsilon$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= := \langle E \rangle$
 $\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$

➤ A gramática está fatorada à esquerda?

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

$\langle S \rangle ::= i \langle A \rangle$
 $\langle A \rangle ::= \langle E \rangle$
 $\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$
 $\langle P \rangle ::= i \mid (\langle E \rangle)$

Transforma regras do tipo
 $\langle A \rangle ::= \alpha \beta \mid \alpha \gamma$
em

$\langle A \rangle ::= \alpha \langle A' \rangle$
 $\langle A' \rangle ::= \beta \mid \gamma$

ANTES
 $\langle F \rangle ::= \langle P \rangle - \langle F \rangle \mid \langle P \rangle$

DEPOIS
 $\langle F \rangle ::= \langle P \rangle \langle X \rangle$
 $\langle X \rangle ::= - \langle F \rangle \mid \varepsilon$

ASD Preditiva

- Gramática LL(1)
 - Exercício
 - A gramática seguinte é LL(1)? Se não é, transforme-a

Gramática original

```
<S> ::= i<A>
<A> ::= := <E>
<E> ::= <E> + <T> | <T>
<T> ::= <T> * <F> | <F>
<F> ::= <P> - <F> | <P>
<P> ::= i | (<E>)
```



Gramática LL(1)

```
<S> ::= i<A>
<A> ::= := <E>
<E> ::= <T><E'>
<E'> ::= +<E> | ε
<T> ::= <F><T'>
<T'> ::= *<T> | ε
<F> ::= <P><X>
<X> ::= -<F> | ε
<P> ::= i | (<E>)
```

ASD Preditiva Recursiva

- O que é?
 - ➔ Um ASD Preditivo Recursivo é
 - Um conjunto de procedimentos possivelmente recursivos
 - Um procedimento para cada não-terminal

- Exemplo:

Para a gramática a seguir teríamos quais procedimentos?

```
<exp> ::= <termo> <exp'>
<exp'> ::= <soma> <exp> | ε
<soma> ::= + | -
<termo> ::= <fator> <termo'>
<termo'> ::= <mult> <termo> | ε
<mult> ::= *
<fator> ::= ( <exp> ) | número
```


ASD Preditiva Recursiva

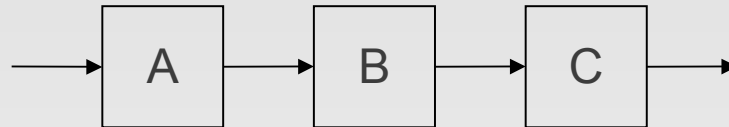
- Como é feita?
 - Regra gramatical para um não-terminal A
 - Procedimento para reconhecer um A
 - O lado direito da regra
 - Especifica a estrutura do código
- Sequência de terminais = casamento com a entrada
- Sequência de não-terminais = chamadas de procedimentos
- Escolhas = comandos de alternativas (case e if)

ASD Preditiva Recursiva

- Método formal para gerar os procedimentos
 - Regras de transformação
 - Regras gramaticais → grafos sintáticos
 - Toda regra é mapeada em um grafo
 - Regras de tradução
 - Grafos sintáticos → procedimentos
 1. Reduzir o número de grafos
 2. Escrever um procedimento para cada grafo

ASD Preditiva Recursiva

- Método formal para gerar os procedimentos
 - A sequência

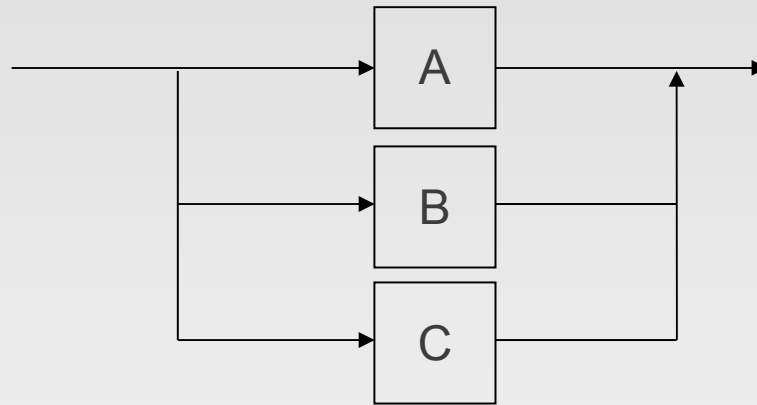


dá origem ao código

```
begin  
  A  
  B  
  C  
end
```

ASD Preditiva Recursiva

- Método formal para gerar os procedimentos
 - A alternativa



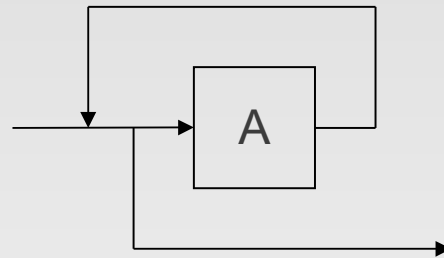
dá origem ao código

```
begin
  se (símbolo está em Primeiro(A)) então A
  senão se (símbolo está em Primeiro(B)) então B
  senão se (símbolo está em Primeiro(C)) então C
end
```

→ onde símbolo é a categoria do token atual

ASD Preditiva Recursiva

- Método formal para gerar os procedimentos
 - A repetição

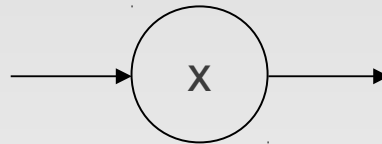


dá origem ao código

```
begin
  enquanto (símbolo está em Primeiro(A)) faça
    A
end
```

ASD Preditiva Recursiva

- Método formal para gerar os procedimentos
 - O terminal

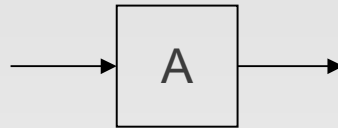


dá origem ao código

```
begin
  se (símbolo=x)
  então obter_simbolo
  senão ERRO
end
```

ASD Preditiva Recursiva

- Método formal para gerar os procedimentos
 - O não-terminal



dá origem ao código

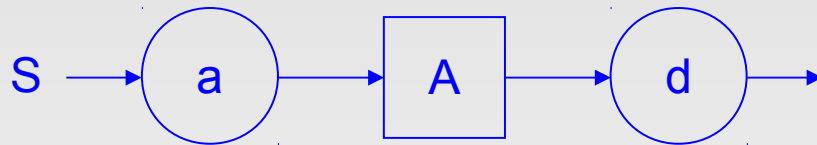
```
begin  
  A  
end
```

ASD Preditiva Recursiva

- Exemplo
 - Escreva os procedimentos correspondentes para
$$\begin{aligned} \langle S \rangle &::= a \langle A \rangle d \\ \langle A \rangle &::= c \langle A \rangle \mid e \langle B \rangle \\ \langle B \rangle &::= f \mid g \end{aligned}$$

ASD Preditiva Recursiva

- Exemplo
 - Escreva os procedimentos correspondentes para $\langle S \rangle ::= a \langle A \rangle d$



```
procedimento S
begin
  se (simbolo='a') então
  begin
    obter_simbolo
    A
    se (simbolo='d')
      então obter_simbolo
      senão ERRO
    end se
  senão ERRO
  end
```

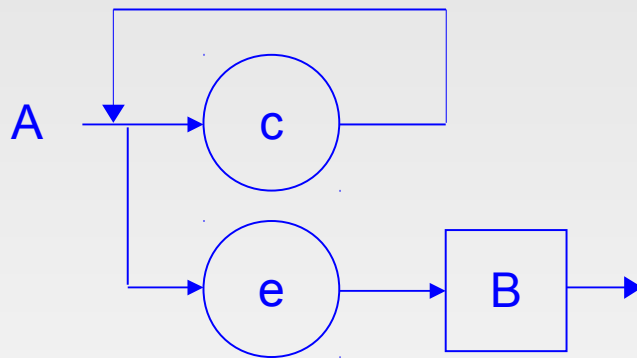
ASD Preditiva Recursiva

- Exemplo

- Escreva os procedimentos correspondentes para

$\langle S \rangle ::= a\langle A \rangle d$

$\langle A \rangle ::= c\langle A \rangle \mid e\langle B \rangle$



```
procedimento A
begin
  se (simbolo='c') então
    begin
      obter_simbolo
      A
    end se
  senão se (simbolo='e') então
    begin
      obter_simbolo
      B
    end se
  senão ERRO
end
```

ASD Preditiva Recursiva

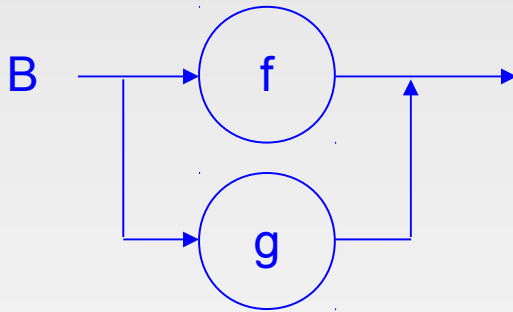
- Exemplo

- Escreva os procedimentos correspondentes para

$\langle S \rangle ::= a\langle A \rangle d$

$\langle A \rangle ::= c\langle A \rangle \mid e\langle B \rangle$

$\langle B \rangle ::= f \mid g$



procedimento B

begin

se (simbolo='f') ou (simbolo='g')

então obter_simbolo

senão ERRO

end

ASD Preditiva Recursiva

- Exemplo

- Escreva os procedimentos correspondentes para

$\langle S \rangle ::= a\langle A \rangle d$

$\langle A \rangle ::= c\langle A \rangle \mid e\langle B \rangle$

$\langle B \rangle ::= f \mid g$

```
procedimento S
begin
  se (simbolo='a') então
    begin
      obter_simbolo
      A
      se (simbolo='d')
        então obter_simbolo
        senão ERRO
      end se
    senão ERRO
  end
end
```

```
procedimento A
begin
  se (simbolo='c') então
    begin
      obter_simbolo
      A
    end se
  senão se (simbolo='e') então
    begin
      obter_simbolo
      B
    end se
  senão ERRO
end
```

```
procedimento B
begin
  se (simbolo='f') ou (simbolo='g')
    então obter_simbolo
  senão ERRO
end
```

```
procedimento ASD
begin
  obter_simbolo
  S
  se (terminou_cadeia)
    então SUCESSO
  senão ERRO
end
```

ASD Preditiva Recursiva

Qual seria o resultado da chamada
ASD('acefd')?

- Exemplo

- Escreva os procedimentos correspondentes para

$\langle S \rangle ::= a\langle A \rangle d$

$\langle A \rangle ::= c\langle A \rangle \mid e\langle B \rangle$

$\langle B \rangle ::= f \mid g$

```
procedimento S
begin
  se (simbolo='a') então
    begin
      obter_simbolo
      A
      se (simbolo='d')
        então obter_simbolo
        senão ERRO
      end se
    senão ERRO
  end
end
```

```
procedimento A
begin
  se (simbolo='c') então
    begin
      obter_simbolo
      A
    end se
  senão se (simbolo='e') então
    begin
      obter_simbolo
      B
    end se
  senão ERRO
end
```

```
procedimento B
begin
  se (simbolo='f') ou (simbolo='g')
    então obter_simbolo
  senão ERRO
end
```

```
procedimento ASD
begin
  obter_simbolo
  S
  se (terminou_cadeia)
    então SUCESSO
  senão ERRO
end
```

ASD Preditiva Recursiva

Qual seria o resultado da chamada
ASD('aecfd')?

- Exemplo

- Escreva os procedimentos correspondentes para

$\langle S \rangle ::= a\langle A \rangle d$

$\langle A \rangle ::= c\langle A \rangle \mid e\langle B \rangle$

$\langle B \rangle ::= f \mid g$

```
procedimento S
begin
  se (simbolo='a') então
    begin
      obter_simbolo
      A
      se (simbolo='d')
        então obter_simbolo
        senão ERRO
      end se
    senão ERRO
  end
end
```

```
procedimento A
begin
  se (simbolo='c') então
    begin
      obter_simbolo
      A
    end se
  senão se (simbolo='e') então
    begin
      obter_simbolo
      B
    end se
  senão ERRO
end
```

```
procedimento B
begin
  se (simbolo='f') ou (simbolo='g')
    então obter_simbolo
  senão ERRO
end
```

```
procedimento ASD
begin
  obter_simbolo
  S
  se (terminou_cadeia)
    então SUCESSO
  senão ERRO
end
```

ASD Preditiva Recursiva

Qual seria o resultado da chamada
ASD('acefda')?

- Exemplo

- Escreva os procedimentos correspondentes para

$\langle S \rangle ::= a\langle A \rangle d$

$\langle A \rangle ::= c\langle A \rangle \mid e\langle B \rangle$

$\langle B \rangle ::= f \mid g$

```
procedimento S
begin
  se (simbolo='a') então
    begin
      obter_simbolo
      A
      se (simbolo='d')
        então obter_simbolo
        senão ERRO
      end se
    senão ERRO
  end
end
```

```
procedimento A
begin
  se (simbolo='c') então
    begin
      obter_simbolo
      A
    end se
  senão se (simbolo='e') então
    begin
      obter_simbolo
      B
    end se
  senão ERRO
end
```

```
procedimento B
begin
  se (simbolo='f') ou (simbolo='g')
    então obter_simbolo
  senão ERRO
end
```

```
procedimento ASD
begin
  obter_simbolo
  S
  se (terminou_cadeia)
    então SUCESSO
  senão ERRO
end
```

ASD Preditiva Recursiva

- Exercício
 - Para a gramática a seguir teríamos quais procedimentos?
 $\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$
 $\langle \text{soma} \rangle ::= + \mid -$
 $\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$
 $\langle \text{mult} \rangle ::= *$
 $\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

ASD Preditiva Recursiva

■ Exercício

- Para a gramática a seguir teríamos quais procedimentos?

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Regras gramaticais \rightarrow grafos sintáticos (2 maneiras)

1. Usar versão LL(1) da gramática

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \langle \text{exp}' \rangle$

$\langle \text{exp}' \rangle ::= \langle \text{soma} \rangle \langle \text{exp} \rangle \mid \epsilon$

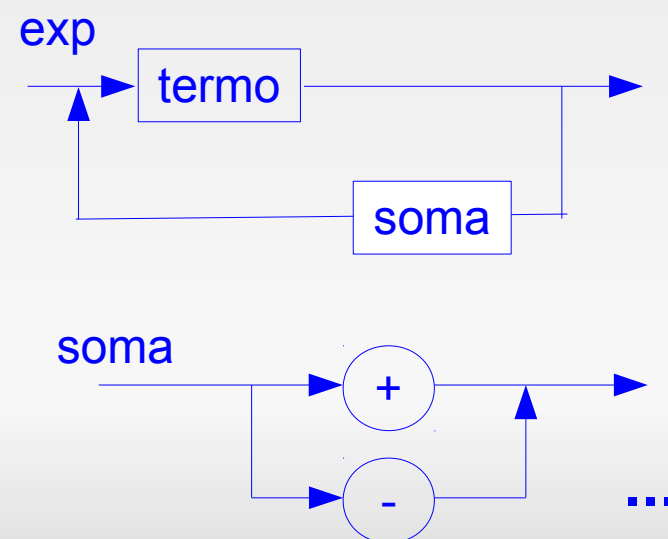
$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \langle \text{termo}' \rangle$

$\langle \text{termo}' \rangle ::= \langle \text{mult} \rangle \langle \text{termo} \rangle \mid \epsilon$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$



ASD Preditiva Recursiva

■ Exercício

- Para a gramática a seguir teríamos quais procedimentos?

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Regras gramaticais \rightarrow grafos sintáticos (2 maneiras)

2. Converter regras para a notação BNF estendida (EBNF¹)

$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \{ \langle \text{soma} \rangle \langle \text{termo} \rangle \}$

$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \{ \langle \text{mult} \rangle \langle \text{fator} \rangle \}$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Transforma regras do tipo
 $\langle A \rangle ::= \langle A \rangle \alpha \mid \beta$
em regras não-recursivas
 $\langle A \rangle ::= \beta \alpha^*$

¹ Extended Backus–Naur Form (EBNF)

ASD Preditiva Recursiva

■ Exercício

- Para a gramática a seguir teríamos quais procedimentos?

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Regras gramaticais \rightarrow grafos sintáticos (2 maneiras)

2. Converter regras para a notação BNF estendida (EBNF¹)

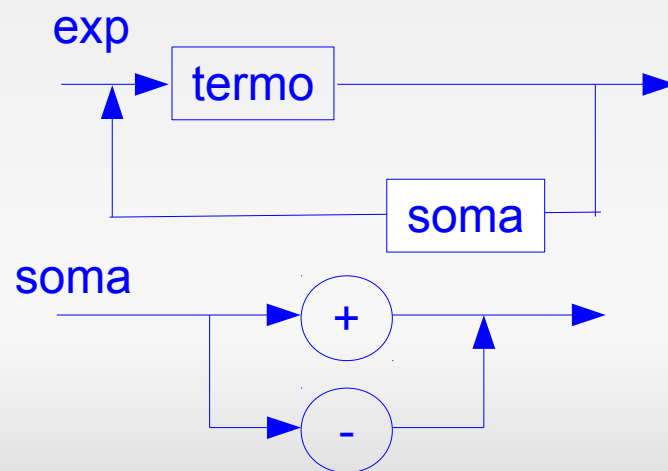
$\langle \text{exp} \rangle ::= \langle \text{termo} \rangle \{ \langle \text{soma} \rangle \langle \text{termo} \rangle \}$

$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{fator} \rangle \{ \langle \text{mult} \rangle \langle \text{fator} \rangle \}$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$



¹ Extended Backus–Naur Form (EBNF)

ASD Preditiva Recursiva

■ Exercício

- Para a gramática a seguir teríamos quais procedimentos?

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

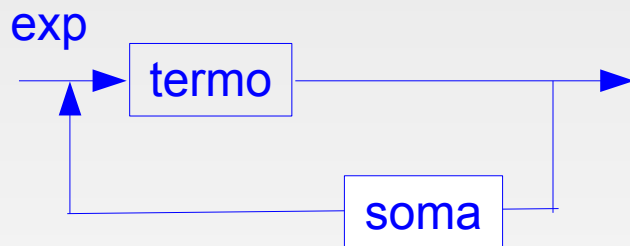
$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Grafos sintáticos → procedimentos



```
procedimento exp
begin
  termo
  se (símbolo está em Primeiro(soma)) então
  begin
    soma
    exp
  end se
end
```

ASD Preditiva Recursiva

■ Exercício

- Para a gramática a seguir teríamos quais procedimentos?

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

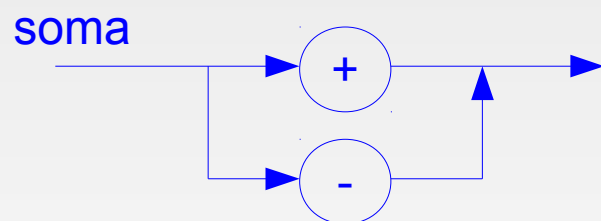
$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Grafos sintáticos → procedimentos



procedimento *soma*

begin

se (símbolo = '+') ou (símbolo = '-')

então obter_simbolo

senão ERRO

end

ASD Preditiva Recursiva

■ Exercício

- Para a gramática a seguir teríamos quais procedimentos?

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

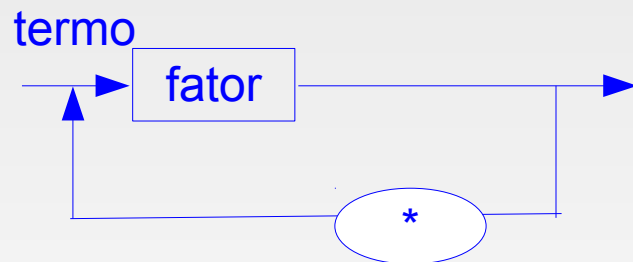
$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Grafos sintáticos → procedimentos



```
procedimento termo
begin
  fator
  se (símbolo = '*') então
  begin
    obter_simbolo
    fator
  end se
end
```

ASD Preditiva Recursiva

Exercício

- Para a gramática a seguir teríamos quais procedimentos?

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

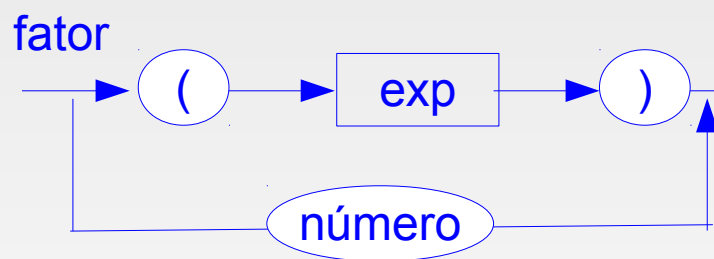
$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

Grafos sintáticos → procedimentos



procedimento *fator*
begin

se (símbolo = '(') então

begin

obter_símbolo

exp

se (símbolo = ')')

então obter_símbolo

senão ERRO

end se

senão se (símbolo = 'número')

então obter_símbolo

senão ERRO

end

ASD Preditiva Recursiva

■ Exercício

- Procedimentos do ASD criado para a gramática:

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

procedimento ASD

begin

obter_simbolo

exp

se (terminou_cadeia)

então SUCESSO

senão ERRO

end

procedimento *exp*

begin

termo

se (símbolo está em

Primeiro(soma)) então

begin

soma

exp

end se

end

procedimento *soma*

begin

se (símbolo = '+') ou (símbolo = '-')

então obter_simbolo

senão ERRO

end

procedimento *termo*

begin

fator

se (símbolo = '*') então

begin

obter_simbolo

fator

end se

end

procedimento *fator*

begin

se (símbolo = '(') então

begin

obter_simbolo

exp

se (símbolo = ')')

então obter_simbolo

senão ERRO

end se

senão se (símbolo = 'número')

então obter_simbolo

senão ERRO

end

ASD Preditiva Recursiva

Qual seria o resultado da chamada
ASD('32+48*9')?

Exercício

- Procedimentos do ASD criado para a gramática:

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

```
procedimento ASD
begin
  obter_simbolo
  exp
  se (terminou_cadeia)
  então SUCESSO
  senão ERRO
end
```

```
procedimento exp
begin
  termo
  se (símbolo está em
    Primeiro(soma)) então
  begin
    soma
    exp
  end se
end
```

```
procedimento soma
begin
  se (símbolo = '+') ou (símbolo = '-')
  então obter_simbolo
  senão ERRO
end
```

```
procedimento termo
begin
  fator
  se (símbolo = '*') então
  begin
    obter_simbolo
    fator
  end se
end
```

```
procedimento fator
begin
  se (símbolo = '(') então
  begin
    obter_simbolo
    exp
    se (símbolo = ')')
    então obter_simbolo
    senão ERRO
  end se
  senão se (símbolo = 'número')
  então obter_simbolo
  senão ERRO
end
```

ASD Preditiva Recursiva

Qual seria o resultado da chamada
ASD('32+-48*9')?

Exercício

- Procedimentos do ASD criado para a gramática:

$\langle \text{exp} \rangle ::= \langle \text{exp} \rangle \langle \text{soma} \rangle \langle \text{termo} \rangle \mid \langle \text{termo} \rangle$

$\langle \text{soma} \rangle ::= + \mid -$

$\langle \text{termo} \rangle ::= \langle \text{termo} \rangle \langle \text{mult} \rangle \langle \text{fator} \rangle \mid \langle \text{fator} \rangle$

$\langle \text{mult} \rangle ::= *$

$\langle \text{fator} \rangle ::= (\langle \text{exp} \rangle) \mid \text{número}$

```
procedimento ASD
begin
  obter_simbolo
  exp
  se (terminou_cadeia)
  então SUCESSO
  senão ERRO
end
```

```
procedimento exp
begin
  termo
  se (símbolo está em
    Primeiro(soma)) então
  begin
    soma
    exp
  end se
end
```

```
procedimento soma
begin
  se (símbolo = '+') ou (símbolo = '-')
  então obter_simbolo
  senão ERRO
end
```

```
procedimento termo
begin
  fator
  se (símbolo = '*') então
  begin
    obter_simbolo
    fator
  end se
end
```

```
procedimento fator
begin
  se (símbolo = '(') então
  begin
    obter_simbolo
    exp
    se (símbolo = ')')
    então obter_simbolo
    senão ERRO
  end se
  senão se (símbolo = 'número')
  então obter_simbolo
  senão ERRO
end
```

ASD Preditiva Recursiva

- Análise sintática descendente preditiva recursiva
 - Vantagem
 - É bastante poderosa
 - Desvantagens
 - É adequada apenas para linguagens pequenas e cuidadosamente projetadas
 - Ainda é *ad hoc*
 - Apresenta problemas com
 - Escolhas que iniciam por não-terminais
 - Solução: construção de conjuntos **Primeiros**
 - Regras que podem dar origem à cadeia vazia
 - Solução: construção de conjuntos **Seguidores**

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

- $$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle^* \langle E \rangle \mid \langle E \rangle^{**} \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$$

1. Transforme-a em LL(1)

2. Faça um ASD preditivo recursivo para a gramática LL(1)

- Desenhe os grafos sintáticos correspondentes
 - Construa os procedimentos recursivos equivalentes

3. Reconheça as cadeias usando o analisador criado

- id ** id
 - id + id id

ASD Preditiva Recursiva

- Exercício
 - Dada a gramática a seguir
$$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle^* \langle E \rangle \mid \langle E \rangle^{**} \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$$
 - 1. Transforme-a em LL(1)
 - A gramática é ambígua?
 - A gramática é recursiva à esquerda?
 - A gramática está fatorada à esquerda?

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid id$

1. Transforme-a em LL(1)

- Removendo a ambiguidade da gramática

- Separa operadores de precedência diferente em não-terminais diferentes
 - Garante que os operadores com precedência $<$ ocorram mais próximos da raiz e os de precedência $>$, mais distantes da raiz
 - Garante a associatividade de operadores de mesma precedência

Sabendo-se que

- $**$ tem maior precedência e é associativo à direita;
- $*$ tem precedência intermediária e é associativo à esquerda;
- $+$ tem menor precedência e é associativo à esquerda

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Gramática resultante após a remoção da ambiguidade

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle T \rangle * \langle F \rangle \mid \langle F \rangle$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- A gramática é recursiva à esquerda?

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle^* \langle E \rangle \mid \langle E \rangle^{**} \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle E \rangle + \langle T \rangle \mid \langle T \rangle$

$\langle T \rangle ::= \langle T \rangle^* \langle F \rangle \mid \langle F \rangle$

$\langle F \rangle ::= \langle P \rangle^{**} \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Removendo a recursão à esquerda

- Transforma regras do tipo

$\langle A \rangle ::= \langle A \rangle \alpha \mid \beta$

em

$\langle A \rangle ::= \beta \langle A' \rangle$

$\langle A' \rangle ::= \alpha \langle A' \rangle \mid \varepsilon$

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid id$

- 1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= id \mid (\langle E \rangle)$

- Gramática resultante após a remoção da recursão à esquerda

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle ** \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- A gramática está fatorada à esquerda?

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle^* \langle E \rangle \mid \langle E \rangle^{**} \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle^{**} \langle F \rangle \mid \langle P \rangle$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Fatorando à esquerda

- Transforma regras do tipo

$\langle A \rangle ::= \alpha \beta \mid \alpha \gamma$

em

$\langle A \rangle ::= \alpha \langle A' \rangle$

$\langle A' \rangle ::= \beta \mid \gamma$

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

$\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle * \langle E \rangle \mid \langle E \rangle ** \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

- 1. Transforme-a em LL(1)

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

$\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

$\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

$\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

$\langle F \rangle ::= \langle P \rangle \langle F' \rangle$

$\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$

$\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

- Gramática resultante após a fatoração à esquerda

ASD Preditiva Recursiva

- Exercício

- Dada a gramática a seguir

- $\langle E \rangle ::= \langle E \rangle + \langle E \rangle \mid \langle E \rangle^* \langle E \rangle \mid \langle E \rangle^{**} \langle E \rangle \mid (\langle E \rangle) \mid \text{id}$

- 1. Transforme-a em LL(1)

- Gramática LL(1) a ser usada no restante do exercício

- $\langle E \rangle ::= \langle T \rangle \langle E' \rangle$

- $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$

- $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$

- $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$

- $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$

- $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$

- $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

ASD Preditiva Recursiva

- Exercício

$$\begin{aligned} \langle E \rangle &::= \langle T \rangle \langle E' \rangle \\ \langle E' \rangle &::= + \langle E \rangle \mid \varepsilon \\ \langle T \rangle &::= \langle F \rangle \langle T' \rangle \\ \langle T' \rangle &::= * \langle T \rangle \mid \varepsilon \\ \langle F \rangle &::= \langle P \rangle \langle F' \rangle \\ \langle F' \rangle &::= ** \langle F \rangle \mid \varepsilon \\ \langle P \rangle &::= \text{id} \mid (\langle E \rangle) \end{aligned}$$

2. Faça um ASD preditivo recursivo para a gramática LL(1)

- Desenhe os grafos sintáticos correspondentes
- Construa os procedimentos recursivos equivalentes

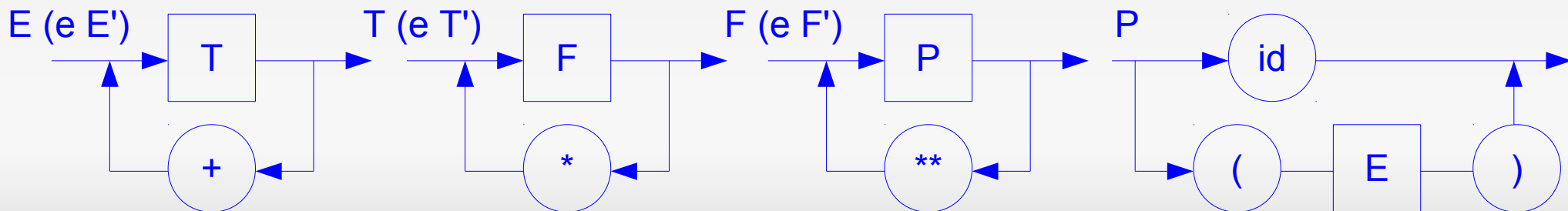
ASD Preditiva Recursiva

Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$
 $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$

2. Faça um ASD preditivo recursivo para a gramática LL(1)

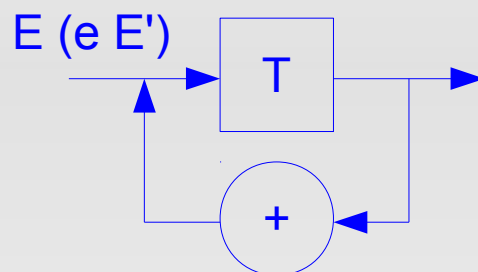
- Desenhe os grafos sintáticos correspondentes



ASD Preditiva Recursiva

Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$
 $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$



```
procedimento E
begin
  T
  se (simbolo='+') então
    begin
      obter_simbolo
      E
    end se
end
```

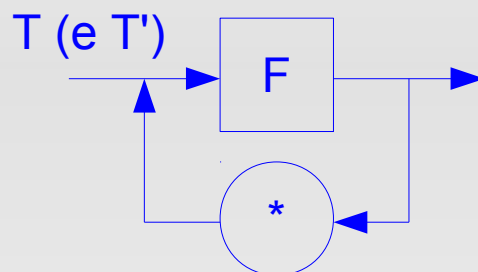
2. Faça um ASD preditivo recursivo para a gramática LL(1)

- Desenhe os grafos sintáticos correspondentes
- Construa os procedimentos recursivos equivalentes

ASD Preditiva Recursiva

Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$
 $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$



```
procedimento T
begin
  faça
    F
    se (simbolo='*')
      então obter_simbolo
    senão fim
  enquanto não fim
end
```

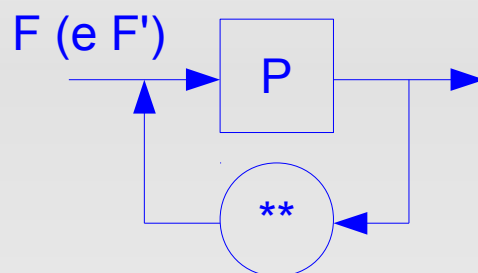
2. Faça um ASD preditivo recursivo para a gramática LL(1)

- Desenhe os grafos sintáticos correspondentes
- Construa os procedimentos recursivos equivalentes

ASD Preditiva Recursiva

Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$
 $\langle P \rangle ::= \text{id} \mid (\langle E \rangle)$



```
procedimento F
begin
  enquanto não fim faça
  begin
    P
    se (simbolo='**')
      então obter_simbolo
    senão fim
  end enquanto
end
```

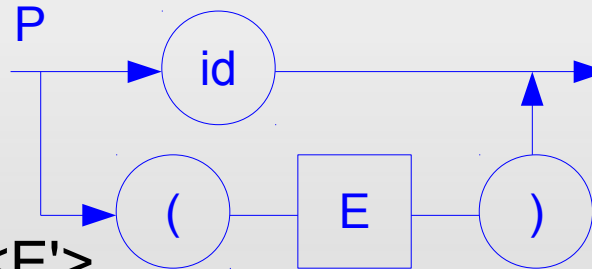
2. Faça um ASD preditivo recursivo para a gramática LL(1)

- Desenhe os grafos sintáticos correspondentes
- Construa os procedimentos recursivos equivalentes

ASD Preditiva Recursiva

Exercício

$\langle E \rangle ::= \langle T \rangle \langle E' \rangle$
 $\langle E' \rangle ::= + \langle E \rangle \mid \varepsilon$
 $\langle T \rangle ::= \langle F \rangle \langle T' \rangle$
 $\langle T' \rangle ::= * \langle T \rangle \mid \varepsilon$
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$
 $\langle F' \rangle ::= ** \langle F \rangle \mid \varepsilon$
 $\langle P \rangle ::= id \mid (\langle E \rangle)$



```
procedimento P
begin
  se (simbolo='id')
  então obter_simbolo
  senão se (simbolo='(')
    então begin
      obter_simbolo
      E
      se (simbolo=')')
      então obter_simbolo
      senão ERRO
    end então
  senão ERRO
end
```

2. Faça um ASD preditivo recursivo para a gramática LL(1)

- Desenhe os grafos sintáticos correspondentes
- Construa os procedimentos recursivos equivalentes

ASD Preditiva Recursiva

- Exercício

$$\begin{aligned} \langle E \rangle &::= \langle T \rangle \langle E' \rangle \\ \langle E' \rangle &::= + \langle E \rangle \mid \varepsilon \\ \langle T \rangle &::= \langle F \rangle \langle T' \rangle \\ \langle T' \rangle &::= * \langle T \rangle \mid \varepsilon \\ \langle F \rangle &::= \langle P \rangle \langle F' \rangle \\ \langle F' \rangle &::= ** \langle F \rangle \mid \varepsilon \\ \langle P \rangle &::= \text{id} \mid (\langle E \rangle) \end{aligned}$$

3. Reconheça as cadeias usando o analisador criado

- id ** id
- id + id id

ASD Preditiva Recursiva

procedimento ASD

```
begin
  obter_simbolo
  E
  se (terminou_cadeia)
    então SUCESSO
  senão ERRO
end
```

$\langle T \rangle \langle E' \rangle$
: $+ \langle E \rangle \mid \epsilon$
 $\langle F \rangle \langle T' \rangle$
: $* \langle T \rangle \mid \epsilon$
 $\langle F \rangle ::= \langle P \rangle \langle F' \rangle$
 $\langle F' \rangle ::= ** \langle F \rangle \mid \epsilon$
 $\langle P \rangle ::= id \mid (\langle E \rangle)$

3. Reconheça as cadeias usando o analisador criado

- $id ** id$
- $id + id id$

procedimento E

```
begin
  T
  se (simbolo='+') então
    begin
      obter_simbolo
      E
    end se
end
```

procedimento T

```
begin
  faça
    F
    se (simbolo='*')
      então obter_simbolo
    senão fim
  enquanto não fim
end
```

procedimento P

```
begin
  se (simbolo='id')
    então obter_simbolo
  senão se (simbolo='(')
    então begin
      obter_simbolo
      E
      se (simbolo=')')
        então obter_simbolo
      senão ERRO
    end então
    senão ERRO
```

end

procedimento F

```
begin
  enquanto não fim faça
    begin
      P
      se (simbolo='**')
        então obter_simbolo
      senão fim
    end enquanto
end
```