Compiladores - Análise Recursiva

Fabio Mascarenhas – 2017.2

http://www.dcc.ufrj.br/~fabiom/comp

Geradores x Reconhecedores

- A definição formal de gramática dá um *gerador* para uma linguagem
- Para análise sintática, precisamos de um *reconhecedor*
- Mas podemos reformular a definição de gramática para dar um reconhecedor, também
- Uma PE-CFG (gramática livre de contexto com expressões de parsing) tem os mesmos conjuntos V, T e P de uma gramática tradicional, mas o conjunto P é uma *função* de não-terminais em *expressões de parsing*
- Podemos ter ou um não-terminal inicial *S* ou uma expressão de parsing inicial *s*

Expressões de Parsing

- Uma expressão de parsing é:
  - Um terminal *a*
  - Um não-terminal *A*
  - Uma *concatenação* de duas expressões *pq*
  - Uma *escolha* entre duas expressões *p*|*q*
- A precedência da concatenação é maior que a da escolha, mas podemos usar parênteses para agrupamento

Reconhecendo uma entrada

- O significado de uma expressão de parsing *p* associada a uma gramática *G*, dada uma entrada qualquer, é dado por uma série de *regras de dedução* que dizem se a expressão *reconhece* um prefixo da entrada

$$G\ p\ xy \to y\ ,\ x \in T^*$$

$$G\ a\ ay \to y$$

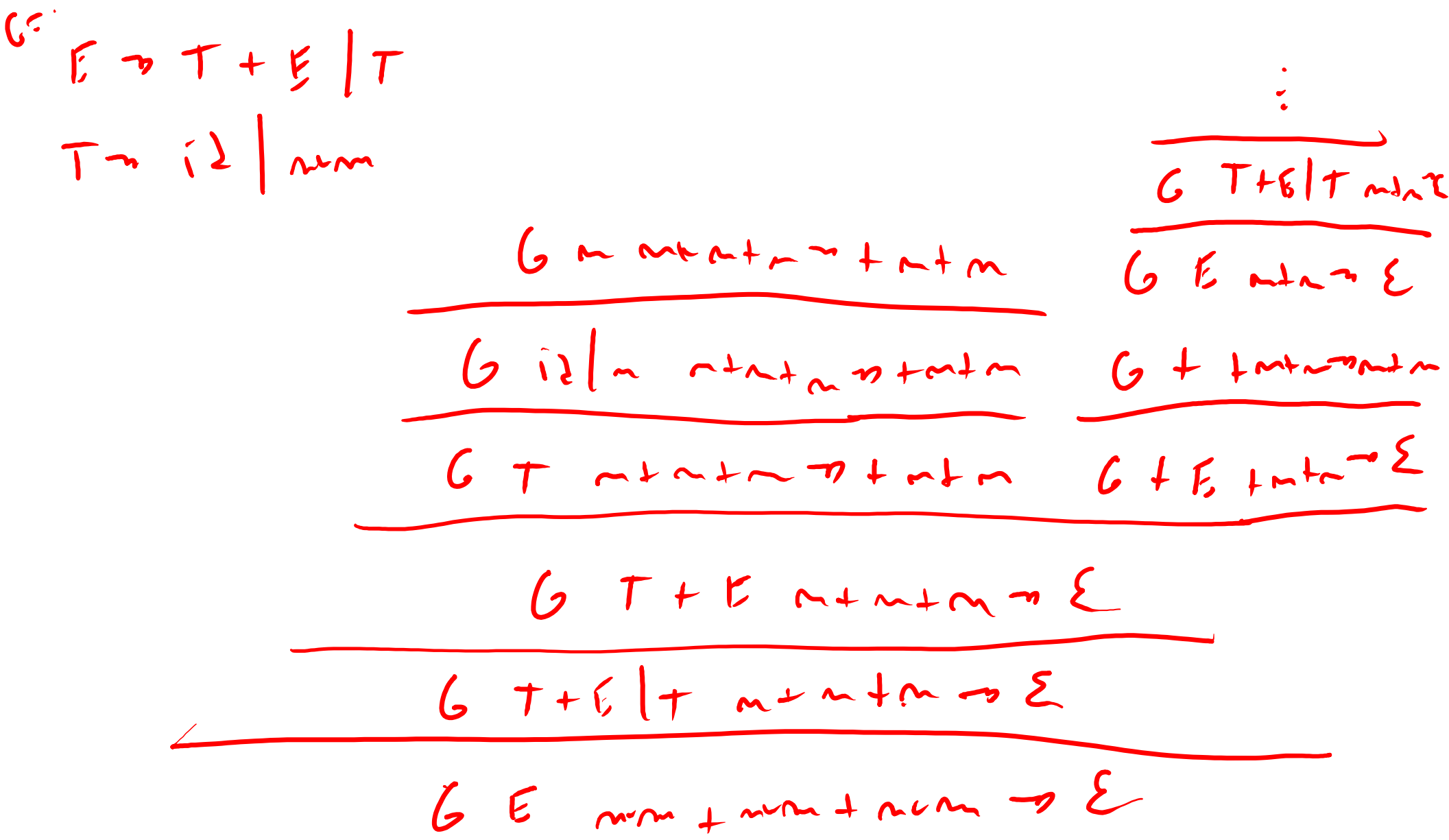
$$\frac{G\ p\ xyz \to yz \qquad G\ q\ yz \to z}{G\ pq\ xyz \to z}$$

$$\frac{G\ G(A)\ xy \to y}{G\ A\ xy \to y}$$

$$\frac{G\ p\ xy \to y}{G\ p|q\ xy \to y} \qquad \frac{G\ q\ xy \to y}{G\ p|q\ xy \to y}$$

$$L(G) = \{ w \in T^* \mid G\ S\ w \to \varepsilon \}$$

Exemplo



Não-determinismo da escolha

- As regras de dedução para a escolha não dizem qual das alternativas escolher: a escolha em uma gramática livre de contexto é *não-determinística*
- Simular não-determinismo em uma implementação real não é difícil, mas não é muito eficiente, e gera problemas de *ambiguidade*
- Todas as técnicas de análise sintática que vamos ver são diferentes maneiras de domar esse não-determinismo
- A primeira técnica, que vamos ver a seguir, reinterpreta a escolha para ser *determinística* e *ordenada*