

Árvores de derivação e ambiguidade

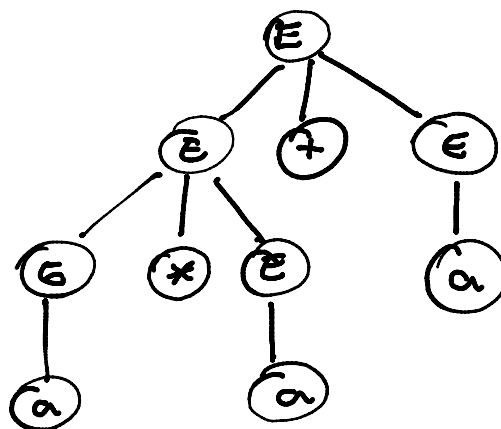
Considere a gramática a seguir que gera expressões aritméticas em uma linguagem de programação hipotética qualquer.

- $E \rightarrow E + E \mid E * E \mid (E) \mid a$

★ Agora considere a palavra (expressão) $a * a + a$. Essa palavra é gerada pela gramática acima. Podemos representar a sequência de derivações que levam a ela através de uma árvore. Essa árvore é chamada de **árvore de derivação**. Ela mostra de forma gráfica, não sequencial, as derivações utilizadas, independentemente da ordem com que as regras da gramática foram escolhidas.

Da mesma forma que em árvores binárias, cada variável ou terminal de uma sentença corresponde a um nó dessa árvore. As regras usadas na expansão de variáveis resultam em filhos do nó correspondente; normalmente, os filhos são inseridos na mesma ordem da regra da esquerda para a direita. Segue o exemplo da árvore de derivação que leva à expressão acima.

📌 Exemplo: Segue a árvore de derivação para a palavra $a * a + a$.



$$E \Rightarrow E + E$$

$$\Rightarrow E * E + E$$

$$\Rightarrow E * E + a$$

$$\Rightarrow a * E + a$$

$$\Rightarrow a * a + a$$

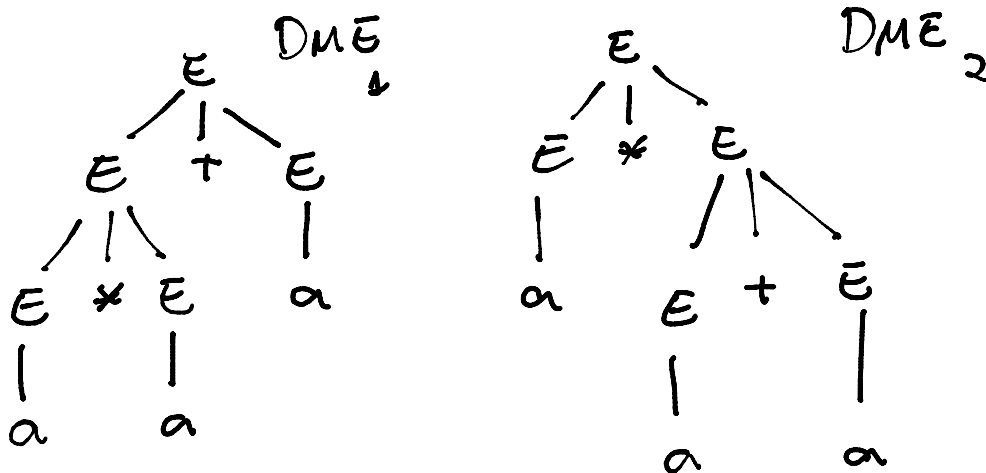
Note que uma derivação possui apenas uma árvore de derivação associada. Contudo, uma árvore de derivação pode estar associada a várias derivações.

★ As árvores de derivação são instrumentos importantes na construção de linguagens de programação porque podemos associar semântica à estrutura sintática (por exemplo, geração de código para calcular aquela expressão; verificação de tipos, etc.). A ordem com que as regras são aplicadas, ou seja, como a derivação é executada interfere diretamente na semântica relacionada. Logo, se existirem duas formas distintas de se derivar uma mesma sentença, então é introduzida uma certa ambiguidade à linguagem. Dizemos que uma GLC é **ambígua** quando existe mais de uma árvore de derivação para alguma sentença que ela gere.

Para definir mais formalmente o conceito de ambiguidade, vamos considerar duas formas de se derivar uma sentença: uma derivação mais à esquerda (DME); e uma derivação mais à direita (DMD). Nessas derivações, a variável mais à esquerda/direita é sempre expandida antes das

subsequentes. Em gramáticas não-ambíguas, existe uma única DME e uma única DMD. Portanto, uma GLC é ambígua se, e somente se, existirem pelo menos duas DMEs/DMDs distintas para uma mesma sentença.

Exemplo: A gramática do exemplo inicial é ambígua.



Observe que falamos que uma gramática é ambígua. Isso porque a linguagem pode ser gerada por outra gramática sem ambiguidade. A gramática abaixo gera a mesma linguagem de expressões da anterior, porém não é ambígua.

- $E \rightarrow E + T \mid T$
- $T \rightarrow T * F \mid F$
- $F \rightarrow (E) \mid a$

Contudo, existem linguagens para as quais só existem gramáticas ambíguas. Essas linguagens são chamadas de linguagens inerentemente ambíguas. Um exemplo é a linguagem $L = \{a^m b^n c^k \mid m = n \vee n = k\}$. Qualquer GLC será inevitavelmente ambígua ao gerar palavras $a^n b^n c^n$.