- write this BNF grammar using the standard context-free grammar notation

$$
\begin{aligned}
E &::= T\ [\ +\ T\ ]\ \ldots \\
T &::= F\ [\ *\ F\ ]\ \ldots \\
F &::= \text{“(”}\ E\ \text{“)”} \mid x \mid y \mid z
\end{aligned}
$$

Parsing

- *parsing* refers to the process of finding a derivation for a string *w* using the rules of grammar *G*, or showing that no such derivation exists
  - that $w \in L(G)$ is important for establishing that *w* is syntactically correct
  - finding a derivation for *w* is an important first step in semantic analysis

Parsing

- find a derivation of *x*+*y***z* in the grammar shown

$$
\begin{aligned}
E &\longrightarrow E + E \\
E &\longrightarrow E * E \\
E &\longrightarrow (E) \\
E &\longrightarrow x \\
E &\longrightarrow y \\
E &\longrightarrow z
\end{aligned}
$$

$$
\begin{aligned}
E &\Longrightarrow E + E \\
&\Longrightarrow E + E * E \\
&\Longrightarrow E + y * E \\
&\Longrightarrow x + y * E \\
&\Longrightarrow x + y * z
\end{aligned}
$$

$$
\begin{aligned}
E &\Longrightarrow E + E \\
&\Longrightarrow x + E \\
&\Longrightarrow x + E * E \\
&\Longrightarrow x + y * E \\
&\Longrightarrow x + y * z
\end{aligned}
$$

$$
\begin{aligned}
E &\Longrightarrow E * E \\
&\Longrightarrow E + E * E \\
&\Longrightarrow x + E * E \\
&\Longrightarrow x + y * E \\
&\Longrightarrow x + y * z
\end{aligned}
$$

Parsing

- in order to have an efficient parsing algorithm for a grammar, we need
  - an *unambiguous* grammar – only a single derivation is possible
  - a *deterministic* process – only one rule can be applicable in a given step

Left Derivations

- in general, there are many possible derivations for a string in $L(G)$
- in a *left derivation*, always replace the leftmost non-terminal in the next step

$$\begin{array}{l} E \longrightarrow E+E \\ E \longrightarrow E*E \\ E \longrightarrow (E) \\ E \longrightarrow x \\ E \longrightarrow y \\ E \longrightarrow z \end{array}$$

$$\begin{array}{l} E \Longrightarrow E+E \\ \quad \Longrightarrow E+E*E \\ \quad \Longrightarrow E+y*E \\ \quad \Longrightarrow x+y*E \\ \quad \Longrightarrow x+y*z \end{array}$$

$$\begin{array}{l} E \Longrightarrow E+E \\ \quad \Longrightarrow x+E \\ \quad \Longrightarrow x+E*E \\ \quad \Longrightarrow x+y*E \\ \quad \Longrightarrow x+y*z \end{array}$$

- any string that has a derivation has a left derivation
  - the same rules are applied, it's just the order that is changed

Ambiguous Grammars

- a context-free grammar G is *ambiguous* if there is a string $w \in L(G)$ such that w has more than one left derivation according to G

$$\begin{array}{l} E \longrightarrow E+E \\ E \longrightarrow E*E \\ E \longrightarrow (E) \\ E \longrightarrow x \\ E \longrightarrow y \\ E \longrightarrow z \end{array}$$

$$\begin{array}{l} E \Longrightarrow E+E \\ \quad \Longrightarrow x+E \\ \quad \Longrightarrow x+E*E \\ \quad \Longrightarrow x+y*E \\ \quad \Longrightarrow x+y*z \end{array}$$

$$\begin{array}{l} E \Longrightarrow E*E \\ \quad \Longrightarrow E+E*E \\ \quad \Longrightarrow x+E*E \\ \quad \Longrightarrow x+y*E \\ \quad \Longrightarrow x+y*z \end{array}$$

1. Show that each of the following grammars is ambiguous by finding a string that has two left derivations according to the grammar:

a)
$$\begin{array}{l} S \longrightarrow SS \\ S \longrightarrow aSb \\ S \longrightarrow bSa \\ S \longrightarrow \varepsilon \end{array}$$

b)
$$\begin{array}{l} S \longrightarrow ASb \\ S \longrightarrow \varepsilon \\ A \longrightarrow aA \\ A \longrightarrow a \end{array}$$

Unambiguous Grammars

$$\begin{array}{l} E \longrightarrow E+E \\ E \longrightarrow E*E \\ E \longrightarrow (E) \\ E \longrightarrow x \\ E \longrightarrow y \\ E \longrightarrow z \end{array}$$

$$\begin{array}{l} E \Longrightarrow E+E \\ \quad \Longrightarrow x+E \\ \quad \Longrightarrow x+E*E \\ \quad \Longrightarrow x+y*E \\ \quad \Longrightarrow x+y*z \end{array}$$

$$\begin{array}{l} E \Longrightarrow E*E \\ \quad \Longrightarrow E+E*E \\ \quad \Longrightarrow x+E*E \\ \quad \Longrightarrow x+y*E \\ \quad \Longrightarrow x+y*z \end{array}$$

- precedence rules govern the interpretation of an expression like x+y*z in order to resolve the ambiguity in meaning
  - since we are usually parsing something as a step towards determining its meaning, the grammar should also be unambiguous and reflect the proper interpretation

$$\begin{array}{l} E ::= T\ [\ +\ T\ ]\ldots \\ T ::= F\ [\ *\ F\ ]\ldots \\ F ::= \text{“(”}\ E\ \text{“)”} \mid x \mid y \mid z \end{array}$$

- write this BNF grammar using the standard context-free grammar notation

$$E ::= T \ [+ T \ ] \ldots$$
$$T ::= F \ [* F \ ] \ldots$$
$$F ::= \text{“(”} \ E \ \text{“)”} \mid x \mid y \mid z$$

$$\begin{array}{l} E \longrightarrow TA \\ A \longrightarrow +TA \\ A \longrightarrow \varepsilon \\ T \longrightarrow FB \\ B \longrightarrow *FB \\ B \longrightarrow \varepsilon \\ F \longrightarrow (E) \\ F \longrightarrow x \\ F \longrightarrow y \\ F \longrightarrow z \end{array}$$

- give a left derivation for *x+y*z*

$$\begin{array}{l} E \longrightarrow TA \\ A \longrightarrow +TA \\ A \longrightarrow \varepsilon \\ T \longrightarrow FB \\ B \longrightarrow *FB \\ B \longrightarrow \varepsilon \\ F \longrightarrow (E) \\ F \longrightarrow x \\ F \longrightarrow y \\ F \longrightarrow z \end{array}$$

$$\begin{array}{ll} E & \Longrightarrow TA \\ & \Longrightarrow FBA \\ & \Longrightarrow xBA \\ & \Longrightarrow xA \\ & \Longrightarrow x+TA \\ & \Longrightarrow x+FBA \\ & \Longrightarrow x+yBA \\ & \Longrightarrow x+y*FBA \\ & \Longrightarrow x+y*zBA \\ & \Longrightarrow x+y*zA \\ & \Longrightarrow x+y*z \end{array}$$

LL(1) Grammars

- in an unambiguous grammar *G*, at each step in a left derivation there's only one production that can be applied that will lead to a correct derivation of *w*
- *G* is an *LL(1) grammar* if that one production can be determined by (only) looking ahead to the next symbol in *w*
  - *LL*(*1*) means *w* is read *L*eft-to-right and a *L*eft derivation is constructed by looking ahead at most *1* character in *w*
  - LL(1) grammars are nice in practice because it is straightforward to write a computer program to parse them

$$\begin{array}{l} E \longrightarrow TA \\ A \longrightarrow +TA \\ A \longrightarrow \varepsilon \\ T \longrightarrow FB \\ B \longrightarrow *FB \\ B \longrightarrow \varepsilon \\ F \longrightarrow (E) \\ F \longrightarrow x \\ F \longrightarrow y \\ F \longrightarrow z \end{array}$$

2. Consider the string $z+(x+y)*x$. Find a left derivation of this string according to each of the grammars G_1, G_2, and G_3, as given in this section.

$$\begin{array}{l} E \longrightarrow E+E \\ E \longrightarrow E*E \\ E \longrightarrow (E) \\ E \longrightarrow x \\ E \longrightarrow y \\ E \longrightarrow z \end{array}$$

$$\begin{array}{l} E \longrightarrow TA \\ A \longrightarrow +TA \\ A \longrightarrow \varepsilon \\ T \longrightarrow FB \\ B \longrightarrow *FB \\ B \longrightarrow \varepsilon \\ F \longrightarrow (E) \\ F \longrightarrow x \\ F \longrightarrow y \\ F \longrightarrow z \end{array}$$

$$\begin{array}{l} E \longrightarrow E+T \\ E \longrightarrow T \\ T \longrightarrow T*F \\ T \longrightarrow F \\ F \longrightarrow (E) \\ F \longrightarrow x \\ F \longrightarrow y \\ F \longrightarrow z \end{array}$$

- find a left derivation for (x+y)*z

$$
\begin{aligned}
E &\longrightarrow E + T \\
E &\longrightarrow T \\
T &\longrightarrow T * F \\
T &\longrightarrow F \\
F &\longrightarrow (E) \\
F &\longrightarrow x \\
F &\longrightarrow y \\
F &\longrightarrow z
\end{aligned}
$$

LR(1) Grammars

- not all unambiguous context-free grammars are LL(1)
- *G* is an *LR(1) grammar* if it is always possible to tell which rule to apply at each step of the *right* derivation by (only) looking ahead to the next symbol in *w*