Contents

Context-Free Grammar

- A context-free grammar is specified by:
  - A finite set of terminal symbols (input language)
  - A finite set of nonterminal symbols
  - A finite set of production sof the form
    - <A>$\rightarrow \alpha$
    - Where <A> is a nonterminal symbol and α is a sequence (possibly the null sequence) of terminal and non terminal symbols
  - A starting nonterminal symbol
- A context-free language is a set of all terminal strings that can be derived from the starting symbol of a context-free grammar

1. <s>$\rightarrow$a<A><B>c
2. <A>$\rightarrow$b<A>
3. <A>$\rightarrow \varepsilon$
4. <B>$\rightarrow$b
5. <B>$\rightarrow \varepsilon$

Derivations

- $\Rightarrow$: Derives
- $\overset{*}{\Rightarrow}$: Derives in 0 or more steps
- $\overset{+}{\Rightarrow}$: Derives in 1 or more steps

- $\overset{N}{\Rightarrow}$: Derives with N step

  Derive string `abc`

- $\texttt{<s>} \overset{1}{\Rightarrow} \texttt{a<A><B>c}$
- $\texttt{a<A><B>c} \overset{2}{\Rightarrow} \texttt{ab<A><B>c}$
- $\texttt{ab<B><B>c} \overset{3}{\Rightarrow} \texttt{ab<B>c}$
- $\texttt{ab<B>c} \overset{5}{\Rightarrow} \texttt{abc}$
- **Leftmost Derivation**
- $\texttt{<s>} \overset{1}{\Rightarrow} \texttt{a<A><B>c}$
- $\texttt{a<A><B>c} \overset{3}{\Rightarrow} \texttt{a<B>c}$
- $\texttt{a<B>c} \overset{4}{\Rightarrow} \texttt{abc}$
- **Leftmost Derivation**
- Can make a string in two leftmost derivations the language is ambiguous
- $\texttt{<s>} \overset{1}{\Rightarrow} \texttt{a<A><B>c}$
- $\texttt{a<A><B>c} \overset{5}{\Rightarrow} \texttt{a<A>c}$
- $\texttt{a<A>c} \overset{2}{\Rightarrow} \texttt{ab<A>c}$
- $\texttt{ab<A>c} \overset{3}{\Rightarrow} \texttt{abc}$
- **Rightmost derivaion*

  If there is only one leftmost and rightmost derivation tree for each string then the language is unambiguous

Grammar for Arithmetic Expressions

1. $\texttt{<E>}\rightarrow\texttt{<E>+<T>}$
2. $\texttt{<E>}\rightarrow\texttt{<T>}$
3. $\texttt{<T>}\rightarrow\texttt{<T>*<P>}$
4. $\texttt{<T>}\rightarrow\texttt{<P>}$
5. $\texttt{<P>}\rightarrow\texttt{(<E>)}$
6. $\texttt{<P>}\rightarrow\texttt{const}$

   1+2*3+4

- <E> - Expression
- <T> - Term
- <P> - Primary
- <F> - Factor (used for exponentiation)

Starting terminal is <E>

- <E> $\stackrel{1}{\Rightarrow}$ <E>+<T>
- <E>+<T> $\stackrel{1}{\Rightarrow}$ <E>+<T>+<T>
- <E>+<T>+<T> $\stackrel{2}{\Rightarrow}$ <T>+<T>+<T>
- <T>+<T>+<T> $\stackrel{4}{\Rightarrow}$ <P>+<T>+<T>
- <P>+<T>+<T> $\stackrel{6}{\Rightarrow}$ $const_1$+<T>+<T>
- $const_1$+<T>+<T> $\stackrel{3}{\Rightarrow}$ $const_1$+<T>*<P>+<T>
- $const_1$+<T>*<P>+<T>$\stackrel{*}{\Rightarrow}$ $const_1 + const_2 * const_3 + const_4$

$const_1$ from the lexical analyser

- Convert derivation tree to the lexical values
- Tree shows the structure of the expression

Another Grammer for Arithmetic Expressions

1. <E> $\rightarrow$ <E>+<T>
2. <E> $\rightarrow$ <T>

- <E> $\Rightarrow$ <T>
- <E> $\Rightarrow$ <E>+<T> $=$ <E>+<T> $\Rightarrow$ <T>+<T>

Replace them with the following

1. <E> $\rightarrow$ <T><E-list>
2. <E-list> $\rightarrow$ +<T><E-list>
3. <E-list> $\rightarrow \varepsilon$

- <E> $\Rightarrow$ <T><E-list>
- <T><E-list> $\stackrel{3}{\Rightarrow}$ <T>
- <T> $\stackrel{2}{\Rightarrow}$ <T>+<T><E-list>
- <T>+<T><E-list> $\Rightarrow$ <T>+<T>+<T><E-list>
- <T>+<T>+<T>+<E-list> $\stackrel{3}{\Rightarrow}$ <T>+<T>+<T>

This is an e-list grammar

E-list Grammar

Are always leftmost derived

1. <E> $\rightarrow$ <T><E-list>
2. <E-list> $\rightarrow$ +<T><E-list>
3. <E-list> $\rightarrow \varepsilon$
4. <T> $\rightarrow$ <P><T-list>
5. <T-list> $\rightarrow$ *<P><T-list>
6. <T-list> $\rightarrow \varepsilon$
7. <P> $\rightarrow$ (<E>)
8. <P> $\rightarrow$ const

1+2*3+4

- <E> $\overset{1}{\Rightarrow}$ <T><E-list>
- <T><E-list> $\overset{4}{\Rightarrow}$ <P><T-list><E-list>
- <P><T-list><E-list> $\overset{8}{\Rightarrow}$ $const_1$<T-list><E-list>
- $const_1$<T-list><E-list> $\overset{6}{\Rightarrow}$ $const_1$<E-list>
- $const_1$<E-list> $\overset{7}{\Rightarrow}$ $const_1$+<T><E-list>
- $const_1$+<T>+<E-list> $\overset{*}{\Rightarrow}$ $const_1 + const_2 * const_3 + const_4$

Consider

1. <E>$\rightarrow$<E><op><T>
2. <E>$\rightarrow$<T>
3. <T>$\rightarrow$ident
4. <op>$\rightarrow$+
5. <op>$\rightarrow$or