Automata and Formal Languages (6)

Email: christian.urban at kcl.ac.uk
Office: S1.27 (1st floor Strand Building)
Slides: KEATS (also home work is there)

Grammars

A (context-free) grammar G consists of

- a finite set of nonterminal symbols (upper case)
- a finite terminal symbols or tokens (lower case)
- a start symbol (which must be a nonterminal)
- a set of rules

$$A \to \text{rhs}$$

where rhs are sequences involving terminals and nonterminals.

Grammars

A (context-free) grammar G consists of

- a finite set of nonterminal symbols (upper case)
- a finite terminal symbols or tokens (lower case)
- a start symbol (which must be a nonterminal)
- a set of rules

$$A \to \text{rhs}$$

where rhs are sequences involving terminals and nonterminals.

We can also allow rules

$$A \to \text{rhs}_1 | \text{rhs}_2 | \ldots$$

Palindromes

$$
\begin{array}{lcl}
S & \rightarrow & \epsilon \\
S & \rightarrow & a \cdot S \cdot a \\
S & \rightarrow & b \cdot S \cdot b
\end{array}
$$

Palindromes

$$\begin{array}{lcl} S & \rightarrow & \epsilon \\ S & \rightarrow & a \cdot S \cdot a \\ S & \rightarrow & b \cdot S \cdot b \end{array}$$

or

$$S \rightarrow \epsilon \mid a \cdot S \cdot a \mid b \cdot S \cdot b$$

Arithmetic Expressions

$$
\begin{array}{lcl}
E & \rightarrow & num\_token \\
E & \rightarrow & E \cdot + \cdot E \\
E & \rightarrow & E \cdot - \cdot E \\
E & \rightarrow & E \cdot * \cdot E \\
E & \rightarrow & (\cdot E \cdot)
\end{array}
$$

Arithmetic Expressions

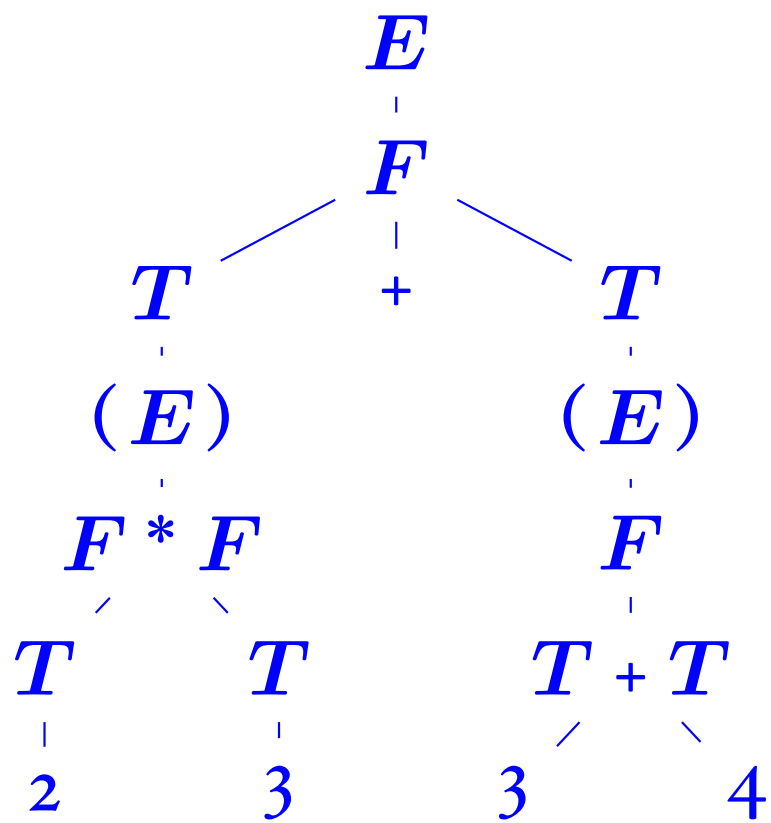
$$
\begin{array}{lcl}
E & \rightarrow & num\_token \\
E & \rightarrow & E \cdot + \cdot E \\
E & \rightarrow & E \cdot - \cdot E \\
E & \rightarrow & E \cdot * \cdot E \\
E & \rightarrow & (\cdot E \cdot)
\end{array}
$$

```
1 + 2 * 3 + 4
```

Parse Trees

$$
\begin{array}{lcl}
E & \rightarrow & F \mid F \cdot * \cdot F \\
F & \rightarrow & T \mid T \cdot + \cdot T \mid T \cdot - \cdot T \\
T & \rightarrow & num\_token \mid (\cdot E \cdot)
\end{array}
$$

(2*3)+(3+4)



Ambiguous Grammars

A grammar is ambiguous if there is a string that has at least parse trees.

$$
\begin{array}{lcl}
E & \rightarrow & num\_token \\
E & \rightarrow & E \cdot + \cdot E \\
E & \rightarrow & E \cdot - \cdot E \\
E & \rightarrow & E \cdot * \cdot E \\
E & \rightarrow & (\cdot E \cdot)
\end{array}
$$

```
1 + 2 * 3 + 4
```

Chomsky Normal Form

All rules must be of the form

$$A \to a$$

or

$$A \to B \cdot C$$

CYK Algorithm

$$
\begin{array}{lcl}
S & \rightarrow & N \cdot P \\
P & \rightarrow & V \cdot N \\
N & \rightarrow & N \cdot N \\
N & \rightarrow & \texttt{students} \mid \texttt{Jeff} \mid \texttt{geometry} \mid \texttt{trains} \\
V & \rightarrow & \texttt{trains}
\end{array}
$$

```
Jeff trains geometry students
```

CYK Algorithm

- runtime is $O(n^3)$
- grammars need to be transferred into CNF