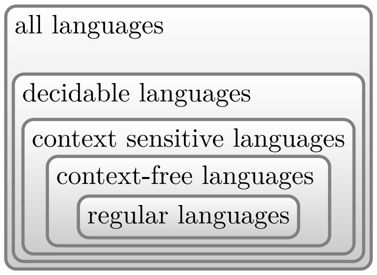
Handout 6

While regular expressions are very useful for lexing and for recognising many patterns (like email addresses), they have their limitations. For example there is no regular expression that can recognise the language $a^n b^n$. Another example is the language of well-parenthesised expressions. In languages like Lisp, which use parentheses rather extensively, it might be of interest whether the following two expressions are well-parenthesised (the left one is, the right one is not):

$$(((()()))())\qquad\qquad (((()()))()))$$

In order to solve such recognition problems, we need more powerful techniques than regular expressions. We will in particular look at *context-free languages*. They include the regular languages as the picture below shows:



Context-free languages play an important role in 'day-to-day' text processing and in programming languages. Context-free languages are usually specified by grammars. For example a grammar for well-parenthesised expressions is

$$P \quad\rightarrow\quad (\cdot P \cdot) \cdot P \mid \epsilon$$

In general grammars consist of finitely many rules built up from terminal symbols (usually lower-case letters) and non-terminal symbols (upper-case letters). Rules have the shape

$$NT \quad\rightarrow\quad rhs$$

where on the left-hand side is a single non-terminal and on the right a string consisting of both terminals and non-terminals including the ϵ-symbol for indicating the empty string. We use the convention to separate components on the right hand-side by using the $\cdot$ symbol, as in the grammar for well-parenthesised expressions. We also use the convention to use $\mid$ as a shorthand notation for several rules. For example

$$NT \quad\rightarrow\quad rhs_1 \mid rhs_2$$

means that the non-terminal NT can be replaced by either rhs_1 or rhs_2. If there are more than one non-terminal on the left-hand side of the rules, then we need to indicate what is the *starting* symbol of the grammar. For example the grammar for arithmetic expressions can be given as follows

$$\begin{array}{lcl} E & \rightarrow & N \\ E & \rightarrow & E \cdot + \cdot E \\ E & \rightarrow & E \cdot - \cdot E \\ E & \rightarrow & E \cdot * \cdot E \\ E & \rightarrow & (\cdot E \cdot) \\ N & \rightarrow & \epsilon \mid 0 \cdot N \mid 1 \cdot N \mid \ldots \mid 9 \cdot N \end{array}$$

where E is the starting symbol. A *derivation* for a grammar starts with the staring symbol of the grammar and in each step replaces one non-terminal by a right-hand side of a rule.