

- A Grammar is **LL(1)** iff it is **unambiguous**, **not left-recursive**, and **left factored**. LL(1) grammars can be parsed with **predictive parsers**, which are top-down parsers which do not backtrack. They use 1 **lookahead**: one token which can be looked at without being used.
- **LL(k) parsers** are top-down (= recursive descent) parsers with **Left to right** scanning which produce **Leftmost** derivations using **k** lookahead tokens. The grammar of a language parsed by an LL(k) parser is called an LL(k) grammar.
- **LR(k) parsers** are bottom-up (= shift-reduce) parsers with **Left to right** scanning which produce **Rightmost** derivations using **k** lookahead tokens. The grammar of a language parsed by an LR(k) parser is called an LR(k) grammar.

