

# LL(\*) Parsing

Jan Černý

`xcerny62@stud.fit.vutbr.cz`

Hana Kadlčíková

`xkadlc07@stud.fit.vutbr.cz`

LL(\*) parser is an LL parser whose lookahead (maximum number of tokens based on which a parser decides which production to use) isn't restricted by any constant. Therefore, the parser decides the size of a lookahead dynamically at run-time by using a lookahead DFA to predict which production should be used. Depending on the complexity of the parsing decisions, it switches from standard  $k \geq 1$  lookahead to arbitrary lookahead and in some cases to backtracking.

LL(\*) parsing is described by predicated grammars, which are grammars augmented by a set of side-effect-free semantic predicates and a set of actions. The strength of LL(\*) parser is similar to GLR (Generalized LR) and PEG (Parser Expression Grammar) parsers, but it also reduces unexpected behaviour while retaining LL's good error handling.