LL Parsing

- can't handle left-recursion
+ intuitive

→ read input from left to right

→ construct the AST "from left to right"

make decisions about grammar rules from top of AST down

left-recursive

<expr> ::= ∅

| <expr> + <expr>

| <expr> * <expr>

1+2+3

LR Parsing

→ make decisions about grammar rules from bottom up

Based on automata theory (push-down automaton)

Today: approximation of LR parsing

We have (1) input stream of tokens and (2) stack to track state.

Each step:

shift

move input from stream onto stack

reduce

collapse items on stack