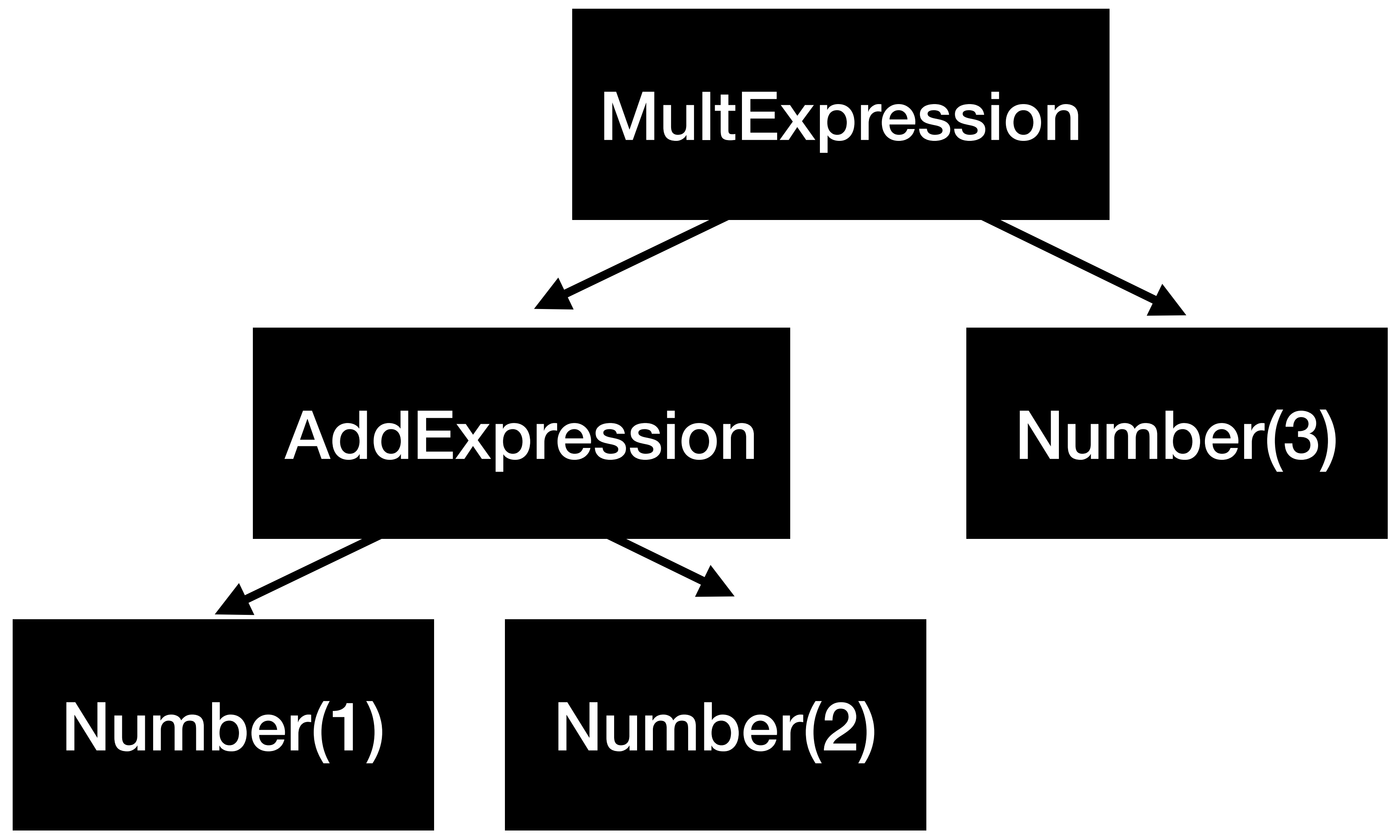
Overview: Lexing

- Turn the program into a list of "lexemes", or distinct pieces of text, based on their character composition
- Typically splits on whitespace (outside of, e.g., comments or strings)
- Example: "(1 + 2) * 3" could turn into [LPAREN, DIGIT("1"), OP("+"), DIGIT("2"), RPAREN, OP("*"), DIGIT("3")]
- Preprocessing step that serves as a starting point for parsing. Ideally doesn't have knowledge of programming language semantics, just syntax
- Typically uses regular expressions :)

Overview: Parsing

- Starts with lexemes produced by lexer
- Groups lexemes into expressions, turning them into an AST (abstract syntax tree)
- Makes sure that program is semantically valid
- Example: "(1 + 2) * 3" could turn into this AST:
- But "(1 + 2" would be thrown out as an invalid program, because the parentheses are mismatched


MultExpression
AddExpression
Number(3)
Number(1)
Number(2)