Concepts of Programming Languages, CSCI 305, Fall 2021
Translation of Token Description to Regular Expressions, 43-48 & 103-105
Sept. 1

Syntax – form
Semantics – meaning

In the context of program translation:
Syntax - what can be determined by the lexical analyzer and parser
Semantics – everything else

Lexeme – lowest-level syntactic unit of a language
Token – category of lexemes

Lexical analyzer assignment - create a table-driven lexical analyzer. The analyzer can be used to "tokenize" any language. The analyzer is language independent. The table is language specific. There are 4 steps to creating the language-specific table:

1. Create regular expressions for each token
2. Create a Non-deterministic Finite Automaton (NFA) for all
3. Convert NFA to a Deterministic Finite Automaton (DFA)
4. Convert the DFA to a minimal DFA

Regular expressions can be used to describe patterns for tokens

Regular expressions are defined recursively as:

Basis

- Character
- Empty string, ε

Induction

- Concatenation, $^{\circ}$ or just write symbols beside each other
- Alteration, |
- Kleene closure, *

Use parenthesis to avoid ambiguity.

Productions can be used to write regular expressions and grammars.

Productions are written:

identifier → definition

When productions are used to describe regular expressions, recursion is NOT allowed, either directly or indirectly. When productions are used to define grammars, recursion is allowed.