CSE 110A - Fundamentals Of Compiler Design

Elijah Hantman

Naive Scanner Tokenizer

- Simple
- Common for early languages
- Can be Unwieldy for many types of tokens
- Can be heavily nested

Regex

- Tokens are restricted to Regular languages
- Extensible
- Modular
- Easy to add tokens
- Many regex implementations have issues with order dependency
- Not as powerful as context free grammers

Side Note on Chompsky Hierarchy
From least to most powerful

- Regular
- Context free
- Context Sensitive
- Recursively Enumerable

What is Regex?

- Python Regex
  - Library re
  - re.fullmatch(pattern, string)
    Only succeeds if entire string matches the pattern
- Recursive definition

  All characters are regular expressions.

  These single character expressions only match on strings which are equal to themselves.

  If we write two regular expressions in sequence they are concatenated. Which means that they only match a string where some part of the string matches the first regex, and the rest of the string matches the second regex.

  We have modifiers which modify how the regular expression to the left matches.

  The modifiers are as follows:

  - *

    The Kleene star, matches zero or more copies of the expression to the left, equivalent to zero or more copies of the expression concatenated together.

  - +

    The plus matches one or more copies of the expression to the left. Equal to one or more copies concatenated together.

  There is one basic operator, choice (—).

  What choice means is that the string matches if it matches the expressin on the left, or the expression on the right.

  There are rules of operator precedence, modifiers come before choice, and concatenation comes before modifiers.

Concatenation can be specified via parentheses. These form groups. Each group is a valid regular expression. Parentheses define a single regular expression as a group which ignores precedence rules.

As an extension, the square brackets mean match against any character inside the bracket. [ab] is equivalent to a—b.

The ^ carrot operator means the complement, so [^ab] can match any character but a or b.

Optional Modifier, equivalent to choice between regex with character and regex without character. Same precedence as other modifiers. This is used to make regex non-greedy, since it will always put the version without the string first.

The . character is used to denote matching any character.

Many languages also use '()' to indicate a capture group. This is an extension for regex matchers that says to extract strings in groups and return them to the user.

When the ^ operator is placed outside square brackets, it matches the NULL at the start of the string. In Multiline mode, regex engines will match the beginning of the line.

The $ operator matches the NULL at the end of the string. In Multiline mode, it will match the end of the line.

Token Actions

- Replacement
- Keywords
- Error reporting
- Symbol table Creation

Actions are functions which run on and modify the token before it is returned. You can use them for keywords or marking them with additional metadata.

First Class Functions

- Python has functional programming
- You can make them, pass them around, do whatever you want
- Idk if function pointers really count, like they are not the same.