Regular Expressions

Outline and Objectives

Outline

- Motivation
- Regular expressions
  - Regular Ops.
  - Formal Definition
- Examples
- Applications

Learning Objectives

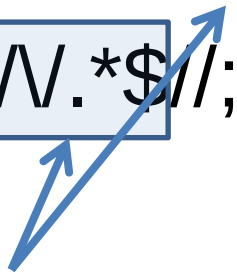
- Evaluate regular expressions.
- Construct regular expressions for simple languages.
- Describe regular operations.

Motivation

Task: Remove comments and white space from a text file. Assume comments start with two forward slashes (ie, //)

Python: `line = re.sub("//.*$", "", line)`

Perl: `$line =~ s/\/\/.*$//;`



Regular Expressions

Evaluating Expression

Arithmetic Expression

(5 + 3) * 2

- This evaluates to a number: 16
- Makes use of standard mathematical operations

Regular Expression

01*0

- This evaluates to a language: A string that starts and ends with ‘0’ and has a continuous string of zero or more ‘1’s
- Makes use of regular operations

Regular Operations

Regular operations operate on languages…

Let X and Y be regular languages, then following are regular operations. The result of each op. can be shown to be a regular language.

Concatenation: $X \bullet Y = \{ xy \mid x \in X \textit{ AND } y \in Y \}$

Union: $X \cup Y = \{ x \mid x \in X \textit{ OR } y \in Y \}$

Star: $X^* = \{x_1 x_2 x_3 .. x_i \mid i \geq 0 \text{ and } xi \in X \}$

(Sipser, 44)

Regular Languages

R is a regular expression if R is any of the following…

- x for some x in an alphabet (set of valid chars.)
- ε or Ø

ε is a language with empty string; Ø is an empty language

- (R_1 U R_2), where R_1 and R_2 are regular languages
- (R_1 • R_2), where R_1 and R_2 are regular languages
- (R_1)*, where R_1 is a regular language

(Sipser, 64)

Examples

Assume the valid set of characters is 0 & 1. Describe or explicitly state the language represented by each expression.

(0 U 1)*0 =	{ s | s ends in a 0}
(1 U ε)0 =	{ 10, 0}
(0 U 1)1(0 U 1) =	{ s | s contains 3 digits w/ the 1 in the middle }

Examples

Assume the valid set of characters is 0 & 1. Provide a regular expression the evaluates to the language described.

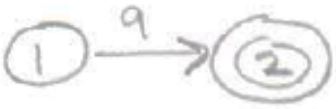
- A language that starts with 0 and ends with 1 and is 4 characters long. = 0(0 U 1)(0 U 1)1

- A language that accepts any string containing a combination of 0's or 1's that has at least one character. = (0 U 1)(0 U 1)*

- A string that has an even pair of 0's and no 1's. = (00)*

Link to FSM

Regular expressions are equivalent to FSM. The idea is to construct a FSM that will accept any string from the language the RE represents.

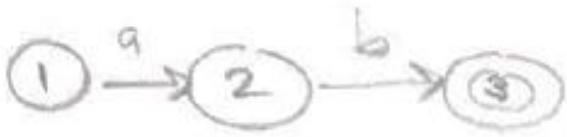
a 

ε 

(Sipser, 66)

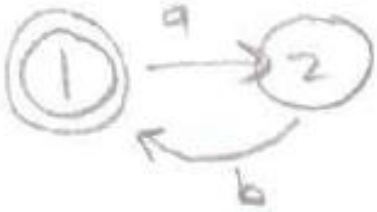
Link to FSM

ab



a U b

(ab)*



Applications

You'll typically see regular expressions used for pattern matching.

When working with text strings, there are number of language specific extensions that make matching strings easier

Symbol	Meaning	Symbol	Meaning
^	Start of line	[ ]	Group of characters
$	End of line	[^]	Group of characters to exclude
+	One or more	.	Any valid character

Examples

Provide a regular expression to match the following.

- White space at the beginning of a line — ^[\t]*
- The word ‘add’ — add[\t\n]
- Any number — [0-9]+
- Any number with punctuation — [0-9.,]+
- Whole line — ^.*$

You may need to escape the . to match a period instead of any character.

References

- Sipser M: *Introduction to the theory of computation*. 2nd ed. Boston: Thomson Course Technology; 2006.