Monash University
Faculty of Information Technology

# FIT2014 Theory of Computation



## Lecture 6

## Regular Expressions



slides by Graham Farr
based in part on previous slides by David Albrecht

# Overview



- Some Problems
- Applications of Regular Expressions
- Regular Expressions
- Regular Languages

# Some Problems



- Find all the files which contain old subject course codes.
- Find all the e-mail addresses in a set of mail files.
- Change the way comments in programs are formatted in your web pages.
- Using web server access files, record how many times each page is visited, and how many times each link is used.

# Applications of Regular Expressions



- Useful way to describe simple patterns.
- Used in several programs:
- Editors: vi, emacs
- Filters: `egrep`, `sed`, `gawk`
- Programming languages: JFlex, CUP, Perl

# Filters

`egrep`

- A program which searches a file for a pattern described by a regular expression.

`sed`

- A program which enables stream editing of files.

`awk`, `nawk`, `gawk`

- Programming languages which enable text manipulation.

# Programming Languages



JFlex, flex, lex

- Languages used to generate lexical analysers.



CUP, bison, yacc

- Languages used to generate compilers.



Perl

- A powerful scripting language, developed in the 1980s by Larry Wall.

# Regular Expressions for Small Languages



The empty language $\emptyset$

Language $\{\varepsilon\}$ consisting only of the empty word $\varepsilon$



Language $\{w\}$ consisting only of the single word $w$ $w$



E.g.:
the language $\{\texttt{abbab}\}$ $\texttt{abbab}$

# Alternatives, Grouping

**Alternatives**
are indicated by $\cup$.
E.g.:

$$1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9$$

is a regular expression for:

$$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

**Grouping**
is indicated by ( ).
E.g.:

$$(ab \cup ba)(e \cup g)$$

is a regular expression for:

$$\{abe, abg, bae, bag\}$$

# Finite Languages

consist of finite number of words.

E.g.:

$$\{\texttt{abaaba}, \texttt{abbbba}, \texttt{abbaba}\}$$

Regular Expression:

$$\texttt{abaaba} \cup \texttt{abbbba} \cup \texttt{abbaba}$$

*. . . or:*

$$\texttt{ab}(\texttt{aa} \cup \texttt{bb} \cup \texttt{ba})\texttt{ba}$$

# Kleene Star

Matching an expression *zero or more times* is indicated by $^*$



For example:



$\mathtt{a}^*$ represents $\{\varepsilon, \mathtt{a}, \mathtt{aa}, \mathtt{aaa}, \mathtt{aaaa}, \ldots\}$

$(\mathtt{ab})^*$ represents $\{\varepsilon, \mathtt{ab}, \mathtt{abab}, \mathtt{ababab}, \ldots\}$



$\mathtt{ab}^*$ represents $\{\mathtt{a}, \mathtt{ab}, \mathtt{abb}, \mathtt{abbb}, \mathtt{abbbb}, \ldots\}$

Note: $\mathtt{ab}^* \neq (\mathtt{ab})^*$

$$(\mathtt{aa} \cup \mathtt{bb})^* = (\mathtt{aa} \cup \mathtt{bb})^0 \cup (\mathtt{aa} \cup \mathtt{bb})^1 \cup (\mathtt{aa} \cup \mathtt{bb})^2 \cup \cdots$$

$$= \varepsilon \cup (\mathtt{aa} \cup \mathtt{bb}) \cup (\mathtt{aa} \cup \mathtt{bb})(\mathtt{aa} \cup \mathtt{bb}) \cup \cdots$$

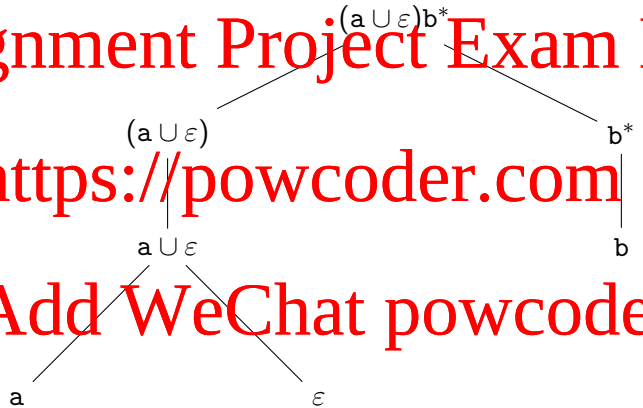


represents:

$$\{\varepsilon, \mathtt{aa}, \mathtt{bb}, \mathtt{aaaa}, \mathtt{aabb}, \mathtt{bbaa}, \mathtt{bbbb}, \mathtt{aaaaaa}, \mathtt{aaaabb}, \mathtt{aabbaa}, \ldots\}$$

# Parse Tree





# Definition

1. $\emptyset$ and $\varepsilon$ are regular expressions
2. All letters in the alphabet are regular expressions.
3. If $R$ and $S$ are regular expressions, then so are:
   (i) $(R)$
   (ii) $RS$
   (iii) $R \cup S$
   (iv) $R^*$

A language which can be described by a *regular expression* is called a **regular language**.



If a word belongs to the language described by a regular expression, then we say it is **matched** by the regular expression.

# Example: EVEN-EVEN



Recall:

EVEN-EVEN $=$ { All strings in which a and b each occur an *even* number of times }

$= \{\varepsilon, \mathtt{aa}, \mathtt{bb}, \mathtt{aaaa}, \mathtt{aabb}, \mathtt{abab}, \mathtt{abba}, \ldots\}$.



Regular Expression:



$$(\mathtt{aa} \cup \mathtt{bb} \cup (\mathtt{ab} \cup \mathtt{ba})(\mathtt{aa} \cup \mathtt{bb})^*(\mathtt{ab} \cup \mathtt{ba}))^*$$

# Things to think about . . .

Is the set of all English words (in some standard dictionary) a regular language?

Is DOUBLEWORD (see Lecture 1) a regular language?

Is PALINDROME a regular language?

Is the set of all grammatical English sentences a regular language?

How would you determine, for a given string and regular expression, whether the string matches the regular expression?

# Example: Floating Point Number



A floating point number has one or more digits, which may begin with a minus sign ($-$), and which may contain a decimal point.



E.g.,

$$0 \quad 1.2 \quad -3 \quad -4.675 \quad 002 \quad 023.50$$

# Sequence of Digits

One Digit

$$0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9$$



Two Digits

$$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)$$



Three Digits

$$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)$$



One or more Digits

$$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$$

# Sequences of Digits



Digit

$$D = (0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)$$

Two Digits

$$DD \quad \text{or} \quad D^2$$



Three Digits

$$DDD \quad \text{or} \quad D^3$$



One or more Digits

$$DD^*$$

# Numbers

One Digit

$$D = (0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)$$



Nonnegative Integers

$$N = DD^* \qquad \text{e.g.:} \quad 1 \quad 123 \quad 1209 \quad 002 \quad 020$$



Integers

$$Z = N \cup (-N)$$



Floating Point Number

$$F = Z \cup (Z.) \cup (.N) \cup (-.N) \cup (Z.N)$$

# Other Notations

| | alternative notation |
|---|---|
| $R \cup S$ | $R \mid S$ |
| $0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9$ | [0-9] |
| any letter a to z | [a-z] |
| $RR^*$ | $R^+$ |
| $\varepsilon \cup R$ | $R?$ |



*Be careful!* Many variations exist. Tools that use regexps differ in many details:

- whether or not they use the above special meanings of $+$ and $?$;
- how they handle the parentheses and vertical bar for alternatives
  - sometimes $(\cdots|\cdots)$
  - sometimes $\backslash(\cdots\backslash|\cdots\backslash)$
- how they use full stop (often represents any non-newline character);
- how they represent newline characters
- ...

# Revision

- Regular Expressions
- Definitions
- How to use them to define languages
- Regular languages



Read Sipser, §1.3, pp 63–66.



Additional Reading

- Jeffrey E.F. Friedl, *Mastering Regular Expressions: Powerful Techniques for Perl and Other Tools*, O'Reilly, 1997.