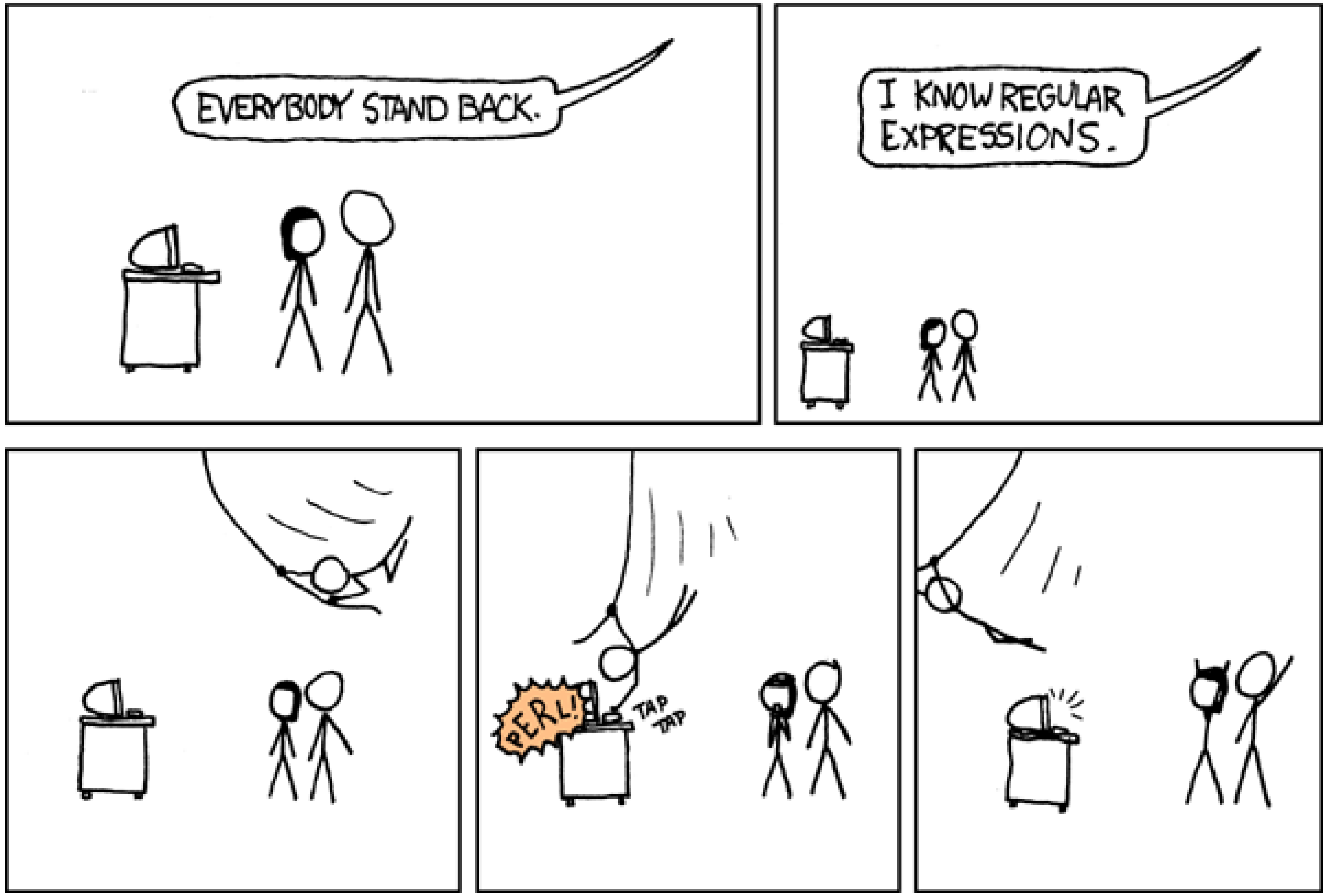


Regular Expressions, Context Free Grammars

CSE 311: Foundations of Computing I
Lecture 19

Announcements

- HW6 due Wednesday at 11:59pm on Gradescope.
- Midterm Corrections due Wednesday at 11:59pm on Gradescope.
  - $\text{midterm\_grade} = 0.75 \cdot \text{original\_grade} + 0.25 \cdot \max(\text{corrected\_grade}, \text{original\_grade})$
  - No Late Days permitted
  - More details posted on Assignments page

Theoretical Computer Science

Recall: Course Goals

1. Learn to make & clearly communicate rigorous formal arguments
   - Mathematical Proofs
2. Understand mathematical objects that are widely used in CS
   - Number Theory, Set Theory, Recursively-Defined Functions
3. Explore and analyze models of computation
   - Regular Expressions, Context-Free Grammars, Finite Automata

Languages

Definition:
A language is ______________________________________.

For Example:

- ____
- ____
- ____

Languages in Theoretical Computer Science

- We want to study different ______________________________, and the ______________________________ of each.
- A computer is said to ______________ a language if it can ______________ ______________________________.
- One way to evaluate how powerful a model of computation is is to ______________________________.

Regular Languages

One class of languages

Regular Expressions

Basis Step:

Recursive Step:

Regular Expressions

Each regular expression **matches** a set of strings (a language).

ε matches ____

a matches ____

Regular Expressions

Each regular expression **matches** a set of strings (a language).

$A \cup B$ matches ____

AB matches ____

A^* matches ____

Examples

a^*b^*

$(0 \cup 1)0(0 \cup 1)0$

$(00 \cup 11)^*$

Examples

Construct a regular expression that matches the given set of strings.

All binary strings.

All binary strings that contain 0110.

Examples

Construct a regular expression that matches the given set of strings.

All binary strings that have an even number of 1s.

All binary strings that don't contain 00.

Practical Advice

- Check ε and single character strings. Those are often edge cases.
- List 5 strings that should be matched, and 5 strings that shouldn't be.

  Test your RegEx against those strings.
- Remember $*$ allows for 0 copies! To say "at least one copy", use AA^*.

Exercises

Construct a regular expression that matches the given set of strings.

The set of all binary strings of odd length.

The set of all binary strings with at most two ones.

The set of all binary strings with equal number of 0s and 1s.

Applications of Regular Expressions

An aside: Regular Expressions are used everywhere, outside of CS Theory too.

- Some search tools allow you to search for regex's instead of strings
  E.g. search for "$(a \cup \cdots \cup z \cup 1 \cup \cdots \cup 9)^*\mathbf{@uw.edu}$".
- Used in $\mathbf{grep}$, a program that does pattern matching searches in LINUX.
- Used to define the "tokens": e.g. legal variable names, keywords, in programming languages and compilers.
- Many implementations of RegExs are more powerful than our theoretical Regular Expression

WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.
OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!
BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!
IT'S HOPELESS!
EVERYBODY STAND BACK.
I KNOW REGULAR EXPRESSIONS.
PERL!
TAP TAP

Regular Languages

Definitions:

Regular Languages are ____

____.

Irregular Languages are ____.

Irregular Languages

It turns out a lot of useful languages are irregular.

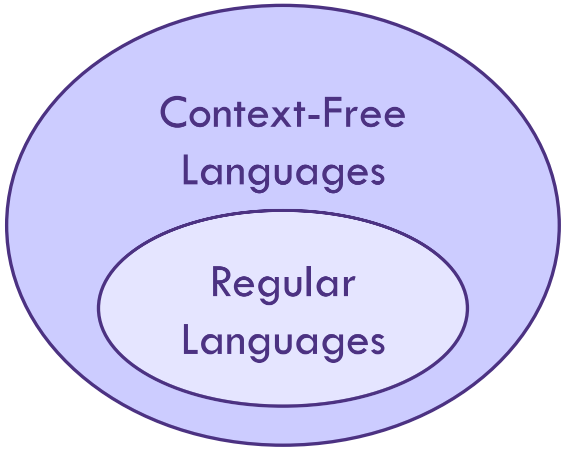
- Binary strings with an equal number of 0s and 1s
- Palindromes (strings that read the same forwards and backwards)
- Matched parentheses, e.g. ((())())
- Properly formed arithmetic expressions

Context Free Languages

Another class of languages

Context-Free Languages

- We just saw some limitations of Regular Languages
- Context-Free Languages are a strictly larger class of languages



- Context-Free Languages are generated by Context-Free Grammars
  (just like Regular Languages are specified by Regular Expressions)

Context-Free Grammars (CFGs)

- A Context-Free Grammar is a ________________________________:
  - Alphabet of ____
  - A finite set ____
  - One special nonterminal __________________________________

- A production rule for a nonterminal A takes the form:

  where each w_i is a ____

Context-Free Grammars (CFGs)

For example:

$S \rightarrow Ab \mid c$

$A \rightarrow Aa \mid \varepsilon$

Context-Free Grammars

For Example:
S → Ab | c
A → Aa | ε

We think of Context-Free Grammars as **generating** strings.

1. Start from the start symbol S.
2. Choose a nonterminal, e.g. S, in the string, and replace it by one of the w's in the rules for S

$$S \rightarrow w_1 \mid w_2 \mid \ldots \mid w_k$$

3. Repeat step 2 until there are no nonterminals left.

The language that the CFG describes is the set of all strings that it generates.

Example

$S \rightarrow 0S \mid S1 \mid \varepsilon$

Example

$S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon$

Example

CFG for the language $\{0^n 1^n : n \geq 0\}$

Example

CFG for the language $\{0^n 1^n 23 : n \geq 0\}$

Exercises

CFG for the set of binary strings with the same number of 0s as 1s.

CFG for the set of balanced parentheses. E.g. ((())())