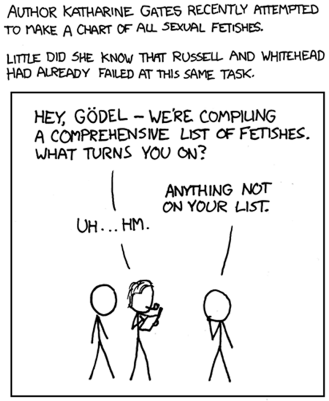
# CSE 311: Foundations of Computing

Fall 2013

Lecture 26: Pattern matching, cardinality


AUTHOR KATHARINE GATES RECENTLY ATTEMPTED TO MAKE A CHART OF ALL SEXUAL FETISHES.
LITTLE DID SHE KNOW THAT RUSSELL AND WHITEHEAD HAD ALREADY FAILED AT THIS SAME TASK.
HEY, GÖDEL – WE'RE COMPILING A COMPREHENSIVE LIST OF FETISHES. WHAT TURNS YOU ON?
UH... HM.
ANYTHING NOT ON YOUR LIST.

## highlights

- DFAs ≡ Regular Expressions
  - No need to know details of NFAs→RegExpressions
- Method for proving no DFAs for languages
  - e.g. $\{0^n1^n : n \geq 0\}$,
    {Palindromes},
    {Balanced Parens}

## how to show language L has no DFA

- Find a "hard" infinite set $S=\{s_0,s_1,\ldots,s_n,\ldots\}$ of strings that might be prefixes of strings in L
- Show that S is hard by showing that no two strings $s_n \neq s_m$ in S can end at the same state of any DFA recognizing L
  - For each pair $s_n \neq s_m$ find an extender string t depending on n,m so that exactly one of $s_nt$ and $s_mt$ is in L
- Conclude that any DFA for L would need ≥ |S|states which is not finite, and so impossible

## pattern matching

- Given
  - a string, **s**, of **n** characters
  - a pattern, **p**, of **m** characters
  - usually **m<<n**
- Find
  - all occurrences of the pattern **p** in the string **s**
- Obvious algorithm:
  - try to see if **p** matches at each of the positions in **s**

    stop at a failed match and try the next position

```
  string s = x y x x y x y x y y x y x y x y y x y x y x x
pattern p = x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y
                 x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y
                 x y x y y
                   x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y
                 x y x y y
                   x
                     x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y
                 x y x y y
                   x
                     x y x y y x y x y x x
                       x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y
                 x y x y y
                   x
                     x y x y y x y x y x x
                       x
                         x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y
                 x y x y y
                   x
                     x y x y y x y x y x x
                       x
                         x y x
                           x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y
                 x y x y y
                   x
                     x y x y y x y x y x x
                       x
                         x y x
                           x
                             x y x y y x y x y x x
```

```
string s = x y x x y x y x y y x y x y x y y x y x y x x
           x y x y
             x
               x y
                 x y x y y
                   x
                     x y x y y x y x y x x
                       x
                         x y x
                           x
                             x
                               x y x y y x y x y x x
```

string **s** = x y x x y x y x y y x y x y x y y x y x y x x

x y x y

x

x y

x y x y y

x

x y x y y x y x y x x

x

x y x

x

x

x y x y y

x y x y y x y x y x x

string **s** = x y x x y x y x y y x y x y x y y x y x y x x

x y x y

x

x y

x y x y y

x

x y x y y x y x y x x

x

x y x

x

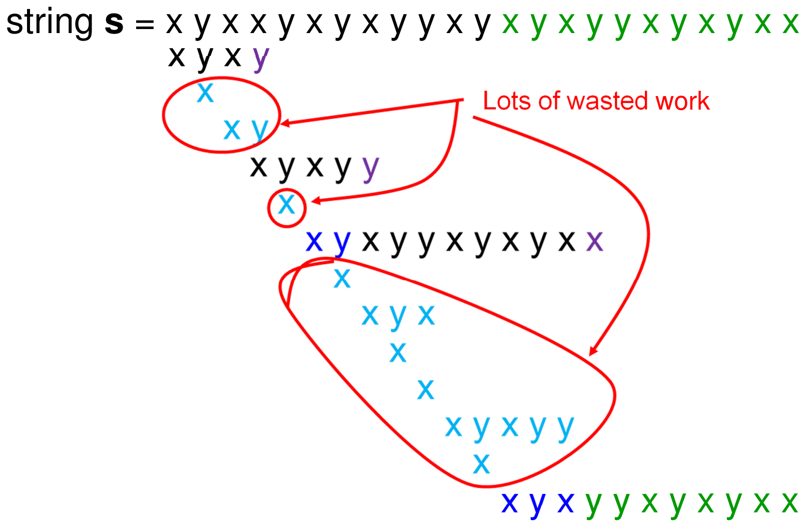
x

x y x y y

x

x y x y y x y x y x x

Worst-case time O(**mn**)


string **s** = x y x x y x y x y y x y x y x y y x y x y x x

x y x y

x

x y

x y x y y

x

x y x y y x y x y x x

x

x y x

x

x

x y x y y

x

x y x y y x y x y x x

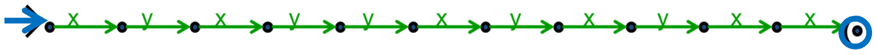
Lots of wasted work

## better pattern matching via finite automata

- **Build a DFA for the pattern (preprocessing) of size O(m)**
  - **Keep track of the 'longest match currently active'**
  - **The DFA will have only m+1 states**
- **Run the DFA on the string n steps**
- **Obvious construction method for DFA will be O($m^2$) but can be done in O(m) time.**
- **Total O(m+n) time**

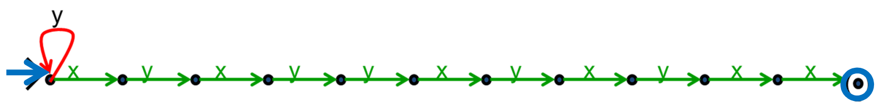
## building a DFA for the pattern

pattern **p**=x y x y y x y x y x x



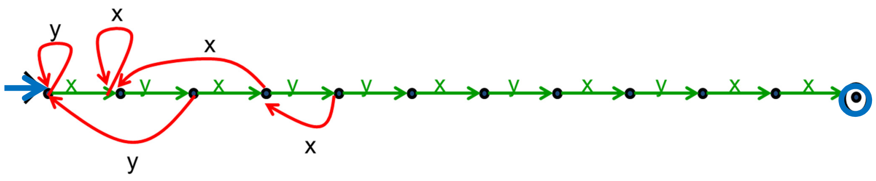
## preprocessing the pattern

pattern **p**=x y x y y x y x y x x



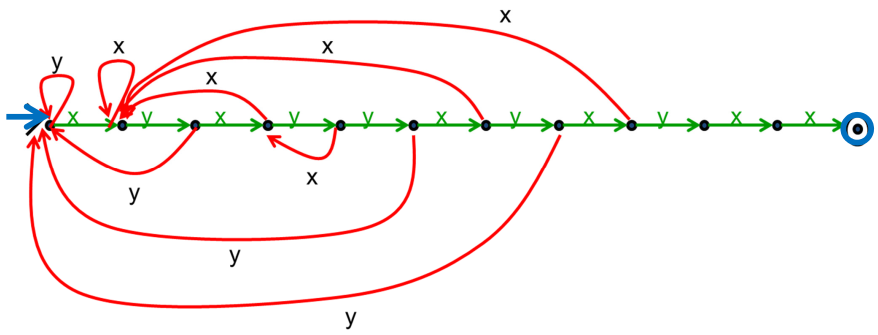
## preprocessing the pattern

pattern **p**=x y x y y x y x y x x



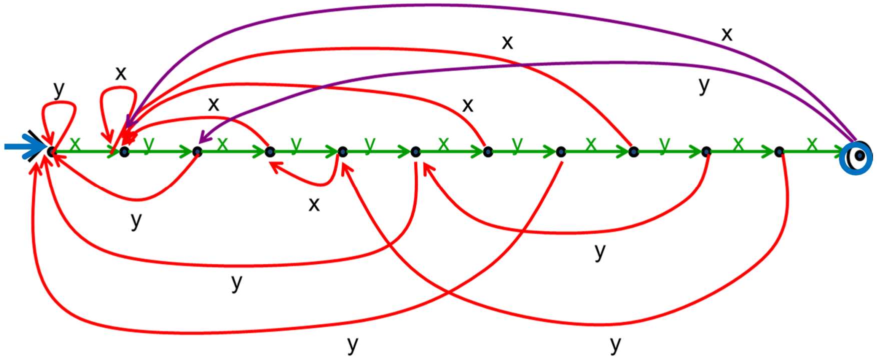
## preprocessing the pattern

pattern **p**=x y x y y x y x y x x



## preprocessing the pattern

pattern **p**=x y x y y x y x y x x



## generalizing

- Can search for arbitrary combinations of patterns
  - Not just a single pattern
  - Build NFA for pattern then convert to DFA ‘on the fly’.

    Compare DFA constructed above with subset construction for the obvious NFA.

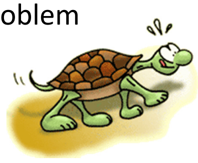
## cardinality and computability

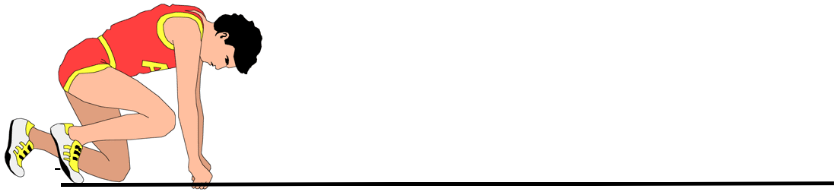
**Computers as we know them grew out of a desire to avoid bugs in mathematical reasoning**

## a brief history of reasoning

Ancient Greece

- Deductive logic
  - Euclid’s Elements
- Infinite things are a problem
  - Zeno’s paradox





## a brief history of reasoning

- 1670’s-1800’s  Calculus & infinite series
  - Suddenly infinite stuff really matters
  - Reasoning about the infinite still a problem

    Tendency for buggy or hazy proofs
- Mid-late 1800’s
  - Formal mathematical logic

    Boole   Boolean Algebra
  - Theory of infinite sets and cardinality

    Cantor

    “There are more real #’s than rational #’s”

## a brief history of reasoning

- 1900
  - Hilbert's famous speech outlines goal: mechanize all of mathematics

    23 problems
- 1930’s
  - Gödel, Turing show that Hilbert’s program is impossible.

    Gödel’s Incompleteness Theorem

    Undecidability of the Halting Problem

Both use ideas from Cantor’s proof about reals & rationals

## starting with Cantor

- How big is a set?
  - If S is finite, we already defined |S| to be the number of elements in S.
  - What if S is infinite?  Are all of these sets the same size?

    Natural numbers $\mathbb{N}$

    Even natural numbers

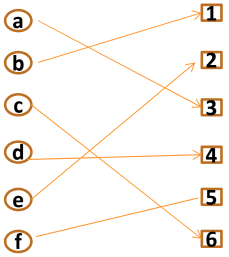
    Integers $\mathbb{Z}$

    Rational numbers $\mathbb{Q}$

    Real numbers $\mathbb{R}$

## cardinality

Definition:  Two sets A and B are the same size (same cardinality) iff there is a 1-1 and onto function f:A→B



Also applies to infinite sets

## cardinality

- The natural numbers and even natural numbers have the same cardinality:

0 1 2 3 4 5 6 7 8 9 10 …

0 2 4 6 8 10 12 14 16 18 20 …

n is matched with 2n

## countability

Definition: A set is *countable* iff it is the same size as some subset of the natural numbers

Equivalent: A set S is *countable* iff there is an onto function g: $\mathbb{N} \rightarrow S$

Equivalent: A set S is *countable* iff we can write $S=\{s_1, s_2, s_3, \ldots\}$

## the set of all integers is countable

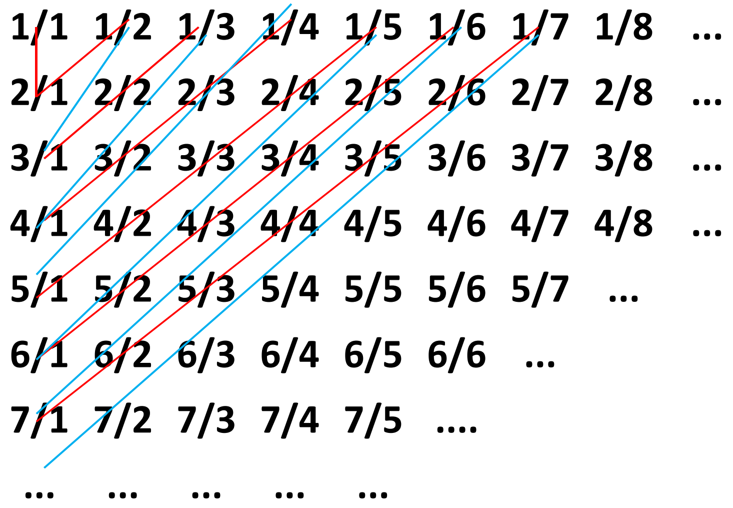
## is the set of positive rational numbers countable?

- We can't do the same thing we did for the integers
  - Between any two rational numbers there are an infinite number of others

## the set of positive rational numbers is countable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1/1 | 1/2 | 1/3 | 1/4 | 1/5 | 1/6 | 1/7 | 1/8 | … |
| 2/1 | 2/2 | 2/3 | 2/4 | 2/5 | 2/6 | 2/7 | 2/8 | … |
| 3/1 | 3/2 | 3/3 | 3/4 | 3/5 | 3/6 | 3/7 | 3/8 | … |
| 4/1 | 4/2 | 4/3 | 4/4 | 4/5 | 4/6 | 4/7 | 4/8 | … |
| 5/1 | 5/2 | 5/3 | 5/4 | 5/5 | 5/6 | 5/7 | … | |
| 6/1 | 6/2 | 6/3 | 6/4 | 6/5 | 6/6 | … | | |
| 7/1 | 7/2 | 7/3 | 7/4 | 7/5 | …. | | | |
| … | … | … | … | … | | | | |

## the set of positive rational numbers is countable



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1/1 | 1/2 | 1/3 | 1/4 | 1/5 | 1/6 | 1/7 | 1/8 | … |
| 2/1 | 2/2 | 2/3 | 2/4 | 2/5 | 2/6 | 2/7 | 2/8 | … |
| 3/1 | 3/2 | 3/3 | 3/4 | 3/5 | 3/6 | 3/7 | 3/8 | … |
| 4/1 | 4/2 | 4/3 | 4/4 | 4/5 | 4/6 | 4/7 | 4/8 | … |
| 5/1 | 5/2 | 5/3 | 5/4 | 5/5 | 5/6 | 5/7 | … | |
| 6/1 | 6/2 | 6/3 | 6/4 | 6/5 | 6/6 | … | | |
| 7/1 | 7/2 | 7/3 | 7/4 | 7/5 | …. | | | |
| … | … | … | … | … | | | | |

## the set of positive rational numbers is countable

$\mathbb{Q}^+$ = {1/1, 2/1,1/2, 3/1,2/2,1/3,
4/1,2/3,3/2,1/4, 5/1,4/2,3/3,2/4,1/5, …}

List elements in order of

- numerator+denominator
- breaking ties according to denominator

Only k numbers have total of k

Technique is called “dovetailing”

## claim: $\Sigma^*$ is countable for every finite $\Sigma$

## the set of all Java programs is countable

## georg cantor

- Set theory
- Cardinality
- Continuum hypothesis

## georg cantor

Cantor's revolutionary ideas were not accepted by the mathematical establishment.

Poincaré referred to them as a "grave disease infecting mathematics."

Kronecker fought to keep Cantor's papers out of his journals.

He spent the last 30 years of his life battling depression, living often in "sanatoriums" (psychiatric hospitals)

## what about the real numbers?

Q: Is *every* set is countable?

A: Theorem [Cantor] The set of real numbers (even just between 0 and 1) is NOT countable

Proof is by contradiction using a new method called diagonalization

## proof by contradiction

- Suppose that $\mathbb{R}^{[0,1)}$ is countable
- Then there is some listing of all elements
  $\mathbb{R}^{[0,1)} = \{ r_1, r_2, r_3, r_4, \ldots \}$
- We will prove that in such a listing there must be at least one missing element which contradicts statement "$\mathbb{R}^{[0,1)}$ is countable"
- The missing element will be found by looking at the decimal expansions of $r_1, r_2, r_3, r_4, \ldots$

## real numbers between 0 and 1: $\mathbb{R}^{[0,1)}$

- Every number between 0 and 1 has an infinite decimal expansion:

$1/2 = 0.500000000000000000000000\ldots$

$1/3 = 0.333333333333333333333333\ldots$

$1/7 = 0.14285714285714285714285\ldots$

$\pi - 3 = 0.14159265358979323846264\ldots$

$1/5 = 0.199999999999999999999999\ldots$

$= 0.200000000000000000000000\ldots$

## representations of real numbers as decimals

Representation is unique except for the cases that decimal ends in all 0's or all 9's.

$$x = 0.19999999999999999999999\ldots$$

$$10x = 1.9999999999999999999999\ldots$$

$$9x = 1.8 \text{ so}$$

$$x = 0.200000000000000000\ldots$$

Won't allow the representations ending in all 9's

All other representations give different elements of $\mathbb{R}^{[0,1)}$

## supposed listing of $\mathbb{R}^{[0,1)}$

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | … |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $r_1$ | 0. | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | … | … |
| $r_2$ | 0. | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | … | … |
| $r_3$ | 0. | 1 | 4 | 2 | 8 | 5 | 7 | 1 | 4 | … | … |
| $r_4$ | 0. | 1 | 4 | 1 | 5 | 9 | 2 | 6 | 5 | … | … |
| $r_5$ | 0. | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | … | … |
| $r_6$ | 0. | 2 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | … | … |
| $r_7$ | 0. | 7 | 1 | 8 | 2 | 8 | 1 | 8 | 2 | … | … |
| $r_8$ | 0. | 6 | 1 | 8 | 0 | 3 | 3 | 9 | 4 | … | … |
| … | …. | … | …. | …. | … | … | … | … | … | … | |

## supposed listing of $\mathbb{R}^{[0,1)}$

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $r_1$ | 0. | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | ... |
| $r_2$ | 0. | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | ... | ... |
| $r_3$ | 0. | 1 | 4 | 2 | 8 | 5 | 7 | 1 | 4 | ... | ... |
| $r_4$ | 0. | 1 | 4 | 1 | 5 | 9 | 2 | 6 | 5 | ... | ... |
| $r_5$ | 0. | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | ... | ... |
| $r_6$ | 0. | 2 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | ... | ... |
| $r_7$ | 0. | 7 | 1 | 8 | 2 | 8 | 1 | 8 | 2 | ... | ... |
| $r_8$ | 0. | 6 | 1 | 8 | 0 | 3 | 3 | 9 | 4 | ... | ... |
| ... | .... | ... | .... | .... | ... | ... | ... | ... | ... | ... | |

## flipped diagonal

**Flipping Rule:**

**If digit is 5, make it 1**
**If digit is not 5, make it 5**

| | | 1 | 2 | 3 | 4 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $r_1$ | 0. | $5^{1}$ | 0 | 0 | 0 | | | | | | |
| $r_2$ | 0. | 3 | $3^{5}$ | 3 | 3 | | | | | | |
| $r_3$ | 0. | 1 | 4 | $2^{5}$ | 8 | 5 | 7 | 1 | 4 | ... | ... |
| $r_4$ | 0. | 1 | 4 | 1 | $5^{1}$ | 9 | 2 | 6 | 5 | ... | ... |
| $r_5$ | 0. | 1 | 2 | 1 | 2 | $2^{5}$ | 1 | 2 | 2 | ... | ... |
| $r_6$ | 0. | 2 | 5 | 0 | 0 | 0 | $0^{5}$ | 0 | 0 | ... | ... |
| $r_7$ | 0. | 7 | 1 | 8 | 2 | 8 | 1 | $8^{5}$ | 2 | ... | ... |
| $r_8$ | 0. | 6 | 1 | 8 | 0 | 3 | 3 | 9 | $4^{5}$ | ... | ... |
| ... | .... | ... | .... | .... | ... | ... | ... | ... | ... | ...$^{...}$ | |

## flipped diagonal number D

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D = | 0. | 1 | 5 | 5 | 1 | 5 | 5 | 5 | 5 | ... | |

D is in $\mathbb{R}^{[0,1)}$

But for all **n**, we have
$D \neq r_n$ since they differ on
$n^{th}$ digit (which is not **0** or **9**)

⇒ list was incomplete

⇒ $\mathbb{R}^{[0,1)}$ is not countable

## the set of all functions f : ℕ→{0,1,...,9} is not countable

## non-computable functions

- We have seen that
  - The set of all (Java) programs is countable
  - The set of all functions f : $\mathbb{N}\rightarrow\{0,1,\ldots,9\}$ is not countable

- So…
  - There must be some function f : $\mathbb{N}\rightarrow\{0,1,\ldots,9\}$ that is not computable by any program!