CSE 311 Foundations of Computing I

Lecture 25

Pattern Matching, Cardinality, Computability

Spring 2013

Announcements

- Reading
  - 7th edition: 2.5 (Cardinality) + p. 201 and 13.5
  - 6th edition: pp. 158-160 (Cardinality)+ p 177 and 12.5
- Pick up graded Homework 6
- Homework 8 out today, due next Friday

Last Lecture 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 FSMs/DFAs are used in designing circuits
  - This was just to give a rough idea

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

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

x y x

x

x

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

Lots of wasted work

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

Better Pattern Matching via Finite Automata

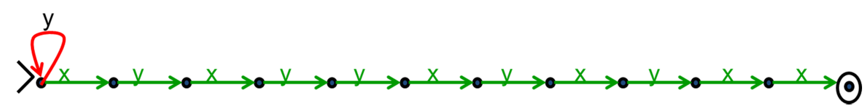
- Build a DFA for the pattern (preprocessing) of size O(**m**)
  - Keep track of the 'longest match currently active'
  - The DFA will have only **m**+1 states
- Run the DFA on the string **n** steps
- Obvious construction method for DFA will be O($\mathbf{m}^2$) but can be done in O(**m**) time.
- Total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

Building a DFA for the pattern

Pattern **p**=x y x y y x y x y x x

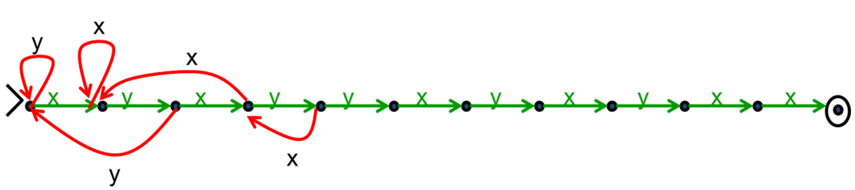
Preprocessing the pattern

Pattern **p**=x y x y y x y x y x x



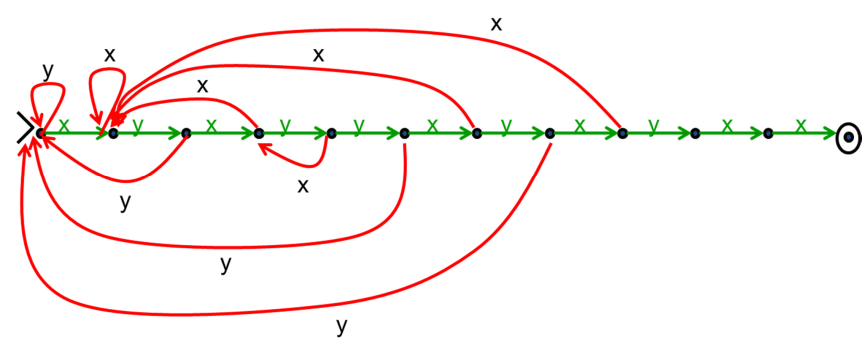
Preprocessing the pattern

Pattern **p**=x y x y y x y x y x x



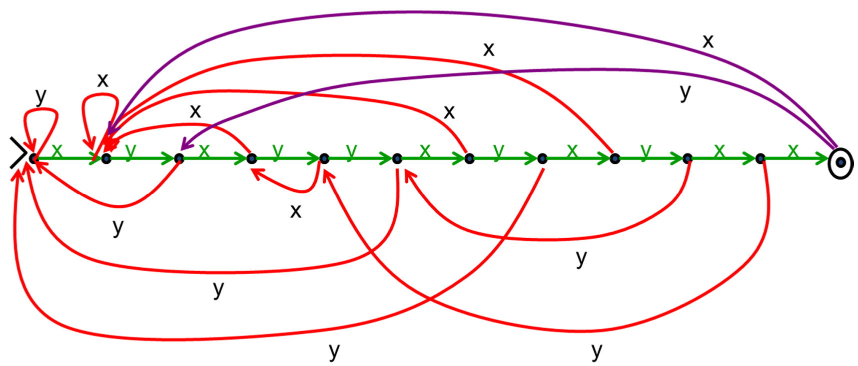
Preprocessing the pattern

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.

CSE 311 Foundations of Computing I

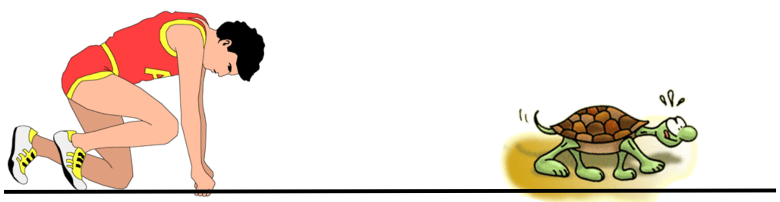
Cardinality and Computability

Computing & Mathematics

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 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

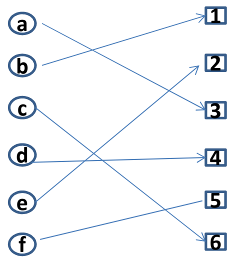
Both use ideas from Cantor's proof about reals & rationals

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

Def: 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 $S=\{s_1, s_2, s_3, ...\}$

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

Positive Rational Numbers

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**	**….**			
…	**…**	**…**	**…**	**…**				

{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**	**….**			
…	**…**	**…**	**…**	**…**				

{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 when the total is k

Technique is called "dovetailing"

Claim: Σ^* is countable for every finite Σ

The set of all Java programs is countable

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 = 0.50000000000000000000000...

1/3 = 0.33333333333333333333333...

1/7 = 0.14285714285714285714285...

π -3 = 0.14159265358979323846264...

1/5 = 0.19999999999999999999999...

= 0.20000000000000000000000...

Representations as decimals

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

x = 0.19999999999999999999999...

10x = 1.9999999999999999999999...

9x = 1.8 so x=0.200000000000000000...

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

All other representations give 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

		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}	...	...
...	....	...	....	....	...	...	...	...	...	......	

Flipping Rule:

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

$\Rightarrow$ list was incomplete

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

The set of all functions $f : \mathbb{N} \rightarrow \{0,1,\ldots,9\}$ 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!