CSE 311 Foundations of Computing I

Lecture 9

Set Theory and Functions

Autumn 2011

Announcements

- Reading assignments
  - Today: Sets and Functions
    - 2.1-2.3 6^{th} and 7^{th} Editions
    - 1.6-1.8 5^{th} Edition
  - Wednesday:
    - 4.1-4.2 7^{th} Edition
    - 3.4, 3.6 up to p. 227 6^{th} Edition
    - 2.4, 2.5 up to p. 177 5^{th} Edition

Highlights from last lecture

- Formal proofs:
  - simple well-defined rules
  - easy to check
- English proofs correspond to those rules
  - designed to be easier for humans to read
  - easily checkable in principle
- Simple proof strategies already do a lot
  - Later we will cover a specific strategy that applies to loops and recursion (mathematical induction)

Set Theory

- Formal treatment dates from late 19^{th} century
- Direct ties between set theory and logic
- Important foundational language

Definition: A set is an unordered collection of objects

$x \in A$: "x is an element of A"
"x is a member of A"
"x is in A"

$x \notin A$: $\neg (x \in A)$

Definitions

- A and B are *equal* if they have the same elements

$$A = B \equiv \forall x\, (x \in A \leftrightarrow x \in B)$$

- A is a *subset* of B if every element of A is also in B

$$A \subseteq B \equiv \forall x\, (x \in A \rightarrow x \in B)$$

Empty Set and Power Set

- Empty set $\emptyset$ does not contain any elements
- Power set of a set A = set of all subsets of A

$$\mathcal{P}(A) = \{ B : B \subseteq A \}$$

Cartesian Product : $A \times B$

$$A \times B = \{ (a, b) \mid a \in A \wedge b \in B \}$$

Set operations

$A \cup B = \{ x \mid (x \in A) \vee (x \in B) \}$	union
$A \cap B = \{ x \mid (x \in A) \wedge (x \in B) \}$	intersection
$A - B = \{ x \mid (x \in A) \wedge (x \notin B) \}$	set difference
$A \oplus B = \{ x \mid (x \in A) \oplus (x \in B) \}$	symmetric difference
$\overline{A} = \{ x \mid x \notin A \}$ (with respect to universe U)	complement

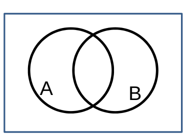
It's Boolean algebra again

- Definition for $\cup$ based on $\vee$
- Definition for $\cap$ based on $\wedge$
- Complement works like $\neg$

De Morgan's Laws

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

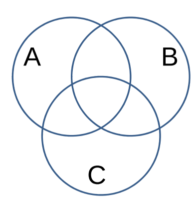
$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

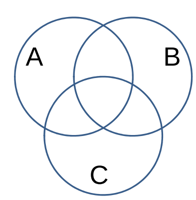


Proof technique:
To show C = D show
$x \in C \rightarrow x \in D$ and
$x \in D \rightarrow x \in C$

Distributive Laws

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$





Characteristic vectors: Representing sets using bits

- Suppose universe U is {1,2,...,n}
- Can represent set $B \subseteq U$ as a vector of bits:

  $b_1 b_2 \ldots b_n$ where $b_i=1 \equiv (i \in B)$

  $b_i=0 \equiv (i \notin B)$

  – Called the *characteristic vector* of set B
- Given characteristic vectors for A and B
  – What is characteristic vector for $A \cup B$? $A \cap B$?

Boolean operations on bit-vectors: (a.k.a. bit-wise operations)

- 01101101 Java: **z=x|y**

  ∨ <u>00110111</u>

  01111111

- 00101010 Java: **z=x&y**

  ∧ <u>00001111</u>

  00001010

- 01101101 Java: **z=x^y**

  ⊕ <u>00110111</u>

  01011010

A simple identity

- If x and y are bits: $(x \oplus y) \oplus y$ = ?
- What if x and y are bit-vectors?

Private Key Cryptography

- Alice wants to be able to communicate message secretly to Bob so that eavesdropper Eve who hears their conversation, cannot tell what Alice's message is
- Alice and Bob can get together and privately share a secret key K ahead of time.

One-time pad

- Alice and Bob privately share random n-bit vector K
  – Eve does not know K
- Later, Alice has n-bit message m to send to Bob
  – Alice computes $C = m \oplus K$
  – Alice sends C to Bob
  – Bob computes $m = C \oplus K$ which is $(m \oplus K) \oplus K$
- Eve cannot figure out m from C unless she can guess K

Unix/Linux file permissions

- **ls –l**

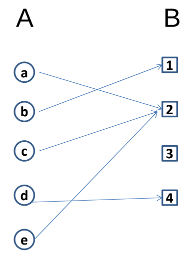
```
drwxr-xr-x ... Documents/
-rw-r--r-- ... file1
```

- Permissions maintained as bit vectors
  – Letter means bit is 1 **-** means bit is 0.

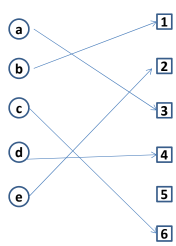
Functions review

- A *function* from *A* to *B*
  - an assignment of exactly one element of *B* to each element of *A.*
  - We write *f: A→B.*
  - "Image of *a*" = *f(a)*
- *Domain* of *f* : A
- *Range* of *f* = set of all images of elements of A

Image, Preimage



Is this a function? one-to-one? onto?



Russell's Paradox

$$S = \{ x \mid x \notin x \}$$