Discrete Structures for Computer Science

William Garrison
bill@cs.pitt.edu
6311 Sennott Square

Lecture #9: Set Identities and Functions

Based on materials developed by Dr. Adam Lee
Today’s Topics

Set identities
- Methods of proof
- Relationships to logical equivalences

Functions
- Important definitions
- Relationships to sets, relations
- Special of functions
Set identities help us manipulate complex expressions

Recall from last lecture that set operations bear a striking resemblance to logical operations

- Disjunction ($\lor$) and set union ($\cup$)
- Conjunction ($\land$) and set intersection ($\cap$)
- Negation ($\neg$) and complement ($\overline{\phantom{1}}$)

Just as logical equivalences helped us manipulate logical expressions, **set identities** help us simplify and understand complex set definitions.
Some important set identities

<table>
<thead>
<tr>
<th>Identity</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \cup \emptyset = A$</td>
<td>Identity laws</td>
</tr>
<tr>
<td>$A \cap U = A$</td>
<td>Domination laws</td>
</tr>
<tr>
<td>$A \cup U = U$</td>
<td>Idempotent laws</td>
</tr>
<tr>
<td>$A \cap \emptyset = \emptyset$</td>
<td>Complementation law</td>
</tr>
<tr>
<td>$A \cup (B \cup C) = (A \cup B) \cup C$</td>
<td>Commutative laws</td>
</tr>
<tr>
<td>$A \cap (B \cap C) = (A \cap B) \cap C$</td>
<td>Associative laws</td>
</tr>
</tbody>
</table>

We don’t have commutative or associative laws for set difference!
Some important set identities

<table>
<thead>
<tr>
<th>Identity</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Distributive laws</td>
</tr>
<tr>
<td></td>
<td>DeMorgan’s laws</td>
</tr>
<tr>
<td></td>
<td>Absorption laws</td>
</tr>
<tr>
<td></td>
<td>Complement laws</td>
</tr>
</tbody>
</table>
There are many ways to prove set identities

Today, we’ll discuss four common methods:

1. Membership tables
2. Logical argument
3. Using set builder notation
4. Applying other known set identities
Membership tables allow us to write proofs like we did using truth tables!

The membership table for an expression has columns for sub-expressions and rows to indicate the ways in which an arbitrary element may or may not be included.

**Example:** A membership table for set intersection

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A ∩ B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

An element is in $A \cap B$ iff it is in both $A$ and $B$. 
Prove that \( A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \)

Since the appropriate columns of the membership table are the same, we can conclude that \( A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \). □
Sometimes, it’s easier to make a logical argument about a set identity.

Recall: \( A = B \iff A \subseteq B \) and \( B \subseteq A \)

As a result, we can prove a set identity by arguing that each side of the equality is a subset of the other.

**Example:** Prove that \( \overline{A \cap B} = \overline{A} \cup \overline{B} \)

1. First prove that \( \overline{A \cap B} \subseteq \overline{A} \cup \overline{B} \)

2. Then prove that \( \overline{A} \cup \overline{B} \subseteq \overline{A \cap B} \)

Let’s see how this is done...
Prove that $\overline{A \cap B} = \overline{A} \cup \overline{B}$
Note: Differences between $\subseteq$ and $\in$

Recall that $A \subseteq B$ if $A$ is a **subset** of $B$, whereas $a \in A$ means that $a$ is an **element** of $A$.

*Examples:*

- Is $\{1\} \in \{1, 2, 3\}$?
- Is $\{1\} \subseteq \{1, 2, 3\}$?
- Is $1 \in \{1, 2, 3\}$?
- Is $\{2, 3\} \subseteq \{1, \{2, 3\}, \{4, 5\}\}$?
- Is $\{2, 3\} \in \{1, \{2, 3\}, \{4, 5\}\}$?
- Is $\emptyset \in \{1, 2, 3\}$?
- Is $\emptyset \subseteq \{1, 2, 3\}$?
Be careful when computing power sets

**Question:** What is \( P(\{1, 2, \{1, 2\}\}) \)?

**Note:** The set \( \{1, 2, \{1, 2\}\} \) has three elements

- 1
- 2
- \( \{1, 2\} \)

So, we need all combinations of those elements:

- \( \emptyset \)
- \{1\}
- \{2\}
- \{\{1,2\}\}
- \{1, 2\}
- \{1, \{1,2\}\}
- \{2, \{1,2\}\}
- \{1, 2, \{1,2\}\}

\[
P(\{1, 2, \{1,2\}\}) = \{\emptyset, \{1\}, \{2\}, \{\{1,2\}\}, \{1, 2\}, \{1, \{1,2\}\}, \{2, \{1,2\}\}, \{1, 2, \{1,2\}\}\}
\]

This power set has \( 2^3 = 8 \) elements.
We can use set builder notation and logical definition to make very precise proofs

**Example:** Prove that $\overline{A \cap B} = \overline{A} \cup \overline{B}$

**Proof:**

1. $\overline{A \cap B} = \{x \mid x \notin A \cap B\}$  
   \hspace{1cm} \text{Def'n of complement}
2. $= \{x \mid \neg(x \in (A \cap B))\}$  
   \hspace{1cm} \text{Def'n of } \notin
3. $= \{x \mid \neg(x \in A \land x \in B)\}$  
   \hspace{1cm} \text{Def'n of } \cap
4. $= \{x \mid \neg(x \in A) \lor \neg(x \in B)\}$  
   \hspace{1cm} \text{DeMorgan's law}
5. $= \{x \mid x \notin A \lor x \notin B\}$  
   \hspace{1cm} \text{Def'n of } \notin
6. $= \{x \mid x \in \overline{A} \lor x \in \overline{B}\}$  
   \hspace{1cm} \text{Def'n of complement}
7. $= \{x \mid x \in \overline{A} \cup \overline{B}\}$  
   \hspace{1cm} \text{Def'n of } \cup
8. $= \overline{A} \cup \overline{B}$  
   \hspace{1cm} \text{Set builder notation}
We can also construct proofs by repeatedly applying known set identities.

**Example:** Prove that \( A \cup (B \cap C) = (C \cup B) \cap \overline{A} \)

**Proof:**

1. \[ A \cup (B \cap C) = A \cap (B \cap C) \quad \text{DeMorgan’s law} \]
2. \[ = A \cap (B \cup C) \quad \text{DeMorgan’s law} \]
3. \[ = (B \cup C) \cap A \quad \text{Commutative law} \]
4. \[ = (C \cup B) \cap A \quad \text{Commutative law} \]

Note how similar this process is to that of proving logical equivalences using known logical equivalences.
Problem 1: Prove DeMorgan’s law for complement over intersection using a membership table.

Problem 2: Prove the complementation law using set builder notation.
Sets give us a way to formalize the concept of a function

**Definition:** Let A and B be nonempty sets. A function, $f$, is an assignment of exactly one element of set B to each element of set A.

**Note:** We write $f : A \to B$ to denote that $f$ is a function from A to B.

**Note:** We say that $f(a) = b$ if the element $a \in A$ is mapped to the unique element $b \in B$ by the function $f$. 

![Diagram](image-url)
1. Explicitly
   - $f : \mathbb{Z} \rightarrow \mathbb{Z}$
   - $f(x) = x^2 + 2x + 1$

2. Using a programming language
   - `int min(int x, int y) = { x < y ? return x : return y; }`

3. Using a relation
   - Let $S = \{\text{Anna, Brian, Christine}\}$
   - Let $G = \{A, B, C, D, F\}$
More terminology

The **domain** of a function is the set that the function maps from, while the **codomain** is the set that is mapped to.

If \( f(a) = b \), \( b \) is called the **image** of \( a \), and \( a \) is called the **preimage** of \( b \).

The **range** of a function \( f : A \to B \) is the set of all images of elements of \( A \).

**Domain** = \( S = \{ \text{Anna, Brian, Christine} \} \)

**Codomain** = \( G = \{ A, B, C, D, E \} \)

**Range** = \( \{ A, C \} \)
What are the domain, codomain, and range of the following functions?

1. \( f : \mathbb{Z} \rightarrow \mathbb{Z}, \ f(x) = x^3 \)
   - Domain:
   - Codomain:
   - Range:

2. \( g : \mathbb{R} \rightarrow \mathbb{R}, \ g(x) = x - 2 \)
   - Domain:
   - Codomain:
   - Range:

3. int foo(int x, int y) = { return (x*y)%2; }
   - Domain:
   - Codomain:
   - Range:
A one-to-one function never assigns the same image to two different elements

**Definition**: A function \( f : A \to B \) is one-to-one, or injective, iff \( \forall x, y \in A \ [(f(x) = f(y)) \to (x = y)] \)

Are the following functions *injections*?

- \( f : \mathbb{R} \to \mathbb{R}, \ f(x) = x + 1 \) **Yes**
- \( f : \mathbb{Z} \to \mathbb{Z}, \ f(x) = x^2 \) **No**
- \( f : \mathbb{R}^+ \to \mathbb{R}^+, \ f(x) = \sqrt{x} \) **Yes**
- \( f : S \to G \) **No**

\[ f : S \to G \]

- Anna \( \bullet \) \( \rightarrow \) A
- Brian \( \bullet \) \( \rightarrow \) B
- Christine \( \bullet \) \( \rightarrow \) C

\[ \bullet \] D, E
An onto function “uses” every element of its codomain

**Definition:** We call a function $f : A \rightarrow B$ **onto**, or **surjective**, iff for every element $b \in B$, there is some element $a \in A$ such that $f(a) = b$.

Think about an onto function as “covering” the entirety of its codomain.

The following function is a **surjection**:
Are the following functions one-to-one, onto, both, or neither?

**$f : A \rightarrow B$**

- **Neither!**

  (Aside: Functions that are both one-to-one and onto are called *bijections*).

- **One-to-one and onto**

**$f : A \rightarrow B$**

- **One-to-one**

- **Onto**
**Definition:** If $f : A \to B$ is a bijection, the inverse of $f$ is the function $f^{-1} : B \to A$ that assigns to each $b \in B$ the unique value $a \in A$ such that $f(a) = b$. That is, $f^{-1}(b) = a$ iff $f(a) = b$.

**Graphically:**

Note: Only a bijection can have an inverse. (Why?)
Do the following functions have inverses?

1. \( f : \mathbb{R} \rightarrow \mathbb{R}, \ f(x) = x^2 \)

2. \( g : \mathbb{Z} \rightarrow \mathbb{Z}, \ g(x) = x + 1 \)

1. \( h : A \rightarrow B \)
Definition: Given two functions $g : A \rightarrow B$ and $f : B \rightarrow C$, the composition of $f$ and $g$, denoted $f \circ g$, is defined as $(f \circ g)(x) = f(g(x))$.

Note: For $f \circ g$ to exist, the domain of $f$ must be a subset of the codomain of $g$. 
Can the following functions be composed? If so, what is their composition?

Let \( f : A \rightarrow A \) such that \( f(a) = b, \ f(b) = c, \ f(c) = a \)
\[ g : B \rightarrow A \text{ such that } g(1) = b, \ g(4) = a \]

1. \((f \circ g)(x)\)?
2. \((g \circ f)(x)\)?

Let \( f : \mathbb{Z} \rightarrow \mathbb{Z}, \ f(x) = 2x + 1 \)
\[ g : \mathbb{Z} \rightarrow \mathbb{Z}, \ g(x) = x^2 \]

1. \((f \circ g)(x)\)?
2. \((g \circ f)(x)\)?

**Note:** There is **not** a guarantee that \((f \circ g)(x) = (g \circ f)(x)\).
**Important functions**

**Definition:** The floor function maps a real number $x$ to the largest integer $y$ that is not greater than $x$. The floor of $x$ is denoted $\lfloor x \rfloor$.

**Definition:** The ceiling function maps a real number $x$ to the smallest integer $y$ that is not less than $x$. The ceiling of $x$ is denoted $\lceil x \rceil$.

**Examples:**

- $\lfloor 1.2 \rfloor = 1$
- $\lfloor 7.0 \rfloor = 7$
- $\lfloor -42.24 \rfloor = -43$

- $\lceil 1.2 \rceil = 2$
- $\lceil 7.0 \rceil = 7$
- $\lceil -42.24 \rceil = -42$
We actually use floor and ceiling quite a bit in computer science...

**Example:** A byte, which holds 8 bits, is typically the smallest amount of memory that can be allocated on most systems. How many bytes are needed to store 123 bits of data?

**Answer:** We need \( \lceil \frac{123}{8} \rceil = \lceil 15.375 \rceil = 16 \) bytes

**Example:** How many 1400 byte packets can be transmitted over a 14.4 kbps modem in one minute?

**Answer:** A 14.4 kbps modem can transmit \( 14,400 \times 60 = 864,000 \) bits per minute. Therefore, we can transmit \( \lfloor \frac{864,000}{(1400 \times 8)} \rfloor = \lfloor 77.1428571 \rfloor = 77 \) packets.
Group work!

Problem 3: Find the domain and range of each of the following functions.

a. The function that determines the number of zeros in some bit string
b. The function that maps an English word to its two rightmost letters
c. The function that assigns to an integer the sum of its individual digits

Problem 4: Compute the following

a. \( \lfloor 435.5 \rfloor \)
b. \( \lceil \frac{89}{90} \rceil \)
c. \( \lceil 5.5 + \lfloor 1.22 \rfloor \rceil \)
Final thoughts

- Set identities are useful tools!

- We can prove set identities in a number of (equivalent) ways

- Sets are the basis of functions, which are used throughout computer science and mathematics

- Next time:
  - Summations (Section 2.4)