Introduction

COT 4210 Discrete Structures II

Summer 2025

Department of Computer Science

Dr. Steinberg

What is the course all about? (0.1 Sipser Textbook)

- We will be diving into the world of **theory of computation**. There are three components.
    - Automata Theory
        - *Why are they important and useful?*
    - Computability Theory
        - *Why are some problems just not solvable?*
    - Complexity Theory
        - *What makes some problems computationally hard and others easy?*
- They all linked to the question: *What are the fundamental capabilities and limitations of computers?*

Review of Mathematical Essentials

0.2 Sipser Textbook

Sets
Given elements *x* and *y*, and sets *A* and *B*

Containment

- $x \in A$ - A contains x.
- $x \notin A$ - A doesn't contain x.
- $A = \{x, y\}$ - A contains only x and y.
- $A = \{x \mid x \in \mathbb{N}, x > 50\}$ - A contains the natural numbers higher than 50.

Operators

- $A \cup B$ – union
- $A \cap B$ – intersection
- $\overline{A}$- complement

Subsets

- $A \subseteq B$ - A is a subset of B.
  - $\forall\, x \in A, x \in B$
- $A \subsetneq B$ - A is a proper subset of B ($\subset$).
  - $\forall\, x \in A, x \in B$ and $A \neq B$.
- The *power set* of A is the set of all subsets of A.

Common sets

- $\mathbb{Z}$ – the set of all integers
- $\mathbb{N}$ – the set of all natural numbers
- $\varnothing$ or ϕ - the empty set

Sequences

- Sequences
  - Like ordered sets
  - Finite sequences are called *k*-tuples
  - 2-tuples are also known as *ordered pairs*
- Cartesian products of sets:
  - $A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$
  - Can take it of any number of sets
  - $A \times A = A^2$, $A \times A \times A = A^3$, etc.

Functions

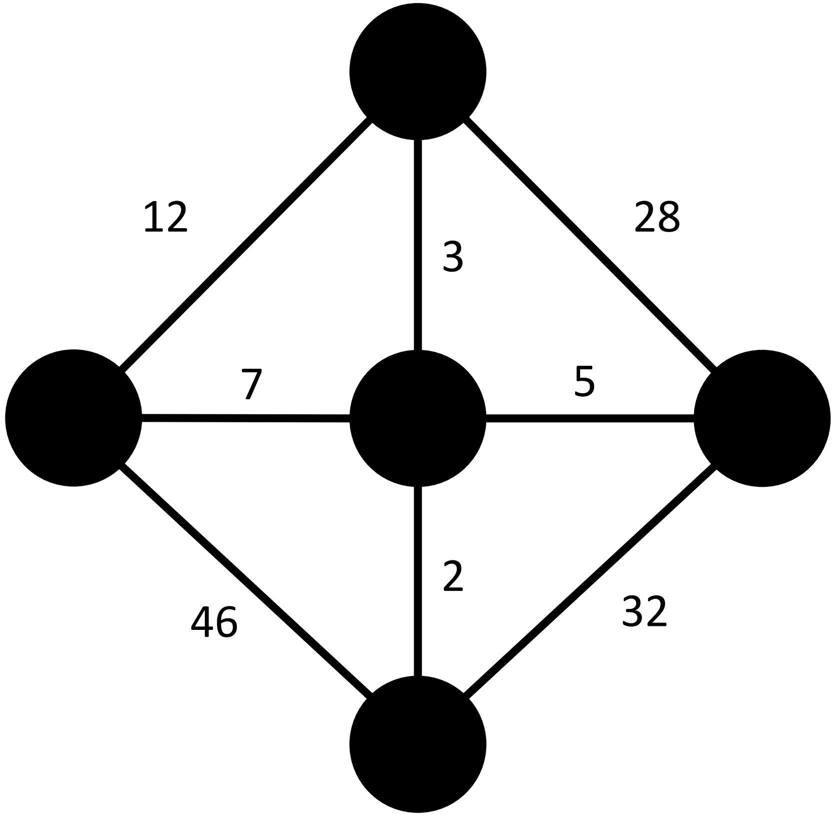
- Functions
  - Map a *domain* onto a *range*
  - *n*-ary functions take *n* arguments
  - *f*: D $\rightarrow$ R
    - *abs*: $\mathbb{Z} \rightarrow \mathbb{N}$
    - *add*: $\mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$
- A function is…
  - *One-to-one* (an *injection*) if it maps every element of the range from at most one element of the domain
  - *Onto* (a *surjection*) if it maps every element of the range from at least one element of the domain
  - A *bijection* if every element of the range is mapped by exactly one element of the domain

Relations

- A *predicate* or *property* is a function with range {TRUE, FALSE}
- A property with a domain of n-tuples A^n is an n-ary relation
- Binary relations are common, and like binary functions, we use infix notations for them
- Let R be a binary relation on A^2. R is:
  - *Reflexive* if $\forall x \in A, x R x$
  - *Symmetric* if $x R y \rightarrow y R x$
  - *Transitive* if $(x R y, y R z) \rightarrow x R z$
  - An *equivalence* relation if it is reflexive, symmetric and transitive

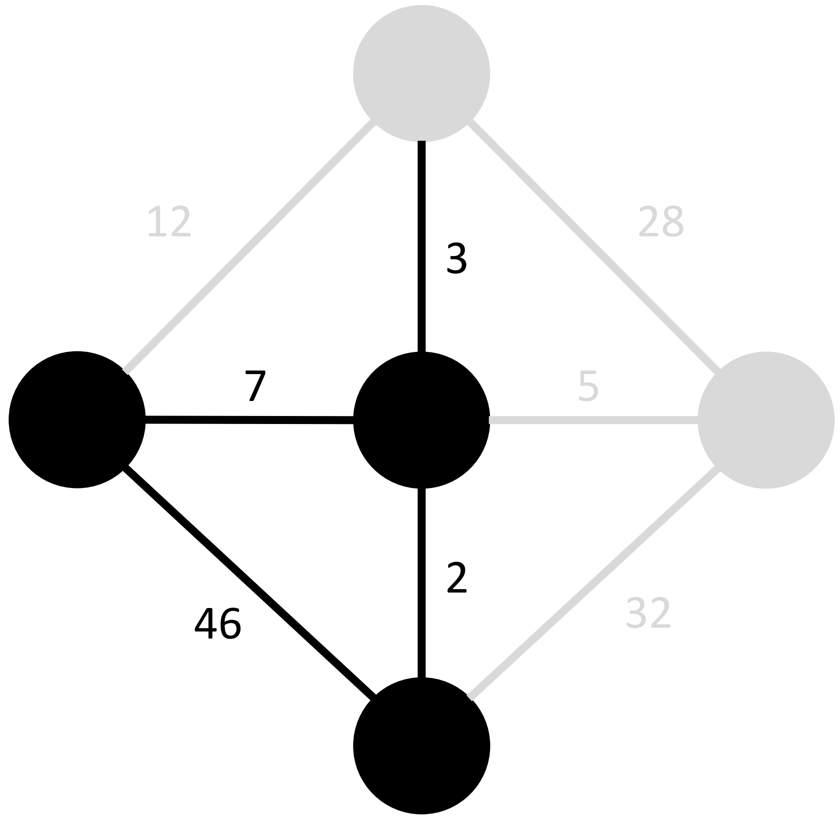
Graphs: Undirected Graphs

- An undirected *graph* is a collection of *nodes* (or *vertices*) and *edges* that connect them
  - The *degree* of a node is the number of edges that connect to that node
  - Edges are unique – you can't have two edges between the same pair of nodes
  - Edges can also be labeled



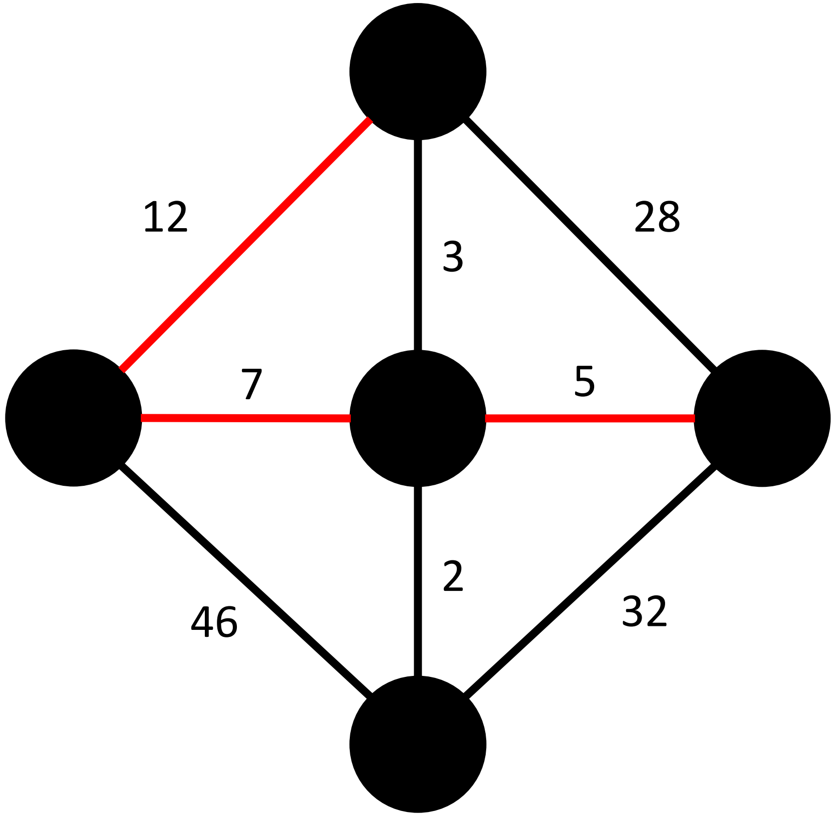
Graphs: Subgraphs

- An undirected *graph* is a collection of *nodes* (or *vertices*) and *edges* that connect them
  - The *degree* of a node is the number of edges that connect to that node
  - Edges are unique – you can't have two edges between the same pair of nodes
  - Nodes can have *self-loops*
  - Edges can also be labeled
- A graph *G* is a *subgraph* of graph *H* if it has a subset of *H*'s nodes and all the related edges



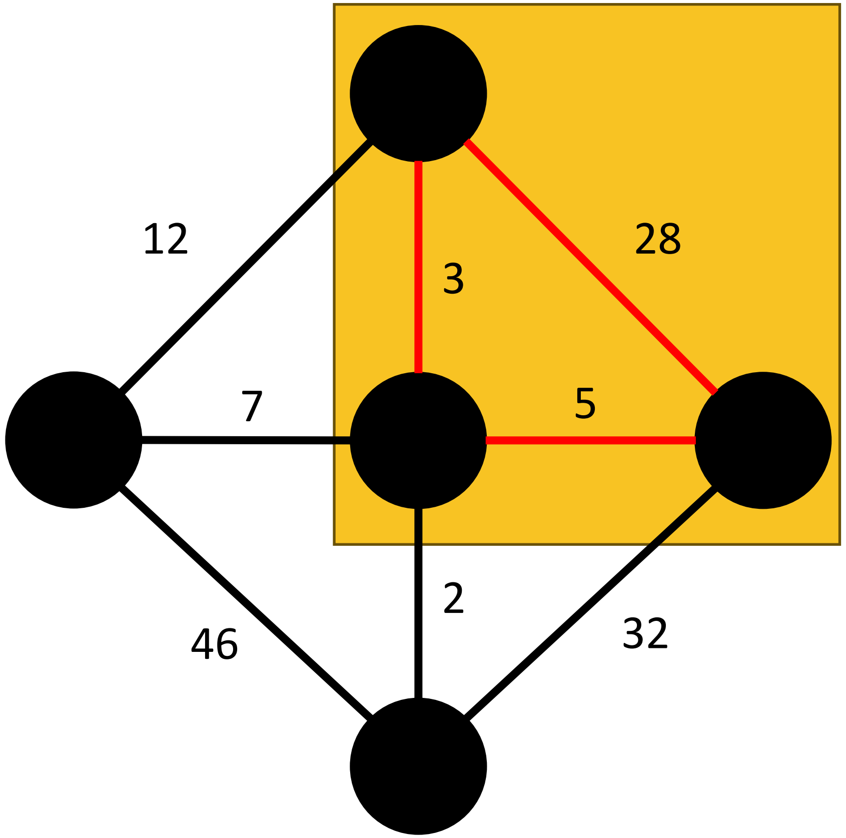
Graphs: Paths

- A *path* is a sequence of nodes connected by edges
  - A *simple path* doesn't repeat any nodes
  - A graph is *connected* if every two nodes have a path



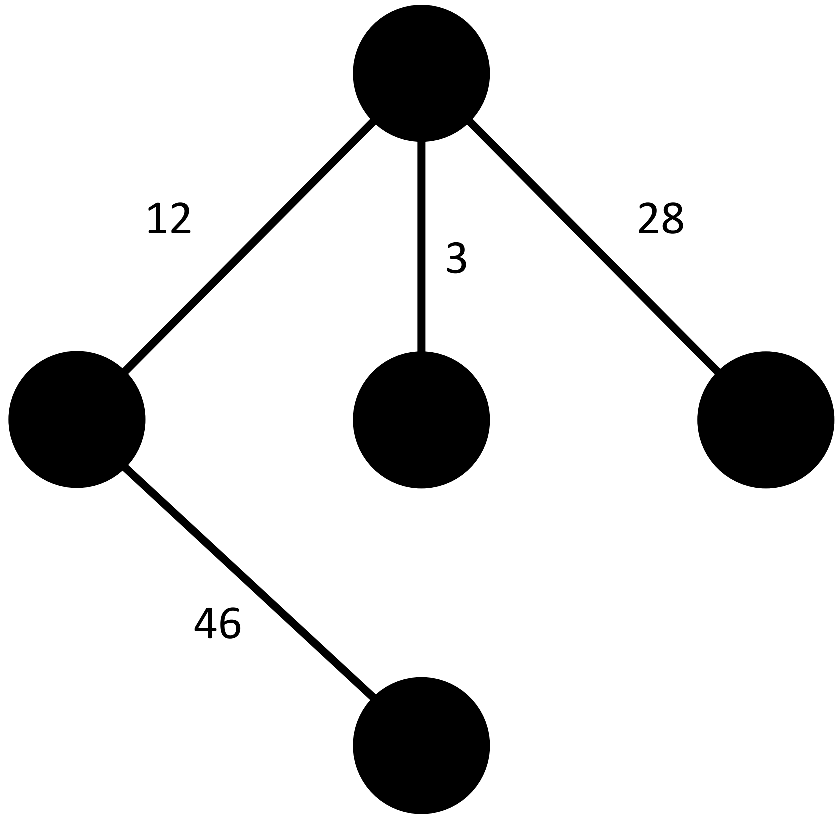
Graphs: Cycles

- A *path* is a sequence of nodes connected by edges
    - A *simple path* doesn't repeat any nodes
    - A graph is *connected* if every two nodes have a path
    - A path is a *cycle* if it starts and ends on the same node
    - A *simple cycle* contains at least three nodes and repeats only the first/last



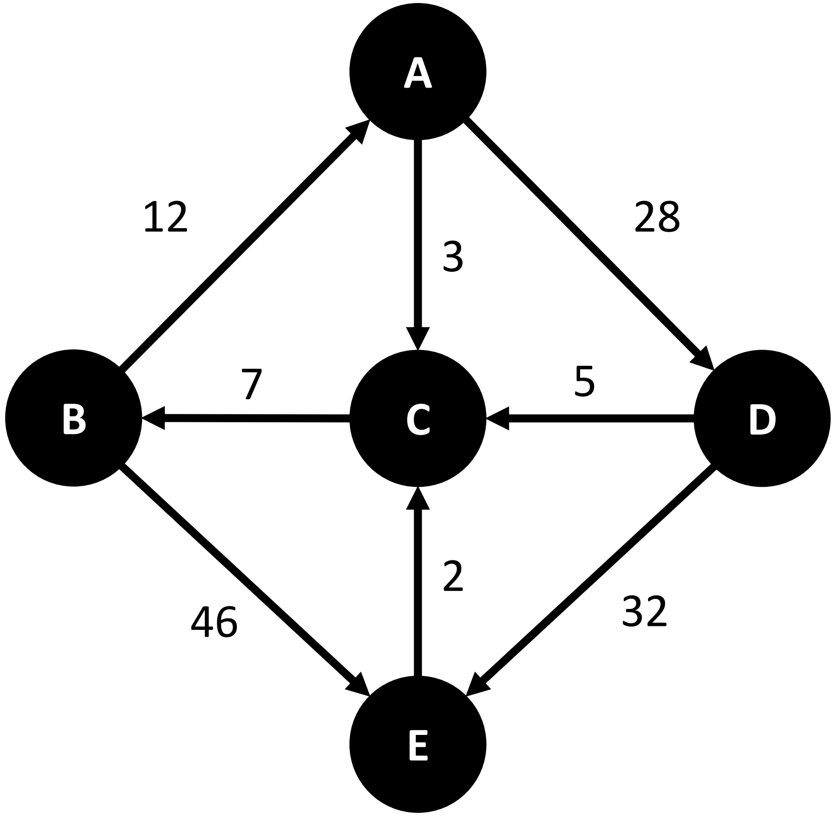
Graphs: Trees

- A *path* is a sequence of nodes connected by edges
  - A *simple path* doesn't repeat any nodes
  - A graph is *connected* if every two nodes have a path
  - A path is a *cycle* if it starts and ends on the same node
  - A *simple cycle* contains at least three nodes and repeats only the first/last
  - A graph is a *tree* if it is connected and has no simple cycles



Graphs: Directed Graphs

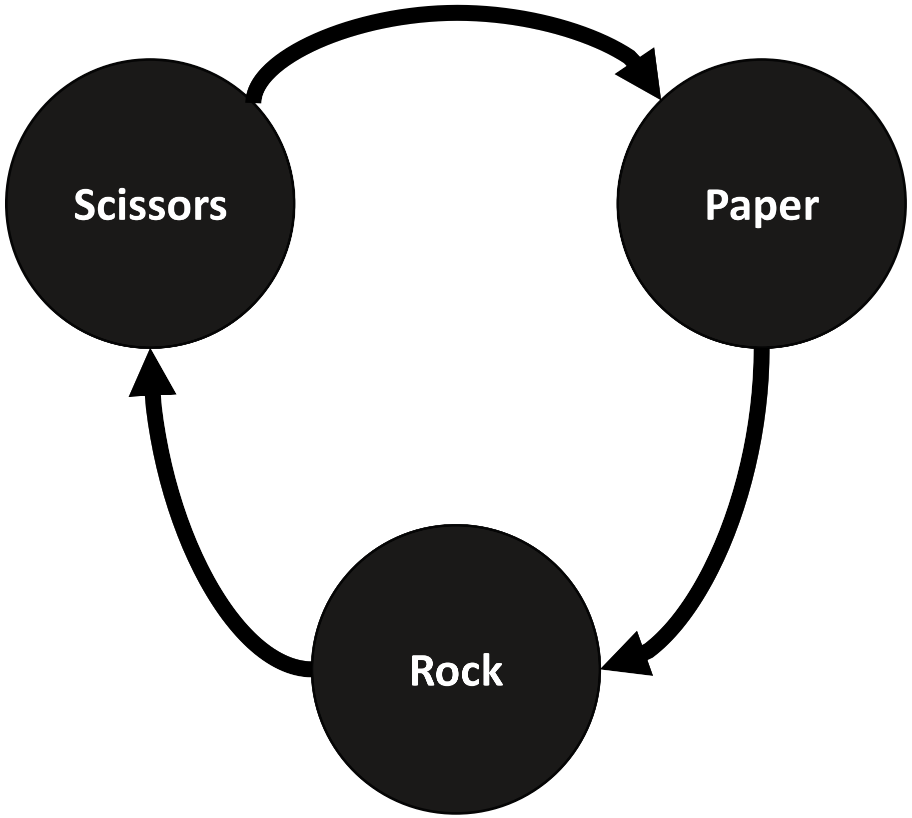
- A *directed graph* is a graph with arrows instead of lines
  - Edges between nodes *i* and *j* are *ordered* pairs (*i, j*)
  - *Directed paths* are paths that follow the direction of the edges
  - A directed graph is *strongly connected* if every pair of nodes has a directed path
  - Nodes can have *self-loops*



Directed Graphs and Binary Relations

Consider the relation "beats"

beats	Scissors	Paper	Rock
Scissors	FALSE	TRUE	FALSE
Paper	FALSE	FALSE	TRUE
Rock	TRUE	FALSE	FALSE



Strings

- An *alphabet* is a non-empty, finite set of *symbols*
  - The symbols Σ and Γ are used to denote an alphabet. We will see this throughout the course.
- A *string* over an alphabet is a finite sequence of symbols from that alphabet
- Strings have *length*, like any sequence; the empty string ε is the string with length 0
- A *language* is a set of strings over a given alphabet
- ***Do not confound the empty language with the empty string***

Strings cont.

Given strings S, T, U and V, we write:

- S_i to denote the i^{th} symbol in S
- ST to denote the *concatenation* of S and T
- S^R to denote the *reverse* of S

We also say…

- S is a *substring* of V if $\exists\ T, U \ni TSU = V$
  - …and a *proper substring* if $S \neq V$
- S is a *prefix* of V if $\exists\ T \ni ST = V$
  - …and a *proper prefix* if $S \neq V$
- S is a *suffix* of V if $\exists\ T \ni TS = V$
  - …and a *proper suffix* if $S \neq V$

Proofs

0.3-0.4 Sipser Textbook

Definitions, Theorems, and Proofs

- **Definitions** describe the mathematical objects and ideas we want to work with
- **Statements** or **assertions** are things we say about mathematics; they can be true or false
- **Proofs** are unassailable logical demonstrations that statements are true
- **Theorems** are statements that have been proven true
- **Lemmas** are theorems that are only any good for proving other theorems
- **Corollaries** are follow-on theorems that are easy to prove once you prove their parent theorems

How To Prove Something

1. Understand the statement
2. Convince *yourself* of whether it is true or false
3. Work out its implications until you have a general sense of *why* it is true or false
    - “Warm fuzzy feelings” don’t prove anything – but they can help you get *ready* to prove something
4. Break down any sub-cases you will need to prove
    - After this you may need to cycle back to step 2
5. Get started

Sipser Textbook Proof Note

The book uses a highly narrative proof format. There are of course other valid techniques and we are going to observe two of them.

Quasi-Narrative Format
Prove $\overline{A \cup B} = \overline{A} \cap \overline{B}$

We can show this by showing $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$ and $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$.

1. Suppose $x \in \overline{A \cup B}$.
2. Then by definition of complement, $x \notin A \cup B$.
3. Then by definition of union, $x \notin A$ and $x \notin B$.
4. Then by def. of complement, $x \in \overline{A}$ and $x \in \overline{B}$.
5. Then by definition of intersection, $x \in \overline{A} \cap \overline{B}$.
6. We have shown that if $x \in \overline{A \cup B}$, $x \in \overline{A} \cap \overline{B}$.
7. Hence by definition of subset, $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$.

Now suppose $x \in \overline{A} \cap \overline{B}$.

1. Then by def. of intersection, $x \in \overline{A}$ and $x \in \overline{B}$.
2. Then by def. of complement, $x \notin A$ and $x \notin B$.
3. Then by definition of union, $x \notin A \cup B$.
4. Then by definition of complement, $x \in \overline{A \cup B}$.
5. We have shown that if $x \in \overline{A} \cap \overline{B}$, $x \in \overline{A \cup B}$.
6. Hence by definition of subset, $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$.
7. We have shown that $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$ and $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$.
8. Hence by set equality, $\overline{A \cup B} = \overline{A} \cap \overline{B}$, QED.

Two-Column Format
Prove $\overline{A \cup B} = \overline{A} \cap \overline{B}$

1. STS $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}, \overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$	set equality
2. Let $x \in \overline{A \cup B}$	
3. $\therefore x \notin A \cup B$	complement
4. $\therefore x \in \overline{A}$, $x \in \overline{B}$	union
5. $\therefore x \in \overline{A} \cap \overline{B}$	intersection
6. $x \in \overline{A \cup B} \Rightarrow x \in \overline{A} \cap \overline{B}$	2-5
7. $\therefore \overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$	subset
8. Let $x \in \overline{A} \cap \overline{B}$	
9. $\therefore x \in \overline{A}, x \in \overline{B}$	intersection
10. $\therefore x \notin A, x \notin B$	complement
11. $\therefore x \notin A \cup B$	union
12. $\therefore x \in \overline{A \cup B}$	complement
13. $x \in \overline{A} \cap \overline{B} \Rightarrow x \in \overline{A \cup B}$	9-13
14. $\therefore \overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$	subset
15. $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}, \overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$	7, 14
16. $\therefore \overline{A \cup B} = \overline{A} \cap \overline{B}$	set equality

Types of Proofs

Direct Argument

Construction

- Prove something exists by showing how to make it

Contradiction

- Prove something is true by showing it can't be false

Weak Induction

- Show that a statement is true for the case of 0
- Show that *if* it's true for the case of i, *then* it's true for the case of $i + 1$

Strong Induction

- Show that a statement is true for the case of 0
- Show that *if* it's true for all of the cases $< i$, *then* it's true for the case of i

Acknowledgement

Some Notes and content come from Dr. Gerber, Dr. Hughes, and Mr. Guha's COT4210 class and the Sipser Textbook, *Introduction to the Theory of Computation*, 3rd ed., 2013