Week 5 at a glance

We will be learning and practicing to:

- Clearly and unambiguously communicate computational ideas using appropriate formalism. Translate across levels of abstraction.
  - Translating between symbolic and English versions of statements using precise mathematical language
  - Using appropriate signpost words to improve readability of proofs, including 'arbitrary' and 'assume'
- Know, select and apply appropriate computing knowledge and problem-solving techniques. Reason about computation and systems. Use mathematical techniques to solve problems. Determine appropriate conceptual tools to apply to new situations. Know when tools do not apply and try different approaches. Critically analyze and evaluate candidate solutions.
  - Judging logical equivalence of compound propositions using symbolic manipulation with known equivalences, including DeMorgan's Law
  - Writing the converse, contrapositive, and inverse of a given conditional statement
  - Determining what evidence is required to establish that a quantified statement is true or false
  - Evaluating quantified statements about finite and infinite domains
- Apply proof strategies, including direct proofs and proofs by contradiction, and determine whether a proposed argument is valid or not.
  - Identifying the proof strategies used in a given proof
  - Identifying which proof strategies are applicable to prove a given compound proposition based on its logical structure
  - Carrying out a given proof strategy to prove a given statement
  - Carrying out a universal generalization argument to prove that a universal statement is true
  - Using proofs as knowledge discovery tools to decide whether a statement is true or false

TODO:

Project due May 7, 2024. No review quiz this week.

Test 1, in class this week, on Friday May 3, 2024. The test covers material in Weeks 1 through 4 and Monday of Week 5. To study for the exam, we recommend reviewing class notes (e.g. annotations linked on the class website, podcast, supplementary video from the class website), reviewing homework (and its posted sample solutions), and in particular *working examples* (extra examples in lecture notes, review quizzes, discussion examples) and getting feedback (office hours and Piazza). Some practice questions (and their solutions) are available on the class website, linked from Week 5 and from the Assignments page.

Week 5 Monday: Nested Quantifiers

Recall the definitions: The set of RNA strands S is defined (recursively) by:

Basis Step:	$\mathtt{A} \in S, \mathtt{C} \in S, \mathtt{U} \in S, \mathtt{G} \in S$
Recursive Step:	If $s \in S$ and $b \in B$, then $sb \in S$

where sb is string concatenation.

The function *rnalen* that computes the length of RNA strands in S is defined recursively by:

		$rnalen : S$	$\to \mathbb{Z}^+$
Basis Step:	If $b \in B$ then	$rnalen(b)$	$= 1$
Recursive Step:	If $s \in S$ and $b \in B$, then	$rnalen(sb)$	$= 1 + rnalen(s)$

The function *basecount* that computes the number of a given base b appearing in a RNA strand s is defined recursively by:

		$basecount : S \times B$	$\to \mathbb{N}$
Basis Step:	If $b_1 \in B, b_2 \in B$	$basecount(\ (b_1, b_2)\ )$	$= \begin{cases} 1 & \text{when } b_1 = b_2 \\ 0 & \text{when } b_1 \neq b_2 \end{cases}$
Recursive Step:	If $s \in S, b_1 \in B, b_2 \in B$	$basecount(\ (sb_1, b_2)\ )$	$= \begin{cases} 1 + basecount(\ (s, b_2)\ ) & \text{when } b_1 = b_2 \\ basecount(\ (s, b_2)\ ) & \text{when } b_1 \neq b_2 \end{cases}$

Alternating nested quantifiers

$$\forall s \in S\ \exists n \in \mathbb{N}\ (\ basecount(\ (s, \mathtt{U})\ ) = n\ )$$

In English: For each strand, there is a nonnnegative integer that counts the number of occurrences of `U` in that strand.

$$\exists n \in \mathbb{N}\ \forall s \in S\ (\ basecount(\ (s, \mathtt{U})\ ) = n\ )$$

In English: There is a nonnnegative integer that counts the number of occurrences of `U` in every strand.

Are these statements true or false?

$$\forall s \in S \; \exists b \in B \; (\; basecount(\; (s, b) \;) = 3 \;)$$

In English: For each RNA strand there is a base that occurs 3 times in this strand.

Write the negation and use De Morgan's law to find a logically equivalent version where the negation is applied only to the BC predicate (not next to a quantifier).

Is the original statement **True** or **False**?

Proof strategies

When a predicate $P(x)$ is over a **finite** domain:

- To show that $\forall x P(x)$ is true: check that $P(x)$ evaluates to T at each domain element by evaluating over and over. This is called "Proof of universal by **exhaustion**".
- To show that $\forall x P(x)$ is false: find a **counterexample**, a domain element where $P(x)$ evaluates to F.
- To show that $\exists x P(x)$ is true: find a **witness**, a domain element where $P(x)$ evaluates to T.
- To show that $\exists x P(x)$ is false: check that $P(x)$ evaluates to F at each domain element by evaluating over and over. DeMorgan's Law gives that $\neg \exists x P(x) \quad \equiv \quad \forall x \neg P(x)$ so this amounts to a proof of universal by exhaustion.

New! Proof by universal generalization: To prove that $\forall x \, P(x)$ is true, we can take an arbitrary element e from the domain of quantification and show that $P(e)$ is true, without making any assumptions about e other than that it comes from the domain.
An **arbitrary** element of a set or domain is a fixed but unknown element from that set.

Suppose $P(x)$ is a predicate over a domain D.

1. True or False: To translate the statement "There are at least two elements in D where the predicate P evaluates to true", we could write

$$\exists x_1 \in D \, \exists x_2 \in D \, (P(x_1) \wedge P(x_2))$$

2. True or False: To translate the statement "There are at most two elements in D where the predicate P evaluates to true", we could write

$$\forall x_1 \in D \, \forall x_2 \in D \, \forall x_3 \in D \; (\; (\; P(x_1) \wedge P(x_2) \wedge P(x_3) \;) \rightarrow (\; x_1 = x_2 \vee x_2 = x_3 \vee x_1 = x_3 \;) \;)$$

Week 5 Wednesday: Proof Strategies and Sets

Definitions:

A **set** is an unordered collection of elements. When A and B are sets, $A = B$ (set equality) means

$$\forall x(x \in A \leftrightarrow x \in B)$$

When A and B are sets, $A \subseteq B$ ("A is a **subset** of B") means

$$\forall x(x \in A \rightarrow x \in B)$$

When A and B are sets, $A \subsetneq B$ ("A is a **proper subset** of B") means

$$(A \subseteq B) \wedge (A \neq B)$$

New! Proof of conditional by direct proof: To prove that the conditional statement $p \rightarrow q$ is true, we can assume p is true and use that assumption to show q is true.

New! Proof of conditional by contrapositive proof: To prove that the implication $p \rightarrow q$ is true, we can assume q is false and use that assumption to show p is also false.

New! Proof of disjuction using equivalent conditional: To prove that the disjunction $p \vee q$ is true, we can rewrite it equivalently as $\neg p \rightarrow q$ and then use direct proof or contrapositive proof.

New! Proof by Cases: To prove q, we can work by cases by first describing all possible cases we might be in and then showing that each one guarantees q. Formally, if we know that $p_1 \vee p_2$ is true, and we can show that $(p_1 \rightarrow q)$ is true and we can show that $(p_2 \rightarrow q)$, then we can conclude q is true.

New! Proof of conjunctions with subgoals: To show that $p \wedge q$ is true, we have two subgoals: subgoal (1) prove p is true; and, subgoal (2) prove q is true.

To show that $p \wedge q$ is false, it's enough to prove that $\neg p$.
To show that $p \wedge q$ is false, it's enough to prove that $\neg q$.

To prove that one set is a subset of another, e.g. to show $A \subseteq B$:

To prove that two sets are equal, e.g. to show $A = B$:

Example: $\{43, 7, 9\} = \{7, 43, 9, 7\}$

Prove or **disprove**: $\{\texttt{A}, \texttt{C}, \texttt{U}, \texttt{G}\} \subseteq \{\texttt{AA}, \texttt{AC}, \texttt{AU}, \texttt{AG}\}$

Prove or **disprove**: For some set B, $\emptyset \in B$.

Prove or **disprove**: For every set B, $\emptyset \in B$.

Prove or **disprove**: The empty set is a subset of every set.

Prove or **disprove**: The empty set is a proper subset of every set.

Prove or **disprove**: $\{4, 6\} \subseteq \{n \mid \exists c \in \mathbb{Z}(n = 4c)\}$

Prove or **disprove**: $\{4, 6\} \subseteq \{n \textbf{ mod } 10 \mid \exists c \in \mathbb{Z}(n = 4c)\}$

Consider ..., an **arbitrary** .... **Assume** ..., we **want to show** that .... Which is what was needed, so the proof is complete □.

or, in other words:

Let ... be an **arbitrary** .... **Assume** ..., **WTS** that ... **QED**.

Cartesian product: When A and B are sets,

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

Example: $\{43, 9\} \times \{9, \mathbb{Z}\} =$

Example: $\mathbb{Z} \times \emptyset =$

Union: When A and B are sets,

$$A \cup B = \{x \mid x \in A \vee x \in B\}$$

Example: $\{43, 9\} \cup \{9, \mathbb{Z}\} =$

Example: $\mathbb{Z} \cup \emptyset =$

Intersection: When A and B are sets,

$$A \cap B = \{x \mid x \in A \wedge x \in B\}$$

Example: $\{43, 9\} \cap \{9, \mathbb{Z}\} =$

Example: $\mathbb{Z} \cap \emptyset =$

Set difference: When A and B are sets,

$$A - B = \{x \mid x \in A \wedge x \notin B\}$$

Example: $\{43, 9\} - \{9, \mathbb{Z}\} =$

Example: $\mathbb{Z} - \emptyset =$

Disjoint sets: sets A and B are disjoint means $A \cap B = \emptyset$

Example: $\{43, 9\}, \{9, \mathbb{Z}\}$ are not disjoint

Example: The sets $\mathbb{Z}$ and $\emptyset$ are disjoint

Power set: When U is a set, $\mathcal{P}(U) = \{X \mid X \subseteq U\}$

Example: $\mathcal{P}(\{43, 9\}) =$

Example: $\mathcal{P}(\emptyset) =$