Predicate Logic and Proofs
(Rosen, Sections 1.4, 1.5, 1.7)

TOPICS

• Predicate Logic
• Nested Quantifiers
• More Proofs

Predicate Logic

- Some statements cannot be expressed in propositional logic, such as:
  - All men are mortal.
  - Some trees have needles.
  - X > 3.
- Predicate logic can express these statements and make inferences on them.

Recap motivation: are the two snippets “the same”?

```java
float x1=0, x2=0, y1=0, y2=0;
// Some code that assigns values to these variables (don't count on them all being zero)
if ((x1 > x2) || !(y1 > y2) || (x1 >= y2))
    System.out.println("Call the paintBlue method");
else
    System.out.println("Call the paintRed method");
```

Let’s build towards the math

- We want to reason about Boolean expressions
- They are built out of numbers (and also strings) and variables and operators
- Operators as functions:
  - The comparison operator > maps two numbers to a Boolean value: \( > : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B} \)
  - So do all comparison/relation operators
  - Boolean operators (||, &&)? They map two Booleans to a Boolean value: \( \& \& : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B} \)
Only Boolean variables/operators inside the condition

```java
float x1=0, x2=0, y1=0, y2=0;
boolean b1,b2,b3,b4,b5;
// Replace
// if ((x1 > x2) || ! ((y1 > y2) || (x1 >= y2)))
// by
b1 = (x1 > x2); b2 = (y1 > y2); b3 = (x1 >= y2);
if (b1 || ! (b2 || b3))
// and in the other one
// replace
// if (((x1 > x2) || (y1 <= y2)) &&
// ((x1 > x2) || (x1 < y2)))
// by
b4 = (y1 <= y2); b5= (x1 < y2);
if ((b1 || b4) && (b1 || b5))
```

Welcome to Propositional Logic

- Also known as:
  - Propositional calculus
  - Boolean algebra
- Propositional logic allows us to prove or disprove equalities that appear in programs:
  - For example, is \( (b1 \lor ! (b2 \lor b3)) \) the same thing as \( (b1 \lor ! b2) \land (b1 \lor ! b3) \)?
  - The answer is yes they are exactly (and always) equivalent by De Morgan’s law, and distributivity.

Propositional Logic Limitations

- Not everything can be reduced to a proposition
- Whether a certain condition is true or not may depend on the values of other (non-Boolean) variables

Statements in Predicate Logic

- \(P(x,y)\)
  - Two parts:
    - A predicate \(P\) describes a relation or property.
    - Variables \((x,y)\) can take arbitrary values from some domain.
    - Example: comparison operator \(> : R \times R \rightarrow B\)
  - Still have two truth values for statements \((T \text{ and } F)\)
  - When we assign values to \(x\) and \(y\), then \(P\) has a truth value.
Let $Q(x,y)$ denote “$x = y + 3$”.
- What are truth values of:
  - $Q(1,2)$ → false
  - $Q(3,0)$ → true

Let $R(x,y)$ denote $x$ beats $y$ in Rock/Paper/Scissors with 2 players with following rules:
- What are the truth values of:
  - $R(\text{rock, paper})$ → false
  - $R(\text{scissors, paper})$ → true

Quantification expresses the extent to which a predicate is true over a set of elements.
- Two forms:
  - Universal $\forall$
  - Existential $\exists$

Universal Quantifier
- $P(x)$ is true for all values in the domain $\forall x \in D, P(x)$
- For every $x$ in $D$, $P(x)$ is true.
- An element $x$ for which $P(x)$ is false is called a counterexample.
- Given $P(x)$ as “$x + 1 > x$” and the domain of $R$, what is the truth value of:
  $\forall x \ P(x)$ → true

Example
- Let $P(x)$ be that $x > 0$ and $x$ is in domain of $R$.
- Give a counterexample for:
  $\forall x \ P(x)$
**Existential Quantifier**

- $P(x)$ is true for at least one value in the domain.
  - $\exists x \in D, P(x)$
- For some $x$ in $D$, $P(x)$ is true.
- Let the domain of $x$ be “animals”, $M(x)$ be “$x$ is a mammal” and $E(x)$ be “$x$ lays eggs”, what is the truth value of: $\exists x (M(x) \land E(x))$

**English to Logic**

- Some person in this class has visited the Grand Canyon.
  - Domain of $x$ is the set of all persons
  - $C(x): x$ is a person in this class
  - $V(x): x$ has visited the Grand Canyon
  - $\exists x(C(x) \land V(x))$
Evaluating Expressions: Precedence and Variable Bindings

- **Precedence:**
  - Quantifiers and negation are evaluated before operators
  - Otherwise left to right

- **Bound:**
  - Variables can be given specific values or
  - Can be constrained by quantifiers

Predicate Logic Equivalences

Statements are **logically equivalent** iff they have the same truth value under all possible bindings.

For example:

\[
\forall x (P(x) \land Q(x)) \equiv \forall x P(x) \land \forall x Q(x)
\]

In English: "Given the domain of students in CS160, all students have passed M118 course (P) and are registered at CSU (Q); hence, all students have passed M118 and all students are registered at CSU.

Other Equivalences

- Someone likes skiing (P) or likes swimming (Q); hence, there exists someone who likes skiing or there exists someone who likes skiing.
  \[
  \exists x (P(x) \lor Q(x)) \equiv \exists x P(x) \lor \exists x Q(x)
  \]

- Not everyone likes to go to the dentist; hence there is someone who does not like to go to the dentist.
  \[
  \neg \forall x P(x) \equiv \exists x \neg P(x)
  \]

- There does not exist someone who likes to go to the dentist; hence everyone does not like to go to the dentist.
  \[
  \neg \exists x P(x) \equiv \forall x \neg P(x)
  \]