Today’s topics

• Boolean algebra preview
• Propositional equivalences
• Predicate logic

Reading: Sections 1.2–1.4

Slides taken from Prof. Michael Frank

Bits and Bit Operations

• A bit is a binary (base 2) digit: 0 or 1.
• Bits may be used to represent truth values.
• By convention:
  0 represents “false”; 1 represents “true”.
• Boolean algebra is like ordinary algebra except that variables stand for bits, + means “or”, and multiplication means “and”.
  – See module 23 (chapter 10) for more details.

Bit Strings

• A Bit string of length n is an ordered sequence (series, tuple) of \( n \geq 0 \) bits.
  – More on sequences in §3.2.
• By convention, bit strings are (sometimes) written left to right:
  – e.g. the “first” bit of the bit string “10011010101001101010” is 1.
  – Watch out! Another common convention is that the rightmost bit is bit #0, the 2nd-rightmost is bit #1, etc.
• When a bit string represents a base-2 number, by convention, the first (leftmost) bit is the most significant bit. Ex. \( 1101_2 = 8+4+1 = 13 \).

Counting in Binary

• Did you know that you can count to 1,023 just using two hands?
  – How? Count in binary!
    • Each finger (up/down) represents 1 bit.
• To increment: Flip the rightmost (low-order) bit:
  – If it changes 1→0, then also flip the next bit to the left,
    • If that bit changes 1→0, then flip the next one, etc.
• \( 0000000000, 0000000001, 0000000010, \ldots, 1111111101, 1111111110, 1111111111 \)
Bitwise Operations

• Boolean operations can be extended to operate on bit strings as well as single bits.
• E.g.:
  01 1011 0110
  11 0001 1101
  Bit-wise OR
  Bit-wise AND
  Bit-wise XOR

End of §1.1

You have learned about:
• Propositions: What they are.
• Propositional logic operators’
  – Symbolic notations.
  – English equivalents.
  – Logical meaning.
  – Truth tables.
• Atomic vs. compound propositions.
• Alternative notations.
• Bits and bit-strings.
• Next section: §1.2
  – Propositional equivalences.
  – How to prove them.

Propositional Equivalence (§1.2)

Two syntactically (i.e., textually) different compound propositions may be the semantically identical (i.e., have the same meaning). We call them equivalent. Learn:
• Various equivalence rules or laws.
• How to prove equivalences using symbolic derivations.

Tautologies and Contradictions

A tautology is a compound proposition that is true no matter what the truth values of its atomic propositions are!
Ex. $p \lor \neg p$ [What is its truth table?]
A contradiction is a compound proposition that is false no matter what! $Ex. p \land \neg p$ [Truth table?]
Other compound props. are contingencies.
Logical Equivalence

Compound proposition \( p \) is *logically equivalent* to compound proposition \( q \), written \( p \iff q \), IFF the compound proposition \( p \iff q \) is a tautology.

Compound propositions \( p \) and \( q \) are logically equivalent to each other IFF \( p \) and \( q \) contain the same truth values as each other in all rows of their truth tables.

Equivalence Laws

- These are similar to the arithmetic identities you may have learned in algebra, but for propositional equivalences instead.
- They provide a pattern or template that can be used to match all or part of a much more complicated proposition and to find an equivalence for it.

Ex. Prove that \( p \lor q \iff \neg(\neg p \land \neg q) \).

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( p \lor q )</th>
<th>( \neg p )</th>
<th>( \neg q )</th>
<th>( \neg p \land \neg q )</th>
<th>( \neg(\neg p \land \neg q) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Equivalence Laws - Examples

- **Identity:** \( p \land T \iff p \quad p \lor F \iff p \)
- **Domination:** \( p \lor T \iff T \quad p \land F \iff F \)
- **Idempotent:** \( p \lor p \iff p \quad p \land p \iff p \)
- **Double negation:** \( \neg\neg p \iff p \)
- **Commutative:** \( p \lor q \iff q \lor p \quad p \land q \iff q \land p \)
- **Associative:** \( (p \lor q) \lor r \iff p \lor (q \lor r) \)
\( (p \land q) \land r \iff p \land (q \land r) \)
More Equivalence Laws

- **Distributive:** \( p \lor (q \land r) \iff (p \lor q) \land (p \lor r) \)
  \( p \land (q \lor r) \iff (p \land q) \lor (p \land r) \)
- **De Morgan’s:**
  \( \lnot(p \land q) \iff \lnot p \lor \lnot q \)
  \( \lnot(p \lor q) \iff \lnot p \land \lnot q \)
- **Trivial tautology/contradiction:**
  \( p \lor \lnot p \iff T \quad p \land \lnot p \iff F \)

An Example Problem

- Check using a symbolic derivation whether
  \( (p \land \lnot q) \implies (p \lor r) \iff \lnot p \lor q \lor \lnot r. \)

\( (p \land \lnot q) \implies (p \lor r) \quad \) [Expand definition of \( \implies \)]
\( \iff \lnot(p \land \lnot q) \lor (p \lor r) \quad \) [Expand definition of \( \lor \)]
\( \iff \lnot(p \land \lnot q) \lor ((p \lor r) \land \lnot(p \land r)) \)
  \[\text{[DeMorgan’s Law]}\]
\( \iff (\lnot p \lor q) \lor ((p \lor r) \land \lnot(p \land r)) \)
  \[\text{cont.}\]

Defining Operators via Equivalences

Using equivalences, we can define operators in terms of other operators.

- **Exclusive or:** \( p \oplus q \iff (p \lor q) \land \lnot(p \land q) \)
  \( p \oplus q \iff (p \land \lnot q) \lor (q \land \lnot p) \)
- **Implies:** \( p \implies q \iff \lnot p \lor q \)
- **Biconditional:** \( p \iff q \iff (p \implies q) \land (q \implies p) \)
  \( p \iff q \iff \lnot(p \oplus q) \)

Example Continued...

\( (\lnot p \lor q) \lor ((p \lor r) \land \lnot(p \land r)) \iff [\lor \text{ commutes}] \)
\( \iff (q \lor \lnot p) \lor ((p \lor r) \land \lnot(p \land r)) \iff [\lor \text{ associative}] \)
\( \iff q \lor ((p \lor r) \land \lnot(p \land r)) \iff [\lor \text{ distrib. over } \land] \)
\( \iff q \lor (((\lnot p \lor (p \lor r)) \land (\lnot p \lor \lnot(p \land r))) \iff [\lor \text{ assoc.}] \)
\( \iff q \lor ((T \lor r) \land (\lnot p \lor \lnot(p \land r))) \iff [\lor \text{ trivial taut.}] \)
\( \iff q \lor ((T \land (\lnot p \lor \lnot(p \land r)))) \iff [\lor \text{ domination}] \)
\( \iff q \lor (\lnot p \lor \lnot(p \land r)) \iff [\lor \text{ identity}] \)
\( \iff q \lor (\lnot p \lor \lnot(p \land r)) \iff \text{cont.} \)
End of Long Example

$q \lor (\neg p \lor \neg (p \land r))$

[DeMorgan’s] $\iff q \lor (\neg p \lor \neg (p \lor \neg r))$

[Assoc.] $\iff q \lor ((\neg p \lor \neg p) \lor \neg r)$

[Idempotent] $\iff q \lor (\neg p \lor \neg r)$

[Assoc.] $\iff (q \lor \neg p) \lor \neg r$

[Commut.] $\iff \neg p \lor q \lor \neg r$

Q.E.D. (quod erat demonstrandum) or $\square$

Review: Propositional Logic
(§§1.1-1.2)

- Atomic propositions: $p$, $q$, $r$, …
- Boolean operators: $\neg \land \lor \oplus \rightarrow \iff$
- Compound propositions: $s := (p \land \neg q) \lor r$
- Equivalences: $p \land \neg q \iff \neg (p \rightarrow q)$
- Proving equivalences using:
  - Truth tables.
  - Symbolic derivations. $p \iff q \iff r$ ...

Predicate Logic (§1.3)

- Predicate logic is an extension of propositional logic that permits concisely reasoning about whole classes of entities.
- Propositional logic (recall) treats simple propositions (sentences) as atomic entities.
- In contrast, predicate logic distinguishes the subject of a sentence from its predicate.
  - Remember these English grammar terms?

Applications of Predicate Logic

It is the formal notation for writing perfectly clear, concise, and unambiguous mathematical definitions, axioms, and theorems (more on these in module 2) for any branch of mathematics.

Predicate logic with function symbols, the “=” operator, and a few proof-building rules is sufficient for defining any conceivable mathematical system, and for proving anything that can be proved within that system!
Other Applications

• Predicate logic is the foundation of the field of mathematical logic, which culminated in Gödel’s incompleteness theorem, which revealed the ultimate limits of mathematical thought:
  – Given any finitely describable, consistent proof procedure, there will always remain some true statements that will never be proven by that procedure.
• i.e., we can’t discover all mathematical truths, unless we sometimes resort to making guesses.

Practical Applications of Predicate Logic

• It is the basis for clearly expressed formal specifications for any complex system.
• It is basis for automatic theorem provers and many other Artificial Intelligence systems.
  – E.g. automatic program verification systems.
• Predicate-logic like statements are supported by some of the more sophisticated database query engines and container class libraries
  – these are types of programming tools.

Subjects and Predicates

• In the sentence “The dog is sleeping”:
  – The phrase “the dog” denotes the subject - the object or entity that the sentence is about.
  – The phrase “is sleeping” denotes the predicate - a property that is true of the subject.
• In predicate logic, a predicate is modeled as a function $P(\cdot)$ from objects to propositions.
  – $P(x) = "x is sleeping"$ (where $x$ is any object).

More About Predicates

• Convention: Lowercase variables $x, y, z...$ denote objects/entities; uppercase variables $P, Q, R...$ denote propositional functions (predicates).
• Keep in mind that the result of applying a predicate $P$ to an object $x$ is the proposition $P(x)$. But the predicate $P$ itself (e.g. $P= "is sleeping"$) is not a proposition (not a complete sentence).
  – E.g. if $P(x) = "x is a prime number"$, $P(3)$ is the proposition “3 is a prime number.”
Propositional Functions

- Predicate logic *generalizes* the grammatical notion of a predicate to also include propositional functions of *any* number of arguments, each of which may take *any* grammatical role that a noun can take.

  - *E.g.* let \( P(x,y,z) = "x \text{ gave } y \text{ the grade } z" \), then if \( x = "\text{Mike}" \), \( y = "\text{Mary}" \), \( z = "\text{A}" \), then \( P(x,y,z) = "\text{Mike gave Mary the grade A}"."

Universes of Discourse (U.D.s)

- The power of distinguishing objects from predicates is that it lets you state things about *many* objects at once.

  - *E.g.*, let \( P(x) = "x+1>x" \). We can then say, "For any number \( x \), \( P(x) \) is true" instead of
    \[
    (0+1>0) \land (1+1>1) \land (2+1>2) \land ... 
    \]

- The collection of values that a variable can take is called *x*’s *universe of discourse*.

Quantifier Expressions

- *Quantifiers* provide a notation that allows us to *quantify* (count) *how many* objects in the univ. of disc. satisfy a given predicate.

  - "\( \forall \)" is the FOR\( \forall \)LL or *universal* quantifier. \( \forall x \ P(x) \) means *for all* \( x \) in the u.d., \( P \) holds.

  - "\( \exists \)" is the \( \exists \)ISTS or *existential* quantifier. \( \exists x \ P(x) \) means *there exists* an \( x \) in the u.d. (that is, 1 or more) *such that* \( P(x) \) is true.

The Universal Quantifier \( \forall \)

- Example:
  Let the u.d. of \( x \) be parking spaces at Duke.
  Let \( P(x) \) be the *predicate* "\( x \) is full."
  Then the *universal quantification* of \( P(x) \), \( \forall x \ P(x) \), is the *proposition*:
    
    - "All parking spaces at Duke are full."
    - *i.e.*, "Every parking space at Duke is full."
    - *i.e.*, "For each parking space at Duke, that space is full."
### The Existential Quantifier \( \exists \)

- Example:
  
  Let the u.d. of \( x \) be parking spaces at Duke. Let \( P(x) \) be the predicate "\( x \) is full."
  
  Then the existential quantification of \( P(x) \), \( \exists x \ P(x) \), is the proposition:
  
  - "Some parking space at Duke is full."
  - "There is a parking space at Duke that is full."
  - "At least one parking space at Duke is full."

### Free and Bound Variables

- An expression like \( P(x) \) is said to have a free variable \( x \) (meaning, \( x \) is undefined).
- A quantifier (either \( \forall \) or \( \exists \)) operates on an expression having one or more free variables, and binds one or more of those variables, to produce an expression having one or more bound variables.

### Example of Binding

- \( P(x,y) \) has 2 free variables, \( x \) and \( y \).
- \( \forall x \ P(x,y) \) has 1 free variable, and one bound variable. [Which is which?]
- "\( P(x) \), where \( x=3 \)" is another way to bind \( x \).
- An expression with zero free variables is a bona-fide (actual) proposition.
- An expression with one or more free variables is still only a predicate: e.g. let \( Q(y) = \forall x \ P(x,y) \)

### Nesting of Quantifiers

Example: Let the u.d. of \( x \) & \( y \) be people.

Let \( L(x,y) = "x \) likes \( y" \) (a predicate w. 2 f.v.’s)

Then \( \exists y \ L(x,y) = "There is someone whom \( x \) likes." (A predicate w. 1 free variable, \( x \))

Then \( \forall x (\exists y \ L(x,y)) = 

"Everyone has someone whom they like.” (A \( \forall \) with 0 free variables.)
Review: Predicate Logic (§1.3)

- Objects $x, y, z, \ldots$
- Predicates $P, Q, R, \ldots$ are functions mapping objects $x$ to propositions $P(x)$.
- Multi-argument predicates $P(x, y)$.
- Quantifiers: $[\forall x P(x)] \equiv “For all x’s, P(x).”$ $[\exists x P(x)] \equiv “There is an x such that P(x).”$
- Universes of discourse, bound & free vars.

Quantifier Exercise

If $R(x,y) = “x$ relies upon $y, ”$ express the following in unambiguous English:

- $\forall x(\exists y R(x,y))=$
- $\exists y(\forall x R(x,y))=$
- $\exists x(\forall y R(x,y))=$
- $\forall y(\exists x R(x,y))=$
- $\forall x(\forall y R(x,y))=$

Natural language is ambiguous!

- “Everybody likes somebody.”
  - For everybody, there is somebody they like,
    - $\forall x \exists y Likes(x,y)$
  - or, there is somebody (a popular person) whom everyone likes?
    - $\exists y \forall x Likes(x,y)$
- “Somebody likes everybody.”
  - Same problem: Depends on context, emphasis.

Game Theoretic Semantics

- Thinking in terms of a competitive game can help you tell whether a proposition with nested quantifiers is true.
- The game has two players, both with the same knowledge:
  - Verifier: Wants to demonstrate that the proposition is true.
  - Falsifier: Wants to demonstrate that the proposition is false.
- The Rules of the Game “Verify or Falsify”:
  - Read the quantifiers from left to right, picking values of variables.
  - When you see “$\forall$”, the falsifier gets to select the value.
  - When you see “$\exists$”, the verifier gets to select the value.
- If the verifier can always win, then the proposition is true.
- If the falsifier can always win, then it is false.
Let’s Play, “Verify or Falsify!”

Let $B(x,y) := “x$’s birthday is followed within 7 days by $y$’s birthday.”

Suppose I claim that among you:

- $orall x \exists y B(x,y)$

Your turn, as falsifier:
- You pick any $x \rightarrow (so-and-so)$
- $\exists y B(so-and-so,y)$

My turn, as verifier:
- I pick any $y \rightarrow (such-and-such)$
- $B(so-and-so,such-and-such)$

Let’s play it in class.

Who wins this game?

What if I switched the quantifiers, and I claimed that

$\exists y \forall x B(x,y)$?

Who wins in that case?

Still More Conventions

- Sometimes the universe of discourse is restricted within the quantification, e.g.,
  - $\forall x > 0 P(x)$ is shorthand for
    “For all $x$ that are greater than zero, $P(x)$.”
    $= \forall x (x > 0 \rightarrow P(x))$
  - $\exists x > 0 P(x)$ is shorthand for
    “There is an $x$ greater than zero such that $P(x)$.”
    $= \exists x (x > 0 \land P(x))$

More to Know About Binding

- $\forall x \exists x P(x)$ - $x$ is not a free variable in $\exists x P(x)$, therefore the $\forall x$ binding isn’t used.
- $(\forall x P(x)) \land Q(x)$ - The variable $x$ is outside of the scope of the $\forall x$ quantifier, and is therefore free. Not a complete proposition!
- $(\forall x P(x)) \land (\exists x Q(x))$ – This is legal, because there are 2 different $x$’s!

Quantifier Equivalence Laws

- Definitions of quantifiers: If u.d.$=a,b,c,\ldots$
  - $\forall x P(x) \iff P(a) \land P(b) \land P(c) \land \ldots$
  - $\exists x P(x) \iff P(a) \lor P(b) \lor P(c) \lor \ldots$
- From those, we can prove the laws:
  - $\forall x P(x) \iff \neg \exists x \neg P(x)$
  - $\exists x P(x) \iff \neg \forall x \neg P(x)$
- Which propositional equivalence laws can be used to prove this?
More Equivalence Laws

- \( \forall x \forall y \ P(x,y) \iff \forall y \forall x \ P(x,y) \)
- \( \exists x \exists y \ P(x,y) \iff \exists y \exists x \ P(x,y) \)
- \( \forall x (P(x) \land Q(x)) \iff (\forall x P(x)) \land (\forall x Q(x)) \)
- \( \exists x (P(x) \lor Q(x)) \iff (\exists x P(x)) \lor (\exists x Q(x)) \)
- Exercise: See if you can prove these yourself.
  - What propositional equivalences did you use?

Review: Predicate Logic (§1.3)

- Objects \( x, y, z, \ldots \)
- Predicates \( P, Q, R, \ldots \) are functions mapping objects \( x \) to propositions \( P(x) \).
- Multi-argument predicates \( P(x, y) \).
- Quantifiers: \( \forall x P(x) \) = “For all \( x \)’s, \( P(x) \)” \( (\exists x P(x)) \) = “There is an \( x \) such that \( P(x) \)”

More Notational Conventions

- Quantifiers bind as loosely as needed: parenthesize \( \forall x (P(x) \land Q(x)) \)
- Consecutive quantifiers of the same type can be combined: \( \forall x \forall y \forall z \ P(x,y,z) \iff \forall x,y,z \ P(x,y,z) \) or even \( \forall xyz \ P(x,y,z) \)
- All quantified expressions can be reduced to the canonical alternating form \( \forall x_1 \exists x_2 \forall x_3 \exists x_4 \ldots P(x_1, x_2, x_3, x_4, \ldots) \)

Defining New Quantifiers

As per their name, quantifiers can be used to express that a predicate is true of any given quantity (number) of objects.
Define \( \exists! x P(x) \) to mean “\( P(x) \) is true of \( exactly one \) \( x \) in the universe of discourse.”
\( \exists! x P(x) \iff \exists x (P(x) \land \neg \exists y (P(y) \land y \neq x)) \)
“\( \text{There is an } x \text{ such that } P(x), \text{ where there is no } y \text{ such that } P(y) \text{ and } y \neq x. \)”
1.45

Some Number Theory Examples

- Let u.d. = the natural numbers 0, 1, 2, ...
- “A number $x$ is even, $E(x)$, if and only if it is equal to 2 times some other number.”
  \[ \forall x \ (E(x) \iff (\exists y \ x=2y)) \]
- “A number is prime, $P(x)$, iff it’s greater than 1 and it isn’t the product of any two non-unity numbers.”
  \[ \forall x \ (P(x) \iff (x>1 \land \neg \exists y z \ x=yz \land y \neq 1 \land z \neq 1)) \]

1.46

Goldbach’s Conjecture (unproven)

Using $E(x)$ and $P(x)$ from previous slide,
\[ \forall E(x>2) : \exists P(p),P(q) : p+q = x \]
or, with more explicit notation:
\[ \forall x [x>2 \land E(x)] \rightarrow \exists p \exists q P(p) \land P(q) \land p+q = x. \]

“Every even number greater than 2 is the sum of two primes.”

1.47

Calculus Example

- One way of precisely defining the calculus concept of a limit, using quantifiers:
  \[ \lim_{x \to a} f(x) = L \iff \forall \varepsilon > 0 : \exists \delta > 0 : \forall x : \left( |x-a| < \delta \rightarrow (|f(x) - L| < \varepsilon) \right) \]

1.48

Deduction Example

- Definitions:
  \[ s \equiv \text{Socrates (ancient Greek philosopher)}; \]
  \[ H(x) \equiv \text{“x is human”}; \]
  \[ M(x) \equiv \text{“x is mortal”}. \]
- Premises:
  \[ H(s) \quad \text{Socrates is human.} \]
  \[ \forall x \ H(x) \rightarrow M(x) \quad \text{All humans are mortal}. \]
Deduction Example Continued

Some valid conclusions you can draw:
\[ H(s) \rightarrow M(s) \quad \text{[Instantiate universal.]} \]
\[ \text{If Socrates is human then he is mortal.} \]
\[ \neg H(s) \lor M(s) \quad \text{Socrates is inhuman or mortal.} \]
\[ H(s) \land (\neg H(s) \lor M(s)) \quad \text{Socrates is human, and also either inhuman or mortal.} \]
\[ (H(s) \land \neg H(s)) \lor (H(s) \land M(s)) \quad \text{[Apply distributive law.]} \]
\[ \text{[Trivial contradiction.]} \]
\[ (H(s) \land M(s)) \quad \text{Socrates is mortal.} \]

Another Example

• Definitions: \( H(x) \triangleq \text{“x is human”} \);
\( M(x) \triangleq \text{“x is mortal”} \);
\( G(x) \triangleq \text{“x is a god”} \)

• Premises:
  – \( \forall x \ H(x) \rightarrow M(x) \) (\text{“Humans are mortal”}) and
  – \( \forall x \ G(x) \rightarrow \neg M(x) \) (\text{“Gods are immortal”}).

• Show that \( \neg \exists x \ (H(x) \land G(x)) \)
  (\text{“No human is a god.”})

The Derivation

• \( \forall x \ H(x) \rightarrow M(x) \) and \( \forall x \ G(x) \rightarrow \neg M(x). \)
• \( \forall x \ \neg M(x) \rightarrow \neg H(x) \quad \text{[Contraposition.]} \)
• \( \forall x \ [G(x) \rightarrow \neg M(x)] \land [\neg M(x) \rightarrow \neg H(x)] \)
• \( \forall x \ G(x) \rightarrow \neg H(x) \quad \text{[Transitivity of \( \rightarrow \).]} \)
• \( \forall x \ \neg G(x) \lor \neg H(x) \quad \text{[Definition of \( \rightarrow \).]} \)
• \( \forall x \ \neg (G(x) \land H(x)) \quad \text{[DeMorgan’s law.]} \)
• \( \neg \exists x \ G(x) \land H(x) \quad \text{[An equivalence law.]} \)

End of §1.3-1.4, Predicate Logic

• From these sections you should have learned:
  – Predicate logic notation & conventions
  – Conversions: predicate logic \( \leftrightarrow \) clear English
  – Meaning of quantifiers, equivalences
  – Simple reasoning with quantifiers
• Upcoming topics:
  – Introduction to proof-writing.
  – Then: Set theory –
    • a language for talking about collections of objects.