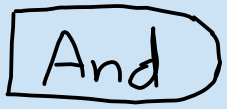
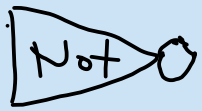
# Predicates and Quantifiers

CSE 311 Autumn 25
Lecture 5

# Meet Boolean Algebra

| Name | Variables | "True/False" | "And" | "Or" | "Not" | Implication |
|---|---|---|---|---|---|---|
| Java Code | `boolean b` | `true, false` | `&&` | `\|\|` | ! | No special symbol |
| Propositional Logic | $"p, q, r"$ | T, F | $\wedge$ | $\vee$ | $\neg$ | $\rightarrow$ |
| Circuits | Wires | 1, 0 |   |   |   | No special symbol |
| Boolean Algebra | $a, b, c$ | 1,0 | $\cdot$ ("multiplication") | $+$ ("addition") | $'$ (apostrophe after variable) | No special symbol |

Propositional logic

$(p \wedge q \wedge r) \vee s \vee \neg t$

Boolean Algebra

$pqr + s + t'$

# Canonical Forms

A truth table is a unique representation of a Boolean Function.

If you describe a function, there's only one possible truth table for it.

Given a truth table you can find many circuits and many compound propositions to represent it.

Think back to when we were developing the law of implication…

It would be nice to have a "standard" proposition (or standard circuit) we could always write as a starting point.

So we have a (possibly) shorter way of telling if we have the same function.

# Disjunctive Normal Form (DNF)

a.k.a. OR of ANDs

a.k.a Sum-of-Products Form

a.k.a. Minterm Expansion

1. Read the true rows of the truth table
2. AND together all the settings in a given (true) row.
3. OR together the true rows.

# Conjunctive Normal Form

a.k.a. AND of ORs

a.k.a. Product-of-Sums Form

a.k.a. Maxterm Expansion

1. Read the false rows of the truth table
2. OR together the negations of all the settings in the false rows.
3. AND together the false rows.

Or take the DNF of the negation of the function you care about, and distribute the negation.

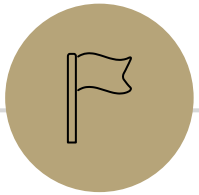
# Normal Forms

Don’t simplify any further! Don’t factor anything out (even if you can). The point of the canonical form is we know exactly what it looks like, you might simplify differently than someone else.

Why? Easier to understand for people.

Inside the parentheses are only ORs between the parentheses are only ANDs (or vice versa).

You’ll use these more in later courses.



# Predicates!

# Predicate Logic

So far our propositions have worked great for fixed objects.

What if we want to say "If $x > 10$ then $x^2 > 100$."

$x > 10$ isn't a proposition. Its truth value depends on $x$.

We need a function that can take in a value for $x$ and output True or False as appropriate.

# Predicates

**Predicate**

**A function that outputs true or false.**

`Cat(x)`:= "`x` is a cat"

`Prime(x)` := "`x` is prime"

`LessThan(x,y)`:= "`x<y`"

`Sum(x,y,z)`:= "`x+y=z`"

`HasNChars(s,n)`:= "string `s` has length `n`"

Numbers and types of inputs can change. Only requirement is output is Boolean.

# Analogy

Propositions were like Boolean variables.

What are predicates? Functions that return Booleans

```
public boolean predicate(…)
```

# Translation (Predicates)

Translation works a lot like when we just had propositions.

Let's try it…

$x$ is prime or $x^2$ is odd or $x = 2$.

$\texttt{Prime}(x) \vee \texttt{Odd}(x^2) \vee \texttt{Equals}(x, 2)$

# Domain of Discourse Problem

$x$ is prime or $x^2$ is odd or $x = 2$.

$\texttt{Prime}(x) \vee \texttt{Odd}(x^2) \vee \texttt{Equals}(x, 2)$

Can $x$ be 4.5? What about "abc" ?

I never intended you to plug 4.5 or "abc" into $x$.

When you read the sentence you probably didn't imagine plugging those values in….

# Domain of Discourse Definition

$x$ is prime or $x^2$ is odd or $x = 2$.

$\texttt{Prime}(x) \vee \texttt{Odd}(x^2) \vee \texttt{Equals}(x, 2)$

To make sure we **can't** plug in 4.5 for $x$, predicate logic requires deciding on the types we'll allow

**Domain of Discourse**

**The set of all inputs allowed as inputs to our predicates.**

Often we give the type(s) of allowed inputs, like "all integers" or "all real numbers.

# Try it…

What's a possible domain of discourse for these lists of predicates?

1. "$x$ is a cat", "$x$ barks", "$x$ likes to take walks"

2. "$x$ is prime", "$x$=5" "$x < 20$" "$x$ is a power of two"

3. "$x$ is enrolled in course $y$", "$y$ is a pre-req for $z$"

# Try it… (answers)

What's a possible domain of discourse for these lists of predicates?

1. "$x$ is a cat", "$x$ barks", "$x$ likes to take walks"

   "Mammals", "pets", "dogs and cats", …

2. "$x$ is prime", "$x$=5" "$x < 20$" "$x$ is a power of two"

   "positive integers", "integers", "numbers", …

3. "$x$ is enrolled in course $y$", "$y$ is a pre-req for $z$"

   "objects in the university course enrollment system", "university entities", "students and courses", …

More than one domain of discourse might be reasonable…if it might affect the meaning of the statement, we specify it.

# Quantifiers (Reasons)

Now that we have variables, let's really use them…

We tend to use variables for two reasons:

1. The statement is true for every $x$, we just want to put a name on it.
2. There's some $x$ out there that works, (but I might not know which it is, so I'm using a variable).

# Quantifiers (Symbols)

We have two extra symbols to indicate which way we're using the variable.

1. The statement is true for every $x$, we just want to put a name on it.

$\forall x\ (\mathrm{p}(\mathrm{x}) \wedge q(x))$ means "for every $x$ in our domain, $p(x)$ and $q(x)$ both evaluate to true."

2. There's some $x$ out there that works, (but I might not know which it is, so I'm using a variable).

$\exists x(p(x) \wedge q(x))$ means "there is an $x$ in our domain, such that $p(x)$ and $q(x)$ are both true.

# Quantifiers (Universal)

We have two extra symbols to indicate which way we're using the variable.

1. The statement is true for every $x$, we just want to put a name on it.

$\forall x\ (\mathrm{p}(\mathrm{x}) \wedge q(x))$ means "for every $x$ in our domain, $p(x)$ and $q(x)$ both evaluate to true."

## Universal Quantifier

"$\forall x$"

**"for each $x$", "for every $x$", "for all $x$" are common translations**

**Remember: upside-down-A for All.**

# Quantifiers (Existential)

## Existential Quantifier

**"$\exists x$"**

**"there is an $x$", "there exists an $x$", "for some $x$" are common translations**

**Remember: backwards-E for Exists.**

2. There's some $x$ out there that works, (but I might not know which it is, so I'm using a variable).

$\exists x(p(x) \wedge q(x))$ means "there is an $x$ in our domain, for which $p(x)$ and $q(x)$ are both true.

# Translations (Quantifiers)

Domain of Discourse: all integers

"For every $x$, if $x$ is even, then $x = 2$."

"There are $\mathrm{x}, y$ such that $\mathrm{x} < y$."

$\exists x\ (\texttt{Odd}(x) \wedge \texttt{LessThan}(x, 5))$

$\forall y\ (\text{Even}(y) \wedge \text{Odd}(y))$

# Translations (Quantifers; with answers)

"For every $x$, if $x$ is even, then $x = 2$."

$\forall x(\texttt{Even}(x) \rightarrow \texttt{Equal}(x, 2))$

"There are $\mathrm{x}, y$ such that $\mathrm{x} < y$."

$\exists x \exists y(\texttt{LessThan}(x, y))$

$\exists x\ (\texttt{Odd}(x) \wedge \texttt{LessThan}(x, 5))$

There is an odd number that is less than 5.

$\forall y\ (\mathrm{Even}(y) \wedge \mathrm{Odd}(y))$

All numbers are both even and odd.

# Translations Resources

More practice in section and on homework.

Also a reading on the webpage –

An explanation of why “for any” is not a great way to translate ∀ (even though it looks like a good option on the surface)

More information on what happens with multiple quantifiers (we’ll discuss more on Monday).

# Evaluating Predicate Logic Intro

"For every $x$, if $x$ is even, then $x = 2$." / $\forall x(\texttt{Even}(x) \rightarrow \texttt{Equal}(x, 2))$

Is this true?

# Evaluating Predicate Logic

"For every $x$, if $x$ is even, then $x = 2$." / $\forall x(\texttt{Even}(x) \rightarrow \texttt{Equal}(x, 2))$

Is this true?

TRICK QUESTION! It depends on the domain.

| Prime Numbers | Positive Integers | Odd integers |
|---|---|---|
| True | False | True (vacuously) |

# One Technical Matter

How do we parse sentences with quantifiers?

What's the "order of operations?"

We will usually put parentheses right after the quantifier and variable to make it clear what's included. If we don't, it's the rest of the expression.

Be careful with repeated variables…they don't always mean what you think they mean.

$\forall x\big(P(x)\big) \wedge \forall x(Q(x))$ are different $x$'s.

# Bound Variables

What happens if we repeat a variable?

Whenever you introduce a new quantifier with an already existing variable, it "takes over" that name until its expression ends.

$$\forall x(P(x) \wedge \forall x[Q(x)] \wedge R(x))$$

It's common (albeit somewhat confusing) practice to reuse a variables when it "wouldn't matter".

Never do something like the above: where a single name switches from gold to purple back to gold. Switching from gold to purple only is usually fine…but names are cheap.

# More Practice

Let your domain of discourse be fruits. Translate these

There is a fruit that is tasty and ripe.

For every fruit, if it is not ripe then it is not tasty.

There is a fruit that is sliced and diced.

# More Practice with answers

Let your domain of discourse be fruits. Translate these

There is a fruit that is tasty and ripe.

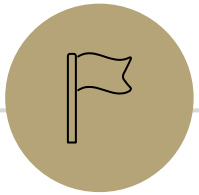
$$\exists x(\texttt{Tasty}(x) \land \texttt{Ripe}(x))$$

For every fruit, if it is not ripe then it is not tasty.

$$\forall x(\neg\texttt{Ripe}(x) \rightarrow \neg\texttt{Tasty}(x))$$

There is a fruit that is sliced and diced.

$$\exists x(\texttt{Sliced}(x) \land \texttt{Diced}(x))$$



# Negation

# Negating Quantifiers

What happens when we negate an expression with quantifiers?

What does your intuition say?

Original

Every positive integer is prime

$\forall x\ \texttt{Prime}(x)$

Domain of discourse: positive integers

Negation

There is a positive integer that is not prime.

$\exists x(\neg\ \texttt{Prime}(x))$

Domain of discourse: positive integers

# Negating Quantifiers Rules

Let's try on an existential quantifier…

**Original**

There is a positive integer which is prime and even.

$\exists x(\texttt{Prime}(x) \wedge \texttt{Even}(x))$

Domain of discourse: positive integers

**Negation**

Every positive integer is composite or odd.

$\forall x(\neg\texttt{Prime}(x) \vee \neg\texttt{Even}(x))$

Domain of discourse: positive integers

To negate an expression with a quantifier
1. Switch the quantifier (∀ becomes ∃, ∃ becomes ∀)
2. Negate the expression inside

# Negation

Let your Domain of Discourse be integers; translate into predicate notation and negate.

There are integers $x, y$ such that $xy = 0$.

Every integer is even.

# Negation (with answers)

Let your Domain of Discourse be integers; translate into predicate notation and negate.

There are integers $x, y$ such that $xy = 0$.

Original $\exists x \exists y(\text{Equal}(xy, 0))$

Negation $\forall x \forall y(\neg \text{Equal}(xy, 0))$

Every integer is even.

Original $\forall x(\text{Even}(x))$

Negation $\exists x(\neg \text{Even}(x))$