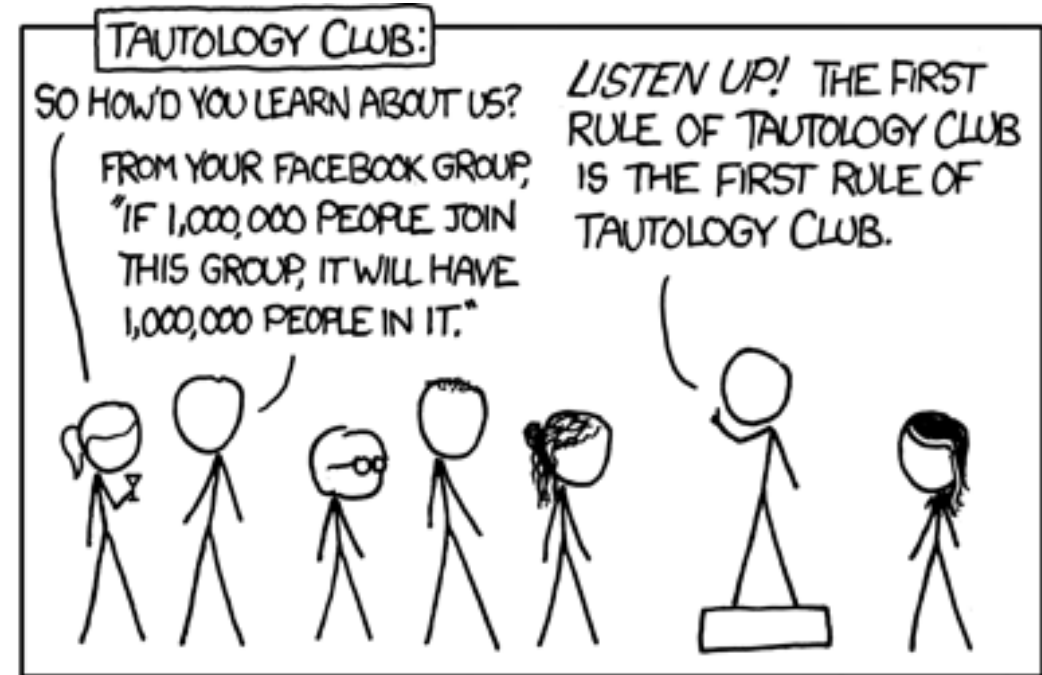


Warm Up

p	q	r	$F(p, q, r)$
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	T

1. Write a propositional logic expression for F in DNF (ORs of ANDs) form
2. Write a propositional logic expression for F in CNF (ANDs of ORs) form



Predicate Logic

CSE 311: Foundations of
Computing I
Lecture 4

Announcements

- HW1
 - Due tonight, 11:59 pm
- HW2
 - Released tonight

Warm Up

p	q	r	$F(p, q, r)$
T	T	T	T
T	T	F	T
T	F	T	F
T	F	F	T
F	T	T	T
F	T	F	F
F	F	T	F
F	F	F	T

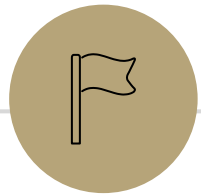
1. Write a propositional logic expression for F in DNF (ORs of ANDs) form

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

2. Write a propositional logic expression for F in CNF (ANDs of ORs) form

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

$$* (\neg p \vee q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee \neg r)$$



Boolean Algebra & Circuits

Notation

- Logic is fundamental
 - Computer scientists use it in programs
 - Mathematicians use it in proofs
 - Engineers use it in hardware
 - Philosophers use it in arguments
- Consequently, everyone has their own notation

Boolean Algebra

Another notation for logic. Preferred by some because it's compact.

Term	Propositional Logic	Boolean Algebra
or	\vee	$+$
and	\wedge	\cdot
not	\neg	$'$
True	T	1
False	F	0

$\neg p$ p'

$(p \wedge q \wedge r) \vee s \vee \neg t$ in Boolean Algebra is $pqr + s + t'$.

Digital Circuits

Computing with Logic

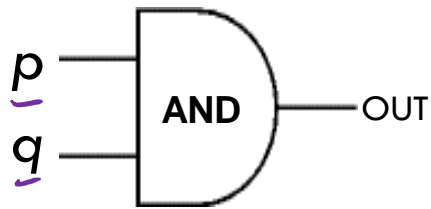
- T corresponds to 1, or high voltage
- F corresponds to 0, or low voltage

Gates

- Gates take inputs and produce outputs

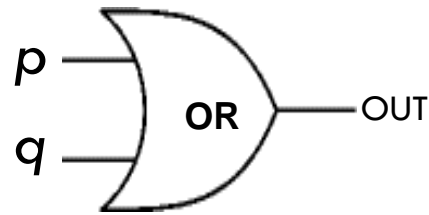
Digital Circuits – Gates

AND Gate



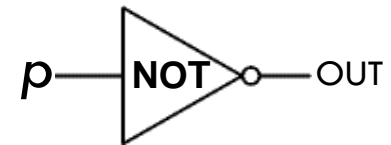
$$p \wedge q$$

OR Gate



$$p \vee q$$

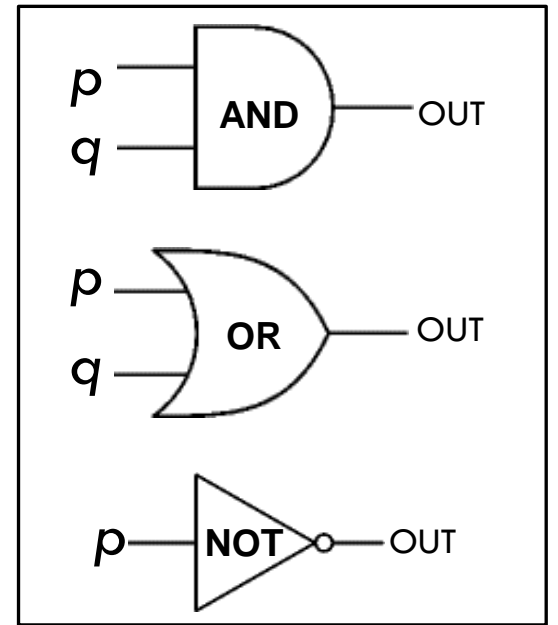
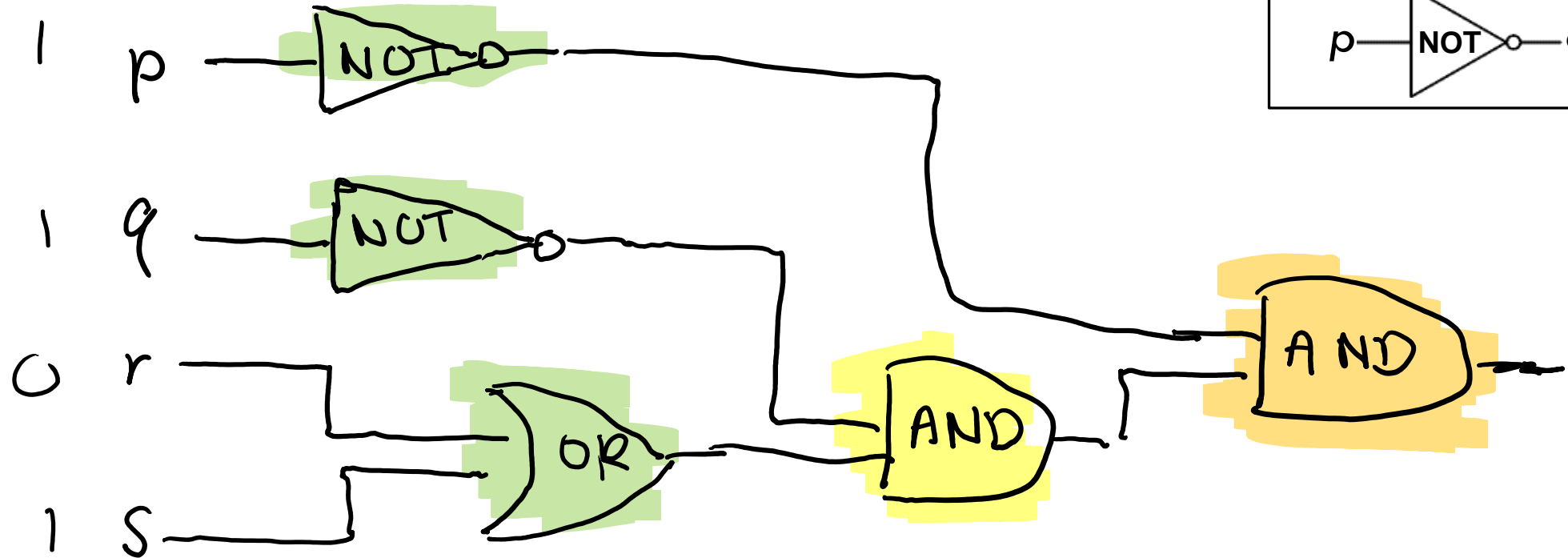
NOT Gate

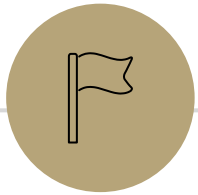


$$\neg p$$

Digital Circuits – Example

Write $\neg p \wedge (\neg q \wedge (r \vee s))$ as a circuit.





Predicate Logic

Motivation

Often we will work with statements of the form:

If $x > 10$, then $x^2 > 100$.

If x is even, x^2 is even.

~~$p: x > 10$
 $q: x^2 > 100$~~
 $p \rightarrow q$

Can you translate these to propositional logic?

NO. we need a function that is T/F depending on x .

Motivation

Propositional Logic

Lets us break down complex true or false statements into atomic parts joined by connectives.

Predicate Logic

Lets us analyze complex true or false statements that are functions of some underlying objects.

Predicate Logic

3 Parts

1. Predicate (& logical connectives)
2. Domain of Discourse
3. Quantifiers

Predicate

Definition:

A **predicate** is a function that outputs true or false.

$:=$ "is defined as"

$\text{Cat}(x) := x$ is a cat

$\text{Even}(x) := x$ is even

$\text{LessThan}(x, y) := x < y$

$\text{Sum}(x, y, z) := x + y = z$

$\text{IsRaining} :=$ It is raining outside

$\text{Cat}(\text{Garfield}) \equiv T$

$\text{Cat}(\text{Anjali}) \equiv F$

$\text{Even}(4) \equiv T$

$\text{Even}(17) \equiv F$

$\text{LessThan}(3, 5) \equiv T$

↑
Proposition

Analogy

Propositions were like Boolean variables.

```
boolean itIsRaining = true
```

Predicates are like functions that return Boolean values.

```
public boolean Even(int x) {...}
```

Even(x)

$x \% 2 == 0$

Predicate Translation: Example

~~Prime (Anjali)~~

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

$\text{Prime}(\underline{x}) := x$ is prime

$\text{odd}(\underline{x}) := x$ is odd

$\text{Equals}(\underline{a}, \underline{b}) := a = b$

$\text{Prime}(\underline{x}) \vee \text{odd}(\underline{x^2}) \vee \text{Equals}(\underline{x}, 2)$

↑
T

↑
T

↑
F

Domain of Discourse

Definition:

The **domain of discourse** for a predicate is the set of possible inputs.

$\text{Cat}(x)$ – Possible domains include mammals, animals, cats.

$\text{LessThan}(x, y)$ – Possible domains include real numbers, integers.

1.2, 3.5 ... -2, -1, 0, 1, 2, ...

Domain of Discourse: Example

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

1. $\text{Prime}(x) := x$ is prime

Integers > 1

2. $\text{Equals}(x, y) := x$ and y are the same object

Real Numbers, Objects, people

* 3. $\text{EnrolledIn}(x, y) := x$ is enrolled in course y

students & courses

Quantifiers: Motivation

We tend to use variables for two reasons:

1. The statement is true for every x .

For every integer x , if x is even then x^2 is even.

2. There is some x for which the statement is true.
There is some problem x that computers cannot solve.

Quantifiers

$\forall x P(x)$

- \forall is called the Universal Quantifier.
- Read out loud as "for all x , P of x ".
- $\forall x P(x)$ means for every x in the domain, $P(x)$ is true.

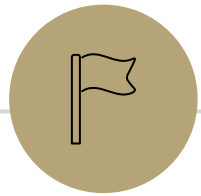
$\exists x P(x)$

- \exists is called the Existential Quantifier.
- Read out loud as "there exists x such that P of x ".
- $\exists x P(x)$ means there is some x in the domain for which $P(x)$ is true.

Predicate Logic Summary

3 Parts

1. Predicate – Function that outputs true or false.
2. Domain of Discourse – Set of possible inputs to a predicate.
3. Quantifiers – A statement about when a predicate is true: \forall or \exists



Predicate Logic Translation

English to Logic

Domain of Discourse
Integers

Predicate Definitions
Even(x) := x is even
Equals(x, y) := $x = y$
LessThan(x, y) := $x < y$

Translate to predicate logic. Then evaluate if the statement is T or F.

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

$$\forall x (\text{Even}(x) \rightarrow \text{Equals}(x, 2)) \equiv \text{F}$$

e.g. $x=4$

There are integers x and y such that $x < y$.

$$\exists x \exists y (\text{LessThan}(x, y)) \equiv \text{T}$$

Logic to English

Domain of Discourse
Integers

Predicate Definitions

Even(x) := x is even

LessThan(x, y) := $x < y$

Odd(x) := x is odd

Translate to English. Then evaluate if the statement is T or F.

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

There is an odd integer less than 5.

T

e.g. $x=3$

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

All integers are even and odd.

F

Examples

Domain of Discourse

Integers

Predicate Definitions

Even(x) := x is even Greater(x, y) := $x > y$

Odd(x) := x is odd Prime(x) := x is prime

Translate to English. Then evaluate if the statement is T or F.

$\exists x \text{ Even}(x)$ There is an even integer. (T)

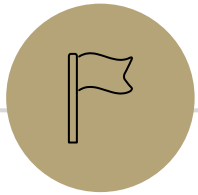
$\forall x \text{ Odd}(x)$ All integers are odd. (F)

$\forall x (\text{Even}(x) \vee \text{Odd}(x))$ All integers are even or odd. (T)

$\exists x (\text{Even}(x) \wedge \text{Odd}(x))$ There is an int. that's even & odd. (F)

$\forall x \text{ Greater}(x + 1, x)$ for every int x , $x + 1 > x$ (T)

$\exists x (\text{Even}(x) \wedge \text{Prime}(x))$ There is an int. that's even & prime. (T)



Domain Restriction

Domain Restriction

Definition:

Domain restriction is the technique of limiting our
domain of discourse to a smaller set of objects.

Domain Restriction

Domain of Discourse
Animals

Predicate Definitions

Cat(x) := x is a cat

Blue(x) := x is blue

All cats are blue.

$$\cancel{\forall x (Cat(x) \wedge Blue(x))} \quad \forall x (Cat(x) \rightarrow Blue(x))$$

There is a blue cat.

$$\exists x (Cat(x) \wedge Blue(x))$$

Domain Restriction

Domain of Discourse
Animals

Predicate Definitions
Cat(x) := x is a cat
Blue(x) := x is blue

$$\forall x (\text{Cat}(x) \rightarrow \text{Blue}(x))$$

vs.

$$\forall x (\text{Cat}(x) \wedge \text{Blue}(x))$$

All cats are blue.

All animals are blue cats.

$$\exists x (\text{Cat}(x) \rightarrow \text{Blue}(x))$$

vs.

$$\exists x (\text{Cat}(x) \wedge \text{Blue}(x))$$

$$\exists x (\neg (\text{Cat}(x) \vee \text{Blue}(x)))$$

There is an animal that is not a cat or is blue.

There is a blue cat

Tip: Avoid \rightarrow under \exists in most situations

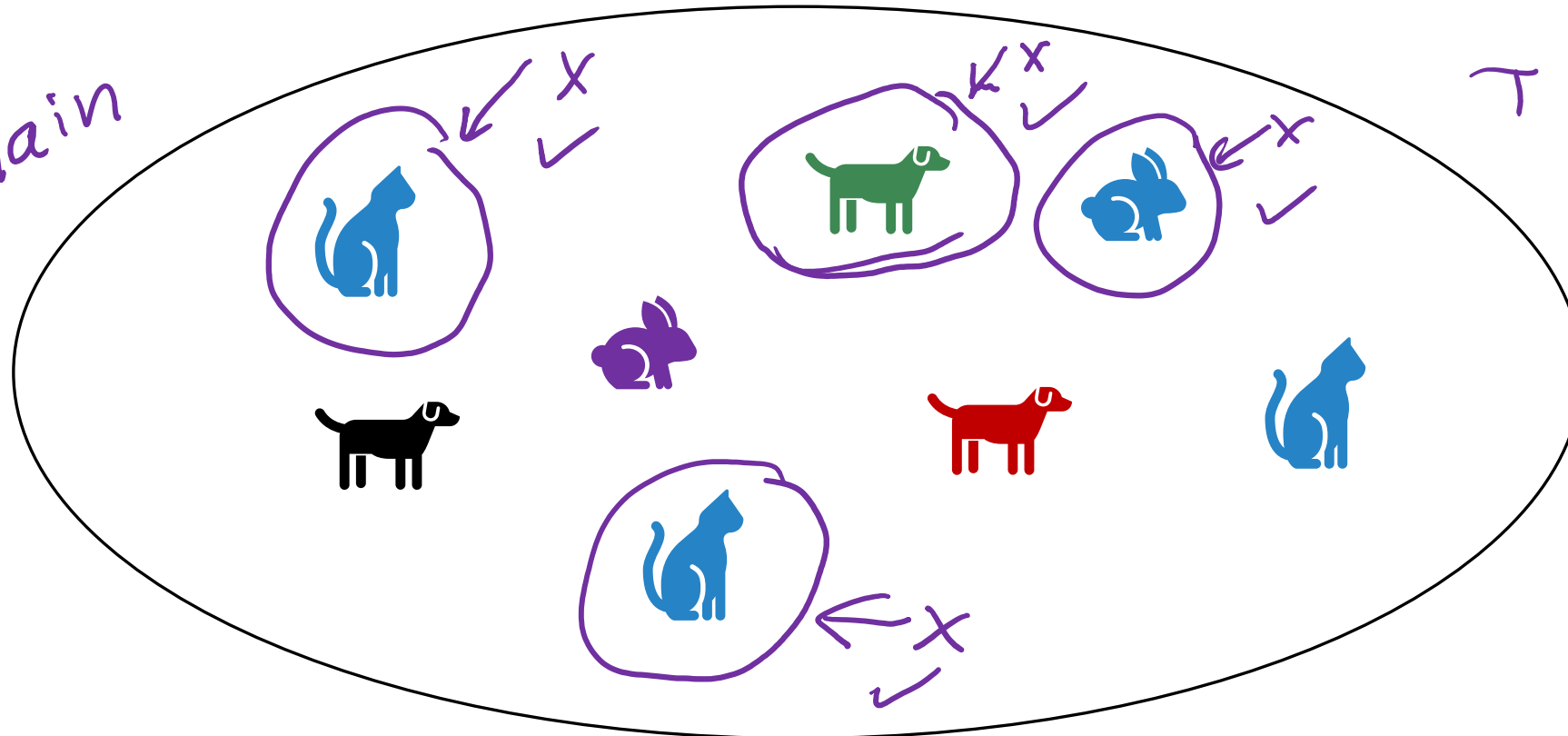
\forall and vacuous truth

$\forall x(\text{Cat}(x) \rightarrow \text{Blue}(x))$ means "all cats are blue".

One way to check if $\forall x P(x)$ is true is to "loop" over every element of the domain and check that $P(x)$ is true.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

domain



T	\rightarrow	T
f	\rightarrow	f
f	\rightarrow	T

Domain Restriction

Translations often sound more **natural** if we:

1. Notice domain restriction patterns. $\forall x (\text{Mouse}(x) \rightarrow \dots)$
2. Avoid using variables when we can.
3. Drop the "for all" or "there exists" when we can.

For example:

$$\forall x (\text{Cat}(x) \rightarrow \text{Blue}(x))$$

✗ For all animals x , if x is a cat then x is blue.

✓ All cats are blue.

Domain Restriction

Domain of Discourse
Food

Predicate Definitions
Fruit(x) := x is a fruit
Tasty(x) := x is tasty
Ripe(x) := x is ripe

Translate these sentences using a **natural-sounding** translation.

$\exists x(\text{Fruit}(x) \wedge \text{Tasty}(x))$

There is a tasty fruit

Some fruits are tasty

$\forall x((\text{Fruit}(x) \wedge \neg \text{Ripe}(x)) \rightarrow \neg \text{Tasty}(x))$

All fruits that aren't ripe
aren't tasty.

Quantifier Scope

$$\star \exists x (P(x) \wedge Q(x)) \quad \text{vs.} \quad \exists x P(x) \wedge \exists x Q(x)$$

The first expression is marked with a circled 'f' (false) and the second with a circled 'T' (true). Both expressions are underlined in purple.

Domain: Integers

$$P(x) := (x=1)$$

x is odd

$$Q(x) := (x=2)$$

x is even

Section Tomorrow

Practice with equivalences & translating

Quantifiers can be nested!

There is a person that all people love.

$\exists x \forall y (\text{Love}(y, x))$