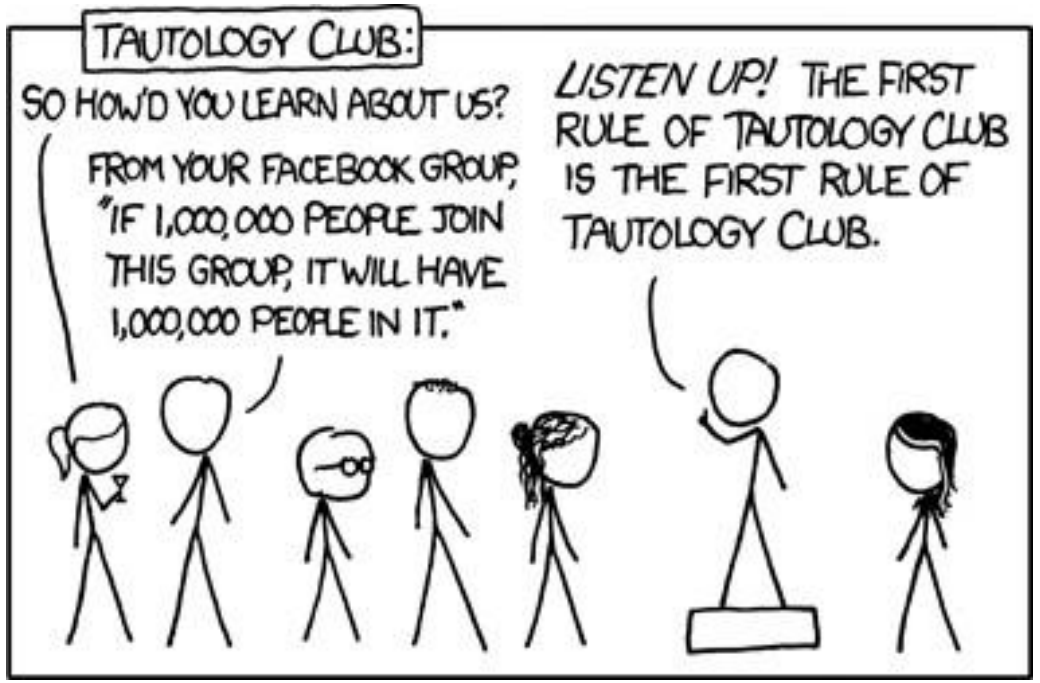
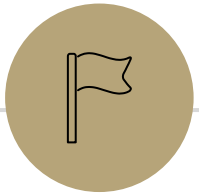


Predicates and Quantifiers

CSE 311 Summer 25
Lecture 4

Announcements

- HW1
  - Due Wednesday, 11:59 pm

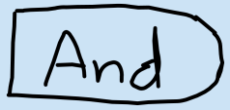
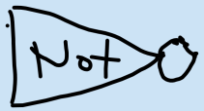


Review

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

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 prepositions 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 (CNF)

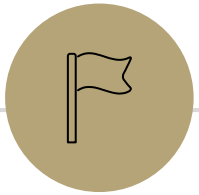
a.k.a. AND of ORs

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.



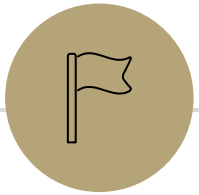
Warm Up

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

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.

Can you translate these to propositional logic?

No. We need a function that is true or false depending on the value of 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
2. Domain of Discourse
3. Quantifiers

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.

```
boolean itIsRaining = true
```

Predicates are like functions that return Boolean values.

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

Predicate Translation

Translation works a lot like when we just had propositions.

Let’s try it…

Predicate Translation: Example

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

$$\text{Prime}(a) \coloneqq a \text{ is Prime}$$
$$\text{Odd}(a) \coloneqq a \text{ is Odd}$$
$$\text{Equals}(a, b) \coloneqq \ a = b$$

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

Domain of Discourse

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

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…

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: Motivation

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

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

Quantifiers

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

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 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.

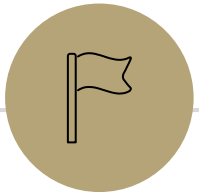
Predicate Logic Summary

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: ∀ or ∃



Predicate Logic Translation

Translations

Domain of Discourse
Integers

Predicate Definitions
$\text{Even}(x) \coloneqq x$ is even
$\text{LessThan}(x, y) \coloneqq x < y$
( )

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

"There are 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

Domain of Discourse
Integers

Predicate Definitions
$\text{Even}(x) \coloneqq x$ is even
$\text{LessThan}(x, y) \coloneqq x < y$
( )

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

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

"There are x, y such that $x < y$."

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

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

There is an odd number that is less than 5.

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

All numbers are both even and odd.

Translations

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

"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

Domain of Discourse
Integers

Predicate Definitions
$\mathrm{Even}(x) :=$ x is even
$\mathrm{LessThan}(x, y) := x < y$
$\mathrm{Odd}(x) :=$ x is odd

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

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

There is an odd integer less than 5. True, e.g., x = 3.

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

All integers are even and odd. False.

Examples

Domain of Discourse
Integers

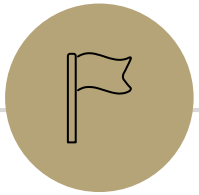
Predicate Definitions

$\text{Even}(x) \coloneqq x$ is even $\text{Greater}(x, y) \coloneqq x > y$

$\text{Odd}(x) \coloneqq x$ is odd $\text{Prime}(x) \coloneqq x$ 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's an integer that's even and odd.	F
$\forall x\ \text{Greater}(x+1, x)$	For all integers x, $x + 1 > x$	T
$\exists x(\text{Even}(x) \wedge \text{Prime}(x))$	There is an integer that's even and prime.	T



Domain Restriction

Domain Restriction

Definition:

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

Quantifiers

∀ (for **A**ll) and ∃ (there **E**xists)

Write these statements in predicate logic with quantifiers. Let your domain of discourse be "cats"

This sentence implicitly makes a statement about all cats!

If a cat is fat, then it is happy.

$$\forall x[\texttt{Fat}(x) \rightarrow \texttt{Happy}(x)]$$

Quantifiers

Writing implications can be tricky when we change the domain of discourse.

For every cat: if the cat is fat, then it is happy.

Domain of Discourse: cats $\forall x[\texttt{Fat}(x) \rightarrow \texttt{Happy}(x)]$

What if we change our domain of discourse to be all mammals?
We need to limit x to be a cat. How do we do that?

$\forall x[(\texttt{Cat}(x) \wedge \texttt{Fat}(x)) \rightarrow \texttt{Happy}(x)]$ $\forall x[\texttt{Cat}(x) \wedge (\texttt{Fat}(x) \rightarrow \texttt{Happy}(x))]$

Quantifiers

Which of these translates "For every cat: if a cat is fat then it is happy."
when our domain of discourse is "mammals"?

$\forall x[(\texttt{Cat}(x) \wedge \texttt{Fat}(x)) \rightarrow \texttt{Happy}(x)]$

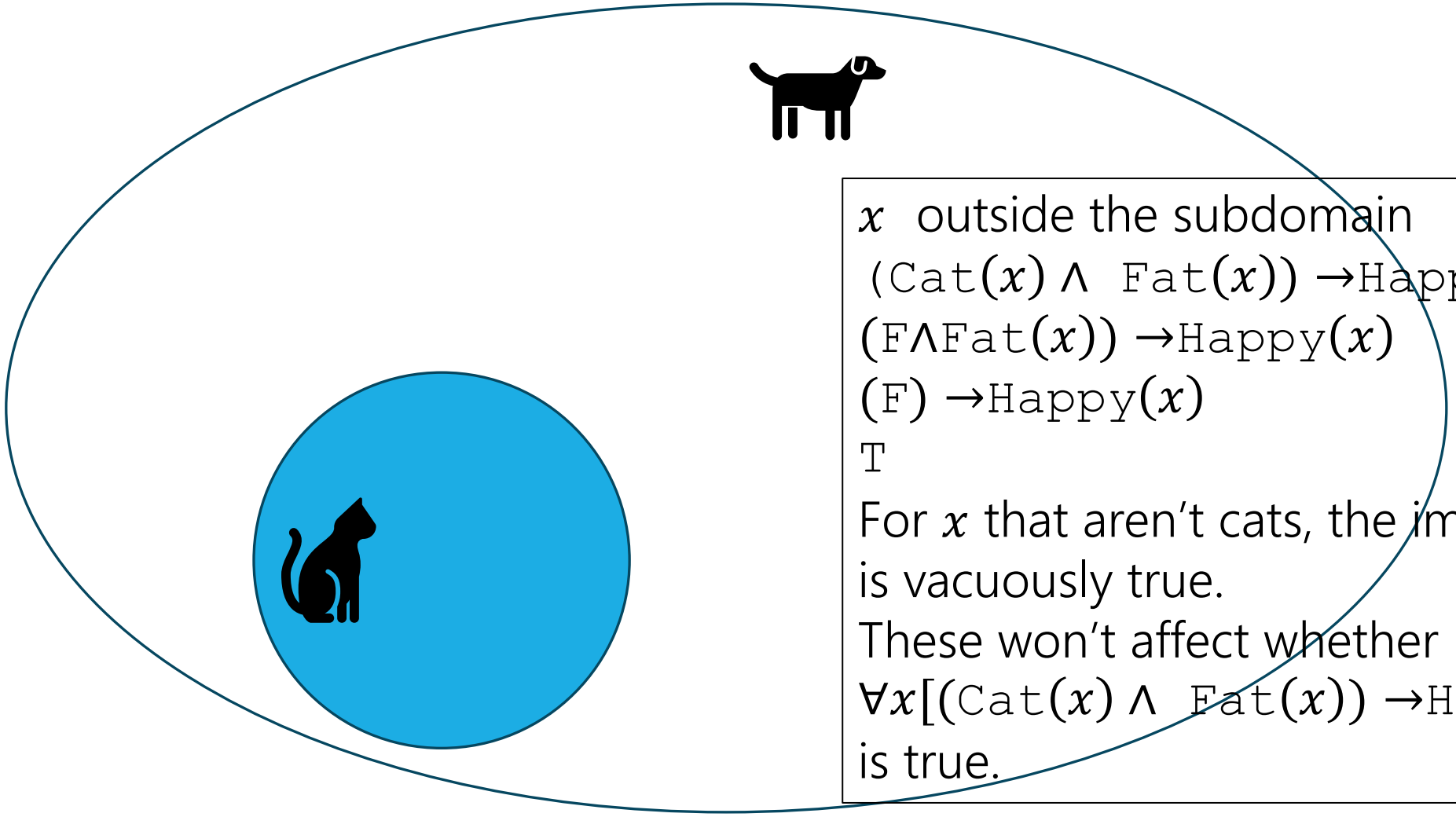
For all mammals, if x is a cat and fat then it is happy
[if x is not a cat, the claim is vacuously true, you can't use the promise for anything]

$\forall x[\texttt{Cat}(x) \wedge (\texttt{Fat}(x) \rightarrow \texttt{Happy}(x))]$

For all mammals, that mammal is a cat and if it is fat then it is happy.
[what if x is a dog? Dogs are in the domain, but…uh-oh. This isn't what we meant.]

To "limit" variables to a portion of your domain of discourse under a universal quantifier add a hypothesis to an implication.

$\forall x[(\mathrm{Cat}(x) \wedge \mathrm{Fat}(x)) \rightarrow \mathrm{Happy}(x)]$



x outside the subdomain

$(\mathrm{Cat}(x) \wedge \mathrm{Fat}(x)) \rightarrow \mathrm{Happy}(x)$

$(\mathrm{F} \wedge \mathrm{Fat}(x)) \rightarrow \mathrm{Happy}(x)$

$(\mathrm{F}) \rightarrow \mathrm{Happy}(x)$

T

For x that aren't cats, the implication is vacuously true.

These won't affect whether $\forall x[(\mathrm{Cat}(x) \wedge \mathrm{Fat}(x)) \rightarrow \mathrm{Happy}(x)]$ is true.

$\forall x[(\mathrm{Cat}(x) \wedge \mathrm{Fat}(x)) \rightarrow \mathrm{Happy}(x)]$

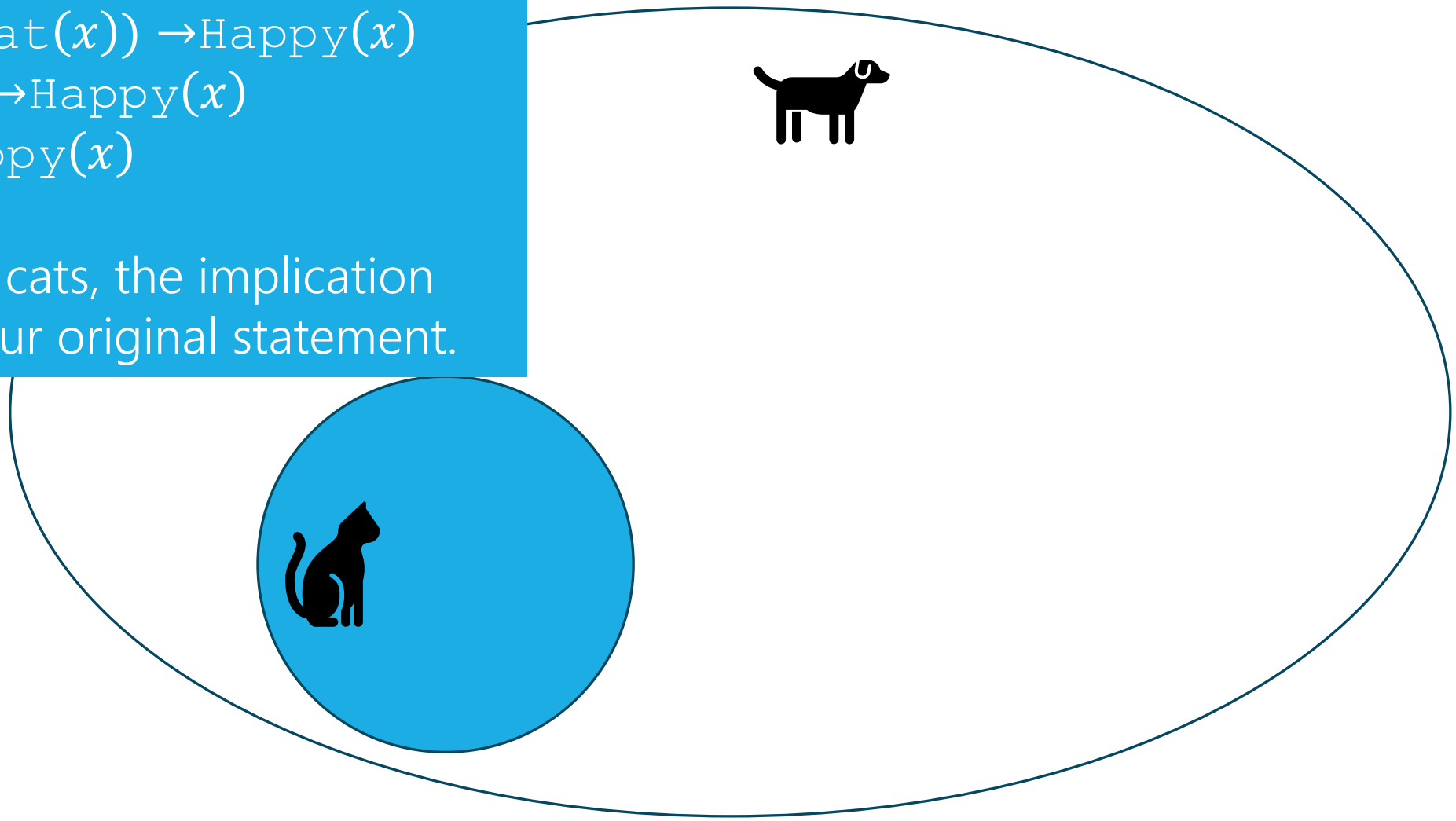
x inside the subdomain

$(\mathrm{Cat}(x) \wedge \mathrm{Fat}(x)) \rightarrow \mathrm{Happy}(x)$

$(\mathrm{T} \wedge \mathrm{Fat}(x)) \rightarrow \mathrm{Happy}(x)$

$\mathrm{Fat}(x) \rightarrow \mathrm{Happy}(x)$

For x that are cats, the implication simplifies to our original statement.



Quantifiers

Existential quantifiers need a different rule:

To “limit” variables to a portion of your domain of discourse under an existential quantifier AND the limitation together with the rest of the statement.

There is a dog who is not happy.

Domain of discourse: dogs

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

Quantifiers

Which of these translates "There is a dog who is not happy."
when our domain of discourse is "mammals"?

$$\exists x[\texttt{Dog}(x) \rightarrow \neg\texttt{Happy}(x)]$$

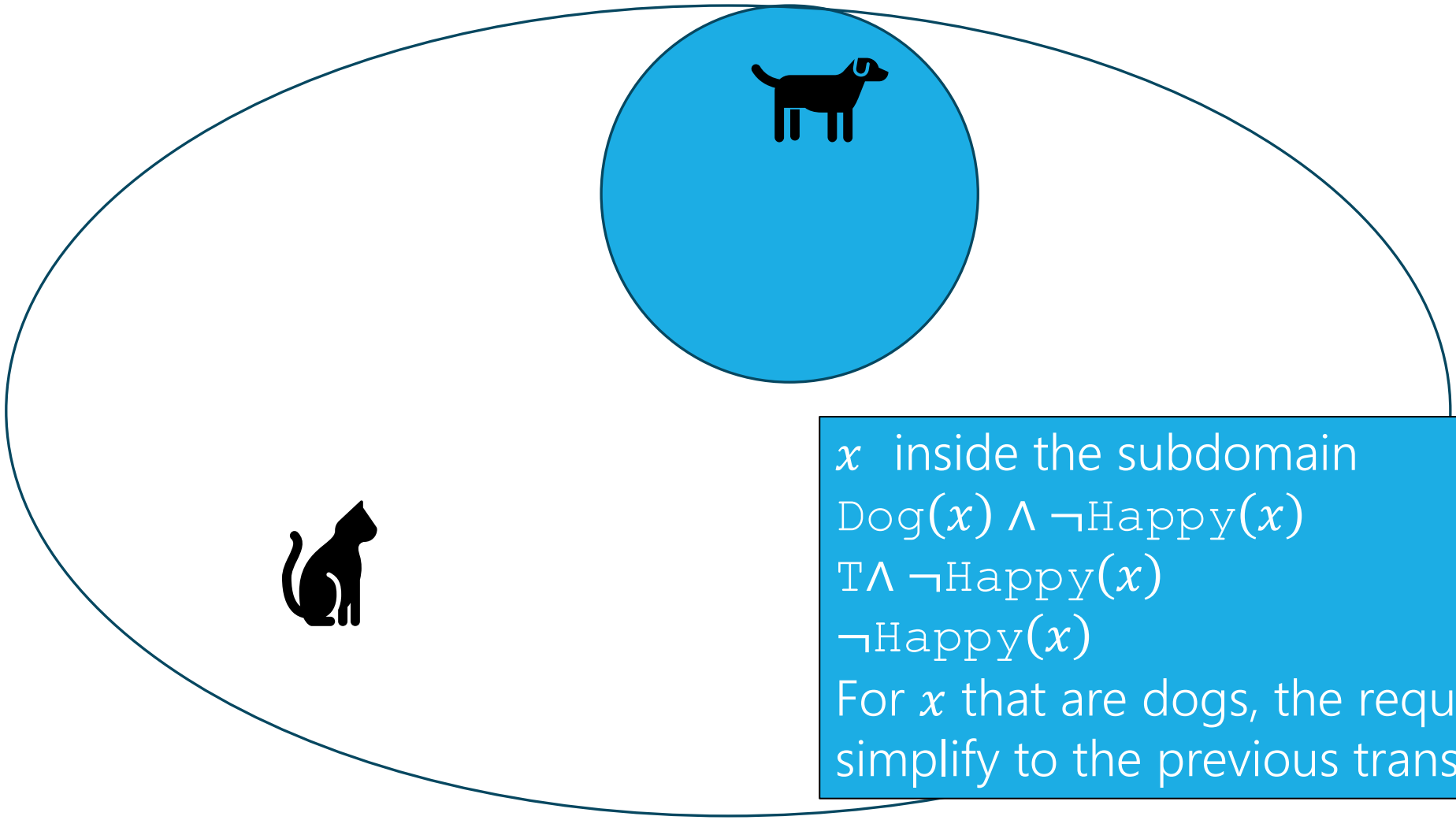
There is a mammal, such that if x is a dog then it is not happy.
[this can't be right – plug in a cat for x and the implication is true]

$$\exists x[\texttt{Dog}(x) \wedge \neg\texttt{Happy}(x)]$$

There is a mammal that is both a dog and not happy.
[this one is correct!]

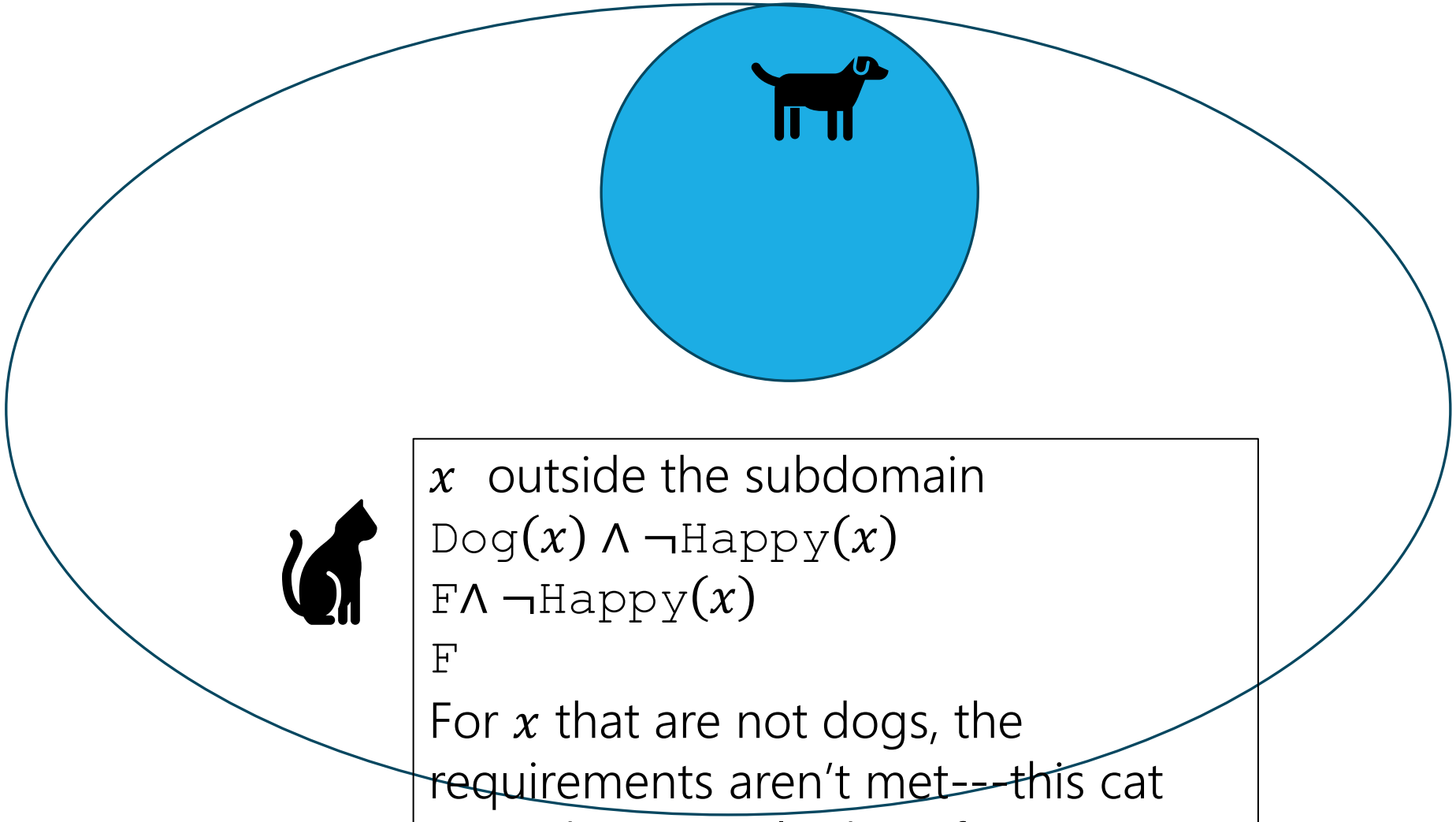
To "limit" variables to a portion of your domain of discourse under an existential quantifier AND the limitation together with the rest of the statement.

$\exists x[(\mathtt{Dog}(x) \wedge \neg \mathtt{Happy}(x)]$



x inside the subdomain
$\mathtt{Dog}(x) \wedge \neg \mathtt{Happy}(x)$
$\mathtt{T} \wedge \neg \mathtt{Happy}(x)$
$\neg \mathtt{Happy}(x)$
For x that are dogs, the requirements simplify to the previous translation

$\exists x[(\mathtt{Dog}(x) \wedge \neg \mathtt{Happy}(x)]$



x outside the subdomain

$\mathtt{Dog}(x) \wedge \neg \mathtt{Happy}(x)$

$\mathtt{F} \wedge \neg \mathtt{Happy}(x)$

$\mathtt{F}$

For x that are not dogs, the requirements aren't met---this cat isn't what we're looking for.

Why are the rules what they are?

A universal quantifier is a "Big AND"

For a domain of discourse of $\{e_1, e_2, \ldots, e_k\}$

$\forall x(P(x))$ means $P(e_1) \wedge P(e_2) \wedge \cdots \wedge P(e_k)$

Now let's say our domain is $\{e_1, e_2, \ldots, e_k, f_1, f_2, \ldots, f_j\}$ where f_i are the irrelevant parts of the bigger domain (non-cat-mammals). We want the expression to be

$P(e_1) \wedge P(e_2) \wedge \cdots \wedge P(e_k) \wedge T \wedge T \ldots \wedge T$

$\forall x(RightSubDomain(x) \rightarrow P(x))$ does that!

Why are the rules what they are?

An existential quantifier is a "Big OR"

For a domain of discourse of $\{e_1, e_2, \ldots, e_k\}$

$\exists x(P(x))$ means $P(e_1) \vee P(e_2) \vee \cdots \vee P(e_k)$

Now let's say our domain is $\{e_1, e_2, \ldots, e_k, f_1, f_2, \ldots, f_j\}$ where f_i are the irrelevant parts of the bigger domain (non-cat-mammals). We want the expression to be

$P(e_1) \vee P(e_2) \vee \cdots \vee P(e_k) \vee F \vee F \ldots \vee F$

$\exists x(RightSubDomain(x) \wedge P(x))$ does that!

Domain Restriction Translations

Translations often sound more natural if we:

1. Notice domain restriction patterns.
2. Avoid using variables when we can.
3. Drop the "for all" or "there exists" when we can.

For example:

$$\forall x\big(\mathrm{Cat}(x) \to \mathrm{Blue}(x)\big)$$

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

✓ All cats are blue.

Translation Practice

Domain of Discourse
Food

Predicate Definitions
$\text{Fruit}(x) \coloneqq x$ is a fruit
$\text{Tasty}(x) \coloneqq x$ is tasty
$\text{Ripe}(x) \coloneqq x$ is ripe

Translate these sentences using a natural-sounding translation.

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

There is a tasty fruit. OR 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.

Todo

Tonight:

Get started on HW1 if you haven’t already!

CC3 due Wednesday at noon