CMPSCI 250: Introduction to Computation

Lecture 9: Predicates and Quantifiers
David Mix Barrington
26 February 2013

Predicates and Quantifiers

- Predicates and Data Types
- Existential and Universal Quantifiers
- Free and Bound Variables
- Logical Equivalences for Quantifiers -- The DeMorgan Rules
- Nested Quantifiers
- Translating Quantified Statements
- Negating Quantified Statements

Predicates and Data Types

- A **proposition** is a statement that is true or false. A **predicate** is a statement that *would be* true or false if the value of a particular variable is known.
- "$3 \le 7$" is a proposition that is true. "$7 \le 3$" is a proposition that is false. But "$x \le y$" is not a proposition. If we set $x = 3$ and $y = 7$ it is true, and if we set $x = 7$ and $y = 3$ it is false.
- Propositions can deal with entities other than numbers. In programming, you know that every variable has a **type** -- the range of possible values it might take. To know what a proposition means, or when a predicate is true or false, we need to know the type of any variables in it. What values for them are possible? In logic, the range of values for a variable is also called the **universe of discourse**.
- We can name predicates. If I define T(x) to be "x is a terrier", where x is of type "dog", then T(Cardie) and T(Duncan) are each propositions.

IClicker Question #1

- Let A(n) be the predicate “n < 7”, where the type of n is “integer”.
- **How many** of the propositions A(3), A(7), and A(4) are true?
- (a) none of them
- (b) one of them
- (c) two of them
- (d) all three of them

Predicates in Program Correctness

- When we reason about code, we are concerned with the truth of statements whose meaning may change during the run of the code.

- When we write a Java method we often state its **preconditions**, statements we expect to be true before the method runs, and its **postconditions**, statements that should be true after it runs if the preconditions were true. We state these as predicates, because their truth usually cannot be determined without the value of some variables.

- A **loop invariant** is a statement that will remain true after a pass through a loop, provided that it is true before the pass. We state loop invariants in terms of the method's variables, for example "x represents the number of inputs seen so far".

- Predicates give us a language to make such statements, on which we can then apply the rules of logic.

Existential and Universal Quantifiers

- Given a predicate, say our T(x), and a data type, say “dogs who live with me”, the predicate may be true for *all* the elements of the type, for *some* of them, or for *none* of them. **Quantifiers** are a formal way to say such things about a predicate.

- The **universal quantifier** $\forall$ is used to make a statement that a predicate is always true. The statement “$\forall x:T(x)$” says that all of my dogs are terriers, a proposition that is false.

- The **existential quantifier** $\exists$ is used to make a statement that a predicate is sometimes true. The statement “$\exists x:T(x)$” says that at least one of my dogs is a terrier, which is true.

IClicker Question #2

- If the type of x is “my dogs” and T(x) means “x is a terrier”, and W(x) means “x likes to go for walks”, **what is the meaning** of the quantified statement “$(\forall x: W(x)) \wedge (\exists x: T(x)) \wedge (\exists x: \neg T(x))$”?

- (a) All my dogs like walks, at least one of my dogs is a terrier, and at least one of my dogs is not a terrier.

- (b) All of my dogs who like walks are terriers that exist, or non-terriers that exist.

- (c) All of my dogs are either terriers or non-terriers, and at least one likes walks.

Free and Bound Variables

- Each use of a quantifier **binds** one variable. Consider the two-place predicate P(x, y), meaning "dog x plays with dog y". Now suppose we quantify this statement to make "∃x:P(x, y)", meaning "there exists a dog x who plays with dog y". To know whether this is true, we need to know the value of y -- for example, "∃x: P(x, Duncan)" might be true while "∃x: P(x, Cardie)" is false.

- In this example y is a free variable, since we need to know its value to determine the meaning of the statement. But x is a bound variable -- it does not make sense to speak of "∃x: P(x, y)" being true for some values of x but not others. The statement "∃z: P(z, y)" would have the same meaning as "∃x:P(x, y)", but "∃x:P(x, z)" would have a meaning depending on z.

IClicker Question #3

- Let $Q(x, y)$ be the predicate “$x + y = 3$”, where x and y are integers.
- Consider the quantified statement “$\forall x$: $Q(x, y)$”. **Which is true?**
- (a) x and y are both free variables
- (b) x is a free variable and y is a bound variable
- (c) x is a bound variable and y is a free variable
- (d) x and y are both bound variables

Logical Equivalences: ANDs and ORs

- Going back to dogs, let $F(x)$ mean "x is furry", $R(x)$ mean "x is a retriever", $T(x)$ mean "x is a terrier", and $W(x)$ mean "x likes to go for walks".

- The statements "$\forall x: F(x) \wedge W(x)$" and "$(\forall x: F(x)) \wedge (\forall x: W(x))$" are logically equivalent. The first means "all my dogs are both furry and like walks", while the second says "all my dogs are furry and all my dogs like walks".

- But we have to be careful using "rules" like this. With my actual dogs, the statement "$\forall x: (R(x) \vee T(x))$" is true, but "$(\forall x:R(x)) \vee (\forall x:T(x))$" is false!

IClicker Question #4

- **Which pair of statements** are logically equivalent?
- (a) “$\exists x$: $(T(x) \wedge R(x))$” and “$(\exists x{:}T(x)) \wedge (\exists x{:}R(x))$”
- (b) “$\exists x$: $(T(x) \vee R(x))$” and “$(\exists x{:}T(x)) \vee (\exists x{:}\ R(x))$”
- (c) “$\exists x$: $T(x)$” and “$\exists x$: $\neg R(x)$”

Translating Quantified Statements

- There are many ways to phrase a universal statement like "$\forall x$: F(x)", where the type of the variable x is "my dogs" and F(x) means "x is furry":
- "All my dogs are furry."
- "For every one of my dogs, it is furry."
- "If x is any of my dogs, then x is furry."
- "There does not exist one of my dogs that is not furry" ($\neg\exists x:\neg F(x)$)
- The last statement uses one of the **DeMorgan laws** and helps us see that the statement would be *true* if I had no dogs at all.

More Translations of Quantified Statements

- Now let's look at "$\exists x$:T(x)", where x's type is "my dogs" and T(x) means "x is a terrier". We can translate this as:
- "There exists one of my dogs that is a terrier."
- "At least one of my dogs is a terrier."
- "There is one of my dogs x with the property that x is a terrier."
- "It is not true that all of my dogs are non-terriers."
- The last statement uses the other DeMorgan law for quantifiers, and in symbols would be "$\neg\forall x:\neg T(x)$". If I had no dogs, $\exists x:T(x)$ would be false.

Negating Quantified Statements

- Recall the DeMorgan Laws of propositional logic: $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$ (and-to-or), and $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ (or-to-and). We have similar rules for quantified statements, because an $\exists$ is in effect a big OR, and a $\forall$ just a big AND.

- The negation of "$\forall x$: F(x)" ("all my dogs are furry") is "$\exists x$: $\neg$F(x)" ("at least one of my dogs is not furry"). To show that a universal statement is *false*, we need a **counterexample**. If the data type is empty, no counterexample can exist and any universal statement is *true*.

- The negation of "$\exists x$: F(x)" ("at least one of my dogs is furry") is "$\forall x$:$\neg$F(x)" ("all my dogs are not furry"). To show that an existential statement is true, we need a **witness**. If the data type is empty, no witness can exist and any existential statement is *false*.

IClicker Question #5

- What is the **negation** of “Either all my dogs are terriers or one of them is not furry”, or “$(\forall x{:}T(x)) \vee (\exists x{:}\neg F(x))$”?

- (a) “All my dogs are terriers and none of them are furry.” $(\forall x{:}T(x)) \wedge (\forall x{:}\neg F(x))$

- (b) “One of my dogs is not a terrier and all of them are furry.” $(\exists x{:}\neg T(x)) \wedge (\forall x{:}F(x))$

- (c) “None of my dogs are terriers or one of them is furry.” $(\forall x{:}\neg T(x)) \vee (\exists x{:}F(x))$

Nested Quantifiers

- If we begin with a predicate that has more than one free variable, we can **nest** more than one quantifier. If P(x, y) means "dog x plays with dog y", we can form such statements as "$\exists x:\forall y:P(x, y)$" ("there is a dog that plays with all dogs") and "$\forall x:\exists y:P(x,y)$" ("each dog has some dog that it plays with").

- Note that these two statements have *no free variables* -- they are propositions that are either true or false with no variable value needing to be supplied.

- Also note that these statements are *different from each other*: the first requires that there be one dog that plays with all the others, while the second would be true as long as each dog has a playmate, not necessarily all the same one.

- But $\exists x:\exists y:P(x,y)$ and $\exists y:\exists x:P(x,y)$ are the same, even if $P(x,y)$ and $P(y,x)$ are not -- the same is true for $\forall x:\forall y:P(x,y)$ and $\forall y:\forall x:P(x,y)$. Can you see why?

Translating Nested Quantifiers

- We can translate the colon after "$\exists x$" as "such that", except in the case where there are adjacent existential quantifiers. So "$\exists x:\forall y:T(x) \wedge P(x, y)$" is "there exists a dog x such that for all dogs y, x is a terrier and x plays with y". But "$\exists x:\exists y:T(x) \wedge P(x,y)$" is "there exist a dog x and a dog y such that x is a terrier and x plays with y".

- The colon after "$\forall x$" is best left as a comma rather than "such that". We translate "$\forall x:\exists y:F(x) \wedge P(x,y)$" as "for every dog x, there exists a dog y such that x is furry and x plays with y". Adjacent universal quantifiers get translated with "and", as in "$\forall x:\forall y:F(x) \wedge P(x, y)$" which means "for all dogs x and all dogs y, x is furry and x plays with y".

- There are more natural English phrasings of most quantified statements -- you often want to first translate the symbols exactly and then rephrase to get something that means the same but sounds better.

Quantified Statements in Number Theory

- The main purpose of quantified statements in mathematics is to *express statements*, and particularly *properties of objects*, precisely. In number theory we begin with variables that range over the **natural numbers N**, and the symbols for arithmetic.

- The predicate D(x, y) or "x divides y" can be defined as "$\exists z: xz = y$".

- The predicate P(x) or "x is prime" can be defined as "$(x > 1) \wedge \neg(\exists y: \exists z: (x = yz) \wedge (y > 1) \wedge (z > 1))$." Using DeMorgan twice and some propositional logic rules, we can write this as "$(x > 1) \wedge \forall y: \forall z: ((x = yz) \rightarrow ((y \leq 1) \vee (z \leq 1))$".

- "$x \equiv y \pmod{m}$" can be written "$\exists k: x + km = y$" where k is an *integer*, or as "D(m, x - y)" where x-y might be negative.