COMPSCI 250: Introduction to Computation

Lecture #8: Proofs With Quantifiers

David Mix Barrington and Ghazaleh Parvini

22 September 2023

Proofs With Quantifiers

- The Meaning of Quantifier Proofs
- The Four Proof Rules
- Instantiation: Eliminating $\exists$
- Existence: Introducing $\exists$
- Specification: Eliminating $\forall$
- Generalization: Introducing $\forall$
- The Dog Example

Powers of Languages

- In algebra we say “x^k” to denote the product of k copies of x. Similarly in language theory, if X is a language, we abbreviate the concatenation product XX as “X^2”, XXX as “X^3”, and so forth.
- It turns out that if we treat concatenation as “multiplication” and union as “addition”, the distributive law holds, and using algebraic rules we can get facts like $(X + Y)^2 = X^2 + XY + YX + Y^2$. (We don’t say “2XY” because XY and YX are not necessarily equal.)

The Kleene Star Operation

- X^0 is a special case -- "not multiplying" gives us the multiplicative identity, which turns out to be the language $\{\lambda\}$.
(Check that $\{\lambda\}X = X$ for any language X.)

- It's convenient sometimes to talk about the language $X^0 + X^1 + X^2 + X^3 + \ldots$, which is the set of all strings that can be made by concatenating together *any number* of strings from X. We call this language X^*, the **Kleene star** of X.
We've used this notation already when we defined Σ^* to be the set of all strings from Σ.

The Meaning of Quantifier Proofs

- A quantified statement talks about a particular situation -- a set of objects divided into data types, and some predicates with arguments from particular types.
- For every legal **atomic statement**, which is a predicate with arguments of the proper type filled in, we need to have the truth value defined.
- We may also have **constants** -- values from specific types that are given names.

The Dog Example

- Our running example today will have a set of dogs D, three unary predicates on dogs: W(x) “x likes walks”, R(x) “x is a Rottweiler”, and T(x) “x is a terrier”, and a binary predicate S(x, y) “dog x is smaller than dog y”.
- We will also have two constants of type D, Cardie (c) and Duncan (d). There are an infinity of possible models of these predicates -- we want to show that any model that satisfies our premises also satisfies our conclusion.

Models of the Predicates?

- What do we mean by a “model” of a set of predicates?
- In order to know whether a statement using these predicates is true, we need to know the set of dogs, which dogs make each unary predicate true, and which pairs of dogs make the binary predicate R true.
- Once all these things are decided, any statement with no free variables is either true or false.

The Four Proof Rules

- In the Forward-Backward method, we have one statement that we want to go forward from, and another we want to go backward from.
- The structure of these statements lets us know what kind of quantifier proof rule we can use. In particular, the *outermost* quantifier controls what we can do.

Going Forward

- To go forward from a universal statement, we want to **eliminate** the universal quantifier, that is, go from $\forall x: P(x)$ to $P(a)$. To go forward from an existential statement, we want to eliminate the $\exists$, going from $\exists x: P(x)$ to $P(a)$.
- The rules of Specification and Instantiation will allow us to do this, but we must be careful of our context and the meaning of our variables.

Going Backward

- To go backward from a universal statement, we want to prove $\forall x: P(x)$ from statements without the $\forall$, thus **introducing** a universal quantifier. The rule of Generalization lets us do this.
- The rule of Existence lets us introduce an existential and prove $\exists x: P(x)$, starting from statements without the $\exists$.
- We thus can find a last step if there is a quantifier in our desired conclusion.

Instantiation: Eliminating ∃

- Given the premise "there exists a dog that is a terrier" ($\exists x$: $T(x)$), the rule of **Instantiation** lets us derive the statement $T(a)$, eliminating the quantifier.
- In English, we would say "let a be the dog that exists by the premise, so that we know $T(a)$". Here "a" must be a new variable, referring to a new dog.

More Instantiation

- We don’t know whether that new dog is equal to any old dogs, or whether any of the other predicates are true for it. We know only its type and the fact “T(a)” that we got from the statement we instantiated.
- In essence we are giving a name to one of the dogs, who may or may not be one of the dogs we already know something about. A common error is to say something like “a terrier exists, therefore that terrier is Duncan”, claiming a name or a property of the instantiated object with no justification.

Clicker Question #1

- Given the premise "$\exists x$: $S(x, d) \wedge W(x)$", or "There exists a dog that is smaller than Duncan and likes walks", which statement *follows* from the premise?

- (a) It is not the case that every dog is either smaller than Duncan or likes walks.
- (b) Every dog that is smaller than Duncan likes walks.
- (c) Duncan is smaller than himself and likes walks.
- (d) Some dog x is both smaller than Duncan and likes walks.

Not the Answer

Clicker Answer #1

- Given the premise "$\exists x: S(x, d) \wedge W(x)$", or "There exists a dog that is smaller than Duncan and likes walks", which statement *follows* from the premise?

- (a) It is not the case that every dog is either smaller than Duncan or likes walks.

  This is $\sim\forall x{:}S(x, d) \vee W(x)$.

  Correct would be $\sim\forall x{:}\sim S(x, d) \vee \sim W(x)$.

- (b) Every dog that is smaller than Duncan likes walks.

  This is $\forall x: S(x, d) \rightarrow W(x)$.

- (c) Duncan is smaller than himself and likes walks.

  This is $S(d, d) \wedge W(d)$, Specifying instead of Instantiating.

- (d) Some dog x is both smaller than Duncan and likes walks.

Existence: Introducing ∃

- How do we prove a statement like $\exists x: P(x)$, introducing an existential quantifier? The rule of **Existence** says that from *any* statement of the form $P(a)$, we can derive $\exists x: P(x)$. Here x is a new bound variable that isn't being used already.

- For example, given the premise "Duncan is a terrier"($T(d)$), we can derive "there exists a dog that is a terrier" ($\exists x: T(x)$).

More Existence

- We have to be careful to introduce the existential quantifier so that its scope covers the entire statement that we are using.
  If I have premises T(d) and R(c), for example,
  I could derive ∃x: T(x) and ∃x: R(x).

- But it would be wrong to derive
  ∃x: (T(x) ∧R(x)), "there is a dog that is both a terrier and a Rottweiler". To get that I would need a single statement T(a) ∧ R(a), with the same a in both places.

Still More Existence

- If I have $\exists x:(T(x) \wedge R(x))$, and I want to derive $\exists y: T(y)$, I should first use Instantiation to say $T(a) \wedge R(a)$ for some a, then Separation to get $T(a)$, then Existence.
- There is no rule to go directly from this premise to this conclusion, and being sloppy with the quantifier rules can lead to errors.

Specification: Eliminating ∀

- If we have a universal statement of the form $\forall x: P(x)$, the rule of **Specification** allows us to derive $P(a)$, where a is any constant or variable of the same type as x. That is, we can derive $P(a)$ for any a *of our choice*.
- If we have the statement $\forall x: W(x)$,"all dogs like walks", we can derive $W(d)$, $W(c)$, or $W(y)$ for a free variable y that already appears in other statements.

More Specification

- An important caveat to remember is that in principle we remove one universal at a time. When we remove it we must set *all* occurrences of the bound variable to the *same* value.

- Given "$\forall x$:[$W(x) \wedge S(x, d) \wedge (T(y) \rightarrow S(y, x))$]", we could get "$W(c) \wedge S(c, d) \wedge (T(y) \rightarrow S(y, c))$" or "$W(y) \wedge S(y, d) \wedge (T(y) \rightarrow S(y, y))$", but we couldn't replace some x's with c's and others with y's.

Still More Specification

- Note, by the way, that in a statement such as "$\forall x$: $W(x) \wedge S(x, d) \wedge (T(y) \rightarrow S(y, x))$", the $\forall x$ has a scope that reaches to the end of the whole statement.

- Another thing we can't do is set x to an existing *bound* variable. We could not go from "$\forall x$: $\forall y$: $T(y) \rightarrow S(x, y)$" to "$\forall y$: $T(y) \rightarrow S(y, y)$". This is because the bound y is to be defined *after* we set the value of x, so we can't force the two to be the same.

Clicker Question #2

- Now we have "$\forall x$: $W(x) \rightarrow \neg S(x, d)$", or "Every dog that likes walks is not smaller than Duncan". Which of these *is* a valid conclusion by Specification?
- (a) If Duncan likes walks, he is not smaller than himself.
- (b) If Rhonda is smaller than Duncan, then dog x does not like walks.
- (c) If every dog likes to walk, then it is not the case that every dog is smaller than Duncan.
- (d) If Cardie is not smaller than Duncan, then she likes walks.

Not the Answer

Clicker Answer #2

- Now we have "$\forall x$: $W(x) \rightarrow \neg S(x, d)$", or "Every dog that likes walks is not smaller than Duncan". Which of these *is* a valid conclusion by Specification?
- (a) If Duncan likes walks, he is not smaller than himself.

  Specifying the premise to Duncan.
- (b) If Rhonda is smaller than Duncan, then dog x does not like walks.

  Wrong, Specifying two different values.
- (c) If every dog likes to walk, then it is not the case that every dog is smaller than Duncan.

  This would be $(\forall x{:}W(x)) \rightarrow \sim(\forall x{:}\ S(x, d))$, a bad "distributive" law.
- (d) If Cardie is not smaller than Duncan, then she likes walks.

  This Specifies the converse $\forall x{:}\sim S(x, d) \rightarrow W(x)$ to Cardie.

Generalization: Introducing ∀

- We've just seen that universal statements are very powerful, so it stands to reason that we should have to work harder to prove them. The rule of **Generalization** allows us to prove new universal statements.

- To prove a statement ∀x: P(x), we first say "let y be arbitrary", where y is a new variable of the type of x. We then have to prove that P(y) is true, without using any information about y other than its type.
  If we do this, we may then derive ∀x: P(x).

More Generalization

- We most often use this in the form $\forall x: (P(x) \rightarrow Q(x))$, so we let y be arbitrary and then have to prove $P(y) \rightarrow Q(y)$.
- To do this we can assume P(y) and use it to derive Q(y), which may be possible if P and Q are related. When we do mathematical induction, we will prove statements of the form $\forall x: P(x)$, where x is a natural, in part by proving $\forall x: (P(x) \rightarrow P(x+1))$.

The Dog Example

- We have a set of dogs D, and predicates R(x) “x is a Rottweiler”, T(x) “x is a terrier”, S(x, y) “x is smaller than y”, W(x) “x likes to go for walks”.
- Our desired conclusion is as follows: “There exists a Rottweiler that is larger than some terrier who likes walks”, which we may write as “$\exists x: \exists y: R(x) \wedge S(y, x) \wedge T(y) \wedge W(y)$”.
- We will work from five premises and find a proof strategy in the next slides.

Clicker Question #3

- Suppose we were trying to prove the statement "$\forall x: \exists y: S(x, y) \vee T(x)$". Which proof strategy is valid?
- (a) Let x be an arbitrary dog, and prove that if it is smaller than some other dog y, then x is a terrier.
- (b) Let x be an arbitrary dog, and find some dog y that, unless x is a terrier, is larger than x.
- (c) Find a dog y such that any arbitrary dog is either smaller than y or is a terrier.
- (d) Find a dog y that is larger than all terriers.

Not the Answer

Clicker Answer #3

- Suppose we were trying to prove the statement "$\forall x:\exists y: S(x, y) \vee T(x)$". Which proof strategy is valid?

Would prove $\forall x:(\exists y:S(x, y)) \rightarrow T(x)$.

- (a) Let x be an arbitrary dog, and prove that if it is smaller than some other dog y, then x is a terrier.
- (b) Let x be an arbitrary dog, and find some dog y that, unless x is a terrier, is larger than x.
- (c) Find a dog y such that any arbitrary dog is either smaller than y or is a terrier.

Would prove $\exists y:\forall x:S(x, y) \vee T(x)$.

- (d) Find a dog y that is larger than all terriers.

Would prove $\exists y:\forall x:T(x) \rightarrow S(x, y)$.

Dog Example Premises

- (1) All dogs like to go for walks: $\forall x: W(x)$
- (2) Duncan is a terrier: $T(d)$
- (3) Cardie is smaller than some Rottweiler: $\exists x: R(x) \land S(c, x)$
- (4) All terriers are smaller than Cardie: $\forall x: T(x) \rightarrow S(x, c)$
- (5) S is transitive: $\forall x: \forall y: \forall z: (S(x, y) \land S(y, z)) \rightarrow S(x, z)$

Dog Example Strategy

- Recall the goal: There exists a Rottweiler that is larger than some terrier who likes walks ($\exists x: \exists y: R(x) \wedge S(y, x) \wedge T(y) \wedge W(y)$).
- Overall strategy: Figure out which dogs x and y ought to be -- maybe constants, maybe dogs forced to exist by the premises.
  In this case y should be Duncan, and x should be the Rottweiler provided by premise (3):
- (3) Cardie is smaller than some Rottweiler: $\exists x: R(x) \wedge S(c, x)$

More of the Dog Example

- Goal: $\exists x: \exists y: R(x) \wedge S(y, x) \wedge T(y) \wedge W(y)$.
- We use Instantiation on (3) $\exists x: R(x) \wedge S(c, x)$ to get a dog r such that $R(r) \wedge S(c, r)$.
- We need four facts about d and r: We have $R(r)$ and we need $W(d)$, $T(d)$, and $S(d, r)$.
- We are given $T(d)$ by (2)
- We get $W(d)$ by Specification on (1) $\forall x: W(x)$

Finishing the Dog Example

- To get S(d, r), we use:
  -- Specification on (4) $\forall x: T(x) \rightarrow S(x, c)$ to get $T(d) \rightarrow S(d, c)$
  -- Modus Ponens to get S(d, c), knowing (2) T(d)
  -- in (5) $\forall x: \forall y: \forall z: (S(x, y) \wedge S(y, z)) \rightarrow S(x, z)$, use Specification for $(S(d, c) \wedge S(c, r)) \rightarrow S(d, r)$ then Conjunction of S(d, c) with S(c, r) from (3) and Modus Ponens to get S(d, r).

- Once we have the four facts R(r), W(d), T(d), and S(d, r), we use Existence twice to get $\exists x: \exists y: R(x) \wedge S(y, x) \wedge T(y) \wedge W(y)$.