CMPSCI 250: Introduction to Computation

Lecture #10: Rules of Inference
David Mix Barrington
28 February 2013

Rules of Inference

- Review of Predicates and Quantifiers
- An Overview of Proofs
- Rules for Propositional Logic
- The Murder Mystery
- Some Fallacies
- Rules for Quantifiers
- A Sample Quantifier Proof

Review of Predicates and Quantifiers

- We defined a **predicate** to be a statement that becomes a proposition if the values of certain **free variables** are supplied.

- The **universal quantifier** $\forall$ and the **existential quantifier** $\exists$ each **bind** a variable, reducing the number of free variables by one. $\forall x:P(x)$ means “for every x, P(x) is true”. $\exists x:P(x)$ means “there exists an x such that P(x) is true”. The type of the bound variable is an essential part of the meaning of the statement.

- We saw some logical equivalences involving quantifiers. $\forall$ **commutes** with $\wedge$ and $\exists$ commutes with $\vee$, but neither commutes with the other. Negations of quantified statements obey the **DeMorgan Laws**: $\neg\exists x:P(x)$ is equivalent to $\forall x:\neg P(x)$ and $\neg\forall x:P(x)$ is equivalent to $\exists x:\neg P(x)$.

An Overview of Proofs

- A **mathematical proof** takes one or more statements called **premises**, and derives one or more statements called **conclusions**. If the premises of a valid proof are true, then the conclusion must be as well. The premises may include statements assumed to be universally true, called **axioms**. A statement that has been proved is called a **theorem**. If we assume a premise P and prove a conclusion C, the theorem we have proved is "$P \rightarrow C$".

- In your previous mathematical career, you are used to constructing chains of algebraic equalities, where each one follows from one of the previous ones by a rule of algebra. In the same way, we may be able to get from our premise to our conclusion by known **logical equivalences**.

- More often we use **rules of inference** in the form of **implications**. For example, if we know that $p \wedge q$ is true, then q must be true. The basic form of our logical proofs will be to write down a sequence of statements, where each one follows from one or more of the previous ones by some rule of inference.

Rules for Propositional Logic

- Any tautology of the form "$p \rightarrow q$" gives us a rule, where if we have p we may derive q. The most generally useful rules have names, but the main point is to only use valid rules -- the validity of a rule may be checked with truth tables.

- Rosen lists eight useful rules on page 72 (7th ed.). We may combine these with the logical equivalences we know, like commutativity and associativity of $\vee$ and $\wedge$.

Modus Ponens: from p and $p \rightarrow q$, derive q
Modus Tollens: from $\neg q$ and $p \rightarrow q$, derive $\neg p$
Hypothetical Syllogism: from $p \rightarrow q$ and $q \rightarrow r$, derive $p \rightarrow r$
Disjunctive Syllogism: from $p \vee q$ and $\neg p$, derive q
Addition: from p, derive $p \vee q$
Simplification: from $p \wedge q$, derive p
Conjunction: from p and q, derive $p \wedge q$
Resolution: from $p \vee q$ and $\neg p \vee r$, derive $q \vee r$

iClicker Question 1: A Valid Inference

- The exclusive or of p and q, $p \oplus q$, is defined to be $(p \land \neg q) \lor (\neg p \land q)$. Which one of these three statements logically follows from $p \oplus q$. That is, which one must be true if $p \oplus q$ is true?

- (a) $p \lor q$

- (b) $\neg p \land \neg q$

- (c) $\neg(p \to q)$

The Murder Mystery

- In our discussion last Monday we looked at two logic problems in the form of murder mysteries. We were given a situation that could be described in terms of a number of boolean variables, and given a number of premises in the form of clues -- compound statements using those variables. Our goal was to determine the truth values of all the variables, if they were determined by the premises. This varies from our usual proofs in that we didn't know the conclusion in advance.

- The basic method was to derive consequences of the premises using valid rules of inference. To get started, we needed to make hypotheses. If we say "assume h", for example, we are starting a **proof by cases**. We may find that if h is true, no valid setting of the variables is possible, because h leads to a **contradiction**. Or we may find that h forces the values of all the variables.

- In a murder mystery, once we have found values of all the variables, we must **check** that these values make *all* the premises evaluate to true.

Some Fallacies

- In *Love and Death*, Woody Allen's character takes the premises "All men are mortal" and "Socrates is a man" and concludes "All men are Socrates". This is an example of a **fallacy** -- the use of a "rule of inference" that is not guaranteed to give a correct conclusion from correct premises. Logicians find it useful to identify and name fallacies that often occur in practice.

- Rosen gives two examples in propositional calculus. The first is the "rule" that derives p from the two statements $p \to q$ and q. (He calls this the **fallacy of affirming the conclusion**.) Of course it's possible for p to be false and q to be true at the same time. You can think of this rule as replacing $p \to q$ by its converse $q \to p$, then using Modus Ponens.

- Similarly from $p \to q$ and $\neg p$, the **fallacy of denying the hypothesis** derives $\neg q$. This can be thought of as replacing $p \to q$ by its inverse $\neg p \to \neg q$, then using Modus Ponens.

iClicker Question 2: Spotting a Fallacy

- Which of these three arguments is **not a fallacy**?
- (a) If Duncan is outside, then he is barking. Duncan is not outside. Therefore he is not barking.
- (b) If Duncan is not barking, then he is asleep. Duncan is asleep. Therefore he is not barking.
- (c) If Duncan is asleep, then he is not barking. Duncan is barking. Therefore he is not asleep.
- (d) Trick question -- all three arguments are fallacies.

Rules for Quantifiers

- There are four basic rules of inference for quantified statements. Two of them **introduce** a new quantifier to the list of proven statements, by deriving a quantified statement with a new bound variable from a statement where that variable is free. The other two **eliminate** a quantifier by taking a statement where a particular variable is quantified, and deriving a statement without that quantifier, where that variable has been replaced by a constant.

- Which rule to use in a given situation depends on what premises we have to work with, and what conclusion we are looking for.

- These rules operate only on the outermost quantifier in a statement, and cannot be used inside another quantifier or a logical operator.

- Three of the rules are very straightforward, and one (the most useful one, of course) is considerably more subtle.

The Three Straightforward Rules

- If we have a premise of the form P(a), where a is a constant of the correct type, we may derive ∃x:P(x) by the rule of **Existential Generalization**. For example, from "Duncan is a terrier", we may derive "There exists one of my dogs that is a terrier", given that Duncan is one of my dogs.

- If we have a premise of the form ∀x:P(x), we may derive P(a), where a is *any* element of the type (of our choice) by the rule of **Universal Instantiation**. For example, from "All my dogs like walks", we may derive "Cardie likes walks", given that Cardie is one of my dogs.

- If we have a premise of the form ∃x:P(x), we may conclude P(a) by **Existential Instantiation**, but here a is a *new* element of the type about which we know nothing except its type and the statement P(a). From the premise "One of my dogs is a retriever", I may derive "Rover is a retriever" and work with Rover in my argument, as long as I don't go assuming that Rover is equal (or is not equal) to any other named dogs occurring in the argument.

iClicker Question #3: Existential Instantiation

- Again the type is "my dogs" and Cardie and Duncan are two of my dogs. Which of these statements is a **valid** inference from the premise "one of my dogs is a terrier and likes walks" using **Existential Instantiation**?
- (a) Let Biscuit be the one of my dogs who is a terrier and likes walks. Then Biscuit likes walks.
- (b) Let Biscuit be the one of my dogs who is a terrier and likes walks. Then Biscuit is not the same dog as either Duncan or Cardie.
- (c) Duncan is a terrier and likes walks.

Universal Generalization

- Universal statements are powerful, because Universal Instantiation lets us use them to derive facts about any elements of the type. It stands to reason, then, that we have to work harder to prove a new universal statement. The rule of **Universal Generalization** allows us to derive a statement of the form $\forall x{:}P(x)$.

- To apply Universal Generalization, we first define a new variable a, of x's type, and assume nothing about a except its type. This is usually expressed by saying "Let a be arbitrary". Then we somehow prove the statement P(a). Finally we conclude that $\forall x{:}P(x)$ is true, by saying "because a was arbitrary, we may conclude $\forall x{:}P(x)$ by Universal Generalization".

- To see this, we can derive "all my dogs like walks" from the premise "all my dogs are both furry and like walks". The premise is "$\forall x{:}(F(x) \wedge W(x))$". Let a be an arbitrary one of my dogs. By Universal Instantiation on the premise, we get "$F(a) \wedge W(a)$". We then derive "W(a)" by simplification. Since a was arbitrary and we proved "W(a)", we may conclude "$\forall x{:}W(x)$".

iClicker Question #4: Universal Modus Ponens

- The rule of **Universal Modus Ponens** lets us conclude Q(a) from the two statements P(a) and $\forall x:(P(x) \rightarrow Q(x))$. Which pair of premises **may not** be used with this rule to derive the proposition "Duncan is noisy"? (Assume again that Duncan is one of my dogs.)

- a) "Duncan is a terrier" and "All terriers are noisy".

- b) "If Rover is one of my dogs and Rover is a terrier, then Rover is noisy" and "Duncan is a terrier".

- c) "Duncan is a terrier" and "There does not exist one of my dogs that is both a terrier and is not noisy".

- d) Trick question, any of the three pairs of premises could be used.

An Example of a Quantifier Proof

- 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".
- 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 ($\exists x: R(x) \wedge S(c, x)$), (4) All terriers are smaller than Cardie ($\forall x: T(x) \rightarrow S(x, c)$) (5) S is transitive ($\forall x: \forall y: \forall z: (S(x, y) \wedge S(y, z)) \rightarrow S(x, z)$).
- Desired conclusion: 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).

More of the Dog Example

- We use EI on (3) 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 have $T(d)$ by (2), and we get $W(d)$ by UI on (1).
- To get $S(d, r)$, we use UI on (4) to get $T(d) \rightarrow S(d, c)$, Modus Ponens to get $S(d, c)$ since we have $T(d)$, and finally UI on (5) to get $(S(d, c) \wedge S(c, r)) \rightarrow S(d, r)$ and Conjunction and Modus Ponens to get $S(d, r)$.
- Once we have these four facts we use EG twice to get our desired conclusion $\exists x: \exists y: R(x) \wedge S(y, x) \wedge T(y) \wedge W(y)$.

Games

- In next Monday's discussion you will see a **two-player game** of perfect information that cannot go on indefinitely (and always has a winner at the end). In such a game, there is always a **winning strategy** for one of the players.

- To prove this rigorously would take mathematical induction, which we haven't done yet, but here is an informal argument. Given the rules of the game, we can create (at least in principle) a **game tree** that has a node for every position and an edge for every move.

- We could (again in principle) label every leaf node of this tree with the winner of the game in that position. We can then label any other node if all of its children are labeled, which will allow us to label more nodes, and so on until the root node (for the start position) is labeled.

- A node is a win for the player whose move it is if there exists a child node where that player wins.