A Proof Primer: Reconciling Velleman style proof strategies with Sequent Proofs

September 4, 2012

Proofs are taught in Dr. Cowles version of COSC 2300 using the methods of Daniel J. Velleman [2]. Velleman's methods are essentially Gentzen's sequent proof rules presented in a more narrative form. My own notes [1] present the proof rules in sequent form. These notes, closely based on notes by Vellman, present the proof strategies in terms of the sets of formulas he calls *givens* and *goals*.

To be completely precise we present the syntax of formulas first.

1 Syntax

1.1 Propositional Logic

We use *propositional variables* to stand for arbitrary propositions and assume there is an infinite supply of these variables.

$$\mathcal{V} = \{p, q, r, p_1, q_1, r_1, p_2, \cdots\}$$

Note that the fact that the set $\mathcal{V}$ is infinite is unimportant since no individual formula will ever require more than some fixed finite number of variables, however it is important that the number of variables we can select from is unbounded. The must always be a way to get another one.

We include the constant symbol $\bot$ (say "bottom").

Complex propositions are constructed by combining simpler ones with *propositional connectives*. For now we leave the meaning of the connectives unspecified and simply present them as one of the symbols $\wedge, \vee, \Rightarrow$ standing for *and*, *or* and *implies* respectively.

Definition 1.1 (Propositional Logic syntax) The syntax of propositional formulas (we denote the set as $\mathcal{P}$) can be described by a grammar as follows:

$$\mathcal{P} ::= \bot \mid x \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \Rightarrow \psi$$

where

$\bot$ is a constant symbol,

$x \in \mathcal{V}$ is a propositional variable, and

$\phi, \psi \in \mathcal{P}$ are meta-variables denoting previously constructed propositional formulas.

1.2 Predicate Logic

In this section we extend propositional logic presented in the previous chapter to allow for *quantification* of the form:

for all things x, $\cdots$
for every x, $\cdots$
there exists a thing x *such that* $\cdots$
for some thing x, $\cdots$

Where "$\cdots$" is some statement referring to the thing denoted by the variable x that specifies a property of the thing denoted by x. The first two forms are called *universal quantification*, they are different ways of asserting that everything satisfies some specified property. The second two forms are called *existential quantification*, they assert that something exists having the specified property.

Symbolically, we write "*for all things* x, $\cdots$ " as $(\forall x. \cdots)$ and "*there exists a thing* x *such that* $\cdots$" as $(\exists x. \cdots)$.

1.3 Predicates

To make this extension to our logic we add truth-valued functions called predicates which map elements from a *domain of discourse* to the values in the set of Boolean values ($\mathbb{B}$).

Definition 1.2 (arity) A function is called *n-ary* if it takes n arguments, $0 \leq n$. If a function is $n - ary$, we say it has *arity* n. A function of arity 0, *i.e.* a function that takes no arguments, is called a *constant*. We say a 0-ary function is *nullary*, 1-ary function is *unary*. We say a 2-ary function is *binary* and, although we could say 3-ary, 4-ary and 5-ary functions

ternary, *quaternary* and *quintary* respectively, we do not insist on carrying this increasingly tortured nomenclature any further.

For example, consider the following functions:

i.) $f() = 5$
ii.) $g(x) = x + 5$
iii.) $h(x, y) = (x + y) - 1$
vi.) $f_1(x, y, z) = x * (y + z)$
v.) $g_1(x, y, z, w) = f_1(x, y, w) - z$

The first function is nullary, it takes *no* arguments. Typically, we will drop the parentheses and write f instead of $f()$. The second function takes one argument and so is a *unary function*. The third function is *binary*. The fourth and fifth are 3-ary and 4-ary functions respectively.

Definition 1.3 (Boolean valued function) A function is *Boolean-valued* if its range is the set $\mathbb{B}$.

Definition 1.4 (predicate) A *predicate* is a n-ary Boolean-valued function over some domain of input.

Example 1.1. In ordinary arithmetic, the binary predicates include *less than* (written $<$) and *equals* (written $=$). Typically these are written in infix notation *i.e.* instead of writing $=(x, y)$ and $<(x, y)$ we write $x = y$ and $x < y$; do not let this infix confuse you, they are still binary predicates. We can define other predicates in terms of these two. For example we can define a binary predicate *less-than-or-equals* as:

$$i \leq j \stackrel{\text{def}}{=} ((i = j) \vee (i < j))$$

We could define a unary predicate which is true when its argument is equal to 0 and is false otherwise:

$$=_0 (i) \stackrel{\text{def}}{=} i = 0$$

We could define a 3-ary predicate which is true if k is strictly between i and j:

$$\texttt{between}(i, j, k) \stackrel{\text{def}}{=} ((i < k) \wedge (k < j))$$

Note that predicate constants act just like propositional variables.

Predicate logic formulas are constructed from two sorts of components: terms and formulas which may contain terms.

i.) parts that refer to objects and functions on those objects in the domain of discourse. These components of the formula are called *terms*.

ii.) parts of a formula that denote truth values, these include predicates over the domain of discourse and formulas constructed inductively by connecting previously constructed formulas.

1.4 Variables

The definitions of the syntactic classes of terms and formulas (both defined below) depend on an unbounded collection of variable symbols, we call this set $\mathcal{V}$.

$$\mathcal{V} = \{x, y, z, w, x_1, y_1, z_1, w_1, x_2, \cdots\}$$

Unlike propositional variables, which denoted truth-values, these variables will range over individual elements in the domain of discourse. So, for formulas where the quantifiers are understood to range over integer values, the variables range over integer values. Like propositional variables, we assume the set $\mathcal{V}$ is fixed (and so we do not include it among the parameters of the definitions that use it.)

1.5 Terms

The syntax of terms (the collection of which we will write as $\mathcal{T}$) is determined by a set of n-ary function symbols, call this set $\mathcal{F}$. We assume the arity of a function symbol can be determined.

Definition 1.5 (Terms) *Terms* are defined over a set of function symbols $\mathcal{F}$ are given by the following grammar:

$$\mathcal{T}_{[\mathcal{F}]} ::= x \mid f(t_1, \cdots t_n)$$

where:

$\mathcal{F}$ is a set of function symbols,
$x \in \mathcal{V}$ is a variable,
$f \in \mathcal{F}$ is a function symbol for a function of arity n, where $n \geq 0$ and
$t_i \in \mathcal{T}_{[\mathcal{F}]}$ denote previously constructed terms, $1 \leq i \leq n$.

Note that the definition of terms is parametrized by the set of function symbols. The set of terms in $\mathcal{T}_{[\mathcal{F}]}$ is determined by the set of function symbols in $\mathcal{F}$ and by the arities of those symbols. Also, note that if $n = 0$, the term $f()$ is a constant and we will write it simply as f.

Example 1.2. Let $\mathcal{F} = \{a, b, f, g\}$ where a and b are constants, f is a unary function symbol and g is a binary function symbol. In this case, $\mathcal{T}$ includes:

$$\begin{array}{l}\{a, x, f(a), f(x),\\ g(a,a), g(a,x), g(a,f(a)), g(a,f(x)),\\ g(x,a), g(x,x), g(x,f(a)), g(x,f(x)),\\ b, y, f(b), f(y), f(f(a)), f(f(x)), f(g(a,a)), \cdots\end{array}$$

1.6 Formulas

Definition 1.6 (Predicate Logic Formula) *Formulas* of predicate logic are defined over a set of function symbols $\mathcal{F}$ and a set of predicate symbols $\mathcal{P}$ and are given by the following grammar.

$$\mathcal{PL}_{[\mathcal{F},\mathcal{P}]} ::= \bot \mid P(t_1, \cdots, t_n) \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \Rightarrow \psi \mid \forall x.\phi \mid \exists x.\phi$$

where:

$\mathcal{F}$ is set of function symbols,

$\mathcal{P}$ is set of predicate symbols,

$\bot$ is a constant symbol,

$P \in \mathcal{P}$ is a predicate symbol for a predicate of arity n, where $n \geq 0$,

$t_i \in \mathcal{T}_{[\mathcal{F}]}$ are terms, $1 \leq i \leq n$,

$\phi, \psi \in \mathcal{PL}_{[\mathcal{F},\mathcal{P}]}$ are previously constructed formulas, and

$x \in \mathcal{V}$ is a variable.

This definition is parametrized by the set of function symbols ($\mathcal{F}$) and the set of predicate symbols ($\mathcal{P}$). As remarked above, a predicate symbol denoting a constant is the equivalent of a propositional variable presented in the previous chapter. Thus, predicate symbols are a generalization of propositional variables; when actual values are substituted for their variables, they denote truth values.

We will sometimes write $\phi[x]$ to indicate that the variable x may occur in the formula ϕ.

1.6.1 Predicate Logic extends Propositional Logic

Given a rich enough set of predicate symbols $\mathcal{P}$ *i.e.* one that includes one constant symbol for each propositional variable, the language of predicate logic extends the language of propositional logic. Specifically, every formula

of propositional logic is a formula of predicate logic. To see this note that: the constant symbol bottom ($\bot$) is included in both languages; the propositional variables are all included in $\mathcal{P}$ as predicate symbols of arity 0. Also, every connective of propositional logic is also a connective of predicate logic. Thus, we conclude that every formula of propositional logic can be identified with a syntactically identical formula of predicate logic.

We will see in later sections that not only is the syntax preserved, both the semantics and the proof system are also preserved.

2 Proofs

At any point in a proof there is a set of the formulas you have (so-far) assumed to be true; these are the *givens*. There is also a set of formulas, one of which you must show is true; these are the *goals*. Thus, pairs of sets of formulas capture the state of a proof. In the sequent presentation the proof state is written $\Gamma \vdash \Delta$ where Γ is the set of givens and Δ is the set of goals. In a sequent, we write $\Gamma_1, \phi, \Gamma_2 \vdash \Delta$ to indicate ϕ is in the set of givens and write $\Gamma \vdash \Delta_1, \phi, \Delta_2$ to indicate ϕ is in the set of goals.

To prove a formula is valid, you start with an empty set of givens and the set of goals includes the single formula to be proved. Proofs proceed by applying one of the proof strategies to a formula in the givens or the goals. Depending on the strategy, this will either result in a completed proof or one or two new given-goal sets. So proofs are a kind of tree structure with the nodes pairs of given-goal sets and the edges the names of the proof strategy used to get to that node.

Proof rules for the propositional sequent calculus have one of the following three forms:

$$\frac{}{\mathcal{C}}(N) \qquad\qquad \frac{\mathcal{H}}{\mathcal{C}}(N) \qquad\qquad \frac{\mathcal{H}_1 \ \mathcal{H}_2}{\mathcal{C}}(N)$$

where $\mathcal{C}, \mathcal{H}, \mathcal{H}_1$, and $\mathcal{H}_2$ are all schematic sequents. N is the name of the rule. The $\mathcal{H}$ patterns are the *premises* (or *hypotheses*) of the rule and the pattern $\mathcal{C}$ is the *goal* (or *conclusion*) of the rule. Rules having no premises are called *axioms*.

The proof strategies are divided into three groups. There are axiom rules, there are rules to apply to formulas in the givens (or left rules) and there are rules to apply to formulas in the goals (or right rules). These left and right rules come in pairs, two for each non-atomic syntactic class of formulas with one rule describing what to do if the formula is in the set of givens and another describing what to do if it is in the goal set.

2.1 Axiom Rules

There are two axioms rules. One form is when the given-goal set contain some formula in common. If $\Gamma \cap \Delta \neq \{\}$ then there is some formula (say ϕ) they have in common. The justification for this rule is that if you have assumed the formula ϕ is true (and you must have because it is in the set Γ of givens) then since ϕ is one of the goals you're done. Here's the Sequent form of the rule.

Proof Rule 2.1 (Ax)

$$\frac{}{\Gamma_1, \phi, \Gamma_2 \vdash \Delta_1, \phi, \Delta_2} \text{ (Ax)}$$

The other axiom rule says if $\bot$ is in the set Γ of givens, you're done. The justification for this rule is that anything follows from a false assumption.

Proof Rule 2.2 ($\bot$Ax)

$$\frac{}{\Gamma_1, \bot, \Gamma_2 \vdash \Delta} \text{ ($\bot$Ax)}$$

2.1.1 Proof Strategies for conjunctions

If a formula of the form $\phi \wedge \psi$ is in the set of givens - then continue with the same set of goals and a new givens set that includes both ϕ and ψ. The justification is if you've assumed $\phi \wedge \psi$ then certainly you can assume ϕ and ψ separately. The sequent proof rule looks like this:

Proof Rule 2.3 ($\wedge$L)

$$\frac{\Gamma_1, \phi, \psi, \Gamma_2 \vdash \Delta}{\Gamma_1, (\phi \wedge \psi), \Gamma_2 \vdash \Delta} \text{ ($\wedge$L)}$$

If a formula of the form $\phi \wedge \psi$ is in the goal set - Then there are two things to show. In both cases you work with the same given set Γ. In pone case you add ϕ to the goal set and in the other you add ψ. The justification is if you must show ϕ and ψ you must show them separately.

Proof Rule 2.4 ($\wedge$R)

$$\frac{\Gamma \vdash \Delta_1, \phi, \Delta_2 \quad \Gamma \vdash \Delta_1, \psi, \Delta_2}{\Gamma \vdash \Delta_1, (\phi \wedge \psi), \Delta_2} \ (\wedge\text{R})$$

2.1.2 Proof Strategies for Disjunctions

If a formula of the form $\phi \vee \psi$ is in the set of givens - there are two things to prove. In both subproofs, the goal set remains the same. In one branch of the proof you add ϕ to the set of givens and in the other you add ψ. The justification for this rule is, if you have assumed $\phi \vee \psi$ then you know that at least one (possibly both) of ϕ or ψ is true, but you don't know which one - so you need to consider both cases.

Proof Rule 2.5 ($\vee$L)

$$\frac{\Gamma_1, \phi, \Gamma_2 \vdash \Delta \quad \Gamma_1, \psi, \Gamma_2 \vdash \Delta}{\Gamma_1, (\phi \vee \psi), \Gamma_2 \vdash \Delta} \ (\vee\text{L})$$

If a formula of the form $\phi \vee \psi$ is in the goal set - you continue with the new proof state having the same set of givens and you add ϕ and ψ individually to the goal set. The justification for this rule is if you are to show $\phi \vee \psi$ then you must show at least one of them.

Proof Rule 2.6 ($\vee$R)

$$\frac{\Gamma \vdash \Delta_1, \phi, \psi, \Delta_2}{\Gamma, \vdash \Delta_1, (\phi \vee \psi), \Delta_2} \ (\vee\text{R})$$

2.1.3 Proof Strategies for Implications

If a formula of the form $\phi \Rightarrow \psi$ is in the set of givens - then you have two subgoals. In one of the subgoals, you keep the same set of givens and add ϕ to the goal set. In the other subgoal, you add ψ to the set of givens and keep the goal set the same.

Proof Rule 2.7 ($\Rightarrow$L)

$$\frac{\Gamma_1, \Gamma_2 \vdash \phi, \Delta \quad \Gamma_1, \psi, \Gamma_2 \vdash \Delta}{\Gamma_1, (\phi \Rightarrow \psi), \Gamma_2 \vdash \Delta} \ (\Rightarrow\text{L})$$

If a formula of the form $\phi \Rightarrow \psi$ is in the goal set - there is one subgoal. Add ϕ to the set of givens and add ψ to the set of goals and continue. The justification for this rule is - to show an implication $\phi \Rightarrow \psi$ assume the antecedent ϕ and show ψ.

Proof Rule 2.8 ($\Rightarrow$R)

$$\frac{\Gamma, \phi \vdash \Delta_1, \psi, \Delta_2}{\Gamma \vdash \Delta_1, (\phi \Rightarrow \psi), \Delta_2} \ (\Rightarrow\text{R})$$

2.1.4 Proof Strategies for Negations

If a formula of the form $\neg\phi$ is in the set of givens - continue the proof with the same given set and a new goal set containing the old goals together with the formula ϕ. Thus, the new goal set is of the form $\{\phi\} \cup \Delta$. The justification for this rule is based on the observation that if you have assumed $\neg\phi$ and you can subsequently show ϕ then there must be a contradiction lurking in your assumptions.

Proof Rule 2.9 ($\neg$L)

$$\frac{\Gamma_1, \Gamma_2 \vdash \phi, \Delta}{\Gamma_1, \neg\phi, \Gamma_2 \vdash \Delta} \ (\neg\text{L})$$

If a formula of the form $\neg\phi$ is in the goal set - continue the proof with a new given set $\{\phi\} \cup \Gamma$ and use the same goal set you started with less the formula $\neg\phi$ and including $\bot$[1]. The justification for this rule is somewhat oblique. If you can derive $\bot$ from assuming ϕ then $\neg\phi$ must be true.

Proof Rule 2.10 ($\neg$R)

$$\frac{\Gamma, \phi \vdash \Delta_1, \Delta_2}{\Gamma \vdash \Delta_1, \neg\phi, \Delta_2} \ (\neg\text{R})$$

[1]Strictly speaking it is not necessary to add $\bot$ to the goal set but it makes the justification easier to understand (I hope) and explains what to do if the formula $\neg\phi$ is the only formula in Γ.

2.1.5 Proof Strategies for Universals

If a formula of the form $\forall x.\phi[x]$ is in the set of givens - then you can choose any term t to replace for x in the formula $\phi[x]$. Add this new formula to the set of givens and leave the goals the same as before and continue. The hard part is that to choose which t is the one that will help in proof. There is a technical complication, you need to make sure that when you are replacing t for x in ϕ you do not capture any free variable in t by a quantifier in ϕ. See [1] for a more detailed description of how this works.

Proof Rule 2.11.

$$\frac{\Gamma_1, \phi[x := t], \Gamma_2 \vdash \Delta}{\Gamma_1, \forall x.\phi, \Gamma_2 \vdash \Delta} \ (\forall \text{L}) \qquad \text{where } t \text{ is any term.}$$

If a formula of the form $\forall x.\phi[x]$ is in the goal set - there is one subgoal to prove. Choose a fresh variable (say y) that does not occur anywhere in any formula in the givens or goals and add the formula $\phi[y]$ to the goal set. Since y does not occur anywhere in the current proof - it must refer to an *arbitrary* element of the domain of discourse. If you can prove the theorem holds for $\phi[y]$ then it must hold for all elements of the domain of discourse.

Proof Rule 2.12.

$$\frac{\Gamma \vdash \Delta_1, \phi[y], \Delta_2}{\Gamma \vdash \Delta_1, \forall x.\phi[x], \Delta_2} \ (\forall \text{R}) \qquad \begin{array}{l}\text{where variable } y \text{ is not free in any} \\ \text{formula of } (\Gamma \cup \Delta_1 \cup \{\forall x.\phi\} \cup \Delta_2).\end{array}$$

2.1.6 Proof Strategies for Existentials

If a formula of the form $\exists x.\phi[x]$ is in the set of givens - then there is one subgoal to show. Introduce a fresh variable (say y) and add the formula $\phi[y]$ to the set of givens and continue with the proof. Since $\exists x.\phi[x]$ was assumed, $\phi[y]$ holds for some y - you just don't know which one. Since y is completely new, it is an arbitrary element of the domain of discourse.

Proof Rule 2.13.

$$\frac{\Gamma_1, \phi[y], \Gamma_2 \vdash \Delta}{\Gamma_1, \exists x.\phi[x], \Gamma_2 \vdash \Delta} \ (\exists \text{L}) \qquad \begin{array}{l}\text{where variable } y \text{ is not free in any} \\ \text{formula of } (\Gamma_1 \cup \Gamma_2 \cup \{\exists x.\phi\} \cup \Delta).\end{array}$$

If a formula of the form $\exists x.\phi[x]$ is in the goal set - In this case choose a term t (the witness) and add $\phi[t]$ to the goal set.

Proof Rule 2.14.

$$\frac{\Gamma \vdash \Delta_1, \phi[t], \Delta_2}{\Gamma \vdash \Delta_1, \exists x.\phi[x], \Delta_2}\ (\exists \mathrm{R}) \qquad \text{where } t \text{ is any term.}$$

2.2 Techniques That Can Be Used in Any Proof

Proof by contradiction: Assume the goal is false and derive a contradiction. Thus, if ϕ is a gormula in the goal set, add $\neg\phi$ to the set of givens and $\perp$ to the goals and proceed.

Proof by cases: Since $\phi \vee \neg\phi$ holds, you can add $\phi \vee \phi$ to the givesn at any time. In other situations you may know that $\phi_1 \vee \phi_2 \vee \cdots \vee \phi_k$ exhausts all possibilities in which case you can add that formula to the givens and proceed. An example would be - when reasoning about integers you know $x < 0 \vee x = 0 \vee x > 0$ - a case split like this one can be useful in some proofs.

References

[1] James Caldwell. *Logic and Discrete Mathematics for Computer Scientists.* College Publications, August 2011.

[2] Daniel J. Velleman. *How To Prove It: A Structured Approach.* Cambridge University Press, 1994.