How to prove it in Natural Deduction: A Tactical Approach *

Favio E. Miranda-Perea[1], P. Selene Linares-Arévalo[1], and Atocha Aliseda[2]

1 **Departamento de Matemáticas, Facultad de Ciencias, Universidad Nacional Autónoma de México. Circuito Exterior s/n, Ciudad Universitaria C.P 04510, México D.F.**
   `favio@ciencias.unam.mx`   `selene_linares@ciencias.unam.mx`
2 **Instituto de investigaciones Filosóficas. Universidad Nacional Autónoma de México. Circuito Mario de la Cueva s/n, Ciudad Universitaria, C.P 04510, México D.F.**
   `atocha@filosoficas.unam.mx`

―――― **Abstract** ――――

The motivation for this paper comes out of our experience with teaching natural deduction (ND) and with the way this formal system is implemented by the COQ proof assistant, namely by means of so-called tactics, which are heuristics that transform a goal formula into a sequence of subgoals whose provability implies that of the original formula. We aim at capturing some of these tactics into a system of ND for minimal logic. Our goal is twofold: formal and didactic. The former delivers a formal system with its underlying heuristics to build proofs, which in turn serves our latter purpose, that of making an ideal system for the teaching of ND at an undergraduate level in a computer science program.

1 Introduction

The importance of logic in mathematics and computer science is unquestionable. The use of proof-assistants, whose kernel are implemented logics, in the verification and certification of software and mathematical proofs is coming of age. A proof-assistant is a computer system that consists of a domain-specific language representing logical objects, as well as definitions and theorems about these objects, together with a mechanism that allows for the interactive construction and validation of proofs. Though proof assistants have been used to teach logic (e.g. [9]), we think there is still a gap between the traditional way of teaching deductive systems, in particular natural deduction (ND from now onwards), using [2] for example, and the use of proof-assistants to solve tasks of software construction and verification, like the ones tackled in a programming language foundations course (c.f. [7]). This paper claims to be a contribution to filling this gap.

Our main goal is to teach ND to undergraduate computer science students in the way this formal system is implemented by a proof-assistant, specifically COQ (`http://coq.inria.fr`). This way, the migration from studying logic, to using it, for example in a programming languages foundations or a formal verification course, will be smooth, as the transition will be mainly syntactical. We pursuit this goal from the theoretical side, that is, we do not discuss here how to use COQ to teach logic. To the best of our knowledge, our approach is novel, for we capture COQ's native proof search mechanism for first order logic by giving a formal definition of its procedural way of constructing proofs, which corresponds to what we may call "usual mathematical reasoning".
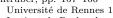
―――――――――

2   Natural Deduction

ND is, as the name says, a formalism that captures natural reasoning, as opposed to the formal reasoning given by Hilbert's axiomatic systems, recall for example the derivation of $A \to A$ in such a system. The flavor of ND we choose is one with localized hypotheses, called S-systems in [3], for this version is more akin to computer science applications. In these systems, inference rules are not applied over formulas, but over judgments (sequents), which are pairs of the form $\Gamma \vdash A$ where A is a formula and Γ is a set of formulas called the context. The intuitive meaning of such a sequent is that formula A is provable from the hypotheses in Γ. This way, all current hypotheses, i.e. formulae in Γ, are available at every derivation step. Therefore there is no need for a global discharging mechanism[1], unlike Fitch-style ND as presented in [2, 5]. The specific inference rules are:

$$\frac{}{\Gamma, A \vdash A} \; (Hyp)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \; (\to I) \qquad\qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B} \; (\to E)$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \; (\wedge I) \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \; (\wedge E) \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \; (\wedge E)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \; (\vee I) \qquad\qquad \frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \; (\vee I)$$

$$\frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C} \; (\vee E)$$

$$\frac{\Gamma \vdash A \;\; x \notin FV(\Gamma)}{\Gamma \vdash \forall x A} (\forall I) \qquad\qquad \frac{\Gamma \vdash \forall x A}{\Gamma \vdash A[x := t]} (\forall E)$$

$$\frac{\Gamma \vdash A[x := t]}{\Gamma \vdash \exists x A} (\exists I) \qquad \frac{\Gamma \vdash \exists x A \quad \Gamma, A \vdash C \quad x \notin FV(\Gamma, C)}{\Gamma \vdash C} (\exists E)$$

As usual, we have a starting rule or axiom scheme (Hyp); introduction rules (I), useful to prove a formula according to its logical form (i.e., its main connective or quantifier) and elimination rules (E), useful to obtain information from a formula. Note that neither the false constant $\bot$ nor the negation operator $\neg$ are present and therefore we are dealing with minimal logic. The reason is that we are interested in logic from an algorithmic point of view and classical logic departs from this view. Our notion of proof is therefore linear as opposed to a tree form. From a theoretical perspective, this choice is irrelevant, but the linear form is in accord with the way proof-assistants operate.

▶ **Definition 1.** A proof or derivation of judgement $\mathcal{J} =_{def} \Gamma \vdash A$ from the set of judgements $\mathbb{H}$ is a finite sequence of judgements $\Pi = \langle \mathcal{J}_1, \ldots, \mathcal{J}_k \rangle$ such that $\mathcal{J}_k = \mathcal{J}$ and for every $1 \le i \le k$ one of the following conditions hold:

---

[1] i.e. One that involves specific previous parts of the derivation.

- $\mathcal{J}_i$ is an instance of the (Hyp) rule
- $\mathcal{J}_i \in \mathbb{H}$
- For every $1 \le i \le k$, $\mathcal{J}_i$ is the conclusion of an instance of some inference rule whose premises are $\mathcal{J}_{l_1}, \ldots, \mathcal{J}_{l_n}$ with $l_1, \ldots, l_n < i$

We say that $\Gamma \vdash A$ is provable, or simply that $\Gamma \vdash A$ holds, if it is provable from the empty set of sequents $\mathbb{H} = \varnothing$.

▶ **Example 2.** Let $\Gamma = \{p \to q \vee r \,,\, q \to r \,,\, r \to s\}$, we want to show that $\Gamma \vdash p \to s$ holds. The following is a derivation of this sequent[2]:

1	$\Gamma, p \vdash p$	(Hyp)
2	$\Gamma, p \vdash p \to q \vee r$	(Hyp)
3	$\Gamma, p \vdash q \vee r$	$(\to E)\ 1, 2$
4	$\Gamma, p, q \vdash q$	(Hyp)
5	$\Gamma, p, q \vdash q \to r$	(Hyp)
6	$\Gamma, p, q \vdash r$	$(\to E)\ 4, 5$
7	$\Gamma, p, r \vdash r$	(Hyp)
8	$\Gamma, p \vdash r$	$(\vee E)\ 3, 6, 7$
9	$\Gamma, p \vdash r \to s$	(Hyp)
10	$\Gamma, p \vdash s$	$(\to E)\ 8, 9$
11	$\Gamma \vdash p \to s$	$(\to I)\ 10$

At this point, following definitions and a few examples, most books demand from the students the task of solving more exercises. There is no discussion as to how to master or even perform a derivation process! The very question of *how to prove it* is unanswered and left to the creativeness and luck of the student. There is however a marginal, though remarkable exception in philosophy, which goes back to the Greeks in their study of *analysis* and *synthesis*; the former depicts the backward process of working a mathematical proof, the latter concerned with the forward derivation in a proof. More recently, a modern pioneering work in the study of *heuristics* is [8], in which heuristic strategies and guidance are given to solve mathematical problems. Some other proposals which provide guidance to proof construction are found in a brief discussion in [2], and a much better one in [5].

As the above proof suggests, performing the derivation process is not easy, perhaps due to the rigidity of definition 1. Just note that more than half of the steps are instances of the (Hyp) rule, which is mainly needed to be able to apply $(\to E)$. These steps do not at all reflect a natural way of reasoning and thus make S-systems inconvenient, as opposed to Fitch-style systems.

The question of how to obtain such proof is quite difficult to answer directly, but a clever student can justify it by appealing to the following mathematical reasoning. The formula we have to proof is an implication, therefore we assume the antecedent p and prove the consequent s, which corresponds to steps 11 and 10 in the previous proof. Next we observe that $r \to s$ is part of the premises, so it suffices to prove $\Gamma, p \vdash r$ (step 8). Here the application of $(\to E)$ is implicit, but the proof must be explicit and forces us to add step 9. To prove the sequent at step 8, we might try to reason as in steps 10 and 11, using

---

[2] The last column is not part of the formal derivation, but consists of justifications for every step, according to definition 1.

the premise $q \to r$, which would lead us to seek a proof of Γ, $p \vdash q$. But after some failed attempts we realize that $q \vee r$ is derivable from the current premises (steps 1 to 3) and that by a case analysis on this formula we can easily obtain r as follows. If q happens then the premise $q \to r$ yields r by $(\to E)$ (steps 4 to 6), and if r is the case, we are done (step 7). Following this reasoning, the student is faced with the problem of constructing the formal proof, which requires that we start with instances of the (Hyp) rule and go forward to obtain what we need. By comparison with the reasoning used to obtain the steps of the proof, this is, quite unnatural. Intuitively, all ND proofs can be constructed by such mathematical reasoning, but it is not clear in general, how to transform this reasoning into a formal derivation. An attempt, that combines both, reasoning forward from the hypotheses and backward from the conclusion, is the intercalation calculus, implemented by the APROS system (`http://www.phil.cmu.edu/projects/apros/`). Our purpose here is not to elaborate further on this, but to put forward a formal notion of backwards derivation, which we call *derivation by tactics* (to be found in definition 5), one that produces proofs equivalent to the usual ones (equivalence proof will be stated as theorem 9) and that captures the procedural reasoning *"understood"* by proof assistants.

Let us start by stating some heuristics.

2.1   Heuristics

The following two strategies can be used systematically to build a proof of a given judgement:

- *Backward reasoning*: To show that $\Gamma \vdash A$ is derivable start by analyzing the judgement and work backwards towards the axioms. To generate a proof, mantain a queue of current goals (sequents whose derivations are to be searched for), which initially includes only the original judgement. Remove a judgment from this queue, and consider a rule whose conclusion is that judgement, adding the premises of such a rule to the queue, as subgoals. This process must be repeated, with the same starting queue, for each possible rule. The process terminates when the queue is empty, meaning all goals have been achieved.
- *Forward reasoning*: To show that $\Gamma \vdash A$ is derivable, start with axioms and work forwards towards the desired judgement. To generate a proof, extend a sequence S of already derived judgements, which is initially empty, by adding to it the conclusion of any rule whose premises are in S. This process generates several sequences and, assuming that all rules are considered to add new sequents, the strategy will eventually find a derivation of the original judgement.

We are interesed in developing a goal-directed system based on backward reasoning that allows for the automation of proof construction, using a mathematical reasoning, as sketched above. To do so we restate our strategy more formally, following the proof strategies discussed in [5].

- Our initial goal is the judgement we want to prove, say $\mathcal{G} =_{def} \Gamma \vdash A$.
- By analyzing Γ and A, we substitute the initial goal $\mathcal{G}$ by simpler subgoals, say $\mathcal{G}_1, \ldots, \mathcal{G}_k$, whose provability implies the provability of $\mathcal{G}$. After the substitution $\mathcal{G}_1$ becomes the *current goal*. This process can be done using one of the following *heuristic rules*:
- *Conclusion Analysis* (CA): The analysis of A leads us to identify its logical form (i.e. its main operator, a connective or a quantifier) and substitute the original goal by the premises of the corresponding introduction rule. Of course, if A is atomic this kind of analysis is not useful.

- *Premise Analysis* (PA): We focus on an specific formula B in Γ which, according to our previous experience and formerly derived judgements, can help us to prove the original formula A. This analysis generates new subgoals either by modifying the context according to the logical form of B or by modifying the conclusion.
- *Lemma Assertion* (LA): Sometimes the current goal is not directly provable by CA and PA, but follows from an intermediate judgement, like a lemma, which is derivable from the current hypotheses.
- A clever combination of the three previous items eventually yields a current goal which is evident (i.e. is an axiom, a hypothesis or an already proved judgement) and therefore discarded from the current sequence of subgoals.
- The process ends succesfully when there are no more subgoals to prove.

A justification for the adequacy of our heuristic rules, will be given in next section. According to the logical form of a conclusion or premise, the heuristics (CA, PA or LA) generate different subgoals. For example, if the current goal is $\Gamma \vdash B \wedge C$, CA yields two subgoals, namely $\Gamma \vdash B$ and $\Gamma \vdash C$, being $\Gamma \vdash B$ the new current goal. Each of these ways of generating subgoals is what we call a *tactic*.

Even if it may seem a bit redundant, we would like to rework example 2 in order to make explicit the application of our heuristic rules: the initial goal is (the judgement in) step 11, and applying CA to this sequent yields step 10; from this PA on the premise $r \to s$ leads to step 8, thus avoiding step 9. To prove step 8, we try PA on $q \to r$, which yields $\Gamma, p \vdash q$. This sequent is not derivable by our own means, but fortunately we realize that, according to the information on Γ, the formula r is a consequence of $q \vee r$, so we can use LA, the lemma being $q \vee r$. This is provable by means of PA on the premise $p \to q \vee r$, this avoids step 2 and yields step 1 as trivial goal, hence the lemma is proved. As we now know $q \vee r$, we can substitute the goal in step 8 by $\Gamma, p, q \vee r \vdash r$, and then use PA on $q \vee r$, which generates two subgoals, namely steps 6 and 7, the latter is a trivial goal whereas the former is consequence of step 4 by means of PA on premise $q \to r$, and thus step 5 is avoided. At this moment, step 4 is the only remaining subgoal, which is trivial and therefore the proof is finished.

Let us present this train of reasoning in a more systematic way, one that gives guidance to the student about *how to prove it*. The following sequence is not a derivation in the sense of definition 1, but a sequence of backward reasonings where step $i + 1$ is obtained by applying our heuristic rules CA,PA or LA to the first sequent in step i.

1	$\Gamma \vdash p \to s$	Original Goal
2	$\Gamma, p \vdash s$	CA.
3	$\Gamma, p \vdash r$	PA($r \to s$).

The current goal is $\Gamma, p \vdash r$. As r is atomic, CA is useless; we try PA and find that $q \to r$ seems a natural candidate to continue the process, but this would lead us to a dead end when trying to prove sequent $\Gamma, p \vdash q$. Instead, LA comes in action, and generates two subgoals (separated by a semicolon):

4   $\Gamma, p \vdash q \vee r$; $\Gamma, p, q \vee r \vdash r$ LA($q \vee r$).

The current goal is now the lemma $\Gamma, p \vdash q \vee r$ and PA yields

5   $\Gamma, p \vdash p$; $\Gamma, p, q \vee r \vdash r$ PA($p \to q \vee r$).

The new current goal is $\Gamma,\ p \vdash p$, which is trivial and can be discarded

    6   $\Gamma,\ p,\ q \vee r \vdash r$ trivial

The case analysis is now a consequence of PA($q \vee r$). The specific tactic generates two subgoals, each assuming one of q and r. In the first case r is proved by means of PA($q \to r$), and in the second the proof is trivial, for r is itself a premise. This way, all subgoals are proved and the original goal succeeds.

    7   $\Gamma,\ p,\ q \vdash r$; $\Gamma,\ p,\ r \vdash r$ PA($q \vee r$).
    8   $\Gamma,\ p,\ q \vdash q$; $\Gamma,\ p,\ r \vdash r$ PA($q \to r$).
    9   $\Gamma,\ p,\ r \vdash r$ trivial.
  10   trivial.

The last step indicates that there are no more subgoals left and therefore the proof is finished.

The aim of this work is to capture the above reasoning by means of a formal system. It is clear that this kind of derivation is sequential, meaning that each step depends solely on the previous one. Therefore we can view this process as a transition system.

2.2 A Tactical Approach to ND: Formalization

We are now in a position to formalize the previously discussed heuristic processes by means of a transition system that manipulates sequences of goals (judgements).

▶ **Definition 3.** A transition system is a triple of the form $\mathcal{T} = \langle \mathbb{S}, \mathcal{F}, \rhd \rangle$ where $\mathbb{S} \neq \varnothing$ is a set of states or configurations; $\mathcal{F} \subseteq \mathbb{S}$ is the set of final or terminal states; $\rhd \subseteq \mathbb{S} \times \mathbb{S}$ is a binary relation on $\mathbb{S}$, called the transition relation, such that for every $F \in \mathcal{F}$ and every $S \in \mathbb{S}$, $F \not\rhd S$. That is, there are no transitions from final states.

For the sake of clarity, let us say that a goal $\mathcal{G}$ is a judgement, we denote a finite sequence of goals as $\mathcal{S} =_{def} \mathcal{G}_1; \ldots; \mathcal{G}_k$. The set of such sequences is $GSeq$, in particular we have $\square \in SeqG$, where $\square$ represents the empty goal sequence.

The transition system below captures the backward reasoning strategy by formalizing the process of substituting a goal with its corresponding subgoals according to the heuristic rules CA,PA or LA.

▶ **Definition 4.** The transition system of tactics is defined as $\mathcal{T} = \langle GSeq, \{\square\}, \rhd \rangle$. That is, a state is a sequence of goals; the only terminal state is the empty sequence of goals $\square$; and the transition relation $\rhd$ between states is given by the following rules, called *tactics*:

- Conclusion Analysis (CA):
    - `intro`: $\Gamma \vdash A \to B; \mathcal{S} \ \rhd \ \Gamma, A \vdash B; \mathcal{S}$
    - `split`: $\Gamma \vdash A \wedge B; \mathcal{S} \ \rhd \ \Gamma \vdash A; \Gamma \vdash B; \mathcal{S}$
    - `left`: $\Gamma \vdash A \vee B; \mathcal{S} \ \rhd \ \Gamma \vdash A; \mathcal{S}$
    - `right`: $\Gamma \vdash A \vee B; \mathcal{S} \ \rhd \ \Gamma \vdash B; \mathcal{S}$
    - `intro`: $\Gamma \vdash \forall x A; \mathcal{S} \ \rhd \ \Gamma \vdash A; \mathcal{S}$ where w.l.o.g., $x \notin FV(\Gamma)$
    - `exists`: $\Gamma \vdash \exists x A; \mathcal{S} \ \rhd \ \Gamma \vdash A[x := t]; \mathcal{S}$

- Premise Analysis (PA):
    - `apply`: $\Gamma, A \to B \vdash B; \mathcal{S} \;\triangleright\; \Gamma, A \to B \vdash A; \mathcal{S}$
    - `destruct`: $\Gamma, A \wedge B \vdash C; \mathcal{S} \;\triangleright\; \Gamma, A, B \vdash C; \mathcal{S}$
    - `destruct`: $\Gamma, A \vee B \vdash C; \mathcal{S} \;\triangleright\; \Gamma, A \vdash C; \Gamma, B \vdash C; \mathcal{S}$
    - `destruct`: $\Gamma, \exists x A \vdash C; \mathcal{S} \;\triangleright\; \Gamma, A \vdash C; \mathcal{S}$ where w.l.o.g. $x \notin FV(\Gamma)$
- Lemma Assertion (LA):
    - `assert`: $\Gamma \vdash C; \mathcal{S} \;\triangleright\; \Gamma \vdash A; \Gamma, A \vdash C; \mathcal{S}$
    - `cut`: $\Gamma \vdash C; \mathcal{S} \;\triangleright\; \Gamma \vdash A \to C; \Gamma \vdash A; \mathcal{S}$
- Discarding tactics:
    - `apply`: $\Gamma, \forall x A \vdash A[x := t]; \mathcal{S} \;\triangleright\; \mathcal{S}$
    - `trivial`: $\Gamma, A \vdash A; \mathcal{S} \;\triangleright\; \mathcal{S}$.

These tactics are classified according to the heuristic rules of the backward reasoning strategy: the first group is generated by each particular case of CA, according to the logical form of the conclusion; the second group comes from PA, in agreement with the logical form of the focused premise; the process of lemma assertion is given by the third group. Observe that both LA tactics are the same from a logical point of view, but, as we are dealing with goal sequences, they are quite different from an operational point of view. Finally, discarding tactics, that is, those whose application decreases the number of subgoals, are provided by the fourth group.

We now give a brief justification of the four tactic groups: discarding tactics correspond to proving axioms, that is, sequents given by the (Hyp) rule or sequents of the form $\Gamma, \forall x A \vdash A[x := t]$ corresponding to universal instantiation. LA tactics are justified by the so-called substitution property of the ND system: if $\Gamma, A \vdash C$ and $\Gamma \vdash A$ then $\Gamma \vdash C$ (the formula A is the lemma). LA tactics rules require the generation of a new formula (A), a difficult, if not an impossible task for theorem provers to implement the transition system. The operational mechanism of CA tactics is supported by the inversion principle[3] of the introduction rules. Finally, PA tactics are justified by the inversion[4] of the following admissible[5] inference rules which generate subgoals either by simplifying a premise or, in the case of implication, by modifying the conclusion:

$$\frac{\Gamma, A \to B \vdash A}{\Gamma, A \to B \vdash B}$$

$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C} \qquad \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C} \qquad \frac{\Gamma, A \vdash C \quad x \notin FV(\Gamma, C)}{\Gamma, \exists x A \vdash C}$$

It is now clear that the backward reasoning strategy is captured by the transition system, therefore a derivation by tactics is simply a sequence of transitions ending on the final state $\square$.

---

[3]   That is, the derivability of the conclusion implies the derivability of the premises. This property becomes a theorem when the context Γ is empty.

[4]   Again, this is a theorem for $\Gamma = \varnothing$.

[5]   That these inference rules are admissible is a direct consequence of the elimination rules (more details in the extended version of this paper).

▶ **Definition 5.** Let $\mathcal{J} =_{def} \Gamma \vdash A$ be a judgement. A derivation of $\mathcal{J}$ by tactics is a sequence of states $\mathcal{S}_1, \ldots, \mathcal{S}_k$ such that $\mathcal{S}_1 = \mathcal{J}$, for every $1 \leq i < k$, $\mathcal{S}_i \rhd \mathcal{S}_{i+1}$ and $\mathcal{S}_k = \square$. If such a derivation exists, we write $\mathcal{J} \rhd^+ \square$.

We now show a derivation by tactics of our running example.

▶ **Example 6.** Let $\Gamma = \{p \rightarrow q \vee r,\ q \rightarrow r,\ r \rightarrow s\}$. The following sequence of transitions shows that $\Gamma \vdash p \rightarrow s \rhd^+ \square$. In the right column of step $i+1$ we annotate the name of the tactic that allows for the transition from step i to step $i+1$.

1	$\Gamma \vdash p \rightarrow s$	
2	$\Gamma, p \vdash s$	intro
3	$\Gamma, p \vdash r$	apply $r \rightarrow s$
4	$\Gamma, p \vdash q \vee r$; $\Gamma, p, q \vee r \vdash r$	assert $q \vee r$
5	$\Gamma, p \vdash p$; $\Gamma, p, q \vee r \vdash r$	apply $p \rightarrow q \vee r$
6	$\Gamma, p, q \vee r \vdash r$	trivial
7	$\Gamma, p, q \vdash r$; $\Gamma, p, r \vdash r$	destruct $q \vee r$
8	$\Gamma, p, q \vdash q$; $\Gamma, p, r \vdash r$	apply $q \rightarrow r$
9	$\Gamma, p, r \vdash r$	trivial
10	$\square$	trivial

Let us now present a more elaborated example.

▶ **Example 7.** Let $\Gamma = \{(x \vee p) \wedge q \rightarrow l,\ m \vee q \rightarrow s \wedge t,\ (s \wedge t) \wedge l \rightarrow x,\ p \rightarrow q\}$. The following is a derivation by tactics of $\Gamma \vdash m \wedge p \rightarrow x$.

1	$\Gamma \vdash m \wedge p \rightarrow x$	
2	$\Gamma, m \wedge p \vdash x$	intro
3	$\Gamma, m \wedge p \vdash (s \wedge t) \wedge l$	apply $(s \wedge t) \wedge l \rightarrow x$
4	$\Gamma, m, p \vdash (s \wedge t) \wedge l$	destruct $m \wedge p$
5	$\Gamma, m, p \vdash s \wedge t$;	
	$\Gamma, m, p \vdash l$	split
6	$\Gamma, m, p \vdash m \vee q$;	
	$\Gamma, m, p \vdash l$	apply $m \vee q \rightarrow s \wedge t$
7	$\Gamma, m, p \vdash m$;	
	$\Gamma, m, p \vdash l$	left
8	$\Gamma, m, p \vdash l$;	trivial
9	$\Gamma, m, p \vdash (x \vee p) \wedge q$	apply $(x \vee p) \wedge q \rightarrow l$
10	$\Gamma, m, p \vdash x \vee p$;	
	$\Gamma, m, p \vdash q$	split
11	$\Gamma, m, p \vdash p$;	
	$\Gamma, m, p \vdash q$	right
12	$\Gamma, m, p \vdash q$	trivial
13	$\Gamma, m, p \vdash p$	apply $p \rightarrow q$
14	$\square$	trivial

In order to show how our approach works in first order (minimal) logic, here is an example.

▶ **Example 8.** We give a derivation by tactics of the sequent

$$\vdash \forall v (Pv \rightarrow Qv) \rightarrow \forall x \big(\exists y (Py \wedge Rxy) \rightarrow \exists z (Qz \wedge Rxz) \big).$$

$$
\begin{array}{lll}
1 & \vdash \forall v(Pv \to Qv) \to \forall x\big(\exists y(Py \wedge Rxy) \to \exists z(Qz \wedge Rxz)\big) & \\
2 & \forall v(Pv \to Qv) \vdash \forall x\big(\exists y(Py \wedge Rxy) \to \exists z(Qz \wedge Rxz)\big) & \text{intro} \\
3 & \forall v(Pv \to Qv) \vdash \exists y(Py \wedge Rxy) \to \exists z(Qz \wedge Rxz) & \text{intro} \\
4 & \forall v(Pv \to Qv),\ \exists y(Py \wedge Rxy) \vdash \exists z(Qz \wedge Rxz) & \text{intro} \\
5 & \forall v(Pv \to Qv),\ Py \wedge Rxy \vdash \exists z(Qz \wedge Rxz) & \text{destruct } \exists y(Py \wedge Rxy) \\
6 & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash \exists z(Qz \wedge Rxz) & \text{destruct } Py \wedge Rxy \\
7 & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash Qy \wedge Rxy & \text{exists } y \\
8 & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash Qy\ ; & \\
  & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash Rxy & \text{split} \\
9 & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash Py \to Qy\ ; & \\
  & \forall v(Pv \to Qv),\ Py,\ Rxy,\ Py \to Qy \vdash Qy\ ; & \\
  & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash Rxy & \text{assert } Py \to Qy \\
10 & \forall v(Pv \to Qv),\ Py,\ Rxy,\ Py \to Qy \vdash Qy\ ; & \\
  & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash Rxy & \text{apply } \forall v(Pv \to Qv) \\
11 & \forall v(Pv \to Qv),\ Py,\ Rxy,\ Py \to Qy \vdash Py\ ; & \\
  & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash Rxy & \text{apply } Py \to Qy \\
12 & \forall v(Pv \to Qv),\ Py,\ Rxy \vdash Rxy & \text{trivial} \\
13 & \square & \text{trivial}
\end{array}
$$

We end by stating the theorem that guarantees that the usual concept of derivation given in definition 1, coincides with our proposed notion of derivation by tactics given in definition 5. The proof of this theorem may be found in the extended version of this paper.

▶ **Theorem 9** (Equivalence of $\vdash$ and $\rhd^+$). *For any sequent $\mathcal{J}$, $\mathcal{J} \rhd^+ \square$ if and only if $\mathcal{J}$ is provable.*

3 Final Remarks

Due to lack of space we were not able to discuss Coq properly. Nevertheless, we want to mention that its underlying first order logic mechanisms can be understood as our transition system $\mathcal{T}$. Actually, the name we give to each tactic in definition 4 is the name of its corresponding Coq command (c.f. [1]). For example, a Coq proof of our running example is given in the following script:

```
Hypotheses (p q r s : Prop)
           (H1: p -> q \/ r)
           (H2: q -> r)
           (H3: r -> s).

Theorem Example1: p -> s.
Proof.
intro.
apply H3.
assert (q \/ r) as H4.
apply H1.
trivial.
destruct H4.
apply H2.
trivial.
trivial.
Qed.
```

The reader may realize that once the tactical approach is understood, this computer-assisted proof is quite clear. Therefore, our goal for a smooth migration from teaching to using logic has been accomplished. Indeed, our backward reasoning strategy (as discussed in 2.1) has been presented to students in a computational logic course, and had a very good reception. At the moment, we will use this material in two of our courses (Computational Logic and Automated Reasoning), where we will test the tactical approach as a way to introduce CoQ. The results of this experience will be reported later. The main difference between our approach and that of CoQ is that in the latter hypotheses are labeled, in order to make it possible to have further reference to them. If we use these labels as variables of a λ-calculus, as is done by CoQ, we can encode the application of every inference rule with a λ-term. From this encoding, the mechanism of derivation by tactics yields a λ-term which encodes such a proof, and we can decode this proof into a usual derivation. This is the well-known Curry-Howard correspondence. Part of our future work is to explore the benefits of this powerful result for teaching logic. With respect to classical logic, we may extend our approach by allowing the use of lemmas of the form $A \vee \neg A$ (without requiring its proof), or by adding, for instance, the tactic $\Gamma, \neg A \vdash B \ ; \ \mathcal{S} \rhd \Gamma \vdash A \ ; \ \mathcal{S}$, which corresponds to proving B by contradicting the hypothesis $\neg A$. However, the inversion principles that support our proposal are invalid in this case and therefore we cannot guarantee the feasibility of the tactical approach. A further line of research is to extend our approach to other logics. We are particularly interested in modal logic. In this respect, the second author of this paper (in [4]) has verified in CoQ the definition and properties of the ND system for modal logic developed in [6], something which allows for a straightforward definition and implementation of our tactical approach for the case of modal logic.

Acknowledgements

References

1 Bertot Y. and Castéran P. *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions.* Texts in Theoretical Computer Science. Springer 1st ed. 2004.

2 Huth M. and Ryan M. *Logic in Computer Science: Modelling and Reasoning about Systems.* Cambridge University Press. Second edition, 2004.

3 Indrzejczak. A. *Natural Deduction, Hybrid Systems and Modal Logics.* Trends in Logic. Vol. 30. Springer 2010.

4 Linares-Arévalo P. S. *Deducción Natural en Lógica Modal: una implementación en Coq.* Master's thesis, Universidad Nacional Autónoma de México. 2015.

5 Manzano M. and Huertas A. *Lógica para principiantes.* Alianza Editorial, 2004.

6 Pfenning F. and Davies R. *A Judgmental Reconstruction of Modal Logic.* Mathematical structures in computer science, vol. 11. 511- 540. 2001.

7 Pierce C.B. *Lambda, the Ultimate TA: Using a Proof Assistant to Teach Programming Language Foundations.* In Proceedings of the 14th ACM SIGPLAN International conference on Functional programming. 2009.

8 Polya G. *How to solve it. A new aspect of mathematical method.* Princeton University Press. 1945.

9 Hendriks, M., Kaliszyk, C., van Raamsdonk, F., Wiedijk, F. *Teaching Logic using a state-of-the-art Proof Assistant.* URL: `http://www.cs.vu.nl/~femke/ps/formed08.pdf`.