Philosophical Communications, Web Series, No. 35, pp. 261-274
Dept. of Philosophy, Göteborg University, Sweden
ISSN 1652-0459

Kvantifikator för en Dag

Essays dedicated to Dag Westerståhl on his sixtieth birthday

GOAL-DRIVEN REASONING IN FIRST-ORDER LOGIC

CLAES STRANNEGÅRD

Abstract. A complete, subformula-preserving proof system for goal-driven reasoning in first-order logic is presented. The proof system is a transition system and its proofs are (linear and local) computations in the transistion system. Thus the proof system fits into a standard framework for modeling problem-solving in cognitive psychology.

§1. Introduction. In [8] and [15], timed psychological experiments were performed in which subjects were asked questions of the form

(1) Is $\Gamma \models A$?

where Γ is a set of sentences of first-order logic (FO), $\models$ is the logical consequence relation, and A is a sentence of FO. Proof systems were then constructed to account for the experimental results.

On the basis of proof systems such as these, various notions of *comprehensible* proof can be defined, e. g. by using complexity measures on proofs such as number of symbols, number of steps and number of temporary assumptions.

A first potential application of comprehensible proofs is in formal verification, where the question (1) arises time and time again in the form

(2) Is $SYSTEM \models REQUIREMENT$?

where $SYSTEM$ is a FO formalization of an engineering system (such as an electronic brake system of a car) and $REQUIREMENT$ is a FO formalization of some requirement on that construction (such as $\forall t(brake(t) \rightarrow brake\_light(t+1)))$. Thus, verification problems are turned into FO decidability problems. The undecidability of FO, however, means that no general algorithm exists for answering (1).

What about (2), then? Could it be easier than the fully general question (1) in some way? Well, experience shows that it is often possible to express (2) in a decidable fragment of FO, like propositional logic or some temporal logic. Large industrial systems, including

Thanks to Thierry Coquand, Per Lindström and Lance Rips.

electronic circuits with millions of gates, have been verified in this way using SAT-solvers like Chaff [10].

What can be done, then, in cases when we cannot restrict attention to such decidable fragments of FO, or in cases when we can, but the available algorithms are too slow on the relevant instances of (2)? One approach, which is described in [16], is to turn to fragments that are originally defined in psychological, rather than mathematical, terms. In essence, the idea is to restrict attention to the relation

(3) $SYSTEM \vdash^{*} REQUIREMENT$

where $\vdash^{*}$ is the provability relation associated with a suitable notion of comprehensible proof. The rationale for this approach in the industrial context is as follows:

- For several reasons, including maintainability, the functionality of the system must be comprehensible and possible to communicate to others. In particular, (3) should hold.
- For manmade systems, the normal case should be that (3) holds, since at least the constructors themselves should be able to explain their constructions to others.
- Evidence from [8] and [15] suggests that the relation $\vdash^{*}$ is radically easier to compute than $\models$. To begin with, in line with the results of the timed experiments, $\vdash^{*}$ is finite (for a fixed finite FO language). Moreover, its finite size is bounded by all sorts of cognitive limitations. This is because the proofs that define $\vdash^{*}$ cannot be too demanding, e. g. , on the working memory, the perceptive capacity, the span of attention, or the processing time.

This suggests using a heuristic for FO automatic theorem proving, which primarily directs the search for proofs towards the comprehensible proofs. Thus one may hope to speed up verification processes and also make more verification problems solvable in practice. This approach can be considered for logics beyond FO as well, including logics whose logical consequences are not computably enumerable.

A second potential application of comprehensible proofs is to automatically generate explanations in the form of comprehensible proofs (that are perhaps translated into natural language). In this case, it is the comprehensible proofs themselves that are intended for human consumption, not just the knowledge that certain things are (comprehensibly) provable.

In [5], representing the so-called *mental models* tradition, cognitive models for example-based reasoning are constructed. This makes the underlying reasoning systems logically unsound and therefore less relevant to formal verification. In [8] and [15], which are exponents of the so-called *mental logic* tradition, cognitive models are constructed on the basis of sound systems of natural deduction (ND), as defined e. g. in [14]. From the perspective of cognitive modeling, however, the ND formalism seems to be somewhat problematic.

First, ND proofs are trees. This is a problem if logical reasoning processes are to be regarded as special cases of general reasoning processes, as modeled in cognitive psychology by computations in transition systems [13]. In transition systems, computations are linear and local sequences of information states. ND proofs, on the other hand, are neither linear (since they are tree-shaped), nor local (since assumptions high up in the trees can be cancelled much further down).

Second, ND does not explicitly support goal-driven reasoning. In fact, the rules of ND are operators on proofs rather than proof-goals. For instance, consider →-introduction:

$$\dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \to B}\ \to\text{-introduction}$$

This rule can be regarded as an operator which transforms proofs of B into proofs of $A \to B$. In practice, proofs in ND are often constructed in a goal-driven fashion, as explained in [21]. Then informal rules that operate on proof-goals can be applied. One such rule replaces the proof-goal "show $A \to B$" by the proof-goal "assuming A, show B". In ND, however, such informal rules, or proof strategies, have no formal counterparts. Cognitive models, on the other hand, frequently support goal-driven reasoning, cf. [15] and [8]. This is in line with the view that it is generally unrealistic to assume that people tackle logic problems like (1) by working forward from the premises only.

Similar problems related to psychological realism pertain to formalisms like Hilbert systems [20], sequent calculi [3], and semantic tableaux [1]. This is not so surprising, however, since none of these formalisms were conceived for the purpose of cognitive modeling. In the next section, a formalism which is intended for cognitive modeling is presented.

§2. System. We consider first-order formulas in a language whose logical symbols are $\bot$, $\neg$, $\wedge$, $\forall$, and whose non-logical symbols are propositional variables, individual constants, and relation symbols. Capital Latin letters are used for formulas and capital Greek letters for sets of formulas.

DEFINITION 1. A *task* is an expression of the form $[\Gamma \vdash A]$. A task $[\Gamma \vdash A]$ is *sound* if $\Gamma \models A$. A *state* is a set of tasks.

Remembering that we are dealing with sets rather than sequences, we shall often write tasks and states in a simplified notation, with certain commas and braces omitted.

DEFINITION 2. The system S consists of the rules of Table 1. In Reductio ad absurdum, A must be different from $\bot$, in Case split it is required that $\neg(B \wedge C) \in \Gamma$ and in Naming, the constant c must be fresh in the sense that it does not appear to the left of the symbol $\rhd$.

$[\Gamma, A \vdash A] \rhd \emptyset$	Hypothesis
$[\Gamma, B, \neg B \vdash A] \rhd \emptyset$	Ex falso quodlibet
$[\Gamma \vdash A] \rhd [\Gamma, \neg A \vdash \bot]$	Reductio ad absurdum
$[\Gamma, \neg\neg B \vdash A] \rhd [\Gamma, \neg\neg B, B, \vdash A]$	Double negation
$[\Gamma, B \wedge C \vdash A] \rhd [\Gamma, B \wedge C, B, C \vdash A]$	Conjunction split
$[\Gamma \vdash A] \rhd [\Gamma, \neg B \vdash A][\Gamma, \neg C \vdash A]$	Case split
$[\Gamma, \forall x B(x) \vdash A] \rhd [\Gamma, \forall x B(x), B(c) \vdash A]$	Instantiation
$[\Gamma, \neg\forall x B(x) \vdash A] \rhd [\Gamma, \neg\forall x B(x), \neg B(c) \vdash A]$	Naming

TABLE 1. The rules of S.

DEFINITION 3. A *proof* in S is a $\rhd$-chain of states. The symbol $\blacktriangleright$ denotes the reflexive, transistive closure of $\rhd$. Such $\blacktriangleright$-chains of states will sometimes be written in vertical style, with annotated horizontal bars replacing $\blacktriangleright$.

The intended interpretations of the notions just introduced are as follows. $[\Gamma \vdash A]$ is the task of showing that $\Gamma \models A$. A state is a set of such tasks. $s \rhd s'$ means that to solve the tasks of s, it is sufficient to solve all the tasks of s'. In particular, if $s \rhd \emptyset$, then the tasks of s can be regarded as solved.

Note that S is a transition system [13] and the proofs of S are computations in this system. Thus, logical reasoning and logical reasoning processes are modeled as special cases in a standard framework for modeling problem solving and problem solving processes in general. This makes it relatively straightforward to model provability with bounded cognitive resources. The idea is to define local complexity measures on states and rules and then combine these local measures into global complexity measures defined on proofs. For instance, working memory, which is one of the well-known bottlenecks in problem solving [9], can be directly modeled as a complexity measure on states. This measure can then be extended to proofs by taking the maximum over all the states. Using such a measure, all proofs that require too much working memory, e. g. for holding temporary assumptions, can be excluded.

§3. Soundness and completeness.

THEOREM 3.1 (Soundness of S). *Suppose* $[\Gamma \vdash A] \blacktriangleright \emptyset$. *Then* $\Gamma \models A$.

PROOF. Let π be a proof in S that ends with $\emptyset$. We show by (backward) induction that no state in π contains an unsound task.

Base step. The last state in π is empty so it contains no unsound task.

Induction step. Suppose that $s \rhd s'$ is a step in π, where s' contains no unsound task. We show that s also contains no unsound task by showing that all the rules preserve soundness of tasks when read backwards. This is easy to see for all rules except Naming. In the case of Naming we must show that

(4) $\Gamma, \neg\forall x B(x), \neg B(c) \models A \Rightarrow \Gamma, \neg\forall x B(x) \models A.$

Let L be the language of $\Gamma + \neg\forall x B(x) + A$. By the restriction on Naming, we may assume that c does not appear in L. Also, let M be any L-model such that $M \models \Gamma, \neg\forall x B(x)$. Then there is an element $a \in M$ which satisfies $\neg B(x)$ in M. Now let us define the $L \cup \{c\}$-model M', which is exactly like M except that it interprets c as a. Then clearly $M' \models \Gamma, \neg\forall x B(x), \neg B(c)$. Hence, by assumption, $M' \models A$. But A does not contain c, so it follows that $M \models A$, as desired. Thus (4) follows. ⊣

To prove completeness of S, we will make use of ideas from [1], [4], [6], and [7].

DEFINITION 4. (Φ, Ψ) is *closed* if the following conditions hold:

1. $\neg B \in \Phi \Rightarrow B \in \Psi$
2. $\neg B \in \Psi \Rightarrow B \in \Phi$

3. $B \wedge C \in \Phi \Rightarrow B \in \Phi$ and $C \in \Phi$

4. $B \wedge C \in \Psi \Rightarrow B \in \Psi$ or $C \in \Psi$

5. $\forall x B(x) \in \Phi \Rightarrow B(c) \in \Phi$, for all constants c

6. $\forall x B(x) \in \Psi \Rightarrow B(c) \in \Psi$, for some constant c.

Definition 5. (Φ, Ψ) is *separated* if $[\Phi, \neg\Psi \vdash \bot] \not\blacktriangleright \emptyset$.

Definition 6. The *canonical model* M of (Φ, Ψ) is defined as follows.

- The domain of M is the set of constants that appear in $\Phi \cup \Psi$ (plus any constant if needed to ensure that the domain is non-empty).
- Let $R(x_1, \dots, x_n)$ be a relation symbol. The interpretation of R in M is the set $\{(c_1, \dots, c_n) : R(c_1, \dots, c_n) \in \Phi\}$.
- Let p be a propositional variable. The interpretation of p in M is "true" if $p \in \Phi$ and "false" otherwise.

Lemma 3.2. Let (Φ, Ψ) be a separated closed pair and let M be the canonical model of (Φ, Ψ). Then $M \models \Phi$ and $M \models \neg\Psi$.

Proof. We use induction on formulas A to show that

(5) $A \in \Phi \Rightarrow M \models A$,

(6) $A \in \Psi \Rightarrow M \not\models A$.

Base case.

Case $A = p$.

Suppose $p \in \Phi$. Then $M \models p$ by the definition of M. Hence (5) holds.

Now suppose $p \in \Psi$. Assuming that $p \in \Phi$ too, we would have the following proof:

$$\frac{[\Phi, \neg\Psi \vdash \bot]}{\emptyset}\text{ Ex falso quodlibet (on } p \in \Phi \text{ and } \neg p \in \neg\Psi)$$

This would contradict the assumption that (Φ, Ψ) is separated. Thus, $p \notin \Phi$ and by the definition of M, $M \not\models p$. Hence (6) holds.

Case $A = \bot$.

Suppose $\bot \in \Phi$. Then we would have the following proof:

$$\frac{[\Phi, \neg\Psi \vdash \bot]}{\emptyset}\text{ Hypothesis (on } \bot \in \Phi)$$

This contradicts the assumption that (Φ, Ψ) is separated. Thus $\bot \notin \Phi$. Hence (5) holds.

Now suppose $\bot \in \Psi$. By the semantics of $\bot$ we have $M \not\models \bot$. Hence (6) holds.

Induction case.

Case $A = \neg B$.

To show (5), note that $A \in \Phi \Rightarrow \neg B \in \Phi \Rightarrow B \in \Psi$ (by Condition 1) $\Rightarrow M \not\models B$ (by induction) $\Rightarrow M \models A$.

To show (6), note that $A \in \Psi \Rightarrow \neg B \in \Psi \Rightarrow B \in \Phi$ (by Condition 2) $\Rightarrow M \models B$ (by induction) $\Rightarrow M \not\models A$.

Case $A = B \wedge C$.

To show (5), note that $A \in \Phi \Rightarrow B \wedge C \in \Phi \Rightarrow B, C \in \Phi$ (by Condition 3) $\Rightarrow M \models B$ and $M \models C$ (by induction) $\Rightarrow M \models A$.

To show (6), note that $A \in \Psi \Rightarrow B \wedge C \in \Psi \Rightarrow B \in \Psi$ or $C \in \Psi$ (by Condition 4) $\Rightarrow M \not\models B$ or $M \not\models C$ (by induction) $\Rightarrow M \not\models A$.

Case $A = \forall x B(x)$.

To show (5), note that $A \in \Phi \Rightarrow \forall x B(x) \in \Phi \Rightarrow B(c) \in \Phi$, for every c (by Condition 5) $\Rightarrow M \models B(c)$, for every c (by induction) $\Rightarrow M \models A$.

To show (6), note that $A \in \Psi \Rightarrow \forall x B(x) \in \Psi \Rightarrow B(c) \in \Psi$, for some c (by Condition 6) $\Rightarrow M \not\models B(c)$ (by induction) $\Rightarrow M \not\models A$.

$\dashv$

THEOREM 3.3 (Completeness of S). *Suppose* $[\Gamma \vdash A] \not\blacktriangleright \emptyset$. *Then* $\Gamma \not\models A$.

PROOF. Suppose that $[\Gamma \vdash A] \not\blacktriangleright \emptyset$. We will define a model witnessing that $\Gamma \not\models A$. To that end we will define a closed, separated pair (Σ, Δ) such that $\Gamma \subseteq \Sigma$ and $A \in \Delta$, and then take the model to be the canonical model of (Σ, Δ).

Let $A_1, A_2, \ldots$ be an enumeration of all formulas in the language of Γ, A, in which each formula appears infinitely often. We also assume some well-ordering of the constants of this language (which enables us to pick the minimal element of any non-empty set of constants). Let (Σ_n, Δ_n) be defined as in Figure 1.

LEMMA 3.4. Each (Σ_n, Δ_n) is separated.

PROOF. Induction on n.

Stage 0. Let $(\Sigma_0, \Delta_0) = (\Gamma, \{A\})$.

Stage $n+1$. Apply the first applicable condition among the following:

1. Case $A_n = \neg B \in \Sigma_n$ and $B \notin \Delta_n$.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n, \Delta_n + B)$.
2. Case $A_n = \neg B \in \Delta_n$, and $B \notin \Sigma_n$.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n + B, \Delta_n)$.
3. Case $A_n = B \wedge C \in \Sigma_n$ and $(B \notin \Sigma_n$ or $C \notin \Sigma_n)$.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n + B + C, \Delta_n)$.
4. Case $A_n = B \wedge C \in \Delta_n, B \notin \Delta_n$ and $C \notin \Delta_n$.

   (a) Subcase $(\Sigma_n, \Delta_n + B)$ is separated.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n, \Delta_n + B)$.

   (b) Subcase $(\Sigma_n, \Delta_n + C)$ is separated.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n, \Delta_n + C)$.

   (c) Otherwise.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n, \Delta_n)$.
5. Case $A_n = \forall x B(x) \in \Sigma_n$ and $B(c) \notin \Sigma_n$ for some (minimal) c.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n + B(c), \Delta_n)$.
6. Case $A_n = \forall x B(x) \in \Delta_n$ and $B(c) \notin \Delta_n$ for every c.

   (a) Subcase $(\Sigma_n, \Delta_n + B(c))$ is separated for some (minimal) c.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n, \Delta_n + B(c))$.

   (b) Otherwise.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n, \Delta_n)$.
7. Case otherwise.

   Let $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n, \Delta_n)$.

FIGURE 1. Definition of the sequence (Σ_n, Δ_n).

Base case. Suppose (Σ_0, Δ_0) is not separated. Since $(\Sigma_0, \Delta_0) = (\Gamma, \{A\})$, this means that $[\Gamma, \neg A \vdash \bot] \blacktriangleright \emptyset$. Then we have the following proof:

$$\dfrac{\dfrac{[\Gamma \vdash A]}{[\Gamma, \neg A \vdash \bot]}\,\text{Reductio ad absurdum}}{\emptyset}\,\text{Assumption}$$

This contradicts our assumption that $[\Gamma \vdash A] \not\blacktriangleright \emptyset$.

Induction case. Our induction hypothesis is that (Σ_n, Δ_n) is separated, i. e.

(7) $[\Sigma_n, \neg\Delta_n \vdash \bot] \not\blacktriangleright \emptyset$.

We want to show that $(\Sigma_{n+1}, \Delta_{n+1})$ is also separated, i. e.

(8) $[\Sigma_{n+1}, \neg\Delta_{n+1} \vdash \bot] \not\blacktriangleright \emptyset$.

We will show that (8) holds no matter what case in the definition of $(\Sigma_{n+1}, \Delta_{n+1})$ in Figure 1 applies at $n+1$. By the induction hypothesis this is automatic for the cases 4, 6 (which are defined in terms of separability) and 7. Hence we only need to consider the cases 1, 2, 3 and 5.

Case 1. Then $A_n = \neg B \in \Sigma_n$ and $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n, \Delta_n + B)$. Thus we need to show

$$[\Sigma_n, \neg\Delta_n, \neg B \vdash \bot] \not\blacktriangleright \emptyset.$$

But since $\neg B \in \Sigma_n$, this follows from the induction hypothesis.

Case 2. Then $A_n = \neg B \in \Delta_n$ and $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n + B, \Delta_n)$. Thus we need to show

$$[\Sigma_n, B, \neg\Delta_n \vdash \bot] \not\blacktriangleright \emptyset.$$

Assume the contrary. Then we have the following proof, contradicting the induction hypothesis:

$$\cfrac{\cfrac{[\Sigma_n, \neg\Delta_n \vdash \bot]}{[\Sigma_n, B, \neg\Delta_n \vdash \bot]}\,\text{Double negation (on } \neg\neg B \in \neg\Delta_n\text{)}}{\emptyset}\,\text{Assumption}$$

Case 3. Then $A_n = B \wedge C \in \Sigma_n$ and $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n + B + C, \Delta_n)$. Thus we need to show

$$[\Sigma_n, B, C, \neg\Delta_n \vdash \bot] \not\blacktriangleright \emptyset.$$

Assume the contrary. Then we have the following proof, contradicting the induction hypothesis:

$$\cfrac{\cfrac{[\Sigma_n, \neg\Delta_n \vdash \bot]}{[\Sigma_n, B, C, \neg\Delta_n \vdash \bot]}\,\text{Conjunction split (on } B \wedge C \in \Sigma_n\text{)}}{\emptyset}\,\text{Assumption}$$

Case 5. Then $A_n = \forall x B(x) \in \Sigma_n$ and $(\Sigma_{n+1}, \Delta_{n+1}) = (\Sigma_n + B(c), \Delta_n)$, for some c. Thus we need to show that

$$[\Sigma_n, B(c), \neg\Delta_n \vdash \bot] \not\blacktriangleright \emptyset.$$

Assume the contrary. Then we have the following proof, contradicting the induction hypothesis:

$$\begin{array}{cl} [\Sigma_n, \neg\Delta_n \vdash \bot] & \\ \hline [\Sigma_n, B(c), \neg\Delta_n \vdash \bot] & \text{Instantiation (on } \forall x B(x) \in \Sigma_n) \\ \hline \emptyset & \text{Assumption} \end{array}$$

$\dashv$

Now let $\Sigma = \cup\Sigma_n$ and $\Delta = \cup\Delta_n$.

LEMMA 3.5. (Σ, Δ) is separated.

PROOF. Suppose (Σ, Δ) is not separated. Then $[\Sigma, \neg\Delta \vdash \bot] \blacktriangleright \emptyset$. Since all the rules only involve a finite number of formulas, this implies that for some n, $[\Sigma_n, \neg\Delta_n \vdash \bot] \blacktriangleright \emptyset$. This contradicts Lemma 3.4. $\dashv$

LEMMA 3.6. (Σ, Δ) is closed.

PROOF. Note that all six conditions of Definition 4 except Condition 5 can be ensured by inserting one or two suitable formulas into Φ or Ψ. Condition 5, on the other hand, may require inserting infinitely many formulas (on the form $B(c)$ for different c). Each case of the construction in Figure 1 was designed specifically to take care of the Condition in Definition 4 with the corresponding number.

Conditions 1, 2 and 3 are clearly met by (Σ, Δ). Condition 5 is also met by (Σ, Δ), since for each formula $\forall x B(x)$, there are infinitely many natural numbers n such that $A_n = \forall x B(x)$.

Conditions 4 and 6 would clearly be met too if we knew that cases 4c and 6b never happened in the construction. Thus, we only need to show this to conclude that (Σ, Δ) is closed.

CLAIM 3.7. Case 4c never happens.

PROOF. Suppose Case 4c happens at stage $n+1$. Then $A_n = B \wedge C \in \Delta_n$. Furthermore, neither $(\Sigma_n, \Delta_n + B)$ nor $(\Sigma_n, \Delta_n + C)$ is separated. Thus we have

(9) $[\Sigma_n, \neg\Delta_n, \neg B \vdash \bot] \blacktriangleright \emptyset$,

(10) $[\Sigma_n, \neg\Delta_n, \neg C \vdash \bot] \blacktriangleright \emptyset$.

Hence we have the following proof:

$$\cfrac{\cfrac{\cfrac{[\Sigma_n, \neg\Delta_n \vdash \bot]}{[\Sigma_n, \neg\Delta_n, \neg B \vdash \bot][\Sigma_n, \neg\Delta_n, \neg C \vdash \bot]}\ \text{Case split (on } \neg(B \wedge C) \in \neg\Delta_n)}{[\Sigma_n, \neg\Delta_n, \neg C \vdash \bot]}\ \text{By (9)}}{\emptyset}\ \text{By (10)}$$

Thus (Σ_n, Δ_n) is not separated. This contradicts Lemma 3.4. $\dashv$

Claim 3.8. Case 6b never happens.

Proof. Suppose Case 6b happens at stage $n + 1$. Then $A_n = \forall x B(x) \in \Delta_n$. Furthermore, $(\Sigma_n, \Delta_n + B(c))$ is not separated for any c. Thus for every c,

(11)

$$[\Sigma_n, \neg\Delta_n, \neg B(c) \vdash \bot] \blacktriangleright \emptyset.$$

Hence we have the following proof for any constant c' not in $[\Sigma_n, \neg\Delta_n \vdash \bot]$:

$$\cfrac{\cfrac{[\Sigma_n, \neg\Delta_n \vdash \bot]}{[\Sigma_n, \neg\Delta_n, \neg B(c') \vdash \bot]}\ \text{Naming (on } \neg\forall x B(x) \in \neg\Delta_n)}{\emptyset}\ \text{By (11)}$$

Thus (Σ_n, Δ_n) is not separated. This contradicts Lemma 3.4. $\dashv$

Hence (Σ, Δ) is closed. $\dashv$

Now we can finish the proof of Theorem 3.3. By Lemma 3.6, (Σ, Δ) is closed. (Σ, Δ) is also separated by Lemma 3.4 and hence by Lemma 3.2, the canonical model M of (Σ, Δ) is such that $M \models \Sigma$ and $M \models \neg\Delta$. In particular $M \models \Gamma$ and $M \not\models A$. Consequently $\Gamma \not\models A$. $\dashv$

Now let us take a closer look at the rules of S. First, note that in a proof that starts with a propositional task, the rules Instantiation and Naming are never applicable. Therefore S is complete for propositional logic even when these rules are excluded.

Second, except for the rule Instantiation, applying the rules of S requires very little creativity. This is in contrast, e. g. to the rule $\bot$-elimination in natural deduction, which allows the introduction of a completely arbitrary formula, which must be selected by the proof-maker. It is also in contrast to the Axiom rule of Sequent calculus [20], which allows the introduction of an arbitrary assumption on the form $A \vdash A$.

Third, inspecting the rules of S reveals that a proof beginning with $[\Gamma \vdash A]$ can only contain subformulas and negated subformulas of formulas in $\Gamma + A$, provided that all formulas $B(c)$ are defined to be subformulas of $\forall x B(x)$. Thus, the rules of S preserve subformulas in this (restricted) sense. Therefore irrelevant formulas never appear in proofs of S. This is in contrast to natural deduction, where there are no control mechanisms against this happening (in non-normal proofs).

§4. Conclusion. The purpose of introducing S was to show that it is possible to define proof systems that are complete, subformula-preserving, goal-driven and on a standard format for cognitive modeling. One of the advantages of this format is that it supports straightforward modeling of resource-bounded reasoning. This may in turn be of interest, i. a. in the context of formal verification for verifying comprehensible engineering constructions and in language technology for generating comprehensible explanations. Although fundamentally different, the system S bears certain similarities to the system G3c of [11], especially as implemented in the proof-editor PESCA [12]. The system S is also reminiscent of the system of [17], whose states, however, contain objects of two kinds, called facts and goals.

To develop specific proof systems for cognitive modeling one must first specify the target of the model. For instance, the target may be to model those questions (1) that a certain mathematician can answer correctly when the time-limit is 30 seconds and the mathematician has access to pencil and paper. A system like S may then need to be modified in several ways. The language may need to be extended with more connectives, quantifiers, function symbols, and the equality sign. The set of rules may also need to be modified and extended. In principle, each deduction rule that the mathematician might use should be included in the system. This generally makes the set of rules redundant from the provability perspective, but not from the resource-bounded provability perspective.

With such a proof system at hand, one may proceed to define the relation $\vdash^*$ in an iterative process of defining complexity measures, setting complexity bounds, adjusting the proof system, and making statistical evaluations. The goal is to refine the model in this way, until a statistically satisfactory level of accuracy is reached.

In [16], it is shown how cognitive models can be constructed on the basis of transition systems and complexity measures on computations. Case studies in which cognitive models were (successfully) constructed in this way include mental addition [18], color sequence recollection [2], and trading games [19].

Claes Strannegård
IT University of Göteborg
402 75 Göteborg
Sweden
claes.strannegard@ituniv.se

REFERENCES

[1] Evert W. Beth, *Semantic entailment and formal derivability*, ***Koninklijke Nederlandse Akademie van Wentenschappen, Proceedings of the Section of Sciences***, vol. 18 (1955), pp. 309–342.

[2] Kristian Freed and Hampus Ram, *Finding patterns in series: measuring complexity of human recollection*, ***Master's thesis***, Chalmers University of Technology, 2005.

[3] Gerhard Gentzen, *Untersuchungen über das Logische Schliessen*, ***Mathematische Zeitschrift***, vol. 39 (1935), pp. 176–210, 405–431.

[4] Jaako Hintikka, *Form and content in quantification theory*, ***Acta Philosohica Fennica***, vol. 8 (1955), pp. 7–55.

[5] Philip N. Johnson-Laird, ***Mental models***, Harvard University Press, 1983.

[6] Stig Kanger, ***Provability in logic***, Almqvist & Wiksell, Stockholm, 1957.

[7] Per Lindström, ***First-order logic***, Department of Philosophy, Göteborg University, 2004.

[8] David P. O'Brien Martin D. S. Braine, ***Mental logic***, L. Erlbaum Associates, 1998.

[9] George A. Miller, *The magical number seven, plus or minus two: Some limtis on our capacity for processing information*, ***Psychological Review***, vol. 63 (1956), pp. 81–97.

[10] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik, *Chaff: Engineering an Efficient SAT Solver*, ***Proceedings of the 38th Design Automation Conference***, 2001.

[11] Sara Negri and Jan von Plato, ***Structural proof theory***, Cambridge University Press, 2001.

[12] ———, ***Structural proof theory***, Cambridge University Press, 2001, Appendix C "PESCA—A Proof Editor for Sequent Calculus" by Aarne Ranta.

[13] Gordon D. Plotkin, *A Structural Approach to Operational Semantics*, ***Technical Report DAIMI FN-19***, University of Aarhus, 1981.

[14] Dag Prawitz, ***Natural deduction: A proof-theoretical study***, Almquist & Wiksell, 1965.

[15] Lance Rips, ***The psychology of proof***, Bradford, 1996.

[16] Claes Strannegård, *Anthropomorphic artificial intelligence*, ***Kapten Mnemos kolumbarium*** (F. Larsson, editor), Philosophical Communications, vol. 32, Department of Philosophy, Göteborg University, 2005.

[17] ———, *A proof system for modeling reasoning processes in propositional logic*, ***The Bulletin of Symbolic Logic***, (2006), Extended abstract accepted for publication.

[18] Claes Strannegård, Kristofer Sundén Ringnér, and John Hughes, *Mental addition: a case study in anthropomorphic artificial intelligence*, Manuscript in preparation. Chalmers University of Technology, 2005.

[19] Johan Tavelin, *A cognitive model for transistion system producibility*, Master's Thesis in preparation. Chalmers University of Technology, 2005.

[20] Anne S. Troelstra and Helmut Schwichtenberg, ***Basic proof theory***, Cambridge University Press, 1996.

[21] Anne S. Troelstra and Dirk van Dalen, ***Constructivism in mathematics, Vol 1***, Studies in Logic and the Foundations of Mathematics, vol. 121, North Holland, Amsterdam, 1988.