A természetes levezetési kalkulus Gentzen-stílusú kiterjesztése - az informatikai felsőoktatáshoz igazítva

A GENTZEN-STYLE EXTENSION OF NATURAL DEDUCTION - ADJUSTED FOR UNDERGRADUATE INFORMATICS STUDIES

Mecsei Zoltán, mecseiz@inf.unideb.hu
Vályi Sándor, valyis@inf.unideb.hu
Debreceni Egyetem Informatikai Kar Számítógéptudományi Tanszék

1. Introduction

Artificial intelligence(AI) is one of the most expandig domains of informatics. Its basic undergraduate study is a common feature of the informatics oriented majors of several Hungarian and European universities. An approved –not only theoretical– approach to AI is automatic reasoning and theorem proving. [1] [2] [3] The current methods of automatic theorem proving –such as resolution and Gentzen-style sequent calculus– are essential in implementing of AI systems based on automatic reasoning. Before learning automated methods, the students should explore a manual proof searching formalism.

The only logical deduction step of resolution is not similar to ones of natural reasoning. The rise of natural deduction system (NDS) is concerned with this artifact. NDS is a proof description formalism [1] [4] [5] whose logical deduction steps and constructions coincide with those of human mathematical reasoning. There are two main differences to its historical precedents, Hilbert-style axiomatic theorem proving systems. In NDS, one administrate not only the goal of the actual stage of proofs, but, in every moment, the actual hypotheses are stored. This is evident in deductions steps containing –for example- ”reductio ad absurdum”or ”proof by cases”, where we change our assumptions dynamically. The second difference is that its deduction rules are all natural for human reasoning.

The educational practice of NDS gives a strong basis for the development of logical skills. For students of mathematics, executing exercises in NDS -even for not purely logical tasks but for real deductions of some mathematical theory as Peano-arithmetics or axiomatic set theory- results in the development of their logical skills and their better understanding of the foundations of mathematics.

The second result cannot be the main goal of teaching formal deduction to infomatics students. It is closer to the real demands of this profession when we establish a solid base of right and wrong conclusions –both in purely logical conclusions and conclusions valid in a special interpretation or interpretation class. We do not argue against teaching mathematical logic for students of informatics. It is a standard base for theoretical computer science (algorithm theory, semantics of programming languages and almost all other areas) . We should only adjust the presented proof formalization system. Original NDS is sharpened to

develop mathematical-logical skills of pure logical deductions. It is short for usual deduction steps of real reasoning. In our opinion, for education of informatics students, a richer set of allowed deduction steps in exercises can be more successful.

In search of solution of this problem, our attention could turn to Gentzen-style deduction calculus(GC). [1] It has the advantage that it provides a semi-automatic decision procedure for propositional tasks. Its rules have the same natural taste as NDS except for one thing: its basic elements are sequents which can have any natural number of actual goals of the current stage of our proof. This fact has good reasons but does not coincide with our intuitive notion of reasoning. This means for us, that GC cannot be called *natural* deduction system generally. The proofs of GC are not for human reading. It can disturb informatics students in learning proof construction.

Our proposal is to use a mixed deduction system –unifying advantages of both NDS and GC without the mentioned disadvantages– in exercises for informatics students. Basically, we propose to extend NDS by some new rules borrowed from or motivated by GC. It can be the object of future examinations to determine efficiency of the proposed system.

In the second chapter we present our proposal and in the third we give two intuitive examples of the advantages of our system.

2. The proposed deduction system

The original NDS consists of the following objects. (Γ, Δ denote multisets of formulae while A, B and C denote formulae. Sequents are pairs (Γ, A) where Γ is a multiset of formulae and A is a formula, written in form of $\Gamma \rightarrow A$.) [1][4]

Axioms are the following sequents: $\Gamma, A \rightarrow A$ for arbitrary Γ and A.

Deduction rules are

Introduction rules	Elimination rules
$\dfrac{\Gamma \rightarrow A;\ \Gamma \rightarrow B}{\Gamma \rightarrow A \wedge B}$	$\dfrac{\Gamma, A, B \rightarrow C}{\Gamma, A \wedge B \rightarrow C}$
$\dfrac{\Gamma \rightarrow A}{\Gamma \rightarrow A \vee B}$ $\dfrac{\Gamma \rightarrow B}{\Gamma \rightarrow A \vee B}$	$\dfrac{\Gamma, A \rightarrow C;\ \Gamma, B \rightarrow C}{\Gamma, A \vee B \rightarrow C}$
$\dfrac{\Gamma, A \rightarrow B}{\Gamma \rightarrow A \supset B}$	$\dfrac{\Gamma \rightarrow A;\ \Gamma \rightarrow A \supset B}{\Gamma \rightarrow B}$
$\dfrac{\Gamma, A \rightarrow B;\ \Gamma, A \rightarrow \neg B}{\Gamma \rightarrow \neg A}$	$\dfrac{\Gamma \rightarrow \neg\neg A}{\Gamma \rightarrow A}$

$\underline{\Gamma \to A}$ $\qquad\qquad$ $\underline{\Gamma \to \forall xA}$

$\Gamma \to \forall xA$ $\qquad\qquad$ $\Gamma \to A(x||t)$

(x is free neither in A nor in Γ) $\quad$ (t is a term)

$\underline{\Gamma \to A(x||t)}$ $\qquad\qquad$ $\underline{\Gamma, A \to B}$

$\Gamma \to \exists xA$ $\qquad\qquad$ $\Gamma, \exists xA \to B$

(t is a term) $\qquad\qquad$ (x is free neither in Γ, $\exists xA$ nor in B)

and in addition, structural rules expressing the facts that

(i) any hypothesis can be used more than once

(ii) the ordering of hypotheses does not count

(iii) we can use extra hypotheses

(iv) cut, that is

$\underline{\Gamma \to A; \quad \Delta, A \to B}$

$\Gamma, \Delta \to B$

We propose some new rules motivated by Gentzen calculus. This set consists of deduction steps needed to formalize everyday reasoning. The user of this proposal can pick a subset which is appropriately tailored to its need. Or we can select a minimal set of deduction rules only used for a maximum level of score and the more rules are used in the exercise the less is the achieved knowledge performance.

1. a reversible and more comfortable $\supset$-elimination

$\underline{\Gamma, \neg A \to C; \ \Gamma, B \to C}$

$\Gamma, A \supset B \to C$

2. contrapositions of any kind

$\underline{\Gamma, \neg A \to B}$ $\qquad$ $\underline{\Gamma, \neg A \to \neg B}$ $\qquad$ $\underline{\Gamma, A \to \neg B}$ $\qquad$ $\underline{\Gamma, B \to A}$

$\Gamma, \ B \to \ \neg A$ $\qquad$ $\Gamma, \ B \to \ A$ $\qquad$ $\Gamma, B \to \neg A$ $\qquad$ $\Gamma, \neg A \to \neg B$

3. $\neg\neg$-introduction

$\underline{\Gamma \to A}$

$\Gamma \to \neg\neg A$

4. reversible $\vee$-introductions from any side

$\underline{\Gamma, \neg A \to B}$ $\qquad\qquad$ $\underline{\Gamma, \neg B \to A}$

$\Gamma \to A \vee B$ $\qquad\qquad$ $\Gamma \to A \vee B$

5. chain reasoning

$$\frac{\Gamma, A \to B;\ \Gamma, B \to C}{\Gamma, A \to C}$$

6. special, more recognizable case of cut

$$\frac{\Gamma, B \vee \neg B \to A}{\Gamma \to A}$$

7. strong rules of substitution of propositional equivalents

$$\frac{\Gamma, A \to B}{\Gamma, A \to C} \qquad \frac{\Gamma, B \to A}{\Gamma, C \to A}$$

(where B and C are propositionally equivalent, justified say by Quine-tables or any learned method)

All these rules are available in ordinary NDS. (This can be justified by two or three lined proofs.) They can be viewed as lemmata, co-routines of proof constructing in NDS. So the correctness and completeness of the proposed system is an obvious corollary of correctness and completeness of NDS. Using these rules, an informatics student need not waste time and effort with easy logical consequences while proving interesting real-world proof searching and constructing tasks in first-order logic and even more sophisticated theories.

3. Demonstration

Let us consider some examples using our new rules.

$$\dfrac{\dfrac{\dfrac{\dfrac{A, \neg B \to A}{\neg A, A \to B}}{\neg A \to A \supset B}}{\neg(A \supset B) \to A} \qquad A \to A}{(A \supset B) \supset A \to A}$$

$$\to((A \supset B) \supset A) \supset A$$

This is the law of Peirce, one of hardest tasks for manual proof in NDS.

$\underline{A(x), \neg B \rightarrow A(x)}$

$\underline{\neg A(x), A(x) \rightarrow B}$

$\underline{\neg A(x) \rightarrow A(x) \supset B}$

$\underline{\neg A(x) \rightarrow \exists x(A(x) \supset B)}$

$\underline{\neg\exists x(A(x) \supset B) \rightarrow A(x)}$ $\qquad$ $\underline{B, A(x) \rightarrow B}$

$\underline{\neg\exists x(A(x) \supset B) \rightarrow \forall x A(x)}$ $\qquad$ $\underline{B \rightarrow A(x) \supset B}$

$\underline{\neg\forall x A(x) \rightarrow \exists x(A(x) \supset B)}$ $\qquad$ $\underline{B \rightarrow \exists x(A(x) \supset B)}$

$\forall x A(x) \supset B \rightarrow \exists x(A(x) \supset B)$ $\qquad$ (x is not free in $\forall x A(x)$,B)

This is one of most challenging tasks among quantified logical laws.

Summarized, while constructing proofs allowing our new rules, one can concentrate to the real properties of the model. Proof of everyday logical laws need no high effort – this may be an advantage of an proof description formalism for students of informatics.

References

[1] Pásztorné Varga K., Várterész M. *A matematikai logika alkalmazásszemléletű tárgyalása*, Panem(Budapest), 2003

[2] Fekete I., Gregorics T., Nagy S. *Bevezetés a mesterséges intelligenciába***,** LSI , 1990

[3] Russell S., Norwig P. *Artificial Intelligence,a modern approach* Prentice Hall, 2003

[4] Dragálin A., Buzási Sz. *Bevezetés a matematikai logikába*, Debrecen, 1986

[5] Troelstra, A. S., Schwichtenberg, H. *Basic Proof Theory*, Cambridge, 2000