Logic & Language from the Outside.

Raffaella Bernardi*
UiL-OTS, University of Utrecht.

Abstract

In this paper we present a method for teaching formal systems using computational linguistic tools. The aim of this method is to present logic in an appealing way for students with different skills. We use intuitive properties of natural language to teach the more esoteric logical ones. Using this approach, a better understanding of the subject is gained. In particular, by means of linguistic sings the important distinction between syntax and semantics in logical systems can be clearly shown. Furthermore, following this method we give the students a concrete application of the logical concepts different from more standard ones (e.g. mathematical language). This will help them assuming a more flexible attitude when studying logic. In the last part of the paper we describe the intuitive interface of a theorem prover which students can use to check their exercises by themselves and we explain the advantages provided by this automatic assistant.

1 Introduction

Logic is usually considered as required knowledge for students interested in mathematics or in computer science. When included in the curricula of other disciplines, it is usually taught only at an introductory level, giving an approximative presentation of propositional and first order logic. We believe logic should play an important role in other curricula too. It is, in fact, a suitable subject for teaching students how to abstract away from the specific context in which a learned concept is usually applied. Moreover, learning logic is an opportunity for students to acquire reasoning skills which find their application when reasoning in everyday life, and in other subjects as well. We aim to show a method for teaching this esoteric subject to students interested in formal reasoning as well as to students with no particular attitude for it. To reach this goal we propose to use the students' intuitions on natural language in an innovative way: natural language expressions are considered as the object on which the logic system reasons. Two different abstractions are here at work: one concerning the formal languages to which logical concepts are usually applied, and a second one regarding linguistic expressions which are treated as the object of the reasoning instead of their vehicle.

The connection between formal and natural languages calls into the mind the slogan "parsing as deduction" proposed in [8]. In the approach labeled by it, logical methods are applied to linguistic data, and programming languages are used to parse linguistic strings. For teaching purposes, the outcome of these combinations can be used in different

*Part of this work is sponsored by *Spinoza Logic in Action*, under the Spinoza Dissemination of Logic effort. Moreover, the author has been supported by the Netherlands Organization for Scientific Research (NWO, SIR-11-2266) for teaching logic at the High school of Grosseto, Italy.

directions stressing one or the other of its components. In this paper we are going to describe how and why this approach is insightful when teaching logic.

One of the most important discoveries in modern logic is the distinction between meaning and form, between the truth and provability of a formula. Teaching this difference is the main goal of introductory logic courses. Students are taught how to reason with logical language looking at the syntax of the language (proof theory), and at its semantic interpretation (model theory). However, it is difficult to find out whether this distinction is clearly understood.

When teaching propositional or first order logic, natural language is used as the object language formalized by the logic in order to get intuitions on the formalization. This can create confusion between the syntactic and semantic levels of the logical language, and proving the validity of a theorem can be mixed up with proving its truth. Natural language expressions, in fact, are the vehicle of natural reasoning which is dominated by the semantic level.

We believe that the tight 'connection' between natural language expressions and meaning can be used in a productive way when teaching the distinction between syntax and semantics in logical systems. By working out the details of the distinction where the connection is close, we help avoiding the confusion between these levels in other cases as well. The link between logic and linguistics is at the heart of Categorial Grammar frameworks as shown by Lambek [5], [7]. We propose to teach the Lambek calculus to students attending logic courses.

The Lambek calculus has interesting logical properties with different levels of difficulty. It offers quite a broad range of logical topics, from the more elementary one, such as modus ponens and hypothetical reasoning, to more complex subjects such as the Curry-Howard isomorphism, sequents calculus and proof nets. Moreover, the Lambek calculus belongs to the family of so called substructural logics which differ from each-other in their structural properties. An important point to teach in modern logic courses is that logic systems are fragments of bigger ones, and the choice of the logic we want to work with depends on the specific application we look at. The family of Lambek systems has nice properties in this respect: a systematic communication between the single systems exists as shown in [4]. Due to its resource sensitive character, each minimal change on the structural side has visible consequences on the expressiveness of the system. This can be easily checked looking at linguistic applications.

Finally, a nice consequence of using a logical inference system as parser, is that two subjects, that are traditionally kept apart, linguistics and logic, get together. This is of particular importance because it shows the richness of an interdisciplinary approach and illustrates that knowledge cannot be compartmentalized, and fixed in separated blocks. We believe that teaching a reasoning strategy in many completely diverse applications is certainly a difficult, but extremely productive goal to achieve, which goes behind the specific subject taught in the logic course.

2 When Logic is Applied to Natural Language

In this section we make things more concrete, showing the advantage of using the Lambek calculus even with respect to such a classical rule as Modus Ponens (MP). This rule is usually taught using natural language expressions in which to translate logic symbols and the schema is presented as an intuitively true inference. In spite of the fact that MP is an inference rule concerning only the syntax of the logic language and not the truth-values of the formula, the proposed translation gives intuitions using the semantic interpretation. Let $\Gamma : A$ mean that the linguistic expression Γ is represented as A in the logic language. Then for example, MP is explained using the application below.

$$\begin{array}{ccc} \text{Modus Ponens} & & \text{Application:} \\ \dfrac{A \to B \quad A}{B} & \rightsquigarrow & \dfrac{\text{If Maria swims, then Maria gets wet} : p \to q \quad \text{Maria swims} : p}{\text{Maria gets wet} : q} \end{array}$$

Using natural language in this way can be misleading. The linguistic exemplification can be easily accepted reasoning on the meaning of the propositions involved. However, this information is irrelevant when applying MP. The intuitions called upon belong to the wrong level of the language. When teaching the syntactical behavior of logical rules the students' attention should be brought to focus on the syntactic level, using their intuitions on it. This is what we do when applying logic to reason on linguistic signs. A basic concept in linguistics is that words are of different syntactic categories, and that they combine following the rules of a grammar. For example, words as proper names are considered of a category called 'noun phrase' and denoted by np; intransitive verbs are a complex category which when combined with a np yields a sentence, s. This intuitive concept is at work when applying MP to syntactic categories as shown below.

$$\frac{\text{Maria} : np \quad \text{swims} : np \to s}{\text{Maria swims} : s}$$

Once again the reasoning schema is MP, but the standard meanings of the words, *Maria* and *swims*, are not involved and the students do not pay attention to them. This example shows that changing the focus of the reasoning from the inside to the outside of natural language, can help in teaching logical properties in a precise way. The Lambek calculus permits us to realize this zooming out, and to take linguistic signs as object of the logic reasoning. In this calculus, parsing linguistic strings means to prove their grammaticality. But parsing sentences is what students do in their everyday life, proving is what they have to learn in a logic course. In addition, intuitive theorem provers have been developed in the approach of "parsing as deduction" and they can be used as automated assistants when teaching the deductive calculus.

2.1 Why to use an automated assistant

Several are the advantages provided by the use of an electronic support. First of all, as shown by [1, 3] when students can see what they are reasoning about and when proofs are given as graphs, they achieve far greater understanding of the subject than otherwise. Moreover, students have different learning rhythms. Using the assistant makes it possible for them to follow their own one, checking their mistakes and working out further solutions. Finally, as claimed in [2] when mistakes are pointed out by a machine, instead of by the teacher, students are more motivated to understand the problems. For these reasons, we believe that electronic tools are an important support in teaching.

2.2 Grail: An automated proof assistant

Grail, developed by R. Moot, is an automated prover working with proof nets, a graph based representation of proofs, and labeled deduction [6][1]. The theorem prover is imple-

[1] Grail is available under GPL at ftp://ftp.let.uu.nl/pub/users/moot

mented in SICStus Prolog, the user graphical interface in TclTk. We show some snapshots in Appendix A.

The parser implemented in Grail consists of a lexicon where the translation from natural language to formal language is established, and a set of logical rules. Using the graphical interface, students can easily practice both with (i) the grammar of the logical language, and (ii) the logical rules of the system. As regarding (i), they can learn how to construct well-formed formulas building the corresponding decomposition tree. As for (ii) several opportunities are at hand: The automated prover, can be used simply as a test-bed for students who wants to check the correctness of their proof in natural deduction format, or in an interactive way using the proof nets. In both cases the visualization and interaction with the computer is appealing for students.

3 Conclusion and Further Research

We have described a new method for teaching logic which by using intuitions on natural language properties brings students to familiarize with the enigmatic notation and principles of formal systems. Following it, students with different skills can gain a deeper understanding of logic.

Grail has already been used as an electronic support to teach Lambek calculus in different environments, from the advanced courses given at the IX and XI ESSLLI Summer Schools, to the more introductory ones taught for undergraduate students at the University of Utrecht, till the elementary lessons addressed to high school students. For all this different audience, it has proved to increase the efforts and the enthusiasm for learning. Works in progress on interactive web pages for long distance learning are carried out by Moortgat and Oehrle [9].

References

[1] J. Barwise and J. Etchmendy. Visual information and valid reasoning. In G. Allwein and J. Barwise, editors, *Logical Reasoning with Diagrams*. Oxford University Press, 1996.

[2] H. Hoover and P. Rudnicki. Teaching freshman logic with mizar-mse. In *Workshop on Teaching Logic and Reasoning in an Illogical world.*, 1996.

[3] R. Cox K. Stenning and J. Oberlander. Contrasting the cognitive effects of graphical and sentential logic teaching: reasoning, representation and individual differences. *Language and Cognitive Processes*, 10(3/4):333–354, 1995.

[4] N. Kurtonina and M. Moortgat. Structural control. In P. Blackburn and M. de Rijke, editors, *Logic, Structures and Syntax*. Dordrecht: Kluwer, 1995.

[5] J. Lambek. The Mathematics of Sentence Structure. *American Mathematical Monthly*, 65:154–170, 1958.

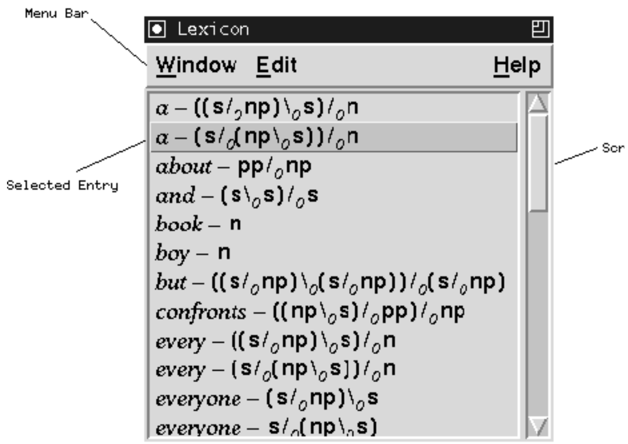
[6] R. Moot. Grail: An Automated Proof Assistant for Categorial Grammar Logics. In R. Backhouse, editor, *Proceedings of the 1998 User Interfaces for Theorem Provers Conference*, pages 120–129, 1998.

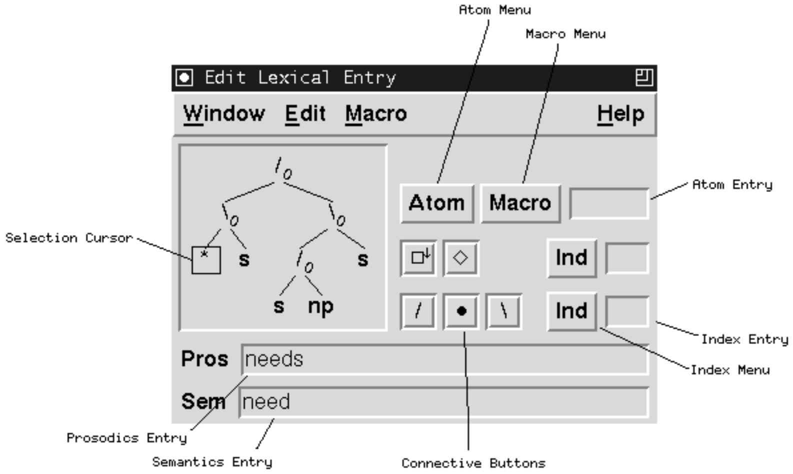
[7] M. Moortgat. Categorial Type Logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*. Elsevier, 1997.

[8] F. Pereira and D. Warren. Parsing as deduction. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*. Massachussetts Institute of Technology, 1983.

[9] M. Moortgat R. Oehrle and R. Moot (course-ware). *Grammatical reasoning: logic, structure and control.* In preparation.

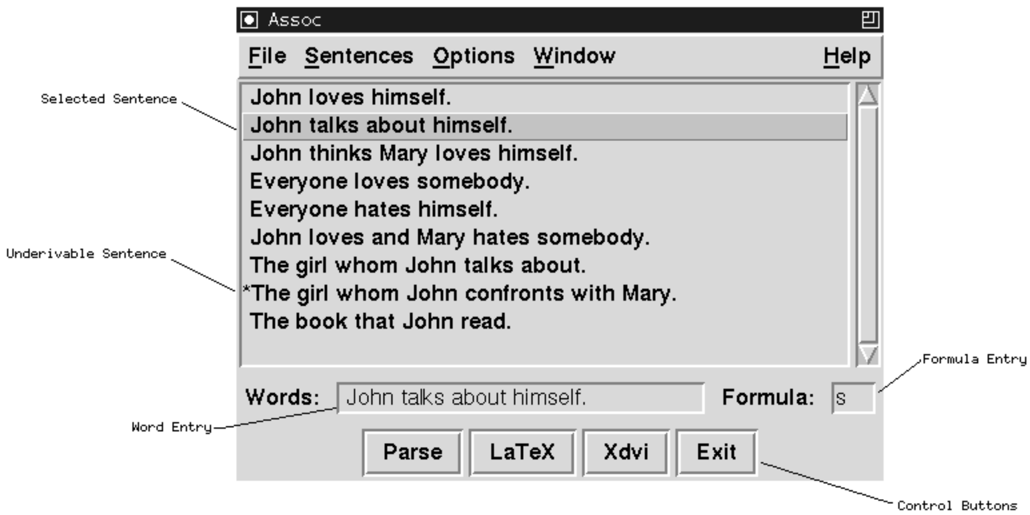
A Grail interface: some examples





The window on the left shows the lexical entries stored in the lexicon. To assign a formula to a linguistic item, the student has to be able to build the decomposition tree as show in the right window. The $\backslash_i, /_i$ are the two implication operators sensitive to the order in which the formulas occur. The index i is the *mode* which distinguish the binary operators from each other with respect of the structures they compose.

Having stored the needed information in the lexicon, the student can start practicing with the proof-system proving the grammaticality of linguistic strings.



After parsing the sentence Grail displays the proof either in natural deduction or with proof net.