Bioastronomy 2002: Life Among the Stars
IAU Symposium, Vol. 213, 2004
R.P.Norris and F.H.Stootman (eds.)

Large-Size Message Construction for ETI: Self-Interpretation in LINCOS

Alexander Ollongren

Leiden University, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

Douglas A. Vakoch

SETI Institute, 2035 Landings Drive, Mountain View CA 94043, U.S.A.

Abstract. Messages for ETI written in a terrestrial language admitting linear notation, should be supplied with extra-linguistic annotations – also linearized in some way – as an aid for interpretation by recipients. In several papers the first author has advocated the use at the second level of an abstract system (a new *Lingua Cosmica*, still under development), based on constructive logic – with a minimal set of primitives. Expressions at the meta level explain the logic contents of messages.

The interaction between any *LINCOS* and text may suffice for interpretation of both – provided the messages and annotations are of considerable size. Finding a measure for the sizes involved is, however, a non-trivial problem. As a result, one is interested in methods for interpreting a *LINCOS* without recourse to natural languages. The present paper is concerned with the problem of interpretation of the new system within itself. For that purpose, interpretation of parts of it and (complete) propositional logic in terms of each other is considered (commutativity). This leads to self-interpretation as all semantic terms reside in the closed context of the system.

1. Introduction

A message, written in some unknown symbolism, supposedly meaningful, but exempt from direct clues for its interpretation, presents formidable challenges: what language is behind it, what are the syntactic and semantic characteristics of it? An example of the kind of intricacies occurring in this respect is provided by the long standing problem of deciphering *Linear A* – even today not completely solved. The instance that there is no (natural) language behind a given message is even more complex, and can only be resolved if the message contains indirect clues (like references to e.g., pictograms) or, alternatively, elements for self-interpretation.

For the task of constructing interstellar messages for extraterrestrial intelligence, the problem can partly be circumvented by using large-size texts in some natural language for formulating the messages themselves, supplemented with a descriptive text *about* the contents – thus supplying at another level in part the

semantics of messages. However, at that level the same problem might show up. It is not *a priori* evident that a recipient will be able to understand the conventions and semantics of the system providing the annotations. It would therefore clearly be advantageous to employ a system admitting self-interpretation. The new *Lingua Cosmica* system, under development by the first author, is capable of doing this. This is because it is based on concepts of type theory, in the form originally proposed and developed by P. Martin-Löf from Sweden (based on concepts from intuitionistic logic formulated by L. Brouwer and A. Heyting in the Netherlands) and later extensively implemented in the French Coq Project at INRIA.

In the present contribution we explain how to use *LINCOS* for interpretation of expressions in the simple case of propositional calculus, more in particular the *rules of inference* (we discuss a complete set). As an illustrative example consider the *elimination rule* for the logic *conjunction*, saying that if the conjunction of X and Y is the case (the premise), then X is the case. The (compound) premise, considered to be a type, is interpreted by giving a *resident* of it. The conclusion that X is the case, is interpreted by *constructing* a resident of X. That Y is also the case under the mentioned premise, is interpreted by constructing a resident of Y. Along these lines all rules of inference are given interpretations in the closed context of *LINCOS*, i.e., the system of propositional logic is not only embedded in *LINCOS* but also interpreted in it. Thus self- interpretation is achieved. A brief description of the theoretical background for this view is contained in the paper.

2. Theory

In the present paper we consider the problem of interpretation for the classical propositional calculus. This calculus consists of logical expressions, built of elementary propositions, expressed by constants $A, B, C, \ldots$ combined with the binary connectives 'and' ($\wedge$), 'or' ($\vee$), 'implication' ($\rightarrow$), and the unary 'negation' ($\sim$), for which the usual binding rules apply. Expressions may contain sub expressions, enclosed in parenthesis. An argument, starting from suppositions leading to a conclusion, is valid if *deduction rules* have been used. These rules are axiomatic by nature, i.e., their validity within the system is assumed. They can be given an interpretation outside the system, e.g., by using truth tables. A set of deduction rules should be consistent, so that no contradictions can be derived, and complete, i.e., that every valid expression in the calculus can be derived. In order to achieve self-interpretation, this calculus including a complete set of deduction rules can be embedded in *LINCOS*, itself based on the implementation of constructive logic in the French Coq system. The paper demonstrates how this is done, and how this road leads to self-interpretation of the cosmic language.

3. Deduction Rules

By way of first example suppose that it is known that A and B is the case. By the *deduction rule* 'and elimination' it can then be concluded that A is the case. It can also be concluded that B is the case. Written formally $A \wedge B \vdash A$,

https://doi.org/10.1017/S0074180900193787 Published online by Cambridge University Press

and $A \wedge B \vdash B$. We shall begin by formulating this rule in *LINCOS*, explaining constructive aspects as they appear along the way.

There are some primitives in the language. Some basic ones are summarised as follows. The set of propositions is Prop and that A is a constant (or elementary) proposition is expressed by the *declaration* CONSTANT A : Prop. A is then a *resident* of Prop (or is *in* Prop). Compound propositions are obtained by using binary and unary connectives in the usual way. That A is the case is expressed by CONSTANT a : A, i.e., declaring that A has a resident. In the same way compound expressions can be supplied with residents, e.g., CONSTANT x : $A \wedge B$. The language admits *mappings*. Supposing that A and B are residents of Prop. Then $A \to B$, also in Prop, *couples* residents of A with those of B. If $y : A \to B, a : A$, then y *applied* to a is in B, written $(ya) : B$.

Consider now above deduction rules. That A and B exist is expressed by a declaration. That A and B is the case is expressed by the existence of a *resident* of the expression $A \wedge B$.

$$\mathbf{CONSTANT\ A,B : Prop.}$$
$$\mathbf{CONSTANT\ x : A \wedge B.}$$

Constructive logic is able to express the validity of propositional deduction rules. For the above deduction rules a resident of A and one of B needs to be *constructed* using the information that x is a resident of the compound proposition $A \wedge B$. So x must be *eliminated*. The constructions follow in a moment. First consider the element Elim_and, defined (or declared) as a *hypothesis*

$$\mathbf{HYPOTHESIS\ Elim\_and : (ALL\ X,Y,P : Prop)(X \to Y \to P) \to (X \wedge Y) \to P.}$$

This is clearly not a propositional constant – it is an expression in *LINCOS*, obeying the following

$$\mathbf{(Elim\_and\ ABA) : (A \to B \to A) \to (A \wedge B) \to A.}$$
$$\mathbf{(Elim\_and\ ABB) : (A \to B \to B) \to (A \wedge B) \to B.}$$

Suppose that residents, $x_1 : A \to B \to A$, and $x_2 : A \to B \to B$, which do not exist globally (since they have not been declared) can be constructed. Then

$$\mathbf{(Elim\_and\ ABAx_1) : (A \wedge B) \to A.}$$
$$\mathbf{(Elim\_and\ ABBx_2) : (A \wedge B) \to B.}$$

and, because $x : A \wedge B$

$$\mathbf{(Elim\_and\ ABAx_1x) : A.}$$
$$\mathbf{(Elim\_and\ ABBx_2x) : B.}$$

so that an element of A and one of B is constructed, precisely as required.

In order to construct the mentioned objects $x_1 : A \to B \to A$, and $x_2 : A \to B \to B$ one more instrument from constructive logic is needed, the so-called LAMBDA *abstraction* for creating types. An example to show how this works is as follows: let $b : B$, then [LAMBDA $h : A$]$b : A \to B$. So, generalising to two arguments

$$\mathbf{[LAMBDA\ h1 : A, h2 : B]h1 : A \to B \to A.}$$

https://doi.org/10.1017/S0074180900193787 Published online by Cambridge University Press

$$[\mathbf{LAMBDA\ h1 : A, h2 : B}]\mathbf{h2 : A \to B \to B}.$$

Summa summarum ('and elimination' rule)

$$\mathbf{HYPOTHESIS\ Elim\_and\ : (ALL\ X, Y, P : Prop)(X \to Y \to P) \to (X \wedge Y) \to P}.$$

$$\mathbf{CONSTANT\ A, B : Prop}.$$
$$\mathbf{CONSTANT\ a : A \wedge B}.$$
$(\mathbf{Elim\_and\ ABA}[\mathbf{LAMBDA\ h1 : A, h2 : B}]\mathbf{h1a}) : \mathbf{A}$ (interpretation of $A \wedge B \vdash A$).
$(\mathbf{Elim\_and\ ABB}[\mathbf{LAMBDA\ h1 : A, h2 : B}]\mathbf{h2a}) : \mathbf{B}$ (interpretation of $A \wedge B \vdash B$).

Consider next the deduction rule 'or introduction', written $A \vdash A \wedge B$, or $B \vdash A \wedge B$. Two hypotheses are needed for the verification. The complete result is

("or introduction" rule)

$$\mathbf{HYPOTHESIS\ Intro\_or1 : (AL\ X, Y : Prop)X \to X \vee Y}.$$
$$\mathbf{HYPOTHESIS\ Intro\_or2 : (ALL\ X, Y : Prop)Y \to X \vee Y}.$$
$$\mathbf{CONSTANT\ A, B : Prop}.$$
$$\mathbf{CONSTANT\ a : A}.\ \mathbf{CONSTANT\ b : B}.$$
$(\mathbf{Intro\_or1\ ABa}) : \mathbf{A \vee B}$ (interpretation of $A \vdash A \vee B$).
$(\mathbf{Intro\_or2\ ABb}) : \mathbf{A \vee B}$ (interpretation of $B \vdash A \vee B$).

The deduction rule "and introduction" written $A, B \vdash A \wedge B$, cannot be verified in *LINCOS*, because there is no means for constructing a resident of $A \wedge B$, given residents of A and B. So this rule is added as a hypothesis

("and introduction" rule)

$$\mathbf{HYPOTHESIS\ Intro\_and : (ALL\ X, Y : Prop)(X \wedge Y)}.$$
$$\mathbf{CONSTANT\ A, B : Prop}.$$
$(\mathbf{Intro\_and\ AB}) : \mathbf{A \wedge B}$ (interpretation of $A, B \vdash A \wedge B$).

An important rule is Modus Ponens (MP), which expresses: if A is the case and if A implies B, then B is the case, written $A, A \to B \vdash B$.

$$\mathbf{CONSTANT\ A, B : Prop}.$$
$$\mathbf{CONSTANT\ a : A}.\ \mathbf{CONSTANT\ x : A \to B}.$$
$(\mathbf{x\ a}) : \mathbf{B}$ (interpretation of $A, A \to B \vdash B$).

Modus Tollens (MT) expresses: if B is not the case and if A implies B, then A in not the case, written $\sim B, A \to B \vdash \sim A$. Can this be expressed constructively? The following base (also called environment) must be used:

$$\mathbf{CONSTANT\ A, B : Prop}.$$
$\mathbf{CONSTANT\ p : \sim B}.$ (expressing that B is not the case)
$$\mathbf{CONSTANT\ x : A \to B}.$$

In classic propositional logic MT is justified as follows. Suppose that $a : A$ (A is the case), then by MP B is the case because $(xa) : B$. But B is not

https://doi.org/10.1017/S0074180900193787 Published online by Cambridge University Press

the case, so we have here a contradiction, and the supposition a : A must be discarded and as a consequence A is not the case. This argument cannot be used in constructive logic because there is no objection against $p :\sim B$ as well as $q : B$ for some q. Supplying A with a resident does not help, so in above base a member of $\sim A$ cannot be constructed. But Modus Tollens can be entered as a hypothesis:

$$\textbf{HYPOTHESIS MT} : (\textbf{ALL X, Y : Prop})(\sim \mathbf{Y} \rightarrow (\mathbf{X} \rightarrow \mathbf{Y})) \rightarrow \sim \mathbf{X}.$$

4. Self-interpretation

The purpose of a formal language L, i.e., a symbolic system with (syntactical) composition rules, is conveying information. So basic units of L as words and sentences admit interpretation. Consider the case that L is a terrestrial, natural language. Formalising (or modelling) the syntax of natural languages has been done on the basis of rewriting rules (e.g., in Chomskyan generative grammars) and is known as a rather complex undertaking in many ways. The difficulties of defining the semantics of a natural language are considered in linguistics as yet to be insurmountable. There is, on the other hand, the remarkable fact that humans can learn to handle a natural language reasonably well in a relatively short period of time (dependent on their ages). The predominant method for achieving this seems to be the use of examples in the real world explained in the language by teachers. Assigning meaning to terms (say sentences) in a language is in practise done at first by referring to concrete objects and relations in reality, while abstraction is entered in later stages in the learning process. Once a language is mastered, a human can appreciate how a language can be used to explain itself, say in textbooks, possibly with the help of more or less sophisticated formalisations. A kind of (partial) self-interpretation becomes then operational in this setting. Using such material and/or by listening to accomplished orators and/or by reading, humans using a kind of bootstrapping can increase their linguistic abilities.

Lingua Cosmica is a symbolic system as well with its own syntax and semantics. It is meant to be used at a secondary level annotating messages written in a natural language – it is therefore a meta language. How do we explain to an ETI recipient the syntax and semantics of *LINCOS*? The syntax is relatively simple. Technically speaking, membership, functional application, lambda abstraction, definitions, hypothesis and facts are all the ingredients. Given a sufficiently rich sample of expressions, the rules of composition of well-formed terms should not present much of a problem. On the other hand there is no immediate evidently useful set of examples (common ground) between humans and ETI.

The use of other symbolic logics, themselves outside *LINCOS*, seems to be useful for explaining the syntax, but especially the semantics of the language, subject to restrictions. The authors consider in the present contribution a restricted form of self-interpretation, i.e., *LINCOS* is explained in terms of *LINCOS*. The examples used are not from reality but occur in the realm of one of the simplest forms of symbolic logic: propositional calculus, itself outside *LINCOS*. Rules of propositional logic (in fact, a set of deduction rules) have

https://doi.org/10.1017/S0074180900193787 Published online by Cambridge University Press

been embedded in the system in the present paper. Each rule is first given a direct interpretation and then transformed to a verifiable fact. The (constructive) verifications of facts as propositional expressions are identical to their direct interpretations. In other words, deduction rules explain their counterparts, the facts, and vice versa. So deduction rules and the derived provable facts are two sides of the same coin.

Fortunately this is not the only way of achieving the goal. Self-interpretation can also be based on examples from predicate logic, and future work will show how this can be done. Earlier work by the first author focused on the idea that interaction between texts in a natural language and *LINCOS* can be used to explain *LINCOS* (and in fact perhaps even both, also by a kind of bootstrapping!). Another idea is to use music for the purpose. The first author did a preliminary study on this recently.

Acknowledgments. Supported by a grant to the second author from The John Templeton Foundation. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of The John Templeton Foundation.

Doug Vakoch (*photo: Seth Shostak*)

https://doi.org/10.1017/S0074180900193787 Published online by Cambridge University Press