A refutation calculus for intuitionistic logic

Gianluigi Bellin and Luca Tranchini

Classically, logical consequence can be equivalently defined as truth transmission or as "backward" falsity transmission, in the following sense: that a consequence statement $\Gamma \vdash \Delta$ holds can be equated either with the condition that if all Γs are true, at least one of the Δs must be true as well, or, equivalently with the condition that if all Δs are false, at least one of the Γs must be false as well.

Intuitionism replaces the classical notion of truth with the constructive one of proof, where proofs are understood, via the Curry-Howard isomorphism, as computer programs: A consequence claim $\Gamma \vdash A$ is justified if there is an algorithm that taken a proof of each of the Γs as input produces a proof of A as output. Such algorithms are associated to deductions of Gentzen-Prawitz natural deduction systems. In particular to each rule for constructing deductions there is associated a basic operation, in the case of the implicational fragment the operations being λ-abstraction and function application:

$$\frac{\begin{array}{c}[x : A]\\ t : B\end{array}}{\lambda x.t : A \supset B} \qquad \frac{t : A \supset B \qquad s : A}{\mathtt{app}(t,s) : B}$$

An inference rule is thus seen as encoding an operation that given proofs of the premises yields a proof of the conclusion.

In the present paper we show how the deductions of the implicational fragment of intuitionistic logic can be equivalently understood as computer programs that taken a refutation of their end-formula as input yield a refutation of (the conjunction of) their assumptions as output, and thus that also in the constructive setting it is possible, like in the classical case, to dualize the account of consequence by exchanging truth with falsity and reversing the direction of transmission.

Refutation is here understood as a primitive (metalinguistic) notion rather than analysed in terms of an (object language) operator of negation. At the level of atomic propositions, such an idea is not new (for instance, it is common to introduced a primitive predicate of apartness as the genuine constructive counterpart of negated equality). Here we propose to go one step further, giving a constructive meaning to the refutation of logically complex formulas starting from the refutation of their components.

To attain this goal, one has to associate to each inference rule particular operations that given a refutation of the conclusion yield a refutation of the premises. But what should these operations be?

$$\frac{\begin{array}{c}[? : A]\\ ? : B\end{array}}{t : A \supset B} \qquad \frac{? : A \supset B \qquad ? : A}{t : B}$$

To answer the question, we do *not* start from the common idea according to which a refutation of $A \supset B$ is a pair consisting of a proof of A and a refutation of B. Though intuitive, this

makes the notion of refutation depend on the one of proof. Our starting point is rather the idea that to refute a $A \supset B$ one has to show that it is impossible to convert a refutation of B into a refutation of A (this we take the closest way of dualizing the idea that a proof of $A \supset B$ is a function from proofs of A to proofs of B).

Taking this idea seriously, it follows that given a refutation t of B we have two alternatives (among which we may not be able to tell which is the case): either it is impossible to convert t into a refutation of A, or it is possible to do so, though we may not know how. We take the operation associated to the elimination rule to encode these two alternatives and thus as splitting the computation into two threads, one representing the alternative in which it is impossible to transform the refutation of B into one of A (that is, in which $A \supset B$ has been refuted), the other representing the alternative in which a refutation of A could be produced out of the refutation of B:

$$\frac{\mathtt{mkc}(t, x) : A \supset B \qquad x_{\langle t \rangle} : A}{t : B}$$

The term $x_{\langle t \rangle}$ should be understood as having no real computational content, since we need not know anything about the refutation of A. In this sense it reminds of a variable, but it is not a real variable: we have not assumed to have a refutation x of A outright, but only postulated it as a possible alternative induced by the availability of a refutation t of B. (That x is not a real variable will be made precise below by taking the displayed occurrence of the variable x to be bound in $x_{\langle t \rangle}$).

The notion of impossibility underlying the informal explanation of the refutation conditions of $A \supset B$ is made precise in operational terms using the notion of error. Suppose that among the threads of a computation starting from a refutation x of B there are some that output refutations $s_1, \ldots, s_n$ of A. If one has a refutation t of $A \supset B$, one knows that these threads are spurious, i.e. that they represent impossible alternatives, since to have a refutation of $A \supset B$ is to know that it is impossible to convert refutations of B into refutations of A. Thus these threads can be closed with an error message that "explains" the incompatibility between the availability of a refutation of $A \supset B$ and the content of the thread (the possibility of producing—among other alternatives—refutations of A from a refutation of B).

We can thereby decorate the introduction rule for implication as follows:

$$\frac{\boxed{\mathtt{err}(t \mid x \mapsto s_i)} \quad [s_i^* : A] \quad x_{\langle t \rangle} : B}{t : A \supset B}$$

where the error message $\mathtt{err}(t \mid x \mapsto s_i)$ plays the role of a discharge index in linking the assumptions with the inference rule at which they are discharged. The operational content of the rule is thus the following: when one has a refutation of $A \supset B$ one can reason as if one had a refutation of B (represented by the "merely stipulated variable" $x_{\langle t \rangle}$) with the additional information that subsequently generated refutations $s_i^* \equiv s_i[\![x_{\langle t \rangle} / x]\!]$ of A are spurious.

From these introductory remarks the following fundamental difference between the usual Curry-Howard correspondence and the correspondence between deduction and terms proposed in the present paper should be clear. In the usual Curry-Howard interpretation a whole deduction is encoded by the (unique) proof-term decorating the end-formula of the deduction, and the free variables of this term are those decorating the undischarged assumptions of the deduction. In the refutation-based interpretation we are proposing, a whole deduction is

encoded by a family of terms decorating the undischarged assumptions and error messages associated to the discharged assumptions, and these terms and error messages will all depend on the (unique) variable decorating the end-formula of the deduction. Intuitively, each term and error message of a family should be thought of as corresponding to a thread of an algorithm that, taken a refutation of the end-formula as input, produces either a refutation of the (undischarged) assumptions or an error message as output. In general we are not in the position of selecting one thread as the one providing the "correct" output (for example, given a refutation of B we cannot tell in general, if we can obtain a refutation of A or of $A \supset B$).

It is worth remarking that to a deduction in which all assumptions have been discharged (i.e. a proof of the end-formula) we associate an algorithm which takes a refutation of the end-formula as input and outputs error messages from all of its threads. Informally, the algorithm shows that it is impossible to refute the end-formula. The analysis of "provability" as "impossibility of refuting" is closely connected to the analysis of classical provability via a double negation translation which lies at the basis of computational accounts of classical logic, especially of the $\lambda\mu$-calculus of Parigot [4]. In particular, in the context of the $\lambda\mu$-calculus, Crolard [3] defined two operations: $\texttt{make-coroutine}(t, \alpha)$ and $\texttt{resume}\ t\ \texttt{with}\ x \mapsto s$ and used them to formulate the Curry-Howard correspondence for subtractive logic (more commonly referred to as bi-intuitionistic logic) an extension of intuitionistic logic with a connective dual to intuitionistic implication.

The first author [1, 2] proposed several "parallel" variants of Crolard's calculus, as a term assignment to dual intuitionistic logic (and linear versions thereof), in which Crolard's operations were taken as primitive rather than defined.

The goal of our paper is twofold: First to provide a very concise and crisp presentation of the first author's "parallel" calculus, with the hope of making the main idea underlying it accessible to the widest audience possible. Second, we show that these operations can be used to develop a term assignment for refutations in implicational logic, rather than for proofs in subtractive logic as Crolard (and the first author) did.

After introducing the calculus by presenting terms, typing rules and conversions, we give a proof of strong normalization via an embedding of our calculus in λ-calculus. We close the paper with a comparison between our calculus and Crolard's original term assignment for subtractive logic, stressing the significance of the parallel nature of the calculus we propose.

References

[1] Bellin, G.: 2005, A term assignment for dual intuitionistic logic. Paper presented at the Workshop *Intuitionistic Modal Logics and Applications* affiliated at the LICS 2005 Conference, Chicago.

[2] Bellin, G.: 2014, Categorical proof theory of co-intuitionistic linear logic, *Logical Methods in Computer Science* **10**(3).
URL: *http://dx.doi.org/10.2168/LMCS-10(3:16)2014*

[3] Crolard, T.: 2004, A formulae-as-types interpretation of subtractive logic, *Journal of Logic and Computation* **14**(4), 529–570.

[4] Parigot, M.: 1992, $\lambda\mu$-calculus: An algorithmic interpretation of classical natural deduction, *in* A. Voronkov (ed.), *Logic Programming and Automated Reasoning,International Conference LPAR'92, St. Petersburg, Russia, July 15-20, 1992, Proceedings*, Springer, pp. 190–201.