Bidirectional Reasoning

Alessandro Agostini[†] and Fausto Giunchiglia[†,‡]
[†] Mechanized Reasoning Group IRST,
38050 Povo - Trento, Italy.
[‡] University of Trento,
via Inama, 5 - 38100 Trento, Italy.
fausto@irst.it

Abstract

The goal of this paper is to present a formal system **FB** for *bidirectional reasoning* which integrates forward and backward deduction. **FB** is proved equivalent to Gentzen's classical system of propositional natural deduction. **FB** is the logic of a theorem prover which supports interactive proof construction in general domains.

1 Introduction

Forward and backward reasoning are well known techniques applied in both knowledge representation and automated theorem proving (see for instance [17, 4, 19, 15, 25, 14, 8]). Basically, any problem can be solved in a forward direction, i.e. from the hypothesis to the conclusion, or in a backward direction, i.e. from the goal to (sub)goals. Forward and backward reasoning can be integrated in a single *bidirectional reasoning* system, where a deduction can be performed working both forward from *facts* and backward from *goals*. Roughly speaking, bidirectional reasoning has been discussed according to three different levels of abstraction and detachment from the underlying logic.

At the first level, which we call the **heuristic** level, bidirectional reasoning is seen as an efficient strategy for problem solving, and it is discussed and formalized almost independently of the underlying logic. Some examples in mathematics are [22, 23], and [27]. One example in AI is [19]. At this level, it does make sense to speak about the psychological plausibility of bidirectional reasoning (see for instance [15, 25, 3]).

At the second level, which we call the **tactic** level, bidirectional reasoning is performed by using rules (called *tactics*), which are built on the underlying logical rules). The most important major examples in theorem proving are `LCF` and its descendants ([12, 7, 21, 20]). In `LCF` forward reasoning is performed by using formal inference rules and backward reasoning is performed at the metalevel by using *tactics*. Another example in AI is GOAL [5], that provides `FOL` [28] with a goal oriented language for interactive proof construction.

At the third level, which we call the **logical** level, bidirectional reasoning is totally performed by *bidirectional deductions* inside a well defined formal system. As far as we know, this approach has never been explored before.

In this paper, we present **FB**, the logic implemented in an interactive theorem prover called GETFOL[1]. **FB** has the following features:

(1) It defines bidirectional reasoning as deduction, in the case of classical (propositional) logic;

(2) **FB** is equivalent to Gentzen's classical system of natural deduction ([10, 24]) and his variant described in [9];

(3) the deductions in **FB** are *natural*, in the sense that they extend the natural way to costruct proofs in the Gentzen's systems of natural deduction. These deductions have some noteworthy properties, the major of them is a *naturalness theorem* for the system **FB**.

FB does not increase the number of theorems, but only the number of possible (natural) strategies which can be used to prove the same theorem. **FB** allows the embedding of a relevant control knowledge - the *direction of reasoning* - into the logic itself. This leads to consider the so called *bidirectionality* of reasoning as a basic property of formal deductions. In this context, the number of inference rules used is irrelevant [16]: "A point worthy of stress is that a deductive system is not "simpler" merely because it employs fewer rules of inference. A more meaningful measure of simplicity is the ease with which heuristic considerations can be absorbed into the system".

This paper is structured as follows. **FB** is developed in Section 2. In particular, the inference rules in **FB** are divided into three main types: forward rules for reasoning forwards, backward rules for reasoning backwards (Section 2.1), and a special rule - called *goal-to-theorem rule* - for matching forward and backward parts of reasoning (Section 2.2). Section 3 describes the naturalness properties of bidirectional reasoning in **FB**. Finally, Section 4 and Section 5 give a short discussion of the related work and some conclusions.

2 The formal system FB

FB is formally defined as an *axiomatic formal system*, i.e. a triple $\langle \mathcal{L}, \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{L}$ is a propositional language, $\mathcal{A}$ is a set of axioms and $\mathcal{R}$ is a set of inference rules. In particular, only logical axioms appear in the theory, i.e. *basic sequents* of the form $A \longrightarrow A$, where A is a well formed formula of $\mathcal{L}$. However, the theory can be easily extended to include axioms. We call **FB** a *bidirectional natural deduction system*. As we will see later, **F** and **B** represent indipendent and equivalent theories of forward and backward reasoning, respectively.

In this work we assume that a part of the deductive reasoning consists of the application of inference rules to the premises and the conclusion(s) of an argument. Premises, or more in general *facts*, are represented in (the language of) **FB** as *sequents*, i.e. pairs $\langle \Gamma, A \rangle$, also written $\Gamma \longrightarrow A$, where A is a formula and Γ is a finite set of formulae. Conclusions, or more generally *goals*, are represented as $B \longleftarrow \Delta$, where B, Δ have the same meaning as in sequents. Briefly, the language $\mathcal{L}$ can be defined as the set of sequents and goals over a propositional language.

[1]GETFOL ([11]) is a reimplementation of the FOL system ([28]). GETFOL has, with minor variations, all the functionalities of FOL plus extension, some of which described here, to perform bidirectional reasoning.

2.1 Forward and backward systems

The inference rules in $\mathcal{R}$ but one are either sequent or goal versions of the natural deduction calculus [24] for classical propositional logic. They allow introduction and elimination only in the succedent of the sequents (goals). The inference rules can be divided into two main types: forward rules for reasoning forwards, and backward rules for reasoning backwards.

The forward inference rules define the *forward system* **F** (Table 1).

$\wedge I$	$\dfrac{\Gamma \longrightarrow A \quad \Delta \longrightarrow B}{\Gamma\ \Delta \longrightarrow A \wedge B}$	$\wedge E_l$	$\dfrac{\Gamma \longrightarrow A \wedge B}{\Gamma \longrightarrow A}$
		$\wedge E_r$	$\dfrac{\Gamma \longrightarrow A \wedge B}{\Gamma \longrightarrow B}$
$\vee I_l$	$\dfrac{\Gamma \longrightarrow A}{\Gamma \longrightarrow A \vee B}$		
$\vee I_r$	$\dfrac{\Gamma \longrightarrow B}{\Gamma \longrightarrow A \vee B}$	$\vee E$	$\dfrac{\Gamma \longrightarrow A \vee B \quad \Delta\ A \longrightarrow C \quad \Theta\ B \longrightarrow C}{\Gamma\ \Delta\ \Theta \longrightarrow C}$
$\supset I$	$\dfrac{\Gamma\ A \longrightarrow B}{\Gamma \longrightarrow A \supset B}$	$\supset E$	$\dfrac{\Gamma \longrightarrow A \quad \Delta \longrightarrow A \supset B}{\Gamma\ \Delta \longrightarrow B}$
$\perp_i$	$\dfrac{\Gamma \longrightarrow \perp}{\Gamma \longrightarrow A}$	$\perp_c$	$\dfrac{\Gamma\ \neg A \longrightarrow \perp}{\Gamma \longrightarrow A}$

Table 1: The deductive machinery of **F**.

The backward inference rules define the *backward system* **B** (Table 2).

$\wedge I_b$	$\dfrac{A \wedge B \xleftarrow{?} \Gamma\ \Delta}{A \xleftarrow{?} \Gamma \quad B \xleftarrow{?} \Delta}$	$\wedge E_{lb}$	$\dfrac{A \xleftarrow{?} \Gamma}{A \wedge B \xleftarrow{?} \Gamma}$
		$\wedge E_{rb}$	$\dfrac{B \xleftarrow{?} \Gamma}{A \wedge B \xleftarrow{?} \Gamma}$
$\vee I_{lb}$	$\dfrac{A \vee B \xleftarrow{?} \Gamma}{A \xleftarrow{?} \Gamma}$		
$\vee I_{rb}$	$\dfrac{A \vee B \xleftarrow{?} \Gamma}{B \xleftarrow{?} \Gamma}$	$\vee E_b$	$\dfrac{C \xleftarrow{?} \Gamma\ \Delta\ \Theta}{A \vee B \xleftarrow{?} \Gamma \quad C \xleftarrow{?} \Delta\ A \quad C \xleftarrow{?} \Theta\ B}$
$\supset I_b$	$\dfrac{A \supset B \xleftarrow{?} \Gamma}{B \xleftarrow{?} \Gamma\ A}$	$\supset E_b$	$\dfrac{B \xleftarrow{?} \Gamma\ \Delta}{A \xleftarrow{?} \Gamma \quad A \supset B \xleftarrow{?} \Delta}$
$\perp_{ib}$	$\dfrac{A \xleftarrow{?} \Gamma}{\perp \xleftarrow{?} \Gamma}$	$\perp_{cb}$	$\dfrac{A \xleftarrow{?} \Gamma}{\perp \xleftarrow{?} \Gamma\ \neg A}$

Table 2: The deductive machinery of **B**.

Remark. We may note in passing that there is a total symmetry between the inference rules in **F** and in **B**, respectively.

2.2 Forward/backward system

At this point, a possible question is why do we need both forward and backward inference rules; in particular, why it is necessary to have rules that are forward and backward versions of the same deductive scheme. The answer is that the two rule types play different roles, both equally important. Intuitively, forward rules are useful in getting started on a problem by carrying out deductions that clarify the meaning of the premises. At the later stage, they may also be helpful for simplifying intermediate results. Only for forward inference rules is normally associated a logical consequence relation. Moreover, forward strategies "have been observed in subjects' performance on other mathematical tasks", and they are also needed in situations where "no particular goal is specified" ([25], p. 48).

On the other hand, forward rules, even if complete[2], are not well suited for ordinary mathematical reasoning, and to see why not, consider the following example. Suppose to prove: $((A\supset B)\wedge(B\supset C))\supset(A\supset C)$. The resulting proof is given as follows:[3]

$$\dfrac{\dfrac{[(A\supset B)\wedge(B\supset C)]}{B\supset C}\wedge E \qquad \dfrac{\dfrac{[(A\supset B)\wedge(B\supset C)]}{A\supset B}\wedge E \qquad [A]}{B}\supset E}{C}\supset E$$

$$\dfrac{\dfrac{C}{A\supset C}\supset I_b}{((A\supset B)\wedge(B\supset C))\supset(A\supset C)}\supset I_b$$

Figure 1: A bidirectional natural proof.

The upper box is used to denote the forward part of the proof, the lower one the backward part. As empirical strategy, at each step of the above proof, backward rules $\supset I_b$ are checked first; this allow us to reduce the complexity of the conclusion $((A\supset B)\wedge(B\supset C))\supset(A\supset C)$ to C. Then, the *usable assumptions*[4] $[A]$ and $[(A\supset B)\wedge(B\supset C)]$ are used for starting the forward proof. Third, forward rules are applied to usable assumptions to deduce C. Finally, it is checked if forward and backward parts of the proof can be matched to a single deduction. The proof succeeds if a such kind of matching exists, it fails otherwise. In Figure 1, the "matching point" is the formula C at the bottom of the upper box and at the top of the lower box, and the proof succeeds. In [2] we argue in more details that this empirical strategy is supported by the existence of a particular normal form for deductions in **FB**.

A goal, e.g. the theorem to prove, can be achieved in only few ways, depending on the goal complexity as well as on the available rules, by applying a precisely chosen sequence of in some way *evident* rules. For such problems it should be more convenient to work backwards. Perhaps this is the reason why backward reasoning is regarded by Polya [23], chapter 8.1, as "a more proper way of solving problems [prove theorems] than forward reasoning". There are also other factors implying that reasoning backwards can

[2] In the sense that any (propositional) theorem is provable in **F**.

[3] In this example, to simplify notation we write A for $\longrightarrow A$ by reasoning forwards and B for $B \overset{?}{\longleftarrow}$ by reasoning backwards.

[4] An usable assumption is used as an assumption. However, an important difference with assumptions is that an usable assumption is introduced in the deduction by a backward inference rule rather than directly by the user and, moreover, it is certainly discharged by reasoning forwards.

be advantageous (see for instance [14]). As described in [14], the most relevant is that backward reasoning can be used to organize the bidirectional reasoning. In particular, backward reasoning gives premises - the usable assumptions - from which we can start a deduction.

A bidirectional deduction contains basically two parts, which are called the *sequent tree*, i.e. a tree of sequents, say Π, and the *goal tree*, i.e. a tree of goals, say $\coprod$. We need also a suitable formalism for representing the possible matching of them. An intuitive idea of "matching" is given in the example in Section 2. Here we shall describe this idea formally. We introduce the following *goal-to-theorem* (*gtt*) rule:

$$\dfrac{\begin{array}{c} A \overset{?}{\leftarrow} \Delta \\ \coprod \\ A_1 \overset{?}{\leftarrow} \Delta_1 \cdots A_n \overset{?}{\leftarrow} \Delta_n \end{array} \quad \Delta_1\ \Gamma_1 \longrightarrow A_1 \cdots \Delta_n\ \Gamma_n \longrightarrow A_n}{\Delta\ \bigcup_{i=1}^{n} \Gamma_i \longrightarrow A}\ gtt$$

where $\coprod$ is a goal tree. The *gtt*-rule denotes a further (actually the least) inference rule in **FB** and has no restrictions. The intuitive meaning of this rule follows: Every theorem to prove can be represented as a goal $A \overset{?}{\leftarrow} \Delta$, with Δ possibly empty. This goal can be reduced to (sub)goals $A_1 \overset{?}{\leftarrow} \Delta_1, ..., A_n \overset{?}{\leftarrow} \Delta_n$ by using backward rules in **B**. If there is a fact $\Delta_i\ \Gamma_i \longrightarrow A_i$ (e.g., obtained by reasoning forward in **F**) for every $i \leq n$, then we may consider every (sub)goal $A_i \overset{?}{\leftarrow} \Delta_i$ to be matched, or *closed*, by the fact $\Delta_i\ \Gamma_i \longrightarrow A_i$ $(i \leq n)$. Thus, we can go up to the goal tree $\coprod$ again to transform the root goal $A \overset{?}{\leftarrow} \Delta$ into the fact $\Delta \longrightarrow A$. This is the conclusion of the *gtt*-rule when Γ_i is empty for every $i \leq n$. However, Γ_i could be not empty for some $i \leq n$. Then, we must add Γ_i to Δ to form the fact $\Delta\ \bigcup_{i=1}^{n} \Gamma_i \longrightarrow A$. Notice that the *gtt*-rule discharges implicitly the formulae Δ_i inside any fact $\Delta_i\ \Gamma_i \longrightarrow A_i$, for every $i \leq n$.

Remark. If the union of all Γ_i for every $i \leq n$ is empty, then *gtt* provides us with a desired formalism to represent *matching*.[5] In the following, we refer to gtt_{base} this restricted *gtt*-rule. However, in the general case, *gtt* allows us to prove a sequent "weaker" than the goal to obtain. This sequent is a partial solution of the goal to prove and it could be used as a lemma to approach the final solution.

Given the *gtt*-rule, we can formalize entirely the bidirectional reasoning inside the formal system **FB**. Then, we define what is meant by $\Pi\coprod$ being a *deduction in* the system **FB** *of* a sequent $\Gamma \longrightarrow A$.

Definition 1 (Deduction in FB of) *: A deduction in the system* **FB** *of a sequent* $\Gamma \longrightarrow A$ *is inductively defined as follows:*

(*i*) *A basic sequent* $A \longrightarrow A$ *is a deduction in* **FB** *of* $A \longrightarrow A$.

(*ii*) *If* $\Pi\coprod_i$ *is a deduction in* **FB** *of* $\Gamma_i \longrightarrow A_i$ *for every* $i \leq n$, *then*

$$\dfrac{\begin{array}{c}\Pi\coprod_1 \\ \Gamma_1 \longrightarrow A_1\end{array} \quad \cdots \quad \begin{array}{c}\Pi\coprod_n \\ \Gamma_n \longrightarrow A_n\end{array}}{\Gamma \longrightarrow A}\ \rho$$

[5] Roughly speaking, matching is only the one-to-one relation between the i-th goal $A_i \overset{?}{\leftarrow} \Delta_i$ and the i-th sequent $\Delta_i\ \Gamma_i \longrightarrow A_i$ of the rule. The matching as given in Figure 1 in the previous section is only a simple case where Γ's is empty, A_i is C and Δ_i is $\{A, (A \supset B) \wedge (B \supset C)\}$.

is a deduction in **FB** *of* $\Gamma \longrightarrow A$ *provided that* ρ *is a (instance of a) forward inference rule.*

(iii) *If* Π *is a goal tree with root* $A \overset{?}{\leftarrow} \Delta$ *and* $\Pi\Pi_i$ *is a deduction in* **FB** *of* $\Delta_i\ \Gamma_i \longrightarrow A_i$ *for every* $i \leq n$, *then*

$$\dfrac{\begin{array}{c} A \overset{?}{\leftarrow} \Delta \\ \Pi \\ A_1 \overset{?}{\leftarrow} \Delta_1 \cdots A_n \overset{?}{\leftarrow} \Delta_n \end{array} \quad \begin{array}{c} \Pi\Pi_1 \\ \Delta_1\ \Gamma_1 \longrightarrow A_1 \end{array} \cdots \begin{array}{c} \Pi\Pi_n \\ \Delta_n\ \Gamma_n \longrightarrow A_n \end{array}}{\Delta\ \bigcup_{i=1}^{n} \Gamma_i \longrightarrow A}\ gtt$$

is a deduction in **FB** *of* $\Delta\ \bigcup_{i=1}^{n} \Gamma_i \longrightarrow A$.

FB has the first of the two main desired properties, i.e. the equivalence with the sequent version **F** of the propositional Gentzen-type system of natural deduction **ND** as modelled by Prawitz [24].[6]

Definition 2 (Equivalence) :

(i) *Identical formulae, sequents and goals are equivalent.*

(ii) *The sequent* $A_1 \cdots A_n \longrightarrow C$ *is equivalent to the following formula: If* $n \geq 1$, *then*

$$A_1 \wedge \cdots \wedge A_n \supset C;$$

otherwise: C.

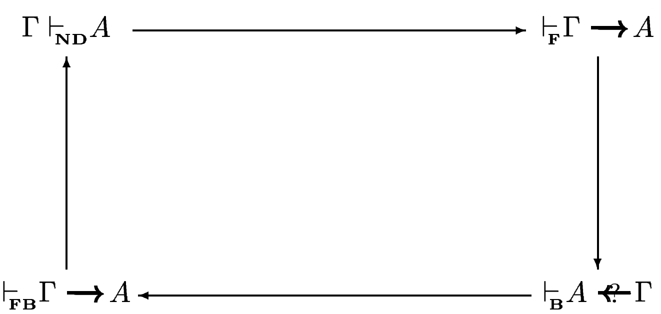
(iii) *The sequent* $A_1 \cdots A_n \longrightarrow C$ *is equivalent to the goal* $C \overset{?}{\leftarrow} A_1 \cdots A_n$.

(iv) *The equivalence is transitive.*

Two *derivations* are called equivalent if the *endformula* (*endsequent*, *rootgoal*) of one is equivalent to that of the other.[7] Two *calculi* are called equivalent if every derivation in one calculus can be transformed into an equivalent derivation in the other calculus.

Theorem 1 (Equivalence) : *The formal systems* **ND**, **F**, **B** *and* **FB** *are equivalent, i.e.* $\Gamma \vdash_{\mathbf{ND}} A$ *iff* $\vdash_{\mathbf{F}} \Gamma \longrightarrow A$ *iff* $\vdash_{\mathbf{B}} A \overset{?}{\leftarrow} \Gamma$ *iff* $\vdash_{\mathbf{FB}} \Gamma \longrightarrow A$.

Proof: (*hint*) We can draw the following diagram (see [1]):



[6]For lack of space, only proof outlines are given. For details see [1].

[7]When we consider derivations in **ND** of a formula A depending on a non-empty set of formulae Γ, this definition falls. For a substantially wider definition of equivalent derivations, which is adequate for our proof of equivalence, see [1].

Let $\bigwedge\Gamma$ be the conjunction of the formulae of Γ. Then, by the deduction theorem in **ND**, it holds $\Gamma \vdash_{\mathbf{ND}} A$ if and only if $\vdash_{\mathbf{ND}} \bigwedge\Gamma \supset A$. Hence, by definition of equivalence, $\bigwedge\Gamma \supset A$ is equivalent to $\Gamma \longrightarrow A$, which is equivalent to $A \overset{?}{\longleftarrow} \Gamma$. Therefore, by the transitivity of equivalence, **ND**, **F**, **B** e **FB** are equivalent. □

The intuitive meaning of Theorem 1 is simple: appropriately formalized forward reasoning (inference rules in **F**, or in **ND**), backward reasoning (inference rules in **B**) and bidirectional reasoning (inference rules in **FB**) are equivalent, in the sense that all assure identical potential ability of finding the proof of a theorem, i.e., they allow us to prove the same theorems. Nothing, however, is said about the efficiency, or the naturalness[8], of bidirectional reasoning and its dependence on the reasoning direction choice; these problems will be discussed in the next section.

3 The naturalness principle

Bidirectional reasoning is *natural* with respect to many points of view. As summarized by Nilsson [19]: "there is evidence that a 'good' choice of the search direction can, at least in some cases, totally reduce the number of nodes that can be generated". Furthermore, "whenever possible, the direction of reasoning ought to be in the direction of a decreasing number of alternatives" and "the best direction in which to apply a [inference] rule sometimes depends on the domain". Here we present a new concept of naturalness of reasoning, as formalized in **FB**, that allows us to choose the best reasoning direction in a natural way and independently of the reasoning domain. Thus, the deductions in **FB** have some noteworthy properties summarized into two main results: (a) the naturalness theorem and his generalization (b) the strong naturalness theorem.

NATURALNESS PRINCIPLE. The relationship between a Prawitz's natural deduction and the bidirectional reasoning as discussed at the euristic level (see Section 1) is roughly expressed by the following principle, which we shall call *the naturalness principle*:

> *If $\Gamma \vdash_{\mathbf{ND}} A$, then there is a deduction*[9] *of A from Γ in which a part is constructed by reasoning backwards from A, and a part is constructed by reasoning forwards from Γ.*

We illustrate this principle by two examples; a more detailed verification follows after Theorem 2 below.

Example 1. $(A \supset B) \supset ((A \supset (B \supset C)) \supset (A \supset C))$ may be deduced in **ND**. In agreement with the naturalness principle, it holds: the deduction in **ND**

[8] In the context of interactive proofs construction we may consider efficiency and naturalness as synonymous. However, it is also necessary to remember that this point of view is misleading in strictly automated theorem proving. There, efficiency and naturalness are usually opposite terms.

[9] As defined in [24], p. 24.

$$\dfrac{\dfrac{[A] \quad [A\supset B]}{B}\supset E \qquad \dfrac{[A] \quad [A\supset (B\supset C)]}{B\supset C}\supset E}{C}\supset E$$

$$\dfrac{\dfrac{\dfrac{C}{A\supset C}\supset I_b}{(A\supset (B\supset C))\supset (A\supset C)}\supset I_b}{(A\supset B)\supset ((A\supset (B\supset C))\supset (A\supset C))}\supset I_b$$

of $(A\supset B)\supset((A\supset(B\supset C))\supset(A\supset C))$ from the empty set is constructed by reasoning backwards from $(A\supset B)\supset((A\supset(B\supset C))\supset(A\supset C))$ (see lower box) and by reasoning forwards from assumptions A, $A\supset B$ and $A\supset(B\supset C)$, successively discharged (see upper box).

Example 2. $A\vee\neg A$ may be deduced in **ND**. In agreement with the naturalness principle, it holds: the deduction in **ND**

$$\dfrac{A}{A\vee\neg A}\vee I_b \qquad\qquad \dfrac{\neg A}{A\vee\neg A}\vee I_b$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[A]}{A\vee\neg A}\vee I \quad [\neg(A\vee\neg A)]}{\bot}\neg E}{\neg A}\neg I \qquad \dfrac{\dfrac{\dfrac{[\neg A]}{A\vee\neg A}\vee I \quad [\neg(A\vee\neg A)]}{\bot}\neg E}{\neg\neg A}\neg I}{\dfrac{\bot}{A\vee\neg A}\bot_c}\neg E$$

of $A \vee\neg A$ from the empty set is constructed by reasoning backwards from $A\vee\neg A$ (see upper boxes) and by reasoning forwards from assumptions A, $\neg A$ and $\neg(A\vee\neg A)$, successively discharged (see lower box).

Remark. The naturalness principle is related to, but not identical to, the "*only if*" part of the equivalence theorem (i.e.: *if* $\Gamma \vdash_{\mathbf{ND}} A$ *then* $\vdash_{\mathbf{FB}} \Gamma \longrightarrow A$). In this respect, the vice versa holds. However, the principle expresses the naturalness of bidirectional reasoning in a deeper sense. If there is a deduction Π in **ND** of A from Γ, then Π can be "read" from the out to the middle, i.e. backwards from A and forwards from Γ - being sure that the two parts of forward and backward reasoning on Π "match". Thus, in this sense, the converse of principle is false, because it does not say how to understand this kind of matching.

The naturalness principle is somewhat informal. Following the remark above, what does it mean to say "read a **ND**-deduction in an out-middle way" and "match two parts of forward and backward reasoning"? To answer this question, we need some terminology.

Let Π be a sequent tree. We first define a *matching line* in Π, in symbols $/\!\!\!\Pi$, as a set $\{s_1, ..., s_n\}$ of (occurrences of) sequents in Π such that, for every pair s_i, s_j $(i, j \leq n)$, s_i does not stand above s_j in Π and s_j does not stand above s_i in Π.[10] A *matching point* is a matching line with only one element.

[10]We take for granted the meaning of saying that a (occurrence of a) sequent s stands *above* a (occurrence of a) sequent s' (or that s' stands *below* s) in a sequent tree Π.

A matching line $\not\Pi$ with n elements splits a sequent tree Π into $n+1$ trees of sequents $\Pi_1, ..., \Pi_n$, Π', or shorter, *sequent sub-trees.* Π_i $(i \leq n)$ is an *upper sequent sub-tree* obtained from $\not\Pi$, and Π' is the *lower sequent sub-tree* obtained from $\not\Pi$. A *sequent sub-tree of* Π is an upper or lower sequent sub-tree obtained from any matching line in Π. A similar terminology may be provided for matching lines in a goal tree; e.g., it could be defined a matching line $\not\amalg$ in a goal tree $\amalg$.

The above terminology will be used to formalize what we mean for matching between forward and backward reasoning. In the following, we set up some terminology for representing "reading" in an *out-middle* way of Prawitz's natural deductions. A sequent tree Π and a goal tree $\amalg$ are *symmetrical* according to the following definition:

Definition 3 (Symmetry Π-$\amalg$) *:*

(*i*) *$\Gamma \longrightarrow A$ and $A \overset{?}{\longleftarrow} \Gamma$ are symmetrical.*

(*ii*) *If Π_i and $\amalg_i$ are symmetrical for every $i \leq n$, then*

$$\dfrac{\begin{array}{ccc}\Pi_1 & & \Pi_n \\ \Gamma_1 \longrightarrow A_1 & \cdots & \Gamma_n \longrightarrow A_n\end{array}}{\Gamma \longrightarrow A}\,\rho$$

and

$$\begin{array}{c}\dfrac{A \overset{?}{\longleftarrow} \Gamma}{A_1 \overset{?}{\longleftarrow} \Gamma_1 \qquad\qquad A_n \overset{?}{\longleftarrow} \Gamma_n}\,\rho_b \\ \amalg_1 \qquad \ldots \qquad \amalg_n\end{array}$$

are symmetrical, where ρ is a (instance of a) forward inference rule.

While reasoning backwards on a deduction in **ND**, we are doing a sort of implicit inversion of every inference step of deduction. Definition 3 makes explicit such kind of "mental" inversion. The naturalness principle thus suggests the following *naturalness theorem*:

Theorem 2 (Naturalness) *: Let Π be deduction in* **F** *of $\Gamma \longrightarrow A$, and let $\not\Pi = \{\Gamma_1 \longrightarrow A_1, ..., \Gamma_n \longrightarrow A_n\}$ be any matching line in Π. Then, there is a deduction in* **FB** *of $\Gamma \longrightarrow A$:*

$$\dfrac{\begin{array}{cc}\begin{array}{c}A \overset{?}{\longleftarrow} \Gamma \\ \amalg' \\ A_1 \overset{?}{\longleftarrow} \Gamma_1 \cdots A_n \overset{?}{\longleftarrow} \Gamma_n\end{array} & \begin{array}{ccc}\Pi_1 & & \Pi_n \\ \Gamma_1 \longrightarrow A_1 & \cdots & \Gamma_n \longrightarrow A_n\end{array}\end{array}}{\Gamma \longrightarrow A}\,gtt_{base}$$

with the following properties:

(1). *$\amalg'$ and Π' are symmetrical, where Π' is the lower sequent sub-tree obtained from $\not\Pi$.*

(2). *Π_i is an upper sequent sub-tree obtained from $\not\Pi$, for every $i \leq n$.*

Theorem 2 gives a result that is somewhat similar to, but more general of, the naturalness principle. Bidirectional reasoning is now fully represented inside the formal system **FB**. This formal description simplifies the proof of the principle and gives a logical meaning of - say metalogical - forward and backward reasoning as expessed in the principle.

Moreover, Theorem 2 provides also a proof of the converse of the principle, otherwise difficult to prove. In other words, Theorem 2 gives a formal meaning of both the naturalness principle and its converse. The proof of the theorem above is directly based on the naturalness principle and is quite simple; indeed, the proof is already suggested in the examples above (for the details, see [1]).

We shall describe Theorem 2 also in a more graphical notation (Figure 2). In Figure 2 the bigger triangle represents a deduction in **F**, while the line on it represents an arbitrary matching line; bold triangles are sequent trees. The lower arrow transforms a sequent tree, the polygon on the left, into a symmetrical goal tree, the polygon on the right (Theorem 2, property (1)). Both upper arrows represent no changes in the proof (Theorem 2, property (2)).

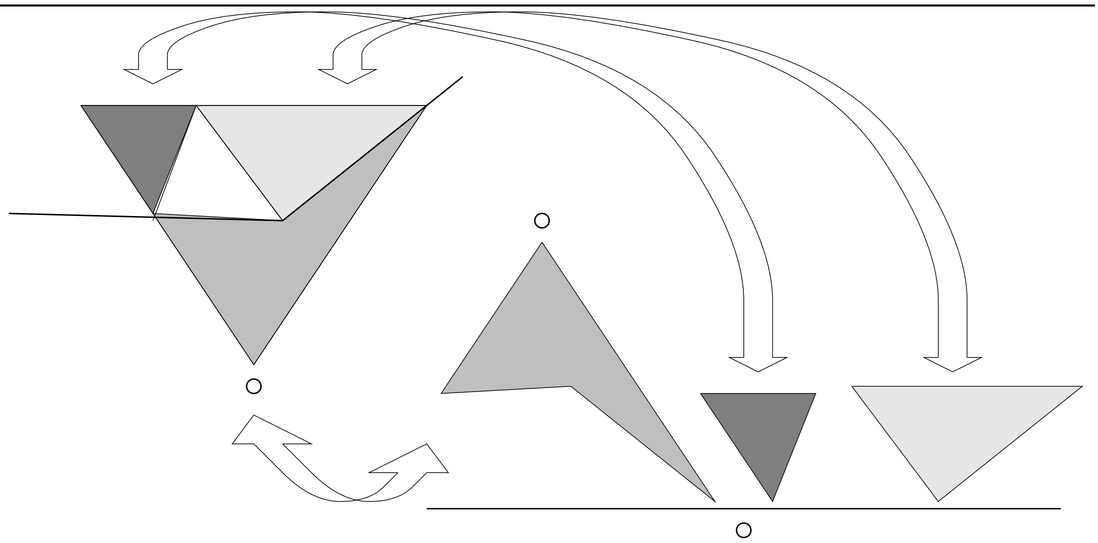


Figure 2: A graphical view of Theorem 2.

Remark. The naturalness theorem holds also for a deduction $\amalg$ in **B**.
Let $\not\amalg = \{A_1 \stackrel{?}{\longleftarrow} \Gamma_1,...,A_n \stackrel{?}{\longleftarrow} \Gamma_n\}$ be an arbitrary matching line in $\amalg$. Then, clauses (1) and (2) become:

- (1′) $\amalg'$ is the upper goal sub-tree obtained from $\not\amalg$, and
- (2′) Π_i and $\amalg_i$ are symmetrical, where $\amalg_i$ is an lower goal sub-tree obtained from $\not\amalg$, for every $i \leq n$.

Moreover, Theorem 2 holds for any sequent or goal tree and it is not restricted to deductions. Theorem 2 does not explain neither how many matching lines can be really constructed in a sequent (goal) sub-tree, nor how to choose a “good” matching line. To answer these questions, we first define certain trees of both sequents and goals, or shorter, *sequent-goal trees.*

Definition 4 (Sequent-goal tree) *: Let* $\Pi\amalg$ *be a deduction in* **FB** *of* $\Theta \longrightarrow C$*, and let* Π *be any sequent tree in* $\Pi\amalg$*. Then, for any matching line* $\not\Pi =$ $\{\Gamma_1 \longrightarrow A_1, ..., \Gamma_n \longrightarrow A_n\}$ *in* Π*,*

(i)

$$\dfrac{\begin{array}{c} A \overset{?}{\longleftarrow} \Delta \\ \Pi \\ A_1 \overset{?}{\longleftarrow} \Delta_1 \cdots A_n \overset{?}{\longleftarrow} \Delta_n \end{array} \quad \begin{array}{c} \Pi_1 \\ \Delta_1\ \Gamma_1 \longrightarrow A_1 \end{array} \cdots \begin{array}{c} \Pi_n \\ \Delta_n\ \Gamma_n \longrightarrow A_n \end{array}}{\begin{array}{c} \Delta\ \bigcup_{i=1}^{n} \Gamma_i \longrightarrow A \\ \Pi \\ \Theta \longrightarrow C \end{array}}\ gtt$$

is a sequent-goal tree.

(ii) *If* ΠП′, ΠП$_i$ *are sequent-goal trees for* $i \leq n$, *then*

$$\dfrac{\begin{array}{c} A \overset{?}{\longleftarrow} \Delta \\ \amalg \\ A_1 \overset{?}{\longleftarrow} \Delta_1 \cdots A_n \overset{?}{\longleftarrow} \Delta_n \end{array} \quad \begin{array}{c} \Pi\amalg_1 \\ \Delta_1\ \Gamma_1 \longrightarrow A_1 \end{array} \cdots \begin{array}{c} \Pi\amalg_n \\ \Delta_n\ \Gamma_n \longrightarrow A_n \end{array}}{\begin{array}{c} \Delta\ \bigcup_{i=1}^{n} \Gamma_i \longrightarrow A \\ \Pi\amalg' \\ \Theta \longrightarrow C \end{array}}\ gtt$$

is a sequent-goal tree.

Π, Π_i $(i \leq n)$ in Definition 4 are *sequent trees in* $\Pi\amalg$ and $\amalg$ is a *goal tree in* $\Pi\amalg$. We say also that $\Pi\amalg$ is a *sequent-goal tree of s* if s is the *endsequent* of $\Pi\amalg$, i.e., the "root" of the lower sequent tree in $\Pi\amalg$ (e.g., $\Theta \longrightarrow C$ above).

A matching line in a sequent-goal tree $\Pi\amalg$ is either a matching line in a sequent tree in $\Pi\amalg$ or a matching line in a goal tree in $\Pi\amalg$. Hence, we may present the following *strong naturalness theorem*:

Theorem 3 (Strong naturalness) *: Let* $\Pi\amalg$ *be a deduction in* **FB** *of* $\Theta \longrightarrow C$, *and let* Π *be any sequent tree in* $\Pi\amalg$. *Then, for any matching line* $\not\Pi = \{\Gamma_1 \longrightarrow A_1, ..., \Gamma_n \longrightarrow A_n\}$ *in* Π,

$$\dfrac{\begin{array}{c} A \overset{?}{\longleftarrow} \Delta \\ \amalg' \\ A_1 \overset{?}{\longleftarrow} \Gamma_1 \cdots A_n \overset{?}{\longleftarrow} \Gamma_n \end{array} \qquad \begin{array}{c} \Pi\amalg_1 \\ \Theta_1 \longrightarrow B_1 \\ \Pi_1 \\ \Gamma_1 \longrightarrow A_1 \end{array} \begin{array}{c} \\ \cdots \\ \\ \cdots \end{array} \begin{array}{c} \Pi\amalg_m \\ \Theta_m \longrightarrow B_m \\ \Pi_n \\ \Gamma_n \longrightarrow A_n \end{array}}{\begin{array}{c} \Delta \longrightarrow A \\ \Pi\amalg' \\ \Theta \longrightarrow C \end{array}}\ gtt_{base} \tag{1}$$

is a deduction in **FB** *of* $\Theta \longrightarrow C$.

Proof: (*hint*) We assume that $\Pi\amalg$ is structured as follows:

$$\begin{array}{c} \begin{array}{ccc} \Pi\amalg_1 & & \Pi\amalg_m \\ \Theta_1 \longrightarrow B_1 & \cdots & \Theta_m \longrightarrow B_m \end{array} \\ \Pi \\ \Delta \longrightarrow A \\ \Pi\amalg' \\ \Theta \longrightarrow C \end{array}$$

Then, by applying Theorem 2 to Π, we construct the following sequent-goal tree:

$$\dfrac{\begin{array}{c} A \overset{?}{\longleftarrow} \Delta \\ \amalg' \\ A_1 \overset{?}{\longleftarrow} \Gamma_1 \cdots A_n \overset{?}{\longleftarrow} \Gamma_n \end{array} \quad \begin{array}{c} \Pi_1 \\ \Gamma_1 \longrightarrow A_1 \end{array} \cdots \begin{array}{c} \Pi_n \\ \Gamma_n \longrightarrow A_n \end{array}}{\Delta \longrightarrow A}\ gtt_{base}$$

The union of all Π_i's leaves, for $i \leq n$, is the set of the Π's leaves. Hence, by the form of $\Pi\amalg$, (1) is a deduction in **FB** of $\Theta \longrightarrow C$. □

Notice that Theorem 3 holds also for any goal tree $\amalg$ in $\Pi\amalg$ and for any matching line $\not\amalg = \{A_1 \overset{?}{\longleftarrow} \Gamma_1,...,A_n \overset{?}{\longleftarrow} \Gamma_n\}$ in $\amalg$. The intuitive meaning of Theorem 3 depends somewhat on the choice of the matching line. A matching line in a sequent tree produces an introduction of the *gtt*-rule; this is the case showed directly by Theorem 3. While a matching line in a goal tree implies a "shifting" of the existing *gtt* into the deduction itself. Theorem 3, with some slight additions, is fundamental for the sequel and gives a result for system **FB** that is equivalent to say that an inversion of the direction of reasoning can be carried out at any time and anywhere in a proof of a theorem. In this sense, the deductions in **FB** extend the way to construct proofs in the Gentzen's systems of natural deduction.

Both Theorem 2 and Theorem 3 do not tell us anything about how to construct in **FB** a "good" proof of a theorem. We may ask whether it is possible to transform every bidirectional deduction to an equivalent "normal" deduction which proceeds, so to say, directly, only by reducing the complexity of the succedent formula of both sequents and goals in the deduction. Some ideas are given in [2]. Because of its particular interest as well as some open points, however, the reduction of bidirectional deductions to normal form seems somewhat more difficult and it is not discussed here.

4 Related work

As far as we know, our formalization to forward and backward reasoning has never been proposed before. However, some comparisons with existing AI systems can be made.

AI research has produced many applications of forward and backward reasoning systems. The idea of using forward and backward reasoning for proving theorems was pioneered in AI by A. Newell and H.A. Simon ([17, 18]) and, successively, by Nilsson ([19]). An attempt towards combining forward and backward reasoning in expert systems was also made by Ligęza [13]. Systems of the kind presented in these papers are often called *rule-based deduction systems*, or also *production systems*, to emphasize the importance of using rules to make deductions.

In other work, more directly related to automated theorem proving, (see for instance [12, 7, 5, 26]), only one direction of reasoning is formalized inside a logical system, while the other direction is left implicit in the control. In particular, the *intercalation calculi* defined by Sieg [26] (see also [6]) represent only forward reasoning, and give to both facts and goals a single formal structure.

A third group of systems was developed for studying cognitive aspects of bidirectional reasoning (beside [17, 18], interesting systems was developed in [15] and [25]). In these systems, facts and goals are represented, as well as a set of "cognitively valid" - as the authors shall say in their papers - forward and backward rules of inference. As a consequence, these systems take the form of a computer model of reasoning, by forcing the psychological validity of arguments; on the other hand, they are sometimes theoretically incomplete (see for instance [25]).

However, all the systems proposed so far, are not based on a formal notion of consequence relation to explain bidirectional reasoning. In particular, no formal notion of matching between facts and goals is given.

5 Conclusions

In this paper, we have presented a formal system **FB**, such that:

(a) forward and backward reasoning is well represented by *facts*, *goals*, and *consequence relations*. The matching between facts and goals, or forward and backward parts of deductions, is also formalized. In this respect, the notion of *matching line* presented in this paper is totally innovative. As an important consequence of these facts, the system developed has been correct and complete.

(b) forward, backward and bidirectional reasoning (everyone based on the presented way of formalization and inference rules), are formally equivalent, i.e. any proof which can be found by reasoning forwards (or backwards) can also be found by reasoning in a bidirectional way, and vice versa. The equivalence is formal since, in the realistic case of a proof construction, it may turn out that a theorem can be proved in a bidirectional way easier than reasoning in only one direction. Thus, what has usually been considered as relevant control knowledge for reasoning systems, i.e. the *direction of reasoning*, is embedded into the inference rules itself. This approach seems a step towards the development of interactive theorem provers, `GETFOL` is a first example, able to provide the user with the primitive tools to construct both efficient and domain independent strategies, or *tactics*, to guide the search for a proof of a theorem;

(c) bidirectional deductions are natural, i.e. they extend the natural way to construct proofs in the Gentzen's systems of natural deduction.

Acknowledgments

The results of this paper are part of the M.Sc. Thesis [1] developed by the first author under the supervision of the second author. Some early ideas were developed in some unpublished notes by Fausto Giunchiglia, Paolo Traverso and Alessandro Cimatti. We thank the Mechanized Reasoning Group at IRST. In particular, we are indebted to Luciano Serafini and Paolo Traverso for their useful suggestions and discussions, and for reviewing a first draft of this paper.

References

[1] A. Agostini. Logiche naturali per la meccanizzazione del ragionamento. Technical Report Thesis 9212-15, IRST, Trento, Italy, 1992. Also M.Sc. Thesis, University of Milan, Milan, Italy, 1992.

[2] A. Agostini. Bidirectional Natural Deduction. *AI*IA Notizie*, 4, 1993.

[3] B. G. Bara. *Scienza cognitiva - Un approccio evolutivo alla simulazione della mente.* Bollati Boringhieri, 1990.

[4] W.W. Bledsoe. Non-resolution theorem proving. *Artificial Intelligence*, 9:1–35, 1977.

[5] J. B. Bulnes. GOAL - A Goal Oriented Command Language for Interactive Proof Costructions. Technical Report STAN-CS-79-743, Computer Science Department, Stanford University, 1979.

[6] S. Cittadini. Intercalation Calculus for Intuitionistic Propositional Logic. Technical Report CMU-PHIL-29, Carnegie Mellon University, Pittsburgh, 1992.

[7] R.L. Constable, S.F. Allen, H.M. Bromley, et al. *Implementing Mathematics with the NuPRL Proof Development System.* Prentice Hall, 1986.

[8] D. Duffy. *Principles of Automated Theorem Proving.* John Wiley & Sons, New York, 1991.

[9] G. Gentzen. The consistency of elementary number theory. In M.E. Szabo, editor, *The collected papers of Gerhard Gentzen*, pages 150–155. North Holland Publishing Company, 1969.

[10] G. Gentzen. Investigations into logical deduction. In M.E. Szabo, editor, *The collected papers of Gerhard Gentzen*, pages 68–128. North Holland Publishing Company, 1969.

[11] F. Giunchiglia. The GETFOL Manual - GETFOL version 1. Technical Report 92-0010, DIST - University of Genova, Genoa, Italy, 1992.

[12] M.J. Gordon, A.J. Milner, and C.P. Wadsworth. *Edinburgh LCF - A mechanized logic of computation*, volume 78 of *Lecture Notes in Computer Science.* Springer Verlag, 1979.

[13] A. Ligęza. An expert systems approach to analysis and control in certain complex systems. In *Preprints Fourth IFAC/IFIP Symposium on Software for Computer Control SOCOCO'86*, pages 147–152, Graz, Austria, 1986.

[14] A. Ligęza. Dynamic backward reasoning systems. *Artificial Intelligence*, (43):127–152, 1990.

[15] J. Martins, D. McKay, and S. Shapiro. Bidirectional inference. Technical Report 174, Department of Computer Science, State University of New York, Buffalo, NY, 1981.

[16] A. Nevins. A human oriented logic for automatic theorem proving. *ACM*, (21):606–621, 1974.

[17] A. Newell, J.C. Shaw, and H.A. Simon. Empirical explorations of the logic theory machine. In Fiegenbaum and Feldman, editors, *Computers & Thought*, pages 134–152. McGraw-Hill, 1963.

[18] A. Newell and H.A. Simon. *Human Problem Solving.* Prentice-Hall, 1972.

[19] N.J. Nilsson. *Principles of Artificial Intelligence.* Tioga Publishing Co., 1980.

[20] L. Paulson. Interactive Theorem Proving with Cambridge LCF: A User's Manual. Technical Report 80, Computer Laboratory, University of Cambridge, 1985.

[21] L. Paulson. The Foundation of a Generic Theorem Prover. *Journal of Automated Reasoning*, 5:363–396, 1989.

[22] G. Polya. *How to Solve It.* Princeton University Press, Princeton, NJ, 1945.

[23] G. Polya. *Mathematical Discovery: On Understanding, Learning and Teaching Problem Solving.* John Wiley & Sons, 1962.

[24] D. Prawitz. *Natural Deduction - A proof theoretical study.* Almquist and Wiksell, Stockholm, 1965.

[25] L. J. Rips. Cognitive processes in propositional reasoning. *Psychological Review*, (90):38–71, 1983.

[26] W. Sieg. Mechanism and Search : Aspects of Proof Theory. Technical Report CMU-PHIL-30, Carnegie Mellon University, Pittsburgh, 1992.

[27] D. Solow. *How to read and do proofs.* John Wiley & Sons, New York, 1982.

[28] R.W. Weyhrauch. Prolegomena to a Theory of Mechanized Formal Reasoning. *Artificial Intelligence*, 13(1):133–176, 1980.