Inhabitants of Intuitionistic Implicational Theorems

Katalin Bimbó(✉)

Department of Philosophy, University of Alberta, Edmonton, AB T6G 2E7, Canada
bimbo@ualberta.ca
http://www.ualberta.ca/~bimbo

Abstract. The aim of this paper is to define an algorithm that produces a *combinatory inhabitant* for an implicational theorem of intuitionistic logic *from a proof in a sequent calculus*. The algorithm is applicable to standard proofs, that exist for every theorem, moreover, non-standard proofs can be straightforwardly transformed into standard ones. We prove that the resulting combinator inhabits the simple type for which it is generated.

Keywords: Combinator · Curry–Howard correspondence
Intuitionistic logic · Sequent calculus

1 Introduction

A connection between implicational formulas and combinatory and λ-terms has been well known for more than half a century. A link is usually established between a natural deduction system and λ-terms, or axioms and combinators (or λ-terms). Moreover, the natural deduction system is usually defined as a type-assignment calculus.

Our interest is in *sequent calculi*, on one hand, and in *combinatory terms*, on the other. A natural deduction type-assignment system with combinators very closely resembles an axiomatic calculus. Neither of the latter two kinds of proof systems is highly suitable for decidability proofs, nor they provide much control over the shape of proofs. There have been certain attempts to use sequent calculi in type assignment systems. However, we believe that no algorithm has been defined so far for the extraction of combinatory inhabitants for intuitionistic theorems from sequent calculus proofs thereof.[1]

The next section briefly overviews some other approaches that link λ-terms and sequent calculi. Section 3 introduces a sequent calculus for $J^{\boldsymbol{T}}_{\rightarrow}$. This sequent calculus falls into the line of sequent calculi that were initiated by the non-associative Lambek calculus. The bulk of the paper, Sect. 4 is devoted to the description of the extraction algorithm, which has several stages, and then, to the proof of its correctness. We conclude the paper with a few remarks.

[1] The paper [9] deals with generating HRM^n_n terms and combinatory inhabitants from proofs of $T_{\rightarrow}$ theorems.

L. S. Moss et al. (Eds.): WoLLIC 2018, LNCS 10944, pp. 1–24, 2018.
https://doi.org/10.1007/978-3-662-57669-4_1

2 Some Other Approaches

We should note that there have been earlier attempts to define a connection between sequent calculus proofs and λ-terms. Girard's idea—when he introduced his sequent calculus LJT—was to add just a minimal amount of structure to antecedents which are otherwise sets of formulas. A special location (the so-called "stoup") is reserved for a single formula, but it may be left empty. All the rules of LJT, except the rule called "mid-cut," affect a formula in the special location of at least one premise. The effect of this focusing on a formula in most of the proof steps is the *exclusion of many proofs* that could be constructed in a more usual formulation of a sequent calculus for implicational intuitionistic logic. *Adding structure to sets* is a step in the right direction in our view, but LJT does not go far enough. (See [17] for a detailed presentation of LJT.)

Another approach is to define a type-assignment system in a sequent calculus form. Barendregt and Ghilezan in [2] introduced λL (and λL^{cf}) to find a correspondence between λ-terms and implicational intuitionistic theorems. It is natural to start with λN in a horizontal presentation (which resembles sequent calculus notation). Then the antecedent is a context (or a set of type-assignments to variables) in which a certain type-assignment to a (possibly) complex λ-term can be derived. The core question is how the left introduction rule for $\rightarrow$ should look like. [2] combines *substitution* and *residuation*: given that $M\colon \mathcal{A}$ and $x\colon \mathcal{B}$, if there is a $y\colon \mathcal{A} \rightarrow \mathcal{B}$, then yM can replace x. Combinators were originally introduced by Schönfinkel in [20] to eliminate the need for substitution, which is a complex operation in a system with variable binding operations (such as λ or $\forall$). Thus reintroducing substitution seems to be a step backward. A more serious complaint about λL is that the $\rightarrow$ introduction rule essentially requires the λ operator. The λ could be replaced by the λ^*, but the latter would have to be deciphered on a case-by-case basis (i.e., by running a λ^* algorithm on every concrete term). In sum, this approach differs from ours in that it uses type-assignments and λ's.

Arguably, Girard's as well as Barendregt and Ghilezan's calculi are directly motivated by a natural deduction calculus (such as NJ) for intuitionistic logic. On the other hand, a connection between *sequent calculus rules* and *combinators* was discovered by Curry; see, for instance, his [11]. It is easy to see that $\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{A}$, the principal type schema of the combinator K cannot be proved without some kind of thinning rule on the left. However, such informal observations about rules and matching combinators cannot be made precise as long as sequents are based on sequences or even sets of formulas, which is inherent both in [17] and in [2].

3 A New Sequent Calculus for $J_{\rightarrow}$

Function application is not an associative operation—unlike conjunction and disjunction are in intuitionistic logic. The analogy between combinators and structural rules in a sequent calculus cannot be made precise without replacing the usual associative structural connective with a non-associative one. Of course, the use of a structural connective that is not stipulated to be associative is by no means new; Lambek's [18] already includes such a connective. Combinators have been explicitly incorporated into a sequent calculus by Meyer in [19].[2]

Taking the reduction patterns of combinators seriously helped to formulate sequent calculi for the relevance logic of positive ticket entailment, $\mathbf{T}_+$. Giambrone's calculi in [15,16] utilized insights concerning B, B' and W in the form of structural rules. Bimbó [3] defined sequent calculi for $\mathbf{T}^{\boldsymbol{t}}_{\to}$ and $\mathbf{T}^{\circ\boldsymbol{t}}_{+}$ (among other logics), in which special rules involving $\boldsymbol{t}$ are added. It is easy to show that $\boldsymbol{t}$ is a left identity for fusion (i.e., intensional conjunction) in $\mathbf{T}$, hence, a special instance of the thinning rule may be added in a sequent calculus. However, the latter rule is not the only one needed that involves $\boldsymbol{t}$, if we algebraize $\mathbf{T}^{\circ\boldsymbol{t}}_{+}$ with an equivalence relation defined from implicational theorems. The introduction of additional structural rules specific to $\boldsymbol{t}$, on the other hand, requires us to refine the proof of the cut theorem, which could otherwise proceed by double induction for a multicut rule or by triple induction for the single cut rule.[3]

A connection between K, C and W, on one hand, and (left) structural rules, on the other hand, was noted by Curry in [10,11], who is probably the first to use combinators in the labels of the thinning, the permutation and the contraction rules. The operational rules of a sequent calculus leave their trace in a sequent in the form of a formula with a matching connective or quantifier; however, structural rules leave more subtle vestige behind. (This is not quite true for the thinning rules, which add a formula into a sequent. However, the formula is arbitrary, which means that there is no single kind of a formula that indicates that an application of a thinning rule has taken place.) *Structurally free logics* that were invented by Dunn and Meyer in [13] introduce a combinator. The "formulas-as-types" slogan is often used to connect typable combinators or λ-terms and (implicational) formulas. "Combinators-as-formulas" could be a similar catchphrase for structurally free logics. To illustrate the idea, we use *contraction* and the combinator M. The axiom for M is $\mathsf{M}x \rhd xx$.[4]

$$\frac{\mathfrak{A}[\mathfrak{B};\mathfrak{B}] \vdash \mathcal{C}}{\mathfrak{A}[\mathfrak{B}] \vdash \mathcal{C}}\ (\mathsf{M}\vdash) \qquad\qquad \frac{\mathfrak{A}[\mathfrak{B};\mathfrak{B}] \vdash \mathcal{C}}{\mathfrak{A}[\mathsf{M};\mathfrak{B}] \vdash \mathcal{C}}\ (\mathsf{M}\vdash)$$

The $(\mathsf{M}\vdash)$ rule on the left is contraction (on structures), whereas the other $(\mathsf{M}\vdash)$ rule is the *combinatory rule*. For the latter, we could omit the label $(\mathsf{M}\vdash)$,

[2] This paper was known for a long time only to a select group of relevance logicians, but nowadays, it is freely available online.

[3] See [1] for information on relevance logics, in general. Dunn [12] introduced a sequent calculus for $\mathbf{R}_+$, whereas [7,8] introduced and used a sequent calculus for $\mathbf{R}^{\boldsymbol{t}}_{\to}$. See also [6] for a comprehensive overview of a variety of sequent calculi.

[4] In combinatory terms, parentheses are omitted by assuming association to the left; for example, $xz(yz)$ is a shorthand for $((xz)(yz))$. For more on combinatory logic, see for example [4]. Brackets in a sequent indicate a hole in a structure in the antecedent, with the result of a replacement put into brackets in the lower sequent.

because M appears in the sequent itself. Combinatory rules supplant structural rules, hence, the label "structurally free." The implicational types of typable combinators are provable in structurally free logic—with an appropriately chosen set of combinatory rules. The following is a pointer toward this (with some of the more obvious steps omitted or compressed).[5]

$$\cfrac{\cfrac{\cfrac{\vdots \atop \mathcal{D} \to \mathcal{A} \to \mathcal{B} \to \mathcal{C}; \mathcal{D}; \mathcal{A}; \mathcal{B} \vdash \mathcal{C}}{\mathsf{C}; (\mathcal{D} \to \mathcal{A} \to \mathcal{B} \to \mathcal{C}; \mathcal{D}); \mathcal{B}; \mathcal{A} \vdash \mathcal{C}}}{\mathsf{B}; \mathsf{C}; \mathcal{D} \to \mathcal{A} \to \mathcal{B} \to \mathcal{C}; \mathcal{D}; \mathcal{B}; \mathcal{A} \vdash \mathcal{C}}}{\mathsf{B}; \mathsf{C} \vdash (\mathcal{D} \to \mathcal{A} \to \mathcal{B} \to \mathcal{C}) \to (\mathcal{D} \to \mathcal{B} \to \mathcal{A} \to \mathcal{C})} \ {\vdash}{\to}\text{'s}$$

The combinators B and C are the standard ones with axioms $\mathsf{B}xyz \rhd x(yz)$ and $\mathsf{C}xyz \rhd xzy$. The principal (simple) type schema of BC is $(\mathcal{A} \to \mathcal{B} \to \mathcal{C} \to \mathcal{D}) \to \mathcal{A} \to \mathcal{C} \to \mathcal{B} \to \mathcal{D}$. In structurally free logics, other (than simple) types can be considered; for instance, $\mathsf{M} \vdash \mathcal{A} \to (\mathcal{A} \circ \mathcal{A})$ is easily seen to be provable, when $\circ$ (fusion) is included with the usual rules. However, our aim is to find a way to obtain a combinator that inhabits a simple type from a sequent calculus proof—without making combinators into formulas and without introducing a type assignment system.

There are several sequent calculi for intuitionistic logic including its various fragments, but most often they use an associative comma in the antecedent (and possibly in the succedent too).[6] We define a *new sequent calculus* $LJ^{\boldsymbol{T}}_{\to}$ for the implicational fragment of intuitionistic logic with the truth constant $\boldsymbol{T}$.

Definition 1. The *language* of $LJ^{\boldsymbol{T}}_{\to}$ contains a denumerable sequence of propositional variables $\langle p_i \rangle_{i \in \omega}$, the arrow connective $\to$ and $\boldsymbol{T}$.

The set of *well-formed formulas* is inductively defined from the set of propositional variables with $\boldsymbol{T}$ added, by the clause: "If $\mathcal{A}$ and $\mathcal{B}$ are well-formed formulas, then so is $(\mathcal{A} \to \mathcal{B})$."

The set of *structures* is inductively defined from the set of well-formed formulas by the clause: "If $\mathfrak{A}$ and $\mathfrak{B}$ are structures, then so is $(\mathfrak{A}; \mathfrak{B})$."

A *sequent* is a structure followed by $\vdash$, and then, by a well-formed formula.

Remark 1. We will occasionally use some (notational) conventions. For example, we might omit some parentheses as mentioned previously, or we might call well-formed formulas simply formulas. Again, we will use [ ] for a context (or hole) in a structure, as usual. The use of a semi-colon as the structural connective is motivated by our desire to consider the structural connective as an analog of function application.

5 Parentheses in the antecedent of a sequent are restored as in combinatory terms, but parentheses in simple types are restored by association to the right.

6 See [14] as well as [10,11] for such calculi.

Remark 2. We already mentioned M, B and C together with their axioms. Further combinators that we will use in this paper are I, B', T, W, K and S. Their axioms, respectively, are $\mathsf{I}x \rhd x$, $\mathsf{B}'xyz \rhd y(xz)$, $\mathsf{T}xy \rhd yx$, $\mathsf{W}xy \rhd xyy$, $\mathsf{K}xy \rhd x$ and $\mathsf{S}xyz \rhd xz(yz)$.

Definition 2. The *sequent calculus* $LJ^{\boldsymbol{T}}_{\rightarrow}$ comprises the axiom and the rules listed below.

$$\mathcal{A} \vdash \mathcal{A} \quad (\mathsf{I})$$

$$\frac{\mathfrak{A} \vdash \mathcal{A} \quad \mathfrak{B}[\mathcal{B}] \vdash \mathcal{C}}{\mathfrak{B}[\mathcal{A} \rightarrow \mathcal{B}; \mathfrak{A}] \vdash \mathcal{C}} \; ({\rightarrow}\vdash) \qquad \frac{\mathfrak{A}; \mathcal{A} \vdash \mathcal{B}}{\mathfrak{A} \vdash \mathcal{A} \rightarrow \mathcal{B}} \; (\vdash{\rightarrow})$$

$$\frac{\mathfrak{A}[\mathfrak{B}; (\mathfrak{C}; \mathfrak{D})] \vdash \mathcal{A}}{\mathfrak{A}[(\mathfrak{B}; \mathfrak{C}); \mathfrak{D}] \vdash \mathcal{A}} \; (\mathsf{B}\vdash) \qquad \frac{\mathfrak{A}[\mathfrak{C}; (\mathfrak{B}; \mathfrak{D})] \vdash \mathcal{A}}{\mathfrak{A}[(\mathfrak{B}; \mathfrak{C}); \mathfrak{D}] \vdash \mathcal{A}} \; (\mathsf{B}'\vdash)$$

$$\frac{\mathfrak{A}[(\mathfrak{B}; \mathfrak{D}); \mathfrak{C}] \vdash \mathcal{A}}{\mathfrak{A}[(\mathfrak{B}; \mathfrak{C}); \mathfrak{D}] \vdash \mathcal{A}} \; (\mathsf{C}\vdash) \qquad \frac{\mathfrak{A}[\mathfrak{C}; \mathfrak{B}] \vdash \mathcal{A}}{\mathfrak{A}[\mathfrak{B}; \mathfrak{C}] \vdash \mathcal{A}} \; (\mathsf{T}\vdash)$$

$$\frac{\mathfrak{A}[(\mathfrak{B}; \mathfrak{C}); \mathfrak{C}] \vdash \mathcal{A}}{\mathfrak{A}[\mathfrak{B}; \mathfrak{C}] \vdash \mathcal{A}} \; (\mathsf{W}\vdash) \qquad \frac{\mathfrak{A}[\mathfrak{B}; \mathfrak{B}] \vdash \mathcal{A}}{\mathfrak{A}[\mathfrak{B}] \vdash \mathcal{A}} \; (\mathsf{M}\vdash)$$

$$\frac{\mathfrak{A}[\mathfrak{B}] \vdash \mathcal{A}}{\mathfrak{A}[\mathfrak{B}; \mathfrak{C}] \vdash \mathcal{A}} \; (\mathsf{K}\vdash) \qquad \frac{\mathfrak{A}[\mathfrak{B}] \vdash \mathcal{A}}{\mathfrak{A}[\boldsymbol{T}; \mathfrak{B}] \vdash \mathcal{A}} \; (\boldsymbol{T}\vdash)$$

Remark 3. The two rules $(\mathsf{T}\vdash)$ and $(\boldsymbol{T}\vdash)$ have a somewhat similar label, but they are clearly different. Both combinators and sentential constants have had their practically standard notation for many years, and so both T's are entrenched. From the point of view of structurally free logics, however, $(\boldsymbol{T}\vdash)$ is like $(\mathsf{I}\vdash)$, a rule that simply inserts the identity combinator on the left of a structure.

The rules may be grouped into operational and structural rules. The former group consists of $({\rightarrow}\vdash)$, $(\vdash{\rightarrow})$ and $(\boldsymbol{T}\vdash)$. The structural rules are all labeled with combinators, that is, they are $(\mathsf{B}\vdash)$, $(\mathsf{B}'\vdash)$, $(\mathsf{C}\vdash)$, $(\mathsf{T}\vdash)$, $(\mathsf{W}\vdash)$, $(\mathsf{M}\vdash)$ and $(\mathsf{K}\vdash)$.

Remark 4. The structural rules clearly suffice for intuitionistic logic; indeed, they all together constitute a redundant set of structural rules. The rules $(\mathsf{T}\vdash)$, $(\mathsf{M}\vdash)$ and $(\mathsf{K}\vdash)$ are permutation, contraction and thinning, moreover, they *operate on structures*, not formulas; thus, in a sense, they are more powerful than the usual versions of these rules. (Of course, in a usual sequent calculus for intuitionistic logic, the notion of a structure cannot be replicated. However, contraction, permutation and thinning could affect a sequence of formulas at once, because those versions of the rules are admissible.) The semi-colon is not associative, but $(\mathsf{B}\vdash)$ is the left-to-right direction of associativity. The other direction of associativity can be obtained by several applications of $(\mathsf{T}\vdash)$ and an application of $(\mathsf{B}\vdash)$. Finally, we note that each of the combinators that appear in the labels to the rules are *proper combinators* (as it is understood in combinatory logic), and they are definable from the combinatory base $\{\, \mathsf{S}, \mathsf{K} \,\}$.

The notion of a *proof* is the usual one for sequent calculi. $\mathcal{A}$ is a *theorem* iff $\boldsymbol{T} \vdash \mathcal{A}$ has a proof. The $\boldsymbol{T}$ cannot be omitted, because $\vdash \mathcal{A}$ is not a sequent in $LJ^{\boldsymbol{T}}_{\rightarrow}$, let alone it has a proof.

The set of rules is not minimal in the sense that some rules could be omitted, because they are derivable from others. As an example, $(\boldsymbol{T}\vdash)$ is easily seen to be derivable by first applying $(\mathsf{K}\vdash)$ with $\boldsymbol{T}$ in place of $\mathfrak{C}$, and then, applying $(\mathsf{T}\vdash)$ to permute $\boldsymbol{T}$ to the left of $\mathfrak{B}$. The abundance of rules facilitates shorter proofs and finding shorter inhabitants.

$J_{\rightarrow}$ can be thought of as an "S–K calculus," because the principal type schemas of those combinators suffice as axioms (with detachment as a rule) for an axiom system. Similarly, in a combinatory type-assignment system, starting with type schemas assigned to S and K allows us to prove all the theorems of $J_{\rightarrow}$ with a combinator attached to each. A widely known theorem is that S and K are sufficient to define any function—in the sense of combinatorial definability.

However, proofs in a sequent calculus are built differently than in an axiomatic calculus, which is one of the advantages sequent calculi supply. Thus, while we included a rule labeled $(\mathsf{K}\vdash)$, there is no $(\mathsf{S}\vdash)$ rule. The three left structural rules in LJ emulate (to some extent) the effect of the combinators K, C and W. In LJ, both $(C\vdash)$ (the left permutation rule) and $(W\vdash)$ (the left contraction rule) would have to be used to prove the type of S. It is not possible to define S from merely C and W, because neither has an associative effect.

Example 5. The principal type schema of S is $(\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow \mathcal{A} \rightarrow \mathcal{C}$. The following is a proof of this formula in $LJ^{\boldsymbol{T}}_{\rightarrow}$, which shows the usefulness of some of the redundant structural rules—permitting several steps to be collapsed into single steps.

$$
\begin{prooftree}
\AxiomC{$\mathcal{A} \vdash \mathcal{A}$}
\AxiomC{$\mathcal{A} \vdash \mathcal{A}$}
\AxiomC{$\mathcal{B} \vdash \mathcal{B}$}
\AxiomC{$\mathcal{C} \vdash \mathcal{C}$}
\RightLabel{$\rightarrow\vdash$}
\BinaryInfC{$\mathcal{B} \rightarrow \mathcal{C}; \mathcal{B} \vdash \mathcal{C}$}
\RightLabel{$\rightarrow\vdash$}
\BinaryInfC{$(\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}; \mathcal{A}); \mathcal{B} \vdash \mathcal{C}$}
\RightLabel{$\mathsf{C}\vdash$}
\UnaryInfC{$(\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}; \mathcal{B}); \mathcal{A} \vdash \mathcal{C}$}
\RightLabel{$\rightarrow\vdash$}
\BinaryInfC{$(\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}; (\mathcal{A} \rightarrow \mathcal{B}; \mathcal{A})); \mathcal{A} \vdash \mathcal{C}$}
\RightLabel{$\mathsf{B}\vdash$}
\UnaryInfC{$((\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}; \mathcal{A} \rightarrow \mathcal{B}); \mathcal{A}); \mathcal{A} \vdash \mathcal{C}$}
\RightLabel{$\mathsf{W}\vdash$}
\UnaryInfC{$(\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}; \mathcal{A} \rightarrow \mathcal{B}); \mathcal{A} \vdash \mathcal{C}$}
\RightLabel{$\boldsymbol{T}\vdash$}
\UnaryInfC{$((\boldsymbol{T}; \mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}); \mathcal{A} \rightarrow \mathcal{B}); \mathcal{A} \vdash \mathcal{C}$}
\RightLabel{$\vdash\rightarrow$'s}
\UnaryInfC{$\boldsymbol{T} \vdash (\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}) \rightarrow (\mathcal{A} \rightarrow \mathcal{B}) \rightarrow \mathcal{A} \rightarrow \mathcal{C}$}
\end{prooftree}
$$

Remark 6. We can show easily that the $(\mathsf{S}\vdash)$ rule is a derived rule of $LJ^{\boldsymbol{T}}_{\rightarrow}$.

$$
\begin{prooftree}
\AxiomC{$\mathfrak{A}[\mathfrak{B}; \mathfrak{D}; (\mathfrak{C}; \mathfrak{D})] \vdash \mathcal{A}$}
\RightLabel{$(\mathsf{T}\vdash)$}
\UnaryInfC{$\mathfrak{A}[\mathfrak{B}; \mathfrak{D}; (\mathfrak{D}; \mathfrak{C})] \vdash \mathcal{A}$}
\RightLabel{$(\mathsf{B}\vdash)$}
\UnaryInfC{$\mathfrak{A}[\mathfrak{B}; \mathfrak{D}; \mathfrak{D}; \mathfrak{C}] \vdash \mathcal{A}$}
\RightLabel{$(\mathsf{W}\vdash)$}
\UnaryInfC{$\mathfrak{A}[\mathfrak{B}; \mathfrak{D}; \mathfrak{C}] \vdash \mathcal{A}$}
\RightLabel{$(\mathsf{C}\vdash)$}
\UnaryInfC{$\mathfrak{A}[\mathfrak{B}; \mathfrak{C}; \mathfrak{D}] \vdash \mathcal{A}$}
\end{prooftree}
$$

Our goal is to find combinatory inhabitants. We hasten to note that despite the combinatory labels in the proof in Example 5 and in the derived ($\mathsf{S}\vdash$) rule, we do not have a combinatory inhabitant on the horizon. WBC is a fancy (and complex) ternary identity combinator, whereas CBW is not even a proper combinator. CWBT is a quaternary combinator that is often denoted by D, but it is not a duplicator and has no permutative effect. $\mathsf{TBWC} \rhd_1 \mathsf{WBC}$, that is, TBWC is an even longer ternary identity combinator. Of course, there are further combinatory terms that can be formed from one occurrence of each of W, B and C, and S is definable from these three combinators. (For example, it is easy to verify that $\mathsf{B(BW)(BC(BB))}$ defines S.) However, we do not need to consider all the combinators with exactly one W, B and C to see the point we are emphasizing here, namely, that we cannot simply convert the labels for the rules into a combinatory inhabitant.

The lack of a direct match between a definition of S and the rules that are used in the proof of the principal type schema of S is perhaps not very surprising, because the ($\to\vdash$) rule bears a slight resemblance to the affixing rule, which is a composite of the suffixing and prefixing rules. The latter are connected to B' and B, respectively.

The cut rule is of paramount interest in any sequent calculus. $LJ^{\boldsymbol{T}}_{\to}$ is a single right-handed calculus, but it has structured antecedents; therefore, we formulate the cut rule as follows.

$$\frac{\mathfrak{A} \vdash \mathcal{C} \qquad \mathfrak{B}[\mathcal{C}] \vdash \mathcal{B}}{\mathfrak{B}[\mathfrak{A}] \vdash \mathcal{B}}\ \text{cut}$$

Theorem 3. *The cut rule is* admissible *in* $LJ^{\boldsymbol{T}}_{\to}$.

Proof. The proof is a standard multi-inductive proof. Here we take as the parameters of the induction the rank of the cut and the degree of the cut formula. We include only some sample steps.

1. First, let us define a multi-cut rule. We will indicate one or more occurrences of a structure $\mathcal{C}$ in the antecedent of the right premise by $[\mathcal{C}]\cdots[\mathcal{C}]$, with the result of replacement indicated as before.

$$\frac{\mathfrak{A} \vdash \mathcal{C} \qquad \mathfrak{B}[\mathcal{C}]\cdots[\mathcal{C}] \vdash \mathcal{B}}{\mathfrak{B}[\mathfrak{A}]\cdots[\mathfrak{A}] \vdash \mathcal{B}}\ \text{multi-cut}$$

The multi-cut rule encompasses the cut rule by permitting a single occurrence of $\mathcal{C}$. On the other hand, multi-cut is a derived rule, because as many applications of cut (with the same left premise) as the number of the selected occurrences of $\mathcal{C}$ yields the lower sequent of multi-cut.[7]

2. The only connective is $\to$, and if the rank of the cut formula is 2, then an interesting case is when the premises are by the ($\to\vdash$) and ($\vdash\to$) rules. The cut is replaced by two cuts, each of lower degree; the latter justifies the change in the

[7] The multi-cut rule is not the same rule as the mix rule, and its use in the proof of the cut theorem is not necessary. It is possible to use triple induction with a parameter called contraction measure as in [5], for example.

proof. The given and the modified proofs are as follows. Due to the stipulation about the rank, this case has a single occurrence of the cut formula in the right premise.[8]

$$\dfrac{{\scriptstyle\vdash\to}\ \dfrac{\mathfrak{A};\mathcal{A}\vdash\mathcal{B}}{\mathfrak{A}\vdash\mathcal{A}\to\mathcal{B}} \qquad \dfrac{\mathfrak{B}\vdash\mathcal{A}\quad \mathfrak{C}[\mathcal{B}]\vdash\mathcal{C}}{\mathfrak{C}[\mathcal{A}\to\mathcal{B};\mathfrak{B}]\vdash\mathcal{C}}\ {\scriptstyle\to\vdash}}{\mathfrak{C}[\mathfrak{A};\mathfrak{B}]\vdash\mathcal{C}}\ (\text{cut})$$

$$\dfrac{(\text{cut})\ \dfrac{\mathfrak{B}\vdash\mathcal{A}\quad \mathfrak{A};\mathcal{A}\vdash\mathcal{B}}{\mathfrak{A};\mathfrak{B}\vdash\mathcal{B}} \qquad \mathfrak{C}[\mathcal{B}]\vdash\mathcal{C}}{\mathfrak{C}[\mathfrak{A};\mathfrak{B}]\vdash\mathcal{C}}\ (\text{cut})$$

3. To illustrate the interaction of the cut and a structural rule, we detail the situation when the right rank of the cut is >1, and the right premise is by $(\mathsf{B}'\vdash)$. It is important to note that the structural rules are applicable to structures. For a quick example to show this, let $\mathfrak{B}, \mathfrak{C}$ and $\mathfrak{D}$ be formulas, and let $\mathfrak{C}$ be an occurrence of the cut formula $\mathcal{C}$, in the proof below. If $\mathfrak{E}$ were not an atomic structure (which is possible), then the transformation of the proof would not result in a proof, if the structural rule could not be applied to complex structures.

$$\dfrac{\mathfrak{E}\vdash\mathcal{C} \qquad \dfrac{\mathfrak{A}[\mathfrak{C};(\mathfrak{B};\mathfrak{D})][\mathcal{C}]\cdots[\mathcal{C}]\vdash\mathcal{A}}{\mathfrak{A}[\mathfrak{B};\mathfrak{C};\mathfrak{D}][\mathcal{C}]\cdots[\mathcal{C}]\vdash\mathcal{A}}\ (\mathsf{B}'\vdash)}{\mathfrak{A}[\mathfrak{B};\mathfrak{C};\mathfrak{D}][\mathfrak{E}]\cdots[\mathfrak{E}]\vdash\mathcal{A}}\ (\text{multi-cut})$$

$$\dfrac{\dfrac{\mathfrak{E}\vdash\mathcal{C}\quad \mathfrak{A}[\mathfrak{C};(\mathfrak{B};\mathfrak{D})][\mathcal{C}]\cdots[\mathcal{C}]\vdash\mathcal{A}}{\mathfrak{A}[\mathfrak{C};(\mathfrak{B};\mathfrak{D})][\mathfrak{E}]\cdots[\mathfrak{E}]\vdash\mathcal{A}}\ (\text{multi-cut})}{\mathfrak{A}[\mathfrak{B};\mathfrak{C};\mathfrak{D}][\mathfrak{E}]\cdots[\mathfrak{E}]\vdash\mathcal{A}}\ (\mathsf{B}'\vdash)$$

We do not include the other cases here in order to keep our paper focused. ⸫

Remark 7. We mentioned that the $(\boldsymbol{T}\vdash)$ rule is like the $(\mathsf{I}\vdash)$ rule, which in turn is like the $(\boldsymbol{t}\vdash)$ rule. In some sequent calculi, especially, in sequent calculi for various *relevance logics*, it should be considered what happens when the cut formula is $\boldsymbol{t}$. The clearest way to prove the cut theorem is to use a separate induction to show that cuts on $\boldsymbol{t}$ are eliminable. In $LJ^{\boldsymbol{T}}_{\to}$, this problem does not arise at all, because the $(\boldsymbol{T}\vdash)$ rule is a special instance of the $(\mathsf{K}\vdash)$ rule.

Theorem 4. *If $\mathcal{A}$ is a theorem of $J_{\to}$, then $\mathcal{A}$ is a theorem of $LJ^{\boldsymbol{T}}_{\to}$.*

Proof. The outline of the proof is as follows. $\mathcal{A}\to\mathcal{B}\to\mathcal{A}$ is easily shown to be a theorem of $LJ^{\boldsymbol{T}}_{\to}$, and we have the proof in Example 5. If $\boldsymbol{T}\vdash\mathcal{A}$ and $\boldsymbol{T}\vdash\mathcal{A}\to\mathcal{B}$ are provable sequents, then $\boldsymbol{T}\vdash\mathcal{B}$ is a theorem, because from $\boldsymbol{T}\vdash\mathcal{A}\to\mathcal{B}$ we can obtain $\boldsymbol{T};\mathcal{A}\vdash\mathcal{B}$ by cut, and then, by another cut, we have $\boldsymbol{T};\boldsymbol{T}\vdash\mathcal{B}$, and by $(\mathsf{M}\vdash)$, we get $\boldsymbol{T}\vdash\mathcal{B}$. ⸫

[8] We omit ⋮'s from above the top sequents, which does not mean that the leaves are instances of the axiom; some of them are obviously not.

The converse of this claim is obviously false, simply because $\boldsymbol{T}$ is not in the language of $J_{\rightarrow}$. However, in intuitionistic logic all theorems are provably equivalent, because $\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{A}$ is a theorem. Hence, we can define $J^{\boldsymbol{T}}_{\rightarrow}$ by adding a single axiom to $J_{\rightarrow}$ (after extending its language). (A3) says that $\boldsymbol{T}$ is the top element of the word algebra of $LJ^{\boldsymbol{T}}_{\rightarrow}$.

(A3) $\mathcal{A} \rightarrow \boldsymbol{T}$

Theorem 5. *If $\mathcal{A}$ is a theorem of $LJ^{\boldsymbol{T}}_{\rightarrow}$, then $\mathcal{A}$ is a theorem of $J^{\boldsymbol{T}}_{\rightarrow}$.*

Proof. The proof is more elaborate than that of the previous theorem, but follows the usual lines for such proofs. Our sequent calculus can be extended by $\wedge$ with the usual rules (which originated in Gentzen's work). The resulting sequent calculus, $LJ^{\boldsymbol{T}}_{\overset{\wedge}{\rightarrow}}$ enjoys the admissibility of the cut rule. The proof of the cut theorem allows us to conclude that $LJ^{\boldsymbol{T}}_{\overset{\wedge}{\rightarrow}}$ is a conservative extension of $LJ^{\boldsymbol{T}}_{\rightarrow}$. The axiomatic calculus can be extended with $\wedge$ too, and that can be shown to be conservative, for example, by relying on semantics for $J^{\boldsymbol{T}}_{\overset{\wedge}{\rightarrow}}$ and $J^{\boldsymbol{T}}_{\rightarrow}$.

Once the framework is set up on both sides, we define a function that is applicable to sequents in $LJ^{\boldsymbol{T}}_{\overset{\wedge}{\rightarrow}}$, and yields a formula in the common language of these calculi. Lastly, we use an induction on the height of a proof in $LJ^{\boldsymbol{T}}_{\overset{\wedge}{\rightarrow}}$ to show that the axiom and the rules can be emulated in $J^{\boldsymbol{T}}_{\overset{\wedge}{\rightarrow}}$. (The detailed proof is quite lengthy, but straightforward; hence, we omit the details.) ⁖

REMARK 8. The constant $\boldsymbol{T}$ has a double role in $LJ^{\boldsymbol{T}}_{\rightarrow}$. First of all, it is an atomic formula, which can be a subformula of a more complex formula. However, we introduced this constant primarily to keep track of the structure of antecedents. There is no difficulty in determining whether $\boldsymbol{T}$ does or does not occur in a formula.

Corollary 6. *A formula $\mathcal{A}$, which contains no occurrences of $\boldsymbol{T}$, is a theorem of $LJ^{\boldsymbol{T}}_{\rightarrow}$ iff it is a theorem of $J_{\rightarrow}$.*

This claim follows from the three previous theorems and their proofs. Most importantly, the cut theorem guarantees that the *subformula property* holds. That is, if $\mathcal{A}$ is a formula, which occurs in a sequent in the proof of $\mathfrak{A} \vdash \mathcal{B}$, then $\mathcal{A}$ is a subformula of a formula in the sequent $\mathfrak{A} \vdash \mathcal{B}$. If $\boldsymbol{T}$ occurs in a proof of the theorem $\mathcal{A}$ (and it does by the definition of a theorem), then it is a subformula of a formula in $\boldsymbol{T} \vdash \mathcal{A}$. By stipulation, $\mathcal{A}$ is $\boldsymbol{T}$-free, therefore, the only occurrence of $\boldsymbol{T}$ in $\boldsymbol{T} \vdash \mathcal{A}$ is $\boldsymbol{T}$ itself.

As a result, we can consider $\boldsymbol{T}$-free theorems of $J^{\boldsymbol{T}}_{\rightarrow}$ in $LJ^{\boldsymbol{T}}_{\rightarrow}$, without any problems. All the occurrences of $\boldsymbol{T}$ will turn out to be ancestors of the $\boldsymbol{T}$ in the root sequent.

4 Standard Proofs, Cafs and an Inhabitant

Sequent calculi maintain the premises at each step in a derivation. Derivations in axiomatic calculi limit the manipulation of the premises to the addition of new premises. In natural deduction calculi, one can not only add new premises to a derivation, but it is possible to decompose premises and it is also possible to incorporate premises into more complex formulas (thereby eliminating them from among the premises).

Sequent calculi offer the most flexibility and the strictest control over the premises (as well as over the conclusions). Thinning on the left allows the addition of new premises, whereas the most typical way to eliminate a premise is via the implication introduction rule on the right-hand side. The possibility to tinker with the premises and carry them along from step to step while they are not affected by an application of a rule is advantageous.

However, exactly the mentioned features of sequent calculi lead to certain complications in the process of extracting combinatory inhabitants from proofs. For example, in a sufficiently large sequent, we might be able to apply rules to different formulas without interfering with the applicability of another rule. The combinatory counterpart of this *indeterminacy* is when the inhabitant cannot differentiate between proofs in which such pairs of applications of rules are permuted. Of course, the permutability of certain rules is a phenomenon that appears not only in $LJ^{\boldsymbol{T}}_{\rightarrow}$, but in other sequent calculi too. Still, with ; structuring antecedents, permutability may be less expected than in sequent calculi with flat sequents.

Another potential problem is that some theorems may have more than one proof. If the two proofs become subproofs in a larger proof, for example, by applications of the $(\rightarrow\vdash)$ rule, then the $(\mathsf{W}\vdash)$ or the $(\mathsf{M}\vdash)$ rule may be applied to contract two complex antecedents, which contain theorems that are (potentially) inhabited by distinct combinators. Contraction may be thought of as *identification*, but it is not possible to identify distinct combinators. There are ways to deal with this problem; we mention two of them.

First, we can apply an abstraction algorithm to obtain terms of the form $\mathsf{Z}_1 x_1 \ldots x_n$ and $\mathsf{Z}_2 x_1 \ldots x_n$. It is straightforward to further obtain $\mathsf{Z}_3 x_1 \ldots x_n$, which reduces to $\mathsf{Z}_1 x_1 \ldots x_n(\mathsf{Z}_2 x_1 \ldots x_n)$. While the x's are variables in the term we started with, it well might be that that term is not of the form $x_1 \ldots x_n$. Intuitionistic logic has full permutation, that is, $(\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C}) \rightarrow \mathcal{B} \rightarrow \mathcal{A} \rightarrow \mathcal{C}$ is a theorem. In combinatory logic, this means that there is a combinatory term that is isomorphic to the starting term possibly except combinators appearing in front of the variables. In other words, instead of an x_i, there may be an occurrence of $\mathsf{Z}_i x_i$. A nice aspect of this approach is that a larger class of proofs can be subjected to the inhabitant extraction procedure. On the other hand, a drawback is that it is less clear how to systematize the various abstraction steps. It is clear from combinatory logic that $\mathsf{Z}_1, \mathsf{Z}_2, \mathsf{Z}_3$ as well as all the necessary Z_i's exist, but it is less clear how to describe an algorithm that is applicable independently of the shape of the starting term, and preferably, efficient too. Of course, as a last resort, we could always use some variant of λ^*.

Second, we can delineate the kind of proofs that we consider. In particular, we can completely avoid the complication caused by contraction on structures that correspond to distinct combinators if we exclude proofs in which $(\mathsf{M}\vdash)$

or $(\mathsf{W}\vdash)$ is applied to a complex structure. A careful reader may immediately notice that even if contraction rules are applied to single formulas only, it may happen that those formulas are inhabited by distinct combinators; hence, the restriction seems not to be sufficient. However, as it will become clear later in this section, we do not run into any problem with combinatory terms that are of the form $\mathsf{Z}_1 x_1(\mathsf{Z}_2 x_1)$. Of course, the limitation of proofs is acceptable, if it does not exclude theorems of $J_\to$ from our consideration. We can apply the same insight from combinatory logic as in the previous approach, however, now we talk about the proofs themselves—before the inhabitant extraction is applied.

Given a proof in which $(\mathsf{M}\vdash)$ or $(\mathsf{W}\vdash)$ has been applied with a complex $\mathfrak{B}$ or $\mathfrak{C}$, respectively, we can apply $(\mathsf{B}\vdash)$, $(\mathsf{B}'\vdash)$, $(\mathsf{T}\vdash)$ and $(\mathsf{C}\vdash)$ together with $(\mathsf{M}\vdash)$ and $(\mathsf{W}\vdash)$ restricted to single formulas and produce the same lower sequent that resulted from the application of $(\mathsf{M}\vdash)$ or $(\mathsf{W}\vdash)$.

Lemma 7. *If $\mathcal{A}$ is a theorem of $J_\to$, then there is a proof of $\mathcal{A}$ in $LJ^{\boldsymbol{T}}_\to$ in which applications of $(\mathsf{M}\vdash)$ and $(\mathsf{W}\vdash)$ (if there are any) involve formulas in place of $\mathfrak{B}$ and $\mathfrak{C}$, respectively.*

Proof. The proof is by induction on the structure of the antecedents, $\mathfrak{B}$ in case of $(\mathsf{M}\vdash)$ and $\mathfrak{C}$ in case of $(\mathsf{W}\vdash)$.

1. If $\mathfrak{B}$ in $(\mathsf{M}\vdash)$ or $\mathfrak{C}$ in $(\mathsf{W}\vdash)$ is $\mathcal{D}$, respectively, then the claim is true.

2. Let us assume that $\mathfrak{C}$ is $\mathfrak{C}_1;\mathfrak{C}_2$.

$$
\begin{array}{cl}
\mathfrak{A}[\mathfrak{B};(\mathfrak{C}_1;\mathfrak{C}_2);(\mathfrak{C}_1;\mathfrak{C}_2)] \vdash \mathcal{A} & \\ \hline
\mathfrak{A}[\mathfrak{C}_1;(\mathfrak{B};(\mathfrak{C}_1;\mathfrak{C}_2));\mathfrak{C}_2] \vdash \mathcal{A} & (\mathsf{B}'\vdash) \\ \hline
\mathfrak{A}[\mathfrak{C}_1;(\mathfrak{B};\mathfrak{C}_1;\mathfrak{C}_2);\mathfrak{C}_2] \vdash \mathcal{A} & (\mathsf{B}\vdash) \\ \hline
\mathfrak{A}[\mathfrak{C}_1;(\mathfrak{B};\mathfrak{C}_1);\mathfrak{C}_2;\mathfrak{C}_2] \vdash \mathcal{A} & (\mathsf{B}\vdash) \\ \hline\hline
\vdots & \\ \hline\hline
\mathfrak{A}[\mathfrak{C}_1;(\mathfrak{B};\mathfrak{C}_1);\mathfrak{C}_2] \vdash \mathcal{A} & \\ \hline
\mathfrak{A}[\mathfrak{B};\mathfrak{C}_1;\mathfrak{C}_1;\mathfrak{C}_2] \vdash \mathcal{A} & (\mathsf{B}'\vdash) \\ \hline\hline
\vdots & \\ \hline\hline
\mathfrak{A}[\mathfrak{B};\mathfrak{C}_1;\mathfrak{C}_2] \vdash \mathcal{A} & \\ \hline
\mathfrak{A}[\mathfrak{C}_2;(\mathfrak{B};\mathfrak{C}_1)] \vdash \mathcal{A} & (\mathsf{T}\vdash) \\ \hline
\mathfrak{A}[\mathfrak{C}_2;(\mathfrak{C}_1;\mathfrak{B})] \vdash \mathcal{A} & (\mathsf{T}\vdash) \\ \hline
\mathfrak{A}[\mathfrak{C}_1;\mathfrak{C}_2;\mathfrak{B}] \vdash \mathcal{A} & (\mathsf{B}'\vdash) \\ \hline
\mathfrak{A}[\mathfrak{B};(\mathfrak{C}_1;\mathfrak{C}_2)] \vdash \mathcal{A} & (\mathsf{T}\vdash)
\end{array}
$$

The thicker lines and $\vdots$'s indicate where the hypothesis of the induction is applied. $\mathfrak{C}_1$ and $\mathfrak{C}_2$ are proper substructures of $(\mathfrak{C}_1;\mathfrak{C}_2)$; hence, $(\mathsf{W}\vdash)$ can be applied. We note that the structure of the bottom sequent is exactly the desired one, because the last four steps restore the association of $\mathfrak{C}_1$ and $\mathfrak{C}_2$.

3. Let us assume that $\mathfrak{B}$ is $\mathfrak{B}_1;\mathfrak{B}_2$.

$$\begin{array}{c}
\mathfrak{A}[\mathfrak{B}_1;\mathfrak{B}_2;(\mathfrak{B}_1;\mathfrak{B}_2)]\vdash\mathcal{A} \\
\hline
\mathfrak{A}[\mathfrak{B}_1;(\mathfrak{B}_1;\mathfrak{B}_2);\mathfrak{B}_2]\vdash\mathcal{A} \\
\hline
\mathfrak{A}[\mathfrak{B}_1;\mathfrak{B}_1;\mathfrak{B}_2;\mathfrak{B}_2]\vdash\mathcal{A} \\
\hline
\vdots \\
\hline
\mathfrak{A}[\mathfrak{B}_1;\mathfrak{B}_2;\mathfrak{B}_2]\vdash\mathcal{A} \\
\hline
\vdots \\
\hline
\mathfrak{A}[\mathfrak{B}_1;\mathfrak{B}_2]\vdash\mathcal{A}
\end{array}
\quad
\begin{array}{l}
(\mathsf{B}'\vdash) \\
(\mathsf{B}\vdash)
\end{array}$$

Once again, the places where the inductive hypothesis is applied are indicated by thick lines and $\vdots$'s. $\mathfrak{B}_1$ and $\mathfrak{B}_2$ are proper substructures of $\mathfrak{B}_1;\mathfrak{B}_2$; hence, the piecewise contractions can be performed by $(\mathsf{M}\vdash)$ and $(\mathsf{W}\vdash)$. ∴

It is easy to see that an application of the $(\mathsf{K}\vdash)$ rule, in which $\mathfrak{C}$ is a compound antecedent can be dissolved into successive applications of $(\mathsf{K}\vdash)$ in each of which $\mathfrak{C}$ is a formula.

NOTE 9. If $\mathcal{A}$ is a theorem of $J_\rightarrow$, then $\mathcal{A}$ is of the form $\mathcal{B}_1 \rightarrow \cdots \mathcal{B}_n \rightarrow p$, for some $\mathcal{B}_1, \ldots, \mathcal{B}_n$ and p.

We know that $J_\rightarrow$ is consistent, hence, no propositional variable is a theorem. Then, a theorem $\mathcal{A}$ contains at least one implication, but the consequent of ... the consequent of ... the consequent of the formula must be a propositional variable, because there are finitely many $\rightarrow$'s in $\mathcal{A}$. We will refer to $\mathcal{B}_1, \ldots, \mathcal{B}_n$ as the *antecedents* of $\mathcal{A}$. Of course, a formula and a sequent are related but different kinds of objects, and so there is no danger that the two usages of the term "antecedent" become confusing.

In order to gain further control over the shape of proofs, we will use the observation we have just stated together with the next lemma to narrow the set of proofs that we will deal with.

Lemma 8. *If $\mathcal{B}_1 \rightarrow \cdots \mathcal{B}_n \rightarrow p$ is a theorem of $J_\rightarrow$, then there is a proof of the sequent $\boldsymbol{T} \vdash \mathcal{B}_1 \rightarrow \cdots \mathcal{B}_n \rightarrow p$ in which the last n steps are consecutive applications of $(\vdash\rightarrow)$.*

Proof. We prove that if there is an application of a rule after $(\rightarrow\vdash)$, which introduces one of the antecedents, then that can be permuted with the application of the $(\rightarrow\vdash)$ rule. We do not (try to) prove, and it obviously does not hold in general, that different applications of the $(\vdash\rightarrow)$ rule can be permuted with each other.[9] (We only consider a couple of cases, and omit the rest of the details.)
1. Let us assume that $(\mathsf{C}\vdash)$ follows $(\vdash\rightarrow)$. The given and the modified proof segments are as follows. (Again, we omit the $\vdots$'s to save some space.)

[9] $\mathcal{B}_i$ is one of $\mathcal{B}_1, \ldots, \mathcal{B}_n$, whereas $\mathcal{C}$ is p (if $\mathcal{B}_i$ is $\mathcal{B}_n$) or $\mathcal{B}_{i+1} \rightarrow \cdots \mathcal{B}_n \rightarrow p$ (otherwise).

$$\dfrac{\dfrac{\mathfrak{A}[\mathfrak{B};\mathfrak{D};\mathfrak{C}];\mathcal{B}_i\vdash\mathcal{C}}{\mathfrak{A}[\mathfrak{B};\mathfrak{D};\mathfrak{C}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\vdash\to)}{\mathfrak{A}[\mathfrak{B};\mathfrak{C};\mathfrak{D}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\mathsf{C}\vdash)\qquad\dfrac{\dfrac{\mathfrak{A}[\mathfrak{B};\mathfrak{D};\mathfrak{C}];\mathcal{B}_i\vdash\mathcal{C}}{\mathfrak{A}[\mathfrak{B};\mathfrak{C};\mathfrak{D}];\mathcal{B}_i\vdash\mathcal{C}}\,(\mathsf{C}\vdash)}{\mathfrak{A}[\mathfrak{B};\mathfrak{C};\mathfrak{D}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\vdash\to)$$

2. Let us suppose that the rule that is to be permuted upward is $(\mathsf{T}\vdash)$.

$$\dfrac{\dfrac{\mathfrak{A}[\mathfrak{C};\mathfrak{B}];\mathcal{B}_i\vdash\mathcal{C}}{\mathfrak{A}[\mathfrak{C};\mathfrak{B}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\vdash\to)}{\mathfrak{A}[\mathfrak{B};\mathfrak{C}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\mathsf{T}\vdash)\qquad\dfrac{\dfrac{\mathfrak{A}[\mathfrak{C};\mathfrak{B}];\mathcal{B}_i\vdash\mathcal{C}}{\mathfrak{A}[\mathfrak{B};\mathfrak{C}];\mathcal{B}_i\vdash\mathcal{C}}\,(\mathsf{T}\vdash)}{\mathfrak{A}[\mathfrak{B};\mathfrak{C}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\vdash\to)$$

3. Now, let the rule be $(\mathsf{M}\vdash)$.

$$\dfrac{\dfrac{\mathfrak{A}[\mathfrak{B};\mathfrak{B}];\mathcal{B}_i\vdash\mathcal{C}}{\mathfrak{A}[\mathfrak{B};\mathfrak{B}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\vdash\to)}{\mathfrak{A}[\mathfrak{B}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\mathsf{M}\vdash)\qquad\dfrac{\dfrac{\mathfrak{A}[\mathfrak{B};\mathfrak{B}];\mathcal{B}_i\vdash\mathcal{C}}{\mathfrak{A}[\mathfrak{B}];\mathcal{B}_i\vdash\mathcal{C}}\,(\mathsf{M}\vdash)}{\mathfrak{A}[\mathfrak{B}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\vdash\to)$$

4. Let us assume that the rule after $(\vdash\to)$ is $(\mathsf{K}\vdash)$.

$$\dfrac{\dfrac{\mathfrak{A}[\mathfrak{B}];\mathcal{B}_i\vdash\mathcal{C}}{\mathfrak{A}[\mathfrak{B}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\vdash\to)}{\mathfrak{A}[\mathfrak{B};\mathfrak{C}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\mathsf{K}\vdash)\qquad\dfrac{\dfrac{\mathfrak{A}[\mathfrak{B}];\mathcal{B}_i\vdash\mathcal{C}}{\mathfrak{A}[\mathfrak{B};\mathfrak{C}];\mathcal{B}_i\vdash\mathcal{C}}\,(\mathsf{K}\vdash)}{\mathfrak{A}[\mathfrak{B};\mathfrak{C}]\vdash\mathcal{B}_i\to\mathcal{C}}\,(\vdash\to)$$

∴

Remark 10. The proof of Lemma 7 relied on applications of some of the structural rules, and it did not involve the $(\vdash\to)$ rule at all. Therefore, the two restrictions on the set of proofs—based on Lemmas 7 and 8—are orthogonal. That is, we know that every theorem of $J_\to$ has a proof that is both free of contractions in which the contracted structures are complex and that ends with the successive introduction of the antecedents of the theorem.

Definition 9. Let $\mathcal{A}$ be a theorem of $J_\to$. A proof $\mathscr{P}$ of $\mathcal{A}$ is *standard* iff $\mathscr{P}$ has no applications of $(\mathsf{M}\vdash)$ or $(\mathsf{W}\vdash)$ or $(\mathsf{K}\vdash)$ in which the subalterns in the rules are complex, and $\mathscr{P}$ ends with n applications of $(\to\vdash)$, when $\mathcal{A}$ is $\mathcal{B}_1\to\cdots\mathcal{B}_n\to p$.

Remark 11. Every theorem of $J_\to$ has a standard proof in $LJ^{\boldsymbol{T}}_\to$, however, not every proof is standard. We should emphasize that standard proofs are *normal* objects. Clearly, every theorem of $J_\to$ has a normal proof, furthermore, a proof that is not normal can be transformed into a normal one. However, the existence of standard proofs for theorems cannot be strengthened to a statement of *unique* existence. For instance, the $(\boldsymbol{T}\vdash)$ and $(\mathsf{M}\vdash)$ rules can always embellish a proof, and so can the $(\mathsf{T}\vdash)$ rule.

The general form of standard proofs is as follows.

$$\dfrac{\dfrac{\dfrac{\vdots}{\boldsymbol{T};\mathcal{B}_1;\mathcal{B}_2;\ldots;\mathcal{B}_n\vdash p}}{\boldsymbol{T};\mathcal{B}_1\vdash\mathcal{B}_2\to\cdots\mathcal{B}_n\to p}\,(\vdash\to)\text{'s}}{\boldsymbol{T}\vdash\mathcal{B}_1\to\mathcal{B}_2\to\cdots\mathcal{B}_n\to p}\,(\vdash\to)$$

Definition 10. Given a standard proof of a theorem $\mathcal{B}_1 \rightarrow \cdots \mathcal{B}_n \rightarrow p$, we call the sequent $\boldsymbol{T}; \mathcal{B}_1; \ldots; \mathcal{B}_n \vdash p$ followed by $(\vdash \rightarrow)$'s the *source sequent*.

Given a standard proof, the part of the proof below the source sequent is uniquely determined. Moreover, for our purposes, the applications of the $(\vdash \rightarrow)$ rule are not very important. Hence, we will mainly ignore the rest of a standard proof below the source sequent; indeed, we will concentrate on the proof of the source sequent itself.

The extraction of a combinatory inhabitant starts with a given standard proof, and as a result of processing the standard proof, we will obtain *one combinatory inhabitant*.

Remark 12. It is always guaranteed that our procedure yields a combinatory inhabitant that has the theorem as its type, however, the procedure does not guarantee that the combinatory inhabitant has the theorem as its *principal type*. In axiomatic calculi, there is a way to ensure that principal types are generated. Starting with axioms (rather than axiom schemas), applications of *condensed detachment* produce theorems in which no propositional variables are identified unless that move is inevitable for the application of the detachment rule. There is no similar mechanism built into sequent calculi, though it might be an interesting question to ask what conditions would be sufficient to guarantee that the theorem is the principal type schema of the combinatory inhabitant we generate.

For the sake of transparency, we divide the whole procedure into *three parts*. First, we want to *trace* how formulas move around in a proof. This is especially pivotal in $LJ^{\boldsymbol{T}}_{\rightarrow}$, because of the plenitude of structural rules. Second, we will introduce a new sort of objects, that we call *caf* 's. By using caf's, we recuperate combinators in accordance with the applications of the rules that bear combinatory labels. Third, we will apply a simple $\mathsf{BB'I}$-*abstraction algorithm*, which will produce the combinatory inhabitant from the caf's that replace the formulas in the source sequent.

4.1 Tracing Occurrences

Curry's observation about a certain connection between structural rules and combinators remained dormant for decades—until Dunn and Meyer in [13] revived this idea. As we already mentioned, they not only established a precise link between structural rules and combinators, but they replaced all the structural rules with *combinatory rules*. The latter kind of rules differ from rules like the structural rules in $LJ^{\boldsymbol{T}}_{\rightarrow}$ by introducing a combinator as a formula into the lower sequent of the rule. The formulation of $LJ^{\boldsymbol{T}}_{\rightarrow}$ drew some inspiration from structurally free logics, but we retained the combinators in the labels of the structural rules (instead of including combinators into the set of formulas).

Following Curry, it is customary to provide an *analysis* for a sequent calculus, which is a classification of the formulas in the rules according to the role that they play in a proof.[10] We give only a partial analysis here, which focuses on

[10] We would have included the analysis earlier, if we would have included all the details of the proof of the cut theorem.

occurrences of formulas that are passed down from the upper sequent to the lower sequent in an application of a rule.

Definition 11. The notion of an *ancestor* of a formula that occurs in a lower sequent of a rule is defined rule by rule as follows.
(1) $(\to\vdash)$: $\mathcal{C}$ (in the upper sequent) is the ancestor of $\mathcal{C}$ (in the lower sequent), and any formula occurring in $\mathfrak{A}$ or $\mathfrak{B}$ in the upper sequent is the ancestor of its copy in the lower sequent.
(2) $(\vdash\to)$: A formula occurring in $\mathfrak{A}$ in the upper sequent is an ancestor of its copy within $\mathfrak{A}$ in the lower sequent.
(3) $(\mathsf{B}\vdash)$, $(\mathsf{B}'\vdash)$, $(\mathsf{C}\vdash)$ and $(\mathsf{T}\vdash)$: $\mathcal{A}$ in the upper sequent is the ancestor of $\mathcal{A}$ in the lower sequent. A formula occurring in $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$ or $\mathfrak{D}$ in the upper sequent is an ancestor of its copy in the identically labeled structure in the lower sequent.
(4) $(\mathsf{W}\vdash)$: $\mathcal{A}$ is the ancestor of $\mathcal{A}$, and a formula in $\mathfrak{A}$ or $\mathfrak{B}$ in the upper sequent is an ancestor of the same formula occurring in the same structure in the lower sequent. A formula occurring in either copy of $\mathfrak{C}$ in the upper sequent is an ancestor of the same formula occurring in $\mathfrak{C}$ in the lower sequent.
(5) $(\mathsf{M}\vdash)$: $\mathcal{A}$ is the ancestor of $\mathcal{A}$, and a formula occurring in $\mathfrak{A}$ in the upper sequent is an ancestor of the same formula occurring in $\mathfrak{A}$ in the lower sequent. A formula occurring in either of the two $\mathfrak{B}$'s in the upper sequent is an ancestor of the same formula occurring in the same position in the structure $\mathfrak{B}$ in the lower sequent.
(6) $(\mathsf{K}\vdash)$: $\mathcal{A}$ is the ancestor of $\mathcal{A}$, and any formula in $\mathfrak{A}$ or $\mathfrak{B}$ is the ancestor of that formula in the identically labeled structure in the lower sequent.
(7) $(\boldsymbol{T}\vdash)$: $\mathcal{A}$ is the ancestor of $\mathcal{A}$, and a formula that occurs in the structure $\mathfrak{A}$ or $\mathfrak{B}$ in the upper sequent is the ancestor of the formula that is of the same shape and has the same location within $\mathfrak{A}$ and $\mathfrak{B}$, respectively, in the lower sequent.

Remark 13. Certain formulas do not have ancestors at all, while some other formulas have exactly one ancestor (even if the same formula has several occurrences in a sequent), and yet another formulas have exactly two ancestors. The part of the analysis that we included here is completely oblivious to the emergence of complex formulas from their immediate subformulas by applications of the connective rules.

A sequent calculus proof—by definition—does not include a labeling of the proof tree with codes for the rules that were applied. However, there is no difficulty in decorating the proof tree, and in most cases, the decoration is *unique.* The possibility of multiple labels arises in the case of some structural rules when certain structures that we denoted by distinct letters turn out to be identical. Nonetheless, the following is straightforward; hence, we do not provide a proof for it.

Claim 12. *Given a proof in $LJ^{\boldsymbol{T}}_{\to}$, it is* decidable *which rules could have been applied at each proof step in the proof tree.*

To put a very wide bound on how many decorations can be added at a step, we can multiply the length of a sequent by the number of rules in $LJ^{\boldsymbol{T}}_{\to}$—still a finite number. This estimate is very much off target, because several pairs of rules cannot be unified at all.

Definition 13. We define a *preferred decoration* of a proof tree by (1)–(6).

(1) If the rule applied is $(\to\,\vdash)$ or $(\vdash\,\to)$, then the appropriate label is the preferred decoration, which is attached to the step.[11]
(2) If the rule applied is either $(\mathsf{B}\vdash)$ or $(\mathsf{B}'\vdash)$, that is, $\mathfrak{B}$ is $\mathfrak{C}$, then the preferred decoration is $(\mathsf{B}\vdash)$.
(3) If the rule applied is either $(\mathsf{W}\vdash)$ or $(\mathsf{M}\vdash)$, that is, $\mathfrak{B}$ is $\mathfrak{C}$, then the preferred decoration is $(\mathsf{M}\vdash)$.
(4) If the rule applied is either $(\mathsf{K}\vdash)$ or $(\boldsymbol{T}\vdash)$, that is, $\mathfrak{B}$ and $\mathfrak{C}$ are $\boldsymbol{T}$, then the preferred decoration is $(\mathsf{K}\vdash)$.
(5) If the rule applied is either $(\mathsf{C}\vdash)$ or $(\mathsf{T}\vdash)$, that is, $\mathfrak{C}$ and $\mathfrak{D}$ are $\mathfrak{B}$, then the preferred decoration is $(\mathsf{T}\vdash)$.
(6) If there is a unique structural rule that is applied, then its label is the preferred decoration.

If a sequent contains several formulas that are of the same shape, then an ambiguity may arise as to where the rule was applied. For instance, if the upper sequent is $\mathcal{A};\mathcal{A};(\mathcal{A};\mathcal{A})\vdash\mathcal{B}$, and it is identical to the lower sequent, then this could be an application of $(\mathsf{T}\vdash)$ in three different ways. In such a case, we assume that the rule has been applied with the least possible scope, and at the leftmost place.[12]

Based on the notion of an ancestor together with the notion of preferred decoration we define an algorithm, which starts off the process of turning sequents into pairs of combinatory terms.

Definition 14. Let a standard proof be given, with the root sequent being the source sequent, which is of the shape $\boldsymbol{T};\mathcal{B}_1;\ldots;\mathcal{B}_n\vdash p$. The formulas in the source sequent, and iteratively, in the sequents above it, are *represented* and *replaced* by variables according to (1)–(5).

(1) The source sequent becomes $x_0;x_1;\ldots;x_n\vdash x_{n+1}$.
(2) The formulas (i.e., formula occurrences) in the $(\mathsf{B}\vdash)$, $(\mathsf{B}'\vdash)$, $(\mathsf{C}\vdash)$ and $(\mathsf{T}\vdash)$ rules are in one-one correspondence between the lower and upper sequents. Thus, the formulas in the upper sequent are replaced by variables according to this correspondence. In the $(\mathsf{K}\vdash)$ and $(\boldsymbol{T}\vdash)$ rules, the same is true, if $\mathfrak{C}$ and $\boldsymbol{T}$ are omitted. That is, the formulas in the upper sequent are ancestors of formulas in the lower sequent, and they are replaced by the variables that stand for them in the lower sequent.

[11] No application of $(\to\,\vdash)$ and $(\vdash\,\to)$ can be unified with an application of any other rule, as it is easy to see.

[12] Of course, this is a futile step to start with, and for instance, in a proof-search tree we would prune proofs to forbid such happenings. However, our present definition for a standard proof does not exclude proofs that contain identical sequents.

(3) In the ($\mathsf{W}\vdash$) and ($\mathsf{M}\vdash$) rules, some formulas have two ancestors. The variables replace the formulas in the upper sequent according to the ancestor relation.
(4) In the ($\vdash\to$) rule, the principal formula of the rule has no ancestor, but it is also absent from the upper sequent. On the other hand, the subalterns are new formulas, which means that they have to be replaced by new variables. If m is the greatest index on a variable used up to this point, then the subaltern, which is the consequent of the arrow formula, is replaced by x_{m+1}. The subaltern, which is the antecedent of the arrow formula, is replaced by x_{m+2}. All the other formulas are replaced by the variable that stands for the formula of which they are an ancestor.
(5) In the ($\to\vdash$) rule, the structures $\mathfrak{A}$ and $\mathfrak{B}$ are the same in the upper and lower sequents, so are the $\mathcal{C}$'s. Here the replacement is carried out according to the one-one correspondence. The immediate subformulas of the arrow formula are handled as in the previous case. The consequent formula is replaced by a new variable with the index $m+1$, if m is the highest index on any variable so far. The antecedent formula is replaced by x_{m+2}.

It is obvious that the definition of the ancestor relation together with the above algorithm guarantees that all formulas are represented by an indexed variable. The proof tree has now been transformed into a tree, in which only x's occur, however, there is an isomorphism between the two trees, and so we continue to modify the new tree by assuming that we can use the information contained in the proof tree itself.

4.2 Formulas Turned into Caf's

As we already mentioned, structurally free logics include combinators into the set of atomic formulas. We do not take combinators to be formulas, but we want to augment the tree of variables with combinators. The variables stand for *formula occurrences*, and we think of them as proxies for identifiable formulas.

Definition 15. The set of *combinatorially augmented formulas* (*caf*'s, for short) is inductively defined by (1)–(2).

(1) If X is a formula, then X is a caf.
(2) If X is a caf and Z is a combinator, then $\mathsf{Z}X$ is a caf.

The ;'s remained in the variable tree, and it is straightforward to define the inductive set generated by ; from the set of x's or from the set of caf's. We expand the range of objects that can instantiate the meta-variables $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \ldots$ to include the variables and the caf's. Context always disambiguates what the structures are composed of.

Using the indexed x's in place of the formula occurrences, we can insert combinators into the tree. However, we need some facts from combinatory logic to make the procedure smooth and easily comprehensible. The following is well known.

Claim 16. $\mathsf{Z}(MN)$ *is* weakly equivalent *to* $\mathsf{BZ}MN$. *In general,* $\mathsf{Z}(M_1 \ldots M_n)$ *is weakly equivalent to* $\underbrace{\mathsf{B}(\ldots(\mathsf{B}}_{n}\,\mathsf{Z})\ldots)M_1 \ldots M_n$.

The impact of the first part of the claim is that compound caf's can be equivalently viewed as being of the form $\mathsf{Z}\mathcal{A}$, for some Z and the formula $\mathcal{A}$ which is the atomic caf component of the compound caf. In other words, if we want to add Z_1 to a caf Z_2x, then we can add Z_2 and immediately move on to $\mathsf{BZ}_1\mathsf{Z}_2x$.

Remark 14. The impact of the second part of this claim is that we can use sufficiently many regular compositors to disassociate a complex combinatory term and to position a combinator *locally* on a caf (rather than on a compound structure), when the structures affected by a structural rule are complex. We will summarily refer to the utilization of this observation by the label (Bs) (or we will use it tacitly).

Now, we define an algorithm that inserts combinators into our tree. The algorithm starts at the leaves of the tree and proceeds in a top-down fashion, level by level. The highest level of the tree contains one or more leaves, each of which is an instance of the axiom.

Remark 15. The ancestor relation is *inherited* by the caf's in an obvious way. Indeed, after the replacement of formula occurrences with indexed variables, the ancestor relation is simpler to spot. If X is a caf in the upper sequent, which is of the form $\mathsf{Z}x_i$, and x_i is a variable in the lower sequent, then the formula occurrence (represented by x_i) in the upper sequent is an *ancestor* of the formula occurrence x_i. As a *default step*, we assume that the lower x_i is changed to $\mathsf{Z}x_i$, and then possibly, similar manipulations are performed on the other caf's too.

We assume that we have the original proof tree with its preferred decoration at hand.

Definition 17. Given a tree of variables, *combinators* are inserted according to (1)–(10), after the default copying of caf's from the upper to the lower sequents.

(1) If the rule applied is $(\boldsymbol{T}\vdash)$, then no combinator is inserted.[13]
(2) If the rule applied is $(\mathsf{K}\vdash)$, then K is added to $\mathfrak{B}$. If $\mathfrak{B}$ is a complex antecedent, then (Bs) is applied to attach K to the leftmost caf in $\mathfrak{B}$.
(3) If the rule applied is $(\mathsf{W}\vdash)$, then we distinguish four subcases. (i) From $\mathfrak{B}; x; x$ we get $\mathsf{W}\mathfrak{B}; x$. (ii) From $\mathfrak{B}; \mathsf{Z}x; x$ we get $\mathsf{BW}(\mathsf{B}'\mathsf{Z})\mathfrak{B}; x$. (iii) From $\mathfrak{B}; x; \mathsf{Z}x$ we get $\mathsf{BW}(\mathsf{B}(\mathsf{B}'\mathsf{Z}))\mathfrak{B}; x$. (iv) From $\mathfrak{B}; \mathsf{Z}_1x; \mathsf{Z}_2x$ we get $\mathsf{BW}(\mathsf{B}(\mathsf{B}'\mathsf{Z}_2)(\mathsf{BB}'\mathsf{Z}_1))\mathfrak{B}; x$. If $\mathfrak{B}$ is complex in any of the subcases, then (Bs) is applied too in order to position the combinator on the leftmost caf within $\mathfrak{B}$.

[13] The variable that is introduced at this step cannot be anything else than x_0, that we will later on turn into I.

(4) If the rule applied is $(\mathsf{M}\vdash)$, then we distinguish four subcases. (i) From $x;x$ we get $\mathsf{M}x$. (ii) From $\mathsf{Z}x;x$ we get $\mathsf{WZ}x$. (iii) From $x;\mathsf{Z}x$ we get $\mathsf{W}(\mathsf{B}'\mathsf{Z})x$. Lastly, (iv) from $\mathsf{Z}_1x;\mathsf{Z}_2x$ we get $\mathsf{W}(\mathsf{B}(\mathsf{B}'\mathsf{Z}_2)\mathsf{Z}_1)x$.
(5) If the rule applied is $(\mathsf{C}\vdash)$, then C is added to $\mathfrak{B}$. Should it be necessary—due to $\mathfrak{B}$'s being complex—$(\mathsf{B}\mathrm{s})$ is applied to move C to the leftmost caf.
(6) If the rule applied is $(\mathsf{T}\vdash)$, then T is added to $\mathfrak{B}$, possibly, with $(\mathsf{B}\mathrm{s})$.
(7) If the rule applied is $(\mathsf{B}\vdash)$, then B is added to $\mathfrak{B}$, with $(\mathsf{B}\mathrm{s})$, if needed.
(8) If the rule applied is $(\mathsf{B}'\vdash)$, then we proceed as in (7), but insert B'.
(9) If the rule applied is $(\vdash\to)$, then we distinguish two subcases. (i) If the caf standing in for the antecedent of the principal formula is x, then no combinator is inserted. (ii) If the sequent is of the form $\mathfrak{A};\mathsf{Z}x_i\vdash x_j$, then $\mathsf{B}'\mathsf{Z}$ is added to $\mathfrak{A}$ (if atomic), or to its leftmost caf using $(\mathsf{B}\mathrm{s})$ (if $\mathfrak{A}$ is complex).
(10) If the rule applied is $(\to\vdash)$, then there are two subcases to consider. (i) The antecedent of the principal arrow formula is x, then no combinator is introduced. (ii) If the caf standing in for the subaltern in the right premise is $\mathsf{Z}x$, then the caf Y representing the principal formula, is modified to $\mathsf{BZ}Y$.

The algorithm is well-defined, because at each step there is exactly one (sub)case that is applicable. Once the source sequent is reached, we have accumulated the information about the combinatory inhabitant in a somewhat dispersed form.

4.3 $\mathsf{BB}'\mathsf{I}$-Abstraction

The source sequent has the general form $\mathsf{Z}_0x_0;\mathsf{Z}_1x_1;\ldots;\mathsf{Z}_nx_n\vdash x_{n+1}$, where any of the Z's may be absent. We want to consider the antecedent as a combinatory term, namely, as $\mathsf{Z}_0\mathsf{I}(\mathsf{Z}_1x_1)\ldots(\mathsf{Z}_nx_n)$. The base $\{\mathsf{S},\mathsf{K}\}$ is combinatorially complete, and so is the base $\{\mathsf{B},\mathsf{B}',\mathsf{C},\mathsf{T},\mathsf{W},\mathsf{M},\mathsf{K}\}$. For instance, I is definable as CKB'. That is, there is no problem with obtaining a Z_{n+1} such that $\mathsf{Z}_{n+1}x_1\ldots x_n$ is weakly equivalent to the previous combinatory term.

However, we want to utilize the insight that the term resulting from the source sequent has a very *special form.* First, we note that if Z_0 is empty, then it is sufficient to consider $\mathsf{Z}_1x_1\ldots(\mathsf{Z}_nx_n)$. Our general idea is to move the Z's systematically to the front, and then to disassociate them from the rest of the term. We achieve this by applying B'- and B-expansion steps.

Definition 18. Given the term $\mathsf{Z}_0\mathsf{I}(\mathsf{Z}_1x_1)\ldots(\mathsf{Z}_nx_n)$, the $\mathsf{BB}'\mathsf{I}$-*abstraction* is defined by replacing the term with that in (1), and then with the term in (2).

(1) $\mathsf{B}'\mathsf{Z}_n(\mathsf{B}'\mathsf{Z}_{n-1}\ldots(\mathsf{B}'\mathsf{Z}_1(\mathsf{Z}_0\mathsf{I})x_1)\ldots x_{n-1})x_n$
(2) $\underbrace{\mathsf{B}(\mathsf{B}\ldots}_{n-1}(\mathsf{B}'\mathsf{Z}_n)\ldots)(\underbrace{\mathsf{B}(\mathsf{B}\ldots}_{n-2}(\mathsf{B}'\mathsf{Z}_{n-1})\ldots)\ldots$
$(\mathsf{B}(\mathsf{B}'\mathsf{Z}_2)(\mathsf{B}'\mathsf{Z}_1(\mathsf{Z}_0\mathsf{I})))\ldots)x_1x_2\ldots x_{n-1}x_n$

Remark 16. Of course, the term in (2) by itself is what we want, however, moving through the term in (1), it becomes obvious that the $\mathsf{BB}'\mathsf{I}$-abstraction yields a term that is weakly equivalent to the term obtained from the source sequent.

Before providing a justification for the whole procedure in the form of a correctness theorem, we return to Example 5 from Sect. 3. It so happens that that proof is a standard proof.

Example 17. We give here the tree of variables, which is isomorphic to the proof tree—save that we completely omit the part of the tree below the source sequent.

$$
\dfrac{\dfrac{\dfrac{\dfrac{x_3 \vdash x_6 \qquad \dfrac{\dfrac{x_3 \vdash x_8 \qquad \dfrac{x_5 \vdash x_{10} \quad x_9 \vdash x_4}{x_7; x_5 \vdash x_4}\,(\to\vdash)}{x_1; x_3; x_5 \vdash x_4}\,(\to\vdash)}{x_1; x_5; x_3 \vdash x_4}\,(\mathsf{C}\vdash)}{x_1; (x_2; x_3); x_3 \vdash x_4}\,(\to\vdash)}{x_1; x_2; x_3; x_3 \vdash x_4}\,(\mathsf{B}\vdash)}{x_1; x_2; x_3 \vdash x_4}\,(\mathsf{W}\vdash)}{x_0; x_1; x_2; x_3 \vdash x_4}\,(\boldsymbol{T}\vdash)
$$

Next is the tree that results by the algorithm that inserts combinators into the tree of variables.

$$
\dfrac{\dfrac{\dfrac{\dfrac{x_3 \vdash x_6 \qquad \dfrac{\dfrac{x_3 \vdash x_8 \qquad \dfrac{x_5 \vdash x_{10} \quad x_9 \vdash x_4}{x_7; x_5 \vdash x_4}\,(\to\vdash)}{x_1; x_3; x_5 \vdash x_4}\,(\to\vdash)}{\mathsf{C}x_1; x_5; x_3 \vdash x_4}\,(\mathsf{C}\vdash)}{\mathsf{C}x_1; (x_2; x_3); x_3 \vdash x_4}\,(\to\vdash)}{\mathsf{BBC}x_1; x_2; x_3; x_3 \vdash x_4}\,(\mathsf{B}\vdash)}{\mathsf{B(BW)(BBC)}x_1; x_2; x_3 \vdash x_4}\,(\mathsf{W}\vdash)}{x_0; \mathsf{B(BW)(BBC)}x_1; x_2; x_3 \vdash x_4}\,(\boldsymbol{T}\vdash)
$$

Despite the fact that there is no separate $(\mathsf{S}\vdash)$ rule, we have a simple proof of the principal type of S. Z_0 is absent, hence, we get $\mathsf{I}(\mathsf{B(BW)(BBC)}x_1)x_2x_3$ as the combinatory term from the source sequent. This term immediately reduces to $\mathsf{B(BW)(BBC)}x_1x_2x_3$, which is already in normal form with the combinator disassociated from the variables.[14] We omit the simple verification that $\mathsf{B(BW)(BBC)}$ is weakly equival to S. Alternatively, one can easily check that $(\mathcal{A} \to \mathcal{B} \to \mathcal{C}) \to (\mathcal{A} \to \mathcal{B}) \to \mathcal{A} \to \mathcal{C}$ is a type of $\mathsf{B(BW)(BBC)}$.

Remark 18. It may be noted that the term WBC was right in spirit. The three atomic combinators in that term appear in the term that resulted (as we indicate by underlining): $\mathsf{B(B\underline{W})(B\underline{B}\,\underline{C})}$.

[14] Incidentally, the combinator that we gave after Example 5 is different. It may be an interesting question how to find sequent calculus proofs given a combinatory inhabitant for a simple type.

4.4 Correctness

The *formulas-as-types* slogan proved to be fruitful, but instead of the separation and recombination of terms and types, we want to view *formulas as terms.* Informally, it is probably clear at this point that the combinatory inhabitant that we extracted from a standard proof has the theorem as its type. However, we will prove the correctness of the whole procedure rigorously. First, we officially turn structures comprising caf's into combinatory terms. That is, we assume that the x's are proxies for certain formula occurrences, and then we reuse them as real variables in combinatory terms.

Definition 19. The *sharpening* operation, denoted by $^{\#}$, is defined by (1)–(4)—relative to a standard proof with caf's in place of formulas.

(1) $x_0^{\#}$, that is, $\boldsymbol{T}^{\#}$ is I.
(2) If $i \neq 0$, then $x_i^{\#}$, that is, $\mathcal{A}^{\#}$ is x_i.
(3) If X is a compound caf $\mathsf{Z}x_i$, then $X^{\#}$ is $\mathsf{Z}(x_i)^{\#}$, that is, $\mathsf{Z}(\mathcal{A})^{\#}$.
(4) $(\mathfrak{A};\mathfrak{B})^{\#}$ is $(\mathfrak{A}^{\#}\mathfrak{B}^{\#})$.

Remark 19. We have three trees, the proof tree we started with, the tree of variables and the tree of caf's. The variables stand for certain formula occurrences. The sharpening operation applied to the antecedent and the succedent in the third tree yields a pair of combinatory terms.

It is easy to see that the following claim, which is important for the consistency in the use of the variables, is true.

Claim 20. *If x_i occurs in the tree of variables that is obtained from the proof tree of a standard proof, then x_i represents occurrences of* only one formula *throughout the tree.*

In other words, it can happen that different occurrences of one formula turn into different variables, but no variable stands for more than one formula.

Given a sharpened sequent $\mathfrak{A}^{\#} \vdash \mathcal{A}^{\#}$, we can determine which variables occur in it; we focus on the left-hand side of the turnstile. The variables ensure that we do not loose track of the formula occurrences across sequents, but otherwise, they simply get as their type the formula for which they stand.

Definition 21. Given the sharpened sequent $\mathfrak{A}^{\#} \vdash \mathcal{A}^{\#}$, let the *context* Δ be defined as $\Delta = \{\, x_i\colon \mathcal{B} \mid x_i \in \mathrm{fv}(\mathfrak{A}^{\#}) \wedge \mathcal{B}^{\#} = x_i \,\}$. The *interpretation* of the sequent is $\Delta \Vdash \mathfrak{A}^{\#} : \mathcal{A}$.

Example 20. The source sequent in our example is $x_0; \mathsf{B}(\mathsf{BW})(\mathsf{BBC})x_1; x_2; x_3 \vdash x_4$. If we display the formulas in the caf's, then the source sequent looks like the following: $\boldsymbol{T}; \mathsf{B}(\mathsf{BW})(\mathsf{BBC})\mathcal{A} \to \mathcal{B} \to \mathcal{C}; \mathcal{A} \to \mathcal{B}; \mathcal{A} \vdash \mathcal{C}$. The sharpening operation turns the antecedent of the sequent into $\mathsf{I}(\mathsf{B}(\mathsf{BW})(\mathsf{BBC})x_1)x_2x_3$. Then, $\mathrm{fv}(\mathfrak{A}^{\#}) = \{\, x_1, x_2, x_3 \,\}$ and $\Delta = \{\, x_1\colon \mathcal{A} \to \mathcal{B} \to \mathcal{C}, x_2\colon \mathcal{A} \to \mathcal{B}, x_3\colon \mathcal{A} \,\}$. Finally, the interpretation of the source sequent is

$$\{\, x_1\colon \mathcal{A} \to \mathcal{B} \to \mathcal{C}, x_2\colon \mathcal{A} \to \mathcal{B}, x_3\colon \mathcal{A} \,\} \Vdash \mathsf{I}(\mathsf{B}(\mathsf{BW})(\mathsf{BBC})x_1)x_2x_3 : \mathcal{C}.$$

Remark 21. We hasten to point out that interpreting the sequents of caf's through statements that closely resemble type-assignment statements is merely a convenience, and it could be completely avoided. However, we presume that it helps to understand the correctness proof for our construction.

Lemma 22. *Given a standard proof that has been transformed into a tree of caf's, the interpretation of each sequent $\mathfrak{A} \vdash \mathcal{A}$ in the tree is correct in the sense that if the combinators are replaced by suitable instances of their principal type schemas, and the variables stand for the formulas as indicated in the context as well as function application is detachment, then $\mathfrak{A}^{\#}$ is $\mathcal{A}$.*

Proof. The proof of this lemma is rather lengthy, hence, we include only two subcases here.
1. If the rule is $(\mathsf{K}\vdash)$, then we have $\Delta \Vdash (\mathfrak{A}[\mathfrak{B}])^{\#} \colon \mathcal{A}$. $\mathfrak{B}^{\#}$ is a subterm of $(\mathfrak{A}[\mathfrak{B}])^{\#}$, hence, in Δ, it computes to a formula, let us say $\mathcal{B}$. $\mathfrak{C}$ may, in general, introduce a new variable, that is, Δ is expanded to Δ'. (We have already pointed out that all the Δ's match one formula to an x.) In the context Δ', $\mathfrak{C}^{\#}$ is assigned a formula, namely, $\mathfrak{C}^{\#} \colon \mathcal{C} \in \Delta'$—whether $\mathfrak{C}^{\#}$ is new in Δ' or not. If we choose for K's type $\mathcal{B} \to \mathcal{C} \to \mathcal{B}$, then $\mathsf{K}\mathfrak{B}^{\#}\mathfrak{C}^{\#}$ gets the type $\mathcal{B}$. Then, we have that $\Delta' \Vdash (\mathfrak{A}[\mathfrak{B};\mathfrak{C}])^{\#} \colon \mathcal{A}$, as we had to show.
2. Let us consider the rule $(\mathsf{C}\vdash)$. We suppose that $\Delta \Vdash (\mathfrak{A}[\mathfrak{B};\mathfrak{D};\mathfrak{C}])^{\#} \colon \mathcal{A}$. There is no change in the context in an application of the $(\mathsf{C}\vdash)$ rule, because $\mathrm{fv}((\mathfrak{A}[\mathfrak{B};\mathfrak{D};\mathfrak{C}])^{\#}) = \mathrm{fv}((\mathfrak{A}[\mathfrak{B};\mathfrak{C};\mathfrak{D}])^{\#})$. The term $(\mathfrak{B};\mathfrak{D};\mathfrak{C})^{\#}$ is a subterm of $(\mathfrak{A}[\mathfrak{B};\mathfrak{D};\mathfrak{C}])^{\#}$, and so it computes to a formula, let us say $\mathcal{E}$. Similarly, $\mathfrak{B}^{\#}$, $\mathfrak{D}^{\#}$ and $\mathfrak{C}^{\#}$ yield some formulas, respectively, $\mathcal{D} \to \mathcal{C} \to \mathcal{E}$, $\mathcal{D}$ and $\mathcal{C}$. The principal type schema of C is $(\mathcal{A} \to \mathcal{B} \to \mathcal{C}) \to \mathcal{B} \to \mathcal{A} \to \mathcal{C}$. Taking the instance where $\mathcal{A}$ is $\mathcal{D}$, $\mathcal{B}$ is $\mathcal{C}$ and $\mathcal{C}$ is $\mathcal{E}$, we get that $\mathsf{C}\mathfrak{B}^{\#}$ is $\mathcal{C} \to \mathcal{D} \to \mathcal{E}$. Then further, $\mathsf{C}\mathfrak{B}^{\#}\mathfrak{C}^{\#}$ is $\mathcal{D} \to \mathcal{E}$, and finally $\mathsf{C}\mathfrak{B}^{\#}\mathfrak{C}^{\#}\mathfrak{D}^{\#}$ is $\mathcal{E}$. Placing back the term into the hole in $(\mathfrak{A}[])^{\#}$, we obtain that $\Delta \Vdash (\mathfrak{A}[\mathfrak{B};\mathfrak{C};\mathfrak{D}])^{\#} \colon \mathcal{A}$. ∴

We have established that up to the $\mathsf{BB'I}$-abstraction the combinators inserted into the caf's in the source sequent are correct. The next lemma provides the last step in the proof of correctness.

Lemma 23. *Let* $\{\, x_1 \colon \mathcal{A}_1, \ldots, x_n \colon \mathcal{A}_n \,\} \Vdash \mathsf{Z}_0\mathsf{I}(\mathsf{Z}_1 x_1)\ldots(\mathsf{Z}_n x_n) \colon \mathcal{A}_{n+1}$. *Then,*

$$\{\, x_1 \colon \mathcal{A}_1, \ldots, x_n \colon \mathcal{A}_n \,\} \Vdash \underbrace{\mathsf{B}(\mathsf{B}\ldots}_{n-1}(\mathsf{B}'\mathsf{Z}_n)\ldots)(\underbrace{\mathsf{B}(\mathsf{B}\ldots}_{n-2}(\mathsf{B}'\mathsf{Z}_{n-1})\ldots)\ldots$$
$$\mathsf{B}(\mathsf{B}'\mathsf{Z}_2)(\mathsf{B}'\mathsf{Z}_1(\mathsf{Z}_0\mathsf{I})))\ldots)x_1x_2\ldots x_{n-1}x_n \colon \mathcal{A}_{n+1}.$$

Proof. The lemma is a special case of B- and B'-expansions, and their well-known properties; hence, we omit the details. ∴

5 Conclusions

We have shown that there is a way to extend the Curry–Howard correspondence to connect *sequent calculus proofs* of intuitionistic implicational theorems and *combinatory inhabitants* over the base $\{\,\mathsf{B}, \mathsf{B}', \mathsf{C}, \mathsf{T}, \mathsf{W}, \mathsf{M}, \mathsf{K}, \mathsf{I}\,\}$. A similar approach has been shown in [9] to be applicable to $LT^{\boldsymbol{t}}_{\rightarrow}$, the implicational fragment of ticket entailment, and we conjecture that the approach can be adapted to other implicational logics that extend the relevance logic $\mathbf{TW}_{\rightarrow}$.

Acknowledgments. I would like to thank the program committee and the organizers of WoLLIC 2018 for inviting me to speak at the conference in Bogotá, Colombia.

References

1. Anderson, A.R., Belnap, N.D., Dunn, J.M.: Entailment: The Logic of Relevance and Necessity, vol. II. Princeton University Press, Princeton (1992)
2. Barendregt, H., Ghilezan, S.: Lambda terms for natural deduction, sequent calculus and cut elimination. J. Funct. Program. **10**, 121–134 (2000)
3. Bimbó, K.: Relevance logics. In: Jacquette, D. (ed.) Philosophy of Logic. Handbook of the Philosophy of Science (D. Gabbay, P. Thagard, J. Woods, eds.), vol. 5, pp. 723–789. Elsevier (North-Holland), Amsterdam (2007). https://doi.org/10.1016/B978-044451541-4/50022-1
4. Bimbó, K.: Combinatory Logic: Pure, Applied and Typed. Discrete Mathematics and its Applications. CRC Press, Boca Raton (2012). https://doi.org/10.1201/b11046
5. Bimbó, K.: The decidability of the intensional fragment of classical linear logic. Theor. Comput. Sci. **597**, 1–17 (2015). https://doi.org/10.1016/j.tcs.2015.06.019
6. Bimbó, K.: Proof Theory: Sequent Calculi and Related Formalisms. Discrete Mathematics and its Applications. CRC Press, Boca Raton (2015). https://doi.org/10.1201/b17294
7. Bimbó, K., Dunn, J.M.: New consecution calculi for $R^{t}_{\rightarrow}$. Notre Dame J. Form. Logic **53**(4), 491–509 (2012). https://doi.org/10.1215/00294527-1722719
8. Bimbó, K., Dunn, J.M.: On the decidability of implicational ticket entailment. J. Symb. Log. **78**(1), 214–236 (2013). https://doi.org/10.2178/jsl.7801150
9. Bimbó, K., Dunn, J.M.: Extracting $\mathsf{BB'IW}$ inhabitants of simple types from proofs in the sequent calculus $LT^{t}_{\rightarrow}$ for implicational ticket entailment. Log. Universalis **8**(2), 141–164 (2014). https://doi.org/10.1007/s11787-014-0099-z
10. Curry, H.B.: A Theory of Formal Deducibility. Notre Dame Mathematical Lectures, no. 6. University of Notre Dame Press, Notre Dame (1950)
11. Curry, H.B.: Foundations of Mathematical Logic. McGraw-Hill Book Company, New York (1963). (Dover, New York, NY, 1977)
12. Dunn, J.M.: A 'Gentzen system' for positive relevant implication (abstract). J. Symb. Log. **38**(2), 356–357 (1973). https://doi.org/10.2307/2272113
13. Dunn, J.M., Meyer, R.K.: Combinators and structurally free logic. Log. J. IGPL **5**(4), 505–537 (1997). https://doi.org/10.1093/jigpal/5.4.505
14. Gentzen, G.: Investigations into logical deduction. Am. Philos. Q. **1**(4), 288–306 (1964)
15. Giambrone, S.: Gentzen systems and decision procedures for relevant logics. Ph.D. thesis, Australian National University, Canberra, ACT, Australia (1983)
16. Giambrone, S.: TW_{+} and RW_{+} are decidable. J. Philos. Log. **14**, 235–254 (1985)
17. Herbelin, H.: A λ-calculus structure isomorphic to Gentzen-style sequent calculus structure. In: Pacholski, L., Tiuryn, J. (eds.) CSL 1994. LNCS, vol. 933, pp. 61–75. Springer, Heidelberg (1995). https://doi.org/10.1007/BFb0022247

18. Lambek, J.: On the calculus of syntactic types. In: Jacobson, R. (ed.) Structure of Language and its Mathematical Aspects, pp. 166–178. American Mathematical Society, Providence (1961)
19. Meyer, R.K.: A general Gentzen system for implicational calculi. Relevance Log. Newsl. **1**, 189–201 (1976)
20. Schönfinkel, M.: On the building blocks of mathematical logic. In: van Heijenoort, J. (ed.) From Frege to Gödel. A Source Book in Mathematical Logic, pp. 355–366. Harvard University Press, Cambridge (1967)