

Intuitionistic Decision Procedures Since Gentzen

Roy Dyckhoff

1 Introduction

Gentzen solved the decision problem for intuitionistic propositional logic in his doctoral thesis [31]; this paper reviews some of the subsequent progress. Solutions to the problem are of importance both for general philosophical reasons and because of their use in implementations of proof assistants (such as *Coq* [4], widely used in software verification) based on intuitionistic logic. Our focus is on calculi and procedures that can be understood in relation to traditional proof theory.

We tend (despite their importance) to avoid implementation issues, e.g. the use of AVL trees rather than lists [73], structure sharing techniques [44], binary decision diagrams [33], caching and dependency directed backjumping [34] and prefix unification [72], in favour of relatively simple calculi where questions such as cut admissibility can be raised and, ideally by syntactic methods, answered. We ignore the first-order case, for which see Schütte [63], Franzén et al. [60] and Otten [55]. For implementations see Otten's ILTP website [56]. We also have our own implementations of several of the calculi mentioned here, using our own Prolog software YAPE ("Yet Another Proof Engine") [15] allowing sequent calculus rules to be coded clearly and proofs to be displayed using L^AT_EX, either as trees or linearly, as illustrated in Sect. 8.7. We are particularly interested in questions of

1. **termination** (hence decidability),
2. **bicompleteness** (extractability of models from failed proof searches),
3. **determinism** (avoidance of backtracking),
4. **simplicity** (allows easier reasoning about systems).

We include a short discussion of labelled calculi; concerning termination therein, we refer to some recent literature.

R. Dyckhoff (✉)

University of St. Andrews, St. Andrews, UK
e-mail: rd@st-andrews.ac.uk

2 Gentzen's Calculus, LJ

Gentzen [31] solved the decision problem for **Int** with a calculus **LJ**, in which the antecedent of each sequent is a list of formulae and the succedent is either empty or a single formula. Since lists rather than sets are used, and the “operational” rules act only on the first element of the list, rules of *Exchange*, *Contraction* and *Thinning* (hereinafter called *Weakening*) are required. *Initial sequents* are of the form $A \Longrightarrow A$, where A is a formula. Let C indicate either a formula or an empty succedent. If A is a formula and Γ a list, then A, Γ is the list with head A and tail Γ ; and similarly Γ, Δ is the list obtained by concatenating lists Γ and Δ . The important rules (for intuitionistic implication) are

$$\frac{\Gamma \Longrightarrow A \quad B, \Delta \Longrightarrow C}{A \rightarrow B, \Gamma, \Delta \Longrightarrow C} L\rightarrow \quad \frac{A, \Gamma \Longrightarrow B}{\Gamma \Longrightarrow A \rightarrow B} R\rightarrow$$

Gentzen took negation as a primitive notion, with rules as follows:

$$\frac{\Gamma \Longrightarrow A}{\neg A, \Gamma \Longrightarrow} L\neg \quad \frac{A, \Gamma \Longrightarrow}{\Gamma \Longrightarrow \neg A} R\neg$$

which have the virtue of being illustrations of the succedent being empty.

But in practice we will take negation as a defined notion, using $\neg A =_{\text{def}} A \rightarrow \perp$, and allowing also $\perp, \Gamma \Longrightarrow C$ as an initial sequent (and now we can replace empty succedents by \perp). For completeness we show also his rules for conjunction and disjunction:

$$\frac{A_i, \Gamma \Longrightarrow C}{A_0 \wedge A_1, \Gamma \Longrightarrow C} L\wedge_i \quad \frac{\Gamma \Longrightarrow A \quad \Gamma \Longrightarrow B}{\Gamma \Longrightarrow A \wedge B} R\wedge$$

$$\frac{A, \Gamma \Longrightarrow C \quad B, \Gamma \Longrightarrow C}{A \vee B, \Gamma \Longrightarrow C} L\vee \quad \frac{\Gamma \Longrightarrow A_i}{\Gamma \Longrightarrow A_0 \vee A_1} R\vee_i$$

Gentzen's approach is not (as one may find suggested in the literature) a root-first approach, but to see what sequents (from the finite range of possibilities) are initial, what can be inferred from them, and so on. It may thus be regarded as an early instance of Maslov's “Inverse Method” [48]. Defining a sequent to be *reduced* if its antecedent contains no more than three occurrences of any formula, and after showing that a derivation of a reduced sequent can be modified into one where all the sequents are reduced, one can see an obvious finiteness argument exploiting the subformula property. Kosta Došen observed in [12] that Gentzen's “three occurrences” can be reduced to “two occurrences”. (B subsumes $A \rightarrow B$, so, in the rule $L\rightarrow$, we may need a copy of $A \rightarrow B$ in Γ but we don't need one in Δ . In other words, $B, \Delta \Longrightarrow C$ and $B, A \rightarrow B, \Delta \Longrightarrow C$ are inter-derivable.)

2.1 *Calculi of Ono, Ketonen and Kleene, Troelstra's G3i*

Gentzen's **LJ** is not a calculus suitable for solving the decision problem root-first—note especially the “context-splitting” nature of $L\rightarrow$. Ono [54], Ketonen [42] and Kleene [43] observed in various ways that it was better to incorporate structural rules (like *Exchange*, *Weakening* and *Contraction*) into the notation and/or into the operational rules: for example, in Kleene's treatment one has

$$\frac{A \rightarrow B, \Gamma \Longrightarrow A \quad B, A \rightarrow B, \Gamma \Longrightarrow C}{A \rightarrow B, \Gamma \Longrightarrow C} L\rightarrow \quad \frac{A, \Gamma \Longrightarrow B}{\Gamma \Longrightarrow A \rightarrow B} R\rightarrow$$

with the convention that two sequents are “cognate” (and thus are interchangeable) iff exactly the same formulae appear in the antecedents (regardless of number and order) and they have the same succedent. Thus, using modern terminology found in [68] but not in [43], A, Γ is now the multiset sum of the multisets A and Γ . Care needs to be taken with rules if sets are used, since A, Γ as a pattern can match both with Γ containing A and with Γ not containing A .

Note that $A \rightarrow B$ can be omitted from the second premiss of $L\rightarrow$ (as before, since it is subsumed by B), but not from the first, lest completeness be lost.

This incorporation of *Weakening* into the rules allows a “root-first” approach, i.e. the root-first construction of a tree with the sequent to be decided at the root and expansion of the tree by choice of a principal formula and generation of the appropriate rule instance and thus of part of the next level of the tree. Typically (provided the search can be shown to terminate) such an expansion is done depth-first rather than breadth-first.

Troelstra's book (with Schwichtenberg) [68] gives a good treatment of this transformation of Gentzen's calculi by incorporation of the structural rules, including acknowledgment of the inspiration provided by Dragalin [13].

2.2 *Maehara's Calculus, m-G3i*

Maehara [47] introduced¹ an important variant of Kleene's calculus: succedents can now be arbitrary (finite) collections Δ of formulae rather than just empty or singular. The rules for implication and disjunction are then

$$\frac{A \rightarrow B, \Gamma \Longrightarrow A, \Delta \quad B, A \rightarrow B, \Gamma \Longrightarrow \Delta}{A \rightarrow B, \Gamma \Longrightarrow \Delta} L\rightarrow \quad \frac{A, \Gamma \Longrightarrow B}{\Gamma \Longrightarrow A \rightarrow B, \Delta} R\rightarrow$$

¹A referee points out that this paper gave a classical provability interpretation of intuitionistic logic and an early syntactic proof of its faithfulness, a result conjectured by Gödel (and proved semantically by McKinsey and Tarski), as could usefully have been mentioned in [20].

$$\frac{A, A \vee B, \Gamma \Longrightarrow \Delta \quad B, A \vee B, \Gamma \Longrightarrow \Delta}{A \vee B, \Gamma \Longrightarrow \Delta} L\vee \quad \frac{\Gamma \Longrightarrow A, B, A \vee B, \Delta}{\Gamma \Longrightarrow A \vee B, \Delta} R\vee$$

which have the virtues (1) that $L\rightarrow$ and $R\vee$ are invertible and hence (2) that all the non-determinism in root-first search pertains to the $R\rightarrow$ rule and the choice of its principal formula $A\rightarrow B$ in the succedent for analysis. (We consider $R\rightarrow$ as a special case of $R\rightarrow$.) Without loss of completeness one may remove the repetition of the principal formula $A \vee B$ from the premisses of $L\vee$ and $R\vee$.

Proofs in this system can be much smaller than those in the single-succedent calculus: see Egly and Schmitt [23].

Approximately this calculus is used as a basis in tableau theorem proving; one advantage is that counter-models can be extracted from failed searches. (Note that the rule $R\vee$ is classical here but not in **G3i**.) In other words, the calculus is *bicomplete*.

The same calculus (presented as a tableau calculus) appears, attributed to Beth [5], in Fitting's thesis [27] and in his book [28]. It can also be found in Curry's book [11]. As noted by von Plato [58], a similar system was developed by Gentzen, and can be found in his Nachlass.

Fitting's notion of "tableau" is a finite sequence of configurations, each obtained from its predecessor by applying a rule to one of the problems (i.e. sequents) therein; each *configuration* is a finite collection of problems (each of which has to be solved for the configuration to be *closed*). A problem is *solved* when it is an initial sequent. The tableau is *closed* (and the proof complete) when one (without loss of generality, the last) of its configurations is closed. Some rule applications replace a problem by just one problem while others replace it by two; in each case the new configuration is just the old configuration in which the old problem has been replaced by the new ones. There is no need to apply rules to solved problems or to closed configurations.

Using sequent notation rather than signed formulae, and (as in proof theory rather than tableau practice) writing the proof from bottom to top, and (for clarity) marking principal formulae with a box), here is a closed tableau proving the sequent $\Longrightarrow (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r$:

$$\begin{array}{c} \frac{p \rightarrow q \rightarrow r, p \rightarrow q, p \Longrightarrow p, r; \quad p \rightarrow q \rightarrow r, p \rightarrow q, p, q \Longrightarrow p, r; \quad p \rightarrow q \rightarrow r, p \rightarrow q, p, q, q \rightarrow r \Longrightarrow q, r; \quad p \rightarrow q \rightarrow r, p \rightarrow q, p, q, q \rightarrow r, r \Longrightarrow r}{p \rightarrow q \rightarrow r, p \rightarrow q, p \Longrightarrow p, r; \quad p \rightarrow q \rightarrow r, p \rightarrow q, p, q \Longrightarrow p, r; \quad p \rightarrow q \rightarrow r, p \rightarrow q, p, q, \boxed{q \rightarrow r} \Longrightarrow r} L\rightarrow \\ \frac{\quad}{p \rightarrow q \rightarrow r, p \rightarrow q, p \Longrightarrow p, r; \quad \boxed{p \rightarrow q \rightarrow r} p \rightarrow q, p, q \Longrightarrow r} L\rightarrow \\ \frac{\quad}{p \rightarrow q \rightarrow r, \boxed{p \rightarrow q} p \Longrightarrow r} R\rightarrow \\ \frac{\quad}{p \rightarrow q \rightarrow r, p \rightarrow q \Longrightarrow \boxed{p \rightarrow r}} R\rightarrow \\ \frac{\quad}{p \rightarrow q \rightarrow r \Longrightarrow \boxed{(p \rightarrow q) \rightarrow p \rightarrow r}} R\rightarrow \\ \frac{\quad}{\Longrightarrow \boxed{(p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r}} R\rightarrow \end{array}$$

As can be seen, each problem in the "last" (i.e. uppermost) configuration is solved; hence, that configuration is closed. Conjunctive branching is handled at each $L\rightarrow$ step by adding an extra problem, thus hiding the tree structure familiar to us from

Gentzen’s work; but, backtracking (because of the rule $R\rightarrow$) is not made explicit. The repetition (as one moves upwards) of closed problems is official, but inessential. Backtracking could be added by allowing branching—one branch should then be required to be closed; only then is it feasible to extract counter-models from failed searches. [In this proof there are no backtracking possibilities.]

Termination is assured by the subformula property: some form of loop-checking is required.

An interesting variation is the calculus **GHPC** of Dragalin [13]; by omitting Δ in the first premiss, without loss of completeness, this has a non-invertible $L\rightarrow$ rule, incorporating a form of focusing useful in the proof theory of the multi-succedent **m-G4ip**.

3 Vorob’ev’s Calculus, **G4ip**

Vorob’ev introduced in [70, 71] an important calculus now known as **G4ip**. Others (Hudelmaier [39–41], Dyckhoff [14]) rediscovered (and refined) the same calculus some 40 years later. See also Lincoln et al. [45]. The goal is to avoid use of loop-checking (messy to reason about, tricky to implement). The formula $\neg\neg(p \vee \neg p)$ easily illustrates the need for loop-checking in Fitting’s calculus. The key idea is to replace, in the single succedent calculus² **G3ip**, the left rule for implication $L\rightarrow$ by four rules, according to the form of the implication’s antecedent A , exploiting the intuitionistic equivalences

1. $\perp \rightarrow B \equiv \top$,
2. $P \wedge (P \rightarrow B) \equiv P \wedge B$,
3. $(C \wedge D) \rightarrow B \equiv C \rightarrow (D \rightarrow B)$,
4. $(C \vee D) \rightarrow B \equiv (C \rightarrow B) \wedge (D \rightarrow B)$,
5. $C \wedge ((C \rightarrow D) \rightarrow B) \equiv C \wedge (D \rightarrow B)$.

to reduce the formula’s complexity (in a carefully measured sense) and a bit of proof theory to show completeness. The effect is that root-first depth-first proof search terminates, i.e. root-first application of inference rules decreases the sequent’s “size” rather than allowing it to oscillate up and down without termination. A measure of “size” (due to Hudelmaier) can be found in [68]; another can be found in [14].

The rules for left-implication are thus as follows (with P atomic):

$$\frac{\Gamma, P, B \Longrightarrow E}{\Gamma, P, P \rightarrow B \Longrightarrow E} L0 \rightarrow \quad \frac{\Gamma, C \rightarrow (D \rightarrow B) \Longrightarrow E}{\Gamma, (C \wedge D) \rightarrow B \Longrightarrow E} L\wedge \rightarrow$$

$$\frac{\Gamma, C \rightarrow B, D \rightarrow B \Longrightarrow E}{\Gamma, (C \vee D) \rightarrow B \Longrightarrow E} L\vee \rightarrow \quad \frac{\Gamma, C, D \rightarrow B \Longrightarrow D \quad \Gamma, B \Longrightarrow E}{\Gamma, (C \rightarrow D) \rightarrow B \Longrightarrow E} L\rightarrow \rightarrow$$

²The final “p” in the name indicates “propositional”.

of which each but the last is invertible. There is no need for the case $\perp \rightarrow B$ to be included, since this formula is equivalent to \top . In each case the size of each premiss is less than that of the conclusion.

Formulae $P \rightarrow B$ are called *atomic implications*; those of the form $(C \rightarrow D) \rightarrow B$ are called *nested implications*. Proving completeness of the resulting system is a challenging exercise.

3.1 Hudelmaier's Refinements of Vorob'ev's Calculus

First appearance of Hudelmaier's rediscovery of Vorob'ev's work is in [39], i.e. in 1988 and then in his thesis, published in 1992 as [40].

The novelty (apart from some different proof methods) w.r.t. **G4ip** is in [41], ensuring proofs are of linear rather than exponential depth, by use of **fresh** proposition variables P in the cases $(L\vee\rightarrow$ and $L\rightarrow\rightarrow)$ where a **compound** subformula (B , resp. D) from the conclusion is duplicated into a premiss. So these rules are replaced by the following:

$$\frac{\Gamma, C \rightarrow P, D \rightarrow P, P \rightarrow B \Rightarrow E}{\Gamma, (C \vee D) \rightarrow B \Rightarrow E} L\vee\rightarrow' \quad \frac{\Gamma, C, D \rightarrow P, P \rightarrow B \Rightarrow P \quad \Gamma, B \Rightarrow E}{\Gamma, (C \rightarrow D) \rightarrow B \Rightarrow E} L\rightarrow\rightarrow'$$

This allows one to show the decision problem to be in $O(n \log n)$ -SPACE. (In 1977 Ladner showed **S4**, and hence also **Int**, to be in PSPACE; in 1979 Statman [65] showed **Int** to be P-SPACE-hard.)

More precisely, Hudelmaier showed the value of $L\vee\rightarrow'$ but had difficulty³ with $L\rightarrow\rightarrow'$, so he adopted a more complicated rule; the difficulty was overcome by Fiorino [3, 26]. The proof of soundness is easy: given proofs of the premisses of $L\rightarrow\rightarrow'$, substitute D for P in the first, cut with a proof of $D \rightarrow D$, and then use the sound rule $L\rightarrow\rightarrow$ to infer the desired conclusion. As for completeness, note that the rule

$$\frac{C, D \rightarrow P, P \rightarrow B, \Gamma \Rightarrow P}{C, (C \rightarrow D) \rightarrow B, \Gamma \Rightarrow D}$$

is invertible⁴ in **G3ip** (or any other standard calculus for **Int**), and thus, in a proof by induction on sequent size of completeness of **G4ip**, a sequent matching the conclusion may be reduced to the corresponding premiss.

³He commented "Unfortunately, this method does not work for the second problem."

⁴As established using a cut of the conclusion with $D, D \rightarrow P \Rightarrow P$ and a cut with $C, D \rightarrow P, P \rightarrow B \Rightarrow (C \rightarrow D) \rightarrow B$.

3.2 Dyckhoff's Refinements of Vorob'ev's Calculus

Novelty⁵ (apart from different proof methods) of [14] is to have (in addition to the single succedent calculus **G4ip**) a multi-succedent calculus **m-G4ip**. ([14] called them **LJT** and **LJT***.) This is closer to tableau methods used in implementations and allows easy extraction of a counter-model from a failed proof search [57] (joint work with Pinto). For the multi-succedent version, use Maehara's rule $R \rightarrow$ and replace (in each of the four special left rules for implication) each succedent formula E by Δ . (But the first premiss of $L \rightarrow$ should have, as in Dragalin's calculus **GHPC**, just one formula D in its succedent, lest the rule be unsound.) This can be combined with Hudelmaier's depth-reduction techniques. Various refinements of the multi-succedent version have been developed and implemented by a group in Milan (Avellone, Ferrari, Fiorentino, Fiorino, Miglioli[†], Moscato and Ornaghi); one of the most recent papers is [24]. Their proof methods are almost entirely semantic.

3.3 Proof Theory of Vorob'ev's Calculus

Vorob'ev's proof of completeness of the calculus rests on a lemma now seen as the completeness of a single-succedent focused calculus **LJQ'**: see Dyckhoff and Lengrand [17] for details, and its extension to a multi-succedent focused calculus **LJQ*** (a variant of a calculus in Herbelin's thesis [36]). Root-first proof search in **LJQ'** occasionally focuses on the succedent and analyses it until either it is atomic or the rule $R \rightarrow$ is used; in particular, the $L \rightarrow$ rule requires (in the first premiss, but not in the second) a focus on the succedent. The completeness of this approach is a useful fact, exploited not just in Vorob'ev's [70, 71] but also in Hudelmaier's [41] (in which it is mentioned as "folklore"). The same trick is applicable elsewhere, e.g. in guarded logic and labelled calculi.

Dyckhoff and Negri [18] give a direct proof of completeness (w.r.t. an axiomatic presentation, via *Cut*-admissibility, rather than w.r.t. semantics), showing that *Contraction* is admissible in **G4ip** and hence (with explicit cut reduction steps) that *Cut* is admissible. This approach generalises to the multi-succedent case, and even shows the completeness of a first-order version (without, alas, the depth-boundedness ...). Dyckhoff, Kesner and Lengrand [16] show (for the implicational fragment **G4ip**[→] only) how to make the cut reduction system strongly normalising.

4 Weich's Thesis

Weich [73, 74] made several excellent contributions: verified constructive completeness proofs, in *MINLOG* and in *Coq*, from which *Scheme* or *OCaml* programs may be extracted; pruning of the search by use of counter-models generated earlier in

⁵As realised by the author after too long a delay.

the search (“an improvement both astonishing and significant”); a “conditional normal form” for formulae, obtained by pre-processing: essentially, $A \rightarrow B$ where A is a conjunction of atoms and B is one of $\perp, P, Q \vee R, (Q \rightarrow R) \rightarrow \perp, (Q \rightarrow R) \rightarrow S$. This reduces some of the run-time expansions that are otherwise repeated in different branches of the search. (P, Q, R, S indicate atoms.)

5 Easy Optimisations

Once the succedent is empty (or just \perp), one can revert to classical logic. Search can be pruned if a new subproblem (arising from choice of instance of non-invertible rule) isn’t solvable classically.

“Simplification”: once an atom p is added to the antecedent, all formulae in the sequent are simplified by putting $p = \top$ and reducing (e.g. with $\top \wedge A \equiv A$).

The same works if a negated atom $\neg p$ is added to the antecedent; the sequent is simplified by replacing p throughout by \perp (and simplifying accordingly, e.g. with $\perp \vee A \equiv A$).

When a problem is analysed into two subproblems, and the first is solved, one may use [73] information from it in the second; e.g. the rules (one for multi-succedent; two for single-succedent calculi).

$$\frac{A, \Gamma \Longrightarrow \Delta \quad B, \Gamma \Longrightarrow A, \Delta}{A \vee B, \Gamma \Longrightarrow \Delta} L \vee' \qquad \frac{A, \Gamma \Longrightarrow G \quad B, \Gamma \Longrightarrow A \vee G}{A \vee B, \Gamma \Longrightarrow G} L \vee^* \\ \frac{A, \Gamma \Longrightarrow G \quad B, A \rightarrow G, \Gamma \Longrightarrow G}{A \vee B, \Gamma \Longrightarrow G} L \vee''$$

Several other easy optimisations are to be found in Franzén’s [29], Ferrari et al’s [24] and Weich’s [73].

6 Goal-Directed Pruning

We say that the atom P occurs *strictly positively* in \perp ; and in P ; and in $A \wedge B$ iff in one of A and B ; in $A \vee B$ iff in both A and B ; and in $A \rightarrow B$ iff in B . In brief, P *sp-occurs* in the formula. The following is based on a result in [67, p. 69].

Theorem 1 *If $\Gamma \Longrightarrow P$ in **G4ip**, then there is some formula in Γ in which P sp-occurs.*

Proof By induction on the derivation and case analysis:

1. The last step has \perp (resp. P) principal; then $\perp \in \Gamma$ (resp. $P \in \Gamma$) and P sp-occurs therein.
2. The last step has $A \wedge B$ principal and derivable premiss $A, B, \Gamma' \Longrightarrow P$; by the induction hypothesis we can find a suitable formula either in Γ' or in $\{A, B\}$. In the latter case, P sp-occurs in $A \wedge B$.
3. The last step has $A \vee B$ principal and derivable premisses $A, \Gamma' \Longrightarrow P$ and $B, \Gamma' \Longrightarrow P$; by the induction hypothesis, we can find a suitable formula either in Γ' or in both A and B . In the latter case, P sp-occurs in both A and B (and hence in $A \vee B$).
4. The remaining cases are similar. □

Thus, the sequent

$$(p \rightarrow s) \rightarrow t, (c \rightarrow p) \rightarrow b \Longrightarrow p$$

cannot be reduced (but would be reduced if we had $p = t$). Reduction of a sequent to two new sequents using (root-first) the rule $L\rightarrow$ generates premisses of which the first may not be derivable even if the conclusion is derivable; the Theorem can be used here to prune the search space. One can see this as a weak form of “goal-directedness”.

One would like to strengthen this to the claim that the formula given by the theorem can be taken as principal, thus (in general) allowing many possible choices of principal formula to be ignored. A counterexample is given by the sequent $q \vee r, q \rightarrow p, r \rightarrow p \Longrightarrow p$. A counter-example with only atomic and nested implications in the antecedent is $(a \rightarrow a) \rightarrow b, b \rightarrow (q \vee r), q \rightarrow p, r \rightarrow p \Longrightarrow p$; the only derivation of this in **G4ip** ends with $(a \rightarrow a) \rightarrow b$ principal—but p does not sp-occur therein.

Weich [73] presents and justifies a much stronger form of goal-directedness.

7 Mints' Classification

Mints [51] gave a convenient classification of subclasses of **Int**, and their complexity. Let $|S|$ be the formula equivalent of a sequent S . By introduction of new variables (following Skolem 1920 and Wajsberg 1938), one can in linear time replace a formula A by a sequent S_A so that A is provable iff $|S_A|$ is provable, where the succedent of S_A is atomic and the antecedent consists of formulae that (with P, Q, R atomic) are one of

- (0) **negated atoms** $\neg P$,
- (1) **atoms** P ,
- (2) **implications** $P \rightarrow Q$,
- (3) **binary implications** $P \rightarrow (Q \rightarrow R)$,
- (4) **nested implications** $(P \rightarrow Q) \rightarrow R$,

- (5) **implied disjunctions** $P \rightarrow (Q \vee R)$,
- (6) **negative implications** $P \rightarrow (\neg Q)$,
- (7) **converse negative implications** $\neg Q \rightarrow P$.

Thus, it suffices to consider only sequents where the antecedent X consists of formulae of these eight types (and the succedent is atomic).

“Simplification” allows us to remove all formulae of type 0 or 1. For consistency with Mints’ paper we avoid this step.

According to the types of formulae used in X , one has complexity results: if all formulae of X are of type 2, 3 or 4 we talk of the class [2,3,4], and similarly for other classes.

One then has that

- The class [2,3,4] (and any superclass) is PSPACE-complete,
- the class [1,2,5,6] is NP-complete (and any superclass is NP-hard),
- the class [0,1,2,3,6] (and any subclass) is in LIN,
- the class [0,1,2,4,5,7] (and any subclass) is in P,
-

Note that the class [0,1,2,3,5,6] is the zero-order case of “coherent logic”, recently the subject of theoretical study and automation [6].

From the perspective of **G4ip**, the difficulty of proof search is dealing with “nested implications”, i.e. formulae of type (4) and their variant (7). So the surprise is that (provided we exclude formulae of type (3) and their variant (6)) while allowing formulae of type (4) and their variant (7), the decision problem is in P. This is achieved using a resolution method [50], a variant of the familiar “forward chaining” method that disposes linearly of [0,1,2,3,6]. Tammet [66] implemented this method. But the verdict [56] by the ILTP website authors is “Prover seems to be incorrect”.

8 Ensuring the Subformula Property

G4ip lacks the subformula property, and has been criticised by some for this failing, apparently on philosophical grounds.

Despite a strong feeling that it doesn’t matter (because it is still analytic in a weak but adequate sense), we consider henceforth some further approaches that ensure that proofs have the subformula property:

1. Underwood’s calculus.
2. Intercalation calculus of Sieg and Cittadini.
3. Implication-locking (Franzén).
4. Loop-checking (two approaches).
5. The calculus **LJ_{Pm}*** of Mints.
6. The calculi **IG^r** and **SIC** of Corsi and Tassi.
7. The calculus **LSJ** of Ferrari, Fiorentino and Fiorino.
8. The calculus **GLJ** of RD (unpublished).

8.1 Underwood's Calculus

Underwood [69] gave a constructive completeness proof for a calculus presented rather in terms of Kripke semantics than proof theory. As reconstructed by Weich [73], this is as follows, with antecedents and succedents regarded as sets: rules for conjunction and disjunction are rather standard, with provisos about not being used if (used root-first) they fail to add a new formula to one of the sets. Rules for implication are thus:

$$\frac{A \rightarrow B, \Gamma \Longrightarrow A, \Delta \quad B, A \rightarrow B, \Gamma \Longrightarrow \Delta}{A \rightarrow B, \Gamma \Longrightarrow \Delta} L \rightarrow (A \notin \Delta; B \notin \Gamma)$$

$$\frac{A, \Gamma \Longrightarrow B, A \rightarrow B, \Delta}{A, \Gamma \Longrightarrow A \rightarrow B, \Delta} RSimp (B \notin \Delta) \quad \frac{A, \Gamma \Longrightarrow B}{\Gamma \Longrightarrow A \rightarrow B, \Delta} R \rightarrow (A \notin \Gamma)$$

Branches are bounded in length by the square of the number of the end-sequent's subformulae, hence termination without loop-checking. This is the basis for the extraction of an algorithm by Caldwell [8]. (An early version was in use by Constable in NuPRL about 1991.)

8.2 Intercalation Calculus of Sieg and Cittadini

Sieg and Cittadini [64], building on earlier work for classical logic by Byrnes, present a system (the “intercalation calculus”) geared towards use in a pedagogical system *AProS*, in use at Carnegie Mellon University and elsewhere; for such a system, lack of the subformula property would be confusing. Thanks to this property, the search space is finite; a loop-checker is required. *Questions* (i.e. sequents) are of the form $\alpha; \beta ? G$, where G is the “goal formula” and α and β are sets of formulae, the former being assumptions and the latter the formulae derived from assumptions. As examples of the rules, we present those for implication in more traditional notation:

$$\frac{\alpha; \beta \Longrightarrow A \quad \alpha; \beta, B \Longrightarrow G}{\alpha; \beta \Longrightarrow G} L \rightarrow \text{ where } A \rightarrow B \in \alpha \cup \beta, B \notin \alpha \cup \beta, A \neq G$$

$$\frac{A, \alpha; \beta \Longrightarrow B}{\alpha; \beta \Longrightarrow A \rightarrow B} R \rightarrow$$

The objective here, of course, is the finding of normal natural deduction proofs rather than being an efficient decision procedure. Herbelin's calculus **LJT** [36] would be another approach to this objective, allowing (in principle) the discovery or enumeration of all such proofs.

8.3 Implication-Locking (Franzén’s approach)

Franzén [29, 60] uses the notion of *covering*: Γ *covers* A if

- $A \in \Gamma$, or
- $A \equiv B \wedge C$ and Γ covers **both** B and C , or
- $A \equiv B \vee C$ and Γ covers **one of** B and C , or
- $A \equiv B \rightarrow C$ and Γ covers C .

The rule $R \rightarrow$ is then specialised to the two cases: the usual one (a *transfer instance*) if Γ does not cover the antecedent A of the principal formula, and the special one (infer $\Gamma \Longrightarrow A \rightarrow B$ from $\Gamma \Longrightarrow B$ when Γ covers A) (similar, respectively, to Underwood’s $R \rightarrow$ and $RSimp$). There is then the restriction that, on each branch, every two instances of $L \rightarrow$ must be separated by a transfer instance of $R \rightarrow$. In other words, implications are “locked” until “released” by a transfer. This is enough to ensure termination.

8.4 Loop-Checking (The Bern Approach)

For simplicity, we ignore disjunction and absurdity. We may therefore restrict $L \rightarrow$ to cases where the succedent formula is an atom. Left rules are *cumulative*, i.e. the principal formula is duplicated to the premiss. So a loop can only occur during a phase when nothing new is added to the antecedent, and in the succedent a formula appears and later (i.e. higher up the proof branch) appears again. Without loss of generality, one can restrict to the case where this formula is an atom. Sequents now contain an extra component, the *history* H (the set of such atoms; quite simple). Then, if as one moves root-first one uses the left rule for implication with conclusion having atomic succedent P , this use is *blocked* if already $P \in H$, but otherwise P is added to the history. If a new formula is added to the antecedent, the history at the premiss is emptied. See [37] (by Heuerding et al. 1996) for details.

8.5 Loop-Checking (The St Andrews Approach)

Howe [38] presented a variation of the Bern approach. Sequents again contain an extra component, the *history*. Loops are found earlier at the cost of some extra data storage. In some cases this dramatically cuts the search time, but in general makes it slightly slower.

8.6 System $LJpm^*$ of Mints

Mints' inference rules [52] operate on *tableaux*, i.e. lists \mathcal{T} of multi-succedent sequents (the *components* of \mathcal{T}). We use “;” for the “append” operation on lists, where [52] uses a “ \star ”; and, for emphasis, we parenthesise components. A tableau is *initial* iff **one** of its components is an initial sequent. A *proof* is a tree, each leaf of which is an initial tableau. Use of (Mints') tableaux rather than just of sequents avoids backtracking at the meta-level: all the inference rules are invertible.

$$\frac{\mathcal{T}; (A, A \vee B, \Gamma \Longrightarrow \Delta); \mathcal{T}' \quad \mathcal{T}; (B, A \vee B, \Gamma \Longrightarrow \Delta); \mathcal{T}'}{\mathcal{T}; (A \vee B, \Gamma \Longrightarrow \Delta); \mathcal{T}'} L\vee$$

$$\frac{\mathcal{T}; (\Gamma \Longrightarrow \Delta, A \vee B, A, B); \mathcal{T}'}{\mathcal{T}; (\Gamma \Longrightarrow \Delta, A \vee B); \mathcal{T}'} R\vee$$

Conjunctive branching (as in $L\vee$) replaces one tableau by two, whereas disjunctive branching (**not yet illustrated**) adds components to a tableau. [Not the same usage as Fitting.] Here are the rules for implication:

$$\frac{\mathcal{T}; (A \rightarrow B, \Gamma \Longrightarrow \Delta, A); \mathcal{T}' \quad \mathcal{T}; (B, A \rightarrow B, \Gamma \Longrightarrow \Delta); \mathcal{T}'}{\mathcal{T}; (A \rightarrow B, \Gamma \Longrightarrow \Delta); \mathcal{T}'} L\rightarrow$$

$$\frac{\mathcal{T}; (\Gamma \Longrightarrow \Delta, A \rightarrow B); (A, \Gamma \Longrightarrow B); \mathcal{T}'}{\mathcal{T}; (\Gamma \Longrightarrow \Delta, A \rightarrow B); \mathcal{T}'} R\rightarrow$$

in which note the conjunctive branching in the first (i.e. the rule has two premisses) and the disjunctive branching (by addition of the new component $(A, \Gamma \Longrightarrow B)$ to the tableau) in the second. Note that the principal formula is always, except in $R\rightarrow$, duplicated into the premisses. $\Gamma \Longrightarrow \Delta$ *subsumes* $\Gamma' \Longrightarrow \Delta'$ iff $\Gamma \subseteq \Gamma'$ and $\Delta \subseteq \Delta'$ (as sets of formulae); then one forbids any tableau extension step if some new sequent subsumes some component of some tableau lower down the tree, i.e. loops must be detected (might be costly) and avoided. A finiteness argument then shows that this ensures termination.

8.7 System IG^r of Corsi and Tassi

We present (in our own 2-D style) the system IG^r of Corsi and Tassi [10] (implicational part: the other parts present no difficulties). Its main features are (a) that it is depth-bounded (b) that it has the subformula property and (c) bicompleteness. *Initial sequents* are, as usual, those with an atom on both left and right.) The superfix r stands for a regularity condition, enforced by the use of \mathcal{B} and \mathcal{H} .

A proof in the **G4ip** calculus looks (in tree form) much the same but with fewer formulae at each node and without branching, using the rule $L0\rightarrow$ rather than $L\rightarrow$. (Here is the **IG'** proof again in linear style, using a one-dimensional layout $\mathcal{B}; \mathcal{H} : \Gamma \Longrightarrow \Delta$ for each sequent)

$$\begin{array}{ll}
[p \rightarrow q]; [p \rightarrow r, (p \rightarrow q) \rightarrow p \rightarrow r, (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [p, p \rightarrow q, p \rightarrow q \rightarrow r] \Longrightarrow [p, r] & (1) \\
[p \rightarrow q \rightarrow r, p \rightarrow q]; [p \rightarrow r, (p \rightarrow q) \rightarrow p \rightarrow r, (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [q, p, p \rightarrow q \rightarrow r] \Longrightarrow [p, r] & (2) \\
[q \rightarrow r, p \rightarrow q \rightarrow r, p \rightarrow q]; [p \rightarrow r, (p \rightarrow q) \rightarrow p \rightarrow r, (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [q \rightarrow r, q, p] \Longrightarrow [q, r] & (3) \\
[q \rightarrow r, p \rightarrow q \rightarrow r, p \rightarrow q]; [p \rightarrow r, (p \rightarrow q) \rightarrow p \rightarrow r, (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [r, q, p] \Longrightarrow [r] & (4) \\
[p \rightarrow q \rightarrow r, p \rightarrow q]; [p \rightarrow r, (p \rightarrow q) \rightarrow p \rightarrow r, (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [q \rightarrow r, q, p] \Longrightarrow [r] & \text{By } L\rightarrow \text{ from 3, 4} \quad (5) \\
[p \rightarrow q]; [p \rightarrow r, (p \rightarrow q) \rightarrow p \rightarrow r, (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [q, p, p \rightarrow q \rightarrow r] \Longrightarrow [r] & \text{By } L\rightarrow \text{ from 2, 5} \quad (6) \\
[] : [p \rightarrow r, (p \rightarrow q) \rightarrow p \rightarrow r, (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [p, p \rightarrow q, p \rightarrow q \rightarrow r] \Longrightarrow [r] & \text{By } L\rightarrow \text{ from 1, 6} \quad (7) \\
[] : [(p \rightarrow q) \rightarrow p \rightarrow r, (p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [p \rightarrow q, p \rightarrow q \rightarrow r] \Longrightarrow [p \rightarrow r] & \text{By } R\rightarrow \text{ from 7} \quad (8) \\
[] : [(p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] : [p \rightarrow q \rightarrow r] \Longrightarrow [(p \rightarrow q) \rightarrow p \rightarrow r] & \text{By } R\rightarrow \text{ from 8} \quad (9) \\
[] : [] \Longrightarrow [(p \rightarrow q \rightarrow r) \rightarrow (p \rightarrow q) \rightarrow p \rightarrow r] & \text{By } R\rightarrow \text{ from 9} \quad (10)
\end{array}$$

The same calculus was rediscovered in 2013–2014 by Goré et al., renamed “IntHistGC” and implemented with several optimisations such as caching and back-jumping [34].

8.8 The Calculus SIC of Corsi and Tassi

SIC is a variant of the system **IG'** in the same paper [10]; the essential difference is that backtracking (because of disjunctive branching) is incorporated into the calculus, and thus each node of the tree is a stack of ordinary sequents rather than just one such sequent. This is very similar to Mints’ notion of tableau. Sequents are pushed onto the stack to indicate all the alternative possibilities (according to the different implicational succedent formulae); as they are tried and found unsolvable, they are popped, and failure occurs when the stack is empty. The goal is thus achieved (as reflected in the paper’s title, “Intuitionistic logic freed of all metarules”) that all use of “global metarules” is thus replaced by use of “local metarules”, incorporated into the rules of the calculus.

8.9 The Calculus LSJ of Ferrari, Fiorentini and Fiorino

Sequents are [25] of the form $\Gamma \xrightarrow{\Theta} \Delta$, the components Γ, Θ, Δ being sets (of formulae) rather than multisets. Let us use $<$ for \leq without equality.

The semantics (using only finite models) of such a sequent is that $(K, V, w) \Vdash \Gamma \xrightarrow{\Theta} \Delta$ iff, whenever both

1. for every $H \in \Theta$ and $w' \in K$ with $w < w'$, one has $(K, V, w') \Vdash H$,
2. for every $G \in \Gamma$, one has $(K, V, w) \Vdash G$,

then for some $D \in \Delta$ one has $(K, V, w) \Vdash D$. Negation is defined as usual. We omit the rules for disjunction, dual to those for conjunction. The rules are

$$\begin{array}{c}
\frac{}{\perp, \Gamma \stackrel{\Theta}{\Rightarrow} \Delta} L\perp \qquad \frac{}{A, \Gamma \stackrel{\Theta}{\Rightarrow} \Delta, A} Id \\
\\
\frac{A, B, \Gamma \stackrel{\Theta}{\Rightarrow} \Delta}{A \wedge B, \Gamma \stackrel{\Theta}{\Rightarrow} \Delta} L\wedge \qquad \frac{\Gamma \stackrel{\Theta}{\Rightarrow} \Delta, A \quad \Gamma \stackrel{\Theta}{\Rightarrow} \Delta, B}{\Gamma \stackrel{\Theta}{\Rightarrow} \Delta, A \wedge B} R\wedge \\
\\
\frac{B, \Gamma \stackrel{\Theta}{\Rightarrow} \Delta \quad \Gamma \stackrel{B, \Theta}{\Rightarrow} \Delta, A \quad \Theta, \Gamma \stackrel{B}{\Rightarrow} A}{A \rightarrow B, \Gamma \stackrel{\Theta}{\Rightarrow} \Delta} L\rightarrow \qquad \frac{A, \Gamma \stackrel{\Theta}{\Rightarrow} \Delta, B \quad A, \Theta, \Gamma \stackrel{\{\}}{\Rightarrow} B}{\Gamma \stackrel{\Theta}{\Rightarrow} \Delta, A \rightarrow B} R\rightarrow
\end{array}$$

A syntactic proof of cut-admissibility for this calculus seems difficult; a semantic proof is in [25]. Using our own implementation of **LSJ**, with Prolog cuts to prune the search space wherever seemed appropriate, the first (indeed, only) proof we found of the formula that is the type of the *S* combinator is 87 lines long. It is possible that, with differently placed cuts in the implementation, a shorter proof would be found. An associated calculus, building on the approach of [57], gives bicompleteness.

8.10 The Calculus GLJ

We recall Sambin and Valentini's system **GLS'** from [61] for the classical provability logic **GL** (implicational and modal part: the other parts are standard). Antecedents and succedents are sets; so Γ, A and Γ, Δ stand for the unions $\{A\} \cup \Gamma$ and $\Gamma \cup \Delta$ (and similarly for A, Γ). When we see $A \supset B, \Gamma$ in the conclusion of a schematic rule, it is implicit that $A \supset B \notin \Gamma$; similarly with Δ rather than Γ , and for other connectives. Actual proofs may for clarity be written with apparent repetitions. All rules (except *RR*) are invertible—[61] calls this *doubly sound*. *Initial sequents* are, as is almost usual, those with a formula common to both left and right (or with \perp on the left). The rules (in each of which Γ and Δ are disjoint, and with Π and Σ disjoint sets of atoms) are:

$$\frac{\Gamma \Rightarrow \Delta, A \quad B, \Gamma \Rightarrow \Delta}{A \supset B, \Gamma \Rightarrow \Delta} L\supset \quad \frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \supset B} R\supset \quad \frac{\Gamma, \Box\Gamma, \Box D \Rightarrow D}{\Pi, \Box\Gamma \Rightarrow \Box\Delta, \Box D, \Sigma} RR$$

The rule *RR* has the property that if the conclusion is valid then, for some choice of a boxed formula in its succedent, the corresponding premiss is valid. ([61] also calls this *doubly sound*.)

Root-first proof search in **GLS'** terminates. The argument (from [61]) is as follows. First, used root-first, every rule other than *RR* reduces the number of connectives. Second, as we proceed up a branch, the set of boxed formulae in the antecedent occasionally expands but never shrinks: thus, if a sequent $\Pi, \Box\Gamma \Rightarrow \Box\Delta, \Sigma$ is a conclusion of *RR*, the antecedent of every sequent above it will contain a formula $\Box D$, with $D \in \Delta$, and the antecedent of every sequent at or below it cannot contain such a formula (since search is required to stop at initial sequents). So all the sequents in a branch are different. By the subformula property their number is finite, so search along any branch terminates.

Consider the standard embedding \cdot^\square of **Int** into **GL**, in which notice the distinctions between *classical* and *intuitionistic* implication, $A \supset B$ and $A \rightarrow B$, and between *classical* and *intuitionistic* negation, $\sim A$ and $\neg A$:

$$\begin{aligned} \perp^\square &:= \perp & P^\square &:= P \wedge \Box P \\ (A \wedge B)^\square &:= A^\square \wedge B^\square & (A \vee B)^\square &:= A^\square \vee B^\square \\ (\neg A)^\square &:= \sim A^\square \wedge \Box(\sim A^\square) & (A \rightarrow B)^\square &:= (A^\square \supset B^\square) \wedge \Box(A^\square \supset B^\square) \end{aligned}$$

The interpretations of the intuitionistic implication rules are then

$$\frac{\Gamma^\square, (A \rightarrow B)^\square \Rightarrow A^\square, \Delta^\square \quad \Gamma^\square, B^\square \Rightarrow \Delta^\square}{\Gamma^\square, (A \rightarrow B)^\square \Rightarrow \Delta^\square} L \rightarrow^\square \quad \frac{\Gamma^\square, A^\square \Rightarrow B^\square}{\Gamma^\square \Rightarrow (A \rightarrow B)^\square, \Delta^\square} R \rightarrow^\square$$

and these need to be justified as sound rules in **GLS'** (in which *Weakening* is known to be admissible). This is routine.

We now specialise the rules of **GLS'** and show them in the language of **Int**, giving us a novel calculus **GLJ**. In the following, P ranges over atoms, Π and Σ are sets of atoms; Θ and Ψ are sets of either atoms, classical negations or classical implications, implicitly treated as boxed; Γ and Δ are arbitrary sets of formulae—and all are just formulae of **Int**, apart from the classical negations and implications (in Θ and Ψ).

Sequents are now of the form $\Pi; \Theta; \Gamma \Rightarrow \Delta; \Psi; \Sigma$. Provability of a formula A will match the derivability of the sequent $[\Box A]; [\Box]; [\Box] \Rightarrow A; [\Box]; [\Box]$. We use the classical notation $A \supset B$ for implications after being moved by rule $L \rightarrow$ from Γ to Θ (or by rule $R \rightarrow$ from Δ to Ψ), so that, when moved back to Γ by variants of RR , they are correctly analysed (by $L \supset$).

$$\begin{aligned} & \overline{[P], \Pi; \Theta; \Gamma \Rightarrow \Delta; \Psi; \Sigma, [P]} \quad Ax1 \quad \overline{\Pi; [A], \Theta; \Gamma \Rightarrow \Delta; \Psi; [A]; \Sigma} \quad Ax2 \\ & \frac{}{\Pi; \Theta; [\Box \perp], \Gamma \Rightarrow \Delta; \Psi; \Sigma} L \perp \quad \frac{\Pi; \Theta; \Gamma \Rightarrow \Delta; \Psi; \Sigma}{\Pi; \Theta; \Gamma \Rightarrow \Delta, [\Box \perp]; \Psi; \Sigma} R \perp \\ & \frac{P, \Pi; P, \Theta; \Gamma \Rightarrow \Delta; \Psi; \Sigma}{\Pi; \Theta; [P], \Gamma \Rightarrow \Delta; \Psi; \Sigma} LA t \\ & \frac{\Pi; \Theta; \Gamma \Rightarrow \Delta; \Psi; P, \Sigma \quad \Pi; \Theta; \Gamma \Rightarrow \Delta; P, \Psi; \Sigma}{\Pi; \Theta; \Gamma \Rightarrow [P], \Delta; \Psi; \Sigma} RA t \\ & \frac{\Pi; \Theta; A, B, \Gamma \Rightarrow \Delta; \Psi; \Sigma}{\Pi; \Theta; [A \wedge B], \Gamma \Rightarrow \Delta; \Psi; \Sigma} L \wedge \\ & \frac{\Pi; \Theta; \Gamma \Rightarrow \Delta, A; \Psi; \Sigma \quad \Pi; \Theta; \Gamma \Rightarrow \Delta, B; \Psi; \Sigma}{\Pi; \Theta; \Gamma \Rightarrow \Delta, [A \wedge B]; \Psi; \Sigma} R \wedge \end{aligned}$$

Rules for \vee are dual to those for \wedge , so need not be shown here.

$$\begin{array}{c}
\frac{\Pi; A \supset B, \Theta; \Gamma \Longrightarrow \Delta, A; \Psi; \Sigma \quad \Pi; A \supset B, \Theta; B, \Gamma \Longrightarrow \Delta; \Psi; \Sigma}{\Pi; \Theta; \boxed{A \rightarrow B}, \Gamma \Longrightarrow \Delta; \Psi; \Sigma} L_{\rightarrow} \\
\\
\frac{\Pi; \Theta; \Gamma \Longrightarrow \Delta, A; \Psi; \Sigma \quad \Pi; \Theta; B, \Gamma \Longrightarrow \Delta; \Psi; \Sigma}{\Pi; \Theta; \boxed{A \supset B}, \Gamma \Longrightarrow \Delta; \Psi; \Sigma} L_{\supset} \\
\\
\frac{\Pi; \Theta; A, \Gamma \Longrightarrow \Delta, B; \Psi; \Sigma \quad \Pi; \Theta; \Gamma \Longrightarrow \Delta; \Psi, A \supset B; \Sigma}{\Pi; \Theta; \Gamma \Longrightarrow \Delta, \boxed{A \rightarrow B}; \Psi; \Sigma} R_{\rightarrow} \\
\\
\frac{\Pi; \sim A, \Theta; \Gamma \Longrightarrow \Delta, A; \Psi; \Sigma}{\Pi; \Theta; \boxed{\neg A}, \Gamma \Longrightarrow \Delta; \Psi; \Sigma} L_{\neg} \quad \frac{\Pi; \Theta; \Gamma \Longrightarrow \Delta, A; \Psi; \Sigma}{\Pi; \Theta; \boxed{\sim A}, \Gamma \Longrightarrow \Delta; \Psi; \Sigma} L_{\sim} \\
\\
\frac{\Pi; \Theta; A, \Gamma \Longrightarrow \Delta; \Psi; \Sigma \quad \Pi; \Theta; \Gamma \Longrightarrow \Delta; \Psi, \sim A; \Sigma}{\Pi; \Theta; \Gamma \Longrightarrow \Delta, \boxed{\neg A}; \Psi; \Sigma} R_{\neg}
\end{array}$$

Finally, we have rules corresponding to the RR rule of **GLS'**. There are three of these, since the formulae in Ψ can be either atoms, classical implications or classical negations. These are the only rules that are not invertible.

$$\frac{[]; A \supset B, \Theta; A, \Theta \Longrightarrow B; []; []}{\Pi; \Theta; [] \Longrightarrow []; \Psi, \boxed{A \supset B}; \Sigma} RR_{\supset} \quad \frac{[]; \sim A, \Theta; A, \Theta \Longrightarrow []; []; []}{\Pi; \Theta; [] \Longrightarrow []; \Psi, \boxed{\sim A}; \Sigma} RR_{\sim} \quad \frac{\Pi; P, \Theta; \Theta \Longrightarrow []; []; P}{\Pi; \Theta; [] \Longrightarrow []; \Psi, \boxed{P}; \Sigma} RRA_t$$

In these rules RR_{\supset} , RR_{\sim} and RRA_t , it is required that Π and Σ are disjoint and Θ is disjoint from (respectively) Ψ , $A \supset B$, from Ψ , $\sim A$ and from Ψ , P .

This calculus **GLJ** doesn't quite have the subformula property: for example, L_{\rightarrow} and R_{\rightarrow} turn intuitionistic implications into classical implications. To obtain it, we can either decree that $A \supset B$ is a subformula of $A \rightarrow B$ and that $\sim A$ is a subformula of $\neg A$, or adjust the calculus slightly (at the expense of some extra search). But it does have the termination property, by an extension of the argument for **GLS'** above. Countermodel construction from failed searches seems to be routine. But, the first proof found (of the type of the S combinator) has 5,185 lines. **GLJ** is thus presented as an example, which could have been written down at any time after 1982, of a complete calculus with an easy termination argument—but not as efficient as **G4ip** or **m-G4ip**.

9 Labelled Calculi

Many authors (Castellini, Catach, Fitting, Gabbay, Kanger, Maslov, Negri, Russo, Schmidt, Simpson, Tishkovsky, Vigano, ...) have exploited labels (aka “prefixes”) in sequent calculi (or tableau calculi), one motivation being to make the inference

rules invertible (and another being to allow uniform development of analytic calculi from frame conditions rather than from axioms). Some have criticised this as a lack of syntactic purity, i.e. as the presence of “semantic pollution”; others defend it as allowing calculi for otherwise unmanageable logics. Read [59] mounts a strong defence. Goré has a useful survey [32] in the context of modal logics. Using labelled tableaux, Schmidt and Tishkovsky have implemented a generic tableau calculus generator [62], geared rather towards description logics; this can generate a JAVA-based prover, or could be combined with a tableau-based theorem prover such as LOTREC [46] or the Tableaux Work Bench [1].

For **Int**, and using sequent calculus notation rather than tableaux, one statement of the method is by Dyckhoff and Negri [20]. This covers a wide range of intermediate logics—all those where the first-order frame conditions in Kripke semantics can be presented as geometric (aka coherent) implications, i.e. (in fact) all that can be presented semantically using first-order formulae, since every first-order theory has a coherent conservative extension [21, 22]. This approach solves the problem of backtracking; but termination is a problem, with various approaches, including the “unrestricted blocking” rule of [62] and another method in [30, 53].

9.1 Calculus G3i

The calculus just mentioned (by Dyckhoff and Negri [20]) is as follows (rules for \vee are omitted, being dual to those for \wedge):

$$\begin{array}{c}
 \frac{\overline{x : \perp}, \Gamma \Longrightarrow \Delta}{x : A, x : B, \Gamma \Longrightarrow \Delta} L\perp \quad \frac{\overline{x \leq y, x : P}, \Gamma \Longrightarrow \Delta, \overline{y : P}}{\Gamma \Longrightarrow \Delta, x : A \quad \Gamma \Longrightarrow \Delta, x : B} Ax \\
 \frac{x : A, x : B, \Gamma \Longrightarrow \Delta}{\overline{x : A \wedge B}, \Gamma \Longrightarrow \Delta} L\wedge \quad \frac{\Gamma \Longrightarrow \Delta, x : A \quad \Gamma \Longrightarrow \Delta, x : B}{\Gamma \Longrightarrow \Delta, \overline{x : A \wedge B}} R\wedge \\
 \frac{x \leq y, x : A \rightarrow B, \Gamma \Longrightarrow \Delta, y : A \quad x \leq y, x : A \rightarrow B, y : B, \Gamma \Longrightarrow \Delta}{\overline{x \leq y, x : A \rightarrow B}, \Gamma \Longrightarrow \Delta} L\rightarrow \\
 \frac{x \leq y, y : A, \Gamma \Longrightarrow \Delta, y : B}{\Gamma \Longrightarrow \Delta, \overline{x : A \rightarrow B}} R\rightarrow \\
 \frac{x \leq x, \Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta} Ref \quad \frac{x \leq z, x \leq y, y \leq z, \Gamma \Longrightarrow \Delta}{\overline{x \leq y, y \leq z}, \Gamma \Longrightarrow \Delta} Trans
 \end{array}$$

with y fresh in $R\rightarrow$, i.e. not occurring in the conclusion.

Derivations can be restricted to those in which the label x used in the *Ref* rule already occurs in the conclusion.

This calculus does not terminate (e.g. on Peirce’s formula).

Negri [53] shows how to add a loop-checking mechanism to ensure termination and build finite counter-models directly from a failed proof search. The effect on complexity isn't clear; the loop-checking is expensive. Further details of related methods are in Garg et al. [30]; there is also related work by Schmidt et al. [62] and by Antonsen and Waaler [2].

10 Focused Calculi

Naive implementations of the calculi mentioned above spend a great deal of time looking along lists to find a formula of a certain form.

A better approach is to take the next formula and either analyse it (i.e. generate appropriate subproblems) or put it aside in a suitable place for later use. For example, atomic formulae can be examined (to see if the branch closes) or (if that fails) put into a list of atoms; and succedent conjunctions can be put aside until all non-branching rules have been dealt with. This can be regarded as a naive form of *focusing*. So can, to some extent, the calculus **G4ip**, with its connections to the focused calculus **LJQ**.

But several authors, notably McLaughlin and Pfenning [49], have more logic-based approaches. For lack of space, we omit their presentation.

11 Challenges and Open Problems

1. Find a simple calculus for **Int** that (a) has the termination property (ideally, with linear depth) and (b) avoids backtracking through rules, but without implementing the usual meta-level “list of disjunctive goals to be tried one after another”. This can be done for classical logic and for Gödel-Dummett logic [19]. Is there a fundamental complexity result (yet to be discovered) that forbids this? Note that linear temporal logic is PSPACE-hard but has a terminating calculus [7] with all rules invertible. (Termination here depends on a form of history mechanism.) Is there a combination of the **G4ip** ideas and labelling that solves this problem?
2. Find, develop and simplify uniform methods for ensuring termination in labelled calculi.
3. Find syntactic (i.e. non-semantic) methods for proving cut admissibility for calculi with sequents with several components, e.g. **LSJ** and **GLJ**.
4. Is there a calculus that combines the good features of **G4ip** (where it is the nested implications—formulae of type (4) and their variant (7)—that are problematic) and Mints' resolution method (where these are less of a problem: his class $[0,1,2,4,5,7]$ is in P). Or do we get, not the good, but the bad features of both?
5. Develop more proofs of correctness and completeness using proof assistants like *NuPRL*, *Coq* and *Agda*, extending work of Underwood [69], Caldwell [8], Weich [73, 74] and allowing extraction of verified software in (e.g.) Haskell, Scheme or

OCaml. There is some recent work (unpublished) by Larchey-Wendling on **LSJ** (and on **G4ip**) in this direction.

Acknowledgments Thanks are especially due to Gerhard Jäger and Helmut Schwichtenberg, whose scientific encouragement over the years has been substantial; and to Grisha Mints, now, alas, no longer with us, for helpful comments on historical matters—regrettably not all incorporated (thanks to a failure of technology).

References

1. P. Abate, R. Goré, The Tableaux Work Bench, in *Proceedings of IJCAR 2003*. LNCS, vol. 2796 (Springer, 2003), pp. 230–236
2. R. Antonsen, A. Waaler, A labelled system for IPL with variable splitting, in *Proceedings of CADE 2007*. LNAI, vol. 4603 (Springer, 2007), pp. 132–146
3. A. Avellone, G. Fiorino, U. Moscato, An implementation of a $O(n \log n)$ -SPACE decision procedure for propositional intuitionistic logic, in *3rd International Workshop on the Implementation of Logics* (Tbilisi, Georgia, Oct 2002)
4. B. Barras, S. Boutin, et al., The Coq proof assistant reference manual, version 6.2.1. Technical Report, INRIA (2000). <http://www.fpt.inria.fr>
5. E.W. Beth, *The Foundations of Mathematics* (North-Holland, 1959)
6. M. Bezem, T. Coquand, Automating coherent logic, in *Proceedings of LPAR 2005*. LNCS, vol. 3835 (Springer, 2005), pp. 246–260
7. K. Brünnler, M. Lange, Cut-free sequent systems for temporal logic. *J. Logic Algebraic Program.* **76**, 216–225 (2008)
8. J. Caldwell, Decidability extracted: synthesizing “correct-by-construction” decision procedures from constructive proofs. Ph.D. dissertation (Cornell University, 1998)
9. L. Catach, TABLEAUX: a general theorem prover for modal logics. *J. Autom. Reason.* **7**, 489–510 (1991)
10. G. Corsi, G. Tassi, Intuitionistic logic freed of all metarules. *J. Symb. Logic* **72**, 1204–1218 (2007)
11. H. Curry, *Foundations of Mathematical Logic* (Dover Publications, 1963)
12. K. Došen, A note on Gentzen’s decision procedure for intuitionistic propositional logic. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* **33**, 453–456 (1987)
13. A.G. Dragalin, *Mathematical Intuitionism, Translations of Mathematical Monographs* 67 (Trans. E. Mendelson). (American Mathematical Society, Providence, R.I., 1988)
14. R. Dyckhoff, Contraction-free sequent calculi for intuitionistic logic. *J. Symb. Logic* **57**, 795–807 (1992)
15. R. Dyckhoff, Yet Another Proof Engine, MS (available from the author) (2014)
16. R. Dyckhoff, D. Kesner, S. Lengrand, Strong cut-elimination systems for Hudelmaier’s depth-bounded sequent calculus for implicational logic, in *IJCAR 2006 Proceedings*. LNCS, vol. 4130, pp. 347–361 (2006)
17. R. Dyckhoff, S. Lengrand, LJQ: a strongly focused calculus for intuitionistic logic, in *Proceedings of Computability in Europe 2006*. LNCS, vol. 3988 (Springer, 2006), pp. 173–185
18. R. Dyckhoff, S. Negri, Admissibility of structural rules for contraction-free systems of intuitionistic logic. *J. Symb. Logic* **65**, 1499–1518 (2000)
19. R. Dyckhoff, S. Negri, Decision methods for linearly ordered Heyting algebras. *Arch. Math. Log.* **45**, 411–422 (2006)
20. R. Dyckhoff, S. Negri, Proof analysis for intermediate logics. *Arch. Math. Log.* **51**, 71–92 (2012)
21. R. Dyckhoff, S. Negri, Geometrisation of first-order logic, *B. Symb. Logic* **21**, 123–163 (2015) *Geometrisation of First-order Formulae*, Submitted June 2014

22. R. Dyckhoff, S. Negri, Coherentisation of accessibility conditions in labelled sequent calculi, in *Extended Abstract (2 pp.)*, *Gentzen Systems and Beyond 2014, Informal Proceedings*, ed. by R. Kuznets & G. Metcalfe. Vienna Summer of Logic, July 2014
23. U. Egly, S. Schmitt, On intuitionistic proof transformations, their complexity, and application to constructive program synthesis. *Fundamenta Informaticae* **39**, 59–83 (1999)
24. M. Ferrari, C. Fiorentini, G. Fiorino, Simplification rules for intuitionistic propositional tableaux. *ACM Trans. Comput. Log.* **13**, 14:1–14:23 (2012)
25. M. Ferrari, C. Fiorentini, G. Fiorino, Contraction-free linear depth sequent calculi for intuitionistic propositional logic with the subformula property and minimal depth counter-models. *J. Autom. Reason.* **51**, 129–149 (2013)
26. G. Fiorino, Decision procedures for propositional intermediate logics. Ph.D. thesis (Milan University, 2000)
27. M. Fitting, *Intuitionistic Logic, Model Theory and Forcing* (North-Holland, 1969)
28. M. Fitting, *Proof Methods for Modal and Intuitionistic Logic* (Reidel, 1983)
29. T. Franzén, Algorithmic aspects of intuitionistic propositional logic, I and II, SICS Research Reports R870X and R8906, 1987 and 1989
30. D. Garg, V. Genovese, S. Negri, Countermodels from sequent calculi in multi-modal logics, in *Proceedings of LICS 2012* (IEEE, 2012), pp. 315–324
31. G. Gentzen, Untersuchungen über das logische Schliessen. *Math. Zeitschrift* **39**, 176–210, 405–431 (1935)
32. R. Goré, Tableau methods for modal and temporal logics, in *Handbook of Tableau Methods* (Kluwer, 1999), pp. 297–396
33. R. Goré, J. Thomson, BDD-based automated reasoning in propositional non-classical logics: progress report, in *Proceedings of PAAR-2012, EPiC Series 21* (EasyChair, 2013), pp. 43–57
34. R. Goré, J. Thomson, J. Wu, A history-based theorem prover for intuitionistic propositional logic using global caching: IntHistGC system description, in *Proceedings of IJCAR 2014*. LNAI, vol. 8562 (Springer, 2014), pp. 262–268
35. J. Goubault-Larrecq, Implementing tableaux by decision diagrams, Unpublished note, Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, 47 pp. (1996)
36. H. Herbelin, Séquents qu'on calcule. Thèse de Doctorat, Université Paris 7 (1995)
37. A. Heuerding, M. Seyfried, H. Zimmermann, Efficient loop-check for backward proof search in some non-classical logics, in *Proceedings of Tableaux 1996*. LNAI, vol. 1071 (Springer, 1996), pp. 210–225
38. J. Howe, Two loop-detection mechanisms: a comparison, in *Proceedings of Tableaux 1997*. LNCS, vol. 1227 (Springer, 1997), pp. 188–200
39. J. Hudelmaier, A Prolog program for intuitionistic propositional logic, SNS-Bericht 88–28, Tübingen (1988)
40. J. Hudelmaier, Bounds for cut elimination in intuitionistic propositional logic. *Arch. Math. Logic* **31**, 331–353 (1992)
41. J. Hudelmaier, An $O(n \log n)$ -SPACE decision procedure for intuitionistic propositional logic. *J. Logic Comput.* **3**, 63–76 (1993)
42. O. Ketonen, Untersuchungen zum Prädikatenkalkül. *Annales Acad. Sci. Fenn. Ser. A.I.* **23** (1944)
43. S.C. Kleene, *Introduction to Metamathematics* (North-Holland, 1952)
44. D. Larchey-Wendling, D. Mery, D. Galmiche, STRIP: structural sharing for efficient proof-search, in *Proceedings of IJCAR 2001*. LNCS, vol. 2083 (Springer, 2001), pp. 696–700
45. P. Lincoln, A. Scedrov, N. Shankar, Linearizing intuitionistic implication. *Ann. Pure Appl. Logic* **60**, 151–177 (1993)
46. O. Gasquet, A. Herzig, D. Longin, M. Sahade, LoTREC: logical tableaux research engineering companion, in *Proceedings of Tableaux 2005*. LNCS, vol. 3702 (Springer, 2005), pp. 318–322
47. S. Maehara, Eine Darstellung der Intuitionistischen Logik in der Klassischen. *Nagoya Math. J.* **7**, 45–64 (1954)
48. S.Yu. Maslov, An inverse method of establishing deducibility in the classical predicate calculus. *Dokl. Akad. Nauk. SSSR* **159**, 17–20 (translated as *Soviet Math. Dokl.* **5**, 1420) (1964)

49. S. McLaughlin, F. Pfenning, Imogen: focusing the polarized inverse method for intuitionistic propositional logic, in *Proceedings of LPAR'08*. LNCS, vol. 5330 (Springer, 2008), pp. 174–181
50. G. Mints, Gentzen-type systems and resolution rule. Part I. LNCS **417**, 198–231 (1990)
51. G. Mints, Complexity of subclasses of the intuitionistic propositional calculus, *Programming Logic* (ed. by B. Nordström). BIT **31**, 64–69 (1992)
52. G. Mints, *A Short Introduction to Intuitionistic Logic*, CSLI Stanford Lecture Notes (Springer, 2000)
53. S. Negri, Proofs and countermodels in non-classical logics. *Logica Universalis* **8**, 25–60 (2014)
54. K. Ono, Logische Untersuchungen über die Grundlagen der Mathematik. J. Fac. Sci. Imperial Univ. Tokyo. Section I. Math. Astron. Phys. Chem. **3**, 329–389 (1938)
55. J. Otten, Clausal connection-based theorem proving in intuitionistic first-order logic, in *Proceedings of Tableaux 2005*. LNCS, vol. 3702 (Springer, 2005), pp. 245–261
56. J. Otten, The ILTP Library. <http://www.cs.uni-potsdam.de/ti/iltp/>
57. L. Pinto, R. Dyckhoff, Loop-free construction of counter-models for intuitionistic propositional logic, *Symposia Gaussiana, Conf. A*, ed. by M. Behara, R. Fritsch, R.G. Lintz (Walter de Gruyter & Co, Berlin, 1995), pp. 225–232
58. J. von Plato, *Saved from the Cellar: Gerhard Gentzen's Shorthand Notes on Logic and Foundations of Mathematics* Springer (to appear, 2016)
59. S.L. Read, Semantic Pollution and Syntactic Purity, *R. Symb. Logic* **8**, 649–691 (2015)
60. D. Sahlin, T. Franzén, S. Haridi, An intuitionistic predicate logic theorem prover. *J. Logic Comput.* **2**, 619–656 (1992)
61. G. Sambin, S. Valentini, The modal logic of provability. The sequential approach. *J. Philos. Logic* **11**, 311–342 (1982)
62. R. Schmidt, D. Tishkovsky, Automated synthesis of tableau calculi. *Logical Methods Comput. Sci.* **7**, 32 (2011)
63. K. Schütte, *Vollständige Systeme modaler und intuitionistischer Logik*, *Ergebnisse der Mathematik* (Springer, 1968)
64. W. Sieg, S. Cittadini, Normal natural deduction proofs (in non-classical logics), in *LNAI 2605* (Springer, 2005), pp. 169–191
65. R. Statman, Intuitionistic propositional logic is polynomial-space complete. *Theoret. Comput. Sci.* **9**, 67–72 (1979)
66. T. Tammet, A resolution theorem prover for intuitionistic logic, in *CADE-13*. LNCS, vol. 1104 (Springer, 1996), pp. 2–16
67. N. Tennant, *Autologic* (Edinburgh University Press, 1992)
68. A.S. Troelstra, H. Schwichtenberg, *Basic Proof Theory* (Cambridge, 2001)
69. J. Underwood, A constructive completeness proof for intuitionistic propositional calculus, TR 90–1179, Department of Computer Science, Cornell University, 1990; also in *Proceedings of the Workshop on Analytic Tableaux* (Marseille, 1993)
70. N.N. Vorob'ev, The derivability problem in the constructive propositional calculus with strong negation. *Doklady Akademii Nauk SSSR* **85**, 689–692 (1952)
71. N.N. Vorob'ev, A new algorithm for derivability in the constructive propositional calculus. *AMS Transl. Ser.* **2**(94), 37–71 (1970)
72. A. Waaler, L. Wallen, Tableaux methods in intuitionistic logic, in *Handbook of Tableaux Methods*, ed. by M. D'Agostino, D.M. Gabbay, R. Hähnle, J. Posegga (Kluwer, Dordrecht, 1999), pp. 255–296
73. K. Weich, Improving proof search in intuitionistic propositional logic. Munich Ph.D. thesis, also from Logos Verlag Berlin (2001)
74. K. Weich, Decision procedures for intuitionistic propositional logic by program extraction, in *Proceedings of Tableaux 1998*. LNCS, vol. 1397 (Springer, 1998), pp. 292–306