Some Formal Considerations on Gabbay's Restart Rule in Natural Deduction and Goal-Directed Reasoning

Michael Gabbay and Murdoch Gabbay

1 Introduction

In this paper we make some observations about Natural Deduction derivations [Prawitz, 1965, van Dalen, 1986, Bell and Machover, 1977]. We assume the reader is familiar with it and with proof-theory in general. Our development will be simple, even simple-minded, and concrete. However, it will also be evident that general ideas motivate our examples, and we think both our specific examples and the ideas behind them are interesting and may be useful to some readers.

In a sentence, the bare technical content of this paper is: *Extending natural deduction with global well-formedness conditions can neatly and cheaply capture classical and intermediate logics.*

The interest here is in the 'neatly' and 'cheaply'. By 'neatly' we mean 'preserving proof-normalisation',[1] and 'maintaining the subformula property', and by 'cheaply' we mean 'preserving the formal structure of deductions' (so that a deduction in the original system is still, formally, a deduction in the extended system, and in particular it requires no extra effort to write just because it is in the extended system).

To illustrate what we have in mind consider intuitionistic first-order logic **(FOL)** [van Dalen, 1986] as a paradigmatic example of a formal notion of deduction. A natural deduction derivation (or **deduction**) is an inductively defined tree structure where each node contains an instance of a formula. A deduction is **valid** when each successive node follows from its predecessors in accordance with some predetermined inference rules.

A particular attraction of Natural Deduction is its clean and economical presentation. Here for example are deduction (fragments) proving $A \wedge B$ from A and B, and $\forall x.\ (P(x) \wedge Q(x))$ from $\forall x.\ P(x)$ and $\forall x.\ Q(x)$:

$$\dfrac{A \quad B}{A \wedge B}\,(\wedge I) \qquad\qquad \dfrac{\dfrac{\dfrac{\forall x.\ P(x)}{P(x)}\,(\forall E) \qquad \dfrac{\forall x.\ Q(x)}{Q(x)}\,(\forall E)}{P(x) \wedge Q(x)}\,(\wedge I)}{\forall x.\ (P(x) \wedge Q(x))}\,(\forall I)$$

Of interest is also the Curry-Howard correspondence [M. H. B. Sorensen, 1998, Barendregt, 2000]: these deductions have a natural notion of proof-normalisation. As is well-known, studying proof-normalisation is an important first step for giving deductions semantics, for example arrows in Cartesian-Closed Categories

[1] Perhaps in this paper we should call it 'deduction-normalisation' or 'derivation-normalisation' but, as the phrase is a common one, we shall use the term 'proof-normalsation' to refer to a property of deductions (the property of reducing to normal forms).

[Taylor, 1999], sets-and-functions-between-them, or normal forms in functional programs [M. H. B. Sorensen, 1998].

Now consider classical FOL. We can express it in Natural Deduction style by adding the law of excluded middle or double-negation elimination to intuitionistic FOL:

$$\frac{}{A \vee \neg A}\,(EM) \qquad\qquad \frac{\neg\neg A}{A}\,(DNE)$$

However, these rules compromise proof-normalisation and are in that sense unsatisfactory (computational semantics of deductions in classical logic are a research area in their own right, see [Coquand, 1996] for a solid survey; we do *not* discuss this huge field here).

So how to recover normal forms? Gentzen's elegant solution [Gentzen, 1934] was a multiple-conclusioned logic. We are so familiar with this idea nowadays, we may forget that this is an expensive option in the sense that we have to add explicit context and co-context *everywhere*. For example, the deduction-fragments above become:

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}\,(\wedge I) \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash \forall x.\ P(x), \Delta}{\Gamma \vdash P(x), \Delta}\,(\forall E) \qquad \dfrac{\Gamma \vdash \forall x.\ Q(x), \Delta}{\Gamma \vdash Q(x), \Delta}\,(\forall E)}{\Gamma \vdash P(x) \wedge Q(x), \Delta}\,(\wedge I)}{\Gamma \vdash \forall x.\ (P(x) \wedge Q(x)), \Delta}\,(\forall I)$$

(In practice we might prefer left- and right-introduction rules, but that is not important here.) So to do classical FOL we have to pay at every stage of the deduction by threading Γ and Δ through every part of the deduction, even the purely intuitionistic parts inherited from the intuitionistic core of the logic. (This is analagous to emulating global state in a purely functional programming language by threading it through function calls [Peyton-Jones, 2001].)

Now for our proposal. We extend intuitionistic FOL with **classical restart**, which is the following Natural Deduction rule:

$$(Restart) \qquad \frac{A}{B}$$

This is not a misprint. From A we may proceed to B.

The *side-condition*, of course there is one, is that below every occurrence of restart from A to B, there is (at least) one occurrence of A. For example $\dfrac{\top}{B}$ is not a valid deduction, but the following deduction is valid:

$$\dfrac{\dfrac{\dfrac{\dfrac{[A]}{B}\,(Restart)^{*}}{A \to B}\,(\to I) \qquad [(A \to B) \to A]}{A^{\dagger}}\,(\to E)}{((A \to B) \to A) \to A}\,(\to I)$$

The restart at * is justified at †. The conclusion of this deduction is *Peirce's Law*, a famous classical tautology, and we have just proved one half of the next theorem.

Natural deduction has a notion of state given by the undischarged assumptions. Although Restart insists we return to A, on our return we may find ourselves in a more clement state with respect to the undischarged assumptions. This turns out to give precisely the extra deductive power of classical logic:

THEOREM 1. *Propositional Intuitionistic Logic plus* (*Restart*) *has the same entailment relation as Classical Logic.*

The proof is elementary; one half follows from the derivation of Peirce's Law above. We give the other half in the next section (see also Theorem 2 on page 24).

So *if* we impose global well-formedness conditions on Natural Deduction derivations, *then* we have a way of strengthening a logic 'cheaply'. There is a bit more: it is 'neat' too and proof-normalisation is inherited in a natural way, and we shall explore that briefly in the next section.

In conclusion: There are (at least) two techniques to imposing side-conditions on the form of a deduction. One is to build them in as sequents at every stage, another is to impose them globally on a (natural) deduction. Sometimes these side-conditions are easy to express in both techniques. For example:

- *Natural Deduction:* In $\forall$-introduction $\dfrac{P}{\forall x.\ P}$, x may not occur in assumptions on which P depends.
- *Sequents:* In $\forall$-left-introduction $\dfrac{\Gamma \ \vdash \ P}{\Gamma \ \vdash \ \forall x.\ P}$, x may not occur in Γ.
- *Natural Deduction:* In $\Rightarrow$-introduction $\dfrac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \Rightarrow B}$, A must occur precisely once (to obtain a 'linearity' condition on logical implication).
- *Sequent Calculus:* In $\Rightarrow$-right-introduction $\dfrac{\Gamma, A \ \vdash \ B}{\Gamma \ \vdash \ A \Rightarrow B}$, Γ is a multiset and comma denotes multiset union.

An interesting historical example where Natural Deduction seemed to do better than sequent systems was Prawitz's natural deduction system for the modal logic S5 [Prawitz, 1965] for which the obvious sequent system does not have cut-elimination (an issue addressed later by [Braüner, 2000]). Conversely Natural Deduction could not neatly (in the sense we define) capture classical logic, until now:

- *Sequent Calculus:* Sequents are multiple-conclusion $\Gamma \ \vdash \ \Delta$ (for classical logic).
- *Natural Deduction:* We propose: Restart.[2]

[2] Methods of obtaining a calculus for classical logic that satisfies a normal form theorem are known. For example in [Stalmarck, 1991, Prawitz, 1965] the rule

$$\dfrac{\begin{array}{c}\neg A \\ \vdots \\ \bot\end{array}}{A}\ (PIP)$$

is used. This rule may be considered as an elimination rule for $\bot$, it is also known as the **principle of indirect proof** (PIP). We confine discussion of this rule to a footnote because it is not purely structural, it cannot be formulated unless at least one of $\bot$ or $\neg$ is in the language. So for example we cannot use it to formulate the implication-only fragment of classical propositional logic which should have $((A \to B) \to A) \to A$ as a theorem independently of whether we have negation or $\bot$ in the language as well. We are interested here in the properties of Restart as a purely structural rule that makes the difference between intuitionistic, classical and intermediate logics.

A little history: A rule corresponding to Classical Restart was invented by Dov Gabbay [Gabbay and Reyle, 1984, Gabbay, 1998, Gabbay and Olivetti, 2000b] in the framework of **goal directed deductions**; these are for proof-search and the rules are read bottom up (we discuss this further below). As a bottom-up goal-directed rule, restart is: at a goal B we can 'restart' at any previous goal A we have considered. The second author then re-presented the same rule top-down in Prawitz natural deduction [Gabbay, 2004].

2 Classical Restart

We begin by studying in detail Intuitionistic Logic (IL) augmented with our restart natural deduction rule. We show it has the same derivability relation as Classical Logic (CL); We prove that the standard proof-normalisation procedure for Natural Deduction is preserved; We then consider consider restart in other deduction formats than natural deduction.

2.1 Proof that IL+(Restart)=CL

For simplicity we consider just the propositional case. Fix a set of **atomic formulae** $p, q, r, \ldots$. **Formulae** $A, B \ldots$ are defined by the grammar[3]

$$A ::= p \mid A \to A$$

The Natural Deduction rules of IL are standard:

$$\dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \to B}\,(\to I) \qquad \dfrac{A \quad A \to B}{B}\,(\to E) \qquad \dfrac{\bot}{C}\,(\bot E)$$

Other connectives such as $\wedge$ and $\vee$ are possible.

A **deduction** of IL is formed from $(\to I)$, $(\to E)$, $(\bot E)$. Call a **proto-deduction** a deduction formed as just described, but also with $(Restart)$. A proto-deduction is **valid** when it satisfies the condition that under every restart $\dfrac{A}{B}$ there is a later instance of A. We represent this in the following diagram:

$$\begin{matrix}\dfrac{A}{B}\,(Restart)\\ \vdots \\ A\end{matrix}$$

Let **Classical Logic (CL)** be IL augmented with Peirce's Law $((A \to B) \to A) \to A$ for all A and B. We have already seen how to use $(Restart)$ to prove Peirce's Law, now we consider the converse.

Any application of $(Restart)$ as above may be replaced by

$$\dfrac{\dfrac{\begin{matrix}\dfrac{A \qquad [A \to B]^1}{B}\,(\to E)\\ \vdots \\ A\end{matrix}}{(A \to B) \to A}\,(\to I)^1 \qquad \dfrac{}{((A \to B) \to A) \to A}\,\text{Assumed}}{A}\,(\to E).$$

[3] Our results easily generalise to $\wedge$ and $\vee$.

This proves Theorem 1, and in view of this we may call (*Restart*) **Classical Restart**.

2.2 Proof-normalisation

Unlike adding Peirce's Law, adding (*Restart*) to IL does not compromise proof-normalisation. The essential cases are as before, for example:

$$\dfrac{\dfrac{\begin{matrix}[A]^1\\ \vdots \\ B\end{matrix}}{A \to B}\,(\to I)^1 \qquad \begin{matrix}\vdots\, \Pi\\ A\end{matrix}}{B}\,(\to E) \qquad \Longrightarrow \qquad \begin{matrix}\vdots\, \Pi \\ A \\ \vdots \\ B\end{matrix}$$

The only possible problem is when $A \to B$ above justifies one or more instances of (*Restart*) in the deduction above; then restructure them as follows:

$$\dfrac{A \to B}{C}\,(Restart) \qquad \Longrightarrow \qquad \dfrac{\dfrac{A \to B \qquad \begin{matrix}\vdots\, \Pi \\ A\end{matrix}}{B}\,(\to E)}{C}\,(Restart)$$

Now each of the problematic instances of (*Restart*) is justified one line further down, by B rather than $A \to B$, and we can proceed with elimination of the essential case. Note how Π is **teleported** deep inside the deduction.

More generally, in the presence of conjunction and disjunction, we can always restructure a deduction so that no formula both completes a restart rule and is the major premise of an elimination rule. Schematically, we can replace

$$\begin{matrix}\dfrac{A}{B}\,(Restart) \\ \vdots \\ \dfrac{A}{C}\,(?E)\end{matrix}$$

with this

$$\begin{matrix}\dfrac{\dfrac{A}{C}\,(?E)}{B}\,(Restart) \\ \vdots \\ \dfrac{A}{C}\,(?E)\end{matrix}$$

and now the premise of the restart is C and its side condition is met by a formula one step closer to the conclusion (so we may conclude that ultimately, we can restructure the deduction so that no formula both is the major premise of an elimination rule and validates a previous application of restart).

Notice also that we can always restructure a deduction so that the conclusion of an application of $\vee E$ (or $\exists E$ if we extend the results to the quantified case) is also not the major premise of any elimination rule.[4]

We can now complete a normalisation argument using familiar methods, we sketch the argument here. Say a *segment* is a sequence of occurrences of a formulae $A_1 \ldots A_n$,

[4] So for example in the case of the existential quantifier (A^x_t is the formula obtained by simultaneously

where each A_i is an instance of the same formula A, and A_i is a minor premise of a rule the conclusion of which is A_{i+1}. Say a segment is *maximal* when A_1 is the conclusion of an introduction rule, and A_n is the major premise of an elimination rule. Notice that since we can always restructure a deduction so that no major premise of an elimination rule is also the conclusion of $\vee E$ (or $\exists E$) then we can assume that any maximal segment is a sequence of only two formulae. But then since we may always restructure a deduction so that no major premise of an elimination rule is necessary to validate a previous application of restart, we may remove the maximal segment altogether (as shown by the essential cases above).

Additionally, we may neaten up a deduction by thinking of $(Restart)$ as an introduction rule and simplify $(Restart)$ followed by an elimination rule as we would any essential case:

$$\dfrac{\dfrac{A}{B \to C}\,(Restart) \qquad\qquad \dfrac{\Pi}{B}}{C}\,(\to E) \quad \Longrightarrow \quad \dfrac{A}{C}\,(Restart).$$

We do not simplify restart in the minor premise ($\dfrac{\Pi}{B}$ above); this destroys confluence of the reductions, we discuss this later.

Disjunction. Our syntax in (2.1) does not have disjunction $\vee$ but later in this paper it will be convenient to suppose we do. The natural reduction is as follows:

$$\dfrac{\dfrac{A}{B \vee C}\,(Restart) \qquad \dfrac{[B]}{D} \qquad \dfrac{[C]}{D}}{D}\,(\vee E) \quad \Longrightarrow \quad \dfrac{A}{D}\,(Restart).$$

Again, a similar simplification is possible if a restart terminates either of the two minor premises to D, but adding it would destroy confluence.

2.3 From restart to multiple conclusions

Suppose we have a deduction $\begin{matrix}\Gamma\\ \vdots\\ A\end{matrix}$ where the side conditions of some restarts, say from members of Δ, above A have not (yet) been met. We can write this as a sequent $\Gamma \vdash A; \Delta$ (the predicate A is 'active'; a minor tweak of standard sequents, see below). We apply the restart rule from A to B to obtain this deduction tree

$$\dfrac{\begin{matrix}\Gamma\\ \vdots\\ A\end{matrix}}{B}\,(Restart)$$

---

replacing all free occurrences of the variable x in A by the term t):

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \exists x A\end{matrix} \qquad \begin{matrix}[A^x_a]\\ \vdots\\ B\end{matrix}}{B}\,(\exists E)}{C}\,(?E) \quad \Longrightarrow \quad \dfrac{\begin{matrix}\vdots\\ \exists x A\end{matrix} \qquad \dfrac{\begin{matrix}[A^x_a]\\ \vdots\\ B\end{matrix}}{C}\,(?E)}{C}\,(\vee E)$$

In such a reduction it may be necessary to replace a (in the deduction of B from A^x_a) by some other constant if a occurs in C. This is because of the side condition on $(\exists E)$, see section 8.1.

Now we describe this new deduction tree as $\Gamma \ \vdash \ B; \Delta \cup \{A\}$ or just $\Gamma \ \vdash \ B; \Delta, A$. We can construct a new sequent system (with the semicolon as an additional structural entity) with rules

$$\frac{\Gamma \ \vdash \ A; \Delta}{\Gamma \ \vdash \ B; \Delta, A}\,(Restart1) \qquad \frac{\Gamma \ \vdash \ A; A, \Delta}{\Gamma \ \vdash \ A; \Delta}\,(Restart2)$$

The second of these two rules captures the side condition 'provided A is deduced again later on'. If we add rules for the other connectives we obtain a sequent system for classical logic. For example:

$$\frac{\Gamma, A \ \vdash \ B; \Delta}{\Gamma \ \vdash \ A \to B; \Delta}\,(\to R) \qquad \frac{\Gamma \ \vdash \ A; \Delta \quad \Gamma, B \ \vdash \ C; \Delta}{\Gamma, A \to B \ \vdash \ C; \Delta}\,(\to L)$$

Here is a (well-known) deduction of Peirce's law:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{A \ \vdash \ A}{A \ \vdash \ B; A}\,(Restart1)}{\vdash \ (A \to B); A}\,(\to R) \qquad A \ \vdash \ A; A}{(A \to B) \to A \ \vdash \ A; A}\,(\to L)}{(A \to B) \to A \ \vdash \ A;}\,(Restart2)}{\vdash \ ((A \to B) \to A) \to A;}\,(\to R)$$

In the usual multiple conclusion sequent calculus the conclusion is not split as above into an 'active formula' and a set of 'outstanding formulae'. This extra structure imposes no restrictions since we can easily change the active formula:

$$\cfrac{\cfrac{\cfrac{\Gamma \ \vdash \ A; \Delta}{\Gamma \ \vdash \ B; \Delta, A}\,(Restart1)}{\Gamma \ \vdash \ A; \Delta, A, B}\,(Restart1)}{\Gamma \ \vdash \ A; \Delta, B}\,(Restart2).$$

Thus we may interpret the sequent $\Gamma \ \vdash \ A; \Delta$ semantically just as we would an ordinary multiple conclusion logic:

> If the Γ are true and the Δ are false then A is true.

or even

> If the Γ are true then so is at least one of A, Δ.

Similarly we may understand a deduction tree $\begin{array}{c}\Gamma \\ \vdots \\ A\end{array}$ as a deduction of A from the assumptions of Γ and further assumptions of the negations of the premises of restarts that do not have their side conditions met.[5]

[5] Perhaps this subsection is too humbly titled. '*No*', we can say: 'a multiple conclusion sequent $\Gamma \ \vdash \ \Delta$ expresses the existence of a/any deduction from assumptions Γ with a conclusion $A \in \Delta$ and incomplete applications of restart to formulae $B \in \Delta - \{A\}$'. This is philosophically more satisfying than an interpretation of multiple conclusions which assumes the logic has an explicit disjunction $\vee$.

The reader may recognise that the observations of this subsection and the last correspond, via the Curry-Howard correspondence, to the $\lambda\mu$-calculus [Parigot, 1992]. In fact (using the terminology we have developed) the $\lambda\mu$-calculus only allows restarts from any A to $\bot$. Also our proof-normalisation procedure is not restricted at all to following that specified by $\lambda\mu$; a restart from A to B need not necessarily be tied to a particular later instance of A, and we can choose arbitrarily from which to teleport; so the system here is a little more general.

Of course, all this is not limited to first-order logic. Indeed restart as described is valid in the presence of any abstract machine with state, and perhaps investigating 'stateful automata with restart' belongs to interesting future work. We return to this in the Conclusion.

3 Restart in frameworks other than Natural Deduction

3.1 Restart in linear natural deduction

We have formulated (*Restart*) for a Prawitz style natural deduction system. Its tree structure makes dependency between two formulae easy to track, being a matter of whether there is a path in the tree between them. (*Restart*) as discussed so far exploits this. For example in this deduction

$$\begin{array}{c} \dfrac{A}{B}\,(Restart) \\ \vdots \\ A \end{array}$$

formulae occurring between B and A may be said to **depend** on that particular application of restart. We do not have a full deduction until this dependency is discharged (until A occurs again).

In a linear natural deduction system, dependency is harder to track. When we extend with restart we must extend whatever mechanism we have for handling dependencies to take the new rule into account.

Consider adding restart to a Lemmon style natural deduction system. Each formula in the deduction has three sets of labels (this is standard): a line number; the line numbers of the premises if it is the conclusion of a rule; and the line numbers of the assumptions it depends on. Here is a deduction of $\neg\neg(A \vee \neg A)$:

Dependency	Line	*Formula*	Wherefrom
1	(1)	$\neg(A \vee \neg A)$	
2	(2)	A	
2	(3)	$A \vee \neg A$	$2(\vee I)$
$1, 2$	(4)	$\bot$	$1, 3(\neg E)$
1	(5)	$\neg A$	$2, 4(\neg I)$
1	(6)	$A \vee \neg A$	$5(\vee I)$
1	(7)	$\bot$	$6, 1(\neg E)$
	(8)	$\neg\neg(A \vee \neg A)$	$1, 7(\neg I)$

The 'deduction rule' $(\neg I)$ is the following schema:

$$\begin{array}{llll}
 & \vdots & & \\
n & (n) & A & \\
 & \vdots & & \\
\{n_1 \dots n_r, n\} & (m) & \bot & \dots \\
 & \vdots & & \\
\{n_1 \dots n_r\} & (k) & \neg A & n, m(\neg I)
\end{array}$$

We say 'schema' because the premises (lines n and m) can occur at some distance above the conclusion (line k). The assumption A is discharged by removing its line number from the set of dependency labels at line k, and this is how dependencies are tracked.

We say that B at line m depends on A at line n when the sets of dependency labels are $\{n\}$ at line n and $\{m_1 \dots m_k, n\}$ at line m (i.e. A occurs as an assumption and B has A's line number in its dependency column). So now we have $\Gamma \vdash A$ when there is a deduction in which there is a line containing A that depends only on elements of Γ.

To add restart, extend labels with a dependency label $R(n)$ where R is for 'restart' and n is the line number of the premise. We then add two rules, one for the restart:

$$\begin{array}{llll}
 & \vdots & & \\
\{n_1 \dots n_r\} & (n) & A & \dots \\
 & \vdots & & \\
\{n_1 \dots n_r, R(n)\} & (k) & B & n(\mathit{Restart}1)
\end{array}$$

...and one for the side condition that the premise of the restart reoccur:

$$\begin{array}{lllll}
 & \vdots & & & \\
\{m_1 \dots m_r, R(n)\} & (m) & A & \dots & \text{Provided the formula} \\
 & \vdots & & & \text{at line } n \text{ is } A \\
\{m_1 \dots m_r\} & (k) & A & m(\mathit{Restart}2) &
\end{array}$$

So for example we can now deduce Peirce's law:

Dependency	Line	*Formula*	Wherefrom
1	(1)	A	
2	(2)	$(A \to B) \to A$	
$1, R(1)$	(3)	B	$1(\mathit{Restart}1)$
$R(1)$	(4)	$A \to B$	$1, 3(\to I)$
$2, R(1)$	(5)	A	$2, 4(\to E)$
2	(6)	A	$5(\mathit{Restart}2)$
	(7)	$((A \to B) \to A) \to A$	$2, 6(\to I)$

The restart rule looks less magical now, since these rules are so like a rule for indirect proof; $R(n)$ resembles a label for $\neg A$, $(\mathit{Restart}1)$ resembles a rule for deducing B from A and $\neg A$, and $(\mathit{Restart}2)$ resembles a rule for deducing that if A follows from $\neg A$ then we can conclude A without a dependency on $\neg A$. It is not hard to translate between deductions using the two rules above, and deductions using Indirect Proof or Peirce's Law.

In Prawitz form, restart is interesting because we get a normalising deduction system for classical logic from an intuitionistic one with no structural changes (labels, multiple conclusions). In linear natural deduction systems, where dependency is tracked by labelling anyway, this advantage is, if not exactly lost, certainly less apparent.

3.2 Goal-directed deduction with restart

A striking formulation of the restart rule is the original one from [Gabbay, 1990]. This is in terms of Goal-Directed reasoning which we can understand here as a Fitch style natural deduction system such that all subdeductions have a goal which must be met. That is, a subdeduction is not open-ended, but can be closed only when the goal is reached. For example, here is a Goal-Directed deduction that $[A \to (B \to C)] \to [(A \to B) \to (A \to C)]$:

1	$\quad A \to (B \to C) \quad ?(A \to B) \to (A \to C)$	
2	$\quad\quad A \to B \quad ?(A \to C)$	
3	$\quad\quad\quad A \quad ?C$	
4	$\quad\quad\quad B \to C$	$(\to E)$, 1, 3
5	$\quad\quad\quad B$	$(\to E)$, 2, 3
6	$\quad\quad\quad C$	$(\to E)$, 4, 5
7	$\quad\quad A \to C$	$(\to I)$, 3, 6
8	$\quad (A \to B) \to (A \to C)$	$(\to I)$, 2, 7
9	$[A \to (B \to C)] \to [(A \to B) \to (A \to C)]$	$(\to I)$, 1, 8

Notice that an individual subdeduction is completed only when its goal is reached, at which point we can terminate the subdeduction and introduce an implication. The restart rule then becomes an additional rule on when a subdeduction is completed.

> A subdeduction with a hypothesis A and goal $?B$ may be terminated (introducing an implication $A \to B$) by the deduction of not only its goal, but by the deduction of any previous goal.

For example here is a Goal-Directed deduction of Peirce's Law:

1	$\quad (A \to B) \to A) \quad ?A$	
2	$\quad\quad A \to B \quad ?A$	
3	$\quad\quad\quad A \quad ?B$	
4	$\quad\quad\quad A$	Recalling , 3
5	$\quad\quad A \to B$	(*Restart*) to $?A$, 4, 3, 2
6	$\quad\quad A$	$(\to E)$, 2, 5
7	$\quad [(A \to B) \to A)] \to A$	$(\to E)$, 1, 6

We get line 5 by restarting the goal $?B$ to a goal $?A$ (and declaring ourselves satisfied since that is what we have at line 4); instead of terminating the subdeduction with B, we terminate it with the previous goal A.

It is not hard to see that this restart is sound: we observe that we can obtain a valid deduction in a system with an indirect proof rule by replacing any goal $?A$ with an additional assumption $\neg A$ and then using the principle of indirect proof. Any previous 'goal' is really an assumption that the goal is false. So note in the example above, reading $?A$ as $\neg A$, that the introduction of $A \to B$ is valid, as we can deduce B from A and the previous assumption that $\neg A$. Because $?A$ was a *previous* goal, when A is deduced again later in the deduction (line 6), we may consider the assumption of $\neg A$ (i.e. $?A$) as being discharged by the Principle of Indirect Proof:

$$\begin{array}{l|l}
\vdots & \vdots \\
n & \quad \begin{array}{|l} \neg A \\ \hline \vdots \\ A \end{array} \\
\vdots & \\
m & \\
m+1 & A \qquad\qquad (PIP),\ n,\ m
\end{array}$$

The system of Goal-Directed logic [Gabbay, 1990, Gabbay and Olivetti, 2000a] is mainly for implication- and quantifier-fragments of various logics.

4 n-depth restart

Say a Kripke model has **height bounded by** n when for all series of worlds $w_1, \ldots, w_{n+1}$, if $w_i \mathrel{R} w_{i+1}$ for $1 \leq i \leq n$ then an i exists with $w_i = w_j$ ("the model has depth at most n"). Here R denotes the Kripke accessibility relation.

A characteristic axiom scheme for models of depth 1 is $P^1(A, B) \equiv ((A \to B) \to A) \to A$. A characteristic axiom scheme for models of depth $i+1$ is taken from [Gabbay, 1981] and given as follows:

$$P^{i+1}(A_1, \ldots, A_{i+1}) \quad \stackrel{\text{def}}{=} \quad P^1(A_{i+1}, P^i(A_1, \ldots, A_i)).$$

For example $P^2(A, B, C) = ((A \to \Big(((B \to C) \to B) \to B\Big)) \to A) \to A$.

4.1 A natural deduction restart for bounded height

An appealing *intuition* (we do not make it formal) behind n-depth restart is that each A_i is proved at world w_i, where w_n is the final world. Whence the condition on discharge after handles, which corresponds to a restriction not to descend below w_i until we have justified the restart at that world. The restart at the final world is an ordinary 1-depth restart, jumping from A_n to an arbitrary B is reasonable because there is no future world and so any B holds in any future world.

$$\dfrac{\begin{array}{ccc} \vdots\,\Pi_1 & & \vdots\,\Pi_n \\ A_1 & \ldots & A_n \end{array}}{\begin{array}{c} B \\ \vdots\,\Pi'_n \\ A_n \\ \vdots\,\Pi'_2 \\ A_2 \\ \vdots\,\Pi'_1 \\ A_1 \end{array}}$$

n-depth Restart

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[B]^1}{B \vee (B \to (A \vee \neg A))^b} \qquad \dfrac{[A]^2}{A \vee \neg A^a}}{\bot}}{\neg A^2}}{A \vee \neg A^a}}{B \to (A \vee \neg A)^1}}{B \vee (B \to (A \vee \neg A))^b}$$

Proof using 2-depth restart

$$\dfrac{\dfrac{\dfrac{\dfrac{[A]^1}{A \vee \neg A^a} \qquad \dfrac{[A]^2}{A \vee \neg A^b}}{\bot}}{\neg A^{1,2}}}{A \vee \neg A^{a,b}}$$

Invalid (two 1-depth restarts)

There is an extra side-condition on n-depth restart. Say that the restart is *handled* by each re-deduction of $A_n \ldots A_n$ (so each application of an n-depth restart rule gets handled n times). The side condition is this

> In n-depth restart no assumption or application of restart in the deduction Π_i of A_i may be discharged or handled under B before A_{i+1} has been handled

We can prove $B \vee (B \to (A \vee \neg A))$ using 2-depth restart, as illustrated. The 'deduction' on the right of $A \vee \neg A$ using the 2-depth restart rule is *invalid* because it discharges A at ^{2} before $A \vee \neg A$ at ^{b}.

Proof-normalisation is fairly evident and similar to the 1-depth case (ordinary restart); an elimination rule following A_i in the lower part of the deduction is teleported to the A_i in the upper part of the deduction. Normal forms are unique. We omit the proof of this, which is not hard.

5 Soundness of n-depth restart

We shall argue that any deduction containing applications of n-depth restart may be replaced by a deduction, of the same conclusion, using the n-depth axiom instead. We shall do this by showing how to replace any application of n-depth restart with a use of the n-depth restart rule.

The construction, although simple to the mind, is unfortunately not so simple to the eye. Take any application of n-depth restart:

$$
\begin{array}{c}
\dfrac{\begin{array}{c}\vdots\,\Pi_1\\ A_1\end{array}\quad \ldots \quad \begin{array}{c}\vdots\,\Pi_n\\ A_n\end{array}}{B}\\
\vdots\,\Pi'_n\\
A_n\\
\vdots\,\Pi'_2\\
A_2\\
\vdots\,\Pi'_1\\
A_1
\end{array}
$$

and we can begin to rewrite it as follows:

$$
\begin{array}{c}
\dfrac{\dfrac{\begin{array}{c}\vdots\,\Pi_1\\ A_1\end{array}\quad \ldots \quad \begin{array}{c}\vdots\,\Pi_{n-1}\\ A_{n-1}\end{array}}{[(A_n \to B) \to A_n] \to A_n}\qquad \dfrac{\begin{array}{c}\dfrac{\begin{array}{c}\vdots\,\Pi_n\\ A_n\end{array}\quad [A_n \to B]^{i_n}}{B}\\ \vdots\,\Pi'_n\\ A_n\end{array}}{(A_n \to B) \to A_n}\,(\to I)^{i_n}}{A_n}\,(\to E)\\
\vdots\,\Pi'_2 \ldots \Pi'_{n-1}\\
A_2\\
\vdots\,\Pi'_1\\
A_1
\end{array}
$$

We can routinely replace all applications of n-depth restart with this 'rule':

$$\begin{array}{c}\dfrac{\dfrac{\begin{matrix}\vdots & & \vdots\\ A_1 & \dots & A_{n-1}\end{matrix}}{P^1(A_n,B)} \qquad \begin{matrix}\vdots\\ (A_n \to B) \to A_n\end{matrix}}{A_n}\;(\to E)\\ \vdots\\ A_1\end{array}$$

If there is no A_{n-1} (i.e. if $n = 1$) then A_{n-1} and Π_{n-1} are empty and we have replaced the n-depth restart with an appeal to Peirce's Law.

$$\overline{P^1(A_1,B)}$$

However, if $n > 1$ then we must make further replacements. To save space, let P^m be short for $P^m(A_{(n-(m-1))} \dots A_n, B)$

We may replace any application of

$$\begin{array}{c}\dfrac{\dfrac{\begin{matrix}\vdots\,\Pi_1 & & \vdots\,\Pi_{n-m}\\ A_1 & \dots & A_{n-m}\end{matrix}}{P^m} \qquad \begin{matrix}\vdots\,\Pi_{n-(m-1)},\Pi'_{n-(m-1)}\dots\Pi_n,\Pi'_n\\ (A_{(n-(m-1)} \to P^{m-1}) \to A_{(n-(m-1)}\end{matrix}}{A_{n-(m-1)}}\;(\to E)\\ \vdots\,\Pi'_1\dots\Pi'_{n-m}\\ A_1\end{array}$$

so that we use instead an application of

$$\begin{array}{c}\dfrac{\dfrac{\begin{matrix}\vdots\,\Pi_1 & & \vdots\,\Pi_{n-(m+1)}\\ A_1 & \dots & A_{n-(m+1)}\end{matrix}}{P^{m+1}} \qquad \begin{matrix}\vdots\,\Pi_{n-m},\Pi'_{n-m}\dots\Pi_n,\Pi'_n\\ (A_{n-m} \to P^m) \to A_{n-m}\end{matrix}}{A_{n-m}}\;(\to E)\\ \vdots\,\Pi'_1\dots\Pi'_{n-(m+1)}\\ A_1\end{array}$$

in a similar way:

$$\begin{array}{c}\dfrac{\dfrac{\begin{matrix}\vdots\,\Pi_1 & & \vdots\,\Pi_{n-(m+1)}\\ A_1 & \dots & A_{n-(m+1)}\end{matrix}}{P^{m+1}} \qquad \dfrac{\begin{array}{c}\dfrac{\dfrac{\begin{matrix}\vdots\,\Pi_{n-m} & \\ A_{n-m} & [A_{n-m} \to P^m]^i\end{matrix}}{P^m}\;(\to E) \qquad \begin{matrix}\vdots\,\Pi/\Pi'_{n-(m-1)}\dots\Pi/\Pi'_n\\ (A_{(n-(m-1)} \to P^{m-1}) \to A_{(n-(m-1)}\end{matrix}}{A_{n-(m-1)}}\;(\to E)\\ \vdots\,\Pi'_{n-m}\\ A_{n-m}\end{array}}{[A_{n-m} \to P^m] \to A_{n-m}}\;(\to I)^i}{A_{n-m}}\\ \vdots\,\Pi'_1\dots\Pi'_{n-(m+1)}\\ A_1\end{array}$$

Notice that the side condition ensures that no assumption in $\Pi_1 \dots \Pi_{n-(m+1)}$ is discharged in Π'_{n-m} so we can bring Π'_{n-m} out and across from underneath $\Pi_1 \dots \Pi_{n-(m+1)}$ without affecting the deduction.

Ultimately, after repeated applications of this we will have replaced all applications of n-depth restart by this:

$$
\begin{array}{c}
\dfrac{\dfrac{\begin{array}{c}\vdots\\ A_1\end{array}\quad \ldots \quad \begin{array}{c}\vdots\\ A_{n-n}\end{array}}{P^n} \qquad (A_{n-(n-1)} \to \overset{\vdots}{P^{n-1}}) \to A_{n-(n-1)}}{A_{n-(n-1)}}\ (\to E)\\
\vdots\\
A_1
\end{array}
$$

and since there is no A_0 this is simply

$$
\dfrac{(A_1 \to \overset{\vdots}{P^{n-1}}) \to A_1 \qquad \overline{P^n}}{A_1}\ (\to E)
$$

which is just an appeal to the n-depth Kripke model axiom.

6 Completeness of n-depth restart

Notice that within an application of an $n+1$-depth restart rule we can find an n-depth restart rule (considering only $A_n \ldots A_2$ and ignoring A_1). We have already shown that any instance of Peirce's law P^1 can be derived using 1-depth restart. We can now argue that P^n can be derived using the n-depth restart rule.

The argument is by induction on n. Suppose for the induction hypothesis that the r-depth restart rule can be used to derive the r-depth axiom, where $r = n-1$ and $r \geq 1$.

$$
\begin{array}{c}
\dfrac{A_2 \ldots A_n}{B}\\
\vdots\\
A_n\\
\vdots\\
A_2
\end{array}
$$

where A_2 is $P^{n-1}(A_2 \ldots A_n, B)$. Now we can edit it in the following way, first we add an extra premise:

$$
\begin{array}{c}
\dfrac{\dfrac{A_1}{P^n(A_1 \ldots A_n, B)}\ (\to I) \qquad A_2 \ldots A_n}{B}\\
\vdots\\
A_n\\
\vdots\\
A_2
\end{array}
$$

and now we continue on from A_2 as follows

$$\dfrac{\dfrac{[(A_1 \to A_2) \to A_1]^l \qquad \dfrac{\begin{array}{c}\dfrac{\dfrac{[A_1]^i}{P^n(A_1 \dots A_n, B)}(\to I) \qquad A_2 \dots A_n}{B} \\ \vdots \\ A_n \\ \vdots \\ A_2\end{array}}{A_1 \to A_2}(\to I)^i}{A_1}(\to E)}{((A_1 \to A_2) \to A_1) \to A_1}$$

and since A_2 is $P^{n-1}(A_2 \dots A_{n+1}, B)$ then $((A_1 \to A_2) \to A_1) \to A_1$ is really $P^n(A_1 \dots A_n, B)$ which is the n-depth axiom. Furthermore we have converted the r-depth restart into an instance of $r+1$-depth restart. This completes the induction.

6.1 Goal-directed n-depth restart

n-depth restart can be formulated in Goal Directed reasoning as well. Say that a goal $?A$ is *embedded* in another goal $?B$ when $?B$ is a goal previous to $?A$,[6] and B occurs, not as a goal, somewhere between $?B$ and $?A$.

Schematically, $?A$ is embedded in $?B$ when this happens:

$$\begin{array}{c|l|l|l|l}
\vdots & \vdots & & & \\
n & & X \quad ?B & & \\
\cline{3-3}
 & & \vdots & & \\
\vdots & & \dots & & \\
 & & & \vdots & \\
m & & & B & \\
 & & & \vdots & \\
\vdots & & & \dots & \\
r & & & & Y \quad ?A \\
\cline{5-5}
\vdots & & & & \vdots
\end{array}$$

The n-depth goal directed restart rule is this:

> A subdeduction with a hypothesis A and goal $?B$ may be terminated (introducing an implication $A \to B$) by the deduction of not only its goal, but of a previous goal $?C$ provided that there are goals $?C_1 \dots ?C_{n-1}$ such that $?C$ is embedded in $C_n - 1$ and each C_i is embedded in C_{i-1}.

So for example, here is a deduction using 2-depth restart of $((A \to (\neg\neg B \to B)) \to$

[6] i.e. the subdeduction for which $?A$ is a goal is a subdeduction (of a subdeduction of...) the subdeduction for which $?B$ is a goal.

$A) \to A$:

1	$(A \to (\neg\neg B \to B)) \to A \quad ?A$	
2	$\quad A \quad ?\neg\neg B \to B$	
3	$\quad\quad \neg\neg B \quad ?B$	
4	$\quad\quad\quad B \quad ?\bot$	
5	$\quad\quad\quad B$	Recalling , 4
6	$\quad\quad \neg B$	2-restart, $?B$ is embedded in $?A$
7	$\quad\quad \bot$	$(\to E)$, 2, 6
8	$\quad\quad B$	$(\bot E)$, 7
9	$\quad \neg\neg B \to B$	$(\to I)$, 3, 8
10	$A \to (\neg\neg B \to B)$	$(\to I)$, 2, 9
11	A	$(\to E)$, 1, 10
12	$((A \to (\neg\neg B \to B)) \to A) \to A$	$(\to I)$, 1, 11

The application of restart was to complete the Goal $?\bot$ by completing instead the previous goal $?B$. Notice that above the goal $?B$ is an occurrence of A and above that is a goal $?A$. So $?B$ is embedded in at least one other goal. This legitimises they use of 2-depth restart.

6.2 Soundness of goal-directed n-depth restart

It is a somewhat tedious matter to show how to convert a deduction using the goal-directed n-depth restart rule into a deduction using the n-depth axioms. We shall not describe the construction in detail here, mainly because it is similar to the construction proving soundness of the natural deduction n-depth restart rule.

But a sketch-proof of why the rule is sound may be found by observing that the goal-directed n-depth restart rule meets the conditions of the natural deduction n-depth restart rule.

To apply the goal directed restart rule, we act as if we have achieved a goal $?B$ when in fact we have deduced only A. Effectively we do this:

$$\frac{A}{B}$$

But A was not just any formula, it was a previous goal, which means it will have to be met at a later point for the deduction to be complete:

$$\begin{array}{c} \underline{A} \\ B \\ \vdots \\ A \end{array}$$

But A is embedded in a goal $?C_1$, which means that we have C_1 to complete the goal

$?C_1$ after we have achieved goal $?A$:

$$\cfrac{A}{\begin{array}{c} B \\ \vdots \\ A \\ \vdots \\ C_1 \end{array}}$$

and also we have already deduced C_1 when deduced A:

$$\cfrac{A \quad C_1}{\begin{array}{c} B \\ \vdots \\ A \\ \vdots \\ C_1 \end{array}}$$

but discharge in goal-directed reasoning is done only by completing goals, it follows (since C_1 was deduced in between the goal $?A$ and the prior goal $?C_1$) that no assumption on which C_1 depends is discharged (has its goal met) before A is deduced again (that is, before the goal $?A$ is achieved). The same holds if $?A$ must be embedded in n further goals:

$$\cfrac{A, C_1 \dots C_n}{\begin{array}{c} B \\ \vdots \\ A \\ \vdots \\ C_1 \\ \vdots \\ C_n \end{array}}$$

This is precisely the side condition on the $n+1$-depth natural deduction restart rule.

7 Hash logic

7.1 Natural deduction for hash logic

We can internalise the restart rule and use it as an elimination rule for a connective. Here are the rules for a unary connective #.

$$\frac{A}{\#A}\ (\#I) \qquad \frac{\#B \quad A}{B}\ (\#E) \qquad \frac{\#A_1 \quad \dots \quad \#A_n \qquad \begin{array}{c}[A_1]^i \dots [A_n]^i \\ \vdots \\ B\end{array}}{\#B}\ (\#P)^i$$

Say an application of $(\#E)$ is *completed* when A is deduced again below it. Until A is deduced again the application of $(\#E)$ is incomplete.

The side condition on the rule $(\#E)$ is this:

(i) A must be deduced again after the application of the rule and
(ii) no (occurrence of an) assumption on which $\#B$ depends may be discharged until A is re-deduced and
(iii) no $(\#E)$ rule on which $\#B$ depends is completed until A is re-deduced.

So the rule is this

$$\begin{array}{c}\dfrac{\#B \quad A}{B}\,(\#E)\\ \vdots\\ A\end{array}$$

where nothing on which $\#B$ depends may be discharged until the $(\#E)$ rule is completed and no $(\#E)$ rule that is incomplete at $\#B$ is completed until A re-occurs.

We should be liberal with our interpretation of these rules. We should not regard the rules as requiring that an application of the $(\#E)$ rule is completed by the *first* re-occurrence of its minor premise that is deduced again. The rule demands only that the minor premise is deduced at some later point, it may appear many times before we can regard the rule as being completed.[7]

To see what the Hash rules do, note the following deduction of $A \vee (A \to B)$:[8]

$$\dfrac{\dfrac{\dfrac{\#B \quad \dfrac{[A]^1}{A\vee(A\to B)}\,\vee I}{B}\,(\#E)}{A\to B}\,(\to I)(1)}{A\vee(A\to B)}\,\vee I$$

Furthermore, we can replace instances of $(\#E)$ with instances of this $\#B \to (A \vee (A \to B))$:

$$\begin{array}{c}\dfrac{\begin{array}{c}\vdots\,\Pi_1\\ \#B\end{array}\quad \begin{array}{c}\vdots\,\Pi_2\\ A\end{array}}{B}\,(\#E)\\ \vdots\,\Pi\\ A\end{array}$$

may be replaced by

$$\dfrac{\dfrac{\#B\to(A\vee(A\to B)) \quad \begin{array}{c}\vdots\,\Pi_1\\ \#B\end{array}}{A\vee(A\to B)}\,(\to E) \qquad [A]^i \qquad \begin{array}{c}\dfrac{[A\to B]^i \quad \begin{array}{c}\vdots\,\Pi_2\\ A\end{array}}{B}\,(\to E)\\ \vdots\,\Pi\\ A\end{array}}{A}\,\vee E(i)$$

And the side condition ensures that moving Π out from underneath Π_1 does not invalidate the deduction.

[7] So suppose we encounter something like this in the deduction

$$\begin{array}{c}\dfrac{\dfrac{\#\#A \quad B}{\#A}\,(\#E) \quad C}{A}\,(\#E)\\ \vdots\\ B\end{array}$$

and the first application of $(\#E)$ cannot be completed by the re-occurrence of B (suppose it is the first) unless C has already re-occurred. If the completed the second application of $(\#E)$ then, even though B has been deduced again, both application of $(\#E)$ remain incomplete.

[8] Here we use disjunction to harmonise our results with [Gabbay, 1981], but from the next section it can be seen disjunction is not essential to make the point made here.

We have shown that for every deduction using $(\#E)$ there is a deduction, of the same conclusion, using an appeal to an axiom schema $\#A \rightarrow (A \vee (A \rightarrow B))$ instead. Furthermore, it is not hard to see that we can replace the $(\#I)$ rule throughout a deduction by an appeal to the axiom schema $A \rightarrow \#A$ and $(\rightarrow E)$. Finally, we can remove all occurrences of the $\#P$ rule and replace them with appeals to this axiom $\#(A \rightarrow B) \rightarrow (\#A \rightarrow \#B)$.[9]

It is shown in [Gabbay, 1981] that adding the axioms

$$\begin{array}{l} \#B \rightarrow (A \vee (A \rightarrow B)) \\ A \rightarrow \#A \\ \#(A \rightarrow B) \rightarrow (\#A \rightarrow \#B) \end{array}$$

to intuitionistic logic gives (the new connective) $\#A$ the following truth conditions:

> $\#A$ is true at world w (in a Kripke model) when A is true in all w' s.t. wRw' and $w \neq w'$.

Given this, $\#^n\bot$ means that the Kripke model has a depth of at most n worlds.[10]

I shall not discuss the hash logic much here. We shall note that the additional rules for # preserve normalisation.

7.2 Normalisation for hash logic

The essential case for # is only slightly more complex than for the other connectives. Suppose # is introduced, subjected to a number of applications of $\#P$, and then eliminated. First note that two (and hence any number) of successive applications of $(\#P)$

$$\dfrac{\dfrac{\#A_1 \quad \ldots \quad \#A_n \qquad \begin{matrix}[A_1]^i \ldots [A_n]^i \\ \vdots \\ B\end{matrix}}{\#B}\,(\#P)(i) \qquad B_1 \quad \ldots \quad \#B_m \qquad \begin{matrix}[B]^j \quad [B_1]^j \ldots [B_m]^j \\ \vdots \\ C\end{matrix}}{\#C}\,(\#P)(j)$$

[9] The minor premise of $(\#P)$ gives us that $A_1 \rightarrow \ldots A_n \rightarrow B$ and so we have $\#(A_1 \rightarrow \ldots A_n \rightarrow B)$ using the axiom $A \rightarrow \#A$ and then the # operator may be distributed across all implications using the axiom above.

[10] Actually this is not true. The axioms for # considered in [Gabbay, 1981] are these:

$$\begin{array}{l} \#B \rightarrow (A \vee (A \rightarrow B)) \\ A \rightarrow \#A \\ (A \rightarrow B) \rightarrow (\#A \rightarrow \#B) \\ \#A \rightarrow \neg\neg A \end{array}$$

and the semantics for # is that $\#A$ is true at world w (in a Kripke model) when either

(i) w is not an endpoint and A is true in all w' s.t. wRw' and $w \neq w'$.

(ii) w is an endpoint and A is true in w.

But it is not hard to convert Gabbay's proof into one that the axioms we present are complete for our simpler semantics. We find our simpler semantics much more convenient. For one thing we can express the hash operator of [Gabbay, 1981] using conjunction: $\#A \wedge \neg\neg A$. Furthermore it is useful not to have $\#\bot$ as necessarily false, for we can use it (on our semantics for #) to express the property of being an endpoint (according to our semantics, $\#A$ is true at w iff w has no worlds accessible to it other than itself).

may be reduced to just one:

$$\dfrac{\#A_1 \quad \dots \quad \#A_n \quad \#B_1 \quad \dots \quad \#B_m \qquad \begin{array}{c} \begin{array}{c}[A_1]^i \dots [A_n]^i \\ \vdots \\ B\end{array} \quad [B_1]^i \dots [B_m]^i \\ \vdots \\ C \end{array}}{\#C}\ (\#P)(i)$$

and now an introduction, permutation ((#P) rule) and subsequent elimination of #

$$\begin{array}{c}\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\ \Pi\\ A\end{array}}{\#A}\ (\#I) \quad \#A_1 \quad \dots \quad \#A_n \qquad \begin{array}{c}[A]^i \quad [A_1]^i\dots[A_n]^i\\ \vdots\\ B\end{array}}{\#B}\ (\#P)(i) \qquad C}{B}\ (\#E) \\ \vdots \\ C\end{array}$$

may be reduced to:

$$\begin{array}{c}\dfrac{\dfrac{\#A_1 \quad \dots \quad \#A_n \qquad \begin{array}{c}\begin{array}{c}\vdots\ \Pi\\ A\end{array} \quad [A_1]^i\dots[A_n]^i\\ \vdots\\ B\end{array}}{\#B}\ (\#P)(i) \qquad C}{B}\ (\#E) \\ \vdots \\ C\end{array}$$

and if any of the $\#A_i$ are introduced by (#I) then they may be dispatched similarly.

The normalisation argument then proceeds in almost identical fashion to the argument for restart. We must argue that we can always restructure the deduction so that the major premise of an elimination rule never be required to complete an application of restart. In such a case we must teleport the elimination rule away, so this

$$\dfrac{\begin{array}{c}\vdots\ \Pi' \\ \dfrac{\begin{array}{c}\vdots\\ \#B\end{array} \quad \begin{array}{c}\vdots\ \Pi\\ A\end{array}}{B}\ (\#E) \\ \vdots\ \Pi' \\ A\end{array}}{C}\ (?E)$$

gets replaced by this:

$$\dfrac{\begin{array}{c}\dfrac{\#B \quad \begin{array}{c}\vdots\ \Pi\\ A\\ \vdots\ (?E)\\ C\end{array}}{B}\ (\#E) \\ \vdots\ \Pi' \\ A\end{array}}{C}\ (?E)$$

and we must treat any incomplete restarts in the minor premises of Π as being completed only *after* Π' (even if the appropriate formulae happen, by chance, to re-occur in Π').[11]

7.3 Goal directed Hash logic

The Goal-directed reasoning of [Gabbay and Reyle, 1984, Gabbay, 1998, Gabbay and Olivetti, 2000b] does not have rules for disjunction. However, we can formulate the axioms for # in terms of implication. It is not hard to show that we obtain equivalent logics when using the axiom $\#B \to [((A \to B) \to A) \to A]$ instead of $\#B \to (A \vee (A \to B))$. For example, here is an argument that there is a deduction of $(A \vee (A \to B))$ if we add $((A \to B) \to A) \to A$ as an axiom, for brevity let X be shorthand for $[A \vee (A \to B)]$.

1	$\vdash ((X \to B) \to X) \to X$	by the alternative axiom
2	$A \vdash X$	by the disjunction axioms
3	$A, X \to B \vdash B$	from 2, by Cut and Modus Ponens
4	$X \to B \vdash A \to B$	from 3 using the Deduction theorem
5	$X \to B \vdash X$	from 4 using the disjunction axioms and Cut
6	$\vdash (X \to B) \to X$	from 5 by the Deduction theorem
7	$\vdash X$	Cut from 1 and 6

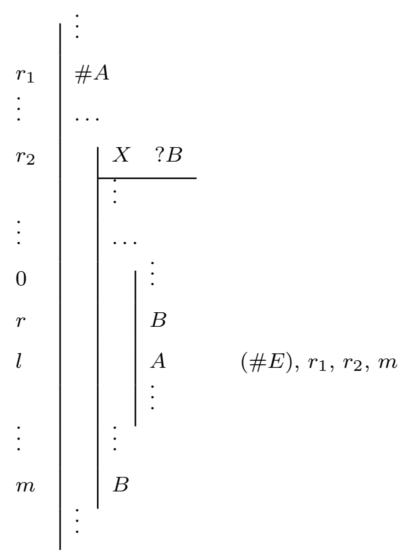
Here are the rules for the # operator in goal directed logic. Firstly is the obvious introduction rule:

> We may infer $\#A$ if we have previously inferred A.

and now the less obvious elimination rule:

> We may infer A from an instance of B if we have a goal $?B$ prior to that instance of B and we have deduced $\#A$ prior to that goal $?B$.

Schematically this rule looks like this



Line l is inferred from r by the rule for #, prior to B is a goal $?B$ and prior that is $\#A$.

[11] Note that for a hash logic normalisation theorem we must work with a slightly different notion of a segment: a segment is either (i) a sequence in a deduction $A_1 \ldots A_n$ where each A_i is an occurrence of A or (ii) a sequence $\#B_1 \ldots \#B_n$ where each $\#B_{i+1}$ is deduced from B_i by the $\#P$ rule.

To see the rule is sound observe that we can convert it into a deduction that uses an instance of the axiom $\#A \to [((B \to A) \to B) \to B]$.

$$
\begin{array}{l|llll}
 & \vdots & & & \\
r_1 & \#A & & & \\
 & \#A \to [((B \to A) \to B) \to B] & & & \text{Axiom} \\
r_1' & ((B \to A) \to B) \to B & & & \\
\vdots & \dots & & & \\
r_2 & \quad \begin{array}{|l} X \quad ?B \\ \hline \end{array} & & & \\
r' & \quad | \quad \begin{array}{|l} B \to A \quad ?B \\ \hline \end{array} & & & \\
 & \quad | \quad | \ \vdots & & & \\
\vdots & \quad | \quad | \ \dots & & & \\
0 & \quad | \quad | \quad | \ \vdots & & & \\
r & \quad | \quad | \quad | \ B & & & \\
l & \quad | \quad | \quad | \ A & & & \text{Modus Ponens, } r',\, r \\
 & \quad | \quad | \quad | \ \vdots & & & \\
\vdots & \quad | \quad | \ \vdots & & & \\
m & \quad | \quad | \ B & & & \\
m' & \quad | \ (B \to A) \to B & & & \\
 & \quad | \ B & & & \text{Modus Ponens, } r_1',\, m' \\
 & \vdots & & &
\end{array}
$$

Looking within this construction we can see an alternative proof that the simple goal-directed restart rule is sound that proceeds by replacing applications of restart to an appeal to Peirce's law rather than the principle of indirect proof.

To ensure that the goal-directed hash calculus is complete we need only add the following rule (the equivalent of the $\#P$ rule) which we represent here schematically:

$$
\begin{array}{l|lll}
 & \vdots & & \\
r_1 & \#A_1 & & \\
\vdots & \vdots & & \\
r_n & \#A_n & & \\
r & \quad \begin{array}{|l} A_1 \dots A_n \quad ?B \\ \hline \end{array} & & \\
\vdots & \quad | \ \vdots & & \\
r' & \quad | \ B & & \\
 & \#B & & (\#P), r_1 \dots r_n, r, r'
\end{array}
$$

To show completeness we must use the new rules to derive the three axioms $\#A \to [((B \to A) \to B) \to B$, $A \to \#A$ and $\#(A \to B) \to \#A \to \#B$, this is left for the reader.

8 Restart in first order logic

8.1 Quantifier rules

We have concentrated on restart for propositional logic, but there is no reason not to do exactly the same with predicate logic.

In first order logic, the restart rule remains unchanged (to extend intuitionistic to classical logic), but the side conditions on $\forall I$ and $\exists E$ rules must be modified a little.[12]

$$\frac{A}{\forall x A}\ (\forall I)$$

Provided x does not occur free in any assumptions on which A depends *nor* (free) in any formula that is required to handle any restarts that are incomplete at A.

$$\frac{\exists x A \qquad \begin{array}{c}[A^x_c]^i\\ \vdots\\ C\end{array}}{C}\ (\exists E)^i$$

Where c is a constant which does not occur in any assumptions on which C depends, except A^x_c, nor in $\exists x A$, nor in any formula that is required to handle any restarts that are incomplete at C.

A formula B is required to handle a restart at (a node containing an instance of) A in a deduction when there is a (prior) application of restart that requires B to be deduced again but (an appropriate instance of) B has not yet been deduced again.

We obtain these side conditions systematically as follows: in the soundness proof of restart (page 4) we convert the instances of restart into an appeal to Peirce's law. That is, we show that for every deduction using the restart rule there is a direct way of replacing the appeal to restart with an appeal to Peirce's law. Actually, it is not hard to see that we can replace the applications of restart with appeals to a particular instance of Peirce's law: $((A \to \bot) \to A) \to A$.

The deduction (segment) with which we replace the application of restart contains the assumption, and subsequent discharge, of $(A \to \bot) \to A$. This assumption of $(A \to \bot) \to A$ (which formerly did not appear in the deduction at all) may interfere with the side conditions of any applications of $(\forall I)$ or $(\exists E)$ below it. Schematically, if we replace an application of restart

$$\begin{array}{c}\dfrac{A}{B}\ (Restart)\\ \vdots\ \Pi\\ A\end{array}$$

by this

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{A \qquad [A \to \bot]^1}{\bot}\ (\to E)}{B}\ (\bot E)\\ \vdots\ \Pi\\ A\end{array}}{(A \to B) \to A}\ (\to I)^1 \qquad \dfrac{}{((A \to \bot) \to A) \to A}\ \text{Assumed}}{A}\ (\to E)$$

then any applications of $(\forall I)$ or $(\exists E)$ in Π may become invalidated if x or c is free in A. The new side condition ensures that this does not happen as the application of restart in question is incomplete throughout Π and so no application of $(\forall I)$ or $(\exists E)$ makes critical use of any x or c that is free in A.

[12] A^x_t is the formula obtained by simultaneously replacing all free occurrences of the variable x in A by the term t

It is now not hard to derive the following theorem:

THEOREM 2. *Intuitionistic FOL plus* (*Restart*) *has the same entailment relation as Classical FOL.*

We have just shown the left right direction of this theorem, and the right left direction is shown in the same way as with Theorem 1

Normalisation follows by an easy extension of the arguments of Section 2.2. For example here is the reduction case for the existential quantifier (given the familiar side condition on $\exists I$ we may suppose that no variable in t is bound in A^x_t).

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\Pi_1\\ A^x_t\end{array}}{\exists xA}\,(\exists I)\qquad \begin{array}{c}[A^x_c]^i\\ \vdots\,\Pi_2\\ B\end{array}}{B}\,(\exists E)^i \qquad\Longrightarrow\qquad \begin{array}{c}\vdots\,\Pi_1\\ A^x_t\\ \vdots\,\Pi_2'\\ B\end{array}$$

Where Π_2' is obtained from Π_1 by making the following replacements:

1. First replace any constant or variable symbol in Π_2 that is not free in any assumption of Π_2 but is free in assumptions of Π_1 by some other variable or constant symbol that is not free in any assumptions of Π_1.

2. Finally, replace c by t throughout Π_2

It is left to the reader to verify, bearing in mind the rules for the quantifiers, that Π_2' may be appended to the end of Π_1 to yield a valid deduction.

There is much interesting research to be done in the relation between more complex restart rules, hash logic and first order intermediate logics. Restart seems quite robust and suggests in a natural way the side-conditions and reduction rules necessary to make things work. We now continue with another example.

8.2 Endpoint Restart

Consider this restart rule, call it *endpoint restart*:

$$\begin{array}{c}\dfrac{A}{B}\\ \vdots\\ A\\ \vdots\\ \bot\end{array}$$

which allows us to infer B from A provided we infer A again later and then $\bot$ still later. In relation to the side conditions on the universal quantifier, if x is free in A, then we cannot introduce $\forall xD$ by $\forall I$ until after A is deduced again (but we can do so before $\bot$ is deduced again as x is not free in $\bot$).

This more complex restart rule allows us to deduce the following formula

$$NS = \forall x\neg\neg A \to \neg\neg\forall xA$$

$$
\begin{prooftree}
\AxiomC{$[\forall x \neg\neg A]^3$}
\RightLabel{$\forall E$}
\UnaryInfC{$\neg\neg A$}
\AxiomC{$[A]^1$}
\UnaryInfC{$\bot$}
\RightLabel{$(\rightarrow I)(1)$}
\UnaryInfC{$\neg A$}
\RightLabel{$(\rightarrow E)$}
\BinaryInfC{$\bot$}
\RightLabel{$(\bot E)$}
\UnaryInfC{A}
\RightLabel{$\forall I$}
\UnaryInfC{$\forall x A$}
\AxiomC{$[\neg\forall x A]^2$}
\RightLabel{$(\rightarrow E)$}
\BinaryInfC{$\bot$}
\RightLabel{$(\rightarrow I)(2)$}
\UnaryInfC{$\neg\neg\forall x A$}
\RightLabel{$(\rightarrow I)(3)$}
\UnaryInfC{$\forall x \neg\neg A \rightarrow \neg\neg\forall x A$}
\end{prooftree}
$$

and the application of $\forall I$ is legitimate because the requirement to re-deduce A has been met (although the application of restart is not fully complete as $\bot$ must be deduced). Furthermore observe that for any deduction using the new restart there is a deduction of the same conclusion using the axiom $\neg\neg\forall x(((A \rightarrow B) \rightarrow A) \rightarrow A)$. We can convert any application of endpoint restart thusly

$$
\begin{prooftree}
\AxiomC{$\neg\neg\forall x(((A \rightarrow B) \rightarrow A) \rightarrow A)$}
\AxiomC{$[\forall x(((A \rightarrow B) \rightarrow A) \rightarrow A)]^2$}
\UnaryInfC{$((A \rightarrow B) \rightarrow A) \rightarrow A$}
\AxiomC{$\vdots$}
\noLine
\UnaryInfC{A}
\AxiomC{$[A \rightarrow B]^1$}
\RightLabel{$(\rightarrow E)$}
\BinaryInfC{B}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{A}
\RightLabel{$(\rightarrow I)^1$}
\UnaryInfC{$(A \rightarrow B) \rightarrow A$}
\RightLabel{$\forall E$}
\BinaryInfC{A}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\bot$}
\RightLabel{$(\rightarrow I)^2$}
\UnaryInfC{$\neg\forall x(((A \rightarrow B) \rightarrow A) \rightarrow A)$}
\BinaryInfC{$\bot$}
\end{prooftree}
$$

But it is a fact of intuisionistic logic that

$$\forall x \neg\neg A \rightarrow \neg\neg\forall x A \;\vdash\; \neg\neg\forall x(((A \rightarrow B) \rightarrow A) \rightarrow A)^{13}$$

and so we can convert any applications of endpoint restart into an appeals to NS.

Adding NS as an axiom to intuitionistic logic yields an intermediate logic complete for Kripke frames where each world has an endpoint accessible to it (see [van Dalen, 1986, Gabbay, 1981]), that is:

$$\forall w \exists w'(wRw' \& \forall w''(w'Rw'' \rightarrow w' = w''))$$

Notice that we can already express such a condition using the hash operator $\neg\neg\#\bot$. And indeed, we can deduce NS from $\neg\neg\#\bot$ (see 8.3).

Interestingly however, we cannot deduce $\neg\neg\#\bot$ if we add NS as an axiom. To see this, consider a Kripke frame with a set of worlds W and an accessibility relation such that wRw' for every $w, w' \in W$, i.e. the accessibility relation is an equivalence class on W. In such a frame NS is clearly true at all worlds, but $\#\bot$ is not true at any world.

What has happened is that NS does not force a Kripke frame to have endpoints, but it does force a frame into something indistinguishable from one with endpoints (a frame where there are always worlds where all propositions are determined). But with the additional expressive power of the # operator we can make such distinctions

[13] $((A \rightarrow B) \rightarrow A) \rightarrow A$ is a propositional classical theorem and so $\neg\neg(((A \rightarrow B) \rightarrow A) \rightarrow A)$ is an intuitionistic theorem, and therefore so is its universal generalisation etc.

and so, in the presence of #, the intermediate logic with NS as an axiom (named MH in [Gabbay, 1981]) is not complete for frames where every world has an accessible endpoint. It follows that NS and endpoint restart are complete for Kripke frames where every world has an accessible endpoint only if the language does not contain #.

Normalisation of endpoint restart follows almost exactly as with ordinary restart: because $\bot$ has no introduction rule, the extra condition that $\bot$ be deduced after A has no effect on the normalisation argument.

The goal-directed formulation of endpoint restart is this

> We may replace a goal $?B$ with a previous goal $?A$ (i.e. act as if we have deduced B when we have really deduced A) provided that there is a goal $?\bot$ previous to the goal $?A$.

To see this work it is convenient to have the following introduction rule for the universal quantifier:

$$\begin{array}{l|l|l}
r & \quad ?A^x_y & y \text{ does not occur above} \\
\vdots & \quad \vdots & \\
r' & \quad A^x_y & \\
 & \forall x A & \forall I,\ r,\ r'
\end{array}$$

That is, we assume nothing and set $A[x/y]$ as our goal where y is new to the deduction, then when we reach that goal we may conclude $\forall x A$. Now, here is a deduction of NS using goal-directed endpoint restart:

$$\begin{array}{l|l|l}
1 & \quad \forall x\neg\neg A \quad ?\neg\neg\forall x A & \\
2 & \qquad \neg\forall x A \quad ?\bot & \\
3 & \qquad\quad ?A^x_y & \\
4 & \qquad\quad \neg\neg A^x_y & \forall E,\ 1 \\
5 & \qquad\qquad A^x_y \quad ?\bot & \\
6 & \qquad\qquad A^x_y & \text{recalling},\ 5 \\
7 & \qquad\quad \neg A^x_y & \text{endpoint restart},\ 3,\ 5,\ 6 \\
8 & \qquad\quad \bot & (\to E),\ 4,\ 7 \\
9 & \qquad\quad A^x_y & (\bot E),\ 8 \\
10 & \qquad \forall x A & \forall I,\ 3,\ 9 \\
11 & \qquad \bot & (\to E),\ 2,\ 10 \\
12 & \quad \neg\neg\forall x A & (\to I),\ 2,\ 11 \\
13 & \forall x\neg\neg A \to \neg\neg\forall x A & (\to I),\ 1,\ 12
\end{array}$$

What is interesting about this particular restart rule is that it does not mention quantifiers at all. Looking carefully at the rule we can see that it has no effect on the propositional fragment of intuitionistic logic. Without quantifiers it corresponds to an appeal to the double negation of Peirce's law which is an intuitionistic theorem.[14] With quantifiers endpoint restart corresponds to an appeal to the double negation of a universal generalisation of Peirce's law (and that is not an intuitionistic theorem).

[14] Also, propositional intuitionistic logic is complete for finite Kripke models. So the condition that every world has an accessible endpoint is trivially met in the propositional case.

8.3 More fun with hash

Endpoint restart was not discovered by accident, it was noticed by externalising the effect of adding $\neg\neg\#\bot$ as an axiom. That is, # internalises into intuitionistic logic the deduction structure of restart. And so more complex formulae involving hash internalise more complex deduction structures. It is interesting to see what restarts we get if we attempt to put these structures back into the external structure of natural deduction (in the form of more complex restart rules). It is not hard to see how the axiom $\neg\neg\#\bot$ justifies endpoint restart:[15]

$$\dfrac{\neg\neg\#\bot \qquad \dfrac{\begin{array}{c}\dfrac{\dfrac{[\#\bot]^1 \quad A}{\bot}\,(\#E)}{B}\,(\bot E)\\ \vdots\\ A\\ \vdots\\ \bot\end{array}}{\neg\#\bot}\,(\to I)(1)}{\bot}$$

and we obtained endpoint restart from considering that deduction. We suspect many more restart rules can be obtained in this manner.

9 Conclusions and future work

It is always nice when we can extend a system without having to reengineer our existing material (say it is **inherited**). For example it is nice to be able to extend a logic in such a way that deductions of the unextended logic are still deductions of the extended logic, and valid proof-reductions of the unextended logic are still valid proof-reductions of the extended logic.

Examples of this principle in action are not hard to find wherever people have invested effort in a particular system, and then find they want to extend it. Besides saved effort, the chances are that if formal structure is preserved, so may the good properties (e.g. proof-normalisation!).

This paper presents a robust and elementary method by which logical derivability may be extended, without affecting the structure of deductions or their normalisation. We have applied it here both to intuitionistic Natural Deduction and to Gabbay's Goal-Directed Deduction. We consider consider restart-like methods of obtaining interesting and sometimes exotic logics. This also gives some new perspectives on their computational content.

On a philosophical note we are following work of Gentzen, Prawitz and Dummett (to name a few) that a deduction theoretic account of the meaning of a logical connective can be found in its introduction and elimination rules (intro/elim pair). In this view, the intro/elim pair of, say implication, define its meaning. A *useful* definition of meaning is an intro/elim pair which normalises in the context of the rest of the logic. As we have seen, natural deduction plus restart is convenient for defining classical logics in a way that, for example, natural deduction plus double negation elimination is not.

We made it obvious that Classical Restart corresponds to multiple conclusions, but it is not *identical* to multiple conclusions, any more than Natural Deduction is *identical* to sequent systems. Indeed, we can argue that Restart is the true meaning of multiple conclusions!

[15] And since endpoint restart can be used to deduce NS it follows that $\neg\neg\#\bot$ deduces NS.

There is a much work on ‘exotic’ deduction systems for logics a little off the beaten track. We have dabbled in this in the paper with the hash logic and with n-depth restart. **Hypersequents**, introduced by Arnon Avron [Avron, 1996], are another example in the literature. Amusingly hypersequents go in entirely the opposite direction to this paper; instead of doing away with sequent structure, they enrich it. And yet they seem to arrive at similar applications to us:

Ciabattoni and Ferrari [Ciabattoni and Ferrari, 2000] present deduction-systems for logics complete for Kripke models with restrictions on their geometry, including bounded depth like our n-depth restart but also bounded cardinality, bounded width, and so on — this is much more general than what we managed.

The connection between our determinedly Natural Deduction restart rules, and the superficially quite different determinedly Sequent Calculus hypersequent systems, is in the connection between the *proof-teleportation* which we have seen in this paper, which arises naturally from restart proof-normalisation, in which fragments of deduction are copied (teleported) deep inside intermediate parts of other deductions; on the other hand hypersequents on the other hand derive their power by allowing us to *rearrange and copy assumptions* inside intermediate steps in the sequent deduction. In the first, we move a fragment of deduction to a new context, and in the second we move the new context to be around a fragment of deduction — in the end, it seems to come to almost the same thing.

Hypersequents seem to be more general and expressive than restart rules in that they express a wider range of intermediate logics. Yet Restart rules are nice when they work, they retain the simple Natural Deduction presentation and so are ‘cheap’ in the sense of the Introduction. This can be seen in the ease by which we can internalise the restart rule into the hash logic which is a conservative extension of classical logic (this follows from the normal form theorem on hash logic). The hash logic itself is interesting and further work into the extra expressive power and computational content of #, and connectives axiomatisable using #, seems promising. Indeed, as far as we are aware, the intermediate logic MH (see section 8.2) has not been formulated in terms of hypersequents.

Viewed computationally, Classical Restart leads immediately to a λ-calculus very similar to Parigot’s $\lambda\mu$-calculus (restarts-yet-to-be-justified, and the current goal, correspond to multiple conclusions). Parigot’s great accomplishment was to ‘pull out of the air’ the $\lambda\mu$-calculus in 1992, and in retrospect we now see that Dov Gabbay had pulled it out of the air too, in 1984 and in a different field as goal-directed restart. This should serve as a taster to interesting calculi that can be produced for the predicate case, for hash logic, and for any other restart rule.

Perhaps the most striking presentation of restart rules is still in the original presentation in goal directed formulations. The restart rule is somewhat obvious in the rigorous and highly annotated structure of (Lemmon style) linear natural deduction systems. The effects of restart and hash rules in their natural deduction formulations are unexpected at first, but then intelligible. But in their goal-directed formulations the restart and hash rules seem, at least to the authors of this article, almost magical. The discoverer of the restart rule, who spotted it directly in its goal directed formulation, is surely to be commended on such a leap of the imagination.

Acknowledgements

We are grateful to Agata Ciabattoni, Alexis Saurin, and David Richter, for useful comments, and to two anonymous referees.

BIBLIOGRAPHY

[Avron, 1996] A. Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In W. Hodges, M. Hyland, C. Steinhorn, and J. Truss, editors, *Logic: from foundations to applications. Proc. Logic Colloquium, Keele, UK, 1993*, pp. 1–32. oup, New York, 1996.

[Barendregt, 2000] H. P. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science, Volume 2*, pp. 117–309. OUP, 2000.

[Bell and Machover, 1977] J. Bell and M. Machover. *A course in mathematical logic*. North-Holland, 1977.

[Braüner, 2000] T. Braüner. A cut-free gentzen formulation of the modal logic s5. *Logic Journal of IGPL*, 8(5):629–643, 2000.

[Ciabattoni and Ferrari, 2000] A. Ciabattoni and M. Ferrari. Hypertableau and path-hypertableau calculi for some families of intermediate logics. In *Automated Reasoning with Tableaux and Related Methods (Tableaux'2000)*, volume 1847 of *LNAI*, pp. 160–175, 2000.

[Coquand, 1996] T. Coquand. Computational content of classical logic. In P. Dybjer and A. Pitts, editors, *CLICS Cambridge Summer School.*, 1996.

[de Groote, 1995] P. de Groote. A simple calculus of exception handling. In *LNCS*, volume 902, pp. 201–215. Springer, 1995. TLCA'95.

[Gabbay and Olivetti, 2000a] D. Gabbay and N. Olivetti. *Goal Directed Algorithmic Proof Theory*. Kluwer Academic, 2000.

[Gabbay and Olivetti, 2000b] D. Gabbay and N. Olivetti. *Goal-directed Proof Theory*. Kluwer Academic Publishers, 2000.

[Gabbay and Reyle, 1984] D. Gabbay and U. Reyle. N-prolog: An extension of prolog with hypothetical implications i. *Journal of Logic Programming*, (1):391–355, 1984.

[Gabbay, 1981] D. Gabbay. *Semantical investigations in Heyting's Intuitionistic Logic*, volume 148 of *Synthèse*. D. Reidel, 1981.

[Gabbay, 1990] D. Gabbay. Algorithmic proof with diminishing resources part i. In *CSL'90*, volume 533 of *LNCS*, pp. 156–173. Springer, 1990.

[Gabbay, 1998] D. Gabbay. *Elementary Logic: A Procedural Perspective*. Prentice Hall, 1998.

[Gabbay, 2004] Michael Gabbay. *Proof-Normalisation and Truth by Definition*. PhD thesis, King's College, London, 2004.

[Gentzen, 1934] G. Gentzen. Untersuchungen über das logische Schliessen. *Mathematische Zeitschrift*, 39:176–210 and 405–431, 1934.

[M. H. B. Sorensen, 1998] P. Urzyczyn M. H. B. Sorensen. Lectures on the curry-howard isomorphism. Technical Report 98/14, DIKU, 1998.

[Parigot, 1992] M. Parigot. Lambda-mu-calculus: An algorithmic interpretation of classical natural deduction. In *LPAR'92*, pp. 190–201, 1992.

[Peyton-Jones, 2001] S. Peyton-Jones. Tackling the awkward squad: monadic input/output, concurrency, exceptions, and foreign-language calls in haskell. In *Engineering theories of software construction, Marktoberdorf Summer School*, NATO ASI, pp. 47–96. 105 Press, 2001.

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

[Stalmarck, 1991] G. Stalmarck. Normalization theorems for full first order classical natural deduction. *Journal of Symbolic Logic*, 56:129–149, 1991.

[Taylor, 1999] P. Taylor. *Practical Foundations of Mathematics*. Number 59 in Cambridge Studies in Advanced Mathematics. 1999.

[van Dalen, 1986] D. van Dalen. Intuitionistic logic. In Gabbay and Günthner, editors, *Handbook of Philosophical Logic vol III*, volume 166 of *Synthèse*, chapter 4. D. Reidel, 1986.