Structuring Metatheory on Inductive Definitions

David Basin and Seán Matthews
Max-Planck-Institut für Informatik,
Im Stadtwald, D-66123 Saarbrücken, Germany,
{basin,sean}@mpi-sb.mpg.de

April 24, 1996

Abstract

We examine a problem in formal metatheory: if theories are structured hierarchically, there are metatheorems which hold in only some extensions. We illustrate this using modal logics and the deduction theorem. We show how statements of metatheorems in such hierarchies can take account of possible theory extensions; i.e. a metatheorem formalizes not only the theory in which it holds, but also under what extensions, both to the language and proof system, it remains valid. We show that FS_0, a system for formal metamathematics, provides a basis for organizing theories this way, and we report on our practical experience.

Supporting documentation: A machine-checked transcript available at `http://www.mpi-sb.mpg.de/~sean/papers.html` contains the definitions and proofs described in this paper.

1 Introduction

Theory structuring plays an important role in the application of theorem provers to nontrivial problems, and many systems provide support for it. For example, HOL [6], Isabelle [14] and their predecessor LCF [7] support simple theory hierarchies: a theory specifies a language, using types and typed constants, and axioms for proving theorems, and each theory inherits the types, constants, axioms and theorems from its ancestors. This way, results proven for some theory can be inherited automatically by extensions. The drawback to this approach is that types associated with the language and proof system must be considered 'open', limiting the ways proofs can be constructed. In particular, we cannot reason by induction on the structure of the language or proofs. We can 'close' the language or proof system by explicitly adding induction principles, but then it is generally unsound later to add new axioms, or constructors to the language; i.e. extensions are impossible.

Our work addresses a difficulty for structured theory development that arises in formal metatheory: the simple hierarchical approach described above, which in effect offers an 'all or nothing' approach to inheriting results, is sometimes too inflexible, because some metatheorems need induction over the proof or language of a theory, but also hold in *some* extensions. If we are to exploit such metatheorems in supertheories then we need a framework allowing finer control over the scope of metatheorems.

We present a solution to this problem. We show how theories and development can be structured using inductive definitions so that we are able to describe not only the theories in which results apply but also the possible valid extensions. This way it is possible to import a metatheorem into a new context simply by checking that the context satisfies the conditions specified in the theory. We demonstrate that FS_0, a framework for formal metatheory due to Feferman [5], provides an adequate basis for organizing theories this way, and report on experiments with an implementation which indicate that our approach works well in practice. The core definitions and proofs presented in this paper have all been machine checked by an implementation of FS_0.

Motivation

We expand on our motivation and our proposed approach. Formal details will follow in subsequent sections.

In formal metatheory, one has a theorem prover with its own logic, the *metalogic*, which formalizes the syntax and proof rules of other logics, the *object logics*. Well-known metalogics are type theories like the Edinburgh LF [8] or minimal logics with quantification over higher-types, like the logic of Isabelle. Such metalogics are used to formalize sets (or types) corresponding to the language and proofs of particular object logics.

The example from formal metatheory we will use concerns hierarchies of Hilbert systems. Hilbert systems are interesting because they are commonly acknowledged to be the easiest to prove theorems *about*, but are also the hardest to prove theorems *in*. In practice, to prove theorems with them we need a metatheorem, the deduction theorem, the application of which allows us to simulate 'natural deduction' style proof under assumption. To prove the deduction theorem we need induction over the structure of derivations in a theory. However, as noted, if this theorem is to be used in theory extensions then we need a new approach to structuring hierarchies of theories. We start by informally developing a hierarchy of theories and considering the sort of facilities required.

At the bottom of our hierarchy is *arrow logic* (minimal implicative propositional logic), the formulae of which correspond to members of the set $\mathcal{L}_{\mathcal{A}}$ of formulae built from the connective $\rightarrow$ and sentential constants. Theorems correspond to members of a second set, $\mathcal{T}_{\mathcal{A}}$, and are instances of the standard Hilbert axioms

$$A \rightarrow B \rightarrow A \qquad (K)$$

and

$$(A \rightarrow B) \rightarrow (A \rightarrow B \rightarrow C) \rightarrow (A \rightarrow C) \qquad (S)$$

or are generated by applying the rule *modus ponens* to previously proven theorems. We write $\vdash_{\mathcal{A}} A$ to indicate that $A \in \mathcal{T}_{\mathcal{A}}$. Assume that our (not yet specified) metalogic contains at least the implies connective $\Rightarrow$, then two theorems we can prove in this metalogic are:

$$\vdash_{\mathcal{A}} A \to A \qquad (I)$$

$$(\vdash_{\mathcal{A}} A \to C) \Rightarrow (\vdash_{\mathcal{A}} A \to B \to C) \qquad (\mathit{thin})$$

The first of these asserts the provability of a formula in the object logic and the second represents a derived proof rule. Both of these can be made schematic, though formalization in a metatheory then requires universal quantification over $\mathcal{L}_{\mathcal{A}}$ in the metalogic; we will often leave such quantifiers implicit.

Now consider the deduction theorem: if we define $A \vdash_{\mathcal{A}} B$ as meta-notation ($\mathcal{L}_{\mathcal{A}}$ is a language of formulae, not sequents) meaning that if the Hilbert system $\mathcal{T}_{\mathcal{A}}$ is extended with the additional axiom A, then B belongs to the resulting set of proofs, then we can state this as

$$A \vdash_{\mathcal{A}} B \Rightarrow \vdash_{\mathcal{A}} A \to B\,. \qquad (DT)$$

Notice that just to state this theorem, which is schematic in A (and B), we must refer to infinitely many extensions of our original Hilbert system.

To go to the next level of our hierarchy we could extend arrow logic to propositional logic (we consider here only conjunction, other operators are similar): we extend $\mathcal{L}_{\mathcal{A}}$ to $\mathcal{L}_{\mathcal{P}}$ by adding the new binary connective $\wedge$ and extend the proof type by adding the three new axiom schemata:

$$\begin{array}{c} A \to B \to A \wedge B \\ A \wedge B \to A \\ A \wedge B \to B \end{array} \qquad (\mathit{Conj})$$

We expect that all theorems of arrow logic, e.g. (I), are still theorems of propositional logic provided that the original axioms ((S) and (K)) and inference rule (modus ponens) are interpreted over the new language. The rule (*thin*) clearly also holds. The status of the deduction theorem is less obvious: this also holds for propositional logic, but since the proof of this theorem is by induction on the structure of arrow logic proofs, we must check the new cases corresponding to the new axioms. If we used a logic like HOL as a metalogic and formulated these two proof systems as inductive definitions, we would need to prove the deduction theorem individually for each system.

Now consider another extension, to the normal propositional modal logic K. Extend the language $\mathcal{L}_{\mathcal{A}}$ to $\mathcal{L}_{\mathcal{M}}$ by adding the unary modality $\Box$ and extend $\mathcal{T}_{\mathcal{A}}$ to $\mathcal{T}_{\mathcal{M}}$ by adding an axiom stating the distributivity of necessity over implication

$$\Box(A \to B) \to (\Box A \to \Box B) \qquad (N)$$

and adding a rule of inference

$$\vdash_{\mathcal{M}} A \Rightarrow \vdash_{\mathcal{M}} \Box A\,. \qquad (\mathit{nec})$$

Which metatheorems about $\mathcal{T}_{\mathcal{A}}$ should be applicable in $\mathcal{T}_{\mathcal{M}}$? Both (I) and $(thin)$ hold in any extension of $\mathcal{T}_{\mathcal{A}}$ with new axioms. Furthermore, they hold when $\mathcal{T}_{\mathcal{A}}$ is extended with new rules and also when the language of $\mathcal{L}_{\mathcal{A}}$ is extended to $\mathcal{L}_{\mathcal{M}}$ provided the old axioms and inference rule modus ponens are interpreted over the new language. The deduction theorem (DT) holds too under extensions of $\mathcal{T}_{\mathcal{A}}$ with new axioms. But it does not hold under extensions with arbitrary new rules; indeed, it fails for $\mathcal{T}_{\mathcal{M}}$ with the additional rule (nec). We will later see, however, that instances of it hold in some extensions provided that certain conditions are satisfied.

To structure theories so that we can use metatheorems in appropriate extensions we have to be able to formalize the above kinds of provisos and prove metatheorems under them. This is supported by the use of inductive definitions in our metalogic, FS_0. An inductive definition corresponds to a term $I(base, step)$ where *base* and *step* are sets that formalize the base and step cases of the definition. The definition defines a new set, which we can quantify over and reason about by induction. The language and proof system of a theory each correspond to such a definition. For example, for $\mathcal{L}_{\mathcal{A}}$, $base_{\mathcal{L}_{\mathcal{A}}}$ is the set of sentential constants, and $step_{\mathcal{L}_{\mathcal{A}}}$ explains how two formulae in $\mathcal{L}_{\mathcal{A}}$ can be combined under a connective $\rightarrow$. For $\mathcal{T}_{\mathcal{A}}$, $base_{\mathcal{T}_{\mathcal{A}}}$ contains all instances of the axioms (S) and (K), and $step_{\mathcal{T}_{\mathcal{A}}}$ corresponds to modus ponens.

Structuring is supported by the way we can form *open inductive definitions*, which are definitions where the base and step are augmented by concrete or hypothetical extensions. For example, let $\vdash_{\mathcal{A}\cup\{A\}}$ be $I(base_{\mathcal{T}_{\mathcal{A}}} \cup \{A\}, step_{\mathcal{T}_{\mathcal{A}}})$ and we can formulate the deduction theorem as

$$\vdash_{\mathcal{A}\cup\{A\}} B \Rightarrow \vdash_{\mathcal{A}} A \rightarrow B\,. \qquad (DT_2)$$

The provisos on, and scope of, metatheorems can then be formalized in terms of extension sets which are as strong as possible, i.e. we state a metatheorem for $I(base \cup \Gamma, step \cup \Delta)$ rather than $I(base, step)$ and then place conditions on Γ and Δ. By taking Γ and Δ to be empty, we get the metatheorem for the original theory. Thus, for instance, (I) and $(thin)$ hold for extensions of the proof system $base_{\mathcal{T}_{\mathcal{A}}}$ and $step_{\mathcal{T}_{\mathcal{A}}}$ with arbitrary Γ and Δ, whereas the deduction theorem holds only for arbitrary Γ.

In §2 we provide an overview of our implementation of FS_0 and show in §3 how we use it to implement the kind of metatheoretic development described above where even theorems like the deduction theorem can be proven, once and for all, to hold for infinitely many proof systems. This provides a way to construct natural deduction style proofs under assumption in infinitely many extensions of our hierarchy. In §4 we relate this to standard metatheory were rules are sometimes classified as *admissible* and *derived*; our approach allows us to formalize other kinds of rules useful for machine supported theorem proving and go beyond what is possible in theorem provers like Isabelle or HOL where theories are hierarchically structured. We also make comparisons with other approaches to structuring theories and discuss directions for future research.

2 Background: The Theory FS_0

We use Feferman's FS_0, which we have implemented in Isabelle [14], for our development. FS_0 is a theory of inductive definitions designed for formal metamathematics. Since we will not give concrete FS_0 proofs in this paper, we shall review, briefly, only those aspects of FS_0 and our implementation that are used in this paper. A full description of FS_0 and our implementation are given in [5, 11, 12].

FS_0 is a theory of inductively defined sets of s-expressions, based on three-sorted classical first-order logic with equality: The first sort, $\mathcal{S}$, of s-expressions contains nil, and is closed under pairing, written, $(\cdot,\cdot)$. The second sort, $\mathcal{F}$, is of (primitive recursive) functions. The third sort, $\mathcal{C}$, is of sets (or classes); it contains $\{nil\}$ and new sets are constructed from defined ones using binary intersection and union, and by taking the inverse image of a set under a function. Equality and the subset relation on sets are defined extensionally.

Central to our work is that sets can be built by inductive definitions: if a and b are sets, then $I(a,b)$ is the least set containing a and closed under a rule

$$\frac{t_1 \quad t_2}{t}$$

where $((t,t_1),t_2) \in b$. We can reason about inductively defined sets using the induction principle

$$base \subset s \Rightarrow \forall a,b,c(b,c \in s \Rightarrow ((a,b),c) \in step \Rightarrow a \in s) \Rightarrow I(base, step) \subset s\,.$$

To help understand this induction principle, consider the set $I(base, step)$. This is the inductively defined set where $base$ defines a set of s-expressions corresponding to the base-case of the inductive definition. In the step case, $step$ corresponds to a three place relation, encoded as a set of s-expressions, where if $((z,x),y) \in step$ then from $x \in I(base, step)$ and $y \in I(base, step)$ we can conclude $z \in I(base, step)$; i.e. it is a rule which explains how given elements belonging to the inductively defined set, we can generate new elements. Note that although in FS_0 we are allowed only inductive definitions with exactly two 'predecessors' this is sufficient for the needs of this paper (in fact it is enough for general finitary definitions). Also notice that there are no typing restrictions; rules are just special instances of sets, and *any* set can be used as a rule (i.e. $step$ in the set above is not a set of 3-tuples, but a set of s-expressions, some of which are 3-tuples).

FS_0 provides comprehension over a large class of first-order (specifically Σ^0_1) predicates. This is automatically provided by our implementation and we write

$$x \in s \Leftrightarrow P(x)$$

to indicate a set defined this way. Such definitions can also be parameterized and the parameters are treated as constants of the appropriate sort; e.g.

$$x \in s(a,b) \Leftrightarrow (x,a) \in b$$

defines a set s parameterized over an s-expression a and a set b.

All the basic results in this paper have been implemented and checked using our implementation of FS_0. For the sake of brevity and readability we use, in this paper, more standard mathematical syntax; however a transcript of the development is available at `http://www.mpi-sb.mpg.de/~sean/papers.html`. We relate the transcript to this paper by using a special definition and theorem format for machine checked mathematics; we write, e.g.,

M-Definition 1 (`a_definition`) *This describes how* `a_definition` *is defined in the transcript.*

M-Theorem 2 (`a_thm`) *This describes the machine checked result associated with the name* `a_thm` *in the transcript.*

3 Structured Metatheory

We now formalize the hierarchy of propositional Hilbert systems given in the introduction. We begin by showing how inductive definitions can be made in an *open* way which allows us to specify that a theorem holds in all extensions of the language and proof system. After, we show how more control over scope is possible and how this allows new and useful kinds of metatheorems, like parameterized versions of the deduction theorem, which hold in some, but not all extensions.

Arrow Logic

We begin by defining the language $\mathcal{L}_\mathcal{A}$ and proof system $\mathcal{T}_\mathcal{A}$. $\mathcal{L}_\mathcal{A}$ is the smallest set satisfying

$$\mathcal{L}_\mathcal{A} ::= P \mid (\to, (\mathcal{L}_\mathcal{A}, \mathcal{L}_\mathcal{A}))\,,$$

where P is some set of (atomic) propositions.

M-Definition 1 (`LA`) *First we define some constant s-expression '$\to$', which we use to mark syntactic objects. Then we define*

$$x \in imp\_rule \Leftrightarrow \exists a, b \in \mathcal{S}(x = (((\to, (a, b)), a), b))\,,$$

at which point we could define $\mathcal{L}_\mathcal{A} \equiv I(P, imp\_rule)$, where P is a set of propositions. However, we make a more general definition, parameterized over arbitrary base case, Γ, and step case, Δ, extensions:

$$\mathcal{L}_\mathcal{A}[\Gamma, \Delta] \equiv I(\Gamma, \Delta \cup imp\_rule)\,.$$

Then the language $\mathcal{L}_\mathcal{A}$ is simply $\mathcal{L}_\mathcal{A}[P, \emptyset]$, where P is some class of atomic propositions.

We can then show that $(\to, (a, a)) \in \mathcal{L}_\mathcal{A}[P, \Delta]$ given that $a \in P$. In the future we will use standard syntax, using context (and different symbols) to disambiguate object-level/meta-level syntax; e.g. rather than $(\rightarrow, (a, a))$, we write simply $a \rightarrow a$. Also, we will write formulae built from '$\rightarrow$' associated to the right.

In a similar manner we can declare the family of proof systems $\mathcal{T}_{\mathcal{A}}$.

M-Definition 2 (`TA`) *Let*

$$\begin{array}{rcl} x \in K & \Leftrightarrow & \exists a, b \in \mathcal{L}_{\mathcal{A}}(x = a \rightarrow b \rightarrow a) \\ x \in S & \Leftrightarrow & \exists a, b, c \in \mathcal{L}_{\mathcal{A}}(x = (a \rightarrow b) \rightarrow (a \rightarrow b \rightarrow c) \rightarrow a \rightarrow c) \\ x \in mp & \Leftrightarrow & \exists a, b \in \mathcal{L}_{\mathcal{A}}(x = ((b, a), a \rightarrow b)) \\ \mathcal{T}_{\mathcal{A}}[\Gamma, \Delta] & \equiv & I(\Gamma \cup S \cup K, \Delta \cup mp) \end{array}$$

In the remainder of the paper we will use the following definitions for $\mathcal{T}_{\mathcal{A}}$ and other theories we introduce: $\mathcal{T}_{\mathcal{A}}$ (without parameters) denotes $\mathcal{T}_{\mathcal{A}}[\emptyset, \emptyset]$, $\vdash_{\mathcal{A}} a$ denotes $a \in \mathcal{T}_{\mathcal{A}}$, and $\vdash_{\mathcal{A}[\Gamma,\Delta]} a$ denotes $a \in \mathcal{T}_{\mathcal{A}}[\Gamma, \Delta]$.

$\mathcal{T}_{\mathcal{A}}$ formalizes arrow logic; to give a feel for it we show how one would prove (I). We begin by proving some trivial metatheorems:

M-Lemma 3 (`mp_arrow`) $\forall a, b \in \mathcal{L}_{\mathcal{A}}(\vdash_{\mathcal{A}} a \Rightarrow \vdash_{\mathcal{A}} a \rightarrow b \Rightarrow \vdash_{\mathcal{A}} b)$.

M-Lemma 4 (`s_arrow`) $\forall a, b, c \in \mathcal{L}_{\mathcal{A}}(\vdash_{\mathcal{A}} (a \rightarrow b) \rightarrow (a \rightarrow b \rightarrow c) \rightarrow a \rightarrow c)$.

M-Lemma 5 (`k_arrow`) $\forall a, b \in \mathcal{L}_{\mathcal{A}}(\vdash_{\mathcal{A}} a \rightarrow b \rightarrow a)$.

All of these follow directly from the definition of $\mathcal{T}_{\mathcal{A}}$. We then have (I):

M-Theorem 6 (`i_arrow1`) $\forall a \in \mathcal{L}_{\mathcal{A}}(\vdash_{\mathcal{A}} a \rightarrow a)$.

which we prove on the machine (see the transcript) using the following steps:

1.	$\vdash_{\mathcal{A}} (a \rightarrow b \rightarrow a) \rightarrow (a \rightarrow (b \rightarrow a) \rightarrow a) \rightarrow a \rightarrow a$	`s_arrow`
2.	$\vdash_{\mathcal{A}} a \rightarrow b \rightarrow a$	`k_arrow`
3.	$\vdash_{\mathcal{A}} a \rightarrow (b \rightarrow a) \rightarrow a$	`k_arrow`
4.	$\vdash_{\mathcal{A}} (a \rightarrow (b \rightarrow a) \rightarrow a) \rightarrow a \rightarrow a$	`mp_arrow` 1,2
5.	$\vdash_{\mathcal{A}} a \rightarrow a$	`mp_arrow` 3,4

This theorem has been shown just for $\mathcal{T}_{\mathcal{A}}$. However, if we examine the proof of `i_arrow`, we see that it made use only of `s_arrow`, `k_arrow` and `mp_arrow`, and that these in turn hold for arbitrary $\mathcal{T}_{\mathcal{A}}[\Gamma, \Delta]$, not just for the particular instance $\mathcal{T}_{\mathcal{A}}$. We call this property *openendedness* and observe that we can prove an openended version of `i_arrow`:

M-Theorem 7 (`i_arrow2`) $\forall a \in \mathcal{L}_{\mathcal{A}}(\vdash_{\mathcal{A}[\Gamma,\Delta]} a \rightarrow a)$.

We can use `i_arrow2` with any extension of our proof type with new axioms Γ and rules Δ. To avoid notational clutter, we leave implicit the universal quantification over these set variables.

This new metatheorem is openended with respect to the proof system: it demonstrates that $a \to a$ holds in any extension of the Hilbert system represented by $\mathcal{T}_\mathcal{A}$ with new axioms or rules. But even more is possible: the proof only requires implication and sentential constants in our language, but there could be other connectives. The most general form then is openended both with respect to the language and the proof system

M-Theorem 8 (`i_arrow`) $\forall a \in \mathcal{L}_\mathcal{A}[\Phi, \Psi](\vdash_{\mathcal{A}[\Gamma,\Delta]} a \to a)$.

Similarly, we can prove an openended version of (*thin*):

M-Theorem 9 (`thin_lem`)

$$\forall a, b, c \in \mathcal{L}_\mathcal{A}[\Phi, \Psi](\vdash_{\mathcal{A}[\Gamma,\Delta]} a \to c \Rightarrow \vdash_{\mathcal{A}[\Gamma,\Delta]} a \to b \to c).$$

In §4 we will classify several kinds of rules. The above theorems are similar to the kinds of rules derivable using theorem provers, such as Isabelle, in which theories are organized hierarchically and theorems proven in subtheories are inherited in supertheories.

The Deduction Theorem

We cannot prove openended versions of all metatheorems. An example of a metatheorem which does not hold in all theory extensions is the deduction theorem, which is one of the more important metatheorems about a Hilbert system. It is important as it is a practical necessity if one actually wants to carry out interactive proofs using a Hilbert system. Consider, for example, `i_arrow`. By the deduction theorem, we can prove $\vdash_\mathcal{A} a \to a$ by establishing $a \vdash_\mathcal{A} a$, i.e. $\vdash_{\mathcal{A}[\{a\},\emptyset]} a$, which is immediate. This is much more natural than the Hilbert proof given previously. By proving the deduction theorem, we gain the ability to simulate natural deduction style proof under assumption in the metatheory of the Hilbert system.

How would we prove the deduction theorem? Let us first sketch an informal proof: we prove $A \to B$ by induction on the structure of possible derivations of B with A is assumed as an axiom. There are two base cases:

1. B is an axiom. Then B follows without A. Since we have, as an axiom, $B \to A \to B$, by modus ponens we have $A \to B$.

2. B is A. Then we need to show $A \to A$, which we can do, following the proof of `i_arrow1` above.

In the step case B is a result of modus ponens from C and $C \to B$. By the induction hypothesis we have $A \to C$ and $A \to C \to B$, and as an axiom we have $(A \to C) \to (A \to C \to B) \to A \to B$. Then by modus ponens twice we have $A \to B$.

We can formalize the deduction theorem in FS_0 as

$$\forall a, b \in \mathcal{L}_\mathcal{A}(\vdash_{\mathcal{A}[\{a\},\emptyset]} b \Rightarrow \vdash_{\mathcal{A}[\emptyset,\emptyset]} a \to b)\,.$$

This theorem does not hold in all extensions; we could even state and prove in FS_0 that

$$\neg\forall\Gamma, \Delta\forall a, b \in \mathcal{L}_{\mathcal{A}}(\vdash_{\mathcal{A}[\{a\}\cup\Gamma,\Delta]} b \Rightarrow \vdash_{\mathcal{A}[\Gamma,\Delta]} a \to b)\,.$$

Although we cannot prove this openended version, a limited kind of extensibility is still possible. Studying the proof sketch we see that it:

- Requires $\to$ in the language $\mathcal{L}_{\mathcal{A}}$, but there could be other connectives.
- Requires the two axioms (S) and (K) but there could also be other axioms.
- Requires *exactly* modus ponens, but no more; if there were other rules then we would have more possibilities in the step case and arbitrary such possibilities are not covered by our proof.

Hence full extensibility is possible on the language side and *base*, but not *step*, thus the result we have verified in FS_0 is parametrised over Φ, Ψ, Γ as follows:

M-Theorem 10 (`dedthm`)

$$\forall a, b \in \mathcal{L}_{\mathcal{A}}[\Phi, \Psi](\vdash_{\mathcal{A}[\{a\}\cup\Gamma,\emptyset]} b \Rightarrow \vdash_{\mathcal{A}[\Gamma,\emptyset]} a \to b)\,.$$

This version allows us to apply the deduction theorem to infinitely many extensions of arrow logic. For example, we can extend $\mathcal{L}_{\mathcal{A}}$ to $\mathcal{L}_{\mathcal{P}}$ by adding conjunction and extending $\mathcal{T}_{\mathcal{A}}$ to $\mathcal{T}_{\mathcal{P}}$ by adding the three axioms given in (*Conj*). Since these are *axioms*, not *rules*, we can directly apply `dedthm` to prove theorems in this extended language and proof system; e.g.:

M-Theorem 11 (`i_prop`) $\forall a \in \mathcal{L}_{\mathcal{P}}(\vdash_{\mathcal{P}} a \to a)$.

Rule Extensions of the Deduction Theorem

The above metatheorem `dedthm` has been generalized to apply to all extensions of the language and those extensions of the proof system of $\mathcal{L}_{\mathcal{A}}$ with axioms. As it stands though, we must prove a new version (when the theorem still holds) for each extension of the theory with new proof rules. We can do better than this.

It is possible to prove a more general theorem by formalizing sufficient conditions for addition of new rules. Returning to the above proof: what is the relationship between (S), (K), and modus ponens? (K) and (I)—which follows from (K) and (S)—are needed for the two subcases of the base case. In the step case, (S) directly reflects the induction hypothesis corresponding to modus ponens. Modus ponens is the rule

$$\frac{C \quad C \to B}{B}$$

and in the induction step, each formula in this rule is prefixed by '$A \to$'; from $A \to C$ and $A \to C \to B$ we must show $A \to B$. If we internalize this (Curried) as a formula in the logic itself we have

$$(A \to C) \to (A \to C \to B) \to (A \to B)$$

which is just an instance of the (S) axiom.

Now suppose we have a general rule with two premises (this can be easily generalized to finitely many assumptions). Such a rule is of the form

$$\frac{B \quad C}{D}$$

A sufficient requirement for the deduction theorem is that in the step case we can use the assumptions $A \to B$ and $A \to C$ to prove $A \to D$. This is possible with the axiom schema

$$(A \to B) \to (A \to C) \to (A \to D)\,.$$

(S) is just the particular case needed to deal with modus ponens.

We can now state a metatheorem based on the above observation that generalizes `dedthm`. Rather than insisting that certain axioms are in the theory whenever the corresponding rules are, we directly formulate the provability requirement given above: the provability of $A \to B$ and $A \to C$ entails the provability of $A \to D$. We directly formalize this requirement within the statement of the generalized deduction theorem.

M-Theorem 12 (`dedthm_abs1`) *Let* $\mathcal{L}$ *be* $\mathcal{L}_{\mathcal{A}}[\Phi, \Psi]$, $\vdash$ *be* $\vdash_{\mathcal{A}[\Gamma,\Delta]}$ *and* $\vdash_{\{a\}}$ *be* $\vdash_{\mathcal{A}[\Gamma\cup\{a\},\Delta]}$ *then*

$$\begin{aligned}\forall a, b \in \mathcal{L}((\forall a, b, c, d \in \mathcal{L}((d, (b, c)) \in \Delta \\ \Rightarrow \vdash (a \to b) \Rightarrow \vdash (a \to c) \Rightarrow \vdash (a \to d))) \\ \Rightarrow \vdash_{\{a\}} b \Rightarrow \vdash a \to b)\,.\end{aligned}$$

This theorem states, in a general sense, the conditions necessary for the deduction theorem to hold. The following section shows, however, that it is not the strongest such theorem.

Modal Logic

In many logics that are simple extensions of arrow logic, the deduction theorem fails, and modal logic is one of these. Prawitz however showed that it is still possible in certain modal logics to reason under assumptions, as long as we place conditions on the sort of assumptions we are allowed [15].

In this subsection we discuss a natural 'partial' deduction theorem based upon the following:

M-Theorem 13 (`dedthm_abs2`) *Let* $\mathcal{L}$ *be* $\mathcal{L}_{\mathcal{A}}[\Phi, \Psi]$, $\vdash$ *be* $\vdash_{\mathcal{A}[\Gamma,\Delta]}$ *and* $\vdash_{\{a\}}$ *be* $\vdash_{\mathcal{A}[\Gamma\cup\{a\},\Delta]}$ *then*

$$\begin{aligned}\forall a, b \in \mathcal{L}(\forall a, b, c, d \in \mathcal{L}(a \in P \Rightarrow (d, (b, c)) \in \Delta \\ \Rightarrow \vdash (a \to b) \Rightarrow \vdash (a \to c) \Rightarrow \vdash (a \to d))) \\ \Rightarrow a \in P \Rightarrow \vdash_{\{a\}} b \Rightarrow \vdash a \to b\,.\end{aligned}$$

This generalizes `dedthm_abs1`, where the extra P can be used to place restrictions on the possible assumptions (set P to the universal set and we get `dedthm_abs1` as a special case). It is important to notice that this result says nothing about modal logic; it is an openended result that we have proved for arrow logic and arbitrary extensions.

Now consider modal logics. The modal logic K (not to be confused with the axiom schema (K)) is formed by extending a propositional language to $\mathcal{L}_{\mathcal{M}}$ by adding the unary modality $\Box$, and extending a propositional proof system to $\mathcal{T}_{\mathcal{M}}$ by adding the axiom (N) and the rule (nec). Note that this rule for necessitation cannot be replaced by any axiom.

We cannot use our parameterized deduction theorem for this logic, nor can we prove a new version for it. Let us return to our informal proof. We have a new step case for (nec). The formula B $(= \Box B')$ is the result of (nec) from B'. By the induction hypothesis we have $A \to B'$, and we want to prove $A \to \Box B'$. Unfortunately, this fails for modal logics in general. We can, however, prove a more specialized theorem.

First we extend our definitions to the case of modal connectives.

M-Definition 14 (`LM`) *We define a new rule box_rule in the obvious way and define* $\mathcal{L}_{\mathcal{M}}[\Gamma, \Delta]$ *to be* $\mathcal{L}_{\mathcal{A}}[\Gamma, \Delta \cup box\_rule]$.

M-Definition 15 (`TM`) *We get the basic modal theory* $\mathcal{T}_{\mathcal{M}}$*, equivalent to the modal logic* K*, by first defining the class of normal axiom schemas norm_ax, and the necessitation rule (nec), then defining* $\mathcal{T}_{\mathcal{M}}[\Gamma, \Delta]$ *to be* $\mathcal{T}_{\mathcal{A}}[\Gamma \cup norm\_ax \cup \mathcal{T}_{\mathcal{A}}[\emptyset, \emptyset], \Delta \cup nec]$. *Notice that, like in the usual formalizations of modal proof systems, all propositional tautologies are taken as axioms.*

$\mathcal{T}_{\mathcal{M}}$ is the basic modal logic K; in what follows we also consider theories where an axiom called '4' or 'transitivity' is available.

M-Definition 16 (`four_ax`) $x \in four\_ax \Leftrightarrow \exists a (x = \Box a \to \Box\Box a)$

Then if we define $\mathcal{T}_{\mathcal{MF}}[\Gamma, \Delta]$ to be $\mathcal{T}_{\mathcal{M}}[\Gamma \cup four\_ax, \Delta]$ we have

M-Lemma 17 (`four_closed`) *If* A *is boxed (i.e.* $A \equiv \Box A'$*), and* B *follows by (nec) from* B' *(so we must also have* $B \equiv \Box B'$*), then* $\vdash_{\mathcal{MF}} \Box A' \to B'$ *implies* $\vdash_{\mathcal{MF}} \Box A' \to \Box B'$.

This follows by

1.	$\vdash_{\mathcal{MF}} \Box A' \to B'$	
2.	$\vdash_{\mathcal{MF}} \Box(\Box A' \to B')$	`nec` 1
3.	$\vdash_{\mathcal{MF}} \Box(\Box A' \to B') \to \Box\Box A' \to \Box B'$	`norm_ax`
4.	$\vdash_{\mathcal{MF}} \Box\Box A' \to \Box B'$	`mp_arrow` 2,3
5.	$\vdash_{\mathcal{MF}} (\Box A' \to \Box\Box A') \to (\Box\Box A' \to \Box B') \to (\Box A' \to \Box B')$	prop taut
6.	$\vdash_{\mathcal{MF}} \Box A' \to \Box\Box A'$	`four_ax`
7.	$\vdash_{\mathcal{MF}} (\Box\Box A' \to \Box B') \to \Box A' \to \Box B'$	`mp_arrow` 5,6
8.	$\vdash_{\mathcal{MF}} \Box A' \to \Box B'$	`mp_arrow` 7,4

Since this result is open for all axiom and language extensions, `dedthm_abs2` immediately gives us a partial deduction theorem for any modal logic with the 4 axiom.

M-Theorem 18 (`MT4dedthm`)

$$\forall a, b \in \mathcal{L}_{\mathcal{M}}[\Phi, \Psi](((\exists a' \in \mathcal{L}_{\mathcal{M}}[\Phi, \Psi](a = \Box a')) \\ \Rightarrow \vdash_{\mathcal{MF}[\{a\} \cup \Gamma, \emptyset]} b \Rightarrow \vdash_{\mathcal{MF}[\Gamma, \emptyset]} a \to b)) \,.$$

Note that this proof uses results such as `mp_arrow`, which were proven originally for $\mathcal{T}_{\mathcal{A}}$.

This is not the full deduction theorem because it only allows us to construct proofs under boxed assumption; still it is general and holds for all modal logics containing the 4 axiom. In his book on natural deduction, Prawitz presents two natural deduction systems, for $S4$ and $S5$, but no general results on this topic. Both our metatheorem and Prawitz' presentation place side conditions on assumptions stating that they must be boxed; one difference is that Prawitz' conditions are on the treatment of $\Box$ not $\to$. Although it is not immediate, our metatheorem provides a basis for giving a natural deduction presentation of modal logics like $S4$ that is essentially equivalent to Prawitz; see [1, § 4.4] for a more detailed discussion.

4 Comparison and Conclusions

Standard Metatheory and Hierarchical Structuring

We have shown that inductive definitions offer considerable power and simplicity; each metatheorem states in which extensions it is applicable; once proved, it can be directly used in these extensions. Most metatheorems only require that certain axioms and rules are available and therefore hold in all extensions with additional axioms and rules. But others depend on certain things being absent (e.g. the deduction theorem); in such cases, one can prove more restricted theorems that are still usable in appropriate extensions.

How does this kind of metatheory compare with what is possible in theorem provers supporting hierarchical theories? We begin by reviewing the two standard notions of proof rules; our definitions are those of Troelstra [16, §1.11.1] translated into our notation, $\mathcal{T}[\Gamma, \Delta]$ is a theory $\mathcal{T}$ extended with sets of axioms Γ and rules Δ.

Fix a language of formulae. A *rule* is an $n+1$-ary relation over formulae $\langle F_1, \ldots, F_n, F_{n+1}\rangle$ where the $F_1, \ldots, F_n$ are the *premises* and F_{n+1} the *conclusion*. A rule is *admissible for* $\mathcal{T}$ iff

$$\vdash_{\mathcal{T}[\emptyset,\emptyset]} F_1 \Rightarrow \cdots \Rightarrow \vdash_{\mathcal{T}[\emptyset,\emptyset]} F_n \Rightarrow \vdash_{\mathcal{T}[\emptyset,\emptyset]} F_{n+1} \,, \qquad (adm)$$

and it is *derivable for* $\mathcal{T}$ iff

$$\forall \Gamma (\vdash_{\mathcal{T}[\Gamma,\emptyset]} F_1 \Rightarrow \cdots \Rightarrow \vdash_{\mathcal{T}[\Gamma,\emptyset]} F_n \Rightarrow \vdash_{\mathcal{T}[\Gamma,\emptyset]} F_{n+1}) \,. \qquad (der)$$

Derivability implies admissibility, but not vice versa. The rule of cut for first-order logic is the classic example of an admissible but not derived rule (since cut cannot be eliminated from, e.g., sequent-calculus Peano arithmetic). It is easy to show that Troelstra's definition of derivability is equivalent to that used by Hindley and Seldin [9]; i.e. $\vdash_{\mathcal{T}[\{F_1,\ldots,F_n\},\emptyset]} F_{n+1}$, and that if a rule is derivable it holds in all extensions of $\mathcal{T}$ with new axioms and rules.

Our use of formal metatheory allows us to formalize both admissible and derived rules, as well as those not fitting the above schema. For example, the languages or proof systems for the F_i can be different. The versions of the deduction theorem that we formalized exemplify this; they are neither derived nor admissible since their statements involve different proof systems.

Standard hierarchical structuring of theories allows the derivation of a subset of the derived rules. For example, to prove a theorem in Isabelle, one begins by stating in which theory it is to be proved. The theorem may then be proved using any axioms, rules, or derived rules for that theory. All theorems proved hold in any extension with new axioms or rules. Not all rules belonging to the class of derived rules can be so formally derived. In particular, one cannot derive an instance of (der) that requires induction on the *language* of the theory, even though such a theorem would hold in all extensions of the *proof* system. But (because of this) theorems proved hold not only in extensions of the proof system, but in extensions of the language as well.

Other Approaches

We briefly survey some alternative approaches. Type theories support an approach to structuring theories (which may include induction principles) by formalizing them as Sigma types [10, 13]. For instance, these types can be used to formalize the languages and proof systems of logics, and after one can prove inclusions and the existence of other kinds of relations between such types and their members. Examples of this approach, which is extremely powerful, are provided by Basin and Constable [2]. However the user is obliged, with each new theory declaration, to prove new metatheorems relating it to previously constructed theories.

The IMPS prover also provides support for the formalization of individual 'little theories' and interpretations between them [4]. Theory interpretation allows us to apply, for example, abstract theorems of group theory to a more concrete mathematical structure like the real numbers, after showing that multiplication over nonzero elements forms a group. In many respects, theory interpretation is more general than the kind of metatheory we have explored (we have not considered translations), but, like with approaches using Sigma types, one must explicitly construct (perhaps with help from the system) an appropriate interpretation from one theory into another.

The approach we have taken based on inductive definitions has been implemented within FS_0, but other logics supporting inductive definitions (e.g. HOL, type theory) could provide alternative candidates. However, FS_0 has been proposed and shown itself useful as a general metalogic and its simplicity greatly aided the development carried out here. Note too, that the inductive definitions we required often depended on parameters given within the statement of metatheorems; hence one cannot directly use the kind of support packages for inductive definitions provided with systems like HOL and Isabelle. One would have to work directly with fixedpoint definitions (or inductive types, in the case of type theory), and additional support would be required to use these definitions in practice.

Practical Experience and Further Work

We have presented an approach to machine checked metatheory based on inductive definitions. Our practical experience with an implementation of FS_0 in Isabelle was very positive. We were able, in our implementation (which provides considerably more support than the minimal theory provided by Feferman) to verify formally the results documented here within a few days. Further, apart from a few technical idiosyncrasies of FS_0, the proofs on the machine closely follow the 'natural' informal proofs we would produce on paper.

The work we have described here concerns the metatheory of logics that extend ordinary minimal implicative propositional logic, i.e. classical, intuitionistic and modal. We start with a basic theory of $\rightarrow$ and prove generalized metatheorems for it, specifically the deduction theorem. We could formalize Hilbert presentations of even weaker, substructural, versions of $\rightarrow$, e.g. relevant or linear logic; however, because FS_0 itself is based on a stronger logic (i.e. classical logic), it is not possible to 'mingle' metatheory and object theory for such encodings in the way we have done above for the deduction theorem. Došen [3] has proposed a possible solution to this: conservatively extending such logics with a special modal connective allowing, e.g., a modified deduction theorem (essentially a generalized version of `MT4dedthm`). We are currently exploring this.

References

[1] A. Avron, F. Honsell, I. Mason, and R. Pollack. Using typed lambda calculus to implement formal systems on a machine. *J. Automated Reasoning*, 9:309–352, 1992.

[2] D. Basin and R. Constable. Metalogical frameworks. In G. Huet and G. Plotkin, editors, *Logical Frameworks*. CUP, Cambridge, 1991.

[3] K. Došen. Modal logic as metalogic. *J. Logic, Language and Information*, 1:173–201, 1992.

[4] W. M. Farmer, J. D. Guttman, and F. J. Thayer. Little theories. In D. Kapur, editor, *Proc. CADE-11*. Springer, Berlin, 1992.

[5] S. Feferman. Finitary inductive systems. In *Logic Colloquium '88*. North-Holland, Amsterdam, 1990.

[6] M. Gordon and T. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic.* CUP, Cambridge, 1993.

[7] M. J. Gordon, R. Milner, and C. P. Wadsworth. *Edinburgh LCF: A Mechanized Logic of Computation.* Springer, Berlin, 1979.

[8] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *J. ACM*, 40:143–184, 1993.

[9] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and λ-Calculus.* CUP, Cambridge, 1986.

[10] D. B. MacQueen. Using dependent types to express modular structure. In *Proc. 13th ACM Symp. Principles Programming Languages.* ACM Press, New York, 1986.

[11] S. Matthews. A general binding facility in FS_0. Available from `http://www.mpi-sb.mpg.de/~sean/`.

[12] S. Matthews. Implementing FS_0 in Isabelle: adding structure at the metalevel. Available from `http://www.mpi-sb.mpg.de/~sean/`, 1995.

[13] J. C. Mitchell and G. Plotkin. Abstract types have existential type. In *Proc. 12th ACM Symp. Principles Programming Languages.* ACM Press, New York, 1985.

[14] L. C. Paulson. *Isabelle: A generic theorem prover.* Springer, Berlin, 1994.

[15] D. Prawitz. *Natural Deduction; A proof theoretic study.* Almqvist and Wiksell, Stockholm, 1965.

[16] A. S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis.* Springer, Berlin, 1982.