Structuring Metatheory on Inductive Definitions

David Basin
Institut für Informatik
Universität Freiburg
Am Flughafen 17, D-79110,
Freiburg, Germany

Seán Matthews
Max-Planck-Institut für Informatik
Im Stadtwald, D-66123
Saarbrücken, Germany

Abstract

We examine a problem for machine supported metatheory. There are statements true about a theory that are true of some (but only some) extensions; however standard theory-structuring facilities do not support selective inheritance. We use the example of the deduction theorem for modal logic and show how a statement about a theory can explicitly formalize the closure conditions extensions should satisfy for it to remain true. We show how metatheories based on inductive definitions allow theories and general metatheorems to be organized this way, and report on a case study using the theory FS_0.

1 Introduction

Hierarchical 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 [13] and their predecessor LCF [7] support simple theory hierarchies. In these systems a theory is a specification of a language, using types and typed constants, and a collection of rules and axioms which are used to prove theorems; then a theory inherits the types, constants, axioms, rules and theorems from its ancestors. The drawback to this is that types associated with the language and proof system must be considered 'open', and this limits the sort of proofs that can be constructed. In particular, it rules out arguments requiring induction on the structure of the language or proofs of a theory. We can 'close' the language or proof system by explicitly adding induction over the structure of the language or proof system, but then it is unsound later to add new constructors, axioms, or rules; i.e. extensions to the theory are impossible.

We investigate this problem, and show that the simple hierarchical approach described above, which in effect offers an 'all or nothing' approach to inheriting results, does not allow a fine enough control over the scope of theorems and is more inflexible than need be. Our solution uses a theory of parameterized inductive definitions as a metatheory in which we can organize collections of *object theories* and *logics*. This allows the same sort of hierarchical structuring as we have described above, but also allows us to state and prove more general metatheoretic results in a way such that they can be inherited by extensions, or *supertheories*, in the hierarchy: instead of closing the theory for which we prove a result, we formalize, as part of the metatheorem itself, the closure properties we need. This means that a theory for which we prove metatheorems can still be extended, and metatheorems can be imported into the supertheory simply by checking that the closure conditions are satisfied.

Our approach makes few special demands of the metatheory: it requires only that we are able to build families of sets using parameterized inductive definitions, and can reason about their elements by induction. We report on an implementation of our ideas in FS_0, a framework theory for formal metatheory due to Feferman [5], which satisfies these requirements. We have used FS_0 because it was designed to provide a simple, minimal theory of inductive definitions specifically for the purposes of syntactic metatheory; if we can use inductive definitions to structure theory development in FS_0, then similar results should be possible with stronger metatheories such as type theory or formalizations of induction definitions in higher-order logic or set-theory (see §4.2 for more on this). The definitions and proofs presented here have all been machine checked in our implementation of FS_0.

1.1 Motivation

The example from formal metatheory we consider is fundamental for hierarchies of Hilbert systems. Hilbert systems are interesting because they are commonly acknowledged to be the easiest kind of proof system about which to prove theorems, but also the hardest in which to prove theorems. In practice, to prove theorems with them we need a metatheorem, the deduction theorem, which allows, essentially, 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 facilities needed.

At the root of our hierarchy is what we call *arrow logic* (minimal implicative propositional logic), the formulae of which correspond to members of the set $\mathsf{L}_{\mathcal{A}}$ of formulae built from the binary connective $\rightarrow$ (written infix, associating to the right) and sentential constants. Theorems correspond to members of a second set, $\mathsf{T}_{\mathcal{A}}$, and are instances of the standard Hilbert axiom schemata

$$A \to B \to A \tag{K}$$

and

$$(A \to B) \to (A \to B \to C) \to (A \to C) \tag{S}$$

or are generated by applying the rule *modus ponens*:

$$\frac{A \quad A \to B}{B\,.}$$

We write $\vdash_{\mathcal{A}} A$ to indicate that $A \in \mathsf{T}_{\mathcal{A}}$. Then, assuming our (not yet formally specified) metatheory contains at least the 'implies' connective $\Rightarrow$, two *meta*theorems we can prove are:

$$\vdash_{\mathcal{A}} A \to A \tag{I}$$

$$(\vdash_{\mathcal{A}} A \to C) \Rightarrow (\vdash_{\mathcal{A}} A \to B \to C)\,. \tag{$Thin$}$$

The first of these asserts the provability of a formula in the object logic and the second represents a derived proof rule. Given universal quantification over $\mathsf{L}_{\mathcal{A}}$ in the metatheory, we could make both of the above formulae schematic by taking their universal closures. (In the future we will assume this reading, leaving the quantifiers implicit.)

Now consider the deduction theorem. For this we need to extend our metalanguage: we define $A \vdash_{\mathcal{A}} B$ as metanotation meaning that if the Hilbert system $\mathsf{T}_{\mathcal{A}}$ is extended with the additional axiom A, then B belongs to the resulting set of proofs.[1] The deduction theorem is, then,

$$A \vdash_{\mathcal{A}} B \Rightarrow \vdash_{\mathcal{A}} A \to B\,, \tag{DT}$$

which is proven by induction on the structure of derivations. Notice that just to state this metatheorem, which is schematic in A (and B), we refer to infinitely many axiomatic extensions of the original Hilbert system.

An obvious candidate for a supertheory of arrow logic is full propositional logic (we consider here only conjunction, other operators are similar): we extend $\mathsf{L}_{\mathcal{A}}$ to $\mathsf{L}_{\mathcal{P}}$ by adding the new binary connective $\wedge$, and extend $\mathsf{T}_{\mathcal{A}}$ to $\mathsf{T}_{\mathcal{P}}$ by adding the three new axiom schemata:

$$\begin{array}{c} A \to B \to A \wedge B \\ A \wedge B \to A \\ A \wedge B \to B \end{array} \tag{$Conj$}$$

What is the status of the theorems and metatheorems of arrow logic in propositional logic? We expect that all theorems, e.g. (I), are still theorems of propositional logic, provided that the original axioms (S) and (K) and inference

[1] Note that this is metanotation, since $\mathsf{L}_{\mathcal{A}}$ is a language of *formulae*, not sequents.

rule (*modus ponens*) are interpreted over the new language. The rule (*Thin*) clearly also holds. The status of the deduction theorem is less obvious. In fact it does hold for propositional logic, but since the proof suggested above is by induction on the structure of arrow logic proofs, it is not valid in this new context; we must check the new cases corresponding to the new axioms. If we used a metatheory based on a logic like higher-order logic and formulated these two proof systems as standard inductive definitions, we would need to prove the deduction theorem individually for each system.

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

$$\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)\,, \tag{\textit{Norm}}$$

and a rule of inference *necessitation*:

$$\frac{A}{\Box A}\,. \tag{\textit{Nec}}$$

Again we can ask, what is the status of the theorems and metatheorems of arrow logic in $\mathsf{T}_{\mathcal{M}}$? Both (I) and (*Thin*) hold in any extension of $\mathsf{T}_{\mathcal{A}}$ with new axioms. They even hold when $\mathsf{T}_{\mathcal{A}}$ is extended with new rules and also when the language of $\mathsf{L}_{\mathcal{A}}$ is extended to $\mathsf{L}_{\mathcal{M}}$, provided the old axioms and inference rule *modus ponens* are interpreted over the new language. The deduction theorem (DT) also holds under extensions of $\mathsf{T}_{\mathcal{A}}$ with new axioms. But it does not hold under extensions with arbitrary new rules; in particular, it fails for $\mathsf{T}_{\mathcal{M}}$ 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. We show that this is possible using parameterized inductive definitions. Informally, an inductive definition of a set is presented as the closure of some base set under a collection of rules that generate new elements from members of the set. Formally, we will explain and use notation for inductive definitions based on their presentation in FS_0. However, ignoring minor differences, other formalizations are fairly similar (an example is provided in §4.2).

In 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 $\mathsf{L}_{\mathcal{A}}$ the set $base_{\mathsf{L}_{\mathcal{A}}}$ defines the sentential constants while $step_{\mathsf{L}_{\mathcal{A}}}$ explains how two formulae in $\mathsf{L}_{\mathcal{A}}$ can be combined under a connective $\rightarrow$. For $\mathsf{T}_{\mathcal{A}}$ the set $base_{\mathsf{T}_{\mathcal{A}}}$ defines all instances of the axioms (S) and (K), while $step_{\mathsf{T}_{\mathcal{A}}}$ corresponds to *modus ponens*.

Theory structuring is supported by our use of parameterized inductive definitions, which are definitions where the base and step sets are augmented with parameters specifying concrete or hypothetical extensions. For example, let $\vdash_{\mathcal{A}[\Gamma,\Delta]} A$ denote that the formula A belongs to the set $I(base_{\mathsf{T}_{\mathcal{A}}} \cup \Gamma, step_{\mathsf{T}_{\mathcal{A}}} \cup \Delta)$. Here, the axioms of $\mathsf{T}_{\mathcal{A}}$ are extended by the set of axioms Γ, and the rules of $\mathsf{T}_{\mathcal{A}}$ are extended by the set of rules Δ. The deduction theorem can now be formulated as

$$\vdash_{\mathcal{A}[\{A\},\emptyset]} B \Rightarrow \vdash_{\mathcal{A}[\emptyset,\emptyset]} A \to B \,. \tag{DT_2}$$

This metatheorem relates the provability of $A \to B$ in the theory $\mathsf{T}_{\mathcal{A}}$ to the provability of B in the concrete extension of $\mathsf{T}_{\mathcal{A}}$ with the axiom A. In general, the provisos on, and scope of, metatheorems can be formalized in terms of constraints on parameter sets, 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. If Γ and Δ are empty, then we get the metatheorem for the original theory. Thus, for instance, (I) and (Thin) hold for extensions of the proof system $base_{\mathsf{T}_{\mathcal{A}}}$ and $step_{\mathsf{T}_{\mathcal{A}}}$ with arbitrary Γ and Δ, whereas we shall see that the deduction theorem holds only for arbitrary Γ.

1.2 Organization

In §2 we provide more details about inductive definitions in FS_0 and show in §3 how we use them to implement the kind of metatheoretic development described above, where metatheorems and their scope can be appropriately parameterized. In §4 we report on practical experience using FS_0 and the possibility of using alternative metatheories such as inductive definitions within higher-order logic. Finally, in §5 we relate our approach to formalizing metatheorems to standard metatheory where rules are sometimes classified as *admissible* or *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 the HOL system where theories are hierarchically structured. We also make comparisons with other approaches to structuring theories, that are not based on inductive definitions.

2 Inductive definitions and the theory FS_0

We use Feferman's FS_0 for our work since it is a particularly simple minimal theory of inductive definitions. Moreover, we have built an implementation for it in which we have tested our ideas. However since we do not give concrete FS_0 proofs in this paper and because our approach is not FS_0 specific, we review only those aspects that are relevant to this paper and are necessary to provide a rigorous account of our approach. A full description of FS_0 and our implementation can be found in [5] and [10].

FS_0 is a first-order theory of inductively defined sets. Elements of sets are S-expressions, which are terms generated from the constant *nil* and the pairing operation *cons*, usually written simply as a pair $(\cdot,\cdot)$, where the comma associates to the right; i.e. (a,b,c) is short for $(a,(b,c))$. Sets of S-expressions can be defined by induction: 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 $(t, t_1, t_2) \in b$. Note that we are allowed only inductive definitions with exactly two 'predecessors', but this is sufficient for the needs of this paper (in fact it is enough for general finitary definitions).

We reason about inductively defined sets using the induction principle

$$\begin{aligned} & \mathit{base} \subset s \Rightarrow \forall a, b, c.\, (b, c \in s \Rightarrow (a, b, c) \in \mathit{step} \\ & \qquad\qquad\qquad\qquad\qquad\qquad \Rightarrow a \in s) \Rightarrow I(\mathit{base}, \mathit{step}) \subset s\,. \end{aligned}$$

Hence a predicate with extension s holds of a set $I(\mathit{base}, \mathit{step})$ if the predicate holds of all elements in *base* and whenever it holds for two elements b and c, and (a, b, c) is an instance of the rule *step*, then it also holds for a, i.e. the predicate is invariant for rule application.

To 'get off the ground', FS_0 provides ways of defining non-inductive sets; it also allows comprehension over a large class of first-order (essentially Σ^0_1) predicates. We write

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

to indicate a set defined this way. Such definitions can be parameterized and the parameters are treated in a simple way:

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

defines a set s parameterized over an element a and a set b.

All definitions and theorems given in this paper have been checked using our implementation of FS_0. For the sake of readability, however, we use standard mathematical syntax (instead of the concrete ASCII syntax supported by our implementation) in our presentation here.

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 the scope of a metatheorem 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.

3.1 Arrow logic

We begin by defining the language $\mathsf{L}_{\mathcal{A}}$ and proof system $\mathsf{T}_{\mathcal{A}}$. $\mathsf{L}_{\mathcal{A}}$ is the smallest set built from some atomic set of atoms and closed under the binary connective $\rightarrow$.

Definition 1 ($\mathsf{L}_{\mathcal{A}}$) *First we define a constant '$\rightarrow$', which we use to mark syntactic objects. Then we define*

$$x \in \mathit{imp\_rule} \Leftrightarrow \exists a, b.\, x = (a \rightarrow b, a, b)$$

at which point we could define $\mathsf{L}_{\mathcal{A}} \equiv I(P, \mathit{imp\_rule})$*, 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:*

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

The language $\mathsf{L}_{\mathcal{A}}$ is then simply $\mathsf{L}_{\mathcal{A}}[P, \emptyset]$, where P is some class of atomic propositions, and we can show, e.g., that $a \rightarrow a \in \mathsf{L}_{\mathcal{A}}[P, \Delta]$ given that $a \in P$. Note that $\rightarrow$ should not be confused with implication in the metalogic, $\Rightarrow$. Also, we will write both implications with formulae associating to the right.

In the same way, we can declare the family of proof systems $\mathsf{T}_{\mathcal{A}}$:

Definition 2 ($\mathsf{T}_{\mathcal{A}}$) *Let*

$$\begin{aligned} x \in K &\Leftrightarrow \exists a, b \in \mathsf{L}_{\mathcal{A}}.\, x = a \rightarrow b \rightarrow a \\ x \in S &\Leftrightarrow \exists a, b, c \in \mathsf{L}_{\mathcal{A}}.\, x = (a \rightarrow b) \rightarrow (a \rightarrow b \rightarrow c) \rightarrow a \rightarrow c \\ x \in mp &\Leftrightarrow \exists a, b \in \mathsf{L}_{\mathcal{A}}.\, x = (b, a, a \rightarrow b) \end{aligned}$$

so that

$$\mathsf{T}_{\mathcal{A}}[\Gamma, \Delta] \equiv I(\Gamma \cup S \cup K, \Delta \cup mp)$$

Below we use the following notation for $\mathsf{T}_{\mathcal{A}}$ and other theories we introduce: $\mathsf{T}_{\mathcal{A}}$ (without parameters) is $\mathsf{T}_{\mathcal{A}}[\emptyset, \emptyset]$, $\vdash_{\mathcal{A}} a$ is $a \in \mathsf{T}_{\mathcal{A}}$, and $\vdash_{\mathcal{A}[\Gamma,\Delta]} a$ is $a \in \mathsf{T}_{\mathcal{A}}[\Gamma, \Delta]$.

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

Lemma 3

$$\forall a, b \in \mathsf{L}_{\mathcal{A}}.\, \vdash_{\mathcal{A}} a \Rightarrow \vdash_{\mathcal{A}} a \rightarrow b \Rightarrow \vdash_{\mathcal{A}} b \tag{$MP_{\mathcal{A}}$}$$

$$\forall a, b, c \in \mathsf{L}_{\mathcal{A}}.\ \vdash_{\mathcal{A}} (a \rightarrow b) \rightarrow (a \rightarrow b \rightarrow c) \rightarrow a \rightarrow c \tag{$S_{\mathcal{A}}$}$$

$$\forall a, b \in \mathsf{L}_{\mathcal{A}}.\ \vdash_{\mathcal{A}} a \rightarrow b \rightarrow a \tag{$K_{\mathcal{A}}$}$$

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

Theorem 4 ($I_{\mathcal{A}_1}$) $\forall a \in \mathsf{L}_{\mathcal{A}}.\ \vdash_{\mathcal{A}} a \rightarrow a\,.$

We prove this (on the machine) using the following steps:

1.	$\vdash_{\mathcal{A}} (a \to b \to a) \to (a \to (b \to a) \to a) \to a \to a$	$S_{\mathcal{A}}$
2.	$\vdash_{\mathcal{A}} a \to b \to a$	$K_{\mathcal{A}}$
3.	$\vdash_{\mathcal{A}} a \to (b \to a) \to a$	$K_{\mathcal{A}}$
4.	$\vdash_{\mathcal{A}} (a \to (b \to a) \to a) \to a \to a$	$MP_{\mathcal{A}}$ 1,2
5.	$\vdash_{\mathcal{A}} a \to a$	$MP_{\mathcal{A}}$ 3,4

This metatheorem has been shown just for $\mathsf{T}_{\mathcal{A}}$. However, if we examine the proof, we see that it made use only of $S_{\mathcal{A}}$, $K_{\mathcal{A}}$ and $MP_{\mathcal{A}}$, and that these hold for arbitrary $\mathsf{T}_{\mathcal{A}}[\Gamma, \Delta]$, not just for the particular instance $\mathsf{T}_{\mathcal{A}}$, thus the result may be parameterized, in an openended way, over all possible extensions to the axioms and rules of $\mathsf{T}_{\mathcal{A}}$.

Theorem 5 $(I_{\mathcal{A}_2})$ $\forall a \in \mathsf{L}_{\mathcal{A}}. \vdash_{\mathcal{A}[\Gamma,\Delta]} a \to a$.

This new metatheorem is parameterized with respect to the proof system: it states that $a \to a$ holds in any extension of the Hilbert system represented by $\mathsf{T}_{\mathcal{A}}$ with new axioms Γ and rules Δ. But in fact we can generalize even further: the proof only requires implication and sentential constants in our language, but there could be other connectives. The most general form then is parameterized both with respect to the language and the proof system

Theorem 6 $(I_{\mathcal{A}+})$ $\forall a \in \mathsf{L}_{\mathcal{A}}[\Phi, \Psi].\ \vdash_{\mathcal{A}[\Gamma,\Delta]} a \to a$.

We then, of course, should also go on to prove similarly parameterized versions of our other basic metatheorems, which we can do in the same way:

Theorem 7

$$\forall a, b \in \mathsf{L}_{\mathcal{A}}[\Phi, \Psi]. \vdash_{\mathcal{A}[\Gamma,\Delta]} a \Rightarrow \vdash_{\mathcal{A}[\Gamma,\Delta]} a \to b \Rightarrow \vdash_{\mathcal{A}[\Gamma,\Delta]} b \qquad (MP_{\mathcal{A}+})$$
$$\forall a, b, c \in \mathsf{L}_{\mathcal{A}}[\Phi, \Psi].\ \vdash_{\mathcal{A}[\Gamma,\Delta]} (a \to b) \to (a \to b \to c) \to a \to c \qquad (S_{\mathcal{A}+})$$
$$\forall a, b \in \mathsf{L}_{\mathcal{A}}[\Phi, \Psi].\ \vdash_{\mathcal{A}[\Gamma,\Delta]} a \to b \to a \qquad (K_{\mathcal{A}+})$$
$$\forall a, b, c \in \mathsf{L}_{\mathcal{A}}[\Phi, \Psi]. \vdash_{\mathcal{A}[\Gamma,\Delta]} a \to c \Rightarrow \vdash_{\mathcal{A}[\Gamma,\Delta]}. a \to b \to c \qquad (\mathit{Thin}_{\mathcal{A}+})$$

These metatheorems are all fully openended: Due to the use of parameterization, they can be used in any extension both of the language and the proof system. This makes them 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. In §5 we will classify these and several other kinds of rules.

3.2 The deduction theorem

We cannot prove fully parameterized versions of all metatheorems. An example of a metatheorem for Hilbert systems (in fact, as we have said, one of the most important) that does not hold in all theory extensions is the deduction theorem. But when this metatheorem does hold, its application allows trivial proofs of theorems that are otherwise tricky to prove; e.g. $I_{\mathcal{A}_1}$, which we could reduce to $a \vdash_{\mathcal{A}} a$, i.e. $\vdash_{\mathcal{A}[\{a\},\emptyset]} a$, which follows immediately.

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

- 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 $A \to B$.
- B is A. Then we need to show $A \to A$, which we can do, following the proof of $I_{\mathcal{A}_1}$ above.

In the step case B follows by an application of *modus ponens* to 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 $(A \to C) \to (A \to C \to B) \to A \to B$. Thus by *modus ponens* twice we have $A \to B$.

We can formalize the deduction theorem in a form where we can prove it using the induction available in FS_0, as

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

This is the 'minimal', closed form of the theorem. An openended version would be much more useful, but the theorem does not hold in all extensions; the class of propositional modal logics, for example, provides a counter example; i.e.

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

Although a completely parameterized version of the deduction theorem is not provable, if we look closely at the proof sketch, we can see that a limited form may be available. We can see that we need

- $\to$ in the language $\mathsf{L}_{\mathcal{A}}$, but there could be other connectives.
- the two axioms (S) and (K), but there could be other axioms.
- *exactly* the rule *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.

Thus full extensibility over the language is possible, but only restricted extensibility, over the *base*, over the theory (i.e. no new rules); so the result we have verified can be parameterized over Φ, Ψ and Γ, to get

Theorem 8 (DT_0) $\forall a, b \in \mathsf{L}_{\mathcal{A}}[\Phi, \Psi].\, \vdash_{\mathcal{A}[\{a\} \cup \Gamma,\emptyset]} b \Rightarrow \vdash_{\mathcal{A}[\Gamma,\emptyset]} a \to b\,.$

This version can then be applied to infinitely many extensions of arrow logic. For example, we can extend $\mathsf{L}_{\mathcal{A}}$ to $\mathsf{L}_{\mathcal{P}}$ by adding conjunction and extending $\mathsf{T}_{\mathcal{A}}$ to $\mathsf{T}_{\mathcal{P}}$ by adding the three axioms given in (*Conj*). Since these are *axioms*, not *rules*, we can directly apply DT_0 to prove theorems in this extended language and proof system, for instance to provide a second, simpler, proof that $\forall a \in \mathsf{L}_{\mathcal{P}}.\, \vdash_{\mathcal{P}} a \to a$.

3.3 Rule extensions of the deduction theorem

The above metatheorem, DT_0, has been generalized to apply to all extensions of the language and to extensions with axioms of the proof system of $\mathsf{L}_{\mathcal{A}}$. 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 the 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*, which, to remind, is the rule

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

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

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

which is precisely the (S) axiom schema.

Now consider the more general case of a rule with two premises (this can be generalized easily 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 $A \to D$.

We formalize this observation to get a more general form of DT_0. We do not, however, insist that certain axioms like (S) be in the theory whenever certain rules are; instead we directly formulate the provability requirement above. Thus we have

Theorem 9 **(DT_1)** *If we define* L *to be* $\mathsf{L}_{\mathcal{A}}[\Phi, \Psi]$, $\vdash$ *to be* $\vdash_{\mathcal{A}[\Gamma,\Delta]}$ *and* $\vdash_{\{a\}}$ *to be* $\vdash_{\mathcal{A}[\Gamma \cup \{a\},\Delta]}$, *then*

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

This metatheorem states general conditions that are sufficient for the deduction theorem to hold. The following section shows, however, that even this is not the most general version possible.

3.4 Modal logic

In many logics that are simple extensions of arrow logic, the deduction theorem fails, and, as is well known, 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:

Theorem 10 (DT_2) *If we define* L *to be* $\mathsf{L}_{\mathcal{A}}[\Phi, \Psi]$, $\vdash$ *to be* $\vdash_{\mathcal{A}[\Gamma,\Delta]}$ *and* $\vdash_{\{a\}}$ *to be* $\vdash_{\mathcal{A}[\Gamma\cup\{a\},\Delta]}$*, then*

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

This generalizes DT_1, where the extra set P can be used to place restrictions on the possible assumptions (setting P to the universal set provides DT_1 as a special case). Note that this result says nothing about modal logic; it is a parameterized 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 $\mathsf{L}_{\mathcal{M}}$ by adding the unary modality $\Box$, and extending a propositional proof system to $\mathsf{T}_{\mathcal{M}}$ by adding the axiom $(Norm)$ and the necessitation rule (Nec). Note that the necessitation rule cannot be replaced by an axiom schema.

Definition 11 ($\mathsf{L}_{\mathcal{M}}$) *We define a new rule, box_rule, in the obvious way and define* $\mathsf{L}_{\mathcal{M}}[\Gamma, \Delta]$ *to be* $\mathsf{L}_{\mathcal{A}}[\Gamma, \Delta \cup \mathit{box\_rule}]$.

Definition 12 ($\mathsf{T}_{\mathcal{M}}$) *We define the basic modal theory* $\mathsf{T}_{\mathcal{M}}$ *by first defining the class of normal axiom schemata norm_ax, and the necessitation rule* (Nec)*, then defining* $\mathsf{T}_{\mathcal{M}}[\Gamma, \Delta]$ *to be* $\mathsf{T}_{\mathcal{A}}[\Gamma \cup \mathit{norm\_ax} \cup \mathsf{T}_{\mathcal{A}}[\emptyset, \emptyset], \Delta \cup \mathit{Nec}]$.

Notice that, like in the usual formalizations of modal proof systems, all propositional tautologies are taken as axioms.

$\mathsf{T}_{\mathcal{M}}$ is the basic modal logic K. Our parameterized deduction theorem fails for this logic. Let us return to our proof sketch above. We have a new step case for (Nec). The formula B $(= \Box B')$ is the result of an application of (Nec) to 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 in theories where an axiom called '4' or 'transitivity' is available.

Definition 13 ($\mathit{four\_ax}$) $x \in \mathit{four\_ax} \Leftrightarrow \exists a.\, x = \Box a \to \Box\Box a$

Then, if we define $\mathsf{T}_{\mathcal{M}\mathcal{F}}[\Gamma, \Delta]$ to be $\mathsf{T}_{\mathcal{M}}[\Gamma \cup \mathit{four\_ax}, \Delta]$ we can prove

Lemma 14 ($\mathit{Closed\_Lem}$) *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* $\vdash_{\mathcal{M}\mathcal{F}} \Box A' \to B'$ *implies* $\vdash_{\mathcal{M}\mathcal{F}} \Box A' \to \Box B'$.

This theorem 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.	$\vdash_{\mathcal{MF}} \Box\Box A' \to \Box B'$	$MP_{\mathcal{A}+}$ 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.	$\vdash_{\mathcal{MF}} (\Box\Box A' \to \Box B') \to \Box A' \to \Box B'$	$MP_{\mathcal{A}+}$ 5,6
8.	$\vdash_{\mathcal{MF}} \Box A' \to \Box B'$	$MP_{\mathcal{A}+}$ 7,4

Since this result is parameterized for all axiom and language extensions, DT_2 immediately gives us a partial deduction theorem for any modal logic with the transitivity axiom.

Theorem 15 ($DedThm_{\mathcal{MF}}$)

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

Note that this proof uses results such as $MP_{\mathcal{A}+}$, which we proved (and machine checked) originally for arbitrary $\mathsf{T}_{\mathcal{A}+}$.

This is not the full deduction theorem: we can use it only to construct proofs under *boxed* assumptions; still it is general and holds for all modal logics containing the transitivity axiom. In his book on natural deduction, Prawitz presents two natural deduction systems, for $S4$ and $S5$, but no general results. 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 of this.

4 Practical experience

4.1 Experience with FS_0

We have presented an approach to machine checked metatheory based on inductive definitions. Our practical experience using this approach to develop metatheory has been very positive. Using our implementation of FS_0 in Isabelle (which provides more support than the minimal theory provided by Feferman) we were able to verify formally the results documented here within a few days. 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.

To give the reader a feel for this, we present a small part of our development.[2] Consider the parameterized version of the deduction theorem proven for arrow logic in §3.3. This was proven for a theory $\mathsf{T}_{\mathcal{A}}$ extended by an arbitrary set of axioms Γ and rules Δ provided that for each rule an appropriate closure condition holds. In FS_0 we begin by declaring the theory $\mathsf{T}_{\mathcal{A}}[\Gamma, \Delta]$ (i.e. `TA`) as we gave it in §3.1.

```
comp("S_ax", "(a->b->c) -> (a->b) -> a->c")
comp("K_ax", "a->b->a")
comp("MP", "(b,a,a->b)")
def("TA",  ["A","R"], "I(A Un S_ax Un K_ax, R Un MP)")
```

We shall not explain all of the syntactic details, but the idea should be clear: the above formalizes the two axioms S and K and the rule for *modus ponens*. The infix function symbol `Un` stands for union, so `TA` is the theory $\mathsf{T}_{\mathcal{A}}[\Gamma, \Delta]$ where `A` and `R` represent the parameters Γ and Δ respectively.

The statement of Theorem DT_1 in Isabelle is then:

```
[| !! a b c d.([| (d,b,c):R; a->b:TA(A,R);  a->c:TA(A,R)|]
             ==> a->d:TA(A,R));
   b:TA(A Un {a},R) |]
  ==> a->b:TA(A,R)
```

Note that `!!` is universal quantification and `==>` is implication in the meta-logic; furthermore `[| A1;...;An |] ==> A` represents the nested implication `A1 ==> ...=> An ==> A`. This metatheorem states that we must show that $a \to b$ is a theorem in the theory $\mathsf{T}_{\mathcal{A}}[A, R]$ (fourth line) when b is a theorem in the theory $\mathsf{T}_{\mathcal{A}}[A \cup \{a\}, R]$ (third line) and the appropriate closure condition holds (first and second line). This closure condition, as we have seen, is that when $\frac{b \quad c}{d}$ is a rule in R then from the provability of $a \to b$ and $a \to c$ it follows that $a \to d$ is provable.

The proof follows in five interactive steps, following the informal sketch. For example, after induction we must show a step case consisting of:

```
[| a->b:TA(A, R); a->c:TA(A, R); (d,b,c):R Un MP |]
      ==> a->d:TA(A, R)
```

We proceed by 'unrolling' the definition of set union, `Un`, which gives us two goals, depending on whether the rule $\frac{b \quad c}{d}$ is an instance of R or *modus ponens*. This yields two subgoals

```
1. [| a->b:TA(A, R); a->c:TA(A, R); (d,b,c):R |]
      ==> a->d:TA(A, R)
2. [| a->b:TA(A, R); a->c:TA(A, R); (d,b,c):MP |]
      ==> a->d:TA(A, R)
```

and the first is proved using the closure condition associated with R and the second, as is standard, using the axioms S.

[2]Apart from some minor tidying, text in `typewriter` font comes directly from a checked proof script.

4.2 Other metatheories supporting inductive definitions

We have indicated how parameterized inductive definitions in FS_0 support structured metatheoretic development. Our approach, based on parameterizing possible extensions is also possible in other metatheories that support similar kinds of inductive definitions. Possible alternative candidates include constructive type theories [3], set-theory [14] or higher-order logic [6].

We have also carried out a development similar to the one sketched in this paper using a theory of inductive definitions based on the Knaster-Tarski Theorem in Isabelle's theory of higher-order logic [12]. The differences were minor.

What follows is a snapshot from this development illustrating some of the close similarities. We can declare in Isabelle the theory $\mathsf{T}_{\mathcal{A}}[\Gamma, \Delta]$ as an inductively defined set `TA` parameterized by `A` and `R`.

```
Hilbert =
TA :: "['a pl set, ('a pl * 'a pl) => 'a pl] => 'a pl set"
inductive "TA A R"
intrs
  A "p:A ==> p:TA A R"
  R "[| a:TA A R; b:TA A R |] ==> R(a,b):TA A R"
  K "a->b->a:TA A R"
  S "(a->b->c) -> (a->b) -> a->c:TA A R"
  MP "[| a->b:TA A R; a:TA A R |] ==> b:TA A R"
```

The declaration declares the type of `TA`. It is a function that returns a set of propositions (over some type `'a`) given two arguments; the first is a set of propositions (the additional axioms), and the second is a function from pairs of propositions to a proposition (the additional rule instances). Following this are the cases of the inductive definition that explain how to generate members of the set `TA A R`. In this context, the statement of the parameterized deduction theorem is as follows.

```
[| !!a b c. [| a->b:TA A R; a->c:TA A R |]
            ==>  a->R(b,c):TA A R;
   b:TA (A Un {a}) R |]
  ==> a->b:TA A R
```

Although higher-order logic and its conservative extension with a theory of inductively defined sets is very different from FS_0 (which is a conservative extension of primitive recursive arithmetic), this example indicates that the details of the formalization are similar. Moreover, so are the proofs themselves, since reasoning in both cases is abstracted away from the underlying metatheory.

5 Comparison and conclusions

5.1 Standard metatheory and hierarchical structuring

We have shown that parameterized inductive definitions offer considerable power and simplicity in organizing metatheories. Each metatheorem states the conditions an extension has to satisfy for it to apply, so once proved, we need only check these conditions before making use of it. Most metatheorems require only that certain axioms and rules are available and therefore hold in all extensions with additional axioms and rules. Others depend on certain things being absent (e.g. other rules, in the case of the deduction theorem); in such cases, we 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 ($\mathsf{T}[\Gamma, \Delta]$ is a theory 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 T iff

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

and *derivable* for T iff

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

Derivability implies admissibility, but not vice versa; e.g. 'cut' for first-order logic in the sequent calculus, is the classic example of an admissible, but not derived, rule (since it is no-longer admissible if the presentation is extended to, e.g., Peano arithmetic). It is easy to show that Troelstra's definition of derivability is equivalent to that used by Hindley and Seldin [8]; i.e. $\models_{\mathsf{T}[\{F_1,\ldots,F_n\},\emptyset]} F_{n+1}$, and that if a rule is derivable it holds in all extensions of T with new axioms and rules.

Our use of formal metatheory allows us to formalize both admissible and derived rules, as well as other sorts not fitting the above schemata. For example, the languages or proof systems for the F_i can be different, like in the various versions of the deduction theorem that we have formalized; our deduction theorems are neither derived nor admissible, since their statements involve different proof systems.

The simple theory hierarchies of systems like Isabelle, which we discussed in the introduction, allow the formal derivation of a subset of the derived rules of a theory. For example, in order to prove a theorem in Isabelle, one has (naturally) to specify the theory in which this is going to be done; then the proof can use any axioms, rules, or formally derived rules of that theory. All theorems proved hold in any extension with new axioms or rules. However, not all rules belonging to the class of derived rules can be formally derived this way. In particular, it is not possible to derive an instance of (der) which needs induction on the *language* of the theory, even though such a theorem would hold in all extensions of the *proof* system. On the other hand (because of this restriction) proven theorems hold not only in extensions of the proof system, but in extensions of the language as well.

5.2 Other approaches

Type theories support an approach to structuring theories (which can include induction principles) by formalizing them as Σ-types [9, 11]. For instance, these types can be used to formalize the languages and proof systems of logics, and afterwards one can prove inclusions and the existence of other kinds of relations between types and their members. Examples of this approach, which is very 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, but, like with approaches using Σ-types, one must explicitly construct (perhaps with help from the system) an appropriate interpretation mapping one theory into another.

References

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

[2] D. Basin and R. Constable. Metalogical frameworks. In G. Huet and G. Plotkin, editors, *Logical Environments.* Cambridge University Press, Cambridge, 1993.

[3] R. Constable et al. *Implementing Mathematics with the Nuprl Proof Development System.* Prentice-Hall, New Jersey, 1986.

[4] W. M. Farmer, J. D. Guttman, and F. J. Thayer. Little theories. In D. Kapur, editor, *Proc. CADE–11*, pages 567–581. Springer, Berlin, 1992.

[5] S. Feferman. Finitary inductive systems. In *Logic Colloquium '88*, pages 191–220. North-Holland, Amsterdam, 1990.

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

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

[8] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and λ-Calculus.* Cambridge University Press, Cambridge, 1986.

[9] D. B. MacQueen. Using dependent types to express modular structure. In *Proc. 13th Ann. ACM Symp. Principles of Programming Languages*, pages 227–286. ACM Press, New York, 1986.

[10] S. Matthews. Implementing FS_0 in Isabelle: adding structure at the metalevel. In J. Calmet and C. Limongelli, editors, *Proc. Disco'96.* Springer, Berlin, 1996.

[11] J. C. Mitchell and G. Plotkin. Abstract types have existential type. In *Proc. 12th Ann. ACM Symp. Principles of Programming Languages*, pages 37–51. ACM Press, New York, 1985.

[12] L. C. Paulson. A fixedpoint approach to implementing (co)inductive definitions. In A. Bundy, editor, *Proc. CADE-12*, pages 148–161. Springer, Berlin, 1994.

[13] L. C. Paulson. *Isabelle: A Generic Theorem Prover.* Springer, Berlin, 1994.

[14] L. C. Paulson. Set theory for verification: II. Induction and recursion. *J. Auto. Reas.*, 15:167–215, 1995.

[15] D. Prawitz. *Natural Deduction.* Almqvist and Wiksell, Stockholm, 1965.

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