Independence questions in a finite axiom-schematization of first-order logic

Benoît Jubin

16 February 2022

Dedicated to the memory of Norman Megill

Abstract

We review some independence results in a finite axiom-schematization of classical first-order logic introduced by Norman Megill. We also prove that a certain axiom scheme of this system is independent although all of its instances are provable from the other axiom schemes.

Contents

Introduction 2

1 The formal system 4

1.1 Schemes 5

1.2 Proofs 6

1.3 From the scheme level to the object level (and back?) 7

2 Soundness and Completeness 9

3 Independence 11

3.1 Object-independence 11

3.2 Supertruth and partial independence of ALLcomm 15

3.3 Variants of supertruth and partial independence of spec and ALLeq 18

A An elementary example of an independent but object-provable scheme 19

B Comments and variants of the TMM system 20

B.1 The propositional calculus bloc 21

B.2 The modal bloc 21

B.3 The vacuous generalization axiom scheme 21

B.4 The equality bloc 22

B.5 The denotation axiom scheme 22

B.6 The substitution axiom scheme 22

B.7 The “quantification over equal variables” axiom scheme 23

B.8 The generalized equality axiom scheme 23

B.9 The predicate axiom schemes 23

C Mario Carneiro’s proof of the independence of the rule of generalization 23

D Table of correspondence of scheme labels 24
Introduction

Many mathematical theories which are not finitely axiomatizable can nevertheless be axiomatized by a finite number of axiom schemes. Examples are classical first-order logic and Zermelo–Fraenkel set theory. If one wants to consider these theories as genuinely finitely axiomatized, one has not only to exhibit a finite number of axiom schemes, but also to carry out the proofs at the scheme level rather than at the object level, as is traditionally done. This gives rise to interesting questions on the relationship between proofs at these two levels, and in particular the relationship between “scheme-independence” and “object-independence.”

We present such a finite schematization of classical first-order logic mainly due to Megill [Meg95] building on earlier work by Tarski [Tar65], Kalish–Montague [KM65], and Monk [Mon65]. It has the advantage of requiring only very simple metalogic, in that it does not use the notions of bound and free variables, but only the notions of a variable occurring in a formula and of two variables being distinct. It has no notion of proper substitution, but only plain substitution.

In this article, “first-order logic” means classical classical\(^1\) one-sorted first-order logic with equality and no terms. Before formally defining the required notions, in particular that of scheme, in Section 1\(^2\) we give the axiom schemes of this system, indicating simply for now that “DV(x, ϕ)” (resp. “DV(x, y)”\(^3\)) means that in the instances of the corresponding scheme, the variable substituted for x should not occur in the formula substituted for ϕ (resp. the variables substituted for x and for y should be distinct). We choose a system with more numerous and weaker axiom schemes, which permits the study of several subsystems axiomatizing various well-studied logics.

\[
\begin{align*}
\text{propcalc} & \quad \\frac{\varphi \land \varphi \rightarrow \psi}{\psi} \quad \text{(mp)} \\
& \quad \frac{\varphi \rightarrow ((\psi \rightarrow \chi) \rightarrow ((\theta \rightarrow \psi) \rightarrow (\chi \rightarrow \tau)) \rightarrow (\psi \rightarrow \tau))}{(\varphi \rightarrow \psi) \rightarrow \varphi} \quad \text{(minimp)} \\
& \quad \frac{(\varphi \rightarrow \neg \psi) \rightarrow (\psi \rightarrow \neg \varphi)}{(\varphi \rightarrow \psi) \rightarrow \varphi} \quad \text{(peirce)} \\
& \quad \frac{\neg \varphi \rightarrow (\varphi \rightarrow \psi)}{(\varphi \rightarrow \psi) \rightarrow \varphi} \quad \text{(notelim)} \\
\end{align*}
\]

\[
\begin{align*}
\text{modal bloc} & \quad \frac{\varphi \rightarrow \forall x \varphi}{\forall x \varphi} \quad \text{(gen)} \\
& \quad \frac{\forall x (\varphi \rightarrow \psi) \rightarrow (\forall x \varphi \rightarrow \forall x \psi)}{(\forall x \varphi \rightarrow \forall x \psi)} \quad \text{ALLDistr} \\
& \quad \frac{\forall x \varphi \rightarrow \varphi}{\forall x \varphi} \quad \text{(spec)} \\
& \quad \frac{-\forall x \varphi \rightarrow \forall x \neg \forall x \varphi}{-\forall x \varphi \rightarrow \forall x \neg \forall x \varphi} \quad \text{(modal5)} \\
& \quad \frac{\varphi \rightarrow \forall x \varphi \land \text{DV}(x, \varphi)}{(\forall x \varphi \rightarrow \varphi \land \text{DV}(x, \varphi)} \quad \text{(vacGen)} \\
& \quad \frac{x \equiv x}{x \equiv x} \quad \text{(ALLcomm)} \\
\end{align*}
\]

\[
\begin{align*}
\text{equality} & \quad \frac{x \equiv x}{x \equiv x} \quad \text{(EQref1)} \\
& \quad \frac{x \equiv y \rightarrow y \equiv x}{x \equiv y \rightarrow y \equiv x} \quad \text{(Eqsymm)} \\
& \quad \frac{x \equiv y \rightarrow (y \equiv z \rightarrow x \equiv z)}{x \equiv y \rightarrow (y \equiv z \rightarrow x \equiv z)} \quad \text{(EQtrans)} \\
& \quad \frac{x \equiv x \rightarrow -\forall y y \equiv x \land \text{DV}(x, y)}{x \equiv x \rightarrow -\forall y y \equiv x \land \text{DV}(x, y)} \quad \text{(denot)} \\
& \quad \frac{\forall x (x \equiv y \rightarrow (\varphi \rightarrow \forall x (x \equiv y \rightarrow \varphi))) \land \text{DV}(x, y), \text{DV}(y, \varphi)}{\forall x (x \equiv y \rightarrow (\forall x \varphi \rightarrow \forall y \varphi)} \quad \text{(subst)} \\
& \quad \frac{\forall x x \equiv y \rightarrow (\forall x \varphi \rightarrow \forall y \varphi)}{\forall x x \equiv y \rightarrow (\forall x \varphi \rightarrow \forall y \varphi)} \quad \text{(ALLEq)} \\
& \quad \frac{-\forall x x \equiv y \rightarrow (-\forall x x \equiv z \rightarrow (y \equiv z \rightarrow \forall x y \equiv z))}{-\forall x x \equiv y \rightarrow (-\forall x x \equiv z \rightarrow (y \equiv z \rightarrow \forall x y \equiv z))} \quad \text{(genEq)} \\
\end{align*}
\]

\(^1\)One “classical” is for “classical propositional calculus”, as opposed to intuitionistic or minimal propositional calculus, and the other “classical” is in opposition to free logic.

\(^2\)Defined terms are written in boldface.
We call this system the Tarski–Monk–Megill system and denote it by $TMM$. It has some variants, that we also informally call the $TMM$ system. We present some of these variants in Appendix B, together with comments on some of these axiom schemes. Many variants have fewer axiom schemes, which can be an advantage for some applications (for instance, finding models), but makes them less modular. Here, we have chosen, on the contrary, to have more (conjecturally independent) axiom schemes, allowing for a piecemeal presentation, in which several subsets of axiom schemes axiomatize well-known logics (see Figure 1 of Appendix B).

For first-order theories on a given language, one adds to the above schemes $n$ “predicate axiom schemes” for each $n$-ary nonlogical predicate, sometimes called the “equality axiom schemes” associated with that predicate. For example, if the language contains exactly one nonlogical predicate, denoted by $\in$, which is binary and written in infix notation, then one adds the two predicate axiom schemes

$$x \equiv y \to (x \in z \to y \in z), \quad (ax-\in_1)$$

$$x \equiv y \to (z \in x \to z \in y). \quad (ax-\in_2)$$

If $\mathcal{L}$ is a language and emphasis is needed, then we denote by $TMM_{\mathcal{L}}$ the full system with the appropriate predicate axiom schemes.

The labels we used for these axiom schemes abbreviate, respectively: modus ponens, minimal implicational calculus, Peirce’s law, contraposition, “not” elimination, rule of generalization, “forall distributes over implication,” specialization, axiom corresponding to the modal logic axiom 5, vacuous generalization, “forall quantifiers commute,” equality is reflexive (resp. symmetric, transitive), denotation, substitution, “forall quantifiers over equal variables,” generalized equality. We make use of the standard translation between modal logic and monadic first-order logic (necessity maps to “$\forall x$” and possibility to “$\exists x$”, which we use as a shorthand for “$\neg \forall x \neg$”). Other common names for some of these axiom schemes are $\text{ALLdistr} = \text{modalK} = \text{kripke}$ and $\text{spec} = \text{modalT}$. Other schemes sometimes used as axioms are:

$$\forall x \neg \varphi \to \neg \forall x \varphi, \quad (\text{modalD})$$

$$\neg \varphi \to \forall x \neg \forall x \varphi, \quad (\text{modalB})$$

$$\forall x \varphi \to \forall x \forall x \varphi. \quad (\text{modal4})$$

The scheme spec implies modalD over propcalc (defined below).

We define the following subsystems of $TMM$:

$$\text{propcalc} := \{\text{mp, minimp, peirce, contrap, notelim}\},$$

$$\text{EQ} := \{\text{EQrefl, EQsymm, EQtrans}\},$$

$$\text{T} := \text{propcalc} \cup \text{EQ} \cup \{\text{gen, ALLdistr, vacGen, denot}\},$$

$$\text{TM} := \text{TMM} \setminus \{\text{ALLeq, genEq}\}.$$  

The system T was proved to be object-complete by Tarski [Tar65] and object-independent by Kalish–Montague [KM65]. The system $TMM$ was proved to be complete by Megill [Meg95]. The question of independence of its axiom schemes is still open, and the main new result in this article is a step in that direction.

Namely, we adapt Kalish–Montague’s and Monk’s proofs of independence (and in some cases provide new proofs)\(^5\) to prove the independence of the axiom schemes of $T \setminus \{\text{ALLdistr}\}$ and of subst in $TMM$.

---

\(^3\)The work of Kalish–Montague consisted in proving that the scheme of specialization $\text{spec}$ is object-provable in a related system, and in proving some independence results, but we think that this contributed slightly less to the final form of the system $TMM$ than the works of the three cited authors.

\(^4\)Any variable “denotes” in the sense of free logic.

\(^5\)The proof of the independence of the rule of generalization is new, and for the axioms of propositional calculus, one truth table is new and the others are classical.
Using new notions of “supertruth,” we prove the independence of \texttt{ALLcomm} in $\text{TMM} \setminus \{\text{spec, ALLeq}\}$ (Proposition 3.12) and of \texttt{spec} and \texttt{ALLeq} from $\text{TMM} \setminus \{\text{spec, ALLeq}\}$ (Proposition 3.13). The remaining open questions are the independence of these three axiom schemes in the whole axiom system, as well as the independence of \texttt{ALLdistr, modal5,} and \texttt{genEq} in $\text{TMM}$.

\textit{Remark} 0.1. A fundamental insight of Tarski, which makes these systems with weaker metalogic work, is that if $x$ and $y$ are disjoint (that is, the variables substituted for $x$ and for $y$ should be distinct), then the formula $\forall x (x \equiv y \rightarrow \varphi)$, which we denote by $[y/x]\varphi$, is equivalent to the result of the proper substitution of $y$ for $x$ in $\varphi$. With this notation, the scheme \texttt{subst} can be written $[y/x](\varphi \rightarrow [y/x]\varphi), \text{DV}(x,y), \text{DV}(y,\varphi)$.

\textit{Remark} 0.2. The finitary property of the axiom-schematization $\text{TMM}$ makes it well-suited to automatic proof verification. In particular, one of its variants is the axiom-schematization of first-order logic used in the main database written in the Metamath language (see [MW19, Appendix C]), which formalizes ZFC set theory (and many other areas of mathematics) on top of it. We give in Appendix D a table of correspondence of scheme labels used in this article and in that Metamath database, called \texttt{set.mm}, and browsable online at \url{http://us.metamath.org/mpeuni/mmset.html}.

\textbf{Plan of the article} In Section 1, we define the formal systems that are the object of study of this paper. In Section 2, we recall the completeness results of Tarski and Megill. In Section 3, we prove the independence results stated above. In particular, we exhibit a natural example in first-order logic of an independent scheme all of whose object-instances are redundant. Appendix A gives an elementary example of a similar phenomenon of independence of a scheme all of whose object-like instances are redundant. Appendix B makes some comments on the various axiom schemes and on some variants of $\text{TMM}$. Appendix C gives an alternate proof due to Mario Carneiro of the independence of \texttt{gen}. Appendix D contains a table of correspondence of scheme labels used here and in the Metamath database \texttt{set.mm}.

\textbf{Acknowledgments} I warmly thank Mario Carneiro and Norman Megill for careful reading of earlier drafts of this article and very useful comments. I also thank the contributors of the Metamath database \texttt{set.mm}, in particular Norman Megill, Mario Carneiro, and Wolf Lammen for their work related to axiomatic questions.

\textbf{Norman Megill} I discovered the notion I called “supertruth” in August 2020 and rapidly began discussing it with Mario Carneiro and Norman Megill. Over the following year, it slowly extended to the present article, which benefited from many exchanges with Mario and Norm. Although I have never met Norm personally, I enjoyed many insightful discussions we had over the years, on this and related topics. The formalized mathematics tool Metamath is his creation, and the vibrant community he gathered around it owes him a lot. I dedicate this article to his memory.

1 The formal system

In this section, we define the formal systems that are the object of study of this paper. They are special cases of “Metamath systems,” whose general definition can be found in [MW19, Appendix C.2] and [Car16, Subsection 2.1].
1.1 Schemes

We denote by $\omega = 0, 1, 2, \ldots$ the set of natural numbers. We define two disjoint sets of symbols: the sets of variable metavariables and the set of formula metavariables, defined respectively by

\begin{align*}
\text{vrMV} &:= \{ x_i \mid i \in \omega \}, \\
\text{fmMV} &:= \{ \varphi_i \mid i \in \omega \},
\end{align*}

and we denote by $\text{MV} := \text{vrMV} \cup \text{fmMV}$ the set of metavariables. We sometimes informally write $x, y, z, \ldots$ instead of $x_0, x_1, x_2, \ldots$ and $\varphi, \psi, \chi, \ldots$ instead of $\varphi_0, \varphi_1, \varphi_2, \ldots$.

A language is a set of symbols (disjoint from the above sets), whose members are called nonlogical predicates, together with a function from this set to $\omega$ called the arity function of the language. We will typically use the notation $L = \{ P_0, P_1, P_2, \ldots \}$, with an often implicit arity function $r : L \to \omega$.

A metaformula (on a language $L$) is a finite string of characters in $\{ \to, \neg, \forall, \equiv \} \cup \text{MV} \cup L$ conforming to the usual formation rules. The height of a metaformula is defined as usual. For the sake of readability, we will use an infix notation, and therefore parentheses, but one could use a prefix or suffix notation, which renders parentheses unnecessary. We denote the set of metaformulas by $\text{MF} \subseteq (\{ \to, \neg, \forall, \equiv \} \cup \text{MV} \cup L)^*$ or $\text{MF}_L$ if emphasis on the language is needed. Note that $\text{fmMV} \subseteq \text{MF}$.

A scheme is a triple consisting of a finite set of metaformulas (its hypotheses), a metaformula (its conclusion), and a (finite) set of pairs of metavariables occurring in its hypotheses or conclusion (its disjoint variable conditions, or DV conditions; this terminology will become clear in Equation (9)). We use $\Phi, \Psi, \chi, \ldots$ to denote metaformulas or schemes (which one will be clear from context). If $\Phi$ is a metaformula or a scheme, then we denote by $\text{OC}(\Phi) \subseteq \text{MV}$ the set of metavariables occurring in it (for schemes, this means: occurring in its hypotheses or conclusion). If $\Phi$ is a metaformula or a scheme, then we denote by $\text{DV}(\Phi)$ its set of DV conditions. With this notation,

$$\text{DV}(\Phi) \subseteq P_2(\text{OC}(\Phi))$$

for any scheme $\Phi$, where $P_2$ denotes the set of subsets of cardinality 2 of the given set.

A scheme will typically be written as $\Phi = (\{ \Phi_1, \ldots, \Phi_n \}, \Phi_0, \text{DV}(\Phi))$ or informally $\Phi_1 \& \ldots \& \Phi_n \implies \Phi_0$, $\text{DV}(\Phi)$ as in the introduction. When there is no ambiguity, especially when there are no hypotheses, we will generally use the same notation for a scheme and its conclusion. We will sometimes use informal self-explanatory notation. For instance, the scheme $(\varnothing, \neg \forall x_0 \neg x_0 \equiv x_1, \{ x_0, x_1 \})$ may be abbreviated as $\exists x_0 x_0 \equiv x_1$, $\text{DV}(x_0, x_1)$. We will often write $(\Phi_I, \Phi_0, D)$ instead of $(\Phi_I, \Phi_0, D \cap P_2(\text{OC}(\Phi)))$. Also, a scheme with no DV conditions may be written simply as $(\Phi_I, \Phi_0)$.

A (type-preserving) substitution is a function

$$\sigma : \text{MV} \to \text{vrMV} \cup \text{MF}$$

More precisely, one can consider two bijective functions $x$ and $\varphi$ from $\omega$ onto two disjoint sets.

Explicitly, a metaformula is either a formula metavariable, a predicate (equality or a non-logical predicate) followed by the number of variable metavariables corresponding to its arity, the universal quantifier followed by a variable metavariable and a metaformula, the negation of a metaformula, or the implication of two metaformulas.

We use the Kleene star to denote the set of finite strings of characters on a given alphabet.
such that $\sigma(\text{vrMV}) \subseteq \text{vrMV}$ and $\sigma(\text{fmMV}) \subseteq \text{MF}$, and $\{m \in \text{MV} \mid \sigma(m) \neq m\}$ is finite. If a substitution is defined as a partial function, it is assumed to be the identity where not defined. The action of a substitution on a metaformula is defined in the natural way. The resulting metaformula is called an “instance” of the original metaformula. We denote instantiation with a self-explanatory notation. For example, the metaformula $(\forall x_0 \varphi_0)^{x_0 \leftarrow x_1, \varphi_0 \leftarrow x_2} \equiv x_2$ is $\forall x_1 x_0 \equiv x_2$.

The action of a substitution on a scheme is defined as follows: the substitution acts on its hypotheses and conclusion as above, and also on its set of DV conditions in the natural way: if $D \subseteq \mathcal{P}_2(\text{MV})$ and $\sigma$ is a substitution, then

$$D^\sigma := \{\{m, n\} \in \mathcal{P}_2(\text{MV}) \mid \exists \{m', n'\} \in D ((m \in \text{OC}(\sigma(m')) \land n \in \text{OC}(\sigma(n'))))\}. \quad (8)$$

A substitution $\sigma$ is legitimate on a scheme $\Phi$ if it does not violate its DV conditions, that is, if

$$\{m, n\} \in \text{DV}(\Phi) \text{ implies } \text{OC}(\sigma(m)) \cap \text{OC}(\sigma(n)) = \emptyset. \quad (9)$$

If a substitution is applied to a scheme, it will be implicitly assumed that it is legitimate on that scheme. An instance of a scheme is the result of a legitimate substitution possibly followed by the addition of any number of DV conditions. Formally, $\Psi \in \text{Inst}(\Phi)$ if and only if there exists a substitution $\sigma$ legitimate on $\Phi$ such that $\Psi_i = \Phi_i^\sigma$ for $0 \leq i \leq n$ and $\text{DV}^\sigma(\Phi) \subseteq \text{DV}(\Psi)$.

Note the following important idempotence or transitivity property: an instance of an instance of a metaformula or scheme is an instance of that metaformula or scheme (and a metaformula or scheme is an instance of itself, so instantiation yields a preorder on the set of metaformulas and a preorder on the set of schemes).

**Remark 1.1.** Schemes are a bit harder to manipulate than formulas of first-order logic. For instance, one should be careful when talking about the “negation” of a scheme. For example, neither $\forall xx \equiv y$ nor $\neg \forall xx \equiv y$ are true. Indeed, the scheme $\forall xx \equiv x$ is true, and the scheme $\neg \forall xx \equiv y$, $\text{DV}(x, y)$ is true in any model with at least two elements, and these two schemes are respective instances of the two above schemes. This added difficulty is akin to the one associated with formulas (possibly open, that is, with free variables) as opposed to sentences (closed formulas) in classical first-order logic.

### 1.2 Proofs

Proofs are defined similarly as in classical logic. The precise definition for general Metamath systems can be found in the cited references. A proof of a scheme $\Phi$ from a set of schemes $S$ is a couple consisting of $\Phi$ and a finite sequence $P$ of metaformulas (called the “lines” of the proof) satisfying the following two conditions:

1. if a line $P_i$ is not a hypothesis of $\Phi$, then there exist $i_1, \ldots, i_n < i$ such that $\{\{P_{i_1}, \ldots, P_{i_n}\}, P_i, \text{DV}(P)\}$ is an instance of a scheme in $S$, where

$$\text{DV}(P) := \text{DV}(\Phi) \cup \{\{m, n\} \mid m \in \text{OC}(P) \setminus \text{OC}(\Phi) \text{ and } n \in \text{OC}(P)\} \quad (10)$$

is the set of DV conditions of $P$;

2. the final line of the proof is the conclusion of $\Phi$.

Note that $\text{DV}(P) \cap \mathcal{P}_2(\text{OC}(\Phi)) = \text{DV}(\Phi)$. A scheme is provable from a set of schemes if there exists a proof of that scheme from that set of schemes. The set of schemes provable from the set of schemes $S$ is denoted by $\overline{S}$. We also write $S \vdash \Phi$ for $\Phi \in \overline{S}$. A metavariable in $\text{OC}(P) \setminus \text{OC}(\Phi)$ is called a
**dummy** metavariable of $P$, and the above condition in the definition of $DV(P)$ means that all dummy metavariables are disjoint from all other metavariables.

We now define the action of a substitution on a proof. Let $\sigma$ be a substitution and $P$ be (the second component of) a proof of the scheme $\Phi$. First, rename every dummy metavariable of $P$ to avoid clashes.\(^{11}\) Then, apply the substitution to each line of $P$. The resulting sequence is denoted by $P^\sigma$ (it can be defined unambiguously as indicated in the previous footnote). Similarly, the action of a substitution on a set of schemes is defined to be the action on each of its elements.

**Proposition 1.2.** If $P$ is a proof of the scheme $\Phi$ from the set $S$ of schemes and $\sigma$ is a substitution, then $P^\sigma$ is a proof of $\Phi^\sigma$ from $S^\sigma$. In particular, if a scheme is provable from a set $S$, then so is any instance of it.

**Proof.** Straightforward. \(\square\)

**Remark 1.3.** The rules of modus ponens and generalization are schemes of the system, like any other. Therefore, the only “metarule” one can use in proofs is instantiation. The semantics we will define to prove our independence results must therefore be compatible with instantiation. The easiest way to achieve this is to use definitions of the form: “a scheme is *true if all its instances (including itself) satisfy . . .”

### 1.3 From the scheme level to the object level (and back?)

We now explain the relationship between the scheme level and the object level of usual logic. The fundamental notion to connect these two levels is that of **object-instantiation**. We define a new set of symbols disjoint from MV, the set of (individual) variables, by\(^{12}\)

$$VR := \{v_i \mid i \in \omega\}. \quad (11)$$

The set of formulas of first-order logic is denoted by $FM \subseteq \{(\rightarrow, \neg, \forall, \equiv) \cup VR \cup L\}^*$ and is defined as usual.\(^{13}\) Formulas will often be denoted by $\phi, \psi, \ldots$ but this should cause no confusion with elements of $fmMV$. An object-substitution is a function $\tau: MV \to VR \cup FM$ such that $\tau(vrMV) \subseteq VR$ and $\tau(fmMV) \subseteq FM$. The action of an object-substitution on a metaformula is defined in the natural way, and results in a formula called an **object-instance** of the metaformula. We use the same notation as for instantiation. For example, the formula $(\forall x_0 \varphi_0)^{x_0 \rightarrow v_1, \varphi_0 \equiv v_2}$ is $\forall v_0 v_1 v_2 \equiv v_2$.

One extends this action to schemes as above, with the same notion of legitimacy (for example, if a scheme has $DV(x_0, \varphi_0)$, then one cannot substitute in it $v_0 \equiv v_1$ for $\varphi_0$ and $v_0$ for $x_0$). We call the resulting couples (consisting of a set of formulas and a formula) **formulas with hypotheses**.

Similarly to the transitivity property of instantiation, an object-instance of an instance of a scheme is an object-instance of that scheme.

Suppose that $P$ is a proof of a scheme $\Phi$ from a set of schemes $S$ and that $F$ is an object-instance of $\Phi$. It is immediate to construct from $P$ a proof (at the object level) of $F$ from object-instances of schemes in $S$. A scheme is **object-provable** from a set of schemes if all its object-instances are provable from object-instances of these schemes. Therefore, the above shows that provability implies object-provability. Does the converse hold? Answer in the next paragraph.

**Remark 1.4.** Object-provability depends on the language, whereas provability does not. More precisely, let $L$ and $L'$ be two languages and let $\Phi$ be a scheme and $S$ be a set of schemes, both on the language $L \cap L'$. Then, $\Phi$ is provable from $S$ in the language $L$ if and only if it is so in the language $L'$: in a proof on

---

\(^{11}\)For instance, rename each dummy metavariable $m_i$ of $P$ to the metavariable $m_{i+N+1}$, where $N$ is the largest $j \in \omega$ such that $\sigma(m_j) \neq m_j$ or $m_j \in OC(\sigma(m_k))$ for some $k$ with $\sigma(m_k) \neq m_k$.

\(^{12}\)More precisely, one can consider a bijective function $v: \omega \to VR$ with $MV \cap VR = \emptyset$.

\(^{13}\)Note that there is no notion of disjoint variable or $DV$ condition at the object level: two variables $v_i$ and $v_j$ are simply the same if $i = j$ or different if $i \neq j$. 
any language, replace all atomic expressions involving a nonlogical predicate not in $\mathcal{L} \cap \mathcal{L}'$ by a constant, say $\top$. The result is a proof of $\Phi$ from $S$ in $\mathcal{L} \cap \mathcal{L}'$. On the other hand, it can be the case that $\Phi$ is object-provable from $S$ in a language but not in a larger one, because object-instantiation involves the language: on a larger language, $\Phi$ has more object-instances. This is detailed in the next paragraph where we prove (after Monk) object-independence of $\text{subst}$ when the language has a non-nullary nonlogical predicate.

**Independence** A scheme is (**object-**)independent from a set of schemes if it is not (object-)provable from them. Equivalently, a scheme is object-independent from a set of schemes if at least one of its object-instances is not provable from their object-instances. A scheme is (**object-**)independent in a set of schemes if it is (object-)independent from the other schemes in that set. A scheme is (**object-**)redundant in a set of schemes if it is not (object-)independent in that set, or equivalently if it is (object-)provable from the other schemes in that set.\(^{14}\)

Since provability implies object-provability, object-independence implies independence. Does the converse hold? It does not. Examples can be given for Metamath systems that are more elementary than first-order logic, even if somewhat artificial (see Appendix A). In first-order logic, an example comes from Monk’s proof of the object-independence of $\text{subst}$ in $\text{TM}$ on any language containing a non-nullary non-logical predicate, even though it is object-redundant on an empty language, as proved by Tarski ([Tar65, Lem. 13 and 16]). The argument for independence is as follows: since $\text{subst}$ is object-independent on any language containing a non-nullary non-logical predicate (and no corresponding predicate axiom scheme), it is independent. Therefore, if one adds the predicate axiom schemes associated with each predicate, one has object-redundancy and independence.

Are there examples which do not require adding a nonlogical predicate and associated predicate axiom schemes? There are. An example is given by our proof of the independence of the instance $\forall x. x \equiv y \rightarrow x \equiv y$ of $\text{spec}$ in $\text{TM} \setminus \{\text{spec}, \text{ALLeq}\}$ in Proposition 3.13, even though it is object-provable from $T$ as proved by Kalish–Montague.

**Emulating the object level at the scheme level** The following remark due to Norman Megill is useful to treat some object-level problems at the scheme level.

The mapping $\text{VR} \rightarrow \text{MV}, v_j \mapsto x_j$ gives rise to an injection $i: \text{FM} \hookrightarrow \text{MF}$. One can extend it to formulas with hypotheses as follows: $i$ acts as above on hypotheses and conclusion, and adds all possible DV conditions among occurring metavariables. In particular, the image of $i$ is the set of schemes containing no formula metavariables and with all DV conditions among occurring metavariables. We call these schemes **object-like**.

Defining the action of $i$ on sets of formulas with hypotheses and on proofs in the natural way, one sees that if $P$ is an object-level proof of the formula with hypotheses $F$ from the set of formulas with hypotheses $S$, then $i(P)$ is a proof of the scheme $i(F)$ from the set of schemes $i(S)$. In particular, a scheme is object-provable from a set of schemes $S$ if and only if all its object-like instances are provable from $S$.

**Remark** 1.5. To demonstrate the usefulness of this method, we prove that $\text{spec}$ is object-provable from $T$.\(^{15}\)

We will actually prove the more general result: for all metaformula $\Phi$ and for all variable metavariable $x$, one has

$$T \vdash \left( \forall x \Phi \rightarrow \Phi, \{\{x, m\} \mid m \in \text{OC}(\Phi) \setminus \{x\}\} \right).$$

(12)

This generalizes object-provability of $\text{spec}$ in two ways: $\Phi$ may contain variable metavariables, and the DV conditions are only with $x$, and not among other metavariables occurring in $\Phi$. First, one has

$$T \vdash \left( x \equiv y \rightarrow (\varphi \leftrightarrow \psi) \implies \forall x \varphi \rightarrow \varphi, \{\{x, y\}, \{x, \psi\}, \{y, \varphi\}\} \right).$$

(13)

\(^{14}\)We use “redundant” for “not independent” because “dependent” has another meaning.

\(^{15}\)That fact is also a consequence of Theorem 2.2.
This is proved as $\spw$ in set.mm. The hypothesis of this scheme can be thought of as “$\psi$ is the result of replacing every occurrence of $x$ in $\varphi$ by a fresh metavariable $\gamma$.”

Now, it suffices to prove the hypothesis of the above scheme for a particular choice of $\psi$ (possibly with additional DV conditions as long as they do not exceed those of (12)). We prove that for all metaformula $\Phi$ and for all variable metavariables $x$ and $y$, if $y$ is fresh (that is, if $y \notin \{x\} \cup \text{OC}(\Phi)$), then

$$\forall \\{x, y\} . \{m \in \text{OC}(\Phi) \cup \{x\} \} .$$

(14)

This is easily proved by induction on the height of $\Phi$. The base cases correspond to $\Phi$ atomic, that is, of the form $\varphi_i$, or $x_i \equiv x_j$, or similarly with a nonlogical predicate. These cases are respectively trivial and proved from $\text{EQ}$ and from the predicate axiom schemes for the nonlogical predicates. As for the induction steps, the propositional calculus cases (that is, $\Phi$ is an implication or a negation) pose no difficulty. If $\Phi$ is of the form $\forall z \Psi$ with $x \neq z$, then the result follows easily from $\text{ALLdistr}$. Finally, suppose that $\Phi$ is of the form $\forall x \Psi$. One has to prove $(x \equiv y \to (\forall x \Psi \leftrightarrow \forall y \Psi^{x\leftrightarrow y}))$, knowing by induction hypothesis that $(x \equiv y \to (\Psi \leftrightarrow \Psi^{x\leftrightarrow y}))$, then follows easily from $\text{cbvalw}$ in set.mm.

The provability result (12) is about the best one can hope for in $\text{T}$, in terms of DV conditions, in view of the independence of $\forall xx \equiv y \to x \equiv y$ proved in Proposition 3.13.

Similarly, one can prove object-provability, and actually similar generalizations thereof as above, corresponding to $\text{modal5}$ (and similarly with $\text{modal1B}$ and $\text{modal4}$), $\text{ALLcomm}$, $\text{subst}$ and $\text{ALLeq}$: for any metaformula $\Phi$ and variable metavariable $x$, one has, respectively,

$$\forall x \Phi \to \forall x \exists x \Phi, \{\{x, m\} | m \in \text{OC}(\Phi) \cup \{x\}\} \}$$

(15)

$$\forall x \forall y \Phi \to \forall y \forall x \Phi, \{\{y, m\} | m \in \{x\} \cup \text{OC}(\Phi) \cup \{y\}\}$$

(16)

$$\forall x (x \equiv y \to (\Phi \to \forall x (x \equiv y \to \Phi)))$$

(17)

$$\text{T} \vdash (\forall xx \equiv y \to (\forall x \Phi \to \forall y \Phi), \{\{x, m\} | m \in \text{OC}(\Phi) \cup \{x\}\} \cup \{\{y, m\} | m \in \{x\} \cup \text{OC}(\Phi) \cup \{y\}\} \}.$$ 

(18)

2 Soundness and Completeness

In order to define soundness and completeness of a system, one has to first define its semantics. At the object level, this is the usual semantics of first-order logic: a statement with hypotheses is true\(^{18}\) if it is true in every nonempty model of first-order logic (and in the case of formulas with hypotheses, if it preserves truth in every such model). At the scheme level, we define, as expected, a scheme to be true\(^{19}\) when all its object-instances are true\(^{19}\).

**Theorem 2.1** (Soundness). *If a scheme is provable from a set of true schemes, then it is true.*

**Proof.** Let $P$ be a proof of $\Phi$ from a set $S$ of true schemes and let $F$ be an object-instance of $\Phi$. As mentioned in the previous section, one can construct from $P$ a proof (at the object level) of $F$ from object-instances of schemes in $S$. These object-instances of true schemes are true formulas with hypotheses.\(^{16}\)

\(^{16}\)If $x = y$, then the result is a tautology (as for the weakened forms of $\text{ALLcomm}$ and $\text{ALLeq}$ below) but these degenerate forms are not needed.

\(^{17}\)The axiom $\text{modalB}$ is provable from $\text{T}$ without need of weakening.

\(^{18}\)A more frequent term is “(universally) valid,” especially when “true” is reserved for specific models and assignments.

\(^{19}\)After all, this is the original aim of the introduction of the scheme level: formally describe the object level.
Therefore, by soundness of first-order logic, $F$ is true. Since $F$ can be any object-instance of $\Phi$, this implies that $\Phi$ is true.

The axiom schemes of $\text{TMM}$ are true, so $\text{TMM}$ and its subsystems can only prove true schemes.

A set of schemes is called complete (resp. object-complete) if all true schemes are provable (resp. object-provable) from it. Since provability implies object-provability, completeness implies object-completeness. We now restate in our language the two fundamental completeness theorems regarding the systems\(^{20}\) considered here.

**Theorem 2.2** ([Tar65, Thm. 5]). The system $T$ is object-complete.

**Theorem 2.3** ([Meg95, Thm. 9.7]). The system $\text{TMM}$ is complete for schemes with no DV conditions involving two formula metavariables.

**Remark 2.4.** Tarski proved his result with the scheme $\text{spec}$ added to $T$, and Kalish–Montague proved ([KM65, Lem. 9]) that $\text{spec}$ is object-redundant, resulting in the present statement.

**Remark 2.5.** The completeness results of Monk are a bit different, since he studies substitutionless logic. Although he stays at the object level, his methods and results in the study of $\text{TM}$ can be considered as an intermediate step between the object-completeness of $\text{T}$ and the completeness of $\text{TMM}$.

As an aside, note that the system $\text{TMM}$ contains two DV conditions: in $\text{vacGen}$ and in $\text{subst}$. The DV condition in $\text{subst}$ can be removed if one uses the variant $\text{ax-12}$ of $\text{subst}$ described in Appendix B. The next proposition shows that one cannot remove the other DV condition. It also shows the necessity of the restriction “with no DV conditions involving two formula metavariables” in Megill’s completeness theorem.

**Proposition 2.6.** An axiom-schematization of first-order logic that is complete (resp. complete for schemes with no DV conditions involving two formula metavariables, resp. with no DV conditions involving formula metavariables), contains at least one scheme that becomes false when removing its DV conditions involving two formula metavariables (resp. its DV conditions involving formula metavariables, resp. its DV conditions).

**Proof.** Suppose that there exists a complete axiom-schematization with no scheme becoming false when removing its DV conditions involving two formula metavariables. Such a system proves the scheme $(\exists x \varphi \rightarrow \forall x \varphi) \lor (\exists x \psi \rightarrow \forall x \psi)$, $\text{DV}(\varphi, \psi)$. Indeed, this scheme is true (if $\varphi$ and $\psi$ are disjoint, then at least one of them is disjoint from $x$). Therefore, that same scheme without its DV condition would also be provable: simply remove all DV conditions involving two formula metavariables from its proof (this is still a proof because DV conditions involving two formula metavariables cannot be produced from DV conditions of other forms by Equation (8)). But that scheme is false: for instance, substitute $x \equiv y$ for both $\varphi$ and $\psi$.

For the second (resp. third) case, proceed similarly using the scheme $\text{vacGen}$ (resp. the scheme $\text{oneObj}$, see next remark).

**Remark 2.7.** A result of Monk ([Mon71]) implies that there is no finite axiom-schematization with no DV conditions that is object-complete (let alone complete for all schemes involving no DV conditions). Indeed, when there are no DV conditions in our schemes, our notion of object-instantiation reduces to that article’s notions of substitution and instantiation, so its main theorem proving the non-existence of finite axiomatizations applies.

\(^{20}\)We use this term synonymously with “set of schemes,” now that the background theory has been established.
Remark 2.8. On the other hand, there exist finite object-complete axiom-schematizations of first-order logic with no DV conditions involving formula metavariables. An example can be obtained from TMM by replacing $\text{vacGen}$ with the axiom scheme

$$\forall xx \equiv y \to (\varphi \to \forall x\varphi), \text{DV}(x,y) \tag{oneObj}$$

and adding for each nonlogical predicate $P$ an axiom scheme analogous to $\text{genEq}$:

$$\neg \forall xx \equiv x_1 \to (\cdots \to (\neg \forall xx \equiv x_n \to (P(x_1,\ldots,x_n) \to \forall xP(x_1,\ldots,x_n)))\ldots). \tag{genP}$$

We prove, following Megill, that this axiom-schematization is object-complete. It suffices to prove that all object-like instances of $\text{vacGen}$ are provable from it. We do this by induction on the height of $\Phi$. If $\Phi$ is an atomic formula involving the predicate $P$, then it is easily proved from $\text{intuitcalc} \cup \{\text{oneObj}, \text{gen}_P\}$ (this includes the equality predicate since we use $\text{gen}_\equiv$ as a synonym of $\text{genEq}$). The induction step in the case of an implication, a negation, or a universal quantification is easily proved from $\text{pure}^{21}$ (see definition of this subsystem in Figure 1 of Appendix B).

Remark 2.9. These results leave open the questions:

- Are there finite axiom-schematizations with no DV conditions involving formula metavariables that are complete for all schemes of the same form? (The previous remark shows that there are object-complete ones.)

- Are there finite axiom-schematizations that are complete for all schemes (including those with DV conditions involving two variable metavariables)?

3 Independence

As mentioned above, since provability implies object-provability, object-independence (on any given language) implies independence. We begin by giving the easier object-independence results. For some results, we suppose that the language contains a non-nullary nonlogical predicate, and object-independence in the empty language remains an interesting open question.

3.1 Object-independence

The proofs in this subsection are at the object level, so we use valuations defined on formulas (and not metaformulas), that is, functions $\text{val}: \text{FM} \to A$ where $A$ is a set of truth values, which will be here of the form $\{0,1,\ldots,n\}$. A subset of these values are considered as true. Valuations will be implicitly defined by induction on formula height.

3.1.1 Propositional calculus

**Independence of the rule of modus ponens** [Mon65, Thm. 9, Part 9 (Detachment)] Without this rule, one cannot prove statements shorter than the axioms, like $v_0 \equiv v_0$.

**Independence of the axioms of propositional calculus** If one ignores quantifiers, then the axiom schemes of TMM not in propositional calculus nor the equality bloc become particularly simple. Formally, “ignoring quantifiers” means choosing valuations $\text{val}$ such that $\text{val}(\forall v_i \varphi) = \text{val}(\varphi)$ for any $i \in \omega$ and any formula $\varphi \in \text{FM}$. We use such valuations to prove the object-independence of the propositional calculus axiom schemes.

$^{21}$See $\text{ax5im}$, $\text{ax5n}$ and $\text{ax5al}$ in set.mm, where the base cases are given by $\text{ax5eq}$ and $\text{ax5el}$.
Independence of minimp Consider the valuation given by the truth table

|   | 0 | 1 | 2 | 3 | 4 | ¬ |
|---|---|---|---|---|---|---|
| *0| 0 | 1 | 1 | 1 | 2 |   |
| 1 | 0 | 0 | 0 | 0 | 0 |   |
| 2 | 0 | 0 | 0 | 0 | 0 |   |
| 3 | 0 | 0 | 4 | 0 | 4 |   |
| 4 | 0 | 0 | 3 | 3 | 0 | 1 |

where only 0 is considered true, and set \( \text{val}(v_i \equiv v_j) := 0 \). This validates \( \text{mp, notnotintro} \) and KI (that is, \( \{\text{simp, id}\} \)), see Appendix D for the labels, and therefore all the axiom schemes not in propositional calculus, including the axiom schemes in the equality bloc, and also \( \text{contrap, notelim, and peirce} \), but falsifies \( \text{minimp} \).

Remark 3.1. If a language contains a nonlogical predicate, say \( P \), then one can evaluate it to 3 regardless of its arguments. Then, the assignment given in the previous footnote is realizable (using also \( \text{val}(\neg v_0 \equiv v_0) = 2 \)). If the language is empty, then no object-instance of any formula has value 3 nor 4, though refuting assignments for \( \text{minimp} \) all require it. I do not know if object-independence holds on the empty language. A computer search might be able to find a valuation to prove it.

Independence of peirce Use the Gödel (2)truth-table from [Rob68]

|   | 0 | 1 | 2 | ¬ |
|---|---|---|---|---|
| *0| 0 | 1 | 2 | 2 |
| 1 | 0 | 0 | 2 | 2 |
| 2 | 0 | 0 | 0 |   |

where only 0 is considered true, and set \( \text{val}(v_i \equiv v_j) := 0 \) if \( i = j \) else 2. It validates intuitionistic propositional calculus, hence also the later axiom schemes, including the axiom schemes in the equality bloc, but falsifies \( \text{peirce} \).

Remark 3.2. Similarly to the previous remark, one needs a nonlogical predicate to be evaluated to 1. I do not know if object-independence holds on the empty language.

Independence of contrap Use the truth-table

|   | 0 | 1 | 2 | ¬ |
|---|---|---|---|---|
| *0| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 1 |

where only 0 is considered true, and set \( \text{val}(v_i \equiv v_j) := 0 \) if \( i = j \) else 2. It validates implicational calculus and \( \text{notnotintro} \), and therefore all the axiom schemes not in propositional calculus. It validates axiom schemes in the equality bloc, and also \( \text{notelim} \), but falsifies \( \text{contrap} \): consider its instance given by \( \varphi \leftarrow v_0 \equiv v_1, \psi \leftarrow v_0 \equiv v_0 \).

From this point on, valuations have to satisfy implicational calculus (that is, \( \{\text{mp, minimp, peirce}\} \)). Therefore, we make the following definitions. If \( P \) is a proposition (in the informal language), then [P] is equal to 1 if \( P \) is true, else 0. A valuation \( \text{val} : \text{FM} \rightarrow \{0, 1\} \) is an imp-valuation if \( \text{val}(\varphi \rightarrow \psi) = 1 \).

22This table was found by computer search to solve a similar problem in [Jub21].
23An assignment that falsifies \( \text{minimp} \) is \( \varphi \leftarrow 0, \psi \leftarrow 3, \chi \leftarrow 0, \theta \leftarrow 0, \tau \leftarrow 2 \).
24An assignment that falsifies \( \text{peirce} \) is \( \varphi \leftarrow 1, \psi \leftarrow 2 \).
25This notation is called the Iverson bracket, see D. E. Knuth, Two notes on notation, Amer. Math. Monthly 99 (1992), no. 5, 403–422, arXiv:math/9205211[math.HO].
ψ) = max(1 − val(φ), val(ψ)) for all formulas φ, ψ ∈ FM. It is a pc-valuation if furthermore val(¬φ) = 1 − val(φ) for all formulas φ ∈ FM. An imp= valuation (resp. a pc= valuation) is an imp-valuation (resp. a pc-valuation) that is standard on equality, that is, such that val(v_i v_j) = [i = j].

**Independence of notelim** Use the imp= valuation which ignores quantifiers and is always true on negations, that is, such that val(¬φ) = 1. Then, all axiom schemes are validated except notelim: consider its instance given by φ ← v_0 v_0, ψ ← v_0 v_1.

### 3.1.2 Modal bloc

**Independence of the rule of generalization** Because subst is a generalization of a scheme over a variable that is free in it, a proof of the independence of gen is a bit harder to find than in [KM65] or [Mon65]. Note also that {propcalc, EQrefl, genEq} ⊢ ∀xx ≡ x.

Since propcalc, EQ and spec have to hold, we define val to be a pc= valuation given on quantified formulas by

\[ \text{val}(∀v_i φ) := \text{val}(φ) \text{ val}_i(φ) \] (19)

for all i ∈ ω and φ ∈ FM, where the val_i’s are valuations. Since val has to satisfy ALLdistr, we assume that each val_i is an imp-valuation. We also assume that val_i ignores universal quantifiers, so that ALLcomm is validated. More precisely, we set val_i(∀v_j φ) := val_i(φ) if i ≠ j and val_i(∀v_i φ) := 1. On atomic formulas, we define

\[ \text{val}_i(v_j ≡ v_k) := \{ j = k \text{ or } i \notin \{ j, k \} \} \] (20)

for i, j, k ∈ ω. Finally, val_i is defined on negated formulas by:

\[ \text{val}_i(¬(φ → ψ)) := \text{val}_i(φ → ψ), \]
\[ \text{val}_i(¬¬φ) := \text{val}_i(φ), \]
\[ \text{val}_i(¬v_j ≡ v_k) := \text{val}_i(v_j ≡ v_k), \]
\[ \text{val}_i(¬∀v_j φ) := 1 \]

for i, j, k ∈ ω and φ, ψ ∈ FM. In other words, val_i ignores negations not followed by a universal quantifier and validates negations of universally quantified formulas.

Since val is a pc-valuation, it validates propcalc. Since val and each val_i are imp-valuations, val validates ALLdistr. Since val(∀v_i φ) ≤ val(φ), val validates spec. Since val is standard on equality, it validates EQ.

Let FV(φ) denote the set of free variables occurring in the formula φ. A proof by induction on formula height shows that if v_i ∉ FV(φ), then val_i(φ) = 1. Therefore, val validates modal5 and vacGen. One has val(∀v_i ∀v_j φ) = val(φ) val_i(φ) val_j(φ) if i ≠ j, which is symmetric in i, j. Therefore, val validates ALLcomm. One has val(∀v_j ¬v_j ≡ v_i) = val(¬v_j ≡ v_i) val_j(v_j ≡ v_i) = 0 (even when i = j). Therefore, val satisfies denot (even when its DV condition is dropped).

If i ≠ j, then val(v_i v_j) = val_i(v_i v_j) = 0, and val_i is an imp-valuation, so val validates subst. Also, val validates genEq, since it actually validates v_j ≡ v_k → ∀v_i v_j ≡ v_k. Note that it also validates the strong denotation axiom ¬∀v_j ¬v_j v_i and all its generalizations. Finally, val validates ALLeq since val(∀v_i v_i ≡ v_j) = [i = j].

To show that val does not validate gen, we have to add a non-nullary predicate, say P, and declare val(P(v_i)) = 1 and val_j(P(v_i)) = [i ≠ j], so that we still have val_i(φ) = 1 if v_i ∉ FV(φ). Then, val(P(v_0)) > val(∀v_0 P(v_0)).

Another proof of independence (but not object-independence), due to Mario Carneiro, is given in Appendix C.

---

26 A priori, we could allow val to interpret ≡ like any equivalence relation on the domain of discourse, but in order that it validate subst, it is nearly constrained to interpret ≡ as equality on the domain of discourse.
Independence of \texttt{ALLdistr} \ One can prove the scheme $\forall x (\varphi \rightarrow \psi) \& \forall x \varphi \implies \forall x \psi$ from \{\texttt{mp, gen, spec}\}. Therefore, the independence of \texttt{ALLdistr} cannot be proved by simple valuations. We use Monk’s valuation ([Monk65, Thm. 9, Part 4 (C4)]) to prove the independence of \texttt{ALLdistr} in \texttt{TMM \setminus \{spec\}}. Let \textit{val} be the \textit{pc}-valuation evaluating atomic formulas to 1 and such that \textit{val}(\forall v_i (\varphi \rightarrow \psi)) = 1 for all $\varphi, \psi \in \texttt{FM}$ and \textit{val}(\forall v_i \varphi) = \textit{val}(\varphi)$ for all $\varphi \in \texttt{FM}$ that is not an implication. Then, \textit{val} validates all the axiom schemes but \texttt{spec} and \texttt{ALLdistr}; in the latter, take for instance $x \leftarrow v_0, \varphi \leftarrow v_0 \equiv v_0, \psi \leftarrow \neg v_0 \equiv v_0$.

We also prove the independence of \texttt{ALLdistr} in \texttt{TMM \setminus \{vacGen, subst\}} as follows. Recall that a neighborhood model for a classical non-normal modal logic is given by a set of worlds $W$ with a neighborhood function $N: W \rightarrow \mathcal{P}(w)$ and a valuation $\models$ satisfying standard conditions, in particular those of classical propositional calculus and $w \models \Box \varphi$ if and only if $\{v \in W \mid v \models \varphi\} \in N(w)$. After translating first-order formulas into modal ones as usual (map universal quantifiers to necessity), consider a neighborhood model $(W, N)$ where $W = \{w_1, w_2, w_3\}$ and $N(w_1) = \{\{w_1\}, \{w_1, w_2\}, W\}$ and $N(w_2) = \{\{w_1, w_2\}, \{w_2, w_3\}, W\}$ and $N(w_3) = \{\{w_3\}, \{w_1, w_3\}, \{w_2, w_3\}, W\}$. Consider a valuation such that $\varphi$ is false exactly at $w_3$ and $\psi$ is true exactly at $w_2$. Then, $(\varphi \rightarrow \psi)$ is true exactly at $\{w_2, w_3\}$, so $w_2 \models \forall x (\varphi \rightarrow \psi)$ and $w_2 \models \forall x \varphi$ but $w_2 \not\models \forall x \psi$. On the other hand, $W$ is a neighborhood of each world, so the model validates the rule of generalization, and the model is such that $w \in U$ for all $w \in W$ and $U \in N(w)$, so it validates \texttt{spec}, and is such that $U \not\in N(v)$ implies $\{w \in W \mid U \not\in N(w)\} \subseteq N(v)$, so it validates \texttt{modal5}. Finally, take this valuation to validate every equality everywhere.

Independence of \texttt{spec} \ The axiom scheme \texttt{spec} is object-provable from \texttt{T} (since \texttt{T} is object-complete) but independent in \texttt{TMM}. We actually prove object-independence in \texttt{TMM} when the language has a non-nullary nonlogical predicate (but no associated predicate axiom scheme). This implies independence when the language has a non-nullary nonlogical predicate, hence also when it does not by Remark \ref{prop:noncomplete}. Consider a model of first-order logic without equality, with domain $D = \{a, b\}$ and interpret $\equiv$ as the total relation (that is, its graph is $D^2$). Consider a unary predicate $P$ which is interpreted to true on $a$ and false on $b$. Finally, interpret “$\forall v_i$” as “$\forall v_i \in \{a\}$” (more precisely, modify the standard interpretation of universally quantified formulas accordingly). Then, all axiom schemes of \texttt{TMM} are true except \texttt{spec}: its instance $\forall v_0 P(v_0) \rightarrow P(v_0)$ is false since the $v_0$ in the consequent can be assigned to $b$.

To rephrase this example, we can say that we have a domain of quantification $\{a\}$ which is strictly included in the domain of discourse $D$, so that specialization does not hold. In order that \texttt{denot} hold, we need that all elements in the domain of discourse be equal to an element in the domain of quantification, which is why we interpreted $\equiv$ as the total relation. Note that \texttt{ax-12} is not true in this model (this is necessary, since one can prove specialization from \texttt{T} and \texttt{ax-12}), although \texttt{subst} is.

In the next section, we give another proof of the independence of \texttt{spec}, and even of its formula-metavariable-free instance $\forall xx \equiv y \rightarrow x \equiv y$, from the other axiom schemes of \texttt{TMM}.

Independence of \texttt{modal5} \ The axiom scheme \texttt{modal5} is object-provable from \texttt{T} (since \texttt{T} is object-complete) but independent in \texttt{TMM \setminus \{vacGen, subst\}}, as a standard Kripke model (see for instance [BRV01]) shows: consider two worlds, $A$ and $B$, with a reflexive accessibility relation which furthermore connects $A$ to $B$. Interpret equality as always true and “$\exists x_i$” for any $i \in \omega$ as necessity. Introduce a predicate $P$ which is true in $A$ but not in $B$. Then, $A \not\models \exists x P \rightarrow \forall x \exists x P$, so \texttt{modal5} does not hold, but the other axioms in \texttt{TMM \setminus \{vacGen, subst\}} do. We leave the question of independence in \texttt{TMM} open.

3.1.3 Vacuous generalization

The axiom scheme \texttt{vacGen} is independent in \texttt{TMM} by Proposition \ref{prop:noncomplete}. Equivalently, declare a scheme to be *-true if the scheme obtained from it by ignoring its DV conditions involving formula metavariables is true. Then, \texttt{vacGen} is the only axiom scheme in \texttt{TMM} which is not *-true. We leave the question of object-independence in \texttt{TMM} open.
3.1.4 Equality bloc

We consider the following models of first-order logic without equality:

- **EQrefl**: a non-empty domain, equality is interpreted as the empty relation (so \( v_i \equiv v_j \) is always evaluated to false).
- **EQsymm**: domain \( \{0, 1\} \), the graph of equality is \( \{0, 1\}^2 \setminus \{(1, 0)\} \).
- **EQtrans**: domain \( \{0, 1, 2\} \), the graph of equality is \( \{0, 1, 2\}^2 \setminus \{(0, 2), (2, 0)\} \).

3.1.5 Denotation axiom scheme

Let \( \text{val} \) be the pc\(_=\)-valuation that ignores quantifiers. Then, all axiom schemes are validated except for \( \text{denot} \).

3.1.6 Substitution axiom scheme

The model used in [Mon65, Thm. 9, Part 8 (C8)] proves the independence of \( \text{subst} \) in a system related to \( \text{TMM} \). We specialize that model to prove independence in \( \text{TMM} \), even when the latter is augmented with the axiom schemes \( \text{OneObj} \) and \( \text{gen}_P \) (see Appendix D for these names).

The language consists of one nonlogical predicate, say \( P \), which is unary. The domain of discourse has three objects, say 0, 1, 2. Equality is interpreted as the equivalence relation with equivalence classes \( \{0, 1\} \) and \( \{2\} \). The predicate \( P \) holds for exactly 0.

Having more than one equivalence classes for \( \equiv \) (i.e., having unequal objects) makes the formula \( \forall v_i v_i \equiv v_j \) true exactly when \( i = j \), making the verifications easier.

3.1.7 Predicate axiom schemes

Suppose that the language has \( n \) nonlogical predicate symbols, \( \mathcal{L} = \{P_1, \ldots, P_n\} \) with \( P_i \) of arity \( a_i \). There is no loss of generality since in any proof can occur only a finite number of predicates. The corresponding \( \sum_{i=1}^{n} a_i \) predicate axiom schemes have the form

\[
y \equiv z \to (P_i(x_1, \ldots, y, \ldots, x_{a_i}) \to P_i(x_1, \ldots, z, \ldots, x_{a_i})).
\]

where \( y \) and \( z \) are at the \( j \)th position, for \( 1 \leq i \leq n \) and \( 1 \leq j \leq a_i \).

Fix \( i \) and \( j \) as above. We consider the following model of first-order logic without equality. The domain has two elements, say \( D = \{0, 1\} \). The interpretation (the graph) of equality is \( D^2 \), that is, \( v_i \equiv v_j \) always evaluates to true. The interpretations of the predicates \( P_k \) for \( k \neq i \) are empty, and the interpretation of the predicate \( P_i \) is \( D^{j-1} \times \{0\} \times D^{n_i-j} \). Then, all the predicate axiom schemes evaluate to true, except \( \text{ax}-P_{ij} \).

3.2 Supertruth and partial independence of \( \text{ALLcomm} \)

Since the system \( T \) is object-complete, proving independence of the axiom schemes in \( \text{TMM} \setminus T \) is generally a harder task. In the cases of \( \text{spec} \) and \( \text{subst} \), this was done by adding nonlogical predicates to the language without the associated predicate axiom schemes. The method presented here is different, and can prove, for instance, that even some true formula-metavariable-free schemes, like the instance \( \forall x x \equiv y \to x \equiv y \) of \( \text{spec} \), or the instance \( \forall x \forall y \forall z t \to \forall y \forall x z \equiv t \) of \( \text{ALLcomm} \), are not provable from \( T \).

Remark 3.3. Monk’s proof of the independence of \( \text{spec} \) in a related system ([Mon65, Thm. 12, Part 3 (A6)]) does not apply here. Indeed, that system contains only sentences (closed formulas), and for instance \( \text{denot} \) is evaluated to false (its closure is of course evaluated to true). His proof of the independence
of ALLcomm (Part 1 (A4) of the same theorem) relies on the order of variables, because he works in a substitutionless calculus, but that method cannot be applied here, where variable metavariables are interchangeable.

To ease the reading of the proof, we introduce the following definitions and notation. A quantified subformula is a subformula beginning with (and not merely containing) a universal quantifier. If it begins with "∀x;" then we say that it is i-quantified. The operation on metaformulas that consists in replacing every occurrence of \( x_j \) within any \( i \)-quantified subformula (equivalently, within the scope of any quantification over \( x_i \)) with \( x_i \) is called the \((i, j)\)-transform. The \((i, j)\)-transform of a scheme is the scheme resulting from the \((i, j)\)-transforms of its hypotheses and conclusion, with its DV conditions unchanged.

The result of the \((i, j)\)-transform on the metaformula or scheme \( \Phi \) is denoted by \( \Phi^{(i,j)} \). If \( \Phi^{(i,j)} = \Phi \), we say that the \((i, j)\)-transform acts trivially, or is trivial, on \( \Phi \). This is the case for instance when \( \{x_i, x_j\} \notin \mathcal{P}_2(\text{OC}(\Phi)) \). The \((i, j)\)-transform is legitimate on the scheme \( \Phi \) if \( \{x_i, x_j\} \notin \text{DV}(\Phi) \).

Remark 3.4. One can think of an \((i, j)\)-transform as a way to set quantified subformulas to True or False with more freedom than with usual models. In the words of Mario Carneiro, it can be thought of as a "deliberate bound variable capture."

A scheme is supertrue if the following is true of all its instances: if all the legitimate \((i, j)\)-transforms of all the instances of its hypotheses are true, then all the legitimate \((i, j)\)-transforms of all the instances of its conclusion are true.

Remark 3.5. A few remarks are in order:

- When a hypothesis or the conclusion of a scheme is considered as a scheme itself (for example when asserting or asking about its truth or supertruth), then it is understood to be the scheme with no hypotheses, with that formula as conclusion, and with the DV conditions inherited from the scheme. For example, if \( \{\Phi_1, \Phi_2\}, \Phi_0, D \) is a scheme, and we mention "the hypothesis \( \Phi_1 \) as a scheme", then we mean, the scheme \( (\emptyset, \Phi_1, D) \) (by our above convention, we do not write \( D \cap \mathcal{P}_2(\text{OC}(\Phi_1)) \) explicitly).

- Recall that we do include the trivial \((i, j)\)-transforms. In other words, "all legitimate \((i, j)\)-transforms of \( \Phi \)" is the same thing as "\( \Phi \) and all its legitimate \((i, j)\)-transforms".

- The definition may seem to have one unnecessary level of instantiation (namely, two levels instead of one), but this is subtly not the case: the first instantiation is of the whole scheme, while the second is, independently, of the given hypothesis or conclusion. As an example, consider the scheme \( \varphi \Rightarrow \psi \). If we did not have the first instantiation, then the definition would read "if all transforms of all instances of \( \varphi \) are true, then all transforms of all instances of \( \psi \) are true", and this is vacuously true since obviously not all instances of \( \varphi \) are true (\( \bot \) is not true). Therefore, the scheme \( \varphi \Rightarrow \psi \) would be supertrue, clearly something that we do not want. With the correct definition, we are allowed to first instantiate the scheme, and one such instance is \( \top \Rightarrow \psi \). Now, all transforms of all instances of \( \top \) are true, but this is not the case for \( \psi \) (again, \( \bot \) is not true). Therefore, \( \varphi \Rightarrow \psi \) is not supertrue.

\[ \forall \Psi \in \text{Inst}(\Phi) \left( \left( \forall i \in [1,n] \left( \Psi_i, \text{DV}(\Psi) \right) \text{ presuptrue} \right) \Rightarrow \left( \Psi_0, \text{DV}(\Psi) \right) \text{ presuptrue} \right). \]  

\[ (21) \]
• For hypothesis-free schemes, however, one can remove one level of instantiation, since, as seen in Subsection 1.1, an instance of an instance is an instance. Namely, a scheme with no hypotheses is supertrue if and only if all the legitimate \((i, j)\)-transforms of all its instances are true. Combined with the first item, this shows that a scheme is supertrue if and only if all its instances are such that whenever their hypotheses are supertrue, so is their conclusion.

• One could be tempted to simplify the definition to: “a scheme is supertrue if all the legitimate \((i, j)\)-transforms of all its instances are true.” That property implies supertruth, but not conversely: with that definition, the rule of generalization would not be supertrue as the following example due to Mario Carneiro shows: the rule of generalization instantiates to \((x \equiv y \to \varphi) \implies \forall x(x \equiv x \to \varphi)\), which \((x, y)\)-transforms to \((x \equiv y \to \varphi) \implies \forall x(x \equiv x \to \varphi)\), which is not true: consider the instance given by \(\varphi \leftarrow x \equiv y\).

• For any metaformula or scheme, \(\Phi^{x_j \leftarrow x_i} = (\Phi^{i,j})^{x_j \leftarrow x_i}\), so \(\Phi^{x_j \leftarrow x_i}\) is an instance of \(\Phi^{i,j}\).

The \((i, j)\)-transform of a proof (resp. of a set of schemes) is defined as the sequence (resp. set) of the \((i, j)\)-transforms of its lines (resp. elements). When a line of a proof is considered as a scheme (for example when asserting or asking about its truth or supertruth), it is understood to be the scheme with no hypotheses, with that formula as conclusion, and with the DV conditions inherited from the proof (for example, for the line \(P_i\) of the proof \(P\), the DV condition is \(DV(P) \cap P_2(OC(P_i))\)).

The following proposition is analogous to Proposition 1.2.

**Proposition 3.6.** If \(P\) is a proof of a scheme \(\Phi\) from a set \(S\) of schemes and \(i, j \in \omega\), then \(P^{(i,j)}\) is a proof of \(\Phi^{(i,j)}\) from \(S^{(i,j)}\). In particular, a scheme provable from a set of supertrue schemes is supertrue.

*Proof.* Let \(P\) be a proof of the scheme \(\Phi\) from the set \(S\) of schemes and let \(i, j \in \omega\). If the line \(P_k\) is a hypothesis of \(\Phi\), then the line \(P^{(i,j)}_k\) is a hypothesis of \(\Phi^{(i,j)}\). Else, there exist \(k_1, \ldots, k_n < k\) such that \({P_{k_1}, \ldots, P_{k_n}, P_k, DV(P)}\) is an instance of a scheme in \(S\). Then, \({P^{(i,j)}_1, \ldots, P^{(i,j)}_n, P^{(i,j)}_k, DV(P^{(i,j)})}\) is the \((i, j)\)-transform of that instance. Finally, the final line of \(P^{(i,j)}\) is \(\Phi_0^{(i,j)}\).

From this and Proposition 1.2, it follows that a scheme provable from a set of supertrue schemes is supertrue.

**Proposition 3.7.** A true quantifier-free scheme is supertrue.

*Proof.* Let \(\Phi\) be a quantifier-free scheme and let \(\Psi\) be an instance of \(\Phi\). Let \(i, j \in \omega\). Since \(\Phi\) is quantifier-free, the \(i\)-quantified subformulas of \(\Psi\) necessarily come from metavariables \(\varphi_k\) occurring in \(\Phi\). Therefore, \(\Psi^{(i,j)}\) is also an instance of \(\Phi\): if \(\Psi\) was obtained by substituting \(\chi\) for \(\varphi_k\), then \(\Psi^{(i,j)}\) is obtained by substituting \(\chi^{(i,j)}\) for \(\varphi_k\). Therefore, if \(\Psi\) is true, then all its \((i, j)\)-transforms are true. Therefore, if \(\Phi\) is true, then it is supertrue.

**Proposition 3.8.** The rule of generalization \(\text{gen}\) is supertrue.

*Proof.* Let \(\Phi\) be a metaformula and let \(i, j, k \in \omega\). One has \((\forall x_k \Phi)^{(k,j)} = \forall x_k \Phi^{x_j \leftarrow x_k}\). If \(i \neq k\), then \((\forall x_k \Phi)^{(i,j)} = \forall x_k \Phi^{(i,j)}\). Since an instance of \(\text{gen}\) has the form \({\{\Phi\}, \forall x_k \Phi, D}\), the above proves that \(\text{gen}\) is supertrue.

**Proposition 3.9.** The schemes \(\text{ALLdistr, modalD, modal4, modal5, vacGen, denot, subst, genEq}\) are supertrue.

*Proof.* \(\text{ALLdistr, modalD, modal4, modal5}:\) An \(i\)-quantified subformula in an instance of one of these schemes with \(i \neq 0\) comes from within a formula metavariable, so as in the case of quantifier-free schemes, any of its \((i, j)\)-transforms with \(i \neq 0\) is an instance of the scheme. On the other hand, the result of a \((0, j)\)-transform is also an instance, obtained by the substitution \(x_j \leftarrow x_0\).
vacGen: Let $\Phi$ be an instance of vacGen. It is of the form $\Psi \to \forall x_k \Psi$, $\text{DV}(x_k, \text{OC}(\Psi))$. If the $(i,j)$-transform is legitimate on $\Phi$ and nontrivial, then $i \neq k$ because of the DV conditions. Therefore, any $i$-quantified subformula of $\Phi$ is a subformula of $\Psi$. Therefore, the $(i,j)$-transform of $\Phi$ is another instance of vacGen (namely, $(\Psi \to \forall x_k \Psi)^{(i,j)} = (\Psi^{(i,j)} \to \forall x_k \Psi^{(i,j)})$), so is true. Therefore, all combinations of $(i,j)$-transforms of $\Phi$ are true, so vacGen is supertrue.

denot: This scheme has no legitimate nontrivial $(i,j)$-transform. Its strengthening obtained by removing its DV condition has a legitimate nontrivial $(i,j)$-transform, which is $x \equiv x \to \exists xx \equiv x$, which is true.

subt: Let $\Phi$ be an instance of subt. Up to variable renaming, it is characterized by the formula $\Psi$ that is substituted for $\varphi$. All legitimate $(i,j)$-transforms act similarly on both occurrences of $\Psi$, so that the resulting scheme $\Phi^{(i,j)}$ is an instance of subt so is true.

genEq: The only nontrivial $(i,j)$-transforms are the $(x,y)$-transform and the $(x,z)$-transform, and both make an antecedent false.

**Proposition 3.10.** Every scheme provable from $TMM \setminus \{\text{spec, ALLcomm, ALLeq}\}$ is supertrue.

**Proof.** By Proposition 3.6 and the previous three propositions. □

**Proposition 3.11.** The scheme ALLcomm is not supertrue.

**Proof.** Even the formula-metavariable-free instance $\forall x \forall y z \equiv t \to \forall y \forall x z \equiv t$ of ALLcomm is not supertrue. Indeed, its $(x,y)$-transform is $\forall x \forall x z \equiv t \to \forall y \forall x z \equiv t$, which is not true, as its instance $z \leftrightarrow y$ shows. □

**Proposition 3.12.** In the axiom system $TMM \setminus \{\text{spec, ALLeq}\}$, the scheme ALLcomm cannot be weakened by adding the DV condition $\text{DV}(x,y)$; in particular, it is independent.

**Proof.** By the above proposition, it suffices to prove that adding the DV condition $\text{DV}(x,y)$ to ALLcomm makes it supertrue. This is the case since all legitimate $(i,j)$-transforms of all the instances of this weakened version reduce to substitutions, hence yield true schemes. □

### 3.3 Variants of supertruth and partial independence of spec and ALLeq

**Hull-supertruth** A first variant is the following notion: define the hull of any set of schemes to be the smallest set of schemes containing it and closed under instantiation and legitimate $(i,j)$-transforms, and then define a scheme to be hull-supertrue if for all of its instances, if all schemes in the hull of its hypotheses are true, then all the schemes in the hull of its conclusion are true. All the axiom schemes in Proposition 3.9 are hull-supertrue.

**Semisupertruth** We define the $(i,j)$-transform to be the combination of the $(i,j)$-transform and the $(j,i)$-transform. It is legitimate on a scheme when any (hence both) of these is legitimate. Performing these two transforms in different orders gives schemes which are identical up to renaming some bound variables, but for the sake of definiteness, one can ask that the $(i,j)$-transform is performed first if $i < j$. If one simply replaces $(i,j)$-transforms with $(i,j)$-transforms in the definition of supertruth, there are already some differences. For instance, the scheme $\forall xx \equiv y \to \forall y x \equiv y$ ($\text{ax-c1lin}$ in set.mm) is not supertrue but satisfies this new notion.

Define a scheme to be semisupertrue if it is true and the following is true of all its formula-metavariable-free instances: if all the legitimate $(i,j)$-transforms of all the instances of its hypotheses are true, then all the legitimate $(i,j)$-transforms of all the instances of its conclusion are true.

Then, proofs preserve semisupertruth and all axiom schemes in $TMM \setminus \{\text{spec, ALLeq}\}$ are semisupertrue. We check this for the non-supertrue scheme ALLcomm. A formula-metavariable-free instance of ALLcomm is of the form $\forall x \forall y \Phi \to \forall y \forall x \Phi$ (with possibly DV conditions) where $\Phi$ contains no formula metavariables.
Its \( \{i,j\} \)-transforms other than the \( \{x,y\} \)-transform affect both occurrences of \( \Phi \) similarly. Its \( \{x,y\} \)-transform is \( \forall x \forall x \Phi^{x \to y} \to \forall y \forall y \Phi^{x \to y} \), which is true by renaming of the bound variable \( y \) to \( x \).

On the other hand, \( \forall xx \equiv y \to (\forall xz \equiv z \to \forall yz \equiv t) \) is not semisupertrue since its \( \{x,y\} \)-transform is \( \forall xx \equiv x \to (\forall xz \equiv t \to \forall yz \equiv t) \), which is not true as its instance \( z \leftarrow y \) shows. Similarly, \( \forall xx \equiv y \to x \equiv y \) is not semisupertrue since its \( \{x,y\} \)-transform is \( \forall xx \equiv x \to x \equiv y \), which is not true. This proves:

**Proposition 3.13.** The axiom schemes \( \text{spec} \) and \( \text{ALLeq} \) are independent from \( \text{TMM} \setminus \{\text{spec}, \text{ALLeq}\} \).

**Remark 3.14.** In \( \text{TMM} \), one can replace \( \text{ALLeq} \) with its generalization \( \forall y (\forall xx \equiv y \to (\forall x \varphi \to \forall y \varphi)) \), which is semisupertrue, and from which \( \text{ALLeq} \) can be recovered using \( \text{spec} \). In this new system, \( \text{spec} \) is therefore independent.

**Remark 3.15.** Building on the previous remark, we see that any true scheme can be rendered semisupertrue (and even supertrue) by taking its universal generalization over all variable metavariables occurring in it. Since schemes can be recovered from their generalizations via \( \text{spec} \), the axiom scheme \( \text{spec} \) can be proved independent in slight variations of basically all systems.

**Remark 3.16.** In this section, I used supertruth and some of its variants as devices to prove independence results. I do not know whether they are of independent interest. I leave their semantic study (for instance, giving an axiom-schematization for supertrue statements) to future work.

### A An elementary example of an independent but object-provable scheme

In this appendix, we present an example of a Metamath system in which a scheme is independent but all its variable-free instances are provable from a set of schemes. This is only superficially similar to the example in the main part of the article, since here there is no distinction between scheme level and object level.

The expressions are formed from the constants Term, Nat, 0, and \( \prime \) (prime). We denote by \( n \) a variable. We posit that every variable, 0, and the prime of a term are terms. We posit that 0 is a Nat, and that if \( n \) is a Nat, then so is \( n' \). Symbolically, our axiom schemes are:

\[
\begin{align*}
\text{Term} & \; n \\
\text{Term} & \; 0 \\
\text{Term} & \; n' \\
\text{Nat} & \; 0 \\
\text{Nat} & \; n \implies \text{Nat} \; n'
\end{align*}
\]

The first three axioms can be considered as syntactic axiom schemes. In this system, the scheme \( \text{Nat} \; n \) is independent although all its variable-free instances, like \( \text{Nat} \; 0' \) or \( \text{Nat} \; 0'' \), are provable. More formally, the following is a valid Metamath database:

\[
\begin{align*}
\text{Term} & \; n \\
\text{Term} & \; 0 \\
\text{Term} & \; n' \\
\text{Nat} & \; 0 \\
\text{Nat} & \; n \implies \text{Nat} \; n'
\end{align*}
\]

The first three axioms can be considered as syntactic axiom schemes. In this system, the scheme \( \text{Nat} \; n \) is independent although all its variable-free instances, like \( \text{Nat} \; 0' \) or \( \text{Nat} \; 0'' \), are provable. More formally, the following is a valid Metamath database:

\[
\begin{align*}
\text{Term} & \; n \\
\text{Term} & \; 0 \\
\text{Term} & \; n' \\
\text{Nat} & \; 0 \\
\text{Nat} & \; n \implies \text{Nat} \; n'
\end{align*}
\]
The reason why the scheme Nat \( n \) is independent is that the system lacks a sort of induction axiom scheme. This is similar to Robinson's arithmetic Q, where commutativity of addition is not provable although any specific instance of it is.

B Comments and variants of the TMM system

Before mentioning the variants, we first show in Figure 1 various subsets of the set of axiom schemes of TMM and the logics they axiomatize. The abbreviations should be self-explanatory (“minimal implicational calculus,” “paraconsistent calculus,” etc.).

For the subsystems of propositional calculus in that figure, we refer to [Rob68]. Beware that the metalogic has no notion of free and bound variables, so the logics labeled \textit{monadic}, \textit{pure}, and their variants may be weaker than expected. In particular, in \textit{pure\textsubscript{=}}, the symbol \( \equiv \) denotes an equivalence relation, but not necessarily equality: it is equality only once \textit{denot} has been assumed.

Some of the comments below relate the axioms of TMM with the axioms used in the Metamath database
set.mm. The result labeled xxx in that database will typically be denoted by set.mm/xxx. A table of correspondence can be found in Table 1 of Appendix D.

B.1 The propositional calculus bloc

The sole inference rule of the propositional calculus part of all the variants mentioned here is modus ponens mp. A family of variants axiomatizes minimal implicational calculus differently than with the sole scheme minimp (due to Łukasiewicz). One can for instance use the axiomatization SK (as done in set.mm) or BCKW or B’KWSTRUCT. Although these systems are known to be independent, it is not the case anymore for BCKW when one adds the Peirce axiom P, since IBCP ⊢ K. As for SK and B’KWSTRUCT, a computer search for truth tables might answer the question of independence, but some later axioms, when quantifiers are ignored, are instances of the identity axiom I and of K, so counterexamples may be harder to find. The truth table used above to prove independence of minimp also proves independence of S in the axiomatization using SK.

Independently of the above, one can replace the set {peirce, contrap, notelim} with the sole scheme $(\neg \varphi \rightarrow \neg \psi) \rightarrow (\psi \rightarrow \varphi)$ (as done in set.mm) and the independence results are obviously preserved. One can also replace {peirce, notelim} with $\neg\neg \varphi \rightarrow \varphi$, and the independence results are obviously preserved as well.

Tarski used the axiomatization of propositional calculus {syl, clavius, notelim}, and Kalish–Montague and Monk proved their independence in the larger system they study (but their proofs do not apply to TMM since some later axiom schemes are not satisfied by the models they use).

There are many other well-known independent axiomatizations of propositional calculus, but proofs of independence in the larger system TMM have to be found.

The axiomatization we use, inspired by [Rob68], has the advantage of having semantic significance: six of its eight subsystems obtained from using minimp and a combination of the other three axioms correspond to a well-studied logic: (minimal implicational, minimal, implicational, intuitionistic, paraconsistent, classical)-calculus.

B.2 The modal bloc

The three modal axioms of our system are ALLdistr = modalK, spec = modalT, and modal5. Using the language of modal logic, one can replace the bloc31 T5 with DB4 (or TB4). The scheme modalB is not semisupertrue (use the same instance and transform as in the proof for spec), while modalD and modal4 are, so it is independent in the variant of TMM \ {ALLEq} using DB4 instead of T5. On the other hand, modalD is provable from modK \ {EQrefl, denot}. Finally, in modK \ {modalB}, the axiom schemes modal4 and modal5 (which are supertrue) are equivalent, so in any system containing modK \ {modalB}, they are either both provable or both independent.

B.3 The vacuous generalization axiom scheme

vacGen : $\varphi \rightarrow \forall x \varphi$, DV($x, \varphi$)

The scheme vacGen says that one can universally quantify a formula over a variable not occurring in it. Over classical calculus, it is equivalent to its dual $\exists x \varphi \rightarrow \varphi$, DV($x, \varphi$) and to $\exists x \varphi \rightarrow \forall x \varphi$, DV($x, \varphi$) (with the help of spec for the reverse implication).

---

31As is customary in modal logic, we denote by “T5” the conjunction of the modal axioms T and 5, which are denoted here by modalT and modal5 respectively, and similarly for similar expressions.
B.4 The equality bloc

\(\text{EQrefl} : x \equiv x\)
\(\text{EQsymm} : x \equiv y \to y \equiv x\)
\(\text{EQtrans} : x \equiv y \to (y \equiv z \to x \equiv z)\)

In TMM, we used the characterization of an equivalence relation as being a reflexive, transitive, symmetric one. One can also use the characterization: reflexive and (left or right)-Euclidean: both characterizations are equivalent over minimimplcalc. Recall that a binary relation \(\equiv\) is left (resp. right)-Euclidean if it satisfies \(\text{ax}_{-\equiv 2}\) (resp. \(\text{ax}_{-\equiv 1}\) or \(\text{EQeucl}\)) of Table 1 of Appendix D. As their names suggest, these axiom schemes can also be seen as the predicate axiom schemes associated with the predicate \(\equiv\).

The independence proofs are similar, since this bloc is fairly well separated from the rest of the axiom schemes: when \(\text{EQeucl}\) replaces \(\{\text{EQsymm, EQtrans}\}\) in the axiomatization, the model given above for the independence of \(\text{EQrefl}\) (resp. \(\text{EQtrans}\)) proves the independence of \(\text{EQrefl}\) (resp. \(\text{EQeucl}\)).

In the presence of right-Euclideanness, one can weaken the requirement of reflexivity to that of right-seriality, that is, the property expressed by the axiom \(\text{denot}^\prime\). See the next subsection for that axiomatization.

One can weaken \(\text{EQsymm}\) by adding the DV condition \(\text{DV}(x, y)\), and recover the full scheme from it and \(\text{monadic}\). One can also weaken \(\text{EQtrans}\) by adding the DV conditions \(\text{DV}(x, y)\) and \(\text{DV}(x, z)\) (or \(\text{DV}(x, z)\) and \(\text{DV}(y, z)\)), and recover the full scheme from it and \(\text{monadic}\). We do not know if we can fully unbundle \(\text{EQtrans}\), that is, require that all variables be disjoint.

We can also take the universal closures of these axiom schemes, since \(\text{spec}\) and \(\text{gen}\) show that they are equivalent to them.

B.5 The denotation axiom scheme

\(\text{denot} : x \equiv x \to \neg \forall y \neg y \equiv x \land \text{DV}(x, y)\)

Tarski used the axiom scheme \(\text{denot}^\prime : \exists xx \equiv y, \text{DV}(x, y)\) in place\(^{32}\) of \(\text{denot}\). More precisely, Tarski’s and the present axiomatizations correspond respectively to the left and right hand side of the bi-entailment \(\{\text{EQeucl, denot}^\prime\} + \text{modK} \vdash \{\text{EQrefl, EQsymm, EQtrans, denot}\}\).

We chose our axiom schemes in order to better separate the equality bloc from the rest of the axiom schemes. In \(\text{denot}\), the addition of the antecedent \(x \equiv x\) could be seen as a cheap trick to guarantee independence of \(\text{EQrefl}\), but more interestingly, \(\text{denot}\) can be seen as existential generalization applied to the unary predicate \(\cdot \equiv x\). It is also related to the proposed view in free logic that an object exists (equivalently, denotes) if and only if it is equal to itself.

As noted above, \(\text{modK} \cup \{\text{EQrefl, denot}\}\) proves \(\text{modalD}\), so if one wanted to use for the modal bloc the axiomatization \(\text{DB4}\) and keep independence of \(\text{modalD}\), then one could weaken \(\text{denot}\) to its generalization \(\forall x (x = x \to \exists y yy = x), \text{DV}(x, y)\).

Finally, one could take as axiom the contrapositive \(\forall y \neg y \equiv x \to \neg x \equiv x\), which can be seen as universal instantiation for the predicate \(\neg \cdot \equiv x\). Then, independence of \(\text{contrap}\) could be simply proved by considering the \(\text{imp}_\equiv\)-valuation which ignores quantifiers and is always false on negations.

B.6 The substitution axiom scheme

\(\text{subst} : \forall x (x \equiv y \to (\varphi \to \forall x (x \equiv y \to \varphi))), \text{DV}(x, y), \text{DV}(y, \varphi)\)

\(^{32}\) Tarski used it both with and without the DV condition. For a logic with terms, the corresponding axiom scheme should be \(\exists xx \equiv t, \text{DV}(x, t)\) (see [KM65, § 4]) and there, the DV condition is necessary (else one could apply the substitution \(t \leftarrow \{x\}\) in a well-founded set theory). This gives a motivation for keeping the DV condition in the term-less case, and proving that it is sufficient.
One can replace the axiom schemes \texttt{spec} and \texttt{subst} by a version of \texttt{subst} without its initial quantifier, call it \texttt{ax12v2} as in \texttt{set.mm}, which also contains the closely related \texttt{ax12v} and \texttt{ax-12}. One has \{\texttt{spec,subst}\} \vdash \texttt{ax12v} \vdash \texttt{ax12v2} \vdash \texttt{ax-12} (with converse entailment when \texttt{genEq} or \texttt{ax-13} is added). The schemes \texttt{ax12v}, \texttt{ax12v2}, \texttt{ax-12} are independent in the new systems, by the same proof of Monk. Independence of \texttt{gen} is also much easier to show in these systems: use the pc-valuation such that val(\forall x\Phi) = 0 if and only if \( x \in OC(\Phi) \), as in [KM65].

The scheme \texttt{ax12v} (as well as \texttt{ax12v2} and \texttt{ax-12}) is not supertrue, as can be seen directly or because \texttt{spec}, which is not supertrue, is implied by it over the supertrue set \( T \).

\section*{B.7 The “quantification over equal variables” axiom scheme}
\texttt{ALLEq} : \forall xx \equiv y \to (\forall x\varphi \to \forall y\varphi)

This axiom (\texttt{set.mm/ax-c11}) was suggested to me by Norman Megill to add it to TMM, which otherwise is probably not complete. In \texttt{set.mm}, it is implied by \texttt{set.mm/ax-12}, which was recognized by several to be “too strong” since it conveys the content of \texttt{ALLEq} in addition to the substitution property conveyed by \texttt{subst}.

This axiom says that one can indifferently quantify over any of two “always equal variables,” and this phrase is in turn justified by the result \texttt{set.mm/ax-c11n}: \forall xx \equiv y \rightarrow \forall yy \equiv z showing the symmetry of that relation and provable from mod\( K \cup \{\texttt{EQsymm,ALLEq}\} \) (while its reflexivity is provable from \{\texttt{ax-gen,EQrefl}\} and its transitivity is provable from mod\( K \cup \{\texttt{EQtrans,ALLEq}\} \)).

\section*{B.8 The generalized equality axiom scheme}
\texttt{genEq} : \forall xx \equiv y \rightarrow (\forall xz \equiv z \to (y \equiv z \to \forall xy \equiv z))

Informally, \texttt{genEq} says that one can universally quantify an equality over a variable which does not occur in that equality. However, it is not a consequence of \texttt{vacGen}, because this non-occurrence is expressed by the antecedents (on a domain with at least two elements, the formula \( \forall u_0v_0 \equiv v_1 \) is false), and not by a DV condition as in \texttt{vacGen}.

The axiom scheme \texttt{genEq} used to be an axiom scheme of \texttt{set.mm} but it got replaced by \texttt{ax-13} (see Appendix D). One has \{\texttt{spec,genEq}\} \vdash \texttt{ax-13} \vdash \texttt{genEq}. Furthermore, the independence proofs given above still work for that system.

The scheme \texttt{ax-13} is not semisupertrue (make the \{\( x,y \)\}-transform or the \{\( x,z \)\}-transform), so in the \texttt{set.mm}-variant of TMM, our results prove the independence of \texttt{ax-13} from \{\texttt{mp, gen, ax-1, \ldots, ax-11}\}.

\section*{B.9 The predicate axiom schemes}
\texttt{ax-\texttt{Pij}} : \texttt{P\textit{i}(z_1, \ldots, x, \ldots, z_A) \to P\textit{i}(y_1, \ldots, z\alpha_1)}

Theses axiom schemes ensure the substitutivity property of the predicate \( P_i \) with respect to each of its \( a_i \) variables. One can add to each predicate axiom scheme the DV condition \( DV(x,y) \) on the first two variables. One can also combine the \( a_i \) predicate axiom schemes associated with an \( a_{ij} \)-ary predicate, obtaining for example in the case of the binary infix operator \( \in \) the axiom \( x_0 \equiv x_1 \to (y_0 \equiv y_1 \to (x_0 \equiv y_0 \to x_1 \equiv y_1)) \).

\section{C Mario Carneiro’s proof of the independence of the rule of generalization}
Here is a proof of the independence (but not object-independence) of the rule of generalization due to Mario Carneiro. It is simpler than the proof in the main part and gives additional insight to the concept
of supertruth.

Fix a model of first-order logic and let \( a \in D \) be a fixed element of the domain of discourse. Let \( P \) be a unary predicate and interpret it as being true at \( a \) and false at at least one element of \( D \). Define a formula as being \(*\)-true if it is true in that model as soon as \( v_0 \) is assigned to \( a \). Since \(*\)-truth is a weaker notion than truth, all hypothesis free axioms are \(*\)-true. It is not hard to verify that modus ponens \( mp \) preserves \(*\)-truth. However, the rule of generalization \( gen \) does not preserve \(*\)-truth. For instance, the formula \( P(v_0) \) is \(*\)-true (since \( v_0 \) has to be assigned to \( a \), where \( P \) is true), but the formula \( \forall v_0 P(v_0) \) is not (here, no variable need be assigned, since this is a closed formula).

Equivalently, one can rephrase the proof in terms of formula transformation: say that the formula \( \Phi \) is \(*\)-true if the formula \( P(v_0) \rightarrow \Phi \) is true in \( D \).

This proof illustrates the fact that, although it is standard to regard a formula as true when it is a true statement for all assignments of its free variables, there are many other possible interpretations.

### D Table of correspondence of scheme labels

We gather in Table 1 some schemes used in this article and in the Metamath database `set.mm` with their labels.

### References

[BRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic*. Cambridge University Press, 2001.

[Car16] Mario Carneiro. “Models for Metamath”. CICM. 2016. arXiv: 1601.07699.

[Jub21] Benoît Jubin. “On the independence of Robinson’s set of axioms for propositional calculus” (2021). arXiv: 2109.14745.

[KM65] Donald Kalish and Richard M. Montague. “On Tarski’s formalization of predicate logic with identity”. *Archiv für Mathematische Logik und Grundlagenforschung* 7 (1965), pp. 81–101.

[Meg95] Norman D. Megill. “A Finitely Axiomatized Formalization of Predicate Calculus with Equality”. *Notre Dame Journal of Formal Logic* 25 (1995), pp. 435–453. URL: [us.metamath.org/downloads/finiteaxiom.pdf](http://us.metamath.org/downloads/finiteaxiom.pdf).

[MW19] Norman D. Megill and David A. Wheeler. *Metamath, A Computer Language for Mathematical Proofs*. Lulu Press, 2019. URL: [us.metamath.org/downloads/metamath.pdf](http://us.metamath.org/downloads/metamath.pdf).

[Mon65] J. Donald Monk. “Substitutionless predicate logic with identity”. *Archiv für Mathematische Logik und Grundlagenforschung* 7 (1965), pp. 102–121.

[Mon71] J. Donald Monk. “Provability with finitely many variables”. *Proceedings of the American Mathematical Society* 27.2 (1971), pp. 353–358.

[Rob68] T. Thacher Robinson. “Independence of two nice sets of axioms for the propositional calculus”. *The Journal of Symbolic Logic* 33.2 (June 1968), pp. 265–270.

[Tar65] Alfred Tarski. “A simplified formalization of predicate logic with identity”. *Archiv für Mathematische Logik und Grundlagenforschung* 7 (1965), pp. 61–79.

---

33 Equivalently, one could add a term \( t \), to be assigned to \( a \), and define the predicate \( P \) to be “\( \equiv t \).”

34 Or: as soon as at least one variable

35 Therefore, substitution of (individual) variables does not preserve \(*\)-truth, but this is not a problem: only substitution of metavariables has to preserve \(*\)-truth, which is the case here by construction.
| Table 1: Correspondence between some scheme labels | this article | set.mm |
|---------------------------------|--------------|--------|
| $\varphi \land \varphi \rightarrow \psi \rightarrow \varphi$ | mp | ax-mp |
| $\varphi \rightarrow ((\psi \rightarrow \chi) \rightarrow ((\theta \rightarrow \psi) \rightarrow (\chi \rightarrow \tau)) \rightarrow (\psi \rightarrow \tau))$ | minimpl | minimpl |
| $(\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi))$ | syl, B $'$ | imim1 |
| $(\varphi \rightarrow \psi) \rightarrow ((\chi \rightarrow \varphi) \rightarrow (\chi \rightarrow \psi))$ | syl $^*$, B | imim2 |
| $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\psi \rightarrow (\varphi \rightarrow \chi))$ | comm, C | pm2.04 |
| $\varphi \rightarrow (\psi \rightarrow \varphi)$ | simp, K | ax-1 |
| $\varphi \rightarrow \varphi$ | id, I | id |
| $(\varphi \rightarrow (\varphi \rightarrow \psi)) \rightarrow (\varphi \rightarrow \psi)$ | hilbert, W | pm2.43 |
| $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$ | frege, S | ax-2 |
| $((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi$ | peirce, P | peirce |
| $(\varphi \rightarrow \neg \psi) \rightarrow (\psi \rightarrow \neg \varphi)$ | contrap | con2 |
| $(-\varphi \rightarrow \neg \psi) \rightarrow (\psi \rightarrow \varphi)$ | notelim | ax-3 |
| $\neg \varphi \rightarrow (\varphi \rightarrow \psi)$ | notelim | pm2.21 |
| $\varphi \rightarrow (-\varphi \rightarrow \psi)$ | notelim | pm2.24 |
| $(\neg \varphi \rightarrow \varphi) \rightarrow \varphi$ | notelim | pm2.18 |
| $\varphi \rightarrow \neg \neg \varphi \rightarrow \varphi$ | notnotintro | notnot |
| $\neg \neg \varphi \rightarrow \varphi$ | notnotelim | notnottr |
| $\varphi \rightarrow \forall x \varphi$ | gen | ax-gen |
| ALLdistr, modalK | ax-4 |
| spec, modalT | ax-10 |
| modal5 | sp |
| modalD | ax-14 |
| modalB | bj-modal |
| modal4 | bj-modalb |
| hba1 | |
| $\forall x \forall y \varphi \rightarrow \forall y \forall x \varphi$ | ALLcomm | ax-11 |
| $\varphi \rightarrow \forall x \varphi, \ DV(x, \varphi)$ | vacGen | ax-5 |
| $\exists x \varphi \rightarrow \forall x \varphi, \ DV(x, \varphi)$ | ax5ea | |
| $x \equiv x$ | | |
| $x \equiv y \rightarrow y \equiv x$ | | |
| $x \equiv y \rightarrow (y \equiv z \rightarrow x \equiv z)$ | | |
| $x \equiv y \rightarrow (x \equiv z \rightarrow y \equiv z)$ | | |
| $x \equiv y \rightarrow (x \equiv z \rightarrow y \equiv z)$ | | |
| $\neg \forall x \neg \equiv y \equiv y$ | denot $'$ | ax-6 |
| $\neg \forall x \neg \equiv y \equiv y, \ DV(x, y)$ | denot | ax6v |
| $x \equiv x \rightarrow \neg \forall y \equiv x, \ DV(x, y)$ | bj-denot | |
| $x \equiv y \rightarrow (\forall y \equiv x \rightarrow \forall x (x \equiv y \rightarrow \varphi))$ | subst | ax-12 |
| $\forall x (x \equiv y \rightarrow (\varphi \rightarrow \forall x (x \equiv y \rightarrow \varphi)), \ DV(x, y), \ DV(y, \varphi)$ | | |
| $\forall x \forall y \equiv x \rightarrow \forall y \equiv x \equiv x$ | ALLEq | ax-c11n |
| $\forall x \equiv y \rightarrow (\forall x \varphi \rightarrow \forall y \varphi)$ | ax-c11 |
| $\forall x \varphi \equiv y \rightarrow (\varphi \rightarrow \forall x \varphi), \ DV(x, y)$ | ax-c16 |
| $\neg \forall x \equiv y \rightarrow (y \equiv z \rightarrow \forall x \equiv z)$ | | |
| $\neg \forall x \equiv y \rightarrow (\neg \forall x \equiv z \rightarrow (y \equiv z \rightarrow \forall x \equiv z))$ | | |
| $\neg \forall x \equiv x_1 \rightarrow (\cdots \rightarrow (\neg \forall x \equiv x_n \rightarrow (P(x_1, \ldots, x_n) \rightarrow \forall x P(x_1, \ldots, x_n)))) \cdots$ | genEq, gen $^=$ | ax-c13 |
| $\neg \forall x \equiv x_1 \rightarrow (\cdots \rightarrow (\neg \forall x \equiv x_n \rightarrow (P(x_1, \ldots, x_n) \rightarrow \forall x P(x_1, \ldots, x_n))) \cdots$ | gen $P_\sim$ | ax-c14 |
| $x \equiv y \rightarrow (x \equiv z \rightarrow y \equiv z)$ | ax-$c_1$ | ax-8 |
| $x \equiv y \rightarrow (z \equiv x \rightarrow z \equiv y)$ | ax-$c_2$ | ax-9 |
| $x \equiv y \rightarrow (P_i (z_1, \ldots, x, \ldots, z_{a_i}) \rightarrow P_i (z_1, \ldots, y, \ldots, z_{a_i}))$ | ax-$P_i j$ | |