Definitions in Metamath

Mario Carneiro

Carnegie Mellon University, Pittsburgh PA, USA

Abstract. Definitions are great.

Keywords: Metamath · Model theory · formal proof · consistency · ZFC · Mathematical logic

1 Introduction

Metamath is a proof language, developed in 1992, on the principle of minimizing the foundational logic to as little as possible [2]. From the beginning, definitions have been treated as the same thing as axioms, which yields an extremely simple structure for verifiers. However, this standpoint is undesirable when we want to assert that a certain small set of axioms leads to a given theorem, because each new definition requires one or more axioms along with it, which leak through to the final theorem, even though these axioms are in principle eliminable.

This paper builds on [1], which defines the model theory of Metamath (we borrow notation and definitions directly from that paper, so readers are encouraged to consult it). In this work, we develop a framework for discussing conservative extensions of a formal system.

Modifications from [1] In [1], the sets TC and VT (typecodes and variable typecodes) were inferred from the axioms and variable definitions, respectively. In this paper, we will assume instead that these sets are given separately as part of the definition of a formal system $\langle \mathsf{CN}, \mathsf{VR}, \mathrm{Type}, \mathsf{VT}, \mathsf{TC}, \Gamma \rangle$, subject to the restrictions $\mathsf{VT} \subseteq \mathsf{TC} \subseteq \mathsf{CN}$, $\mathrm{Type} : \mathsf{VR} \to \mathsf{VT}$, and $\mathsf{EX}_C = \{e \in \mathsf{EX} \mid (|e| > 0 \wedge e_0 \in \mathsf{TC})\}$. This does not change any of the results of [1].

We will also assume that η is surjective on U, that is, for each $c \in \mathsf{TC}, v \in \mathsf{U}_c$ there is some $\mu \in \mathsf{VL}$ and $e \in \mathsf{EX}_C$ such that $\eta_\mu(e)$ is defined and equals v. For $c \in \mathsf{VT}$ this is trivially true, and in grammatical formal systems it is also true, but in general it must be added as an extra requirement.

Extra notation from [1] We will use some extra notation for the definitions in [1].

- Formal systems are usually denoted by $T = \langle \mathsf{CN}, \mathsf{VR}, \mathrm{Type}, \mathsf{VT}, \mathsf{TC}, \Gamma \rangle$; in each case we will assume that all the definitions associated to T such as $\mathsf{DV}, \mathsf{EX}_C$ etc. are available.

- When multiple theories are in use these will be disambiguated by primes or sub/superscripts as in TC' or TC^T.
- Similarly, models are denoted by $M = \langle \mathsf{U}, \#, \eta \rangle$ with associated definitions such as VL.
- The notation $M \models T$ means that M is a model for T.
- The statement "$\eta_\mu(e)$ is defined", which means that e is true in the model relative to the valuation μ, is written $M \models_\mu e$.
- The notation $M \models e$ means that $M \models_\mu e$ for all $\mu \in \mathsf{VL}$.
- The notation $M \models \langle D, H, A \rangle$ means that for every $\mu \in \mathsf{VL}$, if $\mu(\alpha) \# \mu(\beta)$ for all $\{\alpha, \beta\} \in D$ and $M \models_\mu H$, then $M \models_\mu A$. (The previous definition is a special case of this, $M \models e$ iff $M \models \langle \emptyset, \emptyset, e \rangle$.)

2 The category of models of a formal system

We begin with a definition of model homomorphism:

Definition 1. *Given two models $M, M' \models T$ for the same formal system, we say that $f : M \to M'$ is a model homomorphism if f is a family of functions $f_c : \mathsf{U}_c \to \mathsf{U}'_c$ for $c \in \mathsf{TC}$ such that:*

- *For $v \in \mathsf{U}_c, w \in \mathsf{U}_{c'}$, if $v \# w$, then $f_c(v) \#' f_{c'}(w)$.*
- *For each $\mu \in \mathsf{VL}$ and $e \in \mathsf{EX}_C$, if $\eta_\mu(e)$ is defined then $\eta'_{f(\mu)}(e) = f_{\mathrm{Type}(e)}(\eta_\mu(e))$, where $f(\mu) \in \mathsf{VL}'$ is defined by $f(\mu)(v) = f_{\mathrm{Type}(v)}(\mu(v))$ for each $v \in \mathsf{VR}$.*

We say that f is injective (surjective) if each f_c is injective (surjective).

From this, we immediately get a derived notion of the category $\mathbf{Mdl}(T)$ of all models of a formal system T, as well as the category $\mathbf{GMdl}(T)$ of models of a grammatical formal system (which differs only in that models of a grammatical system must respect the Syn function as well; the morphisms are defined the same as in the general case). All of the constructions in this section apply with $\mathbf{GMdl}(T)$ in place of $\mathbf{Mdl}(T)$.

There is also a preorder on models (which we will treat as a partial order, implicitly taking equivalence classes):

Definition 2. *Given two models $M, M' \models T$ for the same formal system, we say that $M \le M'$, or "M is* stronger *than M'", if $M \models \langle D, H, A \rangle$ implies $M' \models \langle D, H, A \rangle$ for all statements $\langle D, H, A \rangle$.*

The model $\mathbf{1}$ in which $\mathsf{U}_c = \{*\}$ for each c, $* \# *$ is true, and $\eta_\mu(e) = *$ for all μ, e, is a terminal object in this category, and the top of the poset.

Theorem 1. $\mathbf{Mdl}(T)$ *has arbitrary products, and furthermore $\forall i\ N \le M_i$ implies $N \le M$ (one direction of the lattice meet property is valid).*

Proof. Let $M = \prod_{i \in I} M_i$ be the model defined by $\mathsf{U}_c = \prod_{i \in I} \mathsf{U}^i_c$, $v \# w$ if $\forall i \in I, v_i \#_i w_i$, and $(\eta_\mu(e))_i = \eta^i_{\mu_i}(e)$, where μ_i is defined by $\mu_i(v) = (\mu(v))_i$,

and $\eta_\mu(e)$ is defined only if each component is defined. The construction of $\mathbf{1}$ above is a special case of this for $I = \emptyset$.

Now suppose that $M_i \models \langle D, H, A\rangle$ for each i, and let $\mu \in \mathsf{VL}$, $\mu(\alpha) \# \mu(\beta)$ for all $\{\alpha, \beta\} \in D$, and $M \models_\mu H$. Then $\mu(\alpha) \# \mu(\beta)$ implies $\mu(\alpha) \#_i \mu(\beta)$ for all i, and $M \models_\mu H$ (i.e. $\eta_\mu(e)$ is defined for all $e \in H$) implies that $M \models_{\mu_i} H$ for each i. Thus $M \models_{\mu_i} A$ for each i, so $M \models_\mu A$. □

Theorem 2. *Given $f, g : X \to Y$, where $X, Y \in \mathbf{Mdl}(T)$, if Y is compatible with the equivalence relation (see below), then there is a coequalizer object Q with morphism $q : Y \to Q$, and furthermore, $Q \leq Y$.*

Proof. Q and $q : Y \to Q$ are defined such that:

- U_c^Q is the quotient of U_c^Y by the smallest equivalence relation $\sim_c$ such that for each $e \in \mathsf{EX}_C$ with $\mathrm{Type}(e) = c$:
  - For each $\mu \in \mathsf{VL}^X$, $f_c(\eta_\mu^X(e)) \sim_c g_c(\eta_\mu^X(e))$, and
  - For all $\mu, \nu \in \mathsf{VL}^Y$, if $\mu(v) \sim_{\mathrm{Type}(v)} \nu(v)$ for all $v \in \mathcal{V}(e)$, then $\eta_\mu^Y(e) \sim_c \eta_\nu^Y(e)$.
  - Y is *compatible with the equivalence relation* if $\eta_\mu^Y(e)$ and $\eta_\nu^Y(e)$ in the previous clause are either both defined or neither defined.
- The morphism q is the canonical map $q_c : \mathsf{U}_c^Y \to \mathsf{U}_c^Y / \sim_c$.
- The freshness relation is given by $q_c(v) \#^Q q_{c'}(w)$ if for some $v' \sim_c v$ and $w' \sim_{c'} w$, we have $v' \#^Y w'$.
- The interpretation function $\eta_\mu^Q(e)$ is defined by $\eta_{q(\mu)}^Q(e) = q_{\mathrm{Type}(e)}(\eta_\mu^Y(e))$ (where $e \in \mathsf{EX}_C$ and $\mu \in \mathsf{VL}^Y$). The second part of the definition of $\sim_c$ ensures that this is well-defined.

Suppose that $Q \models \langle D, H, A\rangle$, and let $\mu \in \mathsf{VL}^Y$, $\mu(\alpha) \#^Y \mu(\beta)$ for all $\{\alpha, \beta\} \in D$, and $Y \models_\mu H$. Then $q(\mu(\alpha)) \#^Q q(\mu(\beta))$, and $Q \models_{q(\mu)} H$ because for each $e \in H$, $\eta_{q(\mu)}^Q(e) = q_{\mathrm{Type}(e)}(\eta_\mu^Y(e))$ is defined. Thus $Q \models_{q(\mu)} A$, so $\eta_\mu^Y(A)$ is defined, which means $Y \models_\mu A$. Thus $Q \leq Y$. □

Theorem 3. $\mathbf{Mdl}(T)$ *does not necessarily have an initial object.*

Proof. Consider the empty formal system, where $\Gamma = \emptyset$. (We can denote this by $T = \emptyset$, in slight abuse of notation, because the other components CN, VR, VT, TC, Type are all still there.) In this case, there is a model in which η_μ is only defined for variables, with all composite expressions being undefined (false in the model), and $v \# w$ is always true. Then the only constraint on M is the choice of U subject to the requirement that $\mathsf{U}_c \neq \emptyset$ for $c \in \mathsf{VT}$, and the constraints on morphisms are trivialized, so the subcategory of models with these trivial $\#, \eta$ functions is naturally isomorphic to the category $\prod_{c \in \mathsf{TC}} \mathbf{Set}$ of TC-indexed set families and functions between them.

It is clear by considering this subcategory that there is not necessarily an initial object, because if VT is nonempty then there is no equivalent of the empty set; concretely, with T defined such that $\mathsf{VT} = \mathsf{TC} = \mathsf{CN} = \{\mathtt{A}\}$ and $\mathsf{VR} = \{x\}$, any model M of T must have some $v \in \mathsf{U}_\mathtt{A}$, so there are at least two morphisms into M' with $\mathsf{U}'_\mathtt{A} = \{a, b\}$, mapping $f(v) = a$ and $f'(v) = b$. Thus $\mathbf{Mdl}(\emptyset)$ has no initial object. □

This also implies that there are not necessarily equalizers, since the equalizer of functions f and f' would be an initial object (if v is the only member of $\mathsf{U}_{\mathtt{A}}$).

Theorem 4. $\mathbf{Mdl}(T)$ *does not necessarily have coproducts.*

Proof. Again, the freshness constraint is the barrier to this property. Consider the same example as in Theorem 3, but allowing # to be sometimes false. Since model homomorphisms go from models with a smaller # relation to a larger one, we need # to be as small as possible given the injection morphisms to the coproduct. But often the minimal such relation will not actually be a freshness relation.

For example, take M_1 to have base set $\{a, b\}$ and M_2 to be $\{c, d\}$ (with only one typecode), and suppose that $\#_1$ and $\#_2$ are always true. The coproduct M must contain at least the elements $\{a, b, c, d\}$, and we wish to know if e.g. $a \# c$ or not. If these four are the only elements in M, then by the freshness constraint at least one of the four must be fresh for all of them, so that if a is fresh from everything we have at most $\neg(b \# \{c, d\})$ (with all other pairs fresh). But this choice is not unique, and we could have $\neg(a \# \{c, d\})$ instead; but then, being minimal, M would need all of these, $\neg(\{a, b\} \# \{c, d\})$, a contradiction (since now no element is fresh for all of them).

Thus there is some other element $e \# \{a, b, c, d\}$. But since we are assuming there are no axioms, nothing else constrains the value of e, that is, the canonical injections are insufficient to determine a morphism from the coproduct. Thus the coproduct cannot exist. □

Theorem 5. *If T is a grammatical formal system, then* $\mathbf{GMdl}(T)$ *is a full reflective subcategory of* $\mathbf{Mdl}(T)$.

Proof. Recall that a model for a grammatical formal system, $M \in \mathbf{GMdl}(T)$, is a model in the sense of general formal systems $M \in \mathbf{Mdl}(T)$, subject to the additional constraints:

- $\mathsf{U}_c \subseteq \mathsf{U}_{\mathrm{Syn}(c)}$
- $v \# \langle c, w\rangle \leftrightarrow v \# \langle \mathrm{Syn}(c), w\rangle$
- $\eta_\mu(e) = \eta_\mu(\mathrm{Syn}(e))$ if $\eta_\mu(\mathrm{Syn}(e)) \in \mathsf{U}_c$, else undefined.

Since the set of homomorphisms is unchanged in $\mathbf{GMdl}(T)$, this is a full subcategory. To show it is reflective, suppose $M \in \mathbf{Mdl}(T)$; we will construct $G \in \mathbf{GMdl}(T)$ and $a : M \to G$ such that any other $a' : M \to G'$ factors uniquely through a.

Each $f \in \mathsf{U}_c^G$ will be a partial function on $\mathrm{Syn}^{-1}\{c\}$, defined at c, such that either $f(d)$ is undefined or $f(d) \in \mathsf{U}_d$. Specifically, for $c \in \mathsf{VT}$, U_c^G is the set, over each $\mu \in \mathsf{VL}$ and $e \in \mathsf{EX}_C$ such that $\mathrm{Type}(e) = c$ and $\eta_\mu(e)$ is defined, of the partial function f defined by $f(d) = \eta_\mu([d/e_0]e)$ for $d \in \mathrm{Syn}^{-1}\{c\}$, where $[d/e_0]e \in \mathsf{EX}_C$ denotes the expression formed by setting the typecode e_0 of e to d.

For $d \in \mathsf{TC} \setminus \mathsf{VT}$, we define U_d^G as the subset of $f \in \mathsf{U}_c^G$ (where $c = \mathrm{Syn}(d)$) such that $f(d)$ is defined.

Define $f \#^G g$ if $f(c) \# g(d)$ for some c, d in the domains of f, g resp. For $e \in \mathsf{EX}_C, \mu \in \mathsf{VL}^G$ with $\mathrm{Type}(e) = c \in \mathsf{VT}$, let $\eta^G_\mu(e)$ be the partial function $\eta^G_\mu(e)(d) = \eta_\nu([d/e_0]e)$, where $\nu(v) = \mu(v)(\mathrm{Type}(v))$ (which is defined because $\mathrm{Type}(v) \in \mathsf{VT}$). The value of $\#^G$ and η^G on TC is now forced by the definition, so this defines the model $G \in \mathbf{GMdl}(T)$.

The homomorphism a is defined such that □ finish the proof

Theorem 6. *A subset of a model is a model iff the restricted # relation is still a freshness relation, and the restricted η function satisfies the type correctness law. In particular, the image of a model under a homomorphism is a model.*

Proof. Since all laws other than axiom application and type correctness are equational, they are still satisfied by a restricted η function. Axiom application is still true because it deals with only one η_μ, for μ in the restricted VL.

The image $f(X)$ of a model homomorphism satisfies the freshness condition because any finite set W is the f-image of some set W', and taking $v \# W'$, we have $f(v) \# W$. For type correctness, if $\eta_\mu(e)$ is defined, then $\eta^{f(X)}_{f(\mu)}(e) = f_{\mathrm{Type}(e)}(\eta^X_\mu(e)) \in \mathsf{U}^{f(X)}_{\mathrm{Type}(e)}$. □

Theorem 7. *A monomorphism is an injective morphism, and an epimorphism is a surjective morphism.*

Proof. Suppose we have a monomorphism $f : X \to Y$, and $c \in \mathsf{TC}, x, y \in \mathsf{U}^X_c$ with $f_c(x) = f_c(y)$. Let $g^1, g^2 : \mathbf{1} \to X$, with $g^1_c(*) = x$, $g^2_c(*) = y$, and all other c' set to the same value $g^1_{c'}(*) = g^2_{c'}(*)$. Then $f \circ g^1 = f \circ g^2 \implies g^1 = g^2$, and hence $x = y$.

To show that an epimorphism $f : X \to Y$ is surjective, we can use the coequalizer construction to build the quotient $Y/f(X)$. That is, we take the coequalizer of $\pi_1 \circ \iota, \pi_2 \circ \iota : f(X) \times f(X) \to Y$, where π_1, π_2 are the projections from the product, and $\iota : f(X) \to Y$ is the inclusion. This gives a function $q : Y \to Y/f(X)$. For the other function $g : Y \to Y/f(X)$, we let $g(v) = q(f(x))$ for each $v \in Y$, where x is some fixed element of X.

To verify that g is a morphism: Given $\mu \in \mathsf{VL}$ and $e \in \mathsf{EX}_C$, with $\eta_\mu(e)$ defined, and $c = \mathrm{Type}(e)$, let $\nu \in \mathsf{VL}^X$ such that $\nu(v) = x_{\mathrm{Type}(v)}$. Then $\eta^{Y/f(X)}_{g(\mu)}(e) = q(\eta_{f(\nu)}(e)) = q(f(\eta_\nu(e)))$, and $g(\eta_\mu(e)) = q(f(x))$. Since $f(\eta_\nu(e)) \sim f(x)$ by definition of the quotient $Y/f(X)$, these two are equal.

Now by the epimorphism property, $q \circ f = g \circ f$, and hence $q = g$. Thus given $y \in Y$, $y \sim f(x)$. But $\sim$ is already closed inside $f(X) \times f(X)$ because f is a homomorphism, so $y \in f(X)$. □

For the next result, we will need the category **Fresh**, whose objects are TC-indexed sets with a freshness relation (that is, models without the η component), and homomorphisms, which are just model homomorphisms with the η preservation rule dropped. (Note that **Fresh** implicitly depends on VT and TC – but not Γ or the other components.)

Theorem 8. $\mathbf{Mdl}(T)$ *has free objects, in the sense that the forgetful functor* $U : \mathbf{Mdl}(T) \to \mathbf{Fresh}$ *has a left adjoint.*

Proof. Given $X = \langle \mathsf{U}^X, \#^X \rangle$, let $W_c = \{p \in (\mathsf{CN} \sqcup \bigsqcup \mathsf{U}^X)^{<\omega} \mid p_0 = c\}$, the set of expressions with variables replaced by elements of U^X. We will define first $\#$ and η on W_c, then pare it down to the actual set $\mathsf{U}_c \subseteq W_c$.

- Let $\mathcal{V}(e)$ for $e \in W_c$ denote the elements of $\bigsqcup \mathsf{U}^X$ in e.
- For $e \in W_c, e' \in W_{c'}$, let $e \# e'$ if for all $v \in \mathcal{V}(e), w \in \mathcal{V}(e')$, $v \#^X w$.
- For $e \in \mathsf{EX}_C$, $\mu \in \mathsf{VL}^W$ (where VL^W is the set of valuations where each variable v takes a value from $W_{\mathrm{Type}(v)}$), define $\eta_\mu(e)$ to be the substitution of each variable in e with the string $\mu(v)$; the result will be a member of $W_{\mathrm{Type}(e)}$.

Now, we define each $\mathsf{U}_c \subseteq W_c$, for $c \in \mathsf{TC}$, simultaneously as the minimal set family subject to the constraints:

- For each $v \in \mathsf{U}_c^X$, $\langle c, v \rangle \in \mathsf{U}_c$.
- For each $\mu \in \mathsf{VL}$ (with VL defined in terms of U) and $\langle D, H, A \rangle \in \Gamma$, if
  - $\mu(\alpha) \# \mu(\beta)$ for all $\{\alpha, \beta\} \in D$, and
  - $\eta_\mu(h) \in \mathsf{U}_{\mathrm{Type}(h)}$ for all $h \in H$,

  then $\eta_\mu(A) \in \mathsf{U}_{\mathrm{Type}(A)}$.

With $\#$ and η restricted to U, this gives the desired free model on X. The canonical injection sets $v \mapsto \langle c, v \rangle$ for each $c \in \mathsf{TC}, v \in \mathsf{U}_c^X$. □

Remark 1. Note that for any family $S_c \subseteq W_c$, we can add $S_c \subseteq \mathsf{U}_c$ to the closure conditions for U to get a different model $F(X, S)$, which is no longer free but is instead thought of as "the free model on X generated by S". Roughly speaking, this model satisfies $F(X, S) \models_\mu s$ for each $s \in S_c$ and is universal among such models. ("Roughly" here because s is not an expression but is a member of the model itself, so there are some problems with stating the universal property. But this is the intuition. In many cases such as Theorem 9, s will be closely related to an expression, and then this can be made precise.)

The Gödelian construction of a formal system as a model of itself in [1] is a special case of this construction, where X sets $\mathsf{U}_c^X = \{v \in \mathsf{VR}' \mid \mathrm{Type}(v) = c\}$ (where VR' chosen such that U_c^X is infinite for each $c \in \mathsf{VT}$), and $v \# w$ iff $v \neq w$. But we can improve the result with a different choice of X:

Theorem 9 (Gödel's completeness theorem 2). *Assuming* VR *has infinitely many variables of each type, a theorem* $\langle D, H, A \rangle$ *of a formal system is provable if and only if* $M \models \langle D, H, A \rangle$ *in every model* M*. (This improves on [1] by allowing a nonempty* H *and a non-full* D*.)*

Proof. As in the first proof, the forward direction of both versions of the theorem is trivial by the definition of a model. Let $\mathsf{U}_c = \{v \in \mathsf{VR} \mid \mathrm{Type}(v) = c\}$. For $v, w \in \mathsf{VR}$, let $v \# w$ if either $\{v, w\} \in D$ or $v \neq w$ and one of v, w is not in

$\mathcal{V}(H \cup \{A\})$. (The set $D^* = \{\{v, w\} \mid v \# w\}$ is the largest DV set whose reduct is D.) Finally, let S_c be the set of all expressions in H of type c.

Let $M = F(\langle \mathsf{U}, \# \rangle, S)$ be the model induced by $\mathsf{U}, \#$, and generated by the set S, by Remark 2, and let $I \in \mathsf{VL}$ map each variable to itself (specifically, $I(v) = \mathsf{VH}_v$), so that $\eta_I(e) = e$. Then $I(v) \#^M I(w)$ for all $v, w \in D$, and for each $h \in H$, $\eta_I(h) \in S_{\mathrm{Type}(h)}$, so $M \models_I H$. Thus by the theorem hypothesis, $M \models_I A$, that is, $A \in \mathsf{U}_{\mathrm{Type}(A)}$.

By induction, we claim that for each $c \in \mathsf{TC}$ and $A \in \mathsf{U}_c$, $\langle D^*, H, A \rangle$ is a provable pre-statement, i.e. $A \in C$, the closure of H with respect to D^*. Thus, the reduct of $\langle D^*, H, A \rangle$, which is $\langle D, H, A \rangle$ by construction, is a theorem.

- If $A = \langle c, v \rangle$ for a variable v of type c, then $A = \mathsf{VH}_v \in C$.
- If $A \in S_c$, then $A \in H \subseteq C$.
- Otherwise, by the induction hypothesis we are given $\langle D', H', A' \rangle$ and $\mu \in \mathsf{VL}$, where
  - $\mu(v) \in C$ for each $v \in \mathsf{VR}$,
  - $\mu(\alpha) \# \mu(\beta)$ for all $\{\alpha, \beta\} \in D'$,
  - $\eta_\mu(h) \in C$ for all $h \in H'$,
  - and $A = \eta_\mu(A')$.

  Define the substitution σ such that $\sigma(\mathsf{VH}_v) = \mu(v)$. Then $\sigma(e) = \eta_\mu(e)$, so $\sigma(h) \in C$ for each $h \in H'$ and $\sigma(\mathsf{VH}_v) = \mu(v) \in C$ for each $v \in \mathsf{VR}$, and for each $\{\alpha, \beta\} \in D'$, $\gamma \in \mathcal{V}(\sigma(\langle \alpha \rangle))$ and $\delta \in \mathcal{V}(\sigma(\langle \beta \rangle))$, by definition of $\#$ we have $\{\gamma, \delta\} \in D^*$, so $\sigma(A') = A \in C$.

□

Remark 2. It is unfortunate that Theorem 9 requires infinitely many variables, because this is not the case in any actual Metamath database, which is a finite thing. However, there is an easy compactness result here: One can take an existing database, extend it with infinitely many variables, then prove the theorem and throw away all the unused variables to achieve the proof in a finite formal system. But there is no bound on how many variables will be needed to prove a given theorem (just as there is no bound on the length of the proof).

2.1 Extensions

All of the above work deals with different models over a fixed formal system. We can also consider modification of the underlying theory.

Definition 3. *A formal system T'* extends *another formal system T, denoted $T \leq T'$, if:*

- $\mathsf{CN} \subseteq \mathsf{CN}'$
- $\mathsf{VR} \subseteq \mathsf{VR}'$
- $\mathsf{VT} \subseteq \mathsf{VT}'$
- $\mathsf{TC} \setminus \mathsf{VT} \subseteq \mathsf{TC}' \setminus \mathsf{VT}'$
- $\mathrm{Type} \subseteq \mathrm{Type}'$ *(that is, the functions agree on their common domain)*
- $\Gamma \subseteq \Gamma'$

If T and T' are grammatical formal systems, then we also require $\mathrm{Syn} \subseteq \mathrm{Syn}'$. *If all components are equal except $\Gamma \subseteq \Gamma'$, then this is called an* axiomatic extension*, denoted $T \trianglelefteq T'$.*

Theorem 10. *If $T \leq T'$, then there is a restriction functor $R : \mathbf{Mdl}(T') \to \mathbf{Mdl}(T)$, which is full, and an embedding if $\mathsf{TC} = \mathsf{TC}'$.*

Proof. Given $M \in \mathbf{Mdl}(T')$, let $\mathsf{U}_c^{RM} = \mathsf{U}_c^M$ for $c \in \mathsf{TC}^T$, $\#^{RM}$ be the restriction of $\#^M$, and $\eta_\mu^{RM}(e) = \eta_\mu^M(e)$ for each $e \in \mathsf{EX}_C^T$ and $\mu \in \mathsf{VL}^{RM}$. On morphisms, it just restricts $(f(c))_{c \in \mathsf{TC}'}$ to $(f(c))_{c \in \mathsf{TC}}$.

The fact that this is full is a simple consequence of the fact that it is a restriction on the set families, and it is an embedding when $\mathsf{TC} = \mathsf{TC}'$ because in this case the functor does nothing to objects or morphisms (it is the inclusion map from a subcategory at this point). □

An important question that will come up later is the "extension problem": given $M \in \mathbf{Mdl}(T)$, what can we say about those $M' \in \mathbf{Mdl}(T')$ such that $RM' = M$? Note that by combining the forgetful functor $U : \mathbf{Mdl}(T) \to \mathbf{Fresh}$ with $F : \mathbf{Fresh} \to \mathbf{Mdl}(T')$, we can build the model $RFUM \in \mathbf{Mdl}(T)$. This

Theorem 11. *If $\mathsf{TC} = \mathsf{TC}'$ in the extension $T \leq T'$, then the restriction functor $R : \mathbf{Mdl}(T') \to \mathbf{Mdl}(T)$ has a left adjoint $E : \mathbf{Mdl}(T) \to \mathbf{Mdl}(T')$.*

Proof. Given $M \in \mathbf{Mdl}(T)$, by combining the forgetful functor $U : \mathbf{Mdl}(T) \to \mathbf{Fresh}$ with $F : \mathbf{Fresh} \to \mathbf{Mdl}(T')$, we get $FUM \in \mathbf{Mdl}(T')$. (The assumption $\mathsf{TC} = \mathsf{TC}'$ ensures that **Fresh** is the same for T and T'.) However, we have forgotten too much of the structure of T with the composition. Given any $N \in \mathbf{Mdl}(T')$ and $f : M \to RN$, Given $M \in \mathbf{Mdl}(T')$, let $\mathsf{U}_c^{RM} = \mathsf{U}_c^M$ for $c \in \mathsf{TC}^T$, $\#^{R(M)}$ be the restriction of $\#^M$, and $\eta_\mu^{R(M)}(e) = \eta_\mu^M(e)$ for each $e \in \mathsf{EX}_C^T$ and $\mu \in \mathsf{VL}^{R(M)}$. □

finish the proof

2.2 Conservative and definitional extensions

Now we have the structure we need to start homing in on conservative extensions, from which we can extract a definition for definitions. In conventional logic, a theory T' is a conservative extension of a theory T if the language of T' extends the language of T, and every theorem of T is a theorem of T' (i.e. T' is an extension of T), and every theorem of T' in the language of T is a theorem of T.

In Metamath, the hard part of the above definition is the notion of "language". *A priori* there is no notion of language beyond what is achievable through the applications of the axioms; but of course this is not suitable because if we allowed "in the language of T" to simply mean "a theorem of T", then the only conservative extensions to a formal system are equivalent formal systems (with the same theorems), which trivializes the notion.

We may notice, however, that expressions are strings of constants and variables, and so perhaps we can use the set of constants as our "language". This brings us to the first definition:

Definition 4. *A* conservative extension *of a formal system T is a formal system $T' \geq T$ such that for every theorem $\langle D, H, A\rangle$ of T', if each $e \in H \cup \{A\}$ is a member of EX_c (that is, an expression in $\mathsf{CN} \cup \mathsf{VR}$), then $\langle D, H, A\rangle$ is a theorem of T.*

We can recharacterize conservative extensions categorically:

Theorem 12. *$T \leq T'$ is a conservative extension iff the restriction functor of Theorem 10 is surjective on objects.*

Proof. In the forward direction, given that $T \leq T'$ is conservative, and given some $M \in \mathbf{Mdl}(T)$ □ finish the proof

This is as good as we can do in general formal systems. However, it is not quite as accurate as we would like. For example, if T is a formal system containing the rule $\mathrm{wff}(\varphi]$ (where φ is a wff variable), and T' has rules $\mathrm{wff}(\varphi]$ and $\mathrm{wff}[\varphi)$, then T' is a conservative extension of T. However, in the very similar case where T has $\mathrm{wff}[: \varphi]$ and T' has $\mathrm{wff}[: \varphi]$ and $\mathrm{wff}[\varphi :]$, T' is not a conservative extension.

The problem is that in the second case, the "language" is the same, because both expression builders draw from the same set of constants "[", ":", "]". To resolve this, we must assume that we have a grammatical formal system, in which case we have a framework of syntax expressions that form the "language". This yields the second and preferred definition for a conservative extension in a grammatical formal system.

Definition 5. *A* conservative extension *of a grammatical formal system T is a grammatical formal system $T' \geq T$ such that for every theorem $\langle D, H, A\rangle$ of T', if for each $e \in H \cup \{A\}$, $e_0 \in \mathsf{TC}$ and $\mathrm{Syn}'(e)$ is a theorem of T (that is, e has a valid parse in T), then $\langle D, H, A\rangle$ is a theorem of T.*

Remark 3. Any unambiguous formal system can be mapped to a tree formal system and then back to a string formal system, where the strings are now (reverse) Polish notation with unique constants for each syntax axiom. Thus we can also go backwards from the more expressive Definition 5 to the original Definition 4, under a new representation.

For the rest of this work, we shall restrict our attention to unambiguous formal systems, because there isn't much more to say about the general case.

3 Definitions

Although conservative extension is the goal, it is not the means. Instead, we make use of "definitions", which can be explained as a syntax axiom which is an abbreviation for an expression in the original language. However, there are some subtleties with bound variables here, and it is easiest to give our initial definition in terms of a model.

Definition 6. *Given a model $M = \langle \mathsf{U}, \#, \pi \rangle$ for the unambiguous formal system T (which we will also write $M \models T$), a* conservative model extension *of M is a model $M' = \langle \mathsf{U}, \#, \pi' \rangle$ for an extended unambiguous formal system $T' \geq T$ such that:*

- $\mathsf{TC}' = \mathsf{TC}$
- $\pi_a = \pi'_a$ *for each* $a \in \mathsf{SA}$

The important thing to notice about this definition is that U and $\#$ are unchanged in the extended model, so the only freedom is in choosing π_a for new syntax axioms in $\mathsf{SA}' \setminus \mathsf{SA}$. However, there are no restrictions on new *logical* axioms in T', provided that M' remains a model of it, which corresponds to the idea that if the π_a "abbreviations" are expanded in these new axioms, one gets the same value in the model, and hence the same truth value as well. If one starts with a very strict model like the Gödelian model of a formal system over itself, this is sufficient to ensure that these axioms are mapped to theorems of the original system.

Theorem 13. *Assume VR_T has infinitely many variables of each type. If for every model $M \models T$ there is some model $N \leq M$ and a conservative model extension $N' \models T'$ of N, then T' is a conservative extension of T.*

Proof. By definition, $T' \geq T$. Now suppose $\langle D, H, A \rangle$ is a theorem of T', and for each $e \in H \cup \{A\}$, $\mathrm{Syn}'(e)$ is a theorem of T. By Theorem 9, it suffices to show $M \models \langle D, H, A \rangle$ for every $M \models T$. Let $N' \models T'$ be a conservative model extension of $N \models T$. $N' \models \langle D, H, A \rangle$ means that for every $\mu \in \mathsf{VL}$, if $\mu(\alpha) \# \mu(\beta)$ for all $\{\alpha, \beta\} \in D$ and $N' \models_\mu H$, then $N' \models_\mu A$. Thus we must show $N \models_\mu e \iff N' \models_\mu e$ for $e \in H \cup \{A\}$ in order to establish $N \models \langle D, H, A \rangle$ and thus $M \models \langle D, H, A \rangle$ (since $N \leq M$).

In these cases, we know $\mathrm{Syn}'(e)$ is a theorem of T, and since $\mathsf{TC} = \mathsf{TC}'$ and Syn agrees with Syn' on their common domain, $\mathrm{Syn}' = \mathrm{Syn}$ and thus $\mathrm{Syn}'(e) = \mathrm{Syn}(e)$ is in the language of T, because it is a theorem of T.

Since $\pi_a = \pi'_a$ for each syntax axiom a, by induction all syntax proofs e satisfy $\eta_\mu(e) = \eta'_\mu(e)$. But $\mathrm{Syn}(e)$ is a syntax proof because it begins with a syntax typecode, so $\eta_\mu(\mathrm{Syn}(e)) = \eta'_\mu(\mathrm{Syn}(e))$ and thus $N \models_\mu e \iff N' \models_\mu e$. □

To make this existential more constructive, we would like a function $\mathcal{D}(M)$ which maps each model $M \models T$ to a conservative model extension $\mathcal{D}(M) \models T'$. If $\mathsf{SA} = \mathsf{SA}'$ then there is nothing to show, but if there is some $a \in \mathsf{SA}' \setminus \mathsf{SA}$ we are tasked with inventing a function $\pi_a : \prod_i \mathsf{U}_{\mathrm{Type}(v_i^a)} \to \mathsf{U}_{\mathrm{Type}(a)}$ using the given arbitrary model M, and there aren't too many options for such.

As the base case, we have the projection functions: the function $f(\{v_i\}) = v_k$ is a valid function $\prod_i \mathsf{U}_{c_i} \to \mathsf{U}_{c_k}$. The induction step is the application of a syntax axiom in T, along with multi-composition: if $a \in \mathsf{SA}$ and for each i, $g_i : X \to \mathsf{U}_{\mathrm{Type}(v_i^a)}$, then $f(x) = \pi_a(\{g_i(x)\})$ is a function $X \to \mathsf{U}_{\mathrm{Type}(a)}$.

This rigorously justifies the usage of abbreviational definitions such as $(\varphi \wedge \psi) \leftrightarrow \neg(\varphi \to \neg\psi)$, but it lacks a mechanism for introducing dummy variables, which are essential to set theoretical definitions.

We can use the above induction to define a function $f : \prod_{i \le n} U_{c_i} \to U_{c'}$ which has "too many variables" $n > k$, and pare it down to the real function $\pi_a : \prod_{i \le k} U_{c_i} \to U_{c'}$ (where the syntax axiom a has k variables) by requiring $\pi_a(\{v_i\}_1^k) = f(\{v_i\}_1^n)$ for all $\{v_i\}_1^n$ such that $v_i \# v_j$ whenever $i, k < j$.

This causes a problem in that the uniqueness of π_a here requires the original function f to take exactly the same value at different dummy arguments, which in many models is not true (such as the Gödelian model in which the statement string is the model element). To resolve this, we use the auxiliary model N, which will be a quotient with respect to the appropriate equivalence relation.

Definition 7. *Given a model M, and $x, y \in \bigsqcup U$, say that x and y are* equivalent *and write $x \simeq y$ if M respects the smallest model equivalence relation $\sim$ satisfying $x \sim y$. (That is, $\sim$ is required to be an equivalence relation closed under η application, so if $\mu(v) \sim \nu(v)$ for all v, then $\eta_\mu(e) \sim \eta_\nu(e)$; and M must respect the equivalence, so $\eta_\mu(e)$ is defined iff $\eta_\nu(e)$ is.)*

Theorem 14. *Equivalence is itself an equivalence relation, and it is also respected by M.*

Proof. Symmetry and reflexivity of $\simeq$ are immediate. If $x \simeq y \simeq z$, let $\sim_{xy}, \sim_{yz}$, etc. denote the smallest model equivalence relating the subscripts. We will prove that $\eta_\mu(e)$ is defined iff $\eta_\nu(e)$ is whenever $\mu(v) \sim_{xyz} \nu(v)$. Because $\eta_\mu(e)$ depends on only a finite number of values of μ, we can assume μ and ν differ at all but finitely many positions; and then by changing each value from $\mu(v)$ to $\nu(v)$ we are reduced to the case of proving $\eta_\mu(e)$ is defined iff $\eta_\nu(e)$ is whenever μ and ν differ at only one point v. This latter property we prove by induction on $\sim_{xyz}$.

In the base case we have $\mu(v) = x$, $\nu(v) = y$, or $\mu(v) = y$, $\nu(v) = z$, which are handled by the assumptions on $\sim_{xy}, \sim_{yz}$. Equivalence properties are immediate because the consequent is of the form "$P(x)$ iff $P(y)$". For η application, suppose that a, b are such that for all expressions e and valuations τ, $\eta_{\tau[a]}(e)$ is defined iff $\eta_{\tau[b]}(e)$ is, where $\tau[a] = \tau[v \to a]$ is the valuation τ with $\tau(v)$ set to a. We want to show that $\eta_\mu(e')$ is defined iff $\eta_\nu(e')$ is, where $\mu(v) = \eta_{\tau[a]}(e)$ and $\nu(v) = \eta_{\tau[b]}(e)$.

Define the substitution σ such that $\sigma(v) = e$ and $\sigma(v') = v'$ for $v' \neq v$. Since $\sigma(\mu[a])(v) = \eta_{\tau[a]}(e) = \mu(v)$ and $\sigma(\mu[a])(v') = \eta_{\tau[a]}(v') = \mu(v')$, we have $\sigma(\mu[a]) = \mu$ so that $\eta_{\mu[a]}(\sigma(e)) = \eta_{\sigma(\mu[a])}(e) = \eta_\mu(e)$, and similarly $\eta_{\mu[b]}(\sigma(e)) = \eta_\nu(e)$ (since $\mu[b] = \nu[b]$). By assumption, $\eta_{\mu[a]}(\sigma(e))$ is defined iff $\eta_{\mu[b]}(\sigma(e))$ is, so $\eta_\mu(e)$ is defined iff $\eta_\nu(e)$ is, as we wanted to show.

Then since $\mu(v) \sim_{xyz} \nu(v)$ implies $\mu(v) \sim_{xz} \nu(v)$, we have $x \simeq z$.

If $\mu(v) \simeq \nu(v)$ for each v, and we wish to show $\eta_\mu(e) \simeq \eta_\nu(e)$ (with one side defined iff the other is), as before by transitivity we are reduced to the case of μ, ν differing only at v. Then $\mu(v') \sim_{\mu\nu} \nu(v')$ for all v', by assumption for $v' = v$ and by reflexivity otherwise, so $\eta_\mu(e)$ is defined iff $\eta_\nu(e)$ is. Then since $\sim (\eta_\mu(e), \eta_\nu(e))$ is contained in $\sim_{\mu\nu}$ we immediately have the definedness requirement. □

Definition 8. *For a fixed formal system T, given an expression e and a variable x, and a set of variables V, $\mathrm{NF}_V(x, e)$, read "x is* not free *in the variables V of e", means that for all models $M \models T$ and valuations $\mu, \nu \in \mathsf{VL}$ that differ only at x, if for all $v \in V$ other than x, $\mu(v) \# \mu(x)$ and $\mu(v) \# \nu(x)$, then $\eta_\mu(e) \simeq \eta_\nu(e)$. $\mathrm{NF}(x, e)$ means $\mathrm{NF}_{\mathcal{V}(e)}(x, e)$ and is read "x is* not free *in e".*

We show that all this defines a valid interpretation function in the following theorem:

Theorem 15. *Let T be an unambiguous formal system, and let $\mathsf{SA}' \supseteq \mathsf{SA}$, such that for each $a \in \mathsf{SA}' \setminus \mathsf{SA}$, $P = P(a)$ is a syntax tree of T, $s = s(a)$ is an injective map from each v_i^a to a variable $s(v_i^a) \in \mathsf{VR}$ (which may appear in the leaves of P), and $\mathrm{NF}(x, P)$ for all $x \in \mathsf{VR} \setminus \operatorname{ran} s$.*

Let T' be defined from T with the new SA'. Then T' is a conservative extension of T.

Proof. By Theorem 13, we are given a model $M \models T$ and must create a conservative model extension $N' \models T'$ over some $N \leq M$. The reason we need the extra step of reduction via N is because the definition of π_a we will make forces η to take equal values on the target expression with different dummy variable assignments.

For our choice of N, we take the quotient of M with respect to $\simeq$ using Theorem 2 to get a quotient model $N = M/\simeq \leq M$.

To build the conservative model extension N', we need a definition of π_a for each $a \in \mathsf{SA}' \setminus \mathsf{SA}$. Set $V = \mathcal{V}(P) \cup \operatorname{ran} s$, and define $\pi_a(\{x_i\}) = \eta_\mu(P)$, where μ is any valuation such that $\mu(s(v_i^a)) = x_i$, and $\mu(v) \# \mu(w)$ whenever $v, w \in V$, $v \neq w$, and $w \notin \operatorname{ran} s$.

We first must show that π_a is well-defined. We can find a valuation with the given properties by assigning first $\mu(s(v_i^a)) = x_i$ (which is consistent because s is injective), then repeatedly extending with some w fresh from all previous values until all $\mu(v)$ for $v \in V$ are assigned, then giving any value to the other "unused" variables.

To show uniqueness, first we note that if μ and ν differ only outside V, the dependence on present variables rule for models ensures that $\eta_\mu(P) = \eta_\nu(P)$. Thus we can assume they coincide in this range, and by definition they coincide on $\operatorname{ran} s$ (since $\mu(s(v_i^a)) = \nu(s(v_i^a)) = x_i$). By changing one variable at a time, by transitivity we are reduced to showing $\eta_\mu(P) = \eta_\nu(P)$ where μ, ν differ only at some $x \in V \setminus \operatorname{ran} s$. Given $v \in \mathcal{V}(P)$ different from x, the π_a assumption gives $\mu(v) \# \mu(x)$ and $\mu(v) \# \nu(x)$, so the $\mathrm{NF}(x, P)$ assumption gives $\eta_\mu(P) \simeq \eta_\nu(P)$ in M and $\eta_\mu(P) = \eta_\nu(P)$ in N.

The only property we must check to ensure that N' is a model is freshness substitution: for all $v \in \bigsqcup \mathsf{U}$ and $x_i \in \mathsf{U}_{\mathrm{Type}(v_i^a)}$, $v \# x_i$ for each i, then $v \# \pi_a(\{x_i\})$. As in the proof of existence, we define $\mu(s(v_i^a)) = x_i$, then extend with w fresh from all previous values, but also fresh from v, until all $\mu(v)$ for $v \in V$ are assigned. The resulting $\pi_a(\{x_i\}) = \eta_\mu(P)$ will satisfy $v \# \mu(w)$ for each $w \in V$, and thus $v \# \eta_\mu(P)$ by freshness substitution in T. □

As it stands, Theorem 15 is not very useful, because it does not introduce any axioms besides new "uninterpreted" syntax. However, there is a whole class of possible new axioms that can now be consistently added to T':

Theorem 16. *The formal system and model $N' \models T'$ in Theorem 15 remains a model if T' is extended with $\langle D', H', A' \rangle$, formed from $N' \models \langle D, H, A \rangle$ by replacing, for some substitution σ, an instance of $\sigma(P)$ one of the subtrees of $H \cup \{A\}$ with the syntax tree $a[\{\sigma(s(v_i^a))\}]$, and adding $\{\sigma(v), \sigma(w)\}$ to D' for each $v, w \in V$ with $v \neq w$ and $w \notin \operatorname{ran} s$.*

Proof. We wish to show that $N' \models \langle D', H', A' \rangle$, so let $\mu \in \mathsf{VL}$, and suppose $\mu(\alpha) \# \mu(\beta)$ for all $\{\alpha, \beta\} \in D'$. Since $D' \supseteq D$, we have that $N' \models_\mu H$ implies $N' \models_\mu A$, and want to show $N' \models_\mu H'$ implies $N' \models_\mu A'$. In other words, $\eta_\mu(e) = \eta_\mu(e')$ where e' is obtained from e by the tree substitution in the theorem assumption.

This follows by induction, with the base case being the subtree substitution itself: $\eta_\mu(a[\{\sigma(s(v_i^a))\}]) = \eta_\mu(\sigma(P))$. If $v, w \in V$, $v \neq w$, and $w \notin \operatorname{ran} s$, then $\{\sigma(v), \sigma(w)\} \in D'$, so $\mu(\sigma(v)) \# \mu(\sigma(w))$. Thus $\sigma(\mu)$ is admissible for expanding the definition of π_a, so

$$\begin{aligned} \eta_\mu(\sigma(P)) &= \eta_{\sigma(\mu)}(P) \\ &= \pi_a(\{\sigma(\mu)(s(v_i^a))\}) \\ &= \pi_a(\{\eta_\mu(\sigma(s(v_i^a)))\}) \\ &= \eta_\mu(a[\{\sigma(s(v_i^a))\}]), \end{aligned}$$

as we wanted to show. □

Now we are in a position to use the results. Theorem 16 produces a new $N' \models \langle D', H', A' \rangle$ from $N' \models \langle D, H, A \rangle$, so it can be iterated as many times as necessary to replace multiple substituted subtrees of P in a theorem. The base case is any theorem from T, which is also a theorem of T and hence modeled by N'.

As an example, if we are defining $\vdash V = \{x \mid x = x\}$, the base theory T satisfies $T \vdash \{x \mid x = x\} = \{x \mid x = x\}$, so T' can be consistently extended with $\vdash V = \{x \mid x = x\}$, which is the result of replacing $\{x \mid x = x\}$ (this is P) with the new constant syntax expression V.

Finally, we would like to make the precondition of Theorem 15 something easily checkable, using some basic theorems for not-freeness.

Theorem 17. *1. If $x \notin \mathcal{V}(e)$, then $\mathrm{NF}_V(x, e)$.*
2. If $\mathrm{NF}_V(x, e_i)$ for each subtree e_i in the tree $a[\{e_i\}]$, then $\mathrm{NF}_V(x, a[\{e_i\}])$.
3. If σ is a substitution and $\mathrm{NF}(x, e)$ for all x such that $y \in \mathcal{V}(\sigma(x))$, then $\mathrm{NF}_V(y, \sigma(e))$.

Proof. Fix a model $M \models T$.

1. If μ, ν are valuations that differ only at $x \notin \mathcal{V}(e)$, then $\eta_\mu(e) = \eta_\nu(e)$, so certainly $\eta_\mu(e) \simeq \eta_\nu(e)$.

2. Let μ, ν be valuations that differ only at x. Now $\eta_\mu(a[\{e_i\}]) = \pi_a(\{\eta_\mu(e_i)\})$, and for each i, $\eta_\mu(e_i) \simeq \eta_\nu(e_i)$, so by Theorem 14 $\eta_\mu(a[\{e_i\}]) \simeq \eta_\nu(a[\{e_i\}])$.
3. Let μ, ν be valuations that differ only at y, and xxx. We have $\eta_\mu(\sigma(e)) = \eta_{\sigma(\mu)}(e)$, and $\sigma(\mu)(x) = \eta_\mu(\sigma(x))$, so $\sigma(\mu)$ differs from $\sigma(\nu)$ only when $y \in \mathcal{V}(\sigma(x))$; then since by assumption $\mathrm{NF}(x, e)$ for each of these x, $\eta_{\sigma(\mu)}(e) \simeq \eta_{\sigma(\nu)}(e)$.

□

These theorems are true under very general circumstances, but a major complicating factor is being able to show the $x \simeq y$ relation, which involves many individual elements of the model. To that end, we introduce a reduction to a single theorem in the object language:

Definition 9. *An* equality *in the formal system T for the type c is an expression $e(x, y)$ containing two variables x, y of type c, such that for all models $M \models T$ and valuations μ, $M \models_\mu e(x, y)$ iff $\mu(x) \simeq \mu(y)$.*

A classic example of an equality is the expression $\vdash x = y$ in set theory, or the relation $\vdash (\varphi \leftrightarrow \psi)$ in predicate logic. That these are actually equalities by this definition remains to be shown, but first let us derive the essential properties of an equality. We write $x \approx y$ instead of $e(x, y)$ for a given equality $\approx$.

Theorem 18. *Let $\approx$ be an equality in T.*

1. *The definition of equality generalizes to expressions: for all M, μ and syntax expressions e, e', $M \models_\mu e \approx e'$ iff $\eta_\mu(e) \simeq \eta_\mu(e')$.*
2. *$\approx$ is provably an equivalence relation in T.*
3. *$\approx$ is the unique equality of type c.*
4. *For any expression e and substitutions σ, σ', if for each $v \in \mathcal{V}(e)$ with $\sigma(v) \neq \sigma'(v)$ there is some equality $\equiv$ on the type $\mathrm{Type}(v)$ such that $T \vdash \sigma(v) \equiv \sigma'(v)$, then $T \vdash \sigma(e) \approx \sigma'(e)$.*

Proof.

1. Applying the definition to the valuation $\sigma(\mu)$, where $\sigma(x) = e$ and $\sigma(y) = e'$, we get $\eta_\mu(e) \simeq \eta_\mu(e')$ iff $\sigma(\mu)(x) \simeq \sigma(\mu)(y)$, iff $M \models_{\sigma(\mu)} x \approx y$, iff $\eta_{\sigma(\mu)}(x \approx y) = \eta_\mu(\sigma(x \approx y)) = \eta_\mu(\sigma(x) \approx \sigma(y)) = \eta_\mu(e \approx e')$ is defined, iff $M \models_\mu e \approx e'$.
2. We will show transitivity; reflexivity and symmetry are proven by an analogous argument. Suppose $T \vdash x \approx y$ and $T \vdash y \approx z$, and let M, μ be given such that $M \models_\mu x \approx y$ and $M \models_\mu y \approx z$. The definition of equality gives $\mu(x) \simeq \mu(y)$ and $\mu(y) \simeq \mu(z)$, and $\simeq$ is an equivalence relation, so $\mu(x) \simeq \mu(z)$ and hence $M \models_\mu x \approx z$. Thus $T \vdash x \approx z$ by Gödel's theorem.
3. $T \vdash x \approx y$ iff for all M, μ, $M \models_\mu x \approx y$, iff for all M, μ, $\mu(x) \simeq \mu(y)$. Since this latter expression does not depend on $\approx$, it follows that if $\equiv$ is another equality for the same type c, then $T \vdash x \approx y$ iff $T \vdash x \equiv y$, so $\approx$ and $\equiv$ are provably equivalent.

4. By Gödel's theorem we are given M, μ such that $M \models_\mu \sigma(v) \approx \sigma'(v)$ for each $v \in \mathcal{V}(e)$, and want to show $M \models_\mu \sigma(e) \approx \sigma'(e)$, or equivalently $\eta_{\sigma(\mu)}(e) \simeq \eta_{\sigma'(\mu)}(e)$.
   By the substitution property of $\simeq$, this follows if for all v, $\sigma(\mu)(v) \simeq \sigma'(\mu)(v)$, equivalently $\eta_\mu(\sigma(v)) \simeq \eta_\mu(\sigma'(v))$, and WLOG we can assume this is satisfied outside $v \in \mathcal{V}(e)$. If $\sigma(v) = \sigma'(v)$ this is clearly satisfied so we can assume $\sigma(v) \neq \sigma'(v)$.
   But then the assumption applies: for each $v \in \mathcal{V}(e)$ we are given an equality for the type such that $T \vdash \sigma(v) \equiv \sigma'(v)$, which implies that $M \models_\mu \sigma(v) \equiv \sigma'(v)$, which is equivalent to $\eta_\mu(\sigma(v)) \simeq \eta_\mu(\sigma'(v))$ or $\sigma(\mu)(v) \simeq \sigma'(\mu)(v)$.

□

We now have definitions of equality and bound variables that make sense in an arbitrary grammatical formal system, even though we did not assume anything resembling a first order logic system in T. The catch is that there may not be any expression in the logic that adequately represents an equality. In order to make use of this architecture, we must have a way to construct actual equalities in T, so we need the converse of Theorem 19(4).

Theorem 19. *Suppose expressions $x \approx_c y$ are given for each $c \in C \subseteq \mathsf{VT}$, and the following are theorems:*

$$\vdash x \approx_c x \tag{1}$$

$$x \approx_c y \vdash y \approx_c x \tag{2}$$

$$x \approx_c y,\ y \approx_c z \vdash x \approx_c z; \tag{3}$$

and for every provable typecode $c \in \mathsf{TC} \setminus \mathsf{VT}$, $\mathrm{Syn}(c) \in C$ and

$$x,\ x \approx_{\mathrm{Syn}(c)} y \vdash y; \tag{4}$$

and for each syntax axiom a and each i, if the i-th variable of a is of type $c' \in C$, a is of some type $c \in C$ and

$$x \approx_{c'} y \vdash a[v_1, \ldots, v_{i-1}, x, v_{i+1}, \ldots, v_k] \approx_c a[v_1, \ldots, v_{i-1}, y, v_{i+1}, \ldots, v_k]. \tag{5}$$

(All variables v_i and x, y, z above are variables, not expressions, of the appropriate type.) Then $\approx_c$ is an equality for each $c \in C$.

Proof. Let a model M and valuation μ be given. We wish to show $M \models_\mu x \approx y$ iff $\mu(x) \simeq \mu(y)$. ($\Leftarrow$) In the reverse direction, $\mu(x) \simeq \mu(y)$ implies $\eta_\mu(x \approx y) = \pi_\approx(\mu(x), \mu(y)) \simeq \pi_\approx(\mu(y), \mu(y))$, so $M \models_\mu x \approx y$ iff $M \models_\mu y \approx y$, which is true by Equation 1.

($\Rightarrow$) In the forward direction, we have $M \models_\mu x \approx_c y$, and want to prove $\mu(x) \simeq \mu(y)$. Define an equivalence relation $\sim$ on the elements of the model such that $v \sim w$ if v, w have the same type c, and:

- If $c \in C$, then there is some ν with $\nu(x) = v$ and $\nu(y) = w$ such that $M \models_\nu x \approx_c y$.

– If $c \notin C$, then $\mu(v) = \mu(w)$.

(Note that the "for some ν" can be replaced by "for any ν", because any two valuations which take the same values at x, y give the same result in the model, since $x \approx_c y$ only depends on x, y.)

Then $x \sim y$, and eqs. (1) to (3) imply that $\sim$ is an equivalence relation. (For transitivity, given $a \sim b \sim c$, we apply the transitivity theorem for a valuation such that $\nu(x) = a$, $\nu(y) = b$, $\nu(z) = c$.)

For η-closure, by induction it is sufficient to prove it for the syntax axioms. So we are given that $v_i \sim w_i$ for each i, and we want to show $\pi_a(\{v_i\}) \sim \pi_a(\{w_i\})$. By transitivity we can reduce to the case when $v_i = w_i$ except at one point k. If $\mathrm{Type}(v_k)$ is not in C, then $v_k = w_k$ so $\pi_a(\{v_i\}) = \pi_a(\{w_i\})$. Otherwise by $v_k \sim w_k$, we can choose variables $\{p_i, q_i\}_i$ such that $p_i = q_i$ if $i \neq k$, and $p_k = x$, $q_k = y$; then choose a valuation such that $\mu(p_i) = v_i$ for each i, and $\mu(y) = w_k$ (so $\mu(q_i) = w_i$ for each i); and such that $M \models_\mu x \approx_c y$.

Equation (5) then applies for the valuation μ, to get $M \models_\mu a[\{p_i\}] \approx_{c'} a[\{q_i\}]$, where $c' \in C$ is the type of a. Thus $\mu(a[\{p_i\}]) \sim \mu(a[\{q_i\}])$, and $\mu(a[\{p_i\}]) = \pi_a(\{\mu(p_i)\}) = \pi_a(\{v_i\})$, and similarly for q_i, so $\pi_a(\{v_i\}) \sim \pi_a(\{w_i\})$.

Finally, we must show that M respects $\sim$, so we want (with the same v_i, w_i as above) that $\eta_\mu\langle c, a[\{p_i\}]\rangle$ is defined iff $\eta_\mu\langle c, a[\{q_i\}]\rangle$ is, where here $c \in \mathsf{TC}$ is the explicitly written root typecode (satisfying $\mathrm{Syn}(c) = \mathrm{Type}(a)$). Equivalently, $M \models_\mu \langle c, a[\{p_i\}]\rangle$ iff $M \models_\mu \langle c, a[\{q_i\}]\rangle$. Equation (4), applied in each direction, proves that this follows from $M \models_\mu, a[\{p_i\}] \approx_{c'} a[\{q_i\}]$, which we have already shown.

Thus, M respects the model equivalence relation $\sim$, and then by Definition 7, $M \models_\mu x \approx_c y$ implies $\mu(x) \simeq \mu(y)$. □

Remark 4. It should be noted that this is not the easiest way to prove that $\approx_c$ satisfies the substitution rules – we would be better served by echewing models altogether and performing induction directly on the recursive definition of well formed formulas in the language. This approach, however, clearly marks the relationship between the formal system definition and the definition from within a model.

3.1 Definitions in `set.mm`

Remark 5. The principal "consumer" of Theorem 19, and the source of intuition, is the case of set theory, with typecodes $\{\mathrm{set}, \mathrm{class}, \mathrm{wff}, \vdash\}$. We take $C = \{\mathrm{class}, \mathrm{wff}\}$, with $\approx_{\mathrm{class}} =$ " $\vdash A = B$" and $\approx_{\mathrm{wff}} =$ " $\vdash \varphi \leftrightarrow \psi$". The closure properties of C are important here: the non-variable typecode $\vdash$ has $\mathrm{Syn}(\vdash) = \mathrm{wff} \in C$, and every syntax axiom produces a value of some type $c \in C$. (This is trivial for $\vdash$ because it is a non-variable typecode, but it is important that set has no syntax axioms, so that only set variables can substitute for other set variables.)

Incidentally, it is not required to restrict C to just these two; in fact set also has an equality on it: $x \approx_{\mathrm{set}} y$ iff $\vdash \forall x\, x = y$. Note that this is saying not that x

and y have the save value in the logic, but they *are the same variable*. In standard FOL x and y are distinct simply because t hey are textually different, so that y is unbound and the statement states that the universe has only one element (which is provably false). But in Metamath it is possible for them to be the same variable, in which case it reads $\vdash \forall x\ x = x$ which is true. You can in fact prove the equality substitution formulas using this: $\vdash \forall x\ x = y \to (\forall x\ \varphi[x] \leftrightarrow \forall y\ \varphi[x])$ is a peculiar but provable assertion in `set.mm`.

How does the not-free predicate translate to `set.mm`? We already have a candidate for $\mathrm{NF}(x, \varphi)$, namely $\vdash (\varphi \to \forall x\ \varphi)$. Now that we know that $=$ is an equality in `set.mm` we may be able to finish the job.

Theorem 20. *In the formal system T of `set.mm`, for every expression φ of type* wff*, $\mathrm{NF}(x, \varphi)$ iff $T \vdash (\varphi \to \forall x\ \varphi)$ is provable with disjoint variable conditions $\{v, x\}$ for each $v \in \mathcal{V}(\varphi)$ with $v \neq x$.*

Proof. ($\Rightarrow$) In the forward direction, we want to show that $M \models_\mu (\varphi \to \forall x\ \varphi)$ for all M, μ given $\mathrm{NF}(x, \varphi)$. By the disjoint variable condition, we can assume $\mu(v)\ \#\ \mu(x)$ for each $v \in \mathcal{V}(\varphi)$, $v \neq x$. Let $\nu(v) = \mu(v)$ for all $v \neq x$, and $\nu(x) = w$, where w is chosen to be fresh from $\mu(v)$ for all $v \in \mathcal{V}(e)$. The definition of $\mathrm{NF}(x, \varphi)$ says that if ν is any other valuation which differs from μ only at x, and $\mu(v)\ \#\ \mu(x)$, $\mu(v)\ \#\ \nu(x)$ for all $v \in \mathcal{V}(e)$ other than x, then $\eta_\mu(\varphi) \simeq \eta_\nu(\varphi)$.

We can expand $M \models_\mu (\varphi \to \forall x\ \varphi)$ as $\pi_\to(\eta_\mu(\varphi), \pi_\forall(\mu(x), \eta_\mu(\varphi)))$, and then $\eta_\mu(\varphi) \simeq \eta_\nu(\varphi)$ implies that $\pi_\to(\eta_\nu(\varphi), \pi_\forall(\mu(x), \eta_\nu(\varphi)))$ is also defined in $\mathsf{U}_\vdash$. Finally, augment ν with $\nu(y) = \mu(x)$, where y is a variable not in $\mathcal{V}(\varphi) \cup \{x\}$. Then we can write the goal as $M \models_\nu (\varphi \to \forall y\ \varphi)$. (In other words, we just rewrote a statement like $(\exists x, x \in z \to \forall x\ \exists x, x \in z)$ to $(\exists y, y \in z \to \forall x\ \exists y, y \in z)$, where y is fresh from x, z.)

Now $\nu(y) = x\ \#\ \mu(v) = \nu(v)$ for each $v \in \mathcal{V}(\varphi)$, so by freshness substitution, $\nu(y)\ \#\ \eta_\mu(\varphi)$. Thus ax-17 applies:

$$\vdash (\varphi \to \forall x\ \varphi), \qquad \text{where } x, \varphi \text{ are disjoint}$$

so that $M \models_\nu (\varphi \to \forall y\ \varphi)$ as desired.

($\Leftarrow$) In the reverse direction, we know $T \vdash (\varphi \to \forall x\ \varphi)$ is provable with disjoint variable conditions $\{v, x\}$ for each $v \in \mathcal{V}(\varphi)$ with $v \neq x$, and we want $\mathrm{NF}(x, \varphi)$. Take a model M and valuations μ, ν that differ only at x, such that for all $v \in \mathcal{V}(\varphi)$ other than x, $\mu(v)\ \#\ \mu(x)$ and $\mu(v)\ \#\ \nu(x)$. Let ν be extended as before, choosing some $y \notin \mathcal{V}(\varphi)$ and setting $\nu(y) = \mu(x)$. Then our goal is to show $M \models_\nu \varphi \leftrightarrow \varphi[x \to y]$, where $\varphi[x \to y]$ is the result of (textually) replacing all x variables in φ with y, or just $M \models_\nu \varphi \to \varphi[x \to y]$ by symmetry.

Since $\varphi \to \forall x\ \varphi$ is provable this reduces to $M \models_\nu \forall x\ \varphi \to \varphi[x \to y]$. Application of a4v reduces the goal to $M \models_\nu x = y \to (\varphi \leftrightarrow \varphi[x \to y])$,which is provable by induction on φ using equality theorems. The interesting case is when $\varphi = \forall z\ \psi$. If $z = x$ (or $z = y$, by symmetry), this is $M \models_\nu x = y \to (\forall x\ \psi \leftrightarrow \forall y\ \psi[x \to y])$, and we can drop the hypothesis $x = y$ and apply cbvalv to reduce; while if $z \neq x$ and $z \neq y$, this is $M \models_\nu x = y \to (\forall z\ \psi \leftrightarrow \forall z\ \psi[x \to y])$ which is reducible using albidv. In either case the disjoint variable conditions are satisfied

because $\nu(x), \nu(y), \nu(z)$ are all fresh from each other by assumption (note that $z \in \mathcal{V}(\varphi)$). □

Remark 6. It is unfortunate that we must reprove that equality is preserved under substitution, since we already have this from Theorem 19. This is necessary because *conditional* equality does not necessarily behave the same way as unconditional equality, which is what we were reasoning about in the previous section, and in any case something resembling `set.mm` implication is not even available in the more general situation.

Remark 7. It should be noted that the assumption of disjointness in Theorem 20, while necessary for the biconditional, is not usually used. That is, not-free theorems in `set.mm` are usually established with few or no disjoint variable conditions, because these compose better. We can safely add these superfluous DV conditions in the final step to use the theorem.

To use Theorem 20, we note that from hbal, $\vdash \forall x\ vph \to \forall x\ \forall x\ \varphi$ (with no DV conditions), $\mathrm{NF}(x, \forall x\ \varphi)$ is true, so Theorem 17(3) allows us to map $x \mapsto x$ and $\varphi \mapsto e$ for any expression e (which may contain x), and then $\mathrm{NF}(x, \forall x\ e)$ is true. That is, x is bound in any expression of the form $\forall x\ -$, which matches our intuition for bound variables. Theorem 17(2) then allows us to prove $\mathrm{NF}(x, \psi \vee \forall x\ \varphi)$ and similar expressions as well.

If $\sigma = \{x \mapsto x, \varphi \mapsto e\}$ is a substitution and $\mathrm{NF}(y, \forall x\ e)$ for all y such that $x \in \mathcal{V}(\sigma(y))$, then $\mathrm{NF}(x, \sigma(\forall x\ e))$.

Using Theorem 15, Theorem 17

References

1. Carneiro, M.: Models for Metamath. Preprint, arXiv:1601.07699 [math.LO].
2. Megill, N.: Metamath: A Computer Language for Pure Mathematics. Lulu Publishing, Morrisville, North Carolina (2007), `http://us.metamath.org/downloads/metamath.pdf`
3. Tarski, A.: "A Simplified Formalization of Predicate Logic with Identity," *Archiv für Mathematische Logik und Grundlagenforschung*, 7:61-79 (1965) [QA.A673].
4. Hofstadter, D.: *Gödel, Escher, Bach.* Basic Books, Inc., New York (1979) [QA9.H63 1980].