The transfinite recursion theorem: a fine structure analysis*

Josep Maria Blasco

Espacio Psicoanalítico de Barcelona
Balmes, 32, 2º 1ª – 08007 Barcelona
jose.maria.blasco@epbcn.com
+34 93 454 89 78

September 25, 2006

Abstract

It is often stated in the set-theoretical literature that some constructions (for example, Gödel's constructible universe L, or forcing extensions) can be carried out using a finite number of axioms. Most of these constructions are based or make heavy use of the transfinite recursion theorem. In this article we provide a completely explicit analysis of the recursion theorem. If φ is the formula on which we do the recursion, we calculate the exact set of axioms needed to prove the recursion theorem for φ, as a recursive function of (the code of) φ. In the way to this result, we develop a framework for the fine-structure analysis of Δ_0 formulas, and exercise it to find explicit Δ_0 expressions for usual concepts, like *being a pair* or *being a function*, that are necessary to develop the basic set-theoretical concepts needed to express the transfinite recursion theorem.

Contents

*URL of this document: https://www.epbcn.com/pdf/jose-maria-blasco/2006-09-25-The-transfinite-recursion-theorem-a-fine-structure-analysis.pdf.
This article was written during the 2005-2006 academic year for the *Diploma d'Estudis Avançats* of the doctorate program "Lógica y Fundamentos de las Matemáticas" (Logic and Foundations of Mathematics) in the Department of Logic, History and Philosophy of Science of the University of Barcelona.

Part I
The basics

1 Introduction

1.1 Statement of our task

This article is about formulas. We want to find a solution to the following problem:

Problem 1.1. *Which is the exact set of axioms of Kripke-Platek Set Theory (KP) needed to prove the transfinite $\in$-recursion theorem for a given formula φ? The solution should ideally be expressed as a primitive recursive formula (i.e., a computer program) that takes φ as an argument and returns as its output the list of axioms of KP* in the pure language of set theory with equality.

Four aspects of this problem make it non-trivial.

1.2 The problem of defined notions

One is the requirement that the axioms are expressed as formulas of the pure language of set theory with equality: there is a (quite large) number of very good books that prove the $\in$-recursion theorem, but all of them make use of *defined notions*; this is standard mathematical practice. As they make use of defined notions, they either extend the language, so that we are no longer working in $\mathcal{L} = \{\in, =\}$, or they state that defined notions are mere abbreviations, and can, *in principle*, be undone so that the formula becomes a formula of $\mathcal{L}$. When they write *in principle*, one has to pay attention: they mean that nobody does it *in practice* (because it is uninteresting, it would be said; but mainly, as we will see, because it is *practically impossible*[1]). But then we have the problem that nobody really knows which are the real, effective formulas we are dealing with, because they are full of defined notions, which are in turn defined over other defined notions, etc.

1.3 The contingency of definitions and proofs

The second problem is that in normal mathematics we don't care which one of several definitions is used, as long as they are all proved equivalent, and we don't care which proof of a theorem we use, as long as the proof is correct. But in our case, since we are interested in effective formulas, one definition or the other, one proof or the other, would lead to vastly different formulas. Clearly, we would need a way to decide, once given two definitions or two proofs, which one is better, that is, from our point of view, which one leads to *simpler* formulas. But this immediately leads us to our third problem.

1.4 The absence of a normal-form theorem for Δ_0 formulas

Remember that we are working in KP. In KP there are two kinds of axioms: simple axioms, such as Extensionality, Empty Set, Pairing and

[1]If this seems abusive, take a look at Appendix B, especially the last formula.

Union, and axiom schemas, namely Separation, Collection and Foundation. Simple axioms do not count, in the sense that most proofs use all of them, and anyway, except maybe for some variable renaming, there is only one instance of each. But the axiom schemas are parameterized in the metalanguage by a formula φ, and this formula cannot be *any* formula. In particular, φ should be Δ_0 for Separation and Collection; and, while stronger logical forms are allowed for Foundation, it is desirable that the complexity of φ is somehow kept *as low as possible*.

And here our third problem arises: while we try to keep a formula "as simple as possible", we are faced with the following question: between two Δ_0 formulas, which one is *simpler*? Unfortunately, there is no definite answer to that question, because there is no normal-form theorem for Δ_0 formulas. The reason for that is very simple: while we can always transform a formula of the form $\forall x \varphi \wedge \psi$ into a formula of the form $\forall x(\varphi \wedge \psi)$ (renaming some variables if necessary), a formula of the form $(\forall x \in y)\varphi \wedge \psi$ can *not* in general be transformed into $(\forall x \in y)(\varphi \wedge \psi)$ (think about the case where y is empty; the $\exists$ case is symmetrical).

This means that we cannot move all quantifiers in an orderly fashion to the beginning of the formula, and therefore that, at least in some cases, there will be no clear, unambiguous way, to decide which of two formulas is the simplest one.

1.5 The unmanageability of pure systems

The fourth problem is easily explained: assume that we have laboriously built a set of primitive recursive formulas that constitute a solution to our problem. These formulas will depend on the exact definitions we have chosen for the usual notions of matematics, and on the concrete, detailed proofs we have chosen for the theorems we need. If we later discover a simpler proof for some of those theorems, or a simpler definition for our defined notions, *we should rebuild all our formulas according to these modifications*, and this is simply *unmanageable*.[2]

1.6 Ways to a solution

In order to be able to manage these four problems, we will have to take a number of decisions which will make the notation used throughout this article a little unusual:

1.6.1 Metafunctions vs. defined notions

Instead of using defined notions, we will define a number of *metafunctions*. These metafunctions will be similar to the defined notions, but they will be also concrete, effective, non-substitutable for equivalent notions. For example, if f is a function then "x belongs to the domain of f", $x \in \operatorname{dom} f$, means that there is a pair $p \in f$ such that $p = \langle x, y \rangle$. But if x is the first component of p, following the Kuratowski definition of ordered pairs, this means that x belongs to all elements e of p. Therefore, $x \in \operatorname{dom} f$ can be expressed as

$$(\exists p \in f)(\forall e \in p)(x \in e). \tag{1}$$

[2]If you are not convinced by this last assertion, you did not take a look at Appendix B. Do it now, and you will come back as a believer.

Hence, we will define a metafunction $\mathsf{InDomain}[x, f]$ to be the formula (1) (the variable e is irrelevant, and will be supplied automatically by the metafunction; if we need to specify it for whatever reason, we will write $\mathsf{InDomain}[x, f; e]$: the semicolon will serve to separate "essential" variables from "auxiliary" ones).

1.6.2 Metafunctions vs. formula transformations

We will need to effectively show that several of the formulas we are dealing with are Σ_1, or can be transformed into Σ_1 formulas. To this effect, we will define another set of metafunctions.

Several of them will transform formulas into logically equivalent ones, using only pure logic as their justification. For example, since a formula of the form

$$(\exists x_1 \in y_1)...(\exists x_n \in y_n)\exists z\varphi \tag{2}$$

can be transformed (if certain conditions about the variables are met) into an equivalent formula of the form

$$\exists z(\exists x_1 \in y_1)...(\exists x_n \in y_n)\varphi \tag{3}$$

by applying only rules of pure logic, we will define a metafunction MoveUp that will transform formulas like (2) into their Σ_1 equivalents (3).

Other metafunctions will transform formulas, but this time applying *axioms of KP*, i.e., not by pure logic alone. A clear example is the following: since we can transform formulas of the form

$$(\forall x \in y)\exists z\varphi \tag{4}$$

into formulas of the form

$$\exists w(\forall x \in y)(\exists z \in w)\varphi \tag{5}$$

by applying Collection, we will define a metafunction Collect that will take as argument a formula like (4) and a new variable w, and return a formula like (5) as its result.

1.6.3 Computer programs do it better

To handle the problem of unmanageability of formulas, and to be able to redo all our calculations in case some defined notion or proof may change, we will use a computer program that effectively implements all our metafunctions and effectively builds our formulas. This way, if something gets altered later, a few changes in the computer program will allow us to automatically rebuild all formulas. Program listings for a preliminary implementation of such a program are included as an Annex to this article.

1.7 Structure of this article

The structure of this article is as follows: part I, "The Basics", is divided into eight sections. The first section is this *Introduction*. Section two, *Notations, basic facts and definitions*, fixes the notation used throughout the article, and introduces a number of syntactical transformations, defined as metafunctions, which will be later used in several of the proofs. Section three, *Denoting complexities*, introduces a notation for a specific measure of complexity for Δ_0 formulas. Section four, *Set Theory: The first axioms*, introduce the Empty Set, Extensionality and

Foundation axioms; some metafunctions are defined for Foundation. Section five, *Enumerations and quantifiers*, introduce the Pairing and Union axioms, define syntactically finite sets, and prove a theorem that collapses several existential quantifiers into one. Section six, *Separation and Collection*, introduces the Separation and Collection axioms and several related metaoperations, and proves a strong form of Collection that will be needed in the proof of the recursion theorem. Section seven, *Tuples*, defines tuples and several related metaoperations. Those will be needed in section eight, *Classes, relations and functions*, when the higher-level mathematical notions used in this article, besides recursion itself, are presented.

Part II, "Transfinite induction and recursion", consists of a single section, in which the Transfinite $\in$-Recursion Theorem is proved in full detail, keeping track of all the axioms used and all the formulas involved.

Three appendixes are included: in Appendix A, "An example: the transitive closure" we evaluate the complexity of the axioms needed to prove the existence of the transitive closure of a set. In Appendix B, "A curiosity: The Π_1-Foundation axiom for the transitive closure case", we calculate in an effective way the instance of Π_1-Foundation needed to prove the existence of the transitive closure of a set. In Appendix C, "Metafunctions reference", we give a complete alphabetical list of all the metafunctions used throughout this article.

1.8 Further work

The results presented in this text can be improved and extended in several ways. Here is a non-comprehensive list, in no particular order.

1) *Optimizing formulas used in this text.* Some of the formulas have alternative equivalents of less complexity. For example, $\mathsf{Tuples}_\mathsf{n}$ can be improved in the following way when $\mathsf{n} \geq 2$: since we already know that all elements are tuples when producing the first tuple, we could avoid some of the tests imposed on subsequent tuples. This would shorten the definition of $\mathrm{Fun}(f)$, for example.

2) *Finding new ways to further optimize formulas.* An example: although $(\exists x \in y)\varphi \wedge \psi$ is not equivalent in general to $(\exists x \in y)(\varphi \wedge \psi)$, if we know that $y \neq \emptyset$ (for example, because there is some $\exists z \in y$ at a higher level in the syntax tree) then the equivalence is possible (with some variable renaming if necessary).

3) *Exploring new measures of complexity.* The notation we have developed for Δ_0 complexities is a good start: it allows us to get an impression of the nature of the involved formulas. However, many other complexity measures suggest themselves: the number of bounded quantifiers used, the height or the width of the syntax tree, or even the length of the formula (i.e., the number of symbols used). All these measures could be used to try to tackle the question of when a Δ_0 formula is simpler than another formula.

4) *Applying the machinery to new problems.* The first, almost mandatory, application of this machinery must be to develop the theory of the ordinal numbers and ordinal recursion. Further applications could be fine-structure analysis of the constructible universe L, or of Forcing.

1.9 Acknowledgements

I'd like to thank Joan Bagaria, ICREA Research Professor at the University of Barcelona and advisor of this article, for his continued patience, support and encouragement.

2 Notation, basic facts and definitions

2.1 Syntax

Metadefinition 2.1. *$\mathcal{L} = \{\in, =\}$ is the **language** of Set Theory with equality.*

Metadefinition 2.2 (Variables and stems)**.** *$\mathsf{Var}(\mathcal{L})$ is the countable set of **variables** of $\mathcal{L}$. In some cases, we will be interested in endowing variables with a well-order: as a consequence, we will be able to speak of "the minimum variable such that..." and similar constructions. We may also consider variables as having some kind of substructure: for example, we may consider variables as being formed from a finite number of **stems** (say the set $A = \{a, b, c, ..., z\}$) enhanced by subscripts and superscripts taken from infinite sets (for example, $\mathbb{N}$, or $\mathbb{N} \cup A$). In this case, we will say that the variables are **derived from** the stem.*

Example. The following are variables: x, y, X, α, Γ, x_1, y^β, x_2^3. If x is a stem, then x_1, x^4, x_α^2, etc., are variables derived from this stem.

$\mathsf{Form}(\mathcal{L})$ is the set of formulas of $\mathcal{L}$, recursively defined below.

Notation (Objects and metaobjects)**.** *Roman italics will be used for variables like v which range over sets; if a variable v ranges over $\mathsf{Var}(\mathcal{L})$, $\mathsf{Form}(\mathcal{L})$, etc., we will use a* sans-serif *font. We will also use a sans-serif font for* meta-functions*.*

Notation (Metafunctions)**.** *If F is a metafunction that operates on formulas, variables, etc., and returns a formula, we will write its arguments between brackets, thus: $\mathsf{F}[\mathsf{h}, \mathsf{v}_1, \mathsf{v}_2]$. In many cases the resulting formula will need to use auxiliary (bound) variables; these can be omitted, as they will be automatically generated by the metaformula. If for whatever reason we want precise control over these auxiliary variables, we can indicate it by specifiying them after a semicolon: for example, $\mathsf{F}[\mathsf{h}, \mathsf{v}_1, \mathsf{v}_2; \mathsf{e}_1, \mathsf{e}_2]$.*

Notation (Metadefinitions by cases)**.** *In many cases we will give partial definitions of metaformulas by showing partially the forms of their arguments. For example we might partially define a metafunction F by writing*

$$\mathsf{F}[\forall \mathsf{v} \mathsf{h}] \stackrel{def}{=} \exists \mathsf{f} \neg \mathsf{h},$$

where h ranges over $\mathsf{Form}(\mathcal{L})$ and v ranges over $\mathsf{Var}(\mathcal{L})$. Such a definition is to be read as partial definition of F, which says nothing about the value of F for other kind of arguments, for example it says nothing about $\mathsf{F}[\exists \mathsf{v} \mathsf{h}]$. Many partial definitions constitute a (maybe still partial) definition by cases. Finally, a metafunction F is to be considered undefined when no partial definition is applicable.

Notation (Vector notation)**.** *We will use vector notation to allow easier reading: if $\mathsf{x}_1, ..., \mathsf{x}_n \in \mathsf{Var}(\mathcal{L})$, we can write $\vec{\mathsf{x}}$ instead of $\mathsf{x}_1, ..., \mathsf{x}_n$ if n is clear from the context.*

We now begin the recursive definition of $\mathsf{Form}(\mathcal{L})$. Formulas will be defined in 2.11 as finite-length *preformulas* (2.10); a preformula es either an *atom* (2.3) or a *factor* (2.7) or a *conjunction* (2.8) or a *disjunction* (2.9); disjunctions are composed of one or more conjunctions, conjunctions of one or more factors, factors are either atomic, parenthesized formulas, negated formulas, or (unboundedly) quantified formulas, and atomic formulas are (in)equalities or formulas of the form $\mathsf{v}_1 \in \mathsf{v}_2$ or $\mathsf{v}_1 \notin \mathsf{v}_2$. Those definitions do not incur in a vicious circle because we are imposing that formulas should be of finite length: without this limitation, we could have formulas of the form

$$\forall x_1 \forall x_2 \forall x_3 ... \varphi$$

with an infinite number of x_i.

Implication, double implication and bounded quantifiers are presented as defined notions.

Metadefinition 2.3 (Atomic formulas)**.**

$$\mathsf{Atom}(\mathcal{L}) \stackrel{def}{=} \{\mathsf{v}_1 = \mathsf{v}_2, \mathsf{v}_1 \in \mathsf{v}_2 : \mathsf{v}_1, \mathsf{v}_2 \in \mathsf{Var}(\mathcal{L})\}.$$

Metadefinition 2.4 (Negated atoms)**.** *Let* $\mathsf{v}_1, \mathsf{v}_2 \in \mathsf{Var}(\mathcal{L})$*. Then*

$$\begin{aligned} \mathsf{v}_1 \neq \mathsf{v}_2 &\stackrel{def}{=} \neg(\mathsf{v}_1 = \mathsf{v}_2), \textit{ and} \\ \mathsf{v}_1 \notin \mathsf{v}_2 &\stackrel{def}{=} \neg(\mathsf{v}_1 \in \mathsf{v}_2). \end{aligned}$$

Metadefinition 2.5 (Implication and double implication)**.** *Let* $\mathsf{f}_1, \mathsf{f}_2 \in \mathsf{Form}(\mathcal{L})$*. Then,*

$$\begin{aligned} \mathsf{f}_1 \to \mathsf{f}_2 &\stackrel{def}{=} \neg(\mathsf{f}_1) \vee \mathsf{f}_2, \textit{ and} \\ \mathsf{f}_1 \leftrightarrow \mathsf{f}_2 &\stackrel{def}{=} (\neg(\mathsf{f}_1) \vee \mathsf{f}_2) \wedge (\neg(\mathsf{f}_2) \vee \mathsf{f}_1). \end{aligned}$$

Metadefinition 2.6 (First-order bounded quantifiers)**.** *Let* $\mathsf{f} \in \mathsf{Form}(\mathcal{L})$ *and* $\mathsf{v}_1, \mathsf{v}_2 \in \mathsf{Var}(\mathcal{L})$*. Then,*

$$(\forall \mathsf{v}_1 \in \mathsf{v}_2)\mathsf{f} \stackrel{def}{=} \forall \mathsf{v}_1(\mathsf{v}_1 \in \mathsf{v}_2 \to \mathsf{f}),$$

and

$$(\exists \mathsf{v}_1 \in \mathsf{v}_2)\mathsf{f} \stackrel{def}{=} \exists \mathsf{v}_1(\mathsf{v}_1 \in \mathsf{v}_2 \wedge \mathsf{f}).$$

Metadefinition 2.7 (Factors)**.**

$$\begin{aligned} \mathsf{Factor}(\mathcal{L}) &\stackrel{def}{=} \mathsf{Atom}(\mathcal{L}) \\ &\cup \{(\mathsf{f}), \neg(\mathsf{f}) : \mathsf{f} \in \mathsf{Form}(\mathcal{L})\} \\ &\cup \{\forall \mathsf{v}(\mathsf{f}) : \mathsf{v} \in \mathsf{Var}(\mathcal{L}), \mathsf{f} \in \mathsf{Form}(\mathcal{L})\} \\ &\cup \{\exists \mathsf{v}(\mathsf{f}) : \mathsf{v} \in \mathsf{Var}(\mathcal{L}), \mathsf{f} \in \mathsf{Form}(\mathcal{L})\}. \end{aligned}$$

It is immediate that $\mathsf{Atom}(\mathcal{L}) \subseteq \mathsf{Factor}(\mathcal{L})$.

Metadefinition 2.8 (Conjunctions)**.**

$$\mathsf{Conj}(\mathcal{L}) \stackrel{def}{=} \{\mathsf{f}_1 \wedge ... \wedge \mathsf{f}_\mathsf{n} : \mathsf{n} > 0, \mathsf{f}_\mathsf{n} \in \mathsf{Factor}(\mathcal{L}) \textit{ for all } \mathsf{n}\}.$$

The case $\mathsf{n} = 1$ shows that $\mathsf{Factor}(\mathcal{L}) \subseteq \mathsf{Conj}(\mathcal{L})$.

Metadefinition 2.9 (Disjunctions)**.**

$$\mathsf{Disj}(\mathcal{L}) \stackrel{def}{=} \{\mathsf{c}_1 \vee ... \vee \mathsf{c}_\mathsf{n} : \mathsf{n} > 0, \mathsf{c}_\mathsf{n} \in \mathsf{Conj}(\mathcal{L}) \textit{ for all } \mathsf{n}\}.$$

The case $\mathsf{n} = 1$ shows that $\mathsf{Conj}(\mathcal{L}) \subseteq \mathsf{Disj}(\mathcal{L})$.

Metadefinition 2.10 (Preformulas)**.**

$$\mathsf{PreForm}(\mathcal{L}) \stackrel{def}{=} \mathsf{Atom}(\mathcal{L}) \cup \mathsf{Factor}(\mathcal{L}) \cup \mathsf{Conj}(\mathcal{L}) \cup \mathsf{Disj}(\mathcal{L}).$$

Notice that, since $\mathsf{Atom}(\mathcal{L}) \subseteq \mathsf{Factor}(\mathcal{L}) \subseteq \mathsf{Conj}(\mathcal{L}) \subseteq \mathsf{Disj}(\mathcal{L})$, we could have defined simply $\mathsf{PreForm}(\mathcal{L}) \stackrel{\text{def}}{=} \mathsf{Disj}(\mathcal{L})$.

Metadefinition 2.11 (Formulas)**.**

$$\mathsf{Form}(\mathcal{L}) \overset{def}{=} \{\mathsf{f} \in \mathsf{PreForm}(\mathcal{L}) : \mathsf{f} \textit{ uses only a finite number of symbols}\}.$$

Metadefinition 2.12 (Variables of a formula)**.** *We define the **variables** of a formula recursively over* $\mathsf{Form}(\mathcal{L})$ *as follows:*

$$\begin{array}{llll} \mathsf{Vars}[\mathsf{v}_1 = \mathsf{v}_2] & \overset{def}{=} \{\mathsf{v}_1, \mathsf{v}_2\}, & \mathsf{Vars}[\mathsf{v}_1 \in \mathsf{v}_2] & \overset{def}{=} \{\mathsf{v}_1, \mathsf{v}_2\}, \\ \mathsf{Vars}[(\mathsf{f})] & \overset{def}{=} \mathsf{Vars}[\mathsf{f}], & \mathsf{Vars}[\neg(\mathsf{f})] & \overset{def}{=} \mathsf{Vars}[\mathsf{f}], \\ \mathsf{Vars}[\forall \mathsf{v}(\mathsf{f})] & \overset{def}{=} \mathsf{Vars}[\mathsf{f}] \cup \{\mathsf{v}\}, & \mathsf{Vars}[\exists \mathsf{v}(\mathsf{f})] & \overset{def}{=} \mathsf{Vars}[\mathsf{f}] \cup \{\mathsf{v}\}, \end{array}$$

$$\begin{array}{ll} \mathsf{Vars}[\mathsf{f}_1 \wedge \ldots \wedge \mathsf{f}_\mathsf{n}] & \overset{def}{=} \mathsf{Vars}[\mathsf{f}_1] \cup \ldots \cup \mathsf{Vars}[\mathsf{f}_\mathsf{n}], \\ \mathsf{Vars}[\mathsf{c}_1 \vee \ldots \vee \mathsf{c}_\mathsf{n}] & \overset{def}{=} \mathsf{Vars}[\mathsf{c}_1] \cup \ldots \cup \mathsf{Vars}[\mathsf{c}_\mathsf{n}]. \end{array}$$

[Where v*,* v_1 *and* v_2 *are variables,* f *is any formula, the* f_i*'s are factors, and the* c_i*'s are conjunctions.]*

Metadefinition 2.13 (Free variables)**.** *We define the **free variables** of a formula recursively over* $\mathsf{Form}(\mathcal{L})$ *as follows:*

$$\begin{array}{llll} \mathsf{Free}[\mathsf{v}_1 = \mathsf{v}_2] & \overset{def}{=} \{\mathsf{v}_1, \mathsf{v}_2\}, & \mathsf{Free}[\mathsf{v}_1 \in \mathsf{v}_2] & \overset{def}{=} \{\mathsf{v}_1, \mathsf{v}_2\}, \\ \mathsf{Free}[(\mathsf{f})] & \overset{def}{=} \mathsf{Free}[\mathsf{f}], & \mathsf{Free}[\neg(\mathsf{f})] & \overset{def}{=} \mathsf{Free}[\mathsf{f}], \\ \mathsf{Free}[\forall \mathsf{v}(\mathsf{f})] & \overset{def}{=} \mathsf{Free}[\mathsf{f}] \setminus \{\mathsf{v}\}, & \mathsf{Free}[\exists \mathsf{v}(\mathsf{f})] & \overset{def}{=} \mathsf{Free}[\mathsf{f}] \setminus \{\mathsf{v}\}, \end{array}$$

$$\begin{array}{ll} \mathsf{Free}[\mathsf{f}_1 \wedge \ldots \wedge \mathsf{f}_\mathsf{n}] & \overset{def}{=} \mathsf{Free}[\mathsf{f}_1] \cup \ldots \cup \mathsf{Free}[\mathsf{f}_\mathsf{n}], \\ \mathsf{Free}[\mathsf{c}_1 \vee \ldots \vee \mathsf{c}_\mathsf{n}] & \overset{def}{=} \mathsf{Free}[\mathsf{c}_1] \cup \ldots \cup \mathsf{Free}[\mathsf{c}_\mathsf{n}]. \end{array}$$

[Where v*,* v_1 *and* v_2 *are variables,* f *is any formula, the* f_i*'s are factors, and the* c_i*'s are conjunctions.]*

It is immediate from the above definitions that for all formulas f, $\mathsf{Free}(\mathsf{f}) \subseteq \mathsf{Vars}(\mathsf{f})$.

Metadefinition 2.14 (New (free) variables)**.** *Let* f_i*,* $i = 1, \ldots, n$ *be formulas, and let* $\mathsf{S} \subset \mathsf{Var}(\mathcal{L})$ *be a set of variables. A **new variable** (with respect to the* f_i*'s and* S *is*

$$\min(\mathsf{Var}(\mathcal{L}) \setminus (\mathsf{S} \cup \bigcup_{1 \leq i \leq n} \mathsf{Vars}(\mathsf{f}_i)),$$

*and a **new free variable** is*

$$\min(\mathsf{Var}(\mathcal{L}) \setminus (\mathsf{S} \cup \bigcup_{1 \leq i \leq n} \mathsf{Free}(\mathsf{f}_i)).$$

Metadefinition 2.15 (Equivalent formulas)**.** *Two formulas* f *and* g *may be logically equivalent, and in this case we will write*

$$\mathsf{f} \equiv \mathsf{g},$$

and say that f *and* g *are logically equivalent formulas, or equivalent by pure logic.*

Similarly, f *and* g *may be made equivalent by assuming some finite set of axioms* $\mathsf{A} = \{\mathsf{a}_i : 1 \leq i \leq n\} \subset KP$*. In this case, we will also write*

$$\mathsf{f} \equiv \mathsf{g},$$

and say that f *and* g *are equivalent by virtue of* A*, or modulo* A*, or, more simply, by* A*.*

Bounded quantifiers, (double) implication and negated relations are defined meta-notions. The introduction of n-way conjunctions and disjunctions in the syntax, for $\mathsf{n} > 2$, is consistent with normal mathematical practice, and a trivial consequence of the associativity of $\wedge$ and $\vee$. Notice that in our syntax, again corresponding to normal practice, $\wedge$ has precedence over $\vee$.

Having defined our syntax, we will now allow for the possibility of slightly relaxing our notation, and use φ, ψ, θ, etc. to denote formulas. If there is no ambiguity, we will also allow the use of $x, y \in \mathsf{Var}(\mathcal{L})$ instead of the more proper but less readable $\mathsf{v}_1, \mathsf{v}_2 \in \mathsf{Var}(\mathcal{L})$.

If $\varphi \in \mathsf{Form}(\mathcal{L})$, we write $\varphi(x_1, ..., x_n) \in \mathsf{Form}(\mathcal{L})$ to indicate that $\{x_1, ..., x_n\} \supseteq \mathsf{Free}[\varphi]$.

2.2 Elementary transformations

Metadefinition 2.16 (Negation of a formula)**.** *The **negation** of a formula* f *is defined recursively as follows:*

$$\begin{array}{llll} \mathsf{Negate}[\mathsf{v}_1 = \mathsf{v}_2] & \stackrel{def}{=} \mathsf{v}_1 \neq \mathsf{v}_2, & \mathsf{Negate}[\mathsf{v}_1 \in \mathsf{v}_2] & \stackrel{def}{=} \mathsf{v}_1 \notin \mathsf{v}_2, \\ \mathsf{Negate}[(\mathsf{f})] & \stackrel{def}{=} (\mathsf{Negate}[\mathsf{f}]), & \mathsf{Negate}[\neg(\mathsf{f})] & \stackrel{def}{=} \mathsf{f}, \\ \mathsf{Negate}[\forall \mathsf{v}(\mathsf{f})] & \stackrel{def}{=} \exists \mathsf{v}\, \mathsf{Negate}(\mathsf{f}), & \mathsf{Negate}[\exists \mathsf{v}(\mathsf{f})] & \stackrel{def}{=} \forall \mathsf{v}\, \mathsf{Negate}(\mathsf{f}), \end{array}$$

$$\begin{array}{ll} \mathsf{Negate}[\mathsf{f}_1 \wedge ... \wedge \mathsf{f}_\mathsf{n}] & \stackrel{def}{=} \mathsf{Negate}[\mathsf{f}_1] \vee ... \vee \mathsf{Negate}[\mathsf{f}_\mathsf{n}], \\ \mathsf{Negate}[\mathsf{c}_1 \vee ... \vee \mathsf{c}_\mathsf{n}] & \stackrel{def}{=} \mathsf{Negate}[\mathsf{c}_1] \wedge ... \wedge \mathsf{Negate}[\mathsf{c}_\mathsf{n}]. \end{array}$$

Lemma 2.17 ("Negate" lemma)**.** *For all* $\mathsf{f} \in \mathsf{Form}(\mathcal{L})$*,*

$$\neg(f) \equiv \mathsf{Negate}(\mathsf{f})$$

by pure logic alone.

Metadefinition 2.18 (Expansion of the reach of a quantifier)**.** *Let* $\odot$ *be one of* $\{\wedge, \vee\}$*, let* f_i *be formulas,* $1 \leq i \leq n$*, let* $j \in \mathbb{N}$ *such that* $1 \leq j \leq n$*, and assume that* $\mathsf{f}_j = \exists \mathsf{v} \mathsf{f}'_j$*. If* f'_j *is of the form*

$$\mathsf{g}_1 \odot ... \odot \mathsf{g}_m, \tag{6}$$

and $\mathsf{v} \notin \mathsf{Free}(\mathsf{f}_i)$ *for all* $i \neq j$*, then* $\mathsf{ExpandExists}[\mathsf{f}_1 \odot ... \odot \mathsf{f}_j \odot ... \odot \mathsf{f}_n, \mathsf{j}]$ *is defined as*

$$\exists \mathsf{v}(\mathsf{f}_1 \odot ... \odot \mathsf{f}_{j-1} \odot \mathsf{g}_1 \odot ... \odot \mathsf{g}_m \odot \mathsf{f}_{j+1} \odot ... \odot \mathsf{f}_n);$$

if f'_j *is not of the form (6) and* $\mathsf{v} \notin \mathsf{Free}(\mathsf{f}_i)$ *for all* $i \neq j$*, then*

$$\mathsf{ExpandExists}[\mathsf{f}_1 \odot ... \odot \mathsf{f}_j \odot ... \odot \mathsf{f}_n, \mathsf{j}] \stackrel{def}{=} \exists \mathsf{v}(\mathsf{f}_1 \odot ... \odot \mathsf{f}'_j \odot ... \odot \mathsf{f}_n);$$

in all other cases, $\mathsf{ExpandExists}$ *is undefined.*

Similarly, if $\mathsf{f}_j = \forall \mathsf{v} \mathsf{f}'_j$*, then if* f'_j *is of the form (6) and* $\mathsf{v} \notin \mathsf{Free}(\mathsf{f}_i)$ *for all* $i \neq j$*, then*

$$\begin{array}{l} \mathsf{ExpandForall}[\mathsf{f}_1 \odot ... \odot \mathsf{f}_j \odot ... \odot \mathsf{f}_n, \mathsf{j}] \stackrel{def}{=} \\ \forall \mathsf{v}(\mathsf{f}_1 \odot ... \odot \mathsf{f}_{j-1} \odot \mathsf{g}_1 \odot ... \odot \mathsf{g}_m \odot \mathsf{f}_{j+1} \odot ... \odot \mathsf{f}_n); \end{array}$$

if f'_j *is not of the form (6) and* $\mathsf{v} \notin \mathsf{Free}(\mathsf{f}_i)$ *for all* $i \neq j$*, then*

$$\mathsf{ExpandForall}[\mathsf{f}_1 \odot ... \odot \mathsf{f}_j \odot ... \odot \mathsf{f}_n, \mathsf{j}] \stackrel{def}{=} \forall \mathsf{v}(\mathsf{f}_1 \odot ... \odot \mathsf{f}'_j \odot ... \odot \mathsf{f}_n);$$

in all other cases, $\mathsf{ExpandForall}$ *is undefined.*

The reason to distiguish two cases in the definitions above is to eliminate redundant parentheses in the results of the metaoperations.

Lemma 2.19 (“ExpandForall” or “ExpandExists” lemma)**.** *Let* $\mathsf{f} \in \mathsf{Form}(\mathcal{L})$ *and* $\mathsf{i} \in \mathbb{N}$*. In all cases where* $\mathsf{ExpandExists}[\mathsf{f}, \mathsf{i}]$ *is defined,*

$$\mathsf{f} \equiv \mathsf{ExpandExists}[\mathsf{f}, \mathsf{i}]$$

by pure logic alone. Similarly, in all cases where $\mathsf{ExpandForall}[\mathsf{f}, \mathsf{i}]$ *is defined,*

$$\mathsf{f} \equiv \mathsf{ExpandForall}[\mathsf{f}, \mathsf{i}]$$

by pure logic alone.

Metadefinition 2.20 (Moving an existential quantifier to the beginning of a formula)**.** *Let* f *be of the form*

$$(\exists \mathsf{a}_1 \in \mathsf{b}_1)...(\exists \mathsf{a}_n \in \mathsf{b}_n)\exists \mathsf{c}\mathsf{g}.$$

If $\mathsf{c} \neq \mathsf{a}_i, \mathsf{b}_i$ *for all* $1 \leq i \leq n$*, then*

$$\mathsf{MoveUp}[\mathsf{f}, \mathsf{n}+1] \stackrel{def}{=} \exists \mathsf{c}(\exists \mathsf{a}_1 \in \mathsf{b}_1)...(\exists \mathsf{a}_n \in \mathsf{b}_n)\exists \mathsf{g};$$

in all other cases, MoveUp *is undefined.*

Lemma 2.21 (”MoveUp” lemma)**.** *Let* $\mathsf{f} \in \mathsf{Form}(\mathcal{L})$ *and* $\mathsf{n} \in \mathbb{N}$*. In all cases where* $\mathsf{MoveUp}[\mathsf{f}, \mathsf{n}]$ *is defined,*

$$\mathsf{f} \equiv \mathsf{MoveUp}[\mathsf{f}, \mathsf{n}]$$

by pure logic alone.

Lemma 2.22 (Negation and bounded quantifiers)**.** *For all* $\mathsf{a}, \mathsf{b} \in \mathsf{Var}(\mathcal{L})$ *and all* $\mathsf{f} \in \mathsf{Form}(\mathcal{L})$*,*

(a) $\neg[(\forall \mathsf{a} \in \mathsf{b})\mathsf{f}] \equiv (\exists \mathsf{a} \in \mathsf{b})(\neg \mathsf{f})$*, and*
(b) $\neg[(\exists \mathsf{a} \in \mathsf{b})\mathsf{f}] \equiv (\forall \mathsf{a} \in \mathsf{b})(\neg \mathsf{f})$

by pure logic.

Lemma 2.23 (Introduction of spurious quantifiers)**.** *If* $\mathsf{f} \in \mathsf{Form}(\mathcal{L})$ *and* $\mathsf{v} \in \mathsf{Var}(\mathcal{L}) \setminus \mathsf{Free}[\mathsf{f}]$*, then*

(a) $\mathsf{f} \equiv \forall \mathsf{v}\mathsf{f}$*, and*
(b) $\mathsf{f} \equiv \exists \mathsf{v}\mathsf{f}$

by pure logic.

Lemma 2.24. *Let* $\mathsf{a}, \mathsf{a}_1, \mathsf{a}_2 \in \mathsf{Var}(\mathcal{L})$*, and let* $\mathsf{f} \in \mathsf{Form}(\mathcal{L})$*. Then,*

$$\exists \mathsf{a}_1 \exists \mathsf{a}_2(\mathsf{a}_1 \in \mathsf{a} \wedge \mathsf{a}_2 \in \mathsf{a} \wedge \mathsf{f}) \equiv (\exists \mathsf{a}_1 \in \mathsf{a})(\exists \mathsf{a}_2 \in \mathsf{a})\mathsf{f}.$$

Proof. Apply definition 2.6 and the MoveUp lemma. □

Metadefinition 2.25. *(a)*

$$\mathsf{Particularize}[\exists \mathsf{a}\mathsf{f}(\mathsf{a}, \vec{\mathsf{b}}); \forall \mathsf{a}\mathsf{g}(\mathsf{a}, \vec{\mathsf{b}})] \stackrel{def}{=} \exists \mathsf{a}(\mathsf{f}(\mathsf{a}, \vec{\mathsf{b}}) \wedge \mathsf{g}(\mathsf{a}, \vec{\mathsf{b}})).$$

(b) Let n *be an integer* ≥ 2*, let* $\vec{\mathsf{a}}$ *be a sequence of* n *variables of* $\mathcal{L}$*, let* $\vec{z} \in \mathsf{Var}(\mathcal{L})$*, and let* $\mathsf{f}(\vec{\mathsf{a}}, \vec{\mathsf{b}}), \mathsf{g}(\vec{\mathsf{a}}, \vec{\mathsf{b}}) \in \mathsf{Form}(\mathcal{L})$*. Then,*

$$\mathsf{Particularize}_\mathsf{n}[\exists \mathsf{a}_1...\exists \mathsf{a}_\mathsf{n}\mathsf{f}(\vec{\mathsf{a}}, \vec{\mathsf{b}}); \forall \mathsf{a}_1...\forall \mathsf{a}_\mathsf{n}\mathsf{g}(\vec{\mathsf{a}}, \vec{\mathsf{b}})] \stackrel{def}{=} \exists \mathsf{a}_1...\exists \mathsf{a}_\mathsf{n}(\mathsf{f}(\vec{\mathsf{a}}, \vec{\mathsf{b}}) \wedge \mathsf{g}(\vec{\mathsf{a}}, \vec{\mathsf{b}})).$$

Lemma 2.26 (The "$\mathsf{Particularize}$" lemma)**.** *Let* $\mathsf{a}, \vec{\mathsf{b}} \in \mathsf{Var}(\mathcal{L})$*, and let* $\mathsf{f}(\mathsf{a}, \vec{\mathsf{b}}), \mathsf{g}(\mathsf{a}, \vec{\mathsf{b}}) \in \mathsf{Form}(\mathcal{L})$*. Then*

$$\exists \mathsf{a} \mathsf{f}(\mathsf{a}, \vec{\mathsf{b}}) \wedge \forall \mathsf{a} \mathsf{g}(\mathsf{a}, \vec{\mathsf{b}}) \rightarrow \mathsf{Particularize}[\exists \mathsf{a} \mathsf{f}(\mathsf{a}, \vec{\mathsf{b}}); \forall \mathsf{a} \mathsf{g}(\mathsf{a}, \vec{\mathsf{b}})].$$

The case with n variables is an immediate consequence.

A set a is **transitive** iff every element of a is a subset of a, that is, if $(\forall \mathsf{b} \in \mathsf{a})(\forall \mathsf{c} \in \mathsf{b})(\mathsf{c} \in \mathsf{a})$.

Metadefinition 2.27 (Transitive set)**.**

$$\mathsf{Tran}[a; b, c] \stackrel{def}{=} (\forall \mathsf{b} \in \mathsf{a})(\forall \mathsf{c} \in \mathsf{b})(\mathsf{c} \in \mathsf{a}).$$

3 Denoting complexities

Metadefinition 3.1. *A formula is Δ_0 (or Σ_0, or Π_0), if all its quantifiers are bounded. If $\varphi(x_1, ...x_k, \vec{z})$ is a Π_n formula, we say that*

$$\exists x_1 ... \exists x_k \varphi(x_1, ...x_k, \vec{z})$$

is a Σ_{n+1} formula. Similarly, if $\varphi(x_1, ...x_k, \vec{z})$ is a Σ_n formula, we say that

$$\forall x_1 ... \forall x_k \varphi(x_1, ...x_k, \vec{z})$$

is a Π_{n+1} formula.

Metadefinition 3.2. *Let $\varphi \in \Delta_0$. Then $|\varphi|$ is the **complexity of** φ, defined recursively as follows:*

a) If φ is atomic, then $|\varphi| = \cdot$.
b) If φ is $\neg\psi$, then $|\varphi| = \neg|\psi|$; $\neg\cdot$ simplifies to $\cdot$.
c) If φ is $\psi_1 \wedge ... \wedge \psi_n$, then $|\varphi| = |\psi_1| \wedge ... \wedge |\psi_n|$; if φ is $\psi_1 \vee ... \vee \psi_n$, then $|\varphi| = |\psi_1| \vee ... \vee |\psi_n|$; $\cdot \wedge \cdot$ simplifies to $\cdot$, as does $\cdot \vee \cdot$.
d) $|(\exists v \in s)\varphi| = \exists(|\varphi|)$, and $|(\forall v \in s)\varphi = \forall(|\varphi|)$; parentheses are omitted where not strictly necessary; strings of identical quantifiers are represented with subindex notation, i.e., $\exists_3 = \exists\exists\exists$, and $\forall_2 = \forall\forall$, etc.; $\forall_n(\cdot)$ simplifies to $\forall_n$, and $\exists_n(\cdot)$ simplifies to $\exists_n$.

Let C be the complexity of φ. Sometimes we will write $\varphi \in C$ to express that φ is of complexity C, enclosing φ between quotes to improve readability if necessary.

Examples

1) $|x = y| = \cdot$; $|x \in y| = \cdot$.
2) $|x \in y \vee y \in x| = |x \in y| \vee |y \in x| = \cdot \vee \cdot = \cdot$.
3) $|(\forall y \in x)(\forall z \in y)(z \in x)| = \forall_2(\cdot) = \forall_2$.
4) "$(\forall y \in x)(\forall z \in y)(z \in x)$" $\in \forall_2$.

Metadefinition 3.3. *If a formula φ is Σ_n (Π_n), $n > 0$, and its Δ_0 part has complexity c, we will say that φ is $\Sigma_n(\mathsf{c})$ ($\Pi_n(\mathsf{c})$).*

Example. Let φ be

$$\exists x_1 \exists x_2 \forall x_3 (x_1 \in a \vee (\forall y \in x_2)(\forall z \in x_3)(y \in z \vee z \in y)).$$

Clearly, φ is a Σ_2 formula; the Δ_0 part is

$$x_1 \in a \vee (\forall y \in x_2)(\forall z \in x_3)(y \in z \vee z \in y),$$

which is $\cdot \vee \forall_2$. Therefore, φ is a $\Sigma_2(\cdot \vee \forall_2)$ formula, or $\varphi \in \Sigma_2(\cdot \vee \forall_2)$.

4 Set Theory: the first axioms

We choose as our axioms a variation of Kripke-Platek set theory without infinity.

Whenever we have to state an axiom schema, we avoid making compromises, and parameterize the axiom schema in a class of formulas. We thus speak of Γ-Foundation, where $\Gamma \subseteq \text{Fm}(\mathcal{L})$; examples are "$\Pi_1$-Foundation" and "$(\forall \wedge (\cdot \vee \exists))$-Separation". This will permit us to postpone the election of the classes of formulas over which our axioms will be defined, while at the same time allowing us to build out the theory step by step, by picking only the axioms that are "absolutely necessary" for our *actual* proofs (for different proofs we would probably have a different set of axioms).

Notice that, in the context of the fine-structure analysis of a proof, the class Γ above will be finite, i.e., we will only need a finite number $\varphi_1, ..., \varphi_n$ of Separation (Foundation, etc.) axioms, that is $\Gamma = \{\varphi_1, ..., \varphi_n\}$.

4.1 "Elegant" axioms vs. "expressive" axioms

It is usual to express axioms of set theory in weak forms that are thought to be *more elegant* than stronger corresponding forms, and then present those stronger forms as theorems. For example, the axiom of Pair would be presented as

$$\forall x \ \forall y \ \exists z \ (x \in z \wedge y \in z); \tag{7}$$

this is in fact the option taken in Kunen [2], p. 12. Such an axiom guarantees that, given two sets x and y, there exists a set z that has x and y amongst its elements; it does not give us automatically a set that has as its only elements x and y. That is, the set z guaranteed to exist by the axiom might be "too big". The usual way of reasoning is to prove a short lemma using Δ_0-separation to separate a subset of z such that $(\forall w \in z)(w = x \vee w = y)$ and extensionality to prove that such set is unique. From then on, everybody proceeds as if the real axiom were

$$\forall x \ \forall y \ \exists ! z \ \forall w [w \in z \leftrightarrow (w = x \vee w = y)]; \tag{8}$$

after all, the lemma is easily demonstrated.

But if the axiom is really (7), then *every use of an unordered pair in a proof is assuming the use of three axioms: Pair, a form of Separation, and Extensionality*; and, clearly, this is not what we desired: we wanted to have a Pair axiom that allowed us to use pairs. The use of a form of Separation is specially inconvenient, because it expands unnecessarily the list of used axioms.

What is going on is very simple: by postulating the aparently simpler forms of our axioms and relegating the intuitive forms of the axioms to apparently trivial lemmas, we are contaminating forever all later proofs (that is, almost all set-theoretical proofs) with axioms that should not be there. The *cleaner* forms of the axioms are in fact *dirtier*, and what was supposed to make things *simpler* in fact makes things much *more complicated*. To express it in another way: nothing makes an axiom like (7) *more elegant* than (8), except accepted practice, and the fact that much respected set theorists, like Kunen [2], seem to prefer (7) to (8).[3]

[3] In fact, Kunen states that he has chosen the axioms so that they will be easy to check later, for example when building L, or forcing extensions.

In this context, it is illuminating to realize that Devlin [1], who is concerned, as we are, with questions of complexity, uses always the reputedly "non-elegant" forms of the axioms.

4.2 The Empty Set Axiom

$$\exists! x\ \forall y\ (y \notin x). \tag{Ept}$$

The unique empty set, guaranteed to exist by axiom (Ept), will be denoted, as usual, by "$\emptyset$".

4.3 The Extensionality Axiom

$$\forall a\ \forall b\ [\ \forall x\ (x \in a \leftrightarrow x \in b) \rightarrow (a = b)\ ]. \tag{Ext}$$

4.4 The Foundation Axiom

Metadefinition 4.1 (The founding metaoperation)**.** *Let* $\varphi(x, \vec{z}) \in \mathsf{Form}(\mathcal{L})$*, and* $y \in \mathsf{Var}(\mathcal{L}) \setminus \mathsf{Free}[\varphi(x, \vec{z})]$*. Then*

$$\mathsf{Found}[\exists x \varphi(x, \vec{z}), y] \stackrel{def}{=} \exists x[\varphi(x, \vec{z}) \wedge (\forall y \in x)(\neg \varphi(y, \vec{z}))].$$

In all other cases, Found *is undefined.*

Metadefinition 4.2 (Foundation formulas)**.** *Let* $\varphi(x, \vec{z}) \in \mathsf{Form}(\mathcal{L})$*, and let* $y \in \mathsf{Var}(\mathcal{L}) \setminus \mathsf{Free}[\varphi(x, \vec{z})]$*. Then*

$$\mathsf{AxFnd}[\exists x \varphi(x, \vec{z}), y]$$

is the formula

$$\forall \vec{z}\ (\exists x\ \varphi(x, \vec{z}) \rightarrow \mathsf{Found}[\exists x \varphi(x, \vec{z}), y]). \tag{Fnd}$$

In all other cases, AxFnd *is undefined.*

Axiom Schema 4.3 (Foundation)**.** *Let* $\Gamma \subseteq \mathsf{Form}(\mathcal{L})$*.* Γ*-Foundation is the axiom schema* $\mathsf{AxFnd}[\exists x \varphi(x, \vec{z}), y]$*, where* $\varphi(x, \vec{z}) \in \Gamma$ *and* $y \in \mathsf{Var}(\mathcal{L}) \setminus \mathsf{Free}[\varphi(x, \vec{z})]$*.*

When $\Gamma = \mathsf{Form}(\mathcal{L})$ *we will speak of "Full Foundation" or, more simply, "Foundation".*

Remark 4.4 ($\in$-Induction Theorem)**.** *The contrapositive of*

$$\mathsf{AxFnd}[\exists x \neg \varphi(x), y]$$

is

$$\forall x[((\forall y \in x)\varphi(y)) \rightarrow \varphi(x)] \rightarrow \forall x\ \varphi(x),$$

called ***$\in$-induction on*** φ*.*

5 Enumerations and quantifiers

5.1 The Pairing Axiom

$$\forall x \; \forall y \; \exists! z \; \forall w \; [w \in z \leftrightarrow (w = x \vee w = y)]. \tag{Pai}$$

Definition 5.1. *Let x, y be sets. The unique set z which contains as its only elements x and y, guaranteed to exist by the Pairing Axiom, is called the* ***(unordered) pair*** *with elements x and y, and is denoted $\{x, y\}$. In the special case where $x = y$, $\{x, y\}$ has only one element, namely $x = y$, and then $\{x, y\} = \{x\} = \{y\}$ is called the* ***singleton*** *of x and is denoted by $\{x\}$.*

5.2 The Union Axiom

$$\forall x \exists! y \forall z [z \in y \leftrightarrow (\exists u \in x)(z \in u)] \tag{Uni}$$

Definition 5.2. *Given a set x, the set of all elements of all elements of x, called the* ***union of*** *x, and guaranteed to exist by the Union axiom, is denoted by $\bigcup x$.*

5.3 Finite sets

Given three elements x, y and z, we can form $\{x, y\}$ and $\{z\}$ by Pairing, $\{\{x, y\}, \{z\}\}$ again by Pairing, and $\bigcup\{\{x, y\}, \{z\}\}$ by Union. This set is normally denoted by $\{x, y, z\}$, since it is trivially seen that its only elements are x, y, and z. A similar operation can be effected for sets of n elements, $n \geq 3$, i.e. to form $\{x_1, ..., x_n\}$ for any $n \in \mathbb{N}$.

It is interesting to notice that forming sets of more that two elements requires the use of the Union axiom, while forming sets of two elements does not. This is clearly an annoying assimetry, which could be overcome by choosing a stronger axiom that allowed forming new sets from any given (syntactically) finite collection of sets.

How can one express in the pure language of set theory that a set is finite? A finite set $f = \{x_1, x_2, ..., x_n\}$ can be completely described by the $\Delta_0(\exists_n \forall)$ formula

$$\begin{aligned}&(\exists x_1 \in f)(\exists x_2 \in f)...(\exists x_n \in f)(\forall e \in f)\\&(e = x_1 \vee e = x_2 \vee ... \vee e = x_n).\end{aligned} \tag{9}$$

This is in itself uninteresting; but when we want to assert a property of the elements of f, two cases present themselves: in the first case (e.g., when n is small and the rôles of the x_i's are different), we may want to use a property of the form $\varphi(f, \vec{x}, \vec{z})$ as in

$$\begin{aligned}&(\exists x_1 \in f)(\exists x_2 \in f)...(\exists x_n \in f)[\varphi(f, \vec{x}, \vec{z}) \wedge\\&(\forall e \in f)(e = x_1 \vee e = x_2 \vee ... \vee e = x_n)];\end{aligned} \tag{10}$$

if, on the contrary, we need to express properties which are common to all of the elements of f, we will write

$$\begin{aligned}&(\exists x_1 \in f)(\exists x_2 \in f)...(\exists x_n \in f)(\forall e \in f)\\&[\varphi(f, e, \vec{x}, \vec{z}) \wedge (e = x_1 \vee e = x_2 \vee ... \vee e = x_n)].\end{aligned} \tag{11}$$

The main difference between (10) and (11) is the placement of φ: in the first case, if φ is Σ_1 then (10) can be transformed into a Σ_1 formula by an application of MoveUp after ExpandExists; in the second case we will need to use Collection first (after ExpandExists).

Metadefinition 5.3 (Enumerations)**.** *Let $f, e, \vec{z}, \vec{x} = x_1, ..., x_n$ be pairwise different variables. Then,*

(a) Enum$[\varphi(f, \vec{x}, \vec{z}), f, e, \vec{x}]$ *is the formula (10), and*
(b) Enum$[\varphi(f, e, \vec{x}, \vec{z}), f, e, \vec{x}]$ *is the formula (11).*

Example 5.4. *a)* Enum$[x = x, f, e, x, y]$ *"says" that f is an (unordered) pair with elements x and y, i.e., $f = \{x, y\}$.*

b) Enum$[x \cap y = \emptyset, f, e, x, y]$ *"says" that f is an (unordered) pair of disjoint elements.*

c) Enum[Tran$[e]$$, f, e, x, y, z]$ *"says" that f is a set with at most three elements which are all transitive.*

5.4 Collapsing quantifiers

For our purposes, a fundamental use of Enum will be to collapse existential quantifiers (for example, to help to reduce the complexity of a formula). It is clear that if we have

$$\exists x_1 \exists x_2 \varphi(x_1, x_2, \vec{z})$$

then we can find a pair $x = \{x_1, x_2\}$ such that

$$\exists x \, \mathsf{Enum}[\varphi(x_1, x_2, \vec{z}), x, e, x_1, x_2],$$

and viceversa; the same is true when dealing with n variables. As we will use this operation quite frequently, we introduce a definition for it:

Metadefinition 5.5 (The collapsing operation)**.** *Let $\mathsf{n} \geq 2$ be an integer, let $\vec{x}$ be a sequence of* n *variables of $\mathcal{L}$, let $\varphi(\vec{x}, \vec{z}) \in \mathsf{Form}(\mathcal{L})$, and let $e, y \in \mathsf{Var}(\mathcal{L})$ be new variables. Then*

$$\mathsf{Collapse_n}[\exists x_1 ... \exists x_n \varphi(\vec{x}, \vec{z}), y; e] \stackrel{def}{=} \exists y \, \mathsf{Enum}[\varphi(\vec{x}, \vec{z}), y, e, \vec{x}].$$

The subindex n *can be dropped when it is clear from the context.*

The next theorem proves that we can collapse existential quantifiers at cost almost zero (i.e., by using only Pairing, and, if $\mathsf{n} > 2$, Union).

Theorem Schema 5.6 (Collapsing existentials (Pai,Uni))**.** *Let $\mathsf{n} \geq 2$ be an integer, let $\vec{x}$ be a sequence of* n *variables of $\mathcal{L}$, let $\varphi(\vec{x}, \vec{z}) \in \mathsf{Form}(\mathcal{L})$, where all the $\vec{x}$ and $\vec{z}$ are pairwise different, and let y, e be new variables. Then*

$$\exists x_1 ... \exists x_n \varphi(\vec{x}, \vec{z}) \equiv \mathsf{Collapse_n}[\varphi(\vec{x}, \vec{z}), y, e]$$

by Pairing (and, if $n > 2$, Union).

Proof. We prove the theorem for the case $\mathsf{n} = 2$. From the Pairing Axiom we derive easily

$$\forall x_1 \forall x_2 \exists y[(x_1 \in x) \wedge (x_2 \in x) \wedge (\forall e \in y)(e = x_1 \vee e = x_2)]. \quad (12)$$

$\Rightarrow$) Assume that

$$\exists x_1 \exists x_2 \varphi(x_1, x_2, \vec{z}). \quad (13)$$

Then, by Particularize$[(13), (12)]$,

$$\exists x_1 \exists x_2[\varphi(x_1, x_2, \vec{z}) \wedge \exists y[(x_1 \in x) \wedge (x_2 \in x) \wedge (\forall e \in y)(e = x_1 \vee e = x_2)]].$$

By $\mathsf{ExpandExists}$,

$$\exists x_1 \exists x_2 \exists y[\varphi(x_1, x_2, \vec{z}) \wedge (x_1 \in y) \wedge (x_2 \in y) \wedge (\forall e \in y)(e = x_1 \vee e = x_2)].$$

By commuting existentials and reordering the conjunction,

$$\exists y \exists x_1 \exists x_2[(x_2 \in y) \wedge (x_1 \in y) \wedge \varphi(x_1, x_2, \vec{z}) \wedge (\forall e \in y)(e = x_1 \vee e = x_2)].$$

By lemma 2.24,

$$\exists y(\exists x_1 \in y)(\exists x_2 \in y)[\varphi(x_1, x_2, \vec{z}) \wedge (\forall e \in y)(e = x_1 \vee e = x_2)].$$

By lemma 2.23, $\varphi(x_1, x_2, \vec{z}) \leftrightarrow \forall e \varphi(x_1, x_2, \vec{z})$, and by pure logic, $\forall e A \wedge \forall e B \rightarrow \forall e(A \wedge B)$; therefore,

$$\exists y(\exists x_1 \in y)(\exists x_2 \in y)\forall e[\varphi(x_1, x_2, \vec{z}) \wedge (e \in y \rightarrow (e = x_1 \vee e = x_2)].$$

Similarly, $\varphi(x_1, x_2, \vec{z})$ implies $e \in y \rightarrow \varphi(x_1, x_2, \vec{z})$, and therefore

$$\begin{aligned} &[\varphi(x_1, x_2, \vec{z}) \wedge (e \in y \rightarrow (e = x_1 \vee e = x_2)] \rightarrow \\ &[(e \in y \rightarrow \varphi(x_1, x_2, \vec{z})) \wedge (e \in y \rightarrow (e = x_1 \vee e = x_2)]. \end{aligned}$$

And since

$$\begin{aligned} &[(e \in y \rightarrow \varphi(x_1, x_2, \vec{z})) \wedge (e \in y \rightarrow (e = x_1 \vee e = x_2)] \rightarrow \\ &[e \in y \rightarrow (\varphi(x_1, x_2, \vec{z}) \wedge (e = x_1 \vee e = x_2)] \end{aligned}$$

we have that

$$\exists y(\exists x_1 \in y)(\exists x_2 \in y)(\forall e \in y)[\varphi(x_1, x_2, \vec{z}) \wedge (e = x_1 \vee e = x_2)].$$

$\Leftarrow$) The reverse direction is much easier: if

$$\exists y(\exists x_1 \in y)(\exists x_2 \in y)(\forall e \in y)[\varphi(x_1, x_2, \vec{z}) \wedge (e = x_1 \vee e = x_2)]$$

then clearly

$$\exists y \exists x_1 \exists x_2(\forall e \in x)[\varphi(x_1, x_2, \vec{z}) \wedge (e = x_1 \vee e = x_2)],$$

and therefore

$$\exists y \exists x_1 \exists x_2(\forall e \in x)\varphi(x_1, x_2, \vec{z}).$$

And since e and y do not occur free in φ,

$$\exists x_1 \exists x_2 \varphi(x_1, x_2, \vec{z}).$$

$\square$

6 Separation and collection

6.1 The Separation Axiom

Metadefinition 6.1 (Separation formulas)**.** *Let* $\varphi(y,\vec{z}) \in \mathsf{Form}(\mathcal{L})$*, and let* $a, x \in \mathsf{Var}(\mathcal{L})$ *such that* $x \notin \mathsf{Free}[\varphi(y,\vec{z})]$*. Then*

$$\mathsf{AxSep}[\varphi(y,\vec{z}), a, x, y] \stackrel{def}{=} \forall\vec{z}\ \forall a\ \exists x\ \forall y\ (y \in x \leftrightarrow y \in a \wedge \varphi(y,\vec{z})). \quad \text{(Sep)}$$

In all other cases AxSep *is undefined.*

Axiom Schema 6.2 (Separation)**.** *Let* $\Gamma \subseteq \mathsf{Form}(\mathcal{L})$*.* Γ*-Separation is the axiom-schema* $\mathsf{AxSep}[\varphi(y,\vec{z}), a, x, y]$ *where* $a, x, y, \vec{z} \in \mathsf{Var}(\mathcal{L})$*,* $\varphi(y,\vec{z}) \in \Gamma$*, and* $x \notin \mathsf{Free}[\varphi(y,\vec{z})]$*.*

What the Separation axiom expresses is that, given a set a and a property $\varphi(z)$ of the elements of a, we can "separate" another set x that has as its elements exactly those elements of a which have the property φ. This is usually denoted

$$x = \{y \in a : \varphi(y)\}.$$

6.2 The Collection Axiom

Metadefinition 6.3 (The Collection metaoperation)**.** *Let* $\varphi(x,y,\vec{z}) \in \mathsf{Form}(\mathcal{L})$*, and let* $a, w \in \mathsf{Var}(\mathcal{L})$ *such that* $a \neq w$ *and* $w \notin \mathsf{Free}[\varphi(x,y,\vec{z})]$*. Then*

$$\mathsf{Collect}[(\forall x \in a)\exists y\varphi(x,y,\vec{z})] \stackrel{def}{=} \exists w(\forall x \in a)(\exists y \in w)\varphi(x,y,\vec{z}).$$

In all other cases, Collect *is undefined.*

Metadefinition 6.4 (Collection formulas)**.** *Let* $\varphi(x,y,\vec{z}) \in \mathsf{Form}(\mathcal{L})$*, and* $a, w \in \mathsf{Var}(\mathcal{L})$ *such that* $a \neq w$ *and* $w \notin \mathsf{Free}[\varphi(x,y,\vec{z})]$*. Then*

$$\mathsf{AxColl}[(\forall x \in a)\exists y\varphi(x,y,\vec{z}), w]$$

is the formula

$$(\forall x \in a)\exists y\varphi(x,y,\vec{z}) \to \mathsf{Collect}[(\forall x \in a)\exists y\varphi(x,y,\vec{z}), w]. \quad \text{(Coll)}$$

In all other cases, AxColl *is undefined.*

Axiom Schema 6.5 (Collection)**.** *Let* $\Gamma \subseteq \mathsf{Form}(\mathcal{L})$*.* Γ*-Collection is the axiom schema* $\mathsf{AxColl}[(\forall x \in a)\exists y\varphi(x,y,\vec{z}), w]$*, for all* $\varphi(x,y,\vec{z}) \in \Gamma$ *and all* $a, w \in \mathsf{Var}(\mathcal{L})$ *such that* $a \neq w$ *and* $w \notin \mathsf{Free}[\varphi(x,y,\vec{z})]$*.*

When $\Gamma = \mathsf{Form}(\mathcal{L})$ *we will speak of "Full Collection" or, more simply, "Collection".*

Theorem Schema 6.6 (Collection equivalence (Γ-Coll))**.** *Let* $\Gamma \subseteq \mathsf{Form}(\mathcal{L})$*. For each* $\varphi(x,y,\vec{z}) \in \Gamma$ *and all* $a, w \in \mathsf{Var}(\mathcal{L})$ *such that* $a \neq w$ *and* $w \notin \mathsf{Free}[\varphi(x,y,\vec{z})]$*,*

$$(\forall x \in a)\exists y\varphi(x,y,\vec{z}) \equiv \mathsf{Collect}[(\forall x \in a)\exists y\varphi(x,y,\vec{z}), w]$$

by $\mathsf{AxColl}[(\forall x \in a)\exists y\varphi(x,y,\vec{z}), w]$*.*

Proof. One direction is Γ-Collection, and the other is immediate. ☐

6.3 Strengthening Collection

Theorem Schema 6.7 (Pai,Uni)**.** *Let* $\varphi(x, y, \vec{z}) \in \mathsf{Form}(\mathcal{L})$ *be of the form* $\exists v \varphi_0(v, x, y, \vec{z})$*, and let* $u \in \mathsf{Var}(\mathcal{L}) \setminus \mathsf{Free}[\varphi(x, y, \vec{z})]$*. For any new variables* r*,* w *and* w'*,*

$$
\begin{array}{lll}
(\forall x \in a)\exists y \varphi(x, y, \vec{z}) \equiv \exists r & [& (\forall x \in a)(\exists y \in r)\varphi(x, y, \vec{z}) \\
& \wedge & (\forall y \in r)(\exists x \in a)\varphi(x, y, \vec{z}) \\
&] &
\end{array}
$$

by

$$\mathsf{AxColl}[(\forall x \in a)\exists u \mathsf{Enum}[\varphi_0(v, x, y, \vec{z}), u, e, y, v], w], \tag{14}$$

and

$$\mathsf{AxSep}[(\exists x \in a)(\exists v \in w')\varphi_0(v, x, y, \vec{z}), w', r, y]. \tag{15}$$

Proof. One direction is immediate. To prove the other direction, assume that

$$(\forall x \in a)\exists y \varphi(x, y, \vec{z}),$$

that is,

$$(\forall x \in a)\exists y \exists v \varphi_0(v, x, y, \vec{z}),$$

If $u = \{y, v\}$, then clearly

$$(\forall x \in a)\exists u(\exists y \in u)(\exists v \in u)(\varphi_0(v, x, y, \vec{z}) \wedge (\forall e \in u)(e = y \vee e = v)),$$

i.e.,

$$(\forall x \in a)\exists u \mathsf{Enum}[\varphi_0(v, x, y, \vec{z}), u, e, y, v].$$

By (14), there exists a set w such that

$$(\forall x \in a)(\exists u \in w)(\exists y \in u)(\exists v \in u)\varphi_0(v, x, y, \vec{z}). \tag{16}$$

Let $w' = \bigcup w$, which exists by Union, and consider

$$r = \{y \in w' : (\exists x \in a)(\exists v \in w')\varphi_0(v, x, y, \vec{z})\}, \tag{17}$$

which exists by (15). Since

$$(\exists v \in w')\varphi_0(v, x, y, \vec{z}) \to \exists v \varphi_0(v, x, y, \vec{z}),$$

i.e.,

$$(\exists v \in w')\varphi_0(v, x, y, \vec{z}) \to \varphi(x, y, \vec{z}),$$

clearly from (17) we get

$$(\forall y \in r)(\exists x \in a)\varphi(x, y, \vec{z});$$

on the other hand, (16) implies

$$(\forall x \in a)(\exists y \in w')(\exists v \in w')\varphi_0(v, x, y, \vec{z}),$$

hence

$$(\forall x \in a)(\exists y \in w')\varphi(x, y, \vec{z}),$$

and, by (17),

$$(\forall x \in a)(\exists y \in r)\varphi(x, y, \vec{z}).$$

□

Remark 6.8. *If in the proof of theorem 6.7 we drop the assumption of the Separation axiom (15), while keeping the assumption of the Collection axiom (14), we can still prove a strengthtened form of Collection:*

$$(\forall x \in a)\exists y \varphi(x, y, \vec{z}) \to \exists r(\forall x \in a)(\exists y \in r)\varphi(x, y, \vec{z}).$$

7 Tuples

7.1 Creating tuples

If we are given two sets x and y, saying that there exists a set $p = \{x, y\}$ is to say that a set p exists such that:

$$(x \in p \wedge y \in p \wedge (\forall e \in p)(e = x \vee e = y)).$$

If we want to express a property of x, y and p, we write

$$x \in p \wedge y \in p \wedge \varphi(x, y, p) \wedge (\forall e \in p)(e = x \vee e = y),$$

which is Σ_1 in case $\varphi \in \Sigma_1$ by ExpandExists; if we need to quantify over all $e \in p$, then we write

$$x \in p \wedge y \in p \wedge (\forall e \in p)((e = x \vee e = y) \wedge \varphi(x, y, p, e)),$$

which is Σ_1 when $\varphi \in \Sigma_1$ with n unbounded existential quantifiers, but we need (Strengthened) Collection to prove it for $n > 0$ ($n > 1$).

Metadefinition 7.1 (Pairing two sets)**.** *Let $x, y, p, e, \vec{z}$ be pairwise different variables. Then,*

(a) If $\varphi(x, y, p, \vec{z}) \in \mathsf{Form}(\mathcal{L})$, then

$$\mathsf{Pair}[\varphi(x, y, p, \vec{z}), x, y, p; e]$$

is the formula

$$x \in p \wedge y \in p \wedge \varphi(x, y, p, \vec{z}) \wedge (\forall e \in p)(e = x \vee e = y).$$

(b) If $\varphi(x, y, p, e, \vec{z}) \in \mathsf{Form}(\mathcal{L})$, then

$$\mathsf{Pair}[\varphi(x, y, p, e, \vec{z}), x, y, p, e]$$

is the formula

$$x \in p \wedge y \in p \wedge (\forall e \in p)((e = x \vee e = y) \wedge \varphi(x, y, p, e, \vec{z}).$$

Notice that the only difference between cases (a) and (b) above radicates in the fact that, in case (b), $e \in \mathsf{Free}(\varphi)$.

Lemma 7.2 (AxColl)**.** *Let $\varphi(x, y, p, e, \vec{z}) \in \Sigma_1$ be*

$$\exists v \varphi_0(v, x, y, p, e, \vec{z}),$$

with $\varphi_0 \in \Delta_0$, and let v' be a new variable. Then there exists a formula

$$\mathsf{Pair}_{\Sigma_1}[\varphi(x, y, p, e, \vec{z}), v', x, y, p, e,] \in \Sigma_1(\cdot \wedge \forall\exists(\cdot \wedge |\varphi_0|))$$

such that

$$\mathsf{Pair}[\varphi(x, y, p, e, \vec{z}), x, y, p, e,] \equiv \mathsf{Pair}_{\Sigma_1}[\varphi(x, y, p, e, \vec{z}), v', x, y, p, e,]$$

by $\mathsf{AxColl}[(\forall e \in p)\mathsf{ExpandExists}[(e = x \vee e = y) \wedge \exists v \varphi_0(v, x, y, p, e, \vec{z})], v']$.

Proof. Pick a new variable v', and consider

$$\begin{aligned}
&\mathsf{Pair}_{\Sigma_1}[\varphi(x, y, p, e, \vec{z}), v', x, y, p, e,] \stackrel{\text{def}}{=} \\
&\quad \mathsf{ExpandExists}[x \in p \wedge y \in p \wedge \\
&\quad\quad \mathsf{Collect}[v', (\forall e \in p) \\
&\quad\quad\quad \mathsf{ExpandExists}[(e = x \vee e = y) \wedge \exists v \varphi_0(v, x, y, p, e, \vec{z})] \\
&\quad\quad] \\
&\quad].
\end{aligned}$$

Recall from the definition of ExpandExists that if φ_0 is a conjunction, then outer parenthesis are automatically removed. ☐

7.2 Ordered pairs

7.2.1 Definition

Definition 7.3 (Kuratowski)**.** *Given two sets a and b, the **ordered pair** $\langle a, b\rangle$ is*

$$\langle a, b\rangle = \{\{a\}, \{a, b\}\}.$$

If $\langle a, b\rangle = \langle c, d\rangle$, then $a = c$ and $b = d$.

7.2.2 Complexity of "being an ordered pair"

Let $t = \langle x, y\rangle$, and assume that $t = \{p_1, p_2\}$, where $p_1 = \{x\}$ and $p_2 = \{x, y\}$. To express this fact in $\mathcal{L} = \{\in, =\}$ alone, we have to ensure that there are two elements p_1 and p_2 in t, that any element e of t is either p_1 or p_2, that $x \in p_1$ and $x, y \in p_2$, and that any element e_1 of p_1 is x, and that any element e_2 of p_2 is either x or y. If, additionally, we want to express some fact φ involving t, x, y, and possibly other variables $\vec{z}$, we are lead naturally to the following definition:

Metadefinition 7.4. *Let $\varphi(t, x, y, \vec{z}) \in \mathsf{Form}(\mathcal{L})$, and let p and e be stems. Then*

$$\begin{aligned}\mathsf{Tuple}[\varphi(t, x, y, \vec{z}), t, x, y; p, e] \stackrel{def}{=} \; & (\exists p_1 \in t)(\exists p_2 \in t)(\exists x \in p_1)(\exists y \in p_2) \\ & (\varphi(t, x, y, \vec{z})\wedge \\ & (\forall e \in t)(\forall e_1 \in p_1)(\forall e_2 \in p_2) \\ & ((e = p_1 \vee e = p_2) \wedge (e_1 = x) \\ & \wedge(e_2 = x \vee e_2 = y) \wedge (x \in p_2))).\end{aligned}$$

Theorem Schema 7.5 (Pai)**.** *If $\varphi(p, x, y, \vec{z}) \in \Delta_0$, then*

$$\mathsf{Tuple}[\varphi(t, x, y, \vec{z}), t, x, y; p, e] \in \Delta_0(\exists_4(|\varphi| \wedge \forall_3));$$

if $\varphi(t, x, y, \vec{z}) \in \Sigma_1$ is $\exists v\varphi_0(v, t, x, y, \vec{z})$, with $\varphi_0(v, t, x, y, \vec{z}) \in \Delta_0$, then there is a formula

$$\mathsf{Tuple}_{\Sigma_1}[\varphi(t, x, y, \vec{z}), t, x, y; p, e] \in \Sigma_1(\exists_4(|\varphi_0| \wedge \forall_3))$$

such that

$$\mathsf{Tuple}[\varphi(t, x, y, \vec{z}), t, x, y; p, e] \equiv \mathsf{Tuple}_{\Sigma_1}[\varphi(t, x, y, \vec{z}), t, x, y; p, e]$$

by Pairing.

Proof. The Δ_0 case is immediate; for the Σ_1 case, consider

$$\begin{aligned}&\mathsf{Tuple}_{\Sigma_1}[\varphi(t, x, y, \vec{z}), t, x, y; p, e] \stackrel{\text{def}}{=} \\ &\quad \mathsf{MoveUp}_5[\\ &\quad\quad (\exists p_1 \in t)(\exists p_2 \in t)(\exists x \in p_1)(\exists y \in p_2) \\ &\quad\quad \mathsf{ExpandExists}[\exists v\varphi_0(v, t, x, y, \vec{z})\wedge \\ &\quad\quad\quad (\forall e \in t)(\forall e_1 \in p_1)(\forall e_2 \in p_2) \\ &\quad\quad\quad (e = p_1 \vee e = p_2) \wedge (e_1 = x) \\ &\quad\quad\quad \wedge(e_2 = x \vee e_2 = y) \wedge (x \in p_2))] \\ &\quad].\end{aligned} \tag{18}$$

Recall from the definition of ExpandExists that if $\varphi_0(v, t, x, y, \vec{z})$ is a conjunction, then its outer parenthesis are removed as a further optimization. □

8 Classes, relations and functions

8.1 Classes

Definition 8.1 (Classes)**.** *Let $C(x, \vec{y})$ be a formula of $\mathcal{L}$ with all free variables shown. We will say that C **defines a class** (with parameters $\vec{y}$). Given a tuple of parameters $\vec{a}$, we will speak of **the class** $\mathbf{C}(\vec{a})$ **(or** $\mathbf{C}_{\vec{a}}$**)** in the following sense: $x \in \mathbf{C}_{\vec{a}}$ will be a shorthand for $C(x, \vec{a})$, and $x \notin \mathbf{C}_{\vec{a}}$ will be a shorthand for $\neg C(x, \vec{a})$. Similarly, given two classes $\mathbf{C}_{\vec{a}}$ and $\mathbf{D}_{\vec{b}}$, we write:*

$\{x : C(x, \vec{a})\}$	*as equivalent to*	$C(x, \vec{a})$;
$\mathbf{C}_{\vec{a}} \subseteq \mathbf{D}_{\vec{b}}$	*as a shorthand for*	$\forall x(C(x, \vec{a} \to D(x, \vec{b}))$;
$\mathbf{C}_{\vec{a}} \cup \mathbf{D}_{\vec{b}}$	*as a shorthand for*	$\{x : C(x, \vec{a}) \lor D(x, \vec{b})\}$;
$\mathbf{C}_{\vec{a}} \cap \mathbf{D}_{\vec{b}}$	*as a shorthand for*	$\{x : C(x, \vec{a}) \land D(x, \vec{b})\}$;
$\mathbf{C}_{\vec{a}} \setminus \mathbf{D}_{\vec{b}}$	*as a shorthand for*	$\{x : C(x, \vec{a}) \land \neg D(x, \vec{b})\}$.

8.2 Class functions

Definition 8.2 (Class functions)**.** *Let T be a $\mathcal{L}$-theory, and let $F(x, \vec{z}, y)$ be a formula with all free variables shown. We will say that F **defines a class function (in the theory T)** iff*

$$T \vdash \forall \vec{z}\ \forall x \exists! y F(x, \vec{z}, y),$$

that is,

$$T \vdash \forall \vec{z}\ \forall x \exists y \big[F(x, \vec{z}, y) \land \forall w (F(x, \vec{z}, w) \to y = w)\big].$$

***Functional notation**: sometimes we will write $y = \mathbf{F}(x, \vec{z})$ instead of $F(x, \vec{z}, y)$, and $y = \mathbf{F}_{\vec{z}}(x)$ instead of $\mathbf{F}(x, \vec{z}, y)$.*

8.3 Relations

Metadefinition 8.3 (Relations)**.** *Let $\varphi(r, a, b, \vec{z}) \in \mathsf{Form}(\mathcal{L})$, and let p and e be stems. Then*

$$\mathsf{Rel}[\varphi(r, x, y, \vec{z}), r, x, y; p, e] \stackrel{def}{=} (\forall t \in r)\mathsf{Tuple}[\varphi(r, x, y, \vec{z}), t, x, y; p, e].$$

Lemma 8.4.

$$\mathsf{Rel}[\varphi(r, x, y, \vec{z}), r, x, y; p, e] \in \forall\exists_4(|\varphi| \land \forall_3).$$

In many cases we will need to handle simultaneously several elements (i.e., several ordered pairs) of a relation. For example, when defining the concept of "being a function" f will be a function iff

$$(\langle x, y \rangle, \langle z, w \rangle \in f \land x = z) \to y = w.$$

Applying definition 8.3 directly, "f is a function" would be

$$\mathsf{Rel}[\mathsf{Rel}[x = z \to y = w, f, z, w], f, x, y],$$

which is $\forall\exists_4\forall_4\exists_4\forall_3$; however, a simpler definition exists:

$$\begin{aligned}
&(\forall p^1 \in f)(\forall p^2 \in f)\\
&(\exists p^1_1 \in p^1)(\exists p^1_2 \in p^1)(\exists p^2_1 \in p^2)(\exists p^2_2 \in p^2)\\
&(\exists x_1 \in p^1_1)(\exists y_1 \in p^1_2)(\exists x_2 \in p^2_1)(\exists y_2 \in p^2_2)\\
&(\forall e^1 \in p^1)(\forall e^2 \in p^2)(\forall e^1_1 \in p^1_1)(\forall e^1_2 \in p^1_2)(\forall e^2_1 \in p^2_1)(\forall e^2_2 \in p^2_2)\\
&(x_1 \in p^1_2 \land x_2 \in p^2_2 \land (e^1 = p^1_1 \lor e^1 = p^1_2) \land (e^2 = p^2_1 \lor e^1 = p^2_2) \land\\
&e^1_1 = x_1 \land e^2_1 = x_2 \land (e^1_2 = x_1 \lor e^1_2 = y_1) \land (e^2_2 = x_2 \lor e^2_2 = y_2) \land\\
&(x_1 = x_2 \to y_1 = y_2)),
\end{aligned}$$

which is $\forall_2\exists_8\forall_6$. In fact, only the last line expresses that f is a function; the rest of the formula is absolutely general:

Metadefinition 8.5. *Let* n *be a positive natural number. Then*

$$\mathsf{Tuples_n}[\varphi(r,\vec{p},\vec{e},\vec{x},\vec{y},\vec{z}),r,p,e,x,y]$$

is defined as

$$\begin{array}{l}(\forall p^1 \in r)...(\forall p^{\mathsf{n}} \in r)\\ (\exists p^1_1 \in p^1)(\exists p^1_2 \in p^1)...(\exists p^{\mathsf{n}}_1 \in p^{\mathsf{n}})(\exists p^{\mathsf{n}}_2 \in p^{\mathsf{n}})\\ (\exists x_1 \in p^1_1)(\exists y_1 \in p^1_2)...(\exists x_{\mathsf{n}} \in p^{\mathsf{n}}_1)(\exists y_{\mathsf{n}} \in p^{\mathsf{n}}_2)\\ (\forall e^1 \in p^1)...(\forall e^{\mathsf{n}} \in p^{\mathsf{n}})\\ (\forall e^1_1 \in p^1_1)(\forall e^1_2 \in p^1_2)...(\forall e^{\mathsf{n}}_1 \in p^{\mathsf{n}}_1)(\forall e^{\mathsf{n}}_2 \in p^{\mathsf{n}}_2)\\ (\bigwedge_{1\le \mathsf{i}\le \mathsf{n}}(x_{\mathsf{i}} \in p^{\mathsf{i}}_2 \wedge (e^{\mathsf{i}} = p^{\mathsf{i}}_1 \vee e^{\mathsf{i}} = p^{\mathsf{i}}_2) \wedge e^{\mathsf{i}}_1 = x_{\mathsf{i}} \wedge (e^{\mathsf{i}}_2 = x_{\mathsf{i}} \vee e^{\mathsf{i}}_2 = y_{\mathsf{i}}))\\ \wedge\varphi(r,\vec{p},\vec{x},\vec{y},\vec{z}))\end{array}$$

Clearly, for each positive n*,*

$$\mathsf{Tuples_n}[\varphi(r,\vec{p},\vec{e},\vec{x},\vec{y},\vec{z}),r,p,e,x,y] \in \forall_{\mathsf{n}}\exists_{4\mathsf{n}}\forall_{3\mathsf{n}}(\cdot \wedge |\varphi(r,\vec{p},\vec{x},\vec{y},\vec{z})|).$$

Notice that p, e, x and y are stems.

8.4 Functions

Definition 8.6. *We say that a relation f is a* ***function****, and write* $\mathrm{Fun}(f)$*, when $\langle x,y\rangle \in f$ and $\langle x,z\rangle \in f$ imply $y = z$.*

Metadefinition 8.7 (Functions)**.**

$$\mathsf{Fun}[\varphi(f,\vec{p},\vec{x},\vec{y},\vec{z}),f;p,e]$$

is defined as

$$\mathsf{Tuples_2}[x_1 = x_2 \to y_1 = y_2 \wedge \varphi(f,\vec{p},\vec{x},\vec{y},\vec{z}),f,p,e,x,y]$$

Metadefinition 8.8 (Functions f_1 and f_2 differ at x, i.e., $f_1(x) \neq f_2(x)$.)**.**

$$\begin{array}{lcl}\mathsf{FunDiff}[f_1,f_2,x;p,e,y] & \stackrel{def}{=} & (\exists p^1 \in f_1)(\exists p^2 \in f_2)\\ & & (\exists p^1_1 \in p^1)(\exists p^1_2 \in p^1)(\exists p^2_1 \in p^2)(\exists p^2_2 \in p^2)\\ & & (\exists y_1 \in p^1_2)(\exists y_2 \in p^2_2)\\ & & (\forall e^1 \in p^1)(\forall e^2 \in p^2)\\ & & (\forall e^1_1 \in p^1_1)(\forall e^1_2 \in p^1_2)(\forall e^2_1 \in p^2_1)(\forall e^2_2 \in p^2_2)\\ & & [(e^1 = p^1_1 \vee e^1 = p^1_2) \wedge (e^2 = p^2_1 \vee e^2 = p^2_2)\wedge\\ & & (e^1_1 = x) \wedge (e^2_1 = x)\wedge\\ & & (e^1_2 = x \vee e^1_2 = y_1) \wedge (e^2_2 = x \vee e^2_2 = y_2)\wedge\\ & & (y_1 \neq y_2)].\end{array}$$

Metadefinition 8.9 (Function f has value y at x, i.e., $f(x) = y$)**.**

$$\begin{array}{lcl}\mathsf{FunVal}[f,x,y;p,e] & \stackrel{def}{=} & (\exists p \in f)(\exists p_1 \in p)(\exists p_2 \in p)\\ & & (\forall e \in p)(\forall e_1 \in p_1)(\forall e_2 \in p_2)\\ & & ((e = p_1 \vee e = p_2) \wedge (e_1 = x) \wedge (e_2 = x \vee e_2 = y))\end{array}$$

Clearly, $\mathsf{FunVal}[f,x,y;p,e] \in \Delta_0(\exists_3\forall_3)$.

Metadefinition 8.10 ($x \in \operatorname{dom} f$)**.** *If f is a function, then saying that $x \in \operatorname{dom} f$ means that there exists a pair $p \in f$ such that x is its first component. But in the Kuratowski definition of ordered pairs, this means that x is an element of every element e of p:*

$$\mathsf{InDomain}[x, f; p, e] \overset{def}{=} (\exists p \in f)(\forall e \in p)(x \in e).$$

It is immediate that $\mathsf{InDomain}[x, f; p, e] \in \Delta_0(\exists\forall)$.

Definition 8.11 (Restriction of a function)**.** *Let f be a function, and a a set. The set*

$$f \restriction a = \{\langle x, y\rangle \in f : x \in a\}$$

is called the ***restriction of f to*** *a.*

Clearly, if $r = f \restriction a$ then all elements p of r have a first component $p_1 \in a$, and p_1 is the first component of an ordered pair p iff p_1 belongs to all the elements e_1 of p.

Metadefinition 8.12 (The $\mathsf{Restrict}$ metaoperation)**.**

$$\begin{aligned}\mathsf{Restrict}[f, a, r; p, e] \overset{def}{=} \quad & (\forall p \in f)(\exists e \in p)(\exists p_1 \in e)(\forall e_1 \in p) \\ & (p_1 \in e_1 \wedge (p \in r \leftrightarrow p_1 \in a)).\end{aligned}$$

Clearly,

$$\mathsf{Restrict}[f, a, r; p, e] \in \Delta_0(\forall\exists_2\forall).$$

Part II
Transfinite induction and recursion

9 Transfinite $\in$-induction and recursion

Recall from remark 4.4:

Theorem Schema 9.1 ($\in$-induction)**.** *Let $\varphi(x, \vec{z})$ be any formula. Then for all $\vec{z}$,*

$$\forall x[((\forall y \in x)\varphi(y, \vec{z})) \to \varphi(x, \vec{z})] \to \forall x\ \varphi(x, \vec{z}).$$

Our goal is to prove the transfinite $\in$-recursion theorem for Σ_1 (Δ_0) class functions. Namely if $F(x, z, \vec{a}, y)$ is a Σ_1 (Δ_0)class function, we want to find a Σ_1 class function $G(x, \vec{a}, y)$ such that for all $\vec{a}$,

$$\forall x(\mathbf{G}_{\vec{a}}(x) = \mathbf{F}_{\vec{a}}(x, \mathbf{G}_{\vec{a}} \restriction x)).$$

The structure of the proof is as follows: we first introduce in 9.2 *suitable functions*, which are set approximations to $\mathbf{G}$ and are readily seen to be definable by a Σ_1 formula (Theorem 9.3) and pairwise compatible (i.e., if σ_1 and σ_2 are suitable functions and $x \in \mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2)$, then $\sigma_1(x) = \sigma_2(x)$ [Theorem 9.4]). We next define in 9.5 a class relation $\mathbf{G}_{\vec{a}}$ as follows: $\mathbf{G}_{\vec{a}}(x, y)$ holds iff there exists a suitable function σ such that $y = \sigma(x)$, that is, iff we can build a set approximation of $\mathbf{G}$ such that $y = \mathbf{G}_{\vec{a}}(x)$ (we can speak of *a* set approximation of $\mathbf{G}$ because we just proved that any two such approximations are compatible). We finally prove that $\mathbf{G}_{\vec{a}}$ has a Σ_1 equivalence (Theorem 9.6) and is total (Theorem 9.8).

For the rest of the section we will assume that $F(x, z, \vec{a}, y)$ is either Δ_0, or Σ_1 of the form

$$\exists v F_0(v, x, z, \vec{a}, y),$$

where $F_0 \in \Delta_0$ (if $F \in \Sigma_1$ with more than one unbounded existential quantifier, apply Collapse first).

9.1 Suitable functions

Definition 9.2 (Suitable functions)**.** *A function σ is **suitable** (for F and for a set of parameters $\vec{a}$), written $S^{\vec{a}}(\sigma)$, iff*

$$\mathrm{Fun}(\sigma) \wedge \mathrm{Tran}(\mathrm{dom}(\sigma)) \wedge (\forall x \in \mathrm{dom}(\sigma))(\sigma(x) = \mathbf{F}_{\vec{a}}(x, \sigma \restriction x)). \quad (19)$$

Theorem 9.3 ("To be suitable for F" has a Σ_1 equivalence if $F \in \Delta_0 \cup \Sigma_1$)**.** *If $F \in \Delta_0$, then there exists a formula $S^{\vec{a}}_{\Sigma_1}(\sigma)$ such that*

$$S^{\vec{a}}(\sigma) \equiv S^{\vec{a}}_{\Sigma_1}(\sigma) \in \Sigma_1(\forall\exists_5(\forall\exists_2\forall \wedge |F| \wedge \forall_3) \wedge \forall_3\exists_{12}\forall_9)$$

by Δ_0-Collection; if F is $\exists v F_0(v, x, z, \vec{a}, y)$, with $F_0 \in \Delta_0$, then there exists a formula $S^{\vec{a}}_{\Sigma_1}(\sigma)$ such that

$$S^{\vec{a}}(\sigma) \equiv S^{\vec{a}}_{\Sigma_1}(\sigma) \in \Sigma_1(\forall_3\exists_{12}\forall_9 \wedge \forall\exists_5(\forall_3 \wedge \exists_2(\forall\exists_2\forall \wedge |F_0| \wedge \forall)))$$

by Δ_0-Collection.

Proof. Consider the definition of suitable functions. $\mathrm{Fun}(\sigma)\wedge\mathrm{Tran}(\mathrm{dom}(\sigma))$ can be expressed as follows

$$\begin{array}{lll}\mathsf{Tuples}_3 & [& (x_1 = x_2 \rightarrow y_1 = y_2) \wedge ((x_1 \in x_2 \wedge x_2 \in x_3) \rightarrow x_1 \in x_3)), \\ & & \sigma, p, e, x, y \\ &]. & \end{array} \tag{20}$$

Following definition 8.5, $\mathrm{Fun}(\sigma) \wedge \mathrm{Tran}(\mathrm{dom}(\sigma))$ is $\Delta_0(\forall_3\exists_{12}\forall_9)$. Additionally, since we can count on σ being a function,

$$(\forall x \in \mathrm{dom}(\sigma))(\sigma(x) = \mathbf{F}_{\vec{a}}(x, \sigma \restriction x))$$

means that every element of σ is a tuple $t = \langle x, y\rangle$ such that $y = \mathbf{F}_{\vec{a}}(x, \sigma \restriction x)$:

$$(\forall t \in \sigma)\mathsf{Tuple}[y = \mathbf{F}_{\vec{a}}(x, \sigma \restriction x), t, x, y]. \tag{21}$$

In turn, $y = \mathbf{F}_{\vec{a}}(x, \sigma \restriction x)$ is $\exists z(z = \sigma \restriction x \wedge F(x, z, \vec{a}, y))$, so that (21) becomes

$$(\forall t \in \sigma)\mathsf{Tuple}[\exists z(\mathsf{Restrict}[\sigma, x, z] \wedge F(x, z, \vec{a}, y)), t, x, y]. \tag{22}$$

Recall from definition 8.12 that $\mathsf{Restrict}[\sigma, x, z] \in \Delta_0(\forall\exists_2\forall)$.

Case $F \in \Delta_0$: If $F \in \Delta_0$, by theorem 7.5 we can interchange Tuple by $\mathsf{Tuple}_{\Sigma_1}$, hence

$$\mathsf{Tuple}_{\Sigma_1}[\exists z(\mathsf{Restrict}[\sigma, x, z] \wedge F(x, z, \vec{a}, y)), t, x, y]$$

is a $\Sigma_1(\exists_4(\forall\exists_2\forall \wedge |F| \wedge \forall_3))$ formula. Pick a new variable v', apply

$$\begin{array}{l}\mathsf{Collect}[(\forall t \in \sigma) \\ \quad \mathsf{Tuple}_{\Sigma_1}[\exists z(\mathsf{Restrict}[\sigma, x, z] \wedge F(x, z, \vec{a}, y)), t, x, y] \\ v'],\end{array}$$

and then $\mathsf{ExpandExists}$ to the conjunction with (20).

Case $F \in \Sigma_1$: Clearly, $\exists z(\mathsf{Restrict}[\sigma, x, z] \wedge \exists v F_0(v, x, z, \vec{a}, y))$ is

$$\exists z\exists v(\mathsf{Restrict}[\sigma, x, z] \wedge F_0(v, x, z, \vec{a}, y))$$

by $\mathsf{ExpandExists}$, and

$$\exists v_0\mathsf{Enum}[\mathsf{Restrict}[\sigma, x, z] \wedge F_0(v, x, z, \vec{a}, y), v_0, e]$$

(a $\Sigma_1(\exists_2(\forall\exists_2\forall\wedge|F_0|\wedge\forall))$ formula) by $\mathsf{Collapse}$. Change Tuple by $\mathsf{Tuple}_{\Sigma_1}$, and then pick a new variable v', apply

$$\begin{array}{l}\mathsf{AxColl}[(\forall t \in \sigma) \\ \quad \mathsf{Tuple}_{\ 1}[\\ \quad\quad \exists v_0\mathsf{Enum}[\mathsf{Restrict}[\sigma, x, z] \wedge F_0(v, x, z, \vec{a}, y), v_0, z, v] \\ \quad t, x, y], \\ v'],\end{array}$$

and then $\mathsf{ExpandExists}$ to the conjunction with (20).

Axioms needed: Pairing, and
 Δ_0 case: $\{\exists_4(\forall\exists_2\forall \wedge |F| \wedge \forall_3)\}$-Collection.
 Σ_1 case: $\{\exists_4(\exists_2(\forall\exists_2\forall \wedge |F| \wedge \forall) \wedge \forall_3)\}$-Collection. □

9.2 Compatibility of suitable functions

Theorem 9.4. *Suitable functions are compatible, in the following sense: fix* $\vec{a}$*, let* σ_1 *and* σ_2 *be such that* $S^{\vec{a}}(\sigma_1) \wedge S^{\vec{a}}(\sigma_2)$*, and pick* $x \in \mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_1)$*. Then* $\sigma_1(x) = \sigma_2(x)$*.*

Proof. Abbreviate

$$C_{\vec{a}}(\sigma_1, \sigma_2, x) \overset{\text{def}}{=} S^{\vec{a}}_{\Sigma_1}(\sigma_1) \wedge S^{\vec{a}}_{\Sigma_1}(\sigma_2) \wedge x \in \mathrm{dom}(\sigma_1) \wedge x \in \mathrm{dom}(\sigma_2).$$

We want to prove that for all $\vec{a}$ all x, and all σ_1, σ_2,

$$C_{\vec{a}}(\sigma_1, \sigma_2, x) \to (\sigma_1(x) = \sigma_2(x)). \tag{23}$$

Assume, in search of a contradiction, that there exist σ_1, σ_2, $\vec{a}$, x such that (23) fails, that is,

$$C_{\vec{a}}(\sigma_1, \sigma_2, x) \wedge \sigma_1(x) \neq \sigma_2(x). \tag{24}$$

Since $\sigma_1(x) \neq \sigma_2(x)$ is

$$\mathsf{FunDiff}[\sigma_1, \sigma_2, x],$$

and we can express $C_{\vec{a}}(\sigma_1, \sigma_2, x)$ as

$$\exists s\, \mathsf{Pair}[S^{\vec{a}}_{\Sigma_1}(\sigma) \wedge \mathsf{InDomain}(x, \sigma), \sigma_1, \sigma_2, s, \sigma],$$

(24) is equivalent to

$$\exists s\, \mathsf{Pair}[S^{\vec{a}}_{\Sigma_1}(\sigma) \wedge \mathsf{InDomain}(x, \sigma), \sigma_1, \sigma_2, s, \sigma] \wedge \mathsf{FunDiff}[\sigma_1, \sigma_2, x]. \tag{25}$$

Consider now $\exists w \varphi(w, \sigma_1, \sigma_2, x, \vec{a})$ defined as

$$
\begin{aligned}
&\mathsf{ExpandExists}[\\
&\quad \mathsf{Collapse}[\\
&\qquad \exists s\, \mathsf{Pair}_{\Sigma_1}[\\
&\qquad\quad \mathsf{ExpandExists}[S^{\vec{a}}_{\Sigma_1}(\sigma) \wedge \mathsf{InDomain}(x, \sigma)],\\
&\qquad\quad \sigma_1, \sigma_2, s, \sigma\\
&\qquad],\\
&\qquad 2, w, k\\
&\quad]\\
&\quad \wedge \mathsf{FunDiff}[\sigma_1, \sigma_2, x]\\
&],
\end{aligned}
$$

which is a Σ_1 formula equivalent to (25) with $\varphi \in \Delta_0$. We have assumed that

$$\exists \vec{a} \exists \sigma_1 \exists \sigma_2 \exists x \exists w \varphi(w, \sigma_1, \sigma_2, x, \vec{a}),$$

which is the same as

$$\exists \vec{a} \exists \sigma_1 \exists \sigma_2 \exists w \exists x \varphi(w, \sigma_1, \sigma_2, x, \vec{a}),$$

and therefore we can apply

$$\mathsf{AxFnd}[\exists x \varphi(w, \sigma_1, \sigma_2, x, \vec{a}), y]$$

to get an x such that

$$C_{\vec{a}}(\sigma_1, \sigma_2, x) \wedge (\sigma_1(x) \neq \sigma_2(x)),$$

while for all $y \in x$,

$$C_{\vec{a}}(\sigma_1, \sigma_2, y) \to (\sigma_1(y) = \sigma_2(y))].$$

For $i = 1, 2$, $S^{\vec{a}}_{\Sigma_1}(\sigma_i)$ implies $\text{Tran}(\text{dom}(\sigma_i))$, hence if $x \in \text{dom}(\sigma_i)$ and $y \in x$, $y \in \text{dom}(\sigma_i)$, and therefore for all $y \in x$, $C_{\vec{a}}(\sigma_1, \sigma_2, y)$ and thus

$$(\forall y \in x)(\sigma_1(y) = \sigma_2(y)),$$

or $\sigma_1 \restriction x = \sigma_2 \restriction x$. But now

$$\sigma_1(x) = \mathbf{F}_{\vec{a}}(x, \sigma_1 \restriction x) = \mathbf{F}_{\vec{a}}(x, \sigma_2 \restriction x) = \sigma_2(x),$$

a contradiction.

Axioms used: Since we use $S^{\vec{a}}_{\Sigma_1}$, we must carry all axioms of theorem 9.3. Additionally, Pair_{Σ_1} uses Collection (lemma 7.2). Hence, the additional axioms are:

For the Δ_0 case, instances of
$\{\cdot \wedge \forall\exists_5(\forall\exists_2\forall \wedge |F| \wedge \forall_3) \wedge \forall_3\exists_{12}\forall_9 \wedge \exists\forall\}$-Collection, and
$\{(\exists_2(\cdot \wedge \forall\exists(\cdot \wedge \forall\exists_5(\forall\exists_2\forall \wedge |F| \wedge \forall_3) \wedge \forall_3\exists_{12}\forall_9 \wedge \exists\forall) \wedge \forall) \wedge \exists_8\forall_6)\}$-Foundation

For the Σ_1 case, instances of
$\{\cdot \wedge \forall\exists_5(\exists_2(\forall\exists_2\forall \wedge |F_0| \wedge \forall) \wedge \forall_3) \wedge \forall_3\exists_{12}\forall_9 \wedge \exists\forall\}$-Collection, and
$\{(\exists_2(\cdot\wedge\forall\exists(\cdot\wedge\forall\exists_5(\exists_2(\forall\exists_2\forall\wedge|F_0|\wedge\forall)\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\wedge\forall)\wedge\exists_8\forall_6)\}$-Foundation

□

9.3 Building G

Definition 9.5. *We define $G(x, \vec{a}, y)$ as follows: $G_{\vec{a}}(x, y)$ iff there exists a function σ that is suitable (for the set of parameters $\vec{a}$) and such that $y = \sigma(x)$:*

$$G(x, \vec{a}, y) \overset{def}{=} \exists\sigma(S^{\vec{a}}_{\Sigma_1}(\sigma) \wedge y = \sigma(x)).$$

9.4 G has a Σ_1 equivalence

Theorem 9.6. *If $F \in \Delta_0$ or $F \in \Sigma_1$, then there exists a formula $G_{\Sigma_1}(x, \vec{a}, y)$ such that $G(x, \vec{a}, y) \equiv G_{\Sigma_1}(x, \vec{a}, y) \in \Sigma_1$.*

Proof. $S^{\vec{a}}_{\Sigma_1}(\sigma) \in \Sigma_1$ by theorem 9.3, and $y = \sigma(x)$ is $\mathsf{FunVal}[\sigma, x, y] \in \exists_3\forall_3$. Hence,

$$\mathsf{ExpandExists}[S^{\vec{a}}_{\Sigma_1}(\sigma) \wedge \mathsf{FunVal}[\sigma, x, y]] \in \Sigma_1,$$

and therefore

$$\exists\sigma\mathsf{ExpandExists}[S^{\vec{a}}_{\Sigma_1}(\sigma) \wedge \mathsf{FunVal}[\sigma, x, y]]$$

has two unbounded existential quantifiers, and we can pick a new variable v' so that

$$G_{\Sigma_1}(x, \vec{a}, y) \overset{\text{def}}{=} \mathsf{Collapse}[\exists\sigma(S^{\vec{a}}_{\Sigma_1}(\sigma) \wedge \mathsf{FunVal}[\sigma, x, y]), v', e] \in \Sigma_1.$$

Axioms used: Apart from minor axioms, the use of $S^{\vec{a}}_{\Sigma_1}$ implies the use of the axioms of theorem 9.3. □

9.5 G is a partial function

Theorem 9.7. *Assume that $G(x,\vec{a},y)$ and $G(x,\vec{a},y')$; then, $y=y'$.*

Proof. We are assuming that there exist suitable functions σ and σ' such that $y=\sigma(x)$ and $y'=\sigma'(x)$, but this implies that $y=y'$ since all suitable functions are compatible (lemma 9.4).

Axioms used: The same as those of lemma 9.4. □

Therefore, we can write $y=\mathbf{G}_{\vec{a}}(x)$, if such an y exists.

9.6 G is total

Theorem 9.8. **G** *is total, i.e.,*

$$\forall\vec{a}\forall x\exists y(y=\mathbf{G}_{\vec{a}}(x)).$$

Proof. Assume otherwise, in search of a contradiction. Then there exist $\vec{a}$, x such that

$$\neg\exists y(y=\mathbf{G}_{\vec{a}}(x)).$$

By $\mathsf{AxFnd}[\exists x\neg\exists y(y=\mathbf{G}_{\vec{a}}(x)),x']$, we can choose x such that

$$(\forall x'\in x)\exists y(y=\mathbf{G}_{\vec{a}}(x')),$$

that is, by definition 9.5,

$$(\forall x'\in x)\exists y\exists\sigma(S^{\vec{a}}_{\Sigma_1}(\sigma)\wedge y=\sigma(x'))$$

(since we have proved in lemma 9.6 that $\mathbf{G}\equiv\mathbf{G}_{\Sigma_1}\in\Sigma_1$, a simple application of $\mathsf{Collapse}$ will show that $\exists y(y=\mathbf{G}_{\vec{a}}(x))$ is also Σ_1).

Notice that $\exists y\exists\sigma(S^{\vec{a}}_{\Sigma_1}(\sigma)\wedge y=\sigma(x'))\rightarrow\exists\sigma(S^{\vec{a}}_{\Sigma_1}(\sigma)\wedge x'\in\mathrm{dom}(\sigma))$, hence

$$(\forall x'\in x)\exists\sigma(S^{\vec{a}}_{\Sigma_1}(\sigma)\wedge\mathsf{InDomain}[x',\sigma]).$$

Since $\mathsf{InDomain}[x',\sigma]\in\Delta_0$,

$$\mathsf{ExpandExists}[S^{\vec{a}}_{\Sigma_1}(\sigma)\wedge\mathsf{InDomain}[x',\sigma]]\in\Sigma_1,$$

and we can apply theorem 6.7 to get a set u such that

$$(\forall x'\in x)(\exists\sigma\in u)(S^{\vec{a}}_{\Sigma_1}(\sigma)\wedge\mathsf{InDomain}[x',\sigma])$$

and

$$(\forall\sigma\in u)(\exists x'\in x)(S^{\vec{a}}_{\Sigma_1}(\sigma)\wedge\mathsf{InDomain}[x',\sigma]).$$

We already know (lemma 9.4) that all suitable functions are compatible; hence $\tau_0=\bigcup u$, which exists by Union, is a function. Additionally, $\mathrm{dom}(\tau_0)$ is clearly transitive; hence, $S^{\vec{a}}(\tau_0)$. Let then

$$\tau=\tau_0\cup\{\langle x,\mathbf{F}_{\vec{a}}(x,\tau_0\restriction x)\rangle\},$$

which exists by Pairing and Union (and because we have assumed that **F** is a function). Clearly, $S^{\vec{a}}(\tau)$; but then

$$\tau(x)=\mathbf{F}_{\vec{a}}(x,\varrho\restriction x),$$

and therefore $G_{\vec{a}}(x,\tau(x))$, contrary to our choice of $\vec{a}$ and x.

Axioms used: All previous axioms of this section, plus Foundation and the axioms needed to apply theorem 6.7:

For the Δ_0 case, instances of

$\{\Pi_1(\forall_2(\forall_2(\exists\forall_5(\exists\forall_2\exists\vee\neg(|F|)\vee\exists_3)\vee\exists_3\forall_{12}\exists_9\vee\forall_3\exists_3\vee\exists)\vee\exists))\}$-Foundation,

$\{\exists_2(\forall\exists_5(\forall\exists_2\forall\wedge|F|\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\}$-Separation, and

$\{\exists_2((\forall\exists_5(\forall\exists_2\forall\wedge|F|\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\wedge\forall\}$-Collection.

For the Σ_1 case, instances of

$\{\Pi_1(\forall_2(\forall_2(\exists\forall_5(\forall_2(\exists\forall_2\exists\vee\neg(|F_0|)\vee\exists)\vee\exists_3)\vee\exists_3\forall_{12}\exists_9\vee\forall_3\exists_3\vee\exists)\vee\exists))\}$-Foundation,

$\{\exists_2(\forall\exists_5(\exists_2(\forall\exists_2\forall\wedge|F_0|\wedge\forall)\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\}$-Separation, and

$\{\exists_2((\forall\exists_5(\exists_2(\forall\exists_2\forall\wedge|F_0|\wedge\forall)\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\wedge\forall)\}$-Collection.

□

9.7 Summary of axioms needed for recursion

9.7.1 Δ_0 case

Collection axioms by complexity:

1) $\Delta_0(\exists_4(\forall\exists_2\forall\wedge|F|\wedge\forall_3))$.
2) $\Delta_0(\cdot\wedge\forall\exists_5(\forall\exists_2\forall\wedge|F|\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)$.
3) $\Delta_0(\exists_2((\forall\exists_5(\forall\exists_2\forall\wedge|F|\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\wedge\forall)$.

Separation axiom complexity:

1) $\Delta_0(\exists_2(\forall\exists_5(\forall\exists_2\forall\wedge|F|\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall))$.

Foundation axioms by complexity:

1) $\Delta_0(\exists_2(\cdot\wedge\forall\exists(\cdot\wedge\forall\exists_5(\forall\exists_2\forall\wedge|F|\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\wedge\forall)\wedge\exists_8\forall_6))$.
2) $\Pi_1(\forall_2(\forall_2(\exists\forall_5(\exists\forall_2\exists\vee\neg(|F|)\vee\exists_3)\vee\exists_3\forall_{12}\exists_9\vee\forall_3\exists_3\vee\exists)\vee\exists))$.

9.7.2 Σ_1 case

Collection axioms by complexity:

1) $\Delta_0(\exists_4(\exists_2(\forall\exists_2\forall\wedge|F|\wedge\forall)\wedge\forall_3))$.
2) $\Delta_0(\cdot\wedge\forall\exists_5(\exists_2(\forall\exists_2\forall\wedge|F_0|\wedge\forall)\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)$.
3) $\Delta_0(\exists_2((\forall\exists_5(\exists_2(\forall\exists_2\forall\wedge|F_0|\wedge\forall)\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\wedge\forall))$.

Separation axiom complexity:

1) $\Delta_0(\exists_2(\forall\exists_5(\exists_2(\forall\exists_2\forall\wedge|F_0|\wedge\forall)\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall))$.

Foundation axioms by complexity

1) $\Delta_0((\exists_2(\cdot\wedge\forall\exists(\cdot\wedge\forall\exists_5(\exists_2(\forall\exists_2\forall\wedge|F_0|\wedge\forall)\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall)\wedge\forall)\wedge\exists_8\forall_6))$.
2) $\Pi_1(\forall_2(\forall_2(\exists\forall_5(\forall_2(\exists\forall_2\exists\vee\neg(|F_0|)\vee\exists)\vee\exists_3)\vee\exists_3\forall_{12}\exists_9\vee\forall_3\exists_3\vee\exists)\vee\exists))$.

Part III
Appendix

A An example: the transitive closure

Let x be a set. The **transitive closure** of x, $\mathrm{TrCl}(x)$, is intuitively defined to be

$$x \cup \bigcup x \cup \bigcup\bigcup x ...,$$

that is,

$$\mathrm{TrCl}(x) = x \cup \bigcup_{y\in x} \mathrm{TrCl}(y),$$

since

$$x \cup \bigcup_{y\in x} \mathrm{TrCl}(y) = x \cup \bigcup_{y\in x} \{y \cup \bigcup_{z\in y} \mathrm{TrCl}(z)\} = x \cup \bigcup x \cup \bigcup_{y\in x}\bigcup_{z\in y} \mathrm{TrCl}(z) = ...$$

To implement this concept as a recursive function, we define

$$\mathbf{F}(x,z) = x \cup \bigcup \mathrm{ran}\, z$$

whenever z is a function ($\mathbf{F}$ is undefined otherwise).

The recursion theorem tells us that there exists an unique class function $\mathbf{G}$ such that

$$\mathbf{G}(x) = \mathbf{F}(x, \mathbf{G} \restriction x) = x \cup \bigcup \mathrm{ran}(\mathbf{G} \restriction x) = x \cup \bigcup_{y\in x} \mathbf{G}(y),$$

and by the unicity of $\mathbf{G}$, $\mathbf{G} = \mathrm{TrCl}$.

Now we have to express $\mathbf{F}$ in full. Assuming, as we can, that z is a function, to say that an element e belongs to $\bigcup \mathrm{ran}\, z$ can be expressed as follows:

$$e \in \bigcup \mathrm{ran}\, z \leftrightarrow (\exists w \in \mathrm{ran}\, z)(e \in w);$$

in turn, $\exists w \in \mathrm{ran}\, z$ means that there exists a pair $p \in z$ such that w is its second component, i.e., if $p = \{p_1, p_2\} = \{\{r\}, \{r, w\}\}$, with $p_1 = \{r\}$, then either $p_1 = p_2$, and then $w \in p_w$, or $p_1 \neq p_2$, and then $w \in p_2$ and there is some other element r in p_2:

$$(\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists w \in p_2)(\forall q \in p)((q = p_1 \vee q = p_2) \\ \wedge(p_1 = p_2 \vee (\exists r \in p_2)(r \neq w))).$$

Since p_1 is not empty, we can rewrite the above formula as

$$(\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists w \in p_2)(\forall q \in p)((q = p_1 \vee q = p_2) \\ \wedge(\exists r \in p_2)(p_1 = p_2 \vee r \neq w)),$$

then as

$$(\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists w \in p_2)(\forall q \in p)(\exists r \in p_2) \\ ((q = p_1 \vee q = p_2) \wedge (p_1 = p_2 \vee r \neq w)),$$

and finally as

$$(\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_2)(\exists w \in p_2)(\forall q \in p) \\ ((q = p_1 \vee q = p_2) \wedge (p_1 = p_2 \vee r \neq w)),$$

which is $\exists_5\forall$. Hence, $e \in \bigcup \operatorname{ran} z$ will be

$$(\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_2)(\exists w \in p_2)(\forall q \in p)$$
$$((q = p_1 \vee q = p_2) \wedge (p_1 = p_2 \vee r \neq w) \wedge (e \in w)).$$

Now let $F(x, z, y)$ be $\forall e((e \in y) \leftrightarrow (e \in x \vee e \in \bigcup \operatorname{ran} z))$, that is, the conjunction of $(\forall e \in y)(e \in x \vee e \in \bigcup \operatorname{ran} z)$, which is Δ_0, $(\forall e \in x)(e \in y)$, which is Δ_0, and $(\forall e \in \bigcup \operatorname{ran} z)(e \in y)$, which we should convert first to Δ_0:

$$(\forall p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_1)(\exists w \in p_2)(\exists e \in w)(\forall q \in p)$$
$$((q = p_1 \vee q = p_2) \wedge ((p_1 = p_2 \vee (r \in p_2 \wedge w \neq r)) \rightarrow e \in y)).$$

Therefore $F(x, z, y)$ is

$$\begin{aligned}
&(\forall e \in y)(e \in x \vee \\
&\quad (\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_2)(\exists w \in p_2)(\forall q \in p)\\
&\quad ((q = p_1 \vee q = p_2) \wedge (p_1 = p_2 \vee r \neq w) \wedge (e \in w)))\\
&\wedge(\forall e \in x)(e \in y)\\
&\wedge(\forall p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_1)(\exists w \in p_2)(\exists e \in w)(\forall q \in p)\\
&\quad ((q = p_1 \vee q = p_2) \wedge ((p_1 = p_2 \vee (r \in p_2 \wedge w \neq r)) \rightarrow e \in y)),
\end{aligned} \tag{26}$$

that is, a $\Delta_0(\forall(\cdot \vee \exists_5\forall) \wedge \forall \wedge \forall\exists_5\forall)$ formula.

To prove the Recursion theorem for F (that is, to prove the existence of the transitive closure) we need to prove that F is indeed a function (this is immediate by Extensionality), one instance of each:

$\{\Delta_0(\exists_4(\forall\exists_2\forall \wedge \forall(\cdot \vee \exists_5\forall) \wedge \forall \wedge \forall\exists_5\forall \wedge \forall_3))\}$-Collection,
$\{\Delta_0((\cdot\wedge\forall\exists_5(\forall\exists_2\forall\wedge\forall(\cdot\vee\exists_5\forall)\wedge\forall\wedge\forall\exists_5\forall\wedge\forall_3)\wedge\forall_3\exists_{12}\forall_9\wedge\exists\forall))\}$-Collection, and
$\{\Delta_0(\exists_2((\forall\exists_5(\forall\exists_2\forall \wedge \forall(\cdot \vee \exists_5\forall) \wedge \forall \wedge \forall\exists_5\forall \wedge \forall_3) \wedge \forall_3\exists_{12}\forall_9 \wedge \exists\forall) \wedge \forall))\}$-Collection;

one instance of

$\{\Delta_0(\exists_2(\forall\exists_5(\forall\exists_2\forall \wedge \forall(\cdot \vee \exists_5\forall) \wedge \forall \wedge \forall\exists_5\forall \wedge \forall_3) \wedge \forall_3\exists_{12}\forall_9 \wedge \exists\forall))\}$-Separation, and

one instance of each:

$\{\Delta_0((\exists_2(\cdot \wedge \forall\exists(\cdot \wedge \forall\exists_5(\forall\exists_2\forall \wedge \forall(\cdot \vee \exists_5\forall) \wedge \forall \wedge \forall\exists_5\forall \wedge \forall_3) \wedge \forall_3\exists_{12}\forall_9 \wedge \exists\forall) \wedge \forall) \wedge \exists_8\forall_6))\}$-Foundation, and
$\{\Pi_1(\forall_2(\forall_2(\exists\forall_5(\exists\forall_2\exists \vee \exists(\cdot \wedge \forall_5\exists) \vee \exists \vee \exists\forall_5\exists \vee \exists_3) \vee \exists_3\forall_{12}\exists_9 \vee \forall_3\exists_3 \vee \exists) \vee \exists))\}$-Foundation.

B A curiosity: The Π_1-foundation axiom for the transitive closure case

As a curiosity, and as a means to prove our assertion that undoing defined notions is almost impossible in practice, we list here one single axiom for the transitive closure case. Recall formula (26): $F(x,z,y)$ is

$$
\begin{aligned}
&(\forall e \in y)(e \in x \vee \\
&\quad (\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_2)(\exists w \in p_2)(\forall q \in p) \\
&\quad ((q = p_1 \vee q = p_2) \wedge (p_1 = p_2 \vee r \neq w) \wedge (e \in w))) \\
&\wedge (\forall e \in x)(e \in y) \\
&\wedge (\forall p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_1)(\exists w \in p_2)(\exists e \in w)(\forall q \in p) \\
&\quad ((q = p_1 \vee q = p_2) \wedge ((p_1 = p_2 \vee (r \in p_2 \wedge w \neq r)) \rightarrow e \in y)).
\end{aligned}
$$

We first must build $S(\sigma)$, the suitability formula, defined as

$$
\begin{aligned}
&(\forall t \in \sigma)\mathsf{Tuple}[\exists z(\mathsf{Restrict}[\sigma, x, z] \wedge F(x,z,y)), t, x, y] \\
&\wedge \mathsf{Tuples}[(x_1 = x_2 \rightarrow y_1 = y_2) \wedge (x_1 \in x_2 \wedge x_2 \in x_3 \rightarrow x_1 \in x_3), 3, \sigma, x, y].
\end{aligned}
$$

Now $\mathsf{Restrict}[\sigma, x, z]$ is

$$(\forall p \in \sigma)(\exists e \in p)(\exists p_1 \in e)(\forall e_1 \in p)(p_1 \in e_1 \wedge (p \in z \leftrightarrow p_1 \in x)),$$

so that

$$\mathsf{Tuple}[\exists z(\mathsf{Restrict}[\sigma, x, z] \wedge F(x,z,y)), t, x, y]$$

is $(\exists p_1 \in t)(\exists p_2 \in t)(\exists x \in p_1)(\exists y \in p_2)(\exists z((\forall p \in \sigma)(\exists e \in p)(\exists p_1 \in e)(\forall e_1 \in p)(p_1 \in e_1 \wedge (p \in z \leftrightarrow p_1 \in x)) \wedge (\forall e \in y)(e \in x \vee (\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_2)(\exists w \in p_2)(\forall q \in p)((q = p_1 \vee q = p_2) \wedge (p_1 = p_2 \vee r \neq w) \wedge e \in w)) \wedge (\forall e \in x)(e \in y) \wedge (\forall p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_1)(\exists w \in p_2)(\exists e \in w)(\forall q \in p)((q = p_1 \vee q = p_2) \wedge ((p_1 = p_2 \vee (r \in p_2 \wedge w \neq r)) \rightarrow e \in y))) \wedge (\forall e \in t)(\forall e_1 \in p_1)(\forall e_2 \in p_2)((e = p_1 \vee e = p_2) \wedge e_1 = x \wedge (e_2 = x \vee e_2 = y) \wedge x \in p_2))$. Therefore

$$S(\sigma)$$

is $(\forall t \in \sigma)(\exists p_1 \in t)(\exists p_2 \in t)(\exists x \in p_1)(\exists y \in p_2)(\exists z((\forall p \in \sigma)(\exists e \in p)(\exists p_1 \in e)(\forall e_1 \in p)(p_1 \in e_1 \wedge (p \in z \leftrightarrow p_1 \in x)) \wedge ((\forall e \in y)(e \in x \vee (\exists p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_2)(\exists w \in p_2)(\forall q \in p)((q = p_1 \vee q = p_2) \wedge (p_1 = p_2 \vee r \neq w) \wedge e \in w)) \wedge (\forall e \in x)(e \in y) \wedge (\forall p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_1)(\exists w \in p_2)(\exists e \in w)(\forall q \in p)((q = p_1 \vee q = p_2) \wedge ((p_1 = p_2 \vee (r \in p_2 \wedge w \neq r)) \rightarrow e \in y)))) \wedge (\forall e \in t)(\forall e_1 \in p_1)(\forall e_2 \in p_2)((e = p_1 \vee e = p_2) \wedge e_1 = x \wedge (e_2 = x \vee e_2 = y) \wedge x \in p_2)) \wedge (\forall p^1 \in \sigma)(\forall p^2 \in \sigma)(\forall p^3 \in \sigma)(\exists p_1^1 \in p^1)(\exists p_2^1 \in p^1)(\exists p_1^2 \in p^2)(\exists p_2^2 \in p^2)(\exists p_1^3 \in p^3)(\exists p_2^3 \in p^3)(\exists x_1 \in p_1^1)(\exists y_1 \in p_2^1)(\exists x_2 \in p_1^2)(\exists y_2 \in p_2^2)(\exists x_3 \in p_1^3)(\exists y_3 \in p_2^3)(\forall e^1 \in p^1)(\forall e^2 \in p^2)(\forall e^3 \in p^3)(\forall e_1^1 \in p_1^1)(\forall e_2^1 \in p_2^1)(\forall e_1^2 \in p_1^2)(\forall e_2^2 \in p_2^2)(\forall e_1^3 \in p_1^3)(\forall e_2^3 \in p_2^3)(x_1 \in p_2^1 \wedge x_2 \in p_2^2 \wedge x_3 \in p_2^3 \wedge (e^1 = p_1^1 \vee e^1 = p_2^1) \wedge (e^2 = p_1^2 \vee e^2 = p_2^2) \wedge (e^3 = p_1^3 \vee e^3 = p_2^3) \wedge e_1^1 = x_1 \wedge e_1^2 = x_2 \wedge e_1^3 = x_3 \wedge (e_2^1 = x_1 \vee e_2^1 = y_1) \wedge (e_2^2 = x_2 \vee e_2^2 = y_2) \wedge (e_2^3 = x_3 \vee e_2^3 = y_3) \wedge (x_1 = x_2 \rightarrow y_1 = y_2) \wedge (x_1 \in x_2 \wedge x_2 \in x_3 \rightarrow x_1 \in x_3))$.

Remember that $G(x,y)$ is defined by $\exists\sigma(S(\sigma) \wedge y = \sigma(x))$. But first we have to bring $S(\sigma)$ into a Σ_1 form, i.e.,

$$
\begin{aligned}
&\mathsf{ExpandExists}[\\
&\quad \mathsf{Collect}[\\
&\qquad (\forall t \in \sigma)\mathsf{Tuple}_{\Sigma_1}[\exists z(\mathsf{Restrict}[\sigma, x, z] \wedge F(x,z,y)), t, x, y], \\
&\quad a] \wedge \\
&\quad \mathsf{Tuples}[(x_1 = x_2 \rightarrow y_1 = y_2) \wedge (x_1 \in x_2 \wedge x_2 \in x_3 \rightarrow x_1 \in x_3), 3, \sigma, x, y], \\
&1],
\end{aligned}
$$

that is, $\exists a((\forall t\in\sigma)(\exists z\in a)(\exists p_1\in t)(\exists p_2\in t)(\exists x\in p_1)(\exists y\in p_2)((\forall p\in\sigma)(\exists e\in p)(\exists p_1\in e)(\forall e_1\in p)(p_1\in e_1\wedge(p\in z\leftrightarrow p_1\in x))\wedge(\forall e\in y)(e\in x\vee(\exists p\in z)(\exists p_1\in p)(\exists p_2\in p)(\exists r\in p_2)(\exists w\in p_2)(\forall q\in p)((q=p_1\vee q=p_2)\wedge(p_1=p_2\vee r\neq w)\wedge e\in w))\wedge(\forall e\in x)(e\in y)\wedge(\forall p\in z)(\exists p_1\in p)(\exists p_2\in p)(\exists r\in p_1)(\exists w\in p_2)(\exists e\in w)(\forall q\in p)((q=p_1\vee q=p_2)\wedge((p_1=p_2\vee(r\in p_2\wedge w\neq r))\to e\in y))\wedge(\forall e\in t)(\forall e_1\in p_1)(\forall e_2\in p_2)((e=p_1\vee e=p_2)\wedge e_1=x\wedge(e_2=x\vee e_2=y)))\wedge(\forall p^1\in\sigma)(\forall p^2\in\sigma)(\forall p^3\in\sigma)(\exists p^1_1\in p^1)(\exists p^1_2\in p^1)(\exists p^2_1\in p^2)(\exists p^2_2\in p^2)(\exists p^3_1\in p^3)(\exists p^3_2\in p^3)(\exists x_1\in p^1_1)(\exists y_1\in p^1_2)(\exists x_2\in p^2_1)(\exists y_2\in p^2_2)(\exists x_3\in p^3_1)(\exists y_3\in p^3_2)(\forall e^1\in p^1)(\forall e^2\in p^2)(\forall e^3\in p^3)(\forall e^1_1\in p^1_1)(\forall e^1_2\in p^1_2)(\forall e^2_1\in p^2_1)(\forall e^2_2\in p^2_2)(\forall e^3_1\in p^3_1)(\forall e^3_2\in p^3_2)(x_1\in p^1_2\wedge x_2\in p^2_2\wedge x_3\in p^3_2\wedge(e^1=p^1_1\vee e^1=p^1_2)\wedge(e^2=p^2_1\vee e^2=p^2_2)\wedge(e^3=p^3_1\vee e^3=p^3_2)\wedge e^1_1=x_1\wedge e^2_1=x_2\wedge e^3_1=x_3\wedge(e^1_2=x_1\vee e^1_2=y_1)\wedge(e^2_2=x_2\vee e^2_2=y_2)\wedge(e^3_2=x_3\vee e^3_2=y_3)\wedge(x_1=x_2\to y_1=y_2)\wedge(x_1\in x_2\wedge x_2\in x_3\to x_1\in x_3)))$.

Now $G(x,y)$ is $\exists\sigma(\exists a((\forall t\in\sigma)(\exists z\in a)(\exists p_1\in t)(\exists p_2\in t)(\exists x\in p_1)(\exists y\in p_2)((\forall p\in\sigma)(\exists e\in p)(\exists p_1\in e)(\forall e_1\in p)(p_1\in e_1\wedge(p\in z\leftrightarrow p_1\in x))\wedge(\forall e\in y)(e\in x\vee(\exists p\in z)(\exists p_1\in p)(\exists p_2\in p)(\exists r\in p_2)(\exists w\in p_2)(\forall q\in p)((q=p_1\vee q=p_2)\wedge(p_1=p_2\vee r\neq w)\wedge e\in w))\wedge(\forall e\in x)(e\in y)\wedge(\forall p\in z)(\exists p_1\in p)(\exists p_2\in p)(\exists r\in p_1)(\exists w\in p_2)(\exists e\in w)(\forall q\in p)((q=p_1\vee q=p_2)\wedge((p_1=p_2\vee(r\in p_2\wedge w\neq r))\to e\in y))\wedge(\forall e\in t)(\forall e_1\in p_1)(\forall e_2\in p_2)((e=p_1\vee e=p_2)\wedge e_1=x\wedge(e_2=x\vee e_2=y)))\wedge(\forall p^1\in\sigma)(\forall p^2\in\sigma)(\forall p^3\in\sigma)(\exists p^1_1\in p^1)(\exists p^1_2\in p^1)(\exists p^2_1\in p^2)(\exists p^2_2\in p^2)(\exists p^3_1\in p^3)(\exists p^3_2\in p^3)(\exists x_1\in p^1_1)(\exists y_1\in p^1_2)(\exists x_2\in p^2_1)(\exists y_2\in p^2_2)(\exists x_3\in p^3_1)(\exists y_3\in p^3_2)(\forall e^1\in p^1)(\forall e^2\in p^2)(\forall e^3\in p^3)(\forall e^1_1\in p^1_1)(\forall e^1_2\in p^1_2)(\forall e^2_1\in p^2_1)(\forall e^2_2\in p^2_2)(\forall e^3_1\in p^3_1)(\forall e^3_2\in p^3_2)(x_1\in p^1_2\wedge x_2\in p^2_2\wedge x_3\in p^3_2\wedge(e^1=p^1_1\vee e^1=p^1_2)\wedge(e^2=p^2_1\vee e^2=p^2_2)\wedge(e^3=p^3_1\vee e^3=p^3_2)\wedge e^1_1=x_1\wedge e^2_1=x_2\wedge e^3_1=x_3\wedge(e^1_2=x_1\vee e^1_2=y_1)\wedge(e^2_2=x_2\vee e^2_2=y_2)\wedge(e^3_2=x_3\vee e^3_2=y_3)\wedge(x_1=x_2\to y_1=y_2)\wedge(x_1\in x_2\wedge x_2\in x_3\to x_1\in x_3)))\wedge(\exists p\in\sigma)(\exists p_1\in p)(\exists p_2\in p)(\forall e\in p)(\forall e_1\in p_1)(\forall e_2\in p_2)((e=p_1\vee e=p_2)\wedge e_1=x\wedge(e_2=x\vee e_2=y)))$, and we need to apply

$$\mathsf{AxFnd}[\exists x\neg\exists y G(x,y)].$$

But first we must bring $G(x,y)$ into a Σ_1 form: $\exists b(\exists\sigma\in b)(\exists a\in b)((\forall t\in\sigma)(\exists z\in a)(\exists p_1\in t)(\exists p_2\in t)(\exists x\in p_1)(\exists y\in p_2)((\forall p\in\sigma)(\exists e\in p)(\exists p_1\in e)(\forall e_1\in p)(p_1\in e_1\wedge(p\in z\leftrightarrow p_1\in x))\wedge(\forall e\in y)(e\in x\vee(\exists p\in z)(\exists p_1\in p)(\exists p_2\in p)(\exists r\in p_2)(\exists w\in p_2)(\forall q\in p)((q=p_1\vee q=p_2)\wedge(p_1=p_2\vee r\neq w)\wedge e\in w))\wedge(\forall e\in x)(e\in y)\wedge(\forall p\in z)(\exists p_1\in p)(\exists p_2\in p)(\exists r\in p_1)(\exists w\in p_2)(\exists e\in w)(\forall q\in p)((q=p_1\vee q=p_2)\wedge((p_1=p_2\vee(r\in p_2\wedge w\neq r))\to e\in y))\wedge(\forall e\in t)(\forall e_1\in p_1)(\forall e_2\in p_2)((e=p_1\vee e=p_2)\wedge e_1=x\wedge(e_2=x\vee e_2=y)))\wedge(\forall p^1\in\sigma)(\forall p^2\in\sigma)(\forall p^3\in\sigma)(\exists p^1_1\in p^1)(\exists p^1_2\in p^1)(\exists p^2_1\in p^2)(\exists p^2_2\in p^2)(\exists p^3_1\in p^3)(\exists p^3_2\in p^3)(\exists x_1\in p^1_1)(\exists y_1\in p^1_2)(\exists x_2\in p^2_1)(\exists y_2\in p^2_2)(\exists x_3\in p^3_1)(\exists y_3\in p^3_2)(\forall e^1\in p^1)(\forall e^2\in p^2)(\forall e^3\in p^3)(\forall e^1_1\in p^1_1)(\forall e^1_2\in p^1_2)(\forall e^2_1\in p^2_1)(\forall e^2_2\in p^2_2)(\forall e^3_1\in p^3_1)(\forall e^3_2\in p^3_2)(x_1\in p^1_2\wedge x_2\in p^2_2\wedge x_3\in p^3_2\wedge(e^1=p^1_1\vee e^1=p^1_2)\wedge(e^2=p^2_1\vee e^2=p^2_2)\wedge(e^3=p^3_1\vee e^3=p^3_2)\wedge e^1_1=x_1\wedge e^2_1=x_2\wedge e^3_1=x_3\wedge(e^1_2=x_1\vee e^1_2=y_1)\wedge(e^2_2=x_2\vee e^2_2=y_2)\wedge(e^3_2=x_3\vee e^3_2=y_3)\wedge(x_1=x_2\to y_1=y_2)\wedge(x_1\in x_2\wedge x_2\in x_3\to x_1\in x_3))\wedge(\exists p\in\sigma)(\exists p_1\in p)(\exists p_2\in p)(\forall e\in p)(\forall e_1\in p_1)(\forall e_2\in p_2)((e=p_1\vee e=p_2)\wedge e_1=x\wedge(e_2=x\vee e_2=y))\wedge(\forall c\in b)(c=\sigma\vee c=a))$, then $\mathsf{Collapse}$

$$\exists y G_{\Sigma_1}(x,y)$$

to get another Σ_1 form, namely $\exists d(\exists y\in d)(\exists b\in d)((\exists\sigma\in b)(\exists a\in b)((\forall t\in\sigma)(\exists z\in a)(\exists p_1\in t)(\exists p_2\in t)(\exists x\in p_1)(\exists y\in p_2)((\forall p\in\sigma)(\exists e\in p)(\exists p_1\in e)(\forall e_1\in p)(p_1\in e_1\wedge(p\in z\leftrightarrow p_1\in x))\wedge(\forall e\in y)(e\in x\vee(\exists p\in z)(\exists p_1\in$

$p)(\exists p_2 \in p)(\exists r \in p_2)(\exists w \in p_2)(\forall q \in p)((q = p_1 \vee q = p_2) \wedge (p_1 = p_2 \vee r \neq w) \wedge e \in w)) \wedge (\forall e \in x)(e \in y) \wedge (\forall p \in z)(\exists p_1 \in p)(\exists p_2 \in p)(\exists r \in p_1)(\exists w \in p_2)(\exists e \in w)(\forall q \in p)((q = p_1 \vee q = p_2) \wedge ((p_1 = p_2 \vee (r \in p_2 \wedge w \neq r)) \to e \in y)) \wedge (\forall e \in t)(\forall e_1 \in p_1)(\forall e_2 \in p_2)((e = p_1 \vee e = p_2) \wedge e_1 = x \wedge (e_2 = x \vee e_2 = y))) \wedge (\forall p^1 \in \sigma)(\forall p^2 \in \sigma)(\forall p^3 \in \sigma)(\exists p_1^1 \in p^1)(\exists p_2^1 \in p^1)(\exists p_1^2 \in p^2)(\exists p_2^2 \in p^2)(\exists p_1^3 \in p^3)(\exists p_2^3 \in p^3)(\exists x_1 \in p_1^1)(\exists y_1 \in p_2^1)(\exists x_2 \in p_1^2)(\exists y_2 \in p_2^2)(\exists x_3 \in p_1^3)(\exists y_3 \in p_2^3)(\forall e^1 \in p^1)(\forall e^2 \in p^2)(\forall e^3 \in p^3)(\forall e_1^1 \in p_1^1)(\forall e_2^1 \in p_2^1)(\forall e_1^2 \in p_1^2)(\forall e_2^2 \in p_2^2)(\forall e_1^3 \in p_1^3)(\forall e_2^3 \in p_2^3)(x_1 \in p_2^1 \wedge x_2 \in p_2^2 \wedge x_3 \in p_2^3 \wedge (e^1 = p_1^1 \vee e^1 = p_2^1) \wedge (e^2 = p_1^2 \vee e^2 = p_2^2) \wedge (e^3 = p_1^3 \vee e^3 = p_2^3) \wedge e_1^1 = x_1 \wedge e_1^2 = x_2 \wedge e_1^3 = x_3 \wedge (e_2^1 = x_1 \vee e_2^1 = y_1) \wedge (e_2^2 = x_2 \vee e_2^2 = y_2) \wedge (e_2^3 = x_3 \vee e_2^3 = y_3) \wedge (x_1 = x_2 \to y_1 = y_2) \wedge (x_1 \in x_2 \wedge x_2 \in x_3 \to x_1 \in x_3)) \wedge (\exists p \in \sigma)(\exists p_1 \in p)(\exists p_2 \in p)(\forall e \in p)(\forall e_1 \in p_1)(\forall e_2 \in p_2)((e = p_1 \vee e = p_2) \wedge e_1 = x \wedge (e_2 = x \vee e_2 = y)) \wedge (\forall c \in b)(c = \sigma \vee c = a)) \wedge (\forall f \in d)(f = y \vee f = b)),$ and finally

$$\mathsf{Negate}[\exists y G_{\Sigma_1}(x, y)]$$

to get a Π_1 formula: $\forall d(\forall y \in d)(\forall b \in d)((\forall \sigma \in b)(\forall a \in b)((\exists t \in \sigma)(\forall z \in a)(\forall p_1 \in t)(\forall p_2 \in t)(\forall x \in p_1)(\forall y \in p_2)((\exists p \in \sigma)(\forall e \in p)(\forall p_1 \in e)(\exists e_1 \in p)(p_1 \notin e_1 \vee (p \in z \wedge p_1 \notin x) \vee (p \notin z \wedge p_1 \in x)) \vee (\exists e \in y)(e \notin x \wedge (\forall p \in z)(\forall p_1 \in p)(\forall p_2 \in p)(\forall r \in p_2)(\forall w \in p_2)(\exists q \in p)((q \neq p_1 \wedge q \neq p_2) \vee (p_1 \neq p_2 \wedge r = w) \vee e \notin w)) \vee (\exists e \in x)(e \notin y) \vee (\exists p \in z)(\forall p_1 \in p)(\forall p_2 \in p)(\forall r \in p_1)(\forall w \in p_2)(\forall e \in w)(\exists q \in p)((q \neq p_1 \wedge q \neq p_2) \vee ((p_1 = p_2 \vee (r \in p_2 \wedge w \neq r)) \wedge e \notin y)) \vee (\exists e \in t)(\exists e_1 \in p_1)(\exists e_2 \in p_2)((e \neq p_1 \wedge e \neq p_2) \vee e_1 \neq x \vee (e_2 \neq x \wedge e_2 \neq y))) \vee (\exists p^1 \in \sigma)(\exists p^2 \in \sigma)(\exists p^3 \in \sigma)(\forall p_1^1 \in p^1)(\forall p_2^1 \in p^1)(\forall p_1^2 \in p^2)(\forall p_2^2 \in p^2)(\forall p_1^3 \in p^3)(\forall p_2^3 \in p^3)(\forall x_1 \in p_1^1)(\forall y_1 \in p_2^1)(\forall x_2 \in p_1^2)(\forall y_2 \in p_2^2)(\forall x_3 \in p_1^3)(\forall y_3 \in p_2^3)(\exists e^1 \in p^1)(\exists e^2 \in p^2)(\exists e^3 \in p^3)(\exists e_1^1 \in p_1^1)(\exists e_2^1 \in p_2^1)(\exists e_1^2 \in p_1^2)(\exists e_2^2 \in p_2^2)(\exists e_1^3 \in p_1^3)(\exists e_2^3 \in p_2^3)(x_1 \notin p_2^1 \vee x_2 \notin p_2^2 \vee x_3 \notin p_2^3 \vee (e^1 \neq p_1^1 \wedge e^1 \neq p_2^1) \vee (e^2 \neq p_1^2 \wedge e^2 \neq p_2^2) \vee (e^3 \neq p_1^3 \wedge e^3 \neq p_2^3) \vee e_1^1 \neq x_1 \vee e_1^2 \neq x_2 \vee e_1^3 \neq x_3 \vee (e_2^1 \neq x_1 \wedge e_2^1 \neq y_1) \vee (e_2^2 \neq x_2 \wedge e_2^2 \neq y_2) \vee (e_2^3 \neq x_3 \wedge e_2^3 \neq y_3) \vee (x_1 = x_2 \wedge y_1 \neq y_2) \vee ((x_1 \in x_2 \wedge x_2 \in x_3) \wedge x_1 \notin x_3)) \vee (\forall p \in \sigma)(\forall p_1 \in p)(\forall p_2 \in p)(\exists e \in p)(\exists e_1 \in p_1)(\exists e_2 \in p_2)((e \neq p_1 \wedge e \neq p_2) \vee e_1 \neq x \vee (e_2 \neq x \wedge e_2 \neq y)) \vee (\exists c \in b)(c \neq \sigma \wedge c \neq a)) \vee (\exists f \in d)(f \neq y \wedge f \neq b)).$

Finally,

$$\mathsf{AxFnd}[\exists x\mathsf{Negate}[\exists y G_{\Sigma_1}(x,y)],k]$$

is $\exists x\forall d(\forall y\in d)(\forall b\in d)((\forall\sigma\in b)(\forall a\in b)((\exists t\in\sigma)(\forall z\in a)(\forall p_1\in$
$t)(\forall p_2\in t)(\forall x\in p_1)(\forall y\in p_2)((\exists p\in\sigma)(\forall e\in p)(\forall p_1\in e)(\exists e_1\in p)(p_1\notin$
$e_1\vee(p\in z\wedge p_1\notin x)\vee(p\notin z\wedge p_1\in x))\vee(\exists e\in y)(e\notin x\wedge(\forall p\in z)(\forall p_1\in$
$p)(\forall p_2\in p)(\forall r\in p_2)(\forall w\in p_2)(\exists q\in p)((q\neq p_1\wedge q\neq p_2)\vee(p_1\neq p_2\wedge r=$
$w)\vee e\notin w))\vee(\exists e\in x)(e\notin y)\vee(\exists p\in z)(\forall p_1\in p)(\forall p_2\in p)(\forall r\in p_1)(\forall w\in$
$p_2)(\forall e\in w)(\exists q\in p)((q\neq p_1\wedge q\neq p_2)\vee((p_1=p_2\vee(r\in p_2\wedge w\neq r))\wedge e\notin$
$y))\vee(\exists e\in t)(\exists e_1\in p_1)(\exists e_2\in p_2)((e\neq p_1\wedge e\neq p_2)\vee e_1\neq x\vee(e_2\neq$
$x\wedge e_2\neq y)))\vee(\exists p^1\in\sigma)(\exists p^2\in\sigma)(\exists p^3\in\sigma)(\forall p^1_1\in p^1)(\forall p^1_2\in p^1)(\forall p^2_1\in$
$p^2)(\forall p^2_2\in p^2)(\forall p^3_1\in p^3)(\forall p^3_2\in p^3)(\forall x_1\in p^1_1)(\forall y_1\in p^1_2)(\forall x_2\in p^2_1)(\forall y_2\in$
$p^2_2)(\forall x_3\in p^3_1)(\forall y_3\in p^3_2)(\exists e^1\in p^1)(\exists e^2\in p^2)(\exists e^3\in p^3)(\exists e^1_1\in p^1_1)(\exists e^1_2\in$
$p^1_2)(\exists e^2_1\in p^2_1)(\exists e^2_2\in p^2_2)(\exists e^3_1\in p^3_1)(\exists e^3_2\in p^3_2)(x_1\notin p^1_2\vee x_2\notin p^2_2\vee x_3\notin$
$p^3_2\vee(e^1\neq p^1_1\wedge e^1\neq p^1_2)\vee(e^2\neq p^2_1\wedge e^2\neq p^2_2)\vee(e^3\neq p^3_1\wedge e^3\neq p^3_2)\vee e^1_1\neq$
$x_1\vee e^2_1\neq x_2\vee e^3_1\neq x_3\vee(e^1_2\neq x_1\wedge e^1_2\neq y_1)\vee(e^2_2\neq x_2\wedge e^2_2\neq y_2)\vee(e^3_2\neq$
$x_3\wedge e^3_2\neq y_3)\vee(x_1=x_2\wedge y_1\neq y_2)\vee((x_1\in x_2\wedge x_2\in x_3)\wedge x_1\notin x_3))\vee(\forall p\in$
$\sigma)(\forall p_1\in p)(\forall p_2\in p)(\exists e\in p)(\exists e_1\in p_1)(\exists e_2\in p_2)((e\neq p_1\wedge e\neq p_2)\vee e_1\neq$
$x\vee(e_2\neq x\wedge e_2\neq y))\vee(\exists c\in b)(c\neq\sigma\wedge c\neq a))\vee(\exists f\in d)(f\neq y\wedge f\neq$
$b))\rightarrow\exists x(\forall d(\forall y\in d)(\forall b\in d)((\forall\sigma\in b)(\forall a\in b)((\exists t\in\sigma)(\forall z\in a)(\forall p_1\in$
$t)(\forall p_2\in t)(\forall x\in p_1)(\forall y\in p_2)((\exists p\in\sigma)(\forall e\in p)(\forall p_1\in e)(\exists e_1\in p)(p_1\notin$
$e_1\vee(p\in z\wedge p_1\notin x)\vee(p\notin z\wedge p_1\in x))\vee(\exists e\in y)(e\notin x\wedge(\forall p\in z)(\forall p_1\in$
$p)(\forall p_2\in p)(\forall r\in p_2)(\forall w\in p_2)(\exists q\in p)((q\neq p_1\wedge q\neq p_2)\vee(p_1\neq p_2\wedge r=$
$w)\vee e\notin w))\vee(\exists e\in x)(e\notin y)\vee(\exists p\in z)(\forall p_1\in p)(\forall p_2\in p)(\forall r\in p_1)(\forall w\in$
$p_2)(\forall e\in w)(\exists q\in p)((q\neq p_1\wedge q\neq p_2)\vee((p_1=p_2\vee(r\in p_2\wedge w\neq r))\wedge e\notin$
$y))\vee(\exists e\in t)(\exists e_1\in p_1)(\exists e_2\in p_2)((e\neq p_1\wedge e\neq p_2)\vee e_1\neq x\vee(e_2\neq$
$x\wedge e_2\neq y)))\vee(\exists p^1\in\sigma)(\exists p^2\in\sigma)(\exists p^3\in\sigma)(\forall p^1_1\in p^1)(\forall p^1_2\in p^1)(\forall p^2_1\in$
$p^2)(\forall p^2_2\in p^2)(\forall p^3_1\in p^3)(\forall p^3_2\in p^3)(\forall x_1\in p^1_1)(\forall y_1\in p^1_2)(\forall x_2\in p^2_1)(\forall y_2\in$
$p^2_2)(\forall x_3\in p^3_1)(\forall y_3\in p^3_2)(\exists e^1\in p^1)(\exists e^2\in p^2)(\exists e^3\in p^3)(\exists e^1_1\in p^1_1)(\exists e^1_2\in$
$p^1_2)(\exists e^2_1\in p^2_1)(\exists e^2_2\in p^2_2)(\exists e^3_1\in p^3_1)(\exists e^3_2\in p^3_2)(x_1\notin p^1_2\vee x_2\notin p^2_2\vee x_3\notin$
$p^3_2\vee(e^1\neq p^1_1\wedge e^1\neq p^1_2)\vee(e^2\neq p^2_1\wedge e^2\neq p^2_2)\vee(e^3\neq p^3_1\wedge e^3\neq p^3_2)\vee e^1_1\neq$
$x_1\vee e^2_1\neq x_2\vee e^3_1\neq x_3\vee(e^1_2\neq x_1\wedge e^1_2\neq y_1)\vee(e^2_2\neq x_2\wedge e^2_2\neq y_2)\vee(e^3_2\neq$
$x_3\wedge e^3_2\neq y_3)\vee(x_1=x_2\wedge y_1\neq y_2)\vee((x_1\in x_2\wedge x_2\in x_3)\wedge x_1\notin x_3))\vee(\forall p\in$
$\sigma)(\forall p_1\in p)(\forall p_2\in p)(\exists e\in p)(\exists e_1\in p_1)(\exists e_2\in p_2)((e\neq p_1\wedge e\neq p_2)\vee e_1\neq$
$x\vee(e_2\neq x\wedge e_2\neq y))\vee(\exists c\in b)(c\neq\sigma\wedge c\neq a))\vee(\exists f\in d)(f\neq y\wedge f\neq$
$b))\wedge(\forall k\in x)\exists d(\exists y\in d)(\exists b\in d)((\exists\sigma\in b)(\exists a\in b)((\forall t\in\sigma)(\exists z\in a)(\exists p_1\in$
$t)(\exists p_2\in t)(\exists x\in p_1)(\exists y\in p_2)((\forall p\in\sigma)(\exists e\in p)(\exists p_1\in e)(\forall e_1\in p)(p_1\in$
$e_1\wedge((p\notin z\vee p_1\in x)\wedge(p\in z\vee p_1\notin x)))\wedge(\forall e\in y)(e\in x\vee(\exists p\in z)(\exists p_1\in$
$p)(\exists p_2\in p)(\exists r\in p_2)(\exists w\in p_2)(\forall q\in p)((q=p_1\vee q=p_2)\wedge(p_1=p_2\vee r\neq$
$w)\wedge e\in w))\wedge(\forall e\in x)(e\in y)\wedge(\forall p\in z)(\exists p_1\in p)(\exists p_2\in p)(\exists r\in p_1)(\exists w\in$
$p_2)(\exists e\in w)(\forall q\in p)((q=p_1\vee q=p_2)\wedge((p_1\neq p_2\wedge(r\notin p_2\vee w=r))\vee e\in$
$y))\wedge(\forall e\in t)(\forall e_1\in p_1)(\forall e_2\in p_2)((e=p_1\vee e=p_2)\wedge e_1=x\wedge(e_2=$
$x\vee e_2=y)))\wedge(\forall p^1\in\sigma)(\forall p^2\in\sigma)(\forall p^3\in\sigma)(\exists p^1_1\in p^1)(\exists p^1_2\in p^1)(\exists p^2_1\in$
$p^2)(\exists p^2_2\in p^2)(\exists p^3_1\in p^3)(\exists p^3_2\in p^3)(\exists x_1\in p^1_1)(\exists y_1\in p^1_2)(\exists x_2\in p^2_1)(\exists y_2\in$
$p^2_2)(\exists x_3\in p^3_1)(\exists y_3\in p^3_2)(\forall e^1\in p^1)(\forall e^2\in p^2)(\forall e^3\in p^3)(\forall e^1_1\in p^1_1)(\forall e^1_2\in$
$p^1_2)(\forall e^2_1\in p^2_1)(\forall e^2_2\in p^2_2)(\forall e^3_1\in p^3_1)(\forall e^3_2\in p^3_2)(x_1\in p^1_2\wedge x_2\in p^2_2\wedge x_3\in$
$p^3_2\wedge(e^1=p^1_1\vee e^1=p^1_2)\wedge(e^2=p^2_1\vee e^2=p^2_2)\wedge(e^3=p^3_1\vee e^3=p^3_2)\wedge e^1_1=$
$x_1\wedge e^2_1=x_2\wedge e^3_1=x_3\wedge(e^1_2=x_1\vee e^1_2=y_1)\wedge(e^2_2=x_2\vee e^2_2=y_2)\wedge(e^3_2=$
$x_3\vee e^3_2=y_3)\wedge(x_1\neq x_2\vee y_1=y_2)\wedge(x_1\notin x_2\vee x_2\notin x_3\vee x_1\in x_3))\wedge(\exists p\in$
$\sigma)(\exists p_1\in p)(\exists p_2\in p)(\forall e\in p)(\forall e_1\in p_1)(\forall e_2\in p_2)((e=p_1\vee e=p_2)\wedge e_1=$
$k\wedge(e_2=k\vee e_2=y))\wedge(\forall c\in b)(c=\sigma\vee c=a))\wedge(\forall f\in d)(f=y\vee f=b)))$.

C Metafunctions reference

The following are only used in the present article.

Free$[\varphi]$ Set of free variables of a formula φ.

Form$(\mathcal{L})$ Formulas of the languaje $\mathcal{L}$.

Var$(\mathcal{L})$ Variables of the languaje $\mathcal{L}$.

Vars$[\varphi]$ Set of all variables of a formula φ.

The following metaformulas return the axioms:

AxColl$[(\forall x \in y)\exists z\varphi(x, y, z, a), w]$ Collection axiom for the listed formula and the collection variable w. See definition 6.4.

AxFnd$[\exists x\varphi(x, \vec{z}), y]$ Foundation axiom whose consequent is $\exists x(\varphi(x, \vec{z}) \wedge (\forall y \in x)\neg\varphi(y, \vec{z}))$. See definition 4.2.

AxSep$[\varphi, a, x, y]$ Separation of $x = \{y \in a : \varphi(y)\}$. See definition 6.1.

The following syntactically manipulate formulas:

Collapse$[\exists x_1 ... \exists x_m \varphi(x_1, ..., x_m, \vec{z}), n]$ Collapses n existential quantifiers into one $(n < m)$. See definition 5.5.

$\mathsf{Collapse_n}[\exists x_1 ... \exists x_m \varphi(x_1, ..., x_m, \vec{z})]$ A more succint way to express the same as Collapse$[\exists x_1 ... \exists x_m \varphi(x_1, ..., x_m, \vec{z}), n]$.

Collect$[(\forall x \in y)\exists z\varphi(x, y, z, \vec{a}), w]$ The right-hand part of the Collection axiom, i.e., $\exists w(\forall x \in y)(\exists z \in w)\varphi(x, y, z, \vec{a})$. See definition 6.3.

Enum$[\varphi, f, e, x_1, ..., x_n]$ Basically it is $f = \{x_1, ..., x_n\} \wedge \varphi(f, e, x_1, ..., x_n, \vec{a})$, but with some subtleties and optimizations. See definition 5.3.

ExpandExists$[\varphi_1 \wedge ... \wedge \varphi_n, i]$ Move the existential quantifier of φ_i outside the conjunction (this has to be possible, i.e., $1 \leq i \leq n$, and the quantifier variable must not be free in any φ_j, $j \neq i$). The same operation, but applied to a disjunction instead to a conjunction. There are some additional optimizations to keep formulas simple, see definition 2.18.

ExpandForall$[\varphi_1 \wedge ... \wedge \varphi_n, i]$ The same as ExpandExists, but with universal quantifiers instead of existential. See definition 2.18.

Found$[\exists x\varphi(x, \vec{z}), y]$ The right-hand side of the corresponding Foundation axiom, i.e., $\exists x(\varphi(x, \vec{z}) \wedge (\forall y \in x)\neg\varphi(y, \vec{z}))$. See definition 4.1.

Fun$[\varphi(f, \vec{p}, \vec{x}, \vec{y}, \vec{z}), f; p, e]$ The set f is a function made of pairs $p_i = \langle x_i, y_i\rangle = \{\{p_i^1\}, \{p_i^1, p_i^2\}\}$, $i = 1, 2$, such that $\varphi(f, \vec{p}, \vec{x}, \vec{y}, \vec{z})$; e is an internal stem and may be omitted; if φ does not use any of the p, p may also be omitted. See definition 8.7.

FunDiff$[f_1, f_2, x; p, e]$ Equivalent to $f_1(x) \neq f_2(x)$; f_1 and f_2 are assumed to be functions; p and e are internal variables and may be omitted. See definition 8.8.

FunVal$[f, x, y; p, e]$ Equivalent to $f(x) = y$; f is assumed to be a function; p and e are internal variables and may be omitted. See definition 8.9.

InDomain$[x, f; p, e]$ Equivalent to $x \in \operatorname{dom} f$; f is assumed to be a function; p and e are internal variables and may be omitted. See definition 8.10.

$\mathsf{MoveUp}[(\exists x_1 \in y_1)...(\exists x_n \in y_n)\exists z\varphi(\vec{z}, \vec{y}, z, \vec{a}), n]$ Moves the unbounded existential to the beginning of the formula, i.e., produces the logically equivalent formula $\exists z(\exists x_1 \in y_1)...(\exists x_n \in y_n)\varphi(\vec{z}, \vec{y}, z, \vec{a})$. See definition 2.20.

$\mathsf{Negate}[\varphi]$ Returns a formula logically equivalent to $\neg\varphi$, but where negation has been recursively applied along the syntax tree until atomic formulas are themselves inverted (negated). See definition 2.16.

$\mathsf{Pair}[\varphi, x, y; p, e]$ Equivalent to $p = \{x, y\}$, where p, x and y are free, but with some additional subtleties and optimizations. See definition 7.1.

$\mathsf{Restrict}[f, a, r; p, e]$ Equivalent to $r = f \restriction a$; p and e are internal variables and may be omitted. See definition 8.12.

$\mathsf{Particularize_n}[\exists x_1...\exists x_n\varphi(\vec{x}, \vec{z}), \forall x_1...\forall x_n\psi(\vec{x}, \vec{z})]$ This metafunction returns $\exists x_1...\exists x_n(\varphi(\vec{x}, \vec{z}) \wedge \psi(\vec{x}, \vec{z}))$. See definition 2.25.

$\mathsf{Rel}[\varphi(r, x, y, \vec{z}), r, x, y; p, e]$ Equivalent to $(\forall t \in r)(t = \langle x, y\rangle \wedge \varphi(r, x, y, \vec{z}))$, i.e., r is a relation such that all pairs $\langle x, y\rangle \in r$ verify φ; p and e are internal variables, and may be omitted. See definition 8.3.

$\mathsf{Tran}[x; y, z]$ Equivalent to $(\forall y \in x)(\forall z \in y)(z \in x)$, i.e., x is transitive. See definition 2.27.

$\mathsf{Tuple}[\varphi(t, x, y, \vec{z}), t, x, y; p, e]$ Equivalent to $t = \langle x, y\rangle \wedge \varphi(t, x, y, \vec{z})$; x and y are bound (i.e., created in the metaformula result as elements of elements of t); p and e are internal variables, and may be omitted. See definition 7.4.

$\mathsf{Tuples}[\varphi(\vec{x}, \vec{y}, \vec{z}), n, r, x, y; p, e]$ A generalization of Tuple: x, y, p and e are *stems*, i.e. initial parts of variable names. The metafunction states that r is a relation, and creates n ordered pairs $\langle x_i, y_i\rangle \in r$, $i = 1, ..., n$, such that $\varphi(x_1, ..., x_n, y_1, ..., y_n, \vec{z})$; p and e are stems for internal variables, and may be omitted. See definition 8.5.

$\mathsf{Tuples_n}[\varphi(\vec{x}, \vec{y}, \vec{z}), r, x, y; p, e]$ Equivalent to $\mathsf{Tuples}[\varphi(\vec{x}, \vec{y}, \vec{z}), n, r, x, y; p, e]$.

References

[1] Keith J. Devlin. *Constructibility.* Perspectives in Mathematical Logic. Heildelberg: Springer-Verlag, 1984.

[2] Kenneth Kunen. *Set Theory — An Introduction to Indepencence Proofs.* Vol. 102. Studies in Logic and the Foundation of Mathematics. The Netherlands: Elsevier, 1980.