PROOFLESS TEXT

by
Harvey M. Friedman
Ohio State University
September 29, 2005

1. INFORMAL INTROUDCTION TO PROOFLESS TEXT.

We give an informal account of some of the essential features of proofless text.

i. Proofless text is based on a variant of ZFC with free logic. Here variables always denote, but not all terms denote. If a term denotes, then all subterms must denote. The sets are all in the usual extensional cumulative hierarchy of sets. There are no urelements.

ii. Every set, in addition to having elements, plays some additional roles. It acts, for each n ≥ 1, as an n-ary relation. It also acts, for each n ≥ 1, as an n-ary function. As an n-ary relation, it may fail at any n arguments. As an n-ary function, it may be undefined at any n arguments. We use $s(t_1,...,t_n)$ and $s[t_1,...,t_n]$ for this action. These are are well formed expressions for any terms $s,t_1,...,t_n$, n ≥ 1. The former is a term and the latter is a formula. The former denotes the unique x such that the ordered tuple $<t_1,...,t_n,x>$ lies in s. (If there is no such unique x then s(t1,...,tn) is undefined). The latter holds if and only if the ordered tuple $<t_1,...,t_n>$ lies in s. If any of $s_1,t_1,...,t_n$ are undefined, then $s(t_1,...,t_n)$ is undefined and $s[t_1,...,t_n]$ is false.

iii. The character set for proofless text consists of 153 characters, including the 52 English letters, the 10 digits, and 28 commonly used Greek letters. All characters come from the 12 point Courier New and Symbol fonts. These two fonts are used in virtually all word processors running on personal computer systems worldwide. However, the more sophisticated TEX systems have many advantages, and it makes sense to convert what we do here to TEX.

iv. A character string is a nonempty finite string from the characters, together with an indication of whether any given character i) is normal or subscripted, ii) is normal, boldface, or italic. Thus there are 2 × 3 = 6 possible formats for a character.

v. The subscripting option is used to write, say, x_1, rather than x1. The boldface option is used solely for the purpose of identifying terms that are being used as infix functions. The italic option is used solely for the purpose of identifying terms that are being used as infix relations. Without these boldface and italic options, it becomes awkward or impossible to use certain useful features of the language without running into ambiguity. Blank space is ignored, as is standard in formal languages in computer science contexts.

vi. There are three fundamental disjoint categories of character strings: the variable strings, the function strings, and the relation strings. Constants are treated as 0-ary functions. The nonempty strings of digits are treated as built in constants, denoting nonnegative integers in the usual way. The nonlogical strings are the function, and relation strings. A given function string can appear in many arities $k \geq 0$, and a given relation string can appear in many arities $k \geq 1$. This can happen even in a single term or formula.

vii. Proofless text consists of a series of definitions, axioms, lemmas, and theorems. Within each of the three categories, the labels are distinct.

viii. Every definition either a) partially defines a function string in one particular mode; or b) partially defines a relation string in one particular mode. A mode is just an arity, or the special mode called infix.

ix. Unorderd and ordered tupling is built in, and these eliminate any need to define a function string or relation string in all arities at once. Here we say "partially defined" in three senses. Firstly, the definition may leave open the meaning of the function string or relation string at some arguments. (This is not allowed in the case of a function string in 0-ary mode, which must be entirely defined). Secondly, function strings may be undefined at certain choices of arguments (this does not apply to 0-ary mode and to relation strings). E.g., we may either leave open the value of ÷(x,0), or, as we recommend for arithmetic, we may assert that ÷(x,0) is undefined as part of the definition of the function string ÷ in 2-ary mode.

x. Any definition of a function string in its infix mode may be accompanied by its precedence integer. This can be any integer whatsoever. Smaller integers indicate tighter binding. Left associativity is used to break ties among equal precedence.

xi. Conflicting definitions can arise when the same function, or relation string, is defined twice (or more) with the same mode. In conflicts, the last applicable definition governs.

xii. Definitions are made via formulas or terms. All nonlogical strings used in these formulas or terms must have been previously the subject of a prior Definition. There is no restriction on the variable strings used in any Definition.

xiii. There is no formal distinction between Lemmas and Theorems. A Theorem consists, roughly speaking, of a formula. There are provisions for stating more than formula using commas instead of ∧, and for separating hypotheses from conclusions, using "Let" and "Then".

xiv. A Theorem can use any variable strings. However, all nonlogical strings used in a Theorem must be previously partially defined - i.e., must be the subject of a prior Definition.

xv. A relational type consists of a list of nonlogical strings, where the function and relation strings are given a list of modes. A mode of a relation string is an arity ≥ 1, or "infix mode". The mode of a function string is an arity ≥ 2, or "infix mode". The infix modes for function strings are assigned integers (for precedence). There are some built in strings that are an implicit part of any relational type (e.g., as mentioned before, the nonempty strings of digits).

xvi. In section 3, the σ terms and σ formulas are simultaneously defined by induction. There are some significant built-ins that are used here - extensional set abstraction, extensional lambda abstraction, digit strings, as well as unordered and ordered tupling. In addition, we have chosen to include the set theoretic primitives ∈,∉,⊆,⊇,⊂,⊃,℘,∪,∩,\,∅, and also {...}, <...>. Axioms and rules involving these notions will be built in.

xvii. There are no user created bracketing operators. There are two built in bracketing operators which have arbitrary arity ≥ 1. These are { }, < >. The first two have proscribed meaning, whereas the third has no proscribed meaning.

xviii. There are no user created variable binding operators. Two variable binding operators are built in. They are set abstraction and lambda abstraction. They have proscribed meaning. Their meaning is strictly extensional, thereby avoiding any issues of intensionality.

2. CHARACTERS, CHARACTER STRINGS, RELATIONAL TYPES.

We now present the alphabet of characters, divided into 6 useful categories.

We also give complete instructions for one workable computer system that we use.

We use a standard kind of MacIntosh keyboard with the fonts

12 pt Courier New.
12 pt Symbol.

We also use friendly certain key combinations that are easily set up by the user by going to Insert at the menu bar and choosing Symbol, and then picking the font symbol. Then click on the desired character and fill out the menu for making an assigned key combination.

When using a Windows keyboard, use the Alt key instead of the Control key.

1. English letters. These are lower case and upper case.

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

These are from 12 pt Courier New and are typed in the normal way on the keyboard.

2. Basic Greek letters.

αβγδεζηθκμνξρστφψ

These are from the 12 pt Symbol font. We recommend that the user set up the following key combinations for these basic Greek letters.

α	Control al	“alpha”
β	Control be	“beta”
γ	Control ga	“gamma”
δ	Control de	“delta”
ε	Control ep	“epsilon”
ζ	Control ze	“zeta”
η	Control et	“eta”
θ	Control th	“theta”
κ	Control ka	“kappa”
μ	Control mu	“mu”
ν	Control nu	“nu”
ξ	Control xi	“xi”
ρ	Control rh	“rho”
σ	Control si	“sigma”
τ	Control ta	“tau”
φ	Control ph	“phi”
ψ	Control ps	“psi”

3. Digits.

0123456789

These are from the 12 pt Courier New font and are typed in the normal way on the keyboard.

4. Function characters.

+-^\/|”
∪∩×÷Δ°∅∞℘πχωΠΣΩΨΞΓℵℑ⊕⊗∑∏
•

The first row is from the 12 pt Courier New font, and are typed in the normal way on the keyboard.

The second row is from the 12 pt Symbol font. We recommend that the user set up the following key combinations for these function characters.

∪	Control cu	“cup”
∩	Control ca	“cap”
×	Control cr	“cross”
÷	Control di	“divide”
Δ	Control cd	“capital delta”
°	Control ci	“circle”
∅	Control em	“empty”
∞	Control iy	“infinity”
℘	Control po	“powerset”
π	Control pi	“pi”
χ	Control ch	“chi”
ω	Control om	“omega”
Π	Control cp	“capital pi”
Σ	Control cs	“capital sigma”
Ω	Control cw	“capital omega”
Ψ	Control sh	“capital psi”
Ξ	Control cx	“capital xi”
Γ	Control cg	“capital gamma”
ℵ	Control ha	“hebrew aleph”
ℑ	Control hb	“hebrew beth”
⊕	Control aa	“alternative addition”
⊗	Control am	“alternative multiplication”
∑	Control su	“sum”
∏	Control pr	“product”

The third row is from the 12 pt Courier New font and is typed as

• Option 8 “dot”

The reason that we do not use the corresponding character from 12 pt Symbol is that it does not show up when converting to pdf on MacIntosh computers. The Option 8 is built in and does not need a user generated key combination.

5. Relation characters.

≤≥<>∈∉⊆⊇⊂⊃≡≈
~

These characters are from the 12 point Symbol font. We recommend that the user set up the following key combinations for these relation characters.

≤	Control <	"less than or equal"
≥	Control >	"greater than or equal"
<	Control lt	"less than"
>	Control gt	"greater than"
∈	Control in	"element of"
∉	Control ni	"not element of"
⊆	Control co	"contained in"
⊇	Control ri	"reverse inclusion"
⊂	Control sc	"strict contained in"
⊃	Control sr	"strict reverse inclusion"
≡	Control qu	"equivalence"
≈	Control wi	"wiggle"

The second row is from the 12 point Courier New font, and is typed in the normal way on the keyboard (far upper left key).

6. Decorating characters.

`'*#;

These are from 12 pt Courier New and are typed in the normal way on the keyboard.

7. Reserved characters.

!=[]{}:,<>.
∃∀⊥¬∧∨→↔↑↓λ≅
≠

The first row is from the 12 pt Courier New font and is typed in the normal way on the keyboard. Note that <> are to be distinguished from the relation characters <>.

The second row is from the 12 pt Symbol font. We recommend that the user set up the following key combinations for these relation characters.

∃	Control ex	"there exists"
∀	Control fo	"for all"
⊥	Control ab	"absurdity"

¬	Control no	"not"
∧	Control an	"and"
∨	Control or	"or"
→	Control /	"implies"
↔	Control if	"if and only if"
↑	Control up	"undefined"
↓	Control dn	"defined"
λ	Control la	"lambda"
≅	Control pe	"partial equality"

The third row is from the 12 pt Courier New font and is typed as

≠ Option + "not equaled to"

The reason that we do not use the corresponding character from 12 pt Symbol is that it does not show up when converting to pdf on MacIntosh computers. The Option + is built in and does not need a user generated key combination.

This concludes the discussion of the characters used in Proofless Text.

A displayed character consists of a character together with an element of {normal, subscript, superscript} × {normal, boldface, italic}. This ordered pair is called the display of the character.

The character strings consist of the nonempty finite strings of displayed characters. Note that a character can appear in a character string in any one of 9 displays.

We now define important categories of character strings.

1. Variable strings. These are character strings starting with any English letter in (normal,normal), or basic Greek letter in (normal,normal), followed by zero or more digits in (subscript,normal), followed by zero or more decorating characters in (normal,normal).

TWO EXAMPLES. x α_5'

2. Function strings. These are character strings that start with either

i. A function character in (normal,normal).
ii. Two or more English letters in (normal,normal), the first of which is capitalized, and the remainder of which are in lower case.
iii. A digit in (normal,normal).

This is followed by zero or more of the following:

any character in (subscript,normal).
any decorating character in (normal,normal).

In case iii, there must be at least one later character.

TWO EXAMPLES. $+_{\Re}$ Exp# 0*

3. Relation strings. These are character strings that start with either

i. A relation character in (normal,normal).
ii. Two or more English letters in (normal,normal), all of which are capitalized.

This is followed by zero or more of the following:

any character in (subscript,normal).
any decorating character in (normal,normal).

TWO EXAMPLES. $\leq_Z$ $CONT_{\Re\Re}$

The nonlogical strings consist of the function strings and the relation strings. Note that there cannot be any itaic or boldface in any nonlogical string.

An applied function string is a pair (x,k) or (x,infix), where x is a function string and $k \geq 0$. The idea is that (x,k) indicates x in the mode of a k-ary function symbol, and (x,infix) indicates x in the mode of an infix function symbol.

An applied relation string is a pair (x,k) or (x,infix), where x is a relation string and $k \geq 1$. The idea is that (x,k) indicates x in the mode of a k-ary relation symbol, and (x,infix) indicates x in the mode of an infix relation symbol.

A relational type σ consists of a list of applied function strings and applied relation strings, provided σ contains the following built in applied function strings and applied relation strings.

a. (∪,1), (∪,infix), (∩,1), (∩,infix), (\,infix), (∅,0), (ω,0), (℘,1).

b. (∈,infix), (∉,infix), (⊆,infix), (⊇,infix), (⊂,infix), (⊃,infix).

3. σ TERMS AND σ FORMULAS.

We fix a relational type σ. We simultaneously define the

i. σ terms.
ii. bracketed σ terms.
iii. σ formulas.
iv. bracketed σ formulas.

1. Every variable string is a bracketed σ term. Every nonempty string of digits is a bracketed σ term. ⊥ is a bracketed σ formula. Let α be an applied function string in σ with arity 0. Then α is a bracketed σ term.

2. Let $k \geq 1$ and $s,t_1,...,t_k$ be σ terms. Let r be a bracketed σ term. Then

(s)
$\{t_1,...,t_k\}$
$<t_1,...,t_k>$
$r(t_1,...,t_k)$

are bracketed σ terms.

3. Let $k \geq 1$ and $r,s,t_1,...,t_k$ be σ terms. Then

$s\uparrow$
$s\downarrow$
$s = t$
$s \neq t$
$s \cong t$
$s[t_1,...,t_k]$

are bracketed σ formulas.

4. Let α be an applied function string in σ with arity 1, whose leading character is a function character, and t be a bracketed σ term. Then αt is a bracketed σ term.

5. Let k ≥ 1 and α be an applied function string in σ with arity k ≥ 1. Let $s_1,...,s_k$ be σ terms. Then $\alpha(s_1,...,s_k)$ is a bracketed σ term.

6. Let k ≥ 2. For each 1 ≤ i ≤ k-1, let α_i be
i) an infix function string in σ; or
ii) a σ term with no boldface or italic characters.
Let $s_1,...,s_k$ be bracketed σ terms. Then

$$s_1\ \alpha_1\ s_2\ ...\ \alpha_{k-1}\ s_k$$

is a σ term, where each α_i with ii) appears in boldface.

7. Let β be an applied relation string in σ with arity k ≥ 1. Let $s_1,...,s_k$ be σ terms. Then $\beta[s_1,...,s_k]$ is a bracketed σ formula.

8. Let p ≥ 2 and $n_1,...,n_p \geq 1$. For 1 ≤ i ≤ p and 1 ≤ j ≤ n_i, let s_{ij} be a σ term. For all 1 ≤ i < p, let β_i be
i) an infix relation string in σ; or
ii) =, ≠, or ≅; or
iii) a σ term with no boldface or italics.
Then

$$s_{11},...,s_{1,n1}\ \beta_1\ s_{21},...,s_{2,n2}\ ...\ \beta_{p-1}\ s_{p1},...,s_{p,np}$$

is a bracketed σ formula, where the β_i with iii) appear in italics.

9. Let k,n ≥ 1, $v_1,...,v_k$ be distinct variables, $t,s_1,...,s_n$ be σ terms, φ be a bracketed σ formula, and ψ be a σ formula. Let α be
i) an infix relation string in σ; or
ii) =, ≠, or ≅; or
iii) a σ term with no boldface or italics.
Then

$(!t)\varphi$
$(!t)(\psi,\ v_1,...,v_k$ fixed)
$(!t\ \alpha\ s_1,...,s_n)\varphi$
$(!t\ \alpha\ s_1,...,s_n)(\psi,\ v_1,...,v_k$ fixed)

$\{t:\ \psi\}$
$\{t:\ \psi,\ v_1,...,v_k$ fixed}
$\{t\ \alpha\ s_1,...,s_n:\ \psi\}$
$\{t\ \alpha\ s_1,...,s_n:\ \psi,\ v_1,...,v_k$ fixed}

are bracketed σ terms, where the α with iii) are italicized.

10. Let $k,n \geq 1$, $v_1,...,v_k$ be distinct variables, $s_1,...,s_n$ be σ terms, t a bracketed σ term, and φ be a σ formula. Let α be
i) an infix relation string in σ; or
ii) =, ≠, or ≅; or
iii) a σ term with no boldface or italics.
Then

$(\lambda v_1,...,v_k)t$
$(\lambda v_1,...,v_k:\varphi)t$
$(\lambda v_1,...,v_k\ \alpha\ s_1,...,s_n)t$
$(\lambda v_1,...,v_k\ \alpha\ s_1,...,s_n:\varphi)t$

are bracketed σ terms, where the α with iii) are italicized.

11. Let $k \geq 2$ and $\varphi,\rho_1,...,\rho_k$ be σ formulas. Let ψ be a bracketed σ formula. Let $\alpha_1,...,\alpha_{k-1}$ be among $\vee,\wedge,\vee\vee,\rightarrow,\leftrightarrow$.
Then

(φ)
$\neg\psi$
$\rho_1\ \alpha_1\ \rho_2\ ...\ \alpha_{k-1}\ \rho_k$
$![\rho_1,...,\rho_k]$

are σ formulas. The first two and the fourth are bracketed σ formulas.

12. Let $k,n,m \geq 1$, $v_1,\ldots,v_k$ be distinct variables, and $t_1,\ldots,t_n,s_1,\ldots,s_m$ be σ terms. Assume that t1,...,tn are distinct. Let α be
i) a relation string listed with infix mode in σ; or
ii) =, ≠, or ≅; or
iii) a σ term with no boldface or italics.
Let φ be a bracketed σ formula and ψ,ρ be σ formulas. Then

$(\forall t_1,\ldots,t_n)\varphi$
$(\exists t_1,\ldots,t_n)\varphi$
$(\exists ! t_1,\ldots,t_n)\varphi$
$(\forall t_1,\ldots,t_n\ \alpha\ s_1,\ldots,s_m)\varphi$
$(\exists t_1,\ldots,t_n\ \alpha\ s_1,\ldots,s_m)\varphi$
$(\exists ! t_1,\ldots,t_n\ \alpha\ s_1,\ldots,s_m)\varphi$
$(\forall t_1,\ldots,t_n)(\psi,\ v_1,\ldots,v_k$ fixed)
$(\exists t_1,\ldots,t_n)(\psi,\ v_1,\ldots,v_k$ fixed)
$(\exists ! t_1,\ldots,t_n)(\psi,\ v_1,\ldots,v_k$ fixed)
$(\forall t_1,\ldots,t_n\ \alpha\ s_1,\ldots,s_m)(\psi,\ v_1,\ldots,v_k$ fixed)
$(\exists t_1,\ldots,t_n\ \alpha\ s_1,\ldots,s_m)(\psi,\ v_1,\ldots,v_k$ fixed)
$(\exists ! t_1,\ldots,t_n\ \alpha\ s_1,\ldots,s_m)(\psi,\ v_1,\ldots,v_k$ fixed)
$(\forall t_1,\ldots,t_n:\rho)\varphi$
$(\exists t_1,\ldots,t_n:\rho)\varphi$
$(\exists ! t_1,\ldots,t_n:\rho)\varphi$

are bracketed σ formulas, where the α for which iii) holds are italicized.

This concludes the definition of σ terms and σ formulas.

4. UNIQUE PARSING.

We claim unique parsing.

5. FREE AND BOUND OCCURRENCES.

6. DEFINITIONS, LEMMAS, AND THEOREMS OVER σ.

Let σ be a relational type. The body of a definition over σ must begin with the phrase

k-ary α.

This is called the subject of the definition. Here either α is a function string and $k \geq 0$, or α is a relation string and $k \geq 1$. There is no other restriction on k,α.

Let us first take up the case where α is a relation string. Then the definition is required to take one of the following forms.

k-ary relation α. $\alpha[v_1,\ldots,v_k] \leftrightarrow \psi$.

k-ary relation α. If φ_1 then $\alpha[v_1,\ldots,v_k] \leftrightarrow \psi_1$. ... If φ_n then $\alpha[v_1,\ldots,v_k] \leftrightarrow \psi_n$.

k-ary relation α. If φ_1 then $\alpha[v_1,\ldots,v_k] \leftrightarrow \psi_1$. ... If φ_n then $\alpha[v_1,\ldots,v_k] \leftrightarrow \psi_n$. Otherwise $\alpha[v_1,\ldots,v_k] \leftrightarrow \psi_{n+1}$.

Infix relation α. $v\ \alpha\ w \leftrightarrow \psi$.

Infix relation α. If φ_1 then $v\ \alpha\ w \leftrightarrow \psi_1$. ... If φ_n then $v\ \alpha\ w \leftrightarrow \psi_n$.

Infix relation α. If φ_1 then $v\ \alpha\ w \leftrightarrow \psi_1$. ... If φ_n then $v\ \alpha\ w \leftrightarrow \psi_n$. Otherwise $v\ \alpha\ w \leftrightarrow \psi_{n+1}$.

For the first four forms, $k,n \geq 1$, $v_1,\ldots,v_k$ are distinct variables, and $\varphi_1,\ldots,\varphi_n,\psi,\psi_1,\ldots,\psi_{n+1}$ are bracketed σ formulas whose free variables are among $v_1,\ldots,v_k$.

For the last three forms, $n \geq 1$, v,w are distinct variables, and $\varphi_1,\ldots,\varphi_n,\psi,\psi_1,\ldots,\psi_{n+1}$ are bracketed σ formulas whose free variables are among v,w.

We now come to the case where α is a function string.

0-ary function α. $\alpha \cong t$.

k-ary function α. $\alpha(v_1,\ldots,v_k) \cong t$.

k-ary function α. If φ_1 then $\alpha(v_1,\ldots,v_k) \cong t_1$. ... If φ_n then $\alpha(v_1,\ldots,v_k) \cong t_n$.

k-ary function α. If φ_1 then $\alpha(v_1,\ldots,v_k) \cong t_1$. ... If φ_n then $\alpha(v_1,\ldots,v_k) \cong t_n$. Otherwise $\alpha(v_1,\ldots,v_k) \cong t_{n+1}$.

Infix function α. v α w $\cong$ t. Precedence r.

Infix function α. If φ_1 then v α w $\cong$ ψ_1. ... If φ_n then v α w $\cong$ ψ_n. Precedence r.

Infix function α. If φ_1 then v α w $\cong$ ψ_1. ... If φ_n then v α w $\cong$ ψ_n. Otherwise v α w $\cong$ ψ_{n+1}. Precedence r.

For the first form, t is a σ term with no free variables.

For the next three forms, $k,n \geq 1$, $v_1,...,v_k$ are distinct variables, and $\varphi_1,...,\varphi_n,t,t_1,...,t_{n+1}$ are σ terms whose free variables are among $v_1,...,v_k$.

For the final three forms, $n \geq 1$, v,w are distinct variables, and $\varphi_1,...,\varphi_n,t,t_1,...,t_{n+1}$ are σ terms whose free variables are among v,w. Also r is either a finite string of digits, or a finite string of digits with a minus sign in front.

We will also have the following two special empty forms of definitions:

k-ary relation α.

Such empty definitions have a special status in the semantics: α is being declared to be a free relation symbol. This has a special role in the semantics.

This concludes the treatment of bodies of Definitions.

As indicated earlier, we will not distinguish between the bodies of Lemmas, and Theorems.

Here are the forms of bodies of Lemmas and Theorems.

$\psi_1,...,\psi_m$.

Let $\varphi_1,...,\varphi_n$. Then $\psi_1,...,\psi_m$.

Here $n,m \geq 1$ and $\varphi_1,...,\varphi_n,\psi_1,...,\psi_m$ are σ formulas.

The idea here is that there are r statements. The i-th statement can be viewed as universally asserting

Definitions take the form

DEFINITION L. body

Here L is the label of the definition. L can be any character string, provided that the entire definition is unambiguous. This will the case unless the label L is truly bizarre.

NOTE: We do not allow definitions of the eight built in applied relation strings and the six built in applied function strings at the end of section 2 above.

Lemmas and Theorems take the form

LEMMA L. body
THEOREM L. body

Here L is the label of the Lemma, or Theorem. L can be any character string.

7. SYNTAX OF PROOFLESS TEXT.

A proofless text is a character string μ subject to several requirements.

The first requirement is that for some $n \geq 1$, μ is the concatenation of n strings $\mu_1,...,\mu_n$, each of which starts with either DEFINITION, LEMMA, or THEOREM (the header).

The second requirement is that there is a sequence of relational types $\sigma_1,...,\sigma_n$ such that

i. σ_1 is the empty relational type.
ii. μ_i is a Definition, Lemma, or Theorem over σ_i, according to its header.
iii. If μ_i is a Lemma or Theorem then $\sigma_{i+1} = \sigma_i$.
iv. If μ_i is a Definition of α in k-ary mode, then σ_{i+1} is σ adjoined with the listing of α in k-ary mode.
v. If μ_i is a Definition of α in infix mode, then σ_{i+1} is σ adjoined with the listing of α in infix mode.

The final requirement is that the labels of all of the Definitions be distinct, the labels of all of the Lemmas be distinct, Theorems be distinct.

Note that $\sigma_1,...,\sigma_n$ are unique if they exist.

The relational type of the proofless text is taken to be σ_n.

8. SEMANTICS OF TERMS AND FORMULAS.

Fix a relational type σ. A σ structure consists of the following data, subject to a requirement.

i. A nonempty class D (the domain).
ii. An interpretation of all function strings in their listed modes in σ.
iii. An interpretation of all relation strings in their listed modes in σ, except for those that are free.
iv. D together with the interpretation of ∈ forms a model of ZFC.
v. The interpretations of infix ∉, 1-ary ∪, infix ∪, 1-ary ∩, infix ∩, infix \, 1-ary ℘, 0-ary ∅, infix ⊆, infix ⊇, infix ⊂, infix ⊃, are as they normally are in the model (D,∈^) of ZFC. (Here ∈^ is from iv).

The interpretation of the function string α with arity 0 is written α:0, and is either an element of D or the special object *. Here * ∉ D, and will be used to indicate "undefined".

The interpretation of the function string α with arity k ≥ 0 is written α:k, and is a function from D^k into D ∪ {*}.

The interpretation of the infix function string α string is written α^, and is a function from D^2 into D ∪ {*}.

The interpretation of the relation string β with arity k ≥ 1 is written β:k, and is a subset of D^k.

The interpretation of the infix relation string β is written β^, and is a subset of D^2.

Note that we do not require α:2 and α^ to be the same, or β:2 and β^ to be the same.

We write σ structures in the form M = (D,∈^,...).

A D assignment is a function f from the set of all variables into D. (Note that we say "function" instead of "partial function" here).

We simultaneously define

Val(M,t,f)
Sat(M,φ,f)

where t is a σ term with no free relations or functions, and φ is a σ formula with no free relations or functions, and f is an D assignment.

Here every Val(M,t,f) is an element of D ∪ {*}.

Here Sat is a ternary relation. Thus Sat(M,φ,f) either holds or fails.

In the definition below, σ and M are fixed, and t,φ,f vary. We take this to be a definition by induction on unambiguous σ terms and unambiguous σ formulas.

1. For variables v, Val(M,v,f) = f(v).

For nonempty strings of digits, μ, Val(M,μ,f) is the usual interpretation of the nonnegative integer represented by μ in the ZFC model (D,∈^), as an internally finite epsilon connected set.

¬Sat(M,⊥,f).

Val(M,α,f) = f(α:0).

2. Val(M,(s),f) = Val(M,s,f).

If Val(M,t_1,f),...,Val(M,t_k,f) ∈ D, then Val(M,{t_1,...,t_k},f) is the usual interpretation of the set {Val(M,t_1,f),...,Val(M,t_k,f)} in the ZFC model (D,∈^). Otherwise, Val(M,{t_1,...,t_k},f) = *.

If $Val(M,t_1,f),\ldots,Val(M,t_k,f) \in D$, then $Val(M,<t_1,\ldots,t_k>,f)$ is the usual interpretation of the ordered tuple $<Val(M,t_1,f),\ldots,Val(M,t_k,f)>$ in the ZFC model $(D,\in\hat{})$. Here we use left associativity, with the usual internal pairing operation {{a},{a,b}}.
Otherwise, $Val(M,<t_1,\ldots,t_k>,f) = *$.

$Val(M,s(t_1,\ldots,t_k),f)$ is the unique x ∈ D such that $Val(M,<t_1,\ldots,t_k,x>,f) \in\hat{}\ Val(M,s,f)$; * otherwise.

3. Sat(M,s↑,f) if and only if Val(M,s,f) ∈ D.

Sat(M,s↓,f) if and only if Val(M,s,f) = *.

Sat(M,s = t,f) if and only if Val(M,s,f) = Val(M,t,f) ∈ D.

Sat(M,s ≠ t,f) if and only if Val(M,s,f),Val(M,t,f) ∈ D and Val(M,s,f) ≠ Val(M,t,f).

Sat(M,s ≅ t,f) if and only if Val(M,s,f) = Val(M,t,f).

Sat(M,s ∈ t,f) if and only if Val(M,s,f) ∈^ Val(M,t,f).

Sat(M,s ∉ t,f) if and only if Val(M,s,f),Val(M,t,f) ∈ D and not Val(M,s,f) ∈^ Val(M,t,f).

$Sat(M,s[t_1,\ldots,t_k],f)$ if and only if $Val(M,<t_1,\ldots,t_k>,f) \in\hat{}$ Val(M,s,f).

4. Val(M,αt,f) = α:1(Val(M,t,f)) if Val(M,t,f) ∈ D; * otherwise.

5. $Val(M,\alpha(s_1,\ldots,s_k),f) = \alpha{:}k(Val(M,s_1,f),\ldots,Val(M,s_k,f))$ provided $Val(M,s_1,f),\ldots,Val(M,s_k,f) \in D$; * otherwise.

6. We now come to $Val(M,s_1\ \alpha_1\ s_2\ \ldots\ \alpha_{k-1}\ s_k,f)$.
case 1. $Val(M,s_1,f),\ldots,Val(M,s_k,f) \in D$, and for σ terms α_i, $Val(M,\alpha_i,f) \in D$. Combine $Val(M,s_1,f),\ldots,Val(M,s_k,f)$ using the interpretations of the α_i as follows. For α_i that are not σ terms, use $\alpha_i\hat{}$ as its interpretation. To determine the order of the k-1 operations, use the precedence numbers for the $\alpha_i\hat{}$, and consider the terms α_i as having precedence number ∞. If we run into * anywhere during the computation, return *.

case 2. Otherwise. Then $\mathrm{Val}(M,s_1\ \alpha_1\ s_2\ \ldots\ \alpha_{k-1}\ s_k) = *$.

7. $\mathrm{Sat}(M,\beta[s_1,\ldots,s_k],f)$ if and only if $\mathrm{Val}(M,\langle s_1,\ldots,s_k\rangle,f) \in \beta{:}k$.

8. $\mathrm{Sat}(M,s_{11},\ldots,s_{1,n1}\ \beta_1\ s_{21},\ldots,s_{2,n2}\ \ldots\ \beta_{p-1}\ s_{p1},\ldots,s_{p,np},f)$ if and only if the following holds for all $1 \leq i \leq p-1$.

case 1. β_i is a relation string or $\in$. For all $1 \leq p \leq n_i$ and $1 \leq q \leq n_{i+1}$, $\langle \mathrm{Val}(M,s_{ip},f),\mathrm{Val}(M,s_{i+1,q},f)\rangle \in \beta_i\hat{}$.

case 2. β_i is $\notin$. For all $1 \leq p \leq n_i$ and $1 \leq q \leq n_{i+1}$, $\mathrm{Val}(M,s_{ip},f),\mathrm{Val}(M,s_{iq},f) \in D$ and not $\mathrm{Val}(M,s_{ip},f) \in\hat{}\ \mathrm{Val}(M,s_{i+1,q},f)$.

case 3. β_i is $=$. For all $1 \leq p \leq n_i$ and $1 \leq q \leq n_{i+1}$, $\mathrm{Val}(M,s_{ip},f) = \mathrm{Val}(M,s_{iq},f) \in D$.

case 4. β_i is $\neq$. For all $1 \leq p \leq n_i$ and $1 \leq q \leq n_{i+1}$, $\mathrm{Val}(M,s_{ip},f),\mathrm{Val}(M,s_{iq},f) \in D$ and $\mathrm{Val}(M,s_{ip},f) \neq \mathrm{Val}(M,s_{i+1,q},f)$.

case 5. β_i is $\cong$. For all $1 \leq p \leq n_i$ and $1 \leq q \leq n_{i+1}$, $\mathrm{Val}(M,s_{ip},f) = \mathrm{Val}(M,s_{i+1,q},f)$.

9. For the sake of brevity, we will only handle the first four items.

We evaluate $\mathrm{Val}(M,(!t)\varphi,f)$ by cases.

case 1. There exists a unique $x \in D$ such that the following holds. There exists a D assignment g such that every argument at which f,g differ is free in t, and

i) $\mathrm{Sat}(M,\varphi,g)$;

ii) $\mathrm{Val}(M,t,g) = x$.

Return x.

case 2. Otherwise. Return *.

We evaluate $\mathrm{Val}(M,(!t)(\varphi,\ v_1,\ldots,v_k \text{ fixed}),f)$ by cases.

case 1. There exists a unique $x \in D$ such that the following holds. There exists a D assignment g such that every argument at which f,g differ is free in t and not among $v_1,\ldots,v_k$, and

i) $\mathrm{Sat}(M,\varphi,g)$;

ii) $\mathrm{Val}(M,t,g) = x$.

Return x.

case 2. Otherwise. Return *.

We evaluate $Val(M,(!t\ \alpha\ s_1,\ldots,s_n)\varphi,f)$ by cases.

case 1. There exists a unique $x \in D$ such that the following holds. There exists a D assignment g such that every argument at which f,g differ is free in t and
i) $Sat(M,t\ \alpha\ s_1,\ldots,s_n,g)$;
ii) $Sat(M,\varphi,g)$;
iii) $Val(M,t,g) = x$.
Return x.
case 2. Otherwise. Return *.

We evaluate $Val(M,(!t\ \alpha\ s_1,\ldots,s_n)(\varphi,\ v_1,\ldots,v_k \text{ fixed}),f)$ by cases.

case 1. There exists a unique $x \in D$ such that the following holds. There exists a D assignment g such that every argument at which f,g differ is free in t and not among $v_1,\ldots,v_k$, and
i) $Sat(M,t\ \alpha\ s_1,\ldots,s_n,g)$;
ii) $Sat(M,\varphi,g)$;
iii) $Val(M,t,g) = x$.
Return x.
case 2. Otherwise. Return *.

10. For the sake of brevity, we only handle the first four items.

We evaluate $Val(M,(\lambda v_1,\ldots,v_k)t,f)$ by cases.

For $x,y_1,\ldots,y_k \in D$, we write $x(y_1,\ldots,y_k)$ for the unique z $\in$ D such that $\langle y_1,\ldots,y_k,z\rangle \in^\wedge x$ if such a unique z exists; * otherwise. Here < > is the internal k+1-tupling in the ZFC model $(D,\in^\wedge)$, with left associativity.

case 1. There exists $x \in D$ such that the following holds. x is a k-ary function internal to the ZFC model $(D,\in^\wedge)$. For all $y_1,\ldots,y_k \in D$, $x(y_1,\ldots,y_k) = Val(M,t,f[v_1/y_1,\ldots,v_k/y_k])$. Since $(D,\in^\wedge)$ is extensional, x is unique. Return x.

case 2. Otherwise. Return *.

We evaluate $Val(M,\lambda v_1,\ldots,v_k{:}\varphi\ t,f)$ by cases.

case 1. There exists $x \in D$ such that the following holds. x is a k-ary function internal to the ZFC model $(D,\in\hat{})$. For all $y_1,\ldots,y_k \in D$ with $Sat(M,\varphi,f[v_1/y_1,\ldots,v_k/y_k])$, we have $x(y_1,\ldots,y_k) = Val(M,t,f[v_1/y_1,\ldots,v_k/y_k])$. For the remaining $y_1,\ldots,y_k \in D$, we have $x(y_1,\ldots,y_k) = *$. Since $(D,\in\hat{})$ is extensional, x is unique. Return x.

case 2. Otherwise. Return *.

We evaluate $Val(M,(\lambda v_1,\ldots,v_k\ \alpha\ s_1,\ldots,s_n)t,f)$ by cases.

case 1. There exists $x \in D$ such that the following holds. x is a k-ary function internal to the ZFC model $(D,\in\hat{})$. For all $y_1,\ldots,y_k \in D$ with $Sat(M,v_1,\ldots,v_k\ \alpha\ s_1,\ldots,s_n, f[v_1/y_1,\ldots,v_k/y_k])$, we have $x(y_1,\ldots,y_k) = Val(M,t,f[v_1/y_1,\ldots,v_k/y_k])$. For the remaining $y_1,\ldots,y_k \in D$, we have $x(y_1,\ldots,y_k) = *$. Since $(D,\in\hat{})$ is extensional, x is unique. Return x.

case 2. Otherwise. Return *.

We evaluate $Val(M,(\lambda v_1,\ldots,v_k\ \alpha\ s_1,\ldots,s_n{:}\varphi)\ t,f)$ be cases.

case 1. There exists $x \in D$ such that the following holds. x is a k-ary function internal to the ZFC model $(D,\in\hat{})$. For all $y_1,\ldots,y_k \in D$ with $Sat(M,v_1,\ldots,v_k\ \alpha\ s_1,\ldots,s_n, f[v_1/y_1,\ldots,v_k/y_k])$, $Sat(M,\varphi,f[v_1/y_1,\ldots,v_k/y_k])$, we have $x(y_1,\ldots,y_k) = Val(M,t,f[v_1/y_1,\ldots,v_k/y_k])$. For the remaining $y_1,\ldots,y_k \in D$, we have $x(y_1,\ldots,y_k) = *$. Since $(D,\in\hat{})$ is extensional, x is unique. Return x.

case 2. Otherwise. Return *.

11. $Sat(M,(\varphi),f)$ if and only if $Sat(M,\varphi,f)$.

$Sat(M,\neg\varphi,f)$ if and only if not $Sat(M,\varphi,f)$.

$Sat(M,\rho_1\ \alpha_1\ \rho_2\ \ldots\ \alpha_{k-1}\ \rho_k,f)$ if and only if the following holds. Let $c_1,\ldots,c_k$ be the truth values of $Sat(M,\rho_1,f),\ldots,Sat(M,\rho_k,f)$, respectively.

First parenthesize $c_1\ \alpha_1\ c_2\ \ldots\ \alpha_{k-1}\ c_k$, using the precedence table

∧
∨∨
∨
→ ↔

This creates blocks of ∧'s, blocks of ∨∨'s, blocks of ∨, and blocks consisting of →,↔.

First evaluate the ∧ blocks. Then evaluate the resulting ∨∨ blocks, where ∨∨ is exclusive or. Then evaluate the resulting ∨ blocks. Finally, evaluate the resulting blocks of →,↔ conjunctively. E.g.,

F ↔ F ↔ F → T ↔ T

comes out T since each of

F ↔ T
F ↔ F
F → T
T ↔ T

come out T.

If the truth value is true, then Sat(M,ρ_1 α_1 ρ_2 ... α_{k-1} ρ_k,f). Otherwise, not Sat(M,ρ_1 α_1 ρ_2 ... α_{k-1} ρ_k,f).

Sat(M,![ρ_1,...,ρ_k],f) if and only if there is a unique i such that Sat(M,ρ_i,f).

12. For the sake of brevity, we only handle items 1,4,8,12.

Sat(M,(∀t_1,...,t_n)φ,f) if and only if the following holds. Let g be a D assignment which agrees with f at the variables that are not free in t_1,...,t_n. If Val(M,t_1,g),...,Val(M,t_n,g) ∈ D then Sat(M,φ,g).

Sat(M,(∀t_1,...,t_n α s_1,...,s_m)φ,f) if and only if the following holds. Let g be a D assignment which agrees with f at the variables that are not free in t_1,...,t_n. If Sat(M,t_1,...,t_n α s_1,...,s_m) then Sat(M,φ,g).

Sat(M,(∃t_1,...,t_n)(φ, v_1,...,v_k fixed),f) if and only if the following holds. There exists a D assignment g which agrees with f at the variables that are not free in $t_1,...,t_n$, and at $v_1,...,v_k$, where Val(M,t_1,g),...,Val(M,t_n,g) ∈ D and Sat(M,φ,g).

Sat(M,(∃!$t_1,...,t_n$ α $s_1,...,s_m$)(φ, $v_1,...,v_k$ fixed)) if and only if the following holds. There exists a D assignment g which agrees with f at the variables that are not free in $t_1,...,t_n$, and at $v_1,...,v_k$, where Sat(M,$t_1,...,t_n$ α $s_1,...,s_m$,g). Furthermore, among such g, the values Val(M,t_1,g),..., Val(M,t_n,g) are unique.

This concludes the definition of Val(M,t,f) and Sat(M,φ,f).

We define Sat(M,φ) if and only if for all D assignments f, Sat(M,φ,f).

We now define Sat(M,ψ), where ψ is a σ formula that may have free relations. We require that Sat(M,ρ) holds for all ρ obtained by substituting σ formulas without free relations for free relations in ψ. Here we require that the substitution be legal in the usual sense.

For sets K of σ formulas, we define Sat(M,K) if and only if for all φ ∈ K, Sat(M,φ).

Let T be a proofless text with relational type σ. We write T(def) for the set of all definitions in T.

We say that T is true if and only if for all σ structures M with D = V and ∈^ = ∈, if Sat(M,T(def)) then Sat(M,T).

We say that T is valid if and only if for all σ structures M, if Sat(M,T(def)) then Sat(M,T).

9. SOME VALID PROOFLESS TEXT 1.

In sections 9-15 below, we present some valid proofless text based on [1]. We follow the presentation of elementary set theory there quite closely.

In this proofless text below, we choose to separately label the Definitions and Lemmas. This is of course optional.

We certainly to not think of the presentation in [1] as any kind of optimal presentation for our purposes. However, we think that this material is Ideal for an initial testing of the power of our setup for proofless text.

We also think it Ideal for an initial intensive investigation into proof structure. In Particular, this section 10 is based on [1], chapter 2, and basically consists of completely straightforward definitions and lemmas. We expect that most of these lemmas are self proving. I.e., the computer should generate completely readable proofs without user interaction. Whatever is not self proving should be generated by the computer with rather minimal user interaction. We will intensively investigate just what the required level of computer interaction should be.

The later sections contain more advanced material requiring more substantial user interaction. These include the equipollence of $\Re$ with Maps(ω,{0,1}), Shroeder Bernstein theorem, and the Zermelo well ordering theorem. There are important intermediate cases requiring some notable level of user interaction but much less than these.

We have found that we can conveniently avoId using certain features of proofless text. It may turn out that because of the power of our setup, we can continue to conveniently avoId using these features when we get into more advanced material in other Parts of mathematics. Alternatively, we may find that we cannot continue to avoId using these features. Another possibility is the use of text macros that the user can activate, but are not Part of the semantics of proofless text. Such macros would serve only to simplify what appears on the computer screen. It could also serve to simplify what appears on the printout of readable text, but with some Segregated printout of the macro being used.

These features we deliberately avoid using are

a. Conflicting definitions. E.g., we have many different reflexive and irreflexive orderings, and we use subscripts to indicate what they are. We do however, use the same nonlogical string with different arity.

b. Incompletely defined nonlogical strings. In all of the definitions made here of functions (function strings), the definitions determine completely what the function returns at all possible arguments - even if it is to return "undefined". The truth value of defined relations (relation strings) at all arguments is also determined.

c. Precedence declarations for infix functions. The additional use of parentheses that this entails does not seem to be visually disturbing - given all of our other rich features.

d. More generally, we do not push unique parsing nearly as hard as we could. E.g., we always put parentheses around quantifiers.

DEFINITION 1.1. Infix function Δ. $x \Delta y \cong (x\backslash y) \cup (y\backslash x)$.

DEFINITION 1.2. Infix function $\times$. $x \times y \cong \{<z,w>: z \in x \wedge w \in y\}$.

LEMMA 1.1. $x \Delta y\downarrow$, $x \times y\downarrow$.

LEMMA 1.2. $\cap x\downarrow \leftrightarrow x \neq \emptyset$.

LEMMA 1.3. $x = \emptyset \leftrightarrow (\forall x)(x \notin A)$.

LEMMA 1.4. $x \subseteq x$.

LEMMA 1.5. $x \subseteq y \wedge y \subseteq x \rightarrow x = y$.

LEMMA 1.6. $x \subseteq \emptyset \leftrightarrow x = \emptyset$.

LEMMA 1.7. $x \subseteq y \wedge y \subseteq z \rightarrow x \subseteq z$.

LEMMA 1.9. $x \subset y \wedge y \subseteq z \rightarrow x \subset y$.

LEMMA 1.9. $x \subseteq y \wedge y \subset z \rightarrow x \subset z$.

LEMMA 1.10. $\neg x \subset x$.

LEMMA 1.11. $x \subset y \rightarrow \neg y \subseteq x$.

LEMMA 1.12. $x \subset y \rightarrow x \subseteq y$.

LEMMA 1.13. $x \in y \cup z \leftrightarrow x \in y \vee x \in z$.

LEMMA 1.14. $x \in y \cap z \leftrightarrow x \in y \wedge x \in z$.

LEMMA 1.15. $x \cup y = y \cup x$, $x \cap y = y \cap x$.

LEMMA 1.16. $x \cup x = x \cap x = x$.

LEMMA 1.17. $x \cap \emptyset = \emptyset$.

LEMMA 1.18. $x \cup \emptyset = A$.

LEMMA 1.19. $x \cap y \subseteq x, y \subseteq x \cup y$.

LEMMA 1.20. $(x \cap y) \cap z = x \cap (y \cap z)$.

LEMMA 1.21. $(x \cup y) \cup z = x \cup (y \cup z)$.

LEMMA 1.22. $x \subseteq y \leftrightarrow x \cap y = x$.

LEMMA 1.23. $x \subseteq y \leftrightarrow x \cup y = y$.

LEMMA 1.24. $x \subseteq z \wedge y \subseteq z \leftrightarrow x \cup y \subseteq z$.

LEMMA 1.25. $(x \cup y) \cap z = (x \cap z) \cup (y \cap z)$.

LEMMA 1.26. $(x \cap y) \cup z = (x \cup z) \cap (y \cup z)$.

LEMMA 1.27. $x \backslash x = \emptyset$.

LEMMA 1.28. $x \backslash (x \cap y) = x \backslash y$.

LEMMA 1.29. $x \cap (x/y) = x \backslash y$.

LEMMA 1.30. $(x \backslash y) \cup y = x \cup y$.

LEMMA 1.31. $(x \cup y) \backslash y = x \backslash y$.

LEMMA 1.32. $(x \cap y) \backslash y = \emptyset$.

LEMMA 1.33. $(x \backslash y) \cap y = \emptyset$.

LEMMA 1.34. $x \backslash (y \cup z) = (x \backslash y) \cap (x \backslash z)$.

LEMMA 1.35. $x \backslash (y \cap z) = (x \backslash y) \cup (x \backslash z)$.

LEMMA 1.36. $\neg(\exists x)(\forall y)(y \in x)$.

LEMMA 1.37. $\neg(\exists x)(\forall y)(y \subseteq x)$.

LEMMA 1.38. $x \Delta \emptyset = x$.

LEMMA 1.39. $x \Delta y = y \Delta x$.

LEMMA 1.40. $(x \Delta y) \Delta z = x \Delta (y \Delta z)$.

LEMMA 1.41. $x \cap (y \Delta z) = (x \cap y) \Delta (x \cap z)$.

LEMMA 1.42. $x\backslash y \subseteq x \Delta y$.

LEMMA 1.43. $x = y \leftrightarrow x \Delta y = \emptyset$.

LEMMA 1.44. $x \Delta y = y \Delta z \rightarrow x = z$.

LEMMA 1.45. $z \in \{x,y\} \leftrightarrow z \in x \vee z \in y$.

LEMMA 1.46. $\{x,y\} = \{u,v\} \leftrightarrow (x = u \wedge y = v) \vee (x = v \wedge y = u)$.

LEMMA 1.47. $\{x\} = \{y\} \leftrightarrow x = y$.

LEMMA 1.48. $\langle x,y\rangle = \langle u,v\rangle \leftrightarrow x = u \wedge y = v$.

LEMMA 1.49. $\{x: x \in A\} = A$.

LEMMA 1.50. $\{x: x = x\}\uparrow$.

LEMMA 1.51. $\cup\{x\} = x$, $\cup\{x,y\} = x \cup y$.

LEMMA 1.52. $\cup(x \cup y) = (\cup x) \cup (\cup y)$.

LEMMA 1.53. $x \subseteq y \rightarrow \cup x \subseteq \cup y$.

LEMMA 1.54. $x \in y \rightarrow x \subseteq \cup y$.

LEMMA 1.55. $(\forall x \in y)(x \subseteq z) \leftrightarrow \cup y \subseteq z$.

LEMMA 1.56. $(\forall x \in y)(x \cap z = \emptyset) \rightarrow (\cup y) \cap z = \emptyset$.

LEMMA 1.57. $\cup\langle x,y\rangle = \{x,y\}$.

LEMMA 1.58. $\cup\cup\langle x,y\rangle = x \cup y$.

LEMMA 1.59. $\cap\{x\} = x$.

LEMMA 1.60. $\cap\{x,y\} = x \cap y$.

LEMMA 1.61. $\cap\langle x,y\rangle = \{x\}$.

LEMMA 1.62. $\cap\cap\langle x,y\rangle = x$.

LEMMA 1.63. $x \subseteq y \wedge (\exists z)(z \in x) \rightarrow \cap y \subseteq \cap x$.

LEMMA 1.64. $x \in y \rightarrow \cap y \subseteq x$.

LEMMA 1.65. $x \in y \wedge x \subseteq z \rightarrow \cap y \subseteq z$.

LEMMA 1.66. $x \in y \wedge x \cap z = \emptyset \rightarrow (\cap y) \cap z = \emptyset$.

LEMMA 1.67. $x \neq \emptyset \wedge y \neq \emptyset \rightarrow \cap(x \cup y) = (\cap x) \cap (\cap y)$.

LEMMA 1.68. $x \neq \emptyset \rightarrow \cap x \subseteq \cup x$.

LEMMA 1.69. $\cup\cap\langle x,y\rangle = x$.

LEMMA 1.70. $\cap\cup\langle x,y\rangle = x \cap y$.

LEMMA 1.71. $x \cap \cup y = \cup\{x \cap z\text{: } z \in y,\ x \text{ fixed}\}$.

LEMMA 1.72. $y \neq \emptyset \rightarrow x \cup \cap y = \cap\{x \cup z\text{: } z \in y,\ x \text{ fixed}\}$.

LEMMA 1.73. $y \in \wp x \leftrightarrow y \subseteq x$, $x \in \wp x$, $\emptyset \in \wp x$.

LEMMA 1.74. $\wp\emptyset = \{\emptyset\}$.

LEMMA 1.75. $\wp\wp\emptyset = \{\emptyset,\{\emptyset\}\}$.

LEMMA 1.76. $x \subseteq y \leftrightarrow \wp x \subseteq \wp y$.

LEMMA 1.77. $\wp x \cup \wp y \subseteq \wp(x \cup y)$.

LEMMA 1.78. $\wp(x \cap y) = \wp x \cap \wp y$.

LEMMA 1.79. $\wp(x\backslash y) \subseteq (\wp x\backslash\wp y) \cup \{\emptyset\}$.

LEMMA 1.80. $x \in A \times B \leftrightarrow \exists y \in A\ \exists z \in B\ x = \langle y,z \rangle$.

LEMMA 1.81. $\langle x,y \rangle \in A \times B \leftrightarrow x \in A \wedge y \in B$.

LEMMA 1.82. $x \times x = \emptyset \leftrightarrow x = \emptyset \vee y = \emptyset$.

LEMMA 1.83. $x \times y = y \times x \leftrightarrow x = \emptyset \vee y = \emptyset \vee x = y$.

LEMMA 1.84. $x \neq \emptyset \wedge x \times y \subseteq x \times z \rightarrow y \subseteq z$.

LEMMA 1.85. $y \subseteq z \rightarrow x \times y \subseteq x \times z$.

LEMMA 1.86. $x \times (y \cap z) = (x \times y) \cap (x \times z)$.

LEMMA 1.87. $x \times (y \cup z) = (x \times y) \cup (x \times z)$.

LEMMA 1.88. $x \times (y \backslash z) = (x \times y) \backslash (x \times z)$.

LEMMA 1.89. $x \notin x$.

LEMMA 1.90. $\neg(x \in y \wedge y \in x)$.

LEMMA 1.91. $x \subseteq x \times x \leftrightarrow x = \emptyset$.

10. SOME VALID PROOFLESS TEXT 2.

The material in this section are adapted From [1], chapter 3.

DEFINITION 2.1. 1-ary relation BR. $BR[x] \leftrightarrow (\forall y \in A)(\exists z,w)(y = \langle z,w \rangle)$.

DEFINITION 2.2. 1-ary relation TR. $TR[x] \leftrightarrow (\forall y \in A)(\exists z,w,u)(y = \langle z,w,u \rangle)$.

DEFINITION 2.3. 1-ary function Dom. If BR[R] then $Dom(R) \cong \{x: (\exists y)(x\ \mathit{R}\ y)\}$. Otherwise $Dom(R)\uparrow$.

DEFINITION 2.4. 1-ary function Rng. If BR[R] then $Rng(R) \cong \{y: (\exists x)(x\ \mathit{R}\ y)\}$. Otherwise $Rng(R)\uparrow$.

DEFINITION 2.5. 1-ary function Fld. $Fld(R) \cong Dom(R) \cup Rng(R)$.

LEMMA 2.1. BR[∅].

LEMMA 2.2. BR[R] ∧ S ⊆ R → BR[S].

LEMMA 2.3. Let BR[R], BR[S]. Then BR[R ∪ S], BR[R ∩ S], BR[R\S].

LEMMA 2.4. Let BR[R]. Then Dom(R)↓, Rng(R)↓, Fld(R)↓.

LEMMA 2.5. Let BR[R], BR[S]. Then Dom(A ∩ B) ⊆ Dom(A) ∩ Dom(B).

LEMMA 2.6. Let BR[R], BR[S]. Then Dom(R)\Dom(S) ⊆ Dom(R\S).

LEMMA 2.7. Let BR[R], BR[S]. Then Rng(R ∪ S) = Rng(R) ∪ Rng(S).

LEMMA 2.8. Let BR[R], BR[S]. Then Rng(R ∩ S) = Rng(R) ∩ Rng(S).

LEMMA 2.9. Let BR[R], BR[S]. Then Rng(R)\Rng(S) ⊆ Rng(R\S).

DEFINITION 2.6. 1-ary function Cnv. If BR[R] then Cnv(R) ≅ {<x,y>: y *R* x}. Otherwise Cnv(R)↑.

LEMMA 2.10. BR[R] ↔ BR[Cnv(R)].

LEMMA 2.11. Let BR[R]. Then x *R* y ↔ y *Cnv(R)* x.

LEMMA 2.12. Let BR[R]. Then Cnv(Cnv(R)) = R.

LEMMA 2.13. Assume BR[R], BR[S]. Then Cnv(R ∩ S) = Cnv(R) ∩ Cnv(S), Cnv(R ∪ S) = Cnv(R) ∪ Cnv(S), Cnv(R\S) = Cnv(R)\Cnv(S).

DEFINITION 2.7. Infix function °. If BR[R], BR[S] then R°S ≅ {y: (∃x,z)(x *R* y ∧ y *S* z)}. Otherwise R°S↑.

LEMMA 2.14. Let BR[R], BR[S]. Then x *R°S* y ↔ (∃z)(y *R* z ∧ z *S* x).

LEMMA 2.15. Let BR[R], BR[S]. Then BR(R°S).

LEMMA 2.16. Let BR[R]. Then $\emptyset \circ R = R \circ \emptyset = \emptyset$.

LEMMA 2.17. Let BR[R], BR[S]. Then $Dom(R \circ S) \subseteq Dom(S)$.

LEMMA 2.18. Let BR[A], BR[B], BR[C], BR[D]. Then $A \subseteq B \wedge C \subseteq D \rightarrow A \circ B \subseteq B \circ D$.

LEMMA 2.19. Let BR[A], BR[B], BR[C]. Then $A \circ (B \cup C) = (A \circ B) \cup (A \circ C)$.

LEMMA 2.20. Let BR[A], BR[B], BR[C]. Then $A \circ (B \cap C) \subseteq (A \circ B) \cap (A \circ C)$.

LEMMA 2.21. Let BR[A], BR[B], BR[C]. Then $(A \circ B) \backslash (A \circ C) \subseteq A \circ (B \backslash C)$.

LEMMA 2.22. Let BR[A], BR[B]. Then $Cnv(A \circ B) = B \circ A$.

LEMMA 2.23. Let BR[A], BR[B], BR[C]. Then $(A \circ B) \circ C = A \circ (B \circ C)$.

DEFINITION 2.8. Infix function /*. If BR(R), BR(S) then R/*S $\cong$ {<x,y>: $(\exists z)(x$ ***R*** $z \wedge z$ ***S*** $y)$}. Otherwise R/*S$\uparrow$.

() We use /* instead of / because we use / later in developing the rationals, and we are avoIding using some features of overloading (see the discussion at the beginning of section 10). We originally used // instead of /*, but now this is not a function string. We changed the setup to support unique parsing. ()

LEMMA 2.24. Let BR(R), BR(S). Then BR(R/*S).

LEMMA 2.25. Let BR(R), BR(S). Then x ***R/*S*** y $\leftrightarrow$ $(\exists z)(x$ ***R*** $z \wedge z$ ***S*** $y)$.

LEMMA 2.26. Let BR(R), BR(S). Then Dom(R/*S) $\subseteq$ Dom(A).

LEMMA 2.27. Let BR(A), BR(B), BR(C). Then $A \subseteq B \wedge C \subseteq D \rightarrow$ A/*C $\subseteq$ B/*D.

LEMMA 2.28. Let BR(A), BR(B), BR(C). Then A/*(B $\cup$ C) = (A/*B) $\cup$ (A/*C).

LEMMA 2.29. Let BR(A), BR(B), BR(C). Then $A/*(B \cap C) \subseteq (A/*B) \cap (A/*C)$.

LEMMA 2.30. Let BR(A), BR(B), BR(C). Then $(A/*B)\backslash(A/*C) \subseteq A/*(B\backslash C)$.

LEMMA 2.31. Let BR(R), BR(S). Then Cnv(A/*B) = Cnv(A)/*Cnv(B).

LEMMA 2.32. Let BR(A), BR(B), BR(C). Then (A/*B)/*C = A/*(B/*C).

DEFINITION 2.9. Infix function |. If BR[R] then $R|A \cong R \cap (A \times Rng(A))$. Otherwise $R|A\uparrow$.

DEFINITION 2.10. Infix function ″. If BR[R] then $R''A \cong Rng(R|A)$. Otherwise $R''A\uparrow$.

LEMMA 2.33. Let BR[R]. Then $x\ \mathit{R|A}\ y \leftrightarrow x\ \mathit{R}\ y \wedge x \in A$.

LEMMA 2.34. Let BR[R]. Then $A \subseteq B \rightarrow R|A \subseteq R|B$.

LEMMA 2.35. Let BR[R]. Then $R|(A \cap B) = (R|A) \cap (R|B)$.

LEMMA 2.36. Let BR[R]. Then $R|(A \cup B) = (R|A) \cup (R|B)$.

LEMMA 2.37. Let BR[R]. Then $R|(A\backslash B) = (R|A)\backslash(R|B)$.

LEMMA 2.38. Let BR[R], BR[S]. Then (R/*S)|A = (R|A)/*S.

LEMMA 2.39. Let BR[R]. Then $y \in R''A \leftrightarrow (\exists x)(x\ \mathit{R}\ y \wedge x \in A)$.

LEMMA 2.40. Let BR[R]. Then $R''(A \cup B) = R''A \cup R''B$.

LEMMA 2.41. Let BR[R]. Then $R''(A \cap B) \subseteq R''A \cap R''B$.

LEMMA 2.42. $(\exists R)(BR[R] \wedge \neg R''(A \cap B) \subseteq R''A \cap R''B)$.

LEMMA 2.43. Let BR[R]. Then $(R''A)\backslash(R''B) \subseteq R''(A\backslash B)$.

LEMMA 2.44. Let BR[R]. Then $A \subseteq B \rightarrow R''A \subseteq R''B$.

LEMMA 2.45. Let BR[R]. Then $R''A = \emptyset \leftrightarrow Dom(R) \cap A = \emptyset$.

LEMMA 2.46. Let BR[R]. Then Dom(R) ∩ A ⊆ Cnv(R″(R″A)).

LEMMA 2.47. Let BR[R]. Then (R″A) ∩ B ⊆ R″(A ∩ Cnv(R)″B).

LEMMA 2.48. (∃A,B)(BR[A] ∧ BR[B] ∧ ¬Dom(A) ∩ Dom(B) ⊆ Dom(A ∩ B)).

LEMMA 2.49. Let BR[A], BR[B]. Then Cnv(A × B) = Cnv(B × A).

DEFINITION 2.11. 2-ary relation RFX. RFX[R,A] ↔ BR[R] ∧ (∀x ∈ A)(x *R* x).

DEFINTION 2.12. 2-ary relation IRFX. IRFX[R,A] ↔ BR[R] ∧ (∀x ∈ A)(¬x *R* x).

DEFINITION 2.13. 2-ary relation SYM. SYM[R,A] ↔ BR[R] ∧ (∀x,y ∈ A)(x *R* y ↔ y *R* x).

DEFINITION 2.14. 2-ary relation ASYM. ASYM[R,A] ↔ BR[R] ∧ (∀x,y ∈ A)(x *R* y → ¬y *R* x).

DEFINTION 2.15. 2-ary relation ANSYM. ANSYM[R,A] ↔ BR[R] ∧ (∀x,y ∈ A)(x *R* y ∧ y *R* x → x = y).

DEFINITION 2.16. 2-ary relation TRAN. TRAN[R,A] ↔ BR[R} ∧ (∀x,y,z ∈ A)(x *R* y ∧ y *R* z → x *R* z).

DEFINITION 2.17. 2-ary relation CONN. CONN[R,A] ↔ BR[R] ∧ (∀x,y ∈ A)(x ≠ y → x *R* y ∨ y *R* x).

DEFINITION 2.18. 2-ary relation SCONN. SCONN[R,A] ↔ BR[R] ∧ (∀x,y ∈ A)(x *R* y ∨ y *R* x).

DEFINITION 2.19. 1-ary relation RFX. RFX[R] ↔ BR[R] ∧ RFX[R,Fld(R)].

DEFINTION 2.20. 1-ary relation IRFX. IRFX[R] ↔ BR[R] ∧ IRFX[R,Fld(R)].

DEFINITION 2.21. 1-ary relation SYM. SYM[R] ↔ BR[R] ∧ SYM[R,Dom(R)].

DEFINITION 2.22. 1-ary relation ASYM. ASYM[R] ↔ BR[R] ∧ ASYM[R,Dom(R)].

DEFINITION 2.23. 1-ary relation ANSYM. ANSYM[R] ↔ BR[R] ∧ ANSYM[R,Dom(R)].

DEFINITION 2.24. 1-ary relation TRAN. TRAN[R] ↔ BR[R] ∧ TRAN[R,Dom(R)].

DEFINITION 2.25. 1-ary relation CONN. CONN[R] ↔ BR[R] ∧ CONN[R,Dom(R)].

DEFINITION 2.26. 1-ary relation SCONN. SCONN[R] ↔ BR[R] ∧ SCONN[R,Dom(R)].

DEFINITION 2.27. 1-ary function Id. Id(x) ≅ {<y,y>: y ∈ x}.

LEMMA 2.50. x ***Id(A)*** x ↔ x ∈ A.

LEMMA 2.51. Dom(Id(A)) = A.

LEMMA 2.52. Id(A)/*Id(A) = Id(A).

LEMMA 2.53. BR[R} ↔ Id(Dom(R))/*R = R.

LEMMA 2.54. RFX[R] ↔ Id(Fld(R)) ⊆ R.

LEMMA 2.55. IRFX[R] ↔ R ∩ Id(Fld(R)) = ∅.

LEMMA 2.56. SYMM[R] ↔ Cnv(Cnv(R)) = Cnv(R).

LEMMA 2.57. ASYM[R] ↔ R ∩ Cnv(R) = ∅.

LEMMA 2.58. ANSYM[R] ↔ R ∩ Cnv(R) ⊆ Id(Dom(R)).

LEMMA 2.59. TRAN[R] ↔ R/*R ⊆ R.

LEMMA 2.60. CONN[R] ↔ (Fld(R) × Fld(R))\Id(Fld(R)) ⊆ R ∪ Cnv(R).

LEMMA 2.61. SCONN[R] ↔ Fld(R) × Fld(R) ⊆ R ∪ Cnv(R).

DEFINITION 2.28. 2-ary relation QORD. QORD[R,A] ↔ RFX[R,A] ∧ TRAN[R,A].

DEFINITION 2.29. 2-ary relation PORD. PORD[R,A] ↔ RFX[R,A] ∧ ANSYM[A,R] ∧ TRAN[R,A].

DEFINITION 2.30. 2-ary relation SORD. SORD[R,A] ↔ ANSYM[R,A] ∧ TRAN[R,A] ∧ SCONN[R,A].

DEFINITION 2.31. 2-ary relation SPORD. SPORD[R,A] ↔ ASYM[R,A] ∧ TRAN[R,A].

DEFINITION 2.32. 2-ary relation SSORD. SSORD[R,A] ↔ ASYM[R,A] ∧ TRAN[R,A] ∧ CONN[R,A].

DEFINITION 2.33. 1-ary relation QORD. QORD[R] ↔ QORD[R,Fld(R)].

DEFINITION 2.34. 1-ary relation PORD. PORD[R] ↔ PORD[R,Fld(R)].

DEFINITION 2.35. 1-ary relation SORD. SORD[R] ↔ SORD[R,Fld(R)].

DEFINITION 2.36. 1-ary relation SPORD. SPORD[R] ↔ SPORD[R,Fld(R)].

DEFINITION 2.37. 1-ary relation SSORD. SSORD[R] ↔ SSORD[R,Fld(R)].

LEMMA 2.62. PORD[R] → QORD[R].

LEMMA 2.63. SORD[R] → PORD[R].

LEMMA 2.64. SORD[R] → SORD[Cnv(R)].

LEMMA 2.65. QORD[R] ∧ QORD[S] → QORD[R ∩ S].

LEMMA 2.66. QORD[R] ∧ QORD[S] ∧ Fld(R) ∩ Fld(S) = ∅ → QORD[R ∪ S].

LEMMA 2.67. PORD[R] → SPORD[R\Id(Fld(R))].

LEMMA 2.68. SPORD[R] → R ∪ PORD[Id(Fld(R))].

LEMMA 2.69. R ⊆ S ⊆ A × A ∧ SSORD[R,A] ∧ SSORD[S,A] → R = S.

DEFINITION 2.38. 3-ary relation MELT. MELT[x,R,A] ↔ BR[R] ∧ x ∈ A ∧ (∀y ∈ A)(¬y *R* x).

DEFINITION 2.39. 3-ary relation FELT. FELT[x,R,A] ↔ BR[R] ∧ x ∈ A ∧ (∀y ∈ A)(x ≠ y → x *R* y).

DEFINITION 2.40. 2-ary relation WO. WO[R,A] ↔ CONN[R,A] ∧ (∀B ⊆ A)(B ≠ ∅ → (∃x)(MELT[x,R,B])).

LEMMA 2.70. WO[R,A] → ASYM[R,A] ∧ TRAN[R,A].

LEMMA 2.71. WO[R,A] ↔ ASYM[R,A] ∧ CONN[R,A] ∧ (∀B ⊆ A)(B ≠ ∅ → (∃x)(FELT(x,R,A])).

LEMMA 2.72. WO[R,A] ∧ A ≠ ∅ → (∃!x)(FELT[x,R,A]).

LEMMA 2.73. WO[R,A] ∧ B ⊆ A → WO[R,B].

DEFINITION 2.41. 3-ary relation ISUC. ISUC[y,R,x] ↔ B[R] ∧ x *R* y ∧ (∀z)(z *R* x → z = y ∨ y *R* z).

DEFINITION 2.42. 3-ary relation LELT. LELT[x,R,A] ↔ BR[R] ∧ x ∈ A ∧ (∀y ∈ A)(x ≠ y → y *R* x).

LEMMA 2.74. Let BR[R]. Then LELT[x,R,A] ↔ FELT[x,Cnv(R),A].

LEMMA 2.75. WO[R,A] ∧ Fld(R) ⊆ A ∧ x ∈ A ∧ ¬LELT[x,R,A] → (∃!y)(ISUC[y,R,x]).

DEFINITION 2.43. 3-ary relation SECT. SECT[B,R,A] ↔ BR[R] ∧ B ⊆ A ∧ A ∩ Cnv(R)”B ⊆ B.

DEFINITION 2.44. 3-ary function Seg. If BR[R] then Seg(A,R,x) ≅ {y ∈ A: y *R* x}. Otherwise Seg(R)↑.

LEMMA 2.76. x ∈ A ∧ TRAN[R,A] → SECT[Seg(A,R,x),R,A].

LEMMA 2.77. WO[R,A] → (SECT[B,R,A] ∧ B ≠ A ↔ (∃x ∈ A)(B = Seg(A,R,x))).

LEMMA 2.78. ASYM[R] → IRFX[R].

LEMMA 2.79. ASYM[R] → ANSYM[R].

LEMMA 2.80. SYM[Id(A)] ∧ ANSYM[Id(A)].

LEMMA 2.81. SYM[R] ∧ ANSYM[R] → (∃A)(R = Id(A)).

LEMMA 2.82. SYM[R] ∧ TRAN[R] → RFX[R].

LEMMA 2.83. SCONN[R] → CONN[R].

LEMMA 2.84. RFX[R] → Dom(R) = Dom(Cnv(R)).

LEMMA 2.85. BR[R] → (SYM[R] ↔ R = Cnv(R)).

LEMMA 2.86. RFX[R] → RFX[Cnv(R)].

LEMMA 2.87. RFX[R] ∧ RFX[S] → RFX[R ∪ S].

LEMMA 2.88. IRFX[R] → IRFX[Cnv(R)].

LEMMA 2.89. IRFX[R] ∧ IRFX[S] → IRFX[R ∩ S] ∧ IRFX[R ∪ S] ∧ IRFX[R\S].

LEMMA 2.90. SYM[R] → SYM[Cnv(R)].

LEMMA 2.91. SYM[R] ∧ SYM[S] → SYM[R ∩ S] ∧ SYM[R ∪ S] ∧ SYM[R\S].

LEMMA 2.92. ASYM[R] → ASYM[Cnv(R)] ∧ ASYM[R ∩ S] ∧ ASYM[R\X].

LEMMA 2.93. ANSYM[R] → ANSYM[Cnv(R)] ∧ ANSYM[R ∩ S] ∧ ANSYM[R\S].

LEMMA 2.94. TRAN[R] → TRAN[Cnv(R)].

LEMMA 2.95. CONN[R] → CONN[Cnv(R)].

LEMMA 2.96. SCONN[R] → SCONN[Cnv(R)].

LEMMA 2.97. BR[R] → RFX[R ∪ Id(Fld(R))].

LEMMA 2.98. BR[R] → IRFX[R\Id(Fld(R))].

LEMMA 2.99. ASYM[R] → ANSYM[R ∪ Id(Fld(R))].

LEMMA 2.100. ANSYM[R] → ASYM[R\Id(Fld(R)].

LEMMA 2.101. TRAN[R] → TRAN[R ∪ Id(Fld(R)].

LEMMA 2.102. TRAN[R] ∧ ANSYM[R] → TRAN[R\Id(Fld(R))].

DEFINITION 2.45. 3-ary relation LB. LB[x,R,A] ↔ BR[R] ∧ (∀y ∈ A)(x *R* y).

DEFINITION 2.46. 3-ary relation INF. INF[x,R,A] ↔ LB[x,R,A] ∧ (∀y)(LB[y,R,A] → y *R* x).

DEFINITION 2.47. 3-ary relation UB. UB[x,R,A] ↔ BR[R] ∧ (∀x ∈ A)(x *R* y).

DEFINITION 2.48. 3-ary relation SUP. SUP[x,R,A] ↔ UB[y,R,A] ∧ (∀x)(UB[x,R,A] → y *R* x).

DEFINITION 2.49. 2-ary relation LAT. LAT[A,R] ↔ PORD[R,A] ∧ (∀x,y ∈ A)(∃z,w ∈ {x,y})(INF[z,R,A) ∧ SUP[w,R,A]).

LEMMA 2.103. LAT[A,R] → LAT[A,Cnv(R)].

LEMMA 2.104. SORD[R,A] → LAT[A,R].

LEMMA 2.105. Let LAT[A,R], LAT[B,S]. Then LAT[A × B, (!T)(∀x,u ∈ A)(∀y,v ∈ B)(<x,y> *T* <u,v> ↔ x *R* u ∧ y *S* v)].

DEFINITION 2.50. 1-ary relation EQUIV. EQUIV[R] ↔ RFX[R] ∧ SYM[R] ∧ TRAN[R].

DEFINITION 2.51. 2-ary relation EQUIV. EQUIV[R,A] ↔ EQUIV[R] ∧ Fld(R) = A.

LEMMA 2.106. EQUIV[R] ↔ R/*Cnv(R) = R.

LEMMA 2.107. QORD[R] → EQUIV[R ∩ Cnv(R)].

DEFINITION 2.52. 2-ary function Coset. If EQUIV[R], x ∈ Fld(R) then Coset(x,R) = {y: x *R* y}. Otherwise Coset(x,R)↑.

LEMMA 2.108. y ∈ Coset(x,R) ↔ x *R* y.

LEMMA 2.109. EQUIV[R] → Coset(x,R) = Coset(y,R) ∨ Coset(x,R) ∩ Coset(y,R) = ∅.

DEFINITION 2.53. 2-ary relation PART. PART[W,A] ↔ ∪W = A ∧ (∀B,C ∈ W)(B ≠ C → B ∩ C = ∅) ∧ (∀B ∈ W)(B ≠ ∅).

DEFINITION 2.54. 1-ary relation PART. PART[W] ↔ (∃A)(PART[W,A]).

LEMMA 2.110. A ≠ ∅ → PART[{A},A].

DEFINITION 2.55. 2-ary relation FINER. Let PART[V], PART[W]. Then FINER[V,W] ↔ V ≠ W ∧ (∀A ∈ V)(∃B ∈ W)(A ⊆ B).

LEMMA 2.111. (∀A)(∃W:PART[W,A])(∀V:PART[V,A])(FINER[W,V]).

DEFINITION 2.56. 1-ary function Part. Let EQUIV[R]. Then Part(R) = {Coset(x,R): x ∈ Fld(R)}.

LEMMA 2.112. EQUIV[R,A] → PART[Part(R),A]. EQUIV[R] → PART[Part(R)].

LEMMA 2.113. Let EQUIV[R], EQUIV[S]. Then R ⊂ S ↔ FINER[Part(R),Part(S)].

DEFINITION 2.57. 1-ary function Reln. Let PART[W]. Then Reln(W) = {<x,y>: (∃B ∈ W)(x ∈ B ∧ y ∈ B)}.

LEMMA 2.114. Let PART[W]. x ***Reln(W)*** y ↔ (∃B ∈ W)(x ∈ B ∧ y ∈ B).

LEMMA 2.115. PART[W,A] → EQUIV[Reln(W),A].

LEMMA 2.116. PART[W,A] ∧ EQUIV[R,A] → (W = Part(R) ↔ Reln(W) = R).

LEMMA 2.117. Let EQUIV[R], EQUIV[S]. Then Coset(x,R ∩ S) = Coset(x,R) ∩ Coset(x,S).

LEMMA 2.118. Let EQUIV[R], EQUIV[S]. Then Coset(x,R ∪ S) = Coset(x,R) ∪ Coset(x,S).

DEFINITION 2.58. 1-ary relation FCN. FCN[f] ↔ f = {<x,y>: f(x) = y}.

LEMMA 2.119. FCN[f] ↔ BR[f] ∧ (∀x ∈ Dom(f))(∃!y)(x *f* y).

LEMMA 2.120. x *f* y ↔ f(x) = y.

LEMMA 2.121. FCN[f] ∧ FCN[g] → FCN[f ∩ g] ∧ FCN[f°g].

LEMMA 2.122. FCN[f] ∧ FCN[g] ∧ x ∈ Dom(f°g) → (f°g)(x) = f(g(x)).

LEMMA 2.123. Let FCN[f], FCN[g]. Then (f°g)|A = f°(g|A).

LEMMA 2.124. Let FCN[f]. Then Cnv(f)″(A ∩ B) = Cnv(f)″A ∩ Cnv(f)″B, (Cnv(f)″A)\(Cnv(f)″B) = Cnv(f)″(A\B).

LEMMA 2.125. Let FCN[f]. Then Rng(f) ∩ B = f″(Cnv(f)″B).

LEMMA 2.126. Let FCN[f], A ∩ B = ∅. Then Cnv(f)″A ∩ Cnv(f)″B = ∅.

DEFINITION 2.59. 1-ary relation MONO. MONO[f] ↔ FCN[f] ∧ FCN[Cnv(f)].

LEMMA 2.127. Let FCN[f]. Then MONO[f] ↔ (∀x,y ∈ Dom(f))(f(x) = f(y) → x = y).

LEMMA 2.128. Let MONO[f]. Then f(x) = y ↔ Cnv(f)(y) = x.

LEMMA 2.129. Let MONO[f]. Then x ∈ Dom(f) → Cnv(f)(f(x)) = x.

LEMMA 2.130. Let MONO[f]. Then y ∈ Rng(f) → f(Cnv(f)(y)) = y.

LEMMA 2.131. Let MONO[f], MONO[g]. Then MONO[f ∩ g], MONO[f ° g].

LEMMA 2.132. Let MONO[f], MONO[g], Dom(f) ∩ Dom(g) = Rng(f) ∩ Rng(g) = ∅. Then MONO[f ∪ g].

DEFINITION 2.60. 3-ary relation FCN. FCN[f,A,B] ↔ FCN[f] ∧ Dom(f) = A ∧ Rng(f) ⊆ B.

DEFINITION 2.61. 3-ary relation SURJ. SURJ[f,A,B] ↔ FCN[f] ∧ Dom(f) = A ∧ Rng(f) = B.

DEFINITION 2.62. 3-ary relation MONO. MONO[f,A,B] ↔ MONO[f] ∧ Dom(f) = A ∧ Rng(f) ⊆ B.

DEFINITION 2.63. 3-ary relation BIJ. BIJ[f,A,B] ↔ MONO[f] ∧ Dom(f) = A ∧ Rng(f) = B.

DEFINITION 2.64. 2-ary function Maps. Maps(A,B) ≅ {f: FCN[f,A,B]}.

LEMMA 2.133. Maps(A,B)↓.

LEMMA 2.134. f ∈ Maps(A,B) ↔ FCN[f,A,B].

LEMMA 2.135. Maps(∅,A) = {∅}.

LEMMA 2.136. A ≠ ∅ → Maps(A,∅) = ∅.

LEMMA 2.137. Maps(A,B) = ∅ ↔ A ≠ ∅ ∧ B = ∅.

LEMMA 2.138. Maps({x},A) = {{<x,y>: y ∈ A, x fixed}}.

LEMMA 2.139. A ⊆ B → Maps(C,A) ⊆ Maps(C,B).

LEMMA 2.140. (λA)(A)↑.

11. SOME VALID PROOFLESS TEXT 3.

The material in this section is adapted From [1], chapter 4. We omit the sections on Cardinal Numbers and Finite Cardinals, as they rely on class theory rather than set theory. We come back to this material in section, after the theory of Von Neumann ordinals is treated in [1], chapter 5.

DEFINITION 3.1. Infix relation $\approx_c$. $A \approx_c B \leftrightarrow (\exists f)(BIJ[f,A,B])$.

() The subscript c stands for cardinality. ()

LEMMA 3.1. $A \approx_c A$.

LEMMA 3.2. $A \approx_c B \rightarrow B \approx_c A$.

LEMMA 3.3. $A \approx_c B \wedge B \approx_c C \rightarrow A \approx_c C$.

LEMMA 3.4. Let $A \approx_c B$, $C \approx_c D$, $A \cap C = B \cap D = \emptyset$. Then $A \cup C \approx_c B \cup D$.

LEMMA 3.5. $A \approx_c B \wedge C \approx_c D \rightarrow A \times C \approx_c B \times D$.

LEMMA 3.6. $A \times B \approx_c B \times A$.

LEMMA 3.7. $A \times (B \times C) \approx_c (A \times B) \times C$.

LEMMA 3.8. $A \times \{x\} \approx_c \{x\} \times A \approx_c A$.

LEMMA 3.9. $(\exists C,D)(A \approx_c C \wedge B \approx_c D \wedge C \cap D =_c \emptyset)$.

LEMMA 3.10. $A \approx_c B \wedge C \approx_c D \rightarrow Maps(A,C) \approx_c Maps(B,D)$.

LEMMA 3.11. $A \cap B = \emptyset \rightarrow Maps(A \cup B,C) \approx_c Maps(A,C) \times Maps(B,C)$.

LEMMA 3.12. $Maps(A,B \times C) \approx_c Maps(A,,B) \times Maps(A,C)$.

LEMMA 3.13. $Maps(A,Maps(B,C)) \approx_c Maps(A \times B,C)$.

LEMMA 3.14. Let $x \neq y$. Then $Maps(A,\{x,y\}) \approx_c \wp A$.

DEFINITION 3.2. Infix relation $\leq_c$. $x \leq_c y \leftrightarrow (\exists z \subseteq y)(x \approx_c z)$.

LEMMA 3.15. $x \leq_c y \leftrightarrow (\exists f)(MONO[f,x,y])$.

LEMMA 3.16. $A \approx_c B \rightarrow A \leq_c B$.

LEMMA 3.17. $A \subseteq B \rightarrow A \leq_c B$.

LEMMA 3.18. $A \leq_c B \wedge B \leq_c C \rightarrow A \leq_c C$.

THEOREM 3.19. $A \leq_c B \wedge B \leq_c A \rightarrow A \approx_c B$.

LEMMA 3.20. Let $A \leq_c B$, $C \leq_c D$. Then $B \cap D = \emptyset \rightarrow A \cup C \leq_c B \cup D$, $A \times C \leq_c B \times D$.

LEMMA 3.21. Let $A \leq_c B$, $C \leq_c D$, $\neg(A = B = C = \emptyset \wedge B \neq \emptyset)$. Then $\text{Maps}(A,C) \leq_c \text{Maps}(B,D)$.

LEMMA 3.22. $A \leq_c A \cup B$.

DEFINITION 3.3. Infix relation $<_c$. $A <_c B \leftrightarrow A \leq_c B \wedge \neg B \leq_c A$.

LEMMA 3.23. $\neg A <_c A$, $A <_c B \rightarrow \neg B <_c A$, $A <_c B \wedge B <_c C \rightarrow A <_c C$.

LEMMA 3.24. $A \leq_c B \rightarrow \neg B <_c A$, $A \leq_c B \wedge B <_c C \rightarrow A <_c C$, $A <_c B \wedge B \leq_c C \rightarrow A <_c C$, $A \leq_c B \leftrightarrow A \approx_c B \vee A <_c B$.

THEOREM 3.25. $A <_c \wp A$.

DEFINITION 3.4. 2-ary relation MNEL. $\text{MNEL}[x,A] \leftrightarrow x \in A \wedge (\forall y \in A)(y \notin x)$.

DEFINITION 3.5. 2-ary relation MXEL. $\text{MXEL}[x,A] \leftrightarrow x \in A \wedge (\forall y \in A)(x \notin y)$.

DEFINITION 3.6. 1-ary relation FIN. $\text{FIN}[x] \leftrightarrow (\forall A \neq \emptyset)(A \subseteq \wp x \rightarrow (\exists y \in A)(\text{MNEL}[y,A])))$.

LEMMA 3.26. $\text{FIN}[\emptyset]$.

LEMMA 3.27. $\text{FIN}[\{x\}]$.

LEMMA 3.28. $\text{FIN}[A] \wedge B \subseteq A \rightarrow \text{FIN}[B]$.

LEMMA 3.29. $\text{FIN}[A] \rightarrow \text{FIN}[A \cap B] \wedge \text{FIN}[A \backslash B]$.

LEMMA 3.30. $\text{FIN}[A] \wedge \text{FIN}[B] \rightarrow \text{FIN}[A \cup B]$.

LEMMA 3.31. $\text{FIN}[A] \rightarrow \text{FIN}[A \cup \{x\}]$.

LEMMA 3.32. FIN[x] $\leftrightarrow$ $(\forall A \neq \emptyset)(A \subseteq \wp x \rightarrow (\exists y \in A)(MXEL[y,A])))$.

LEMMA 3.33. Let FIN[x], $A \subseteq \wp x$, $\emptyset \in A$, $(\forall y \in A)(\forall z \in x)(y \cup \{z\} \in A)$. Then $x \in A$.

LEMMA 3.34. FIN[x] $\leftrightarrow$ $(\forall A \subseteq \wp x)(\emptyset \in A \wedge (\forall y \in A)(\forall z \in x)(y \cup \{z\} \in A) \rightarrow x \in A)$.

LEMMA 3.35. FCN[f] $\wedge$ FIN[Dom(f)] $\rightarrow$ FIN[Rng(f)].

LEMMA 3.36. FIN[x] $\rightarrow$ FIN[$\wp$x].

LEMMA 3.37. FIN[x] $\wedge$ $(\forall y \in x)$(FIN[y]) $\rightarrow$ FIN[$\cup$x].

LEMMA 3.38. FIN[$\wp$A] $\rightarrow$ FIN[A].

LEMMA 3.39. FIN[$\cup$A] $\rightarrow$ FIN[A] $\wedge$ $(\forall y \in x)$(FIN[y]).

LEMMA 3.40. FIN[A] $\wedge$ $A \approx_c B$ $\rightarrow$ FIN[B].

LEMMA 3.41. FIN[A] $\wedge$ $B \leq_c A$ $\rightarrow$ FIN[B].

LEMMA 3.42. FIN[A] $\rightarrow$ $A <_c B \vee A \approx_c B \vee B <_c A$.

LEMMA 3.43. FIN[A] $\wedge$ $\neg$FIN[B] $\rightarrow$ $A <_c B$.

DEFINITION 3.7. 1-ary relation DFIN. DFIN[x] $\leftrightarrow$ $(\forall y \subset x)(\neg x \approx_c y)$.

LEMMA 3.44. FIN[A] $\rightarrow$ DFIN[A].

LEMMA 3.45. FIN[A] $\wedge$ $B \subset A \rightarrow B <_c A$.

LEMMA 3.46. Let FIN[B], FIN[C], $A <_c B$, $B \cap C = \emptyset$. Then $A \cup C <_c B \cup C$.

LEMMA 3.47. Let FIN[A], FIN[B], FIN[C], FIN[D], $A <_c B$, $C <_c D$, $B \cap D = \emptyset$. Then $A \cup C <_c B \cup D$.

LEMMA 3.48. FIN[A] $\wedge$ $x \notin A \rightarrow A <_c A \cup \{x\}$.

LEMMA 3.49. FIN[A] $\wedge$ FIN[B] $\rightarrow$ FIN[$A \times B$].

LEMMA 3.50. FIN[A] ∧ FIN[B] → FIN[Maps(A,B)].

LEMMA 3.51. (∃C)(C ∩ B = ∅ ∧ C $\approx_c$ A).

12. SOME VALID PROOFLESS TEXT 4.

The material in this section is adapted From [1], chapter 5.

DEFINITION 4.1. 1-ary relation TRANS. TRANS(x) ↔ (∀y ∈ x)(∀z ∈ y)(z ∈ x).

DEFINITION 4.2. 1-ary relation ECONN. ECONN[x] ↔ (∀y,z ∈ x)(y ∈ x ∨ z ∈ x ∨ y = z).

DEFINITION 4.3. 1-ary relation ORD. ORD[x] ↔ TRANS(x) ∧ ECONN(x).

DEFINITION 4.4. 1-ary function Eps. Eps(x) ≅ {<y,z>: y ∈ z ∧ z,y ∈ x}.

() Eps(x) is the Epsilon relation on x. ()

LEMMA 4.1. Eps(x)↓, ORD[x] ↔ WO[Eps(x),x].

LEMMA 4.2. Let ORD[A], TRANS[B], B ⊂ A. Then B ∈ A.

LEMMA 4.3. Let ORD[A], ORD[B]. Then A ⊂ B ↔ A ∈ B.

LEMMA 4.4. ORD[A] ∧ B ∈ A → ORD[B].

LEMMA 4.5. ORD[A] ∧ ORD[B] → A ⊆ B ∨ B ⊆ A.

LEMMA 4.6. ORD[A] ∧ ORD[B] → ![A ∈ B, B ∈ A, A = B].

LEMMA 4.7. Let (∀x ∈ A)(ORD[x]). Then ORD[∪A].

DEFINITION 4.5. Infix relation $<_o$. A $<_o$ B ↔ ORD[A] ∧ ORD[B] ∧ A ∈ B.

DEFINITION 4.6. Infix relation $\leq_o$. A $\leq_o$ B ↔ ORD[A] ∧ ORD[B] ∧ (A ∈ B ∨ A = B).

DEFINITION 4.7. Infix relation $>_o$. A $>_o$ B ↔ ORD[A] ∧ ORD[B] ∧ B ∈ A.

DEFINITION 4.8. Infix relation $\geq_o$. A $\geq_o$ B ↔ ORD[A] ∧ ORD[B] ∧ (B ∈ A ∨ A = B).

LEMMA 4.8. ¬A $<_o$ A, A $<_o$ B → ¬B $<_o$ A, A $<_o$ B ∧ B $<_o$ C → A $<_o$ C.

LEMMA 4.9. Let ORD[A], ORD[B]. Then ![A $<_o$ B, A = B, B <o A].

LEMMA 4.10. Let ORD[A]. Then A = {B: ORD[B] ∧ B $<_o$ A}.

LEMMA 4.11. {x: ORD[x]}↑.

DEFINITION 4.9. 1-ary function Suc. If ORD[x] then Suc(x) ≅ {y: y $\leq_o$ x}. Otherwise Suc(x)↑.

LEMMA 4.12. ORD[x] ↔ ORD[Suc(x)].

LEMMA 4.13. ORD[x] → ∪Suc(x) = x.

LEMMA 4.14. Let ORD[x], ORD[y]. ¬(∃B:ORD[y])(x $<_o$ y $<_o$ Suc(x)).

LEMMA 4.15. A ∈ B → B ∉ Suc(A).

LEMMA 4.16. Let (∀x ∈ B)(ORD[x]). Then C ∈ B → C $\leq_o$ ∪B.

LEMMA 4.17. (∀C ∈ B)(C $\leq_o$ D) → ∪B $\leq_o$ D.

DEFINITION 4.10. 1-ary relation NAT. NAT[x] ↔ ORD[x] ∧ WO[Cnv(Eps(x)),x].

LEMMA 4.18. NAT[x] ↔ ORD[x] ∧ FIN[x].

LEMMA 4.19. NAT[∅].

LEMMA 4.20. NAT[x] → NAT[Suc(x)].

LEMMA 4.21. ¬(∃x)(NAT[x] ∧ Suc(x) = ∅).

LEMMA 4.22. Suc(x) = Suc(y) → x = y.

LEMMA 4.23. NAT[x] $\wedge$ y $\leq_{\circ}$ x $\rightarrow$ NAT[y].

DEFINITION 4.19. 0-ary function ω. ω ≅ {x: NAT[x]}.

LEMMA 4.24. NAT[ω], x ∈ ω ↔ NAT[x].

DEFINITION 4.11. 0-ary function 0. 0 ≅ ∅.

DEFINITION 4.12. 0-ary function 1. 1 ≅ {∅}.

LEMMA 4.25. NAT[0], NAT[1].

LEMMA 4.26. Let 0 ∈ A, (∀n ∈ A)(NAT[n] → Suc(n) ∈ A). Then ω ⊆ A.

LEMMA 4.27. (∃!x)(∀y)(∀z ∈ ω)(x(y,0) ≅ w(y) ∧ x(y,Suc(z)) ≅ u(y,z,x(y,z))).

DEFINITION 4.13. 0-ary function Addnat. Addnat ≅ (!x)((∀y,z ∈ ω)(x(y,0) = y ∧ x(y,Suc(z)) = Suc(x(y,z))) ∧ (∀y,z)(x(y,z)↓ ↔ y,z ∈ ω)).

LEMMA 4.28. 2-ary function $+_N$. x $+_N$ y ≡ Addnat(x,y).

LEMMA 4.29. Let x,y ∈ ω. Then $x+_N y$↓, $x+_N 1$ = Suc(x).

LEMMA 4.30. Let x,y,z ∈ ω. Then $(x+_N y)+_N 1 = x+_N (y+_N 1)$.

LEMMA 4.31. Let x,y,z ∈ ω. Then $x+_N y = y+_N x$, $(x+_N y)+_N z = x+_N (y+_N z)$.

DEFINITION 4.15. 0-ary function Mulnat. Mulnat ≅ (!x)((∀y,z ∈ ω)(x(y,0) = 0 ∧ x(y,z+1) = x(y,z)+y ∧ (∀y,z)(x(y,z)↓ ↔ y,z ∈ ω)).

LEMMA 4.32. Mulnat↓.

DEFINITION 4.16. Infix function $\bullet_N$. $x \bullet_N y$ ≅ Mulnat(x,y).

LEMMA 4.33. Let x,y ∈ ω. Then $x \bullet_N y$↓.

LEMMA 4.34. Let x,y,z ∈ ω. Then $x \bullet_N y = y \bullet_N x$, $(x \bullet_N y) \bullet_N z = x \bullet_N (y \bullet_N z)$, $x \bullet_N (y +_N z) = (x \bullet_N y) + (x \bullet_N z)$.

DEFINITION 4.17. 0-ary function Expnat. Expnat ≅ (!x)((∀y,z ∈ ω)(x(y,0) = 1 ∧ x(y,z+1) = x(y,z)•y ∧ (∀y,z)(x(y,z)↓ ↔ y,z ∈ ω)).

LEMMA 4.35. Expnat↓.

DEFINITION 4.18. Infix function $^{\wedge N}$. $x^{\wedge N}y$ ≅ Expnat(x,y).

LEMMA 4.36. Let x,y ∈ ω. Then $x^{\wedge N}y$↓.

LEMMA 4.37. Let x,y,z ∈ ω. Then $x^{\wedge N}(y+_N z) = (x^{\wedge N}y) \bullet_N (x^{\wedge N}z)$, $x^{\wedge N}(y \bullet_N z) = (x^{\wedge N}y)^{\wedge N}z$, $x^{\wedge N}0 = 1$, $x^{\wedge N}1 = x$.

LEMMA 4.38. FIN[x] ↔ (∃y ∈ ω)($x \approx_c y$) ↔ (∃!y ∈ ω)($x \approx_c y$).

LEMMA 4.39. FIN[x] ↔ (∃R)(WO[R,x] ∧ WO[Cnv(R),x]).

LEMMA 4.40. ∪ω = ω.

DEFINITION 4.19. 1-ary relation INF. INF[x] ↔ ¬FIN[x].

LEMMA 4.41. Let $x \approx_c y$. Then FIN[x] ↔ FIN[y], INF[x] ↔ INF[y].

LEMMA 4.42. Let x ⊆ y. Then INF[x] → INF[y].

LEMMA 4.43. INF[x] ↔ (∀n ∈ ω)(∃y ⊆ x)($y \approx_c n$).

LEMMA 4.44. INF[ω].

LEMMA 4.45. FIN[x] ↔ $x <_c \omega$.

DEFINITION 4.20. 1-ary relation DEN. DEN[x] ↔ $x \approx_c \omega$.

LEMMA 4.46. DEN[x] → INF[x].

DEFINITION 4.21. 1-ary relation DINF. DINF[x] ↔ ¬DFIN[x].

LEMMA 4.47. DINF[x] ↔ (∃y ⊂ x)($y \approx_c x$).

LEMMA 4.48. DINF[x] ↔ $x \approx_c x \cup \{x\}$.

LEMMA 4.49. DINF[x] → INF[x].

LEMMA 4.50. $(\exists x \subseteq y)(\mathrm{DEN}[x]) \rightarrow \mathrm{DINF}[x]$.

LEMMA 4.51. $\mathrm{DINF}[x] \leftrightarrow (\exists y \subseteq x)(\mathrm{DEN}[y])$.

LEMMA 4.52. Let DEN[x]. Then $y \subseteq x \rightarrow \mathrm{DEN}[y] \vee \mathrm{FIN}[y]$.

LEMMA 4.53. $\mathrm{FIN}[A] \wedge \mathrm{DEN}[B] \rightarrow \mathrm{DEN}[A \cup B]$.

LEMMA 4.54. $\mathrm{DEN}[A] \wedge \mathrm{DEN}[B] \rightarrow \mathrm{DEN}[A \cup B]$.

LEMMA 4.55. Let FIN[A], $A \neq \emptyset$, DEN[B]. Then $\mathrm{DEN}[A \times B]$.

LEMMA 4.56. $\omega \times \omega \approx_c \omega$.

LEMMA 4.57. Let $n \in \omega\backslash\{0\}$. Then $\mathrm{Maps}(n,\omega) \approx_c \omega$.

LEMMA 4.58. $\mathrm{DEN}[A] \wedge \mathrm{DEN}[B] \rightarrow \mathrm{DEN}[A \times B]$.

LEMMA 4.59. $\mathrm{DEN}[A] \wedge n \in \omega\backslash\{0\} \rightarrow \mathrm{DEN}[\mathrm{Maps}(n,A)]$.

13. SOME VALID PROOFLESS TEXT 5.

The material in this section is adapted From [1], chapter 6.

DEFINITION 5.1. Infix function /. Let $x,y \in \omega$, $y \neq 0$. Then $x/y \cong \langle x,y\rangle$. Otherwise, $x/y\uparrow$.

DEFINITION 5.2. 0-ary function Fr. $\mathrm{Fr} \cong \{x/y: x/y\downarrow\}$.

LEMMA 5.1. $\mathrm{Fr}\downarrow$.

DEFINITION 5.3. Infix relation $\equiv_{Fr}$. $x \equiv_{Fr} y \leftrightarrow (\exists a,b,c,d)(x = a/b \wedge y = c/d \wedge a\bullet d = b\bullet c\}$.

LEMMA 5.2. $\mathrm{EQUIV}[\{\langle x,y\rangle: x \equiv_{Fr} y,\mathrm{Fr}]$.

LEMMA 5.3. $m/n \equiv_{FR} r/s \leftrightarrow m,n,r,s \in \omega \wedge n,s \neq 0 \wedge m\bullet s = n\bullet r$.

LEMMA 5.4. $m/n \in \mathrm{Fr} \wedge p \in \omega\backslash\{0\} \rightarrow m/n \equiv_{Fr} (m\bullet_N p)/(n\bullet_N p)$.

DEFINITION 5.4. Infix relation $<_{Fr}$. $x <_{Fr} y \leftrightarrow (\exists a,b,c,d)(x = a/b \wedge y = c/d \wedge a\bullet_N d <_o b\bullet_N c\}$.

DEFINITION 5.5. Infix relation $>_{Fr}$. $x >_{Fr} y \leftrightarrow y <_{Fr} x$.

DEFINITION 5.6. Infix relation $\leq_{Fr}$. $x \leq_{FR} y \leftrightarrow x <_{Fr} y \vee x \equiv_{Fr} y$.

DEFINITION 5.7. Infix relation $\geq_{Fr}$. $x \geq_{Fr} y \leftrightarrow x >_{Fr} y \vee x \equiv_{Fr} y$.

LEMMA 5.5. SPORD[{<x,y>: $x <_{Fr} y$}], SPORD[{<x,y>: $x >_{Fr} y$}].

LEMMA 5.6. Let $x,y \in Fr$. Then $![x \equiv_{Fr} y, x <_{Fr} y, y <_{Fr} x]$.

LEMMA 5.7. $(\forall x \in Fr)(\exists y \in Fr)(x <_{Fr} y)$.

LEMMA 5.8. $(\forall x,y)(x <_{Fr} y \rightarrow (\exists z \in Fr)(x <_{FR} z <_{Fr} y))$.

LEMMA 5.9. Let $x,y,u,v \in Fr$, $x <_{FR} y$, $x \equiv_{FR} u$, $y \equiv_{Fr} v$. Then $u <_{Fr} v$.

DEFINITION 5.8. Infix function $+_{Fr}$. $x +_{Fr} y \cong (!z)(\exists a,b,d,c,e,f)(x = a/b \wedge y = c/d \wedge z = e/f \wedge a \bullet_N d + b \bullet_N c = e \wedge b \bullet_N d = f)$.

LEMMA 5.10. $(\forall x,y \in Fr)(x +_{FR} y \in Fr)$, $x +_{FR} y\downarrow \leftrightarrow x,y \in Fr$.

LEMMA 5.11. Let $x \equiv_{FR} u$, $y \equiv_{FR} v$. Then $x +_{FR} y \equiv_{FR} u +_{FR} v$.

LEMMA 5.12. Let $n,m,r \in \omega \wedge r \neq 0$. Then $m/n +_{FR} r/n \equiv_{Fr} (m+r)/n$.

LEMMA 5.13. Let $x,y,z \in Fr$. Then $x +_{Fr} y = y +_{Fr} x$, $(x +_{Fr} y) +_{FR} z = x +_{Fr} (y +_{Fr} z)$.

LEMMA 5.14. Let $z \in Fr$, $x <_{Fr} y$. Then $x +_{Fr} z <_{Fr} y +_{Fr} z$, $z +_{Fr} x <_{Fr} z +_{Fr} y$.

LEMMA 5.15. Let $x +_{Fr} z \equiv_{Fr} y +_{Fr} z \vee z +_{Fr} x \equiv_{Fr} z +_{Fr} y$. Then $x \equiv_{Fr} y$.

LEMMA 5.16. Let $x +_{Fr} z <_{Fr} y +_{Fr} z$. Then $x <_{Fr} y$.

DEFINITION 5.9. Infix function $\bullet_{Fr}$. $x \bullet_{Fr} y \cong (!t)(\exists a,b,c,d,e,f)(x = a/b \wedge y = c/d \wedge z = e/f \wedge a\bullet_N c = e \wedge b\bullet_N d = f)$.

LEMMA 5.17. $(\forall x,y \in Fr)(x \bullet_{FR} y \in Fr)$, $x \bullet_{FR} y\downarrow \leftrightarrow x,y \in Fr$.

LEMMA 5.18. Let $x \equiv_{Fr} u$, $y \equiv_{Fr} v$. Then $x \bullet_{Fr} y \equiv_{FR} u \bullet_{Fr} v$.

LEMMA 5.19. Let $x,y,z \in Fr$. Then $x \bullet_{Fr} y = y \bullet_{Fr} x$, $(x \bullet_{Fr} y) \bullet_{Fr} z = x \bullet_{Fr} (y \bullet_{Fr} z)$.

LEMMA 5.20. Let $x,y,z \in Fr$. Then $x \bullet_{Fr} (y +_{Fr} z) = (x \bullet_{Fr} y) +_{Fr} (x \bullet_{Fr} z)$.

LEMMA 5.21. Let $z \in Fr$, $x <_{Fr} y$, $\neg z \equiv_{Fr} 0/1$. Then $x \bullet_{Fr} z <_{Fr} y \bullet_{Fr} z$.

LEMMA 5.22. Let $\neg z \equiv_{Fr} 0/1$, $x \bullet_{Fr} z \equiv_{FR} y \bullet_{Fr} z$. Then $x \equiv_{Fr} y$.

LEMMA 5.23. Let $x \bullet_{Fr} z <_{Fr} y \bullet_{Fr} z$. Then $x <_{Fr} y$.

LEMMA 5.24. Let $x,y \in Fr$, $\neg y \equiv_{Fr} 0/1$. Then $(\exists z \in FR)(x \equiv_{Fr} y \bullet_{Fr} z)$.

DEFINITION 5.10. 0-ary function Nra. $Nra \cong \{Coset(x,\equiv_{Fr}): x \in Fr\}$.

() Nra is the set of all nonnegative rationals. ()

LEMMA 5.25. $Nra\downarrow$, PART[Nra,Fr].

DEFINITION 5.11. Infix relation $<_{Nra}$. $x <_{Nra} y \leftrightarrow x,y \in Nra \wedge (\exists u,v)(u \in x \wedge v \in y \wedge u <_{Fr} v)\}$.

DEFINITION 5.12. Infix relation $>_{Nra}$. $x >_{Nra} y \leftrightarrow x,y \in Nra \wedge (\exists u,v)(u \in x \wedge v \in y \wedge u >_{Fr} v)\}$.

DEFINITION 5.13. Infix relation $\leq_{Nra}$. $x \leq_{Nra} y \leftrightarrow x,y \in Nra \wedge (\exists u,v)(u \in x \wedge v \in y \wedge u \leq_{Fr} v)\}$.

DEFINITION 5.14. Infix relation $\geq_{Nra}$. $x <_{Nra} y \leftrightarrow x,y \in Nra \wedge (\exists u,v)(u \in x \wedge v \in y \wedge u \geq_{Fr} v)\}$.

LEMMA 5.26. SSORD[{<x,y>: x $<_{NQ}$ y}], SSORD[{<x,y>: x $>_{Nra}$ y}], SORD[{<x,y>: x $\leq_{Nra}$ y}], SORD[{<x,y>: x $\leq_{Nra}$ y}].

DEFINITION 5.15. Infix function $+_{NQ}$. x $+_{Nra}$ y $\cong$ (!z)(x,y,z $\in$ Nra $\wedge$ $(\exists$u,v,w)(u $\in$ x $\wedge$ v $\in$ y $\wedge$ z $\in$ w $\wedge$ u $+_{FR}$ v $\equiv_{Fr}$ w)).

LEMMA 5.27. ($\forall$x,y $\in$ Nra)(x $+_{Nra}$ y $\in$ Nra), ($\forall$x,y)(x $+_{Nra}$ y$\downarrow$ $\leftrightarrow$ x,y $\in$ Nra).

LEMMA 5.28. Let x,y,z $\in$ Nra. Then x $+_{Nra}$ y = y $+_{Nra}$ x, (x $+_{Nra}$ y) $+_{Nra}$ z = x $+_{Nra}$ (y $+_{Nra}$ z).

LEMMA 5.29. Let x,y,z $\in$ Nra. Then x = y $\leftrightarrow$ x $+_{Nra}$ z = y $+_{Nra}$ z.

LEMMA 5.30. Let x,y,z $\in$ Nra. Then x $<_{Nra}$ y $\leftrightarrow$ x $+_{Nra}$ z $<_{Nra}$ y $+_{Nra}$ z.

DEFINITION 5.16. Infix function $\bullet_{Nra}$. x $\bullet_{Nra}$ y $\cong$ (!z)(x,y,z $\in$ Nra $\wedge$ $(\exists$u,v,w)(u $\in$ x $\wedge$ v $\in$ y $\wedge$ z $\in$ w $\wedge$ u $\bullet_{Fr}$ v $\equiv_{Fr}$ w)).

LEMMA 5.31. ($\forall$x,y $\in$ Nra)(x $\bullet_{Nra}$ y $\in$ Nra), ($\forall$x,y)(x $\bullet_{Nra}$ y$\downarrow$ $\leftrightarrow$ x,y $\in$ Nra).

LEMMA 5.32. Let x,y,z $\in$ Nra. Then x $\bullet_{Nra}$ y = y $\bullet_{Nra}$ x, (x $\bullet_{Nra}$ y) $\bullet_{Nra}$ z = x $\bullet_{Nra}$ (y $\bullet_{Nra}$ z), x $\bullet_{Nra}$ (y $+_{Nra}$ z) = (x $\bullet_{Nra}$ y) $+_{Nra}$ (x $\bullet_{Nra}$ z).

LEMMA 5.33. Let x,y,z $\in$ Nra, 0/1 $\notin$ z. Then x = y $\leftrightarrow$ x $\bullet_{Nra}$ z = y $\bullet_{Nra}$ z.

LEMMA 5.34. Let x,y,z $\in$ Nra, 0/1 $\notin$ z. Then x $<_{Nra}$ y $\leftrightarrow$ x $\bullet_{Nra}$ z $<_{Nra}$ y $\bullet_{Nra}$ z.

DEFINITION 5.17. 0-ary function 0_{Nra}. 0_{Nra} $\cong$ Coset(0/1,{<x,y>: x $\equiv_{Fr}$ y}).

DEFINITION 5.18. 0-ary function 1_{Nra}. 1_{Nra} $\cong$ Coset(1/1,{<x,y>: x $\equiv_{Fr}$ y}).

LEMMA 5.35. $0_{Nra} \neq 1_{Nra}$.

LEMMA 5.36. Let x,y,z ∈ Nra. Then $x +_{Nra} 0_{Nra} = x$, $x \bullet_{Nra} 0_{Nra} = 0_{Nra}$, $x \bullet_{Nra} 1_{Nra} = x$.

LEMMA 5.37. Let x,y,z ∈ Nra, $0_{Nra} <_{Nra} x \wedge y <_{Nra} z$. Then $x \bullet_{Nra} y <_{Nra} x \bullet_{Nra} z$.

LEMMA 5.38. Let x,y ∈ Nra, $0_{Nra} <_{Nra} x$. Then $(\exists! z \in Nra)(x = y \bullet_{Nra} z$.

DEFINITION 5.19. Infix relation $\equiv_{SUB}$. $x \equiv_{SUB} y \leftrightarrow (\exists a,b,c,d)(x = <a,b> \wedge y = <c,d> \wedge a +_{Nra} d = b +_{Nra} c)$.

LEMMA 5.39. EQUIV[$\{<x,y>: x \equiv_{SUB} y\}$,Nra × Nra].

DEFINITION 5.20. Infix relation $<_{SUB}$. $x <_{SUB} y \leftrightarrow (\exists a,b,c,d)(x = <a,b> \wedge y = <c,d> \wedge a +_{Nra} d <_{Nra} b +_{Nra} c)$.

LEMMA 5.40. SSORD[$\{<x,y>: x <_{SUB} y\}$,Nra × Nra].

DEFINITION 5.21. Infix function $+_{SUB}$. $x +_{SUB} y \cong (!z)(\exists a,b,c,d,e,f)(x = <a,b> \wedge y = <c,d> \wedge z = <e,f> \wedge a +_{Nra} c +_{Nra} f = b +_{Nra} d +_{Nra} e)$.

LEMMA 5.41. $(\forall x,y \in Nra \times Nra)(x +_{SUB} y \in Nra)$, $(\forall x,y)(x +_{SUB} y\downarrow \leftrightarrow x,y \in Nra \times Nra)$.

DEFINITION 5.22. Infix function $\bullet_{SUB}$. $x \bullet_{SUB} y \cong (!z)(\exists a,b,c,d,e,f)(x = <a,b> \wedge y = <c,d> \wedge z = <e,f> \wedge (a \bullet_{Nra} c) +_{Nra} (b \bullet_{Nra} d) +_{Nra} f = (a \bullet_{Nra} d) +_{Nra} (b \bullet_{Nra} c) +_{Nra} e)$.

LEMMA 5.42. $(\forall x,y \in Nra \times Nra)(x \bullet_{SUB} y \in Nra)$, $(\forall x,y)(x \bullet_{SUB} y\downarrow \leftrightarrow x,y \in Nra \times Nra)$.

LEMMA 5.43. Let $x \equiv_{SUB} z$, $y \equiv_{SUB} w$. Then $x <_{SUB} y \leftrightarrow z <_{SUB} w$, $x +_{SUB} y \equiv_{SUB} z +_{SUB} w$, $x \bullet_{SUB} y \equiv_{SUB} z \bullet_{SUB} w$.

DEFINITION 5.23. 0-ary function Ra. Ra $\cong$ {Coset(x,$\{<x,y>: x \equiv_{SUB} y\}$): x ∈ Nra × Nra}.

DEFINITION 5.24. Infix relation $<_{Ra}$. $x <_{Ra} y \leftrightarrow (\exists z,w)(x,y \in Ra \wedge z \in x \wedge w \in y \wedge z <_{SUB} w)$.

DEFINITION 5.25. Infix function $+_{Ra}$. $x +_{Ra} y \cong (!z)(x,y,z \in Ra \wedge (\exists a,b,c)(a \in x \wedge b \in y \wedge c \in z \wedge a +_{SUB} b = c))$.

DEFINITION 5.26. Infix function $\bullet_{Ra}$. $x \bullet_{Ra} y \cong (!z)(x,y,z \in Ra \wedge (\exists a,b,c)(a \in x \wedge b \in y \wedge c \in z \wedge a \bullet_{SUB} b = c))$.

LEMMA 5.44. $(\forall x,y \in Ra)(x +_{Q} y \in Ra)$, $(\forall x,y \in Ra)(x \bullet_{Ra} y \in Ra)$, $(\forall x,y)(x +_{Ra} y\downarrow \leftrightarrow x,y \in Ra)$, $(\forall x,y)(x \bullet_{Ra} y\downarrow \leftrightarrow x,y \in Ra)$.

DEFINITION 5.27. 0-ary function 0_{Ra}. $0_{Ra} \cong Coset(<0_{Nra},0_{Nra}>,\equiv_{SUB})$.

DEFINITION 5.28. 0-ary function 1_{Ra}. $1_{Ra} \cong Coset(<1_{Nra},0_{Nra}>,\equiv_{SUB})$.

LEMMA 5.45. $0_{Ra} \neq 1_{Ra} \in Ra$.

LEMMA 5.46. Let $x,y,z \in Ra$. Then $x +_{Ra} y = y +_{Ra} x$, $x \bullet_{Ra} y = y \bullet_{Ra} x$, $(x +_{Ra} y) +_{Ra} z = x +_{Ra} (y +_{Ra} z)$, $(x \bullet_{Ra} y) \bullet_{Ra} z = x \bullet_{Ra} (y \bullet_{Ra} z)$, $x \bullet_{Ra} (y +_{Ra} z) = (x \bullet_{Ra} y) +_{Ra} (x \bullet_{Ra} z)$, $x +_{Ra} 0_{Ra} = x$, $x \bullet_{Ra} 1_{Ra} = x$, $(\exists w \in Q)(x +_{Ra} w = 0_{Ra})$, $x \neq 0_{Ra} \rightarrow (\exists w \in Ra)(x \bullet_{Ra} w = 1_{Ra})$, $x <_{Ra} y \rightarrow \neg y <_{Ra} x$, $x <_{Ra} y \wedge y <_{Ra} z \rightarrow x <_{Ra} z$, $x \neq y \rightarrow x <_{Ra} y \vee y <_{Ra} x$, $y <_{Ra} z \rightarrow x +_{Ra} y <_{Ra} x +_{Ra} z$, $0_{Ra} <_{Ra} x \wedge y <_{Ra} z \rightarrow x \bullet_{Ra} y <_{Ra} x \bullet_{Ra} z$.

DEFINITION 5.29. Infix relation $>_{Ra}$. $x >_{Ra} y \leftrightarrow y <_{Ra} x$.

DEFINITION 5.30. Infix relation $\leq_{Ra}$. $x \leq_{Ra} y \leftrightarrow x <_{Ra} y \vee x = y$.

DEFINITION 5.31. Infix function $\geq_{Ra}$. $x \geq_{Ra} y \leftrightarrow x >_{Ra} y \vee x = y$.

DEFINITION 5.32. Infix function $-_{Ra}$. $x -_{Ra} y \cong (!z)(x = y +_{Ra} z)$.

DEFINITION 5.3. 1-ary function Av_{Ra}. $Av_{Ra}(x) \cong (!y \in Ra)((x \geq_{Ra} Ra \rightarrow y = x) \wedge (x <_{Ra} 0_{Ra} \rightarrow y = 0_{Ra} -_{Ra} x))$.

() Av_{Ra} is absolute value on the rationals. ()

LEMMA 5.47. $(\forall x,y \in Ra)(x -_{Ra} y, Av_{Ra}(x) \in Q)$, $(\forall x,y)(x -_Q y\downarrow \leftrightarrow x,y \in Ra)$, $(\forall x)(Av_{Ra}(x)\downarrow \leftrightarrow x \in Ra)$.

LEMMA 5.48. Let $x,y \in Q$. Then $Av_{Ra}(x) \geq_{Ra} 0_{Ra}$, $Av_{Ra}(x \bullet_{Ra} y) = Av_{Ra}(x) \bullet_{Ra} Av_{Ra}(y)$, $Av_{Ra}(x +_{Ra} y) \leq_{Ra} Av_{Ra}(x) +_{Ra} Av_{Ra}(y)$, $Av_{Ra}(x) -_{Ra} Av_{Ra}(y) \leq_{Ra} Av_{Ra}(x -_{Ra} y)$, $x \bullet_{Ra} Av_{Ra}(y) \leq_{Ra} Av_{Ra}(x \bullet_{Ra} y)$.

LEMMA 5.49. DEN[Ra].

DEFINITION 5.35. 0-ary function Nat_{Ra}. $Nat_{Ra} \cong (!x)(\forall y)(y \in x \leftrightarrow (y = 0_{Ra} \vee (y >_{Ra} 0_{Ra} \wedge y -_{Ra} 1_{Ra} \in x)))$.

DEFINITION 5.36. 0-ary function Int_{Ra}. $Int_{Ra} \cong (!x)(\forall y)(y \in x \leftrightarrow (y \in Nat_{Ra} \vee 0_{Ra} -_{Ra} y \in Nat_{Ra}))$.

LEMMA 5.50. $Nat_{Ra} \subset Int_{Ra} \subset Ra$.

DEFINITION 5.37. 0-ary function Seq_{Ra}. $Seq_{Ra} \cong \{x: FCN[x,\omega,Ra]\}$.

LEMMA 5.51. $Seq_{Ra}\downarrow$.

DEFINITION 5.38. Infix function $+_{SeqRa}$. $x +_{SeqRa} y \cong (!z)(x,y,z \in Seq_{Ra} \wedge (\forall n \in \omega)(z(n) = x(n) +_{Ra} y(n)))$.

DEFINITION 5.39. Infix function $\bullet_{SeqRa}$. $x \bullet_{SeqRa} y \cong (!z)(x,y,z \in Seq_{Ra} \wedge (\forall n \in \omega)(z(n) = x(n) \bullet_{Ra} y(n)))$.

LEMMA 5.52. $(\forall x,y \in Seq_{Ra})(x +_{SeqRa} y, x \bullet_{SeqRa} y \in Seq_{Ra})$, $(\forall x,y)(x +_{SeqRa} y\downarrow \leftrightarrow x \bullet_{SeqRa} y\downarrow \leftrightarrow x,y \in Seq_{Ra})$.

LEMMA 5.53. Let $x,y,z \in Seq_{Ra}$. Then $x +_{SeqRa} y = y +_{SeqRa} x$, $x +_{SeqRa} y +_{SeqRa} z = x +_{SeqRa} y +_{SeqRa} z$, $x \bullet_{SeqRa} y = y \bullet_{SeqRa} x$, $x \bullet_{SeqRa} y \bullet_{SeqRa} z = x \bullet_{SeqRa} y \bullet_{SeqRa} z$, $x +_{SeqRa} y = x +_{SeqRa} z \leftrightarrow y = z$, $x \bullet_{SeqRa} (y +_{SeqRa} z) = (x \bullet_{SeqRa} y) +_{SeqRa} (x \bullet_{SeqRa} z)$.

DEFINITION 5.40. Infix relation $<_N$. $x <_N y \leftrightarrow x,y \in \omega \wedge x <_o y$.

DEFINITION 5.41. Infix relation $>_N$. $x <_N y \leftrightarrow x,y \in \omega \wedge y <_o x$.

DEFINITION 5.42. Infix relation $\leq_N$. $x \leq_N y \leftrightarrow x <_N y \vee x = y$.

DEFINITION 5.43. Infix relation $\geq_N$. $x \geq_N y \leftrightarrow x >_N y \vee x = y$.

DEFINITION 5.44. 0-ary function symbol $Cseq_{Rq}$. $Cseq_{Ra} \cong \{x \in Seq_{Ra}: (\forall \varepsilon >_Q 0_{Ra})(\exists n \in \omega)(\forall m,r >_N n)(Av_{Ra}(x(m) -_{Ra} x(r)) <_{Ra} \varepsilon)\}$.

LEMMA 5.54. $Cseq_{Ra} \subset Seq_{Ra}$.

LEMMA 5.55. $(\forall x \in Cseq_{Ra})(\exists n \in Nat_{Ra})(\forall m)(x(m) < n)$.

LEMMA 5.56. $(\forall x,y \in Cseq_{Ra})(x +_{SeqRa} y, x \bullet_{SeqRa} y \in Cseq_{Ra})$.

DEFINITION 5.45. Infix relation $\equiv_{CseqRa}$. $x \equiv_{CseqRa} y \leftrightarrow (\forall \varepsilon >_{Ra} 0_{Ra})(\exists n \in \omega)(\forall m >_N n)(Av_{Ra}(x(m) -_{Ra} y(m)) <_{Ra} \varepsilon)$.

LEMMA 5.57. $EQUIV[\{<x,y>: x \equiv_{CseqRa} y\},Cseq_{Ra}]$.

DEFINITION 5.46. Infix relation $<_{CseqRa}$. $x <_{CseqRa} y \leftrightarrow x,y \in Cseq_{Ra} \wedge (\exists \delta >_{Ra} 0_{Ra})(\exists n \in \omega)(\forall m >_N n)(x(m) +_{Ra} \delta <_{Ra} y(m))$.

LEMMA 5.58. Let $x,y \in Cseq_{Ra}$. Then $![x <_{CseqRa} y, x \equiv_{CseqRa} y, y <_{CseqRa} x]$.

LEMMA 5.59. Let $x,y \in Cseq_{Ra}$. Then $x <_{CseqRa} y \rightarrow \neg y <_{CseqRa} x$, $x <_{CseqRa} y \wedge y <_{CseqRa} z \rightarrow x <_{CseqRa} z$.

LEMMA 5.60. Let $x \equiv_{CseqRa} u$, $y \equiv_{CseqRa} v$. Then $x <_{CseqRa} y \rightarrow u <_{CseqRa} v$, $x +_{CseqRa} y \equiv_{CseqRa} u +_{CseqRa} v$, $x \bullet_{CseqRa} y \equiv_{CseqRa} u \bullet_{CseqRa} v$.

DEFINITION 5.47. 0-ary function $\mathfrak{R}$. $\mathfrak{R} \cong \{Coset(x,\{<x,y>: x \equiv_{CseqRa} y\}): x \in Cseq_{Ra}\}$.

LEMMA 5.61. $PART[\mathfrak{R},Cseq_{Ra}]$.

DEFINITION 5.48. Infix relation $<_{\mathfrak{R}}$. $x <_{\mathfrak{R}} y \leftrightarrow (\exists z,w)(x,y \in \mathfrak{R} \wedge z \in x \wedge w \in y \wedge z <_{CseqRa} w)$.

DEFINITION 5.49. Infix function $+_{\mathfrak{R}}$. $x +_{\mathfrak{R}} y \cong (!z)(x,y,z \in \mathfrak{R} \wedge (\exists a,b,c)(a \in x \wedge b \in y \wedge c \in z \wedge a +_{CseqRa} b \equiv_{CseqRa} c))$.

DEFINITION 5.50. Infix function $\bullet_{\Re}$. $x \bullet_{\Re} y \cong (!z)(x,y,z \in \Re \wedge (\exists a,b,c)(a \in x \wedge b \in y \wedge c \in z \wedge a \bullet_{CseqRa} b \equiv_{CseqRa} c))$.

DEFINITION 5.51. 0-ary function $0_{\Re}$. $0_{\Re} \cong (!x \in \Re)(\forall n \in \omega)(x(n) = 0_{Ra})$.

DEFINITION 5.52. 0-ary function $1_{\Re}$. $1_{\Re} \cong (!x \in \Re)(\forall n \in \omega)(x(n) = 1_{Ra})$.

DEFINITION 5.53. 1-ary function $Av_{\Re}$. $Av_{\Re}(x) \cong (!y \in \Re)((x \geq_{\Re} 0 \rightarrow y = x) \wedge (x <_{\Re} 0_{\Re} \rightarrow y = 0_{\Re} -_{\Re} x))$.

LEMMA 5.62. Let $x,y,z \in \Re$. Then $x +_{\Re} y = y +_{\Re} x \in \Re$, $x \bullet_{\Re} y = y \bullet_{\Re} x \in \Re$, $(x +_{\Re} y) +_{\Re} z = x +_{\Re} (y +_{\Re} z)$, $(x \bullet_{\Re} y) \bullet_{\Re} z = x \bullet_{\Re} (y \bullet_{\Re} z)$, $x \bullet_{\Re} (y +_{\Re} z) = (x \bullet_{\Re} y) +_{\Re} (x \bullet_{\Re} z)$, $x +_{\Re} 0_{\Re} = x$, $x \bullet_{\Re} 1_{\Re} = x$, $(\exists w \in Q)(x +_{\Re} w = 0_{\Re})$, $x \neq 0_{\Re} \rightarrow (\exists w \in \Re)(x \bullet_{\Re} w = 1_{\Re})$, $x <_{\Re} y \rightarrow \neg y <_{\Re} x$, $x <_{\Re} y \wedge y <_{\Re} z \rightarrow x <_{\Re} z$, $x \neq y \rightarrow x <_{\Re} y \vee y <_{\Re} x$, $y <_{\Re} z \rightarrow x +_{\Re} y <_{\Re} x +_{\Re} z$, $0_{\Re} <_{\Re} x \wedge y <_{\Re} z \rightarrow x \bullet_{\Re} y <_{\Re} x \bullet_{\Re} z$, $0_{\Re} \neq 1_{\Re}$.

LEMMA 5.63. 1-ary function $Id_{Q\Re}$. $Id_{Q\Re}(x) \cong (!y \in \Re)(\exists z \in y)(\forall n \in \omega)(z(n) = x)$.

DEFINITION 5.54. 0-ary function symbol $Ra_{\Re}$. $Ra_{\Re} \cong \{Id_{Q\Re}(x): x \in Q\}$.

LEMMA 5.64. $Ra_{\Re} \subset \Re$, $Id_{Q\Re}(x)\downarrow \leftrightarrow x \in Ra$.

LEMMA 5.65. $DEN[\{Ra_{\Re}]$.

LEMMA 5.66. $x <_{\Re} y \rightarrow (\exists z \in Ra_{\Re})(x <_{\Re} z <_{\Re} y)$.

DEFINITION 5.55. 0-ary function $S\Re$. $S\Re \cong \{x: FCN[x,\omega,\Re]\}$.

DEFINITION 5.56. $Cseq_{\Re} \cong \{x \in Seq_{\Re}: (\forall \varepsilon >_{Ra} 0_{Ra})(\exists n \in \omega)(\forall m,r >_{N} n)(Av_{Ra}(x(m) -_{Ra} x(r)) <_{Ra} \varepsilon)\}$.

DEFINITION 5.57. 1-ary function lim. Let $x \in Seq_{\Re}$. Then $\lim(x) = (!y \in \Re)(\forall \varepsilon >_{\Re} 0_{\Re})(\exists n \in \omega)(\forall m >_{N} n)(Av_{\Re}(x(n) -_{Ra} y) <_{\Re} \varepsilon)$.

LEMMA 5.67. Let $x \in Seq_{\Re}$. Then $x \in Cseq_{\Re} \leftrightarrow \lim(x) \in \Re$.

DEFINITION 5.58. 2-ary relation $UB_{\Re}$. $UB_{\Re}[x,A] \leftrightarrow x \in \Re \wedge A \subseteq \Re \wedge (\forall y \in A)(y \leq_{\Re} x)$.

DEFINITION 5.59. 1-ary function $min_{\Re}$. $min_{\Re}(A) \cong (!x \in A)(A \subseteq \Re \wedge (\forall \in A)(\neg y <_{\Re} x))$.

DEFINITION 5.60. 1-ary function $lub_{\Re}$. Let $A \subseteq \Re$. Then $lub_{\Re}(A) \cong min\{x: UB[x,A]\}$.

LEMMA 5.68. $A \neq \varnothing \wedge UB_{\Re}[x,A] \rightarrow lub_{\Re}(A)\downarrow$.

LEMMA 5.69. Let DEN[A], $\{0,1\} \leq_c B \leq_c \omega$. Then $\Re \approx_c$ Maps(A,B).

LEMMA 5.70. $A,B \leq_c \Re \rightarrow A \cup B \leq_c \Re$.

LEMMA 5.71. $A \approx_c \Re \wedge B \leq_c \Re \rightarrow A \cup B \approx_c \Re$.

LEMMA 5.72. $a <_{\Re} b \rightarrow \Re \approx_c \{x \in \Re: a <_{\Re} \times <_{\Re} b\}$,

LEMMA 5.73. $A,B \approx_c \Re \rightarrow A \times B \approx_c \Re$.

14. SOME VALID PROOFLESS TEXT 6.

The material in this section is adapted From [1], chapter 7.

15. SOME VALID PROOFLESS TEXT 7.

The material in this section is adapted From [1], chapter 8.

REFERENCE

[1] P. Suppes, Axiomatic Set Theory, original 1960, republished 1972, Dover Publications.