

Axiomatic Theory of Formulas

Victor Porton

Last updated: 2019-11-04 09:46 PM

Part 1.

Axiomatic Theory of

Formulas

Preface

Overview

This new mathematical theory developed by the book author researches the *properties of mathematical formulas* (aka *expressions*). Naturally this theory is essential for the mathematics.

Formulas are encountered in all areas of mathematics. Probably of special interest are propositional formulas in math logic. Using Operator Theory of Formulas may probably simplify proofs of math logic theorems.

The Theory of Formulas is a new foundation for math logic which is the foundation of mathematics. That is the Theory of Formulas is the *foundation of the foundation of mathematics*.

This theory researches mathematical formulas as well as any other objects which have constituent parts of different kinds (*without* the limit that a whole cannot be a part of its part, that is loops are not disallowed). For example, consider relations between objects in the memory of a computer or components of an electronic scheme connected with different kinds of links.

The Operator Theory of Formulas is not limited only to finite neither only to acyclic formulas. Infinite and cyclic (e.g. $a(b(a(b(\dots))))$) formulas are also modeled by this theory. A common example of a system of infinite formulas is the set of irrational numbers. Another example of an infinite formula may be the entire mathematics in the form of some formalism (e.g. all groups of Group Theory).

This book contains a new axiomatic theory, the theory of *constructs*, and its specializations, including the theory of *formulas*:

- Informally, a construct is anything what has indexed components (parts). *E.g.*, the $a-(b+c)$ can be considered as a construct which has “ a ” as the part with index 1, “ $-$ ” as the part with index 2, and $(b+c)$ (a subformula) as the part with index 3.
- Formulas is a particular case of constructs. Informally, a formula is a construct which contains symbols. *E.g.* $a-(b+c)$ would be a formula which contains symbols such as a , b , c , $+$, $-$.

To increase flexibility and power of the theory I study operators (functions) on the sets of constructs rather than the constructs themselves. So different kinds of constructs can be researched with a uniform method.

An important aspect of the theory of formulas accordingly this book is so called *specialization* that is adding additional subexpressions to expressions. For example if $+$ would mean an abelian group operation, $+_1$ ($+$ with added additional subformula “1”) would mean e.g. addition of whole numbers which is a specialization of abelian group operation etc. Specialization can be used instead of variable substitution.

Recall that *formalist philosophy* is a school in philosophy of mathematics which considers mathematical formulas as physical objects. For this new theory of formulas which includes infinite logical formulas that philosophy (“formalism”) is not applicable as infinite formulas cannot be written and are not physical objects in

this sense. So this theory also changes the philosophy of mathematics.

Draft Status

This is a preliminary version. If you will find any mistakes, please contact me.

Bases

Definition of a Basis

Before studying systems of constructs and formulas, I will introduce the concept which I call a *basis*:

A *basis*¹ is a pair of a set E and some set X (set of *indices* or *indexes*) of functions on E .

Remark. In other words, a basis is an universal algebra (with possibly infinite set of operations X) having only unary operations. Such algebras are sometimes called *unary algebras*. (There are a subtle difference about different operations to which corresponds the same function, but I will lay aside this issue, to be more concise.)

I will call a function from E_1 to E_2 an *operator* from a basis (E_1, X_1) to a basis (E_2, X_2) . An operator from a basis (E, X) will be called an operator *on* this basis.

Obvious. Indices are operators on the basis.

By definition \mathcal{S} (I call it the *set of sequences*) is the set of all finite sequences $p_n \dots p_0$ of zero or more indices $p_i \in X$. To a sequence of indices corresponds an operator on E : $(p_n \dots p_0)\alpha = p_n(p_{n-1} \dots (p_0\alpha) \dots)$.

Remark. \mathcal{S} is a monoid with empty sequence $()$ being the identity element.

¹ Plural form of the word *basis* is *bases*.

Morphisms of Bases

Below I will analyze relations between several bases. For simplicity of explanation I will consider only the case when $X_1 = X_2 = \dots = X$.

The case of two bases (E_1, X_1) and (E_2, X_2) such that $X_1 \neq X_2$ can be eliminated using *reindexing*. *Reindexation* as defined below is a generalization of coordinate system transformation (as found for example in linear algebra).

Reindexation (Coordinate Transformation)

For two sets E_1 and E_2 some f can be operator on both E_1 and E_2 (for example if f is a function $E_1 \cup E_2 \rightarrow E_1 \cap E_2$).

For simplicity I will implicitly assume below that $E_1 \cap E_2 = \emptyset$. (If $E_1 \cap E_2 \neq \emptyset$ we can replace E_1 with $E_1 \times \{1\}$ and E_2 with $E_2 \times \{2\}$.)

Definition. Let $X, \lambda X$ be sets, λ (called a *reindexation function*) be an operator $X \rightarrow \lambda X$ and let $(E_2, \lambda X)$ be a basis. For $i \in X, \beta \in E_2$ we will define $i\beta = (\lambda i)\beta$. After this definition i became an operator also on E_2 and (E_2, X) becomes a basis. The basis $(E_2, X) = \lambda^{-1}(E_2, \lambda X)$ is called *reindexed* (with λ) basis $(E_2, \lambda X)$.

$\text{Reind}(\lambda)$ (called *reindexation*) is a function from $(E_2, \lambda X)$ to $\lambda^{-1}(E_2, \lambda X) = (E_2, X)$ that maps every element to itself (but changes the meaning of indices as shown in the following proposition).

Proposition. $i\text{Reind}(\lambda) = \text{Reind}(\lambda)(\lambda i)$ for $i \in X$.

Proof. Let α is a construct of the system of constructs $(E_2, \lambda X)$. Then $\text{Reind}(\lambda)\alpha$ is a construct from (E_2, X) . So $i\text{Reind}(\lambda)\alpha = \text{Reind}(\lambda)(\lambda i)\alpha$. **End of proof.**

Remark. The above proposition may be not true for $i \in S!$

When we have two bases (E_1, X_1) , (E_2, X_2) and a pair (Ψ, λ) of an operator Ψ from E_1 to E_2 and a reindexation function λ from X_1 to X_2 , we can first apply reindexing to get the basis (E_2, X_1) and then consider Ψ as an operator from (E_1, X_1) to (E_2, X_1) . (I recall that below we will consider only the cases of bases with the same set of indices.)

Coordinate Transformation with an Operator

Let (E, X) be a basis. Let f be an operator on E .

Definition. Functions $\lambda_L(f)$ and $\lambda_R(f)$ on X are defined as $\lambda_L(f)i = fi$; $\lambda_R(f)i = if$.

Definition. *Left coordinate transformation* and *right coordinate transformation* of a basis are defined correspondingly as:

$$L(f) = \text{Reind}(\lambda_L(f)) \text{ and } R(f) = \text{Reind}(\lambda_R(f)).$$

Proposition. $iL(f) = L(f)fi$ and $iR(f) = R(f)if$ for $i \in X$.

Proof. By properties of reindexation. **End of proof.**

Proposition. For $p_1, \dots, p_n \in X$

- $p_n \dots p_1 R(f) = R(f) p_n f \dots p_1 f$;
- $p_n \dots p_1 L(f) = R(f) f p_n \dots f p_1$.

Proof. By properties of reindexation:

- $p_n \dots p_1 R(f) = p_n \dots p_2 R(f) p_1 f = p_n \dots p_3 R(f) p_2 f p_1 f = \dots = R(f) p_n f \dots p_1 f.$
- $p_n \dots p_1 L(f) = p_n \dots p_2 L(f) f p_1 = p_n \dots p_3 L(f) f p_2 f p_1 = \dots = L(f) f p_n \dots f p_1.$ **End of proof.**

Morphisms of Bases

Morphisms of bases (with fixed set of indices X) are particular cases of universal algebra morphisms. Indeed for completeness I will define these here without using universal algebra.

By definition a *basis homomorphism* (or as a less fortunate term, *basis morphism*) from a basis (E_1, X) to a basis (E_2, X) is a function $\Psi: E_1 \rightarrow E_2$ such that $\forall i \in X: \Psi i = i \Psi.$

Proposition. Ψ is a homomorphism iff $\forall i \in S: \Psi i = i \Psi.$

Proof. *Reverse implication.* Obvious.

Direct implication. $\Psi p_n \dots p_0 = p_n \Psi p_{n-1} \dots p_1 = \dots = p_n \dots p_1 \Psi p_0 = p_n \dots p_0 \Psi.$ **End of proof.**

By definition a *basis monomorphism* is an injective basis homomorphism.

By definition a *basis epimorphism* is a basis homomorphism such that $\text{im } \Psi = E_2.$

By definition *basis isomorphism* is a basis homomorphism which is both monomorphism and epimorphism.

Parts. Closed Sets

Let (E, X) is a basis.

Definition. The set of *direct parts* of $\alpha \in E$ is $\{i\alpha \mid i \in X\}$.

Definition. The set of *parts* of $\alpha \in E$ is $\text{parts}(\alpha) = \{i\alpha \mid i \in S\}$.

Definition. The set $A \subseteq E$ is *closed* iff $\forall \alpha \in A : \text{parts}(\alpha) \subseteq A$.

Proposition. If A is a set of element of a basis, the following statements are equivalent:

1. A is subconstruct closed.
2. $\forall \alpha \in A, i \in X : i\alpha \in A$.
3. $\forall \alpha \in A, i \in S : i\alpha \in A$.

Proof. (1) \Leftrightarrow (3) is obvious. (3) \Rightarrow (2) is obvious. (2) \Rightarrow (3) is easily provable by induction. **End of proof.**

Let the set $H \subseteq E$ is closed. A basis $\text{Sys} = (E, X)$ *limited* to the set H is by definition the basis $\text{Sys}|_H = (H, X)$.

(The requirement that H is closed warrants that $\text{Sys}|_H$ is really a system of constructs.)

Operator Theory of Constructs

Definition of Constructs

In the operator theory of constructs *constructs* are defined axiomatically:

I will call the *system of constructs* a pair consisting of a basis (E, X) and a subset $Z \subseteq E$ (elements of Z are called *null* or *nil* constructs) such that $\forall i \in X, 0 \in Z : i0 = 0$.

Elements of E will be called *constructs*. I will denote constructs (as well as formulas, see below) with small Greek letters.

Remark. The above is a reasonable formalization of often used informal notion of *mathematical construct*.

Remark. A basis can be considered as a system of constructs with $Z = \emptyset$.

Remark. In practice most often $Z = \{0\}$ for some $0 \in E$; 0 is called the *null element* (or $Z = \emptyset$). [In the previous version of book there were considered only variant of single element set Z ($Z = \{0\}$). The second part of this book was not yet updated for the cases of card $Z \neq 1$.]

Remark. To a system of constructs corresponds an universal algebra with indices being unary operations and null constructs being constant symbols (nullary operations). That is a system of constructs is essentially the same as an algebra *having only nullary and unary operations* (the number of operations is not required to be finite) with axioms $i0 = 0$ for every unary operation i and nullary operation 0 .

Remark. An other variant of definition of *systems of constructs* is a pair of a set E and a set X of functions from subsets of E to E . [TODO]

I will denote systems of constructs as Sys , Sys_1 , Sys_2 , ... and the corresponding sets of constructs as E , E_1 , E_2 , ..., corresponding sets of indices as X , X_1 , X_2 , ... etc.

Example. Let Y is an arbitrary set (its element are called *symbols* or *terminals*). Let $0 \notin Y$. Let E is the set consisting of 0 , Y and (recursively) of lists (finite sequences) of elements of E . Let $X = \mathbb{N}$. Let for $i \in \mathbb{N}$, $\alpha \in E$ define $i\alpha = 0$ if $\alpha \in Y$ or α is a list shorter than $n+1$ elements, and otherwise $i\alpha$ is the i -th component of α (considered as a list). Then $(E, X, 0)$ is a system of constructs. (Moreover $(E, X, \{0\}, Y)$ is a system of formulas, see below.) This system is called *lists of lists*.

Let the set $H \subseteq E$ is closed. A system of constructs $\text{Sys} = (E, X, Z)$ limited to the set H is by definition the system of constructs $\text{Sys}|_H = (H, X, Z \cap H)$. $\text{Sys}|_H$ is called a *subsystem* of Sys .

Operators on Systems of Constructs

I will call an *operator* from a system of constructs (E_1, X, Z) to a system of constructs (E_2, X, Z) a function from E_1 to E_2 such that $\forall 0 \in Z : f0 = 0$.

Obvious. Indices are operators.

Conditional Pass Operator

Let p is an operator. The *conditional pass* operator $[p]$ is defined by the equations:

$$p\alpha \notin Z \Rightarrow [p]\alpha = \alpha; \quad p\alpha \in Z \Rightarrow [p]\alpha = p\alpha.$$

It is really an operator because $p0 = 0$ and so $[p]0 = p0 = 0$ for any $0 \in Z$.

Obvious. When $Z = \{0\}$ these formulas take the form $[p]\alpha = \alpha \Leftarrow p\alpha \neq 0$ and $[p]\alpha = 0 \Leftarrow p\alpha = 0$.

Proposition. $p[p] = p$.

Proof. $p\alpha \notin Z \Rightarrow p[p]\alpha = p\alpha$ and $p\alpha \in Z \Rightarrow [p]\alpha = pp\alpha = p\alpha$. **End of proof.**

Proposition. $p\alpha \in Z \Rightarrow [p]\alpha \in Z$.

Proof. $p\alpha \in Z \Rightarrow [p]\alpha = p\alpha \in Z$.

Theorem. $[p]q = q[pq]$ for any operators p and q .

Proof. $pq\alpha \notin Z \Rightarrow [pq]\alpha = \alpha; pq\alpha \in Z \Rightarrow [pq]\alpha = pq\alpha;$

$pq\alpha \notin Z \Rightarrow [p]q\alpha = q\alpha; pq\alpha \in Z \Rightarrow [p]q\alpha = pq\alpha;$

$pq\alpha \notin Z \Rightarrow q[pq]\alpha = q\alpha; pq\alpha \in Z \Rightarrow q[pq]\alpha = qpq\alpha = pq\alpha.$

So $pq\alpha \notin Z \Rightarrow q[pq]\alpha = [p]q\alpha$ and $pq\alpha \in Z \Rightarrow q[pq]\alpha = [p]q\alpha. \forall \alpha \in E : q[pq]\alpha = [p]q\alpha$. **End of proof.**

Obvious consequence. $[p_0]p_1 \dots p_n = p_1 \dots p_n [p_0 p_1 \dots p_n]$ for any operators p_0, \dots, p_n .

Theorem. $[p_m \dots p_0][p_n \dots p_0] = [p_{\max(n,m)} \dots p_0]$.

Proof. It enough to prove $[qp][p] = [p][qp] = [qp]$. We have $[qp][p] = [p][qp[p]] = [p][qp]$.

$p\alpha \in Z \Rightarrow [p]\alpha = p\alpha \in Z \Rightarrow [qp][p]\alpha = qp[p]\alpha = qp\alpha$;
 $p\alpha \in Z \Rightarrow qp\alpha \in Z \Rightarrow [qp]\alpha = qp\alpha$; $p\alpha \in Z \Rightarrow [qp][p]\alpha = [qp]\alpha$.

$p\alpha \notin Z \Rightarrow [p]\alpha = \alpha \Rightarrow [qp][p]\alpha = [qp]\alpha$.

So in any case $[qp][p]\alpha = [qp]\alpha$. **End of proof.**

Construct Morphisms

Reindexation of Constructs

Let $\text{Sys}_1 = (E_1, X_1, Z_1)$ and $\text{Sys}_2 = (E_2, X_2, Z_2)$ be two systems of constructs. Let λ be a reindexation function from the basis (E_1, X_1) to the basis (E_2, X_2) . Let the basis (E_2, X_1) be the reindexed (with λ) basis (E_2, X_2) .

Then λ is called *reindexation function* from the system of constructs Sys_1 to the system of constructs Sys_2 . The system of constructs $(E_2, X_1, Z_2) = \lambda^{-1}(E_2, X_2, Z_2)$ is called *reindexed* (with λ) system of constructs (E_2, X_2, Z_2) .

To show that (E_2, X_1, Z_2) is really a system of constructs we need to prove that $i0 = 0$ for $i \in X_1, 0 \in Z_2$. We have $i0 = (\lambda i)0 = 0$.

The things said above about coordinate transformations of bases by functions apply as well to transforming a system (E, X, Z) of constructs (with additional requirement that $\forall 0 \in Z : \Psi 0 = 0$.)

Likewise we can make two systems of constructs (considered as universal algebras) to make agree on the same set of unary operations Z . [TODO: Explain in more details about this. Is order of two operations: reindexation and making agreeing on Z significant?]

As a conclusion, we can limit our further consideration to the case of systems of constructs having the same indices and the same nulls: (E_1, X, Z) and (E_2, X, Z) . (However in the second part of this book we will again return to the topic of reindexation.)

Homomorphisms and Isomorphisms

By definition *construct homomorphism* is such a basis homomorphism Ψ that $\forall 0 \in Z : \Psi 0 = 0$. (I recall that we have limited our consideration to the case $Z_1 = Z_2 = \dots = Z$.)

Obvious. Composition of construct homomorphisms is a construct homomorphism.

Definition. *Construct monomorphism* is an injective construct homomorphism. *Construct epimorphism* is construct homomorphism such that its image of E_1 is E_2 and of Z_1 is Z_2 . *Construct isomorphism* is construct homomorphism which is both construct homomorphism and construct epimorphism.

Obvious. The reverse of a construct isomorphism is a construct isomorphism.

Pseudomorphisms

Definition. Construct pseudomorphism from Sys_1 to Sys_2 is such operator from E_1 to E_2 that $\forall i \in X : \Psi i = i\Psi[i]$.

Proposition. An operator Ψ from E_1 to E_2 is a construct pseudomorphism iff $\forall i \in S : \Psi i = i\Psi[i]$.

Proof. *Reverse implication.* Obvious.

Direct implication. By induction by n for $p_i \in X$:

$$\Psi p_{n\dots p_0} = p_{n\dots p_1} \Psi[p_{n\dots p_1}] p_0 = p_{n\dots p_1} \Psi p_0[p_{n\dots p_0}] = p_{n\dots p_0} \Psi[p_0][p_{n\dots p_0}] = p_{n\dots p_0} \Psi[p_{n\dots p_0}]. \text{ End of proof.}$$

Proposition. Construct homomorphism is a construct pseudomorphism.

Proof. Let $\Psi i = i\Psi$. Then $i\alpha \notin Z \Rightarrow \Psi i\alpha = i\Psi\alpha = i\Psi[i]\alpha$; $i\alpha \in Z \Rightarrow \Psi i\alpha = i\alpha = i\Psi i\alpha = i\Psi[i]\alpha$. **End of proof.**

Proposition. Composition of construct pseudomorphisms is a construct pseudomorphism.

Proof. Let Ψ_1 and Ψ_2 are construct pseudomorphisms. We have $\Psi_2 \Psi_1 i\alpha = \Psi_2 i\Psi_1[i]\alpha = i\Psi_2[i]\Psi_1[i]\alpha$.

$$\Psi_1[i]\alpha \notin Z \Rightarrow \Psi_2 \Psi_1 i\alpha = i\Psi_2 \Psi_1[i]\alpha; \Psi_1[i]\alpha \in Z \Rightarrow \Psi_2 \Psi_1 i\alpha = \Psi_1[i]\alpha = i\Psi_2 \Psi_1[i]\alpha.$$

So $\Psi_2 \Psi_1 i\alpha = i\Psi_2 \Psi_1[i]\alpha$. **End of proof.**

Pseudomorphic Image

Definition. Let $\alpha \in E_1$, $\beta \in E_2$, Then $\text{Pseud}(\alpha, \beta) \Leftrightarrow \beta$ is a *pseudomorphic image* of $\alpha \Leftrightarrow$ there exists a pseudomorphism Ψ from Sys_1 to Sys_2 such that $\beta = \Psi\alpha$.

Proposition. $\text{Pseud}(\alpha, \beta) \wedge \text{Pseud}(\beta, \gamma) \Rightarrow \text{Pseud}(\alpha, \gamma)$ for $\alpha \in E_1, \beta \in E_2, \gamma \in E_3$.

Proof. Because composition of pseudomorphisms is a pseudomorphism. **End of proof.**

Theorem. $\text{Pseud}(\alpha, \beta)$ iff $\forall i, j \in S : (i\alpha = j\alpha \notin Z \Rightarrow i\beta = j\beta)$.

Proof. *Direct implication.* Let $\beta = \Psi\alpha$ where Ψ is a construct pseudomorphism and $i\alpha = j\alpha \notin Z$. Then

$$i\beta = i\Psi\alpha; \quad i\alpha \notin Z \Rightarrow i\beta = i\Psi[i]\alpha = \Psi i\alpha; \quad i\alpha = j\alpha \notin Z \\ \Rightarrow i\beta = \Psi i\alpha = \Psi j\alpha = j\beta.$$

Reverse implication. Let $\forall i, j \in S : (i\alpha = j\alpha \notin Z \Rightarrow i\beta = j\beta)$. Then $\forall i \in S : (i\beta = fi\alpha \Leftarrow i\alpha \notin Z)$ where f is some function defined on $\text{parts}(\alpha)$. From this $\beta = f\alpha$. We will prove that f is a pseudomorphism that is $fi = if[i]$. It is enough to prove $fi\gamma = if[i]\gamma$ for $\gamma = j\alpha \notin Z$ where $j \in S$. We have $fi\gamma = fj\alpha = ij\beta[ij\alpha] = ifj\alpha[ij\alpha] = if\gamma[i\gamma] = if[i]\gamma$. **End of proof.**

Definition. Let $\text{Pseud}(\alpha, \beta)$ where $\alpha \in E_1, \beta \in E_2$. β/α (*factor* of β by α) is the operator which is a construct pseudomorphism from $\text{Sys}_1|_{\text{parts}(\alpha)}$ to $\text{Sys}_2|_{\text{parts}(\beta)}$ such that $(\beta/\alpha)\alpha = \beta$. (Below we will prove that there exist only one factor β/α .)

Proposition. If $\text{Pseud}(\alpha, \beta)$ then $\forall i \in S : (i\alpha \notin Z \Rightarrow (\beta/\alpha)i\alpha = i\beta)$.

Proof. If $i\alpha \notin Z$ then $(\beta/\alpha)i\alpha = i(\beta/\alpha)[i]\alpha = i(\beta/\alpha)\alpha = i\beta$. **End of proof.**

Obvious consequence. If $\text{Pseud}(\alpha, \beta)$ then $\beta/\alpha = \{(i\alpha, i\beta) \mid i \in S, i\alpha \neq 0\}$.

Remark. The above statement allows to extend the definition of β/α to the case when $\neg\text{Pseud}(\alpha, \beta)$ but in this case β/α would be a relation rather than a function.

Consequence. If $\text{Pseud}(\alpha, \beta)$ then exist exactly one β/α .

Proof. We need to prove that if both f and g are factors of β by α then $\forall i \in S : fi\alpha = gi\alpha$. We have $i\alpha \notin Z \Rightarrow fi\alpha = i\beta = gi\alpha$. So $fi\alpha = gi\alpha$. **End of proof.**

Theorem. [TODO: Can't this theorem be made stronger?]

1. Let $\text{Pseud}(\alpha, \beta)$ where $\alpha \in E_1, \beta \in E_2$. If Ψ is a construct pseudomorphism from Sys_2 to Sys_3 then $\text{Pseud}(\alpha, \Psi\beta)$ and $\Psi(\beta/\alpha) = (\Psi\beta)/\alpha$.
2. If $\alpha, \beta \in E_1$ and $\text{Pseud}(\alpha, \beta)$ and Ψ is a construct isomorphism from Sys_1 to Sys_2 then $\text{Pseud}(\Psi\alpha, \Psi\beta)$.

Proof. 1. We have $(\beta/\alpha)\alpha = \beta; \Psi\beta = \Psi(\beta/\alpha)\alpha$. From the previous formula, taking in account that $\Psi(\beta/\alpha)$ is a pseudomorphism (as a composition of pseudomorphisms). $\Psi(\beta/\alpha)$ is a pseudomorphism defined on $\text{parts}(\alpha)$ such that $((\Psi\beta)/\alpha)\alpha = \Psi\beta$; from this by definition of factor we have $(\Psi\beta)/\alpha = \Psi(\beta/\alpha)$.

2. We have $\forall i, j \in S : (i\alpha = j\alpha \notin Z \Rightarrow i\beta = j\beta)$ and need to prove

$$\forall i, j \in S : (i\Psi\alpha = j\Psi\alpha \notin Z \Rightarrow i\Psi\beta = j\Psi\beta)$$

what is equivalent to $\forall i, j \in S : (\Psi i\alpha = \Psi j\alpha \notin Z \Rightarrow \Psi i\beta = \Psi j\beta)$.

Taking in account that Ψ is a injection, $\Psi i\alpha = \Psi j\alpha \notin Z \Rightarrow i\alpha = j\alpha \notin Z \Rightarrow i\beta = j\beta \Rightarrow \Psi i\beta = \Psi j\beta$. **End of proof.**

Reconstructions and Systems of Formulas

Reconstructions

A *reconstruction* is a pair of two systems of constructs Sys_1 and Sys_2 and a relation \rightarrow (called “*can be replaced with*”) between E_1 and E_2 . [TODO: More general case when Sys_1 and Sys_2 are bases rather than constructs.]

Remark. \rightarrow can denote such things as allowed variable substitutions. [TODO: More about this.]

Reconstruction homomorphism from a reconstruction $(\text{Sys}_1, \text{Sys}_2, \rightarrow)$ to a reconstruction $(\text{Sys}'_1, \text{Sys}'_2, \rightarrow')$ is a pair (Ψ_1, Ψ_2) of construct homomorphisms Ψ_1 from Sys_1 to Sys'_1 and Ψ_2 from Sys_2 to Sys'_2 such that

$$\forall \alpha, \beta \in E_1 : (\alpha \rightarrow \beta \Rightarrow \Psi_1 \alpha \rightarrow' \Psi_2 \beta).$$

Reconstruction monomorphism is such reconstruction homomorphism that both Ψ_1 and Ψ_2 are injective.

Reconstruction monomorphism has the *reverse* defined as $(\Psi_1, \Psi_2)^{-1} = (\Psi_1^{-1}, \Psi_2^{-1})$.

Reconstruction epimorphism is such reconstruction homomorphism that Ψ_1, Ψ_2 are construct homomorphisms and

$$\forall \alpha, \beta \in E_1 : (\alpha \rightarrow \beta \Leftrightarrow \Psi_1 \alpha \rightarrow' \Psi_2 \beta).$$

Reconstruction isomorphism is reconstruction homomorphism which is both reconstruction monomorphism and reconstruction epimorphism.

Proposition. The reverse of a reconstruction isomorphism is a reconstruction isomorphism.

Proof. Ψ_1 and Ψ_2 are construct isomorphisms; consequently Ψ_1^{-1} and Ψ_2^{-1} are also construct isomorphisms. We have left to prove that $\forall \alpha, \beta \in E_2 : (\alpha \rightarrow' \beta \Rightarrow \Psi^{-1} \alpha \rightarrow \Psi^{-1} \beta)$ what directly follows from the definition of reconstruction epimorphism. **End of proof.**

Definition. Two reconstructions are *isomorphic* iff there exist a reconstruction isomorphism between them.

Systems of Formulas

A *system of formulas* is a pair of a system of constructs (E, X, Z) and a set $Y \subseteq E$. Elements of Y are called *symbols* (or *constant symbols*).

Remark. Often $\forall \omega \in Y, i \in X : i\omega \in Z$ (that is symbols are *terminals* of the formula). But this is not required by our axiomatic; for example, it can be instead $i\omega = \omega$. Then the graph of subformulas would have loops in the place of symbols.

[TODO: Research also systems of formulas with variable symbols and variable substitution.]

Remark. Our method is more flexible and powerful than traditional variable substitution as (provided that we have enough indices) we can in principle specialize any formula (not only formulas with variables).

To every system of formulas (E, X, Z, Y) corresponds a reconstruction with $\text{Sys}_1 = \text{Sys}_2 = (E, X, Z)$ and “can be replaced with” relation defined as $\alpha \rightarrow \beta \Leftrightarrow \alpha = \beta \vee (\alpha \notin Y \wedge \beta \notin Z)$.

Obvious. For systems of formulas the following statements are equivalent:

- $\alpha \rightarrow \beta$;
- $\alpha = \beta \vee (\alpha \notin Y \wedge \beta \notin Z)$;
- $(\alpha \in Y \vee \beta \in Z) \Rightarrow \alpha = \beta$;
- $\alpha \neq \beta \Rightarrow (\alpha \notin Y \wedge \beta \notin Z)$.

Remark. It seems that it is God who has chosen the letters (X, Z, Y) above straight from the end of the alphabet (I have chosen the letters from the words “index”, “zero”, “symbol”, not the last alphabet letters.) But why then Z and Y are reversed in the order? Isn't it a hint that we should first define formulas without Z straight from the bases and only then add the concept of nulls? Anyway He made Y and Z symmetric (after I replaced a single 0 with more general case of arbitrary set of nulls Z). “It is already not I who writes this, but Christ writes through me.” (a *modified* quote from Bible).

Theorem. “Can be replaced with” relation for a system of formulas is reflexive and transitive.

Proof. *Reflexivity.* Obvious.

Transitivity. If $\alpha \rightarrow \gamma$ then $\alpha \rightarrow \alpha \wedge \alpha \rightarrow \gamma$.

Let $\alpha \rightarrow \beta \wedge \beta \rightarrow \gamma$. If $\alpha \notin Y$ and $\gamma \notin Z$, then $\alpha \rightarrow \gamma$ is obvious. If $\alpha \in Y$, then $\alpha \rightarrow \beta \Leftrightarrow \alpha = \beta$ and so $\alpha \rightarrow \beta \wedge \beta \rightarrow \gamma \Rightarrow \alpha = \beta \wedge \beta \rightarrow \gamma \Rightarrow \alpha \rightarrow \gamma$. If $\gamma \in Z$, then $\beta = \gamma$; $\alpha \rightarrow \gamma$. **End of proof.**

[TODO: It seems that studying formulas (an probably even reconstructions) can be reduced to studying constructs (This seems useful in practice because constructs

are simpler than formulas.) by “encoding” symbols as different topological structures of constructs. Probably it may be further reduced to studying bases. This is similar to that symbols (e.g. letters) in typography are encoded by different shapes.]

Specialization Functions

Definition. A *specialization function* Ψ on a reconstruction $(\text{Sys}_1, \text{Sys}_2, \rightarrow)$ is a pseudomorphism Ψ from Sys_1 to Sys_2 such that $\Psi \subseteq (\rightarrow)$ that is $\forall \alpha \in E_1 : \alpha \rightarrow \Psi\alpha$.

Definition. A *specialization function* on a system of formulas is a specialization function on the corresponding reconstruction.

Proposition. Ψ is a specialization function on a system Sys of formulas iff all of the following is true:

- Ψ is a pseudomorphism;
- $\forall \alpha \in E : (\alpha \in Y \vee \Psi\alpha \in Z \Rightarrow \alpha = \Psi\alpha)$.

Proof. $\alpha \rightarrow \beta \Leftrightarrow \alpha = \Psi\alpha \vee (\alpha \notin Y \wedge \Psi\alpha \notin Z) \Leftrightarrow (\alpha = \Psi\alpha \vee \alpha \notin Y) \wedge (\alpha = \Psi\alpha \vee \Psi\alpha \notin Z) \Leftrightarrow$

$\alpha \in Y \vee \Psi\alpha \in Z \Rightarrow \alpha = \Psi\alpha$ for any $\alpha, \beta \in E$. **End of proof.**

Proposition. If f and g are specialization functions on reconstructions $(\text{Sys}_1, \text{Sys}_2, \rightarrow_1)$ and $(\text{Sys}_2, \text{Sys}_3, \rightarrow_2)$ correspondingly, then gf is a specialization function on $(\text{Sys}_1, \text{Sys}_3, (\rightarrow_2) \circ (\rightarrow_1))$.

Proof. That gf is a pseudomorphism is proved above. We have left to prove that $gf \subseteq (\rightarrow_2) \circ (\rightarrow_1)$. It follows from that $f \subseteq (\rightarrow_1)$, $g \subseteq (\rightarrow_2)$. **End of proof.**

Obvious consequence. Composition of specialization functions on a system of formulas is a specialization function on this system of formulas.

Specialization

By definition (for $\alpha \in E_1$, $\beta \in E_2$) $\alpha \leq \beta \Leftrightarrow \forall i \in S : (i\alpha \notin Z \Rightarrow i\alpha \rightarrow i\beta)$. [TODO: Change the notation, it is not a partial order.]

Proposition. On a system of formulas \leq is reflexive and transitive.

Proof. Directly follows from reflexivity and transitivity of \rightarrow on systems of formulas. **End of proof.**

Specialization $\alpha(\dots)$ (regarding the reconstruction $(\text{Sys}_1, \text{Sys}_2, \rightarrow)$) of the formula α is the set

$$\alpha(\dots) = \{\beta \in E_2 \mid \alpha \leq \beta \wedge \text{Pseud}(\alpha, \beta)\}.$$

Remark. $\alpha(\dots) \subseteq E_2$.

Obvious. $\alpha \in \alpha(\dots)$ for a system of formulas.

The following theorem shows that similarly named concepts of *specialization* and *specialization function* are really related:

Theorem. Let $\alpha \in E_1$, $\beta \in E_2$. Then (1) \Rightarrow (2) \Rightarrow (3):

(1) $\beta \in \alpha(\dots)$.

(2) Exists a specialization function Ψ on $(\text{Sys}_1|_{\text{parts}(\alpha)}, \text{Sys}_2|_{\text{parts}(\beta)}, \rightarrow|_{\text{parts}(\alpha) \times \text{parts}(\beta)})$ such that $\Psi\alpha = \beta$.

(3) $\text{Pseud}(\alpha, \beta)$ and β/α is a specialization function on $(\text{Sys}_1|_{\text{parts}(\alpha)}, \text{Sys}_2|_{\text{parts}(\beta)}, \rightarrow|_{\text{parts}(\alpha) \times \text{parts}(\beta)})$.

If also $\forall 0 \in Z, \beta \in E_2 : 0 \rightarrow \beta$, then the above statements are equivalent.

Proof. (2) \Rightarrow (3). $\text{Pseud}(\alpha, \beta)$ because a specialization function is a pseudomorphism. $\Psi = \beta/\alpha$ because β/α is the only pseudomorphism from $\text{Sys}_1|_{\text{parts}(\alpha)}$ to $\text{Sys}_2|_{\text{parts}(\beta)}$ whose value on α is equal to β .

(1) \Rightarrow (2). We have $\alpha \leq \beta \wedge \text{Pseud}(\alpha, \beta)$. That is

$$\forall i \in S : (i\alpha \notin Z \Rightarrow i\alpha \rightarrow i\beta) \text{ and } \forall i, j \in S : (i\alpha = j\alpha \notin Z \Rightarrow i\beta = j\beta).$$

So exists an operator Ψ defined on $\text{parts}(\alpha)$ such that $\forall i \in S : (i\alpha \notin Z \Rightarrow \Psi i\alpha = i\beta)$.

We have $\forall i, j \in S : (ji\alpha \notin Z \Rightarrow \Psi ji\alpha = j\Psi i\alpha)$.

That is $\forall \gamma \in \text{dom } \Psi, j \in S : (j\gamma \notin Z \Rightarrow \Psi j\gamma = j\Psi \gamma)$, what is the same as

$$\forall \gamma \in \text{dom } \Psi, j \in S : \Psi j\gamma = j\Psi[j]\gamma.$$

So Ψ is a pseudomorphism from $\text{Sys}_1|_{\text{parts}(\alpha)}$ to $\text{Sys}_2|_{\text{parts}(\beta)}$.

From the above $\forall i \in S : (i\alpha \notin Z \Rightarrow i\alpha \rightarrow \Psi i\alpha)$. Consequently $\forall \gamma \in \text{parts}(\alpha) : \gamma \rightarrow \Psi \gamma$, that is Ψ is a specialization function.

(3) \Rightarrow (1). (Provided $\forall 0 \in Z, \beta \in E_2 : 0 \rightarrow \beta$) Let $\forall i, j \in S : (i\alpha = j\alpha \notin Z \Rightarrow i\beta = j\beta)$ and β/α is a specialization function on $(\text{Sys}_1|_{\text{parts}(\alpha)}, \text{Sys}_2|_{\text{parts}(\beta)}, \rightarrow|_{\text{parts}(\alpha) \times \text{parts}(\beta)})$. Because β/α is a specialization function, $\forall \gamma \in \text{parts}(\alpha) : \gamma \rightarrow (\beta/\alpha)\gamma$.

We have $\forall i \in S : i\alpha \rightarrow (\beta/\alpha)i\alpha$. But $i\alpha \notin Z \Rightarrow (\beta/\alpha)i\alpha = i\beta$. So $\forall i \in S : (i\alpha \notin Z \Rightarrow i\alpha \rightarrow i\beta)$, $\forall i \in S : i\alpha \rightarrow i\beta$, that is $\alpha \leq \beta$. Finally $\beta \in \alpha(\dots)$ because $\text{Pseud}(\alpha, \beta)$.
End of proof.

By definition $\langle f \rangle x = fx \Leftrightarrow x \in \text{dom } f$; $\langle f \rangle x = x \Leftrightarrow x \notin \text{dom } f$.

Consequence. Let Sys is a system of formulas, $\alpha, \beta \in E$. Then $\beta \in \alpha(\dots)$ (regarding Sys) iff $\langle \beta/\alpha \rangle|_E$ is a specialization function on Sys . [TODO: Check once more that the proof is correct in all cases.]

Proof. We need to prove that $\Psi = \langle \beta/\alpha \rangle|_E$ is a specialization function on Sys iff β/α is a specialization function on $(\text{Sys}_1|_{\text{parts}(\alpha)}, \text{Sys}_2|_{\text{parts}(\beta)}, \neg|_{\text{parts}(\alpha) \times \text{parts}(\beta)})$.

The direct implication of the above is obvious.

Let β/α is a specialization function on $(\text{Sys}_1|_{\text{parts}(\alpha)}, \text{Sys}_2|_{\text{parts}(\beta)}, \neg|_{\text{parts}(\alpha) \times \text{parts}(\beta)})$. We need to prove that Ψ is a pseudomorphism and that

$$\forall \gamma \in E : (\gamma \in Y \vee \Psi\gamma \in Z \Rightarrow \gamma = \Psi\gamma).$$

On $\gamma \in \text{parts}(\alpha)$ this follows from that β/α is a specialization function on $(\text{Sys}_1|_{\text{parts}(\alpha)}, \text{Sys}_2|_{\text{parts}(\beta)}, \neg|_{\text{parts}(\alpha) \times \text{parts}(\beta)})$.

On $\gamma \notin \text{parts}(\alpha)$ this follows from $\gamma = \Psi\gamma$.

To prove that Ψ is a pseudomorphism it is enough to prove that $\Psi i\gamma = i\Psi[i]\gamma$ for $i\gamma \notin \text{parts}(\alpha)$ because for $i\gamma \in \text{parts}(\alpha)$ we have $\Psi i\gamma = (\alpha/\beta)i\gamma = i(\alpha/\beta)[i]\gamma = i\Psi[i]\gamma$. Let $i\gamma \notin \text{parts}(\alpha)$. We have $\Psi i\gamma = i\gamma$.

If $i\gamma \in Z$, then $\Psi i\gamma = i\gamma \in Z$ and $i\Psi[i]\gamma = i\gamma$. If $i\gamma \notin Z$, then $i\Psi[i]\gamma = i\Psi\gamma = i\gamma$ because $\gamma \notin \text{parts}(\alpha)$. So $i\Psi[i]\gamma = i\gamma = \Psi i\gamma$. **End of proof.**

Proposition. Let Sys is a system of formulas, $\alpha, \beta, \gamma \in E$. Then $\beta \in \alpha(\dots) \wedge \gamma \in \beta(\dots) \Rightarrow \gamma \in \alpha(\dots)$.

Proof. It follows from distributivity of both Pseud and \rightarrow (on systems of formulas). **End of proof.**

Remark. The above proposition can be easily generalized for the case of arbitrary reconstructions.

Proposition. If $\Psi = (\Psi_1, \Psi_2)$ is a reconstruction homomorphism from $(\text{Sys}_1, \text{Sys}_2, \rightarrow)$ to $(\text{Sys}_1', \text{Sys}_2', \rightarrow')$, then $\alpha \leq \beta \Rightarrow \Psi_1\alpha \leq \Psi_2\beta$ for $\alpha \in E_1, \beta \in E_2$.

Proof. Let $\alpha \leq \beta$ that is $\forall i \in S : (i\alpha \notin Z_1 \Rightarrow i\alpha \rightarrow i\beta)$. Let also $\Psi i\alpha \neq 0'$. We need to prove $\Psi_1 i\alpha \rightarrow' \Psi_2 i\beta$. We have $i\alpha \neq 0$. So $i\alpha \rightarrow i\beta$; $\Psi_1 i\alpha \rightarrow' \Psi_2 i\beta$. **End of proof.**

Proposition. Let $\text{Sys}_1, \text{Sys}_2, \text{Sys}_1', \text{Sys}_2'$ be systems of constructs, Ψ_1 is construct isomorphism from Sys_1 to Sys_1' and Ψ_2 is construct homomorphism from Sys_2 to Sys_2' . Then $\text{Pseud}(\alpha, \beta) \Rightarrow \text{Pseud}(\Psi_1\alpha, \Psi_2\beta)$.

Proof. We have $\forall i, j \in S : (i\alpha = j\alpha \notin Z \Rightarrow i\beta = j\beta)$. Let $i\Psi_1\alpha = j\Psi_1\alpha \notin Z$ or equivalently $\Psi_1 i\alpha = \Psi_1 j\alpha \notin Z$. Then because Ψ_1 is a bijection, $i\alpha = j\alpha$ and moreover $i\alpha = j\alpha \notin Z$. So $i\beta = j\beta$. From this $\Psi_2 i\beta = \Psi_2 j\beta$ or equivalently $i\Psi_2\beta = j\Psi_2\beta$. **End of proof.**

Theorem. If $\Psi = (\Psi_1, \Psi_2)$ is a reconstruction isomorphism from $(\text{Sys}_1, \text{Sys}_2, \rightarrow)$ to $(\text{Sys}_1', \text{Sys}_2', \rightarrow')$, then $\beta \in \alpha(\dots) \Leftrightarrow \Psi_2\beta \in (\Psi_1\alpha)(\dots)$ for $\alpha \in E_1, \beta \in E_2$.

Proof. $\beta \in \alpha(\dots) \Rightarrow \Psi_2\beta \in (\Psi_1\alpha)(\dots)$ directly follows from two previous statements. Because Ψ_1, Ψ_2 are bijections, then also analogously $\Psi_2\beta \in (\Psi_1\alpha)(\dots) \Rightarrow \beta \in \alpha(\dots)$. **End of proof.**

Least Common Specialization

Let $(\text{Sys}_1, \text{Sys}', \rightarrow_1)$, $(\text{Sys}_2, \text{Sys}', \rightarrow_2)$, $(\text{Sys}_3, \text{Sys}', \rightarrow_3)$ be some reconstructions.

The *least common specialization* $\text{LCS}(\alpha, \beta)$ of constructs $\alpha \in \text{Sys}_1$ and $\beta \in \text{Sys}_2$ is construct $\gamma \in \text{Sys}_3$ such that $\alpha(\dots) \cap \beta(\dots) = \gamma(\dots)$.

Remark. The above definition may probably somehow vary in a future version of this document.

Remark. LCS is similar to least common dividend (LCD) in arithmetic. [TODO: Explain this similarity.] BTW, we could attempt to associate some logical formulas with numbers and consider number theory in some logical form. Maybe it will lead to a new proof of Last Fermat Theorem, etc. We can also reversely study formulas with number theory like Godel was doing.

Proposition. Let $\text{Sys}_3 = \text{Sys}'$ is a system of formulas and \rightarrow_3 is the “can replace with” relation on that system of formulas. If

$$\forall \gamma, \delta \in E' : (\gamma \in \delta(\dots) \wedge \delta \in \gamma(\dots) \Rightarrow \gamma = \delta),$$

then exists no more than one $\text{LCS}(\alpha, \beta)$. [TODO: Formulate this better.]

Proof. Let γ and δ are both $\text{LCS}(\alpha, \beta)$. Then (because Sys' is a system of formulas) $\gamma \in \gamma(\dots) \subseteq \alpha(\dots)$. So $\gamma \in$

$\alpha(\dots)$; analogously $\gamma \in \beta(\dots)$. From this $\gamma \in \alpha(\dots) \cap \beta(\dots) = \delta(\dots)$, that is $\gamma \in \delta(\dots)$; analogously $\delta \in \gamma(\dots)$. So $\gamma = \delta$. **End of proof.**

Definition. Formulas α and β are *top-comatched* iff $\alpha(\dots) \cap \beta(\dots) \neq \emptyset$.

Unsolved problem. Find the condition to warrant that if two constructs are top-comatched then they have LCS. *[Previously I have tried without success to solve it for the particular cases of lists of lists and binary trees. Now I expect it to be simpler in more abstract formulation, but I have not yet attempted to solve it.]*

[TODO: Find an algorithm for calculating $LCS(\alpha, \beta)$ knowing α and β (in such cases as lists of lists).]

Part 2.

Representation Related Properties of Formulas

Preface

Overview

This is the second part “Representation of Formulas” of the book that describes new axiomatic theory of mathematical formulas (including infinite formulas) as well as of anything what has indexed parts.

This part “Representation of Formulas” is about concrete representations of formulas such as:

- lists of lists,
- n -ary trees,
- binary trees,
- graphs,

etc. as well as about some more complex properties of formulas, including pattern substitutions in formulas.

Infinite Trees

Let X (the set of *indexes* or *indices*) is a set.

By definition S is the set of all finite sequences (zero or more elements) of elements of X .

Given some sets Q and X , the set of *infinite trees* is the set Q^S of all functions from S to Q .

By definition $i\alpha = \{(p, \alpha(pi)) \mid p \in S\}$ for $\alpha \in Q^S$.

$\text{Tree}(Q, X) = (Q^S, X)$ is a basis, called the *system of infinite trees*.

If element $0 \in Q$ then $\text{Tree}(Q, X, 0)$ by definition is the system of constructs $(Q^S, X, \{(i, 0) \mid i \in S\})$.

Constantness of a Construct

Definition. Given a reconstruction $(\text{Sys}_1, \text{Sys}_2, \rightarrow)$ *constantness* $\Omega\alpha$ of a construct $\alpha \in E_1$ is

$$\Omega\alpha = \{\beta \in E_2 \mid \neg(\alpha \rightarrow \beta)\}.$$

That is constantness of a construct is the set of these constructs by which it cannot be replaced.

Proposition. For systems of formulas:

- $\alpha \in Y \Rightarrow \Omega\alpha = E \setminus \{\alpha\};$
- $\alpha \notin Y \Rightarrow \Omega\alpha = .$

Proof. Obvious.

To every construct α corresponds the *tree of constantness*: Form $\alpha = \{(i, \Omega i\alpha) \mid i \in S\}$.

Proposition. Form is a construct homomorphism.

Proof. $j\text{Form } \alpha = \{(p, \Omega p j \alpha) \mid p \in S\} = \text{Form } j \alpha$ for $j \in S$. **End of proof.**

Proposition. $\text{Form } \alpha = \text{Form } \beta \Leftrightarrow \forall i \in S : \Omega i \alpha = \Omega i \beta$
 $\Leftrightarrow \forall i \in S, \gamma \in E_2 : (i \alpha \rightarrow \gamma \Leftrightarrow i \beta \rightarrow \gamma)$.

Proof. Obvious.

Theorem. For systems of formulas the following statements are equivalent:

1. $\text{Form } \alpha = \text{Form } \beta$;
2. $\forall i \in S : (i \alpha \in Y \vee i \beta \in Y \Rightarrow i \alpha = i \beta)$;
3. $\forall i \in S : (i \alpha \neq i \beta \Rightarrow i \alpha, i \beta \notin Y)$;
4. $\alpha \leq \beta \wedge \beta \leq \alpha$.

Proof. (1) \Leftrightarrow (2) \Leftrightarrow (3) is obvious.

$\alpha \leq \beta \Leftrightarrow \forall i \in S : (i \alpha \in Y \Rightarrow i \beta = i \alpha)$.

So $\alpha \leq \beta \wedge \beta \leq \alpha \Leftrightarrow \forall i \in S : (i \alpha \in Y \vee i \beta \in Y \Rightarrow i \beta = i \alpha)$. **End of proof.**

Shadow of a Construct. Equivalence of Constructs

Shadow of a construct α is by definition $\text{Shadow } \alpha = \{(i, i \alpha \neq 0) \mid i \in S\}$.

Proposition. *Shadow* is a construct homomorphism from the system of constructs to the system of infinite trees $\text{Tree}(\{0, 1\}, X, 0)$.²

Proof. [TODO]

2 In this formula the first “0” means boolean false, and the second “0” means null construct from the set E .

Definition. Constructs α and β are equivalent $\Leftrightarrow \alpha \sim \beta$
 $\Leftrightarrow \text{Form } \alpha = \text{Form } \beta \wedge \text{Shadow } \alpha = \text{Shadow } \beta$.

Theorem. The following statements are equivalent:

- $\alpha \sim \beta$ regarding a system of formulas $(E, X, 0, Y)$;
- $\text{Form } \alpha = \text{Form } \beta$ regarding the system of formulas $(E, X, 0, Y \cup \{0\})$.

Proof. We will prove [TODO: Explain why it is enough] that

$$\alpha \sim \beta \Leftrightarrow \forall i \in S : (i\alpha = i\beta \in Y \cup \{0\} \vee i\alpha, i\beta \notin Y \cup \{0\}).$$

We have $\alpha \sim \beta \Leftrightarrow \text{Form } \alpha = \text{Form } \beta \wedge \text{Shadow } \alpha = \text{Shadow } \beta \Leftrightarrow$

$$\forall i \in S : ((i\alpha, i\beta \notin Y \vee i\alpha = i\beta \in Y) \wedge i\alpha = 0 \Leftrightarrow i\beta = 0).$$

Equivalently transforming the propositional formula under the quantifier, we get

$$\begin{aligned} & i\alpha, i\beta \notin Y \vee i\alpha = i\beta \in Y) \wedge i\alpha = 0 \Leftrightarrow i\beta = 0 \Leftrightarrow \\ & (i\alpha, i\beta \notin Y \vee i\alpha = i\beta \in Y) \wedge (i\alpha, i\beta \notin \{0\} \vee i\alpha = i\beta \in \{0\}) \Leftrightarrow \end{aligned}$$

$$\begin{aligned} & i\alpha, i\beta \notin Y \cup \{0\} \vee i\alpha = i\beta \in \{0\} \setminus Y \vee (i\alpha = i\beta \in Y \wedge \\ & \quad i\alpha, i\beta \notin \{0\}) \vee i\alpha = i\beta \in Y \cap \{0\} \Leftrightarrow \end{aligned}$$

$$\begin{aligned} & i\alpha, i\beta \notin Y \cup \{0\} \vee i\alpha = i\beta \in \{0\} \vee (i\alpha = i\beta \in Y \wedge i\alpha, \\ & \quad i\beta \notin \{0\}) \Leftrightarrow \end{aligned}$$

$$i\alpha, i\beta \notin Y \cup \{0\} \vee i\alpha = i\beta \in Y \cup \{0\}.$$

End of proof.

Theorem. For systems of formulas if $\alpha \notin Y$ and $\forall i \in X : \alpha \neq i\alpha$ then then Form α can be restored from $\{(i, \text{Form } i\alpha) \mid i \in X\}$.

Proof. $\alpha \notin Y \Rightarrow \Omega\alpha = \emptyset$. So $\Omega i\alpha$ can be restored for any $i \in S$. **End of proof.**

Applying Operators to Parts of Formulas

Applying an Operator to a Subformula

I will denote $i \leq j$ iff the sequence i is a start part of the sequence j for $i, j \in S$. (Note that $i = j \Rightarrow i \leq j$.)

Let $\alpha \in E$, $i \in X$, f is an operator on the set of formulas (or more generally constructs).

If $i \in X$ then by definition applying f to the i -th ($i \in S$) component (or i -th part) of α is

$$(f@i)\alpha = \{(k, \Omega k\alpha) \mid k \in S, \neg(i \leq k)\} \cup \{(ji, \Omega jfi\alpha) \mid j \in S\}.$$

Remark. $(f@i)\alpha$ may be a tree of constantness of some formula β . Then $(f@i)\alpha \sim \beta$.

Proposition.

- $((f@i)\alpha)k = \Omega k\alpha \Leftarrow \neg(i \leq k)$;
- $((f@i)\alpha)(ji) = \Omega jfi\alpha$.

Proof. Obvious.

Proposition. $f@(ji) = (f@j) @ i$.

Proof. [TODO]

[TODO: @ and R, @ and \geq are somehow related. It seems that R corresponds to the customary notion of repeated applying an operator.]

[TODO] Better Variant

Attaching a set A to a basis (E, X) by a function $f: A \rightarrow E$ is by definition the basis $(E \cup A, X)$ with indices extended onto the set A by the formula $i\alpha = if\alpha$ for $\alpha \in A, i \in X$.

$$i\alpha = i\langle f \rangle \alpha$$

$$(E, X) \cup f^{-1}A$$

Recursive Applying an Operator

Now let's define *recursive applying an operator to parts of an object*.

Let (E, X) is a basis.

Recursive Applying an Operator with Closed Image

[TODO: Illustrations for this section.]

On a Basis

First consider the particular case of an operator $f: A \rightarrow B$ where sets $A, B \subseteq E$ such that:

- $A \cap B = \emptyset$;
- B is closed.

Let $f_D = f \cup (\text{=})|_B$. (f_D is an operator $(A \cup B) \rightarrow B$.)

Definition. *Recursive applying operator f is the operator $\text{Rec}(f) = R(f_D)$. [TODO: What if we would replace R by L in this formula? (L seems less perspective than R however.)]*

Remark. See below about the reason of usage of the word *recursive*.

Let $\alpha \in E$.

Lemma. $\forall i \in X : i\text{Rec}(f)\alpha = \text{Rec}(f)if_D\alpha$.

Proof. $i\text{Rec}(f)\alpha = iR(f_D)\alpha = R(f_D)if_D\alpha = \text{Rec}(f)if_D\alpha$.

End of proof.

Lemma. For $i \in X$

- $i\text{Rec}(f)\alpha = \text{Rec}(f)i\alpha \Leftarrow \alpha \in B$;
- $i\text{Rec}(f)\alpha = \text{Rec}(f)if\alpha \Leftarrow \alpha \in A$.

Proof. Obvious consequence of the previous lemma.

End of proof.

Theorem. For $i \in S$

1. $i\text{Rec}(f)\alpha = \text{Rec}(f)i\alpha \Leftarrow \alpha \in B$;
2. $i\text{Rec}(f)\alpha = \text{Rec}(f)if\alpha \Leftarrow \alpha \in A \wedge i \neq ()$.

Proof. 1. From the lemma, taking in account that B is closed.

2. $p_n \dots p_1 \text{Rec}(f)\alpha = p_n \dots p_2 \text{Rec}(f)p_1 f\alpha = \text{Rec}(f)p_n \dots p_2 p_1 f\alpha$. (I used the previous item.) **End of proof.**

On a System of Formulas

Let we have a system Sys of formulas.

From the above

- $\Omega i\text{Rec}(f)\alpha = \Omega i\alpha \Leftarrow \alpha \in B$;
- $\Omega i\text{Rec}(f)\alpha = \Omega if\alpha \Leftarrow \alpha \in A \wedge i \neq ()$.

So

- $i\text{Rec}(f)\alpha \sim i\alpha \Leftarrow \alpha \in B$;
- $i\text{Rec}(f)\alpha \sim if\alpha \Leftarrow \alpha \in A \wedge i \neq ()$.

If we will consider equivalent formulas equal, that is will replace \sim by $=$ in the above formulas, we will get exactly the traditional definition (with $i \in X$) of recursive applying a function to parts of a formula, except of that $\text{Rec}(f)\omega = \omega$ (not $f\omega$ as it is defined traditionally) for $\omega \in Y$.

So our operator $\text{Rec}(f)$ is a generalization of the traditional concept of recursive applying a function to parts of a formula.

Remark. The issue about the value of $\text{Rec}(f)$ (and also of $R(f)$) on symbols can be reasonably solved by never taking into consideration formulas consisting of single symbols, nor any operators which map a formula which isn't a symbol into a formula which is a symbol. For example (in the case of a system of lists of lists) instead of the formula ω consisting of single symbol we would use the formula (ω) and instead of the substitution (see below) $(\omega \rightarrow \psi)$ (where $\omega, \psi \in Y$) we would use the formula $((\omega) \rightarrow (\psi))$. We also could mean (ω) when we write ω . [TODO: Explain this more clearly. Probably we should introduce special term "single symbol" (distinguished from symbol) to overcome this problem.]

The General Case of Recursive Applying an Operator

Above we have defined $\text{Rec}(f)$ in certain special case.

Now let (E, X) be a basis and f be a function from a subset of E to E . We will define $\text{Rec}(f)$ in this general case.

Let D be an arbitrary bijection from E to any set E^* such that $E \cap E^* = \emptyset$ (that is E^* is a duplicate of E).

Let X^* be a set related to X by a bijective relation $i \leftrightarrow i^*$ for $i \in X, i^* \in X^*$. (I also assume $X \cap X^* = \emptyset$.) By definition $i^* \alpha = i^* D \alpha = D i \alpha$. So $(E \cup E^*, X^*)$ is a basis.

Let the sets $A = \text{dom } f; B = (E \setminus A) \cup E^*$. Easy to show that these sets conform to the requirements for A and B from the previous subsection, that is that the set B is closed (regarding indices X^*).

So then $\text{Rec}(D \circ f)$ defined accordingly the previous subsection is a reindexation of the basis $(E \cup E^*, X^*)$.

We will denote it simply $\text{Rec}(f)$ instead of $\text{Rec}(D \circ f)$.
[See “TODO” below.]

$\text{Rec}(f)$ defined this way is exactly the customary notation of recursively applying a function (except of that thing that f was replaced by $D \circ f$, what is not essential because D is a bijection, and except of the above mentioned issue with values of $\text{Rec}(f)$ on symbols).

[TODO: There is a little glitch in the definition of recursive applying an operator, because the special case (“with closed image”) and the general case (defined in this section) are not the same but differ to the extent of the bijection D . This mess should be somehow eliminated in a good way.]

Remark. Traditional recursive applying function (that is operation $\text{Rec}(f)$) from [my](#) viewpoint looks being

defined so unbeautifully that I'd recommend to not use operation $\text{Rec}(f)$ anymore for any serious work (except of probably teaching math children) but instead use $R(f)$ instead of $\text{Rec}(f)$ exclusively.

Remark. Probably the above is a general method of solving many kinds of recursion related problems (by reducing $\text{Rec}(f)$ to $R(f)$). I have not yet tried it with solving particular problems yet however.

Conjecture. If f is a pseudomorphism then:

- $\text{Rec}(f)$ is a pseudomorphism.
- $\text{Rec}(f)|_{\text{dom } f} = D \circ f$.

Remark. Additionally to the above conjecture we could define *pseudomorphic continuation* and show that $\text{Rec}(f)$ is D combined with pseudomorphic continuation.

Formulas with Variables

Now we will model traditional mathematical formulas in our system.

Let there are following sets none of which intersect others:

- $\{0\}$ (*null or nil*);
- V (set of *variables* or *variable symbols*);
- $Y_0, Y_1, Y_2, \dots, Y_n, \dots$ be correspondingly nullary, unary, binary, ..., n -ary operations.

We will define: $Z = \{0\}$; $Y = Y_0 \cup Y_1 \cup Y_2 \cup \dots$; $X = \mathbb{N}$.

E^* is recursively defined as the union of:

- V ;
- all lists of the form $(\omega_n \alpha_0 \dots \alpha_{n-1})$ where $\omega_n \in Y_n, \alpha_i \in E^*$.

$$E = E^* \cup \{0\}.$$

Applying an index i to a list is defined as its i -th element, or 0 if there are no such element. Applying an index to an element of $V \cup Y \cup Z$ is always 0.

(E, X, Z, Y) is a system of formulas. (It can be called *the traditional system of finite formulas* or *the traditional system of finite expressions*.)

Remark. If we want to use the same symbol as operations of different arity, we can make $Y_i = Y^* \times \{i\}$ where Y^* is the set of symbols denoting operations.

Theorem. For the traditional system of finite formulas specialization is the same as *variable substitution* (as it is defined traditionally).

Proof. It can be proved by induction:

- Let $\alpha \in V, \beta \notin Z$. Then $\alpha \notin Z \wedge \alpha \notin Y$. So α can be replaced with β : $\alpha \rightarrow \beta$.
- $\forall i \in S \setminus \{()\} : i\alpha = 0 \notin Y$. So $\forall i \in S \setminus \{()\} : i\alpha \rightarrow i\beta$. Assembling this we have $\forall i \in S : i\alpha \rightarrow i\beta$ that is $\beta \geq \alpha$.
- $\forall i, j \in S : (i\alpha = j\alpha \notin Z \Rightarrow i\beta = j\beta)$ because $i\alpha \notin Z \Rightarrow i = ()$. So $\text{Pseud}(\alpha, \beta)$. As a result $\beta \in \alpha(\dots)$.
- Let $\alpha \in Y$. Then $\alpha \rightarrow \beta \Leftrightarrow \alpha = \beta$.
- Let $\alpha = (\omega_n \alpha_0 \dots \alpha_{n-1})$ where $\omega_n \in Y_n, \alpha_i \in E^*$.
- $\alpha \rightarrow \beta \Leftrightarrow \alpha = \beta \vee (\alpha \notin Y \wedge \beta \notin Z)$.
- Traditional variable substitution is replacing all variables with some formulas, the same variables with the same formulas. Replacing the same variables with the same formulas is equivalent to replacing all subformulas with the same formulas. It is exactly the requirement $\beta \geq \alpha$ that is $\forall i \in S : i\alpha \rightarrow i\beta$.

Substitutions

By definition *top-substitution* of α by β in γ is $(\alpha \text{ -top-} \beta) = \{(\gamma, \text{Rec}(\gamma/\alpha)\beta) \mid \gamma \in \alpha(\dots)\}$.

In other words,

- $\text{dom}(\alpha \text{ -top-} \beta) = \alpha(\dots)$;
- $(\alpha \text{ -top-} \beta)\gamma = \text{Rec}(\gamma/\alpha)\beta$ for $\gamma \in \alpha(\dots)$.

[TODO: Demonstrate that it corresponds to the traditional definition of top-substitution.]

Example. $((A+B) \rightarrow (A\#B))(U^2+V^3) = U^2\#V^3$ if A and B are variables. *[TODO: Explain details, especially define the concept of variables.]*

Proposition. If $\gamma \in \text{dom}(\alpha \text{ -top-} \beta)$ then $(\alpha \text{ -top-} \beta)\gamma \in \beta(\dots)$. *[TODO: Check.]*

Proof. If $\gamma \in \text{dom}(\alpha \text{ -top-} \beta)$ then $\gamma \in \alpha(\dots)$, γ/α is a specialization function, $(\alpha \text{ -top-} \beta)\gamma = (\gamma/\alpha)\beta$ is a specialization of β . **End of proof.**

By definition $(\alpha \rightarrow \beta)\gamma = \{((\alpha \text{ -top-} \beta) @ i)\gamma \mid i \in S, i\gamma \in \alpha(\dots)\}$.

$(\alpha \rightarrow \beta)\gamma$ is called *substitution* of α by β in γ .

Informally $(\alpha \rightarrow \beta)\gamma$ is the set of all replacements of subformulas of γ which match pattern α with corresponding specialization of β .

Remark. Substitutions have many usages:

- asymmetric analog of equality (with left part which may be indefinite);
- inference rule;

- a sign that a subformula can be substituted with other formula;
- $\alpha \rightarrow \beta$ means “ β is α ”.
- a step in an algorithm;
- declaration that the class α is derived from the class β (in the sense of object oriented programming).

See references below for a list of books where different usages of substitutions are considered.

Definition. *Substitution of all matching patterns* operator $\alpha \text{--all} \rightarrow \beta = \text{Rec}(\alpha \text{--top} \rightarrow \beta)$.

Definition. *Full substitution or repeated substitution* operator $\alpha \text{--full} \rightarrow \beta = R(\langle \alpha \text{--top} \rightarrow \beta \rangle)$.

(I recall that by definition $\langle f \rangle = fx \Leftarrow x \in \text{dom } f$ and $\langle f \rangle x = x \Leftarrow x \notin \text{dom } f$.)

Remark. I deem full substitution a more perspective operation (at least for fundamental mathematics) than traditional substitution of all matching patterns.

Different Representations of Formulas

We have already considered *lists of lists* and *infinite trees* above.

Now we will consider some other examples of systems of formulas.

Symbol With Arguments Representation of Formulas

Let Y be a set. The system of *symbols with arguments* is a subsystem of the lists of lists where the first argument of every list is a symbol (that is an element of Y) and the rest arguments of every list are not symbols (but are lists).

Formulas in the form of symbols with arguments may be denoted as $\omega(\alpha_0 \dots \alpha_n)$ instead of $(\omega \alpha_0 \dots \alpha_n)$.

By definition $H(\omega(\alpha_0 \dots \alpha_n)) = \omega$ for $n \geq 0$. (H is called the *head* of the formula $\omega(\alpha_0 \dots \alpha_n)$).

For $\omega \in Y$ we can define either $H\omega = \omega$ or $H\omega = \omega$ depending on the problem domain and on our tastes. After this we may replace the first element of the set X by ω and get a new (reindexed) system of formulas. [TODO: Say this clearly.]

We can add special symbol "formula" to denote a formula consisting of several symbols:

Example. $2+(3/4)$ may be represented as formula $(2() +() \text{ formula } (3() /() 4()))$ in the form of a symbol with arguments.

Example. Omitting the symbol "formula" in the above formula, we'd get $(2() +() (3() /() 4()))$, removing also empty parentheses we get $(2 + (3 / 4))$.

Remark. This could be also represented simpler as $+(2 /(3 4))$ but this form is less flexible regarding possible specializations.

Binary Tree Representation of Formulas

I call (*infinite*) *binary trees* infinite trees with the set $X = \{0, 1\}$.³

Theorem. If $\text{Sys} = (E, X, 0, Y)$ is a system of formulas and the set $\text{card } X \leq \text{card } \mathbb{N}$ then Sys is isomorphic to a subsystem of the system of binary trees. [TODO: Formulate exactly and prove this theorem.]

Graph Representation of Formulas

[TODO: This section should be illustrated.]

Directed Multigraph with Indexed Edges

Let X be a set.

I will call a *directed multigraph with indexed edges* (DMGWIE) a pair of sets (called the *set of vertices* and the *set of edges*) such that the set of edges contains triples (a, b, n) (called *arrows* from a to b) where a and b are vertices and $n \in X$.

3 They are called *binary* (from Latin *binarius* meaning "double") trees because the set $\{0, 1\}$ is two-element.

I will call *top vertex* (*bottom vertex*) of a DMGWIE a vertex to (from) which there are no arrows.

I will call a *DMGWIE homomorphism* a function f on the set of vertices extended to the set of edges so that $f(a, b, n) = (f(a), f(b), n)$.

An injective homomorphism is called *monomorphism*, homomorphism whose image is the union of the entire set of vertices and the set of edges is called *epimorphism*. A homomorphism which is both monomorphism and epimorphism is called *isomorphism*.

Let G be a DMGWIE, $a \in \text{vertices}(G)$, $n \in X$. I will denote $n_G a$ such $b \in \text{vertices}(G)$ that $(a, b, n) \in \text{edges}(G)$.

Let G be a DMGWIE and $v \in \text{vertices}(G)$. I will denote $Q(G, v)$ the DMGWIE consisting of all vertices which can be reached from v moving in the direction of the arrows (including v itself) and all arrows between these vertices.

A System of Constructs as a DMGWIE

[TODO: About graph representation of bases and bases corresponding to graphs.]

[TODO: Show that the above relations of systems of constructs with DMGWIE are the reverse of each other.]

[TODO: Show how construct morphisms and DMGWIE morphisms correspond to each other.]

[TODO: Two different definitions of depth of a construct.]

Human Representation of Formulas

[TODO: This section is severely incomplete. You can skip its reading for now.]

The following representation of formulas is practically identical to the conventional human mathematical notation. So I call it the human representation of formulas:

A formula is a pair of the main part and the list of modifiers:

- the main part is either a symbol or a list of formulas;
- the list of modifiers is a list of formulas.

The modifiers can be conveniently denoted as lower indices in formulas.

Remark. Modifiers are similar to indexes in the traditional math notation. (I use the word “modifier” instead of “index” to not mess with numeric indexes of elements of lists.) For example, two different kinds of additions (e.g. in two abelian groups or calculated by two different algorithms) may be denoted as $+_1$ and $+_2$. It can be formalized in our system as the constant symbol “+” with a modifier: $2_{+1}2$ and $2_{+2}2$.

Formally human form can be represented e.g. as lists of formulas with special element | intended to separate the modifiers from the main part: $((2|) (+|) (2|) |)$.

Informally, in the human form a specialization is adding additional modifiers to a formula.

Example.

- $(a + b)_1 \in (a + b)(\dots)$ or $((a|) (+|) (b|) | 1) \in ((a|) (+|) (b|) |)(\dots)$;
- $(a +_1 b) \in (a + b)(\dots)$ or $((a|) (+|1) (b|) |) \in ((a|) (+|) (b|) |)(\dots)$.

Example. We could denote real number variables as the constant symbol “real” with indexes (specializations of “real”). Then the school algebra equation $a+b = b+a$ could be formalized as $\text{real}_a + \text{real}_b = \text{real}_b + \text{real}_a$.