Acta Cybernetica **15** (2002) 633–652.

A Framework for Studying Substitution*

Härmel Nestra†

Abstract

This paper describes a framework for handling bound variable renaming and substitution mathematically rigorously with the aim at the same time to stay as close as possible to human intuitive preconception about the phenomena, so that proofs could be deduced from intuitive motivations more directly than in the case of standard approaches.

The theory is developed for general multi-sorted term algebras with variable binding. Therefore, the results hold for a wide class of term calculi such as the λ-calculus, first-order predicate logic, the abstract syntax of programming languages etc.

1 Introduction

1.1 About the Matter

In the area of formal logic, one can detect some kind of discrepancy between human intuition and the standard treatment of the underlying notions like substitution and bound variable renaming. Namely, these concepts are defined inductively on the structure of a term and therefore also the proofs of theorems concerning them require structural induction. But when we are thinking of substitution or variable renaming intuitively, we imagine direct replacement of subterm occurrences at some places instead. As a consequence, our primary intuitive reasoning of claims concerning substitution and variable renaming turns often out to be useless when a mathematically rigorous proof is needed.

In this paper, we introduce a method for overcoming the discrepancy described. In our algebraic framework, substitution is defined via subterm replacement at arbitrary positions not using induction explicitly. Therefore, the notions of subterm occurrence positions and replacement become underlying in the theory and matters of particular examining. These notions are often used by authors but quite seldom investigated in themselves. Nevertheless, many methods have been developed for handling substitution. In the following, some possible approaches are listed.

- The direct definition by induction on term structure. Bound variable renaming is performed only if this is inevitable. For a typical instance, see [6].

*Partially supported by Estonian Science Foundation under grant no 4155.

†Institute of Computer Science, University of Tartu, J. Liivi 2, 50409 Tartu, Estonia; e-mail: nestra@cs.ut.ee

- The direct definition by induction on term structure assuming the variable convention. The latter simplifies the definition essentially because it excludes the cases where variable capturing arise. For an instance, see [3].
- The direct definition by induction on term structure, but always making bound variable renaming. Despite the increased amount of work in performing substitution, the formal definition is quite short. See [11].
- Defining substitutions as a part of object language, not as a family of meta-level operations. This method is suitable especially for computerizing. See [1] for instance.

1.2 More about This Paper

Substitution is a very general concept in itself, occurring as a matter of investigation whenever one deals with some kind of term system. One aim of this work is to develop all the theory uniformly for a large class of calculi.

In Sect. 2, the concept of term algebras being our setting of reasoning is specified. We give a definition of term algebras which captures the concept of absolutely free first-order multi-sorted algebra. The only formal difference from standard is our use of so-called indices in signature. For treating variable binding, we then add some elements to the signature and refine the term forming rules. Variable binding is encoded into terms standardly using place-holding variables because of our pursuit to stay as close to standard practice as possible.

So we reduce studying variable binding to almost standard universal algebra framework with some restrictions. This is possible when dealing with the formal side of the matter only. Evaluation of the terms cannot go standardly any more, because homomorphisms—the evaluation maps in universal algebra—do not work correctly in the binding case. In this paper, semantics is not considered. For ways to define algebra homomorphisms and its related concepts (subalgebras, quotient algebras etc) in cases where binding is involved, see [12].

Section 3 presents a rigorous treatment of subterm occurrence locations using positions. Our treatment of positions in principle is standard.

In Sect. 4, replacement operators are defined and some of their basic properties are considered. A replacement operator is a mapping from terms to terms whose task is to replace the subterms at some (maybe no) positions simultaneously with some other terms. It is determined by a rule establishing which terms have to be placed at which positions.

There is no need in our treatment for the argument term of a replacement operator to possess all the positions at which replacing should be performed according to the formal rule. The redundant part of the rule is ignored by the operator.

Replacement operators can be defined inductively as in [2]. An example for pure λ-calculus is found also in [6] (Sect. 9A). Inductive definition of replacement is rather troublesome, particularly in the light of the simplicity of the idea behind. Intuitively, if we replace some subterms of a term with some other terms, then in the result term, the new terms occur at positions where the replacing was done and the rest is as in the original term. We use a definition which is almost direct translation of this intuitive idea to mathematically rigorous form.

In some works on term rewriting, e.g. [10], terms are treated as functions from so-called tree domains to a ranked alphabet. Replacement operation becomes quite elementary in this approach. The price of it is relatively non-traditional, complicated and being far from human intuition treatment of terms. We succeed in maintaining an almost completely standard definition of terms together with easy and intuitive handling of replacement.

The proofs of the theorems about positions and replacement are omitted in this paper (a few theorems are equipped with proof sketches) because replacement is not the main topic of this paper and the claims are mostly intuitively credible.

The purpose of the remaining sections is to exhibit how the theory of variable binding looks in context of our framework. Section 5 gives the definitions, as well as some basic facts. Section 6 develops the theory for α-congruence. Some of the proved facts are common, some are not. Among other things, we express α-congruence as the reflexive transitive closure of a simple relation. In Sect. 7, we define substitution as an operation on α-classes and prove a couple of its nice properties from this aspect.

We have introduced our method earlier in [7, 8] which are preliminary reports of this paper.

1.3 Related Work

Several methods have been developed for describing uniformly all binding situations arising in computer science. Bring up the following.

- The *higher-order abstract syntax* where all possible binding constructs are expressed via λ-terms. This seems to be the most famous and used method of treating arbitrary binding. For example, see [9].
- The *term algebras of binding signature* of [12, 13, 4]. In these papers, also some categorical viewpoints of the area are provided.
- A theory of binding in Fraenkel–Mostowski set theory is presented in [5].
- The *binding structures* of [14]. This work provides a notion of binding algebras which generalize the de Bruijn λ-calculus.

2 Term Algebras

Defining terms in principle requires fixing a signature and some rules for building new terms from given ones. All terms built under the same rules in the same signature form a term algebra with respect to the rules as algebraic operations.

Take the following system for signature.

S1. A set $\mathbf{\Gamma}$ of *types* (or *sorts*).

S2. A set $\mathbf{\Omega}$ of *term-builders*.

S3. A set $\mathbf{I}_\omega$ of *indices* of ω for every $\omega \in \mathbf{\Omega}$. Assume thereby that all the sets $\mathbf{I}_\omega$ are finite and pairwise disjoint. Take $\mathbf{L} = \bigcup_{\omega \in \mathbf{\Omega}} \mathbf{I}_\omega$.

S4. A type $\tau_i \in \mathbf{\Gamma}$ for each $i \in \mathbf{L}$, and a type $\tau_\omega \in \mathbf{\Gamma}$ for each $\omega \in \mathbf{\Omega}$.

For each type $\gamma \in \mathbf{\Gamma}$, let $\mathcal{X}_\gamma$ be a given set of variables.

T1. For each $\gamma \in \mathbf{\Gamma}$, all the variables from the set $\mathcal{X}_\gamma$ are terms of type γ.

T2. Let ω be any term-builder. For every $i \in \mathbf{I}_\omega$, let u_i be any term of type τ_i. Then $\omega(u)$ is a term of type τ_ω, where u denotes the vector which has all the terms u_i as its components, i.e. $u = (u_i : i \in \mathbf{I}_\omega)$.

T3. All terms are constructed by T1 and T2.

T4. Terms constructed in different ways are different.

These rules suggest that, for any $\omega \in \mathbf{\Omega}$, the τ_i, $i \in \mathbf{I}_\omega$, could naturally be called the *argument types* of ω and τ_ω the *result type* of ω. Roughly speaking, ω has the function type $(\tau_i : i \in \mathbf{I}_\omega) \to \tau_\omega$.

The indices play the same role here as numbers $1, \ldots, n$ in indexing function arguments usually, $|\mathbf{I}_\omega|$ being the arity of symbol ω. Pairwise disjointness of the index sets is essentially used in proofs of some underlying theorems of the theory, e.g. of Theorem 4.3. Hence using the index sets instead of numbers is substantial.

On the other hand, term construction does not depend on what the elements of $\mathbf{I}_\omega$ actually are, since rule T2 uses u_i in building a new term, not i. Therefore renaming the indices does not change the terms essentially.

Denote our term algebra by $\mathcal{T}$, i.e. $\mathcal{T} = (\mathcal{T}_\gamma : \gamma \in \mathbf{\Gamma})$ where $\mathcal{T}_\gamma$ is the set of all the terms of type γ. The term constructing rules imply that any term which is not a variable has exactly one type. Variables may have several types.

For accommodating variable binding into the framework, a partitioning

$$\mathbf{I}_\omega = \mathbf{A}_\omega \cup \bigcup_{a \in \mathbf{A}_\omega} \mathbf{B}_a \tag{1}$$

is determined for each $\omega \in \mathbf{\Omega}$ by the signature (both unions must be disjoint). Call the indices of set $\mathbf{A}_\omega$ the *argument indices* of ω; the others are *binding indices*. Each binding index belongs to exactly one set $\mathbf{B}_a$, being thus associated to a fixed argument index. This is for determining the scope of binding.

The idea is that, in any term of shape $\omega(u)$, the arguments u_b of ω for binding indices b stand for binding occurrences of variables. This is achieved by treating the variables of each type as they were forming a separate type.

To be precise, we have a partitioning $\mathbf{\Gamma} = \Gamma \cup \tilde{\Gamma}$ where the elements of Γ are the basic types and the elements of $\tilde{\Gamma} = \{\tilde{\gamma} : \gamma \in \Gamma\}$ are the corresponding variable types. Thereby, τ_b is restricted to be a variable type whenever b is a binding index, while τ_a and τ_ω are restricted to be basic types for every argument index

a and term-builder ω. So each variable has simultaneously its basic type and the corresponding variable type. Assume in the rest for simplicity that each variable has exactly one basic type.

Assume additionally that a term $\omega(u)$, $u = (u_i : i \in \mathbf{I}_\omega)$, can be constructed only if $u_{b_1} = u_{b_2}$ implies $b_1 = b_2$ for any $a \in \mathbf{A}_\omega$ and $b_1, b_2 \in \mathbf{B}_a$. That means, it is allowed to bind one variable only once with the same scope.

For a simple characterizing example, take pure λ-calculus. It has two term-builders—juxtaposition denoting application and the binding symbol λ. As there are no restrictions imposed on the construction of terms except that the first argument of λ must be a variable, pure λ-calculus has one basic type *TERM* and its subtype *VARIABLE*. One may take index sets as in [6] (Sect. 9A): $\mathbf{I}_\cdot = \{1, 2\}$ ($\cdot$ denoting juxtaposition) and $\mathbf{I}_\lambda = \{*, 0\}$, whereby $*$ is the only binding index. So $\tau_1 = \tau_2 = \tau_0 = \mathit{TERM}$ and $\tau_* = \mathit{VARIABLE}$.

For a concrete example, take the λ-term $M = (\lambda x.y)x$ (x and y being variables). Use index 1 for the left-hand argument of juxtaposition and 2 for the other. Then we get $M = \cdot(u)$ where $u = (u_1, u_2)$ with $u_1 = \lambda x.y$ and $u_2 = x$. Further, $u_1 = \lambda(v)$ where $v = (v_0, v_*)$ with $v_0 = y$ and $v_* = x$.

For a bit more complicated example, take untyped first-order predicate logic in some signature. There are two binders $\forall$ and $\exists$. For term-builders, take all the individual, function and predicate symbols of the signature, as well as the propositional connectives. In predicate logic, one distinguishes between so-called "terms" and "formulas". Hence take *TERM* and *FORMULA* to be the two basic types. A variable type *VARIABLE* is corresponding to *TERM* (no propositional variables are used in predicate logic usually, so one can manage without a variable type corresponding to *FORMULA*). One can take the argument index sets arbitrarily in a way that the condition S3 from Sect. 2 holds and the arities match (individual symbols must be taken as 0-ary function symbols). For all function symbols, the argument types equal to *TERM* and also the result type; for all predicate symbols, the argument types equal to *TERM*, but the result type is *FORMULA*. For propositional connectives, the argument types, as well as the result type, equal to *FORMULA*. For quantifiers, the type of the binding argument is *VARIABLE* and the type of the other argument, as well as the result type, is *FORMULA*.

3 Positions and Occurrences

Think of $\mathbf{L}$ as a (possibly infinite) alphabet (recall from Sect. 2 that $\mathbf{L} = \bigcup_{\omega \in \Omega} \mathbf{I}_\omega$).

Definition 3.1. *Positions* are strings over the alphabet $\mathbf{L}$, i.e. elements of $\mathbf{L}^*$.

Definition 3.2. Define the notion *term t occurs in term s at position p* inductively as follows:

- t occurs in s at ϵ (the empty string) iff $s = t$;

- if $p = p'i$ with $p' \in \mathbf{L}^*$, $i \in \mathbf{L}$, then t occurs in s at p iff a term of shape $\omega(u)$ occurs in s at p', such that $i \in \mathbf{I}_\omega$ and $u_i = t$.

Say that t is a *subterm* of s iff t occurs in s at some position. For intuitive understanding of positions, consider term trees. For each $i \in \mathbf{I}_\omega$, label the edge connecting the root of the tree of $\omega(u)$ with the root of the tree of u_i by i. The position where a subterm occurs in a term is found by writing down in order the labels of edges appearing on the path which connects the root of the term with the root of the subterm. So the only difference from standard treatment of positions is that we use the indices given by signature instead of integers for constructing them.

According to Definition 3.2, only terms of type τ_i can occur in term s at position $p'i$. Therefore define $\tau_{p'i} = \tau_i$ for any $p' \in \mathbf{L}^*$ and $i \in \mathbf{L}$.

Definition 3.2 implies also that, in any term, at most one term can occur at a given position. Denote by $s\,.\,p$ the term occurring in s at p, if any. For each $p \in \mathbf{L}^*$, $(.p)$ is a partial function working on terms. However, we treat it as a total function with a special value $\perp$ for denoting undefinedness. So an equality like $s\,.\,p = t\,.\,p$ is valid if neither $s\,.\,p$ nor $t\,.\,p$ exists, but if exactly one of $s\,.\,p$ and $t\,.\,p$ exists, then it is not valid. Assume additionally $\perp.\,p = \perp$ for every p. Then $s\,.\,\epsilon = s$ for all terms s and $s\,.\,pq = s\,.\,p\,.\,q$ for all terms s and positions p, q.

Say that p *is a position of* s iff something occurs in term s at position p, i.e. $s\,.\,p \neq \perp$. For all terms s, denote $\operatorname{pos} s = \{p : s\,.\,p \neq \perp\}$, i.e. $\operatorname{pos} s$ is the set of all positions of s. Clearly $\operatorname{pos} x = \{\epsilon\}$ for any variable x.

Definition 3.3. (i) Let $p, q \in \mathbf{L}^*$. Say that q is a *refinement* of p iff p is a prefix of q, i.e. $q = pr$ for some $r \in \mathbf{L}^*$. If this holds, we write $p \leq q$ (or $q \geq p$).

(ii) If $p, q, r \in \mathbf{L}^*$ with $q = pr$, then write $q\big|_p$ for r.

(iii) For arbitrary $Q \subseteq \mathbf{L}^*$ and $p \in \mathbf{L}^*$, define $pQ = \{pq : q \in Q\}$ and $Q\big|_p = \{r : pr \in Q\} = \left\{q\big|_p : q \in Q, p \leq q\right\}$.

(iv) Let $p, q \in \mathbf{L}^*$. If neither p is a prefix of q nor q is a prefix of p, then p and q are called *divergent* and denoted $p \asymp q$.

(v) Let $P \subseteq \mathbf{L}^*$ and $q \in \mathbf{L}^*$. If $q \asymp p$ for each $p \in P$, then call q *divergent from* P and write $q \asymp P$.

For a concrete example, take the λ-term $M = (\lambda x.y)x$ studied in Sect. 2. Choose the argument indices as above. Then the term $(\lambda x.y)$ occurs in M at position 1 and the term x occurs in M at position 2. Analogously, y occurs in $\lambda x.y$ at position 0 and x occurs in $\lambda x.y$ at position $*$. So the term y occurs in M at position 10 and the term x occurs in M also at position $1*$. Moreover, M itself occurs in M at position ϵ. As the variables are not constructed terms but primitive, no terms occur in variables at non-empty positions. Thus we can find no other occurrences of terms in M, so $\operatorname{pos} M = \{\epsilon, 1, 2, 10, 1*\}$. Among the positions of M, 1 and 2 are divergent, 10 and $1*$ are divergent, and 2 is divergent from both 10 and $1*$.

The relation $\leq$ is a partial order on $\mathbf{L}^*$. So one can speak about maximality and minimality with respect to the relation $\leq$. If $P \subseteq \mathbf{L}^*$, then any two positions both maximal in P are divergent. The same holds for positions minimal in P. If p and q are divergent, then any refinements of these are, too. Any set $Q \subseteq \mathbf{L}^*$

whose elements are pairwise divergent is called *antichain*, whereby Q is said to be an antichain of P whenever $Q \subseteq P$.

The following is one of the theorems underlying our theory. The proof uses essentially the pairwise disjointness of the index sets. Note that we write Σ^r, Σ^t and Σ^c for reflexive, transitive and compatible, respectively, closure of relation Σ. Recall from universal algebra that a binary relation Σ is called compatible iff, for any $\omega \in \boldsymbol{\Omega}$ and vectors $(u_i : i \in \mathbf{I}_\omega)$ and $(v_i : i \in \mathbf{I}_\omega)$, $(u_i, v_i) \in \Sigma$ for all $i \in \mathbf{I}_\omega$ implies $(\omega(u), \omega(v)) \in \Sigma$.

Theorem 3.4. *Let Σ be a binary relation on $\mathcal{T}$ and s, t be terms. Then $(s,t) \in \Sigma^c$ iff there exists a common maximal w.r.t. set inclusion antichain P of both* $\operatorname{pos} s$ *and* $\operatorname{pos} t$ *such that $(s \mathbin{.} p, t \mathbin{.} p) \in \Sigma$ for all $p \in P$.*

Proof. ($\Rightarrow$) Build the compatible closure of Σ iteratively. Argue by induction on the number of steps it takes to get (s,t).

($\Leftarrow$) Let $\|P\|$ be the sum of the lengths of positions of P. The claim follows by induction on $\|P\|$. □

As a corollary, we get the following theorem which provides a method for proving that terms are congruent if we know that some corresponding subterms of them are congruent. Note that equality is just a particular congruence relation.

Theorem 3.5. *Let $\equiv$ be any congruence relation on the term algebra $\mathcal{T}$. Let s, t be terms and $P \subseteq$* $\operatorname{pos} s \cap \operatorname{pos} t$ *an antichain. If $s \mathbin{.} p \equiv t \mathbin{.} p$ for each $p \in P$ and $s \mathbin{.} q = t \mathbin{.} q$ for each $q \asymp P$, then $s \equiv t$.*

4 Replacement Operators

Definition 4.1. Call a function f a *placing rule* iff its domain $\operatorname{dom} f$ is an antichain of $\mathbf{L}^*$ and $f(p) \in \mathcal{T}_{\tau_p}$ for every non-empty $p \in \operatorname{dom} f$.

Definition 4.2. If f is a function with domain P and $q \in \mathbf{L}^*$, then $f\big|_q$ denotes the function with domain $P\big|_q$ and $f\big|_q(r) = f(qr)$ for each $r \in P\big|_q$.

Note that $f\big|_q$ is a placing rule whenever f is.

Our treatment of replacement is grounded on the following theorem.

Theorem 4.3. *Let s be a term and f be a placing rule with domain $P \subseteq$* $\operatorname{pos} s$. *Then there exists a unique term t such that $t.p = f(p)$ for each $p \in P$ and $t.q = s.q$ for each $q \asymp P$. Thereby, if $\epsilon \notin P$ then t is of the same type as s.*

Proof. Argue by induction on $\|P\|$ as in Theorem 3.4. An alternative way is to prove the claim for singleton sets P at first, using induction on the only element of P, and generalize to any P by induction on $|P|$. The uniqueness part of this theorem can be deduced also from Theorem 3.5. □

Theorem 4.3 justifies the following definition.

Definition 4.4. Let s be a term. Let f be a placing rule.

(i) If $\operatorname{dom} f = P \subseteq \operatorname{pos} s$, then define $[f](s)$, the result of simultaneous replacement of the subterms of s at positions $p \in P$ by corresponding terms $f(p)$, to be the unique term whose existence is claimed by Theorem 4.3.

(ii) If $\operatorname{dom} f \not\subseteq \operatorname{pos} s$, then define $[f](s) = [f|_{\operatorname{pos} s}](s)$ where $f|_{\operatorname{pos} s}$ is the function with domain $\operatorname{dom} f \cap \operatorname{pos} s$ behaving like f on it.

Hence replacement operators are defined for cases only for which the positions where the replacement must be performed simultaneously are pairwise divergent.

The replacement operator $[f]$ will frequently be denoted similarly to set comprehension syntax by $[f(p) : p \in \operatorname{dom} f]$ (with some concrete expressions at place of $f(p)$ and $\operatorname{dom} f$ of course; we often practise even writings like $[z : r \in R]$—this means that the placing rule is constantly z with domain R). Assuming $\operatorname{dom} f = \{p_1, \dots, p_n\}$ and $u_i = f(p_i)$ for each $i = 1, \dots, n$, one can write also $[p_1 \mapsto u_1, \dots, p_n \mapsto u_n]$ instead of $[f]$.

Theorem 4.5. *Let s be a term. Let f be a placing rule. Denoting $P = \operatorname{dom} f \cap \operatorname{pos} s$, we have*

$$\operatorname{pos}[f](s) = \bigcup_{p \in P} (p \operatorname{pos} f(p) \cup \{r : r < p\}) \cup \{q \in \operatorname{pos} s : q \asymp P\}.$$

If we replace variables with variables only, then P is a subset of all positions maximal in $\operatorname{pos} s$ and, for each $p \in P$, $\operatorname{pos} f(p) = \{\epsilon\}$. Hence Theorem 4.5 implies that replacing variables with variables does not change the set of positions.

The following theorems state some basic properties of replacement. Theorem 4.7 is for computing expressions of form $[f](s) \centerdot q$. Theorem 4.8 states that if the replacement operators do not "disturb" each other, then the order of their application is unimportant. Theorem 4.9 states some conditions under which one replacement operator absorbs another in consecutive application. Note that if function composition is denoted by ;, then the left function is applied first.

Theorem 4.6. *Let s be a term and f a placing rule such that $f(p) = s \centerdot p$ for each $p \in \operatorname{dom} f \cap \operatorname{pos} s$. Then $[f](s) = s$.*

Theorem 4.7. *Let s be a term, f be a placing rule and $q \in \mathbf{L}^*$.*

(i) *If $p \le q$ for some $p \in \operatorname{dom} f \cap \operatorname{pos} s$, then $[f](s) \centerdot q = f(p) \centerdot q|_p$.*

(ii) *If $p < q$ for no $p \in \operatorname{dom} f \cap \operatorname{pos} s$, then $[f](s) \centerdot q = [f|_q](s \centerdot q)$.*

Note that the conditions of Theorem 4.7 (ii) hold whenever $q \le p$ for some $p \in \operatorname{dom} f \cap \operatorname{pos} s$ because $\operatorname{dom} f$ is an antichain.

Theorem 4.8. *Let $f_1, \dots, f_n$ be placing rules. Assume that the positions of $\operatorname{dom} f_i$ are divergent from the positions of $\operatorname{dom} f_j$ whenever $i \ne j$. Then $[f_1] ; \dots ; [f_n] = [h]$ where $\operatorname{dom} h = \operatorname{dom} f_1 \cup \dots \cup \operatorname{dom} f_n$ and $h(p) = f_i(p)$ whenever $p \in \operatorname{dom} f_i$. So the composition does not depend on the order of application.*

Theorem 4.9. *Let f, g be placing rules such that each element of* $\operatorname{dom} f$ *has a prefix in* $\operatorname{dom} g$. *Then* $[f]\,;[g] = [g]$.

Theorems 4.9, 4.7 (ii) and 4.6 together give the following.

Theorem 4.10. *Let s be a term and f a placing rule. Let Q be an antichain of* $\operatorname{pos} s$ *such that every* $p \in \operatorname{dom} f \cap \operatorname{pos} s$ *has a prefix in Q. Then* $[f](s) = [[f|_q](s \centerdot q) : q \in Q](s)$.

These properties of replacement are intuitively rather clear, therefore we will often use them in the rest without explicit mentioning.

We now discuss briefly the so-called replacement property. This is one of the main properties assumed about rewrite relations in term rewriting theory.

Definition 4.11. A binary relation Σ on algebra $\mathcal{T}$ is said to have the *replacement property* iff, for any terms s, u and position $p \in \operatorname{pos} s$, $(s \centerdot p, u) \in \Sigma$ implies $(s, [p \mapsto u](s)) \in \Sigma$.

In other words, a relation Σ has the replacement property iff replacing any subterm of s with a term related to it gives a term related to s.

The following theorem shows the connections between the replacement property and compatibility.

Theorem 4.12. (i) *Any reflexive compatible relation Σ on $\mathcal{T}$ has the replacement property.*

(ii) *Any transitive relation having the replacement property is compatible.*

(iii) *The reflexive closure of any relation having the replacement property has the replacement property.*

(iv) *The transitive closure of any relation having the replacement property both is compatible and has the replacement property.*

5 Variable Binding in Terms

Our treatment of binding uses positions essentially: binding of positions is the primary, binding of variable occurrence the secondary notion.

Definition 5.1. Call a position q *binding* iff $q = pb$ for some $p \in \mathbf{L}^*$, $b \in \mathbf{B}_a$, $a \in \mathbf{A}_\omega$ and $\omega \in \mathbf{\Omega}$. Thereby call position pa the *root of binding* corresponding to pb.

If either a binding position pb or its root of binding pa, where $b \in \mathbf{B}_a$ and $a \in \mathbf{A}_\omega$, belongs to $\operatorname{pos} s$, then $s \centerdot p = \omega(u)$ for some vector $u = (u_i : i \in \mathbf{I}_\omega)$ which gives $s \centerdot pb = u_b$ and $s \centerdot pa = u_a$. Hence a binding position belongs to $\operatorname{pos} s$ iff its root of binding does.

Definition 5.2. Let x be a variable and s a term.

(i) Let p be a position. If $s \centerdot pb = x$ for $b \in \mathbf{B}_a$, $a \in \mathbf{A}_\omega$ and $\omega \in \mathbf{\Omega}$, then say that the occurrence of x at pb in s is *binding*, or equivalently, *x occurs binding* at pb in s. We may say additionally that position pa is the root of binding of x.

(ii) Let p, r be positions of s. Take $\omega \in \mathbf{\Omega}$, $a \in \mathbf{A}_\omega$ and $b \in \mathbf{B}_a$ such that x occurs binding at pb in s. We say that the occurrence of x at pb in s *binds* position r iff r is not a binding position and pa is the longest prefix of r being a root of binding of x in s. If additionally $s \centerdot r = x$, then say that the occurrence of x at pb binds the occurrence of x at r.

(iii) Let r be a position of s. Say that x is *bound* at r in s iff there exists an occurrence of x in s which binds it at r. If additionally $s \centerdot r = x$, then say that the occurrence of x at r in s is bound, or equivalently, x *occurs bound* at r in s.

(iv) Let r be a position of s. Say that x is *free* at r in s iff r is not binding and x is not bound in r. If additionally $s \centerdot r = x$, then say that the occurrence of x at r in s is free, or equivalently, x *occurs free* at r in s.

Note that an occurrence of x can be binding even if there are no occurrences of x bound by that occurrence. Moreover, note that x can be bound or free at r even if it does not occur at r.

In the paper [7] of ours, binding was defined in a slightly simpler way. This was possible because of some more restrictions imposed on terms there.

Proposition 5.3. *Let s be a term, x a variable and r a non-binding position of s.*

(i) x is bound at r in s iff some prefix of r is a root of binding of x in s.

(ii) x is free at r in s iff no prefix of r is a root of binding of x in s.

Proof. Straightforward by Definition 5.2. □

Definition 5.4. (i) Let $q \in \operatorname{pos} s$ be any binding position and $s \centerdot q = x$. The set consisting of position q, as well as all positions r such that $s \centerdot r = x$ and the occurrence of x at q binds the occurrence of x at r, is called the *binding unit* (of x) corresponding to q in s and denoted by $\operatorname{bu}(q, s)$.

(ii) Let s be a term. For any variable x, let $\operatorname{fpos}_x s$ denote the set of all positions where x occurs free in s, and $\operatorname{bpos}_x s$ denote the set of all positions where x occurs non-free (i.e. binding or bound) in s. Define

$$\operatorname{fpos} s = \bigcup_x \operatorname{fpos}_x s,. \qquad \operatorname{bpos} s = \bigcup_x \operatorname{bpos}_x s. \tag{2}$$

(iii) For term s, let $\operatorname{fv} s$ denote the set of all variables having a free occurrence in s, and $\operatorname{bv} s$ denote the set of all variables having a binding occurrence in s.

Proposition 5.5. *Let s be a term and p its position. Then* $\operatorname{fpos}(s \centerdot p) = \operatorname{fpos} s\big|_p \cup R$ *where R is the set consisting of all positions where some variable not being free at p in s occurs free in $s \centerdot p$.*

Proof. If a variable occurs free at pr in s, then it clearly occurs free at r in $s \centerdot p$. A variable occurring free at r in $s \centerdot p$ but non-free at pr in s means that the root of binding pr at s is a prefix of p, i.e. the variable is not free at p in s. □

Definition 5.6. (i) Let ϱ be any type-preserving mapping of variables to variables. Define the *naive renaming* $\lceil\varrho\rceil$ on arbitrary term s by $\lceil\varrho\rceil(s) = [\varrho(s \centerdot p) : p \in \operatorname{bpos} s](s)$.

(ii) Let σ be any type-preserving mapping of variables to terms. Define the *naive substitution* $\lfloor\sigma\rfloor$ on arbitrary term s by $\lfloor\sigma\rfloor(s) = [\sigma(s \centerdot p) : p \in \operatorname{fpos} s](s)$.

(iii) For any mapping σ of variables to terms, define $\operatorname{supp}\sigma = \{x : \sigma(x) \neq x\}$ (the *support* of σ). The application of $\lfloor\sigma\rfloor$ to s is called *sound* iff $\operatorname{bv} s$ is disjoint with $\operatorname{fv}\sigma(x)$ whenever $x \in \operatorname{fv} s \cap \operatorname{supp}\sigma$.

Naive renaming and substitution are not entirely satisfactory because variable capturing has not been taken into account. They provide a starting point for defining and studying correct capture-avoiding substitution. Correct bound variable renaming is the matter of the next definition.

Definition 5.7. (i) Let s be a term and q a binding position of s. Let z be a variable of type τ_q. Define $\operatorname{ren}_{q\to z}(s) = [z : p \in \operatorname{bu}(q, s)](s)$.

(ii) Say that terms s, t are in relation A (the renaming-step relation) iff $t = \operatorname{ren}_{pb\to z}(s)$ for some binding position pb, $b \in \mathbf{B}_a$, of s and a *fresh* variable z, i.e. z not occurring in $s \centerdot pa$.

(iii) The least congruence relation containing A is denoted by α.

Proposition 5.8. *Let s be any term.*

(i) *For any type-preserving mapping ϱ of variables to variables,* $\operatorname{pos}\lceil\varrho\rceil(s) = \operatorname{pos} s$.

(ii) *If t is a term such that $(s, t) \in$ A, then* $\operatorname{pos} s = \operatorname{pos} t$.

Proof. By the remark made after Theorem 4.5. □

The following three lemmas are rather straightforward corollaries of the definitions presented above. Their proofs however require quite technical and uninteresting study of details, so we omit them as the aim of this paper is to exhibit the proofs of more complicated theorems in the framework developed.

Lemma 5.9. *Let s be a term, $r \in \operatorname{pos} s$ and q a binding position of $s \centerdot r$. Then* $\operatorname{bu}(q, s \centerdot r) = \operatorname{bu}(rq, s)\big|_r$.

Lemma 5.10. *Let s be a term and pb, $b \in \mathbf{B}_a$, a binding position of s with $s \centerdot pb = x$. Let f be a placing rule such that* $\operatorname{dom} f$ *contains neither binding positions nor prefixes of p, and x occurs free in $f(r)$ for no $r \in \operatorname{dom} f$. Then* $\operatorname{bu}(pb, [f](s)) = \operatorname{bu}(pb, s) \setminus \bigcup_{r\in\operatorname{dom} f}\{q : r \leq q\}$.

Lemma 5.11. *Let s, t be terms such that $(s, t) \in$ A.*

(i) *Terms s and t have same binding units.*

(ii) *For every variable x,* $\operatorname{fpos}_x s = \operatorname{fpos}_x t$.

We end the section with proving three propositions using the facts stated so far.

Proposition 5.12. *Relation* A *has the replacement property.*

Proof. Take a term s and a position $r \in \operatorname{pos} s$. Assume $(s \bullet r, u) \in \mathrm{A}$, i.e. $u = \operatorname{ren}_{pb \to z}(s \bullet r)$ for some binding position pb, $b \in \mathbf{B}_a$, of $s \bullet r$ and z not occurring in $s \bullet r \bullet pa = s \bullet rpa$. Now

$$\begin{aligned}
[r \mapsto u](s) &= [r \mapsto \operatorname{ren}_{pb \to z}(s \bullet r)](s) \\
&= [r \mapsto [z : q \in \operatorname{bu}(pb, s \bullet r)](s \bullet r)](s) && \text{(by 5.7 (i))} \\
&= [r \mapsto [z : q \in \operatorname{bu}(rpb, s)\big|_r](s \bullet r)](s) && \text{(by 5.9)} \\
&= [z : q \in \operatorname{bu}(rpb, s)](s) && \text{(by 4.10)} \\
&= \operatorname{ren}_{rpb \to z}(s) && \text{(by 5.7 (i))}
\end{aligned}$$

gives the desired result. □

Proposition 5.13. *Let s be a term and q its binding position. Let σ be a type-preserving mapping of variables to terms such that applying $\lfloor \sigma \rfloor$ to s is sound. Then* $\operatorname{bu}(q, \lfloor \sigma \rfloor (s)) = \operatorname{bu}(q, s)$.

Proof. Theorem 4.5 implies $\operatorname{pos} s \subseteq \operatorname{pos} \lfloor \sigma \rfloor (s)$. Thus $\lfloor \sigma \rfloor (s)$ has the same binding positions and the same roots of bindings as s plus maybe some more which do not belong to $\operatorname{pos} s$. For any $r \asymp \operatorname{fpos} s$, among the rest for each $r \in \operatorname{bpos} s$, we have $\lfloor \sigma \rfloor (s) \bullet r = s \bullet r$. Consequently $\operatorname{bu}(q, s) \subseteq \operatorname{bu}(q, \lfloor \sigma \rfloor (s))$ for every binding position $q \in \operatorname{pos} s$, and any position being divergent from $\operatorname{fpos} s$ belongs to $\operatorname{bu}(q, s)$ iff it belongs to $\operatorname{bu}(q, \lfloor \sigma \rfloor (s))$. Consider the case $r \not\asymp \operatorname{fpos} s$ now. Take $p \in \operatorname{fpos} s$ such that $r \not\asymp p$. If $r < p$, then $r \notin \operatorname{bu}(q, s)$, as well as $r \notin \operatorname{bu}(q, \lfloor \sigma \rfloor (s))$, since binding units contain maximal positions only. If $r \geq p$, then $r \notin \operatorname{bu}(q, s)$, but $r \notin \operatorname{bu}(q, \lfloor \sigma \rfloor (s))$ either because otherwise the variable $s \bullet q$ would occur free at $r\big|_p$ in $\sigma(s \bullet p)$ contradicting the soundness. Hence the claim follows. □

Proposition 5.14. *Let s be a term and $p \in \operatorname{pos} s$. Let σ be a type-preserving mapping of variables to terms. Then $\lfloor \sigma \rfloor (s) \bullet p = \lfloor \sigma' \rfloor (s \bullet p)$ where*

$$\sigma'(x) = \begin{cases} \sigma(x), & \textit{if } x \in \operatorname{supp} \sigma \textit{ and } x \textit{ is free at } p \textit{ in } s, \\ x & \textit{otherwise.} \end{cases}$$

Proof. Proposition 5.3 (ii) implies that a set of positions where a certain variable is free is closed w.r.t. taking prefixes. This gives that $\sigma'(s \bullet pr) = \sigma(s \bullet pr)$ for each $r \in \operatorname{fpos} s\big|_p$. Denote by R the set consisting of all positions where some variable not being free at p in s occurs free in $s \bullet p$. Then $\sigma'(s \bullet pr) = s \bullet pr$ for each $r \in R$. We have

$$\begin{aligned}
\lfloor \sigma \rfloor (s) \bullet p &= [\sigma(s \bullet r) : r \in \operatorname{fpos} s](s) \bullet p && \text{(by 5.6 (ii))} \\
&= [\sigma(s \bullet pr) : r \in \operatorname{fpos} s\big|_p](s \bullet p) && \text{(by 4.7 (ii))} \\
&= [\sigma(s \bullet pr) : r \in \operatorname{fpos} s\big|_p]([s \bullet pr : r \in R](s \bullet p)) && \text{(by 4.6)} \\
&= [\sigma'(s \bullet pr) : r \in \operatorname{fpos} s\big|_p]([\sigma'(s \bullet pr) : r \in R](s \bullet p)) \\
&= [\sigma'(s \bullet pr) : r \in \operatorname{fpos} s\big|_p \cup R](s \bullet p) && \text{(by 4.8)} \\
&= [\sigma'(s \bullet pr) : r \in \operatorname{fpos}(s \bullet p)](s \bullet p) && \text{(by 5.5)} \\
&= \lfloor \sigma' \rfloor (s \bullet p). && \text{(by 5.6 (ii))}
\end{aligned}$$

□

6 Investigating α-congruence

6.1 Expressing α as reflexive transitive closure

Lemma 6.1. *Let s be a term and $p_1b_1, \ldots, p_nb_n$, $n > 0$, different binding positions in s, $b_i \in \mathbf{B}_{a_i}$ for each $i = 1, \ldots, n$. Let ϱ be a type-preserving injective on its support mapping of variables to variables such that $s \mathbin{.} p_ib_i \in \operatorname{supp} \varrho$ and $\varrho(s \mathbin{.} p_ib_i)$ does not occur in $s \mathbin{.} p_ia_i$ for any $i = 1, \ldots, n$. Let f be the placing rule with $\operatorname{dom} f = \bigcup_{i=1}^{n} \operatorname{bu}(p_ib_i, s)$ and $f(r) = \varrho(s.r)$ for all $r \in \operatorname{dom} f$. Then $(s, [f](s)) \in \mathrm{A}^t$.*

Proof. Without loss of generality, assume the ordering $p_1, \ldots, p_n$ being such that, for all $i, j = 1, \ldots, n$, $p_ia_i < p_ja_j$ implies $i < j$. Denote $x_i = s \mathbin{.} p_ib_i$ and $z_i = \varrho(x_i)$. Define $s_0 = s$ and $s_{i+1} = \operatorname{ren}_{p_{i+1}b_{i+1} \to z_{i+1}}(s_i)$ for all $i = 0, \ldots, n-1$.

Prove now that $(s_i, s_{i+1}) \in \mathrm{A}$ for all $i = 0, \ldots, n-1$, i.e. z_{i+1} does not occur in $s_i \mathbin{.} p_{i+1}a_{i+1}$. Suppose the contrary, i.e. $s_i \mathbin{.} p_{i+1}a_{i+1} \mathbin{.} r = z_{i+1}$ for some r. Assume thereby that i is the least number for which such situation arises. This ensures that terms $s_0, \ldots, s_i$ have same binding units by Lemma 5.11 (i). Since $s \mathbin{.} p_{i+1}a_{i+1}r \neq z_{i+1}$ by conditions, we can find the biggest k such that $s_k \mathbin{.} p_{i+1}a_{i+1}r \neq z_{i+1}$. Therefore $p_{i+1}a_{i+1}r$ belongs to the binding unit corresponding to $p_{k+1}b_{k+1}$. This implies $z_{i+1} = z_{k+1}$ giving $x_{i+1} = x_{k+1}$ by the injectivity of ϱ.

If $p_{i+1}a_{i+1}r$ is a binding position, then $p_{i+1}a_{i+1}r = p_{k+1}b_{k+1}$ is the only possibility. Since $p_{i+1}a_{i+1}$ is not binding, it must be $r = p'b_{k+1}$ for some p'. So $p_{i+1}a_{i+1} \leq p_{i+1}a_{i+1}p' = p_{k+1} < p_{k+1}a_{k+1}$ giving $i < k$, a contradiction. If $p_{i+1}a_{i+1}r$ is not binding, then it is bound by the occurrence at $p_{k+1}b_{k+1}$, so $p_{k+1}a_{k+1}$ is the longest prefix of $p_{i+1}a_{i+1}r$ being a root of binding of x_{i+1}. Thus $p_{i+1}a_{i+1} \leq p_{k+1}a_{k+1}$. Strict inequality is impossible as earlier, so $p_{i+1}a_{i+1} = p_{k+1}a_{k+1}$. But this implies b_{i+1} and b_{k+1} both belonging to $\mathbf{B}_a$ for $a = a_{i+1} = a_{k+1}$. By the restriction imposed on term construction rules in Sect. 2, these positions cannot bind the same variable, a contradiction.

It remains to prove $s_n = [f](s)$. We have

$$\begin{aligned} s_{i+1} &= \operatorname{ren}_{p_{i+1}b_{i+1} \to z_{i+1}}(s_i) = \\ &= [z_{i+1} : r \in \operatorname{bu}(p_{i+1}b_{i+1}, s_i)](s_i) = \\ &= [z_{i+1} : r \in \operatorname{bu}(p_{i+1}b_{i+1}, s)](s_i). \end{aligned}$$

So s_n is expressed as an application of a composition of replacement operators to $s_0 = s$. This composition equals to $[f]$ by Theorem 4.8 which applies since positions of different binding units of the same term are divergent. □

Corollary 6.2. *Let s be a term. Let ϱ be a type-preserving injective on its support mapping of variables to variables such that $\varrho(x)$ does not occur in s for any $x \in \operatorname{supp} \varrho$. Then $(s, \lceil \varrho \rceil (s)) \in \mathrm{A}^{rt}$.*

Proof. If the variables of $\operatorname{supp} \varrho$ do not occur binding in s, then $\lceil \varrho \rceil (s) = s$ and we have done. Otherwise let $p_1b_1, \ldots, p_nb_n$ be the binding positions where variables of $\operatorname{supp} \varrho$ occur in s. Define f as in the formulation of Lemma 6.1. Then Lemma 6.1 applies and also the result follows since $\lceil \varrho \rceil (s) = [f](s)$. □

Lemma 6.3. *Let s be a term and pb its binding position, $b \in \mathbf{B}_a$. Let z be a variable having the same type as x and not occurring in $s \,.\, pa$. Let f be the placing rule with* $\operatorname{dom} f = \{pb\} \cup \{paq : s \,.\, paq = x\}$ *and $f(r) = z$ for all $r \in \operatorname{dom} f$. Let g be the placing rule with* $\operatorname{dom} g = \operatorname{dom} f$ *and $g(r) = x$ for all $r \in \operatorname{dom} g$. Denote $u = [f](s)$ and $t = \operatorname{ren}_{pb \to z}(s)$.*

(i) $\operatorname{dom} g = \{pb\} \cup \{paq : u \,.\, paq = z\}$ *and $s = [g](u)$, whereby x does not occur in $u \,.\, pa$.*

(ii) $(s,t) \in \mathrm{A}$, $(t,u) \in \mathrm{A}^t$.

(iii) $(u,s) \in \mathrm{A}^t$.

Proof. (i) We must show that $s \,.\, paq = x \iff u \,.\, paq = z$. It suffices to consider positions paq maximal in s (hence also in u). If $s \,.\, paq = x$, then $paq \in \operatorname{dom} f$, therefore $u \,.\, paq = [f](s) \,.\, paq = f(paq) = z$. If $s \,.\, paq \neq x$, then $paq \notin \operatorname{dom} f$ which implies $paq \asymp \operatorname{dom} f$ because paq and the positions of $\operatorname{dom} f$ are all maximal. Thus $u \,.\, paq = [f](s) \,.\, paq = s \,.\, paq \notin \{x, z\}$. This proves also that x does not occur in $u \,.\, pa$. Now $[g](u) = [g]([f](s)) = [g](s) = s$ by Theorems 4.9 and 4.6.

(ii) Any occurrence of x at a position of $\operatorname{dom} f$ neither is free nor is bound by an occurrence at a position outside $\operatorname{dom} f$. Hence $\operatorname{dom} f$ partitions into binding units of x, whereby all the roots of binding of them are refinements of pa, so pa is the least among them. Since z does not occur in $s \,.\, pa$, it does not occur in $s \,.\, par$ for any root of binding par. Defining ϱ with $\operatorname{supp} \varrho = \{x\}$, $\varrho(x) = z$, Lemma 6.1 gives $(s,u) \in \mathrm{A}^t$, whereby the proof of it gives more precisely $(s,t) \in \mathrm{A}$ and $(t,u) \in \mathrm{A}^t$.

(iii) Part (i) of this lemma proves the assumptions of this lemma for the case of taking u, s, z and g at place of s, u, x and f, respectively. Part (ii) gives then $(u,s) \in \mathrm{A}^t$. □

Claim 6.4. *Let s,t be terms. If $(s,t) \in \mathrm{A}$, then $(t,s) \in \mathrm{A}^t$.*

Proof. If $(s,t) \in \mathrm{A}$, then $t = \operatorname{ren}_{qb \to z}(s)$ for some binding position qb, $b \in \mathbf{B}_a$, of s and variable z not occurring in $s \,.\, pa$. Denote $x = s \,.\, pb$ and let f be the placing rule with $\operatorname{dom} f = \{pb\} \cup \{paq : s \,.\, paq = x\}$ and $f(r) = z$ for all $r \in \operatorname{dom} f$. If we take $u = [f](s)$, then Lemma 6.3 gives $(t,u) \in \mathrm{A}^t$ and $(u,s) \in \mathrm{A}^t$. Hence $(t,s) \in \mathrm{A}^t$ by transitivity. □

Theorem 6.5. $\alpha = \mathrm{A}^{rt}$.

Proof. We have $\mathrm{A}^{rt} \subseteq \alpha$ by definition of α. For the opposite inclusion, we must show that A^{rt} is a congruence. As it is reflexive and transitive, it suffices to show symmetry and compatibility. Compatibility follows from Theorems 5.12 and 4.12 (iii), (iv). For symmetry, apply Claim 6.4 iteratively. □

Using this theorem, we can simply generalize some facts about relation A to α-congruence as in the following corollary.

Corollary 6.6. *Let s,t be α-congruent terms. Then:*

(i) $\operatorname{pos} s = \operatorname{pos} t$;

(ii) *s and t have same binding units;*

(iii) *for any variable x,* $\operatorname{fpos}_x s = \operatorname{fpos}_x t$;

(iv) $\operatorname{fv} s = \operatorname{fv} t$.

Corollary 6.6 can be proved also without using the theory of this subsection. For (i), construct an algebra in our signature such that pos appears to be a homomorphism from $\mathcal{T}$ to it. Then the kernel relation of pos is a congruence containing A. Now use the definition of α. The other equalities can be proved analogously.

Theorem 6.5 helps also to prove the following fact.

Claim 6.7. *Let s, t be α-congruent terms. Then $s \centerdot q = t \centerdot q$ for all binding positions q implies $s = t$.*

Proof. Let P be the set of positions maximal in s (so also in t), and take $p \in P$. If $p \in \mathrm{bu}(q, s)$ for some q (so also $p \in \mathrm{bu}(q, t)$), then $s \centerdot p = s \centerdot q = t \centerdot q = t \centerdot p$. Otherwise, no renaming steps of the sequence transforming s to t influence position p, so $s \centerdot p = t \centerdot p$. By Theorem 3.5, $s = t$. □

6.2 Substitutivity

At this point, it is inevitable to make one more assumption about our algebra. Namely, assume in the rest that there is infinitely many variables of each type τ_b where b is any binding index. This is a standard restriction which guarantees that we can rename a bound variable with a fresh one whenever necessary. The following proposition states this in more detail.

Proposition 6.8. (i) *Let $\mathcal{Y}$ be any finite subsystem of $\mathcal{X}$ (i.e. $\mathcal{Y} = (\mathcal{Y}_\gamma : \gamma \in \Gamma)$ where $\mathcal{Y}_\gamma \subseteq \mathcal{X}_\gamma$ for each $\gamma \in \Gamma$). Then each term is α-congruent to some term in which no variables of $\mathcal{Y}$ occur binding.*

(ii) *Whenever we have a finite family of terms, there exists a family of α-congruent to them, respectively, terms such that no variable occurs both free and binding in these terms.*

(iii) *Let σ be any type-preserving mapping of variables to terms. Then each term s is α-congruent to some term t such that applying $\lfloor \sigma \rfloor$ to t is sound.*

Proof. (i) Let s be any term. As $\mathcal{Y}$ is finite, we can find a type-preserving injective on its support mapping ϱ of variables to variables with $\operatorname{supp} \varrho = \mathcal{Y}$ such that $\varrho(y)$ neither belongs to $\mathcal{Y}$ nor occurs in s for each $y \in \mathcal{Y}$. Now apply Corollary 6.2.

(ii) Take $\mathcal{Y}$ to be the system of all variables which occur free in some of the given terms. Then apply (i) for each of the terms.

(iii) Take $\mathcal{Y}$ to be the system of all variables which occur free in $\sigma(x)$ for some $x \in \operatorname{fv} s \cap \operatorname{supp} \sigma$. Then apply (i) for s. □

Lemma 6.9. *Let s be a term and pb, $b \in \mathbf{B}_a$, its binding position. Denote $x = s \centerdot pb$ and take a variable z of the same type as x not occurring in $s \centerdot pa$. Denote $t = \mathrm{ren}_{pb \to z}(s)$. Let f be any placing rule such that $\operatorname{dom} f$ does not contain binding positions and, for each $r \in \operatorname{dom} f$, x does not occur free in $f(r)$.*

(i) *If z does not occur in $f(r)$ for any $r \in \operatorname{dom} f$, then $([f](s), [f](t)) \in \mathrm{A}^r$.*

(ii) *If z does not occur free in $f(r)$ for any $r \in \operatorname{dom} f$, then $([f](s), [f](t)) \in \alpha$.*

Proof. If $\operatorname{dom} f$ contains a prefix of p, then Theorem 4.9 gives $[f](t) = [f](s)$ from which both parts follow. Assume further no prefixes of p belonging to $\operatorname{dom} f$.

(i) It suffices to prove $[f](t) = \mathrm{ren}_{pb\to z}([f](s))$ since z is fresh by conditions. Take arbitrary $q \in \operatorname{dom} f \cap \operatorname{pos} s$ and $r \in \mathrm{bu}(pb, [f](s))$. If $r < q$, then r is not a maximal position of $[f](s)$ (since also $q \in \operatorname{pos}[f](s))$ and hence cannot belong to $\mathrm{bu}(pb, [f](s))$, a contradiction. Consider the case $q \leq r$. Taking into account that q is not a prefix of p, we have that $q = r = pb$ or $pa \leq q$. The former case cannot arise because of $\operatorname{dom} f$ not containing binding positions. The case $pa \leq q$ leads to $f(q)\centerdot r\big|_q = [f](s)\centerdot r = x$, whereby this occurrence of x is free in $f(q)$ since otherwise r could not belong to the binding unit of pb in $[f](s)$. As this is also excluded by the conditions, we have as the only possibility that $q \asymp r$. We conclude from all this that any position of $\operatorname{dom} f \cap \operatorname{pos} s$ is divergent from any position of $\mathrm{bu}(pb, [f](s))$.

Denote by $P(r)$ the assertion "$\operatorname{dom} f \cap \operatorname{pos} s$ contains a prefix of r". Implicitly using $[f](s) = [f\big|_{\operatorname{pos} s}](s)$ and $\operatorname{pos}[z : r \in \mathrm{bu}(pb, s), \neg P(r)](s) = \operatorname{pos} s$, we have

$$\begin{aligned}
[f](t) &= [f](\mathrm{ren}_{pb\to z}(s)) & \\
&= [f]\left([z : r \in \mathrm{bu}(pb, s)](s)\right) & \text{(by 5.7 (i))}\\
&= [f]\left([z : r \in \mathrm{bu}(pb, s), P(r)]\left([z : r \in \mathrm{bu}(pb, s), \neg P(r)](s)\right)\right) & \text{(by 4.8)}\\
&= [f]\left([z : r \in \mathrm{bu}(pb, s), \neg P(r)](s)\right) & \text{(by 4.9)}\\
&= [f]\left([z : r \in \mathrm{bu}(pb, [f](s))](s)\right) & \text{(by 5.10)}\\
&= [z : r \in \mathrm{bu}(pb, [f](s))]([f](s)) & \text{(by 4.8)}\\
&= \mathrm{ren}_{pb\to z}([f](s)). & \text{(by 5.7 (i))}
\end{aligned}$$

(ii) Find a placing rule f' with $\operatorname{dom} f = \operatorname{dom} f'$ such that, for any $r \in \operatorname{dom} f$, $(f(r), f'(r)) \in \alpha$ and z does not occur in $f'(r)$. Theorem 3.5 gives $([f](s), [f'](s)) \in \alpha$ and $([f](t), [f'](t)) \in \alpha$. By (i), $([f'](s), [f'](t)) \in \mathrm{A}$, so $([f](s), [f](t)) \in \alpha$. □

Lemma 6.10. *Let s, t be terms, $(s, t) \in \mathrm{A}$. Let ϱ be any type-preserving injective on its support mapping of variables to variables such that, for any variable $x \in$* $\operatorname{supp} \varrho$, *$\varrho(x)$ occurs in neither s nor t. Then* $(\lceil\varrho\rceil(s), \lceil\varrho\rceil(t)) \in \mathrm{A}$.

Proof. Let $t = \mathrm{ren}_{pb\to z}(s)$ for binding position pb, $b \in \mathbf{B}_a$, of s, and z not occurring in $s\,.\,pa$. By Corollary 6.2, we have $(s, \lceil\varrho\rceil(s)) \in \alpha$, as well as $(t, \lceil\varrho\rceil(t)) \in \alpha$. This gives $(\lceil\varrho\rceil(s), \lceil\varrho\rceil(t)) \in \alpha$.

Prove now that $\varrho(z)$ does not occur in $\lceil\varrho\rceil(s)\,.\,pa$. Suppose the contrary, i.e. $\lceil\varrho\rceil(s)\,.\,par = \varrho(z)$ for some r. Denote $y = s\,.\,par$, then $z \neq y$. Dependently on whether this occurrence of y is free or not in s, we have either $\varrho(z) = s\,.\,par = y$ or $\varrho(z) = \varrho(s\,.\,par) = \varrho(y)$. Observe that neither $\varrho(y) = z$ nor $\varrho(z) = y$ is possible because of the condition about ϱ in the formulation. So the former case cannot arise. The latter leads to exactly one of y, z belonging to $\operatorname{supp} \varrho$. But this gives either $\varrho(y) = \varrho(z) = z$ or $\varrho(z) = \varrho(y) = y$, contradicting anyway.

Hence $(\lceil\varrho\rceil(t), \mathrm{ren}_{pb\to\varrho(z)}(\lceil\varrho\rceil(s))) \in \alpha$. We show actually that $\lceil\varrho\rceil(t) = \mathrm{ren}_{pb\to\varrho(z)}(\lceil\varrho\rceil(s))$. By Claim 6.7, it remains to prove that these terms have same variables at binding positions. For position pb, we have $\mathrm{ren}_{pb\to\varrho(z)}(\lceil\varrho\rceil(s))\,.\,pb = \varrho(z) = \varrho(t\centerdot pb) = \lceil\varrho\rceil(t)\centerdot pb$. For arbitrary binding position $q \neq pb$, q does not belong to the binding unit corresponding to pb. So $\mathrm{ren}_{pb\to\varrho(z)}(\lceil\varrho\rceil(s))\centerdot q = \lceil\varrho\rceil(s)\centerdot q = \varrho(s\centerdot q) = \varrho(t\centerdot q) = \lceil\varrho\rceil(t)\centerdot q$. □

Lemma 6.11. *Let $(s,t) \in \alpha$. Let $\mathcal{Y}$ be any finite subsystem of $\mathcal{X}$ such that any variable of $\mathcal{Y}$ occurs binding in neither s nor t. Then there exist terms $s = u_0, u_1, \ldots, u_n = t$ such that, for each $i = 0,\ldots,n-1$, $(u_i, u_{i+1}) \in \mathrm{A}$ and any variable of $\mathcal{Y}$ does not occur binding in u_i.*

Proof. Since $\alpha = \mathrm{A}^{rt}$, we can find terms $v_0,\ldots,v_n$ such that $v_0 = s$, $v_n = t$ and $(v_i, v_{i+1}) \in \mathrm{A}$ for all $i = 0,\ldots,n-1$. Define a type-preserving injective on its support mapping ϱ of variables to variables such that
(1) $x \in \operatorname{supp}\varrho$ iff x is in $\mathcal{Y}$ and x occurs binding in some v_i,
(2) $x \in \operatorname{supp}\varrho$ implies $\varrho(x)$ not being in $\mathcal{Y}$ and not occurring in terms $v_0,\ldots,v_n$.
Define $u_i = \lceil\varrho\rceil(v_i)$ for all $i = 0,\ldots,n-1$. Consider arbitrary variable y from $\mathcal{Y}$, suppose it occurring binding in u_i. If y occurs at the same position also in v_i then $\varrho(y) = y$ which contradicts with the choice of ϱ (item (1)). Otherwise $y = \varrho(x)$ for some $x \neq y$ which also contradicts to the choice of ϱ (item (2)). Thus variables of $\mathcal{Y}$ do not occur binding in terms u_i. Lemma 6.10 gives $(u_i, u_{i+1}) \in \mathrm{A}$ for each $i = 0,\ldots,n-1$, so we have done. □

Theorem 6.12. *Let s,t be any α-congruent terms. Let f be any placing rule such that* $\operatorname{dom} f$ *contains no binding positions and, for any $p \in \operatorname{dom} f \cap \operatorname{pos} s$, variables occurring free in $f(p)$ occur binding in neither s nor t. Then $([f](s), [f](t)) \in \alpha$.*

Proof. Let $\mathcal{Y}$ be the system of variables occurring free in terms $f(p)$, $p \in \operatorname{dom} f \cap \operatorname{pos} s$. By conditions, variables of $\mathcal{Y}$ occur binding in neither s nor t. Take $u_0,\ldots,u_n$ whose existence is claimed by Lemma 6.11. By Lemma 6.9, we have $([f](u_i), [f](u_{i+1})) \in \alpha$ for each $i = 0,\ldots,n-1$. Thus $([f](s), [f](t)) \in \alpha$. □

Theorem 6.13. *Let s,t be any α-congruent terms. Let σ be any type-preserving mapping of variables to terms such that applying $\lfloor\sigma\rfloor$ to both s and t is sound. Then $(\lfloor\sigma\rfloor(s), \lfloor\sigma\rfloor(t)) \in \alpha$.*

Proof. We can express $\lfloor\sigma\rfloor(s) = [f](s)$ and $\lfloor\sigma\rfloor(t) = [f](t)$ for some f since $\mathrm{fpos}_x(s) = \mathrm{fpos}_x(t)$ for all variables x. Now apply Theorem 6.12. □

Theorem 6.13 states that different instances of an α-class are equivalent w.r.t. sound application of substitution to them. (This property is called *substitutivity* of α-congruence.) This means that the frequent practice to identify α-congruent terms is mathematically justified indeed. On the other hand, it enables to define substitution as an operation on α-classes. We go this way in the next section. The following theorem is inevitable for this.

Theorem 6.14. *Let s,t be terms, $(s,t) \in \alpha$. Let σ_1, σ_2 be type-preserving mappings of variables to terms such that $(\sigma_1(x), \sigma_2(x)) \in \alpha$ for any variable x. If applying $\lfloor\sigma_1\rfloor$ to both s and t is sound, then $(\lfloor\sigma_1\rfloor(s), \lfloor\sigma_2\rfloor(t)) \in \alpha$.*

Proof. For any variable x, we have $\operatorname{fv}\sigma_1(x) = \operatorname{fv}\sigma_2(x)$ since $(\sigma_1(x), \sigma_2(x)) \in \alpha$. As no renaming steps can be made in a pure variable, each variable is α-congruent to no other term. This leads to $\operatorname{supp}\sigma_1 = \operatorname{supp}\sigma_2$. Thus applying $\lfloor\sigma_2\rfloor$ to both s and t is sound, too. Now $(\lfloor\sigma_1\rfloor(s), \lfloor\sigma_2\rfloor(s)) \in \alpha$ by Theorem 3.5 (for the proof, take $P = \operatorname{fpos} s$ in the formulation of Theorem 3.5). By Theorem 6.13, we have $(\lfloor\sigma_2\rfloor(s), \lfloor\sigma_2\rfloor(t)) \in \alpha$. Transitivity gives the required claim. □

7 Substitution on α-classes

We denote s/α for the α-class of term s. Note that Corollary 6.6 allows us to extend the functions pos, fpos_x and fv naturally to α-classes. It is also reasonable to speak about binding units of α-classes although the variable which occurs in the positions of a binding unit is not determined.

As each variable form a separate α-class, we make no difference between variables and their α-classes.

Definition 7.1. Let $\boldsymbol{s}$ be any α-class of terms. Let $\boldsymbol{\sigma}$ be any type-preserving mapping of variables to α-classes of terms. Then define $\lfloor\boldsymbol{\sigma}\rfloor(\boldsymbol{s}) = \lfloor\sigma\rfloor(s)/\alpha$ where $s \in \boldsymbol{s}$ and σ, $\sigma(x) \in \boldsymbol{\sigma}(x)$ for any variable x, are chosen in such a way that applying $\lfloor\sigma\rfloor$ to s is sound.

The value of $\lfloor\sigma\rfloor(s)/\alpha$ does not depend on the choices made in the definition due to Theorem 6.14. A strong point of this "substitution modulo α" is that substitutions apply legally to any α-class, no worrying about variable captures is needed.

Definition 7.2. Let $\boldsymbol{s}$ be a congruence class of terms and q a position.

(i) Say that q is *significant* for $\boldsymbol{s}$ iff $s \bullet q$ belongs to the same congruence class independently on the choice of $s \in \boldsymbol{s}$.

(ii) If q is significant for $\boldsymbol{s}$, then write $\boldsymbol{s} \bullet q$ for the only class which contains the terms $s \bullet q$, $s \in \boldsymbol{s}$.

(iii) Denote by $\operatorname{sigpos}\boldsymbol{s}$ the set of all positions significant for $\boldsymbol{s}$.

Lemma 7.3. *Let $\boldsymbol{s}$ be an α-class and $p \in \operatorname{pos}\boldsymbol{s}$. Then $p \notin \operatorname{sigpos}\boldsymbol{s}$ iff there exists a binding position $q \in \operatorname{pos}\boldsymbol{s}$ such that p is not a proper prefix of q and p is a prefix of some $r \in \operatorname{bu}(q, \boldsymbol{s})$.*

Proof. ($\Rightarrow$) Assume $p \notin \operatorname{sigpos}\boldsymbol{s}$. Take $s, t \in \boldsymbol{s}$ such that $(s \bullet p, t \bullet p) \notin \alpha$. By Theorem 6.5, find $s = u_0, u_1, \ldots, u_n = t$ such that $(u_i, u_{i+1}) \in \mathrm{A}$ for each $i = 0, \ldots, n-1$. There must exist i such that $(u_i \bullet p, u_{i+1} \bullet p) \notin \alpha$. Take q and z such that $u_{i+1} = \operatorname{ren}_{q \to z}(u_i)$. As $u_i \bullet p \neq u_{i+1} \bullet p$, an $r \in \operatorname{bu}(q, \boldsymbol{s})$ must exist such that $p \not\succ \operatorname{bu}(q, \boldsymbol{s})$. It cannot be $r < p$ since r is maximal in u_i. Thus $p \leq r$. It remains to show that p is not a proper prefix of q. If it were, then $q|_p$ would be a binding position in both $u_i \bullet p$ and $u_{i+1} \bullet p$. We would get

$$\begin{aligned}
u_{i+1} \bullet p &= [z : r \in \operatorname{bu}(q, u_i)](u_i) \bullet p && \text{(by 5.7 (i))}\\
&= [z : r \in \operatorname{bu}(q, u_i)|_p](u_i \bullet p) && \text{(by 4.7 (ii))}\\
&= [z : r \in \operatorname{bu}(q|_p, u_i \bullet p)](u_i \bullet p) && \text{(by 5.9)}\\
&= \operatorname{ren}_{q|_p \to z}(u_i \bullet p) && \text{(by 5.7 (i))}
\end{aligned}$$

contradicting $(u_i \bullet p, u_{i+1} \bullet p) \notin$ A.

($\Leftarrow$) Take $s \in \boldsymbol{s}$ and denote $x = s \bullet q$. Define $t = \mathrm{ren}_{q \to z}(s)$ where z is fresh, so also $t \in \boldsymbol{s}$. Then in terms $s \bullet p$ and $t \bullet p$, variables x and z, respectively, occur at position $r\big|_p$. These occurrences are free because the position q binding r in s would otherwise be a proper refinement of p contradicting the assumption. Hence $(s \bullet p, t \bullet p) \notin \alpha$ giving $p \notin \mathrm{sigpos}\, \boldsymbol{s}$. □

Lemma 7.3 implies that $\mathrm{sigpos}\, \boldsymbol{s}$ can actually be determined by any $s \in \boldsymbol{s}$. As another implication, we have $\mathrm{fpos}\, \boldsymbol{s} \subseteq \mathrm{sigpos}\, \boldsymbol{s}$ for any $\boldsymbol{s}$.

Proposition 7.4. *Let $\boldsymbol{\sigma}$ be a type-preserving mapping of variables to α-classes of terms, and $\boldsymbol{s}$ be an α-class. Assume $q \in \mathrm{sigpos}\, \boldsymbol{s}$. Then $q \in \mathrm{sigpos}\, \lfloor \boldsymbol{\sigma} \rfloor (\boldsymbol{s})$ whereby $\lfloor \boldsymbol{\sigma} \rfloor (\boldsymbol{s}) \bullet q = \lfloor \boldsymbol{\sigma} \rfloor (\boldsymbol{s} \bullet q)$.*

Proof. Take $s \in \boldsymbol{s}$ and σ, $\sigma(x) \in \boldsymbol{\sigma}(x)$ for each variable x, such that applying $\lfloor \sigma \rfloor$ to s is sound. Then $\lfloor \sigma \rfloor (s) \in \lfloor \boldsymbol{\sigma} \rfloor (\boldsymbol{s})$.

Suppose $q \notin \mathrm{sigpos}\, \lfloor \boldsymbol{\sigma} \rfloor (\boldsymbol{s})$. Lemma 7.3 states that there exists a binding position $q' \in \mathrm{pos}\, \lfloor \sigma \rfloor (s)$ such that $q \not< q'$ and q is a prefix of some $r \in \mathrm{bu}(q', \lfloor \sigma \rfloor (s))$. If $q' \notin \mathrm{pos}\, s$, then $p < q'$ for some $p \in \mathrm{fpos}\, s$ by Theorem 4.5 and thus also $p < r$. Now $q \leq p$ since $q \leq r$ and $q \in \mathrm{pos}\, s$, contradicting $q \not< q'$. Hence $q' \in \mathrm{pos}\, s$ which gives $r \in \mathrm{bu}(q', s)$ by Proposition 5.13. But then Lemma 7.3 gives $q \notin \mathrm{sigpos}\, \boldsymbol{s}$, contradicting the assumption. Hence we have proved that $q \in \mathrm{sigpos}\, \lfloor \boldsymbol{\sigma} \rfloor (\boldsymbol{s})$.

By Proposition 5.14, $\lfloor \sigma \rfloor (s) \bullet q = \lfloor \sigma' \rfloor (s \bullet q)$ where σ' works differently from σ on variables $x \in \mathrm{supp}\, \sigma$ only which are not free at q in s. Suppose z being such a variable. Let pb and pa be the binding occurrence and root of binding, respectively, which bind z at q in s. Suppose moreover that z occurs free at some position r in $s \bullet q$. Then pa is the longest prefix of qr being a root of binding of z in s, so $qr \in \mathrm{bu}(pb, s)$. For a fresh variable z', define $t = \mathrm{ren}_{pb \to z'}(s)$. We have clearly $qr \in \mathrm{bu}(pb, t)$ and therefore z' occurs free at r in $t \bullet q$. On one hand, $(s, t) \in \alpha$ by the choice of t. On the other hand, $(s \bullet q, t \bullet q) \notin \alpha$ since the terms have different free variables. This contradicts the significance of q for s.

So $\lfloor \sigma' \rfloor (s \bullet q) = \lfloor \sigma \rfloor (s \bullet q)$ as no variable on which σ and σ' work differently occurs free in $s \bullet q$. Hence $\lfloor \sigma \rfloor (s) \bullet q \in \lfloor \sigma \rfloor (s \bullet q)/\alpha = \lfloor \boldsymbol{\sigma} \rfloor (\boldsymbol{s} \bullet q)$, we have done. □

Theorem 7.5. *Let $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2$ be any type-preserving mappings of variables to α-classes of terms. Then $\lfloor \boldsymbol{\sigma}_1 \rfloor \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor = \lfloor \boldsymbol{\sigma}_1 \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor \rfloor$.*

Proof. We show for any α-class $\boldsymbol{s}$ that $\lfloor \boldsymbol{\sigma}_2 \rfloor (\lfloor \boldsymbol{\sigma}_1 \rfloor (\boldsymbol{s})) = \lfloor \boldsymbol{\sigma}_1 \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor \rfloor (\boldsymbol{s})$.

Take $p \in \mathrm{fpos}\, \boldsymbol{s}$. Then $p \in \mathrm{sigpos}\, \boldsymbol{s}$ whereby $x = \boldsymbol{s} \bullet p$ is a variable. Thus Proposition 7.4 gives $\lfloor \boldsymbol{\sigma}_1 \rfloor (\boldsymbol{s}) \bullet p = \lfloor \boldsymbol{\sigma}_1 \rfloor (\boldsymbol{s} \bullet p) = \lfloor \boldsymbol{\sigma}_1 \rfloor (x) = \boldsymbol{\sigma}_1(x)$ and $\lfloor \boldsymbol{\sigma}_2 \rfloor (\lfloor \boldsymbol{\sigma}_1 \rfloor (\boldsymbol{s})) \bullet p = \lfloor \boldsymbol{\sigma}_2 \rfloor (\lfloor \boldsymbol{\sigma}_1 \rfloor (\boldsymbol{s}) \bullet p) = \lfloor \boldsymbol{\sigma}_2 \rfloor (\boldsymbol{\sigma}_1(x)) = (\boldsymbol{\sigma}_1 \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor)(x) = \lfloor \boldsymbol{\sigma}_1 \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor \rfloor (x) = \lfloor \boldsymbol{\sigma}_1 \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor \rfloor (\boldsymbol{s} \bullet p) = \lfloor \boldsymbol{\sigma}_1 \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor \rfloor (\boldsymbol{s}) \bullet p$.

Take now $q \asymp \mathrm{fpos}\, \boldsymbol{s}$. Take $s \in \boldsymbol{s}$ and σ_1, σ_2, σ_3, $\sigma_1(x) \in \boldsymbol{\sigma}_1(x)$, $\sigma_2(x) \in \boldsymbol{\sigma}_2(x)$, $\sigma_3(x) \in (\boldsymbol{\sigma}_1 \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor)(x)$ for every variable x, such that applying $\lfloor \sigma_1 \rfloor$ and $\lfloor \sigma_3 \rfloor$ to s, as well as $\lfloor \sigma_2 \rfloor$ to $\lfloor \sigma_1 \rfloor (s)$, is sound. So $\lfloor \boldsymbol{\sigma}_2 \rfloor (\lfloor \boldsymbol{\sigma}_1 \rfloor (\boldsymbol{s})) = \lfloor \sigma_2 \rfloor (\lfloor \sigma_1 \rfloor (s))/\alpha$ and $\lfloor \boldsymbol{\sigma}_1 \,;\, \lfloor \boldsymbol{\sigma}_2 \rfloor \rfloor (\boldsymbol{s}) = \lfloor \sigma_3 \rfloor (s)/\alpha$. Clearly $\lfloor \sigma_1 \rfloor$ and $\lfloor \sigma_3 \rfloor$ do not replace at q when applied to s. Also $\lfloor\sigma_2\rfloor$ does not replace at q when applied to $\lfloor\sigma_1\rfloor(s)$ since even if $\lfloor\sigma_1\rfloor(s) \mathbin{.} q = s \mathbin{.} q$ were a variable, it would not be free in s and the same root of binding would bind it also in $\lfloor\sigma_1\rfloor(s)$. Hence $\lfloor\sigma_2\rfloor(\lfloor\sigma_1\rfloor(s)) \mathbin{.} q = s \mathbin{.} q = \lfloor\sigma_3\rfloor \mathbin{.} q$.

Now theorem 3.5 applies and gives the needed result. □

References

[1] Abadi, M., Cardelli, L., Curien, P.-L., Lévy, J.-J.: Explicit substitutions. Journal of Functional Programming **1** (1991) 375–416

[2] Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press (1998)

[3] Barendregt, H. P.: The Lambda Calculus. North-Holland (1984)

[4] Fiore, M., Plotkin, G., Turi, D.: Abstract Syntax and Variable Binding. Proceedings of the 14th Ann. IEEE Symp. in Logic in Computer Science. IEEE CS Press (1999) 193–202

[5] Gabbay, M., Pitts, A.: A New Approach to Abstract Syntax Involving Binders. Proceedings of the 14th Ann. IEEE Symp. in Logic in Computer Science. IEEE CS Press (1999) 214–224

[6] Hindley, J. R.: Basic Simple Type Theory. Cambridge University Press (1997)

[7] Nestra, H.: Handling Substitution without Induction. In: Pilière, C. (ed.): Proceedings of the ESSLLI-2000 Student Session. University of Birmingham (2000) 178–188.

[8] Nestra, H.: A Framework for Studying Substitution. In: Gyimóthy, T. (ed.): Seventh Symposium on Programming Languages and Software Tools. University of Szeged (2001) 184–198

[9] Pfenning, F., Elliott, C.: Higher-Order Abstract Syntax. The ACM SIGPLAN '88 Conference on Programming Language Design and Implementation (1988)

[10] Rosen, B. K.: Tree-Manipulating Systems and Church-Rosser Theorems. Journal of the Association for Computing Machinery **20** (1973) 160–197

[11] Stoughton, A.: Substitution Revisited. Theoretical Computer Science **59** (1988) 317-325

[12] Sun Y.: A Framework for Binding Operators. PhD thesis, LFCS, Edinburgh (1992)

[13] Sun Y.: An algebraic generalization of Frege structures—binding algebras. Theoretical Computer Science **211** (1999) 189-232

[14] Talcott, C.: A theory of binding structures and applications to rewriting. Theoretical Computer Science **112** (1993) 99–143