Epsilon Substitution Method for Elementary Analysis

Grigori Mints
Department of Philosophy, Stanford University, Stanford, CA 94305
Sergei Tupailo
Institute of Cybernetics, Estonian Academy of Sciences, Tallinn, EE0026, Estonia
Wilfried Buchholz
Mathematisches Institut der Universität München,
Theresienstrasse 39, D-80333 München, Bundesrepublik Deutschland

July 25, 1997

Abstract

We formulate epsilon substitution method for elementary analysis $EA$ (second order arithmetic with comprehension for arithmetical formulas with predicate parameters). Two proofs of its termination are presented. One uses embedding into ramified system of level one and cut-elimination for this system. The second proof uses non-effective continuity argument.

Introduction

The epsilon substitution method is based on the language introduced by Hilbert [7] (and used later by Bourbaki [3]). The main non-boolean construction of this language is $exF[x]$, read as "an $x$ satisfying the condition $F[x]"$. In number-theoretic contexts it is often interpreted as least $x$ satisfying $F[x]$. Existential and universal quantifiers become explicitly definable by

$$
\exists x F[x] = F[exF[x]];
\forall x F[x] = F[x\cdot F[x]]
$$

The main axioms of the corresponding formalism are critical formulas

$$
F[x] \rightarrow F[exF[x]]
$$

Hilbert's approach (Ansatz) to transforming arbitrary (non-finitistic) number-theoretic proofs into finitistic (combinatorial) proofs by means of the substitution method is described in [7]. Cf. also the short and lively presentation by Herrmann Weyl in [19]. The approach is as follows.

Take all critical formulas (2) occurring in a given proof $P$. There is only a finite number of them, so one always deals with a finite system $E$ of critical formulas. Consider any substitution $S$ of numerals for constant epsilon-terms. If all critical formulas (2) are true under $S$, it is called a solving substitution for the system $E$. Hilbert proposed a specific plan for finding a solving substitution by a series of successive approximations, described below. If it succeeds and if the last formula of the proof $P$, i.e. the formula proved by $P$, is a constant combinatorial identity such as $1 + 2 + \ldots + 10 = 55$, replacing all free variables by any numerals and then each epsilon-term $t$ by $S(t)$ immediately yields a variable-free (finitistic, combinatorial) proof of the same identity. Moreover, it was noted by Ackermann and stressed later by Kreisel that the same device allows one to extract the numerical content of existential proofs, i.e. proofs of existential formulas $\exists x F[x]$ with combinatorial (free variable) $F[x]$. Indeed, $\exists x F[x]$ is translated as $F[exF[x]]$. If $S$ is a solving substitution for the proof $P$ of such a formula, and $N = S(exF[x])$ then $S(P)$ is a proof of $F[N]$. So $N$ is a numerical realization of the existential quantifier in $\exists x F[x]$.

Hilbert's suggestion for finding a solving substitution by a successive approximation method is based on the following idea of generating substitutions of numerals for closed epsilon-terms. The initial approximation $S_0$ is identically 0. At every stage only a finite number of epsilon-terms are assigned non-zero values.
approximations $S_0, \ldots, S_i$ are already generated, and $S_i$ is not yet a solving substitution, then $S_{i+1}$ is found as follows. Fix appropriate ordering of the critical formulas (2) and take the first formula in this ordering which is false under $S_i$, i.e. for which

$$S_i(F'[,]) = \text{true}, S_i(F[:x]) = \text{false}$$

This means that the value $S_i((x:F))$ is incorrect: this value does not satisfy $F$, while $F[S_i(t)]$ is true under $S_i$. Then the value of $exF$ is corrected by putting

$$S_{i+1}(exF) = \text{(the least } N \leq t) \ (S_i(F[n]) = \text{true})$$

The problem stated by Hilbert was to prove termination of the sequence $S_0, S_1, S_2, \ldots$ after a finite number of steps for any system of critical formulas (2). After von Neumann's [15] attack on this (see below) Hilbert [6] stated further problems:

find a proof of termination for pure number theory, for analysis, and for analysis with the axiom of choice, when each of these systems is suitably reformulated in the epsilon-calculus.

The first attempt by Ackermann [1] to prove termination for analysis was shown by von Neumann [15] to contain a serious defect connected with the treatment of equality. Von Neumann introduced a device allowing to avoid this defect, gave an exact formulation based on Hilbert's Ansatz and presented a termination proof for the case when the terms $exF[x]$ in critical formulas involve only free-variable (combinatorial) formulas $F[x]$. This corresponds to number theory with the induction axiom for quantifier free formulas.

After Gentzen's proof [4] of cut-elimination and consequently of consistency for arithmetic (with respect to closed equations) revealed the role of the ordinal $\varepsilon_0$, Ackermann [2] was able to find a final formulation and to give a termination proof for full first order arithmetic (pure number theory). His formulation for first order arithmetic, described also by Hilbert and Bernays [7], and used in all the subsequent research, consists of Hilbert's Ansatz modified in accordance with von Neumann's remarks plus the following stipulation: after the value of $exF$ was corrected, the values of all terms of greater complexity (rank) are set to 0.

Ackermann's proof [2] is rather involved. On the other hand, there exists much easier non-effective termination proof of the substitution method for the first order arithmetic. It uses a very simple non-effective proof of the existence of a solution of a system of critical formulas from function parameters (cf. [18], section 5.1) and continuous dependence of a system of critical formulas on function parameters (cf. [18], end of the section 5.3).

These ideas were elaborated by Mints [19] for the first order arithmetic and in [11] for the theory of hereditarily finite sets into a non-effective termination proof which uses continuity in Baire topology in the space of number-theoretic functions. A possibility to simplify a system of critical formulas by substituting computable values of subordinate $\varepsilon$-matrices and a need for some restrictions on the functionals was pointed out in [8], p. 259 (ii), 260.

The existence of a solving substitution for the case of analysis (second order arithmetic) is proved as simply as in the first order case, but nothing similar to the simple termination proof is known for the second order case. G. Kreisel [8, 9] further investigated the substitution method for the first order case and made it a basis for the no-counterexample interpretation, the first published functional interpretation producing non-trivial constructive results for non-constructive proofs. W. Tait [18] presented a finer analysis of the rate of termination in terms of relevant parameters based on his analysis [17] of recursion schemata.

The next step was made in [12] where a Gentzen-type system $PAe$ in the epsilon language for first order arithmetic was proposed. The axioms of this calculus depend on a particular system $E$ of critical formulas. Normalization (cut-elimination) steps for this calculus were defined and the following statements proved.

Theorem 1. The sequent expressing existence of a solving substitution for the system $E$ of critical formulas is derivable in $PAe$ by a derivation of special form, called $f$-derivation.

Theorem 2. $f$-derivations can be normalized, i.e. the cut rule can be eliminated by a series of normalization steps.

Theorem 3. A cut-free derivation of the sequent in Theorem 1 encodes a finite sequence

$$S_0, S_1, \ldots, S_n$$

of successive substitutions (in the sense of Ackermann) terminating in a solving substitution for the original system $E$ of critical formulas.
Hence the rate of convergence (i.e., the value of \( n \) in (3) as a function of the system \( E \)) is determined by theorems 3 and 2, and is measured by \( \varepsilon_0 \). The derivation mentioned in Theorem 1 in fact formalizes the non-effective proof of the existence of a solving substitution mentioned earlier. In this paper we apply the proof strategy expressed by Theorems 1-3 to the subsystem of the second order arithmetic based on the comprehension schema for arithmetical formulas with predicate parameters. This system is called \( EA_\varepsilon \) in [16], and is obviously predicative: it can be easily embedded into ramified analysis of the level 1, and its proof-theoretic ordinal is proved in [16] to be \( \varepsilon_0 \). This opens the way to the extension of our approach to ramified systems where bound predicate variables have levels and range over predicate terms of lower levels.

Present approach to the definition of the epsilon substitution for the second order epsilon-terms has two important new features. The values are closed arithmetical abstracts, and the passage to the new substitution is done according to the rank of some intermediate object (canonical form, cf. Section 5). These definitions have been found in the framework of the termination proof via cut-elimination which is presented in the section 6. After this it became possible to give a shorter non-effective proof which is presented in the section 7.

This research was supported by Center for the Study of Language and Information, Stanford University, and by the NSF grant DMS-9200976, and its results with proofs appeared in the preprint [13] by the first two authors. Chapters 2 and 4 of the preprint were written by S. Tupailo. Chapter 3 was due to G. Mints. W. Buchholz wrote a new version of the core parts of the paper with several technical improvements and simplifications.

1 The System \( EA_\varepsilon \)

The formulation here is very close to [7].

1.1 The language \( I_\varepsilon \)

Let us describe in detail the language of \( \varepsilon \)-calculus.

**Basic Symbols**

0-variables (denoted by \( x, y, z, \ldots \)),
1-variables (denoted by \( X, Y, Z, \ldots \)),
the 0-ary function constant 0 (zero), and the unary function constant \( S \) (successor);
the propositional logical connectives \( \land, \lor, \rightarrow \);
the epsilon symbol \( \varepsilon \) and the application symbol App.

**Definition 1** (Terms and formulas)

1. Each \( i \)-variable is a i-term (\( i = 0, 1 \)).
2. The constant 0 is a 0-term.
3. If \( t \) is a 0-term, then \( S t \) is a 0-term.
4. If \( t_1, \ldots, t_n \) are 0-terms and \( p \) is an n-ary predicate constant, then \( pt_1 \ldots t_n \) is a formula.
5. If \( t \) is a 0-term and \( P \) is a 1-term then \( \text{App} t P \) is a formula.
6. If \( A, B \) are formulas then \( \neg A, \land A B, \rightarrow A B \) are formulas.
7. If \( F \) is a formula and \( \xi \) is a i-variable then \( \varepsilon\xi F \) is a i-term (\( i = 0, 1 \)).

To increase readability, we sometimes use infix notation for binary logical connectives, insert parentheses and use standard abbreviations like \( A \lor B = \neg A \rightarrow B \).

**Definition 2** (\( \lambda \)-terms)

If \( G \) is a formula and \( z \) is a 0-variable occurring free in \( G \), then \( \lambda z G \) is a \( \lambda \)-term.

\( i \)-terms (\( i = 0, 1 \)), formulas and \( \lambda \)-terms are called expressions.
Terms of the form \( \varepsilon\xi F \) are called \( \varepsilon \)-terms.
The 0-terms 0, S0, SS0, \ldots are called numerals.
Var denotes the set of all variables, \( \mathbb{N} \) denotes the set of all numerals.
We define \( 0^0 := 0, \quad 0^1 := \lambda z(z = 0). \)
For each term \( u \) we set
\[
\iota(u) := \begin{cases} 
0 & \text{if } u \text{ is a 0-term} \\
1 & \text{if } u \text{ is a 1-term} 
\end{cases}
\]

**Note 1.** Instead of usual computable functions we use computable predicates (their graphs) to simplify technical details. Computable functions can be introduced in a familiar way via their graphs [14].

**Note 2.** Lambda-symbols are not allowed to occur inside other expressions. The result of their substitution into other expressions is understood via \( \lambda \)-conversion (cf. Definitions 6,7 below), which allows to treat them together with 1-terms.

**Syntactic variables:**
- \( e, u, v, w \) for expressions,
- \( \xi, \eta \) for variables,
- \( p \) for any predicate constant and the symbols \( S, \neg, \land, \to, \text{App} \),
- \( s, t \) for 0-terms,
- \( n \) for a numeral \( S^0 \),
- \( P \) for 1-terms,
- \( T \) for 1-terms and \( \lambda \)-terms,
- \( A, B, F, G \) for formulas.

The set \( FV(e) \) of free variables of an expression \( e \) is defined in the standard way.

**Definition 3**
\[
FV(0) := \emptyset \text{ and } FV(\xi) = \{\xi\} \text{ for each variable } \xi, \\
FV(pe_1 \ldots e_n) = FV(e_1) \cup \ldots \cup FV(e_n), \\
FV(\xi F) := FV(F) \setminus \{\xi\}, \\
FV(\lambda zG) := FV(G) \setminus \{z\}.
\]
\( e \) is called closed iff \( FV(e) = \emptyset \)

**Substitution**
We identify expressions which are equivalent modulo renaming of bound variables. If \( u \) is a \( \iota(\xi) \)-term then \( e[\xi/u] \) denotes the result of substituting \( u \) for each free occurrence of \( \xi \) in \( e \), where bound variables in \( e \) are renamed if necessary. If \( \xi \) is known from the context we write \( e[u] \) for \( e[\xi/u] \).

The next series of definitions is needed mainly in the section 3 below to determine computations \( \dot{\varepsilon} \) from inside replacing closed \( e \)-terms by their values. The depth \( d(e) \) is a measure of nesting of closed \( e \)-subterms, taking into account that 1-\( e \)-subterms can be substituted by \( \lambda \)-terms, and this can increase nesting by an arbitrary finite amount. It uses the natural sum function \( \# \) on ordinal numbers, which is a commutative associative analog of the ordinal sum \( + \):
\[
\omega^\alpha \# \omega^\beta = \omega^{\max(\alpha, \beta)} + \omega^{\min(\alpha, \beta)}
\]

**Definition 4**
1. \( d(\xi) := d(0) := 0 \)
2. \( d(pe_1 \ldots e_n) := d(e_1) \# \ldots \# d(e_n), \) for \( p \neq \text{App} \)
3. \( d(\text{App} P) := \omega \cdot (d(P) \# d(t)) \)
4. \( d(\xi F) := \begin{cases} 
1 + d(F) & \text{if } \xi F \text{ is closed} \\
F(F) & \text{otherwise}
\end{cases} 
\)
5. \( d(\lambda zG) := d(G) \)

**Note 1.** \( d(e) < \omega^\omega \).
**Note 2.** \( d(e) = 0 \) iff \( e \) does not contain closed \( e \)-subterms.

**Definition 5**
1. An \( e \)-term \( e = \xi F \) is called canonical if it is closed and \( d(e) = 1 \) (i.e. \( d(F) = 0 \)).
2. A \( \lambda \)-term \( \lambda zG \) is called canonical if \( FV(G) = \{z\} \) and \( d(G) = 0 \).
3. $e$ is called simple if $d(e) = 0$ and $e$ is closed.
4. $e$ is called arithmetical if $e$ contains no 1-variable.
5. TRUE (FALSE) denotes the set of all true (false) simple formulas. [Note that a simple formula contains no variables and is constructed from computable atomic formulas by boolean connectives. Every simple term is a numeral.]
6. $\mathbb{B}_0 := \mathbb{N}; \mathbb{B}_1 :=$ the set of all canonical arithmetical $\lambda$-terms.

The objects to be immediately evaluated are canonical $e$-terms, and the values of $\iota$-terms will be elements of $\mathbb{B}_1$.

To define substitution $e[Y/T]$ for expressions $e$ and $\lambda$-terms $T$, we extend the operation $\text{App}$:

**Definition 6**

$$\text{App}(T, t) := \begin{cases} 
G[t] & \text{if } T = \lambda x G \\
\text{App}Tt & \text{otherwise}
\end{cases}$$

For $p \neq \text{App}$ we set $p(e_1, \ldots, e_n) := p(e_1, \ldots, e_n)$.

**Definition 7**

1. If $e \in \text{Var} \cup \{0\}$ then $e[Y/T] := \begin{cases} 
T & \text{if } e = Y \\
e & \text{otherwise}
\end{cases}$
2. $(e_1 \ldots e_n)[Y/T] := p(e_1[Y/t], \ldots, e_n[Y/t])$
3. $(e\xi)[Y/T] := \begin{cases} 
\xi & \text{if } \xi = Y \text{ where it is assumed without loss of generality that } \xi \notin FV(T) \\
e\xi[Y/T] & \text{otherwise}
\end{cases}$
4. $(\lambda z F)[Y/T] := \lambda z F[Y/T]$ assuming that $z \notin FV(T)$.

The next definition concerns a restriction on the second order critical formulas which arise in the process of translating EA-derivations (Section 2 below) into EAE-derivations.

**Definition 8** (Regular $\lambda$-terms)

A $\lambda$-term $\lambda z G$ is called regular if it can be obtained from a $\lambda$-term without bound 1-variables by substitution of $e$-terms for free variables and $z \in FV(G)$.

### 1.2 Axioms and inference rules of EAE

The language of EAE is L2e.

The only inference rule of EAE is modus ponens: $\frac{F \to G}{G}$.

**Axioms of EAE**

Propositional axioms: all propositional tautologies of the language L2e.

All substitution instances of defining axioms for the predicate constants, including the predicates of addition and multiplication:

- $\text{add}(s, 0, s)$ and $\text{add}(s, t, r) \to \text{add}(s, St, Sr)$
- $\text{prod}(s, 0, 0)$ and $\text{prod}(s, t, r) \land \text{add}(r, s, r_1) \to \text{prod}(s, St, r_1)$
- Equality axioms: $t = t$ and $s = t \to (F[s] \to F[t])$.
- Peano axioms for $S$: $St \neq 0$ and $Ss = St \to s = t$.
- Minimality axioms: $\exists x F[x] = St \to \neg F[t]$.

Critical formulas:

- $F[t] \to F[ex F[x]]$
- $s \neq 0 \to F[ex F[x]]$ with $F := (s = Sx)$.
- $F[t] \to F[ex F[X]]$ with $T$ being a 1-term or a regular $\lambda$-term.

**Comment.** Critical formulas of the second kind are not present in [7]. They are needed here to interpret Robinson’s axiom $s \neq 0 \to \exists x (s = Sx)$.

This concludes the description of EAE. Note that the formulas:

$$s = t \to (ex F[x, s] = ex F[x, t])$$

are consequences of the equality axioms of EAE.

**Note:** EAE is closed under the substitution rule: if $F$ is derivable and $u$ is a $u(\xi)$-term, then $F[x/u]$ is derivable. Indeed, all axioms and inference rules of EAE are closed under substitution.
2 Embedding $EA$ into $EAE$

$EA$ is the usual system of elementary analysis (i.e., second order arithmetic with the axiom scheme of arithmetical comprehension). The language $L_2$ of $EA$ is obtained from $L_2e$ by dropping the epsilon symbol $\epsilon$ and adding the existential quantifier $\exists$. In the definition of terms and formulas case 7 is replaced by:

7. If $F$ is a formula and $\xi$ is a $i$-variable ($i = 0, 1$) then $\exists \xi F$ is a formula.

Note that the only 1-terms of $L_2$ are 1-variables.

The universal quantifier is defined via $\forall$: $\forall F := \neg \exists \xi \neg F$.

Axioms of $EA$

1. Propositional axioms: all propositional tautologies of the language $L_2$.
2. Defining axioms for the predicate constants, including the predicates of addition and multiplication:
   \[
   \begin{align*}
   \text{add}(x, y) &= x + y, \\
   \text{add}(x, y, z) &= \text{add}(x, \text{S}y, z) \\
   \text{prod}(x, y) &= xy, \\
   \text{prod}(x, y, z) &= \text{add}(\text{prod}(x, y), \text{add}(z, x, \text{S}z)) \quad (\text{S}x \neq 0 \text{ and } Sx = S\xi \rightarrow x \neq y).
   \end{align*}
   \]
3. Equality axioms: $x = x$ and $x = y \rightarrow (F[x] \rightarrow F[y])$.
4. Peano axioms for $S$: $Sx \neq 0$ and $Sx = S\xi \rightarrow x = y$.
5. Induction axioms: $F[0] \rightarrow \forall x F[x] \rightarrow F[S\xi] \rightarrow \forall x F[x]$.
6. First order existential axioms: $F[t] \rightarrow \exists \xi F[x]$.
7. Second order existential axioms: $F[T] \rightarrow \exists X F[X]$.

where $T$ is a 1-variable or a $\lambda$-term containing no bound 1-variables.

Inference rules of $EA$

\[
\begin{array}{c}
\text{modus ponens} \\
\hline
F, F \rightarrow G \\
\hline
G
\end{array}
\]
\[
\exists \xi \rightarrow \frac{F[\xi] \rightarrow G}{\exists \xi F[\xi] \rightarrow G}
\]

where $\xi, \eta$ are both 0-variables or 1-variables and the standard proviso is satisfied: the eigenvariable $\eta$ does not occur free in the conclusion $\exists \xi F[\xi] \rightarrow G$.

Definition 9

For any formula $F$ of $EA$ define inductively an $L_2e$-formula $F^*$:

$F^* := F$ for atomic $F$

$(-F^*) := \neg F^*$ and $(F \circ G)^* := (F^* \circ G^*)$ for $\circ = \land, \rightarrow$

$(\exists \xi F[\xi])^* := F^*[\xi F[\xi]^*]$

$(\lambda \xi G)^* := \lambda \xi G^*$

Theorem 2.1

a) If $EA \vdash F$ then $EAE \vdash F^*$.

b) If $EAE \vdash F$ and $F$ is closed then there exists an $EAE$-derivation of $F^*$ in which all formulas are closed.

Proof: a) Let $EA'$ be the set of all $L_2$-formulas $F$ with $EAE \vdash F^*$, $EA'$ is closed under modus ponens and the existential rule $\exists \rightarrow$. (Note that $(\exists \xi F[\xi] \rightarrow G)^* = F^*[\xi F[\xi]^*] \rightarrow G^*$, and therefore closure of $EA'$ under $\exists \rightarrow$ follows from the fact that $EAE$ is closed under substitution.)

The *-translation of any $EA$-axiom of kind (1)-4 is an $EAE$-axiom of the same kind.

The *-translation of any existential axiom of $EA$ is a critical formula.

Below we show that the *-translation of every induction axiom of $EA$ is derivable in $EAE$. This will finish the proof of a).

Observe that b) is an immediate consequence of a): Given an $EAE$-derivation $d$ of a closed formula $F$ one simply replaces every free $i$-variable in $d$ by $0^i$.

Now we consider the *-translation $A$ of an induction axiom.

Obviously $A$ is of the form $G[0] \rightarrow \exists \xi [u] \rightarrow G[e]$ with $\exists \xi [u] := G[x] \rightarrow G[S\xi x]$, $u := \xi e \rightarrow \exists \xi [u]$, $e := \xi e \rightarrow G[x]$. Let $e^* := eq(e = S\xi y)$.

We derive $G[0] \rightarrow \exists \xi [u] \rightarrow G[e]$ by case distinction $e = 0, e \neq 0$. The first case is settled by the equality axiom $0 = e \rightarrow G[0] \rightarrow G[e]$. For the second case observe that $e \neq 0 \rightarrow \exists \xi [u] \rightarrow G[e]$ follows by propositional logic from the minimality axiom $e = S\xi e \rightarrow \neg G[e^*]$, the critical formulas $e \neq 0 \rightarrow e = S\xi e$, $\neg \exists \xi [e^*] \rightarrow \neg \exists \xi [u]$ and the equality axiom $e = S\xi e \rightarrow G[S\xi e^*] \rightarrow G[e]$. \hfill \Box
3 Computations with the $\epsilon$-Substitutions

Definition 10
An $\epsilon$-substitution is a function $S$ such that
\[ \text{dom}(S) \text{ (domain of } S) \text{ is a set of canonical } \epsilon\text{-terms}, \]
if $e \in \text{dom}(S)$ then $S(e) \in B^{(\epsilon)} \cup \{?\}$.

An $\epsilon$-substitution $S$ is called total if $\text{dom}(S)$ is the set of all canonical $\epsilon$-terms.
\[ \mathfrak{S} := S \cup \{(e, ?) : e \text{ is a canonical } \epsilon\text{-term } \notin \text{dom}(S)\} \text{ is called the standard extension of } S. \]

Comment. We consider a function as a set of ordered pairs. So an $\epsilon$-substitution is a set of pairs $(e, u)$ where $e$ is a canonical $\epsilon$-term and $u \in B^{(\epsilon)} \cup \{?\}$. Hilbert's finitist position allowed only for finite $\epsilon$-substitutions $S$. All (canonical) $\epsilon$-terms $\epsilon F$ not mentioned in $S$ explicitly, have default value $0^{(\epsilon)}$. In fact the possible value $?$ for a term indicates exactly this default value. The $\epsilon$-substitution process is defined entirely in terms of finite substitutions: the standard extension $\mathfrak{S}$ of $S$ is used only to simplify notation for computations with default values. Essentially infinite substitutions appear only in section 7 below.

3.1 Computation Steps
Let $S$ be an arbitrary $\epsilon$-substitution.

Definition 11 (Inductive definition of $e \rightsquigarrow_{S} \epsilon'$)
1. If $(e, u) \in S$ and $u \neq ?$, then $e \rightsquigarrow_{S} u$.
2. If $(e, ?) \in S$ then $e \rightsquigarrow_{S}^{1} \epsilon^{(e)}$.
3. If $1 \leq i \leq n$, $e_{i} \rightsquigarrow_{S} \epsilon_{i}$ and $\epsilon'_{i}$ is not a $\lambda$-term then $\epsilon_{1} \ldots \epsilon_{n} \rightsquigarrow_{S} \epsilon_{1} \ldots \epsilon_{1-n} \epsilon'_{i_{n+1}} \ldots \epsilon_{n}$.
4. If $F \rightsquigarrow_{S}^{1} F'$ then $\epsilon F \rightsquigarrow_{S}^{1} \epsilon F'$.
5. If $G \rightsquigarrow_{S}^{1} G'$ then $\lambda z G \rightsquigarrow_{S}^{1} \lambda z G'$.

Comment. Since $S$ is defined only for canonical terms which do not contain any proper closed $\epsilon$-subterms, computations proceed from inside, and since canonical terms are closed, no term containing variables is immediately computed according to $S$: only its closed subterms can be replaced.

Note. If $e \rightsquigarrow_{S}^{1} \lambda z G$ and $e$ is not a $\lambda$-term then either $(e, \lambda z G) \in S$ or $(e, ?) \in S$ and $\lambda z G = 0^{1}$.

Definition 12
1. $e$ is called $S$-reducible if there exists an $\epsilon'$ with $e \rightsquigarrow_{S}^{1} \epsilon'$. Otherwise $e$ is called $S$-irreducible or in $S$-normal form.
2. $\rightsquigarrow_{S}$ denotes the transitive and reflexive closure of $\rightsquigarrow_{S}^{1}$.

Note. $e$ is $S$-reducible if $e$ contains canonical $\epsilon$-subterms in the domain of $S$. In particular no default computations are allowed unless $S$ contains pairs $(e, ?)$.

We are going to prove that $\rightsquigarrow_{S}$ is well-founded, i.e. every sequence of computation steps is terminating, by showing that each step decreases the depth $d(e)$ (Definition 4).

Lemma 3.1 If $e \rightsquigarrow_{S}^{1} \epsilon'$ then $FV(\epsilon') = FV(e)$.

Proof. Consider the cases in the definition of $\rightsquigarrow_{S}^{1}$. If the computation is by 1.1 or 1.2, then $FV(e) = \emptyset = FV(\epsilon')$. In the cases 4, 5 use the induction hypothesis. The same works in case 2 when $p \neq \text{App}$ or $p = \text{App}$ but the second argument is not changed.

Let now $e = \text{App} P t$ and $e' = \text{App}(P', t)$ with $P \rightsquigarrow_{S}^{1} P'$ (recall the definition 6). Then $P$ is an $\epsilon$-term, since 1-variables are not reducible.

1. $P$ is canonical. Then $P' = \lambda z G \in B_{1}$, $FV(G) = \{z\}$ and $\epsilon' = G[t]$. Hence $FV(\epsilon') = FV(t) = FV(e)$.
2. $P$ is not canonical. Then $e' = \text{App} P t$ and $FV(\epsilon') = FV(P') \cup FV(t) \supseteq FV(P) \cup FV(t) = FV(e)$. \hfill \qed

Lemma 3.2 For each arithmetical expression $e$ one has:
\begin{enumerate}
  \item $d(e) < \omega$,
  \item $d(e y/t) < (d(t) + 1) \cdot \omega$.
\end{enumerate}
Proof. a) is obvious, since App does not occur in e and all clauses of the definition 4 except 3 add only finite amount.
b) Induction on e. Let y ∈ FV(e), since otherwise d(ε[e/y][t]) = d(e) < ω. Set α := (d(t) + 1) · ω.
1. If e = y then d(ε[e/y][t]) = d(t) < d(t) + 1 < α.
2. If e = pε1...εn with p ≠ App then d(ε[e][t]) = d(pε1[t]...εn[t]) = d(e1[t])#...#d(en[t]) < α, since by the induction hypothesis (IH) d(ei[t]) < α and α is closed under #.
3. If e = exF then we can assume x ∉ FV(t) and d(e[t]) = d(exF[t]) ≤ 1 + d(F[t]) II
4. If e = λzF, apply IH. □

Lemma 3.3 If e ⊢_S e' then d(e') < d(e).
Proof by induction on the definition of ⊢_S.
1. 1. 2. If e is a canonical ε-term then d(e') = 0 < 1 = d(e).
2. If e = pε1...εn, e' = pε1...εi'...εn and ei ⊢_S e'i', then:
2.1. if p ≠ App then d(e') = d(e1)#{...#d(ei')#{...#d(en) < d(e1)#...#d(en) = d(e)
2.2. if e = AppP and e' = AppP', then d(e') = ω · (d(P')#d(t)) II
2.3. if e = AppP and e' = AppP', one proceeds as in the case 2.2.
3. Let e = AppP and e' = G[t]. Note that ∀α < ω(α · ω ≤ ω · α).
Then by Lemma 3.2 d(e') < d(t' + 1) · ω ≤ ω · (d(t' + 1) ≤ ω · (d(P)#d(t)) = d(e).
4. 5. If e = λF, e' = λe'F or e = λzF, e' = λzF' with F ⊢_S F' then, since FV(F) = FV(F'), we have d(e') = j + d(F') < j + d(F) = d(e) for some j = 0, 1. □
We are going to prove that ⊢_S is locally confluent.

Lemma 3.4 If e is arithmetical and t ⊢_S t' then e[t] ⊃_S e'[t].
Proof is routine.

Lemma 3.5 If e ⊢_S e' and e ⊢_S e'' then there is an expression u such that e' ⊃_S u and e'' ⊃_S u.
Proof by induction on e.
1. If e ∈ dom(S) then e' = e.
2. Let e = pε1...εn, e' = pε1...εi...εn and ei ⊢_S e'i.
2.1. e'' = pε1...εi'...εn with ei ⊢_S e'i'. Then by the IH one has ε1 ⊃_S u1 and e'i' ⊃_S u1 for some u1.
Take u := pε1...εi...εn.
2.2. e'' = pε1...εj...εn with ε1 ⊃_S e'i' and i ≠ j. Take u := pε1...εi...εj...εn.
3. e = AppP, e' = G[t], e'' = AppP' with P ⊢_S λzG and t ⊢_S t'. Then e'' ⊃_S G[t'] and, by Lemma 3.4 e'' ⊃_S G[t']
The remaining cases are easy: apply I.H. □

Theorem 3.6 (Church-Rosser Property)
For each expression e there exists a unique S-irreducible expression e * with e ⊃_S e *.
Proof. By Lemma 3.3 computations terminate, and together with Lemma 3.5 this implies uniqueness. □

Definition 13 (Normal form [e]_S for expression e)
The unique expression e * in the previous theorem is called the S-normalform of e and denoted by [e]_S.

Definition 14 An expression e is S-computable iff d([e]_S) = 0
Note.
1. e is S-computable iff [e]_S = [e]_S.
2. If S is total then every expression is S-computable.
We prove next that ⊃_S is preserved under substitution. The main problem will be with λ-terms.
Lemma 3.7
a) \( e \rightarrow_{\frac{1}{2}} e' \) implies \( \eta \epsilon u \downarrow_{S} e'[\eta/u] \).
b) \( u \rightarrow_{\frac{1}{2}} u' \) implies \( \eta \epsilon u \downarrow_{S} e[\eta/u'] \).
c) \( e[\eta/u] \downarrow_{S} [e][\eta][u][u] \).

Proof. We first prove a) and b) under the assumption that \( u \) is not a \( \lambda \)-term.
a) We use induction on the definition of \( \rightarrow_{\frac{1}{2}} \). The only non-trivial case is \( e = \text{App} pt, e' = G[z/t], P \rightarrow_{\frac{1}{2}} \lambda z G \).
Then since \( P \) is closed and \( FV(G) = \{ z \} \), one has \( e[u] = \text{App} pt[u], e'[u] = G[z/t[u]] \). Hence by the definition of \( \rightarrow_{\frac{1}{2}} \) one has \( e[u] \rightarrow_{\frac{1}{2}} G[z/t[u]] = e'[u] \).
b) We use induction on \( e \) assuming \( \eta \in FV(e) \). Again the only non-trivial case is \( e = \text{App} pt, P[u'] = \lambda z G \).
Then \( e[u] = \text{App} pt[u][t], e[u'] = G[z/t[u']] \) and by III \( P[u] \downarrow_{S} \lambda z G, t[u] \downarrow_{S} t[u'] \). Since \( u \) is not a \( \lambda \)-term, we have \( \lambda z G \in \mathbb{B}_1 \). Hence there exists a canonical \( \epsilon \)-term \( Q \) such that \( P[u] \downarrow_{S} Q \downarrow_{\frac{1}{2}} \lambda z G \). Hence \( e[u] \downarrow_{S} \text{App} pt[u'][u'] \downarrow_{\frac{1}{2}} G[z/t[u']] = e'[u] \).
Now we assume that \( u \) is a \( \lambda \)-term. We have to consider two additional cases.
a') \( e = \text{App} pt \) and \( P[u] = \lambda z G \). Then \( e = \eta \) (otherwise \( P \) contains \( \lambda \)) and thus \( u = \lambda z G, e' = \text{App} pt[t] \) with \( t \rightarrow_{\frac{1}{2}} t' \). Hence \( e[u] = G[z/t[u]], e'[u'] = G[z/t[u']] \) and \( t[u] \downarrow_{S} t[u'] \) by III. Since \( t[u] \) is not a \( \lambda \)-term, by b) we obtain \( e[u] \downarrow_{S} e'[u] \).
b') \( e = \text{App} pt \) and \( P[u] = \lambda z G \). Then \( e = \eta \) (otherwise \( P \) contains \( \lambda \)) and thus \( u = \lambda z G, u' = \lambda z G' \) with \( G \rightarrow_{S} G' \). Hence \( e[u] = G[z/t[u]] \) and \( e[u'] = G[z/t[u']] \). By III \( t[u] \downarrow_{S} t[u'] \). Since \( t[u] \) is not a \( \lambda \)-term, by a), b) we obtain \( G[z/t[u]] \rightarrow_{S} G' G[z/t[u']] \rightarrow_{S} G'[z/t[u']] \).
By induction on the number of computation steps, we obtain versions of a), b) with \( \rightarrow_{\frac{1}{2}} \) replaced by \( \rightarrow_{S} \). These imply c): \( e[u] \downarrow_{S} \left[ e[u][u][u][u] \right] \rightarrow_{S} \left[ e[u][u][u][u] \right] \). \( \Box \)

The next proposition clarifies the structure of normal forms for partial substitutions \( S \), where some canonical \( \epsilon \)-terms are irreducible. It is needed in section 6.3.

Lemma 3.8
a) If \( v \) is an \( S \)-irreducible subterm of \( e \), \( d(v) > 0 \) and \( e \rightarrow_{\frac{1}{2}} e' \) then \( v \) is also a subterm of \( e' \).
b) If \( e[\eta/u] \) is \( S \)-computable and \( \eta \in FV(e) \) then \( u \) is \( S \)-computable.
c) If \( e \) is \( S \)-computable and \( e \rightarrow_{\frac{1}{2}} e' \) then \( e \rightarrow_{S} e' \).

Proof. a) We use induction on the definition of \( \rightarrow_{\frac{1}{2}} \). Since \( e \) is reducible, \( v \) has to be a proper subterm of \( e \) and thus \( e \) cannot be a canonical \( \epsilon \)-term.
1. Let \( e = \epsilon e_1 \ldots e_n \) and \( e' \rightarrow_{\frac{1}{2}} e_i \). Then the claim follows immediately from the III.
2. \( e = \text{App} pt, e' = G[z/t] \) with \( P \rightarrow_{\frac{1}{2}} \lambda z G \). Then \( P \) is a canonical \( \epsilon \)-term \( \neq v \), and therefore \( v \) is not a subterm of \( P \). Hence \( v \) is a subterm of \( t \) and thus of \( e' \) since \( z \) occurs free in \( G \).
3. The cases \( e = \epsilon F, e' = \epsilon F' \) or \( e = \lambda z F, e' = \lambda z G' \) with \( F \rightarrow_{\frac{1}{2}} F' \) follow immediately from III.
b) Assume for contradiction that \( d([u][u]) > 0 \). Then \([u][u] \) and hence \([e][u][u] \) contains an \( S \)-irreducible, closed \( \epsilon \)-subterm \( v \). By Lemma 3.7 and Theorem 3.6 we have \([v][u][u] \downarrow_{S} [v][u][u] \), and therefore \( [v][u][u] \) is a subterm of \([e][u][u] \). This contradicts the \( S \)-computability of \( e[u] \).
c) It suffices to prove the Lemma for one step reductions. Assume that \( e \) is \( S \)-computable and \( e \rightarrow_{\frac{1}{2}} e' \) holds.
By b) every canonical \( \epsilon \)-subterm \( v \) of \( e \) is \( S \)-computable and therefore in \( dom(S) \). Hence \( e \xrightarrow{\frac{1}{2}} e' \). \( \Box \)

Let us establish some properties of regular \( \lambda \)-terms (Definition 8) needed in section 5.

Lemma 3.9
a) If \( T \) is regular and \( T \xrightarrow{\frac{1}{2}} T' \) then \( T' \) is regular.
b) If \( T \) is a closed regular \( \lambda \)-term with \( d(T) = 0 \) then \( T \in \mathbb{B}_1 \).
c) If \( T \) is a closed regular \( \lambda \)-term then \( \downarrow T \in \mathbb{B}_1 \).

Proof. We have \( T = e[\eta_1/u_1, \ldots, \eta_n/u_n] \) where \( u_1, \ldots, u_n \) are \( \epsilon \)-terms and no bound 1-variable occurs in \( e \).
a) Without loss of generality we may assume that either \( T' = e[u_1', u_2, \ldots, u_n] \) with \( u_1 \xrightarrow{\frac{1}{2}} u_1' \) or \( T' = e'[u_1, \ldots, u_n] \) with \( e' \xrightarrow{\frac{1}{2}} e' \). In both cases one easily sees that \( T' \) is again regular. (Only if \( u_1' \in \mathbb{B}_1 \) a short argument is needed.)
b) For contradiction let us assume that \( \eta_i \in FV(e) \). Then, since \( T \) is closed, \( u_i \) is a closed \( \epsilon \)-term and thus \( 0 < d(u_i) \leq d(T) \). Hence none of \( \eta_1, \ldots, \eta_n \) occurs free in \( e \) and we have \( T = e \) which means that \( T \) is a closed \( \lambda \)-term containing no 1-variables. Hence \( T \in \mathbb{B}_1 \), since \( d(T) = 0 \).

c) follows from a) and b). □

4 The rank function

The rank will measure nesting of bound variables. We extend to the \( \epsilon \)-language a definition known for Ramified Analysis [16] .

Set \( o(x) := 0 \), \( o(X) := \omega \), \( o(App) := \omega \), \( o(p) := o(0) := 0 \), for \( p \neq \text{App} \).

In the following \( \sigma \) denotes elements from \( \text{Var} \cup \{ * \} \).

**Definition 15** Definition of \( rk_\sigma(e) \)

1. If \( \sigma \not\in FV(e) \cup \{ * \} \) then \( rk_\sigma(e) := 0 \).
2. For \( \sigma \in FV(e) \cup \{ * \} \) we define:

\[
  rk_\sigma(e) := \begin{cases} 
    o(e) & \text{if } e \in \text{Var} \cup \{ 0 \} \\
    \max \{ o(p), rk_\sigma(e_1), \ldots, rk_\sigma(e_n) \} & \text{if } e = pe_1 \ldots e_n \\
    \max \{ o(\xi), rk_\sigma(F), rk_\sigma(F + 1) \} & \text{if } e = \epsilon F \\
    rk_\sigma(G) & \text{if } e = \lambda G
  \end{cases}
\]

**Note 1.** \( rk_\sigma(e) \leq \omega \cdot 2 \).

**Note 2.** \( rk_\sigma(e) \) is a measure of nesting of bound variables in subterms of \( e \) containing free variable \( \sigma \), and \( rk_\sigma \) takes account of all \( \epsilon \)-subterms. More precisely,

**Lemma 4.1**

\[
  rk_\sigma(e) = \sup \{ o(\eta), rk_\sigma(F) + 1 : \eta, \epsilon F \text{ occur in } e \} 
\]

**Proof** by induction on \( e \). Define

\[
  o(e) := \sup \{ o(\eta) : \eta \text{ occurs in } e \}, \quad rk_1(e) := \sup \{ rk_\sigma(F) + 1 : \epsilon F \text{ occurs in } e \}
\]

Note that \( o(pe_1 \ldots e_n) = \max \{ o(e_1), \ldots, o(e_n) \} \).

1. \( e \in \text{Var} \cup \{ 0 \} \). Then \( rk_\sigma(e) = o(e), \quad rk_1(e) = 0 \)

2. \( e = pe_1 \ldots e_n \). Then

   \[
   rk_\sigma(e) := \max \{ o(p), \max \{ o(e_i), rk_1(e_i) \} \} = \max \{ o(p), \max \{ o(e_i) \} \} = \max \{ o(e), rk_1(e) \}.
   \]

3. \( e = \epsilon F \). Then

   \[
   rk_\sigma(e) := \max \{ o(\xi), rk_\sigma(F), rk_\sigma(F + 1) \} \stackrel{IH}{=} \max \{ o(\xi), o(F), rk_1(F), rk_1(F + 1) \} = \max \{ o(\xi), rk_1(e) \}
   \]

4. \( \epsilon = \lambda G \). Then \( rk_\sigma(e) = rk_\sigma(G) \stackrel{IH}{=} \max \{ o(G), rk_1(G) \} = \max \{ o(e), rk_1(e) \} \). □

Next lemmas establish properties of rank.

**Lemma 4.2**

a) \( e \) arithmetical \( \implies \) \( rk_\sigma(e) < \omega \)

b) \( X \in FV(e) \implies \omega \leq rk_X(e) \).

c) \( \omega \leq \text{rk}_X(P) \), for each \( 1 \)-term \( P \).

**Proof.** Easy induction on \( e, P \).

**Lemma 4.3**

If \( d(e) = 0 \) and \( e \) is not a \( \lambda \)-term then \( rk_\sigma(e) \leq \sup \{ rk_\sigma(e) : \sigma \in FV(e) \} \).

**Proof:**

1. \( e = 0 \). Then \( rk_\sigma(e) = 0 \). \( e = \xi \in \text{Var} \). Then \( rk_\sigma(e) = o(\xi) = rk_\xi(e) \).

2. \( e = \epsilon F \). Then, since \( d(e) = 0 \), we have \( d(F) = 0 \) and \( FV(e) \neq \emptyset \). Therefore
\[
\max \{ o(\xi), r_{k}(F) + 1 \} \leq \sup \{ r_{k}(e) : \sigma \in FV(e) \} \quad \text{and} \quad \text{by I.H.,} \quad \sup \{ r_{k}(F) : \sigma \in FV(F) \}. \\
\text{Hence} \quad r_{k}(e) = \max \{ o(\xi), r_{k}(F_{1}), r_{k}(F_{2}) + 1 \} \leq \sup \{ r_{k}(e) : \sigma \in FV(e) \}. \\
3. \quad e = p_{e_{1}} \ldots e_{n}. \quad \text{Since} \ \omega \leq r_{k}(P) \quad \text{for each 1-term} \ \mathcal{P}, \ \text{we have} \\
r_{k}(e) = \max \{ r_{k}(e_{1}), \ldots, r_{k}(e_{n}) \} \quad \text{I.H.} \\
\sup \{ r_{k}(e_{i}) : \sigma \in FV(e_{i}) \} \leq \sup \{ r_{k}(e) : \sigma \in FV(e) \}. \quad \square \\
\text{Lemma 4.4} \\
\text{If} \ \epsilon \in F \quad \text{is canonical then} \quad r_{k}(F) \leq r_{k}(\epsilon) \quad \text{and} \quad r_{k}(\epsilon F) = \max \{ o(\xi), r_{k}(\epsilon F) + 1 \}. \\
\text{Proof:} \\
\text{Since} \ \epsilon \in F \quad \text{is canonical, we have} \ d(F) = 0. \quad \text{By 4.3 this yields} \\
r_{k}(F) \leq \sup \{ r_{k}(F) : \sigma \in FV(F) \} \quad \text{and thus (since} \ FV(F) \subseteq \{ \xi \} \quad \text{)} \quad r_{k}(F) \leq r_{k}(\epsilon). \\
\text{Hence} \ r_{k}(\epsilon F) = \max \{ o(\xi), r_{k}(F), r_{k}(F + 1) \} = \max \{ o(\xi), r_{k}(\epsilon + 1) \}. \quad \square \\
\text{The next two lemmas show that the rank does not increase during computation.} \\
\text{Lemma 4.5} \\
r_{k}(\epsilon[y/t]) < \max \{ \omega, r_{k}(t) + 1 \}, \quad \text{if} \ \epsilon \ \text{is arithmetical.} \\
\text{Proof:} \quad \text{Let} \ y \in FV(e) \quad \text{and} \ \sigma \in FV(e) \cup \{ * \}. \quad \text{(Otherwise} \ r_{k}(e[t]) = r_{k}(e) < \omega \quad \text{or} \ r_{k}(e[t]) = 0. \) \\
1. \ \epsilon = y: \quad r_{k}(e[t]) = r_{k}(t). \\
2. \ \epsilon = p_{e_{1}} \ldots e_{n}: \quad \text{Then} \ r_{k}(e[t]) = \max \{ o(p), r_{k}(e_{1}[t]), \ldots, r_{k}(e_{n}[t]) \}, \quad \text{and the assertion follows by I.H..} \\
3. \ \epsilon = \epsilon F. \quad \text{Then} \ o(\epsilon) = 0. \quad r_{k}(e[t]) = \max \{ r_{k}(F) \}, \quad r_{k}(F[t] + 1) \quad \text{and, by I.H.,} \\
r_{k}(F[t]) < \max \{ o(\xi), r_{k}(F[t] + 1) \} \quad \text{and} \quad r_{k}(F[t]) < \max \{ \omega, r_{k}(t) + 1 \} = \omega. \\
\text{The last equation holds by clause 1 in the definition of rank, since} \ x \notin FV(t) \cup \{ * \}. \\
4. \ \epsilon = \lambda \xi G. \quad \text{The assertion follows by I.H.} \quad \square \ \\
\text{Lemma 4.6} \\
\text{If} \ \epsilon \rightarrow \frac{1}{2} \epsilon' \quad \text{then} \ r_{k}(\epsilon') \leq r_{k}(\epsilon). \\
\text{Proof by induction on the definition of} \ \epsilon \rightarrow \frac{1}{2} \epsilon. \\
\text{Let} \ \epsilon \in FV(e) \cup \{ * \}. \quad \text{(Otherwise} \ r_{k}(e') = 0. \) \\
1. \ If \ \epsilon \ \text{is a canonical 0-\epsilon-term then} \ r_{k}(\epsilon') = 0, \quad \text{since} \ \epsilon' \in N. \\
2. \ If \ \epsilon \ \text{is a canonical 1-\epsilon-term then} \ r_{k}(\epsilon') < \omega \leq r_{k}(\epsilon), \quad \text{since} \ \epsilon' \ \text{is arithmetical and} \ \sigma = * \ \\
3. \ \epsilon = p_{e_{1}} \ldots e_{n} \quad \text{or} \ \epsilon = \lambda \xi G. \quad \text{Then} \\
r_{k}(e'[t]) = \max \{ \omega, r_{k}(t) \} < \max \{ \omega, r_{k}(e[t]) \}. \\
\text{2. otherwise: immediate from I.H.} \\
3. \ \epsilon = \epsilon F \quad \text{and} \quad \epsilon' = \epsilon F' \quad \text{with} \ F \rightarrow \frac{1}{2} F'. \\
r_{k}(\epsilon'[t]) = \max \{ o(\xi), r_{k}(F'), r_{k}(F') + 1 \} \leq \max \{ o(\xi), r_{k}(F), r_{k}(F) + 1 \} = r_{k}(\epsilon). \quad \square \\
\text{Substitution of a variable by an appropriate canonical value also does not increase the rank.} \\
\text{Lemma 4.7} \\
r_{k}(\epsilon[\eta/u]) \leq r_{k}(\epsilon), \quad \text{for each} \ u \in B_{\eta}. \\
\text{Proof by induction on} \ e: \quad \text{Let} \ \epsilon' := \epsilon[\eta/u] \quad \text{and assume that} \ \eta \in FV(e) \quad \text{and} \ \sigma \in FV(e'). \\
\text{Note that} \ FV(e') = FV(e) \setminus \{ \eta \}. \\
1. \ \epsilon = \eta: \quad \text{Then} \ r_{k}(\epsilon'[t]) = 0 \quad \text{(if} \ \varphi(\eta) = 0) \quad \text{or} \ r_{k}(\epsilon') < \omega = r_{k}(\epsilon) \quad \text{(if} \ \varphi(\eta) = 1). \\
2. \ \epsilon = App \eta \quad \text{and} \ \lambda \xi G. \quad \text{Then} \ e'[t] = G[z / t u] \quad \text{and thus} \\
r_{k}(\epsilon'[t]) \leq \max \{ \omega, r_{k}(t u) \} \quad \text{I.H} \\
3. \ \epsilon = \epsilon F. \quad \text{Then} \\
r_{k}(\epsilon'[t]) = r_{k}(\epsilon[\eta/u]) \leq \max \{ o(\xi), r_{k}(F'), r_{k}(F') + 1 \} \quad \text{I.H} \\
\text{In all other cases the assertion follows immediately from I.H.} \quad \square \\
\text{The next statement shows that our definition of rank is suitable: the rank decreases when the 'body' of a canonical \epsilon-term is substituted by a canonical value.}
Lemma 4.8
If $\xi F$ is canonical then $\rk_s(F[u]) < \rk_s(\xi F)$, for each $u \in B_i(\xi)$.

Proof: $\rk_s(F[u]) \leq \rk_s(\xi F) \leq \rk_q(F) < \rk_s(\xi F)$. □

Definition 16 $\rk(e) := \rk_s(e)$ is called the rank of $e$.

Definition 17 (Truncation to a given rank)
For each $\epsilon$-substitution $S$ and $r \in \mathbb{N}$ we set $S_{\leq r} := \{ (e, u) \in S : \rk(e) \leq r \}$. Analogously we define $S_{< r}, S_{= r}, S_{> r}$.

Lemma 4.9
If $S, S'$ are $\epsilon$-substitutions with $S_{\leq r} = S'_{\leq r}$ then $\models S[.] = \models S'[.]$ holds for all expressions $e$ of rank $\leq r$.

Proof:
Since all subterms of an expression $e$ have ranks $\leq \rk(e)$, we have:
$\rk(e) \leq r \Rightarrow \forall e' (e \to_S e' \iff e \to_{S'} e')$.
Together with Lemma 4.6 this yields the assertion by induction on $d(e)$. □

5 The H-process

Let us recall that critical formulas are formulas of three types:

$F[\emptyset] \rightarrow F[\epsilon x F[\emptyset]]$,
$s \neq 0 \rightarrow F[\epsilon x F[s]]$ with $F := (s = \emptyset)$,
$F[T] \rightarrow F[\epsilon x F[T]]$ with $T$ a 1-term or regular lambda-term.

We assume that $C_{r_0}, \ldots, C_{r_N}$ (with $N \in \mathbb{N}$) is a fixed sequence of closed critical formulas.

In this section we define a successful approximation process for finding a solution of this system according to Hilbert’s approach. It is useful to recall here a Comment from Section 3. The H-process will be arranged in such a way that all non-default values of $\epsilon$-terms are correct: $S(\epsilon x F[\emptyset])$ is the least $n$ satisfying $F[n]$, and $S(\epsilon x F[T])$ is an arithmetical $\lambda$-term $T$ satisfying $F[T]$.

Definition 18
$F[x/n] := F[x/n] \land \neg F[x/n-1] \land \ldots \land \neg F[0]$, $F[X/T] := F[X/T]$.

Definition 19
Let $S$ be an $\epsilon$-substitution:

$e \to_S \text{TRUE (FALSE)} : \iff \models S[e] \in \text{TRUE (FALSE)}$.

$F(S) := \{ F[k/u] : (\epsilon F, u) \in S \wedge u \neq ? \}$

$S$ is correct iff $A \to_S \text{TRUE for all } A \in F(S)$. $S$ is solving iff $C_{r_I} \to_S \text{TRUE for } I = 0, \ldots, N$. Otherwise $S$ is nonsolving.

$\bar{S} := S \cup \{(e, ?) : e \text{ canonical } \epsilon \text{-term } \notin \text{ dom}(S)\}$ is called the standard extension of $S$, cf. Section 3.

The definition of a term whose value is to be corrected (H-term) to a new value (H-value) and of the $\epsilon$-substitution $H(S)$ to which $S$ is reduced by this change of value, is given in terms of the standard extension $\bar{S}$. In fact the first term of minimal rank is chosen, and all values of higher rank are cancelled. Since $\epsilon$-substitutions are defined only for canonical terms, all these operations are preceded by transforming arbitrary $\epsilon$-term $\epsilon x F$ into its canonical form $\epsilon x [F]$.

Definition 20
Let $S$ be an $\epsilon$-substitution such that $\bar{S}$ is nonsolving. (Then $|C_{r_I}[\bar{S}] \in \text{FALSE for some } I \leq N$.)
Set $r_{0} := \rk(\epsilon x [F][\bar{S}])$, where $C_{r_I} = F_{0} \rightarrow F[\epsilon x F]$.

$C_{r}(S) := C_{r_{0}}$, where $I \leq N$ is such that

$|C_{r_{0}}[\bar{S}] \in \text{FALSE } \& \forall J \leq N \exists C_{r_{J}}[\bar{S}] \in \text{FALSE } \Rightarrow r_{J} < r_{J} \lor (r_{J} = r_{J} \land I \leq J)|$. 

12
Let \( C_r(S) = F_0 \rightarrow F[\xi F] \):
\( \xi[F] \) is called the H-term of \( S \).

The H-value \( v \) of \( S \) is defined as follows
a) if \( s(\xi) = 1 \) and \( F_0 = F[T] \) then \( v := [F[T]] \).
b) if \( s(\xi) = 0 \), \( F_0 = (s \neq 0) \), and \( F = (s = \bar{S} \bar{v}) \) then \( v := [\bar{S} \bar{v} - 1] \).
c) if \( s(\xi) = 0 \) and \( F_0 = F[T] \) then \( v := \) the unique \( n \in \mathbb{N} \) with \( [F][\overline{\xi}] \rightarrow_S \) TRUE.

Remark: If \( e \) is the H-term and \( v \) the H-value of \( S \), then \( v \in \mathbb{B}_{\lambda(e)} \). (For \( s(e) = 1 \) cf. Lemma 3.9c.)

The next definition is central for the substitution method.

**Definition 21** If \( S \) is nonsolving then
\( H(S) := (S \setminus \{(e, ?)\}) \cup \{(e, v)\} \), where \( e \) is the H-term and \( v \) the H-value of \( S \).

Let us prove that \( H(S) \) is indeed a correct \( e \)-substitution if \( S \) is correct and nonsolving.

**Lemma 5.1**
Let \( S \) be an \( e \)-substitution such that \( S \) is correct and nonsolving,
and let \( e \) be the H-term, \( v \) the H-value of \( S \). Then the following holds:

\[ a) (e, ?) \in S, \]
\[ b) [\Delta[S]] = v \neq 0^{(e)}, \]
\[ c) H(S) \text{ is correct.} \]

**Proof:**
Let \( C_r(S) = F_0 \rightarrow F[\xi F] \). Then \( e = \xi[F] \).

a) \textbf{Assumption:] \((e, w) \in S \text{ and } w \neq ? \). Then, since \( S \) is correct, by Lemma 3.7c \([F[\xi F]] = [F[w]] = \text{TRUE}.\]

On the other hand, since \( C_r(S) \rightarrow_S \) FALSE, we have \([F[\xi F]] = \text{FALSE}. \) \textbf{Contradiction}.

b) By Lemma 3.7b we have \([F[e]] = [F[w]] \neq \text{TRUE}. \) Hence, by 3.7b, \( v \neq 0^{(e)} \).

The equation \([\Delta[S]] = v \) holds, since \((e, v) \in H(S)\).

c) Let \((e, w) \in H(S) \) with \( w \neq ? \). Then \((e, w) \in S \) or \((e, w) \in H(S) \).
One has \([A[w]]_S \in \text{TRUE}\): in the first case since \( S \) is correct, and in the second case by definition of \( H(S) \) and Lemma 3.7c. By Lemma 4.8 one has \( \text{rk}(A[w]) < \text{rk}(e) \), and by Lemma 4.9 \([A[w]]_H(S) = [A[w]]_S \). \( \square \)

**Definition 22**
The H-process \((for \ C_{r_0}, \ldots, C_{r_N})\) is defined as follows:
\( S_0 := \emptyset, \quad S_{n+1} := \begin{cases} H(S_n) & \text{if } S_n \text{ is nonsolving} \\ \emptyset & \text{otherwise} \end{cases} \)

The H-process terminates iff there exists an \( n \in \mathbb{N} \) such that \( S_n \) is solving.

The next definition determines when the substitution \( S \) itself, (and not \( \overline{S} \)) contains sufficient information to compute all necessary values. An important instance is the provable incorrectness of \( S \).

**Definition 23**
Let \( S \) be an \( e \)-substitution.

\( S \) is computationally inconsistent if \( S \) FALSE for some \( A \in F(S) \).

Otherwise \( S \) is computationally consistent.

An expression \( e \) is \( S \)-computable if \( d([\overline{\xi}]) = 0 \).

\( S \) is computing all formulas \( A \in F(S) \) are \( S \)-computable.

\( S \) is deciding if \( S \) is computing and the critical formulas \( C_{r_0}, \ldots, C_{r_N} \) are \( S \)-computable.

**Remark**
1. A closed formula \( A \) is \( S \)-computable iff \( A \rightarrow_S \) TRUE or \( A \rightarrow_S \) FALSE.
2. \( e \) is \( S \)-computable iff \( [\overline{\xi}] = [\xi] \).
3. If \( S \) is computing and \( e \) then \( S \) is correct.

The next definition requires that all steps for computing \( H(S) \) are possible in terms of \( S \) itself.
Definition 24
Let $S$ be an $e$ substitution. We say that the H-rule applies to $S$ iff
(1) $S$ is oc, deciding, nonsol, and
(2) if $\text{Cr}(S) = F[t] \rightarrow F[\epsilon x F]$ then there exists an $n \in \mathbb{N}$ with $[F][s][n] \rightarrow_s \text{TRUE}$.

Lemma 5.2
Let $S$ be oc, deciding, nonsol, and $\text{Cr}(S) = F[t] \rightarrow F[\epsilon x F]$. Then
a) $\|s\| \in \mathbb{N}$ and $F\lfloor s \rfloor \rightarrow_s \text{TRUE}$.
b) If the formulas $F[k]$ ($i < \|s\|$) are $S$-computable then the H-rule applies to $S$.

Proof:
a) Since $\text{Cr}(S) \rightarrow_s \text{FALSE}$ and $\text{Cr}(S)$ is S-computable, we have $F[t] \rightarrow_s \text{TRUE}$. Now the assertion follows by the Lemmas 3.7b and 3.8b.
b) Let $m := \|s\|$. Since $F[w] \rightarrow_s \text{TRUE}$ and since $F[0], ..., F[n-1]$ are $S$-computable, for
$n := \min \{k : F[k] \rightarrow_s \text{TRUE}\}$ we have $n = m$ and $[F][s][n] \rightarrow_s \text{TRUE}$. □

Lemma 5.3
If the H-rule applies to $S$ then $H(S)$ is correct.

Proof: (cf. proof of 5.1)
Let $(\epsilon \xi A, w) \in H(S)$ with $w \neq \epsilon$. We have to prove $[A[w]]_{H(S)} \in \text{TRUE}$. But $\text{rk}(A[w]) < \text{rk}(e)$ (with e the H-term of $S$) and therefore $[A[w]]_{H(S)} = [A[w]]_{S}$. If $(\epsilon \xi A, w) \in S$ then $[A[w]]_{S} \in \text{TRUE}$, since $S$ is correct. Otherwise $\epsilon \xi A = e$ and $w = \text{H-value of } S$. We treat the case where $\text{Cr}(S) = F[T] \rightarrow F[\epsilon x F][X]$ and leave the other cases to the reader. Then $\epsilon \xi A = e = \epsilon [F[T]]_{S}$ and $w = [T]_{S}$. Since $S$ is deciding, $F[T]$ is $S$-computable and therefore (by Lemmas 3.7c and 3.8c) $[F[T]]_{S} = [A[w]]_{S}$. Since $\text{Cr}(S) = F[T] \rightarrow \text{FALSE}$ and $S$ is deciding, we have $[F[T]]_{S} = [F[T]]_{S} \in \text{TRUE}$. □

6 Termination proof via Cut-elimination

In this section we apply the general schema from [12] for reducing the termination problem for the H-process to the cut-elimination problem in some specially devised sequent calculus with a kind of $\omega$-rule. The termination proof is constructive and uses induction up to $\epsilon_{\alpha}$. In the sequel we use $r$ as syntactic variable for ordinals ($< \epsilon_{\alpha}$).

6.1 The system $\epsilon EA$

The original infinite derivation constructed for a given system $E$ of critical formulas represents the tree of all finite $\epsilon$-substitutions: empty substitution is put at the bottom, and all one-component extensions of $S$ are placed above $S$ (the rule Cut below). The branch of this tree is terminated if $\epsilon$ is correct. Otherwise we start another substitution, solution, or a substitution admitting H-rule is reached. This 'bottom-up' view of the inference rules is helpful below.

The Cut-elimination process introduces steps of the H-process into this tree in the form of the rules Fr, H. The rule Fr temporarily 'freezes' the default value of $e$, which may be changed by the rule H into non-default values.

Definition 25
A sequent is a function $\Theta$ such that $\text{dom}(\Theta)$ is a set of canonical $e$-terms, and $\Theta(e) \in B_{i(e)} \cup \{?, ?^e\}$ for each $e \in \text{dom}(\Theta)$.

So a sequent is almost the same as an $e$-substitution. A component $(e, ?)$ of a sequent $S$ indicates that the default value for $e$ is fixed and will not be changed in the extensions of $S$ to be considered. By identifying $?^e$ with $\epsilon \xi A , ?$, we associate with every sequent $\Theta$ an $e$-substitution which is also denoted by $\Theta$. A sequent $\Theta$ is called correct (oc, deciding, computing) if the associated $e$-substitution has the respective property. An analogous convention is followed with respect to notations like $\|s\|$, $\rightarrow_s$, etc.
Abbreviation:
\[(e, u), \Theta := \{(e, u) \cup \Theta, \text{ if } e \not\in \text{dom}(\Theta)\}.
\]
\[\Theta \leq r \iff \forall (e, u) \in \Theta(r(e) \leq r) \quad (\iff \Theta = \Theta_{\leq r}). \text{ In the same way we define } \Theta \geq r\].

**Definition 26**
\[\Theta f := \{(e, u) \in \Theta : u = ?\}, \Theta t := \{(e, u) \in \Theta : u = ?\}: \text{ fixed and temporary part of a sequent.}\]

**The system eEA**

**Rules of inference:**

\hspace{1cm} (Cut) \[\frac{(e, \), \Theta \ldots (e, u), \Theta \ldots (u \in \mathbb{B}_{i(e)})}{\Theta} \]

\hspace{1cm} (CutFr) \[\frac{(e, \), \Theta \ldots (e, u), \Theta \ldots (u \in \mathbb{B}_{i(e)})}{\Theta} \]

\hspace{1cm} (Fr) \[\frac{(e, ?), \Theta}{\Theta} \]

\hspace{1cm} (H) \[\frac{(e, v), \Theta_{\leq r(e)}}{(e, ?), \Theta}, \text{ if the H-rule applies to } (e, ?), \Theta, \text{ and } e \text{ is the H-term, } v \text{ the H-value of } (e, ?), \Theta.\]

**Axioms:**

\[\text{(As F)} \quad \Theta \quad \text{ if } \Theta \text{ is ci} \]

\[\text{(As S)} \quad \Theta \quad \text{ if } \Theta \text{ is cc, deciding, and solving} \]

\[\text{(As H)} \quad (e, ?), \Theta \quad \text{ if the H-rule applies to } (e, ?), \Theta \text{ and } e \text{ is the H-term of } (e, ?), \Theta.\]

In the above rules and axioms e always denotes a canonical e-term not in dom(\(\Theta\)).

We call e the main term of the respective inference.

**Definition 27** By a deduction in eEA we mean a deduction (i.e. wellfounded tree) according to the rules of inference of eEA from axioms of eEA and additional sequents. By a derivation in eEA we mean a deduction in eEA from axioms of eEA only.

By h(\(d\)) we denote the height of the deduction \(d\), i.e. h(\(d\)) := \(\sup\{h(d_u) + 1 : u \in I\}\) where (\(d_u\))\(_{u \in I}\) is the family of immediate subdeductions of \(d\).

If \(I\) is an inference then rk(\(I\)) denotes the rank of its main term.

If \(d\) is a deduction, and \(X\) is one of the symbols Cut, CutFr, Fr, H, and \(\bowtie\) is one of the symbols \(<, \leq, >, \geq, =\) then \(X(d) \bowtie r \iff \text{rk}(I) \bowtie r\) for every \(X\)-inference \(I\) in \(d\).

Hence “Cut(d) \(< r\)” means that all cuts in \(d\) have rank \(< r\), and “\(X(d) \neq 0\)” means that there are no \(X\)-inferences in \(d\).

**6.2 Cut-elimination**

Cuts will be eliminated in the usual way beginning with the maximal rank \(r\). Eliminated cuts will be replaced by CutFr and H with the same main term, i.e. with the same rank. More precisely, a cut will be replaced by CutFr and then moved (permuted) up the derivation until one encounters AsS with a main term \(e\) traceable to the main formula \((e, ?)\) of CutFr. Then the AsS is replaced by the rule H, and the derivation of the corresponding right premise of the cut is placed over the rule H. After all cuts of rank \(r\) are eliminated, these CutFr will be pruned by Fr. So finally cuts of rank \(r\) will be replaced by Fr of rank \(r\). This motivates the following

**Definition 28** Let \(d\) be a deduction.

\(d\) is an \(r\)-derivation iff Cut(d) \(< r \&\& \text{CutFr}(d) \,< 0 \&\& \text{Fr}(d) \geq r \&\& \text{H}(d) \geq r.\)

\(d\) is an \(r^{+}\)-derivation iff Cut(d) \(< r \&\& \text{CutFr}(d) \equiv r \&\& \text{Fr}(d) \geq r \&\& \text{H}(d) \geq r.\)

**Lemma 6.1**

Every \(r^{+}\)-derivation \(d\) of \(\Theta\) can be transformed into an \(r\)-derivation \(d'\) of \(\Theta\) with h(\(d'\)) \(\leq h(d)\) by pruning each CutFr to Fr.
Proof is obvious. □

Lemma 6.2 (Properties of fixed and temporary parts of a sequent)
a) If \( \Theta \) is a sequent in an \( r+1 \)-deduction of \( \emptyset \) then \( \Theta \Gamma > r \) and \( \Theta \Gamma \leq r \).
b) If \( \Sigma \) is a sequent in an \( r^+ \)-deduction of a sequent \( \Theta \) then

1. \( \Theta _{<r} \setminus \Theta \subseteq \Sigma \).
2. \( (\Sigma f)_{>r} \subseteq \Theta \).
3. \( \Theta \Gamma > r \Rightarrow \Sigma \Gamma > r \).

Proof.
a) is proved by the bottom-up induction on the given deduction. This statement is obviously true for \( \emptyset \), and is inherited from conclusion to the premise of a rule: temporary part \( \Theta \Gamma \) is increased only by Fr which has rank \( \Gamma > r+1 \) and \( \Theta f \) is increased by Cut which has rank \( < r+1 \) (i.e. \( \Gamma > r \)).
b) (1) The only non-trivial case is \( (H) \). Let \( \Theta = (e, ?) \), \( \Upsilon \), and \( \Sigma = (e, v) \). Then \( \Theta _{<r} \setminus \Theta \subseteq \Upsilon _{<r} \subseteq \Sigma \), since \( r < \Gamma \).
(2) Going from \( \Sigma \) down to \( \Theta \) the only points where some \( (e, ?) \) could vanish are Cut-inferences. But each Cut in an \( r^+ \)-derivation has rank \( \leq r \).
(3) Holds, since \( Fr(d) > r \) and \( CutFr(d) = r \), so that formulas added to \( \Theta \Gamma \) are of the rank \( \geq r \). □

Applicability of the standard cut-elimination transformations in our case is seriously restricted by non-admissibility (in general) of the weakening rule: adding components to a sequent can fail to produce a sequent.

Definition 29 Two sequents \( \Theta, \Sigma \) are multiplicable if \( \Theta \cup \Sigma \) is a function. In this case we write \( \Theta * \Sigma \) for \( \Theta \cup \Sigma \), and say that \( \Theta * \Sigma \) is defined.

Lemma 6.3
Let \( d \) be an \( r^+ \)-derivation of \( \Theta \). Let \( \Sigma \leq r \) be a correct sequent such that \( \Theta * \Sigma \) is defined and \( (\Sigma f)_{>r} \subseteq \Theta \). Then there exists an \( r^+ \)-derivation \( d * \Sigma \) of \( \Theta * \Sigma \) with \( h(d * \Sigma) \leq h(d) \).

Proof by induction on \( d \):
We distinguish cases according to the last inference of \( d \).
1. Cut with main term \( e \): Then \( \text{rk}(e) < r \).
   (a) \( e \notin \text{dom}(\Sigma) \); \( \Sigma \) is obtained from \( (d_{\text{cut}} \Sigma)_{\text{Fr}} \in \text{Fr} \{e\} \) by the same Cut.
   (b) \( (e, ?) \in \Sigma \). This cannot happen, since \( \Theta \Gamma > r \) and \( \text{rk}(e) < r \).
   (c) \( (e, u) \in \Sigma \) with \( u \in \text{Fr} \{e\} \). Then \( (e, u) \notin \Sigma \), and therefore the Cut is pruned: the derivation \( d * \Sigma := d_{\text{cut}} \Sigma \) has endsequent \( \Theta * \Sigma \).
2. Fr with main term \( e \): Then \( \text{rk}(e) = r \) and \( (e, ?) \notin \Sigma \). Hence \( (e, ?) \notin \Sigma \), since \( (\Sigma f)_{>r} \subseteq \Theta \).
   (a) \( e \notin \text{dom}(\Sigma) \): As (a) above.
   (b) \( (e, u) \notin \Sigma \) with \( u \in \text{Fr} \{e\} \). As (c) above.
3. H with main term \( e \): Then \( \text{rk}(e) = r \), and \( \Theta = (e, ?) \). \( \Upsilon \) is derived from \( \Theta' := (e, v) \).
   From \( \Sigma \leq r \) \( \Phi(e) \) it follows that \( \Sigma' := \Sigma \setminus \{ (e, ...) \} \) is still correct, and that \( (\Sigma f)_{>r} \subseteq \Theta' \).
   From \( e \notin \text{dom}(\Sigma) \) and \( \Theta' \subseteq (e, v) \), \( \Upsilon \) it follows that \( \Theta' * \Sigma' \) is defined.
   Since \( \Sigma' \leq r \) \( \Phi(e) \), we also get \( \Theta' * \Sigma' = (e, v) \). \( \Upsilon \subseteq (e, v) \), \( (\Sigma f)_{>r} \subseteq \Theta' \).
   So \( d * \Sigma \) is obtained from \( d_{\text{Fr}} \Sigma \) by an H-inference of the same kind.
5. Axioms: If \( \Theta \) is an axiom then \( \Theta * \Sigma \) is an axiom of the same kind.

We continue to investigate admissibility of weakening .

Definition 30 \((\Theta_0, ..., \Theta_n)\) is an \( r \)-path (for \( \Theta_n \)) if it is a path in some \( r \)-deduction of \( \Theta_0 = \emptyset \).

Lemma 6.4
Let \((\Theta_0, ..., \Theta_n)\) be an \( r+1 \)-path for \( \Theta := \Theta_n \). Let \( \Sigma \subseteq r \) be a correct sequent such that \( \Theta_{<r} \subseteq \Sigma \).
Then \( \Theta * \Sigma \) is defined, and there exists a deduction of \( \Sigma \) from \( \Theta * \Sigma \) consisting only of Fr- and H-inferences of ranks \( \Gamma > r \).
Proof by induction on n: Let $n > 0$ and $\Theta' := \Theta_{n-1}$.

$\Theta \ast \Sigma$ is defined, since $\Theta_{<r} \subseteq \Sigma = \Sigma_{<r}$. Note that $\Theta'_{<r} \subseteq \Theta_{<r}$, and therefore by I.H. there is a deduction of $\Sigma$ from $\Theta' \ast \Sigma$. We now show that either $\Theta' \ast \Sigma = \Theta \ast \Sigma$ or $\Theta' \ast \Sigma$ derives from $\Theta \ast \Sigma$ by a $Fr$ or $H$-inference of rank $> r$. For this we distinguish cases according to the topmost inference in $(\Theta_0, ..., \Theta_n)$, i.e. the inference from $\Theta$ to $\Theta'$.

1. Cut: We have $\Theta = (e, u), \Theta$ and $rk(e) \leq r$. Then the Cut is pruned. Indeed, $(e, u) \in \Sigma$ and thus $\Theta \ast \Sigma = \Theta' \ast \Sigma$.

2. $CutFr$: This cannot happen.

3. $Fr$ with main term $e$, and $\Theta = (e, ?), \Theta'$. Then $rk(e) > r$ and $\Theta \ast \Sigma = (e, ?), \Theta' \ast \Sigma$.

4. $H$ with main term $e$, and $\Theta = (e, v), \gamma \subseteq \Theta \ast \Sigma$ and $\Theta' = (e, ?), \gamma$.

Then $\Theta' \ast \Sigma$ is derived from $\Theta \ast \Sigma$ by $H$. Indeed $rk(e) > r$, $\Theta \ast \Sigma = (e, v), (\gamma \ast \Sigma) \subseteq \Theta \ast \Sigma$, and $\Theta' \ast \Sigma = (e, ?), \gamma \ast \Sigma$.

The $H$-rule applies to $\Theta' \ast \Sigma$ (cf. Definition 24), since the $H$-rule applies to $\Theta'$, and $\Sigma$ is correct.

Applicability of the cut-reduction transformation is restricted more or less to subderivations of ‘good’ derivations of an empty sequent $\emptyset$.

**Lemma 6.5**

Let $d$ be a derivation ending with a cut $\mathcal{C}$ of rank $r$ such that the immediate subderivations of $d$ are $r^+$-derivations, and there exists an $(r+1)$-path for the endsequent $\Theta$ of $d$.

Then there exists an $r^+$-derivation $d'$ of $\Theta$ with $h(d') \leq h(d) + \omega + h(d)$.

**Proof:** Let $e$ be the main term of $\mathcal{C}$, and $d_e$ the immediate subderivations of $d$.

We transform $d$ as follows:

1. The cut $\mathcal{C}$ is turned into an inference $CutFr$ by changing every sequent $(e, ?), \Theta'$ of $d_{\Theta'}$ into $(e, ?), \Theta'$.

2. The only rules that are damaged by this transformation are axioms $AxH$ of the form $(e, ?), \gamma$, which now become $(e, ?), \gamma$. At each of these points we extend the deduction by the corresponding $H$-inference, obtaining thereby the new top sequent $(e, v), \gamma$.

3. By Lemma 5.3 $\Sigma = \Sigma_{<r}$ is correct, and by 6.2a,b(1) we get $\Theta_{<r} = \Theta_{<r} \setminus \Theta \subseteq \Sigma$.

4. By Lemma 6.4 $\Theta \ast \Sigma$ is defined, and there exists a deduction of $\Sigma$ from $\Theta \ast \Sigma$ consisting only of $Fr$- and $H$-inferences of ranks $> r$.

5. To derive top sequents $\Theta \ast \Sigma$ in (3) we consider now the subderivation $d_e$ of $\Theta' := (e, v), \Theta$ in the original derivation $d$.

$\Theta \ast \Sigma$ is defined, since $\Theta \ast \Sigma$ is defined and $(e, v) \in \Sigma$. By Lemma 6.2b(2),(3) (applied to $d_{\Theta'}$) we have $((e, ?), \gamma)_{\gamma \subseteq \Theta \ast \Sigma}$, and $((e, ?), \gamma)_{\gamma \subseteq \Theta \ast \Sigma}$.

6. By 6.2a (applied to the $(r+1)$-path for $\Theta$) we have $\Theta \ast \Sigma > r$. Hence $(\Sigma \ast \gamma)_{\gamma \subseteq \Theta}$ and $\Sigma \ast \gamma > r$. Now by Lemma 6.3 there is an $r^+$-derivation $d_e \ast \Sigma$ of $\Theta' \ast \Sigma$, i.e. of $\Theta \ast \Sigma$.

The structure of $d'$ is the following: to some top of the deduction $d_{\Theta'}$ of $(e, ?), \Theta$ (which is an $r^+$-derivation) one-branch deductions from Lemma 6.4 are added, and the tops of the latter deductions are $r^+$-derived by Lemma 6.3. Hence the entire derivation is an $r^+$-derivation.

Now cut reduction is iterated in the standard way.

**Lemma 6.6**

If $d$ is an $r+1$-derivation of $\Theta$, and $\Theta$ has an $r+1$-path then there exists an $r^+$-derivation $d'$ of $\Theta$ with $h(d') \leq \omega^{h(d)+1}$.

**Proof by induction on $h(d)$:**

If $h(d) = 0$, i.e. $d$ consists of an axiom, the assertion is obvious. Assume $h(d) > 0$.

Let $I$ be the last inference of $d$. Let $(d_u)_{u \in I}$ be the family of immediate subderivations of $d$, and $\Theta_u$ the endsequent of $d_u$. Then by I.H. for each $u \in I$ we have an $r^+$-derivation $d'_u$ of $\Theta_u$ with $h(d'_u) \leq \omega^{h(d_u)+1} \leq \omega^{h(d)+1}$. Let $d^+$ be the derivation of $\Theta$ which is obtained from the family $(d'_u)_{u \in I}$ by $I$.

Obviously $h(d^+) \leq \omega^{h(d)+1} + 1$.

1. $I$ is not a cut of rank $r$: Then $d^+$ is an $r^+$-derivation of $\Theta$.

2. $I$ is a cut of rank $r$: In that case we apply Lemma 6.5 to $d^+$, and obtain an $r^+$-derivation $d'$ of $\Theta$ with $h(d') \leq h(d^+) + \omega + h(d^+) < \omega^{h(d)+1}$.

□

17
Lemma 6.7
If \( d \) is an \( \omega \)-derivation of \( \Theta \), and \( \Theta \) has an \( \omega \)-path \( f \) in which all cuts are of rank \( \leq r \) (with \( r < \omega \)) then there exists an \( r^+ \)-derivation \( d' \) of \( \Theta \) with \( h(d') \leq \epsilon_{h(d)+1} \).

Proof by induction on \( h(d) \):
1. Suppose that \( d \) ends in a cut of rank \( r + n \). Let \( d_u \) be the \( \omega \)-subderivations of its premises. By I.H. each \( d_u \) can be transformed into an \( (r + n)^+ \)-derivation \( d'_u \). By replacement of \( d_u \) by \( d'_u \) for all \( u \) the derivation \( d \) is transformed into a derivation \( c \) with \( h(c) = \sup_u(h(d'_u) + 1) \leq \sup_u(\epsilon_{h(d_u)+1} + 1) \leq \epsilon_{h(d) + 1} \).
To the derivation \( c \) we apply Lemma 6.5 and obtain an \( (r+n)^+ \)-derivation \( d' \) of \( \Theta \) with \( h(d') \leq h(c) + \omega + h(c) < \epsilon_{h(d)+1} \).

Now \( n \) applications of Lemma 6.1.6.6 yield the claim.
Note that for each \( i \in \mathbb{N} \), \( f \) is an \((r+i+1)\)-path, and \( (\alpha < \epsilon_{h(d)+1} \Rightarrow \omega^{\alpha+1} < \epsilon_{h(d)+1}) \).
2. Suppose that \( d \) ends in some other inference \( I \). Again by I.H. each immediate subderivation \( d_u \) transforms into an \( r^+ \)-derivation \( d'_u \). By replacement of \( d_u \) by \( d'_u \) for all \( u \) the derivation \( d \) is turned into an \( r^+ \)-derivation \( d' \) with \( h(d') = \sup_u(h(d'_u) + 1) \leq \sup_u(\epsilon_{h(d_u)+1} + 1) \leq \epsilon_{h(d)+1} \).

Lemma 6.8 (Cut-elimination)
Let \( d \) be an \( r \)-derivation of \( \varnothing \) with \( r < \omega + \omega \) and \( h(d) < \epsilon_0 \).
Then there exists a \( 0 \)-derivation \( d' \) of \( \varnothing \) with \( h(d') < \epsilon_0 \).

Proof: w.l.o.g. \( r = \omega + n \). Then \( n \) applications of 6.6.6.1 yield an \( \omega \)-derivation \( d^* \) of \( \varnothing \) with \( h(d^*) < \epsilon_0 \). To \( d^* \) we apply 6.7, 6.1 and obtain a \( 0 \)-derivation \( d' \) of \( \varnothing \) with \( h(d') \leq \epsilon_{h(d^*)+1} < \epsilon_0 \).

The next Lemma says in fact that a cut-free derivation of the empty sequent is a protocol of a terminating \( H \)-process.

Lemma 6.9
A \( 0 \)-derivation \( d \) of \( \varnothing \) consists of exactly one branch and the following holds:
(a) all sequents in \( d \) are correct;
(b) the top sequent of \( d \) is an axiom \( \mathsf{AxS} \), and all other inferences in \( d \) are of the kind \( \mathsf{Fr} \) or \( \mathsf{H} \).

Proof.
(a) Proof by bottom-up induction: \( H : \) cf. Lemma 5.3. \( Fr : \) If \( \Theta \) is correct and \( e \notin \text{dom}(\Theta) \) then \((e,?)\), \( \Theta \) is correct.
(b) Since \( d \) is a \( 0 \)-derivation, there are no \( \mathsf{Cut} \) or \( \mathsf{CutFr} \)-inferences in \( d \). Hence \( d \) is linear. By bottom-up induction we obtain \( \Theta f = \varnothing \) for each sequent \( \Theta \) in \( d \). Since \( d \) is wellfounded, there exists a top sequent \( \Upsilon \). This has to be an axiom. By (a) \( \Upsilon \) is not \( \mathsf{Fr} \). Since \( \Theta f = \varnothing \), \( \Upsilon \) cannot be an axiom \( \mathsf{AxH} \).

6.3 Construction of the original derivation
Here we construct the tree of finite \( \epsilon \)-substitutions mentioned at the beginning of the section 6. General idea here is the same as in [12]. At each stage leaves of the tree are extended (by the bottom-up application of \( \mathsf{Cut} \)) to make them "more computed" till the axioms are reached. Subterms of the non-computed (but needed) \( \epsilon \)-terms of maximum rank are computed till these maximum \( \epsilon \)-terms can be reduced to a canonical form and then computed. Note that the following definitions are stated for a given substitution \( S \), and not for its completion \( \overline{S} \).

Definition 31
Let \( S \) be an \( \epsilon \)-substitution and \( \Phi \) a finite set of closed formulas.

\[\begin{align*}
\rho_S(\Phi) & := \max \{ \text{rk}(A) : A \in \Phi, \ d(A) > \{0\} \} \cup \{0\} \\
\nu_S(\Phi) & := \omega \cdot r + \# s(\Phi, r) \text{ where } r := \rho_S(\Phi) \\
\# s(\Phi, r) & := d_s(\{A_i|A_i \in \Phi\}) + \ldots + d_s(\{A_n|A_n \in \Phi\}) \text{ where } \Phi = \{A_1, \ldots, A_n\} \text{ without repetitions}, \text{ } \quad \text{ if } \text{rk}(F) < r \\
& := \left\{ \begin{array}{ll}
0 & \text{if } \text{rk}(F) < r \\
d(F) & \text{otherwise}
\end{array} \right.
\end{align*}\]

Note that \( \rho_S(\Phi) < \omega + \omega \), \( \# s(\Phi, r) < \omega^\omega \), \( \nu_S(\Phi) < \omega^{\omega+1} \cdot 3 \).
Lemma 6.10 (One step of extension)
Let $S$ be an $e$-substitution and $\Phi$ a finite set of closed formulas.
Let $e = e(F), \not\in \text{dom}(S)$ be a canonical $e$-subterm of a formula $|A_0|_S$ with $A_0 \in \Phi$. $\mathrm{rk}(|A_0|_S) = \rho_S(\Phi)$. 
Let $u \in B_{\Phi}(e)$ or $\{e\}$, $S' := S \cup \{e, u\}$, and $\Phi' := \{u = ? \text{ if } u \not\in \Phi \text{ else } \Phi \cup \{F[u]\}\}$. 
Then $\rho_{S'}(\Phi') \leq \rho_S(\Phi)$ and $\nu_{S'}(\Phi') < \nu_S(\Phi)$. 

Proof:
(a) Since $S \subseteq S'$, we have $|w|_{S'} = |w|_S$ and thus $\mathrm{rk}(|w|_{S'}) \leq \mathrm{rk}(|w|_S), d(|w|_S) \leq d(|w|_S)$ for each $w$.
Let $r := \rho_S(\Phi)$ and $r' := \rho_{S'}(\Phi')$.
One easily sees that $r' \leq r$. Indeed, $\mathrm{rk}(|F[u]|_S) \leq \mathrm{rk}(|F[u]|) < \mathrm{rk}(e) \leq r$, and for remaining terms $w$ occurring in $\Phi$ of $(a)$.
Let $r' = r$ (for $r' < r$ the claim is trivial: $\nu_S(\Phi') < \omega^\omega \cdot (r' + 1) = \omega^\omega \cdot r$).
For each $A \in \Phi$ we have $\mathrm{rk}(|A|_{S'}) \leq \mathrm{rk}(|A|_S), d(|A|_S) \leq d(|A|_S)$ and thus $d_r(|A|_{S'}) \leq d_r(|A|_S)$.
Moreover, $|A_0|_S$ is $S'$-reducible and $\mathrm{rk}(|A_0|_S) = r$. Hence $d_r(|A_0|_{S'}) < d_r(|A_0|_S)$ (Lemma 3.3).
Finally $d_r(|F[u]|_S) = 0$, since $\mathrm{rk}(|F[u]|_S) < r$. 

Lemma 6.11 (Rank reduction)
Let $\Theta$ be a sequent, $L$ a finite set of closed formulas, and $r := \rho_\Theta(F(\Theta) \cup L)$ (cf. Definition 19).
Then there is a deduction $d$ of $\Theta$ by cuts of ranks $\leq r$ from computing sequents $\Upsilon$ containing $\Theta$ and computing all formulas in $L$. Moreover $h(d) \leq \nu_\Theta(F(\Theta) \cup L)$.

Proof by induction on $\nu_\Theta(F(\Theta) \cup L)$:
Let $\Phi := F(\Theta) \cup L$. If $\Theta$ computes all formulas in $\Phi$ we are done.
Otherwise there exists a canonical $e$-subterm $e = e(F)$ of a formula $|A_0|_\Theta$ with $A_0 \in \Phi$, $\mathrm{rk}(|A_0|_\Theta) = r$.
Let $u \in B_{\Phi}(e)$ or $\{e\}$ and $\Theta' := \Theta \cup \{e, u\}$. Then $F(\Theta') = \Theta$ if $u = ?$ then $F(\Theta') = \Theta \cup \{F[u]\}$.
By the Lemma 6.10 $r' := \rho_\Theta(F(\Theta') \cup L) \leq r$ and $\nu_\Theta(F(\Theta') \cup L)$.
Hence (by I.H.) there exists a deduction $d_\Upsilon$ of $\Theta'$ by cuts of ranks $\leq r'$ from computing sequents $\Upsilon$ containing $\Theta'$ and computing all formulas in $L$, and $h(d_\Upsilon) \leq \nu_\Theta(F(\Theta') \cup L)$.
A cut with main term $e$ yields the desired deduction $d$. 

Lemma 6.12
There exists an $r < \omega + \omega$ and an $r+1$-derivation $d$ of the empty sequent containing only axioms and cuts. In addition we have $h(d) < \omega^{r+2}$.

Proof:
First apply Lemma 6.11 to the empty sequent and the set $L := \{C_{r_0}, ..., C_{r_N}\}$. Let $r := \rho_\Theta(L)$, and consider an arbitrary top sequent $\Theta$ of the resulting $r+1$-deduction $d^o$ which is not an axiom. Then $\Theta$ is cc, deciding and nonsolving. Since the only inferences in $d^o$ are cuts of rank $\leq r$, we have $\Theta = \emptyset$ and $\mathrm{rk}(e) \leq r$ for each $e \in \text{dom}(\Theta)$. Let $e$ be the $\Pi$-term of $\Theta$. Since $\Theta$ is deciding, $e$ is $\Theta$-computable (cf. Lemmas 3.7c and 3.8b). Together with Lemma 5.1a and $\Theta t = \emptyset$ this implies $\{e, ?\} \in \Theta$. Since $\Theta$ is not an axiom $\Theta$ it follows that $C_\Theta(e)$ is of the form $F[t] \rightarrow F[exF]$ and $e = ex.A$ with $A := |F|_\Theta$. By Lemma 5.2 $A[n] \leftrightarrow_\Theta \text{TRUE}$ for $n := |t|$.
Now let $L' := \{A[n-1], ..., A[0]\}$. Then $\rho_\Theta(F(\Theta) \cup L') \leq r$. Apply Lemma 6.11 to $\Theta, L'$ and consider any cc top sequent $\Upsilon$ of the resulting $r+1$-deduction. $\Upsilon$ contains $\Theta$ and computes all formulas in $L'$. Now Lemma 5.2b yields that $\Upsilon$ is an axiom $\Theta$ with main term $e$. 

Theorem 6.13 . The $\Pi$-process terminates.

Proof. Combine Lemmata 6.12, 6.8, 6.9. Cf. [12].

6.4 Producing a substitution in terms of finite predicates

6.4.1 Soundness

Lemma 6.14. Let $S$ be a correct and total $e$-substitution. Then all closed axioms of $E\Theta e$ except maybe critical formulas are satisfied by $S$. Modus ponens rule preserves truth under $S$. 

19
Proof. All instances of propositional tautologies and defining axioms for predicate constants are satisfied by $S$ by the Lemma 3.7. Modus ponens preserves truth under $S$, since values of composite formulas are calculated according to standard boolean rules. Equality axioms are satisfied, since by Lemma 3.7c: if $[t]_S = [u]_S$ for 0-terms $t$, $u$ then $[\varepsilon t]_S = [\varepsilon u]_S$ for any expression $e$.

Consider a minimality axiom $\varepsilon F x = S \rightarrow \neg F' x$. Assume $[\varepsilon F x] = [S]_S \in \text{TRUE}$ and let $n := [F]_S$. Then $(\varepsilon [F]_S, n+1) \in S$ and by correctness and Lemma 3.7 we have $[F[n+1]]_S = \neg [F[n+1]]_S \in \text{TRUE}$, in particular $[\neg \varepsilon F x]_S = [\neg F' x]_S \in \text{TRUE}$.  

6.4.2 $\epsilon$-free derivation

Let $d^*$ be a closed derivation in $EAx$, and let $S$ be a correct, total, and solving $\epsilon$-substitution for the system $C_0, \ldots, C_N$ of critical formulas of $d^*$. (For example $S = \overline{S}_n$ where $S_n$ is produced by the $H$-process for $C_0, \ldots, C_N$, cf. Definition 22.) Since $S$ is correct and total, all axioms of $d^*$ except critical formulas are satisfied by $S$ and modus ponens rule preserves truth under $S$. Since $S$ is solving, critical formulas are satisfied, too. Hence all formulas in $d^*$ are true under $S$.

Closed formulas are constructed by propositional connectives $\varepsilon$ from atomic formulas of the form $p_1 \ldots p_n$ and $\text{App}(\epsilon X F) e_1$ where $e_i$ are numerals or closed 0-$\epsilon$-terms possibly preceded by several $S$, and $\epsilon X F$ is a closed 1-term. Let $M$ be the maximum of all numerals, including all values $|n|_S$ of all 0-terms $u$ mentioned in all computations above needed to verify the axioms of the derivation $d^*$.

Replace exterior occurrences of 1-epsilon-terms $\epsilon X F$ in $d^*$ by finite predicates

$$\{n \leq M \mid |\text{App}(\epsilon X F)|_S \in \text{TRUE}\}$$

and exterior occurrences of 0-epsilon-terms $u$ by their $S$-values. We obtain an $\epsilon$-free derivation from formulas which are true under the standard interpretation of predicate constants and boolean connectives, and the rule

$$(\text{App} \{n_1, \ldots, n_k\}(n) \text{ is true}) \quad \frac{\text{Rule}}{\text{H}} \quad (n \text{ occurs among } n_1, \ldots, n_k).$$

Note that thereby an $EAx$ derivation $d^*$ of a formula $F[\varepsilon F] \equiv (\exists x F)^*$ or $F[\varepsilon X F] \equiv (\exists X F)^*$ with $F$ quantifier- and $\epsilon$-free is transformed into a derivation of $F[n]$ or $F[P]$ for some numeral $n$ or finite predicate $P$.

7 Non-constructive proof of termination

In this section, $S, S_n, \ldots$ always denote $\epsilon$-substitutions with $\{e \in \text{dom}(S) : S(e) = \varepsilon\} = \emptyset$. For each pair $(e, u)$ we set $\text{rk}(e, u) := \text{rk}(e)$.

Definition 32 Let $S$ be an $\epsilon$-substitution such that $\overline{S}$ is correct and nonsolving.
Let $e$ be the $H$-term and $v$ the $H$-value of $S$.
We set $\text{rk}(S) := \text{rk}(e)$ and $\pi(S) := (e, v)$.

Note that if $\overline{S}$ is correct and nonsolving, then according to our general assumption on $S$ we have $e \notin \text{dom}(S)$ and $H(S) = S_{\leq \text{rk}(e)} \cup \{(e, v)\}$.

Definition 33 An $\epsilon$-substitution $S$ is called r-substitution ($r \in \text{On}$) iff $\overline{S}$ is correct and $\text{rk}(\pi) < r$ for all pairs $\pi \in S$, i.e. $S = S_{< r}$.

Definition 34 Let $r \in \text{On}$. An $r$-process is a sequence $\{S_i\}_{i \leq r}$ such that:
- $0 < r \leq \omega$,
- $S_0$ is an $r$-substitution,
- if $i < r$ and $\overline{S_i}$ is nonsolving and $\text{rk}(S_i) \geq r$ then $i + 1 < r$ and $S_{i+1} = H(S_i)$,
- if $i < r$ and $(S_i$ is solving or $\text{rk}(S_i) < r$) then $\nu = i + 1$.
(In this case $S_i$ is the last substitution, and $S_r = S_{i+1}$ is not defined.)

20
Remark
1. For each r-substitution S there is a unique r-process (S_i)_{i<\nu} with S = S_0.
2. If (S_i)_{i<\nu} is an r-process then
   a) for all i < \nu, S_i is correct and (S_i)_{<\nu} = S_0.
   b) for all i with i+1 < \nu, \overline{S_i} is nonsolving and rk(S_i) \geq r.

The values of rank r are preserved in an r-process.

Lemma 7.1
If (S_i)_{i<\nu} is an r-process and i \leq j < \nu then (S_i)_{i\leq r} \subseteq S_j.

Proof: If i+1 < \nu then S_{i+1} = (S_i)_{i\leq r} \cup \{(e, v)\} with rk(e) \geq r, and therefore (S_i)_{i\leq r} \subseteq S_{i+1}. From this the claim follows by induction on j. □

An infinite r-process if it existed, would necessarily introduce values of rank > r.

Lemma 7.2 If (S_i)_{i\in\omega} is an r-process then \forall k \exists i \geq k (rk(S_i) > r).

Proof:
Assumption: k \in \mathbb{N} & \forall i \geq k (rk(S_i) = r). — We write \kappa_i for \kappa_{S_i}.
(1) For each canonical e-term e there is an n such that \forall i \geq n (\kappa_i = \kappa_{S_i}).
Proof: 1. rk(e) = r and e \not\in \bigcup_{i < \nu} dom(S_i): Then \kappa_{S_i} = 0(i).
2. rk(e) = r and e \in dom(S_i): By Lemma 7.1 it follows that \kappa_i = \kappa_{S_i} for all i \geq n.
3. rk(e) < r: Then \kappa_i = \kappa_{S_i} for all i.
4. rk(e) > r: By assumption we have \forall i > k (e \not\in dom(S_i)) and therefore \kappa_i = 0(i) for all i > k. □
(2) For each expression e there is an n such that \forall i \geq n (\kappa_i = \kappa_{S_i}).
Proof by induction on d(e). If d(e) = 0 then \kappa_i = e = \kappa_{S_i}. Assume now that d(e) > 0 and let u be some canonical e-subterm of e. By (1) there is an m such that \kappa_i = \kappa_{S_i} for all i \geq m. Let e' result from e by 'contraction' of u. Then e \rightarrow_{S_i} e' for all i \geq m, and by I.H. there is an n \geq m such that \kappa_i = \kappa_{S_i} for all i \geq m. Hence \kappa_i = \kappa_{S_i} for all i \geq n. □
By (2) there is an n \geq k such that \kappa_{S_{i+1}} = \kappa_{S_i} for each critical formula F_0 \rightarrow F[\xi F] from the list \Gamma_0, ..., \Gamma_N. So especially for the H-term e = \xi F, \forall i \geq n such that \kappa_{S_{i+1}} = \kappa_{S_i} for each critical formula F_0 \rightarrow F[\xi F].

Each infinite r-process can be extended to an infinite r+1-process.

Theorem 7.3
Let (S_i)_{i\in\omega} be an r-process. Then (by adding pairs of rank r) S_0 can be extended to an (r+1)-substitution S+ such that the (r+1)-process beginning with S+ is infinite.

Proof: Again we set \kappa_i := \kappa_{S_{i+1}}.
S+ := \bigcup_{i\in\omega} (S_i)_{i<\nu}. By Lemma 7.1 S+ is a function. We show that the (r+1)-process beginning with S+ is an accelerated version of the given infinite r-process (S_i)_{i\in\omega}. (cf. (2) below).
(1) S+ is correct.
Proof: Let (e\xi A, u) \in (S_i)_{i<\nu}. Then for each v \in \mathbb{B}_{i\in\omega}, rk(A[v]) \leq rk(e\xi A) \leq r and therefore \{A[v]\}_i := \{A[v]\}_{S_i}. Hence \{A[u]\}_i := \{A[u]\}_{S_i}. Since S_i is correct, we have \{A[u]\}_i \in \text{TRUE}.
Let (S_{i+1})_{i<\nu} be the (r+1)-process starting with S+. Abbreviation: \kappa_i := \kappa_{S_{i+1}}.
The following proposition (2a) together with (1) yields the theorem.
(2) For every i \in \omega holds
(a) i < \nu.
(b) there exists j \in \omega such that S+ is an extension of S_j by some pairs of rank r.
Proof by induction on i:
I. i = 0: (a) trivial. (b) Take j := 0.
II. i \rightarrow i+1:
By I.H. \( i < \nu \), and there is a \( j \) such that \( S_{+i} = S_j \cup \Psi \) with \( \forall \pi \in \Psi(\pi(\tau) = r) \).
(The symbol \( \cup \) indicates disjoint union).

Let \( \Delta \) be the finite set of all pairs \( \tau \in S_+ \) of rank \( r \) which are used in the computation of \( [C_{r, f}^{\Omega}], \ldots, [C_{r, f}^{\Omega}] \) (\( I = 0, \ldots, N \)) as well as \( \pi(S_{+0}), \ldots, \pi(S_{+i}) \).

Let \( k = \min \{ \ell \geq j \colon \Delta \subseteq S_{k} \land \text{rk}(S_k) > r \} \) (cf. Lemma 7.2). We have \( k \geq j \land \Delta \subseteq S_k \land \text{rk}(S_k) > r \)

CASE A: \( k = j \).
(a) Then \( S_{+i} = S_k \cup \Psi \) and \( \Psi \) is not used in the computation of \( [C_{r, f}^{\Omega}] \) since \( S_k \) already contains \( \Delta \). Hence \( [C_{r, f}^{\Omega}] = [C_{r, f}^{\Omega}] \) for \( I = 0, \ldots, N \). Since \( S_{+i} \) is non-solving and \( \text{rk}(S_k) > r \), it follows that \( S_{+i} \) is non-solving, \( \pi(S_{+i}) = \pi(S_k) \), and \( \text{rk}(S_{+i}) = r + 1 \). Hence \( i + 1 < \nu \).
(b) \( S_{+i+1} = (S_{+i})_{\leq \text{rk}(S_{+i})} \cup \{ \pi(S_{+i}) \} = (S_k \cup \Psi)_{\leq \text{rk}(S_k)} \cup \{ \pi(S_k) \} = (S_k)_{\leq \text{rk}(S_k)} \cup \{ \pi(S_k) \} \cup \Psi = S_{k+1} \cup \Psi \).

CASE B: \( j < k \). Then \( \text{rk}(S_{k-1}) = r \). Otherwise \( (S_k)_{\leq \text{rk}(S_k)} \cup \{ \pi(S_k) \} \cup \Psi = S_{k+1} \cup \Psi \).

Hence in the step from \( S_{k-1} \) to \( S_k \) all pairs of rank \( r \) are removed. It follows that \( S_{+0} = S_k \cup \Phi \) where \( \Phi \) consists of pairs of rank \( r \) which are not used in the computation of \( [C_{r, f}^{\Omega}], \ldots, [C_{r, f}^{\Omega}] \) and \( \pi(S_{+i}) \).

**Proposition.** \( S_{+i} = S_{k+i} \cup \Phi \), for \( i = 0, \ldots, \min \{ i + 1, \nu - 1 \} \) (\( * \)).

Proof by induction on \( i \): Let \( i \leq \min \{ i + 1, \nu - 1 \} \) and \( S_{+i} = S_{k+i} \cup \Phi \).

By definition of \( \Phi \), all pairs of rank \( r \) which are used in the computation of \( [C_{r, f}^{\Omega}], \ldots, [C_{r, f}^{\Omega}] \) and \( \pi(S_{+i}) \) are contained in \( \Delta \subseteq S_k \). Since \( i \leq j \), we have \( \pi(S_{+i}) = \pi(S_{k+i}) \) and \( \text{rk}(S_{+i}) = \text{rk}(S_{k+i}) \). Since \( i + 1 < \nu \), the substitution \( S_{+i} \) is non-terminal, and we have \( \text{rk}(S_{+i}) = r + 1 \). Hence \( S_{+i+1} = (S_{+i})_{\leq \text{rk}(S_{+i})} \cup \{ \pi(S_{+i}) \} = (S_{k+i} \cup \Phi)_{\leq \text{rk}(S_{k+i})} \cup \{ \pi(S_{k+i}) \} = S_{k+i+1} \cup \Phi \).

(a) The above proposition \( * \) yields \( [C_{r, f}^{\Omega}] = [C_{r, f}^{\Omega}] \) since \( \Psi \) is not used in the computation of \( [C_{r, f}^{\Omega}] \).

Hence \( S_{+i} \) is non-solving, since \( S_{k+i} \) is non-solving. Now by \( * \) we also have \( \pi(S_{+i}) = \pi(S_{k+i}) \) and \( \text{rk}(S_{+i}) = \text{rk}(S_{k+i}) \). Assume \( \text{rk}(S_{k+i}) = r \); then \( \pi(S_{k+i}) \in S_+ \cup S_{+i} \) and thus \( \pi(S_{k+i}) \neq \pi(S_{+i}) \), since \( \pi(S_{+i}) \in S_+ \). Contradiction.

Hence \( \text{rk}(S_{+i}) = \text{rk}(S_{k+i}) \geq r + 1 \) and thus \( i + 1 < \nu \).

(b) As in the proof of \( * \) we get \( S_{+i+1} = S_{k+i+1} \cup \Phi \). \( \square \)

**Theorem 7.4**
The \( \emptyset \)-process \( \Pi \) beginning with the empty substitution \( \emptyset \) terminates in a solving substitution.

**Proof:** Obviously it suffices to prove that \( \Pi \) is finite (terminates).

For contradiction we assume that \( \Pi \) is infinite.

Below we define substitutions \( S^r \) for all \( r \in \mathbb{N} \) such that:

1. \( S^0 = \emptyset \).
2. \( S^r \) is an \( r \)-substitution.
3. the \( \emptyset \)-process starting with \( S^r \) is infinite.
4. \( S^q \subseteq S^q \) for all \( q < r \).

Let \( R := \max \{ \text{rk}(C_{r, f}) : r \leq N \} + 1 \). Then \( \text{rk}(S) < R \) for each substitution \( S \). But on the other hand \( \text{rk}(S^R) \geq R \), since by (3) the \( \emptyset \)-process starting with \( S^R \) is infinite. Contradiction.

Definition of \( S^r \) by transfinite recursion on \( r \):

\( S^0 := \emptyset \).

\( S^{r+1} := (S^r) + \) (cf. Theorem 7.3),

\( S^r := \bigcup_{q < r} S^q \), if \( r \in \text{Lim} \).

In parallel with that definition we prove by transfinite induction on \( r \) that \( S^r \) satisfies the above conditions (2), (3), (4). The successor step is settled by theorem 7.3. Now assume \( r \in \text{Lim} \).

Then (4) is trivial, and (2) follows from I.H. (2), (4), since each particular value \( (e, u) \) in \( S^r \) is verified by computation in some \( S^q \), \( q < r \).

For (3) assume that the \( r \)-process for \( S^r \) is finite. Then it uses information only from \( S^q \) for finitely many \( q < r \), hence there is a \( q < r \) such that \( S^q \) is finite which contradicts I.H. (3). \( \square \)
References