The finite stages of inductive definitions *

Robert F. Stärk**

Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104

Summary. In general, the least fixed point of a positive elementary inductive definition over the Herbrand universe is Π_1^1 and has no computational meaning. The finite stages, however, are computable, since validity of equality formulas in the Herbrand universe is decidable. We set up a formal system BID for the finite stages of positive elementary inductive definitions over the Herbrand universe and show that the provably total functions of the system are exactly that of Peano arithmetic. The formal system BID contains the so-called inductive extension of a logic program as a special case. This first-order theory can be used to prove termination and correctness properties of pure Prolog programs, since notions like negation-as-failure and left-termination can be turned into positive inductive definitions.

1. Why inductive definitions over the Herbrand universe?

In traditional logic programming, the semantics of a program is always given by the least fixed point of a monotonic operator over the Herbrand universe. The first example is the well-known van Emden-Kowalski operator for definite Horn clause programs in [25]. This operator is defined by a purely existential formula and is therefore continuous. The least fixed point of the operator is recursively enumerable. Moreover, the finite stages of the inductive definition are exactly what is computed by SLD-resolution.

In [11], Fitting has generalized the van Emden-Kowalski operator using three-valued logic to programs which may also contain negation in the bodies of the clauses. Although Fitting's operator is still monotonic it is no longer continuous. It follows from Blair [2] and Kunen [14] that the least fixed point of this operator can be Π_1^1-complete and that the closure ordinal can be ω_1^{CK} even for definite Horn clause programs.

The finite stages of Fitting's operator, however, are decidable and correspond to what is computed by SLDNF-resolution. This has been shown by Kunen in [15] for allowed logic programs and by the author in [21] for mode-correct programs. The class of allowed programs is considered as too restrictive in general. The class of mode-correct programs, however, contains most programs of practical interest, since a programmer has always modes in mind when he writes a program. Moreover, every allowed program is also mode-correct.

* This paper is in its final form and no similar paper has been or is being submitted elsewhere.

** Research supported by the Swiss National Science Foundation. This article has been written at the Department of Mathematics, Stanford University.

Even though the use of three-valued logic can be eliminated in Fitting's operator (cf. eg. [13]), the completeness results of [15] and [21] cannot be applied to existing implementations of logic programming like, for example, Prolog. The reason is that these systems use special search-strategies which depend on the order of clauses in the program and on the order of literals in the bodies of clauses. Apt and Pedreschi, however, have observed in [1] that only the order of literals in the bodies is important. They say that most programs used in practice terminate independently of the order of clauses in the program — at least for the intended inputs.

Based on this observation we assign in [22] to a predicate R of a logic program three positive elementary inductive definitions for new relations R^{s}, R^{f} and R^{t}. The relation R^{s} corresponds to Fitting's truth-value true; R^{f} corresponds to the truth-value false; R^{t} expresses left-termination in the sense of Apt and Pedreschi. The corresponding formal system is called the *inductive extension* of a logic programs and can be used for proving termination and correctness properties of Prolog programs (see also [24]).

A natural question is then: What is the proof-theoretic strength of the inductive extension? — We answer this problem below and show that the provably total functions of the inductive extension are exactly those of Peano Arithmetic. As a byproduct we obtain a proof-theoretic proof of a key lemma used in the completeness proofs in [22] and [23]. Our proof-theoretical analysis of the inductive extension in terms of provably total functions is similar to Pohlers' treatment of ID_1 in [7] and [18].

The plan of this article is as follows. After recalling some well-known facts on Clark's equality theory CET in Sect. 2, we present a framework for inductive definitions over the Herbrand universe in Sect. 3 and set up a formal system for such in Sect. 4. Using the number-theoretic ordinal functions of Sect. 5 we embed the formal system into an infinitary sequent calculus in Sect. 6. Partial cut-elimination allows then to consider the positive/negative fragment of the calculus only and to perform an asymmetric interpretation that yields the main results of the article. Sect. 7 finally provides some hints how the general framework for inductive definitions relates to the so-called inductive extension of a logic program and to notions like negation as failure and left-termination.

2. Preliminaries

Let $\mathfrak{L}$ be a set of function symbols. Constants are considered as 0-ary function symbols. We assume that $\mathfrak{L}$ contains at least one constant symbol. Function symbols are denoted by f and g. The terms built up with symbols from $\mathfrak{L}$ are denoted by small letters a, b, s, t with and without subscripts. We write $s = t$ for a formal equations and use $s \equiv t$ to express that s and t are syntactically equal. Clark's equality theory $\mathrm{CET}_{\mathfrak{L}}$ for the language $\mathfrak{L}$ comprises the following axioms (cf. [10]):

1. $x = x$
2. $x = y \to y = x$
3. $x = y \wedge y = z \to x = z$
4. $x_1 = y_1 \wedge \ldots \wedge x_m = y_m \to f(x_1, \ldots, x_m) = f(y_1, \ldots, y_m)$
5. $f(x_1, \ldots, x_m) = f(y_1, \ldots, y_m) \to x_i = y_i$
6. $f(x_1, \ldots, x_m) \neq g(y_1, \ldots, y_n)$ [if $f \not\equiv g$]
7. $x \neq t$ [if $x \in \mathrm{FV}(t)$ and $x \not\equiv t$]

Axiom (5) is called *decomposition axiom*, (6) is called *function clash axiom* and (7) is called *occurs check axiom*. Note that (7) is an axiom scheme. It can not be replaced by finitely many axioms.

Example 2.1. Let $\mathfrak{L} = \{0, s\}$, where 0 is a constant and s is a unary function symbol. Then the decomposition, function clash and occurs check axioms for $\mathfrak{L}$ are:

(a) $s(x) = s(y) \to x = y$,
(b) $s(x) \neq 0$,
(c) $s^n(x) \neq x$ for $n > 0$.

For $\mathfrak{L} = \{nil, cons\}$ we have in $\mathrm{CET}_{\mathfrak{L}}$ among others the following axioms:

(a) $cons(x, y) = cons(u, v) \to x = u$, $cons(x, y) = cons(u, v) \to y = v$,
(b) $cons(x, y) \neq nil$,
(c) $cons(x, y) \neq x$, $cons(x, y) \neq y$, $cons(cons(x, y), z) \neq x$, etc.

The theory $\mathrm{CET}_{\mathfrak{L}}$ is strongly related to unification. In fact, an equivalent axiomatization of $\mathrm{CET}_{\mathfrak{L}}$ consists of the following two schemata for arbitrary terms a, b, s_i, t_i:

1′. $s_1 = t_1 \wedge \ldots \wedge s_n = t_n \to a = b$,
 if $\sigma = \mathrm{mgu}\{s_1 = t_1, \ldots, s_n = t_n\}$ and $a\sigma \equiv b\sigma$,
2′. $\neg(s_1 = t_1 \wedge \ldots \wedge s_n = t_n)$,
 if $\{s_1 = t_1, \ldots, s_n = t_n\}$ is not unifiable.

That (1′) and (2′) follow from (1)–(7) is shown in [10]. Conversely, it is easy to see that any of the axioms (1)–(7) is an instance of (1′) or (2′). Thus the two axiomatizations are equivalent.

We denote by $U_{\mathfrak{L}}$ be the set of all closed terms built up with symbols from $\mathfrak{L}$. The set $U_{\mathfrak{L}}$ is called the *Herbrand universe* of $\mathfrak{L}$. By $\mathfrak{U}_{\mathfrak{L}}$ we denote the algebraic structure with domain $U_{\mathfrak{L}}$ and the free interpretation of the function symbols, i.e. $\mathfrak{U}_{\mathfrak{L}}(f)(t_1, \ldots, t_m) = f(t_1, \ldots, t_m)$ for all function symbols $f \in \mathfrak{L}$ and all terms $t_1, \ldots, t_m \in U_{\mathfrak{L}}$. Equality is interpreted as identity. The structure $\mathfrak{U}_{\mathfrak{L}}$ is a model of $\mathrm{CET}_{\mathfrak{L}}$ and, moreover, every model of $\mathrm{CET}_{\mathfrak{L}}$ contains an isomorphic copy of $\mathfrak{U}_{\mathfrak{L}}$. Sometimes, $\mathfrak{U}_{\mathfrak{L}}$ is called the *standard model* of $\mathrm{CET}_{\mathfrak{L}}$.

Term models of $\mathrm{CET}_{\mathfrak{L}}$ which are non-standard can be obtained by the following construction of [3]. Let $Term_{\mathfrak{L}}$ be the set of all terms of $\mathfrak{L}$ (with variables). Let Ψ be a directed set of substitutions. This means that for all substitutions $\sigma, \tau \in \Psi$ there exists a substitution $\theta \in \Psi$ such that $\sigma \leq \theta$ and $\tau \leq \theta$, where $\sigma \leq \theta$ is defined as $\exists \sigma' (\sigma \circ \sigma' = \theta)$. Define on $Term_{\mathfrak{L}}$ the congruence relation $\sim_{\Psi}$ by

$$s \sim_{\Psi} t \;:\Longleftrightarrow\; \exists \sigma \in \Psi \, (s\sigma \equiv t\sigma).$$

Then the term structure $\langle Term_{\mathfrak{L}}, \sim_{\Psi} \rangle$ is a model of $\mathrm{CET}_{\mathfrak{L}}$, if equality is interpreted by the relation $\sim_{\Psi}$. Moreover, every term model of $\mathrm{CET}_{\mathfrak{L}}$ can be obtained in this way.

Models of $\mathrm{CET}_{\mathfrak{L}}$ are also called *locally free algebras.* In [16], Mal'cev gives an algorithm that transforms an arbitrary first-order formula containing the equality symbol only into a normal form over locally free algebras. A slightly more general algorithm with a different normal form is investigated by Shepherdson in [20]. From both normal form algorithms the following proposition can be derived.

Proposition 2.1 (Mal'cev [16], Shepherdson [20]).

1. *Two models of* $\mathrm{CET}_{\mathfrak{L}}$ *are elementary equivalent if, and only if, they have the same number of indecomposable elements.*
2. $\mathrm{CET}_{\mathfrak{L}}$ *is decidable.*
3. $\mathrm{CET}_{\mathfrak{L}}$ *is complete provided that* $\mathfrak{L}$ *is infinite.*
4. *The first-order theory of* $\mathfrak{U}_{\mathfrak{L}}$ *is decidable.*

In this proposition, an element is called indecomposable if it is not in the range of any function. For example, the Herbrand structure $\mathfrak{U}_{\mathfrak{L}}$ has no indecomposable elements. Same number of indecomposable elements means that either both models have the same finite number of indecomposable elements or both models have infinitely many.

If $\mathfrak{L}$ is finite, then $\mathrm{CET}_{\mathfrak{L}}$ is not always complete. Take $\mathfrak{L} = \{c\}$. Then the formula $\forall x \, (x = c)$ is true in some models but false in other models. Though the assumption of an infinite language $\mathfrak{L}$ is technically very useful because it makes $\mathrm{CET}_{\mathfrak{L}}$ a complete theory, it can be an absurd assumption in practice. Therefore we do not make any assumption on whether $\mathfrak{L}$ is finite or infinite in the following.

3. Inductive definitions over the Herbrand universe

We consider simultaneous positive elementary inductive definitions over the Herbrand universe. Our interest in such definitions comes from the declarative analysis of *negation-as-failure* and *left-termination* in logic programming (cf. Sect. 7). Both notions can be turned into positive inductive definitions of the kind we investigate in the following.

Let $R_1, \ldots, R_k$ be k relation symbols each of a given arity. An *operator form* is a formula in the language $\mathfrak{L}(R_1, \ldots, R_k, =)$ which is positive in

$R_1, \dots, R_k$. More formally, we define Pos to be the set of formulas of the following form:

$$\top \mid \bot \mid s = t \mid s \neq t \mid R_j(\mathbf{t}) \mid A \wedge B \mid A \vee B \mid \forall x\, A \mid \exists x\, A.$$

An operator form $\mathcal{A}[\mathbf{R}, \mathbf{x}]$ is then an element of Pos. By writing $\mathcal{A}[\mathbf{R}, \mathbf{x}]$ we indicate that all the relation symbols of the formula different from the equality symbol are among the list $\mathbf{R} = R_1, \dots, R_k$ and that all the free variables are among the list $\mathbf{x}$.

Let $\mathcal{A}_j[\mathbf{R}, \mathbf{x}_j]$ be operator forms such that the length of $\mathbf{x}_j$ is equal to the arity of R_j for $j = 1, \dots, k$. Associated to these operator forms are monotonic operators

$$\Gamma_j : \mathcal{P}(U_{\mathfrak{L}}^{n_1}) \times \dots \times \mathcal{P}(U_{\mathfrak{L}}^{n_k}) \to \mathcal{P}(U_{\mathfrak{L}}^{n_j}), \quad \text{for } j = 1, \dots, k,$$

where n_j is the arity of R_j. The operator Γ_j is defined by

$$\Gamma_j(X_1, \dots, X_k) := \{\, \langle \mathbf{y} \rangle \in U_{\mathfrak{L}}^{n_j} : \mathfrak{U}_{\mathfrak{L}} \models \mathcal{A}[X_1, \dots, X_k, \mathbf{y}] \,\}.$$

The stages I_j^α of the simultaneous inductive definitions are defined as usual for $j = 1, \dots, k$ (cf. [17]):

$$I_j^\alpha := \Gamma_j(I_1^{<\alpha}, \dots, I_k^{<\alpha}), \quad I_j^{<\alpha} := \bigcup_{\beta < \alpha} I_j^\beta, \quad I_j^\infty := \bigcup_{\alpha \in On} I_j^\alpha.$$

It is clear that the relations I_j^∞ are the least fixed points of the operators Γ_j. By $\mathfrak{I}^\alpha$ we denote the structure $\langle U_{\mathfrak{L}}, I_1^\alpha, \dots, I_k^\alpha \rangle$ and by $\mathfrak{I}^\infty$ the structure $\langle U_{\mathfrak{L}}, I_1^\infty, \dots, I_k^\infty \rangle$. In the following we will mainly be interested in the finite stages $\mathfrak{I}^n$ for $n < \omega$, since these structures are decidable. This can be seen as follows. By induction on the natural number n we define equality formulas $R_j^{(n)}[\mathbf{x}_j]$ which characterize the nth stage I_j^n in the inductive generation of R_j. For $j = 1, \dots, k$ and $n \in \mathrm{N}$ we define

1. $R_j^{(0)}[\mathbf{x}_j] :\equiv \bot$,
2. $R_j^{(n+1)}[\mathbf{x}_j] :\equiv \mathcal{A}_j[R_1^{(n)}, \dots, R_k^{(n)}, \mathbf{x}_j]$.

As usual this notion means that in $\mathcal{A}_j[R_1, \dots, R_k, \mathbf{x}_j]$ all subformulas of the form $R_i(\mathbf{t})$ are replaced by the formula $R_i^{(n)}[\mathbf{t}]$ for $i = 1, \dots, k$. Note, that the size of the formulas $R_j^{(n)}[\mathbf{x}_j]$ grows exponentially in n. For arbitrary formulas $A[R_1, \dots, R_k]$ we define

$$A[R_1, \dots, R_k]^{(n)} :\equiv A[R_1^{(n)}, \dots, R_k^{(n)}].$$

The formulas $A^{(n)}$ can also be constructed in a more explicit way. We define by main induction on n and side induction on the size of a formula A equality formulas $\mathrm{E}_n A$. Since we later use it only for positive formulas, we restrict the definition to this class of formulas.

$$\begin{array}{llll}
\mathrm{E}_0\, R_j(\mathbf{t}) & :\equiv \bot, & \mathrm{E}_n(A \wedge B) & :\equiv \mathrm{E}_n\, A \wedge \mathrm{E}_n\, B, \\
\mathrm{E}_{(n+1)}\, R_j(\mathbf{t}) & :\equiv \mathrm{E}_n\, \mathcal{A}_j[\mathbf{R},\mathbf{t}], & \mathrm{E}_n(A \vee B) & :\equiv \mathrm{E}_n\, A \vee \mathrm{E}_n\, B, \\
\mathrm{E}_n(s = t) & :\equiv (s = t), & \mathrm{E}_n \forall x\, A & :\equiv \forall x\; \mathrm{E}_n\, A, \\
\mathrm{E}_n(s \neq t) & :\equiv (s \neq t), & \mathrm{E}_n\, \exists x\, A & :\equiv \exists x\; \mathrm{E}_n\, A.
\end{array}$$

Since $A^{(n)}$ is the same as $\mathrm{E}_n\, A$, we are free to use both notions in the following interchangeably. $A^{(n)}$ is used for expressing that A is true at the nth stage of the inductive definition and $\mathrm{E}_n\, A$ is used if we want to point out that it is an equality formula.

Lemma 3.1. $\mathfrak{U}_{\mathfrak{L}} \models A^{(n)} \iff \mathfrak{I}^n \models A$.

From this lemma it follows immediately that the finite stages I_j^n are decidable. To test whether $\langle \mathbf{a} \rangle$ belongs to I_j^n one just has to compute the equality formula $R_j^{(n)}[\mathbf{a}]$ and then to apply the decision procedure of Proposition 2.1 for $\mathfrak{U}_{\mathfrak{L}}$.

Lemma 3.2. $\mathrm{E}_n\, A$ *implies* $\mathrm{E}_{n+1}\, A$ *for all* $n \in \mathbb{N}$.

The proofs of both lemmas are standard.

4. The formal system BID

Given a list of relation symbols $\mathbf{R} = R_1, \dots, R_k$ and operator forms $\mathcal{A}_j[\mathbf{R}, \mathbf{x}_j]$ for $j = 1, \dots, k$ we define a first-order theory $\mathrm{BID}(R_1, \mathcal{A}_1; \dots; R_k, \mathcal{A}_k)$. The word BID stands for *basic inductive definitions*. Sometimes we also write $\mathrm{BID}(R_1, \dots, R_k)$ or just BID for short. The theory BID comprises the axioms of $\mathrm{CET}_{\mathfrak{L}}$ and the following two principles for arbitrary formulas $B_1(\mathbf{x}_1), \dots, B_k(\mathbf{x}_k)$ of the language $\mathfrak{L}(R_1, \dots, R_k, =)$:

$$\forall \mathbf{x}_j\ (\mathcal{A}_j[\mathbf{R}, \mathbf{x}_j] \rightarrow R_j(\mathbf{x}_j)) \quad \text{for } j = 1, \dots, k, \tag{CLS}$$

$$\bigwedge_{j=1}^{k} \forall \mathbf{x}_j\ (\mathcal{A}_j[\mathbf{B}, \mathbf{x}_j] \rightarrow B_j(\mathbf{x}_j)) \rightarrow \bigwedge_{j=1}^{k} \forall \mathbf{x}_j\ (R_j(\mathbf{x}_j) \rightarrow B_j(\mathbf{x}_j)). \tag{MIN}$$

We write $\mathrm{BID} \vdash A$ if the formula A is derivable from $\mathrm{CET}_{\mathfrak{L}}$, CLS and MIN in classical predicate logic. The principle CLS expresses that the operator forms are closed under the relations R_j and the principle MIN expresses that the relations R_j are minimal with respect to this property. It is well-known that from CLS and MIN one can derive that the relations R_j are fixed points of the operator forms. Choosing $B_j(\mathbf{x}_j) :\equiv \mathcal{A}_j[\mathbf{R}, \mathbf{x}_j]$ one derives from CLS using the monotonicity property of positive formulas

$$\bigwedge_{j=1}^{k} \forall \mathbf{x}_j\ (\mathcal{A}_j[\mathbf{B}, \mathbf{x}_j] \rightarrow \mathcal{A}_j[\mathbf{R}, \mathbf{x}_j]).$$

By an application of MIN one obtains

$$\forall \mathbf{x}_j \left(R_j(\mathbf{x}_j) \to \mathcal{A}_j[\mathbf{R}, \mathbf{x}_j]\right) \quad \text{for } j = 1, \ldots, k. \tag{FIX}$$

Not only these fixed point principles are derivable but also the following equality axioms for relations. These axioms are usually considered as part of predicate logic:

$$x_1 = y_1 \wedge \ldots \wedge x_{n_j} = y_{n_j} \wedge R_j(x_1, \ldots, x_{n_j}) \to R_j(y_1, \ldots, y_{n_j}). \tag{EQ}$$

To see that these formulas are derivable in BID, define for $j = 1, \ldots, k$ the formulas

$$B_j[x_1, \ldots, x_{n_j}] :\equiv \forall y_1, \ldots, y_{n_j} \left(x_1 = y_1 \wedge \ldots \wedge x_{n_j} = y_{n_j} \to R_j(y_1, \ldots, y_{n_j})\right).$$

Then it is easy to verify that for any positive formula $A[\mathbf{R}, x_1, \ldots, x_n]$ we have

$$A[\mathbf{B}, x_1, \ldots, x_n] \to \forall y_1, \ldots, y_n \left(x_1 = y_1 \wedge \ldots \wedge x_n = y_n \to A[\mathbf{R}, y_1, \ldots, y_n]\right).$$

Using CLS we obtain

$$\bigwedge_{j=1}^{k} \forall \mathbf{x}_j \left(\mathcal{A}_j[\mathbf{B}, \mathbf{x}_j] \to B_j(\mathbf{x}_j)\right).$$

Now we can apply MIN and obtain the equality axioms (EQ).

The question is now, what is the theory BID? How does it compare to other systems? What is the proof theoretic strength of BID? What are the provably total functions of BID? What is the definition of provably total function in BID at all?

Note that unlike other formal theories for inductive definitions like ID_1 or $\widehat{\mathrm{ID}}_1$ the theory BID does not automatically include Peano Arithmetic. Therefore, in a first step, we show how one can embed PA into a theory $\mathrm{BID}(nat, add, mul, leq)$ for suitable predicates nat, add, mul and leq. We assume that the language $\mathfrak{L}$ contains at least the constant 0 and the successor function s. Instead of writing an operator form explicitly as

$$\mathcal{A}[nat, x] :\equiv (x = 0) \vee \exists y \left(x = s(y) \wedge nat(y)\right)$$

we use Prolog-like notation and assume that it is implicitly given by the following clauses:

$$nat(0).$$
$$nat(s(x)) \leftarrow nat(x).$$

The operator forms for *add*, *mul* and *leq* are given by the following clauses:

$$\begin{array}{l} add(0,y,y). \\ add(s(x),y,s(z)) \leftarrow add(x,y,z). \\ mul(0,y,0). \\ mul(s(x),y,z) \leftarrow mul(x,y,u) \land add(u,y,z). \\ leq(0,y). \\ leq(s(x),s(y)) \leftarrow leq(x,y). \end{array}$$

It is easy to see that the theory BID(nat, add, mul, leq) proves the induction axiom for natural numbers as a special case of MIN:

$$A(0) \land \forall x\,(nat(x) \land A(x) \to A(s(x))) \to \forall x\,(nat(x) \to A(x)). \tag{4.1}$$

Using (4.1) one can then derive that the relations *add* and *mul* are the graphs of functions, eg.

$$\forall x, y\,(nat(x) \land nat(y) \to \exists! z\,(nat(z) \land add(x,y,z)).$$

Therefore one can embed PA into BID(nat, add, mul, leq) by interpreting universal quantifiers $\forall x\, A$ by $\forall x\,(nat(x) \to A)$ and existential quantifiers $\exists x\, A$ by $\exists x\,(nat(x) \land A)$.

But can we prove more in BID than in PA? — Is it, for example, possible to extend BID(nat, add, mul, leq) by a truth predicate for the structure of natural numbers? Can we inductively define a relation $tr(a,e)$ expressing that the formula a is true in N under the assignment e? The truth definition could then be used to prove the consistency of PA.

We show that this is not possible, at least not in the obvious way, since a truth definition typically has the following property:

$$tr(forall(x,a),e) \leftrightarrow \forall y\,(nat(y) \to tr(a, cons(sub(x,y),e))). \tag{4.2}$$

The relation *nat* occurs negatively on the right-hand side of (4.2). So we cannot obtain (4.2) directly from an inductive definition.

In an attempt to save the truth definition we define the complement of natural numbers by

$$notnat(x) \leftarrow x \neq 0 \land \forall y\,(x = s(y) \to notnat(y))$$

and write (4.2) as

$$tr(forall(x,a),e) \leftrightarrow \forall y\,(notnat(y) \lor tr(a, cons(sub(x,y),e))). \tag{4.3}$$

This equivalence can be obtained from an inductive definition, since the right-hand side is positive. But there is no way to obtain (4.2) from (4.3), since although BID proves

$$\forall x\, \neg(nat(x) \land notnat(x)), \tag{UNIQ}$$

it does not prove

$$\forall x\,(nat(x) \lor notnat(x)). \tag{TOT}$$

And for the applications we have in mind it is essential that TOT is not provable in BID (see Sect. 7).

One of the important features of BID is that if it proves a positive formula A then there exists a natural number $n \in \mathrm{N}$ such that CET $\vdash \mathrm{E}_n A$ (see Theorem 6.2). Thus if TOT would be provable in CET then there would exist an $n \in \mathrm{N}$ such that TOT would already be true at the finite stage $\mathfrak{I}^n$. This, however, is not the case.

What about the provably total functions of BID(nat, add, mul, leq)? What about the provably total functions of BID in general? — First we have to define what we mean by provably total functions of BID. We assume that the language $\mathfrak{L}$ contains the constant 0 and the successor function s. Moreover, we assume that we have a unary predicate nat defining the natural numbers and a distinguished binary relation symbol R. We indicate this by writing BID($nat, R, \ldots$). For $n \in \mathrm{N}$ we denote by $\overline{n}$ the term

$$\underbrace{s(\cdots(s}_{n \text{ times}}(0)\cdots).$$

Definition 4.1. A function $f\colon \mathrm{N} \to \mathrm{N}$ is called *provably total* in BID($nat, R, \ldots$), if

1. BID $\vdash R(\overline{m}, \overline{n})$ for all $m, n \in \mathrm{N}$ such that $f(m) = n$,
2. BID $\vdash \forall x\,(nat(x) \to \exists! y\,(nat(y) \land R(x,y)))$.

This definition can be justified as follows. Let $f\colon \mathrm{N} \to \mathrm{N}$ be provably total in BID($nat, R, \ldots$). Then Theorem 6.3 below says that there exists an ordinal $\alpha < \varepsilon_0$ and a function $F\colon \mathrm{N} \to \mathrm{N}$ which is α-recursive such that

$$\text{CET} \vdash \forall x\,(\mathrm{E}_m\, nat(x) \to \exists y\,(\mathrm{E}_{F(m)}\, nat(y) \land \mathrm{E}_{F(m)}\, R(x,y))) \quad \text{for all } m \in \mathrm{N}.$$

Let $m \in \mathrm{N}$. Then CET $\vdash \mathrm{E}_m\, nat(\overline{m})$ and thus

$$\text{CET} \vdash \exists y\,(\mathrm{E}_{F(m)}\, nat(y) \land \mathrm{E}_{F(m)}\, R(\overline{m}, y)).$$

Since $\mathfrak{U}_{\mathfrak{L}}$ is a model of CET, there exists a closed term t such that

$$\mathfrak{U}_{\mathfrak{L}} \models \mathrm{E}_{F(m)}\, nat(t) \quad \text{and} \quad \mathfrak{U}_{\mathfrak{L}} \models \mathrm{E}_{F(m)}\, R(\overline{m}, t).$$

This implies that $nat(t)$ is true at stage $\mathfrak{I}^{F(m)}$ and therefore there exists an $n \leq F(m)$ such that $t = \overline{n}$. Thus we have

$$\mathfrak{U}_{\mathfrak{L}} \models \mathrm{E}_{F(m)}\, R(\overline{m}, \overline{n}) \quad \text{and} \quad \mathfrak{I}^{F(m)} \models R(\overline{m}, \overline{n}).$$

Since $\mathfrak{I}^{\infty} \models R(\overline{m}, \overline{n})$ and $\mathfrak{I}^{\infty}$ is a model of BID it follows that $f(m) = n$. Thus we have $f(m) \leq F(m)$ for all $m \in \mathrm{N}$. Moreover, given $m \in \mathrm{N}$ we can compute the value $f(m)$ in the following way. It is the least $n \leq F(m)$ such that

$$\mathfrak{U}_{\mathfrak{L}} \models \mathrm{E}_{F(m)}\, R(\overline{m}, \overline{n}).$$

Since the truth of equality formulas in the Herbrand universe is primitive recursively decidable (cf. Proposition 2.1), if follows that the function f is α-recursive. Thus the provably total functions of BID$(nat, R, \ldots)$ are exactly those computable functions which are provably total in Peano Arithmetic. What remains to show is the above mentioned Theorem 6.3.

5. Some remarks on ordinals less than ε_0

The reader familiar with the relations $<_n$ of Weiermann [26] can skip this section. The purpose of this section is to introduce number-theoretic functions $n \mapsto \theta_\alpha(n)$ which will later be used in an asymmetric interpretation.

Let $\varepsilon_0 := \min\{\xi : \omega^\xi = \xi\}$. Ordinals less than ε_0 are denoted by α, β, γ. Finite ordinals are denoted by i, j, m, n. The natural sum of α and β is denoted by $\alpha \# \beta$. The norm of an ordinal $\alpha < \varepsilon_0$ is the number of occurrences of ω in its Cantor normal form. The norm $\mathrm{N}: \varepsilon_0 \to \mathrm{N}$ is defined by

1. $\mathrm{N}(0) := 0$,
2. $\mathrm{N}(\omega^{\alpha_1} + \ldots + \omega^{\alpha_n}) := n + \mathrm{N}(\alpha_1) + \ldots + \mathrm{N}(\alpha_n)$, if $\alpha_1 \geq \ldots \geq \alpha_n$.

We write $\mathrm{N}\,\alpha$ for $\mathrm{N}(\alpha)$. The norm N has the following properties:

1. $\mathrm{N}\,\omega = 2$,
2. $\mathrm{N}\,\omega^\alpha = 1 + \mathrm{N}\,\alpha$,
3. $\mathrm{N}(\alpha \# \beta) = \mathrm{N}\,\alpha + \mathrm{N}\,\beta$.
4. $N(\alpha + n) = \mathrm{N}\,\alpha + n$ and $\mathrm{N}\,n = n$.

For each $n \in \mathrm{N}$ there are only finitely many ordinals $\alpha < \varepsilon_0$ with $\mathrm{N}\,\alpha \leq n$.

In the definition of the relations $<_n$ Weiermann uses (in a more general context) a recursive function $\Phi: \mathrm{N} \to \mathrm{N}$ which is a variant of the Ackermann function. For our purposes, however, it is sufficient to require the following:

$$\Phi(n) + \Phi(n) + 2 \leq \Phi(n+1) \quad \text{for all } n \in \mathrm{N}. \tag{$*$}$$

For example, take $\Phi(n) := 3^{n+1}$. If Φ satisfies $(*)$, then it also has the following properties:

1. $\Phi(n) < \Phi(n+1)$ and $n \leq \Phi(n)$,
2. $m + \Phi(n) \leq \Phi(m+n)$,
3. $m \cdot n \leq \Phi(m+n)$.

Definition 5.1 (Weiermann [26]).

1. $\alpha <^1_n \beta \;:\Longleftrightarrow\; \alpha < \beta$ and $\mathrm{N}\,\alpha \leq \Phi(\mathrm{N}\,\beta + n)$.
2. Let $<_n$ be the transitive closure of $<^1_n$.

On the interval $[\alpha, \alpha + \omega)$ the relations $<_n$ agree with $<$. If $i < j$ then $\alpha + i <_n \alpha + j$. Note, that $\alpha <_m \beta$ implies $\alpha <_n \beta$ for all n with $m \leq n$. Moreover, the usual operations on ordinals are monotonic with respect to the relations $<_n$.

Lemma 5.1 (Weiermann [26]).

1. $\alpha <_n \beta \implies \omega^\alpha <_n \omega^\beta$,
2. $\alpha <_n \beta \implies \alpha \# \gamma <_n \beta \# \gamma$,
3. $\alpha <_n \gamma$ *and* $\beta <_n \gamma \implies \omega^\alpha \# \omega^\beta <_n \omega^\gamma$,
4. $\alpha <_n \beta \implies \alpha + n + 1 <_0 \beta + n + 1$,
5. $m \cdot n <_n \omega + m$.

Proof. We prove assertions (1)–(5) first for the one step relations $<^1_n$. The versions with $<_n$ follow then immediately, since $<_n$ is the transitive closure of $<^1_n$. For (1), assume $\alpha <^1_n \beta$. By definition, we have $\mathrm{N}\,\alpha \le \Phi(\mathrm{N}\,\beta + n)$. Thus,

$$\mathrm{N}\,\omega^\alpha = 1 + \mathrm{N}\,\alpha \le 1 + \Phi(\mathrm{N}\,\beta + n) \le \Phi(\mathrm{N}\,\beta + n + 1) = \Phi(\mathrm{N}\,\omega^\beta + n).$$

For (2), assume $\alpha <^1_n \beta$. Then we have

$$\mathrm{N}(\alpha\#\gamma) = \mathrm{N}\,\alpha + \mathrm{N}\,\gamma \le \Phi(\mathrm{N}\,\beta + n) + \mathrm{N}\,\gamma \le \Phi(\mathrm{N}\,\beta + n + \mathrm{N}\,\gamma) = \Phi(\mathrm{N}(\beta\#\gamma) + n).$$

For (3), assume $\alpha <^1_n \gamma$ and $\beta <^1_n \gamma$. Then we have

$$\mathrm{N}(\omega^\alpha \# \omega^\beta) \le \Phi(\mathrm{N}\,\gamma + n) + \Phi(\mathrm{N}\,\gamma + n) + 2 \le \Phi(\mathrm{N}\,\gamma + n + 1) = \Phi(\mathrm{N}\,\omega^\gamma + n).$$

For (4), assume $\alpha <^1_n \beta$. By definition, we have $\mathrm{N}\,\alpha \le \Phi(\mathrm{N}\,\beta + n)$. Thus,

$$\mathrm{N}(\alpha + n + 1) \le \Phi(\mathrm{N}\,\beta + n) + n + 1 \le \Phi(\mathrm{N}\,\beta + n + 1) = \Phi(\mathrm{N}(\beta + n + 1)).$$

In (5), we have $\mathrm{N}(m \cdot n) = m \cdot n \le \Phi(m+n) \le \Phi(2+m+n) = \Phi(\mathrm{N}(\omega+m)+n)$. □

Since the set $\{\beta : \beta <^1_n \alpha\}$ is finite for each $\alpha < \varepsilon_0$, one can define the following functions $\theta_\alpha \colon \mathrm{N} \to \mathrm{N}$ for $\alpha < \varepsilon_0$.

Definition 5.2. $\theta_\alpha(n) := \max(\{n+1\} \cup \{\theta_\beta(\theta_\gamma(n)) \mid \beta <^1_n \alpha,\ \gamma <^1_n \alpha\})$.

Similar definitions can be found in [4] and [6].

Lemma 5.2. *The functions θ_α have the following properties:*

1. $n < \theta_\alpha(n)$,
2. $\theta_\alpha(n) < \theta_\alpha(n+1)$,
3. $\alpha <_m \beta \implies \theta_\alpha(m+n) < \theta_\beta(m+n)$,
4. $\alpha <_0 \gamma$ *and* $\beta <_0 \gamma \implies \theta_\alpha(\theta_\beta(n)) \le \theta_\gamma(n)$.

Proof. (1) is trivial. (2) is proved by induction on α. If $\theta_\alpha(n) = n + 1$ then the inequality is obvious. Otherwise there exist $\beta <^1_n \alpha$ and $\gamma <^1_n \alpha$ such that $\theta_\alpha(n) = \theta_\beta(\theta_\gamma(n))$. By the induction hypothesis, we obtain $\theta_\gamma(n) < \theta_\gamma(n+1)$ and $\theta_\beta(\theta_\gamma(n)) < \theta_\beta(\theta_\gamma(n+1))$. Thus $\theta_\alpha(n) < \theta_\alpha(n+1)$, since $\beta <^1_{n+1} \alpha$ and $\gamma <^1_{n+1} \alpha$.

Assertion (3) is first shown for the one step relation $<^1_m$. Assume that $\alpha <^1_m \beta$. By (1), $\theta_\alpha(m+n) < \theta_\alpha(\theta_\alpha(m+n))$ and thus $\theta_\alpha(m+n) < \theta_\beta(m+n)$, since $\alpha <^1_{m+n} \beta$. The generatlization to $<_m$ now follows by transitivity.

In (4), assume that $\alpha <_0 \gamma$ and $\beta <_0 \gamma$. Then there exist α' and β' such that $\alpha \leq_0 \alpha' <^1_0 \gamma$ and $\beta \leq_0 \beta' <^1_0 \gamma$. Since $\alpha' <^1_n \gamma$ and $\beta' <^1_n \gamma$, it follows by (2), (3) and the definition of θ_γ that $\theta_\alpha(\theta_\beta(n)) \leq \theta_{\alpha'}(\theta_{\beta'}(n)) \leq \theta_\gamma(n)$. □

6. The infinitary system $\mathbf{BID}_\infty$

The idea is to replace the induction scheme MIN of BID by the following infinitary rules for $j = 1, \dots, k$. Every rule has countably many hypotheses:

$$\frac{R_j^{(n)}[\mathbf{t}] \to A \quad \text{for all } n \in \mathrm{N}}{R_j(\mathbf{t}) \to A} \qquad (\omega)$$

Similar rules are used by Cantini in [9] in a different context. Note that the nth premise grows exponentially in the size of n. The formula A is arbitrary.

How can we prove the induction scheme MIN with this rule? — Assume that $B_j(\mathbf{x}_j)$ are given formulas, $\mathbf{B} = B_1, \dots, B_k$ and that we have

$$\forall \mathbf{x}_j \left(\mathcal{A}_j[\mathbf{B}, \mathbf{x}_j] \to B_j(\mathbf{x}_j)\right) \quad \text{for } j = 1, \dots, k. \tag{6.1}$$

We have to show

$$\forall \mathbf{x}_j \left(R_j(\mathbf{x}_j) \to B_j(\mathbf{x}_j)\right) \quad \text{for } j = 1, \dots, k. \tag{6.2}$$

In order that we can apply (ω) for proving (6.2) we show, by induction on n, that

$$\forall \mathbf{x}_j \left(R_j^{(n)}[\mathbf{x}_j] \to B_j(\mathbf{x}_j)\right) \quad \text{for } j = 1, \dots, k \tag{6.3}$$

is derivable from (6.1). For $n = 0$ this is trivial, since $R_j^{(0)}[\mathbf{x}_j]$ is the constant $\perp$. Under assumption of (6.3), since $R_j^{(n+1)}[\mathbf{x}_j]$ is the formula $\mathcal{A}_j[R_1^{(n)}, \dots, R_k^{(n)}, \mathbf{x}_j]$ and positive formulas are monotonic, we obtain

$$R_j^{(n+1)}[\mathbf{x}_j] \to \mathcal{A}_j[\mathbf{B}, \mathbf{x}_j] \quad \text{for } j = 1, \dots, k.$$

Now we can apply (6.1) and obtain

$$\forall \mathbf{x}_j \left(R_j^{(n+1)}[\mathbf{x}_j] \to B_j(\mathbf{x}_j)\right) \quad \text{for } j = 1, \dots, k.$$

This is (6.3) for $n + 1$. Hence MIN is proved using rule (ω). We will treat this informal argument below in more detail taking care of the exact length of the derivations.

The infinitary system BID_∞ is formulated in a Tait calculus. Sequents Γ and Δ are finite sets of formulas. As usual Γ, Δ stands for $\Gamma \cup \Delta$ and Γ, A for $\Gamma \cup \{A\}$. Negation is defined by DeMorgan's laws. This requires that we have also complementary relation symbols $\neq, \overline{R}_1, \ldots, \overline{R}_k$. Implication $A \to B$ is defined as $\neg A \vee B$. We define Neg to be the set of formulas of the following form:

$$\top \mid \bot \mid s = t \mid s \neq t \mid \overline{R}_j(\mathbf{t}) \mid A \wedge B \mid A \vee B \mid \forall x\, A \mid \exists x\, A.$$

The length of a formula A is the number of $\wedge$, $\vee$, $\forall$ and $\exists$ occurring in A. It is denoted by $|A|$. The rank of a formula is defined in such a way that that the rank of purely positive or purely negative formulas is zero. The rank is used to measure cut formulas.

1. $\mathrm{rk}(A) := 0$, if $A \in \mathrm{Pos} \cup \mathrm{Neg}$,
2. $\mathrm{rk}(A{*}B) := \max(\mathrm{rk}(A), \mathrm{rk}(B)) + 1$, if $(A{*}B) \notin \mathrm{Pos}\cup\mathrm{Neg}$ and $* \in \{\wedge, \vee\}$,
3. $\mathrm{rk}(Qx\, A) := \mathrm{rk}(A) + 1$, if $A \notin \mathrm{Pos} \cup \mathrm{Neg}$ and $Q \in \{\forall, \exists\}$.

The system BID_∞ is given by the derivation relation $\vdash^{\alpha}_{r} \Gamma$. This relation means that the sequent Γ is derivable with length α and cut rank r. The assignment of ordinals to proofs, however, is with respect to the relations $<_n$ and not with respect to the usual ordering relation $<$ for ordinal numbers. This assignment is due to Weiermann [26] and based on the principle of *local predicativity* of Pohlers [19] and Buchholz [5]. The axiom sequents of the infinitary system are:

1. $\top$
2. $s_1 \neq t_1, \ldots, s_n \neq t_n, a = b$
   if $\sigma = \mathrm{mgu}\{s_1 = t_1, \ldots, s_n = t_n\}$ and $a\sigma \equiv b\sigma$,
3. $s_1 \neq t_1, \ldots, s_n \neq t_n$
   if $\{s_1 = t_1, \ldots, s_n = t_n\}$ is not unifiable,
4. $\overline{R}_j(\mathbf{t})$, $R_j(\mathbf{t})$

The relation $\mathrm{BID}_\infty \vdash^{\alpha}_{r} \Gamma$ (or $\vdash^{\alpha}_{r} \Gamma$ for short) is defined by induction on the ordinal $\alpha < \varepsilon_0$.

Definition 6.1. $\vdash^{\alpha}_{r} \Gamma$, if

(A) $\Delta \subseteq \Gamma$ and Δ is an axiom sequent; or
($\wedge$) $(A \wedge B) \in \Gamma$ and $\vdash^{\alpha_1}_{r} \Gamma, A$ and $\vdash^{\alpha_2}_{r} \Gamma, B$ and $\alpha_1 <_0 \alpha$ and $\alpha_2 <_0 \alpha$; or
($\vee$) $(A \vee B) \in \Gamma$ and $\big(\vdash^{\alpha_1}_{r} \Gamma, A$ and $\alpha_1 <_0 \alpha\big)$ or $\big(\vdash^{\alpha_2}_{r} \Gamma, B$ and $\alpha_2 <_0 \alpha\big)$; or
($\forall$) $\forall x\, A(x) \in \Gamma$ and $\vdash^{\beta}_{r} \Gamma, A(u)$ and $\beta <_0 \alpha$ and $u \notin \mathrm{FV}(\Gamma)$; or
($\exists$) $\exists x\, A(x) \in \Gamma$ and $\vdash^{\beta}_{r} \Gamma, A(t)$ and $\beta <_0 \alpha$; or
(R) $R_j(\mathbf{t}) \in \Gamma$ and $\vdash^{\beta}_{r} \Gamma, A_j[\mathbf{R}, \mathbf{t}]$ and $\beta <_0 \alpha$; or
($\overline{R}$) $\overline{R}_j(\mathbf{t}) \in \Gamma$ and $\forall n \in \mathbb{N}\,\big(\vdash^{\alpha_n}_{r} \Gamma, \neg R^{(n)}_j[\mathbf{t}]$ and $\alpha_n <_n \alpha\big)$; or
(C) $\vdash^{\alpha_1}_{r} \Gamma, A$ and $\vdash^{\alpha_2}_{r} \Gamma, \neg A$ and $\alpha_1 <_0 \alpha$ and $\alpha_2 <_0 \alpha$ and $\mathrm{rk}(A) < r$.

Note, that the ordinal of a premise must be less in the sense of $<_0$ than the the ordinal of the conclusion except in rule ($\overline{R}$). There, the ordinal of the nth premise must be less in the sense of $<_n$.

Lemma 6.1. *1. Substitution: If* $\vdash^{\alpha}_{r} \Gamma(u)$ *then* $\vdash^{\alpha}_{r} \Gamma(t)$.

2. *Weakening: If* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma$ *and* $\alpha \leq_0 \alpha'$ *and* $r \leq r'$ *then* $\mathop{|}\!\frac{\alpha'}{r'}\, \Gamma, \Delta$.
3. *Inversion of* $\wedge$: *If* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma, A \wedge B$ *then* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma, A$ *and* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma, B$.
4. *Inversion of* $\forall$: *If* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma, \forall x\, A(x)$ *then* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma, A(t)$ *for all terms* t.
5. *Inversion of* $\vee$: *If* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma, A \vee B$ *then* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma, A, B$.

Proof. By induction on α. □

Lemma 6.2. *(Reduction) If* $\mathop{|}\!\frac{\alpha}{r}\, \Gamma, A$ *and* $\mathop{|}\!\frac{\beta}{r}\, \Delta, \neg A$ *and* $1 \leq \mathrm{rk}(A) \leq r$ *then* $\mathop{|}\!\frac{\alpha \# \beta}{r}\, \Gamma, \Delta$.

Proof. Since $\alpha \# \beta = \beta \# \alpha$ we can assume that A is either a disjunction or an existentially quantified formula. A cannot be atomic, since atomic formulas are in Pos ∪ Neg and therefore have rank 0. The formula $\neg A$ is then a conjunction or a universally quantified formula and we can apply inversion to it. The lemma is proved by induction on α. □

Lemma 6.3. *(Partial cut-elimination) If* $\mathop{|}\!\frac{\alpha}{r+2}\, \Gamma$ *then* $\mathop{|}\!\frac{\omega^\alpha}{r+1}\, \Gamma$.

Proof. By induction on α. □

For a sequent $\Gamma \subseteq \mathrm{Pos} \cup \mathrm{Neg}$ which consists of positive or negative formulas only and $m, n \in \mathbb{N}$ we define $\Gamma[m,n]$ to be the equality formula

$$\bigwedge_{A \in \Gamma \cap \mathrm{Neg}} \mathrm{E}_m \neg A \rightarrow \bigvee_{A \in \Gamma \cap \mathrm{Pos}} \mathrm{E}_n A.$$

Similar asymmetric interpretations are used in [8] and [12]. Note, that the interpretation has the following monotonicity property. If $m' \leq m$ and $n \leq n'$ then $\Gamma[m,n]$ implies $\Gamma[m',n']$. Thus $\Gamma[m,n]$ is monotonically decreasing in its first argument and monotonically increasing in its second argument.

Theorem 6.1. *If* $\mathop{|}\!\frac{\alpha}{1}\, \Gamma$ *and* $\Gamma \subseteq \mathrm{Pos} \cup \mathrm{Neg}$ *then* $\mathrm{CET} \vdash \Gamma[m, \theta_\alpha(m)]$ *for all* $m \in \mathbb{N}$.

Proof. Case (A). If $\{\overline{R}_j(\mathbf{t}), R_j(\mathbf{t})\} \subseteq \Gamma$ then $\Gamma[m, \theta_\alpha(m)]$ is provable in CET, since m is less than $\theta_\alpha(m)$ and hence $\mathrm{E}_m\, R_j(\mathbf{t})$ implies $\mathrm{E}_{\theta_\alpha(m)}\, R_j(\mathbf{t})$. If $\Delta \subseteq \Gamma$ and Δ is an axiom sequent corresponding to an axiom of CET then $\Delta[m, \theta_\alpha(m)]$ is equivalent to $\bigvee \Delta$ and thus $\Gamma[m, \theta_\alpha(m)]$ is provable in CET.

Case ($\wedge$). Assume that $(A \wedge B) \in \Gamma$, $\alpha_1 <_0 \alpha$, $\alpha_2 <_0 \alpha$,

$$\mathrm{CET} \vdash \Gamma, A[m, \theta_{\alpha_1}(m)] \quad \text{and} \quad \mathrm{CET} \vdash \Gamma, B[m, \theta_{\alpha_2}(m)] \quad \text{for all } m \in \mathbb{N}.$$

Since $\theta_{\alpha_i}(m) \leq \theta_\alpha(m)$ for $i = 1, 2$ we also have

$$\mathrm{CET} \vdash \Gamma, A[m, \theta_\alpha(m)] \quad \text{and} \quad \mathrm{CET} \vdash \Gamma, B[m, \theta_\alpha(m)].$$

If $(A \wedge B) \in \mathrm{Pos}$ then $A \in \mathrm{Pos}$, $B \in \mathrm{Pos}$ and $\mathrm{E}_{\theta_\alpha(m)}(A \wedge B)$ is the formula $\mathrm{E}_{\theta_\alpha(m)}\, A \wedge \mathrm{E}_{\theta_\alpha(m)}\, B$. If $(A \wedge B) \in \mathrm{Neg}$ then $A \in \mathrm{Neg}$, $B \in \mathrm{Neg}$ and

$\mathrm{E}_m(\neg(A \wedge B))$ is $\mathrm{E}_m(\neg A) \vee \mathrm{E}_m(\neg B)$. In both cases we obtain that $\Gamma[m, \theta_\alpha(m)]$ is provable in CET for all $m \in \mathbb{N}$.

Case $(\vee)$. This case goes similar to the previous one.

Case $(\forall)$. In this case one needs that $\mathrm{FV}(\Gamma[m,n]) \subseteq \mathrm{FV}(\Gamma)$.

Case $(\exists)$. In this case one needs that $(\mathrm{E}_m\, A)_u[t]$ is the same as $\mathrm{E}_m(A_u[t])$.

Case (R). Assume that $R_j(\mathbf{t}) \in \Gamma$, $\beta <_0 \alpha$ and

$$\mathrm{CET} \vdash (\Gamma, \mathcal{A}_j[\mathbf{R}, \mathbf{t}])[m, \theta_\beta(m)] \quad \text{for all } m \in \mathbb{N}.$$

The formula $\mathcal{A}_j[\mathbf{R}, \mathbf{t}]$ is positive, and since $\mathrm{E}_{\theta_\beta(m)}\, \mathcal{A}_j[\mathbf{R}, \mathbf{t}]$ is the same as $\mathrm{E}_{\theta_\beta(m)+1}\, R_j(\mathbf{t})$ and $\theta_\beta(m) + 1 \leq \theta_\alpha(m)$, we obtain that $\Gamma[m, \theta_\alpha(m)]$ is provable in CET.

Case $(\overline{R})$. Assume that $\overline{R}_j(\mathbf{t}) \in \Gamma$, $\alpha_n <_n \alpha$ for all $n \in \mathbb{N}$ and

$$\mathrm{CET} \vdash \Gamma, \neg\, \mathrm{E}_n\, R_j(\mathbf{t})[m, \theta_{\alpha_n}(m)] \quad \text{for all } m, n \in \mathbb{N}.$$

In the special case $m = n$ we have

$$\mathrm{CET} \vdash \Gamma, \neg\, \mathrm{E}_m\, R_j(\mathbf{t})[m, \theta_{\alpha_m}(m)].$$

Since $\theta_{\alpha_m}(m) \leq \theta_\alpha(m)$, we obtain that

$$\mathrm{CET} \vdash \Gamma, \neg\, \mathrm{E}_m\, R_j(\mathbf{t})[m, \theta_\alpha(m)]$$

and thus $\mathrm{CET} \vdash \Gamma[m, \theta_\alpha(m)]$.

Case (C). Assume that $\mathrm{rk}(A) < 1$, $\alpha_1 <_0 \alpha$, $\alpha_2 <_0 \alpha$ and

$$\mathrm{CET} \vdash \Gamma, A[m, \theta_{\alpha_1}(m)] \quad \text{and} \quad \mathrm{CET} \vdash \Gamma, \neg A[m, \theta_{\alpha_2}(m)] \quad \text{for all } m \in \mathbb{N}.$$

Since $\mathrm{rk}(A) = 0$, we obtain that $A \in \mathrm{Pos} \cup \mathrm{Neg}$. Without loss of generality we can assume that $A \in \mathrm{Pos}$. Let $m \in \mathbb{N}$. Then we have

$$\mathrm{CET} \vdash \Gamma, A[m, \theta_{\alpha_1}(m)] \quad \text{and} \quad \mathrm{CET} \vdash \Gamma, \neg A[\theta_{\alpha_1}(m), \theta_{\alpha_2}(\theta_{\alpha_1}(m))].$$

From this we obtain $\mathrm{CET} \vdash \Gamma[m, \theta_{\alpha_1}(m)] \vee \Gamma[\theta_{\alpha_1}(m), \theta_{\alpha_2}(\theta_{\alpha_1}(m))]$. Since $m \leq \theta_{\alpha_1}(m)$, $\theta_{\alpha_1}(m) \leq \theta_\alpha(m)$ and $\theta_{\alpha_2}(\theta_{\alpha_1}(m)) \leq \theta_\alpha(m)$, by the monotonicity property of the asymmetric interpretation we obtain that $\Gamma[m, \theta_\alpha(m)]$ is provable in CET. □

In the context of finitary systems Jäger proves in [12] a similar theorem with $\Gamma[m, m+2^\alpha]$. Cantini uses in [8] an asymmetric interpretation $\Gamma[m, F(\mathcal{D}, m)]$ where F is an effective function and $\mathcal{D}$ is the code of an infinitary derivation. Also Pohlers Collapsing Lemma 43 in [19] has to be mentioned here.

Corollary 6.1. *If* $\overset{\alpha}{\underset{1}{\vdash}} A$ *and* $A \in \mathrm{Pos}$ *then* $\mathrm{CET} \vdash \mathrm{E}_{\theta_\alpha(0)}\, A$.

Corollary 6.2. *If* $\overset{\alpha}{\underset{1}{\vdash}} \forall x\, (R(x) \to \exists y\, S(x,y))$ *then*

$$\mathrm{CET} \vdash \forall x\, (\mathrm{E}_m\, R(x) \to \exists y\ \mathrm{E}_{\theta_\alpha(m)}\, S(x,y))$$

for all $m \in \mathbb{N}$.

Proof. Assume that $\vdash_1^\alpha \forall x\,(R(x) \to \exists y\, S(x,y))$. By inversion of $\forall$ and $\vee$, we obtain that $\vdash_1^\alpha \overline{R}(u), \exists y\, S(u,y)$. Since $\overline{R}(u) \in$ Neg and $\exists y\, S(u,y) \in$ Pos we can apply the asymmetric interpretation and obtain that the formula

$$\mathrm{E}_m\, R(u) \to \mathrm{E}_{\theta_\alpha(m)}\, \exists y\, S(u,y)$$

is provable in CET. □

What remains to show is that BID can be embedded into the infinitary system BID_∞. In a first step we show that every instance of MIN is cut-free provable in BID_∞ with length $\omega + m$ for some $m \in \mathrm{N}$ (cf. Generalized Induction Theorem 28.10 in [18]).

Lemma 6.4. *If A is an instance of MIN then there exists an $m \in \mathrm{N}$ such that $\vdash_0^{\omega+m} A$.*

Proof. Let $B_j(\mathbf{x}_j)$ be formulas for $j = 1,\dots,k$ and $\mathbf{B} = B_1,\dots,B_k$. Let $cl(\mathbf{B})$ be the formula

$$\bigwedge_{j=1}^{k} \forall \mathbf{x}_j\, (\mathcal{A}_j[\mathbf{B},\mathbf{x}_j] \to B_j(\mathbf{x}_j)).$$

We claim that there exists a constant $c \in \mathrm{N}$ such that for all $n \in \mathrm{N}$ and all terms $\mathbf{t}$

$$\vdash_0^{c\cdot n} \neg cl(\mathbf{B}), \neg R_j^{(n)}[\mathbf{t}], B_j(\mathbf{t}) \quad \text{for } j = 1,\dots,k. \tag{6.4}$$

Let $c := 2c_1 + 2c_2 + c_3 + k + 1$, where

$$c_1 := \max_{1\le j\le k} |\mathcal{A}_j|, \quad c_2 := \max_{1\le j\le k} |B_j|, \quad c_3 := \max_{1\le j\le k} arity(R_j).$$

We show (6.4) by induction on n. If $n = 0$ then (6.4) holds, since $\neg R_j^{(0)}[\mathbf{t}]$ is the constant $\top$. In the induction step we obtain from (6.4), since $\mathcal{A}_j[\mathbf{R},\mathbf{t}]$ is positive in $\mathbf{R}$,

$$\vdash_0^{c\cdot n+2c_1} \neg cl(\mathbf{B}), \neg\mathcal{A}_j[R_1^{(n)},\dots,R_k^{(n)},\mathbf{t}], \mathcal{A}_j[B_1,\dots,B_k,\mathbf{t}].$$

Since $\vdash_0^{2c_2} \neg B_j(\mathbf{t}), B_j(\mathbf{t})$, we obtain

$$\vdash_0^{c\cdot n+2c_1+2c_2+1} \neg cl(\mathbf{B}), \neg R_j^{(n+1)}[\mathbf{t}], B_j(\mathbf{t}), \mathcal{A}_j[\mathbf{B},\mathbf{t}] \wedge \neg B_j(\mathbf{t}).$$

After several applications of ($\exists$) we obtain

$$\vdash_0^{c\cdot n+2c_1+2c_2+c_3+1} \neg cl(\mathbf{B}), \neg R_j^{(n+1)}[\mathbf{t}], B_j(\mathbf{t}), \exists \mathbf{x}_j\, (\mathcal{A}_j[\mathbf{B},\mathbf{x}_j] \wedge \neg B_j(\mathbf{x}_j))$$

and finally after several applications of ($\vee$)

$$\vdash_0^{c\cdot n+2c_1+2c_2+c_3+k+1} \neg cl(\mathbf{B}), \neg R_j^{(n+1)}[\mathbf{t}], B_j(\mathbf{t}).$$

Thus we have (6.4) for $n + 1$. Since $c \cdot n <_n \omega + c$, we can apply ($\overline{R}$) and obtain

$$\vdash_0^{\omega+c} \neg cl(\mathbf{B}), \neg R_j(\mathbf{u}), B_j(\mathbf{u}).$$

Applications of ($\vee$), ($\forall$) and ($\wedge$) give the desired result. □

In a second step we show that if a formula is provable in BID then it is provable in the infinitary system BID_∞ with length $\omega + m$ and cut-rank r for appropriate $m, r \in \mathbb{N}$.

Lemma 6.5. *If* $\mathrm{BID} \vdash A$ *then there exist* $m, r \in \mathbb{N}$ *such that* $\mathrm{BID}_\infty \overset{\omega+m}{\underset{r}{\vdash}} A$.

Proof. By induction on the length of the proof. Remember that on the interval $[\omega, \omega + \omega)$ the relation $<$ is the same as $<_0$. □

Putting everything together we obtain the following two main theorems.

Theorem 6.2. *If* $\mathrm{BID} \vdash A$ *and* A *is positive, then there exists an* $\alpha < \varepsilon_0$ *such that* $\mathrm{CET} \vdash \mathrm{E}_{\theta_\alpha(0)} A$ *and hence* $\mathfrak{J}^{\theta_\alpha(0)} \models \forall(A)$.

This theorem is used in the completeness proof of LDNF-resolution (cf. [22, 23]). The bound $\theta_\alpha(0)$ is not that much important. What is important is that if a positive formula is provable in BID then it is already true at a finite stage. However, the bounds are important for the proof-theoretic strength of BID namely for characterizing the provable total functions of BID.

Theorem 6.3. *If* $\mathrm{BID} \vdash \forall x\,(nat(x) \rightarrow \exists y\,(nat(y) \wedge R(x,y)))$ *then there exists an* $\alpha < \varepsilon_0$ *such that*

$$\mathrm{CET} \vdash \forall x\,(\mathrm{E}_m\, nat(x) \rightarrow \exists y\,(\mathrm{E}_{\theta_\alpha(m)}\, nat(y) \wedge \mathrm{E}_{\theta_\alpha(m)}\, R(x,y)))$$

for all $m \in \mathbb{N}$.

The second theorem can be expressed in a more general form as well. If A and B are positive formulas such that $A \rightarrow B$ is provable in BID then there exists an $\alpha < \varepsilon_0$ such that $\mathrm{E}_m A \rightarrow \mathrm{E}_{\theta_\alpha(m)} B$ is provable in CET for all $m \in \mathbb{N}$. Since $\mathfrak{U}_\mathcal{L}$ is a model of CET, it follows that if A is true a stage $\mathfrak{J}^m$ then B is true a stage $\mathfrak{J}^{\theta_\alpha(m)}$.

7. Negation-as-failure and left-termination

In this section we show how negation-as-failure and left-termination can be expressed by inductive definitions. What does this mean? — Consider a predicate R in a logic program P. From the clauses that define R we construct operator forms for three new relation symbols R^{s} (success), R^{f} (finite failure) and R^{t} (left-termination). Even though in most cases the operator forms for R^{s} are purely existential formulas, the operator forms for R^{f} and R^{t} are more complicated. They are universal formulas in general.

It can be shown that $R^{\mathrm{t}}(a)$ is derivable in the corresponding system BID if, and only if, the goal $R(a)$ is left-terminating in the sense of Apt and Pedreschi [1]. Moreover, the formula $R^{\mathrm{s}}(a) \wedge R^{\mathrm{t}}(a)$ is derivable in BID if, and only if, the goal $R(a)$ succeeds and is left-terminating; $R^{\mathrm{f}}(a) \wedge R^{\mathrm{t}}(a)$

is derivable in BID if, and only if, the goal $R(a)$ fails finitely and is left-terminating. These results are proved in [23]. What remains to show in this paper is that the system used in [23] can be embedded into BID. In order to do this we need some terminology of logic programming. For a more detailed presentation we refer the reader to [23].

The syntactic objects in logic programming that correspond to formulas are called *goals*. We use different connectives for negation, conjunction and disjunction of goals to point out that they have a different meaning in logic programming. Negation means negation-as-failure; conjunction is a non-commutative operation; disjunction means alternatives in the sense of Prolog. *Goals* (denoted by G, H) are expressions of the following form:

$$\mathit{true} \mid \mathit{fail} \mid s = t \mid R(\mathbf{t}) \mid \mathit{not}\, G \mid G \,\&\, H \mid G \;\mathit{or}\; H \mid \exists x\, G.$$

A goal of the form $G_1 \,\&\, \ldots \,\&\, G_n \,\&\, \mathit{true}$ is called a *query* and we abbreviate it by $[G_1, \ldots, G_n]$. A *program clause* is an expression of the form $R(\mathbf{t}) \leftarrow G$, where G is a goal. $R(\mathbf{t})$ is called the *head* of the clause and G its *body*. For a clause C of the form

$$R(t_1[\mathbf{y}], \ldots, t_n[\mathbf{y}]) \leftarrow G[\mathbf{y}]$$

we define its *definition form* $D_C[\mathbf{x}]$ to be the goal

$$D_C[x_1, \ldots, x_n] :\equiv \exists \mathbf{y}(x_1 = t_1[\mathbf{y}] \,\&\, \ldots \,\&\, x_n = t_n[\mathbf{y}] \,\&\, G[\mathbf{y}]),$$

where $\mathbf{y}$ are new variables. A *logic program* P is a finite list of program clauses. For every relation symbol R and logic program P we define a goal $D_R^P[\mathbf{x}]$ called the *definition form* of R with respect to P.

Let $C_1, \ldots, C_m$ be the list of program clauses in P the head of which is of the form $R(\ldots)$. Then we set

$$D_R^P[\mathbf{x}] :\equiv D_{C_1}[\mathbf{x}] \;\mathit{or}\; \ldots \;\mathit{or}\; D_{C_m}[\mathbf{x}].$$

One could as well define a logic program to be a function that assigns to every relation symbol R a goal $D_R^P[\mathbf{x}]$ with distinguished variables $\mathbf{x}$.

Given a logic program P the user can ask queries. In the following we describe a simple query evaluation procedure which directly reflects the stack based memory management of most implementations of logic programming systems.

An *environment* is a finite set of bindings $\{t_1/x_1, \ldots, t_n/x_n\}$ such that the x_i's are pairwise different variables. It is not required that $t_i \not\equiv x_i$. A *frame* consists of a query G and an idempotent environment η. Idempotent means that if $t_i \not\equiv x_i$ then x_i does not occur in $t_1, \ldots, t_n$. Remember that a query is a list of goals. A *frame stack* consists of a (possibly empty) sequence $\langle G_1, \eta_1; \ldots; G_n, \eta_n\rangle$ of frames. The query G_n together with the environment η_n is called the *topmost frame* of the stack. Capital greek letters Φ, Ψ and Θ denote finite, possibly empty, sequences of the form $G_1, \eta_1; \ldots; G_n, \eta_n$. Thus $\langle \Phi; G, \eta\rangle$ denotes a stack with topmost frame G, η. A *state* of a computation is a finite sequence $\langle \Phi_1\rangle \ldots \langle \Phi_n\rangle$ of frame stacks. $\langle \Phi_n\rangle$ is called the topmost stack of the state. States are denoted by the capital greek letter Σ. For a query G with free variables $x_1, \ldots, x_n$ let *init*(G) be the state $\langle G, \{x_1/x_1, \ldots, x_n/x_n\}\rangle$. There are three kinds of final states: *yes*(η), *no* and *error*.

Definition 7.1. The transition rules of the query evaluation procedure are:

1. $\Sigma\ \langle \Phi; \mathit{true}\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi; G, \eta\rangle$
2. $\Sigma\ \langle \Phi; \mathit{fail}\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi\rangle$
3. $\Sigma\ \langle \Phi; s = t\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi; G, \eta\tau\rangle$ [if $\tau = \mathrm{mgu}(s\eta, t\eta)$]
4. $\Sigma\ \langle \Phi; s = t\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi\rangle$ [if $s\eta$ and $t\eta$ are not unifiable]
5. $\Sigma\ \langle \Phi; R(\mathrm{t})\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi; D_R^P[\mathrm{t}]\ \&\ G, \eta\rangle$
6. $\Sigma\ \langle \Phi; (E\ \&\ F)\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi; E\ \&\ (F\ \&\ G), \eta\rangle$
7. $\Sigma\ \langle \Phi; (E\ \mathit{or}\ F)\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi; F\ \&\ G, \eta; E\ \&\ G, \eta\rangle$
8. $\Sigma\ \langle \Phi; (E\ \mathit{or}\ F)\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi; E\ \&\ G, \eta; F\ \&\ G, \eta\rangle$
9. $\Sigma\ \langle \Phi; (\exists x\, F)\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi; F\{y/x\}\ \&\ G, \eta \cup \{y/y\}\rangle$ [where y is new]
10. $\Sigma\ \langle \Phi; (\mathit{not}\, F)\ \&\ G, \eta\rangle \longrightarrow \Sigma\ \langle \Phi; (\mathit{not}\, F)\ \&\ G, \eta\rangle\ \langle [F], \eta\rangle$ [if $F\eta$ is ground]
11. $\Sigma\ \langle \Phi; (\mathit{not}\, F)\ \&\ G, \eta\rangle \longrightarrow \mathit{error}$ [if $F\eta$ is not ground]
12. $\Sigma\ \langle \Phi; (\mathit{not}\, F)\ \&\ G, \eta\rangle\ \langle \Psi; \mathit{true}, \tau\rangle \longrightarrow \Sigma\ \langle \Phi\rangle$
13. $\Sigma\ \langle \Phi; (\mathit{not}\, F)\ \&\ G, \eta\rangle\ \langle\rangle \longrightarrow \Sigma\ \langle \Phi; G, \eta\rangle$
14. $\langle \Phi; \mathit{true}, \eta\rangle \longrightarrow \mathit{yes}(\eta)$
15. $\langle\rangle \longrightarrow \mathit{no}$

Without Rule (8) this definition describes the operational model of pure Prolog with occurs check. We say that

1. a query G *succeeds with answer* σ, if there exists a computation with initial state *init*(G) and final state *yes*(η) such that σ is the restriction of η to the variables of G;
2. a query G *succeeds with answer including* σ, if there exist substitutions τ and θ such that G succeeds with answer τ and $G\tau\theta \equiv G\sigma$;
3. a query G *fails*, if there exists a computation with initial state *init*(G) and final state *no*;
4. a query G is *left-terminating*, if all computations with initial state *init*(G) are finite and do not end in *error*.

Note, that termination means universal termination. For Prolog-like systems this means that one can hit the semicolon key a finite number of times until one finally obtains the message *no more solutions*.

For defining operator forms for success, finite failure and left-termination we need a language with new predicate symbols R^{s}, R^{f} and R^{t}. We also need a special unary predicate *gr* which is used to express that a term is ground. The formulas A, B of the extended language are

$$\top \mid \bot \mid s = t \mid gr(t) \mid R^{\mathrm{s}}(\mathrm{t}) \mid R^{\mathrm{f}}(\mathrm{t}) \mid R^{\mathrm{t}}(\mathrm{t}) \mid$$

$$\neg A \mid A \wedge B \mid A \vee B \mid A \to B \mid \forall x\, A \mid \exists x\, A.$$

Three syntactic operators S, F and T are defined that transform goals into formulas of the extended language. What is important in the definition is that for every goal G the formulas $\mathsf{S}\,G$, $\mathsf{F}\,G$ and $\mathsf{T}\,G$ are positive.

$$\begin{array}{ll}
\mathsf{S}\,\mathit{true} :\equiv \top, & \mathsf{F}\,\mathit{true} :\equiv \bot, \\
\mathsf{S}\,\mathit{fail} :\equiv \bot, & \mathsf{F}\,\mathit{fail} :\equiv \top, \\
\mathsf{S}\,s = t :\equiv s = t, & \mathsf{F}\,s = t :\equiv s \neq t, \\
\mathsf{S}\,R(\mathbf{t}) :\equiv R^{\mathrm{s}}(\mathbf{t}), & \mathsf{F}\,R(\mathbf{t}) :\equiv R^{\mathrm{f}}(\mathbf{t}), \\
\mathsf{S}\,\mathit{not}\,G :\equiv \mathsf{F}\,G, & \mathsf{F}\,\mathit{not}\,G :\equiv \mathsf{S}\,G, \\
\mathsf{S}(G\,\&\,H) :\equiv \mathsf{S}\,G \wedge \mathsf{S}\,H, & \mathsf{F}(G\,\&\,H) :\equiv \mathsf{F}\,G \vee \mathsf{F}\,H, \\
\mathsf{S}(G\ \mathit{or}\ H) :\equiv \mathsf{S}\,G \vee \mathsf{S}\,H, & \mathsf{F}(G\ \mathit{or}\ H) :\equiv \mathsf{F}\,G \wedge \mathsf{F}\,H, \\
\mathsf{S}\,\exists x\,G :\equiv \exists x\,\mathsf{S}\,G, & \mathsf{F}\,\exists x\,G :\equiv \forall x\,\mathsf{F}\,G, \\
\\
\mathsf{T}\,\mathit{true} :\equiv \top, & \mathsf{T}\,\mathit{not}\,G :\equiv \mathsf{T}\,G \wedge gr(G), \\
\mathsf{T}\,\mathit{fail} :\equiv \top, & \mathsf{T}(G\,\&\,H) :\equiv \mathsf{T}\,G \wedge (\mathsf{F}\,G \vee \mathsf{T}\,H), \\
\mathsf{T}\,s = t :\equiv \top, & \mathsf{T}(G\ \mathit{or}\ H) :\equiv \mathsf{T}\,G \wedge \mathsf{T}\,H, \\
\mathsf{T}\,R(\mathbf{t}) :\equiv R^{\mathrm{t}}(\mathbf{t}), & \mathsf{T}\,\exists x\,G :\equiv \forall x\,\mathsf{T}\,G.
\end{array}$$

In the definition of $\mathsf{T}\,\mathit{not}\,G$ the expression $gr(G)$ is an abbreviation for $gr(x_1) \wedge \ldots \wedge gr(x_n)$, where $\mathrm{FV}(G) = \{x_1, \ldots, x_n\}$. In the definition of $\mathsf{T}(G\ \&\ H)$ it becomes clear the the operator T expresses left-termination. The definition of $\mathsf{T}(G\ \mathit{or}\ H)$ shows that T means universal termination. Note, that by defining $\forall x\,G$ as $\mathit{not}\,\exists x\,\mathit{not}\,G$ we can construct goals G such that $\mathsf{S}\,G$ is any positive first-order formula.

We consider $\mathsf{S}\,D^P_R[\mathbf{x}]$, $\mathsf{F}\,D^P_R[\mathbf{x}]$ and $\mathsf{T}\,D^P_R[\mathbf{x}]$ as operator forms defining the relations R^{s}, R^{f} and R^{t} and define $\mathrm{IND}(P)$ to be the theory

$$\mathrm{IND}(P) := \mathrm{BID}(R^{\mathrm{s}}, \mathsf{S}\,D^P_R[\mathbf{x}]; R^{\mathrm{f}}, \mathsf{F}\,D^P_R[\mathbf{x}]; R^{\mathrm{t}}, \mathsf{T}\,D^P_R[\mathbf{x}]; \ldots).$$

Note that $\mathsf{S}\,D^P_R[\mathbf{x}]$ and $\mathsf{F}\,D^P_R[\mathbf{x}]$ contain the relation symbols R^{s} and R^{f} but not the relations R^{t}. The operator form $\mathsf{T}\,D^P_R[\mathbf{x}]$, however, contains all three kind of relations R^{s}, R^{f} and R^{t}. Without the relations symbols R^{t} and the operator T, if we just consider $\mathsf{S}\,D^P_R[\mathbf{x}]$ and $\mathsf{F}\,D^P_R[\mathbf{x}]$ as inductive definitions over the Herbrand universe for the relations R^{s} and R^{f}, then we obtain the tree-valued semantics of Fitting [11] mentioned in Sect. 1.

The theory $\mathrm{IND}(P)$ is called the *inductive extension* of P. Of course, we need also axioms for the predicate gr. The axioms are:

1. $gr(c)$, [if $c \in \mathfrak{L}$ is a constant symbol]
2. $gr(f(x_1, \ldots, x_n)) \leftrightarrow gr(x_1) \wedge \ldots \wedge gr(x_n)$. [if $f \in \mathfrak{L}$ is n-ary]

These axioms can be added to $\mathrm{CET}_{\mathfrak{L}}$ and the results of the previous sections remain valid. Note, that it is not possible to prove $\forall x\,gr(x)$, since there is no complete induction on the universe in $\mathrm{IND}(P)$.

What we call inductive extension in [23] differs from the definition here. The additional axioms of [23], however, are derivable in $\mathrm{IND}(P)$ as is stated in the following two lemmata. The first lemma is the fomal statement that it is not possible for a goal to both, succeed and fail.

Lemma 7.1. $\mathrm{IND}(P) \vdash \neg(\mathsf{S}\,G \wedge \mathsf{F}\,G)$ *for arbitrary goals* G.

Proof. Let $(A)^+$ be the result of replacing all subformulas of the form $R^{\mathrm{s}}(\mathbf{a})$ in A by $\neg R^{\mathrm{f}}(\mathbf{a})$ and all subformulas of the form $R^{\mathrm{f}}(\mathbf{a})$ by $\neg R^{\mathrm{s}}(\mathbf{a})$ for each predicate R. It is easy to see, by induction on the length of the goal G, that the implications

$$(\mathsf{S}\,G)^+ \rightarrow \neg\,\mathsf{F}\,G \quad \text{and} \quad (\mathsf{F}\,G)^+ \rightarrow \neg\,\mathsf{S}\,G$$

are provable in pure logic. As a special case we thus have

$$(\mathsf{S}\,D_R^P[\mathbf{x}])^+ \rightarrow \neg\,\mathsf{F}\,D_R^P[\mathbf{x}] \quad \text{and} \quad (\mathsf{F}\,D_R^P[\mathbf{x}])^+ \rightarrow \neg\,\mathsf{S}\,D_R^P[\mathbf{x}].$$

By the contraposition of FIX (Sect. 4), we obtain that

$$(\mathsf{S}\,D_R^P[\mathbf{x}])^+ \rightarrow \neg R^{\mathrm{f}}(\mathbf{x}) \quad \text{and} \quad (\mathsf{F}\,D_R^P[\mathbf{x}])^+ \rightarrow \neg R^{\mathrm{s}}(\mathbf{x})$$

are provable in $\mathrm{IND}(P)$. Since $\neg R^{\mathrm{f}}(\mathbf{x})$ is the same as $(R^{\mathrm{s}}(\mathbf{x}))^+$ and $\neg R^{\mathrm{s}}(\mathbf{x})$ is the same as $(R^{\mathrm{f}}(\mathbf{x}))^+$ we can apply MIN and obtain that

$$R^{\mathrm{s}}(\mathbf{x}) \rightarrow \neg R^{\mathrm{f}}(\mathbf{x}) \quad \text{and} \quad R^{\mathrm{f}}(\mathbf{x}) \rightarrow \neg R^{\mathrm{s}}(\mathbf{x})$$

are provable in $\mathrm{IND}(P)$. From this we obtain that $\neg(\mathsf{S}\,G \wedge \mathsf{F}\,G)$ is provable for arbitrary goals G. □

The second lemma says that a goal succeeds or fails provided that it terminates. Note, that without the condition of termination this statement is not true in general, since a goal may loop.

Lemma 7.2. $\mathrm{IND}(P) \vdash \mathsf{T}\,G \rightarrow (\mathsf{S}\,G \vee \mathsf{F}\,G)$ *for arbitrary goals* G.

Proof. Let $(A)^*$ be the result of replacing all subformulas of the form $R^{\mathrm{t}}(\mathbf{a})$ in A by $R^{\mathrm{s}}(\mathbf{a}) \vee R^{\mathrm{f}}(\mathbf{a})$ for each predicate R. It is easy to see, by induction on the length of the goal G, that

$$(\mathsf{T}\,G)^* \rightarrow \mathsf{S}\,G \vee \mathsf{F}\,G$$

is provable in pure logic. As a special case we thus have

$$(\mathsf{T}\,D_R^P[\mathbf{x}])^* \rightarrow \mathsf{S}\,D_R^P[\mathbf{x}] \vee \mathsf{F}\,D_R^P[\mathbf{x}].$$

By CLS we obtain that

$$(\mathsf{T}\,D_R^P[\mathbf{x}])^* \rightarrow R^{\mathrm{s}}(\mathbf{x}) \vee R^{\mathrm{f}}(\mathbf{x})$$

is provable in $\mathrm{IND}(P)$. Since $R^{\mathrm{s}}(\mathbf{x}) \vee R^{\mathrm{f}}(\mathbf{x})$ is the same as $(R^{\mathrm{t}}(\mathbf{x}))^*$, we can apply MIN and obtain that

$$R^{\mathrm{t}}(\mathbf{x}) \rightarrow R^{\mathrm{s}}(\mathbf{x}) \vee R^{\mathrm{f}}(\mathbf{x})$$

is provable in $\mathrm{IND}(P)$ for each predicate R. From this we can derive the formula $\mathsf{T}\,G \rightarrow \mathsf{S}\,G \vee \mathsf{F}\,G$ for arbitrary goals G. □

An interesting consequence of the previous two lemmata is that IND(P) proves the following equivalence:

$$\mathbf{T}\,G \wedge (\mathbf{F}\,G \vee \mathbf{T}\,H) \leftrightarrow \mathbf{T}\,G \wedge (\mathbf{S}\,G \rightarrow \mathbf{T}\,H). \tag{7.1}$$

For, assume that we have $\mathbf{T}\,G$. Then the previous two lemmata yield that $\mathbf{S}\,G$ is equivalent to $\neg\,\mathbf{F}\,G$. Thus $\mathbf{F}\,G \vee \mathbf{T}\,H$ is equivalent to $\mathbf{S}\,G \rightarrow \mathbf{T}\,H$. Note that on the left-hand side of the equivalence (7.1) there is a positive formula but the formula on the right-hand side can contain negative occurences of predicate symbols. This means that the termination of a conjunctive goal, $\mathbf{T}(G \mathbin{\&} H)$, can be expressed in a non-positive way.

A second consequense of the two lemmata is that the system of [23] can be embedded into IND(P) and the following main theorem of [23] carries over to IND(P).

Theorem 7.1. *Let P be a logic program.*

1. IND(P) $\vdash \mathbf{T}\,G$ *if, and only if, G is left-terminating.*
2. *If G succeeds with answer σ, then* IND(P) $\vdash \mathbf{S}\,G\sigma$.
3. *If G fails, then* IND(P) $\vdash \mathbf{F}\,G$.
4. *If* IND(P) $\vdash \mathbf{T}\,G \wedge \mathbf{S}\,G\sigma$, *then G succeeds with answer including σ.*
5. *If* IND(P) $\vdash \mathbf{T}\,G \wedge \mathbf{F}\,G$, *then G fails.*

We see that the framework for inductive definitions over the Herbrand universe developed in this paper can be used as a foundation of logic programming.

As a last remark we want to point out that it is possible to obtain bounds on the length of computations from the result of this paper. Assume that P is a program and A is a ground atom such that $\mathbf{T}\,A$ is provable with length n in IND(P). Depending on n, we obtain an ordinal $\alpha < \varepsilon_0$ such that $\mathbf{T}\,A$ is true at stage $\mathfrak{J}^{\theta_\alpha(0)}$. From the results in [23] it now follows that the number of states in the computation tree for A is bounded by $r^{\theta_\alpha(0)}$, where r is a bound on the size of the definition forms for the predicates in P.

We do not claim that these bounds are best possible, however, the reader should be aware that, for example, in the following program

$$\begin{array}{l} \mathit{test}(0) \\ \mathit{test}(s(x)) \leftarrow \mathit{test}(x) \mathbin{\&} \mathit{test}(x) \end{array}$$

the computation tree for $\mathit{test}(\overline{n})$ has size 2^n whereas $\mathbf{T}\,\mathit{test}(\overline{n})$ can be proved with length $O(n)$ in the inductive extension of the program.

Acknowledgment

I am grateful to Thomas Strahm and Andreas Weiermann for helpful comments.

References

1. K. R. Apt and D. Pedreschi. Reasoning about termination of pure Prolog programs. *Information and Computation*, 106(1):109–157, 1993.
2. H. A. Blair. The recursion-theoretic complexity of the semantics of predicate logic as a programming language. *Information and Control*, 54:25–47, 1982.
3. H. A. Blair and A. L. Brown. Definite clause programs are canonical (over a suitable domain). *Annals of Mathematics and Artificial Intelligence*, 1:1–19, 1990.
4. B. Blankertz and A. Weiermann. How to characterize provably total functions by the Buchholz operator method. This volume.
5. W. Buchholz. A simplified version of local predicativity. In P. Aczel, H. Simmons, and S. S. Wainer, editors, *Proof Theory. A selection of papers from the Leeds Proof Theory Programme 1990*, pages 115–147. Cambridge University Press, 1992.
6. W. Buchholz, A. Cichon, and A. Weiermann. A uniform approach to fundamental sequences and hierarchies. *Mathematical Logic Quarterly*, 40:273–286, 1994.
7. W. Buchholz, S. Feferman, W. Pohlers, and W. Sieg. *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, volume 897 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1981.
8. A. Cantini. Levels of implication and type free theories of classifications with approximation operator. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 38:107–141, 1992.
9. A. Cantini. Proof-theoretic aspects of self-referential truth. Technical report, Department of Philosophy, Università degli Studi di Firenze, 1995.
10. K. L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.
11. M. Fitting. A Kripke-Kleene semantics for logic programs. *J. of Logic Programming*, 2:295–312, 1985.
12. G. Jäger. Fixed points in Peano arithmetic with ordinals. *Annals of Pure and Applied Logic*, 60:119–132, 1993.
13. G. Jäger and R. F. Stärk. A proof-theoretic framework for logic programming. In S. R. Buss, editor, *Handbook of Proof Theory*. To appear.
14. K. Kunen. Negation in logic programming. *J. of Logic Programming*, 4(4):289–308, 1987.
15. K. Kunen. Signed data dependencies in logic programs. *J. of Logic Programming*, 7(3):231–245, 1989.
16. A. I. Mal'cev. Axiomatizable classes of locally free algebras of various types. In *The Metamathematics of Algebraic Systems, Collected Papers*, chapter 23, pages 262–281. North-Holland, Amsterdam, 1971.
17. Y. N. Moschovakis. *Elementary Induction on Abstract Structures*. North-Holland, Amsterdam, 1974.
18. W. Pohlers. *Proof theory: an introduction*, volume 1407 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1989.
19. W. Pohlers. A short course in ordinal analysis. In P. Aczel, H. Simmons, and S. S. Wainer, editors, *Proof Theory. A selection of papers from the Leeds Proof Theory Programme 1990*, pages 26–78. Cambridge University Press, 1992.
20. J. C. Shepherdson. Language and equality theory in logic programming. Technical Report PM-88-08, University of Bristol, 1988.
21. R. F. Stärk. Input/output dependencies of normal logic programs. *J. of Logic and Computation*, 4(3):249–262, 1994.

22. R. F. Stärk. First-order theories for pure Prolog programs with negation. *Archive for Mathematical Logic*, 34(2):113–144, 1995.
23. R. F. Stärk. Formal methods for logic programming systems. Technical report, Department of Mathematics, Stanford University, 1995.
24. R. F. Stärk. Total correctness of pure Prolog programs: A formal approach. In R. Dyckhoff, H. Herre, and P. Schroeder-Heister, editors, *Proceedings of the 5th International Workshop on Extensions of Logic Programming, ELP '96*, pages 237–254, Leipzig, Germany, 1996. Springer-Verlag, Lecture Notes in Artificial Intelligence 1050.
25. M. H. van Emden and R. A. Kowalski. The semantics of predicate logic as a programming language. *J. of the Association for Computing Machinery*, 4(23):733–742, 1976.
26. A. Weiermann. How to characterize provably total functions by local predicativity. *J. of Symbolic Logic*. To appear.