CONSISTENCY PROOF OF A FRAGMENT OF PV WITH SUBSTITUTION IN BOUNDED ARITHMETIC

YORIYUKI YAMAGATA

Abstract. This paper presents proof that Buss's S^2_2 can prove the consistency of a fragment of Cook and Urquhart's PV from which induction has been removed but substitution has been retained. This result improves Beckmann's result, which proves the consistency of such a system without substitution in bounded arithmetic S^1_2.

Our proof relies on the notion of "computation" of the terms of PV. In our work, we first prove that, in the system under consideration, if an equation is proved and either its left- or right-hand side is computed, then there is a corresponding computation for its right- or left-hand side, respectively. By carefully computing the bound of the size of the computation, the proof of this theorem inside a bounded arithmetic is obtained, from which the consistency of the system is readily proven.

This result apparently implies the separation of bounded arithmetic because Buss and Ignjatović stated that it is not possible to prove the consistency of a fragment of PV without induction but with substitution in Buss's S^1_2. However, their proof actually shows that it is not possible to prove the consistency of the system, which is obtained by the addition of propositional logic and other axioms to a system such as ours. On the other hand, the system that we have considered is strictly equational, which is a property on which our proof relies.

1. Introduction

Ever since Buss showed the relation between his hierarchy of bounded arithmetic, $S^i_2, i = 1, 2, \ldots$, and the polynomial time hierarchy of computational complexity [3], the question of whether his hierarchy collapses at some $i = n$ has become a central question in bounded arithmetic. This is because the collapse of Buss's hierarchy implies the collapse of polynomial time hierarchy.

A classical way to prove the separation of theories is to use the second incompleteness theorem of Gödel. For example, if it is proved that S_2 proves the consistency of S^1_2, $S^1_2 \neq S_2$ is obtained, because S^1_2 cannot prove its own consistency.

Wilkie and Paris showed that S_2 cannot prove the consistency of Robinson arithmetic Q [11], which is a much weaker system. Although this result stems more from the free use of unbounded quantifiers than from the power of arithmetic, Pudlák showed that S_2 cannot prove the consistency of bounded proofs (proofs in which the formulas only have bounded quantifiers) of S^1_2 [9]. The result was refined by Takeuti [10], as well as by Buss and Ignjatović [4], who showed that, even if induction is removed from S^1_2, S_2 is still not able to prove the consistency of its bounded proofs.

2000 *Mathematics Subject Classification.* 03F30, 03D15.

Key words and phrases. bounded arithmetic, consistency proof, computational complexity.

Thus, it will be interesting to delineate theories that can be proven to be consistent in S_2 and S^1_2 in order to find a theory T that can be proven to be consistent in S_2 but not in S^1_2. In particular, we focus on Cook and Urquhart's system PV [6], which is essentially an equational version of S^1_2. Buss and Ignjatović stated that PV cannot prove the consistency of PV$^-$, a system based on PV from which induction has been removed but substitution is retained. On the other hand, Beckmann [1] later proved that S^1_2 can prove the consistency of a theory obtained from PV$^-$ by removing the substitution rule.

This paper presents proof that S^2_2 is capable of proving the consistency of purely equational PV$^-$, in which proofs are formulated as trees. This result apparently implies that $S^1_2 \subsetneq S^2_2$ is based on the result of Buss and Ignjatović. However, their proof actually shows that PV cannot prove the consistency of the extension of PV$^-$ that contains propositional logic and $BASIC^e$ axioms. On the other hand, our PV$^-$ is strictly equational, which is a property on which our proof relies. Although Buss and Ignjatović stated that their proof can be extended to purely equational PV$^-$, there is a gap in their reasoning. We discuss this in Section 8.4 in detail.

The consistency of PV$^-$ can be proven by using the following strategy. Beckmann used a rewriting system to prove the consistency of PV$^-$ by excluding the substitution rule. According to the terminology of programming language theory, the use of a rewriting system to define the evaluation of terms is known as *small-step semantics* (referred to as *structural operational semantics* in [8]).

There is an alternative approach toward obtaining the abovementioned definition, namely, *big-step semantics* (referred to as *natural semantics* in [7]). In big-step semantics, the relation $\langle t, \rho \rangle \downarrow v$, where t is a term, ρ is an assignment to free variables in t, and v is the value of t under assignment ρ, is defined. We treat $\langle t, \rho \rangle \downarrow v$ as a statement in a derivation and provide rules for deriving $\langle t, \rho \rangle \downarrow v$. For technical reasons, it is assumed that such derivations are directed acyclic graphs (DAGs) in this paper.

However, it is still not possible to prove the induction step for the substitution rule, because bounded arithmetic cannot prove the existence of a value for each term of PV. We overcome this difficulty by allowing an approximate value of a computation, in a way similar to that described in Beckmann's paper [1].

Then, we attempt to prove that $\langle t, \rho \rangle \downarrow v$ implies that $\langle u, \rho \rangle \downarrow v$ for any given assignment ρ by induction on the construction of the proof χ of $t = u$. We call this fact soundness (with respect to our computational semantics). It is possible to set bounds for all quantifiers that appear in the induction hypothesis of this induction by setting a bound on the Gödel number of ρ and bounds on the Gödel numbers of the derivation of $\langle t, \rho \rangle \downarrow v$ and $\langle u, \rho \rangle \downarrow v$. Because induction is carried out on bounded formulas, the proof can be carried out inside S^i_2 for some i. Let the number of primitive symbols in a size(a). We can show that size(ρ) is polynomially bounded by size(χ).

The bounds for the derivations are more difficult to obtain. Although it is possible to bound the number of *nodes* in the above-mentioned derivations, bounds for the *Gödel numbers* of these derivations are not trivially obtained, because there are no (obvious) bounds for the terms that appear in the derivations. This difficulty is overcome by employing the *call-by-value* style of big-step semantics, in which a derivation has the form

$$\frac{\langle f_1(\vec{x}), \nu_1\rangle \downarrow w_1, \quad \dots \quad \langle f_k(\vec{x}, y_1, \dots, y_{l-1}), \nu_l\rangle \downarrow w_l, \quad (\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\dots,m}}{\langle f(\vec{t}), \rho\rangle \downarrow v.} \tag{1}$$

where ν_j denotes the environment that maps x_i to v_i and y_k to w_k for $i = 1, \dots, m$ and $k = 1, \dots, j-1$. m is the number of the arguments of f. Because the numbers of symbols in $t_1, \dots, t_m$ and $f_1, \dots, f_l$ are bounded by $\text{size}(f(\vec{t}))$, and the size of the values appearing in the derivation can be proven to be polynomially bounded by the number of nodes in the derivation and the size of conclusions, the size of the terms that appear in this derivation can be polynomially bounded by the number of nodes and size of the conclusions of the derivation. Thus, all the quantifiers in the induction hypothesis are bounded by the Gödel number of χ.

The part of the induction step that is most difficult to prove is the soundness of the substitution rule. The proof is divided into two parts. First, it is proven that if σ derives $\langle t_1[u/x], \rho\rangle \downarrow v_1, \dots, \langle t_n[u/x], \rho\rangle \downarrow v_n$ and contains a computation of $\langle t, \rho\rangle \downarrow v$, then there exists τ that derives $\langle t_1, \rho[x \mapsto v]\rangle \downarrow v_1, \dots, \langle t_n, \rho[x \mapsto v]\rangle \downarrow v_n$ (Substitution I). Next, it is proven that if σ derives $\langle t_1, \rho[x \mapsto v]\rangle \downarrow v_1, \dots, \langle t_n, \rho[x \mapsto v]\rangle \downarrow v_n$ and contains a computation of $\langle t, \rho\rangle \downarrow v$, then there exists τ that derives $\langle t_1[u/x], \rho\rangle \downarrow v_1, \dots, \langle t_n[u/x], \rho\rangle \downarrow v_n$ (Substitution II).

The intuition underlying the proof of Substitution I is explained as follows. The naïve method, which uses induction on the length of σ, is ineffective. This is because an assumption of the last inference of σ may be used as an assumption of another inference; thus, it may not be a conclusion of σ_1, which is obtained from σ by removing the last inference. Therefore, it is not possible to apply the induction hypothesis to σ_1. To transform all the assumptions into conclusions, it is necessary to increase the length of σ_1 from σ by duplicating the inferences from which the assumptions are derived. Therefore, induction cannot be used on the length of σ.

Instead, we use induction on $\text{size}(t_1[\varepsilon/x]) + \cdots + \text{size}(t_n[\varepsilon/x])$ where ε is a constant symbol. Then, we prove that for all $\text{nodes}(\sigma) \leq U - \text{size}(t_1[\varepsilon/x]) - \cdots - \text{size}(t_n[\varepsilon/x])$, where U is a large integer that is fixed during the proof of soundness, we have τ, which derives $\langle t_1, \rho[x \mapsto w]\rangle \downarrow v_1, \dots, \langle t_n, \rho[x \mapsto w]\rangle \downarrow v_n$ and satisfies $\text{nodes}(\tau) \leq \text{nodes}(\sigma) + \text{size}(t_1[\varepsilon/x]) + \cdots + \text{size}(t_n[\varepsilon/x])$ where w is a value of u. Because all the quantifiers are bounded, the proof can be carried out in S_2, in particular S_2^2.

This paper is a revised version of the paper titled "Consistency proof of a feasible arithmetic inside a bounded arithmetic," [12] which was posted to ArXiv. It is revised from two aspects. First, it addresses the problem in the proof that causes Beckmann's counter-example. Second, it strengthens the meta-theory from S_2^1 to S_2^2, which is used to prove consistency. S_2^2 is necessary to prove the soundness of transitivity and substitution rules We discuss this point in Section 8.3.

The remainder of this paper is organized as follows. Section 2 summarizes the preliminaries. Section 3 introduces PV and PV^-, which is the target of our consistency proof. Section 4 introduces the notion of *(approximate) computation.* Section 5 shows that for each computation σ, $\text{size}(\sigma)$ is polynomially bounded by the number of nodes in σ and the number of primitive symbols in the conclusion of σ. Section 6 presents technical lemmas that are used in the consistency proof. Section 7 presents the proofs of the consistency of PV^-. Finally, Section 8 concludes the paper with a brief discussion.

2. Preliminary

The sequence $a_1, a_2, \dots, a_n$ is often abbreviated as $\vec{a}$. If we treat the sequence $a_1, a_2, \dots, a_n$ as a single object, we write $[a_1, a_2, \dots, a_n]$. For each sequence a, $(a)_i$ is its i-th element. We denote an empty sequence by $[\,]$ in the meta-language. For integer n, $|n|$ denotes its length of binary representation. For a set A of integers, $\sum A$ denotes the sum of all members of A.

Many types of objects are considered as proofs of PV, terms of PV, or the *computation* of these terms, all of which require the assignment of Gödel numbers to them. As all the objects under consideration can be coded as finite sequences of primitive symbols, it will suffice to encode these sequences of symbols. Variables $x_1, x_2, \dots$ are encoded by variable names x and natural numbers $1, 2, \dots$, which can be represented by binary strings. Function symbols for all polynomial time functions are encoded using trees of the primitive functions and labels that show how the function is derived using Cobham's inductive definition of polynomial time functions. Thus, the symbols that are used in our systems are finite, which enables us to use the numbers $0, \dots, N$ to code these symbols. Then, the sequence of symbols is coded as $N+1$-adic numbers.

For each object a consisting of symbols, $\text{size}(a)$ denotes the number of primitive symbols in a, that is, the number of $N+1$-adic numbers in its Gödel number. If a is a sequence or tree in an object language, $\text{nodes}(a)$ denotes the number of nodes in a.

We use the notation $a \equiv b$ when a and b are syntactically equivalent.

For a given term t, the notion of subterm u is defined in the usual way. Further, u may be identical to t. If $t \not\equiv u$, we call u a proper subterm.

3. PV and related systems

In this section, we introduce our version of PV and PV^-.

PV is formulated as a theory of binary strings rather than integers. We identify binary strings and integers which are represented in little endian (the least significant bit appears at the right most position). The differences between our version of PV and the original PV are discussed in Section 8.1.

PV provides the symbols for the empty sequence ε and its binary successors $0, 1$ denoted by $b, b_1, \dots$, which add 0 or 1 to the leftmost positions of strings. If a term is solely constructed by $\varepsilon, 0, 1$, it is referred to as a *numeral*. Although binary successors are functions, the notation we use for them employs a special convention to omit the parentheses after the function symbol. Thus, we write $01x$ instead of $0(1(x))$.

The language of PV contains function symbols for all polynomial time functions. In particular, it contains the constant ε, binary successors $0, 1$ and ε^n, proj^i_n. The intuitive meaning is that ε^n is the n-ary constant function whose value is ε, and proj^k_n is the projection function. From here, a function symbol for a polynomial time function f and f itself are often identified. The terms are denoted by $t, t_1, \dots, u, r, s, \dots$

For each function symbol f of a polynomial time function, let $\text{Base}(f)$ be the set of function symbols that are used in Cobham's recursive definition of f. We assume that $\text{Base}(f)$ always contains ε, 0, and 1, regardless of f. For a set of function symbols S, we define $\text{Base}(S) = \bigcup_{f \in S} \text{Base}(f)$. If α represents any sequence of symbols, $\text{Base}(\alpha)$ is defined by the union of $\text{Base}(f)$ for the function symbols f

that appear in α. Base(f) is computable by a polynomial time function. For each function symbol f, ar(f) is the arity of f. We encode $f \equiv \varepsilon, 0, 1, \varepsilon^n, \mathrm{proj}_n^i$ by

$$\ulcorner \varepsilon \urcorner = \ulcorner [\mathrm{Fun}, \varepsilon] \urcorner \tag{2}$$

$$\ulcorner 0 \urcorner = \ulcorner [\mathrm{Fun}, 0] \urcorner \tag{3}$$

$$\ulcorner 1 \urcorner = \ulcorner [\mathrm{Fun}, 1] \urcorner \tag{4}$$

$$\ulcorner \varepsilon^n \urcorner = \ulcorner [\mathrm{Fun}, \varepsilon, \overbrace{\# \cdots \#}^{n}] \urcorner \tag{5}$$

$$\ulcorner \mathrm{proj}_n^i \urcorner = \ulcorner [\mathrm{Fun}, \mathrm{proj}, i, \overbrace{\# \cdots \#}^{n}] \urcorner \tag{6}$$

where # is a "filler" symbol. Then, a function defined by the composition of g and $h_1, \ldots, h_m$ is encoded as $[\mathrm{Fun}, \mathrm{comp}, g, h_1, \ldots, h_n]$. A function defined by the recurrence of g_ε, g_0, g_1 is encoded as $[\mathrm{Fun}, \mathrm{rec}, g_\varepsilon, g_0, g_1]$. Then, for any function symbols f, which are defined by Cobham's inductive definition, ar$(f) \leq$ size(f) is satisfied.

The only predicate in the vocabulary of PV is the equality $=$. Our PV does not have inequalities $\leq, \geq, \ldots$. The formulas $t_1 = t_2$ of PV are formed by connecting two terms t_1, t_2 by the equality $=$. We consider PV to be purely equational; hence, the formulas do not contain propositional connectives and quantifiers.

There are three types of axioms and inferences in PV: defining axioms, equality axioms, and induction. We consider that the proofs in PV are all tree-like, not DAG-like. This restriction to the representation of proofs is essential to our consistency proof.

3.1. **Defining axioms.** For all of Cobham's defining equations of polynomial time functions, there are corresponding defining axioms in PV. For the constant function ε^n, the defining axiom is

$$\varepsilon^n(x_1, \ldots, x_n) = \varepsilon \tag{7}$$

for a positive integer n. For the projection function, the defining axiom is

$$\mathrm{proj}_n^i(x_1, \ldots, x_n) = x_i \tag{8}$$

for a positive integer n and an integer $i, 1 \leq i \leq n$. For the binary successor functions 0 and 1, there is no defining axiom. If the function $f(x_1, \ldots, x_n)$ is defined by the composition of g and $h_1, \ldots, h_m$, the defining axiom is

$$f(x_1, \ldots, x_n) = g(h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n)). \tag{9}$$

For the function defined by recursion of binary strings, there are three defining axioms:

$$f(\varepsilon, x_1, \cdots, x_n) = g_\varepsilon(x_1, \cdots, x_n) \tag{10}$$

$$f(0x, x_1, \cdots, x_n) = g_0(x, f(x, \vec{x}), x_1, \cdots, x_n) \tag{11}$$

$$f(1x, x_1, \cdots, x_n) = g_1(x, f(x, \vec{x}), x_1, \cdots, x_n). \tag{12}$$

Using Cobham's recursive definition of polynomial time functions, it is easy to see that all polynomial time functions can be defined using these defining axioms. Even though Cook and Urquhart's PV [6] requires all recursion schema to be bounded by a function with a polynomial growth rate, we do not impose this restriction.

Thus, our theory can be extended beyond polynomial time functions. However, this paper focuses on the theory based on polynomial time functions.

We present defining axioms of forms $f(\vec{x}) = t$, but we also introduce defining axioms of forms $t = f(\vec{x})$.

3.2. **Equality axioms.** The identity axiom is formulated as

$$t = t \tag{13}$$

The remaining equality axioms are formulated as inference rules rather than axioms.

$$\frac{u = t}{t = u} \tag{14}$$

$$\frac{t = u \quad u = r}{t = r} \tag{15}$$

$$\frac{t_1 = u_1 \quad \cdots \quad t_n = u_n}{f(t_1, \dots, t_n) = f(u_1, \dots, u_n)} \tag{16}$$

$$\frac{t(x) = u(x)}{t(r) = u(r)} \tag{17}$$

for any term r.

3.3. **Induction.**

$$\frac{t_1(\varepsilon) = t_2(\varepsilon) \quad t_1(s_i x) = v_i(t_1(x)) \quad t_2(s_i x) = v_i(t_2(x)) \quad (i = 0, 1)}{t_1(x) = t_2(x)} \tag{18}$$

The system PV contains defining axioms, equality axioms, and induction as axioms and inference rules. In contrast, the system PV^- contains only defining axioms and equality axioms as axioms and inference rules. This paper demonstrates that the consistency of PV^- can be proven by S^2_2.

4. Approximate computation

In this section, we define the notion of *approximate computations* as being the representation of the evaluations of the terms of PV. The idea that the computation values can be approximated using the $*$ symbol comes from Beckmann [1] but we only allow $*$ contained in numerals.

Definition 1 (Approximate values)**.** Let $\mathbb{D}$ be terms that are created by $0, 1$ from constants ε, $*$. $*$ stands for the *unknown* value. The elements of $\mathbb{D}$ are called *g-numerals*. For $v \in \mathbb{D}$, nodes(v) is defined as the number of symbols $*, \varepsilon, 0, 1$. $\mathbb{D}$ has an order structure. For any $v, w \in \mathbb{D}$, let $\trianglelefteq$ be the relation that is recursively defined by

(1) $\varepsilon \trianglelefteq \varepsilon$
(2) $v \trianglelefteq *$
(3) $v \trianglelefteq w \implies bv \trianglelefteq bw$ for $i = 0, 1$.

If $v \trianglelefteq w$, w is often written as v^*.

Lemma 1 (S^1_2)**.** *$\trianglelefteq$ in $\mathbb{D}$ is the order relation.*

Proof. Transitivity law : we prove that f $v \trianglelefteq w$ and $w \trianglelefteq z$, $v \trianglelefteq z$ by induction on $\mathrm{nodes}(v)+\mathrm{nodes}(w)+\mathrm{nodes}(z)$. If $z \equiv *$ then the conclusion follows. If $v \equiv *$ then $w \equiv *$ and $z \equiv *$ must hold. Therefore, $v \trianglelefteq z$. Next, if $z \equiv \varepsilon$, then $v \equiv w \equiv \varepsilon$. Thus, $v \trianglelefteq z$. If $v \equiv \varepsilon$ then z must be either $*$ or ε. For both cases, $v \trianglelefteq z$. Finally, $v \equiv b_1 v'$, $w \equiv b_2 w'$ and $z \equiv b_3 z'$. Then, $b_1 \equiv b_2 \equiv b_3$, $v' \trianglelefteq w'$ and $z \trianglelefteq z'$ hold. By induction hypothesis, $v' \trianglelefteq z'$ is therefore $v \trianglelefteq z$. Other cases are trivial.

Anti-symmetry law : we prove f $v \trianglelefteq w$ and $w \trianglelefteq v$, $v \equiv w$ by induction on $\mathrm{nodes}(v) + \mathrm{nodes}(w)$. If $v \equiv *$, then w must be $*$; therefore, $v \equiv w$. Similarly, if $v \equiv \varepsilon$, then w must be ε; therefore, $v \equiv w$. By a symmetric argument, we can assume that v and w are neither $*$ nor ε. Then, $v \equiv bv'$ and $w \equiv bw'$. $v' \trianglelefteq w'$ and $w' \trianglelefteq v'$ must hold. Therefore, $v' \equiv w'$. Thus, $v \equiv w'$. □

Definition 2. Let $v_1, \ldots, v_n$ be g-numerals and t be a term of PV^- with free variables $x_1, \ldots, x_n$. An *environment* ρ of t is a map from $x_1, \ldots, x_n$ to $v_1, \ldots, v_n$ respectively. Let $\mathrm{dom}(\rho)$ be $\{x_1, \ldots, x_n\}$. Let

$$B(\rho) = \max_{i=1}^{m} \mathrm{nodes}(\rho(x_i)), \tag{19}$$

$L(\rho) = n$ and $S(\rho) = \mathrm{size}(\rho_i)$. The empty environment is denoted by [ ].

Let v be a g-numeral, t be a term of PV^-, and ρ be an environment of t. The form $\langle t, \rho \rangle \downarrow v$ is referred to as a *(computation) judgment*, t as the main term, ρ as the environment, and v as the value (of t). Because we allow approximate computations, a term t may have several g-numerals as values under the same environment.

If a computational judgment $\langle t, \rho \rangle \downarrow v$ has a form $\langle f(x_1, \ldots, x_n), \rho \rangle \downarrow w$ or $\langle v, \rho \rangle \downarrow w$ where v is a numeral, then it is called *purely numerical.* An inference of a computational judgment is also called purely numerical if its conclusion and premises are purely numerical.

A computation judgment can be derived using the following rules. Each rule is attached by a symbol such as $*$ called a *label.*

In the following rules, on the contrary to the case of terms $t(t_1, \ldots, t_n)$, $f(t_1, \ldots, t_n)$ for any function symbol f means that $t_1, \ldots, t_n$ really appears in $f(t_1, \ldots, t_n)$.

$$\frac{}{\langle t, \rho \rangle \downarrow *} \; * \tag{20}$$

for any term t.

$$\frac{}{\langle x, \rho[x \mapsto v] \rangle \downarrow v^*} \; \mathrm{Env} \tag{21}$$

where $v \trianglelefteq v^*$.

$$\frac{}{\langle v, \rho \rangle \downarrow v^*} \; v \tag{22}$$

where v is a numeral and v^* is an approximation of v.

$$\frac{\langle t, \rho \rangle \downarrow v}{\langle bt, \rho \rangle \downarrow bv^*} \; b \tag{23}$$

where b is either 0 or 1, v^* is an approximation of v, v is a g-numeral and t is not a numeral.

$$\frac{(\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\ldots,m}}{\langle \varepsilon^m(t_1, \ldots, t_m), \rho\rangle \downarrow \varepsilon}\ \varepsilon^m \tag{24}$$

where ε^m is the m-ary constant function of which the value is always ε. $(\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\ldots,m}$ is the sequence $\langle t_{i_1}, \rho\rangle \downarrow v_1, \ldots, \langle t_{i_k}, \rho\rangle \downarrow v_m$ of judgments. We use the similar notation from here.

$$\frac{(\langle t_j, \rho\rangle \downarrow v_j)_{j=1,\ldots,m}}{\langle \mathrm{proj}^i_m(t_1, \ldots, t_m), \rho\rangle \downarrow v_i^*}\ \mathrm{proj}^i_m \tag{25}$$

for $i = 1, \cdots, m$. v_i^* is an approximation of v_i.

If f is defined by composition, we have the following rule.

$$\frac{\langle g(\vec{y}, \xi\rangle \downarrow z \quad \langle h_1(\vec{x}), \nu\rangle \downarrow w_1 \cdots \langle h_m(\vec{x}), \nu\rangle \downarrow w_m \quad (\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\ldots,n}}{\langle f(t_1, \ldots, t_n), \rho\rangle \downarrow z^*}\ \mathrm{comp} \tag{26}$$

where $\vec{y} = y_1, \ldots, y_m$, $\vec{x} = x_1, \ldots, x_n$, $\nu(x_i) = v_i, i = 1, \ldots, n$ and $\xi(y_j) = w_j, j = 1, \ldots, m$.

If f is defined by recursion, we have the following rules.

$$\frac{\langle g_\varepsilon(x_1, \ldots, x_n), \xi\rangle \downarrow z \quad \langle t, \rho\rangle \downarrow \varepsilon \quad (\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\ldots,n}}{\langle f(t, t_1, \ldots, t_n), \rho\rangle \downarrow z^*}\ \text{rec-}\varepsilon \tag{27}$$

where $\xi(x_i) = v_i$ for $i = 1, \ldots, n$.

$$\frac{\langle g_i(x_0, y, \vec{x}), \xi\rangle \downarrow z \quad \langle t, \rho\rangle \downarrow iv_0 \quad \langle f(x_0, \vec{x}), \nu\rangle \downarrow w \quad (\langle t_j, \rho\rangle \downarrow v_j)_{j=1,\ldots,n}}{\langle f(t, t_1, \ldots, t_n), \rho\rangle \downarrow z^*}\ \text{rec-}b \tag{28}$$

where $b = 0, 1$ and $\vec{x} = x_1, \ldots, x_n$. The environment ν is defined by $\nu(x_j) = v_j$ for $j = 1, \ldots, n$ and $\nu(x_0) = v_0$, while ξ is defined by $\xi(x_j) = v_j, \xi(x_0) = v_0, \xi(y) = w$.

Definition 3 (Computation Sequence)**.** A *computation sequence* σ is a sequence $\sigma_1, \ldots \sigma_L$, where each σ_i is a sequence

$$[R, \langle t_i, \rho_i\rangle \downarrow v_i, n_{1i}, \ldots, n_{l_i i}] \tag{29}$$

which satisfies $n_{ji} < i, j = 1, \ldots, l_i$. Each inference

$$\frac{(\sigma_{n_{1i}})_2 \quad \cdots \quad (\sigma_{n_{l_i i}})_2}{(\sigma_i)_2}\ (\sigma_i)_1 \tag{30}$$

must be a valid computation rule. Here, $(a)_i$ is a projection of $[a_1, \ldots, a_n]$ to a_i. The computation judgments that are not used as assumptions of some inference rule are referred to as *conclusions* of σ. If $\langle t, \rho\rangle \downarrow v$ is the only conclusion of σ, it is written as $\sigma \vdash \langle t, \rho\rangle \downarrow v$; however, if σ has multiple conclusions $\vec{\alpha}$, it is written as $\sigma \vdash \vec{\alpha}$. If $\sigma \vdash \langle t, \rho\rangle \downarrow v, \vec{\alpha}$, σ is often considered to be a computation of $\langle t, \rho\rangle \downarrow v$. Although a computation sequence σ is a sequence, σ is often considered to be a DAG, of which the conclusions form the lowest elements.

If there is a computation sequence σ with conclusions $\langle t, \rho\rangle \downarrow v, \vec{\alpha}$ such that $\mathrm{nodes}(\sigma) \leq b$, we write $\vdash_b \langle t, \rho\rangle \downarrow v, \vec{\alpha}$.

For any sequence of computational judgments $\vec{\alpha} = \langle t_1, \rho_1\rangle \downarrow v_1, \ldots, \langle t_n, \rho_n\rangle \downarrow v_n$, $T(\vec{\alpha}) = \max\{\mathrm{size}(t_1), \ldots, \mathrm{size}(t_n)\}$. For a computation σ, $M(\sigma)$ is defined as the maximal size of the main terms of computational judgments in σ, and $T(\sigma) = T(\vec{\alpha})$ if $\vec{\alpha}$ are conclusions of σ. For computational judgments α above, $B(\alpha)$, $L(\alpha)$, and $S(\alpha)$ are defined by $\max_{i=1}^{n} B(\rho_i)$, $\max_{i=1}^{n} L(\rho_i)$ and $\max_{i=1}^{n} S(\rho_i)$ respectively. For a computation σ with the conclusion α, $B(\sigma) = B(\alpha)$, $L(\sigma) = L(\alpha)$ and $S(\sigma) = S(\alpha)$.

We would like to show that $\vdash_{|B|} \langle t, \rho \rangle \downarrow v, \vec{\alpha}$ is definable in S_2^1. The obstacle to do this is that, in the above definition, only the number of nodes of σ, and not the number of primitive symbols, is bounded. Thus, it is required to bound polynomially size(σ) by nodes(σ). This task is carried out in Section 5.

5. Estimating the size of a computation

This section proves the polynomial upper bound of the size of a computation with respect to the number of nodes of the computation together with the size of its conclusions. (S_2^1) means that a statement is provable in the theory S_2^1, and (S_2^2) means that a statement is provable in the S_2^2.

Lemma 2 (S_2^1)**.** *Let σ be a computation of $\langle t_1, \rho_1 \rangle \downarrow v_1, \dots, \langle t_m, \rho_m \rangle \downarrow v_m$. Then,* Base$(t_1, \dots, t_m)$ *contains all function symbols that appear in the main terms of σ. If $f(x_1, \dots, x_n)$ is a main term that appears in σ,* size$(f) \leq T(\sigma)$ *and*

$$\text{size}(f(x_1, \dots, x_n)) \leq T(\sigma) + 2 + \sum_{i=1}^{n} \text{size}(x_i) \tag{31}$$

$$\leq p_M(T(\sigma)) \tag{32}$$

for a polynomial p_M.

Proof. The first half of the lemma is proven by induction on σ. Because f is contained in Base$(t_1, \dots, t_m)$, size$(f) \leq T(\sigma)$. size(x_i) is polynomially increased by size(i), as x_i is a compound symbol constructed from the symbol x and i. Because $i \leq n \leq \text{ar}(f) \leq \text{size}(f) \leq T(\sigma)$, size$(i) \leq T(\sigma)$. Thus, there is a polynomial p_M that satisfies (32). □

Lemma 3 (S_2^1)**.** *Let v is a g-numeral that appears as a value in σ. Then,* nodes$(v) \leq \max\{B(\sigma), T(\sigma)\} +$ nodes(σ).

To prove this lemma, we define a wighted directed graph G_σ and prove related lemmas.

Definition 4. In a computational judgement $\langle t, \rho \rangle \downarrow v$, we call v the righthand and $\langle t, \rho \rangle$ the lefthand. The nodes of the weighted directed graph G_σ consist of all righthands and lefthands of computational judgements of σ. For each node η, we define an integer $N(\eta)$ as nodes(v) if η is a value v, and as $\max(N(t), B(\rho))$, where $N(t)$ is defined as the maximal nodes(v) of numerals v which are contained in t, if η is the lefthand $\langle t, \rho \rangle$. For each inference of σ, edges of G_σ are defined as follows.

$$\frac{}{\langle t, \rho \rangle \downarrow *} \; * \tag{33}$$

In this case, we connect a edge from t to $*$. The weight is 0.

$$\frac{}{\langle x, \rho[x \mapsto v] \rangle \downarrow v^*} \; \text{Env} \tag{34}$$

In this case, we connect a edge from $\langle x, \rho[x \mapsto v]\rangle$ to the the righthand side v^*. The weights of the edge are 0.

$$\frac{}{\langle v, \rho\rangle \downarrow v^*}\ v \tag{35}$$

In this case, we connect a edge from v to v^*. The weight is 0.

$$\frac{\langle t, \rho\rangle \downarrow v}{\langle bt, \rho\rangle \downarrow bv^*}\ b \tag{36}$$

In this case, we connect the lefthand of the lower judgement to the lefthand of the upper judgement and the righthand of the upper judgement to the righthand of the lower judgement. The weight is 0 for the edge which connects the lefthand and 1 for the edge which connects the righthand.

$$\frac{(\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\dots,n}}{\langle \varepsilon^n(t_1, \dots, t_n), \rho\rangle \downarrow \varepsilon}\ \varepsilon^n \tag{37}$$

In this case, we connect the lefthand of the conclusion to all lefthands of the premises. The weights are all 0. ε is connected from its lefthand side of the judgement. The weight is 0.

$$\frac{(\langle t_j, \rho\rangle \downarrow v_j)_{j=1,\dots,n}}{\langle \mathrm{proj}^i_n(t_1, \dots, t_n), \rho\rangle \downarrow v_i^*}\ \mathrm{proj}^i_n \tag{38}$$

In this case, we connect the lefthand of the conclusion to all lefthands of the premises, and v_i in the premise to v_i^* in the conclusion. The weights are all 0.

$$\frac{\langle g(\vec{y}), \mu\rangle \downarrow z \quad \langle h_1(\vec{x}), \nu\rangle \downarrow w_1 \cdots \langle h_m(\vec{x}), \nu\rangle \downarrow w_m \quad (\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\dots,n}}{\langle f(t_1, \dots, t_n), \rho\rangle \downarrow z}\ \mathrm{comp} \tag{39}$$

In this case, we connect the lefthand of the conclusion to all $\langle t_i, \rho\rangle, i = 1, \dots, m$. The weights are 0. Further, all $v_i, i = 1, \dots, n$ in the premises are connected to $\langle h_j(\vec{x}), \nu\rangle, j = 1, \dots, m$. z in the premises is connected to z in the conclusion. The weights are all 0.

$$\frac{\langle g_\varepsilon(\vec{x}), \nu\rangle \downarrow z \quad \{\langle t, \rho\rangle \downarrow \varepsilon\} \quad (\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\dots,n}}{\langle f(t, t_1, \dots, t_n), \rho\rangle \downarrow z}\ \text{rec-}\varepsilon \tag{40}$$

In this case, $\langle t, \rho\rangle$ and $\langle t_i, \rho\rangle, i = 1, \dots, n$ in the premises are connected from $\langle f(t, t_1, \dots, t_n), \rho\rangle$. $v_i, i = 1, \dots, n$ is connected to $\langle g_\varepsilon(\vec{x}), \nu\rangle$. z in the premise is connected to z in the conclusion. The weights are all 0.

(41)

$$\frac{\langle g_b(x_0, y, \vec{x}), \xi\rangle \downarrow z \quad \{\langle t, \rho\rangle \downarrow bv_0\} \quad \langle f(y, \vec{x}), \nu\rangle \downarrow w \quad (\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\dots,n}}{\langle f(t, t_1, \dots, t_n), \rho\rangle \downarrow z}\ \text{rec-}b$$

In this case, $\langle t, \rho\rangle$ and $\langle t_i, \rho\rangle, i = 1, \dots, n$ in the premises are connected from $\langle f(t, t_1, \dots, t_n), \rho\rangle$. All $v_i, i = 1, \dots, n$ and bv_0 are connected to $\langle f(y, \vec{x}), \nu\rangle$. bv_0, all $v_i, i = 1, \dots, n$ and w are connected to $\langle g_i(x_0, y, \vec{x}), \xi\rangle$. Finally, z in the premise is connected to z in the conclusion. The weights are all 0.

Lemma 4 (S^1_2)**.** *Let η be a node of G_σ. Assume that from $\eta_1, \dots, \eta_k$ the edges $e_1, \dots, e_k$ run to η. If $k \geq 1$, $N(\eta) \leq w(e_i) + N(\eta_i)$ for some $i = 1, \dots, k$ where $w(e_i)$ is the weight of e_i.*

Let $p = \eta_0 \overset{e_1}{\leftarrow} \eta_1 \overset{e_2}{\leftarrow} \eta_2 \cdots \overset{e_l}{\leftarrow} \eta_l$ be a path in G_σ. p is called a *bounding path* if for each $k = 0, \dots, l-1$, $N(\eta_k) \leq w(e_{k+1}) + N(\eta_{k+1})$ holds.

Lemma 5. *(S^1_2) If a judgement $\langle t, \rho \rangle \downarrow v$ appears in σ, there is a bounding path from $\langle t, \rho \rangle$ to v in G_σ.*

Proof. Assume that $\sigma_i \equiv \langle t, \rho \rangle \downarrow v$. We prove the statement of the lemma by induction on i. □

Lemma 6. *(S^1_2) If a judgement $\langle t, \rho \rangle \downarrow v$ appears in σ, there is a bounding path of $\langle t, \rho \rangle$ from a lefthand of a conclusion.*

Proof. Assume that $\sigma_i \equiv \langle t, \rho \rangle \downarrow v$. We prove the statement of the lemma by induction on i. □

Lemma 7. *(S^1_2) To each righthand side of an assumption of an inference in σ, there is a bounding path of G_σ from a lefthand of a conclusion.*

Proof. By Lemmas 5 and 6. □

Lemma 8. *(S^1_2) For each node η of G_σ, there is an acyclic path $\eta \overset{e_1}{\leftarrow} \eta_1 \overset{e_2}{\leftarrow} \eta_2 \cdots \overset{e_l}{\leftarrow} \eta_l$ such that*

(1) *η_l is a lefthand of a conclusion of σ.*
(2) *$N(\eta) \leq N(\eta_l) + \sum_{i=1}^{l} w(e_i)$.*

Proof. By Lemma 5 and 7, there is a bounding path p from a lefthand of conclusion to η. Assume that it contains a cycle $\eta_i, \dots, \eta_j = \eta_i$. Because p is a bounding path, $N(\eta_i) = N(\eta_j) \leq w(e_{j+1}) + N(\eta_{j+1})$. Thus, $\eta_1, \dots, \eta_i, \eta_{j+1}, \eta_l$ is also a bounding path. In this way, we can remove all cycles from p. □

Proof of Lemma 3. For each value v which appears in a judgement of σ, $N(v) \leq N(\langle t_k, \rho_k \rangle) + \sum_{i=1}^{l} w(e_i)$ holds, where $v \overset{e_1}{\leftarrow} \eta_1 \overset{e_2}{\leftarrow} \eta_2 \cdots \overset{e_l}{\leftarrow} \eta_l = \langle t_k, \rho_k \rangle$ is an acyclic bounding path from a righthand of a conclusion to v. $N(\langle t_k, \rho_k \rangle) \leq \max(T(\sigma), B(\sigma))$ holds. $\sum_{i=1}^{l} w(e_i)$ is bounded by the number of b-rules in σ. Thus, $N(v)$ is bounded by $\max(T(\sigma), B(\sigma)) + \mathrm{nodes}(\sigma)$. □

Lemma 9 (S^1_2)**.** *Let σ be a computation of $\langle t_1, \rho_1 \rangle \downarrow v_1, \dots, \langle t_m, \rho_m \rangle \downarrow v_m$. Then, for any environment ρ that appears in σ, $L(\rho) \leq \max\{L(\sigma), T(\sigma)\}$, $B(\rho) \leq \max\{B(\sigma), T(\sigma)\} + \mathrm{nodes}(\sigma)$ and $S(\rho) \leq p_S(B(\sigma), T(\sigma), \mathrm{nodes}(\sigma))$ for some polynomial p_S.*

Proof. $L(\rho) \leq \max\{L(\sigma), T(\sigma)\}$ holds because $\mathrm{ar}(f) \leq T(\sigma)$ for f that appears in σ. $B(\rho) \leq \max\{B(\sigma), T(\sigma)\} + \mathrm{nodes}(\sigma)$ holds by Lemma 3. The polynomial bound for $S(\sigma)$ is obtained from the bounds of $L(\sigma)$ and $B(\sigma)$ together with the fact that all variables in ρ appear in either environments of conclusions of σ or $x_1, \dots, x_{\mathrm{ar}(f)}, f \in \mathrm{Base}(\vec{\alpha})$ where $\vec{\alpha}$ are the conclusions of σ. □

Lemma 10. *(S^1_2) There is a polynomial p such that if there is a computation σ of $\langle t_1, \rho_1 \rangle \downarrow v_1, \dots, \langle t_m, \rho_m \rangle \downarrow v_m$ then $\mathrm{size}(\sigma) \leq p(\mathrm{size}(\vec{t}), \mathrm{size}(\vec{\rho}), \mathrm{nodes}(\sigma))$ holds.*

Proof. By Lemma 2, 3 and 9. □

Lemma 11. *The relation $\vdash_{|B|} \vec{\alpha}$ on integers B and judgments $\vec{\alpha}$ can be defined using a Σ^b_1-formula.*

Proof. Immediate from Lemma 10. □

6. Basic properties of computations

In this section, the basic properties of computations are proved. After proving technical lemmas (Lemma 12, 13), we prove the lemmas concerning the forms of values of computations of $\varepsilon, 0t, 1t$ and numerals (Lemma 14, 15). Lemma 15 is crucial for our consistency proof because it shows that the numerals are only computed to the equal numerals. Next we prove Lemma 16, which states the values v_1 and v_2 obtained by computations of the same term t are always compatible, that is, either $v_1 \trianglelefteq v_2$ or $v_2 \trianglelefteq v_1$. This enables us to extract the most "accurate" value $v(t, \rho, \sigma)$ of a term t from a computation σ of t under an assignment ρ (Definition 5). Substitution lemmas (Lemma 20, 21) establish the relation between substitution into a term which is evaluated by a computation, and assignment in the environment in which the term is evaluated. These lemmas enable us to extend a consistency proof to the substitution rule. Unlike other lemmas in this section, substitution lemmas are proved in S^2_2. Finally, we prove Lemmas 22 and 23 which are used to prove "soundness" of defining axioms in Section 7.

Lemma 12 (S^1_2)**.** *If $\langle t, \rho \rangle \downarrow v$ appears as a node in the computation sequence $\sigma, \sigma \vdash \vec{\alpha}$ (as a DAG), there is a computation sequence τ such that $\tau \vdash \langle t, \rho \rangle \downarrow v, \vec{\alpha}$, $\mathrm{nodes}(\tau) \leq \mathrm{nodes}(\sigma) + 1$ and $M(\tau) = M(\sigma)$.*

Proof. If $\langle t, \rho \rangle \downarrow v$ is derived according to the inference R, another instance of R is added to σ, which uses the same assumptions as R, in which case τ is obtained. □

Lemma 13 (S^1_2)**.** *If there is a computation σ such that $\sigma \vdash \alpha, \vec{\alpha}$, then there exists τ such that $\tau \vdash \vec{\alpha}$ and $\mathrm{nodes}(\tau) \leq \mathrm{nodes}(\sigma)$.*

Lemma 14 (S^1_2)**.** *If $\langle \varepsilon, \rho \rangle \downarrow v$ is contained in a computation σ, then either $v \equiv \varepsilon$ or $v \equiv *$. If $\langle bt, \rho \rangle \downarrow v$, where t is not a numeral, is contained in σ, then either $v \equiv iv_0$ for some g-numeral v_0 or $v \equiv *$. If $v \equiv iv_0$, then σ contains $\langle t, \rho \rangle \downarrow v'_0$ and $v'_0 \trianglelefteq v_0$.*

Proof. By induction on $\mathrm{nodes}(\sigma)$. The only rule that can derive $\langle \varepsilon, \rho \rangle \downarrow v$ is either v or $*$-rule. Thus, v is either ε or $*$. Similarly, if σ derives $\langle bt, \rho \rangle \downarrow v$, the only rule that can derive this is $*$ or i-rule. If $v \equiv iv_0$, $\langle bt, \rho \rangle \downarrow iv_0$ can only be derived by i. Thus, the assumptions contain $\langle t, \rho \rangle \downarrow v'_0$ where $v'_0 \trianglelefteq v_0$. □

Lemma 15 (S^1_2)**.** *If $\langle v, \rho \rangle \downarrow w$, in which v is a numeral, is contained in a computation σ, then $v \trianglelefteq w$.*

Proof. The only rules that can derive $\langle v, \rho \rangle \downarrow w$ are $*$ and v-rules. □

Lemma 16 (S^1_2)**.** *Let t be a term and v, w are g-numerals. If both $\langle t, \rho \rangle \downarrow v$ and $\langle t, \rho \rangle \downarrow w$ are present in a computation, then $v \trianglelefteq w$ or $w \trianglelefteq v$.*

We write $v \bigtriangleup w$ when $v \trianglelefteq w$ or $w \trianglelefteq v$. $\bigtriangleup$ is reflexive and symmetric, but not transitive.

To prove the lemma, the next lemma, same as Lemma 4.4. in [1], is to be observed.

Lemma 17 (S^1_2)**.** *If g-numerals w, u, v have relations $w \trianglelefteq u, w \trianglelefteq v$, then $u \bigtriangleup v$.*

Proof. By induction on $\mathrm{nodes}(w) + \mathrm{nodes}(u) + \mathrm{nodes}(v)$.

If $w = *$, $u = v = *$; therefore, $u \bigtriangleup v$. If $w = \varepsilon$, u, v are either $*$ or ε. Therefore, $u \bigtriangleup v$. If $w = bw'$ for some $b = 0, 1$ and u', there are two possibilities of u.

(1) u is $*$
(2) u is bu'

If u is $*$, $u \bigtriangleup v$. Therefore, $u \bigtriangleup v$. If u is bu', $w' \trianglelefteq u'$. There are also two possibilities of v. The only non-trivial case is the case in which $v = bv'$. The other case is symmetric. By induction hypothesis, $u' \bigtriangleup v'$. Therefore, $u \bigtriangleup v$. □

Corollary 1 (S^1_2)**.** *If $u \bigtriangleup v$ for g-numerals u, v and $u \trianglelefteq u'$ and $v \trianglelefteq v'$, then $u' \bigtriangleup v'$.*

Proof. We assume $u \trianglelefteq v$. Then, $u \trianglelefteq v \trianglelefteq v'$. Thus, $u \trianglelefteq u'$ and $u \trianglelefteq v'$. Using Lemma 17, $u' \bigtriangleup v'$. □

Lemma 18 (S^1_2)**.** *Let f be a function symbol and $w_1, \dots, w_m$, $v_1, \dots, v_m$ be g-numerals such that $w_i \bigtriangleup v_i$ for any $i = 1, \dots, m$. Let $\rho(x_1) = w_1, \dots, \rho(x_m) = w_m$ and $\nu(x_1) = v_1, \dots, \rho(x_m) = v_m$. We assume that $\langle f(x_1, \dots, x_m), \rho\rangle \downarrow w$ and $\langle f(x_1, \dots, x_m), \nu\rangle \downarrow v$ are present in the same computation σ. Then, $w \bigtriangleup v$.*

Proof. We assume that $\langle f(x_1, \dots, x_m), \rho\rangle \downarrow w$ is i-the judgment of σ, while $\langle f(x_1, \dots, x_m), \nu\rangle \downarrow v$ is the j-th judgment of σ. The lemma is proved by induction on $i + j$ and case analysis of the rules that derive these statements. Although it appears that we use induction on Π^0_1-formula (because f and g-numerals $w_1, \dots, w_m, v_1, \dots, v_m$ are not bounded), we actually need to consider only those that are included in the computation σ. Thus, the quantifier is bounded. Because $*$ is compatible with any g-numerals, we assume that w and v are not $*$. This assumption makes it possible to uniquely determine the label R of the rules deriving i- and j-th judgment by f.

The possible R are among the b, ε^m, proj^i_m, comp, rec-ε, and rec-b-rule.

The case in which R is b, ε^m or proj^i_m is trivial.

The case in which R is comp is considered. The last rules have the following forms.

(42)
$$\frac{\langle g(\vec{y}), \rho_0\rangle \downarrow w \quad \langle h_1(\vec{x}), \rho^*\rangle \downarrow z_1 \cdots \langle h_k(\vec{x}), \rho^*\rangle \downarrow z_k \quad (\langle x_i, \rho\rangle \downarrow w^*_i)_{i=1,\dots,m}}{\langle f(x_1, \dots, x_m), \rho\rangle \downarrow w}\ \mathrm{comp}$$

(43)
$$\frac{\langle g(\vec{y}), \nu_0\rangle \downarrow v \quad \langle h_1(\vec{x}), \nu^*\rangle \downarrow z'_1 \cdots \langle h_k(\vec{x}), \nu^*\rangle \downarrow z'_k \quad (\langle x_i, \rho\rangle \downarrow v^*_i)_{i=1,\dots,m}}{\langle f(x_1, \dots, x_m), \nu\rangle \downarrow v}\ \mathrm{comp}$$

where $\rho^*(x_1) = w^*_1$ for $i = 1, \dots, m$, $\nu^*(x_i) = v^*_i$ for $i = 1, \dots, m$, $\rho_0(y_1) = z_1, \dots, \rho_0(y_k) = z_k$ and $\nu_0(y_1) = z'_1, \dots, \nu_k(y_k) = z'_k$. By induction hypothesis, $z_i \bigtriangleup z'_i$ for each $i = 1, \dots, m$. By induction hypothesis, $w \bigtriangleup v$.

Finally, the case in which the last rule is either rec-ε or rec-$b, b = 0, 1$ is considered. Because the case for rec-ε is similar to comp, we consider the case in which the last rule is rec-b.

$$\frac{\langle g_i(y, x_0, \vec{x}), \rho^*[y \mapsto z]\rangle \downarrow w \quad \langle f(x_0, \vec{x}), \rho^*\rangle \downarrow z \quad (\langle x_i, \rho\rangle \downarrow w_i)_{i=0,\dots,m}}{\langle f(x_0, \vec{x}), \rho\rangle \downarrow w} \tag{44}$$

$$\frac{\langle g_i(y, x_0, \vec{x}), \nu^*[y \mapsto z']\rangle \downarrow v \quad \langle f(x_0, \vec{x}), \nu^*\rangle \downarrow z' \quad (\langle x_i, \rho\rangle \downarrow v_i)_{i=0,\dots,m}}{\langle f(x_0, \vec{x}), \nu\rangle \downarrow v} \tag{45}$$

Here, $\vec{x} = x_1, \dots, x_m$, $\rho^*(x_1) = w_1^*, \dots, \rho^*(x_m) = w_m^*$ while $\rho^*(x_0) = w_0'$ when $w_0^* = bw_0'$. Similarly, $\nu^*(x_1) = v_1^*, \dots, \nu^*(x_m) = v_m^*$, while $\nu^*(x_0) = v_0^*$ when $v_0^* = bv_0'$. w_1^*, v_1^* must be in the above forms, otherwise the inference is not valid. Because $w_0 \bigtriangleup v_0, \dots, w_m \bigtriangleup v_m$, $w_0^* \bigtriangleup v_0^*, \dots, w_m^* \bigtriangleup v_m^*$. Further, by definition of $\trianglelefteq$, $w_0' \bigtriangleup v_0'$. By induction hypothesis, $z \bigtriangleup z'$. Again, by applying induction hypothesis, $w \bigtriangleup v$. □

Proof of Lemma 16. We assume that $\langle t, \rho\rangle \downarrow v$ is i-the judgment of σ, while $\langle t, \rho\rangle \downarrow w$ is the j-th judgment of σ. The lemma is proved by induction on $i + j$ and case analysis of the rules that derives these statements. Because the case in which either v or w is $*$ is trivial, we can assume that the rules that derive $\langle t, \rho\rangle \downarrow v$ and $\langle t, \rho\rangle \downarrow w$ has the same label R.

If R is Env, $t \equiv x$ for a variable x and both derivations have forms

$$\frac{}{\langle x, \rho[x \mapsto v_0]\rangle \downarrow v} \tag{46}$$

$$\frac{}{\langle x, \rho[x \mapsto v_0]\rangle \downarrow w} \tag{47}$$

where $v_0 \trianglelefteq v, w$. By Lemma 17, $v \bigtriangleup w$ holds.

Thus, we can assume that $t = f(t_1, \dots, t_n)$. If $f \equiv 0, 1$, then the proof is trivial by induction hypothesis.

Otherwise, the derivations have the following forms.

$$\frac{\vec{\beta} \quad (\langle t_i, \rho\rangle \downarrow v_i)_{i=1,\dots,n}}{\langle f(t_1, \dots, t_n), \rho\rangle \downarrow v} \tag{48}$$

$$\frac{\vec{\gamma} \quad (\langle t_i, \rho\rangle \downarrow w_i)_{i=1,\dots,n}}{\langle f(t_1, \dots, t_n), \rho\rangle \downarrow w} \tag{49}$$

where

$$\beta = \langle f_1(y_1^1, \dots, y_{m_1}^1), \rho_1\rangle \downarrow z_1, \dots \langle f_k(y_1^k, \dots, y_{m_1}^k), \rho_k\rangle \downarrow z_k \tag{50}$$

$$\gamma = \langle f_1(y_1^1, \dots, y_{m_1}^1), \nu_1\rangle \downarrow z_1', \dots \langle f_k(y_1^k, \dots, y_{m_1}^k), \nu_k\rangle \downarrow z_k'. \tag{51}$$

By induction hypothesis, $v_i \bigtriangleup w_i$ for all $i = 1, \dots, n$. Using Lemma 18 repeatedly, we obtain $z_1 \bigtriangleup z_1'$. Because $z_1 \trianglelefteq v$ and $z_1' \trianglelefteq w$, $v \bigtriangleup w$. □

Note that if $v_1 \bigtriangleup v_2$, the infimum of v_1 and v_2 exists. This fact enables the following definition.

Definition 5. Let t be a term of PV^-, and σ be a computation. Assume σ contains computational judgments $\langle t, \rho\rangle \downarrow v_1, \dots, \langle t, \rho\rangle \downarrow v_n$. By Lemma 16, $v_1, \dots, v_n$ are compatible. $v(t, \rho, \sigma)$ is defined as infimum of them. If there is no computational judgment of t with the environment ρ in σ, $v(t, \rho, \sigma) = *$.

Lemma 19. *Let σ be a computation, t be a term, and ρ be an environment. Let $v = v(t, \rho, \sigma)$. If v is a g-numeral other than $*$, σ contains a judgment $\langle t, \rho\rangle \downarrow v$.*

Lemma 20 (S^2_2, Substitution Lemma I)**.** *Let σ be a computation that contains occurrences of judgments*

$$\langle t_1(u_1, \dots, u_n), \rho \rangle \downarrow v_1, \dots, \langle t_m(u_1, \dots, u_n), \rho \rangle \downarrow v_m. \tag{52}$$

as conclusions. Assume that $v(u_i, \rho, \sigma) \trianglelefteq w_i$ for $i = 1, \dots, n$ and let $\rho' = \rho[x_1 \mapsto w_1, \dots, x_n \mapsto w_n]$ where $x_1, \dots, x_n$ are fresh variables. Then, there is a computation τ such that $\mathrm{nodes}(\tau) \leq \mathrm{nodes}(\sigma) + \sum_{j=1}^{m} \mathrm{size}(t_j(\varepsilon, \dots, \varepsilon))$ and τ has conclusions

$$\langle t_1, \rho' \rangle \downarrow v_1, \dots, \langle t_m, \rho' \rangle \downarrow v_m. \tag{53}$$

Further, τ contains all judgments in σ as judgments and all conclusions in σ as conclusions.

Each $\langle t_j(u_1, \dots, u_n), \rho \rangle \downarrow v_j, j = 1, \dots, m$ is an occurrence of a judgment but denoted as if it is a judgment by abusing notations. Similarly, $u_1, \dots, u_n$ in each $\langle t_j(u_1, \dots, u_n), \rho \rangle \downarrow v_j, j = 1, \dots, m$ are occurrences of terms but denoted as if they are terms.

Proof. Let U be a fixed integer larger then $\mathrm{nodes}(\sigma) + \sum_{j=1}^{m} \mathrm{size}(t_j(\varepsilon, \dots, \varepsilon))$. By induction on

$$\sum \{\mathrm{size}(s_d(\varepsilon, \dots, \varepsilon)) \mid \langle s_d(u_1, \dots, u_n), \rho \rangle \downarrow z_d \in A\} \tag{54}$$

and subinduction on $\mathrm{nodes}(\kappa)$, we prove the following induction hypothesis (Claim 1).

Claim 1. *Let κ be a computation with distinguished occurrences of judgments*

$$A \equiv \langle s_1(u_1, \dots, u_n), \rho \rangle \downarrow z_1, \dots, \langle s_k(u_1, \dots, u_n), \rho \rangle \downarrow z_k \tag{55}$$

among conclusions and satisfies

$$\mathrm{nodes}(\kappa) \leq U - \sum_{d=1}^{k} \mathrm{size}(s_d(\varepsilon, \dots, \varepsilon)) \tag{56}$$

$$T(\kappa) \leq T(\sigma) \tag{57}$$

$$B(\kappa) \leq B(\sigma). \tag{58}$$

Further, κ contains all judgments of σ. Then, there is a computation λ that has all conclusions of κ and $\langle s_1, \rho' \rangle \downarrow z_1, \dots, \langle s_k, \rho' \rangle \downarrow z_k$ as conclusions. λ satisfies

$$\mathrm{nodes}(\lambda) \leq \mathrm{nodes}(\kappa) + \sum_{d=1}^{k} \mathrm{size}(s_d(\varepsilon, \dots, \varepsilon)) \tag{59}$$

and contains all judgments in κ.

The claim is a Π^b_2-formula because first κ is universally quantified and λ is existentially quantified next. Because κ changes through the induction steps, quantification over κ is necessary. The quantification on κ is polynomially bounded, because of the conditions of (56), (57), and (58). The quantification of λ is also polynomially bounded, because $T(\lambda)$ and $B(\lambda)$ are polynomially bounded by $|\kappa|$. Therefore, the proof can be formalized by Π^b_2-PIND. From the claim, our lemma is readily proven. Therefore, our proof can be formalized in S^2_2.

We can safely assume that the last judgment is $\langle s_1(u_1, \dots, u_n), \rho \rangle \downarrow w_1$. We use case analysis of the last rule of κ. Because κ contains all judgments of σ, $v(u_i, \rho, \kappa) \trianglelefteq w_i$ for $i = 1, \dots, n$.

If the last rule of κ is either $*$ or Env-rule, the proof is trivial.

If the last rule of κ is v-rule, $s_1(u_1,\dots,u_n) \equiv b_1 \cdots b_l(u_1)$ and u_1 is a numeral. First we remove $\langle s_1, \rho\rangle \downarrow v_1$ from A and use the induction hypothesis. We obtain the computation τ_1. We add the rules

$$\dfrac{\dfrac{}{\langle x, \rho'\rangle \downarrow z_1^*}\ \text{Env}}{\begin{array}{c}\vdots\\ \langle b_1 \cdots b_l(x), \rho'\rangle \downarrow b_1 \cdots b_l z_1^*\end{array}} \tag{60}$$

to τ_1. Let this computation τ.

$$\text{nodes}(\tau) \leq \text{nodes}(\sigma) + 1 + \sum_{d=2}^{k} \text{size}(s_d(\varepsilon,\dots,\varepsilon)) \tag{61}$$

Therefore, we can prove the lemma.

The case in which the last rule of κ is either $0, 1$, ε^m, proj, comp or rec-rule. Then, the computation rule has a form

$$\frac{\vec{\beta} \quad (\langle r_q(u_1,\dots,u_n), \rho\rangle \downarrow p_q)_{q=1,\dots,l}}{\langle f(r_1(u_1,\dots,u_n),\dots,r_l(u_1,\dots,u_n)), \rho\rangle \downarrow w_1}\ \text{pair} \tag{62}$$

where $\vec{\beta}$ is purely numerical, which can be empty. Let κ_1 be a computation obtained by making $(\langle r_q(u_1,\dots,u_n), \rho\rangle \downarrow p_q)_{q=1,\dots,l}$ conclusions. This increases nodes(κ_1) from nodes(κ) at most l. By explicitly counting parentheses and a comma, $\sum_{q=1}^{l} \text{size}(r_q(\varepsilon,\dots,\varepsilon)) + l + 2 \leq \text{size}(s_1(\varepsilon,\dots,\varepsilon))$. Because

$$\text{nodes}(\kappa_1) \tag{63}$$
$$\leq \text{nodes}(\kappa) + l \tag{64}$$
$$\leq U - \sum_{d=1}^{k} \text{size}(s_d(\varepsilon,\dots,\varepsilon)) + l \tag{65}$$
$$\leq U - \sum_{q=1}^{l} \text{size}(r_q(\varepsilon,\dots,\varepsilon)) - l - 2 - \sum_{d=2}^{k} \text{size}(s_d(\varepsilon,\dots,\varepsilon)) + l \tag{66}$$
$$\leq U - \sum_{q=1}^{l} \text{size}(r_q(\varepsilon,\dots,\varepsilon)) - \sum_{d=2}^{k} \text{size}(s_d(\varepsilon,\dots,\varepsilon)) - 2 \tag{67}$$

we can apply the induction hypothesis to κ_1. Therefore, we obtain λ_1 that contains all conclusions of κ plus $(\langle r_q, \rho'\rangle \downarrow p_q)_{q=1,\dots,l}, (\langle s_d, \rho'\rangle \downarrow z_d)_{d=1,\dots,m}$ as conclusions.

By adding one rule to λ_1, we obtain λ.

$$\text{nodes}(\lambda) \tag{68}$$

$$\leq \text{nodes}(\lambda_1) + 1 \tag{69}$$

$$\leq \text{nodes}(\kappa_1) + \sum_{q=1}^{l} \text{size}(r_q(\varepsilon, \dots, \varepsilon)) + \sum_{i=2}^{k} \text{size}(s_d(\varepsilon, \dots, \varepsilon)) + 1 \tag{70}$$

$$\leq \text{nodes}(\kappa) + l + \sum_{q=1}^{l} \text{size}(r_q(\varepsilon, \dots, \varepsilon)) + \sum_{d=2}^{k} \text{size}(s_d(\varepsilon, \dots, \varepsilon)) + 1 \tag{71}$$

$$\leq \text{nodes}(\kappa) + \sum_{d=1}^{k} \text{size}(s_d(\varepsilon, \dots, \varepsilon)) - 1 \tag{72}$$

By construction, λ contains all judgments in κ. □

Lemma 21 (S_2^2, Substitution Lemma II)**.** *Let σ be a computation with conclusions $\langle t_j(x_1, \dots, x_n), \rho[x_1 \mapsto w_1, \dots, x_n \mapsto w_n]\rangle \downarrow v_j$ for $j = 1, \dots, m$. We assume that variables $x_1, \dots, x_n$ are not in the domain of ρ and does not appear in the main terms of conclusions, except $t_1, \dots, t_m$. Assume that $v(u_i, \rho, \sigma) \trianglelefteq w_i$ for $i = 1, \dots, n$. Then, there is a computation τ that has all the conclusions of σ plus $\langle t_j(u_1, \dots, u_n), \rho\rangle \downarrow v'_j$ where $v'_j \trianglelefteq v_j$ for $j = 1, \dots, m$ as conclusions and*

$$\text{nodes}(\tau) \leq \text{nodes}(\sigma) + \sum_{j=1}^{m} \text{size}(t_j(\varepsilon, \dots, \varepsilon)). \tag{73}$$

Further, τ contains all judgments in σ.

Proof. Similar to Lemma 20, let U be an integer larger than $\text{nodes}(\sigma) + \sum_{j=1}^{m} \text{size}(t_j(\varepsilon, \dots, \varepsilon))$. By induction on

$$\sum \{\text{size}(s_d(\varepsilon, \dots, \varepsilon)) \mid \langle s_d(x_1, \dots, x_n), \rho\rangle \downarrow z_d \in A\} \tag{74}$$

and subinduction on nodes(κ), we prove the following induction hypothesis (Claim 2).

Claim 2. *Let $\rho' = \rho[x_1 \mapsto w_1, \dots, x_n \mapsto w_n]$. Let κ be a computation with conclusions*

$$A \equiv \langle s_1(x_1, \dots, x_n), \rho'\rangle \downarrow z'_1, \dots, \langle s_k(x_1, \dots, x_n), \rho'\rangle \downarrow z'_k \tag{75}$$

Assume that κ contains all judgments of σ. We assume that variables $x_1, \dots, x_n$ do not appear in the main terms of conclusions, except $s_1, \dots, s_k$. Further, we assume that κ satisfies

$$\text{nodes}(\kappa) \leq U - \sum_{d=1}^{k} s_d(\varepsilon, \dots, \varepsilon) \tag{76}$$

$$T(\kappa) \leq T(\sigma) \tag{77}$$

$$B(\kappa) \leq B(\sigma). \tag{78}$$

Then, there is a computation λ of which the conclusions are $\langle s_d(u_1, \dots, u_n), \rho\rangle \downarrow z'_d$ where $z'_d \trianglelefteq z_d$ for $d = 1, \dots, k$ and

$$\mathrm{nodes}(\lambda) \leq \mathrm{nodes}(\kappa) + \sum_{d=1}^{k} \mathrm{size}(s_d(\varepsilon, \dots, \varepsilon)). \tag{79}$$

Further, λ contains all judgments in κ.

As Claim 1, the claim is Π^b_2-formula, because first κ is universally quantified and λ is existentially quantified next. Because κ changes through the induction steps, quantification over κ is necessary. The quantification on κ is polynomially bounded, because of the conditions of (76), (77), and (78). The quantification of λ is also polynomially bounded, because $T(\lambda)$ and $B(\lambda)$ are polynomially bounded by $|\kappa|$. Therefore, the proof can be formalized by Π^b_2-PIND. Therefore, our proof can be formalized in S^2_2. From the claim, our lemma is readily proven. Because κ contains all judgments of σ, $v(u_i, \rho, \kappa) \trianglelefteq w_i$.

We can safely assume that the last judgment is $\langle s_1(x_1, \dots, x_n), \rho'\rangle \downarrow z_1$. We use case analysis on the last rule of κ.

If the last rule of κ is either $*$ or v-rule, the proof is trivial.

If the last rule of κ is Env-rule for x_i, we replace it by $\langle u_i, \rho\rangle \downarrow v(u_i, \rho, \kappa)$.

The case in which the last rule of κ is either ε^m, $0, 1$, proj, comp, or rec-rule. Then, the computation rule has the form

$$\frac{\vec{\beta} \quad (\langle r_q, \rho'\rangle \downarrow p_q)_{q=1,\dots,l}}{\langle f(r_1, \dots, r_l), \rho'\rangle \downarrow z_1} \tag{80}$$

where $\vec{\beta}$ is purely numerical, which can be empty. Let κ_1 be the computation obtained by making $(\langle r_q, \rho\rangle \downarrow p_q)_{q=1,\dots,l}$ conclusions by increasing $\mathrm{nodes}(\kappa)$ at most l. We add $(\langle r_q, \rho\rangle \downarrow p_q)_{q=1,\dots,l}$ to A while removing the occurrence of $\langle s_1(x_1, \dots, x_n), \rho'\rangle \downarrow z_1$ from A. By explicitly counting parentheses and a comma,

$$\sum_{d=1}^{l} \mathrm{size}(r_q(\varepsilon, \dots, \varepsilon)) + l + 2 \leq \mathrm{size}(s_1(\varepsilon, \dots, \varepsilon)). \tag{81}$$

Because

$$\begin{aligned}
&\mathrm{nodes}(\kappa_1) && (82)\\
&\leq \mathrm{nodes}(\kappa) + l && (83)\\
&\leq U - \sum_{d=1}^{k} \mathrm{size}(s_d(\varepsilon, \dots, \varepsilon)) + l && (84)\\
&\leq U - \sum_{q=1}^{l} \mathrm{size}(r_q(\varepsilon, \dots, \varepsilon)) - l - 2 - \sum_{d=2}^{k} \mathrm{size}(s_d(\varepsilon, \dots, \varepsilon)) + l && (85)\\
&\leq U - \sum_{q=1}^{l} \mathrm{size}(r_q(\varepsilon, \dots, \varepsilon)) - \sum_{d=2}^{k} \mathrm{size}(s_d(\varepsilon, \dots, \varepsilon)) - 2 && (86)
\end{aligned}$$

Thus, we can apply the induction hypothesis to κ_1. Therefore, we obtain λ_1 of which conclusions are $(\langle r_q(u_1, \dots, u_n), \rho\rangle \downarrow p'_q)_{q=1,\dots,l}, (\langle s_d(u_1, \dots, u_n), \rho\rangle \downarrow z'_d)_{d=2,\dots,k}$, where $p'_q \trianglelefteq p_q$ and $z'_d \trianglelefteq z_d$. By adding one rule to λ_1, we obtain λ

$$\frac{\vec{\beta} \quad (\langle r_q(u_1, \ldots, u_n), \rho\rangle \downarrow p'_q)_{q=1,\ldots,l}}{\langle s_1(u_1, \ldots, u_n), \rho\rangle \downarrow z_1.} \tag{87}$$

nodes(λ) is bounded by

$$\text{nodes}(\lambda_1) + 1 \tag{88}$$

$$\leq \text{nodes}(\kappa_1) + \sum_{q=1}^{l} \text{size}(r_q(\varepsilon, \ldots, \varepsilon)) + \sum_{d=2}^{k} \text{size}(s_d(\varepsilon, \ldots, \varepsilon)) + 1 \tag{89}$$

$$\leq \text{nodes}(\kappa) + l + \sum_{q=1}^{l} \text{size}(r_q(\varepsilon, \ldots, \varepsilon)) + \sum_{d=2}^{m} \text{size}(s_d(\varepsilon, \ldots, \varepsilon)) + 1 \tag{90}$$

$$\leq \text{nodes}(\kappa) + \sum_{d=1}^{m} \text{size}(s_d(\varepsilon, \ldots, \varepsilon)) - 1 \tag{91}$$

By construction, λ contains all judgments in κ. □

Lemma 22 (S^1_2). *Let*

$$f(\vec{u}) = t \tag{92}$$

is a substitution instance of the defining axiom of a function f. If $\vdash_{|B|} \langle f(\vec{u}), \rho\rangle \downarrow v, \vec{\alpha}$, then

$$\vdash_{|B|+\text{size}(f(\vec{u})=t)} \langle t, \rho\rangle \downarrow v', \vec{\alpha} \tag{93}$$

such that $v' \trianglelefteq v$.

Proof. Let σ be a computation sequence that derives $\vdash_{|B|} \langle f(\vec{u}), \rho\rangle \downarrow v, \vec{\alpha}$. The lemma is proven by conducting a case analysis of the inference rule R of $\langle f(\vec{u}), \rho\rangle \downarrow v$ and the defining axioms of f.

If R is $*$-rule, the proof is obvious. Therefore, we assume that R is not a $*$-rule. Then, R is determined by the defining axiom of f.

For the case that $f \equiv \varepsilon$ or $f \equiv 0, 1$, the defining axioms do not exist. Thus, the lemma vacuously holds.

For the case in which $f \equiv \varepsilon^n$, R has the form

$$\frac{(\langle u_i, \rho\rangle \downarrow w_i)_{i=1,\ldots,n}}{\langle \varepsilon^n(u_1, \ldots, u_n), \rho\rangle \downarrow \varepsilon.} \tag{94}$$

Then,

$$\frac{}{\langle \varepsilon, \rho\rangle \downarrow \varepsilon.} \tag{95}$$

The case is valid.

For the case in which $f \equiv \text{proj}^n_i$, R has the form

$$\frac{(\langle u_j, \rho\rangle \downarrow w_j)_{j=1,\ldots,n}}{\langle \text{proj}^n_i(u_1, \ldots, u_n), \rho\rangle \downarrow w^*_i.} \tag{96}$$

Because $w_i \trianglelefteq w^*_i$, the lemma is proved.

For the case in which f is defined by the composition $g(h_1(\overline{x}), \dots, h_n(\overline{x}))$, the inference R of σ that derives $\langle f(u_1, \dots, u_n), \rho \rangle \downarrow v$, has the following form.

$$\frac{\langle g(\vec{y}), \xi \rangle \downarrow v \quad \langle h_1(\vec{x}), \nu \rangle \downarrow w_1 \quad \cdots \quad \langle h_m(\vec{x}), \nu \rangle \downarrow w_m \quad (\langle u_i, \rho \rangle \downarrow z_i)_{i=1,\dots,n}}{\langle f(u_1, \dots, u_n), \rho \rangle \downarrow v} \tag{97}$$

Because $\nu(x_i) = z_i, i = 1, \dots, n$, using Lemma 21 repeatedly, $\tau_1 \vdash \langle g(\vec{y}), \xi \rangle \downarrow v, \langle h_1(\vec{u}), \rho \rangle \downarrow w'_1, \dots, \langle h_m(\vec{u}), \rho \rangle \downarrow w'_m$ where $w'_1 \trianglelefteq w_1, \dots, w'_m \trianglelefteq x_m$ is obtained, in which

$$\mathrm{nodes}(\tau_1) \le \mathrm{nodes}(\sigma) + \sum_{j}^{m} \mathrm{size}(h_j(\vec{\varepsilon})). \tag{98}$$

Again, using Lemma 21, $\tau \vdash \langle g(\vec{h}(\vec{u})), \rho \rangle \downarrow v', \vec{\alpha}$ where $v' \trianglelefteq v$, is obtained where

$$\mathrm{nodes}(\tau) \le \mathrm{nodes}(\tau_1) + \mathrm{size}(g(\vec{\varepsilon})) \tag{99}$$

$$\le \mathrm{nodes}(\sigma) + \mathrm{size}(g(\vec{\varepsilon})) + \sum_{j}^{m} \mathrm{size}(h_j(\vec{\varepsilon})) \tag{100}$$

$$\le \mathrm{nodes}(\sigma) + \mathrm{size}(g(\vec{h}(\vec{u}))) + m \tag{101}$$

$$\le \mathrm{nodes}(\sigma) + \mathrm{size}(f(\vec{u}) = g(\vec{h}(\vec{u}))). \tag{102}$$

The case is valid.

If f is defined by recursion, the inference that derives $\langle f(u_1, \dots, u_n), \rho \rangle \downarrow v$ has the form.

$$\frac{\langle g_\varepsilon(\vec{x}), \nu \rangle \downarrow v \quad \langle \varepsilon, \rho \rangle \downarrow \varepsilon \quad (\langle u_i, \rho \rangle \downarrow z_i)_{i=2,\dots,n}}{\langle f(\varepsilon, u_2, \dots, u_n), \rho \rangle \downarrow v} \tag{103}$$

or for each $b = 0, 1$,

$$\frac{\langle g_b(y, \vec{x}), \nu[y \mapsto w] \rangle \downarrow v \quad \langle f(x_0, \vec{x}), \nu \rangle \downarrow w \quad \langle bu, \rho \rangle \downarrow bz_0 \quad (\langle u_i, \rho \rangle \downarrow z_i)_{i=2,\dots,n}}{\langle f(bu, \vec{u}), \rho \rangle \downarrow v} \tag{104}$$

where $\nu(x_0) = z_0$, while $\nu(x_k) = z_k, k = 2, \dots, n$. First, consider the case of (103). According to Lemma 21 and Lemma 13, we have τ that satisfies $\tau \vdash \langle g(u_1, \dots, u_n), \rho \rangle \downarrow v', \vec{\alpha}$ where $v' \trianglelefteq v$ and $\mathrm{nodes}(\tau) \le \mathrm{nodes}(\sigma) + \mathrm{size}(g(\vec{\varepsilon}))$. Thus, the case has been shown to be valid. Next, consider the case of (104). According to Lemma 14, $\langle u, \rho \rangle \downarrow z'_0, z'_0 \trianglelefteq z_0$ is contained in σ. According to Lemma 21, we have τ_1 that has the conclusion $\langle f(u, \vec{u}), \rho \rangle \downarrow w'$, where $w \trianglelefteq w'$. Because τ_1 contains all the judgments of σ, $\langle g_b(y, \vec{x}), \nu[y \mapsto w] \rangle \downarrow v$, and $\langle u_i, \rho \rangle \downarrow z_i, i = 2, \dots, n$ appear in τ_1. Using Lemma 21 again, we obtain $\tau \vdash \langle g_b(u, f(u, \vec{u}), \vec{u}), \rho \rangle \downarrow v'$, where $v' \trianglelefteq v$.

$$\mathrm{nodes}(\tau) \le \mathrm{nodes}(\tau_1) + \mathrm{size}(g_b(\vec{\varepsilon})) \tag{105}$$

$$\le \mathrm{nodes}(\sigma) + \mathrm{size}(f(\vec{\varepsilon})) + \mathrm{size}(g_b(\vec{\varepsilon})) \tag{106}$$

$$\le \mathrm{nodes}(\sigma) + \mathrm{size}(g_b(u, f(u, \vec{u}), \vec{u})) + 1 \tag{107}$$

The case is valid. □

Lemma 23. *(S^1_2) Let*

$$f(\vec{u}) = t \tag{108}$$

is a substitution instance of the defining axiom of a function f. If $\vdash_{|B|} \langle t, \rho \rangle \downarrow v, \vec{\alpha}$, then

$$\vdash_{|B|+\mathrm{size}(f(\vec{u})=t)} \langle f(\vec{u}), \rho \rangle \downarrow v', \vec{\alpha} \tag{109}$$

such that $v' \trianglelefteq v$.

Proof. Let σ be a computation that derives $\vdash_{|B|} \langle t, \rho \rangle \downarrow v, \vec{\alpha}$. The lemma is proven by conducting a case analysis of the defining axiom of f. Because the case in which $v \equiv *$ is trivial, we assume that $v \not\equiv *$.

In the case in which (108) has the form

$$\varepsilon^n(u_1, \dots, u_m) = \varepsilon, \tag{110}$$

By the computation rule

$$\frac{\langle u_1, \rho \rangle \downarrow * \quad \dots \quad \langle u_m, \rho \rangle \downarrow *}{\langle \varepsilon^n(u_1, \dots, u_m), \rho \rangle \downarrow \varepsilon} \tag{111}$$

$\vdash_{\mathrm{size}(\varepsilon^n(u_1,\dots,u_m)=\varepsilon)} \langle \varepsilon^n(u_1, \dots, u_m), \rho \rangle \downarrow \varepsilon$ holds. By Lemma 14, if $\langle \epsilon, \rho \rangle \downarrow v$, then v is either $*$ or ε. Therefore, the case is valid.

In the case for which (108) has the form

$$\mathrm{proj}^n_i(u_1, \dots, u_m) = u_i, \tag{112}$$

the computation rule

$$\frac{\langle u_1, \rho \rangle \downarrow * \cdots \langle u_{i-1}, \rho \rangle \downarrow * \quad \langle u_i, \rho \rangle \downarrow v \quad \langle u_{i+1}, \rho \rangle \downarrow * \cdots \langle u_m, \rho \rangle \downarrow *}{\langle \mathrm{proj}^n_i(u_1, \dots, u_m), \rho \rangle \downarrow v.} \tag{113}$$

applies. By assumption, $\vdash_{|B|} \langle t_i, \rho \rangle \downarrow v$. Thus, $\vdash_{|B|+m} \langle \mathrm{proj}^n_i(t_1, \dots, t_m), \rho \rangle \downarrow v$. Therefore, the case is valid.

The case for which f is defined by the composition of $g, h_1, \dots, h_m$ is presented next. Let σ be a computation of $\langle g(h_1(\vec{u}), \dots, h_m(\vec{u}))), \rho \rangle \downarrow v$. Then, σ has the following form.

$$\frac{\vec{\beta} \quad \langle h_1(\vec{u}), \rho \rangle \downarrow w_1 \quad \dots \quad \langle h_m(\vec{u}), \rho \rangle \downarrow w_m}{\langle g(h_1(\vec{u}), \dots, h_m(\vec{u})), \rho \rangle \downarrow v} \tag{114}$$

where $\vec{\beta}$ is purely numerical. Let $v_1 = v(u_1, \rho, \sigma), \dots, v_n = v(u_n, \rho, \sigma)$ and $\nu(x_1) = v_1, \dots, \nu(x_n) = v_n$. By repeatedly applying Lemma 20, we obtain a computation τ_1 that has the conclusion $\langle g(h_1(\vec{x}), \dots, h_m(\vec{x})), \nu \rangle \downarrow v$. Because v is not $*$, τ_1 contains the inference

$$\frac{\vec{\gamma} \quad \langle h_1(\vec{x}), \nu \rangle \downarrow w_1 \quad \dots \quad \langle h_m(\vec{x}), \nu \rangle \downarrow w_m}{\langle g(h_1(\vec{x}), \dots, h_m(\vec{x})), \nu \rangle \downarrow v} \tag{115}$$

where $\mathrm{nodes}(\tau_1) \le \mathrm{nodes}(\sigma) + \sum_{j=1}^m \mathrm{size}((h_j(\vec{\varepsilon}))$. By applying Lemma 20, we obtain δ_1 that contains the judgment $\langle g(y_1, \dots, y_m), \xi \rangle \downarrow v$, where $\xi(y_1) = w_1, \dots, \xi(y_m) = w_m$, and satisfies $\mathrm{nodes}(\delta_1) \le \mathrm{nodes}(\tau_1) + \mathrm{size}(g(\vec{\varepsilon}))$. δ_1 contains $\langle h_1(\vec{x}), \nu \rangle \downarrow w_1, \dots, \langle h_m(\vec{x}), \nu \rangle \downarrow w_m$. δ_1 also contains $\langle u_i, \rho \rangle \downarrow v_i$ for $i = 1, \dots, m$ unless v_i is $*$. If v_i is $*$, we add $\langle u_i, \rho \rangle \downarrow *$ to δ_1, increasing $\mathrm{nodes}(\delta_1)$ by at most n. Then, we obtain δ_1 which satisfies $\mathrm{nodes}(\delta_1) \le \mathrm{nodes}(\sigma) + \sum_{j=1}^m \mathrm{size}(h_j(\vec{\varepsilon})) + \mathrm{size}(g(\vec{\varepsilon})) + n$. Using these judgements, we can assemble an inference of the judgement $\langle f(x_1, \dots, x_n), \rho \rangle \downarrow v$.

$$\frac{\langle g(y_1, \dots, y_m), \xi \rangle \downarrow v \quad (\langle h_j(\vec{x}), \nu \rangle \downarrow w_j)_{j=1,\dots,m} \quad (\langle u_i, \rho \rangle \downarrow v_i)_{i=1,\dots,n}}{\langle f(u_1, \dots, u_n), \rho \rangle \downarrow v} \tag{116}$$

Let τ be the computation that is created in this way.

$$\text{nodes}(\tau) \leq \text{nodes}(\sigma) + \sum_{j=1}^{m} \text{size}(h_j(\vec{\varepsilon})) + \text{size}(g(\vec{\varepsilon})) + n + 1 \tag{117}$$

$$\leq \text{nodes}(\sigma) + \text{size}(g(h_1(\vec{\varepsilon}), \dots, h_m(\vec{\varepsilon}))) + m + n + 1 \tag{118}$$

$$\leq \text{nodes}(\sigma) + \text{size}(f(\vec{u}) = t) \tag{119}$$

because

$$\text{size}(g(h_1(\vec{\varepsilon}), \dots, h_m(\vec{\varepsilon}))) \leq \text{size}(t) \tag{120}$$

$$m \leq \text{size}([\text{Fun}, \text{comp}, g, h_1, \dots, h_m]) = \text{size}(f) \tag{121}$$

$$n + 1 \leq \text{size}((u_1, \dots, u_n)). \tag{122}$$

The case for which f is defined by recursion using g_ε, g_0, g_1 is presented next. For the case of g_ε, the proof is similar to that of the case of the composition. Consider the case in which the defining equation is $f(bu_0, \vec{u}) = g_b(u_0, f(u_0, \vec{u}), \vec{u})$. Then, there exists a derivation σ with the value of $g_b(u_0, f(u_0, \vec{u}), \vec{u})$. Let $w_0 = v(u_0, \rho, \sigma), w_i = v(u_i, \rho, \sigma), i = 2, \dots, n$ and $v_0 = v(f(u_0, \vec{u}), \rho, \sigma)$. The environment ξ is defined by $\xi(x_0) = w_0, \xi(x_2) = w_2, \dots, \xi(x_n) = w_n$ and $\xi(y) = v_0$. By Lemma 20, a computation τ with the conclusion $\langle g_b(x_0, y, \vec{x}), \xi \rangle \downarrow v$ is obtained. We can assume that τ contains $\langle f(u_0, \vec{u}), \rho \rangle \downarrow v_0$ as a conclusion by increasing nodes(τ) by one. The environment ν is defined by $\nu(x_0) = w_0, \nu(x_2) = w_2, \dots, \nu(x_n) = w_n$. By Lemma 20, a computation μ with the conclusion $\langle f(x_0, \vec{x}), \nu \rangle \downarrow v_0$ is obtained. μ still contains $\langle g_i(x_0, y, \vec{x}), \xi \rangle \downarrow v$. Using these judgments, we can assemble a computation δ of $\langle f(bu_0, \vec{u}), \rho \rangle \downarrow v$

$$\frac{\langle g(x_0, y, \vec{x}), \xi \rangle \downarrow v \quad \dfrac{\langle u_0, \rho \rangle \downarrow v_0}{\langle bu_0, \rho \rangle \downarrow bv_0} \quad \langle f(x_0, \vec{x}), \nu \rangle \downarrow w_0 \quad (\langle u, \rho \rangle \downarrow w_i)_{i=1,\dots,n}}{\langle f(bu_0, \vec{u}), \rho \rangle \downarrow v} \tag{123}$$

by adding at most $n + 2$ $*$-rules to derive assumptions. By summing up,

$$\text{nodes}(\delta) \leq \text{nodes}(\mu) + n + 2 \tag{124}$$

$$\leq \text{nodes}(\tau) + n + 2 + \text{size}(f(\vec{\varepsilon})) + 1 \tag{125}$$

$$\leq \text{nodes}(\sigma) + n + 3 + \text{size}(f(\vec{\varepsilon})) + \text{size}(g_i(\vec{\varepsilon})) \tag{126}$$

$$\leq \text{nodes}(\sigma) + \text{size}(f(bu_0, \vec{u}) = g_b(u_0, f(u_0, \vec{u}), \vec{u})) \tag{127}$$

because

$$n + 3 \leq \text{size}(f(bu_0, \vec{u})) \tag{128}$$

$$\text{size}(f(\vec{\varepsilon})) + \text{size}(g_b(\vec{\varepsilon})) \leq \text{size}(g_b(u_0, f(u_0, \vec{u}), \vec{u})) + 1. \tag{129}$$

□

7. Consistency proof

This section proves the consistency of PV^- inside S^2_2. To this end, we first prove a type of soundness PV^- by the notion of computation. We prove that, whenever an equation $t = u$ is proved, for each computation σ of t with an environment ρ whose value is v, there is a computation τ of u with the environment ρ whose value is $v', v' \trianglelefteq v$. Further, the proof is carried out in S^2_2. The soundness of our semantics implies consistency because the value of 1 is never 0 by Lemma 15; therefore, it is impossible to derive $0 = 1$. The use of S^2_2, and not S^1_2, is essential because we need to quantify over a computation σ and an environment ρ in the proof of soundness. This introduces two alternate quantifiers in the induction hypothesis.

Proposition 1 (S^2_2)**.** *Let π be a tree-like* PV^-*-proof that derives $t = u$. Then, for any environment ρ for the free variables of t and u and computation $\sigma \vdash \langle t, \rho\rangle \downarrow v$, there is a computation $\tau \vdash \langle u, \rho\rangle \downarrow v'$ such that $v' \trianglelefteq v$,* $\mathrm{nodes}(\tau) \leq \mathrm{nodes}(\sigma) + \mathrm{size}(\pi)$.

Proof. We prove the following claim using induction on a tree-like PV^--proof χ.

Claim 3. *Let U be an integer. Let χ be a tree-like* PV^- *proof that derives $r = s$. Then, for any*

- *environment ρ for free variables of r and s such that $B(\rho) \leq \lfloor \frac{1}{2}(U - \mathrm{size}(\chi))^2 \rfloor$ and $L(\rho) \leq U - \mathrm{size}(\chi)$,*
- *computational judgements $\vec{\alpha} \equiv \alpha_1, \ldots, \alpha_l$ such that $M(\alpha) \leq U - \mathrm{size}(\chi)$, $B(\alpha) \leq \lfloor \frac{1}{2}(U - \mathrm{size}(\chi))^2 \rfloor$ and $L(\alpha) \leq U - \mathrm{size}(\chi)$,*
- *computation $\sigma \vdash \langle r, \rho\rangle \downarrow v, \vec{\alpha}$ such that* $\mathrm{nodes}(\sigma) \leq U - \mathrm{size}(\chi)$,

there is a computation $\tau \vdash \langle s, \rho\rangle \downarrow v', \vec{\alpha}$ such that $v' \trianglelefteq v$, $\mathrm{nodes}(\tau) \leq \mathrm{nodes}(\sigma) + \mathrm{size}(\chi)$.

From the claim, the proposition is immediate by letting U to be sufficiently large. The claim is proven by induction on χ. Because the induction hypothesis can be written by a formula with bounded universal quantifiers and one bounded existential quantifier inside, the induction hypothesis can be written by a Π^b_2-formula with two free variable χ and U. Therefore, the claim can be proven in S^2_2. Use of Π^b_2-PIND is essential because quantification over computations is necessary to interpret the transitivity rule and quantification over environments is necessary to interpret substitution. The proof uses case analysis of the last rule of χ. Because for $\mathrm{size}(\chi) > U$ the claim vacuously holds, we can assume that $\mathrm{size}(\chi) \leq U$.

The case for which the conclusion of χ is a defining axiom is proven using Lemma 22 and 23.

The case for which the axiom is the reflexive axiom is trivial.

The case for which the last inference of χ is a symmetry rule is trivial.

The case for which the last inference of χ is a transitivity rule is considered next.

$$\frac{\begin{array}{cc} \vdots\ \chi_1 & \vdots\ \chi_2 \\ t = u & u = w \end{array}}{t = w} \tag{130}$$

Let σ be a computation a conclusion of which has a $\langle t, \rho\rangle \downarrow v$. By induction hypothesis, there is a computation τ of $\langle u, \rho\rangle \downarrow v'$ such that $\mathrm{nodes}(\tau) \leq \mathrm{nodes}(\sigma) + \mathrm{size}(\chi_1)$ and $v' \trianglelefteq v$. Because $\mathrm{nodes}(\tau) \leq U - \mathrm{size}(\chi) + \mathrm{size}(\chi_1) \leq U - \mathrm{size}(\chi_2)$, by induction hypothesis on χ_2, there is a computation δ of $\langle w, \rho\rangle \downarrow v''$ such that $\mathrm{nodes}(\delta) \leq \mathrm{nodes}(\sigma) + \mathrm{size}(\chi_1) + \mathrm{size}(\chi_2)$ and $v'' \trianglelefteq v$. Thus, the claim holds.

The case in which the last inference of χ is

$$\frac{\begin{array}{ccc} \vdots\ \chi_1 & & \vdots\ \chi_n \\ u_1 = s_1 & \cdots & u_n = s_n \end{array}}{f(u_1, \ldots, u_n) = f(s_1, \ldots, s_n),} \tag{131}$$

is considered. Let σ be a computation of $\langle f(u_1,\dots,u_n),\rho\rangle \downarrow v$. Let $w_1 = v(u_1,\rho,\sigma),\dots,w_n = v(u_n,\rho,\sigma)$. By Lemma 19, increasing nodes(σ) by n, we obtain a computation σ_0 such that $\langle u_1,\rho\rangle \downarrow w_1,\dots,\langle u_n,\rho\rangle \downarrow w_n$ are contained in σ as conclusions. The σ_0 satisfies induction hypothesis on χ_1, because

$$\text{size}(f(u_1,\dots,u_n)) \leq \text{size}(\chi) - \text{size}(\chi_1) \tag{132}$$
$$\leq U - \text{size}(\chi_1) \tag{133}$$
$$\text{nodes}(\sigma) + n \leq U - \text{size}(\chi) + \text{size}(f(u_1,\dots,u_n)) \tag{134}$$
$$\leq U - \text{size}(\chi_1) \tag{135}$$

and $\text{size}(u_i) \leq U - \text{size}(\chi_1)$ for $i = 1,\dots,n$ by the similar reason as (133). Therefore, we can transform σ to σ_1 that has the same conclusions to σ except one of $\langle u_1,\rho\rangle \downarrow w_1$, which is replaced to $\langle s_1,\rho\rangle \downarrow w_1'$ where $w_1' \trianglelefteq w_1$. This increases nodes(σ_1) by size(χ_1). Assume that we construct a computation σ_j that has the same conclusions to σ, except $\langle u_i,\rho\rangle \downarrow w_i, i = 1,\dots,j$, which is replaced to $\langle s_i,\rho\rangle \downarrow w_i'$, where $w_i' \trianglelefteq w_i$ and $\text{nodes}(\sigma_j) \leq \text{nodes}(\sigma) + n + \sum_{i=1}^{j} \text{size}(\chi_i)$. Then $\text{size}(f(u_1,\dots,u_n)) \leq \text{size}(\chi) - \text{size}(\chi_{j+1}) \leq U - \text{size}(\chi_{j+1})$ and $\text{nodes}(\sigma_j) \leq U - \text{size}(\chi) + n + \sum_{i=1}^{j} \text{size}(\chi_i) \leq U - \text{size}(\chi_{j+1})$ hold. Further, $\text{size}(u_i) \leq U - \text{size}(\chi_{j+1})$ for $i = 1,\dots,n$ holds. Therefore, we can apply the induction hypothesis on χ_{j+1} to σ_j and obtain σ_{j+1} which has the same conclusions to σ except $\langle u_i,\rho\rangle \downarrow w_i, i = 1,\dots,j+1$, which is replaced to $\langle s_i,\rho\rangle \downarrow w_i'$ where $w_i' \trianglelefteq w_i$ and $\text{nodes}(\sigma_j) \leq \text{nodes}(\sigma) + n + \sum_{i=1}^{j+1} \text{size}(\chi_i)$. Finally, we obtain a computation σ_n that has the same conclusions to σ, except $\langle u_i,\rho\rangle \downarrow w_i, i = 1,\dots,n$, which is replaced to $\langle s_i,\rho\rangle \downarrow w_i'$, where $w_i' \trianglelefteq w_i$ and $\text{nodes}(\sigma_j) \leq \text{nodes}(\sigma) + n + \sum_{i=1}^{n} \text{size}(\chi_i)$. Let $\rho' = \rho[y_1 \mapsto w_1,\dots,y_n \mapsto w_n]$. Because σ_n has the conclusion $\langle f(u_1,\dots,u_n),\rho\rangle \downarrow v$, by Lemma 20 we obtain a computation τ_1 of $\langle f(y_1,\dots,y_n),\rho'\rangle \downarrow v$. τ_1 contains computation judgements $\langle s_1,\rho\rangle \downarrow w_1',\dots,\langle s_n,\rho\rangle \downarrow w_n'$ and satisfies $\text{nodes}(\tau_1) \leq \text{nodes}(\sigma_n) + \text{size}(f(\varepsilon,\dots,\varepsilon))$. By Lemma 21, we obtain a computation τ of $\langle f(s_1,\dots,s_n),\rho\rangle \downarrow v$.

$$\text{nodes}(\tau) \tag{136}$$
$$\leq \text{nodes}(\sigma) + n + \sum_{i=1}^{n} \text{size}(\chi_i) + 2\text{size}(f(\varepsilon,\dots,\varepsilon)) \tag{137}$$
$$\leq \text{nodes}(\sigma) + \sum_{i=1}^{n} \text{size}(\chi_i) + \text{size}(f(u_1,\dots,u_n) = f(s_1,\dots,s_n)) \tag{138}$$
$$\leq \text{nodes}(\sigma) + \text{size}(\chi). \tag{139}$$

Therefore, the claim holds.

Finally, we consider the substitution rule.

$$\dfrac{\dfrac{\vdots\ \chi_1}{r_0(x) = s_0(x)}}{r_0(q) = s_0(q)} \tag{140}$$

Let σ be a computation of $\langle r_0(q),\rho\rangle \downarrow v$ that satisfies the conditions of the proposition. Let $w = v(q,\rho,\sigma)$ and $\rho' = \rho[x \mapsto w]$. $L(\rho') \leq L(\rho) + 1 \leq U - \text{size}(\chi) + 1 \leq$

$U - \text{size}(\chi_1)$.

$$B(\rho') \leq \max(B(\rho), \text{nodes}(w)) \tag{141}$$
$$\leq \max(B(\rho), \max(B(\sigma), M(\sigma)) + \text{nodes}(\sigma)) \tag{142}$$
$$\leq \max(\lfloor \frac{1}{2}(U - \text{size}(\chi))^2 \rfloor, U - \text{size}(\chi)) + U - \text{size}(\chi) \tag{143}$$
$$\leq \lfloor \frac{1}{2}(U - \text{size}(\chi_1))^2 \rfloor \tag{144}$$

By increasing nodes(σ) by 1, we can assume that σ contains $\langle q, \rho \rangle \downarrow w$ as a conclusion. By Lemma 20, there is a computation σ_1 that derives $\langle r_0(x), \rho' \rangle \downarrow v$ such that $\text{nodes}(\sigma_1) \leq \text{nodes}(\sigma) + 1 + \text{size}(r_0(\varepsilon))$. It is easy to see that σ_1 satisfies assumptions of induction hypothesis for χ_1. Therefore, there is a computation τ_1 of $\langle s_0(x), \rho' \rangle \downarrow v'$ such that $v' \trianglelefteq v$ and $\text{nodes}(\tau_1) \leq \text{nodes}(\sigma_1) + \text{size}(\chi_1)$. Finally, because the conclusion $\langle q, \rho \rangle \downarrow w$ is preserved by all operations above, $v(q, \rho, \tau_1) \trianglelefteq w$ holds. By Lemma 21, there is a computation τ of $\langle s_0(q), \rho \rangle \downarrow v'$ such that $\text{nodes}(\tau) \leq \text{nodes}(\tau_1) + \text{size}(s_0(\varepsilon))$ and $v' \trianglelefteq v$.

$$\text{nodes}(\tau) \leq \text{nodes}(\tau_1) + \text{size}(s_0(\varepsilon)) \tag{145}$$
$$\leq \text{nodes}(\sigma_1) + \text{size}(\chi_1) + \text{size}(s_0(\varepsilon)) \tag{146}$$
$$\leq \text{nodes}(\sigma) + 1 + \text{size}(r_0(\varepsilon)) + \text{size}(\chi_1) + \text{size}(s_0(\varepsilon)) \tag{147}$$
$$\leq \text{nodes}(\sigma) + \text{size}(\chi) \tag{148}$$

Therefore, the claim holds. □

Theorem 1. *S^2_2 proves* $\mathrm{PV}^- \nvdash 0\varepsilon = 1\varepsilon$

Proof. Assume that there is a proof π of $0\varepsilon = 1\varepsilon$ in PV^-. Let σ be a computation of $\langle 0\varepsilon, [\,] \rangle \downarrow 0\varepsilon$. By Proposition 1, there is a computation τ of $\langle 1\varepsilon, [\,] \rangle \downarrow 0\varepsilon$, which contradicts Lemma 15. □

8. Discussion

8.1. **Relation to original PV.** Cook and Urquhart's original PV [6] has some differences from our PV.

Their PV uses lambda abstraction to create new function symbols from terms. Because Cook and Urquhart's PV uses only lambda abstraction for first-order variables, the functions that are defined by lambda abstraction can be defined by compositions, projections, and constant functions.

Another difference is that the intended domain of Cook and Urquhart's PV is the set of natural numbers. Natural numbers are represented by the constant 0 and binary successors s_0, s_1 of which the intended meaning is $2 \cdot x$ and $2 \cdot x + 1$, respectively. On the other hand, our formalism uses the set of binary strings as the intended domain. Our system can interpret natural numbers by using a binary number system, using little endian (the least significant bit appears at the right most position). Then, using our system, we can define all polynomial time functions. However, the schema of limited recursion on notation

$$R[g, h, k](x, \vec{y}) = \text{Cond}(x, g(\vec{y}), Cond(t \dot{-} k(x, \vec{y}), t, k(x, \vec{y}))) \tag{149}$$
$$t \equiv h(x, \vec{y}, R[g, h, k](\lfloor \frac{x}{2} \rfloor, \vec{y}))$$

which appears in Cook and Urquhart's PV would not be derived by our system. This is because to derive (149), it appears that the case analysis on x, which does not seem to be derived from our system, is required.

8.2. **Beckmann's counter-example.** The proof in the previous draft [12] allows a counter-example, which was pointed out by Arnold Beckmann [2]. Let $g(x)$ be the function defined by $g(\varepsilon) = \varepsilon, g(0x) = \overbrace{0 \cdots 0}^{k} g(x), k \geq 1$. $h(x)$ is defined recursively by $h(\varepsilon) = \varepsilon, h(0x) = \varepsilon(x, h(x))$. Then, for any numeral n, we have the PV^--proof of $h(g(0n)) = \varepsilon$, whose length is constant.

$$h(g(0n)) = h(\overbrace{0 \cdots 0}^{k} g(n)) \tag{150}$$

$$= \varepsilon^2(\overbrace{0 \cdots 0}^{k-1} g(n), h(\overbrace{0 \cdots 0}^{k-1} g(n))) \tag{151}$$

$$= \varepsilon \tag{152}$$

However, the computation of $h(g(0n))$, which is defined in [12], becomes

$$\frac{\dots \quad \langle g(0n), \rho \rangle \downarrow v_0}{\langle h(g(0n)), \rho \rangle \downarrow v} \tag{153}$$

and the length of the computation of $g(0n)$ rapidly increases depending on n. Because ε can be computed by a computation with a constant length, this contradicts Proposition 1 of [12].

This indicates that there is a gap in the proof of [12]. Indeed, the computation of $\varepsilon(0 \cdots 0g(n), h(0 \cdots 0g(n)))$ does not have a form such as (92) in the proof of Lemma 14 in [12], because it contains computations neither for $0 \cdots 0g(n)$ nor for $h(0 \cdots 0g(n))$.

In this paper, we reformulate the computation rules such that their forms have greater uniformity. Therefore, to compute $\varepsilon(0 \cdots 0g(n), h(0 \cdots 0g(n)))$, we need to compute $0 \cdots 0g(n)$ and $h(0 \cdots 0g(n))$. Thus, Proposition 1 holds for the equality (151). However, to ensure that Proposition 1 holds for the equality (152), we introduce approximate computations, in which the value can be approximated by $*$. By evaluating $0 \cdots 0g(n)$ and $h(0 \cdots 0g(n))$ to $*$, the number of steps of the computation of $\varepsilon(0 \cdots 0g(n), h(0 \cdots 0g(n)))$ can be bounded by a constant. Then, instead of (153), we use the computation that has a constant size.

(154)

$$\dfrac{\dfrac{\langle x, [x, y \mapsto *] \rangle \downarrow * \quad \langle h(y), [x, y \mapsto *] \rangle \downarrow *}{\langle \varepsilon^2(x, h(y)), [x, y \mapsto *] \rangle \downarrow \varepsilon} \qquad \dfrac{\overline{\overline{\langle 0 \cdots 0x, [x \mapsto *] \rangle \downarrow 0*}} \quad \langle g(n), \rho \rangle \downarrow *}{\langle g(0n), \rho \rangle \downarrow 0*}}{\langle h(g(0n)), \rho \rangle \downarrow \varepsilon}$$

where $[x, y \mapsto *]$ denotes the environment that assigns $*$ to x and y. $[x \mapsto *]$ has a similar meaning. Thus, Proposition 1 holds for (152).

8.3. **Meta-theories.** In this paper, we strengthen the meta-theory from S^1_2, which is claimed to be sufficient in the previous draft [12], to S^2_2. This is because the proof of Lemma 20, 21 and Proposition 1 requires $\Pi^b_2 - \mathrm{PIND}$. The reason for Lemma 20 and 21 is that the conclusions of a computation used for induction step change and their number increases. The reasons for Proposition 1 are the transitivity and substitution rules. To interpret the transitivity rule, the induction hypothesis must hold for all computations with certain conditions. Similarly, to interpret substitution, the induction hypothesis must hold for all environments with certain conditions. Therefore, the induction hypothesis has universal quantifiers in the outmost position. Further, the induction hypothesis claims that for each computation of the term in the left-hand side of the conclusion, there is a computation of the term in the right-hand side of the conclusion. Therefore, induction hypothesis becomes Π^b_2.

8.4. **Relation to result of Buss and Ignjatović.** This paper presents proof that Buss's S^2_2 is capable of proving the consistency of purely equational PV^-, which is obtained by removing induction from PV of Cook and Urquhart but retaining the substitution rule. Because Buss and Ignjatović stated that this is impossible in S^1_2, at first glance, it implies that $S^1_2 \subsetneq S^2_2$. However, this is not the case.

Although they stated that S^1_2 cannot prove the consistency of purely equational PV^-, what they actually prove is that S^1_2 cannot prove the consistency of PV^-, which is extended by propositional logic and $BASIC^e$ axioms. According to them, we can obtain the same unprovability for purely equational PV^- by translating propositional connectives into numerical functions. For example, $t = u$ is translated into $\mathrm{Eq}(t, u)$, where the function Eq is defined as

$$\mathrm{Eq}(t,u) = \begin{cases} 0 & \text{if } t = u \\ 1 & \text{otherwise,} \end{cases} \tag{155}$$

and $p \vee q$ into $p \cdot q$, etc. Then, every proposition p is translated to a numerical term t_p. Then, they assume that whenever a proposition p is proved by PV^-, which is extended with propositional logic and $BASIC^e$, $t_p = 0$ can be proved in purely equational PV^-. However, although such translation is possible in PV [5], it depends on the existence of induction. For example, the reflexive law $x = x$ is translated into $\mathrm{Eq}(x, x) = 0$. It is impossible to derive the latter from the former without using induction. Therefore, we cannot conclude that the consistency of PV^- with propositional logic and $BASIC^e$ axioms from the consistency of purely equational PV^- in S^1_2. Thus, our result does not appear to imply that $S^1_2 \subsetneq S^2_2$.

One possible way to prove that $S^1_2 \subsetneq S_2$ would be to prove the consistency of PV^- with propositional logic and $BASIC^e$ axioms in S_2, which is the system considered by Buss and Ignjatović. However, because our method relies on the fact that PV^- is formulated as an equational theory, our method cannot be extended to PV^- with propositional logic and $BASIC^e$ axioms. Thus, as a long-term goal, it would be interesting to develop a technique to prove the consistency of such a system in bounded arithmetic.

Acknowledgment. The author is grateful to Arnold Beckmann for pointing out a counter-example in the previous draft. The author is also grateful to Toshiyasu Arai, Satoru Kuroda, Jun Inoue, Izumi Takeuti, and Kazushige Terui for insightful discussions and comments. The comments of an anonymous referee in response to the previous submission were also very helpful in improving the paper. We would like to thank Editage (www.editage.jp) for English-language editing. This work was partially supported by the Research Institute for Mathematical Sciences, a Joint Usage/Research Center located in Kyoto University.

References

[1] A. Beckmann, *Proving consistency of equational theories in bounded arithmetic*, Journal of Symbolic Logic **67** (2002), no. 1, 279–296.

[2] ———, Personal Communication, 2015.

[3] S. R. Buss, *Bounded Arithmetic*, Bibliopolis, Naples, 1986.

[4] S. R. Buss and A. Ignjatović, *Unprovability of consistency statements in fragments of bounded arithmetic*, Annals of pure and applied Logic **74** (1995), 221–244.

[5] S. Cook and A. Urquhart, *Functional interpretations of feasibly constructive arithmetic*, Annals of Pure and Applied Logic **63** (1993), no. 2, 103–200.

[6] Stephen A Cook, *Feasibly constructive proofs and the propositional calculus (preliminary version)*, Proceedings of seventh annual ACM symposium on theory of computing, 1975, pp. 83–97.

[7] G. Kahn, *Natural semantics*, Annual symposium on theoretical aspects of computer science, 1987, pp. 22–39.

[8] G. D. Plotkin, *A structural approach to operational semantics*, Computer Science Department, Aarhus University Denmark, 1981.

[9] P. Pudlák, *A note on bounded arithmetic*, Fundamenta mathematicae **136** (1990), 85–89.

[10] G. Takeuti, *Some relations among systems for bounded arithmetic*, Mathematical logic, 1990, pp. 139–154.

[11] A. Wilkie and J. Paris, *On the scheme of induction for bounded arithmetic formulas*, Annals of pure and applied Logic **35** (1987), 261–302.

[12] Y. Yamagata, *Consistency proof of a feasible arithmetic inside a bounded arithmetic*, 2014. arXiv:1411.7087v2.

National Institute of Advanced Science and Technology (AIST), 1-8-31 Midorigaoka, Ikeda, Osaka 563-8577 Japan

URL: `https://staff.aist.go.jp/yoriyuki.yamagata/en/`

E-mail address: `yoriyuki.yamagata@aist.go.jp`