Kazushige Terui

Light Affine Set Theory: A Naive Set Theory of Polynomial Time

Abstract. In [7], a naive set theory is introduced based on a polynomial time logical system, *Light Linear Logic* (**LLL**). Although it is reasonably claimed that the set theory inherits the intrinsically polytime character from the underlying logic **LLL**, the discussion there is largely informal, and a formal justification of the claim is not provided sufficiently. Moreover, the syntax is quite complicated in that it is based on a non-traditional hybrid sequent calculus which is required for formulating **LLL**.

In this paper, we consider a naive set theory based on *Intuitionistic Light Affine Logic* (**ILAL**), a simplification of **LLL** introduced by [1], and call it *Light Affine Set Theory* (**LAST**). The simplicity of **LAST** allows us to rigorously verify its polytime character. In particular, we prove that *a function over* $\{0,1\}^*$ *is computable in polynomial time if and only if it is provably total in* **LAST**.

Keywords: naive set theory, polynomial time, linear logic, light logic, substructural logics.

1. Introduction

In [7], *Light Linear Logic* (**LLL**) is introduced as a subsystem of *Linear Logic* [8] and it is proved that its proofs exactly correspond to the polynomial time functions in the sense of the Curry-Howard correspondence. Furthermore, in the appendix of the same paper, **LLL** is enriched with the *unrestricted comprehension principle* to result in a consistent naive set theory. However, the syntax of the set theory is quite complicated in that it is based on a non-traditional hybrid sequent calculus which is required for formulating **LLL**. Later on, a simplified logical system with the same computational power, called *Intuitionistic Light Affine Logic* (**ILAL**), is introduced by adding the unrestricted weakening rule to the intuitionistic fragment of **LLL**[1]. Although **LLL** and **ILAL** have been well studied (see, e.g., [12, 3, 15, 2, 16]), the set theories associated to them have not been fully explored yet. As a matter of fact, the **LLL**-based set theory is outlined and the fundamental theorem, called the *fixpoint theorem*, is proved in [7], but its main feature, namely expressivity over polytime computation, is discussed only in an informal way.

Presented by **Name of Editor**; *Received* December 1, 2002

Studia Logica **0**: 1–32, 2004.

The aim of this paper is to complement Girard's work and to formally verify the intrinsically polytime character of the set theories based on Light Logics. We consider a naive set theory based on **ILAL** (rather than on **LLL**, as it is much simpler than **LLL**), and call it *Light Affine Set Theory* (**LAST**). The theory is consistent, has a fixpoint for every formula, and is undecidable (Section 2). Natural numbers and arithmetic operations such as addition and multiplication are definable in it, and the *light induction principle* is supported (Section 3). We then show that every polytime function is provably total in it (Section 4). The converse also holds; every provably total function in **LAST** is polytime computable. To show this, we give an interpretation of **LAST** proofs as terms of *Light Affine Lambda Calculus* (λLA, [21]), that is an untyped term calculus based on the ideas of Light Logics and satisfies the polynomial time (strong) normalization theorem. As a consequence, we obtain a complete characterization of polytime: a function is computable in polynomial time if and only if it is provably total in **LAST** (Section 5).

Naive set theories have been investigated in the framework of contraction-free logics (see, e.g., [9, 10, 13, 23, 20]). Although such a theory is descriptively rich (as it numeralwise represents all recursive functions [20]), they are proof-theoretically very weak (as its consistency is established by the induction up to ω).[*] To overcome this weakness, several extensions have been considered (see, e.g., [24, 19, 17]). While **LAST** can be seen as one of such extensions, it has a distinctive feature: it precisely captures a computationally interesting class of functions, the polynomial time functions.

2. Fundamentals of Light Affine Set Theory

We first describe the syntax of **LAST** (in 2.1), then overview its basic notions and properties, such as cut-elimination and its consequences (in 2.2), equality and basic set-theoretic operations (in 2.3), the fixpoint theorem (in 2.4) and the undecidability of **LAST** (in 2.5). We owe most materials below to [7] and [20], except the last undecidability result. All the results of this section hold for the modality-free fragment of **LAST**, i.e., Grishin's naive set theory based on a contraction-free logic too.

[*] In this respect, contraction-free set theories are analogous to Robinson's system Q in arithmetic. Both are descriptively rich, but yet to be extended, either by restricted contraction or by induction schema, to gain computational power.

2.1. Syntax

The following definition comes from the appendix of [7], except that the underlying logic is **ILAL** [1, 2] rather than **LLL**.

Definition 2.1 (Light Affine Set Theory **LAST**). The *terms* and *formulas* of **LAST** are defined simultaneously as follows:

- Term variables $x, y, z, \ldots$ are terms;
- If A is a formula and x is a term variable, then $\{x|A\}$ is a term;
- If t and u are terms, then $t \in u$ is a formula;
- If A and B are formulas, then so are $A \multimap B$, $!A$ and $\S A$;
- If A is a formula and x is a term variable, then $\forall x.A$ is a formula.

We use $t, u, v, \ldots$ to denote terms, $A, B, C, \ldots$ to denote formulas, and $\Gamma, \Delta, \Sigma, \ldots$ to denote multisets of formulas. If Γ stands for $A_1, \ldots, A_n$, then $!\Gamma$ stands for $!A_1, \ldots, !A_n$. $\S\Gamma$ is defined analogously. The notation $!^d A$ stands for $\underbrace{!\cdots!}_{d\text{ times}} A$ and $\S^d A$ for $\underbrace{\S\cdots\S}_{d\text{ times}} A$. A variable x is *bound* in $\{x|A\}$ and $\forall x.A$. Following the standard convention (see [18]), we identify two formulas/terms which differ only in the names of bound variables; e.g., $\forall x.A \equiv \forall y.(A[y/x])$. Notation $u[t/x]$ is used to denote the term which is obtained from u by substituting t for all free occurrences of x. A similar substitution notation is used for formulas.

The *inference rules* of **LAST** are listed in Figure 1.[†]

Other connectives of Linear/Affine Logic, such as $\exists$, $\otimes$ (multiplicative conjunction), $\oplus$ (additive disjunction) and $\mathbf{0}$ (absurdity), can be defined in the spirit of [1]. Fix an arbitrary closed term t_0 (e.g., $\{x|x \in x\}$) and define:

$$\begin{array}{rcl} A \otimes B & \equiv & \forall x.((A \multimap B \multimap t_0 \in x) \multimap t_0 \in x); \\ A \oplus B & \equiv & \forall x.((A \multimap t_0 \in x) \multimap (B \multimap t_0 \in x) \multimap t_0 \in x); \\ \mathbf{0} & \equiv & \forall x.t_0 \in x; \\ \exists y.A & \equiv & \forall x.(\forall y.(A \multimap t_0 \in x) \multimap t_0 \in x), \end{array}$$

where x is a fresh variable which does not occur in A and B.[‡] We further define $A \multimapboth B \equiv (A \multimap B) \otimes (B \multimap A)$, $\neg A \equiv A \multimap \mathbf{0}$ and $t \notin u \equiv \neg(t \in u)$. Note that negation is defined in terms of $\mathbf{0}$ rather than $\perp$. These definitions

[†] $\Gamma, \Delta, \ldots$ are *multisets*. So the exchange rule is implicitly assumed.

[‡] Additive conjunction & and constants $\top$, $\mathbf{1}$ $\perp$ could also be defined. But they are not used in this paper.

Identity and Cut:

$$\frac{}{A \vdash A}\ (Id) \qquad\qquad \frac{\Gamma_1 \vdash A \quad A, \Gamma_2 \vdash C}{\Gamma_1, \Gamma_2 \vdash C}\ (Cut)$$

Structural Rules:

$$\frac{\Gamma \vdash C}{A, \Gamma \vdash C}\ (Weak) \qquad\qquad \frac{!A, !A, \Gamma \vdash C}{!A, \Gamma \vdash C}\ (Contr)$$

Linear Implication:

$$\frac{\Gamma_1 \vdash A \quad B, \Gamma_2 \vdash C}{A \multimap B, \Gamma_1, \Gamma_2 \vdash C}\ (\multimap l) \qquad\qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A \multimap B}\ (\multimap r)$$

Modalities:

$$\frac{B \vdash A}{!B \vdash !A}\ (!),\ B \text{ can be absent.} \qquad\qquad \frac{\Gamma, \Delta \vdash A}{!\Gamma, \S\Delta \vdash \S A}\ (\S)$$

Comprehension:

$$\frac{A[t/x], \Gamma \vdash C}{t \in \{x|A\}, \Gamma \vdash C}\ (\in l) \qquad\qquad \frac{\Gamma \vdash A[t/x]}{\Gamma \vdash t \in \{x|A\}}\ (\in r)$$

Set Quantifiers:

$$\frac{A[t/x], \Gamma \vdash C}{\forall x.A, \Gamma \vdash C}\ (\forall l) \qquad\qquad \frac{\Gamma \vdash A}{\Gamma \vdash \forall x.A}\ (\forall r), x \text{ is not free in } \Gamma$$

Figure 1. Inference Rules of Light Affine Set Theory (**LAST**)

satisfy the basic laws of Linear/Affine Logic:

$$\begin{array}{ll}
A \multimap B \multimap A \otimes B & (A \multimap B \multimap C) \multimap (A \otimes B \multimap C) \\
A \multimap A \oplus B & (A \multimap C) \multimap (B \multimap C) \multimap (A \oplus B \multimap C) \\
A \multimap \neg A \multimap \mathbf{0} & \mathbf{0} \multimap A \\
A[t/x] \multimap \exists x.A & A \multimap C \text{ implies } (\exists x.A) \multimap C \text{ if } x \notin FV(C).
\end{array}$$

As in Linear Logic, contraction is allowed only for !-prefixed formulas. The modality ! is, however, properly weaker than that of Linear Logic in that it does not satisfy *dereliction* (axiom **T** of Modal Logic): $!A \multimap A$, *digging* (axiom **4**): $!A \multimap !!A$ and *monoidalness* (axiom **K**): $!A \otimes !B \multimap !(A \otimes B)$. The dereliction principle, in the presence of unrestricted comprehension, leads to inconsistency; let D be a formula which satisfies $D \multimapboth !\neg D$,[§] then we can derive **0** as follows:

[§] D is essentially Russell's paradox; define $R \equiv \{x|!(x \notin x)\}$ and $D \equiv R \in R$.

$$\dfrac{\dfrac{\dfrac{\dfrac{D \vdash !\neg D \qquad \dfrac{\dfrac{\dfrac{!\neg D \vdash D}{\neg D, !\neg D \vdash \mathbf{0}}}{!\neg D, !\neg D \vdash \mathbf{0}}\,(der)}{!\neg D \vdash \mathbf{0}}}{D \vdash \mathbf{0}}}{\vdash \neg D}}{\vdash !\neg D} \qquad \dfrac{\dfrac{\dfrac{!\neg D \vdash D}{\neg D, !\neg D \vdash \mathbf{0}}}{!\neg D, !\neg D \vdash \mathbf{0}}\,(der)}{!\neg D \vdash \mathbf{0}}}{\vdash \mathbf{0}}$$

The monoidalness principle is consistent but leads to exponential explosion of the cut-elimination procedure.[¶] The status of the digging principle is less clear; although the principle itself is consistent [5], it causes inconsistency in conjunction with *weak dereliction* (axiom **D**): $!A \multimap ?A$ in the classical setting [7]. Since the lack of monoidalness is too severe, an auxiliary modality § is introduced as a compensation. It satisfies *monoidalness*: $\S A \otimes \S B \multimap \S(A \otimes B)$ and *stratified dereliction*: $!A \multimap \S A$. We refer to the introduction of [7] for further detail.

2.2. Some Basic Facts

To begin with, **LAST** enjoys cut-elimination:

THEOREM 2.2 (Cut-Elimination, cf. Girard [7]). *If A is provable in* **LAST**, *then it is cut-free provable in* **LAST**.

A proof will be outlined in Section 5. As a consequence, we have

COROLLARY 2.3 (Consistency of **LAST**). **LAST** *does not prove* **0**.

COROLLARY 2.4.

(1) Disjunction Property: If $A \oplus B$ is provable, then either A or B is provable.

(2) Existence Property: If $\exists x.A$ is provable, then $A[t/x]$ is provable for some term t.

(3) Modality Property: If $!A$ or $\S A$ is provable, then A is provable.

2.3. Equality and Set Theoretic Operations

Next, we must define the notion of equality. Unfortunately, the standard *extensional equality*

$$t =_e u \;\equiv\; \forall x(x \in t \multimapboth x \in u)$$

[¶] If we allow monoidalness for !, we immediately obtain *Elementary Affine Set Theory*, which is of independent interest. In particular, the provably total functions of that theory will be exactly the elementary recursive functions (cf. [7, 6]).

is too weak in **LAST**, as it does not satisfy the basic properties of equality. Alternatively, we adopt the following *Leibniz equality* which is given in [7]:

Definition 2.5 (Leibniz Equality). $t = u \ \equiv\ \forall x(t \in x \multimap u \in x)$.

It should be remarked that Leibniz equality can be considered as an internal representation of syntactic identity $\equiv$:

Proposition 2.6. *$t = u$ is provable in* **LAST** *iff t and u are syntactically identical.*

Proof. **LAST** proves $\forall x(t \in x \multimap u \in x), t \in x \vdash u \in x$. Hence if $t = u$ is provable, then $t \in x \vdash u \in x$ is also provable. By the cut-elimination theorem, it should be an axiom. Hence t and u should be syntactically identical. The other direction is immediate. ■

For example, **LAST** does not prove $\{x|A \oplus B\} = \{x|B \oplus A\}$. The following basic properties are easily verified:

Proposition 2.7. *The following formulas are provable in* **LAST***;*

(1) $t = t$.
(2) $t = u \multimap (A[t/x] \multimap A[u/x])$.
(3) $t = u \multimap u = t$.
(4) $t = u \otimes u = r \multimap t = r$.
(5) $t = u \multimap t = u \otimes t = u$.

Note that statement (5) above means that contraction is freely available for all equational formulas. **LAST** proves $t = u \multimap t =_e u$, but not the converse, as it is inconsistent [10] (see [17] for a detailed argument).

In what follows, we use the following abbreviations: $\forall x \in t.A \equiv \forall x(x \in t \multimap A)$; $\exists x \in t.A \equiv \exists x(x \in t \otimes A)$; $\exists^! x.A \equiv \exists x(A \otimes \forall y(A[y/x] \multimap y = x))$; $\exists^! x \in t.A \equiv \exists x \in t(A \otimes \forall y(A[y/x] \multimap y = x))$.

Let us define some set theoretic operations.

Definition 2.8.

$$\begin{array}{rlrl} \emptyset \equiv & \{x|\mathbf{0}\}; & \{t\} \equiv & \{x|x = t\}; \\ \{t,u\} \equiv & \{x|x = t \oplus x = u\}; & \{t_1,\ldots,t_n\} \equiv & \{x|x = t_1 \oplus \cdots \oplus x = t_n\}; \\ t \cup u \equiv & \{x|x \in t \oplus x \in u\}; & \langle t,u\rangle \equiv & \{\{t\},\{t,u\}\}; \\ \langle t_1,\ldots,t_n\rangle \equiv & \langle \cdots \langle\langle t_1,t_2\rangle,t_3\rangle \cdots, t_n\rangle. & & \end{array}$$

Proposition 2.9. *The following are provable in* **LAST***;*

(1) $t \not\in \emptyset$.

(2) $t \in \{u\} \multimap\!\circ\, t = u$.
(3) $t \in \{u, v\} \multimap\!\circ\, t = u \oplus t = v$.
(4) $\langle t, u\rangle = \langle r, s\rangle \multimap\!\circ\, t = r \otimes u = s$.

Proof. As for (1), derive $t \in \{x|\mathbf{0}\} \vdash \mathbf{0}$ from $\mathbf{0} \vdash \mathbf{0}$ by rule $(\in l)$. (2) and (3) are by definition. The proof of (4) is familiar in the case of the standard axiomatic set theory, and we can repeat just the same argument in **LAST**, since contraction is available for all equational formulas. A complete proof can be found in [20]. ■

2.4. Fixpoint Theorem

One of the most fascinating features of naive set theory is that any formula has a fixpoint:

Theorem 2.10 (Fixpoint Theorem, Girard[7], Shirahata[20]).
(1) For any formula A, there exists a term f such that $t \in f \multimap\!\circ\, A[f/y, t/x]$ is provable for any t.
(2) More generally, for any formula A, there exists a term f such that $\langle t_1, \ldots, t_n\rangle \in f \multimap\!\circ\, A[f/y, t_1/x_1, \ldots, t_n/x_n]$ is provable for any $t_1, \ldots, t_n$.

Proof. As for the first claim, define

$$\begin{array}{rcl} s & \equiv & \{z \mid \exists u \exists v(z = \langle u, v\rangle \otimes A[\{w \mid \langle w, v\rangle \in v\}/y, u/x])\}; \\ f & \equiv & \{w \mid \langle w, s\rangle \in s\}, \end{array}$$

where u, v and w are fresh variables. Then we can derive the desired property. A complete proof can be found in [20]. The second claim is just a generalization of the first. ■

2.5. Undecidability of LAST

Propositional **ILAL** is decidable [22], whereas second order **ILAL** is undecidable [14] (see also [22]). In this section, we prove that **LAST** is also undecidable.

In [20], Shirahata defines a *numeral* $\underline{n}$ for each natural number n by $\underline{0} \equiv \emptyset$; $S(t) \equiv t \cup \{t\}$; $\underline{n} \equiv S^n(\underline{0})$. Then he proves the following fact with the help of the fixpoint theorem:

Theorem 2.11 (Shirahata[20]). *Every total recursive function is numeral-wise representable in (the modality-free fragment of)* **LAST***; i.e., for every k-ary recursive function ϕ, there exists a term f such that*

- *for any* $\vec{n} \in \mathbb{N}^k$, $\phi(\vec{n}) = m$ *implies that* $\vdash \langle \underline{\vec{n}}, \underline{m} \rangle \in f$ *and* $\vdash \forall x(\langle \underline{\vec{n}}, x\rangle \in f \multimap x = \underline{m})$ *are provable.*[‖]

In what follows, we extend this result to weak numeralwise representability of all recursively enumerable predicates. Let N^* be the fixpoint

$$x \in N^* \multimapboth\ x = \underline{0} \oplus \exists y \in N^*(x = S(y)).$$

Then we have

LEMMA 2.12. $\vdash t \in N^*$ *is provable in* **LAST** *if and only if* t *is a numeral* $\underline{n}$.

PROOF. The "if" direction is proved by induction on n. When $n = 0$, we have

$$\cfrac{\cfrac{\vdash \underline{0} = \underline{0}}{\vdash \underline{0} = \underline{0} \oplus \exists y \in N^*(\underline{0} = S(y))}}{\vdash \underline{0} \in N^*} \quad .$$

When $n = m + 1$, by the induction hypothesis, $\vdash \underline{m} \in N^*$ is provable in **LAST**. Therefore we derive:

$$\cfrac{\cfrac{\cfrac{\cfrac{\vdash \underline{m} \in N^* \quad \vdash S(\underline{m}) = S(\underline{m})}{\vdash \underline{m} \in N^* \otimes S(\underline{m}) = S(\underline{m})}}{\vdash \exists y \in N^*(S(\underline{m}) = S(y))}}{\vdash S(\underline{m}) = \underline{0} \oplus \exists y \in N^*(S(\underline{m}) = S(y))}}{\vdash S(\underline{m}) \in N^*} \quad .$$

The "only-if" direction is proved by induction on the size of t (i.e., the number of symbols in t). Suppose that $\vdash t \in N^*$ is provable. Then either $\vdash t = \underline{0}$ or $\vdash \exists y \in N^*(t = S(y))$ is provable by the disjunction property. In the former case, t is syntactically equivalent to $\underline{0}$ by Proposition 2.6. In the latter case, there is some term u such that $\vdash u \in N^*$ and $\vdash t = S(u)$ are provable by the existence property. Thus t is syntactically equivalent to $S(u)$, and hence the induction hypothesis applies to u. It follows that $u \equiv \underline{m}$ for some $m \in \mathbb{N}$. Therefore $t \equiv \underline{m+1}$. ■

[‖] This result is analogous to the numeralwise representability of recursive functions in Robinson's Q, but the proof here is much simpler, because we do not have to elaborate coding of sequences and Gödel's β-function. We just have to write down a recursive definition of each function. Then existence of the corresponding term is automatically assured by the fixpoint theorem. This simplicity is one of the main advantages of having unrestricted comprehension.

THEOREM 2.13 (Weak numeralwise representability of r.e. predicates). *For every k-ary recursively enumerable predicate $\psi \subseteq \mathbb{N}^k$ there exists a (modality-free) formula A of* **LAST** *such that*

$$\langle n_1, \dots, n_k \rangle \in \psi \iff \vdash A[\underline{n_1}/x_1, \dots, \underline{n_k}/x_k]$$

for any $\langle n_1, \dots, n_k \rangle \in \mathbb{N}^k$.

PROOF. There is a total recursive predicate $\xi \subseteq \mathbb{N}^{k+1}$ such that $\langle \vec{n} \rangle \in \psi \iff$ there exists $m \in \mathbb{N}$ such that $\langle \vec{n}, m \rangle \in \xi$. It is an easy consequence of Theorem 2.11 that there is a formula B of **LAST** which (weakly) numeralwise represents ξ. Now,

$$\begin{aligned} \langle \vec{n} \rangle \in \psi &\iff \text{there exists } m \in \mathbb{N} \text{ such that } \langle \vec{n}, m \rangle \in \xi \\ &\iff \text{there exists } m \in \mathbb{N} \text{ such that } \vdash B[\underline{\vec{n}}/\vec{x}, \underline{m}/y] \\ &\iff \vdash \exists y \in N^*.B[\underline{\vec{n}}/\vec{x}]. \end{aligned}$$

The last equivalence follows from the existence property and the previous lemma. ■

Note that the converse holds trivially, since we have a semi-decision procedure for checking if a formula A satisfies $\vdash A[\underline{\vec{n}}/\vec{x}]$. Therefore, the weakly numeralwise representable predicates in (the modality-free fragment of) **LAST** are *exactly* the r.e. predicates.

Since the class of recursively enumerable predicates exceeds the class of recursive (decidable) predicates, we may conclude:

COROLLARY 2.14. **LAST** *is undecidable (and so is the modality-free fragment of* **LAST***).*[**]

3. Arithmetic

In this section, we investigate natural numbers and their properties in **LAST**. We define the set of natural numbers (in 3.1), then show that a restricted form of induction, called *light induction*, is available in **LAST** (in 3.2). Using light induction, we show that addition and multiplication are provably total (in 3.3).

[**] The undecidability of Grishin's set theory was independently proved by Cantini [5].

3.1. Natural Numbers

The previous definition of natural numbers based on unordered pairs is not satisfactory, because it does not yield $\mathsf{S}(x) = \mathsf{S}(y) \multimap x = y$. Alternatively, we define natural numbers based on ordered pairs as in [7]:

Definition 3.1 (Natural Numbers). $\mathsf{0} \equiv \emptyset$; $\mathsf{S}(t) \equiv \langle \emptyset, t \rangle$; $\mathsf{n} \equiv \mathsf{S}^n(\mathsf{0})$.

This definition yields the desired properties:

Proposition 3.2. *The following are provable in* **LAST***:*

(1) $\mathsf{S}(t) \neq \mathsf{0}$.

(2) $\mathsf{S}(t) = \mathsf{S}(u) \multimapboth t = u$.

Proof. (1) Suppose $\mathsf{S}(t) = \mathsf{0}$, which is equivalent to $\langle \emptyset, t \rangle = \emptyset$. But $\{\emptyset\} \in \langle \emptyset, t \rangle$ whereas $\{\emptyset\} \notin \emptyset$, a contradiction.
(2) By Proposition 2.7 (2) and Proposition 2.9 (4). ■

Next, we *internally* define the set of natural numbers in **LAST**:

Definition 3.3. $\mathsf{N} \equiv \{x | \forall \alpha . !\forall y (y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap x \in \alpha)\}$.

The term N surely represents the set of natural numbers in the usual sense:

Proposition 3.4.

(1) $\mathsf{0} \in \mathsf{N}$ *is provable in* **LAST**.

(2) $t \in \mathsf{N} \multimap \mathsf{S}(t) \in \mathsf{N}$ *is provable in* **LAST**.

(3) $t \in \mathsf{N}$ *is provable in* **LAST** *if and only if* $t \equiv \mathsf{n}$ *for some* $n \in \mathbb{N}$.

Proof. 1. As follows.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{0} \in \alpha \vdash \mathsf{0} \in \alpha}{\vdash \mathsf{0} \in \alpha \multimap \mathsf{0} \in \alpha}}{\vdash \S(\mathsf{0} \in \alpha \multimap \mathsf{0} \in \alpha)}}{!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \vdash \S(\mathsf{0} \in \alpha \multimap \mathsf{0} \in \alpha)}}{\vdash !\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap \mathsf{0} \in \alpha)}}{\vdash \forall \alpha . !\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap \mathsf{0} \in \alpha)}}{\vdash \mathsf{0} \in \mathsf{N}}$$

2. As follows.

$$
\begin{prooftree}
\AxiomC{$\mathsf{0} \in \alpha \vdash \mathsf{0} \in \alpha$}
\AxiomC{$t \in \alpha \vdash t \in \alpha$}
\AxiomC{$\mathsf{S}(t) \in \alpha \vdash \mathsf{S}(t) \in \alpha$}
\BinaryInfC{$t \in \alpha \multimap \mathsf{S}(t) \in \alpha, t \in \alpha \vdash \mathsf{S}(t) \in \alpha$}
\UnaryInfC{$\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha), t \in \alpha \vdash \mathsf{S}(t) \in \alpha$}
\BinaryInfC{$\mathsf{0} \in \alpha, \forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha), \mathsf{0} \in \alpha \multimap t \in \alpha \vdash \mathsf{S}(t) \in \alpha$}
\UnaryInfC{$\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha), \mathsf{0} \in \alpha \multimap t \in \alpha \vdash \mathsf{0} \in \alpha \multimap \mathsf{S}(t) \in \alpha$}
\UnaryInfC{$!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha), \S(\mathsf{0} \in \alpha \multimap t \in \alpha) \vdash \S(\mathsf{0} \in \alpha \multimap \mathsf{S}(t) \in \alpha)$}
\doubleLine
\UnaryInfC{$!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha)^2, !\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap t \in \alpha) \vdash \S(\mathsf{0} \in \alpha \multimap \mathsf{S}(t) \in \alpha)$}
\doubleLine
\UnaryInfC{$!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap t \in \alpha) \vdash !\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap \mathsf{S}(t) \in \alpha)$}
\doubleLine
\UnaryInfC{$\forall\alpha.!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap t \in \alpha) \vdash \forall\alpha.!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap \mathsf{S}(t) \in \alpha)$}
\doubleLine
\UnaryInfC{$t \in \mathsf{N} \vdash \mathsf{S}(t) \in \mathsf{N}$}
\end{prooftree}
$$

3. The "if" direction follows from 1 and 2 above. As for the "only-if" direction, observe that the last part of the cut-free proof of $t \in \mathsf{N}$ must be of the following form:

$$
\begin{prooftree}
\AxiomC{$\vdots$}
\noLine
\UnaryInfC{$\mathsf{0} \in \alpha, \forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha)^n \vdash t \in \alpha$}
\UnaryInfC{$\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha)^n \vdash \mathsf{0} \in \alpha \multimap t \in \alpha$}
\doubleLine
\UnaryInfC{$!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \vdash \S(\mathsf{0} \in \alpha \multimap t \in \alpha)$}
\UnaryInfC{$\vdash !\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap t \in \alpha)$}
\UnaryInfC{$\vdash \forall\alpha.!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap t \in \alpha)$}
\UnaryInfC{$\vdash t \in \mathsf{N}$}
\end{prooftree}
$$

for some $n \geq \mathsf{0}$. From this, we conclude that $t \equiv \mathsf{m}$ for some $m \leq n$. ■

3.2. Light Induction

With N defined above, a certain restricted form of induction is available:

PROPOSITION 3.5 (Light Induction). *The following inference rule is derivable in* **LAST***:*

$$
\dfrac{\Gamma \vdash A[\mathsf{0}/x] \quad B, A[y/x] \vdash A[\mathsf{S}(y)/x]}{\S\Gamma, !B, t \in \mathsf{N} \vdash \S A[t/x]} \quad ,
$$

where y does not occur in A and B, and B may be absent.

PROOF. See the following derivation.

$$
\begin{prooftree}
\AxiomC{$B, A[y/x] \vdash A[\mathsf{S}(y)/x]$}
\doubleLine
\UnaryInfC{$B, y \in \{x|A\} \vdash \mathsf{S}(y) \in \{x|A\}$}
\doubleLine
\UnaryInfC{$B \vdash \forall y(y \in \{x|A\} \multimap \mathsf{S}(y) \in \{x|A\})$}
\UnaryInfC{$!B \vdash !\forall y(y \in \{x|A\} \multimap \mathsf{S}(y) \in \{x|A\})$}
\AxiomC{$\Gamma \vdash A[\mathsf{0}/x]$}
\UnaryInfC{$\Gamma \vdash \mathsf{0} \in \{x|A\}$}
\AxiomC{$A[t/x] \vdash A[t/x]$}
\UnaryInfC{$t \in \{x|A\} \vdash A[t/x]$}
\BinaryInfC{$\Gamma, \mathsf{0} \in \{x|A\} \multimap t \in \{x|A\} \vdash A[t/x]$}
\UnaryInfC{$\S\Gamma, \S(\mathsf{0} \in \{x|A\} \multimap t \in \{x|A\}) \vdash \S A[t/x]$}
\BinaryInfC{$\S\Gamma, !B, !\forall y(y \in \{x|A\} \multimap \mathsf{S}(y) \in \{x|A\}) \multimap \S(\mathsf{0} \in \{x|A\} \multimap t \in \{x|A\}) \vdash \S A[t/x]$}
\UnaryInfC{$\S\Gamma, !B, \forall\alpha.!\forall y(y \in \alpha \multimap \mathsf{S}(y) \in \alpha) \multimap \S(\mathsf{0} \in \alpha \multimap t \in \alpha) \vdash \S A[t/x]$}
\UnaryInfC{$\S\Gamma, !B, t \in \mathsf{N} \vdash \S A[t/x]$}
\end{prooftree}
$$

■

In what follows, we are particularly interested in those sequents of the form $\vec{u} \in \mathsf{N} \vdash \S^p A$ $(p \geq 0)$, where $\vec{u} \in \mathsf{N}$ stands for a sequence of the form $u_1 \in \mathsf{N}, \ldots, u_n \in \mathsf{N}$. For such sequents, the following useful principles are available:

PROPOSITION 3.6.

(1) Coercion: $t \in \mathsf{N} \multimap \S^p !^q t \in \mathsf{N}$ *is provable for any* $p \geq 1$ *and* $q \geq 0$.

(2) N*-Contraction: The following inference rule is derivable in* **LAST***:*

$$\frac{t \in \mathsf{N}, t \in \mathsf{N}, \vec{u} \in \mathsf{N} \vdash \S^p A}{t \in \mathsf{N}, \vec{u} \in \mathsf{N} \vdash \S^{p+1} A} \textit{ for any } p \geq 0.$$

PROOF. (1) For any $p \geq 1$ and $q \geq 0$, we have $\vdash \S^{p-1} !^q \mathsf{0} \in \mathsf{N}$ and $\S^{p-1} !^q x \in \mathsf{N} \vdash \S^{p-1} !^q \mathsf{S}(x) \in \mathsf{N}$. Hence the desired formula is obtained by light induction.

(2) We have $\vdash \mathsf{0} \in \mathsf{N} \otimes \mathsf{0} \in \mathsf{N}$ and $x \in \mathsf{N} \otimes x \in \mathsf{N} \vdash \mathsf{S}(x) \in \mathsf{N} \otimes \mathsf{S}(x) \in \mathsf{N}$. Hence by light induction, it holds that $t \in \mathsf{N} \vdash \S(t \in \mathsf{N} \otimes t \in \mathsf{N})$. On the other hand, we have

$$\S(t \in \mathsf{N} \otimes t \in \mathsf{N}), \S\vec{u} \in \mathsf{N} \vdash \S^{p+1} A$$

by assumption and rule $(\S)$. Hence the desired sequent is obtained by (Cut) and coercion. ■

3.3. Addition and Multiplication

The graphs of addition and multiplication are defined by fixpoint:

DEFINITION 3.7. Let add be a term which satisfies

$$\begin{aligned}\langle x, y, z\rangle \in \mathsf{add} \quad \multimap\!\!\circ \quad & (y = \mathsf{0} \otimes x = z) \oplus \\ & \exists y' \exists z' (y = \mathsf{S}(y') \otimes z = \mathsf{S}(z') \otimes \langle x, y', z'\rangle \in \mathsf{add}).\end{aligned}$$

Such a term exists by the fixpoint theorem. Similarly, let mult be a term which satisfies

$$\begin{aligned}\langle x, y, z\rangle \in \mathsf{mult} \quad \multimap\!\!\circ \quad & (y = \mathsf{0} \otimes z = \mathsf{0}) \oplus \\ & \exists y' \exists z' (y = \mathsf{S}(y') \otimes \langle z', x, z\rangle \in \mathsf{add} \otimes \langle x, y', z'\rangle \in \mathsf{mult}).\end{aligned}$$

LEMMA 3.8. *The following are provable in* **LAST***:*

(1) $\langle x, \mathsf{0}, z\rangle \in \mathsf{add} \multimap\!\!\circ\, x = z$.

(2) $\langle x, \mathsf{S}(y), z\rangle \in \mathsf{add} \multimap\!\!\circ\, \exists z'(z = \mathsf{S}(z') \otimes \langle x, y, z'\rangle \in \mathsf{add})$.

(3) $\langle x, \mathsf{0}, z\rangle \in \mathsf{mult} \multimap\!\!\circ\, z = \mathsf{0}$.

(4) $\langle x, \mathsf{S}(y), z\rangle \in \mathsf{mult} \multimapboth \exists z'(\langle z', x, z\rangle \in \mathsf{add} \otimes \langle x, y, z'\rangle \in \mathsf{mult})$.

Proof. Argue within **LAST**.
(1) Assume $x = z$. Then we have $\mathsf{0} = \mathsf{0} \otimes x = z$, hence $\langle x, \mathsf{0}, z\rangle \in \mathsf{add}$. Conversely, assume $\langle x, \mathsf{0}, z\rangle \in \mathsf{add}$. Then either $\mathsf{0} = \mathsf{0} \otimes x = z$ or $\exists y' \exists z'(\mathsf{0} = \mathsf{S}(y') \otimes z = \mathsf{S}(z') \otimes \langle x, y', z'\rangle \in \mathsf{add})$. But the latter is impossible, as $\mathsf{0} = \mathsf{S}(y')$ implies a contradiction by Proposition 3.2 (1). Therefore, we have $x = z$.
(2) Assume that $z = \mathsf{S}(z') \otimes \langle x, y, z'\rangle \in \mathsf{add}$ for some z'. Since we have $\mathsf{S}(y) = \mathsf{S}(y)$, it follows that $\exists y' \exists z'(\mathsf{S}(y) = \mathsf{S}(y') \otimes z = \mathsf{S}(z') \otimes \langle x, y', z'\rangle \in \mathsf{add})$. Therefore, $\langle x, \mathsf{S}(y), z\rangle \in \mathsf{add}$. Conversely, assume $\langle x, \mathsf{S}(y), z\rangle \in \mathsf{add}$. Then either $\mathsf{S}(y) = \mathsf{0} \otimes x = z$ or $\exists y' \exists z'(\mathsf{S}(y) = \mathsf{S}(y') \otimes z = \mathsf{S}(z') \otimes \langle x, y', z'\rangle \in \mathsf{add})$. Since the former is impossible, we have $\mathsf{S}(y) = \mathsf{S}(y') \otimes z = \mathsf{S}(z') \otimes \langle x, y', z'\rangle \in \mathsf{add}$ for some y' and z'. By Proposition 3.2 (2), it follows that $y = y'$. Thus, $z = \mathsf{S}(z') \otimes \langle x, y, z'\rangle \in \mathsf{add}$. Hence we have $\exists z'(z = \mathsf{S}(z') \otimes \langle x, y, z'\rangle \in \mathsf{add})$, as required.

It is easy to formalize the above proofs in **LAST**. (3) and (4) can be shown similarly. ■

The following two propositions show that addition and multiplication are provably total in **LAST**.

Proposition 3.9.
(1) $\langle \mathsf{n}, \mathsf{m}, \mathsf{k}\rangle \in \mathsf{add}$ *is provable in* **LAST** *if* $n + m = k$.
(2) $\langle \mathsf{n}, \mathsf{m}, \mathsf{k}\rangle \in \mathsf{mult}$ *is provable in* **LAST** *if* $n \cdot m = k$.

Proof. Both are proved by (external) induction on m, using Lemma 3.8. ■

Proposition 3.10. *The following are provable in* **LAST***:*
(1) $\forall x \in \mathsf{N}.\forall y \in \mathsf{N}.\S\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{add})$.
(2) $\forall x \in \mathsf{N}.\forall y \in \mathsf{N}.\S^3\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{mult})$.

Proof. (1) We prove
(i) $\vdash \forall x \in \mathsf{N}.\exists^! z \in \mathsf{N}(\langle x, \mathsf{0}, z\rangle \in \mathsf{add})$ and
(ii) $\forall x \in \mathsf{N}.\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{add}) \vdash \forall x \in \mathsf{N}.\exists^! z \in \mathsf{N}(\langle x, \mathsf{S}(y), z\rangle \in \mathsf{add})$.
It then follows by light induction that

$$y \in \mathsf{N} \vdash \S(\forall x \in \mathsf{N}.\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{add})). \qquad (*)$$

By an easy manipulation, we derive $\S(x \in \mathsf{N}), y \in \mathsf{N} \vdash \S\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{add})$, hence by coercion $x \in \mathsf{N}, y \in \mathsf{N} \vdash \S\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{add})$, as required.

Now let us show (i). By Lemma 3.8(1), $\langle x, \mathsf{0}, x\rangle \in \mathsf{add} \otimes \forall y(\langle x, \mathsf{0}, y\rangle \in \mathsf{add} \multimap y = x)$ is provable. From this, $x \in \mathsf{N} \vdash \exists^! z \in \mathsf{N}(\langle x, \mathsf{0}, z\rangle \in \mathsf{add})$, i.e., (i) is provable.

As for (ii), argue within **LAST**. Assume that $\forall x \in \mathsf{N}.\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{add})$ and let $x \in \mathsf{N}$. Then there is a unique $z \in \mathsf{N}$ such that $\langle x, y, z\rangle \in \mathsf{add}$. By Proposition 3.4 (2), $\mathsf{S}(z) \in \mathsf{N}$, and by Lemma 3.8 (2),

$$\langle x, \mathsf{S}(y), \mathsf{S}(z)\rangle \in \mathsf{add}. \tag{$**$}$$

It remains to show that $\mathsf{S}(z)$ is the unique element satisfying $(**)$. So assume that $\langle x, \mathsf{S}(y), w\rangle \in \mathsf{add}$. Then by Lemma 3.8 (2), there exists w' such that $w = \mathsf{S}(w')$ and $\langle x, y, w'\rangle \in \mathsf{add}$. By the uniqueness of z, we have $w' = z$. Therefore $w = \mathsf{S}(z)$ as required. It is easy to check that this informal proof can be formalized in **LAST**.

(2) We prove

(iii) $\vdash \S\exists^! z \in \mathsf{N}(\langle x, 0, z\rangle \in \mathsf{mult})$ and

(iv) $x \in \mathsf{N}, \S\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{mult}) \vdash \S\exists^! w \in \mathsf{N}(\langle x, \mathsf{S}(y), w\rangle \in \mathsf{mult})$.

It then follows, by light induction, that $!x \in \mathsf{N}, y \in \mathsf{N} \vdash \S^2\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{mult})$. Hence by coercion, we derive the desired formula.

As for (iii), observe that $\vdash 0 \in \mathsf{N} \otimes \langle x, 0, 0\rangle \in \mathsf{mult} \otimes \forall w(\langle x, 0, w\rangle \in \mathsf{mult} \multimap w = 0)$ is provable by Proposition 3.4 (1) and Lemma 3.8 (3).

As for (iv), first show

$$\langle x, y, z\rangle \in \mathsf{mult}, \langle z, x, w\rangle \in \mathsf{add} \vdash \langle x, \mathsf{S}(y), w\rangle \in \mathsf{mult}$$

and

$$\forall z'(\langle x, y, z'\rangle \in \mathsf{mult} \multimap z' = z), \forall w'(\langle z, x, w'\rangle \in \mathsf{add} \multimap w' = w) \\ \vdash \forall w'(\langle x, \mathsf{S}(y), w'\rangle \in \mathsf{mult} \multimap w' = w)$$

by using Lemma 3.8 (4). From these two, we derive

$$\exists^! z \in \mathsf{N}(\langle x, y, z\rangle \in \mathsf{mult}), \forall z \in \mathsf{N}.\exists^! w \in \mathsf{N}(\langle z, x, w\rangle \in \mathsf{add}) \\ \vdash \exists^! w \in \mathsf{N}(\langle x, \mathsf{S}(y), w\rangle \in \mathsf{mult}).$$

Therefore, we obtain the desired sequent by rule (§) and sequent $(*)$ above (with variables renamed suitably). ∎

4. Representing Sets and Functions

We first clarify the notion of representability (in 4.1), then demonstrate that various sets and functions are representable in **LAST** (in 4.2 – 4.5). We finally show that every polytime function is representable, i.e., provably total (in 4.6).

4.1. Representation in LAST

Let us make precise what it means to represent sets and functions in **LAST**.

DEFINITION 4.1.

(1) A set $\mathcal{T}$ is *represented by* a term t of **LAST** if there is a bijection $(\cdot)^*$ from $\mathcal{T}$ to the set of terms u such that $\vdash u \in t$ is provable in **LAST**.

(2) A function $\phi : \vec{\mathcal{T}} \longrightarrow \mathcal{U}$ is *represented by* a term f *with domains* $\vec{t}$ *and range* u if

- $\vec{\mathcal{T}}$ and $\mathcal{U}$ are represented by $\vec{t}$ and u respectively;
- For any $\vec{m} \in \vec{\mathcal{T}}$ and $n \in \mathcal{U}$ such that $\phi(\vec{m}) = n$, $\vdash \langle \vec{m}^*, n^* \rangle \in f$ is provable in **LAST**;
- $\vdash \forall \vec{x} \in t.\S^d(\exists^! y \in u.\langle \vec{x}, y\rangle \in f)$ is provable in **LAST** for some $d \geq 0$, where $\forall \vec{x} \in t.A$ stands for $\forall x_1 \in t \dots \forall x_n \in t.A$.

In particular, we say that ϕ is *flatly represented by* f in case $d = 0$. A representable function is also said to be *provably total in* **LAST**, following the standard terminology.

Note that the notion of representability in the above sense is stronger than that of numeralwise representability studied in Section 2.5, as it requires the totality of a function to be *internally* provable in the formal system **LAST**. We have already seen that the term N represents the set $\mathbb{N}$ of natural numbers (Proposition 3.4) and plus and mult represent addition and multiplication of natural numbers (Propositions 3.9 and 3.10). It is also clear that the term $\{x | \exists y(x = \langle y, \mathsf{S}(y)\rangle)\}$ represents the successor function. Further examples of representations are provided below.

The uniqueness property incorporated by the third condition of Definition 4.1(2) implies the following:

PROPOSITION 4.2. *Suppose that a function $\phi : \vec{\mathcal{T}} \longrightarrow \mathcal{U}$ be represented by a term f with domains $\vec{t}$ and range u. Let $\vec{m} \in \vec{\mathcal{T}}$. Then $\langle \vec{m}^*, v\rangle \in f$ is provable in* **LAST** *iff $v \equiv n^*$, where $n = \phi(\vec{m})$.*

PROOF. The "if" direction holds by definition. To show the "only-if" direction, observe that $\vdash \S^d\exists^! y \in u(\langle \vec{m}^*, y\rangle \in f)$ is provable in **LAST** for any $\vec{m} \in \vec{\mathcal{T}}$. By Corollary 2.4, it follows that

$$\forall z(\langle \vec{m}^*, z\rangle \in f \multimap z = v_0) \qquad (\dagger)$$

is provable for some v_0. Now suppose that $\langle \vec{m}^*, v\rangle \in f$ be provable. Then we have $v = v_0$ by $(\dagger)$. On the other hand, we also have $\langle \vec{m}^*, n^*\rangle \in f$ by definition, where $n = \phi(\vec{m})$. Hence it holds that $n^* = v_0$ by $(\dagger)$. Hence $v = v_0 = n^*$ is provable in **LAST**. Therefore $v \equiv n^*$ by Proposition 2.6. ■

Definition 4.1 gives our intended notion of representability. The main theorems (Theorem 4.17, Corollary 5.12) will be stated in terms of this. It is, however, problematic in that representable functions in this sense are not necessarily composable[††]. For this reason, we introduce a slightly stronger notion of representability.

DEFINITION 4.3. A function $\phi : \vec{\mathcal{T}} \longrightarrow \mathcal{U}$ is *§-represented by* a term f *with domains* $\vec{t}$ *and range* u if it is represented so in the sense of Definition 4.1 (2) with the third condition strengthened as follows: There exists $d \geq 0$ such that for any $e \geq 0$,

$$\vdash \forall \vec{x} \in t.\S^d \exists y \in u.\S^e(\langle \vec{x}, y\rangle \in f \otimes \forall z(\langle \vec{x}, z\rangle \in f \multimap z = y))$$

is provable in **LAST**. We say that ϕ is *flatly* §-represented by f in case $d = 0$.

Note that §-representability implies the original representability (the latter corresponds to the case $e = 0$). It is not hard to see that addition, multiplication and successor are §-representable as well.

4.2. Finite Sets

A finite set $\mathbb{Q}_n$ with n elements is represented by the term $\mathsf{Q}_n \equiv \{\mathsf{0}, \dots, \mathsf{n-1}\}$ with the obvious bijection. In particular, the set $\mathbb{Q}_2$ of boolean values is represented by Q_2. It is easy to prove

- Flat Contraction: $t \in \mathsf{Q}_n \multimap (t \in \mathsf{Q}_n \otimes t \in \mathsf{Q}_n)$;
- Coercion: $t \in \mathsf{Q}_n \multimap \S^d !^e t \in \mathsf{Q}_n$ for any $d \geq 1$ and $e \geq 0$.

Moreover, we have

PROPOSITION 4.4. *Every finite function* $\phi : \mathbb{Q}_{n_1} \times \cdots \times \mathbb{Q}_{n_k} \longrightarrow \mathbb{Q}_m$ *is flatly §-representable with domains* $\mathsf{Q}_{n_1}, \dots, \mathsf{Q}_{n_k}$ *and range* Q_m.

PROOF. Instead of giving a detailed proof, we just describe an example, which should be sufficient for convincing ourselves. We show that boolean conjunction $\wedge : \mathbb{Q}_2 \times \mathbb{Q}_2 \longrightarrow \mathbb{Q}_2$ is flatly §-represented by

$$\mathsf{conj} \equiv \{x | (x = \langle \mathsf{0}, \mathsf{0}, \mathsf{0}\rangle) \oplus (x = \langle \mathsf{0}, \mathsf{1}, \mathsf{0}\rangle) \oplus (x = \langle \mathsf{1}, \mathsf{0}, \mathsf{0}\rangle) \oplus (x = \langle \mathsf{1}, \mathsf{1}, \mathsf{1}\rangle)\}.$$

The first and the second conditions for §-representability are obvious. As for the third condition, prove $\vdash \mathsf{0} \in \mathsf{Q}_2 \otimes \S^e(\langle \mathsf{0}, \mathsf{0}, \mathsf{0}\rangle \in \mathsf{conj} \otimes \forall z(\langle \mathsf{0}, \mathsf{0}, z\rangle \in \mathsf{conj} \multimap z = \mathsf{0}))$ for any $e \geq 0$, from which it follows that $x = \mathsf{0}, y = \mathsf{0} \vdash$

[††] Pointed out by Daniel de Carvalho.

$\exists z \in \mathsf{Q}_2.\S^e(\langle x,y,z\rangle \in \mathsf{conj} \otimes \forall z'(\langle x,y,z'\rangle \in \mathsf{conj} \multimap z' = z))$. Similarly, we can prove:

$$x = \mathsf{0}, y = \mathsf{1} \vdash \exists z \in \mathsf{Q}_2.\S^e(\langle x,y,z\rangle \in \mathsf{conj} \otimes \forall z'(\langle x,y,z'\rangle \in \mathsf{conj} \multimap z' = z)),$$
$$x = \mathsf{1}, y = \mathsf{0} \vdash \exists z \in \mathsf{Q}_2.\S^e(\langle x,y,z\rangle \in \mathsf{conj} \otimes \forall z'(\langle x,y,z'\rangle \in \mathsf{conj} \multimap z' = z)),$$
$$x = \mathsf{1}, y = \mathsf{1} \vdash \exists z \in \mathsf{Q}_2.\S^e(\langle x,y,z\rangle \in \mathsf{conj} \otimes \forall z'(\langle x,y,z'\rangle \in \mathsf{conj} \multimap z' = z)).$$

Therefore, it holds that $x \in \mathsf{Q}_2, y \in \mathsf{Q}_2 \vdash \exists z \in \mathsf{Q}_2.\S^e(\langle x,y,z\rangle \in \mathsf{conj} \otimes \forall z'(\langle x,y,z'\rangle \in \mathsf{conj} \multimap z' = z))$, by noting that $x \in \mathsf{Q}_2 \multimapboth x = \mathsf{0} \oplus x = \mathsf{1}$. ■

As a generalization, we also have

PROPOSITION 4.5. *Given a function $\psi_i : \vec{\mathcal{T}} \longrightarrow \mathcal{U}$ for each $0 \leq i \leq n-1$, one defines a new function $\phi : \mathbb{Q}_n \times \vec{\mathcal{T}} \longrightarrow \mathcal{U}$ by $\phi(i, \vec{v}) = \psi_i(\vec{v})$.*

Suppose that $\vec{\mathcal{T}}$ and $\mathcal{U}$ are represented by terms $\vec{t}$ and u, and ψ_i be flatly $\S$-represented by term h_i with domains $\vec{t}$ and range u ($0 \leq i \leq n-1$). Then the above ϕ is flatly $\S$-represented by the following term f with domains $\mathsf{Q}_n, \vec{t}$ and range u:

$$f \equiv \{x | (\exists w \in h_0.w = \langle \vec{y}, z\rangle \otimes x = \langle \mathsf{0}, \vec{y}, z\rangle) \oplus \\ \cdots \oplus (\exists w \in h_{n-1}.w = \langle \vec{y}, z\rangle \otimes x = \langle \mathsf{n-1}, \vec{y}, z\rangle).$$

4.3. Words over Finite Alphabets 1

Let us consider the set $\mathbb{W}_n$ of words over a finite alphabet $\mathbb{Q}_n$. First of all, define

$$\begin{array}{rcl} \epsilon & \equiv & \emptyset; \\ \mathsf{S}_i(t) & \equiv & \langle \mathsf{i}, t\rangle, \text{ for each } 0 \leq i < n. \end{array}$$

Then we can naturally associate a term w to each word $w \in \mathbb{W}_n$. For example we associate to $010 \in \mathbb{W}_2$ the term $\mathsf{S}_0(\mathsf{S}_1(\mathsf{S}_0(\epsilon)))$.

There are two ways to represent the set $\mathbb{W}_n$, both of which are useful. The first is a generalization of N. Define W_2 by

$$\mathsf{W}_2 \equiv \{x | \forall \alpha.!\forall y(y \in \alpha \multimap \mathsf{S}_0(y) \in \alpha) \multimap \\ !\forall y(y \in \alpha \multimap \mathsf{S}_1(y) \in \alpha) \multimap \S(\epsilon \in \alpha \multimap x \in \alpha)\}.$$

More generally, define for each n

$$\mathsf{W}_n \equiv \{x | \forall \alpha.!\forall y(y \in \alpha \multimap \mathsf{S}_0(y) \in \alpha) \multimap \\ \cdots !\forall y(y \in \alpha \multimap \mathsf{S}_{n-1}(y) \in \alpha) \multimap \S(\epsilon \in \alpha \multimap x \in \alpha)\}.$$

In what follows, $\mathbb{W}_2$ and W_2 are also written as $\mathbb{W}$ and W. The second representation of $\mathbb{W}_n$ will be given in the next subsection. Similarly to Proposition 3.4, we have

PROPOSITION 4.6.

(1) $\epsilon \in \mathsf{W}_n$ *is provable in* **LAST**.

(2) $t \in \mathsf{W}_n \multimap \mathsf{S}_i(t) \in \mathsf{W}_n$ *is provable in* **LAST** *for each* $0 \leq i < n$.

(3) $t \in \mathsf{W}_n$ *is provable in* **LAST** *if and only if* t *is of the form* w *for some* $w \in \mathbb{W}_n$.

Analogously to N, W_n admits the light induction principle:

$$\frac{\Gamma \vdash A[\epsilon/x] \quad B_0, A[y/x] \vdash A[\mathsf{S}_0(y)/x] \quad \cdots \quad B_{n-1}, A[y/x] \vdash A[\mathsf{S}_{n-1}(y)/x]}{\S\Gamma, !B_0, \ldots, !B_{n-1}, t \in \mathsf{W}_n \vdash \S A[t/x]}$$

where y does not occur in A and $B_0, \ldots, B_{n-1}$. Hence it also admits coercion and contraction (cf. Proposition 3.6):

- Coercion: $t \in \mathsf{W}_n \multimap \S^p !^q t \in \mathsf{W}_n$ is provable for any $p \geq 1$ and $q \geq 0$.
- W_n-Contraction: The following inference rule is derivable in **LAST**:

$$\frac{t \in \mathsf{W}_n, t \in \mathsf{W}_n, \vec{u} \in \mathsf{W}_n \vdash \S^p A}{t \in \mathsf{W}_n, \vec{u} \in \mathsf{W}_n \vdash \S^{p+1} A} \text{ for any } p \geq 0.$$

PROPOSITION 4.7.

(1) Let $n \leq m$. *Then* $t \in \mathsf{W}_n \vdash t \in \mathsf{W}_m$ *is provable in* **LAST**. *Therefore the inclusion map* $\iota : \mathbb{W}_n \longrightarrow \mathbb{W}_m$ *is flatly* $\S$*-representable with domain* W_n *and range* W_m.

(2) The length map $|\bullet|\ : \mathbb{W}_n \longrightarrow \mathbb{N}$ *is* $\S$*-representable with domain* W_n *and range* N.

PROOF. (1) Proving $t \in \mathsf{W}_n \vdash t \in \mathsf{W}_m$ is easy. The term $\{x | \exists y. x = \langle y, y \rangle\}$ represents the inclusion map. (2) We consider the case $n = 2$. Define the term len by:

$$\begin{aligned}\langle x, y \rangle \in \mathsf{len} \multimapboth\ & (x = \epsilon \otimes y = 0) \oplus \\ & \exists x' \exists y' ((x = \mathsf{S}_0(x') \oplus x = \mathsf{S}_1(x')) \otimes y = \mathsf{S}(y') \otimes \langle x', y' \rangle \in \mathsf{len}).\end{aligned}$$

Then, using light induction for W_2, we can show that it represents the length map. ■

4.4. Words over Finite Alphabets 2

The second representation of $\mathbb{W}_n$ is given by fixpoint:

$$x \in \mathsf{W}'_n \multimapboth x = \epsilon \oplus \exists x' \in \mathsf{W}'_n (x = \mathsf{S}_0(x') \oplus \cdots \oplus x = \mathsf{S}_{n-1}(x')).$$

Then similarly to Lemma 2.12, we have

Proposition 4.8.

(1) $\epsilon \in \mathsf{W}'_n$ *is provable in* **LAST**.

(2) $t \in \mathsf{W}'_n \multimap \mathsf{S}_i(t) \in \mathsf{W}'_n$ *is provable in* **LAST** *for each* $0 \leq i < n$.

(3) $t \in \mathsf{W}'_n$ *is provable in* **LAST** *if and only if* t *is of the form* w *for some* $w \in \mathbb{W}_n$.

W'_n does not admit induction. Instead, it provides a flat discriminator function:

Proposition 4.9. *Given functions* $\psi_\epsilon : \vec{\mathcal{T}} \longrightarrow \mathcal{U}$ *and* $\psi_i : \mathbb{W}_n \times \vec{\mathcal{T}} \longrightarrow \mathcal{U}$ *for* $0 \leq i \leq n-1$, *one defines a new function* $\phi : \mathbb{W}_n \times \vec{\mathcal{T}} \longrightarrow \mathcal{U}$, *called* discriminator, *by* $\phi(\epsilon, \vec{v}) = \psi_\epsilon(\vec{v})$; $\phi(i \cdot w, \vec{v}) = \psi_i(w, \vec{v})$.

Suppose that $\vec{\mathcal{T}}$ *and* $\mathcal{U}$ *are represented by terms* $\vec{t}$ *and* u, *and* ψ_ϵ *and* ψ_i*'s be flatly §-represented by* h_ϵ *and* h_i*'s. Then the above* ϕ *is flatly §-representable with domains* $\mathsf{W}'_n, \vec{t}$ *and range* u.

Proof. For simplicity, we consider the case $n = 2$ and assume $\vec{\mathcal{T}} \equiv \mathcal{T}$. Furthermore, we only consider the weaker form of representability here. Define the term f by

$$\begin{aligned}\langle x, y, z\rangle \in f \multimapboth{}& (x = \epsilon \otimes \langle y, z\rangle \in h_\epsilon) \oplus \\ & \exists x'((x = \mathsf{S}_0(x') \otimes \langle x', y, z\rangle \in h_0) \oplus (x = \mathsf{S}_1(x') \otimes \langle x', y, z\rangle \in h_1)).\end{aligned}$$

By assumption, $y \in t \vdash \exists^! z \langle y, z\rangle \in h_\epsilon$, from which we obtain

$$x = \epsilon, y \in t \vdash \exists^! z(\langle x, y, z\rangle \in f). \tag{i}$$

And also, $x' \in \mathsf{W}'_2, y \in t \vdash \exists^! z(\langle x', y, z\rangle \in h_0)$ by assumption, from which we obtain

$$x' \in \mathsf{W}'_2 \otimes x = \mathsf{S}_0(x'), y \in t \vdash \exists^! z(\langle x, y, z\rangle \in f). \tag{ii}$$

Similarly we have

$$x' \in \mathsf{W}'_2 \otimes x = \mathsf{S}_1(x'), y \in t \vdash \exists^! z(\langle x, y, z\rangle \in f). \tag{iii}$$

By (i), (ii) and (iii), $x \in \mathsf{W}'_2, y \in t \vdash \exists^! z(\langle x, y, z\rangle \in f)$ is provable as required. ■

As an instance of the above proposition, we have:

PROPOSITION 4.10. *Predecessor for $\mathbb{W}_n$, defined by $\rho(\epsilon) \equiv \epsilon$; $\rho(i \cdot w) \equiv w$ for $0 \leq i < n$, is flatly §-representable with domain W'_n and range W'_n.*

The following proposition shows, in some sense, that W_n is a "subset" of W'_n.

PROPOSITION 4.11. *$t \in \mathsf{W}_n \vdash \S(t \in \mathsf{W}'_n)$ is provable in* **LAST**. *Therefore, the identity map $\iota : \mathbb{W}_n \longrightarrow \mathbb{W}_n$ is §-representable with domain W_n and range W'_n.*

PROOF. We have $\vdash \epsilon \in \mathsf{W}'_n$ and $x \in \mathsf{W}'_n \vdash \mathsf{S}_i(x) \in \mathsf{W}'_n$ for each $0 \leq i < n$ by Proposition 4.8. Therefore, by light induction for W_n, we obtain $t \in \mathsf{W}_n \vdash \S(t \in \mathsf{W}'_n)$. ■

The converse cannot be proved, since W'_n does not admit induction. As a compensation, we have the following semi-identity map:

PROPOSITION 4.12. *There is a term tau such that $\langle \mathsf{n}, \mathsf{w}, \mathsf{w} \rangle \in \mathsf{tau}$ is provable whenever $n \geq |w|$, and*

$$\forall x \in \mathsf{N}.\S\forall y \in \mathsf{W}'_n.\exists z \in \mathsf{W}_n.\S^e(\langle x, y, z\rangle \in \mathsf{tau} \otimes \forall z'(\langle x, y, z'\rangle \in \mathsf{tau} \multimap z' = z))$$

is provable for any $e \geq 0$.

The term tau can be obtained from the function $\tau : \mathbb{N} \times \mathbb{W}_n \longrightarrow \mathbb{W}_n$, defined by $\tau(0, w) = \epsilon$; $\tau(n+1, \epsilon) = \epsilon$; $\tau(n+1, i \cdot w) = i \cdot \tau(n, w)$. The proof of Proposition 4.12 is left to the reader.

4.5. Cartesian Product, Composition and Iteration

If two sets $\mathcal{T}$ and $\mathcal{U}$ are represented by terms t and u respectively, then their Cartesian product $\mathcal{T} \times \mathcal{U}$ is represented by the term $t \times u \equiv \{x | \exists y \exists z(x = \langle y, z\rangle \otimes y \in t \otimes z \in u)\}$, with the associated bijection $\langle x, y\rangle^* = \langle x^*, y^*\rangle$.

PROPOSITION 4.13. *Let $\phi_i : \mathcal{T}_i \longrightarrow \mathcal{U}_i$ be flatly §-represented by term f_i with domains t_i and range u_i ($i = 1, 2$). Then their product $\phi_1 \times \phi_2 : \mathcal{T}_1 \times \mathcal{T}_2 \longrightarrow \mathcal{U}_1 \times \mathcal{U}_2$ is flatly §-represented by the following term $f_1 \times f_2$ with domain $t_1 \times t_2$ and range $u_1 \times u_2$:*

$$f_1 \times f_2 \equiv \{x | \exists y_1 y_2 z_1 z_2(x = \langle\langle y_1, y_2\rangle, \langle z_1, z_2\rangle\rangle \otimes \langle y_1, z_1\rangle \in f_1 \otimes \langle y_2, z_2\rangle \in f_2)\}.$$

Proposition 4.14. *Suppose that $\vec{\mathcal{T}}$, $\vec{\mathcal{T}'}$, $\mathcal{U}$ and $\mathcal{V}$ are represented by terms $\vec{t}$, $\vec{t'}$, u and v, where terms $\vec{t}$ admit coercion. Suppose also that function $\psi : \vec{\mathcal{T}} \times \mathcal{U} \longrightarrow \mathcal{V}$ is §-represented by term g with domain $\vec{t}, u$ and range v, and that function $\xi : \vec{\mathcal{T}'} \longrightarrow \mathcal{U}$ is §-represented by term h with domains $\vec{t'}$ and range u. Then their composition $\psi \circ \xi(\vec{x}, \vec{y}) = \psi(\vec{x}, \xi(\vec{y}))$, where $\vec{x}$ and $\vec{y}$ are disjoint, is §-represented by the following term $g \circ h$ with domain $\vec{t}, \vec{t'}$ and range v:*

$$g \circ h \;\equiv\; \{x' | \exists \vec{x}\vec{y}zw(x' = \langle \vec{x}, \vec{y}, z\rangle \otimes \langle \vec{y}, w\rangle \in h \otimes \langle \vec{x}, w, z\rangle \in g)\}.$$

Proof. It is easy to check that the term $g \circ h$ satisfies the first and the second conditions for §-representability in Definition 4.3. To show the third one, let $e \geq 0$. Then by definition, we have

$$\vec{x} \in \vec{t} \vdash \forall w \in u.§^d \exists z \in v.§^e(\langle \vec{x}, w, z\rangle \in g \otimes \forall z'(\langle \vec{x}, w, z'\rangle \in g \multimap z' = z)), \qquad \text{(iv)}$$

$$\vec{y} \in \vec{t'} \vdash §^{d'} \exists w \in u.§^{d+e}(\langle \vec{y}, w\rangle \in h \otimes \forall w'(\langle \vec{y}, w'\rangle \in h \multimap w' = w)) \qquad \text{(v)}$$

for some $d, d' \geq 0$. Let

$$\begin{array}{rcl} G & \equiv & \langle \vec{x}, w, z\rangle \in g \otimes \forall z'(\langle \vec{x}, w, z'\rangle \in g \multimap z' = z), \\ H & \equiv & \langle \vec{y}, w\rangle \in h \otimes \forall w'(\langle \vec{y}, w'\rangle \in h \multimap w' = w). \end{array}$$

Then, by applying (§) and coercion to (iv), we obtain $\vec{x} \in \vec{t} \vdash §^{d'}\forall w \in u.§^d \exists z \in v.§^e G$, while we can rewrite (v) as $\vec{y} \in \vec{t'} \vdash §^{d'} \exists w \in u.§^{d+e} H$.

From these two, we successively derive:

$$\vec{x} \in \vec{t}, \vec{y} \in \vec{t'} \vdash §^{d'} \exists w.(§^d(\exists z \in v.§^e G) \otimes §^{d+e} H),$$
$$\vec{x} \in \vec{t}, \vec{y} \in \vec{t'} \vdash §^{d'} \exists w.§^d \exists z \in v.§^e(G \otimes H),$$
$$\vec{x} \in \vec{t}, \vec{y} \in \vec{t'} \vdash §^{d+d'} \exists z \in v.§^e \exists w(G \otimes H).$$

In the meantime, it can be shown that

$$\exists w(G \otimes H) \vdash \langle \vec{x}, \vec{y}, z\rangle \in g \circ h \otimes \forall z'(\langle \vec{x}, \vec{y}, z'\rangle \in g \circ h \multimap z' = z),$$

from which the third condition for §-representability follows easily. ■

Iteration is supported by **LAST**, but there is a crucial limitation that only *flatly* representable functions with *at most one* auxiliary argument can be iterated.

Proposition 4.15. *Suppose that $\vec{\mathcal{T}}$, $\mathcal{U}$ and $\mathcal{V}$ are represented by terms $\vec{t}$, u and v, where terms $\vec{t}$ admit coercion. If $\psi : \vec{\mathcal{T}} \longrightarrow \mathcal{U}$ is §-represented* *by g and $\xi : \mathcal{U} \times \mathcal{V} \longrightarrow \mathcal{U}$ is* flatly *§-represented by h, then the function $\phi : \mathbb{N} \times \vec{\mathcal{T}} \times \mathcal{V} \longrightarrow \mathcal{U}$ defined by*

$$\begin{array}{rcl} \phi(0, \vec{y}, z) & = & \psi(\vec{y}); \\ \phi(n+1, \vec{y}, z) & = & \xi(\phi(n, \vec{y}, z), z), \end{array}$$

is §-represented by

$$\begin{array}{lcl} \langle x, \vec{y}, z, w \rangle \in f & \multimap\!\circ & (x = \mathsf{0} \otimes \langle \vec{y}, w \rangle \in g) \oplus \\ & & \exists x'w'(x = \mathsf{S}(x') \otimes \langle x', \vec{y}, z, w' \rangle \in f \otimes \langle w', z, w \rangle \in h). \end{array}$$

We do not give a (rather tedious) proof here, since it is just a generalization of the proof of Proposition 3.10 (2).

4.6. Encoding Turing Machines

Now let us encode Turing machines in **LAST**. Our treatment of Turing machines below is essentially the same as in [7, 2]. We refer to [11] for the general background on Turing machines.

Let M be a single-tape Turing machine over the alphabet $\Sigma = \{0, 1, b\}$ (where b is for blank) and with the states $Q = \{q_0, \ldots, q_{n-1}\}$ (where q_0 is the initial state). M is endowed with a transition function

$$\delta : \Sigma \times Q \longrightarrow \Sigma \times Q \times \{L, R, C\},$$

here L stands for left, R for right, and C for no-move. M has an infinite tape, which has infinitely many cells in both directions. A *step* of M consists of reading one symbol from the cell which the head is scanning, writing a symbol on the same cell, moving the head at most one tape cell, and entering a new state, in accordance with the transition function δ.

A *configuration* of M consists of a triple $\langle q, w_1, w_2 \rangle \in Q \times \Sigma^* \times \Sigma^*$; q denotes the current state, w_1 describes the non-blank part of the tape to the left of the head, w_2 describes the non-blank part of the tape to the right of the head. By convention, w_1 is written in the reverse order, and w_2 includes the content of the cell currently scanned.

We say that M *with input w yields w' after n steps* when, starting from the *initial configuration* $\langle q_0, \epsilon, w \rangle$, M arrives at a configuration $\langle q, w_1, w' \rangle$ for some q, w_1 after n steps, and moreover $w' \in \mathbb{W}$. If $w' \notin \mathbb{W}$, M yields nothing.

Definition 4.16. A function $\phi : \mathbb{W} \longrightarrow \mathbb{W}$ is a *polynomial time function* if there is a Turing machine M and a monotone polynomial p such that for any $w \in \mathbb{W}$, M with input w yields $\phi(w)$ after $p(|w|)$ steps.

Now we claim:

Theorem 4.17. *Every polynomial time function is provably total in* **LAST** *with domain* W *and range* W.

Proof. All the requisite materials are given in the previous subsections. It just suffices to put them together. Let ϕ be a polynomial time function. Then there is a Turing machine M and a polynomial p by Definition 4.16.

- The set $Conf \equiv Q \times \Sigma^* \times \Sigma^*$ of configurations is represented by the term $\mathsf{Conf} \equiv \mathsf{Q}_n \times \mathsf{W}'_3 \times \mathsf{W}'_3$.
- The function $\hat{\delta} : Conf \longrightarrow Conf$ for the one-step transition is obtained by combining successors and predecessor for Σ^*, a case function for Q and a discriminator for Σ^*. Note that all of these are *flatly* §-representable in **LAST** by Propositions 4.8, 4.10, 4.5 and 4.9. Hence $\hat{\delta}$ is also flatly §-representable with domain Conf and range Conf by Proposition 4.13. (Here it is crucial to use W'_3 rather than W_3.)
- By iteration, a function $\phi_M : \mathbb{N} \times Conf \longrightarrow Conf$ is defined so that the value of $\phi_M(n, c)$ is the configuration of M after n steps starting from the configuration c. By Proposition 4.15, this ϕ_M can be §-represented by a term of **LAST** with domain $\mathsf{N} \times \mathsf{Conf}$ and range Conf.
- On the other hand, any monotone polynomial can be composed of addition and multiplication, and thus can be §-represented by a **LAST** term with domain N and range N (with the help of composition and N-Contraction). By composing it with the length map $|\bullet| : \mathbb{W} \longrightarrow \mathbb{N}$ in Proposition 4.7 (2), we obtain a term §-representing the function $p(|w|)$ with domain W and range N.
- The initializing function which transforms an input $w \in \mathbb{W}$ to an initial configuration is §-represented by a term with domain W and range Conf by using Proposition 4.7 (1) and Proposition 4.11. The output function from $Conf$ to $\mathbb{W}$ can be similarly §-represented with domain Conf and range W'.
- Therefore, we obtain a term §-representing ϕ with domain W and range W'; the input is used twice, once for initialization and once for the time bound $p(|w|)$, but it does not matter since we have W-Contraction.
- Although the above ϕ is §-represented with range W', we may apply $\tau(n, w)$ in Proposition 4.12 to change the range into W, by taking a sufficiently large n; such an n is always provided, since the length of the output is obviously bounded by $p(|w|)$.

- Finally, recall that §-representability implies representability, thus ϕ is provably total in **LAST**.

■

5. Light Affine Lambda Calculus and Interpretation of Proofs

In the preceding sections, we have shown that every polytime function is provably total in **LAST**. In this section, we show the converse, i.e., that any provably total function in **LAST** is a polytime function. To achieve this, we interpret proofs of **LAST** as terms of *Light Affine Lambda Calculus* (λLA, [21]), which are known to be polytime (strongly) normalizable.

We recall λLA and its main properties in 5.1, then give an interpretation of proofs in 5.2. Finally we show how to extract a program from a proof of a totality statement in 5.3.

5.1. Light Affine Lambda Calculus

Here we recall only the requisite materials about λLA. See [21] for a full exposition.

Definition 5.1. Let $\mathtt{x}, \mathtt{y}, \mathtt{z} \ldots$ range over variables. The set $\mathcal{PT}$ of *pseudo-terms* is defined by the following grammar:

$$\mathtt{M}, \mathtt{N} ::= \mathtt{x} \mid \lambda\mathtt{x.M} \mid \mathtt{MN} \mid \mathtt{!M} \mid \mathtt{let\ N\ be\ !x\ in\ M} \mid \S\mathtt{M} \mid \mathtt{let\ N\ be\ }\S\mathtt{x\ in\ M}.$$

In the sequel, symbol † stands for either ! or §. Pseudo-terms ($\lambda\mathtt{x.M}$) and ($\mathtt{let\ N\ be}\ \dagger\ \mathtt{x\ in\ M}$) *bind* each occurrence of $\mathtt{x}$ in $\mathtt{M}$. As usual, pseudo-terms are considered up to α-equivalence, and subject to the *variable convention* (see [4]); namely, the bound variables are chosen to be different from the free variables, so that variable clash is never caused by substitution. The notation $\mathtt{M[N/x]}$ denotes the pseudo-term obtained by substituting $\mathtt{N}$ for the free occurrences of $\mathtt{x}$ in $\mathtt{M}$. $FV(\mathtt{M})$ denotes the set of free variables in $\mathtt{M}$. $FO(\mathtt{x}, \mathtt{M})$ denotes the number of free occurrences of $\mathtt{x}$ in $\mathtt{M}$ and $FO(\mathtt{M})$ denotes the number of all free variable occurrences in $\mathtt{M}$.

The *size* $|\mathtt{M}|$ of a pseudo-term $\mathtt{M}$ is the number of symbols occurring in it. Given $\mathtt{M}$ and its subexpression $\mathtt{N}$, the *depth* of $\mathtt{N}$ in $\mathtt{M}$ is the number of modal operators $!, \S$ enclosing it. The *depth* of $\mathtt{M}$ is the maximum of the depths of all its subexpressions.

In order to define well-formedness, it is convenient to distinguish three kinds of variables: *undischarged*, !-*discharged*, and §-*discharged* variables.

These are to be bound by λ-abstraction, let-! operator and let-§ operator, respectively. (Below, $\uplus$ stands for the disjoint union.)

DEFINITION 5.2. Let X, Y, Z range over the finite sets of variables. The 4-ary relation $\mathtt{M} \in \mathcal{T}_{X,Y,Z}$ (saying that $\mathtt{M}$ is a well-formed term with undischarged variables X, !-discharged variables Y and §-discharged variables Z) is defined as follows (in writing $\mathtt{M} \in \mathcal{T}_{X,Y,Z}$, we implicitly assume that X, Y and Z are mutually disjoint):

(1) $\mathtt{x} \in \mathcal{T}_{X,Y,Z} \iff \mathtt{x} \in X$.
(2) $\lambda\mathtt{x}.\mathtt{M} \in \mathcal{T}_{X,Y,Z} \iff \mathtt{M} \in \mathcal{T}_{X\uplus\{\mathtt{x}\},Y,Z},\ FO(\mathtt{x},\mathtt{M}) \leq 1$.
(3) $\mathtt{MN} \in \mathcal{T}_{X,Y,Z} \iff \mathtt{M} \in \mathcal{T}_{X,Y,Z},\ \mathtt{N} \in \mathcal{T}_{X,Y,Z}$.
(4) $!\mathtt{M} \in \mathcal{T}_{X,Y,Z} \iff \mathtt{M} \in \mathcal{T}_{Y,\emptyset,\emptyset},\ FO(\mathtt{M}) \leq 1$.
(5) $\S\mathtt{M} \in \mathcal{T}_{X,Y,Z} \iff \mathtt{M} \in \mathcal{T}_{Y\uplus Z,\emptyset,\emptyset}$.
(6) $\mathtt{let\ M\ be\ !x\ in\ N} \in \mathcal{T}_{X,Y,Z} \iff \mathtt{M} \in \mathcal{T}_{X,Y,Z},\ \mathtt{N} \in \mathcal{T}_{X,Y\uplus\{\mathtt{x}\},Z}$.
(7) $\mathtt{let\ M\ be\ \S x\ in\ N} \in \mathcal{T}_{X,Y,Z}$
$\iff \mathtt{M} \in \mathcal{T}_{X,Y,Z}, \mathtt{N} \in \mathcal{T}_{X,Y,Z\uplus\{\mathtt{x}\}},\ FO(\mathtt{x},\mathtt{N}) \leq 1$.

We say that $\mathtt{M}$ is a *well-formed term*, or simply a *term*, if $\mathtt{M} \in \mathcal{T}_{X,Y,Z}$ for some X, Y and Z. The set of terms is denoted by $\mathcal{T}$.

The *reduction rules* of λLA are the following:

(β) $(\lambda\mathtt{x}.\mathtt{M})\mathtt{N} \longrightarrow \mathtt{M[N/x]}$;
(§) $\mathtt{let\ \S N\ be\ \S x\ in\ M} \longrightarrow \mathtt{M[N/x]}$;
(!) $\mathtt{let\ !N\ be\ !x\ in\ M} \longrightarrow \mathtt{M[N/x]}$;
(*com*1) $\mathtt{(let\ N\ be\ \dagger x\ in\ M)L} \longrightarrow \mathtt{let\ N\ be\ \dagger x\ in\ (ML)}$;
(*com*2) $\mathtt{let\ (let\ N\ be\ \dagger x\ in\ M)\ be\ \dagger y\ in\ L}$
$\longrightarrow \mathtt{let\ N\ be\ \dagger x\ in\ (let\ M\ be\ \dagger y\ in\ L)}$.

The reduction relation $\longrightarrow$ is defined as usual. The reflexive-transitive closure of $\longrightarrow$ is denoted by $\longrightarrow^*$.

The following are the main properties of λLA:

THEOREM 5.3 (Polytime strong normalization). *Any term* $\mathtt{M}$ *of depth* d *reduces to a normal form within* $O(|\mathtt{M}|^{2^{d+1}})$ *reduction steps (and within time* $O(|\mathtt{M}|^{2^{d+2}})$ *on Turing machines). This result holds independently of which reduction strategy we take.*

THEOREM 5.4 (Church-Rosser property). *If* $\mathtt{M}_0$ *is a term and* $\mathtt{M}_1 \longleftarrow^* \mathtt{M}_0 \longrightarrow {}^*\mathtt{M}_2$*, then* $\mathtt{M}_1 \longrightarrow^* \mathtt{M}_3 \longleftarrow^* \mathtt{M}_2$ *for some term* $\mathtt{M}_3$.

5.2. Interpretation of LAST proofs

Here we give an interpretation of **LAST** proofs into λLA terms.

$$\frac{}{\mathtt{x}\!:\!A \vdash \mathtt{x}\!:\!A}\;Id \qquad \frac{\Gamma_1 \vdash \mathtt{N}\!:\!A \quad \mathtt{x}\!:\!A, \Gamma_2 \vdash \mathtt{M}\!:\!C}{\Gamma_1, \Gamma_2 \vdash \mathtt{M[N/x]}:C}\;Cut$$

$$\frac{\Gamma \vdash \mathtt{M}\!:\!C}{\Delta, \Gamma \vdash \mathtt{M}\!:\!C}\;Weak \qquad \frac{\mathtt{x}\!:\![A]_!, \mathtt{y}\!:\![A]_!, \Gamma \vdash \mathtt{M}\!:\!C}{\mathtt{z}\!:\![A]_!, \Gamma \vdash \mathtt{M[z/x,\ z/y]}:C}\;Cntr$$

$$\frac{\Gamma_1 \vdash \mathtt{N}\!:\!A_1 \quad \mathtt{x}\!:\!A_2, \Gamma_2 \vdash \mathtt{M}\!:\!C}{\Gamma_1, \mathtt{y}\!:\!A_1 \multimap A_2, \Gamma_2 \vdash \mathtt{M[yN/x]}:C}\;{\multimap}l \qquad \frac{\mathtt{x}\!:\!A_1, \Gamma \vdash \mathtt{M}\!:\!A_2}{\Gamma \vdash \lambda\mathtt{x.M}\!:\!A_1 \multimap A_2}\;{\multimap}r$$

$$\frac{\mathtt{x}\!:\![A]_!, \Gamma \vdash \mathtt{M}\!:\!C}{\mathtt{y}\ :!A, \Gamma \vdash \mathtt{let\ y\ be\ !x\ in\ M}:C}\;!l \qquad \frac{\mathtt{x}\!:\!B \vdash \mathtt{M}\!:\!A}{\mathtt{x}\!:\![B]_! \vdash \mathtt{!M}\ :!A}\;!r$$

$$\frac{\mathtt{x}\!:\![A]_\S, \Gamma \vdash \mathtt{M}\!:\!C}{\mathtt{y}\!:\!\S A, \Gamma \vdash \mathtt{let\ y\ be\ }\S\mathtt{x\ in\ M}:C}\;\S l \qquad \frac{\Gamma, \Delta \vdash \mathtt{M}\!:\!A}{[\Gamma]_!, [\Delta]_\S \vdash \S\mathtt{M}\!:\!\S A}\;\S r$$

$$\frac{\mathtt{x}\!:\!A[u/x], \Gamma \vdash \mathtt{M}\!:\!C}{\mathtt{x}\!:\!\forall x.A, \Gamma \vdash \mathtt{M}\!:\!C}\;\forall l \qquad \frac{\Gamma \vdash \mathtt{M}\!:\!A}{\Gamma \vdash \mathtt{M}\!:\!\forall x.A}\;\forall r$$

$$\frac{\mathtt{x}\!:\!A[u/x], \Gamma \vdash \mathtt{M}\!:\!C}{\mathtt{x}\!:\!u \in \{x|A\}, \Gamma \vdash \mathtt{M}\!:\!C}\;\in l \qquad \frac{\Gamma \vdash \mathtt{M}\!:\!A[u/x]}{\Gamma \vdash \mathtt{M}\!:\!u \in \{x|A\}}\;\in r$$

In rule (!), $\mathtt{x}\!:\!B$ may be absent. In rule ($\forall r$), x is not free in Γ.

Figure 2. Interpretation of **LAST** proofs

A *declaration* is either of the form $\mathtt{x} : A$ or of the form $\mathtt{x} : [A]_\dagger$ ($[A]_\dagger$ is called a $\dagger$-*discharged formula*). Throughout this section, $\Gamma, \Delta, \cdots$ stand for a finite set of declarations. Now consider the *term assignment rules* in Figure 2. To each **LAST** proof of conclusion $\vec{B} \vdash A$, we assign a pseudo-term M with free variables $\vec{x}$ such that $\vec{x}\!:\!\vec{B} \vdash M : A$ is derivable according to those rules. Note that here we use discharged formulas in addition to the ordinary formulas, and that the contraction rule is applied to !-discharged formulas rather than !-prefixed formulas. It is, however, easy to see that it makes no significant difference between two proof systems. As expected, we have

THEOREM 5.5. *Proofs of* **LAST** *are interpreted by (well-formed) terms of*

λLA. *More exactly, if* $\vec{\boldsymbol{x}}\!:\!\vec{A}, \vec{\boldsymbol{y}}\!:\![\vec{B}]_!, \vec{\boldsymbol{z}}\!:\![\vec{C}]_\S \vdash \mathtt{M}\!:\!D$*, then* $\mathtt{M} \in \mathcal{T}_{\{\vec{x}\},\{\vec{y}\},\{\vec{z}\}}$.

Proof. By induction on the length of the derivation. As for (Cut) and ($\multimap l$), we use the substitution lemma:

- If $\mathtt{M} \in \mathcal{T}_{X \uplus \{x\},Y,Z}$ and $\mathtt{N} \in \mathcal{T}_{X,Y,Z}$ then $\mathtt{M[N/x]} \in \mathcal{T}_{X,Y,Z}$.

As for ($Weak$), we use the weakening lemma:

- If $\mathtt{M} \in \mathcal{T}_{X,Y,Z}$ and $X \subseteq X'$, $Y \subseteq Y'$ and $Z \subseteq Z'$, then $\mathtt{M} \in \mathcal{T}_{X',Y',Z'}$.

Both are proved by induction on M. ■

In [21] we proved the subject reduction theorem for **ILAL**, and it is routine to accommodate the proof to **LAST** (see Section 6.2 of [22]). Thus we have

Theorem 5.6 (Subject Reduction for **LAST**). *If* $\Gamma \vdash \mathtt{M}\!:\!A$ *is derivable and* $\mathtt{M} \longrightarrow \mathtt{N}$*, then* $\Gamma \vdash \mathtt{N}\!:\!A$ *is derivable.*

The normalization theorem and the subject reduction theorem together imply that every *explicit* cut, which corresponds to a redex of λLA, can be eliminated from a **LAST** derivation. In the meantime, it is easy to see that every *implicit* cut, which does not correspond to a redex of λLA, can be eliminated without increasing the size of a derivation. For example, a principal cut for $\in$ can be eliminated as follows:

$$\dfrac{\dfrac{\overset{\vdots\ \pi_1}{\Gamma_1 \vdash \mathtt{N}\!:\!A[v/y]}}{\Gamma_1 \vdash \mathtt{N}\!:\!v \in \{y|A\}} \quad \dfrac{\overset{\vdots\ \pi_2}{x\!:\!A[v/y], \Gamma_2 \vdash \mathtt{M}\!:\!C}}{x\!:\!v \in \{y|A\}, \Gamma_2 \vdash \mathtt{M}\!:\!C}}{\Gamma_1, \Gamma_2 \vdash \mathtt{M[N/x]}\!:\!C} \quad \longrightarrow \quad \dfrac{\overset{\vdots\ \pi_1}{\Gamma_1 \vdash \mathtt{N}\!:\!A[v/y]} \quad \overset{\vdots\ \pi_2}{x\!:\!A[v/y], \Gamma_2 \vdash \mathtt{M}\!:\!C}}{\Gamma_1, \Gamma_2 \vdash \mathtt{M[N/x]}\!:\!C}$$

See Section 6.3 of [22] for a precise argument. As a consequence, we have

Theorem 5.7 (Cut-Elimination Theorem for **LAST**). *If* $\Gamma \vdash \mathtt{M}\!:\!C$ *is derivable, then* $\Gamma \vdash \mathtt{N} : C$ *is cut-free derivable, where* $\mathtt{N}$ *is the normal form of* $\mathtt{M}$.

Corollary 5.8.

(1) If $\vdash \mathtt{M}\!:\!\S A$ *is derivable and* $\mathtt{M}$ *is normal, then* $\mathtt{M}$ *is of the form* $\S\mathtt{N}$ *and* $\vdash \mathtt{N}\!:\!A$ *is derivable.*

(2) If $\vdash \mathtt{M} : \exists y.A$ *is derivable and* $\mathtt{M}$ *is normal, then* $\mathtt{M}$ *is of the form* $\lambda\mathtt{z.zN}$ *and* $\vdash \mathtt{N}\!:\!A[u/y]$ *is derivable for some term* u *of* **LAST**.

(3) If $\vdash \mathtt{M} : A \otimes B$ *is derivable and* $\mathtt{M}$ *is normal, then* $\mathtt{M}$ *is of the form* $\lambda\mathtt{z.zNL}$ *and* $\vdash \mathtt{N}\!:\!A$ *and* $\vdash \mathtt{L}\!:\!B$ *are derivable.*

Proof. (1) is obvious. As for (2), recall that $\exists y.A$ is defined as $\forall x.(\forall y.(A \multimap t_0 \in x) \multimap t_0 \in x)$. The last part of the cut-free derivation of $\vdash \mathtt{M}\!:\!\exists y.A$ must be of the following form:

$$\cfrac{\cfrac{\cfrac{\cfrac{\overset{\vdots}{\vdash \mathtt{N}:A[u/y]} \quad \mathtt{w}:t_0 \in x \vdash \mathtt{w}:t_0 \in x}{\mathtt{z}:A[u/y] \multimap t_0 \in x \vdash \mathtt{zN}:t_0 \in x}}{\mathtt{z}:\forall y.(A \multimap t_0 \in x) \vdash \mathtt{zN}:t_0 \in x}}{\vdash \lambda\mathtt{z.zN}:\forall y.(A \multimap t_0 \in x) \multimap t_0 \in x}}{\vdash \lambda\mathtt{z.zN}:\exists y.A},$$

and $\mathtt{M} \equiv \lambda\mathtt{z.zN}$. (3) can be shown similarly. ∎

5.3. Program Extraction

Now we describe how to extract a term of λLA from a proof of **LAST**. Instead of dealing with the general case, we confine ourselves to proofs of formulas of the form $\forall x \in \mathsf{W}.\S^d\exists y \in \mathsf{W}.A$, which should be sufficient for illustrating the general pattern.

The following definition is needed to clarify the notion of representation in λLA. Let $\approx$ be the least binary congruence relation that satisfies the following:

(η) $\lambda\mathtt{x.Mx} \approx \mathtt{M}$, if $\mathtt{x} \notin FV(\mathtt{M})$.
$(\mathtt{let})$ $\mathtt{let\ N\ be\ \dagger x\ in\ M} \approx \mathtt{M}$, if $\mathtt{x} \notin FV(\mathtt{M})$.
$(\lambda\text{-}\mathtt{let})$ $\lambda\mathtt{x}.(\mathtt{let\ M\ be\ \dagger y\ in\ N}) \approx \mathtt{let\ M\ be\ \dagger y\ in\ }\lambda\mathtt{x.N}$, if $\mathtt{x} \notin FV(\mathtt{M})$.
$(\mathtt{let}\text{-}\mathtt{let})$ $\mathtt{let\ M\ be\ \dagger x\ in\ (let\ N\ be\ \dagger y\ in\ L)}$
$\approx \mathtt{let\ N\ be\ \dagger y\ in\ (let\ M\ be\ \dagger x\ in\ L)}$,
if $\mathtt{x} \notin FV(\mathtt{N})$ and $\mathtt{y} \notin FV(\mathtt{M})$.

It is easy to see that $\approx$ is compatible with $\longrightarrow^*$. Namely, if $\mathtt{M} \approx \mathtt{N}$ and $\mathtt{M} \longrightarrow^* \mathtt{M}'$, then there is a term $\mathtt{N}'$ such that $\mathtt{N} \longrightarrow^* \mathtt{N}'$ and $\mathtt{M}' \approx \mathtt{N}'$.

Definition 5.9. Let $w \equiv i_0 \cdots i_n \in \mathbb{W}$. Then $\overline{w}$ denotes

$$\lambda\mathtt{x}_0\mathtt{x}_1.(\mathtt{let\ x_0\ be\ !z_0\ in}(\mathtt{let\ x_1\ be\ !z_1\ in\ }\S\lambda\mathtt{y}.(\mathtt{z}_{i_0}\cdots(\mathtt{z}_{i_n}\mathtt{y})\cdots))).$$

In particular, when w is the empty word ϵ, $\overline{w}$ denotes

$$\lambda\mathtt{x}_0\mathtt{x}_1.(\mathtt{let\ x_0\ be\ !z_0\ in}(\mathtt{let\ x_1\ be\ !z_1\ in\ }\S\lambda\mathtt{y.y})).$$

A λLA-*representation of* w is a term $\mathtt{M}$ of λLA such that $\vdash \mathtt{M}:\mathtt{w} \in \mathsf{W}$ is derivable.

It is easy to show that $\overline{w}$ is a λLA-representation of $w \in \mathbb{W}$. Conversely, if $\mathtt{M}$ is a λLA-representation of w, then $\mathtt{M} \approx \overline{w}$. Furthermore, λLA-representations of two distinct words are not related by $\approx$. Therefore, there is a canonical one-one mapping from $\mathbb{W}$ to $\mathcal{W}/{\approx}$, where $\mathcal{W}$ is the set of λLA-representations. Since $\approx$ is compatible with $\longrightarrow^*$, the following definition makes sense:

Definition 5.10. A function $f : \mathbb{W} \longrightarrow \mathbb{W}$ is λLA-*represented* by a term $\mathtt{F}$ if there is a natural number $d \geq 0$ such that for every $w \in \mathbb{W}$ and its λLA-representation $\mathtt{M}$, $\mathtt{FM} \longrightarrow^{*} \S^{d}\mathtt{N}$ and $\mathtt{N}$ is a λLA-representation of $f(w)$.

The following terms are useful in program extraction. For each $d \geq 0$, define λLA terms $\mathtt{Fst}^{d}\mathtt{(z)}$ and $\mathtt{Ext}^{d}(\mathtt{z})$, both having a free variable $\mathtt{z}$, as follows:

$$\begin{aligned}
\mathtt{Fst}^{0}(\mathtt{z}) &\equiv \mathtt{z}(\lambda\mathtt{xy.x}) \\
\mathtt{Fst}^{d+1}(\mathtt{z}) &\equiv \mathtt{let\ z\ be\ } \S\mathtt{y}\ \mathtt{in}\ \S\mathtt{Fst}^{d}(\mathtt{y}) \\
\mathtt{Ext}^{0}(\mathtt{z}) &\equiv \mathtt{z}(\lambda\mathtt{x.x}) \\
\mathtt{Ext}^{d+1}(\mathtt{z}) &\equiv \mathtt{let\ z\ be\ } \S\mathtt{y}\ \mathtt{in}\ \S\mathtt{Ext}^{d}(\mathtt{y})
\end{aligned}$$

Then it is easy to see that $\mathtt{Fst}^{d}(\S^{d}\lambda\mathtt{z.zNL}) \longrightarrow^{*} \S^{d}\mathtt{N}$ and $\mathtt{Ext}^{d}(\S^{d}\lambda\mathtt{z.zM}) \longrightarrow^{*} \S^{d}\mathtt{M}$ for every $d \geq 0$.

Our main theorem in this section is the following:

Theorem 5.11 (Program Extraction). *Let* $\vdash \forall x \in \mathsf{W}.\S^{d}(\exists y \in \mathsf{W}.A)$ *be provable in* **LAST**. *Then there is a term* $\mathtt{F}$ *of* λLA *such that for every* $w \in \mathbb{W}$, $\mathtt{F}\overline{w}$ *reduces to* $\S^{d}\mathtt{N}$, $\mathtt{N}$ *is a* λLA-*representation of some* $w' \in \mathbb{W}$, *and* $A[\mathsf{w}/x, \mathsf{w}'/y]$ *is provable in* **LAST**.

Proof. Let $\mathtt{G}$ be a term of λLA such that

$$\vdash \mathtt{G}\!:\!\forall x \in \mathsf{W}.\S^{d}\exists y \in \mathsf{W}.A$$

is derivable. We claim that the desired term $\mathtt{F}$ is $\lambda\mathtt{z.Fst}^{d}(\mathtt{Ext}^{d}(\mathtt{Gz}))$.

Let $w \in \{0,1\}^{*}$. Then, $\vdash \overline{w} : \mathsf{w} \in \mathsf{W}$ is derivable. On the other hand, it is easy to see that $\vdash \mathtt{G} : \mathsf{w} \in \mathsf{W} \multimap \S^{d}\exists y \in \mathsf{W}.A[\mathsf{w}/x]$ is derivable, so that we have $\vdash \mathtt{G}\overline{w} : \S^{d}\exists y \in \mathsf{W}.A[\mathsf{w}/x]$. By the subject reduction theorem, the normal form of $\mathtt{G}\overline{w}$ is of the same type and by Corollary 5.8, it must be of the form $\S^{d}\lambda\mathtt{z.z}(\lambda\mathtt{w.wNL})$. Moreover, $\vdash \mathtt{N}\!:\!u \in \mathsf{W}$ and $\vdash \mathtt{L}\!:\!A[\mathsf{w}/x, u/y]$ must be derivable for some term u of **LAST**. By Proposition 4.6 (3), $u \equiv \mathsf{w}'$ for some $w' \in \{0,1\}^{*}$. Hence $\mathtt{N}$ is a λLA-representation of w' such that $\vdash A[\mathsf{w}/x, \mathsf{w}'/y]$ is provable in **LAST**.

To put things together, we obtain:

$$\begin{aligned}
&(\lambda\mathtt{z.Fst}^{d}(\mathtt{Ext}^{d}(\mathtt{Gz})))\overline{w} \longrightarrow \mathtt{Fst}^{d}(\mathtt{Ext}^{d}(\mathtt{G}\overline{w})) \\
&\quad\longrightarrow^{*}\mathtt{Fst}^{d}(\mathtt{Ext}^{d}(\S^{d}\lambda\mathtt{z.z}(\lambda\mathtt{w.wNL}))) \longrightarrow^{*}\mathtt{Fst}^{d}(\S^{d}\lambda\mathtt{w.wNL}) \longrightarrow^{*}\S^{d}\mathtt{N} \approx \S^{d}\overline{w'},
\end{aligned}$$

as required. ■

As a corollary, we have the following characterization theorem:

COROLLARY 5.12. **(Characterization of the Polytime Functions)** *Let* $\phi : \mathbb{W} \longrightarrow \mathbb{W}$ *be a function. The following are equivalent:*

(1) ϕ *is computable in polynomial time.*

(2) ϕ *is provably total with domain* W *and range* W *in* **LAST**.

(3) ϕ *is* λLA*-representable.*

PROOF. ($1 \Rightarrow 2$) By Theorem 4.17.
($2 \Rightarrow 3$) By definition, there is a term f of **LAST**, a natural number $d \geq 0$ and a term $\mathtt{G}$ of λLA such that $\vdash \mathtt{G} : \forall x \in \mathsf{W}.\S^d \exists^! y \in \mathsf{W}(\langle x, y\rangle \in f)$. From this, we can easily obtain $\vdash \mathtt{G}' : \forall x \in \mathsf{W}.\S^d \exists y \in \mathsf{W}(\langle x, y\rangle \in f)$. Hence the program extraction theorem applies, and we obtain a term $\mathtt{F}$ of λLA such that for any $w \in \mathbb{W}$, $\mathtt{F}\overline{w}$ reduces to $\S^d\mathtt{N}$, $\mathtt{N}$ is a λLA-representation of some $w' \in \mathbb{W}$, and $\langle \mathsf{w}, \mathsf{w}'\rangle \in f$ is provable in **LAST**. The latter is equivalent to $\phi(w) = w'$ by Proposition 4.2. Therefore $\mathtt{F}$ λLA-represents ϕ.
($3 \Rightarrow 1$) By the polytime normalization theorem (see [21]). ■

6. Conclusion

Substructural logics are often criticized as lacking good reasons for restricting structural inference rules. Here, however, we find at least two reasons for restricting contraction: to save unrestricted comprehension (as observed by Grishin) and to keep feasibility of computation. These two reasons have nicely combined in Girard's work and resulted in a polytime naive set theory, Light Linear Set Theory. A drawback in his set theory is that, in contrast to its "light" computational complexity, the syntax is too "heavy." Our contributions in this paper are, firstly, to simplify the syntax by adding unrestricted weakening (as has been done by Asperti for Light Linear Logic), and secondly, to formally verify the truly polytime character of the resulting set theory, **LAST**. We believe that our work clarifies the essentials of Girard's idea, and thus makes this important application of substructural logics more accessible.

It should be noted, however, that **LAST** is hardly considered as a *working* system of mathematics, because the reasoning allowed by **LAST** is too poor to formalize proofs of mathematically interesting theorems. Indeed, we have found difficulty even in proving the totality of division (when it is inductively defined based on subtraction). Nevertheless, there is much room for extending **LAST** to a more flexible and yet polytime naive set theory. One approach would be to extend it with function symbols and non-logical axioms.

Another research direction is to look for naive set theories which capture other complexity classes. We already have Elementary Affine Set Theory, which captures the elementary recursive functions, but can there be a stronger one? And if so, does it make sense to look for the "strongest" one? We leave these investigations for future work.

Ackowledments

The author would like to thank Professor Mitsuhiro Okada, Professor Hiroakira Ono and Dr. Masaru Shirahata for helpful suggestions. He would also like to thank Dr. Patrick Baillot and Daniel de Carvalho for stimulating discussions, and also for pointing out a mistake in an earlier version of this paper.

References

[1] Asperti, A., 'Light affine logic', *Proceedings of the Thirteenth Annual IEEE Symposium on Logic in Computer Science*, pp. 300 – 308, 1998.

[2] Asperti, A., and L. Roversi, 'Intuitionistic light affine logic (proof-nets, normalization complexity, expressive power, programming notation)', *ACM Transactions on Computational Logic*, 3(1): 137 – 175, 2002.

[3] Baillot, P., 'Stratified coherent spaces: a denotational semantics for light linear logic', *Theoretical Computer Science*, to appear.

[4] Barendregt, H. P., *The Lambda Calculus: Its Syntax and Semantics*, Elsevier North-Holland, 1981.

[5] Cantini, A., 'The undecidability of Grishin's set theory', *Studia Logica*, 74: 345 – 368, 2003.

[6] Danos, V., and J.-B. Joinet, 'Linear logic & elementary time', *Information and Computation*, 183(1): 123 – 137, 2003.

[7] Girard, J.-Y., 'Light linear logic', *Information and Computation*, 14(3): 175 – 204, 1998.

[8] Girard, J.-Y., 'Linear logic', *Theoretical Computer Science*, 50: 1 – 102, 1987.

[9] Grishin, V. N., 'A nonstandard logic and its application to set theory', In *Studies in Formalized Languages and Nonclassical Logics (Russian)*, pp. 135 – 171. Izdat, Nauka, Moskow, 1974.

[10] Grishin, V. N., 'Predicate and set theoretic calculi based on logic without contraction rules' (Russian), *Izvestiya Akademii Nauk SSSR Seriya Matematicheskaya*, 45(1): 47 – 68, 1981. English translation in *Math. USSR Izv.*, 18(1): 41 – 59, 1982.

[11] Hopcroft, J., and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Mass, 1979.

[12] Kanovitch, M, M. Okada, and A. Scedrov, 'Phase semantics for light linear logic', *Theoretical Computer Science*, 244(3): 525 – 549, 2003.

[13] Komori, Y., 'Illative combinatory logic based on BCK-logic', *Mathematica Japonica*, 34(4): 585 – 596, 1989.

[14] Lincoln, P., A. Scedrov, and N. Shankar, 'Decision problems for second order linear logic', *Proceedings of the Tenth Annual IEEE Symposium on Logic in Computer Science*, pp. 476 – 485, 1995.

[15] Murawski, A. S., and C.-H. L. Ong, 'Discreet games, light affine logic and PTIME computation', *Proceedings of Computer Science Logic 2000*, pp. 427 – 441. Springer-Verlag, LNCS 1862, 2000.

[16] Neergaard, P., and H. Mairson, 'LAL is square: Representation and expressiveness in light affine logic', presented at the Fourth International Workshop on Implicit Computational Complexity, 2002.

[17] Peterson, U., 'Logic without contraction as based on inclusion and unrestricted abstraction', *Studia Logica*, 64(3): 365 – 403, 2000.

[18] Schwichtenberg, H., and A. S. Troelstra, *Basic Proof Theory*, Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1996.

[19] Shirahata, M., 'A linear conservative extension of Zermelo-Fraenkel set theory', *Studia Logica*, 56: 361 – 392, 1996.

[20] Shirahata, M., 'Fixpoint theorem in linear set theory', unpublished manuscript, available at http://www.fbc.keio.ac.jp/∼sirahata/Research, 1999.

[21] Terui, K., 'Light affine lambda calculus and polytime strong normalization', *Proceedings of the Sixteenth Annual IEEE Symposium on Logic in Computer Science*, pp. 209 – 220, 2001. The full version is available at http://research.nii.ac.jp/∼terui.

[22] Terui, K., *Light Logic and Polynomial Time Computation*, PhD thesis, Keio University, 2002. Available at http://research.nii.ac.jp/∼terui.

[23] White, R., 'A demonstrably consistent type-free extension of the logic BCK', *Mathematica Japonica*, 32(1): 149 – 169, 1987.

[24] White, R., 'A consistent theory of attributes in a logic without contraction', *Studia Logica*, 52: 113 – 142, 1993.

Kazushige Terui
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan.
terui@nii.ac.jp