A Semantic Proof of Polytime Soundness of Light Affine Logic

Ugo Dal Lago[1] and Martin Hofmann[2]

[1] Dipartimento di Scienze dell'Informazione, Università di Bologna
[2] Institut für Informatik, LMU München

Abstract. We define a denotational semantics for Light Affine Logic (LAL) which has the property that denotations of functions are polynomial time computable by construction of the model. This gives a new proof of polytime-soundness of LAL which is considerably simpler than the standard proof based on proof nets and also is entirely semantical in nature. The model construction uses a new instance of a resource monoid; a general method for interpreting variations of linear logic with complexity restrictions introduced earlier by the authors.

1 Introduction

In recent years, a large number of characterizations of complexity classes based on logics and lambda calculi have appeared. At least three different principles have been exploited, namely linear types [3,10], restricted modalities in the context of linear logic [8,2,13] and non-size-increasing computation [9,1]. These systems have been studied with different, often unrelated methodologies. In particular, proofs of soundness (any function which is representable in the system lies in a complexity class) are usually quite complex and cannot be easily generalized. As a consequence, unifying, reasonably simple frameworks for the analysis of quantitative properties of computation are desirable. This would help to improve the understanding on existing systems, since proofs of soundness, especially conceptually simple ones, often shed light on the *reasons why* the system under consideration enjoys certain quantitative properties.

While we take the significance of LAL itself more or less for granted in this paper we may point out that it is the first system that characterises polynomial time without recourse to explicit resource bounds as found e.g. in Bounded Arithmetic and in addition allows one to define inductive datatypes as certain fomulas by impredicative quantification. Functions acting on those datatypes can thus be naturally represented as proofs via the well-known Curry-Howard correspondence (e.g. logical implication corresponds to functional types). And, noticeably, the class of representable first-order functions *equals* the polynomial time functions. One can thus view LAL as the first resource-free and purely proof-theoretic characterisation of polynomial time.

In a previous paper [7], we have introduced a new semantical framework which consists of an innovative modification of realizability whereby realizers and their

E.A. Hirsch et al. (Eds.): CSR 2008, LNCS 5010, pp. 134–145, 2008.

runtime are bounded by elements of a certain algebraic structure, a *resource monoid*. The axioms for resource monoids are such that for any resource monoid the category of corresponding realizability sets is symmetric monoidal closed and supports second-order quantification, i.e., impredicative polymorphism. With particular resource monoids one can then realize further constructs and type formers such as modalities or recursors. In [7] we have introduced resource monoids and provided concrete instances for LFPL [9] and Elementary Affine Logic (EAL, see [5]). A fairly complicated resource monoid for LAL with a consequently rather technical and unenlightening proof of correctness has been presented in [7].

In this paper we provide a very simple resource monoid for LAL. Not only do we obtain in this way a new, simpler, and conceptually appealing proof of polytime soundness (all definable functions on binary strings are polynomial time computable) for LAL; we also find that the resource monoid we obtain is quite natural; its members are triples (n, m, f) with $n, m \in \mathbb{N}$ and f a monotonically increasing polynomial-time function; the monoid operation which interprets tensor product is given by

$$(n, m, f) + (l, k, g) = (n + l, \max(m, k), \max(f, g)).$$

The order relation between these monoid elements is given by

$$(n, m, f) \leq (l, k, g) \iff (n \leq l) \wedge (n + m \leq l + k) \wedge (f \leq g).$$

The interpretation of the modalities ! and § of LAL uses the functional $f \mapsto \lambda x.x^2 f(x^2)$ which explains that bounding functions extracted from the interpretation are polynomials whose degree grows exponentially with the nesting depth of the modalities as is expected from the known proof based on proof nets and also the known hardness examples. Some formulae which are not provable syntactically can be justified in the semantics. An example is the distributive law $\S(A \otimes B) \multimap \S A \otimes \S B$.

The formal similarity of our resource monoid with the one for LFPL from [7] raises hopes for a system that somehow combines LFPL and LAL; we have to admit that these hopes have not, as yet, materialised if one discounts trivial solutions like the disjoint union of the two systems.

Related work. Semantic models for LAL exist [15,14]; however none of these models yields a proof of polytime soundness. More generally, the method of realizability has been used in connection with resource-bounded computation in several places. The most prominent is Cook and Urquhart's work [4], where terms of a language called PV^ω are used to realize formulas of bounded arithmetic. The contribution of that paper is related to ours in that realizability is used to show "polytime soundness" of a logic. There are important differences though. First, realizers in [4] are typed and very closely related to the logic that is being realized. Second, the language of realizers PV^ω only contains first-order recursion and is therefore too weak for systems with like LFPL or LAL that contain or define recursion with higher-order result types. In contrast, we use untyped realizers and interpret types as certain partial equivalence relations on those. This links

our work to the untyped realizability model HEO (due to Kreisel [12]). This, in turn, has also been done by Crossley et al. [6]. There, however, one proves externally that untyped realizers (in this case of bounded arithmetic formulas) are polytime, whereas our realizers are polytime bounded by construction.

2 Preliminaries

As in any realizability model, we need to introduce a language L in which to write realizers. We stay abstract here: all the results presented in this paper hold for every L satisfying some basic conditions, which we will now explain.

Let $L \subseteq \Sigma^*$ be the set of finite sequences over some finite alphabet Σ. We assume a pairing function $\langle \cdot, \cdot \rangle : L \times L \to L$ and a length function $| \cdot | : L \to \mathbb{N}$ such that $|\langle x, y \rangle| = |x| + |y| + cp$ and $|x| \leq \mathit{length}(x)$ yet $|x| = \Omega(\mathit{length}(x)^\varepsilon)$ for some $\varepsilon > 0$, where $\mathit{length}(x)$ is the number of symbols in x and $cp > 0$ is a fixed constant. We assume a reasonable encoding of algorithms as elements of L. We write $\{e\}(x)$ for the (possibly undefined) application of algorithm $e \in L$ to input $x \in L$. We furthermore assume an abstract time measure $\mathit{Time}(\{e\}(x)) \in \mathbb{N}$ such that $\mathit{Time}(\{e\}(x))$ is defined whenever $\{e\}(x)$ is and $\{e\}(x)$ can be evaluated on a Turing machine in time bounded by $p(\mathit{Time}(\{e\}(x)), |e|, |x|)$, where $p : \mathbb{N}^3 \to \mathbb{N}$ is a fixed polynomial. We require that algorithms manipulating higher-order functions and 0-1 strings can be represented in L and their abstract time measures satisfy intuitive bounds. For example, we assume the existence of e_{comp} (composition) and e_{contr} (duplication, copying) such that for every x, y it holds that $\{e_{comp}\}(\langle x, y \rangle) = z$ where $|z| = |x| + |y| + O(1)$ and $\{z\}(w) = \{y\}(\{x\}(w))$ and $\{e_{contr}\}(x) = \langle x, x \rangle$. Moreover, $\mathit{Time}(\{e_{contr}\}(x)) = O(|x|)$ and $\mathit{Time}(\{e_{comp}\}(\langle x, y \rangle)) = O(1)$ and $\mathit{Time}(\{z\}(w)) = \mathit{Time}(\{x\}(w)) + \mathit{Time}(\{y\}(\{x\}(w))) + O(1)$.

This abstract framework can be instantiated with call-by-value lambda terms [7] or Turing machines [10]. Since the instantiation is irrelevant for our purposes we do not give any details.

3 Resource Monoids and Length Spaces

In this section, we recall the notion of a resource monoid [7] and the corresponding category of realizability sets, called *length spaces*, as well as its general properties.

A *resource monoid* is a quadruple $M = (|M|, +, \leq_M, \mathcal{D}_M)$ where
(i) $(|M|, +)$ is a commutative monoid;
(ii) $\leq_M$ is a pre-order on $|M|$ which is compatible with $+$;
(iii) $\mathcal{D}_M : \{(\alpha, \beta) \mid \alpha \leq_M \beta\} \to \mathbb{N}$ is a function such that for every α, β, γ

$$\mathcal{D}_M(\alpha, \beta) + \mathcal{D}_M(\beta, \gamma) \leq \mathcal{D}_M(\alpha, \gamma)$$
$$\mathcal{D}_M(\alpha, \beta) \leq \mathcal{D}_M(\alpha + \gamma, \beta + \gamma)$$

and, moreover, for every $n \in \mathbb{N}$ there is α such that $\mathcal{D}_M(0, \alpha) \geq n$.

Given a resource monoid $M = (|M|, +, \leq_M, \mathcal{D}_M)$, the function $\mathcal{F}_M : |M| \to \mathbb{N}$ is defined by putting $\mathcal{F}_M(\alpha) = \mathcal{D}_M(0, \alpha)$.

We shall use elements of a resource monoid to bound data, algorithms, and runtimes in the following way: an element φ bounds an algorithm e if $\mathcal{F}_M(\varphi) \geq |e|$ and, more importantly, whenever α bounds an input x to e then there must be a bound $\beta \leq_M \varphi + \alpha$ for the result $y = \{e\}(x)$ and, most importantly, the runtime of that computation must be bounded by $\mathcal{D}_M(\beta, \varphi + \alpha)$. So, in a sense, we have the option of either producing a large output fast or to take a long time for a small output. The "inverse triangular" law above ensures that the composition of two algorithms bounded by φ_1 and φ_2, respectively, can be bounded by $\varphi_1 + \varphi_2$ or a simple modification thereof. In particular, the contribution of the unknown intermediate result in a composition cancels out using that law. Another useful intuition is that $\mathcal{D}_M(\alpha, \beta)$ behaves like the difference $\beta - \alpha$, indeed, $(\beta - \alpha) + (\gamma - \beta) \leq \gamma - \alpha$.

A *length space* on a resource monoid $M = (|M|, +, \leq_M, \mathcal{D}_M)$ is a pair $A = (|A|, \Vdash_A)$, where $|A|$ is a set and $\Vdash_A \subseteq |M| \times L \times |A|$ is a(n infix) relation satisfying the following conditions:

(i) if $\alpha, e \Vdash_A a$, then $\mathcal{F}_M(\alpha) \geq |e|$;
(ii) for every $a \in |A|$, there are α, e such that $\alpha, e \Vdash_A a$;
(iii) if $\alpha, e \Vdash_A a$ and $\alpha \leq_M \beta$, then $\beta, e \Vdash_A a$;
(iv) if $\alpha, e \Vdash_A a$ and $\alpha, e \Vdash_A b$, then $a = b$.

The last requirement implies that each element of $|A|$ is uniquely determined by the (nonempty) set of its realizers and in particular limits the cardinality of any length space to the number of partial equivalence relations on L.

A *morphism* from length space $A = (|A|, \Vdash_A)$ to length space $B = (|B|, \Vdash_B)$ (on the same resource monoid $M = (|M|, +, \leq_M, \mathcal{D}_M)$) is a function $f : |A| \to |B|$ such that there exist $e \in L = \Sigma^*$, $\varphi \in |M|$ with $\mathcal{F}_M(\varphi) \geq |e|$ and whenever $\alpha, d \Vdash_A a$, there must be β, c such that

(i) $\beta, c \Vdash_B f(a)$;
(ii) $\beta \leq_M \varphi + \alpha$;
(iii) $\{e\}(d) = c$;
(iv) $\mathit{Time}(\{e\}(d)) \leq \mathcal{D}_M(\beta, \varphi + \alpha)$.

We call e a *realizer* of f and φ a *majorizer* of f. The set of all morphisms from A to B is denoted as $\mathit{Hom}(A, B)$. If f is a morphism from A to B realized by e and majorized by φ, then we will write $f : A \xrightarrow{e,\varphi} B$ or $\varphi, e \Vdash_{A \multimap B} f$.

Given two length spaces $A = (|A|, \Vdash_A)$ and $B = (|B|, \Vdash_B)$ on the same resource monoid M, we define $A \otimes B = (|A| \times |B|, \Vdash_{A \otimes B})$ (on M) where $e, \alpha \Vdash_{A \otimes B} (a, b)$ iff $\mathcal{F}_M(\alpha) \geq |e|$ and there are f, g, β, γ with

$$\begin{gathered} f, \beta \Vdash_A a \\ g, \gamma \Vdash_B b \\ e = \langle f, g \rangle \\ \alpha \geq_M \beta + \gamma \end{gathered}$$

The following result is from [7]:

Theorem 1. *The category of length spaces for any resource monoid is symmetric monoidal closed with respect to the tensor product given above. In particular, there is a neutral object I and for any two length spaces an exponential $A \multimap B$.*

3.1 Interpreting Multiplicative Affine Logic

We can now formally show that second order multiplicative affine logic (i.e., multiplicative linear logic plus full weakening) can be interpreted inside the category of length spaces on any monoid M. Doing this will simplify the analysis of LAL, since the latter can be obtained by enriching multiplicative affine logic with two modalities. Formulae of (intuitionistic, second order) multiplicative affine logic are generated by the following productions:

$$A ::= \alpha \mid A \multimap A \mid A \otimes A \mid \forall \alpha.A$$

where α ranges over a countable set of atoms. Rules are reported in Figure 1. A *realizability environment* is a partial function assigning length spaces (on the

Identity, Cut and Weakening.

$$\frac{}{A \vdash A}\ I \qquad \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B}\ U \qquad \frac{\Gamma \vdash A}{\Gamma, B \vdash A}\ W$$

Multiplicative Logical Rules.

$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}\ L_{\otimes} \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ R_{\otimes}$$

$$\frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}\ L_{\multimap} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ R_{\multimap}$$

Second Order Logical Rules.

$$\frac{\vdash \Gamma, A[C/\alpha] \vdash B}{\Gamma, \forall \alpha.A \vdash B}\ L^{\forall} \qquad \frac{\Gamma \vdash A \quad \alpha \notin FV(\Gamma)}{\Gamma \vdash \forall \alpha.A}\ R^{\forall}$$

Fig. 1. Intuitionistic Multiplicative Affine Logic

same resource monoid) to atoms. Realizability semantics $[\![A]\!]^{\mathscr{R}}_{\eta}$ of a formula A on the realizability environment η is defined by induction on A:

$$\begin{aligned}
[\![\alpha]\!]^{\mathscr{R}}_{\eta} &= \eta(\alpha) \\
[\![A \otimes B]\!]^{\mathscr{R}}_{\eta} &= [\![A]\!]^{\mathscr{R}}_{\eta} \otimes [\![B]\!]^{\mathscr{R}}_{\eta} \\
[\![A \multimap B]\!]^{\mathscr{R}}_{\eta} &= [\![A]\!]^{\mathscr{R}}_{\eta} \multimap [\![B]\!]^{\mathscr{R}}_{\eta} \\
[\![\forall \alpha.A]\!]^{\mathscr{R}}_{\eta} &= (|[\![\forall \alpha.A]\!]^{\mathscr{R}}_{\eta}|, \Vdash_{[\![\forall \alpha.A]\!]^{\mathscr{R}}_{\eta}})
\end{aligned}$$

where

$$|[\![\forall\alpha.A]\!]^{\mathscr{R}}_{\eta}| \;=\; \prod_{C\in\mathscr{U}} |[\![A]\!]^{\mathscr{R}}_{\eta[\alpha\to C]}|$$

$$\alpha, e \Vdash_{[\![\forall\alpha.A]\!]^{\mathscr{R}}_{\eta}} a \iff \forall C.\alpha, e \Vdash_{[\![A]\!]^{\mathscr{R}}_{\eta[\alpha\to C]}} a$$

Here $\mathscr{U}$ stands for the class of all length spaces. Some care is needed when defining the product since strictly speaking it does not exist for size reasons. The standard way out is to let the product range over those length spaces whose underlying set equals the set of equivalence classes of a partial equivalence relation on L. As already mentioned, every length space is isomorphic to one of that form. When working with the product one has to insert these isomorphisms in appropriate places which, however, we elide to increase readability.

If $n \geq 0$ and $A_1, \ldots, A_n$ are formulas, the expression $[\![A_1 \otimes \ldots \otimes A_n]\!]^{\mathscr{R}}_{\eta}$ stands for I if $n = 0$ and $[\![A_1 \otimes \ldots \otimes A_{n-1}]\!]^{\mathscr{R}}_{\eta} \otimes [\![A_n]\!]^{\mathscr{R}}_{\eta}$ if $n \geq 1$.

4 Light Length Spaces

Light Affine Logic extends Multiplicative Affine Logic by two modalities ! and § which are governed by the rules in Figure 2. In LAL, we can use variations on

Exponential Rules and Contraction.

$$\frac{\Gamma, \Delta \vdash A}{\S\Gamma, !\Delta \vdash \S A}\ P_{\S} \qquad \frac{A \vdash B}{!A \vdash !B}\ P^1_{!} \qquad \frac{\vdash A}{\vdash !A}\ P^2_{!} \qquad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\ C$$

Fig. 2. Intuitionistic Light Affine Logic

the usual impredicative encodings of natural numbers, lists, etc. For example, binary lists can be encoded as cut-free proofs for

$$List_{\mathsf{LAL}} \equiv \forall\alpha.!(\alpha \multimap \alpha) \multimap !(\alpha \multimap \alpha) \multimap \S(\alpha \multimap \alpha)$$

while natural numbers correspond to proofs for

$$Int_{\mathsf{LAL}} \equiv \forall\alpha.!(\alpha \multimap \alpha) \multimap \S(\alpha \multimap \alpha).$$

Now, let π be an LAL proof with conclusion $List_{\mathsf{LAL}} \vdash List_{\mathsf{LAL}}$. If we cut π against proofs corresponding to binary lists and we normalize the obtained proof, we get a proof corresponding to a binary list. So any proof like π *represents* a function from binary lists to binary lists. The above definition can be easily generalized to the cases when π has conclusion in the form $\{!, \S\}^j List_{\mathsf{LAL}} \vdash \{!, \S\}^k List_{\mathsf{LAL}}$.

But what is the expressive power of Light Affine Logic? The class of representable functions include all the polytime functions (see [16]):

Theorem 2 (Polytime Completeness). *Every polytime function on binary lists is represented by a* LAL *proof* $\pi : List_{\mathsf{LAL}} \multimap \S^n List_{\mathsf{LAL}}$.

We will now describe a resource monoid with the property that the ensuing category of length spaces provides structure for the interpretation of these modalities while allowing us to extract polytime bounds for functions of basic type. For ease of notation we denote $\max(m, n)$ by $m \mid n$:

Definition 1. *The algebraic structure* $\mathcal{L}$ *is the quadruple* $(|\mathcal{L}|, +, \leq_{\mathcal{L}}, \mathcal{D}_{\mathcal{L}})$ *such that:*

- *Elements of* $|\mathcal{L}| \subseteq \mathbb{N} \times \mathbb{N} \times \mathbb{N}^{\mathbb{N}}$ *are triples* (n, m, f) *such that* $f : \mathbb{N} \to \mathbb{N}$ *is a monotonically increasing polytime function.*
- *For every* $(n, m, f), (l, k, g) \in |\mathcal{L}|$, $(n, m, f) + (l, k, g) = (n + l, m \mid k, f \mid g)$.
- *For every* $(n, m, f), (l, k, g) \in |\mathcal{L}|$, $(n, m, f) \leq_{\mathcal{L}} (l, k, g)$ *iff* $n \leq l$, $n+m \leq l+k$ *and* $f \leq g$.
- *For every* $(n, m, f), (l, k, g) \in |\mathcal{L}|$ *such that* $(n, m, f) \leq_{\mathcal{L}} (l, k, g)$,

$$\mathcal{D}_{\mathcal{L}}((n, m, f), (l, k, g)) = (l - n)g(l + k).$$

The triple $(0, 0, 0) \in |\mathcal{L}|$, *denoted* $0_{\mathcal{L}}$, *is an identity for* $+$.

The binary relation $\leq_{\mathcal{L}}$ is trivially reflexive. Moreover, it is transitive:

Lemma 1 (Transitivity). *If* α, β, γ *are in* $|\mathcal{L}|$, $\alpha \leq_{\mathcal{L}} \beta$ *and* $\beta \leq_{\mathcal{L}} \gamma$, *then* $\alpha \leq_{\mathcal{L}} \gamma$.

Proof. Let $(n, m, f), (l, k, g), (p, q, h) \in |\mathcal{L}|$. Moreover, let $(n, m, f) \leq_{\mathcal{L}} (l, k, g)$ and $(l, k, g) \leq_{\mathcal{L}} (p, q, h)$. Trivially:

$$n \leq l \leq p;$$
$$n + m \leq l + k \leq p + q;$$
$$f \leq g \leq h.$$

In other words $(n, m, f) \leq_{\mathcal{L}} (p, q, h)$. □

But $\leq_{\mathcal{L}}$ is even compatible with $+$:

Lemma 2 (Compatibility). $0_{\mathcal{L}} \leq_{\mathcal{L}} \alpha$ *for every* $\alpha \in |\mathcal{L}|$. *Moreover, if* α, β, γ *are in* $|\mathcal{L}|$ *and* $\alpha \leq_{\mathcal{L}} \beta$, *then* $\alpha + \gamma \leq_{\mathcal{L}} \beta + \gamma$.

The following is now immediate.

Lemma 3. $\mathcal{L}$ *is a resource monoid.*

Definition 2. *A* light length space *is a length space over the resource monoid* $\mathcal{L}$. *Given a light length space* $A = (|A|, \Vdash_A)$, *the light spaces* $!A = (|A|, \Vdash_{!A})$ *and* $\S A = (|A|, \Vdash_{\S A})$ *both with underlying set* $|A|$, *are defined by:*

$$(n, m, f), e \Vdash_{!A} a \iff \exists (l, k, g) \in |\mathcal{L}|.(l, k, g), e \Vdash_A a \wedge (1, l + k, g^+) \leq_{\mathcal{L}} (n, m, f)$$
$$(n, m, f), e \Vdash_{\S A} a \iff \exists (l, k, g) \in |\mathcal{L}|.(lk, k, g), e \Vdash_A a \wedge (l, k, g^+) \leq_{\mathcal{L}} (n, m, f)$$

where $g^+(x) = x^2 g(x^2)$.

The constructions ! and § on light length spaces serve to capture the exponential modalities of light affine logic. Their definition is the crucial contribution of this

paper. The relations $\Vdash_{!A}, \Vdash_{\S A}$ can equivalently be defined inductively by the following rules:

$$\frac{\alpha, e \Vdash_{!A} a \quad \alpha \leq \beta}{\beta, e \Vdash_{!A} a} \qquad \frac{\alpha, e \Vdash_{\S A} a \quad \alpha \leq \beta}{\beta, e \Vdash_{\S A} a} \qquad \frac{(l,k,g), e \Vdash_A a}{(1,(l+k),g^+), e \Vdash_{!A} a} \qquad \frac{(lk,k,g), e \Vdash_A a}{(l,k,g^+), e \Vdash_{\S A} a}$$

Before we embark on the verification that these settings admit an interpretation of all the constructions of LAL let us illustrate the definitions using the particular length space $\mathbf{N} = (\mathbb{N}, \Vdash_{\mathbf{N}})$ where $(l,k,g), e \Vdash_{\mathbf{N}} n$ if e encodes n and $l \geq n$ and $k \geq 0$ and $g(x) \geq c$ for c a constant large enough so that $lc \geq |e|$. Note that the constant c may be chosen independent of n. This length space is isomorphic to the denotation in the model of the LAL-type $\forall\alpha.!(\alpha \multimap \alpha) \multimap \S(\alpha \multimap \alpha)$ which is the LAL-version of Church numerals.

Then $(l,k,g), e \Vdash_{\mathbf{N}\otimes\mathbf{N}} (n_1, n_2)$ if $l \geq n_1 + n_2$ and $e = \langle e_1, e_2\rangle$ where e_i encodes n_i and $g(x) \geq c$ and $lg(l+k) \geq |e|$. Note that the latter can be achieved by choosing $g(x) = c + d$ for some fixed constant d.

We see that the diagonal map $n \mapsto (n,n)$ cannot be realized for then we would need a fixed l_0 such that $l_0 + l \geq l + l$ for all l. Similarly, we see that all realisable maps f from $\mathbf{N}$ to $\mathbf{N}$ must satisfy $f(x) \leq x + O(1)$. The runtime of such a function is governed by the third ("g") component of its realiser and is hence an arbitrary polynomial.

On the other hand, the length space $!\mathbf{N}$ has $(l,k,g), e \Vdash_{!\mathbf{N}} n$ if $l \geq 1$ and $k \geq n$ and $g(x) \geq cx^2$. Now note that $lg(k) \geq c(1+k)^2 \geq cn \geq |e|$. On the other hand, since the l-slot (first component) may be chosen 1 we find that the diagonal map $!\mathbf{N} \to !\mathbf{N}\otimes!\mathbf{N}$ is realisable. The identity function from $!\mathbf{N}$ to $\mathbf{N}$ is not realisable because the first component l_0 of its realiser would have to satisfy $l_0 + 1 \geq n$ for all n.

Now consider the length space $\S\mathbf{N}$. We have $(l,k,g), e \Vdash_{\S\mathbf{N}} n$ if $lk \geq n$ and $g(x) \geq cx^2$. We are now able to realise the identity from $!\mathbf{N}$ to $\S\mathbf{N}$ noticing that, in particular $(1, n, \lambda x.cx^2), e \Vdash_{\S\mathbf{N}} n$ when e encodes n. We also note that, for instance the doubling function can be realised as a map from $\mathbf{N}$ to $\S\mathbf{N}$. To do this, we need a realiser with first component $l_0 = 1$. Given input n realised by $(n,1,g)$ we then realise the result $2n$ by $(n, 2, \lambda x.cx^2)$. We can even realise the function $1/4n^2 + O(n)$ but not n^2. For this, two §-s are needed.

Proving light length spaces to be a model for LAL amounts to prove that certain constructions involving the modalities ! and § can be justified in the model. First of all, the diagonal map is a morphism, as can be easily proved:

Lemma 4. *Given light length spaces A, B, there is the morphism: contr $:!A \to !A\otimes!A$ where $contr(a) = (a,a)$.*

Proof. The majoriser for the obvious realiser e_{contr} is given by a suitably padded version of $(2, 0, x \mapsto 0)$. The central part of the verification is the observation that

$$\begin{aligned} 2.(1, l+k, g^+) &= (2, 2(l+k), g^+) \leq_{\mathcal{L}} (2, 0, x \mapsto 0) + (1, l+k, g^+) \\ &= (3, l+k, g^+) \end{aligned}$$

□

On the other hand, ! is a functor.

Lemma 5 (Functoriality of !). *If $f : A \xrightarrow{e,\alpha} B$, then there is β such that $f :!A \xrightarrow{e,\beta}!B$.*

Proof (of Lemma 5). Let α be (n, m, f) and suppose $d, (l, k, g) \Vdash_{!A} a$. Then $(l, k, g) \geq_{\mathcal{L}} (1, p + q, h^+)$, where $d, (p, q, h) \Vdash_A a$. Observe that there must be $(i, j, r), c$ such that $c, (i, j, r) \Vdash_B f(a)$, $(i, j, r) \leq_{\mathcal{L}} (n, m, f) + (p, q, h)$ and $\mathit{Time}(\{e\}(d)) \leq \mathcal{D}_{\mathcal{L}}((i, j, r), (n, m, f) + (p, q, h))$. As a consequence, $c, (1, i + j, r^+) \Vdash_{!B} f(a)$. But

$$(1, i + j, r^+) \leq_{\mathcal{L}} (n + m + 1, m, f^+) + (1, p + q, h^+)$$

because:

- The inequality $1+i+j \leq n+m+2+m \mid (p+q)$ holds, because if $m \leq q$, then $1+i+j \leq 1+n+p+m \mid q = 1+n+p+q \leq 2+n+m+m \mid (p+q)$. and if $m > q$, then $1+i+j \leq 1+n+p+m \mid q = 1+n+p+m \leq 2+n+m+p+q \leq 2+n+m+m \mid (p+q)$.
- For every $x \in \mathbb{N}$,

$$\begin{aligned} r^+(x) &= x^2 r(x^2) \leq x^2 (f \mid h)(x^2) \\ &\leq x^2 (f(x^2) \mid h(x^2)) = (x^2 f(x^2) \mid x^2 h(x^2)) \\ &= (f^+(x) \mid h^+(x)) = (f \mid h)^+(x). \end{aligned}$$

Moreover:

$$\begin{aligned} \mathit{Time}(\{e\}(d)) &\leq \mathcal{D}_{\mathcal{L}}((i, j, r), (n, m, f) + (p, q, h)) \\ &\leq (n + p)(f \mid h)(n + p + m \mid q) \\ &\leq (n + m + 1) \cdot (n + m + 2 + m \mid (p + q))^2 \cdot (f \mid h)((n + m + 2 + m \mid (p + q))^2) \\ &= (n + m + 1) \cdot (f \mid h)^+(n + m + 2 + m \mid (p + q)) \\ &= (n + m + 1) \cdot (f^+ \mid h^+)(n + m + 2 + m \mid (p + q)) \\ &= \mathcal{D}_{\mathcal{L}}((1, i + j, r^+), (n + m + 2, m \mid (p + q), f^+ \mid h^+)) \\ &\leq \mathcal{D}_{\mathcal{L}}((1, i + j, r^+), (n + m + 1, m, f^+) + (1, p + q, h^+)) \end{aligned}$$

This means that $f :!A \xrightarrow{e,(n+m+1,m,f^+)}!B$. □

Notice that the distributive law $!A \otimes !B \multimap !(A \otimes B)$ *cannot* be proved in the syntax and is not validated in the model either. Indeed, its introduction would collapse LAL to elementary affine logic, which is elementary time complete. The modality § is a functor itself:

Lemma 6 (Functoriality of §). *If $f : A \xrightarrow{e,\alpha} B$, then there is β such that $f : \S A \xrightarrow{e,\beta} \S B$.*

Proof (of Lemma 6). Let α be (n, m, f) and suppose $d, (l, k, g) \Vdash_{\S A} a$. Then $(l, k, g) \geq_{\mathcal{L}} (p, q, h^+)$, where $d, (pq, q, h) \Vdash_A a$. Observe that there must be $(i, j, r), c$ such that $c, (i, j, r) \Vdash_B f(a)$, $(i, j, r) \leq_{\mathcal{L}} (n, m, f) + (pq, q, h)$ and $\mathit{Time}(\{e\}(d)) \leq \mathcal{D}_{\mathcal{L}}((i, j, r), (n, m, f) + (pq, q, h))$. As a consequence, we obtain $c, (n, m, f) + (pq, q, h) \Vdash_B f(a)$. But notice that

$$\begin{aligned} (n, m, f) + (pq, q, h) &= (n + pq, m \mid q, f \mid h) \\ \leq_{\mathcal{L}} (((m + 1) \mid q)(n + 1 + p), (m + 1) \mid q, f \mid h). \end{aligned}$$

which implies $c, (n+1+p, (m+1) \mid q, (f \mid h)^+) \Vdash_{\S B} f(a)$. Now:

$$\begin{aligned}(n+1+p, (m+1) \mid q, (f \mid h)^+) &= (n+1, m+1, f^+) + (p, q, h^+) \\ &\leq_{\mathcal{L}} (n+2, m+1, f^+) + (l, k, g).\end{aligned}$$

Moreover:

$$\begin{aligned}Time(\{e\}(d)) &\leq \mathcal{D}_{\mathcal{L}}((i,j,r),(n,m,f) + (pq,q,h)) \\ &\leq (n+pq)(f \mid h)(n+pq+m \mid q) \\ &\leq (n+p+2+q(m+1))^2(f \mid h)((n+p+2+q \mid (m+1))^2) \\ &= (f \mid h)^+(n+p+2+q \mid (m+1)) \\ &= (f^+ \mid h^+)(n+p+2+q \mid (m+1)) \\ &= (p+n+2-(p+n+1))(f^+ \mid h^+)(n+p+2+q \mid (m+1)) \\ &= \mathcal{D}_{\mathcal{L}}((n+1+p,(m+1),(f \mid h)^+),(n+2,m+1,f^+) + (p,q,h^+)) \\ &= \mathcal{D}_{\mathcal{L}}((n+1+p,(m+1),(f \mid h)^+),(n+2,m+1,f^+) + (l,k,g)).\end{aligned}$$

This means that $f : \S A \overset{e,(n+2,m+1,f^+)}{\longrightarrow} \S B$. □

The following lemma whose proof we elide establishes the remaining properties required to model LAL: distributivity of § over $\otimes$ and the dereliction axiom relating the two modalities.

Lemma 7. *Given light length spaces A, B, there are morphisms: derelict* $:!A \to \S A$ *and distr* $: \S A \otimes \S B \to \S(A \otimes B)$ *where, for every $a \in |A|$ and $b \in |B|$, $derelict(a) = a$ and $distr(a,b) = (a,b)$.*

As anticipated in the introduction, a principle which cannot be proved syntactically, *can* be justified in the semantics:

Lemma 8. *Given light length spaces A_1, A_2, there is a morphism codistr* : $\S(A_1 \otimes A_2) \to \S A_1 \otimes \S A_2$ *where $codistr(a_1, a_2) = (a_1, a_2)$.*

Proof. Let $e_{codistr} = e_{id}$. We know $\{e_{id}\}(d)$ takes constant time.

Suppose that $\langle d_1, d_2\rangle, \gamma \Vdash_{\S(A_1 \otimes A_2)} (a_1, a_2)$. We have $\gamma \geq_{\mathcal{L}} (l, k, f^+)$ and $(lk, k, f) \geq_{\mathcal{L}} (l_1, k_1, f_1) + (l_2, k_2, f_2)$ where $d_i, (l_i, k_i, f_i) \Vdash_{A_i} a_i$ for $i = 1, 2$. By upward closure we also have $d_i, (l_i, k, f_i) \Vdash_{A_i} a_i$ and then $d_i, (\lceil l_i/k \rceil, k, f_i^+) \Vdash_{\S A_i} a_i$ and finally $\langle d_1, d_2\rangle, (\lceil l_1/k \rceil + \lceil l_2/k \rceil, k, f^+) \Vdash_{\S A_i \otimes \S A_2} (a_1, a_2)$. But now

$$(\lceil l_1/k \rceil + \lceil l_2/k \rceil, k, f^+) \leq_{\mathcal{L}} (2,0,0) + (l,k,f^+)$$

so that a realiser for $e_{codistr}$ may be given by padding $(2,0,0)$ so as to cover the (constant) runtime of this algorithm.

This shows that light length spaces are not fully complete as a model of LAL. On the other hand, results like Lemma 8 are potentially very interesting, since soundness holds for any extension of LAL which can be interpreted in the model.

4.1 Interpreting Light Affine Logic

Interpretations of the modalities § and ! are the obvious ones: $[\![!A]\!]^{\mathscr{R}}_{\eta} = ![\![A]\!]^{\mathscr{R}}_{\eta}$ and $[\![\S A]\!]^{\mathscr{R}}_{\eta} = \S[\![A]\!]^{\mathscr{R}}_{\eta}$. Since all the axioms needed are justifiable in our semantics, we get:

Theorem 3. *Light length spaces form a model of* LAL.

As a consequence, we can prove that the set of functions which can be represented in LAL is a subset of the class of polytime functions:

Corollary 1 (Soundness). *Let π be an* LAL *proof with conclusion of the form $\vdash \{!,\S\}^j List_{\mathsf{LAL}} \multimap \{!,\S\}^k List_{\mathsf{LAL}}$ and let $f : B \to B$ be the function induced by $[\![\pi]\!]^{\mathscr{R}}$. Then f is computable in polynomial time.*

Proof (Sketch). Intuitively, this is clear since runtimes of realizers are always bounded by the third components of majorizers. A slight complication arises from the fact that the third component of the argument to a map also influences the runtime; indeed, without this feature maps like application from $(A \multimap B) \otimes A$ to B could not be interpreted in the model. However, we can define a light length space of binary lists $\mathbf{B}$ whose realisers have constant third component analogous to the light length space $\mathbf{N}$ in the motivation after Definition 2. We then show using definable iterators that this length space is isomorphic to the denotation of the type B above. It is, however, obvious that all functions on $\mathbf{B}$ are polytime. For details see [11] where such an argument has been carried out in detail for Bounded Linear Logic. □

5 Conclusions

We have introduced a new model for LAL based on realizability. This allows us to give a simplified proof of soundness for the same logic. As any kind of semantics, our model can be used to identify certain axioms as not derivable in LAL (if it's not in the model it can't be in the syntax). Examples of such principles are the identification of the two modalities or commutation of the-modality with tensor. More interestingly, there are formulas which are syntactically not provable but are justified in the semantics. The fact that our semantics has polynomial time computability built-in means that such formulas can be added to LAL without compromising soundness for polynomial time. One example of such a formula has been given in Lemma 8 (distributivity of § over tensor.). We are confident that more examples can be found; of particular interest would be principles that enable more algorithms to be expressed in their natural form.

References

1. Amadio, R.M.: Max-plus quasi-interpretations. In: Proceedings of the 7th International Conference on Typed Lambda Calculi and Applications, pp. 31–45 (2003)
2. Asperti, A., Roversi, L.: Intuitionistic light affine logic. ACM Transactions on Computational Logic 3(1), 137–175 (2002)

3. Bellantoni, S., Niggl, K.H., Schwichtenberg, H.: Higher type recursion, ramification and polynomial time. Annals of Pure and Applied Logic 104, 17–30 (2000)
4. Cook, S., Urquhart, A.: Functional interpretations of feasible constructive arithmetic. Annals of Pure and Applied Logic 63(2), 103–200 (1993)
5. Coppola, P., Martini, S.: Typing lambda terms in elementary logic with linear constraints. In: Proceedings of the 6th International Conference on Typed Lambda-Calculus and Applications, pp. 76–90 (2001)
6. Crossley, J., Mathai, G., Seely, R.: A logical calculus for polynomial-time realizability. Journal of Methods of Logic in Computer Science 3, 279–298 (1994)
7. Lago, U.D., Hofmann, M.: Quantitative models and implicit complexity. In: Proc. Foundations of Software Technology and Theoretical Computer Science, pp. 189–200 (2005)
8. Girard, J.-Y.: Light linear logic. Information and Computation 143(2), 175–204 (1998)
9. Hofmann, M.: Linear types and non-size-increasing polynomial time computation. In: Proceedings of the 14th IEEE Syposium on Logic in Computer Science, pp. 464–473 (1999)
10. Hofmann, M.: Safe recursion with higher types and BCK-algebra. Annals of Pure and Applied Logic 104, 113–166 (2000)
11. Hofmann, M., Scott, P.: Realizability models for BLL-like languages. Theoretical Computer Science 318(1-2), 121–137 (2004)
12. Kreisel, G.: Interpretation of analysis by means of constructive functions of finite types. In: Heyting, A. (ed.) Constructiviey in Mathematics, pp. 101–128. North-Holland, Amsterdam (1959)
13. Lafont, Y.: Soft linear logic and polynomial time. Theoretical Computer Science 318, 163–180 (2004)
14. Lago, U.D., Martini, S.: Phase semantics and decidability of elementary affine logic. Theor. Comput. Sci. 318(3), 409–433 (2004)
15. Murawski, A.S., Luke Ong, C.-H.: Discreet games, light affine logic and ptime computation. In: Clote, P.G., Schwichtenberg, H. (eds.) CSL 2000. LNCS, vol. 1862, pp. 427–441. Springer, Heidelberg (2000)
16. Roversi, L.: A p-time completeness proof for light logics. In: Flum, J., Rodríguez-Artalejo, M. (eds.) CSL 1999. LNCS, vol. 1683, pp. 469–483. Springer, Heidelberg (1999)