Logical Methods in Computer Science
Vol. 6 (4:7) 2010, pp. 1–31
www.lmcs-online.org

Submitted Nov. 16, 2009
Published Dec. 18, 2010

BOUNDED LINEAR LOGIC, REVISITED *

UGO DAL LAGO [a] AND MARTIN HOFMANN [b]

[a] Dipartimento di Scienze dell'Informazione, Università di Bologna
e-mail address: dallago@cs.unibo.it

[b] Institut für Informatik, LMU München
e-mail address: mhofmann@informatik.uni-muenchen.de

Abstract. We present QBAL, an extension of Girard, Scedrov and Scott's bounded linear logic. The main novelty of the system is the possibility of quantifying over resource variables. This generalization makes bounded linear logic considerably more flexible, while preserving soundness and completeness for polynomial time. In particular, we provide compositional embeddings of Leivant's RRW and Hofmann's LFPL into QBAL.

1. Introduction

After two decades from the pioneering works that started it [3, 12, 13], implicit computational complexity is now an active research area at the intersection of mathematical logic and computer science. Its aim is the study of machine-free characterizations of complexity classes. The correspondence between an ICC system and a complexity class holds only *extensionally*, i.e., the class of functions (or problems) which are representable in the system equals the complexity class. Usually, the system is a fragment or subsystem of a larger programming language or logical system, the *base system*, in which other functions besides the ones in the complexity class can be represented. Sometimes, one of the two inclusions is shown by proving that any program (or proof) can be reduced with a bounded amount of resources; in this case, we say that the system is intensionally sound. On the other hand, ICC systems are very far from being *intensionally complete*: there are many programs (or proofs) in the base system which are not in the ICC system, even if they can be evaluated with the prescribed complexity bounds. Observe that this does not contradict extensional completeness, since many different programs or proofs compute the same function.

Of course, a system that captures all and only the programs of the base system running within a prescribed complexity bound will in all but trivial cases (e.g., empty base system) fail to be recursively enumerable. Thus, in practice, one strives to improve intensional expressivity by capturing important classes of examples and patterns.

1998 ACM Subject Classification: F.4.1.
Key words and phrases: linear logic, proof theory, implicit computational complexity.
* This is a revised and extended version of a paper with the same title appeared in the Proceedings of the Ninth International Conference on Typed Lambda Calculi and Applications.

An obstacle towards applying ICC characterizations of complexity classes to programming language theory is their poor intensional expressive power: most ICC systems do not capture natural programs and therefore are not useful in practice. This problem has been already considered in the literature. Some papers try to address the poor intensional expressive power of ICC systems by defining new languages or logics allowing to program in ways which are not allowed in existing ICC systems. This includes quasi-interpretations [14] and LFPL, by the second author [9]. Other papers analyze the intensional expressive power of existing systems either by studying necessary conditions on captured programs or, more frequently, by studying relations between existing ICC systems. One nice example is Murawski and Ong's paper [15], in which the authors prove that a subsystem BC^- of Bellantoni and Cook's function algebra BC [3] can be embedded into light affine logic [1] and that the embedding cannot be extended to the whole BC. In this work, we somehow combine the two approaches, by showing that:

- A new logical system, called Quantified Bounded Affine Logic (QBAL for short) can be defined as a generalization of Girard, Scedrov and Scott's bounded linear logic (BLL, [7]), itself the first characterization of polynomial time computable functions as a fragment of Girard's linear logic [5].
- QBAL is intensionally at least as expressive as two heterogeneous, existing systems, namely Leivant's RRW [13] and LFPL.

Bounded linear logic has received relatively little attention in the past [10, 16]. This is mainly due to its syntax, which is more involved than the one of other complexity-related fragments of linear logic appeared more recently [6, 11, 4]. In bounded linear logic, polynomials are part of the syntax and, as a consequence, computation time is controlled *explicitly*; in other words, BLL cannot be claimed to be a truly implicit characterization of polynomial time. Moreover, it seems that BLL is not expressive enough to embed any existing ICC system corresponding to polynomial time (except Lafont's SLL [11], which anyway was conceived as a very small fragment of BLL).

QBAL is obtained by endowing BLL with bounded quantification on resource variables. In other words, formulas of QBAL includes the ones of BLL, plus formulas like $\exists x : \{x \leq y^2\}.A$ or $\forall x, y : \{x \leq z, y \leq z^3\}.B$. Rules governing bounded quantification can be easily added to BLL, preserving its good properties: QBAL is still a characterization of the polytime functions in an extensional sense. Bounded quantification on resource variables, on the other hand, has tremendous consequences from an intensional point of view: both RRW and LFPL can be compositionally embedded into QBAL. This means, arguably, that programs in either RRW or LFPL can be rewritten in QBAL without major changes, i.e., by mimicking their syntactic structure. Similar results are unlikely to hold for BLL, as argued in sections 6 and 7 below.

Logical systems like QBAL or BLL cannot be considered as practical programming languages, although proofs can be interpreted as programs in the sense of Curry and Howard: the syntax is too complicated and the potential programmer would have to provide quantitative information in the form of polynomials while writing programs. On the other hand, considering BLL or QBAL as type systems for the (linear) lambda calculus is interesting, although type inference would be undecidable in general. In this paper, we advocate the usefulness of QBAL as an intermediate language in which to prove soundness results about other ICC systems.

For all these reasons, QBAL is *not* just another system capturing polynomial time computable functions.

The rest of this paper is organized as follows:

- In Section 2 the syntax of QBAL is introduced, and some interesting properties of the system are proved, together with some examples on how to write programs as proofs of QBAL.
- Section 3 introduces a set-theoretic semantics for QBAL, which is exploited in Section 4 to show polytime soundness of the system.
- Section 5 contains some informal discussion about compositional embeddings.
- Section 6 and 7 present compositional embeddings of LFPL and RRW into QBAL, respectively.

2. Syntax

In this section, we present the syntax of QBAL, together with some of its main properties. In the following, we adhere to the notation adopted in the relevant literature on BLL [7, 10].

2.1. Resource Polynomials and Constraints.

Polynomials appears explicitly in the formulas of QBAL, exactly as in BLL. A specific notation for polynomials was introduced in [7], and will be adopted here. In the following, $\mathbb{N}$ is the set of natural numbers. Sometimes, we use the vector notation $\overline{x}$, which stands for the sequence $x_1, \ldots, x_n$, where n is assumed to be known from the context.

Definition 2.1.

- Given a set X of *resource variables*, a *resource monomial over* X is any finite product of binomial coefficients
$$\prod_{i=1}^{m} \binom{x_i}{n_i}$$
where the resource variables $x_1, \ldots, x_m$ are pairwise distinct and $n_1, \ldots, n_m$ are natural numbers.
- A *resource polynomial over* X is any finite sum of monomials over X. $FV(p)$ denotes the set of resource variables in a resource polynomial p.

Resource polynomials are notations for polynomials with rational coefficients. However, by construction, every resource polynomial maps natural numbers to natural numbers. An example of a resource polynomial on $\{x\}$ (actually a monomial) is
$$\binom{x}{2} = \frac{x \cdot (x-1)}{2!} = \frac{1}{2} \cdot x \cdot (x-1).$$

Resource polynomials satisfy some nice closure properties:

Lemma 2.2. *All constant functions and the identity are resource polynomials. Moreover, resource polynomials are closed under binary sums, binary products, composition, and bounded sums.*

Proof. Every constant function n is simply the resource polynomial

$$\underbrace{1+\ldots+1}_{n \text{ times}} = \underbrace{\binom{x}{0}+\ldots+\binom{x}{0}}_{n \text{ times}}.$$

The identity is the resource polynomial

$$\binom{x}{1}.$$

Closure under binary sums is trivial. To prove closure under products, it suffices to show that the product of two monomials (on the same variable) is a resource polynomial, but this boils down to show that the product of two binomial coefficients $\binom{x}{n}$ and $\binom{x}{m}$ can be expressed itself as a resource polynomial. Actually, if $m \leq n$, then

$$\binom{x}{m}\cdot\binom{x}{n} = \sum_{k=0}^{m}\binom{m+n-k}{k, m-k, n-k}\cdot\binom{x}{m+n-k}$$

where

$$\binom{m+n-k}{k, m-k, n-k}$$

is a multinomial coefficient. The bounded sum

$$\sum_{x<y} p$$

where p is a resource polynomial not mentioning the variable y (but possibly mentioning x) can be formed by observing that

$$\sum_{k=0}^{n}\binom{j}{k} = \binom{n+1}{k+1}.$$

Closure by composition can be proved similarly. □

As a consequence, every polynomial with natural number coefficients is a resource polynomial. The main reason why resource polynomials were originally chosen as a notation for polynomials in BLL [7] was closure under bounded sums, a property which is not true in more traditional notation schemes. We follow the original paper here.

Already in BLL, an order relation on resource polynomials is an essential ingredient in defining the syntax of formulas and proof. In QBAL, the notion is even more important: two polynomials can be compared unconditionally or with an implicit assumption in the form of a set of constraints.

Definition 2.3 (Constraints)**.**

- A *constraint* is an inequality in the form $p \leq q$, where p and q are resource polynomials. A constraint $p \leq q$ *holds* (and we write $\models p \leq q$) if it is true in the standard model. The expression $p < q$ stands for the constraint $p + 1 \leq q$. Variables in p appear negatively, while those in q appear positively in every constraint $p \leq q$.
- A *constraint set* is a finite set of constraints. Constraint sets are denoted with letters like $\mathscr{C}$, $\mathscr{D}$ or $\mathscr{E}$. A constraint $p \leq q$ is a consequence of a constraint set $\mathscr{C}$ (and we write $\mathscr{C} \models p \leq q$) if $p \leq q$ is logical consequence of $\mathscr{C}$. For every constraint sets $\mathscr{C}$ and $\mathscr{D}$, $\mathscr{C} \models \mathscr{D}$ iff $\mathscr{C} \models p \leq q$ for every constraint $p \leq q$ in $\mathscr{D}$.

- For each constraint set $\mathscr{C}$, we define an order $\sqsubseteq_{\mathscr{C}}$ on resource polynomials by imposing $p \sqsubseteq_{\mathscr{C}} q$ iff $\mathscr{C} \models p \leq q$.

Resource polynomials are ordered extensionally: $p \leq q$ holds if p is smaller than q in the standard model of arithmetic. This definition is different from the one from [7] which is weaker but syntactical, defining p to be smaller or equal to q if and only if $q - p$ is itself a resource polynomial. This choice is motivated by the necessity of reasoning about resource polynomials under some assumptions in a constraint set. On the other hand, it has some consequences for the decidability of type checking, discussed in Section 2.4.

The following is a useful technical result about constraints:

Lemma 2.4. *If $x_1, \ldots, x_n$ occur positively (negatively, respectively) in $\mathscr{C}$, $p_i \sqsubseteq_{\mathscr{D}} q_i$ for every i and $\overline{x} \notin FV(\mathscr{D})$, then $\mathscr{C}\{\overline{p}/\overline{x}\} \cup \mathscr{D} \models \mathscr{C}\{\overline{q}/\overline{x}\}$ ($\mathscr{C}\{\overline{q}/\overline{x}\} \cup \mathscr{D} \models \mathscr{C}\{\overline{p}/\overline{x}\}$, respectively).*

Proof. Take any constraint $r \leq t$ in $\mathscr{C}$ and suppose $x_1, \ldots, x_n$ occur positively in $\mathscr{C}$. Then $x_1, \ldots, x_n$ can occur in t but they *cannot* occur in r. So:

$$\begin{aligned}(r \leq t)\{\overline{p}/\overline{x}\} &= r \leq (t\{\overline{p}/\overline{x}\}) \\ (r \leq t)\{\overline{q}/\overline{x}\} &= r \leq (t\{\overline{q}/\overline{x}\})\end{aligned}$$

Now, since $p_i \sqsubseteq_{\mathscr{D}} q_i$ for every i, $t\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{D}} t\{\overline{q}/\overline{x}\}$. As a consequence, $\mathscr{C}\{\overline{p}/\overline{x}\} \cup \mathscr{D} \models \mathscr{C}\{\overline{q}/\overline{x}\}$. Analogously if $x_1, \ldots, x_n$ occur only negatively in $\mathscr{C}$. □

2.2. **Formulas.** Resource polynomials, constraints and constraint sets are the essential ingredients in the definition of QBAL formulas:

Definition 2.5. Formulas of QBAL are defined as follows:

$$\begin{aligned}A ::= \quad & \alpha(p_1, \ldots, p_n) \mid A \otimes A \mid A \multimap A \mid \forall \alpha.A \mid !_{x<p}A \mid \\ & \forall(x_1, \ldots, x_n) : \mathscr{C}.A \mid \exists(x_1, \ldots, x_n) : \mathscr{C}.A\end{aligned}$$

where $x \notin FV(p)$, α ranges over a countable class of atoms (each with an arity n). We will restrict ourselves to *bounded* first order quantification. In other words, whenever we write $\forall(x_1, \ldots, x_n) : \mathscr{C}.A$ or $\exists(x_1, \ldots, x_n) : \mathscr{C}.A$ we implicitly assume that for every i there is a resource polynomial p_i not containing the variables $x_1, \ldots, x_n$ such that $\mathscr{C} \models \{x_1 \leq p_1, \ldots, x_n \leq p_n\}$.

Checking the boundedness condition on formulas is undecidable in general (see Section 2.4 for further discussion). The notions of a free atom or a free resource variable in a formula are defined as usual, keeping in mind that $\forall\overline{x}$, $\exists\overline{x}$ and $!_{x<p}$ act as binders for resource variables, while $\forall\alpha$ acts as a binder for atoms.

Notice that resource polynomials and the variables in them can occur inside constraints, constraint sets and formulas. The following definition becomes natural:

Definition 2.6 (Positive and Negative Occurrences)**.** The definition of a positive (or negative) free occurrence of a variable in a formula A proceeds by induction on A:

- All the variables in $FV(p_1) \cup \ldots \cup FV(p_n)$ occur positively in $\alpha(p_1, \ldots, p_n)$.
- Polarities are propagated through compound formulas by noting that $\multimap$ is negative in the first slot, $!_{x<p}$ is negative in p and $\forall\overline{x}{:}\mathscr{C}$ is negative in $\mathscr{C}$. All other slots are positive. We omit the detailed definition.

For example, the first occurrence of x in $\forall y{:}\{y{\leq}x\}.\beta(z) \multimap \alpha(x,y)$ is negative while the second one is positive; the only occurrence of z is negative; all occurrences of y are bound.

When resource variables occur positively in a formula, one can substitute the formula for an atom in another formula:

Definition 2.7 (Substitution)**.** Let B be a formula where the free variables $x_1,\ldots,x_n$ occur only positively. Then $A\{B/\alpha(x_1,\ldots,x_n)\}$ denotes the formula obtained by replacing every free occurrence of $\alpha(p_1,\ldots,p_n)$ with $B\{p_1/x_1,\ldots,p_n/x_n\}$ inside A.

As an example , if A is $\alpha(p) \multimap \alpha(p+1)$ and B is $\exists x : \{x \leq y\}.\beta(y)$, then $A\{B/\alpha(y)\}$ is

$$(\exists x : \{x \leq p\}.\beta(p)) \multimap (\exists x : \{x \leq p+1\}.\beta(p+1)).$$

Formulas can be compared with respect to assumptions in the form of a constraint set:

Definition 2.8 (Ordering Formulas)**.** The order $\sqsubseteq_{\mathscr{C}}$ on resource polynomials can be extended to an order on formulas as follows:

$$\begin{array}{rcl}
\alpha(p_1,\ldots,p_n) & \sqsubseteq_{\mathscr{C}} & \alpha(q_1,\ldots q_n) \;\; \textit{iff} \;\; \forall i.p_i \sqsubseteq_{\mathscr{C}} q_i \\
A\otimes B & \sqsubseteq_{\mathscr{C}} & C\otimes D \;\; \textit{iff} \;\; A \sqsubseteq_{\mathscr{C}} C \wedge B \sqsubseteq_{\mathscr{C}} D \\
A\multimap B & \sqsubseteq_{\mathscr{C}} & C\multimap D \;\; \textit{iff} \;\; C \sqsubseteq_{\mathscr{C}} A \wedge B \sqsubseteq_{\mathscr{C}} D \\
\forall\alpha.A & \sqsubseteq_{\mathscr{C}} & \forall\alpha.B \;\; \textit{iff} \;\; A \sqsubseteq_{\mathscr{C}} B \\
!_{x<p}A & \sqsubseteq_{\mathscr{C}} & !_{x<q}B \;\; \textit{iff} \;\; q \sqsubseteq_{\mathscr{C}} p \wedge x \notin FV(\mathscr{C}) \wedge A \sqsubseteq_{\mathscr{C}\cup\{x<q\}} B \\
\forall\overline{x} : \mathscr{D}.A & \sqsubseteq_{\mathscr{C}} & \forall\overline{x} : \mathscr{E}.B \;\; \textit{iff} \;\; \mathscr{C}\cup\mathscr{E} \models \mathscr{D} \wedge \overline{x} \notin FV(\mathscr{C}) \wedge A \sqsubseteq_{\mathscr{C}\cup\mathscr{E}} B \\
\exists\overline{x} : \mathscr{D}.A & \sqsubseteq_{\mathscr{C}} & \exists\overline{x} : \mathscr{E}.B \;\; \textit{iff} \;\; \mathscr{C}\cup\mathscr{D} \models \mathscr{E} \wedge \overline{x} \notin FV(\mathscr{C}) \wedge A \sqsubseteq_{\mathscr{C}\cup\mathscr{D}} B
\end{array}$$

Please observe that if $A \sqsubseteq_{\mathscr{C}} B$, then A and B have the same "logical skeleton" and only differ in the corresponding resource polynomials and constraint sets. Resource polynomials in positive position are smaller in A, while those in negative position are smaller in B. Moreover, constraint sets in positive position are stronger in A, while those in negative position are stronger in B. Consider, as an example, the constraint set $\mathscr{C} = \{x \leq y+1\}$. It is easy to check that

$$\forall z : \{z \leq y+2\}.\alpha(y+1) \multimap \alpha(z) \sqsubseteq_{\mathscr{C}} \forall z : \{z \leq x\}.\alpha(z) \multimap \alpha(y+2).$$

Indeed $\mathscr{C} \cup \{z \leq x\} \models \{z \leq y+1\} \models \{z \leq y+2\} = \mathscr{D}$, $z \sqsubseteq_{\mathscr{C}\cup\mathscr{D}} y+2$ and $z \sqsubseteq_{\mathscr{C}\cup\mathscr{D}} y+1$. Intuitively, $\sqsubseteq_{\mathscr{C}}$ can be seen as a subtyping relation such that subtypes of a formula A are those formulas which are "smaller" than A whenever the constraints in $\mathscr{C}$ hold. This is in accordance to, e.g., the way $\sqsubseteq_{\mathscr{C}}$ is defined on implicational formulas, which is very reminiscent of the usual rule defining subtyping for arrow types.

The order relations $\sqsubseteq_{\mathscr{C}}$ satisfy some basic properties:

Lemma 2.9 (Strengthening and Transitivity)**.** *If $A \sqsubseteq_{\mathscr{C}} B$ and $\mathscr{D} \models \mathscr{C}$, then $A \sqsubseteq_{\mathscr{D}} B$. Moreover, if $A \sqsubseteq_{\mathscr{C}} B$ and $B \sqsubseteq_{\mathscr{C}} C$, then $A \sqsubseteq_{\mathscr{C}} C$.*

Proof. Strengthening can be proved by an induction on A. Some cases:

- If $!_{x<p}A \sqsubseteq_{\mathscr{C}} !_{x<q}B$ and $\mathscr{D} \models \mathscr{C}$, then $q \sqsubseteq_{\mathscr{C}} p$, $x \notin FV(\mathscr{C})$, and $A \sqsubseteq_{\mathscr{C}\cup\{x<q\}} B$. By induction hypothesis, $A \sqsubseteq_{\mathscr{D}\cup\{x<q\}} B$. We can assume that $x \notin FV(\mathscr{D})$. Finally, $q \sqsubseteq_{\mathscr{D}} p$. The thesis easily follows.
- If $\forall\overline{x} : \mathscr{D}.A \sqsubseteq_{\mathscr{C}} \forall\overline{x} : \mathscr{E}.B$ and $\mathscr{F} \models \mathscr{C}$, then $\mathscr{C}\cup\mathscr{E} \models \mathscr{D}$, $\overline{x} \notin FV(\mathscr{C})$ and $A \sqsubseteq_{\mathscr{C}\cup\mathscr{E}} B$. Again, we can assume that $\overline{x} \notin FV(\mathscr{C})$. Since $\mathscr{F}\cup\mathscr{E} \models \mathscr{C}\cup\mathscr{E}$, we can apply the induction hypothesis, obtaining $A \sqsubseteq_{\mathscr{F}\cup\mathscr{E}} B$, from which the thesis easily follows.

Transitivity can be handled itself by induction on the structure of A. Some cases:

- If $!_{x<p}A \sqsubseteq_{\mathscr{C}} !_{x<q}B$ and $!_{x<q}B \sqsubseteq_{\mathscr{C}} !_{x<r}C$, then $q \sqsubseteq_{\mathscr{C}} p$, $r \sqsubseteq_{\mathscr{C}} q$, $x \notin FV(\mathscr{C})$, $A \sqsubseteq_{\mathscr{C}\cup\{x<q\}} B$ and $B \sqsubseteq_{\mathscr{C}\cup\{x<r\}} C$. From $A \sqsubseteq_{\mathscr{C}\cup\{x<q\}} B$ and $r \sqsubseteq_{\mathscr{C}} q$, it follows by strengthening that $A \sqsubseteq_{\mathscr{C}\cup\{x<r\}} B$. By the induction hypothesis, $A \sqsubseteq_{\mathscr{C}\cup\{x<r\}} C$. The thesis follows, once we observe that $r \sqsubseteq_{\mathscr{C}} p$ by transitivity.
- If $\forall \overline{x} : \mathscr{D}.A \sqsubseteq_{\mathscr{C}} \forall \overline{x} : \mathscr{E}.B$ and $\forall \overline{x} : \mathscr{E}.B \sqsubseteq_{\mathscr{C}} \forall \overline{x} : \mathscr{F}.C$, then $\mathscr{C} \cup \mathscr{E} \models \mathscr{D}$, $\mathscr{C} \cup \mathscr{F} \models \mathscr{E}$, $\overline{x} \notin FV(\mathscr{C})$, $A \sqsubseteq_{\mathscr{C}\cup\mathscr{E}} B$ and $B \sqsubseteq_{\mathscr{C}\cup\mathscr{F}} C$. Since $\mathscr{C} \cup \mathscr{F} \models \mathscr{C} \cup \mathscr{E}$, $A \sqsubseteq_{\mathscr{C}\cup\mathscr{F}} B$ by strengthening. By the induction hypothesis, $A \sqsubseteq_{\mathscr{C}\cup\mathscr{F}} C$. The thesis follows observing that $\mathscr{C} \cup \mathscr{F} \models \mathscr{D}$.
- If $\exists \overline{x} : \mathscr{D}.A \sqsubseteq_{\mathscr{C}} \exists \overline{x} : \mathscr{E}.B$ and $\exists \overline{x} : \mathscr{E}.B \sqsubseteq_{\mathscr{C}} \exists \overline{x} : \mathscr{F}.C$, then $\mathscr{C} \cup \mathscr{D} \models \mathscr{E}$, $\mathscr{C} \cup \mathscr{E} \models \mathscr{F}$, $\overline{x} \notin FV(\mathscr{C})$, $A \sqsubseteq_{\mathscr{C}\cup\mathscr{D}} B$ and $B \sqsubseteq_{\mathscr{C}\cup\mathscr{E}} C$. Since $\mathscr{C} \cup \mathscr{D} \models \mathscr{C} \cup \mathscr{E}$, $B \sqsubseteq_{\mathscr{C}\cup\mathscr{D}} C$ by strengthening. By the induction hypothesis, $A \sqsubseteq_{\mathscr{C}\cup\mathscr{D}} C$. The thesis follows observing that $\mathscr{C} \cup \mathscr{D} \models \mathscr{F}$.

This concludes the proof. □

Some other auxiliary results about the relations $\sqsubseteq_{\mathscr{C}}$ will be useful in the following. We give them here. First of all, we can perform substitution of resource polynomials for resource variables in formulas being sure that the underlying order is preserved:

Lemma 2.10 (Monotonicity, I)**.** *If A is a formula where the variables $x_1, \ldots, x_n$ occur only positively (negatively, respectively), $p_i \sqsubseteq_{\mathscr{C}} q_i$ for every i and $\overline{x} \notin FV(\mathscr{C})$, then $A\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}} A\{\overline{q}/\overline{x}\}$ ($A\{\overline{q}/\overline{x}\} \sqsubseteq_{\mathscr{C}} A\{\overline{p}/\overline{x}\}$, respectively).*

Proof. By induction on A. Let's just check the most interesting cases:

- If $A = \exists \overline{y} : \mathscr{D}.B$, then the variables in $\overline{y}$ can be assumed to be distinct from $x_1 \ldots, x_n$. Moreover:
$$
\begin{aligned}
A\{\overline{p}/\overline{x}\} &= \exists \overline{y} : \mathscr{D}\{\overline{p}/\overline{x}\}.B\{\overline{p}/\overline{x}\} \\
A\{\overline{q}/\overline{x}\} &= \exists \overline{y} : \mathscr{D}\{\overline{q}/\overline{x}\}.B\{\overline{q}/\overline{x}\}
\end{aligned}
$$
Now, suppose that $x_1, \ldots, x_n$ occur only positively in A. Then, by induction hypothesis, $B\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}} B\{\overline{q}/\overline{x}\}$. Moreover, by Lemma 2.4, $\mathscr{D}\{\overline{p}/\overline{x}\} \cup \mathscr{C} \models \mathscr{D}\{\overline{q}/\overline{x}\}$. By definition, this implies $A\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}} A\{\overline{q}/\overline{x}\}$. Similarly if $x_1, \ldots, x_n$ occur only negatively in A.
- If $A = \forall \overline{y} : \mathscr{D}.B$, then we can proceed exactly in the same way.

This concludes the proof. □

On the other hand, the same polynomial can be substituted in formulas, again preserving the underlying order:

Lemma 2.11 (Monotonicity, II)**.** *If $A \sqsubseteq_{\mathscr{C}} B$ then $A\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}\{\overline{p}/\overline{x}\}} B\{\overline{p}/\overline{x}\}$.*

Proof. By induction on A. Some interesting cases:

- If $A = !_{x<q}C$, then $B = !_{x<r}D$, $r \sqsubseteq_{\mathscr{C}} q$, $x \notin FV(\mathscr{C})$, $C \sqsubseteq_{\mathscr{C}\cup\{x<r\}} D$ and x can be assumed not to appear among the variables in $\overline{x}$ nor the ones in $\overline{p}$. Then:
$$
\begin{aligned}
A\{\overline{p}/\overline{x}\} &= !_{x<q\{\overline{p}/\overline{x}\}}C\{\overline{p}/\overline{x}\} \\
B\{\overline{p}/\overline{x}\} &= !_{x<r\{\overline{p}/\overline{x}\}}D\{\overline{p}/\overline{x}\}
\end{aligned}
$$
By induction hypothesis, $C\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}\{\overline{p}/\overline{x}\}\cup\{x<r\{\overline{p}/\overline{x}\}\}} D\{\overline{p}/\overline{x}\}$. Moreover, $r\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}\{\overline{p}/\overline{x}\}} q\{\overline{p}/\overline{x}\}$. The thesis follows.

- If $A = \exists\overline{y} : \mathscr{D}.C$, then $B = \exists\overline{y} : \mathscr{E}.D$, $\mathscr{C} \cup \mathscr{D} \models \mathscr{E}$, $\overline{y} \notin FV(\mathscr{C})$, $C \sqsubseteq_{\mathscr{C}\cup\mathscr{E}} D$ and, again, variables in $\overline{y}$ can be assumed not to appear among the variables in $\overline{x}$ nor in the ones in $\overline{p}$. Then:
$$\begin{aligned} A\{\overline{p}/\overline{x}\} &= \exists\overline{y} : \mathscr{D}\{\overline{p}/\overline{x}\}.C\{\overline{p}/\overline{x}\} \\ B\{\overline{p}/\overline{x}\} &= \exists\overline{y} : \mathscr{E}\{\overline{p}/\overline{x}\}.D\{\overline{p}/\overline{x}\} \end{aligned}$$
Clearly, $\mathscr{C}\{\overline{p}/\overline{x}\} \cup \mathscr{D}\{\overline{p}/\overline{x}\} \models \mathscr{E}\{\overline{p}/\overline{x}\}$. By the assumptions above, $\overline{y} \notin FV(\mathscr{C}\{\overline{p}/\overline{x}\})$. By the induction hypothesis, $C\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}\{\overline{p}/\overline{x}\}\cup\mathscr{E}\{\overline{p}/\overline{x}\}} D\{\overline{p}/\overline{x}\}$. The thesis follows.

This concludes the proof. □

Finally, substitution of formulas for *atoms* preserves itself the order $\sqsubseteq_{\mathscr{C}}$:

Lemma 2.12 (Monotonicity, III). *If C is a formula where the free variables $x_1, \ldots, x_n$ occur only positively, α is an atom with arity n and $A \sqsubseteq_{\mathscr{C}} B$, then $A\{C/\alpha\} \sqsubseteq_{\mathscr{C}} B\{C/\alpha\}$.*

Proof. By induction on A. Some cases:

- If $A = \alpha(\overline{p})$, then $B = \alpha(\overline{q})$ where $p_i \sqsubseteq_{\mathscr{C}} q_i$, $A\{C/\alpha\} = C\{\overline{p}/\overline{x}\}$ and $B\{C/\alpha\} = C\{\overline{q}/\overline{x}\}$. The thesis easily follows from Lemma 2.10.
- If $A = \forall\beta.D$, then $B = \forall\beta.E$ and we can assume that $\beta \neq \alpha$ and that β does not appear free in C. Then $A\{C/\alpha\} = \forall\beta.(D\{C/\alpha\})$ and $B\{C/\alpha\} = \forall\beta.(E\{C/\alpha\})$. From $A \sqsubseteq_{\mathscr{C}} B$, it follows that $D \sqsubseteq_{\mathscr{C}} E$ and, by induction hypothesis, that $D\{C/\alpha\} \sqsubseteq_{\mathscr{C}} E\{C/\alpha\}$. The thesis easily follows.

This concludes the proof. □

2.3. **Rules.** A QBAL *judgement* is an expression in the form $\Gamma \vdash_{\mathscr{C}} A$, where $\mathscr{C}$ is a constraint set, Γ is a multiset of formulas and A is a formula. The meaning of such an expression is the following: A is a consequence of the formulas in Γ, provided the constraints in $\mathscr{C}$ hold.

Rules of inference for QBAL are in Figure 1. All rules except first order ones are the natural generalizations of BLL rules. In particular, observe that the only rules modifying the underlying constraint set(s) are $P_!$, $R_{\forall x}$, $L_{\forall x}$, $R_{\exists x}$ and $L_{\exists x}$. The multiplicative connectives are governed by the usual rules from intuitionistic linear logic. The modality !, on the other hand, is a functor governed by the following axioms, which come from BLL:
$$\begin{aligned} !_{x<1}A &\multimap A\{1/x\}; \\ !_{x<p+q}A &\multimap !_{x<p}A \otimes !_{y<q}A\{p+y/x\}; \\ !_{x<\sum_{w<p} q}A &\multimap !_{y<p}!_{z<q\{y/w\}}A\{(z+\textstyle\sum_{w<y} q)/x\}. \end{aligned}$$
Moreover, given a constraint set $\mathscr{C}$, it holds that $!_pA \multimap !_qA$ whenever $q \sqsubseteq_{\mathscr{C}} p$. Weakening holds for every formula, contrary to what happens in BLL; as is usual in systems derived from linear logic, this does not break good quantitative properties like polynomial time soundness. Rules W, X, $P_!$, $D_!$ and $N_!$ capture the just described behaviour. Observe how rule $P_!$ allows to take advantage of the inequality $x < q$ in the premise.

First order quantification on resource variables, on the other hand, is governed by the four inference rules $R_{\forall x}$, $L_{\forall x}$, $R_{\exists x}$ and $L_{\exists x}$. Let us consider, as an example, rule $R_{\forall x}$, which can be read as follows: if A can be inferred from Γ provided $\mathscr{C} \cup \mathscr{D}$ hold *and* the variables $\overline{x}$ do not appear in Γ nor in $\mathscr{C}$, then A holds for every value of $\overline{x}$ satisfying $\mathscr{C}$.

Axiom and Cut

$$\frac{A \sqsubseteq_{\mathscr{C}} B}{A \vdash_{\mathscr{C}} B}\,A \qquad \frac{\Gamma \vdash_{\mathscr{C}} A \quad \Delta, A \vdash_{\mathscr{C}} B}{\Gamma, \Delta \vdash_{\mathscr{C}} B}\,U$$

Structural Rules

$$\frac{\Gamma \vdash_{\mathscr{C}} B}{\Gamma, A \vdash_{\mathscr{C}} B}\,W \qquad \frac{\Gamma, !_{x<p}A, !_{y<q}A\{p+y/x\} \vdash_{\mathscr{C}} B \quad p+q \sqsubseteq_{\mathscr{C}} r}{\Gamma, !_{x<r}A \vdash_{\mathscr{C}} B}\,X$$

Multiplicative Logical Rules

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

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

Exponential Rules

$$\frac{A_1, \ldots, A_n \vdash_{\mathscr{C}} B \quad \mathscr{D} \cup \{x < p\} \models \mathscr{C} \quad x \notin FV(\mathscr{D}) \quad p \sqsubseteq_{\mathscr{D}} q_i}{!_{x<q_1}A_1, \ldots, !_{x<q_n}A_n \vdash_{\mathscr{D}} !_{x<p}B}\,P_!$$

$$\frac{A\{1/x\}, \Gamma \vdash_{\mathscr{C}} B \quad 1 \sqsubseteq_{\mathscr{C}} p}{!_{x<p}A, \Gamma \vdash_{\mathscr{C}} B}\,D_!$$

$$\frac{!_{y<p}!_{z<q\{y/w\}}A\{(z + \sum_{w<y} q)/x\}, \Gamma \vdash_{\mathscr{C}} B \quad \sum_{w<p} q \sqsubseteq_{\mathscr{C}} r}{!_{x<r}A, \Gamma \vdash_{\mathscr{C}} B}\,N_!$$

Second Order Rules

$$\frac{\Gamma \vdash_{\mathscr{C}} A \quad \alpha \notin FV(\Gamma)}{\Gamma \vdash_{\mathscr{C}} \forall\alpha.A}\,R_{\forall\alpha} \qquad \frac{\Gamma, A\{B/\alpha(x_1, \ldots, x_n)\} \vdash_{\mathscr{C}} C}{\Gamma, \forall\alpha.A \vdash_{\mathscr{C}} C}\,L_{\forall\alpha}$$

First Order Rules

$$\frac{\Gamma \vdash_{\mathscr{C}\cup\mathscr{D}} A \quad \overline{x} \notin FV(\Gamma) \cup FV(\mathscr{C})}{\Gamma \vdash_{\mathscr{C}} \forall\overline{x} : \mathscr{D}.A}\,R_{\forall x} \qquad \frac{\Gamma, A\{\overline{p}/\overline{x}\} \vdash_{\mathscr{C}} C \quad \mathscr{C} \models \mathscr{D}\{\overline{p}/\overline{x}\}}{\Gamma, \forall\overline{x} : \mathscr{D}.A \vdash_{\mathscr{C}} C}\,L_{\forall x}$$

$$\frac{\Gamma \vdash_{\mathscr{C}} A\{\overline{p}/\overline{x}\} \quad \mathscr{C} \models \mathscr{D}\{\overline{p}/\overline{x}\}}{\Gamma \vdash_{\mathscr{C}} \exists\overline{x} : \mathscr{D}.A}\,R_{\exists x} \qquad \frac{\Gamma, A \vdash_{\mathscr{C}\cup\mathscr{D}} C \quad \overline{x} \notin FV(\Gamma) \cup FV(C) \cup FV(\mathscr{C})}{\Gamma, \exists\overline{x} : \mathscr{D}.A \vdash_{\mathscr{C}} C}\,L_{\exists x}$$

Figure 1: A sequent calculus for QBAL

Notice that BLL can be embedded into QBAL: for every BLL proof $\pi : \Gamma \vdash A$, there is a QBAL proof $\langle\pi\rangle : \Gamma \vdash_{\emptyset} A$: this can be proved by an easy induction on π.

If π is a proof of QBAL, then $|\pi|$ is the number of rule instances in π.

2.4. **On Decidability of Proof Checking.** The problem of checking the correctness of a proof is undecidable in QBAL, since the correctness of a formula is itself an undecidable problem: remember that a formula like $\exists x : \mathscr{C}.A$ is correct only if an inequality $x \leq p$ can be deduced from $\mathscr{C}$ for *some* polynomial p. Moreover, the relation $\models$ between constraint sets is undecidable. This is in contrast to what happens in more implicit ICC systems or in BLL itself, where conditional equality $\sqsubseteq_{\mathscr{C}}$ is replaced by unconditional inequality and $p \leq q$ iff $q - p$ is a resource polynomial.

We do not see undecidability of proof checking as a fundamental problem of QBAL for at least two reasons:

- On the one hand, the main role of QBAL is the one of a metasystem in which to prove quantitative properties of other systems. As a consequence, it is crucial to keep the system as expressive as possible.
- On the other hand, simple, decidable fragments of QBAL can possibly be built by considering decidable, although necessarily incomplete, formal systems for assertions in the form $\mathscr{C} \vdash p \leq q$ (or, more generally, $\mathscr{C} \vdash \mathscr{D}$) and by imposing that bounds on quantified variables are given explicitly when forming existential or universal formulas. This, however, is a topic outside the scope of this paper, which we leave for future work.

The way we define QBAL is, in other words, similar to the one Xi adopts when he introduces Dependent ML [19].

2.5. **QBAL and Second Order Logic.** Second order intuitionistic logic can be presented as a context-independent sequent calculus with explicit structural rules [17], LJ. Rules of LJ are in Figure 2. There is a forgetful map $[\cdot]$ from the space of QBAL proofs to the space of LJ proofs. In particular $\multimap$ corresponds to $\rightarrow$ and $\otimes$ corresponds to $\wedge$. Essentially, $[\pi]$ has the same structure as π, except for exponential and first order rules, which have no formal correspondence in LJ. From our point of view, if $[\pi] = [\rho]$, then π and ρ correspond to the same *program*, i.e. QBAL can be seen as a proper decoration of second order logic proofs with additional information which is not necessary to perform the underlying computation.

Axiom, Cut and Structural Rules

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

Logical Rules

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

Second Order Rules

$$\frac{\Gamma \vdash A \quad \alpha \notin FV(\Gamma)}{\Gamma \vdash \forall \alpha.A}\,R_{\forall} \qquad \frac{\Gamma, A\{B/\alpha\} \vdash C}{\Gamma, \forall \alpha.A \vdash C}\,L_{\forall}$$

Figure 2: A sequent calculus for LJ

2.6. **Properties.** QBAL inherits some nice properties from BLL. In particular, proofs can be manipulated in a uniform way by altering their conclusion without changing their structure, i.e., without changing the underlying second order logic proof.

First of all, a useful transformation is the strengthening of the underlying constraint set $\mathscr{C}$:

Proposition 2.13 (Strengthening)**.** *If $\pi : \Gamma \vdash_{\mathscr{C}} B$ is a proof and $\mathscr{D} \models \mathscr{C}$, then there is a proof $\rho : \Gamma \vdash_{\mathscr{D}} B$ such that $[\rho] = [\pi]$ and $|\rho| \leq |\pi|$.*

Proof. An easy induction on π. As an example, if π consists of an instance of rule A, then the thesis follows from Lemma 2.9. As another example, take a proof $\pi : \Gamma \vdash_{\mathscr{C}} \forall \overline{x} : \mathscr{D}.A$ obtained from $\rho : \Gamma \vdash_{\mathscr{C}\cup\mathscr{E}} A$ applying rule $R_{\forall x}$. We can assume without losing generality that $\overline{x} \notin FV(\mathscr{D})$. From $\mathscr{D} \models \mathscr{C}$, it follows that $\mathscr{D}\cup\mathscr{E} \models \mathscr{C}\cup\mathscr{E}$, from which the thesis easily follows. □

QBAL is monotone with respect to the relation $\sqsubseteq_{\mathscr{C}}$ on formulas.

Proposition 2.14 (Monotonicity). *If $\pi : A_1, \ldots, A_n \vdash_{\mathscr{C}} B$, $B \sqsubseteq_{\mathscr{C}} D$ and $C_i \sqsubseteq_{\mathscr{C}} A_i$ for every $1 \leq i \leq n$, then there is $\rho : C_1, \ldots, C_n \vdash_{\mathscr{C}} D$ such that $[\rho] = [\pi]$.*

Proof. By induction on $|\pi|$. Some interesting cases are the following ones:

- Suppose that π is simply
$$\frac{A \sqsubseteq_{\mathscr{C}} B}{A \vdash_{\mathscr{C}} B}$$
and that $C \sqsubseteq_{\mathscr{C}} A$ and $B \sqsubseteq_{\mathscr{C}} D$. Then, by transitivity of $\sqsubseteq_{\mathscr{C}}$ (see Lemma 2.9), $C \sqsubseteq_{\mathscr{C}} D$ and ρ is
$$\frac{C \sqsubseteq_{\mathscr{C}} D}{C \vdash_{\mathscr{C}} D}\ .$$
- Suppose that π is
$$\frac{A_1, \ldots, A_n \vdash_{\mathscr{C}} B \quad \mathscr{D} \cup \{x < p\} \models \mathscr{C} \quad x \notin FV(\mathscr{D}) \quad p \sqsubseteq_{\mathscr{D}} q_i}{!_{x<q_1}A_1, \ldots, !_{x<q_n}A_n \vdash_{\mathscr{D}} !_{x<p}B}$$
and that $!_{x<r_i}C_i \sqsubseteq_{\mathscr{D}} !_{x<q_i}A_i$ (for every i) and $!_{x<p}B \sqsubseteq_{\mathscr{D}} !_{x<s}D$. By definition, $q_i \sqsubseteq_{\mathscr{D}} r_i$ for every i and $s \sqsubseteq_{\mathscr{D}} p$. By the side condition to the premise of π (and by transitivity of $\sqsubseteq_{\mathscr{D}}$), we obtain $s \sqsubseteq_{\mathscr{D}} r_i$ for every i. Moreover, we have $C_i \sqsubseteq_{\mathscr{D}\cup\{x<q_i\}} A_i$ for every i and $B \sqsubseteq_{\mathscr{D}\cup\{x<s\}} D$. This implies $C_i \sqsubseteq_{\mathscr{D}\cup\{x<s\}} A_i$ for every i, because
$$\mathscr{D} \cup \{x < s\} \models \mathscr{D} \cup \{x < p\} \models \mathscr{D} \cup \{x < q_i\}$$
and by Lemma 2.9. Now, since $\mathscr{D} \cup \{x < s\} \models \mathscr{D} \cup \{x < p\} \models \mathscr{C}$, we can obtain, by Proposition 2.13, a proof σ of $A_1, \ldots, A_n \vdash_{\mathscr{D}\cup\{x<s\}} B$ such that $|\sigma| < |\pi|$. Then, we can easily apply the induction hypothesis on σ and conclude.

This concludes the proof. □

Another useful transformation on proofs is the substitution of resource polynomials for free variables.

Proposition 2.15 (Substitution for Variables). *If $\pi : A_1, \ldots, A_n \vdash_{\mathscr{C}} B$ is a proof and $p_1, \ldots, p_n$ are resource polynomials, then there is a proof*
$$\pi\{\overline{p}/\overline{x}\} : A_1\{\overline{p}/\overline{x}\}, \ldots, A_n\{\overline{p}/\overline{x}\} \vdash_{\mathscr{C}\{\overline{p}/\overline{x}\}} B\{\overline{p}/\overline{x}\}$$
such that $[\pi\{\overline{p}/\overline{x}\}] = [\pi]$.

Proof. By induction on π. An interesting case is the following one:

- Suppose that π is simply
$$\frac{A \sqsubseteq_{\mathscr{C}} B}{A \vdash_{\mathscr{C}} B}\ .$$
Now, if $A \sqsubseteq_{\mathscr{C}} B$ then $A\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}\{\overline{p}/\overline{x}\}} B\{\overline{p}/\overline{x}\}$, by Lemma 2.11. As a consequence,
$$\frac{A\{\overline{p}/\overline{x}\} \sqsubseteq_{\mathscr{C}\{\overline{p}/\overline{x}\}} B\{\overline{p}/\overline{x}\}}{A\{\overline{p}/\overline{x}\} \vdash_{\mathscr{C}\{\overline{p}/\overline{x}\}} B\{\overline{p}/\overline{x}\}}\ .$$

- Suppose that π is

  $$\frac{\rho : \Gamma \vdash_{\mathscr{C}} C\{\overline{q}/\overline{y}\} \quad \mathscr{C} \models \mathscr{D}\{\overline{q}/\overline{y}\}}{\Gamma \vdash_{\mathscr{C}} \exists \overline{y} : \mathscr{D}.C} R_{\exists x}$$

  where, without losing generality, $\overline{y}$ can be chosen as fresh variables not appearing among the ones in $\overline{x}$, nor in the ones in $\overline{p}$. Applying the induction hypothesis to ρ, we obtain a proof of $\Gamma\{\overline{p}/\overline{x}\} \vdash_{\mathscr{C}\{\overline{p}/\overline{x}\}} C\{\overline{q}/\overline{y}\}\{\overline{p}/\overline{x}\}$. But, by the assumptions above,

  $$C\{\overline{q}/\overline{y}\}\{\overline{p}/\overline{x}\} = C\{\overline{p}/\overline{x}\}\{(\overline{q}\{\overline{p}/\overline{x}\})/\overline{y}\}.$$

  Analogously, from $\mathscr{C} \models \mathscr{D}\{\overline{q}/\overline{y}\}$, it follows that $\mathscr{C}\{\overline{p}/\overline{x}\} \models \mathscr{D}\{\overline{q}/\overline{y}\}\{\overline{p}/\overline{x}\}$ and

  $$\mathscr{D}\{\overline{q}/\overline{y}\}\{\overline{p}/\overline{x}\} = \mathscr{D}\{\overline{p}/\overline{x}\}\{(\overline{q}\{\overline{p}/\overline{x}\})/\overline{y}\}.$$

  The thesis follows.
- Suppose that π is

  $$\frac{\rho : \Gamma, C \vdash_{\mathscr{C} \cup \mathscr{D}} D \quad \overline{y} \notin FV(\Gamma) \cup FV(D) \cup FV(\mathscr{C})}{\Gamma, \exists \overline{y} : \mathscr{D}.C \vdash_{\mathscr{C}} D} L_{\exists x}$$

  where, as usual, $\overline{y}$ can be chosen as fresh variables not appearing among the ones in $\overline{x}$, nor in $\overline{p}$. Applying the induction hypothesis to ρ, we obtain a proof of

  $$\Gamma\{\overline{p}/\overline{x}\}, C\{\overline{p}/\overline{x}\} \vdash_{\mathscr{C}\{\overline{p}/\overline{x}\} \cup \mathscr{D}\{\overline{p}/\overline{x}\}} D\{\overline{p}/\overline{x}\}.$$

  By the assumptions above, variables in $\overline{y}$ do not appear free in $\Gamma\{\overline{p}/\overline{x}\}$ nor in $D\{\overline{p}/\overline{x}\}$ nor in $\mathscr{C}\{\overline{p}/\overline{x}\}$. The thesis follows.

This concludes the proof. □

But formulas themselves can be substituted (for atoms) into a proof:

Proposition 2.16 (Substitution for Atoms)**.** *If $\pi : A_1, \ldots, A_n \vdash_{\mathscr{C}} B$ is a proof, C is a formula where the free variables $x_1, \ldots, x_m$ occur only positively and α is an atom with arity m, then there is a proof $\pi\{C/\alpha\} : A_1\{C/\alpha\}, \ldots, A_n\{C/\alpha\} \vdash_{\mathscr{C}} B\{C/\alpha\}$ such that $[\pi\{C/\alpha\}] = [\pi]$.*

Proof. By induction on π. Some interesting cases are the following ones:

- Suppose, again, that π is simply

  $$\frac{A \sqsubseteq_{\mathscr{C}} B}{A \vdash_{\mathscr{C}} B}$$

  Now, if $A \sqsubseteq_{\mathscr{C}} B$ then $A\{C/\alpha\} \sqsubseteq_{\mathscr{C}} B\{C/\alpha\}$ by Lemma 2.12. As a consequence:

  $$\frac{A\{C/\alpha\} \sqsubseteq_{\mathscr{C}} B\{C/\alpha\}}{A\{C/\alpha\} \vdash_{\mathscr{C}} B\{C/\alpha\}}$$

- Suppose that π is

  $$\frac{\rho : \Gamma \vdash_{\mathscr{C}} D\{\overline{q}/\overline{y}\} \quad \mathscr{C} \models \mathscr{D}\{\overline{q}/\overline{y}\}}{\Gamma \vdash_{\mathscr{C}} \exists \overline{y} : \mathscr{D}.D} R_{\exists x}$$

  where, without losing generality, $\overline{y}$ can be chosen as fresh variables not appearing among the ones in $\overline{x}$ nor in C. Applying the induction hypothesis to ρ, we obtain a proof of $\Gamma\{C/\alpha\} \vdash_{\mathscr{C}} D\{\overline{q}/\overline{y}\}\{C/\alpha\}$. But, by the assumptions above,

  $$D\{\overline{q}/\overline{y}\}\{C/\alpha\} = D\{C/\alpha\}\{\overline{q}/\overline{y}\}.$$

  The thesis follows.

- Suppose that π is

$$\frac{\rho : \Gamma, D \vdash_{\mathscr{C}\cup\mathscr{D}} E \quad \overline{y} \notin FV(\Gamma) \cup FV(E) \cup FV(\mathscr{C})}{\Gamma, \exists\overline{y} : \mathscr{D}.D \vdash_{\mathscr{C}} E}\ L_{\exists x}$$

  where, as usual, $\overline{y}$ can be chosen as fresh variables not appearing among the ones in $\overline{x}$, nor in C. Applying the induction hypothesis to ρ, we obtain a proof of

$$\Gamma\{C/\alpha\}, D\{C/\alpha\} \vdash_{\mathscr{C}\cup\mathscr{D}} E\{C/\alpha\}.$$

  By the assumptions above, variables in $\overline{y}$ do not appear free in $\Gamma\{C/\alpha\}$ nor in $E\{C/\alpha\}$ nor in $\mathscr{C}\{C/\alpha\}$: they are either in Γ, E, $\mathscr{C}$ or in C. The thesis follows.

This concludes the proof. □

2.7. **Cut-Elimination.** A nice application of the results we have just given is cut-elimination. Indeed, the new rules $R_{\forall x}$, $L_{\forall x}$, $R_{\exists x}$ and $L_{\exists x}$ do not cause any problem in the cut-elimination process. For example, the cut

$$\frac{\dfrac{\pi : \Gamma \vdash_{\mathscr{C}\cup\mathscr{D}} A \quad \overline{x} \notin FV(\Gamma) \cup FV(\mathscr{C})}{\Gamma \vdash_{\mathscr{C}} \forall\overline{x} : \mathscr{D}.A}\ R_{\forall x} \qquad \dfrac{\Delta, A\{\overline{p}/\overline{x}\} \vdash_{\mathscr{C}} C \quad \mathscr{C} \models \mathscr{D}\{\overline{p}/\overline{x}\}}{\Delta, \forall\overline{x} : \mathscr{D}.A \vdash_{\mathscr{C}} C}\ L_{\forall x}}{\Gamma, \Delta \vdash_{\mathscr{C}} C}\ U$$

can be eliminated as follows:

$$\frac{\sigma : \Gamma \vdash_{\mathscr{C}} A\{\overline{p}/\overline{x}\} \quad \rho : \Delta, A\{\overline{p}/\overline{x}\} \vdash_{\mathscr{C}} C}{\Gamma, \Delta \vdash_{\mathscr{C}} C}\ U$$

where σ is obtained by applying Proposition 2.13 to

$$\pi\{\overline{p}/\overline{x}\} : \Gamma \vdash_{\mathscr{C}\cup\mathscr{D}\{\overline{p}/\overline{x}\}} A\{\overline{p}/\overline{x}\},$$

itself obtained from π applying Proposition 2.15. In this paper, we will not study cut-elimination. And polynomial time soundness will be itself proved semantically.

2.8. **Programming in QBAL.** The Curry-Howard correspondence allows to see BLL and QBAL as programming languages endowed with rich type systems. In particular, following the usual impredicative encoding of data into second order intuitionistic logic, natural numbers can be represented as cut-free proofs of the formula

$$\mathbf{N}_p = \forall\alpha.!_{y<p}(\alpha(y) \multimap \alpha(y+1)) \multimap \alpha(0) \multimap \alpha(p).$$

However, only natural numbers less or equal to p are representable this way.

This can be generalized to any word algebra. Given a word algebra $\mathbb{W}$, we will denote by $\varepsilon_{\mathbb{W}}$ the only 0-ary constructor of $\mathbb{W}$ and by $c^1_{\mathbb{W}}, \ldots, c^w_{\mathbb{W}}$ the 1-ary constructors of the same algebra. Notice that these objects can be thought of both as term formers and as (0-ary or unary) functions on terms. Terms of a free algebra $\mathbb{W}$ of length at most p can be represented as cut-free proofs of the formula

$$\mathbf{W}_p = \underbrace{\forall\alpha.!_{y<p}(\alpha(y) \multimap \alpha(y+1)) \multimap \ldots \multimap !_{y<p}(\alpha(y) \multimap \alpha(y+1))}_{w \text{ times}} \multimap \alpha(0) \multimap \alpha(p).$$

Functions on natural numbers can be represented by proofs with conclusion $\mathbf{N}_x \vdash \mathbf{N}_p$, where p is a resource polynomial depending on x, only. More generally, functions on the word algebra $\mathbb{W}$ can be represented by proofs with conclusion $\mathbf{W}_x \vdash \mathbf{W}_p$. For example, all constructors $c^1_{\mathbb{W}}, \ldots, c^w_{\mathbb{W}}$ correspond to proofs with conclusion $\mathbf{W}_x \vdash \mathbf{W}_{x+1}$, while $\varepsilon_{\mathbb{W}}$

corresponds to a proof of $\vdash \mathbf{W}_0$. More generally, the polynomial p gives a bound on the size of the result, as a function of the size of the input. QBAL supports iteration on any word algebra (including natural numbers). As an example, for every p and for every A where x only appears positively, there is a proof π_p^A of

$$\mathbf{N}_p, !_{y<p}(A\{y/x\} \multimap A\{y+1/x\}), A\{0/x\} \vdash A\{p/x\}.$$

namely:

$$\dfrac{\dfrac{\dfrac{}{!_{y<p}(A\{y/x\} \multimap A\{y+1/x\}) \vdash !_{y<p}(A\{y/x\} \multimap A\{y+1/x\}}\,A \quad \dfrac{}{A\{0/x\} \vdash A\{0/x\}}\,A \quad \dfrac{}{A\{p/x\} \vdash A\{p/x\}}\,A}{!_{y<p}(A\{y/x\} \multimap A\{y+1/x\}) \multimap A\{0/x\} \multimap A\{p/x\}, !_{y<p}(A\{y/x\} \multimap A\{y+1/x\}), A\{0/x\} \vdash A\{p/x\}}\,L_{\multimap}}{\mathbf{N}_p, !_{y<p}(A\{y/x\} \multimap A\{y+1/x\}), A\{0/x\} \vdash A\{p/x\}}\,L_{\forall\alpha}$$

This will be essential to prove Lemma 7.1.

2.9. **Unbounded First Order Quantification is Unsound.** One may wonder why quantification on numerical variables is restricted to be bounded (see Definition 2.5). The reason is very simple: in presence of unbounded quantification, QBAL would immediately become unsound. To see that, define $\mathbf{N}_\infty$ to be the formula $\exists(x) : \emptyset.\mathbf{N}_x$. The composition of the successor with itself yields a proof with conclusion $\mathbf{N}_x \vdash \mathbf{N}_{x+2}$ which, by rules $R_{\exists x}$ and $L_{\exists x}$, becomes a proof with conclusion $\mathbf{N}_\infty \vdash \mathbf{N}_\infty$. Iterating it, we obtain a proof of $\mathbf{N}_x \vdash \mathbf{N}_\infty$ which represents the function $n \mapsto 2n$. But by rule $L_{\exists x}$, it can be turned into a proof of $\mathbf{N}_\infty \vdash \mathbf{N}_\infty$, and iterating it again we obtain a proof representing the exponential function. The boundedness assumption will be indeed critical in Section 4, where we establish that any functions which is representable in QBAL is polynomial time computable.

It is not clear whether unbounded existential quantification would be sufficient to embed the whole of second order intuitionistic logic into QBAL.

3. Set-Theoretic Semantics

In this Section, we give a set-theoretic semantics for QBAL. We assume that our ambient set-theory is constructive. This way we have a set of sets $\mathcal{U}$ which contains the natural numbers, closed under binary products, function spaces and $\mathcal{U}$-indexed products. An alternative to assuming a constructive ambient set theory consists of replacing plain sets with PERs (partial equivalence relations) or domains or similar structures. See [10] for a more detailed discussion on this issue.

Formulas of QBAL can be interpreted as sets as follows, where ρ is an environment mapping atoms to sets:

$$\begin{aligned}
[\![\alpha(\overline{p})]\!]_\rho &= \rho(\alpha);\\
[\![A \otimes B]\!]_\rho &= [\![A]\!]_\rho \times [\![B]\!]_\rho;\\
[\![A \multimap B]\!]_\rho &= [\![A]\!]_\rho \Rightarrow [\![B]\!]_\rho;\\
[\![\forall\alpha.A]\!]_\rho &= \prod_{C\in\mathcal{U}} [\![A]\!]_{\rho[\alpha\mapsto C]};\\
[\![!_{x<p}A]\!]_\rho = [\![\forall\overline{x} : \mathscr{C}.A]\!]_\rho = [\![\exists\overline{x} : \mathscr{C}.A]\!]_\rho &= [\![A]\!]_\rho.
\end{aligned}$$

Please observe that the interpretation of any formula A is completely independent from the resource polynomials appearing in A.

To any QBAL proof π of $A_1, \ldots, A_n \vdash_{\mathscr{C}} B$ we can associate a set-theoretic function $[\![\pi]\!]_\rho : [\![A_1 \otimes \ldots \otimes A_n]\!]_\rho \to [\![B]\!]_\rho$ by induction on π, in the obvious way. $[\![\pi]\!]_\rho$ is *equal* to the set-theoretic semantics of $[\pi]$ as a proof of second order intuitionistic logic. Set-theoretic semantics of proofs is preserved by cut-elimination: if π reduces to σ by cut-elimination, then $[\![\pi]\!]_\rho = [\![\sigma]\!]_\rho$.

Observe that $[\![A]\!]_\rho$ only depends on the values of ρ on atoms appearing free in A. So, in particular,

$$[\![\mathbf{N}_q]\!]_\rho = \prod_{C \in \mathcal{U}} (C \Rightarrow C) \Rightarrow (C \Rightarrow C)$$

is independent on ρ and on q, since $\mathbf{N}_q$ is a closed formula. Similarly for $[\![\mathbf{W}_q]\!]_\rho$. Actually, there are functions $\varphi_{\mathbb{N}} : \mathbb{N} \to [\![\mathbf{N}_p]\!]$ and $\psi_{\mathbb{N}} : [\![\mathbf{N}_p]\!] \to \mathbb{N}$ such that $\psi_{\mathbb{N}} \circ \varphi_{\mathbb{N}}$ is the identity on natural numbers. They are defined as follows:

$$\begin{aligned} (\varphi_{\mathbb{N}}(n))_C(f, z) &= f^n(z); \\ \psi_{\mathbb{N}}(x) &= x_{\mathbb{N}}(x \mapsto x+1)(0); \end{aligned}$$

where A_C is the projection of A on the component C whenever A is some product $\prod_{D \in \mathcal{U}} B$.

So, given a proof $\pi : \mathbf{N}_x \vdash \mathbf{N}_p$, the *numeric function represented by* π is simply $\psi_{\mathbb{N}} \circ [\![\pi]\!] \circ \varphi_{\mathbb{N}}$. Similar arguments hold for functions with conclusion $\mathbf{W}_x \vdash \mathbf{W}_p$.

4. QBAL and Polynomial Time

In this section we show that all functions on natural numbers definable in QBAL are polynomial time computable. To this end, we follow the semantic approach in [10] which we now summarizes.

4.1. **Realizability Sets.** Let X be a finite set of resource variables. We write $\mathcal{V}(X)$ for $\mathbb{N}^X$ — the elements of $\mathcal{V}(X)$ are called *valuations* (over X). If $\eta \in \mathcal{V}(X)$ and $c \in \mathbb{N}$ then $\eta[x \mapsto c]$ denotes the valuation which maps x to c and acts like η otherwise. We assume some reasonable encoding of valuations as natural numbers allowing them to be passed as arguments to algorithms.

If $\mathscr{C}$ is a constraint set involving at most the variables in X (*over* X) then $\mathcal{V}_{\mathscr{C}}(X)$ (or simply $\mathcal{V}_{\mathscr{C}}$) is the set of valuations in $\mathcal{V}(X)$ satisfying all the constraints in $\mathscr{C}$.

We write $\mathcal{P}(X)$ for the set of resource polynomials over X. If $p \in \mathcal{P}(X)$ and $\eta \in \mathcal{V}(X)$ we write $p(\eta)$ for the number obtained by evaluating p with $x \mapsto \eta(x)$ for each $x \in X$. A *substitution* $\sigma : X \to Y$ is a function mapping any variable in Y to a polynomial in $\mathcal{P}(X)$. Given a substitution $\sigma : X \to Y$ and a valuation $\eta \in \mathcal{V}(X)$, the valuation $\sigma[\eta] \in \mathcal{V}(Y)$ assigns to every variable $y \in Y$ the natural number $\sigma(y)(\eta)$.

We assume known the untyped lambda calculus as defined e.g. in [2]. A lambda term is *affine linear* if each variable (free or bound) appears at most once (up to α-congruence). For example, $\lambda x.\lambda y.yx$ and $\lambda x.\lambda y.y$ and $\lambda x.xy$ are affine linear while the term $\lambda x.xx$ is not. Notice that every affine linear term t is strongly normalisable in less than $|t|$ steps where $|t|$ is the size of the term. Moreover, the size $|s|$ of any reduct of t is at most $|t|$. The runtime of the computation leading to the normal form is therefore $O(|t|^2)$. We will henceforth use the expression *affine linear term* for an affine linear lambda term which is in normal form. If s, t are affine linear terms, then their application st is defined as the normal form of the lambda term st. Notice that the application st can be computed in time $O((|s| + |t|)^2)$.

If s, t are affine linear terms we write $s \otimes t$ for the affine linear term $\lambda f.fst$. If t is an affine linear term possibly containing the free variables x, y then we write $\lambda x \otimes y.t$ for $\lambda u.u(\lambda x \lambda y.t)$. Notice that $(\lambda x \otimes y.t)(u \otimes v) = t\{u/x, v/y\}$.

More generally, if $(t_i)_{i<n}$ is a family of affine linear terms, we write $\bigotimes_{i<n} t_i$ and $\lambda \bigotimes_{i<n} x_i.t$ for $\lambda f.ft_0t_1 \ldots t_{n-1}$, respectively, $\lambda u.u(\lambda x_0 \lambda x_1 \ldots \lambda x_{n-1}.t)$. Again,

$$(\lambda \bigotimes_{i<n} x_i.t)(\bigotimes_{i<n} t_i) = t\{t_0/x_0, \ldots, t_{n-1}/x_{n-1}\}.$$

We write Λ_a for the set of closed affine linear terms.

There is a canonical way of representing terms of any word algebra $\mathbb{W}$ as affine linear terms, which is attributed to Dana Scott [18]. If the unary constructors of the word algebra $\mathbb{W}$ are $c^1_{\mathbb{W}}, \ldots, c^w_{\mathbb{W}}$ and $\varepsilon_{\mathbb{W}}$ is the only 0-ary constructor of $\mathbb{W}$, the terms of $\mathbb{W}$ are mapped to affine linear terms as follows:

$$\begin{aligned} \ulcorner \varepsilon_{\mathbb{W}} \urcorner &= \lambda x_1. \ldots \lambda x_w.\lambda y.y; \\ \forall i \in \{1, \ldots, w\}.\ulcorner c^i_{\mathbb{W}} s \urcorner &= \lambda x_1. \ldots \lambda x_w.\lambda y.x_i \ulcorner s \urcorner. \end{aligned}$$

As an example, the natural number 2 seen as a term of the word algebra $\mathbf{N}$ becomes

$$\ulcorner 2 \urcorner = \lambda x.\lambda y.x(\lambda x.\lambda y.x(\lambda x.\lambda y.y))$$

Definition 4.1 (Realizability Set)**.** Let X be a finite set of resource variables. A *realizability set over* X is a pair $A = (|A|, \Vdash_A)$ where $|A|$ is a set and $\Vdash_A \subseteq \mathcal{V}(X) \times \Lambda_a \times |A|$ is a ternary relation between valuations over X, affine lambda terms, and the set $|A|$. We write $\eta, t \Vdash_A a$ for $(\eta, t, a) \in \Vdash_A$. Given a substitution σ from X to Y and a realizability set A over Y, then a new realizability set $A[\sigma]$ over X is defined by $|A[\sigma]| = |A|$ and $\eta, t \Vdash_{A[\sigma]} a$ iff $\sigma[\eta], t \Vdash_A a$.

The intuition behind $\eta, t \Vdash_A a$ is that a is an abstract semantic value, η measures the abstract size of a, and the affine linear term t encodes the abstract value a. This is a generalization of what normally happens in realizability models, where $\Vdash_A$ is a *binary* relation between realizers and denotations.

Example 4.2.

- The realizability set $\mathbf{N}_x$ over $\{x\}$ of *tally natural numbers* ("of size at most x") is defined by: $|\mathbf{N}_x| = \mathbb{N}$ and $\eta, t \Vdash_{\mathbf{N}_x} n$ if $t = \ulcorner n \urcorner$ and $\eta(x) \geq n$;
- The realizability set $\mathbf{W}_x$ over $\{x\}$ of *free terms of* $\mathbb{W}$ ("of length at most x") is defined by: $|\mathbf{W}_x| = \mathbb{W}$ and $\eta, t \Vdash_{\mathbf{W}_x} w$ if $t = \ulcorner w \urcorner$ and $\eta(x) \geq |w|$.

These realizability sets $\mathbf{N}_x$ and $\mathbf{W}_x$ turn out to be retracts of the denotations of the eponymous BLL formulas from Section 2.8.

Definition 4.3 (Positive and Negative Variables)**.** Let A be a realizability set over X. We say that $x \in X$ is *positive* (*negative*, respectively) in A, if for all $\eta, \mu \in \mathcal{V}(X), t \in \Lambda_a, a \in |A|$ where η and μ agree on $X \setminus \{x\}$ and $\eta(x) \leq \mu(x)$ ($\eta(x) \geq \mu(x)$, respectively), $\eta, t \Vdash_A a$ implies $\mu, t \Vdash_A a$.

We notice that x is positive in $\mathbf{N}_x$ and $\mathbf{W}_x$. Indeed, if e.g. $\eta, t \Vdash_{\mathbf{N}_x} n$ and $\eta(x) \leq \mu(x)$, then $\mu(x) \geq \eta(x) \geq n$ and $\mu, t \Vdash_{\mathbf{N}_x} n$ by definition.

Realizability sets can be thought of as the object of a category whose arrows are functions, themselves realized by affine linear terms, one for every possible valuation of the underlying resource variables:

Definition 4.4 (Morphisms)**.** Let A, B be realizability sets over some set X. A *morphism* from A to B is a function $f : |A| \to |B|$ satisfying the following conditions:

- there exist a function $e : \mathcal{V}(X) \to \Lambda_a$, an algorithm $\mathcal{A}$ and a resource polynomial q such that for every $\eta \in \mathcal{V}(X)$, $\mathcal{A}$ computes $e(\eta)$ in time bounded by $q(\eta)$;
- for each $\eta \in \mathcal{V}(X)$, $t \in \Lambda_a$, $a \in |A|$, we have

$$\eta, t \Vdash_A a \quad \text{implies} \quad \eta, e(\eta)t \Vdash_B f(a).$$

In this case we say that e *witnesses* f and write $A \to^f_e B$ where in the notation the algorithm $\mathcal{A}$ computing e is presumed to exist.

Noticeably, morphisms compose.

The following definitions summarises the interpretation of formulas according to [10]. First of all, multiplicative connectives $\otimes$ and $\multimap$ correspond to constructions on realizability sets:

Definition 4.5 (Multiplicatives)**.** Let A, B be realizability sets over X. Then the following are realizability sets over X:

- $A \otimes B$ as given by $|A \otimes B| = |A| \times |B|$ and $\eta, t \Vdash_{A\otimes B} (a, b)$ iff $t = u \otimes v$, where $\eta, u \Vdash_A a$ and $\eta, v \Vdash_B b$.
- $A \multimap B$ is given by $|A \multimap B| = |A| \Rightarrow |B|$ and $\eta, t \Vdash_{A\multimap B} f$ iff whenever $\eta, u \Vdash_A a$ it holds that $\eta, t\, u \Vdash_B f(a)$.

Another logical connective needs to be justified, namely the exponential modality:

Definition 4.6 (Modality)**.** If C is a realizability set over $X \cup \{x\}$ and $p \in \mathcal{P}(X)$ then a realizability set $!_{x<p}C$ over X is defined by $|!_{x<p}C| = |C|$ and $\eta, t \Vdash_{!_{x<p}C} a$ if

- $t = \bigotimes_{i<p(\eta)} t_i$ for some family $(t_i)_{i<p(\eta)}$;
- $\eta[x \mapsto i], t_i \Vdash_C a$ for each $i < p(\eta)$.

Lastly, a semantical counterpart of second order universal quantification must be defined. The following are essential preliminary definitions.

Definition 4.7 (Second Order Environments)**.** Let X be a set of resource variables. A *second-order environment over* X is a partial function ρ which assigns to a second-order variable α of arity n a pair (l, C) such that:

- $l = (y_1, \dots, y_n)$ is an n-tuple of pairwise different resource variables not occurring in X;
- C is a realizability set over $X \cup \{y_1, \dots, y_n\}$ in which the y_i are positive.

For a second-order environment ρ we write $|\rho|$ for the mapping $\alpha \mapsto |C|$ when $\rho(\alpha) = (l, C)$. If $\sigma : X \to Y$ is a substitution and ρ is a second-order environment over Y we define a second-order environment $\rho[\sigma]$ over X by $\rho[\sigma](\alpha) = (l, C[\sigma])$ when $\rho(\alpha) = (l, C)$. We assume here that the variables in l are not contained in Y. Otherwise, the substitution cannot be defined.

Using these semantic constructions one defines for each formula A with free resource variables contained in X and second-order environment ρ over X, a realizability set $[\![A]\!]^{\mathcal{B}}_\rho$ over X in such a way that $|[\![A]\!]^{\mathcal{B}}_\rho| = [\![A]\!]_{|\rho|}$ (where $|\rho|$ is the assignment of sets to atoms obtained from ρ in the obvious way), that is to say, the underlying set of the realizability set interpreting a formula A coincides with the set-theoretic meaning of A (see Section 3):

$$[\![\alpha(p_1, \dots, p_n)]\!]^{\mathcal{B}}_\rho = C[\sigma],$$

where $\rho(\alpha) = ((y_1, \ldots, y_n), C)$ and $\sigma(y_i) = p_i$;

$$\llbracket A \otimes B \rrbracket^{\mathcal{B}}_{\rho} = \llbracket A \rrbracket^{\mathcal{B}}_{\rho} \otimes \llbracket B \rrbracket^{\mathcal{B}}_{\rho};$$
$$\llbracket A \multimap B \rrbracket^{\mathcal{B}}_{\rho} = \llbracket A \rrbracket^{\mathcal{B}}_{\rho} \multimap \llbracket B \rrbracket^{\mathcal{B}}_{\rho};$$
$$\llbracket \forall\alpha.A \rrbracket^{\mathcal{B}}_{\rho} = \left(\prod_{C \in \mathcal{U}} \llbracket A \rrbracket_{|\rho|[\alpha \mapsto C]}, \Vdash_{\llbracket \forall\alpha.A \rrbracket^{\mathcal{B}}_{\rho}} \right);$$

where $\eta, t \Vdash_{\llbracket \forall\alpha.A \rrbracket^{\mathcal{B}}_{\rho}} f$ iff $\eta, t \Vdash_{\llbracket A \rrbracket^{\mathcal{B}}_{\rho[\alpha \mapsto (l,C)]}} f_C$ for all (l, C);

$$\llbracket !_{x<p} A \rrbracket^{\mathcal{B}}_{\rho} = !_{x<p} \llbracket A \rrbracket^{\mathcal{B}}_{\rho[\xi_x]},$$

where $\xi_x : X \cup \{x\} \to X$ is the substitution mapping any variable in X into the same variable as an element of $\mathcal{P}(X \cup \{x\})$.

The main result of [10] then asserts that if π is a proof (in BLL) of a sequent $\Gamma \vdash B$ then the function $\llbracket \pi \rrbracket_{|\rho|}$ is a morphism from $\llbracket \Gamma \rrbracket^{\mathcal{B}}_{\rho}$ to $\llbracket B \rrbracket^{\mathcal{B}}_{\rho}$ (where we interpret a context Γ as a $\otimes$-product over its components as usual). From this, polynomial time soundness is a direct corollary since polynomial time computability is built into the notion of a morphism.

It thus only remains to extend the realizability model to cover the constructs of QBAL which we do in the next section.

4.2. **Extending the Realizability Model to QBAL.** The notion of a realizability set above is adequate to model formulas of QBAL. The notion of a morphism, however, should be slightly generalized in order to capture constraints:

Definition 4.8 ($\mathscr{C}$-Morphisms)**.** Let A, B be realizability sets over some set X and $\mathscr{C}$ a constraint set over X. A function $f : |A| \to |B|$ is a $\mathscr{C}$-morphism from A to B iff the following conditions hold:

- there exist a function $e : \mathcal{V}_{\mathscr{C}}(X) \to \Lambda_a$ and an algorithm A such that A computes $e(\eta)$ from η in time bounded by $q(\eta)$ for some resource polynomial q;
- for each $\eta \in \mathcal{V}_{\mathscr{C}}(X)$, $t \in \Lambda_a$, $a \in |A|$, we have that $\eta, t \Vdash_A a$ implies $\eta, e(\eta)t \Vdash_B f(a)$.

In order to define realizability sets $\forall\overline{y}{:}\mathscr{C}.A$ and $\exists\overline{y}{:}\mathscr{C}.A$, we fix some encoding of environments η as affine lambda terms using the $\ulcorner\cdot\urcorner$ encoding of natural numbers. As an example, the environment η on $\{x_0, \ldots, x_{n-1}\}$ could be encoded as $\bigotimes_{i<n} \ulcorner\eta(x_i)\urcorner$; this clearly relies on a total order on resource variables. We do not notationally distinguish environments from their encodings.

Definition 4.9 (First-order Quantification)**.** Let X, Y be disjoint sets of variables. Let A be a realizability set over $X \cup Y$ and $\mathscr{C}$ a constraint set over $X \cup Y$ where we put $Y = \{y_1, \ldots, y_n\}$ and $\overline{y} = (y_1, \ldots, y_n)$. Furthermore, for each $i = 1, \ldots, n$ let $p_i \in \mathcal{P}(X)$ be such that $\mathscr{C} \models \{\overline{y} \leq \overline{p}\}$.

- $|\forall\overline{y}{:}\mathscr{C}.A| = |\exists\overline{y}{:}\mathscr{C}.A| = |A|$,
- $\eta, t \Vdash_{\forall\overline{y}:\mathscr{C}.A} a \iff \forall\mu \in \mathcal{V}(Y).\eta{\cup}\mu \in \mathcal{V}_{\mathscr{C}} \Rightarrow \eta{\cup}\mu, t\mu \Vdash_A a$.
- $\eta, \mu \otimes t \Vdash_{\exists\overline{y}:\mathscr{C}.A} a \iff \mu \in \mathcal{V}(Y) \wedge \eta{\cup}\mu \in \mathcal{V}_{\mathscr{C}} \wedge \eta{\cup}\mu, t \Vdash_A a$

Recall that $\forall\overline{y}{:}\mathscr{C}.A$ and $\exists\overline{y}{:}\mathscr{C}.A$ are well-formed only if there are resource polynomials $\overline{p}$ such that $\mathscr{C} \models \overline{y} < \overline{p}$. Therefore, the set $\{\mu \mid \eta{\cup}\mu \in \mathcal{V}_{\mathscr{C}}\}$ is finite and in fact computable in polynomial time from η. Indeed, its cardinality at most

$$\prod_{i=1}^{n} p_i(\eta),$$

and the size of any of its elements is at most

$$|\eta| + \prod_{i=1}^{n} p_i(\eta).$$

We are now able to prove the main result of this Section:

Theorem 4.10. *Let π be a proof of a sequent $\Gamma \vdash_{\mathscr{C}} B$ and ρ a mapping of atoms to realizability sets. Then $[\![\pi]\!]_{|\rho|}$ is a $\mathscr{C}$-morphism from $[\![\Gamma]\!]_\rho^{\mathcal{B}}$ to $[\![B]\!]_\rho^{\mathcal{B}}$.*

Proof. The proof is by induction on derivations. We only show the cases that differ significantly from the development in [10].
Case $P_!$. For simplicity, suppose that $n = 1$, $q_1 = p$ and $A_1 = A$. The induction hypothesis shows that $[\![\pi]\!]_{|\rho|}$ is a $\mathscr{C}$-morphism from $[\![A]\!]_\rho^{\mathcal{B}}$ to $[\![B]\!]_\rho^{\mathcal{B}}$ witnessed by e. As in the proof of the main result in [10], we define

$$d(\eta) = \lambda \bigotimes_{i<p(\eta)} x_i. \bigotimes_{i<p(\eta)} e(\eta[x \mapsto i])x_i.$$

Now, if $\eta \in \mathcal{V}_{\mathscr{D}}$, then $\eta[x \mapsto i] \in \mathcal{V}_{\mathscr{C}}$ whenever $i < p(\eta)$ by the side condition from rule $P_!$. We obtain that $[\![\pi]\!]$ is a $\mathscr{D}$-morphism from $[\![!_{x<p}A]\!]_\rho^{\mathcal{B}}$ to $[\![!_{x<p}B]\!]_\rho^{\mathcal{B}}$ witnessed by d.
The remaining cases are the four rules for first order quantifiers. In each case, we assume by the induction hypothesis that $[\![\pi]\!]$ is a morphism realizing the premise of the rule and let e be its witness. We have to show that $[\![\pi]\!]$ is a morphism realizing the conclusion of the rule. Note that the set-theoretic meaning of a proof does not change upon application of any of the quantifier rules.
Case $R_{\forall x}$. Suppose that $\eta \in \mathcal{V}_{\mathscr{C}}$ and $\eta, t_\gamma \Vdash_{[\![\Gamma]\!]_\rho^{\mathcal{B}}} \gamma$. Now suppose $\eta \cup \mu \in \mathcal{V}_{\mathscr{D}}$. By the induction hypothesis $\eta \cup \mu, e(\eta \cup \mu)t_\gamma \Vdash_{[\![A]\!]_\rho^{\mathcal{B}}} [\![\pi]\!](\gamma)$. We thus define d by $d(\eta) = u$ where $u \in \Lambda_a$ is such that $ut\mu = e(\eta \cup \mu)t$ whenever $\eta \cup \mu \in \mathcal{V}_{\mathscr{C}}$. Recall that for a given η there are only $q(\eta)$ such μ (for a fixed resource polynomial q), so that t can be constructed as a big case distinction over all those μ. It is then clear that d is polynomial time computable and realizes the conclusion of the rule.
Case $L_{\forall x}$. Assume $\eta \in \mathcal{V}_{\mathscr{C}}$ and $\eta, t_\gamma \Vdash_{[\![\Gamma]\!]_\rho^{\mathcal{B}}} \gamma$ and $\eta, t_a \Vdash_{[\![\forall \overline{x}:\mathscr{D}.A]\!]_\rho^{\mathcal{B}}} a$. Define $\mu_\eta \in \mathcal{V}(X)$ by $\mu_\eta(x_i) = p_i(\eta)$ so that $\eta \cup \mu_\eta \in \mathcal{V}_{\mathscr{D}}$ by the side condition to the rule. Now, $\eta \cup \mu_\eta, t_a\mu_\eta \Vdash_A a$ by Definition 4.9. Hence, $\eta, t_a\mu_\eta \Vdash_{A\{\overline{p}/\overline{x}\}} a$. By the induction hypothesis, $e(\eta)(t_\gamma \otimes t_a\mu_\eta) \Vdash_C [\![\pi]\!](\gamma, a)$, so $d(\eta) = \lambda x_\gamma \otimes x_a.e(\eta)(x_\gamma \otimes x_a(\mu_\eta))$ does the job. The remaining two cases are essentially dual to the previous two. We merely define the witnesses.
Case $R_{\exists x}$. Define $\mu(\eta)$ as in $L_{\forall x}$. We can then put $d(\eta) = \lambda t_\gamma.\mu(\eta) \otimes e(\eta)t_\gamma$.
Case $L_{\exists x}$. We define $d(\eta)$ to be such that $d(\eta)t_\gamma(\mu \otimes t) = e(\eta \cup \mu)t_\gamma t$. This is possible by hard-wiring separate cases for each of the polynomially in η many μ like in case $R_{\forall x}$. □

Corollary 4.11. *Every function on word algebras representable in* QBAL *is polynomial time computable.*

5. On Compositional Embeddings

In this Section, we justify our emphasis on compositional embeddings. An embedding of a logical system or programming language L into QBAL is a function $\langle\cdot\rangle$ from the space of proofs (or programs) of L into the space of proofs for QBAL. Clearly, for an embedding to be relevant from a computational point of view, any proof π of L should be mapped to an equivalent proof $\langle\pi\rangle$, e.g., $\llbracket\langle\pi\rangle\rrbracket = \llbracket\pi\rrbracket$. The existence of an embedding of L into QBAL implicitly proves that QBAL is *extensionally* at least as powerful as L. Such an embedding $\langle\cdot\rangle$ is not necessarily computable nor natural. But whenever L is a sound and complete ICC characterization of polynomial time, a large class of proofs or programs of L can be mapped to QBAL, since the classes of definable *first order* functions are exactly the same in L and QBAL. Indeed, QBAL is both extensionally sound (see Section 4) and extensionally complete (since BLL can be compositionally embedded into it).

Typically, one would like to go beyond extensionality and prove that QBAL is *intensionally* as powerful as L. And if this is the goal, $\langle\cdot\rangle$ should be easily computable. Ideally, we would like $\langle\cdot\rangle$ to act homeomorphically on the space of proofs of L. In other words, whenever a proof π of L is obtained applying a proof-forming rule R to $\rho_1,\ldots,\rho_n$, then $\langle\pi\rangle$ should be obtainable from $\langle\rho_1\rangle,\ldots,\langle\rho_n\rangle$ in a uniform way, i.e., dependently on R but independently on $\langle\rho_1\rangle,\ldots,\langle\rho_n\rangle$. An embedding satisfying the above constraint is said to be *strongly compositional*. The embeddings we will present in the following two sections are only *weakly* compositional: $[\langle\pi\rangle]$ can be uniformly built from $[\langle\rho_1\rangle],\ldots,[\langle\rho_n\rangle]$ whenever π is obtained applying R to $\rho_1,\ldots,\rho_n$. We believe that the existence of a weakly compositional embedding of L into QBAL is sufficient to guarantee that QBAL is intensionally as powerful as L because, as we pointed out in Section 2.5, $[\pi]$ can be thought as the program hidden in the proof π.

6. Embedding LFPL

LFPL is a calculus for non-size-increasing computation introduced by the second author [9]. It allows to capture natural algorithms computing functions such that the size of the result is smaller or equal to the size of the arguments. This way, polynomial time soundness is guaranteed despite the possibility of arbitrarily nested recursive definitions.

We here show that a core subset of LFPL can be compositionally embedded into QBAL. LFPL types are generated by the following grammar:

$$A ::= \diamond \mid \mathbf{N} \mid A\otimes A \mid A \multimap A.$$

Rules for LFPL in natural-deduction style are in Figure 3. We omit terms, since the computational content of type derivations is implicit in their skeleton. The set-theoretic semantics $\llbracket A\rrbracket$ of an LFPL formula A can be defined very easily: $\llbracket\diamond\rrbracket = \prod_{C\in\mathcal{U}} C \Rightarrow C$,

Axiom, Base Types and Weakening

$$\frac{}{A\vdash A}\,A \qquad \frac{}{\diamond,\mathbf{N}\vdash\mathbf{N}}\,S \qquad \frac{\diamond\vdash A\multimap A \quad \vdash A}{\vdash \mathbf{N}\multimap A}\,T \qquad \frac{\Gamma\vdash A}{\Gamma,B\vdash A}\,W$$

Multiplicative Rules

$$\frac{\Gamma,A\vdash B}{\Gamma\vdash A\multimap B}\,I_{\multimap} \qquad \frac{\Gamma\vdash A\multimap B \quad \Delta\vdash A}{\Gamma,\Delta\vdash B}\,E_{\multimap}$$

$$\frac{\Gamma\vdash A \quad \Delta\vdash B}{\Gamma,\Delta\vdash A\otimes B}\,I_{\otimes} \qquad \frac{\Gamma\vdash A\otimes B \quad \Delta,A,B\vdash C}{\Gamma,\Delta\vdash C}\,E_{\otimes}$$

Figure 3: LFPL

$[\![\mathbf{N}]\!] = \prod_{C \in \mathcal{U}} (C \Rightarrow C) \Rightarrow (C \Rightarrow C)$, while the operators $\otimes$ and $\multimap$ are interpreted as usual. Notice that the interpretation of an LFPL formula does not depend on any environment ρ. This way, any LFPL proof $\pi : A_1, \ldots, A_n \vdash B$ can be given a semantics $[\![\pi]\!] : [\![A_1 \otimes \ldots \otimes A_n]\!] \to [\![B]\!]$, itself independent on any ρ. For example, rule T corresponds to iteration, while rule $E_{\multimap}$ corresponds to function application.

LFPL types can be translated to QBAL formulas in the following way:

$$
\begin{aligned}
\langle \diamond \rangle_p^q &= \exists \varepsilon : \{1 \leq p\}. \forall \alpha. \alpha \multimap \alpha \\
\langle \mathbf{N} \rangle_p^q &= \mathbf{N}_p \\
\langle A \otimes B \rangle_p^q &= \exists (x, y) : \{x + y \leq p\}. \langle A \rangle_x^q \otimes \langle B \rangle_y^q \\
\langle A \multimap B \rangle_p^q &= \forall (x) : \{x + p \leq q\}. \langle A \rangle_x^q \multimap \langle B \rangle_{p+x}^q
\end{aligned}
$$

Please observe that the interpretation of any LFPL formulas is parametrized by two resource polynomials p and q. If a variable x occurs in p, but not in q, then x occurs only positively in $\langle A \rangle_p^q$: this can be proved by an easy induction on the structure of A.

The correspondence scales from types to proofs:

Theorem 6.1. *LFPL can be embedded into QBAL. In other words, for every LFPL proof $\pi : A_1, \ldots, A_n \vdash B$, there exists a QBAL proof*

$$\langle \pi \rangle : \langle A_1 \rangle_{x_1}^b, \ldots, \langle A_n \rangle_{x_n}^b \vdash_{\{\sum_i x_i \leq b, 1 \leq b\}} \langle B \rangle_{\sum_i x_i}^b$$

such that $[\![\pi]\!] = [\![\langle \pi \rangle]\!]$.

Proof. As expected, the proof goes by induction on π.

- If the only rule in π is A, then $\langle \pi \rangle$ is simply the axiom

$$\dfrac{}{\langle A \rangle_x^b \vdash_{\{x \leq b, 1 \leq b\}} \langle A \rangle_x^b}\; A$$

- If the only rule in π is S, then $\langle \pi \rangle$ is

$$\dfrac{\dfrac{\rho : \langle \mathbf{N} \rangle_y^b \vdash_{\{x+y \leq b, 1 \leq b, 1 \leq x\}} \langle \mathbf{N} \rangle_{x+y}^b}{\forall \alpha. \alpha \multimap \alpha, \langle \mathbf{N} \rangle_y^b \vdash_{\{x+y \leq b, 1 \leq b, 1 \leq x\}} \langle \mathbf{N} \rangle_{x+y}^b}\; W}{\langle \diamond \rangle_x^b, \langle \mathbf{N} \rangle_y^b \vdash_{\{x+y \leq b, 1 \leq b\}} \langle \mathbf{N} \rangle_{x+y}^b}\; L_{\exists x}$$

  where $\sigma : \langle \mathbf{N} \rangle_y^b \vdash \langle \mathbf{N} \rangle_{y+1}^b$ is the QBAL proof for the successor on natural numbers inherited from BLL and ρ is obtained by first strengthening σ into a proof of $\langle \mathbf{N} \rangle_y^b \vdash_{\{x+y \leq b, 1 \leq b, 1 \leq x\}} \langle \mathbf{N} \rangle_{y+1}^b$ (by Proposition 2.13) and then applying to it Proposition 2.14, after observing that $\langle \mathbf{N} \rangle_{y+1}^b \sqsubseteq_{\{x+y \leq b, 1 \leq b, 1 \leq x\}} \langle \mathbf{N} \rangle_{y+x}^b$.
- If the last rule in π is W and the immediate premise of π is ρ, then $\langle \pi \rangle$ is

$$\dfrac{\sigma : \langle \Gamma \rangle_{\overline{x}}^b \vdash_{\{\sum_i x_i + y \leq b, 1 \leq b\}} \langle B \rangle_{\sum_i x_i + y}^b}{\langle \Gamma \rangle_{\overline{x}}^b, \langle A \rangle_y^b \vdash_{\{\sum_i x_i + y \leq b, 1 \leq b\}} \langle B \rangle_{\sum_i x_i + y}^b}\; W$$

  where σ can be obtained from $\langle \rho \rangle$ by Proposition 2.13 and Proposition 2.14, because $\{\sum_i x_i + y \leq b, 1 \leq b\} \models \{\sum_i x_i \leq b, 1 \leq b\}$ and $\langle B \rangle_{\sum_i x_i}^b \sqsubseteq \langle B \rangle_{\sum_i x_i + y}^b$

- If the last rule in π is $E_{\multimap}$ and the immediate premises of π are ρ and σ, then $\langle\pi\rangle$ is

$$\dfrac{\dfrac{\theta : \langle\Gamma\rangle^{b}_{\frac{b}{x}} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b\}} \langle A \multimap B\rangle^{b}_{\sum_i x_i}}{\langle\Gamma\rangle_{\frac{b}{x}} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b\}} \langle A\rangle^{b}_{\sum_i y_i} \multimap \langle B\rangle^{b}_{\sum_i x_i+\sum_i y_i}}\,L_{\forall x}, A, U \qquad \xi : \langle\Delta\rangle_{\frac{b}{y}} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b\}} \langle A\rangle^{b}_{\sum_i y_i}}{\langle\Gamma\rangle_{\frac{b}{x}}, \langle\Delta\rangle_{\frac{b}{y}} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b\}} \langle B\rangle^{b}_{\sum_i x_i+\sum_i y_i}}\,L_{\multimap}, A, U$$

  where θ and ξ can be obtained from $\langle\rho\rangle$ and $\langle\sigma\rangle$, respectively, by applying Proposition 2.13.
- If the last rule in π is $I_{\multimap}$ and the immediate premise of π is ρ, then $\langle\pi\rangle$ is

$$\dfrac{\dfrac{\sigma : \langle\Gamma\rangle_{\frac{b}{x}}, \langle A\rangle^{b}_{y} \vdash_{\{\sum_i x_i\leq b,\sum_i x_i+y\leq b,1\leq b\}} \langle B\rangle^{b}_{\sum_i x_i+y}}{\langle\Gamma\rangle_{\frac{b}{x}} \vdash_{\{\sum_i x_i\leq b,\sum_i x_i+y\leq b,1\leq b\}} \langle A\rangle^{b}_{y} \multimap \langle B\rangle^{b}_{\sum_i x_i+y}}\,R_{\multimap}}{\langle\Gamma\rangle_{\frac{b}{x}} \vdash_{\{\sum_i x_i\leq b,1\leq b\}} \langle A \multimap B\rangle^{b}_{\sum_i x_i}}\,R_{\forall x}$$

  where σ can be obtained from $\langle\rho\rangle$ by applying Proposition 2.13.
- If the last rule in π is $I_{\otimes}$ and the immediate premises of π are ρ and σ, then $\langle\pi\rangle$ is

$$\dfrac{\dfrac{\theta : \langle\Gamma\rangle_{\frac{b}{x}} \vdash_{\{\sum_i x_i\leq b,\sum_i y_i\leq b,1\leq b\}} \langle A\rangle^{b}_{\sum_i x_i} \qquad \xi : \langle\Delta\rangle_{\frac{b}{y}} \vdash_{\{\sum_i x_i\leq b,\sum_i y_i\leq b,1\leq b\}} \langle B\rangle^{b}_{\sum_i y_i}}{\langle\Gamma\rangle_{\frac{b}{x}}, \langle\Delta\rangle_{\frac{b}{y}} \vdash_{\{\sum_i x_i\leq b,\sum_i y_i\leq b,1\leq b\}} \langle A\rangle^{b}_{\sum_i x_i} \otimes \langle B\rangle^{b}_{\sum_i y_i}}\,R_{\otimes}}{\langle\Gamma\rangle_{\frac{b}{x}}, \langle\Delta\rangle_{\frac{b}{y}} \vdash_{\{\sum_i x_i\leq b,\sum_i y_i\leq b,1\leq b\}} \langle A \otimes B\rangle^{b}_{\sum_i x_i+\sum_i y_i}}\,R_{\exists x}$$

  where θ and ξ can be obtained from $\langle\rho\rangle$ and $\langle\sigma\rangle$, respectively, by Proposition 2.13.
- If the last rule in π is $E_{\otimes}$ and the immediate premises of π are ρ and σ, then $\langle\pi\rangle$ is

$$\dfrac{\theta \qquad \dfrac{\dfrac{\xi : \langle\Gamma\rangle_{\frac{b}{x}}, \langle A\rangle^{b}_{w}, \langle B\rangle^{b}_{z} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b,w+z\leq\sum_i y_i\}} \langle C\rangle^{b}_{\sum_i x_i+\sum_i y_i}}{\langle\Gamma\rangle_{\frac{b}{x}}, \langle A\rangle^{b}_{w} \otimes \langle B\rangle^{b}_{z} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b,w+z\leq\sum_i y_i\}} \langle C\rangle^{b}_{\sum_i x_i+\sum_i y_i}}\,L_{\otimes}}{\langle\Gamma\rangle_{\frac{b}{x}}, \langle A \otimes B\rangle^{b}_{\sum_i y_i} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b\}} \langle C\rangle^{b}_{\sum_i x_i+\sum_i y_i}}\,L_{\exists x}}{\langle\Gamma\rangle_{\frac{b}{x}}, \langle\Delta\rangle_{\frac{b}{y}} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b\}} \langle C\rangle^{b}_{\sum_i x_i+\sum_i y_i}}\,U$$

  where ξ can be obtained from $\langle\rho\rangle$ by Proposition 2.13 and Proposition 2.14 and

$$\theta : \langle\Delta\rangle_{\frac{b}{y}} \vdash_{\{\sum_i x_i+\sum_i y_i\leq b,1\leq b\}} \langle A \otimes B\rangle^{b}_{\sum_i y_i}$$

  can be obtained from $\langle\sigma\rangle$ by Proposition 2.13.
- If the last rule in π is T and the immediate premises of π are ρ and σ, then $\langle\pi\rangle$ is

$$\dfrac{\dfrac{\dfrac{\dfrac{\sigma : \vdash_{\{y+1\leq b,1\leq b\}} \langle\diamond\rangle^{b}_{1} \qquad \theta : \langle\diamond\rangle^{b}_{1} \vdash_{\{y+1\leq b,1\leq b\}} \langle A \multimap A\rangle^{b}_{1}}{\vdash_{\{y+1\leq b,1\leq b\}} \langle A \multimap A\rangle^{b}_{1}}\,U}{\vdash_{\{y+1\leq b,1\leq b\}} \langle A\rangle^{b}_{y} \multimap \langle A\rangle^{b}_{y+1}}\,L_{\forall x}, A, U}{\vdash_{\{x\leq b,1\leq b\}} !_{y<x}(\langle A\rangle^{b}_{y} \multimap \langle A\rangle^{b}_{y+1})}\,P_{!} \qquad \xi : \vdash_{\{x\leq b,1\leq b\}} \langle A\rangle^{b}_{0} \quad \pi^{A}_{x}}{\langle\mathbf{N}\rangle^{b}_{x} \vdash_{\{x\leq b,1\leq b\}} \langle A\rangle^{b}_{x}}\,U$$

  where θ and ξ can be obtained from $\langle\rho\rangle$ and $\langle\sigma\rangle$, respectively, by Proposition 2.13 and σ can be easily built.

This concludes the proof. □

Proposition 6.2. *The correspondence* $\langle\cdot\rangle$ *is weakly compositional.*

Proof. A quick inspection on the proof of Theorem 6.1 shows that $\langle\pi\rangle$ *cannot* be obtained uniformly from $\langle\rho_1\rangle,\dots,\langle\rho_1\rangle$ (where $\rho_1,\dots,\rho_n$ are the immediate sub-proofs of π), because results like Proposition 2.13 or Proposition 2.14 are often applied to $\langle\rho_1\rangle,\dots,\langle\rho_n\rangle$ before they are plugged together (in a uniform way) to obtain $\langle\pi\rangle$. All the results in Section 2.6, however, transform proofs to proofs preserving the underlying LJ proof. As a consequence the embedding is weakly compositional. □

One may ask whether such an embedding might work for BLL proper. We believe this to be unlikely for several reasons. In particular, it seems that BLL lacks a mechanism for turning the information about the size of the manipulated objects from being global to being local. In QBAL, this rôle is played by first order quantifiers. As an example, consider the split function for lists of natural numbers that splits a list into two lists, one containing the even entries and one containing the odd entries. The type of that function in LFPL is $\mathbf{L}(\mathbf{N}) \multimap \mathbf{L}(\mathbf{N}) \otimes \mathbf{L}(\mathbf{N})$ where $\mathbf{L}(\cdot)$ denotes the type of lists that we have elided from our formal treatment for the sake of simplicity. In QBAL this function gets the type

$$\mathbf{L}_x(\mathbf{N}_y) \multimap \exists(u,v) : \{u+v \le x\}.\mathbf{L}_u(\mathbf{N}_y) \otimes \mathbf{L}_v(\mathbf{N}_y).$$

The only conceivable BLL formula for this function is $\mathbf{L}_x(\mathbf{N}_y) \multimap \mathbf{L}_x(\mathbf{N}_y) \otimes \mathbf{L}_x(\mathbf{N}_y)$. In LFPL and in QBAL we can compose the split function with "append" yielding a function of type $\mathbf{L}_x(\mathbf{N}_y) \multimap \mathbf{L}_x(\mathbf{N}_y)$ that can be iterated. In BLL this composition receives the type $\mathbf{L}_x(\mathbf{N}_y) \multimap \mathbf{L}_{2x}(\mathbf{N}_y)$ which of course is not allowed in an iteration. But a hypothetical compositional embedding of LFPL into BLL would have to be able to mimic this construction.

7. Embedding RRW

Ramified recurrence on words (RRW) is a function algebra extensionally corresponding to polynomial time functions introduced by Leivant in the early nineties [13]. Bellantoni and Cook's algebra BC can be easily embedded into RRW.

Let $\mathbb{W}$ be a word algebra, let $c^1_{\mathbb{W}},\dots,c^w_{\mathbb{W}}$ be the unary constructors of $\mathbb{W}$ and let $\varepsilon_{\mathbb{W}}$ be the only 0-ary constructor of $\mathbb{W}$. id denotes the identity function on $\mathbb{W}$. If m is a natural number and $1 \le i \le m$, π^m_i denotes the i-th projection on m arguments in $\mathbb{W}$. Given a n-ary function g on $\mathbb{W}$ and n m-ary functions $f_1,\dots,f_n$ on $\mathbb{W}$, we can define the m-ary *composition* of g and $f_1,\dots,f_n$, denoted $\mathsf{comp}(g,f_1,\dots,f_n)$, as follows:

$$\mathsf{comp}(g,f_1,\dots,f_n)(t_1,\dots,t_m) = g(f_1(t_1,\dots,t_m),\dots,f_n(t_1,\dots,t_m)).$$

Given an n-ary function $f_{\varepsilon_{\mathbb{W}}}$ on $\mathbb{W}$ and $n+2$-ary functions $f_{c^1_{\mathbb{W}}},\dots,f_{c^w_{\mathbb{W}}}$ on $\mathbb{W}$, we can define an $n+1$-ary function g, denoted $\mathsf{rec}(f_{c^1_{\mathbb{W}}},\dots,f_{c^w_{\mathbb{W}}},f_{\varepsilon_{\mathbb{W}}})$, by *primitive recursion* as follows:

$$\begin{array}{rcl} g(t_1,\dots,t_n,\varepsilon_{\mathbb{W}}) & = & f_{\varepsilon_{\mathbb{W}}}(t_1,\dots,t_n) \\ g(t_1,\dots,t_n,c^i_{\mathbb{W}}(t)) & = & f_{c^i_{\mathbb{W}}}(t_1,\dots,t_n,g(t_1,\dots,t_n,t),t) \end{array}$$

Given an n-ary function $f_{\varepsilon_{\mathbb{W}}}$ on $\mathbb{W}$ and $n+1$-ary functions $f_{c^1_{\mathbb{W}}},\dots,f_{c^w_{\mathbb{W}}}$ on $\mathbb{W}$, we can define an $n+1$-ary function g, denoted $\mathsf{cond}(f_{c^1_{\mathbb{W}}},\dots,f_{c^w_{\mathbb{W}}},f_{\varepsilon_{\mathbb{W}}})$, as a *conditional* as follows:

$$\begin{array}{rcl} g(t_1,\dots,t_n,\varepsilon_{\mathbb{W}}) & = & f_{\varepsilon_{\mathbb{W}}}(t_1,\dots,t_n) \\ g(t_1,\dots,t_n,c^i_{\mathbb{W}}(t)) & = & f_{c^i_{\mathbb{W}}}(t_1,\dots,t_n,t) \end{array}$$

We can generate functions starting from id, π_i^m, $\varepsilon_{\mathbb{W}}$ and $c_{\mathbb{W}}$ by freely applying composition, primitive recursion and conditional.

Not every function obtained this way is in RRW: indeed, they correspond to primitive recursive functions on $\mathbb{W}$. In Figure 4, a formal system for judgements in the form $\vdash f : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \to \mathbb{W}^i$ (where $i_1, \ldots, i_n, i$ are natural numbers) is defined. If such a judgement can be derived from the rules in Figure 4, then f is said to be an RRW function (the definition of RRW given here is slightly different but essentially equivalent to the original one [13]). Leivant [13] proved that RRW functions are exactly the polytime computable functions on $\mathbb{W}$. But RRW can be compositionally embedded into QBAL, at least in a weak sense. Before embarking in the proof of that, however, we need a preliminary

$$\frac{}{\vdash \mathsf{id} : \mathbb{W}^i \to \mathbb{W}^i} \qquad \frac{}{\vdash \varepsilon_{\mathbb{W}} : \mathbb{W}^i} \qquad \frac{}{\vdash c_{\mathbb{W}} : \mathbb{W}^i \to \mathbb{W}^i} \qquad \frac{j_i = j}{\vdash \pi_i^m : \mathbb{W}^{j_1} \times \ldots \times \mathbb{W}^{j_m} \to \mathbb{W}^j}$$

$$\frac{\begin{array}{c}\vdash g : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \to \mathbb{W}^i \\ \forall k \in \{1, \ldots, w\}. \vdash f_k : \mathbb{W}^{j_1} \times \ldots \times \mathbb{W}^{j_m} \to \mathbb{W}^{i_k}\end{array}}{\vdash \mathsf{comp}(g, f_1, \ldots, f_n) : \mathbb{W}^{j_1} \times \ldots \mathbb{W}^{j_m} \to \mathbb{W}^i}$$

$$\frac{\begin{array}{c}\forall k \in \{1, \ldots, w\}. \vdash f_{c_{\mathbb{W}}^k} : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \times \mathbb{W}^i \times \mathbb{W}^j \to \mathbb{W}^i \\ \vdash f_\varepsilon : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \to \mathbb{W}^i\end{array} \quad i < j}{\vdash \mathsf{rec}(f_{c_{\mathbb{W}}^1}, \ldots, f_{c_{\mathbb{W}}^w}, f_{\varepsilon_{\mathbb{W}}}) : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \times \mathbb{W}^j \to \mathbb{W}^i}$$

$$\frac{\begin{array}{c}\forall k \in \{1, \ldots, w\}. \vdash f_{c_{\mathbb{W}}^k} : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \times \mathbb{W}^j \to \mathbb{W}^i \\ \vdash f_\varepsilon : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \to \mathbb{W}^i\end{array}}{\vdash \mathsf{cond}(f_{c_{\mathbb{W}}^1}, \ldots, f_{c_{\mathbb{W}}^w}, f_{\varepsilon_{\mathbb{W}}}) : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \times \mathbb{W}^j \to \mathbb{W}^i}$$

Figure 4: RRW as a formal system.

result.

Lemma 7.1 (Contraction Lemma)**.** *Every word algebra is duplicable, i.e., for every word algebra* $\mathbb{W}$ *there is a proof* $\pi_{\mathbb{W}}$ *of* $\mathbf{W}_x \vdash \mathbf{W}_x \otimes \mathbf{W}_x$ *such that* $[\![\pi_{\mathbb{W}}]\!](x) = (x, x)$.

Proof. For simplicity, consider the algebra $\mathbb{N}$ of natural numbers. The proof we are looking for is the following:

$$\dfrac{\dfrac{\dfrac{\sigma : \mathbf{N}_y \vdash \mathbf{N}_{y+1} \quad \sigma : \mathbf{N}_y \vdash \mathbf{N}_{y+1}}{\vdash \mathbf{N}_y \otimes \mathbf{N}_y \multimap \mathbf{N}_{y+1} \otimes \mathbf{N}_{y+1}} R_\otimes, L_\otimes, R_\multimap}{\vdash !_{y<x}(\mathbf{N}_y \otimes \mathbf{N}_y \multimap \mathbf{N}_{y+1} \otimes \mathbf{N}_{y+1})} P_! \qquad \dfrac{\rho :\vdash \mathbf{N}_0 \quad \rho :\vdash \mathbf{N}_0}{\vdash \mathbf{N}_0 \otimes \mathbf{N}_0} R_\otimes \qquad \pi_x^{\mathbf{N}_x \otimes \mathbf{N}_x}}{\mathbf{N}_x \vdash \mathbf{N}_x \otimes \mathbf{N}_x} U$$

where σ are ρ are the proofs corresponding to 0 and the successor coming from BLL. This concludes the proof. □

The following is the main result of this section:

Theorem 7.2. *RRW can be embedded into QBAL. Suppose, in other words, that*

$$\pi :\vdash f : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \to \mathbb{W}^i$$

and that $i < i_{j_1}, \ldots, i_{j_m}$, while $i = i_{k_1}, \ldots, i_{k_h}$. Then, there exist a QBAL *proof π and a resource polynomial q such that*

$$\langle \pi \rangle : \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n} \vdash_{\{x_{k_1} \leq x, \ldots, x_{k_h} \leq x\}} \mathbf{W}_{q(x_{j_1}, \ldots, x_{j_m}) + x}.$$

where $(\prod_{i=1}^{n} \psi_{\mathbb{W}}) \circ [\![\pi]\!] \circ \varphi_{\mathbb{W}} = f$.

Proof. By induction on the proof of

$$\vdash f : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \to \mathbb{W}^{i}.$$

Some interesting cases:

- Consider the identity function id. Clearly:

$$\frac{}{\mathbf{W}_{x_1} \vdash_{\{x_1 \leq x\}} \mathbf{W}_x}$$

- Suppose $f = \mathsf{comp}(g, f_1, \ldots, f_n)$ and

$$\frac{\vdash g : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \to \mathbb{W}^{i} \quad \vdash f_k : \mathbb{W}^{j_1} \times \ldots \times \mathbb{W}^{j_m} \to \mathbb{W}^{i_k}}{\vdash \mathsf{comp}(g, f_1, \ldots, f_n) : \mathbb{W}^{j_1} \times \ldots \mathbb{W}^{j_m} \to \mathbb{W}^{i}}$$

  We partition the sequence $i_1, \ldots, i_n$ into three sequences containing elements which are equal to i, strictly greater than i and strictly smaller than i, respectively:

$$\begin{array}{c} i_{k_1}, \ldots, i_{k_h} \\ i_{u_1}, \ldots, i_{u_v} \\ i_{a_1}, \ldots, i_{a_b} \end{array}$$

  Clearly, $n = h + v + b$. Similarly for the sequence $j_1, \ldots, j_m$:

$$\begin{array}{c} j_{t_1}, \ldots, j_{t_e} \\ j_{c_1}, \ldots, j_{c_l} \\ j_{z_1}, \ldots, j_{z_d} \end{array}$$

  Again, $m = e + d + l$. By induction hypothesis, there are proofs $\pi_g, \pi_{f_1}, \ldots, \pi_{f_n}$ with the appropriate conclusions. Now, consider the proofs $\pi_{f_{k_1}}, \ldots, \pi_{f_{k_h}}$: they are the ones such that $i = i_{k_1}, \ldots, i_{k_h}$. By Proposition 2.14, we can assume that their conclusion is exactly the same, i.e., the polynomials $r_{k_1}, \ldots, r_{k_h}$ in the rhs are indeed the same polynomial r. In other words, we have proofs

$$\begin{array}{rcl} \rho_{f_{k_1}} & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_m} \vdash_{x_{t_1} \leq x, \ldots, x_{t_e} \leq x} \mathbf{W}_{q_{k_1}(x_1, \ldots, x_m, x)} \\ & \vdots & \\ \rho_{f_{k_h}} & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_m} \vdash_{x_{t_1} \leq x, \ldots, x_{t_e} \leq x} \mathbf{W}_{q_{k_h}(x_1, \ldots, x_m, x)} \end{array}$$

  where $q_{k_0}(x_1, \ldots, x_m, x) = r(x_{z_1}, \ldots, x_{z_d}) + x$ for each o. The proofs $\pi_{f_{u_1}}, \ldots, \pi_{f_{u_v}}$ are the ones such that $i < i_{u_1}, \ldots, i_{u_v}$. Observe that $x_{t_1}, \ldots, x_{t_e}$ do not appear in the right hand side of their conclusions. By Proposition 2.15 and Proposition 2.13, we can obtain proofs:

$$\begin{array}{rcl} \rho_{f_{u_1}} & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_m} \vdash_{x_{t_1} \leq x, \ldots, x_{t_e} \leq x} \mathbf{W}_{q_{u_1}(x_1, \ldots, x_m, x)} \\ & \vdots & \\ \rho_{f_{u_v}} & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_m} \vdash_{x_{t_1} \leq x, \ldots, x_{t_e} \leq x} \mathbf{W}_{q_{u_v}(x_1, \ldots, x_m, x)} \end{array}$$

where $q_{u_1},\ldots,q_{u_v}$ only depend on $x_{z_1},\ldots,x_{z_d}$. Similarly, we can obtain proofs

$$\begin{array}{rcl}\rho_{f_{a_1}} & : & \mathbf{W}_{x_1},\ldots,\mathbf{W}_{x_m}\vdash_{x_{t_1}\leq x,\ldots,x_{t_e}\leq x}\mathbf{W}_{q_{a_1}(x_1,\ldots,x_m,x)}\\ & \vdots & \\ \rho_{f_{a_b}} & : & \mathbf{W}_{x_1},\ldots,\mathbf{W}_{x_m}\vdash_{x_{t_1}\leq x,\ldots,x_{t_e}\leq x}\mathbf{W}_{q_{a_b}(x_1,\ldots,x_m,x)}\end{array}$$

corresponding to $\pi_{f_{a_1}},\ldots,\pi_{f_{a_b}}$. Consider the proof π_g. By induction hypothesis it is a proof of

$$\mathbf{W}_{y_1},\ldots,\mathbf{W}_{y_n}\vdash_{y_{k_1}\leq y,\ldots,y_{k_h}\leq y}\mathbf{W}_{p(y_{u_1},\ldots,y_{u_v})+y}$$

Now, define a substitution mapping y_i to $q_i(x_1,\ldots,x_m,x)$ and y to $r(x_{z_1},\ldots,x_{z_d})+x$. By Proposition 2.15, we can obtain a proof

$$\mathbf{W}_{q_1(x_1,\ldots,x_m,x)},\ldots,\mathbf{W}_{q_n(x_1,\ldots,x_m,x)}\vdash_{\mathscr{C}}\mathbf{W}_{s(y_{u_1},\ldots,y_{u_v})+y}$$

where

$$\mathscr{C}=\{q_{k_1}(x_1,\ldots,x_m,x)\leq r(x_{z_1},\ldots,x_{z_d})+x,\ldots,q_{k_h}(x_1,\ldots,x_m,x)\leq r(x_{z_1},\ldots,x_{z_d})+x\}.$$

But $\mathscr{C}$ is always true, and as a consequence

$$\{x_{t_1}\leq x,\ldots,x_{t_e}\leq x\}\models\mathscr{C}.$$

By Proposition 2.13, we can obtain a proof

$$\rho_g:\mathbf{W}_{q_1(x_1,\ldots,x_m,x)},\ldots,\mathbf{W}_{q_n(x_1,\ldots,x_m,x)}\vdash_{x_{t_1}\leq x,\ldots,x_{t_e}\leq x}\mathbf{W}_{s(x_{z_1},\ldots,x_{z_d})+x}$$

Plugging the obtained proofs and remembering that base types are duplicable, we can construct a proof corresponding to $f=\mathsf{comp}(g,f_1,\ldots,f_n)$:

$$\frac{\rho_g\quad\rho_{f_1}\quad\ldots\rho_{f_n}}{\mathbf{W}_{x_1},\ldots,\mathbf{W}_{x_m}\vdash_{x_{t_1}\leq x,\ldots,x_{t_e}\leq x}\mathbf{W}_{p(x_{z_1},\ldots,x_{z_d})+x}}$$

- Suppose $f=\mathsf{rec}(f_{c^1_{\mathbb{W}}},\ldots,f_{c^w_{\mathbb{W}}},f_{\varepsilon_{\mathbb{W}}})$ and

$$\frac{\vdash f_{c^k_{\mathbb{W}}}:\mathbb{W}^{i_1}\times\ldots\times\mathbb{W}^{i_n}\times\mathbb{W}^{i}\times\mathbb{W}^{j}\to\mathbb{W}^{i}\quad\vdash f_{\varepsilon}:\mathbb{W}^{i_1}\times\ldots\times\mathbb{W}^{i_n}\to\mathbb{W}^{i}\quad i<j}{\vdash\mathsf{rec}(f_{c^1_{\mathbb{W}}},\ldots,f_{c^w_{\mathbb{W}}},f_{\varepsilon_{\mathbb{W}}}):\mathbb{W}^{i_1}\times\ldots\times\mathbb{W}^{i_n}\times\mathbb{W}^{j}\to\mathbb{W}^{i}}$$

By induction hypothesis, there are proofs π_ε and $\pi^1_{\mathbb{W}},\ldots,\pi^w_{\mathbb{W}}$ with the appropriate conclusions. By Proposition 2.14 we can assume that:

$$\begin{array}{rcl}\pi_\varepsilon & : & \mathbf{W}_{x_1},\ldots,\mathbf{W}_{x_n}\vdash_{\mathscr{C}}\mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},0)+x}\\ \pi_1 & : & \mathbf{W}_{x_1},\ldots,\mathbf{W}_{x_{n+2}}\vdash_{\mathscr{C}\cup\{x_{n+1}\leq x\}}\mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},x_{n+2})+x}\\ & \vdots & \\ \pi_w & : & \mathbf{W}_{x_1},\ldots,\mathbf{W}_{x_{n+2}}\vdash_{\mathscr{C}\cup\{x_{n+1}\leq x\}}\mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},x_{n+2})+x}\end{array}$$

where $\mathscr{C}=\{x_{k_1}\leq x,\ldots,x_{k_h}\leq x\}$ and q is a fixed resource polynomial. Applying the substitution μ defined as

$$\begin{array}{rcl}x & \mapsto & (y+1)q(x_{j_1},\ldots,x_{j_m},y+1)+x\\ x_{n+1} & \mapsto & (y+1)q(x_{j_1},\ldots,x_{j_m},y)+x\\ x_{n+2} & \mapsto & y\end{array}$$

to $\pi_1, \ldots, \pi_w$ and by using Proposition 2.14, we obtain

$$\begin{array}{lcl}\rho_1 & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}, \mathbf{W}_{(y+1)q(x_{j_1},\ldots,x_{j_m},y)+x}, \mathbf{W}_y \vdash_{\mathscr{D}} \mathbf{W}_{(y+2)q(x_{j_1},\ldots,x_{j_m},y+1)+x} \\ & \vdots & \\ \rho_w & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}, \mathbf{W}_{(y+1)q(x_{j_1},\ldots,x_{j_m},y)+x}, \mathbf{W}_y \vdash_{\mathscr{D}} \mathbf{W}_{(y+2)q(x_{j_1},\ldots,x_{j_m},y+1)+x}\end{array}$$

where $\mathscr{D} = \mathscr{C}[\sigma] \cup \{(y+1)q(x_{j_1}, \ldots, x_{j_m}, y) + x \leq (y+1)q(x_{j_1}, \ldots, x_{j_m}, y+1) + x\}$. Now, notice that

$$(y+1)q(x_{j_1}, \ldots, x_{j_m}, y) + x \leq (y+1)q(x_{j_1}, \ldots, x_{j_m}, y+1) + x$$

always holds because q is monotone. Moreover, $\mathscr{C} \models \mathscr{C}[\sigma]$. By Proposition 2.13, we can then replace $\mathscr{D}$ with $\mathscr{C}$ in $\rho_1 \ldots, \rho_w$. For simplicity, we use the following abbreviations:

$$\begin{array}{lcl}A & = & \mathbf{W}_{x_1} \otimes \ldots \otimes \mathbf{W}_{x_n} \\ B & = & \mathbf{W}_{x_1} \multimap \ldots \multimap \mathbf{W}_{x_n} \multimap (A \otimes \mathbf{W}_{(y+1)q(x_{j_1},\ldots,x_{j_m},y)+x} \otimes \mathbf{W}_y) \\ \Delta & = & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}\end{array}$$

Now, from π_ε and from a proof corresponding to $\varepsilon_{\mathbb{W}}$ (with conclusion $\vdash_{\mathscr{C}} \mathbf{W}_0$), we construct σ_ε as follows:

$$\dfrac{\dfrac{\dfrac{\overline{\mathbf{W}_{x_1} \vdash_{\mathscr{C}} \mathbf{W}_{x_1}} \quad \cdots \quad \overline{\mathbf{W}_{x_n} \vdash_{\mathscr{C}} \mathbf{W}_{x_n}} \quad \pi_\varepsilon : \Delta \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},0)+x} \quad \overline{\vdash_{\mathscr{C}} \mathbf{W}_0}}{\Delta, \Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},0)+x} \otimes \mathbf{W}_0}\ R_\otimes}{\Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},0)+x} \otimes \mathbf{W}_0}\ \text{Lemma 7.1}}{\vdash_{\mathscr{C}} B[y \leftarrow 0]}\ R_\multimap$$

where we have used several times Lemma 7.1. Similarly, from ρ_i and from a proof corresponding to $c^i_{\mathbb{W}}$ (with conclusion $\mathbf{W}_y \vdash_{\mathscr{C}} \mathbf{W}_{y+1}$), we construct σ_i as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{A \vdash_{\mathscr{C}} A} \quad \rho_i : \Delta, \mathbf{W}_{(y+1)q(x_{j_1},\ldots,x_{j_m},y)+x}, \mathbf{W}_y \vdash_{\mathscr{C}} \mathbf{W}_{(y+2)q(x_{j_1},\ldots,x_{j_m},y+1)+x} \quad \overline{\mathbf{W}_y \vdash_{\mathscr{C}} \mathbf{W}_{y+1}}}{A, \mathbf{W}_{(y+1)q(x_{j_1},\ldots,x_{j_m},y)+x}, \mathbf{W}_y, \mathbf{W}_y, \Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{(y+2)q(x_{j_1},\ldots,x_{j_m},y+1)+x} \otimes \mathbf{W}_{y+1}}\ R_\otimes}{B, \Delta, \Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{(y+2)q(x_{j_1},\ldots,x_{j_m},y+1)+x} \otimes \mathbf{W}_{y+1}}\ L_\otimes, \text{Lemma 7.1}}{B, \Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{(y+2)q(x_{j_1},\ldots,x_{j_m},y+1)+x} \otimes \mathbf{W}_{y+1}}\ \text{Lemma 7.1}}{\vdash_{\mathscr{C}} B \multimap B[y \leftarrow y+1]}\ R_\multimap$$

where, again, we have used several times Lemma 7.1. And now we are ready to iterate over the step functions:

$$\dfrac{\dfrac{\sigma_\varepsilon \quad \sigma_1 \quad \ldots \quad \sigma_w}{\mathbf{W}_{x_{n+1}} \vdash_{\mathscr{C}} B[y \leftarrow x_{n+1}]}\ L_\multimap, A, U \quad \dfrac{}{\Delta, B[y \leftarrow x_{n+1}] \vdash_{\mathscr{C}} \mathbf{W}_{(x_{n+1}+1)q(x_{j_1},\ldots,x_{j_m},x_{n+1})+x}}\ A, R_\otimes, L_\multimap}{\Delta, \mathbf{W}_{x_{n+1}} \vdash_{\mathscr{C}} \mathbf{W}_{(x_{n+1}+1)q(x_{j_1},\ldots,x_{j_m},x_{n+1})+x}}\ U$$

- Suppose $f = \mathsf{cond}(f_{c^1_{\mathbb{W}}}, \ldots, f_{c^w_{\mathbb{W}}}, f_{\varepsilon_{\mathbb{W}}})$ and

$$\dfrac{\vdash f_{c_{\mathbb{W}}} : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \times \mathbb{W}^j \to \mathbb{W}^i \quad \vdash f_\varepsilon : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \to \mathbb{W}^i}{\vdash \mathsf{cond}(f_{c^1_{\mathbb{W}}}, \ldots, f_{c^w_{\mathbb{W}}}, f_{\varepsilon_{\mathbb{W}}}) : \mathbb{W}^{i_1} \times \ldots \times \mathbb{W}^{i_n} \times \mathbb{W}^j \to \mathbb{W}^i}$$

We can distinguish three subcases:

- If $i < j$, there are proofs π_ε and $\pi^1_{\mathbb{W}}, \ldots, \pi^w_{\mathbb{W}}$ with the appropriate conclusions. By Proposition 2.14 we can assume that:

$$\begin{array}{rcl}\pi_\varepsilon & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n} \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},0)+x} \\ \pi_1 & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_{n+1}} \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},x_{n+1})+x} \\ & \vdots & \\ \pi_w & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_{n+1}} \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},x_{n+1})+x}\end{array}$$

where $\mathscr{C} = \{x_{k_1} \leq x, \ldots, x_{k_h} \leq x\}$. Applying the substitution $x_{n+1} \mapsto y$ to $\pi_1, \ldots, \pi_w$ and by using Proposition 2.14 and Proposition 2.13, we obtain

$$\begin{array}{rcl}\rho_1 & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}, \mathbf{W}_y \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},y+1)+x} \\ & \vdots & \\ \rho_w & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}, \mathbf{W}_y \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},y+1)+x}\end{array}$$

For simplicity, wee use the following abbreviations:

$$\begin{array}{rcl}A & = & \mathbf{W}_{x_1} \otimes \ldots \otimes \mathbf{W}_{x_n} \\ B & = & \mathbf{W}_{x_1} \multimap \ldots \multimap \mathbf{W}_{x_n} \multimap (A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},y)+x} \otimes \mathbf{W}_y) \\ \Delta & = & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}\end{array}$$

Now, from π_ε and from a proof corresponding to $\varepsilon_{\mathbb{W}}$ (with conclusion $\vdash_{\mathscr{C}} \mathbf{W}_0$), we construct σ_ε as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{W}_{x_1} \vdash_{\mathscr{C}} \mathbf{W}_{x_1}} \quad \cdots \quad \dfrac{}{\mathbf{W}_{x_n} \vdash_{\mathscr{C}} \mathbf{W}_{x_n}} \quad \pi_\varepsilon : \Delta \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},1)+x} \quad \dfrac{}{\vdash_{\mathscr{C}} \mathbf{W}_0}}{\Delta, \Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},1)+x} \otimes \mathbf{W}_0}\, R_\otimes}{\Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},1)+x} \otimes \mathbf{W}_0}\, \text{Lemma 7.1}}{\vdash_{\mathscr{C}} B[y \leftarrow 0]}\, R_\multimap$$

Similarly, from ρ_i and from a proof corresponding to $c^i_{\mathbb{W}}$ (with conclusion $\mathbf{W}_y \vdash_{\mathscr{C}} \mathbf{W}_{y+1}$), we construct σ_i as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash_{\mathscr{C}} A} \quad \rho_i : \Delta, \mathbf{W}_y \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},y+1)+x} \quad \dfrac{}{\mathbf{W}_y \vdash_{\mathscr{C}} \mathbf{W}_{y+1}}}{A, \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},y)+x}, \Delta, \mathbf{W}_y, \mathbf{W}_y \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},y+1)+x} \otimes \mathbf{W}_{y+1}}\, R_\otimes, W}{B, \Delta, \Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},y+1)+x} \otimes \mathbf{W}_{y+1}}\, L_\multimap, A, U}{B, \Delta \vdash_{\mathscr{C}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},y+1)+x} \otimes \mathbf{W}_{y+1}}\, \text{Lemma 7.1}}{\vdash_{\mathscr{C}} B \multimap B[y \leftarrow y+1]}\, R_\multimap$$

And now we are ready to iterate over the step functions:

$$\dfrac{\dfrac{\sigma_\varepsilon \quad \sigma_1 \quad \ldots \quad \sigma_w}{\mathbf{W}_{x_{n+1}} \vdash_{\mathscr{C}} B[y \leftarrow x_{n+1}]}\, L_\multimap, A, U \quad \dfrac{}{\Delta, B[y \leftarrow x_{n+1}] \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},x_{n+1})+x}}}{\Delta, \mathbf{W}_{x_{n+1}} \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},x_{n+1})+x}}\, U$$

- If $i = j$, there are proofs π_ε and $\pi^1_{\mathbb{W}}, \ldots, \pi^w_{\mathbb{W}}$ with the appropriate conclusions. By Proposition 2.14 we can assume that:

$$\begin{array}{rcl}\pi_\varepsilon & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n} \vdash_{\mathscr{C}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m},0)+x} \\ \pi_1 & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_{n+1}} \vdash_{\mathscr{D}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \\ & \vdots & \\ \pi_w & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_{n+1}} \vdash_{\mathscr{D}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x}\end{array}$$

where $\mathscr{C} = \{x_{k_1} \leq x, \ldots, x_{k_h} \leq x\}$ and $\mathscr{D} = \mathscr{C} \cup \{x_{n+1} \leq x\}$. Applying the substitution $x_{n+1} \mapsto y$ to $\pi_1, \ldots, \pi_w$ and by using Proposition 2.15, we obtain

$$\begin{array}{rcl}\rho_1 & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}, \mathbf{W}_y \vdash_{\mathscr{E}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \\ & \vdots & \\ \rho_w & : & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}, \mathbf{W}_y \vdash_{\mathscr{E}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x}\end{array}$$

where $\mathscr{E} = \mathscr{C} \cup \{y \leq x\}$ For simplicity, we use the following abbreviations:

$$\begin{array}{rcl}A & = & \mathbf{W}_{x_1} \otimes \ldots \otimes \mathbf{W}_{x_n} \\ B & = & \mathbf{W}_{x_1} \multimap \ldots \multimap \mathbf{W}_{x_n} \multimap (A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \otimes \mathbf{W}_y) \\ \Delta & = & \mathbf{W}_{x_1}, \ldots, \mathbf{W}_{x_n}\end{array}$$

Now, from π_ε and from a proof corresponding to $\varepsilon_{\mathbb{W}}$ (with conclusion $\vdash_{\mathscr{E}} \mathbf{W}_0$), we construct σ_ε as follows:

$$\dfrac{\dfrac{\dfrac{\overline{\mathbf{W}_{x_1} \vdash_{\mathscr{E}} \mathbf{W}_{x_1}} \quad \cdots \quad \overline{\mathbf{W}_{x_n} \vdash_{\mathscr{E}} \mathbf{W}_{x_n}} \quad \pi_\varepsilon : \Delta \vdash_{\mathscr{E}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \quad \overline{\vdash_{\mathscr{E}} \mathbf{W}_0}}{\Delta, \Delta \vdash_{\mathscr{E}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \otimes \mathbf{W}_0}\ R_\otimes}{\Delta \vdash_{\mathscr{E}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \otimes \mathbf{W}_0}\ \text{Lemma 7.1}}{\vdash_{\mathscr{E}} B[y \leftarrow 0]}\ R_\multimap$$

Similarly, from ρ_i and from a proof corresponding to $c^i_{\mathbb{W}}$ (with conclusion $\mathbf{W}_y \vdash_{\mathscr{E}} \mathbf{W}_{y+1}$), we construct σ_i as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{A \vdash_{\mathscr{E}} A} \quad \rho_i : \Delta, \mathbf{W}_y \vdash_{\mathscr{E}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \quad \overline{\mathbf{W}_y \vdash_{\mathscr{E}} \mathbf{W}_{y+1}}}{A, \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x}, \mathbf{W}_y, \mathbf{W}_y, \Delta \vdash_{\mathscr{E}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \otimes \mathbf{W}_{y+1}}\ R_\otimes}{B, \Delta, \Delta \vdash_{\mathscr{E}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \otimes \mathbf{W}_{y+1}}\ \text{Lemma 7.1}}{B, \Delta \vdash_{\mathscr{E}} A \otimes \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x} \otimes \mathbf{W}_{y+1}}\ \text{Lemma 7.1}}{\vdash_{\mathscr{E}} B \multimap B[y \leftarrow y+1]}\ R_\multimap$$

And now we are ready to iterate over the step functions:

$$\dfrac{\dfrac{\sigma_\varepsilon : \vdash_{\mathscr{D}} B[y \leftarrow 0] \quad \dfrac{\sigma_i : \vdash_{\mathscr{E}} B \multimap B[y \leftarrow y+1]}{\vdash_{\mathscr{D}} !_{y<x_{n+1}}(B \multimap B[y \leftarrow y+1])}}{\mathbf{W}_{x_{n+1}} \vdash_{\mathscr{D}} B[y \leftarrow x_{n+1}]} \quad \dfrac{}{\Delta, B[y \leftarrow x_{n+1}] \vdash_{\mathscr{D}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x}}}{\Delta, \mathbf{W}_{x_{n+1}} \vdash_{\mathscr{D}} \mathbf{W}_{q(x_{j_1},\ldots,x_{j_m})+x}}\ U$$

- If $i > j$, the proof is similar to the case $i < j$.

Observe how, in all the three cases, the proof corresponding to f is structurally the same. This concludes the proof. □

Quite interestingly, the proof of Theorem 7.2 is very similar in structure to the proof of polynomial time soundness for BC given in [3], which is based on the following observation: the size of the output of a BC function is bounded by a polynomial on the sizes of normal arguments plus the *maximum* of sizes of safe arguments. This cannot be formalized in BLL, because the resource polynomials do not include any function computing the maximum of its arguments. On the other hand, this can be captured in QBAL by way of constraints.

Proposition 7.3. *The correspondence* $\langle\cdot\rangle$ *is weakly compositional.*

Proof. The proof is essentially identical to the one of Proposition 6.2. □

8. Conclusions

We presented QBAL, a new ICC system embedding two distinct and unrelated systems for impredicative recursion in the sense of [8], namely ramified recurrence and non-size increasing computation. QBAL allows to overcome the main weakness of BLL, namely that all resource variables are global. In the authors' view, this constitutes the first step towards unifying ICC systems into a single framework. The next step consists in defining an embedding of light linear logic into QBAL and the authors are currently investigating on that.

References

[1] Andrea Asperti and Luca Roversi. Intuitionistic light affine logic. *ACM Transactions on Computational Logic*, 3(1):137–175, 2002.
[2] Henk Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. Studies in Logic and the Foundations of Mathematics. North Holland, 1984.
[3] Stephen Bellantoni and Stephen Cook. A new recursion-theoretic characterization of the polytime functions. *Computational Complexity*, 2:97–110, 1992.
[4] Vincent Danos and Jean-Baptiste Joinet. Linear logic and elementary time. *Information and Computation*, 183(1):123–137, 2003.
[5] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
[6] Jean-Yves Girard. Light linear logic. *Information and Computation*, 143(2):175–204, 1998.
[7] Jean-Yves Girard, Andre Scedrov, and Philip Scott. Bounded linear logic: A modular approach to polynomial-time computability. *Theoretical Computer Science*, 97(1):1–66, 1992.
[8] Martin Hofmann. Programming languages capturing complexity classes. *SIGACT News Logic Column*, 9, 2000. 12 pp.
[9] Martin Hofmann. Linear types and non-size-increasing polynomial time computation. *Information and Computation*, 183(1):57–85, 2003.
[10] Martin Hofmann and Philip Scott. Realizability models for BLL-like languages. *Theoretical Computer Science*, 318(1-2):121–137, 2004.
[11] Yves Lafont. Soft linear logic and polynomial time. *Theoretical Computer Science*, 318(1-2):163–180, 2004.
[12] Daniel Leivant. A foundational delineation of computational feasiblity. In *Sixth IEEE Symposium on Logic in Computer Science, Proceedings*, pages 2–11, 1991.
[13] Daniel Leivant. Stratified functional programs and computational complexity. In *20th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Proceedings*, pages 325–333, 1993.
[14] Jean-Yves Marion and Jean-Yves Moyen. Efficient first order functional program interpreter with time bound certifications. In *Logic for Programming and Automated Reasoning, 7th International Conference, Proceedings*, pages 25–42, 2000.
[15] Andrzej Murawski and Luke Ong. On an interpretation of safe recursion in light affine logic. *Theoretical Computer Science*, 318(1-2):197–223, 2004.

[16] Ulrich Schöpp. Stratified bounded affine logic for logarithmic space. In *22nd IEEE Symposium on Logic in Computer Science, Proceedings*, pages 411–420, 2007.
[17] Anne Troelstra and Helmut Schwichtenberg. *Basic proof theory.* Cambridge University Press, 1996.
[18] Christopher Wadsworth. Some unusual λ-calculus numeral systems. In J.P. Seldin and J.R. Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism.* Academic Press, 1980.
[19] Hongwei Xi. Dependent ML. An approach to practical programming with dependent types. *J. Funct. Program.*, 17(2):215–286, 2007.