Graded Multicategories of Polynomial-time Realizers

R.A.G. Seely[*]
Department of Mathematics and Computer Science
John Abbott College, Ste. Anne de Bellevue
and
McGill University, Montréal, Québec

Extended Abstract—Preliminary Version

Abstract

We present a logical calculus which imposes a grading on a sequent-style calculus to account for the runtime of the programmes represented by the sequents. This system is sound for a notion of polynomial-time realizability. An extension of the grading is also considered, giving a notion of "dependant grades", which is also sound. Furthermore, we define a notion of closed graded multicategory, and show how the structure of polynomial-time realizers has that structure.

0 Introduction

In [4], a restricted notion of realizability is defined, a special case of which is polynomial-time realizability: this is like Kleene's original realizability, save for three features. First, closed atomic formulae are realized only by realizers that express a reason for the "truth" (or provability) of the formula, unlike Kleene's system which only reflects the fact that the formula is provable. Second, open formulae are treated as the corresponding closed formulae with all free variables universally quantified simultaneously. (There is a difference between the quantifiers $\forall\langle\xi,\eta\rangle$ and $\forall\xi\forall\eta$.) And third, the realizers code polynomial-time ("p-time") functions, rather than arbitrary recursive functions.

In [4], only the p-time realizability of single formulae is discussed—in [5] these notions are extended to logical rules, to give a sequent calculus that is sound for p-time realizability. This sequent calculus is much like Gentzen's formulation for intuitionist logic, with three main points of difference, which we summarize again here. First, a sequent of the form $A, B \longrightarrow C$ is interpreted as if it were a formula $A \supset (B \supset C)$, rather than $(A \wedge B) \supset C$ (which would be the Gentzen interpretation). As was shown in [4], these are not equivalent. Indeed, if $\Vdash (A \wedge B) \supset C$ then $\Vdash A \supset (B \supset C)$, but not conversely.

[*]This work was done following a visit to Monash University and is based on the joint work done there by J. N. Crossley, G.L. Mathai, and the author, who wishes to express his thanks to those two for their many kindnesses during his visit.

Research partially supported by a grant from Le Fonds F.C.A.R., Québec.

($\Vdash A$ means "A is realizable"; similarly $e \Vdash A$ means "e realizes A".) Second, among the structure rules we keep thinning, but drop exchange and contraction, (roughly the opposite of Girard's linear logic [7].) Again, it was shown in [4] that one could have $\Vdash A \supset (B \supset C)$ without having $\Vdash B \supset (A \supset C)$.

Our structure is then a closed multicategory [9, 10] with finite products and coproducts: $\supset$ gives the internal hom but this is not a hom for the product structure given by $\wedge$. We do not have a tensor $\otimes$—the comma in the sequent notation takes that role—but if we did, it would not be symmetric. Furthermore, it would not satisfy the expected axiom $A, B \longrightarrow A \otimes B$, for we could then deduce $\Vdash A \otimes B \equiv A \wedge B$, which is false. However, $\Vdash A, B \longrightarrow A \wedge B$ is valid (take $C = A \wedge B$ in the result quoted above).

Third, we extend the usual notion of "sequent" by the addition of variable declarations. A variable declaration $(x :)$ delimits the scope of a free variable x in a sequent, and gives finer control over the functional aspect of realizing open formulae. For example, suppose B is a closed formula and $A(x)$ contains exactly x as free variable. It is *not* the case that $\Vdash B \supset A(x)$ implies $\Vdash B \supset \forall\xi A(\xi)$, according to the definitions of [4]. The point is that such an implication involves an exchange in the order of the realizers m, a standard numeral realizing x, and $b \Vdash B$—in the first m comes first, while in the second b does. If we can shift the functional dependence so that m only enters when wanted then this problem will disappear. The (valid) rule corresponding to our discussion then becomes

$$\frac{\Gamma, (x :) \longrightarrow A(x)}{\Gamma \longrightarrow \forall\xi A(\xi)}$$

where $(x :)$ indicates the beginning of the scope of the variable x, (its end being the end of the sequent), Γ being a finite sequence of formulae (and possibly other variable declarations). Such variable declarations should be thought of as the eponymous statements in computer programmes; as in programming, a variable may only appear locally, and so we wish to capture that locality in the functionality of our realizers. In the above rule, both lines in fact are realized by the same realizer.

In this note, we shall modify the calculus presented in [5] by the introduction of grading: instead of the usual implication $A \supset B$, we shall have a countable set of different "graded" implications, $A \overset{k}{\supset} B$, which indicate a bound on the degree of the runtime of the relevant algorithm. These grades will, initially, be natural numbers, as in the graded λ-calculus of [11], but we shall see that this requirement leads to a weakening of the system of [5], and so at the end of this note we shall present a variant in which the grades may be natural-number-valued functions, dependant on "previously realized data".

Aside The use of dependent grades is really the point of the exercise, since it is the ability to capture instances of "dependent realizability" that distinguishes [4, 5] from [11]. For example, in [4] there is an example (my "Favorite Example") that occurs frequently. In the notation of this paper, it can be given in these three flavours:

$$(x :), (y :) \longrightarrow x^{y+1} = x^y \cdot x \qquad (1)$$
$$(y :), (x :) \longrightarrow x^{y+1} = x^y \cdot x \qquad (2)$$
$$(x, y :) \longrightarrow x^{y+1} = x^y \cdot x \qquad (3)$$

(These examples may also be presented as quantified formulae.) Of course, there is no way that Examples 1, 3 can be realizable in polynomial time, but with dependent realizers, Example 2 *is* realizable in polynomial time. However, in the system of this paper, with integer grades, we cannot realize Example 2 because the second realizer (of $(x:) \longrightarrow x^{m+1} = x^m \cdot x$) depends on the first realizer (the m realizing y) for the degree of its runtime. However, at the time of this writing, [5] is still in preparation, and so I have thought it best to concentrate on the integer-graded system for the time being. This system is in some ways an extension of [11], from which it derives its inspiration, but there are some differences, as the reader will see.

Analagous to the graded implication, we shall have graded universal quantifiers $\overset{k}{\forall}\xi A(\xi)$. We mentioned in [5] that the usual quantifier $\forall\xi A(\xi)$ could be thought of as $(x:) \supset A(x)$; analagously $\overset{k}{\forall}\xi A(\xi)$ may be thought of as $(x:) \overset{k}{\supset} A(x)$. Further, the sequents in [5] are directly related to the notion of implication, and so in the graded system we shall have graded sequents: for example $A, B, C \xrightarrow{\mathbf{k}} D$ will be interpreted as $A \overset{k_1}{\supset} (B \overset{k_2}{\supset} (C \overset{k_3}{\supset} D))$, where $\mathbf{k} = \langle k_1, k_2, k_3\rangle$.

Finally, we shall define a notion of graded multicategory, which extends Lambek's notion of a multicategory by associating with each morphism an index or grade in a coherent manner. In fact, the graded multicategorical structure suitable for the graded sequent calculus described above will have much more structure, *e.g.* it will be "graded closed" since it has a "graded internal hom" given by the graded implication. Of course, all this is directly analagous to the ungraded case, and follows the paradigm case of a Gentzen multicategory [10]. (Also, in the intended example the grading is filtered—if a morphism has grade $\mathbf{k}$ then it also has grade $\mathbf{k}'$, for any $\mathbf{k}' \geq \mathbf{k}$, in a suitable ordering. However, it is not clear this should be part of the general definition.)

One final point: for simplicity, in this note we shall not consider the structure of the natural numbers, and so shall make no reference to the non-logical axioms needed to describe that structure. Of course, this is a serious oversight, but that structure (essentially Buss's axioms ([1, 2], see also [3]) together with the expected equality axioms) is discussed in [5], as well as in [11] *mutatis mutandis.*

1 Polynomial-time realizers

We recall the basic definitions from [4] and [5]—for full details, the reader should refer to those papers.

1.1 Realizers

We assume a formal language $\mathcal{L}$ with constants including 0, function letters including $+$, $\cdot$, and *exp* (exponentiation), and predicate letters including $=$ (equality). As indicated in the Introduction, $\mathcal{L}$ is modified by replacing $\supset$ with $\overset{k}{\supset}$ and $\forall$ with $\overset{k}{\forall}$ (for all k) in the formation rules for formulae. Realizers are taken as (codes of) pairs $e = \langle e_m, e_t\rangle$, where e_m is (the code of) a Turing machine and e_t is (the code of) a polynomial: e acts as e_m except that it turns off after $e_t(|\mathbf{x}|)$ steps, where $|\mathbf{x}|$ is the length of the input. In the following, "function" means such a realizer, unless otherwise specified. (This includes constants, as functions of 0 variables.)

We shall write $e(f)$ to denote $\{e_m\}(f)$, $\{\ \}$ denoting the Kleene bracket.

Definition 1 *A realizer $e = \langle e_m, e_t \rangle$ is said to have grade k if the logarithm of the degree of e_t is $\leq k$; that is, if*

$$deg(e_t) \leq 2^k.$$

For some of the function symbols, in particular, $+, \cdot$, but not exp, there is associated a realizer e_f with the function symbol f, and likewise for some of the predicate symbols, including $=$, a realizer e_P is associated with P. Terms are realized by this definition:

Definition 2 • *If t is closed:*

1. *If t is a constant c, with realizer e_c, then $e_c \mathrel{||}\!\!- t$.*
2. *If $t = f(t_1, \ldots, t_n)$, $e_i \mathrel{||}\!\!- t_i$ for $i = 1, \ldots, n$, e_f is the realizer of f, then $e_f(e_1, \ldots, e_n) \mathrel{||}\!\!- t$.*

- *If t contains free variables $x_1, \ldots, x_n$, then $e \mathrel{||}\!\!- t$ iff e is (the code of) a function that, for standard numerals $m_1, \ldots, m_n$, gives $e(m_1, \ldots, m_n) \mathrel{||}\!\!- t(m_1, \ldots, m_n)$.*

Definition 3 *Closed atomic formulae are realized by the realizers of predicate symbols: if $e_i \mathrel{||}\!\!- t_i$ (for $i = 1, \ldots, n$) and e_P is the realizer of P, then $e_P(e_1, \ldots, e_n) \mathrel{||}\!\!- P(t_1, \ldots, t_n)$.*

Finally, we modify Kleene's original inductive definition of $e \mathrel{||}\!\!- A$ as in [5], with the addition of grading to account for the graded connective $\overset{k}{\supset}$ and the graded quantifier $\overset{k}{\forall}$.

Definition 4 *$e \mathrel{||}\!\!- A$ in the following situations:*

- *If A is closed:*

  1. *$\bot$ is never realized.*
  2. *A is $B \wedge C$, $e = \langle e_0, e_1 \rangle$, $e_0 \mathrel{||}\!\!- B$ and $e_1 \mathrel{||}\!\!- C$.*
  3. *A is $B \vee C$, $e = \langle e_0, e_1 \rangle$, and either e_0 is 0 and $e_1 \mathrel{||}\!\!- B$ or e_0 is not 0 and $e_1 \mathrel{||}\!\!- C$.*
  4. *A is $B \overset{k}{\supset} C$, the grade of e is k, and, for all f, if $f \mathrel{||}\!\!- B$ then $e(f)$ is defined and $e(f) \mathrel{||}\!\!- C$.*
  5. *A is $\exists \xi B(\xi)$, $e = \langle e_0, e_1 \rangle$, e_0 is a standard numeral and $e_1 \mathrel{||}\!\!- B(x := e_0)$, (where $B(x := e_0)$ denotes the substitution of e_0 for x in B.)*
  6. *A is $\overset{k}{\forall} \xi B(\xi)$, the grade of e is k, and for all standard numerals k, $e(k) \mathrel{||}\!\!- B(x := k)$.*

- *If A is open with free variables exactly $x_1, \ldots, x_n$ and for all standard numerals $m_1, \ldots, m_n$,*

$$e(m_1, \ldots, m_n) \mathrel{||}\!\!- A(x_1 := m_1, \ldots, x_n := m_n).$$

1.2 The sequent calculus

In the logical calculus we shall develop, it will be convenient to assume that all formulae are "homogeneous" in their occurrences of free variables: in forming a compound formula $A\diamond B$, for any connective $\diamond$, we shall suppose A and B have exactly the same free variables, which then also occur in $A\diamond B$. This may be done without loss in expressive power by a liberal use of "dummy free variables"; perhaps the simplest technical way to do this is to add new function symbols to $\mathcal{L}$ corresponding to projections (—see [12] for example, where first order logic with equality is handled this way.) Note that in quantifying a formula, exactly one free variable disappears—*viz.* the one quantified.

Definition 5 *A graded sequent* $\Gamma \overset{\mathbf{k}}{\rightarrow} A$ *consists of a usual sequent* $\Gamma \longrightarrow A$ *together with a finite sequence* $\mathbf{k}$ *(of natural numbers) of the same length as the sequence* Γ*. Recall from [5] that* Γ *is a finite sequence of formulae and variable declarations, where a variable declaration* $(x :)$ *must precede all formulae in which* x *appears. Furthermore, a variable may only be declared once within the sequent.*

Remarks

1. There may be formulae within the scope of x in which x does not appear—such formulae will be treated as if they have 'dummy' occurrences of x.

2. We can declare several variables simultaneously via pairing: $(\mathbf{x} :)$ will mean $(\langle x_1, \ldots, x_n\rangle :)$, where $\mathbf{x} = x_1, \ldots, x_n$. These variables can also be declared sequentially: $(x_1 :), (x_2 :), \ldots, (x_n :)$. These forms of declaration are not equivalent.

Definition 6 *Our theory consists of all sequents generated* [1] *from the following axioms by the following rules:*

Logical Axioms

1. $(\mathbf{x} :), A \xrightarrow{\langle 0,0\rangle} A$ *(where* $\mathbf{x}$ *lists all free variables of* A*.)*
2. $(\mathbf{x} :), A, B \xrightarrow{\langle 0,0,1\rangle} A \wedge B$ *(where* $\mathbf{x}$ *lists all free variables of* $A \wedge B$*.)*
3. $(\mathbf{x} :), \overset{k}{\forall}\xi A(\xi), (x :) \xrightarrow{\langle 0,0,k\rangle} A(x)$ *(where* $\mathbf{x}$ *lists all free variables of* A *other than* x*.)*

Structural Rules

(grade filtering)
$$\frac{\Gamma \overset{\mathbf{k}}{\rightarrow} A}{\Gamma \overset{\mathbf{k}'}{\rightarrow} A}$$

where $\mathbf{k}' \geq \mathbf{k}$ *in the sense that each coordinate* $k_i \geq k'_i$.

[1] As outlined by Definition 7.

(thinning)
$$\frac{\Gamma, \Delta \xrightarrow{\mathbf{k}} A}{\Gamma, B, \Delta \xrightarrow{\mathbf{k}'} A}$$
where $\mathbf{k}'$ *is the sequence* $\mathbf{k}$ *with a 0 inserted in the position corresponding to the position of* B *in the sequent.*

(curry)
$$\frac{\Gamma, (\langle x, y\rangle :), \Delta \xrightarrow{\mathbf{k}} A}{\Gamma, (x :), (y :), \Delta \xrightarrow{\mathbf{k}'} A} \qquad \frac{\Gamma, (\langle y, x\rangle :), \Delta \xrightarrow{\mathbf{k}} A}{\Gamma, (x :), (y :), \Delta \xrightarrow{\mathbf{k}'} A}$$
where $\mathbf{k}'$ *is the sequence* $\mathbf{k}$ *with a 1 inserted in the position corresponding to the position of* $(x :)$ *in the conclusion-sequent.*

(cut)
$$\frac{\Delta, B, \Theta \xrightarrow{\langle \mathbf{k}_1, k, \mathbf{k}_2\rangle} A \qquad \Gamma \xrightarrow{\mathbf{l}} B}{\Delta, \Gamma', \Theta \xrightarrow{\langle \mathbf{k}_1', \mathbf{l}'+k, \mathbf{k}_2\rangle} A}$$
where

- Γ' *is* Γ *with the variable declarations for* B *omitted, (see Remarks following),*
- $\mathbf{l}'$ *is* $\mathbf{l}$ *with the entries corresponding to variable declarations for* B *deleted,*
- $\mathbf{k}_1'$ *is* $\mathbf{k}_1$ *with the entries corresponding to the variable declarations for* B *augmented by adding 1 + the corresponding entries from* $\mathbf{l}$ *(*i.e. *those deleted to get* $\mathbf{l}'$*), and*
- $\mathbf{k}_1', \mathbf{l}' + k$ *means add* k *to the* last *entry in the sequence* $\mathbf{k}_1', \mathbf{l}'$.

Logical Rules

($\overset{k}{\supset}$ L)
$$\frac{\Delta, B, \Theta \xrightarrow{\langle \mathbf{k}_1, k, \mathbf{k}_2\rangle} C \qquad \Gamma \xrightarrow{\mathbf{l}} A}{\Delta, (A \overset{h}{\supset} B), \Gamma', \Theta \xrightarrow{\langle \mathbf{k}_1', 0, \mathbf{l}'+k+h, \mathbf{k}_2\rangle} C}$$
where Γ', $\mathbf{k}'$, $\mathbf{l}'$ *are as in* (cut).

($\overset{k}{\supset}$ R)
$$\frac{\Gamma, A \xrightarrow{\langle \mathbf{k}, k\rangle} B}{\Gamma \xrightarrow{\mathbf{k}} (A \overset{k}{\supset} B)}$$

($\wedge$ L)
$$\frac{\Gamma, A, \Delta \xrightarrow{\mathbf{k}} C}{\Gamma, (A \wedge B), \Delta \xrightarrow{\mathbf{k}'} C} \qquad \frac{\Gamma, B, \Delta \xrightarrow{\mathbf{k}} C}{\Gamma, (A \wedge B), \Delta \xrightarrow{\mathbf{k}'} C}$$
where $\mathbf{k}'$ *is the sequence* $\mathbf{k}$ *with no less than 1 in the position corresponding to the position of* $A \wedge B$ *in the conclusion-sequent.*

($\wedge$ R)
$$\frac{\Gamma \xrightarrow{\mathbf{k}} A \qquad \Gamma \xrightarrow{\mathbf{l}} B}{\Gamma \xrightarrow{max'(\mathbf{k}, \mathbf{l})} A \wedge B}$$
where $max'(\mathbf{k}, \mathbf{l})$ *means take the maximum at each coordinate, and where the last entry is no less than 1.*

(∨ L) $$\frac{\Gamma, A, \Delta \xrightarrow{\mathbf{k}} C \quad \Gamma, B, \Delta \xrightarrow{\mathbf{l}} C}{\Gamma, (A \vee B), \Delta \xrightarrow{max'(\mathbf{k},\mathbf{l})} C}$$

where $max'(\mathbf{k}, \mathbf{l})$ means take the maximum at each coordinate, and where the entry in the $A \vee B$ position is no less than 1.

(∨ R) $$\frac{\Gamma \xrightarrow{\mathbf{k}} A}{\Gamma \xrightarrow{\mathbf{k}'} A \vee B} \qquad \frac{\Gamma \xrightarrow{\mathbf{k}} B}{\Gamma \xrightarrow{\mathbf{k}'} A \vee B}$$

where $\mathbf{k}'$ is the sequence $\mathbf{k}$ with the last position no less than 1.

($\overset{k}{\forall}$ R) $$\frac{\Gamma, (x :) \xrightarrow{\langle \mathbf{k}, k \rangle} A(x)}{\Gamma \xrightarrow{\mathbf{k}} \overset{k}{\forall} \xi A(\xi)}$$

(∃ L) $$\frac{\Gamma, (x :), A, \Delta \xrightarrow{\langle \mathbf{k}, k, l, \mathbf{l} \rangle} B}{\Gamma, \exists \xi A(\xi), \Delta \xrightarrow{\langle \mathbf{k}, max'(k,l), \mathbf{l} \rangle} B}$$

where $max'(k, l)$ means $max(k, l, 1)$.

(∃ R) $$\frac{\Gamma \xrightarrow{\mathbf{k}} A}{\Gamma \xrightarrow{\mathbf{k}'} \exists \xi A(\xi)}$$

where $\mathbf{k}'$ means $\mathbf{k}$ with the last entry no less than 1.

Remarks: There are several restrictions on these rules, as discussed in [5]. Some of the restrictions are no longer relevant, because they are subsumed by the grading. However, the following restrictions remain:

(thinning) B must be a formula whose variables are declared in Γ.

(cut) The variable declarations for B must be identical in Γ and in Δ, and must occur at the beginning of Γ. In the conclusion of the rule, these declarations are dropped from Γ to give Γ'.

($\overset{k}{\supset}$ L) Similar restrictions on variable declarations to those above in (cut).

(∃ L) Implicit in the syntax is that x occurs free only in A, not in Δ nor in B.

(∃ R) Γ must begin with the declaration $(x :)$, where x is the variable quantified.

Definition 7 *A derivation of a graded sequent consists of a finite tree such that each branch ends in an axiom and each step is one of the rules of inference given in Definition 6 above, or is a substitution instance of such an axiom or rule.*

We remark here that by a "substitution instance" of a graded sequent $\Gamma \xrightarrow{\mathbf{k}} A$ we mean

- the replacement throughout the sequent of a free variable x, say, occurring in the sequent, by a *realizable* term $t(\mathbf{x})$ which has only new free variables $\mathbf{x}$, not occurring in the sequent,
- the replacement of the declaration $(x :)$ by the simultaneous declaration $(\mathbf{x} :)$, and

- the replacement of $\mathbf{k}$ by the sequence $\mathbf{k}'$, which differs from $\mathbf{k}$ by adding to the coordinate in the $(x :)$ position the grade of the realizer of t of least grade, (or 1 if that is larger.)

A substitution instance of a rule is defined similarly—take the corresponding substitution instances of the sequents involved in the rule.

Note that we only allow substitution instances of realizable terms.

1.3 Graded realizability

As discussed in the Introduction, we shall treat realizability for sequents as if the sequents consisted of a successive introduction of premisses, *i.e.* a nested sequence of implications. Within these successive hypotheses, a variable declaration amounts, in effect, to another such hypothesis. Finally, a rule is realized by a function (not necessarily p-time, however) that assigns a realizer of the conclusion to a simultaneously-presented tuple of realizers of the premisses of the rule. These points are summarized in the following definitions:

Definition 8 *For a graded sequent $\Gamma \xrightarrow{\mathbf{k}} A$, we define $e \Vdash \Gamma \xrightarrow{\mathbf{k}} A$ inductively:*

1. *$e \Vdash B \xrightarrow{k} A$ iff $e \Vdash B \overset{k}{\supset} A$.*

2. *$e \Vdash (x :) \xrightarrow{k} A$ iff $e \Vdash \overset{k}{\forall} \xi A(\xi)$.*

3. *$e \Vdash B, \Gamma \xrightarrow{\langle k, \mathbf{k} \rangle} A$ iff e is (the code of) a function of grade k which to any $b \Vdash B$, produces an output $e(b) \Vdash \Gamma \xrightarrow{\mathbf{k}} A$.*

4. *$e \Vdash (x :), \Gamma \xrightarrow{\langle k, \mathbf{k} \rangle} A$ iff e is (the code of) a function of grade k which to any standard numeral m, produces an output $e(m) \Vdash \Gamma(x := m) \xrightarrow{\mathbf{k}} A(x := m)$.*

Definition 9 *Given a rule of the form*

$$\frac{P_1, \ldots, P_n}{C}$$

where $P_i (i = 1, \ldots, n)$, C, are sequents, we say f realizes the rule iff f is a function (not necessarily p-time) so that for realizers $e_i \Vdash P_i$, $f(e_1, \ldots, e_n) \Vdash C$.

Proposition 1 *Each of the rules of Definition 6 is realizable.*

Proof In each case we shall define the realizer by giving an equation for the "fully evaluated form". Lower case letters will represent realizers of the corresponding formulae given by upper case letters. We shall write $e(\gamma)$ for $e(a_1)(a_2)\ldots(a_n)$ when $\Gamma = A_1, A_2, \ldots, A_n$. Variable declarations are realized by standard numerals m. Formulae involving functions ($A \overset{k}{\supset} B$ or $\overset{k}{\forall} \xi A(\xi)$, as appropriate) will be realized by p; e, f will denote realizers of the premisses of the rule. Finally we shall denote the realizer of the rule by the same name as the rule.

- $(\text{thin})(e)(\gamma)(b)(\delta) = e(\gamma)(\delta)$
- $(\text{curry})(e)(\gamma)(m)(m')(\delta) = e(\gamma)\langle m, m'\rangle(\delta)$
- $(\text{cut})\langle e, f\rangle(\delta)(\gamma)(\theta) = e(\delta)(f(\gamma))(\theta)$
- $(\overset{k}{\supset}\ \text{L})\langle e, f\rangle(\delta)(p)(\gamma)(\theta) = e(\delta)(p(f(\gamma)))(\theta)$
- $(\overset{k}{\supset}\ \text{R})(e)(\gamma)(a) = e(\gamma)(a)$
- $(\wedge\ \text{L})(e)(\gamma)\langle a, b\rangle(\delta) = e(\gamma)(a)(\delta)$
- $(\wedge\ \text{L})'(e)(\gamma)\langle a, b\rangle(\delta) = e(\gamma)(b)(\delta)$
- $(\wedge\ \text{R})\langle e, f\rangle(\gamma) = \langle e(\gamma), f(\gamma)\rangle$
- $(\vee\ \text{L})\langle e, f\rangle(\gamma)\langle i, a\rangle(\delta) = \begin{cases} e(\gamma)(a)(\delta) & \text{if } i = 0 \\ f(\gamma)(a)(\delta) & \text{if } i \neq 0 \end{cases}$
- $(\vee\ \text{R})(e)(\gamma) = \langle 0, e(\gamma)\rangle$
- $(\vee\ \text{R})'(e)(\gamma) = \langle 1, e(\gamma)\rangle$
- $(\overset{k}{\forall}\ \text{R})(e)(\gamma)(m) = e(\gamma)(m)$
- $(\exists\ \text{L})(e)(\gamma)\langle m, a\rangle(\delta) = e(\gamma)(m)(a)(\delta)$
- $(\exists\ \text{R})(e)(\gamma) = \langle m, e(m)(\gamma)\rangle$ where m is the realization of the variable declaration $(x :)$ in Γ.

Of course, now we must justify the grading given in the rules. For the most part this is based on that in [11], but reflecting the difference in our definition of grading, which makes it unnecessary to go up a grade when adding a routine of low degree runtime to a given routine. Notice in particular the frequent references to "not less than 1"—these are to account for the pairing – unpairing operations—which in [11] require a "+ 1" instead.

The main point here is our grading of cut, which we now illustrate with some examples.

First, consider the simplest case, "composition":

$$\frac{B \xrightarrow{k} A \quad C \xrightarrow{l} B}{C \xrightarrow{k+l} A}$$

This grading is the result of the way polynomial-time functions are composed: the runtime of the composite is given by composing the runtimes (see [4, 5, 11].)

Now, let's extend this one step:

$$\frac{e \Vdash D, B \xrightarrow{k_1, k} A \quad f \Vdash C \xrightarrow{l} B}{g \Vdash D, C \xrightarrow{k_1, l+k} A}$$

The programme g is as follows. First we compute $g(d)$, for $d \Vdash D$. This is: compute $e(d)$, (this has a runtime of grade k_1, by assumption); output $e(d)$ composed with f, (this is the previous algorithm, and amounts to just providing the given f with a tail end. Notice that we are not yet running this, so we have merely added some constant to the runtime, and have not changed the grade.)

Next we compute $g(d)(c)$ for $c \Vdash C$. This is where we actually run the composition above, and so as we saw earlier, this step has grade $l + k$.

Next, consider a simple case with variable declarations:

$$\frac{e \Vdash (x:), B \xrightarrow{\langle k_1, k\rangle} C \quad f \Vdash (x:) \xrightarrow{l} B}{g \Vdash (x:) \xrightarrow{k_1+l+1+k} C}$$

(Notice how the grading instructions work here: we have, in the notation of Definition 6, that $\mathbf{k}_1 = k_1$, $\mathbf{l} = l$, $\mathbf{l}' = \langle\rangle$, $\mathbf{k}'_1 = \langle k_1 + l + 1\rangle$, and $\mathbf{k}'_1, \mathbf{l}' + k = \langle k_1 + l + 1 + k\rangle$, as shown.)

The point about this instance of cut is that now we have an instance of an implicit *contraction* rule at play here: we have lost an hypothesis $(x:)$ in the conclusion (because we cannot declare variables more than once in a sequent). This will require us to step up one degree to access an appropriate Universal Turing Machine (UTM) which can perform the required joint application/composition operation for us. So, the algorithm $g(m)$, for $m \Vdash (x:)$, is to calculate both $e(m) \Vdash B \xrightarrow{k} C$ and $f(m) \Vdash B$, plug these into the UTM for grade k to compose the results and get a realizer for C. (Notice we actually run this, we do not just output the instructions "compose $e(m)$, $f(m)$". Calculating $e(m)$ takes runtime of grade k_1, calculating $f(m)$ takes runtime of grade l, and since $e(m)$ itself has grade k, using the UTM pushes us up a grade to give an additional grade of $k + 1$ (see [11]), as shown above.)

Now an induction on the structure of the sequents in the cut rule, similar to that above (going from the simple composition to the one-step extension), gives the general grading formula. As an example, consider:

$$\frac{e \Vdash A_1, (y:), (x:), A_2, B \xrightarrow{\langle k_1,k_2,k_3,k_4,k\rangle} C \quad f \Vdash (x:), (z:), D \xrightarrow{\langle l_1,l_2,l_3\rangle} B}{g \Vdash A_1, (y:), (x:), A_2, (z:), D \xrightarrow{\langle k_1,k_2,k_3+l_1+1,l_2,k_5+l_3\rangle} C}$$

The crucial step is the third, where $m_x \Vdash (x:)$ is read, $e(a_1)(m_y)(m_x)$ and $f(a_1)(m_y)(m_x)$ are calculated, plugged into an appropriate UTM, and so composed to produce a realizer of $A_2, (z:), D \longrightarrow C$. (Note that if we wish, we can always bring hypotheses to the right hand side, via $(\overset{k}{\supset}$ R) and $(\overset{k}{\forall}$ R).)

Theorem 1 *The sequent calculus given in Definition 6 is sound with respect to realizability.*

Proof All that remains to be shown is that the axioms are realizable. But Axiom 1 is trivially realized. Also Axiom 2 is proven in [4], and discussed in the introduction here. Axiom 3 is virtually the identity, given our interpretation of $\overset{k}{\forall}$ and $\xrightarrow{k}$. The grading is quite straightforward.

2 Multicategories

In this section, we consider the structure of the propositional part of the calculus of polynomial-time realizers. In particular, we shall ignore variable declarations, and so the "primes" in the cut rule may be dropped. This gives us the structure of a "graded multicategory". (Due to deadline constraints, I must leave the strange structure of the quantifiers to a promised sequel. It is clear that we have something more complicated there than a straightforward notion of "weak adjoint", especially with the universal quantifiers $\overset{k}{\forall}$. The existential quantifier is a little more straightforward, and will be remarked upon briefly at the end of the paper.)

2.1 Definitions

Recall from [9, 10] that

Definition 10 *A multicategory* $\mathbf{C}$ *consists of a set* $Ob(\mathbf{C})$ *of objects and a set* $M\varphi(\mathbf{C})$ *of morphisms, (also called arrows, multimorphisms, ...,) just like a category, except that the source of a morphism is a finite sequence of objects, rather than a single object. The target of a morphism is a single object as usual. So we have the two maps*

$$source : M\varphi(\mathbf{C}) \longrightarrow Ob(\mathbf{C})^*$$

$$target : M\varphi(\mathbf{C}) \longrightarrow Ob(\mathbf{C})$$

(where X^* *= the free monoid generated by* X*.)*

As with categories, we have identity morphisms $1_A : A \rightarrow A$*, and a notion of composition which is most simply given by the following "inference diagram":*

$$\frac{\Gamma, A, \Delta \xrightarrow{g} B \quad \Theta \xrightarrow{f} A}{\Gamma, \Theta, \Delta \xrightarrow{g\langle f\rangle} B}$$

We have the following axioms:

1. $$\Gamma \xrightarrow{f} A \quad = \quad \frac{A \xrightarrow{1_A} A \quad \Gamma \xrightarrow{f} A}{\Gamma \xrightarrow{1_A\langle f\rangle} A}$$

2. $$\Gamma, A, \Delta \xrightarrow{g} B \quad = \quad \frac{\Gamma, A, \Delta \xrightarrow{g} B \quad A \xrightarrow{1_A} A}{\Gamma, A, \Delta \xrightarrow{g\langle 1_A\rangle} B}$$

3. $$\frac{\Phi, B, \Psi \xrightarrow{h} C \qquad \dfrac{\Gamma, A, \Delta \xrightarrow{g} B \quad \Theta \xrightarrow{f} A}{\Gamma, \Theta, \Delta \xrightarrow{g\langle f\rangle} B}}{\Phi, \Gamma, \Theta, \Delta, \Psi \xrightarrow{h\langle g\langle f\rangle\rangle} C}$$

$$= \quad \frac{\dfrac{\Phi, B, \Psi \xrightarrow{h} C \quad \Gamma, A, \Delta \xrightarrow{g} B}{\Phi, \Gamma, A, \Delta, \Psi \xrightarrow{h\langle g\rangle} C} \qquad \Theta \xrightarrow{f} A}{\Phi, \Gamma, \Theta, \Delta, \Psi \xrightarrow{h\langle g\rangle\langle f\rangle} C}$$

4. $$\dfrac{\dfrac{\Phi, A, \Theta, B, \Psi \xrightarrow{h} C \quad \Delta \xrightarrow{g} B}{\Phi, A, \Theta, \Delta, \Psi \xrightarrow{h\langle g\rangle} C} \qquad \Theta \xrightarrow{f} A}{\Phi, \Gamma, \Theta, \Delta, \Psi \xrightarrow{h\langle g\rangle\langle f\rangle} C}$$

$$= \quad \dfrac{\dfrac{\Phi, A, \Theta, B, \Psi \xrightarrow{h} C \quad \Gamma \xrightarrow{f} A}{\Phi, \Gamma, \Theta, B, \Psi \xrightarrow{h\langle f\rangle} C} \qquad \Delta \xrightarrow{g} B}{\Phi, \Gamma, \Theta, \Delta, \Psi \xrightarrow{h\langle f\rangle\langle g\rangle} C}$$

(We have made slight changes to the notation of [10], mainly in reversing the order of presenting the composition, in order to make the comparison with Definition 6 more obvious. As Lambek pointed out, there is an ambiguity in the notation $g\langle f\rangle$, which we shall ignore, appealing to diagrams when necessary.)

As an example of a multicategory, we offer the following:

Example: Finite sequences of natural numbers form a multicategory $\mathbf{N}^{<\omega}$, which has one object (which we shall not bother to name), and whose morphisms are finite sequences of natural numbers. We shall denote such a morphism by the sequence concerned, and not refer to the objects giving the source and target, as they are obvious. The identity is the singleton $\langle 0\rangle$; composition is given by the diagram:

$$\dfrac{\langle \mathbf{k}_1, k, \mathbf{k}_2\rangle \qquad\qquad \mathbf{l}}{\langle \mathbf{k}_1, \mathbf{l} + k, \mathbf{k}_2\rangle}$$

Of course, this example is inspired by the grading of Definition 6. It is a simple exercise to verify the four axioms in this case.

Definition 11 *A graded multicategory* $\langle \mathbf{C}, G\rangle$ *consists of a multifunctor* G *between a multicategory* $\mathbf{C}$ *and* $\mathbf{N}^{<\omega}$.

This means that to every morphism $f : \Gamma \longrightarrow A$ of $\mathbf{C}$ there is associated a "grading" $G(f) = \mathbf{k}$, usually denoted by a superscript, as we have been doing for our graded sequent calculus:

$$f : \Gamma \xrightarrow{\mathbf{k}} A.$$

The point of this grading being a multifunctor is that it should be defined in such a way as to make the grading of a composite as suggested by the cut rule (of Definition 6), *viz.*:

$$\dfrac{g : \Gamma, A, \Delta \xrightarrow{\langle \mathbf{k}_1, k, \mathbf{k}_2\rangle} B \quad f : \Theta \xrightarrow{\mathbf{l}} A}{g\langle f\rangle : \Gamma, \Theta, \Delta \xrightarrow{\langle \mathbf{k}_1, \mathbf{l}+k, \mathbf{k}_2\rangle} B}$$

And finally, the grading ought to respect the axioms 1 to 4 of Definition 10.

I have presented this definition in such a way as to leave open the possibility of other types of grading than by sequences of natural numbers. However, in the absence of interesting examples, I prefer to leave this definition in this more restrictive form. A point, however: it will not be correct to merely change the target multicategory in order to capture the notion of dependent grading, since there must be a greater degree of interaction between the morphism $f : \Gamma \longrightarrow A$ being graded and the grade $\mathbf{k} = G(f)$. For instance, if $f : A, B \longrightarrow C$, then $G(f) = \langle k_1, k_2 \rangle$, where k_1 is a number, and k_2 is a number-valued function that may depend on both k_1 and $a \in A$, (where $\in$ is a suitable "membership" or "typing" relation—in our main example, $\in$ would be $\Vdash$.)

2.2 The main example

Now we come to the point of this note: it is clear from the notation that what is intended is to have the structure of realizers form a multicategory, with grading as given in Definition 6.

So we define the multicategory $\mathbf{C}$ as follows:

$Ob(\mathbf{C})$ consists of all formulae of $\mathcal{L}'$, the propositional part of our logical system. (In fact we shall really be dealing with a set $[A]$ of graded realizers of A, rather than with A itself, for any formula A—however, the identification is generally harmless.) $M\varphi(\mathbf{C})$ consists of all graded realizers of the corresponding sequents: this means that we are not allowing dependencies of the sort exemplified by Favorite Example 2. To have the axioms of Definition 10, we must factor out by those equations, in the usual manner. The grading G is of course then given by the "least grading" function, (*i.e.* take the least possible entry in each coordinate.) (Here is where we have included less structure in the definition of grading than our model shows, by ignoring the filtering.) We have then already shown that

Proposition 2 *The structure $\langle \mathbf{C}, G \rangle$ is in fact a graded multicategory.*

2.3 Structured graded multicategories

In [10], Lambek defines the notion of a *right closed multicategory, with Cartesian products and coproducts.* Here we sketch the corresponding notions for the graded case, and indicate how the main example has these properties.

Definition 12 *A right closed graded multicategory is a graded multicategory $(\mathbf{C}, G)$ with a family of (graded) internal homs $\overset{k}{\supset}$, (one for each natural number k), and graded morphisms*

$$ev_{AB} : A \overset{k}{\supset} B, A \xrightarrow{\langle 0,k \rangle} B$$

inducing a bijection

$$\frac{f \quad : \quad \Gamma \xrightarrow{\mathbf{k}} (A \overset{k}{\supset} B)}{ev_{AB}\langle f \rangle \quad : \quad \Gamma, A \xrightarrow{\langle \mathbf{k},k \rangle} B}$$

This means to each $f : \Gamma, A \xrightarrow{\langle \mathbf{k},k \rangle} B$ there is a (unique) $f^\star : \Gamma \xrightarrow{\mathbf{k}} A \overset{k}{\supset} B$ such that $ev_{AB}\langle f^\star \rangle = f$. (Uniqueness amounts to $(ev_{AB}\langle f \rangle)^\star = f$.) (The equations for grading are automatic.)

In our situation, $f^\star$ is constructed by the ($\overset{k}{\supset}$ R) rule, and ev_{AB} is constructed from ($\overset{k}{\supset}$ L) as follows:

$$\frac{B \xrightarrow{0} B \quad A \xrightarrow{0} A}{A \overset{k}{\supset} B, A \xrightarrow{\langle 0,k\rangle} B}$$

(We might note that **C** is not left closed in any meaningful sense, as that would require some exchange:)

$$\frac{A, \Gamma \longrightarrow C}{\Gamma \longrightarrow A \not\subset C}$$

The structure of Cartesian products and coproducts is straightforward, and can be found in [10] without alteration. Of course, the product structure is not a tensor for $\overset{k}{\supset}$. However, **C** does have some extra structure given by the axiom scheme $A, B \xrightarrow{\langle 0,1\rangle} A \wedge B$.

3 Final remarks

3.1 Dependent grading

We have made reference throughout to allowing the grades of realizers of formulae that ocurr in sequents to depend on realizers of previous formulae in the sequent and on previous grades. Some final remarks are all that need be made here.

1. With dependent grading, we must now be more careful with the restrictions imposed on the deduction rules in Definition 6. Of course, the restriction from [5] must be imported: (we use the notation of Definition 6, but now understanding that the grades may be functions of appropriate arities.)

($\exists$ L) k must be a constant, (or at least, must be a bounded function.)

In addition to this we also have two further restrictions caused by the fact that $\overset{k}{\supset}$ and $\overset{k}{\forall}$ are graded by constants:

($\overset{k}{\supset}$ R) k must be a constant.

($\overset{k}{\forall}$ R) k must be a constant.

2. In the graded system described in Definition 6, the rule ($\overset{k}{\supset}$ R) in effect defines a bijection. In a system with dependent grades, this would no longer be possible, unless one replaced $\overset{k}{\supset}$ with $\supset$—but as pointed out by [11], there are problems there. It is hoped that this will be discussed in the final version of [5]. A similar remark holds for $\overset{k}{\forall}$.

3. From the preceeding remarks, it is clear that the multicategorical structure for the dependently graded situation is no longer as simple. (For example, the closed structure has been weakened by the absence of the bijections referred to above.) I have not had time to look into this properly, so that will have to await a sequel.

3.2 The quantifiers

The structure of the quantifiers is more complicated than I have time to give it at the time of writing, so as I mentioned earlier must await a sequel. The main complication is that with variable declarations, the definition of "graded multicategory" must change, in view of the implicit contraction allowed by the cut rule. A further complication with the universal quantifiers (not to mention that there are many of these) comes from the absence of a ($\forall$ L) rule—in place of that, we have a "counit" represented by Axiom 3 of Definition 6.

However, the existential quantifier does have some more recognizable structure—ignoring for the moment the problems with cut, we can see the possibility of a bijection

$$\frac{\Gamma, A, \pi\Delta \longrightarrow \pi B}{\Gamma, \Sigma A, \Delta \longrightarrow B}$$

where Σ is the multifunctor corresponding to the existential quantifier, and π the multifunctor corresponding to adding a dummy free variable. But we must leave the details of this for another day.

References

[1] S. Buss, *Bounded Arithmetic,* Doctoral dissertation, Princeton University, 1985.

[2] S. Buss, "The polynomial time hierarchy and intuitionistic bounded arithmetic", Proceedings of the First Symposium on Structures and Complexity, 1986, (IEEE Publications).

[3] S. Cook and A. Urquhart, "Functional interpretations of feasibly constructive arithmetic", Technical Report 210/88, Department of Computer Science, University of Toronto, 1988.

[4] J.N. Crossley, "Proofs, programs and run-times", Preprint, Monash University, 1989.

[5] J.N. Crossley, G.L. Mathai, and R.A.G. Seely, "A logical calculus for polynomial time realizability", Preprint, Monash University, 1989, (final version in preperation).

[6] G. Gentzen, "Investigations into logical deductions", in M.E. Szabo (ed.), *The Collected Papers of Gerhard Gentzen*, North-Holland, 1969.

[7] J.-Y. Girard, "Linear logic", J. Theoretical Computer Science 50 (1987), 1 – 102.

[8] J. Lambek, "Deductive systems and categories I", J. Math. Systems Theory 2 (1968), 278 – 318.

[9] J. Lambek, "Deductive systems and categories II", Springer LNM 86 (1969), 76 – 122.

[10] J. Lambek, "Multicategories revisited", Proceedings of the A.M.S. Summer Conference on Categories in Logic and Computer Science, Boulder 1987.

[11] A. Nerode, J.B. Remmel, and A. Scedrov, "Polynomially graded logic", Proceedings of the Fourth Symposium on Logic in Computer Science, Asilomar 1989 (IEEE Publications).

[12] R.A.G. Seely, "Hyperdoctrines, natural deduction, and the Beck condition", Zeitsch. f. math. Logik und Grundlagen d. Math. 29 (1983), 505 – 542.