Functionals Using Bounded Information and the Dynamics of Algorithms

Serge Grigorieff
LIAFA, CNRS & Université Paris 7
seg@liafa.jussieu.fr

Pierre Valarcher
LACL, Université Paris-Est
pierre.valarcher@u-pec.fr

June 30, 2012

Contents

Abstract

We consider computable functionals mapping the Baire space into the set of integers. By continuity, the value of the functional on a given function depends only on a "critical" finite part of this function. Care: there is in general no way to compute this critical finite part without querying the function on an arbitrarily larger finite part! Nevertheless, things are different in case there is a uniform bound on the size of the domain of this critical finite part. We prove that, modulo a quadratic blow-up of the bound, one can compute the value of the functional by an algorithm which queries the input function on a uniformly bounded finite part. Up to a constant factor, this quadratic blow-up is optimal. We also characterize such functionals in topological terms using uniformities.

As an application of these results, we get a topological characterization of the dynamics of algorithms as modeled by Gurevich's Abstract State Machines.

1 Introduction

1.1 A combinatorico-topological problem...

Consider the discrete topology on $\mathbb{N}$ and the usual Baire topology on $\mathbb{N}^{\mathbb{N}}$ generated by the basis of clopen sets

$$[u] = \{f \in \mathbb{N}^{\mathbb{N}} \mid f \textit{ extends } u\}$$

where u varies over partial functions $\mathbb{N} \to \mathbb{N}$ with finite domains. As is well-known, a functional $\Phi : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is continuous if and only if $\mathbb{N}^{\mathbb{N}}$ is covered by the clopen sets $[u_i]$'s associated to some family $\pi = (u_i)_{i\in\mathbb{N}}$ such that Φ is constant on each $[u_i]$. The property we are interested in is:

(*) *There is a uniform bound on the size of the* $\mathtt{Dom}(u_i)$*'s.*

Let us consider an example.

Example 1.1. 1. Let $\Phi(f) = f(\alpha(f(0), f(1)))$ where $\alpha : \mathbb{N}^2 \to \mathbb{N}$ is some fixed function. To compute $\Phi(f)$ we only need 3 values of f, namely those at 0, 1 and $\alpha(f(0), f(1))$. For x, y, z in $\mathbb{N}$, let $u_{x,y,z}$ be the partial function $\mathbb{N} \to \mathbb{N}$ with domain $\{0, 1, \alpha(x, y)\}$ such that $u_{x,y,z}(0) = x$, $u_{x,y,z}(1) = y$ and, in case $\alpha(x, y) \neq 0, 1$, $u_{x,y,z}(\alpha(x, y)) = z$. The $[u_{x,y,z}]$'s, x, y, z in $\mathbb{N}$, constitute a partition of $\mathbb{N}^{\mathbb{N}}$ in clopen sets on which Φ is constant. Moreover, the $u_{x,y,z}$'s are functions with two or three elements in their domains.

In this example we see that what is used, i.e. the u of condition (*), coincides with what is queried about f during the computation. Though this coincidence may seem to be an ovbious necessity, it is not the case. A priori, the computation may query f on a very large set and use only a much smaller part of it (an example is given in §2.3). Indeed, condition (*) does not give any means to go from f to some u having a size $\leq k$ domain such that $f \in [u]$. Even with a computable enumeration of the u_i's, the natural way to find a convenient $[u_i]$ in which lies the argument f is a loop which successively queries f on $\texttt{Dom}(u_0)$, $\texttt{Dom}(u_1)$, ..., until u_i is found. But this loop queries f on an unbounded set of points!

What we prove (cf. Theorems 2.10 and 3.3):

1. *Example 1.1 is paradigmatic: functionals definable in that way are the sole functionals satisfying property (*).*

2. *If (*) is true with bound k (for the size of the $\texttt{Dom}(u_i)$'s) then one can define Φ à la Example 1.1 so that $\Phi(f)$ is obtained by querying the values of the argument f on at most k^2 points.*

3. *In Point (2), the quadratic blow-up k^2 is optimal.*

4. *If Φ is computable and (*) is true with bound k for a computably enumerable π then one can define Φ à la Example 1.1 with computable auxiliary fixed functions (like the α in Example 1.1) so that $\Phi(f)$ is obtained by querying the values of f on at most $2k^2$ points.*

5. *Condition (*) is topological: it is uniform continuity with respect to a transitive uniformity on the Baire space which gives the Baire topology but strictly refines the uniformity of the usual Baire metric.*

Thus, we prove a kind of fixed point process, modulo a quadratic blow-up: one can go from f to a convenient restriction u of f to a size $\leq k^2$ domain by querying f on the sole points of $\texttt{Dom}(u)$.

1.2 ... also relevant to the question: what is an algorithm?

Though the problem can be seen as relevant to the theory of continuous functionals and to type 2 computability, we came to it via the question of modelling the notion of algorithm.

The question *What is an algorithm?* has not been answered by the solution given to the question *What is a computable function?*. It long remained a pending question up to the solution brought by Yuri Gurevich with Abstract State Machines (ASM). In §4.1, 4.2 we recall the successive trials to answer the question and what is the notion of ASM.

For this introduction, let us just say that we consider the dynamics of an algorithm as a functional from a fixed product of function spaces into itself. What are these function spaces? They model the environment: each environment parameter (item, unbounded array of items...) is viewed as a function (arity 0 is allowed) over finitely many countable sorts. Thus, letting the M_i's be the involved fixed sorts, the dynamics of an algorithm is a functional $\Psi : T \to T$ where

$$T = \prod_{i=1}^{i=q} \left((\prod_{j \in J_i} M_j) \to M_{\ell_i} \right)$$

is a space in general homeomorphic to the usual Baire space $\mathbb{N}^{\mathbb{N}}$. Now, following Gurevich's analysis [11], the algorithms we consider manipulate their environment in quite a gentle way: for some fixed k, the transition step leads from $\vec{f}$ to $\Psi(\vec{f})$ obeying the following rules: for some k

$(1)_k$ *(Bounded effect.)* $\vec{f}$ and $\Psi(\vec{f})$ differ only on at most k points.

$(2)_k$ *(Bounded cause.)* These $\leq k$ points and the values of $\Psi(\vec{f})$ on them depend only on the values of $\vec{f}$ on at most k points.

$(3)_k$ *(Bounded query.)* The computation of $\Psi(\vec{f})$ queries f on at most k points.

Applying our results on functionals satisfying condition (*) (cf. §1.1), we prove the following results (cf. Theorems 4.2, 4.4):

1. *A computable functional is the transition functional of an algorithm if and only if it satisfies properties $(1)_k$, $(2)_k$ for some k.*

2. *If Ψ is computable and satisfies $(1)_k$ and $(2)_k$ then one can compute the value of $\Psi(\vec{g})$ by querying $\vec{g}$ on at most $O(k^2)$ points. In other words, $(3)_{O(k^2)}$ is true.*

3. *A computable functional is the transition functional of an algorithm if and only if it satisfies property $(1)_k$ and a topological condition in the vein of (5) of § 1.1.*

2 Functionals using bounded information

In §2.1, 2.2, we present some notions of covering of the Baire space by clopen sets to which are associated notions of modulus of continuity for total functionals $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}$. Our main theorems relate these notions, cf. §2.3, 2.4, 2.5. A topological interpretation is given in §3 and applications to a characterization of algorithms is the subject of §4.

2.1 Deterministic coverings of the Baire space

Notation 2.1. *Let $k \in \mathbb{N}$. 1. We denote by $\mathcal{F}^{\mathbb{N}\to\mathbb{N}}_{<\omega}$ (resp. $\mathcal{F}^{\mathbb{N}\to\mathbb{N}}_{\leq k}$) the family of partial functions $u : \mathbb{N} \to \mathbb{N}$ with finite domains (resp. with domains having at most k elements).*
2. If $u \in \mathcal{F}^{\mathbb{N}\to\mathbb{N}}_{<\omega}$, we let $[u]$ be the set of total functions $\mathbb{N} \to \mathbb{N}$ which extend u.
3. We denote by $\mathcal{P}_{<\omega}(\mathbb{N})$ (resp. $\mathcal{P}_{\leq k}(\mathbb{N})$) the family of finite (resp. cardinality $\leq k$) subsets of $\mathbb{N}$.

First, we introduce some very simple notions.

Definition 2.2. *Let π be a a subfamily of $\mathcal{F}^{\mathbb{N}\to\mathbb{N}}_{<\omega}$ and $k \in \mathbb{N}$.*
1. We say that π is a covering *if $\mathbb{N}^{\mathbb{N}} = \bigcup_{u\in\pi}[u]$. A covering π is* unambiguous *if the $[u]$'s, $u \in \pi$, are pairwise disjoint, hence form a partition of $\mathbb{N}^{\mathbb{N}}$.*
2. A covering π is k-bounded *if $\texttt{Dom}(u)$ has at most k elements for every $u \in \pi$. It is* bounded *if it is k-bounded for some k.*

Remark 2.3. There is only one 0-covering: it is the singleton family consisting of the empty domain function. For any fixed $a \in \mathbb{N}$, the family $\mathbb{N}^{\{a\}}$ of all functions $\{a\} \to \mathbb{N}$ is an unambiguous 1-covering. Every unambiguous 1-covering is of that form or is the 0-covering. For $k \geq 2$ there are non trivial k-coverings, cf. Example 1.1

As we already noticed, even with a computable enumeration of the u_i's, property (*) in §1.1 gives no obvious way to compute Φ with a bounded number of queries. This is why we strengthen the notion of unambiguous bounded covering to that of deterministic one. This last notion is, in fact, the core of our study. It is based on the following simple result.

Proposition-Definition 2.4. *Let $\mathcal{X} : \mathbb{N}^{\mathbb{N}} \to \mathcal{P}_{<\omega}(\mathbb{N})$ be a total functional satisfying the following condition.*

- (†) *There exists an algorithm – using a function $\Omega : \mathbb{N} \to \mathbb{N}$ as an oracle – which, on input $f \in \mathbb{N}^{\mathbb{N}}$, computes $\mathcal{X}(f)$ and, during its computation, queries f exactly on the points of $\mathcal{X}(f)$.*

Then the family $\pi = \{f \restriction \mathcal{X}(f) \mid f \in \mathbb{N}^{\mathbb{N}}\}$ is an unambiguous covering which is computable in oracle Ω. Such coverings are called deterministic.

Proof. It is obvious that π is a covering. Let us show it is unambiguous. Suppose f is in $[g \restriction \mathcal{X}(g)]$. The run of the algorithm on f queries f successively on $a_1, \ldots, a_m$ such that $\mathcal{X}(f) = \{a_1, \ldots, a_m\}$. Similarly, the run on g queries g successively on $b_1, \ldots, b_p$ such that $\mathcal{X}(g) = \{b_1, \ldots, b_p\}$.

The first query a_1 to f is independent of any value of f hence it is the same as the first query b_1 to g. Thus, $a_1 = b_1$. Since $f \in [g \upharpoonright \mathcal{X}(g)]$, we have $f(a_1) = g(a_1)$. The second query a_2 to f depends only on the value of f on a_1 hence it is the same as the second query b_2 to g. Thus, $a_2 = b_2$. And so on... Thus, the computations on f and g are exactly the same, hence $\mathcal{X}(f) = \mathcal{X}(g)$ and $f \upharpoonright \mathcal{X}(f) = g \upharpoonright \mathcal{X}(g)$. This proves that $u = f \upharpoonright \mathcal{X}(f)$ is the unique $u \in \pi$ such that $f \in [u]$.

Finally, for every $u \in \mathcal{F}^{\mathbb{N}\to\mathbb{N}}_{<\omega}$, let $0 \oplus u$ be the total function $\mathbb{N} \to \mathbb{N}$ which extends u and takes value 0 on $\mathbb{N} \setminus \mathtt{Dom}(u)$. Observe that a partial function u is in π if and only if the algorithm applied to $0 \oplus u$ computes exactly $\mathtt{Dom}(u)$. This shows that π is computable (in oracle Ω). □

Remark 2.5. Though this notion is independent of ASM theory, nevertheless, it is interesting to illustrate it with ASMs having a unique dynamic symbol (cf. §4.2). Consider the terms occurring in the ASM program. In order to compute the next state, we have to evaluate them in the current state. To do so, one considers the forest of subterms and proceeds via a bottom-up evaluation of more and more complex subterms. The functional $\mathcal{X}$ just gives the values of all these subterms. Of course, $\mathcal{X}$ queries its argument f exactly on the values it outputs. In general ASMs, the functional $\mathcal{X}$ would be more complex and would take as arguments the current interpretations of all dynamic symbols.

The notion of deterministic k-bounded covering can also be defined with no functional at all.

Proposition 2.6. *Let π be a covering. The following conditions are equivalent.*

1. *π is deterministic k-bounded,*
2. *There exists a total function $\alpha : \mathcal{F}^{\mathbb{N}\to\mathbb{N}}_{<k} \times \mathbb{N} \to \mathbb{N}$ such that $\pi = \{u \mid \mathtt{Dom}(u) = \{a_i(u) \mid i < k\}\}$ where $a_i(u)$'s is defined inductively: $a_i(u) = \alpha(u \upharpoonright \{a_j(u) \mid j < i\}, i)$.*

Proof. (2) ⇒ (1) is trivial (let Ω be α). As for (1) ⇒ (2), let $\alpha(u, i)$ be the i-th point on which the algorithm queries its argument when the answer to any query on a is given by $u(a)$ if $a \in \mathtt{Dom}(u)$ and is a otherwise. Due to equality $\mathtt{Dom}(u) = \{a_i(u) \mid i < k\}$, case $a \notin \mathtt{Dom}(u)$ is vacuous. □

Remark 2.7. Observe that, in (1) ⇒ (2), α is computable in an enumeration of π. In (2) ⇒ (1), an enumeration of π can be taken computable in α. In particular, if π is computably enumerable then α can be taken computable and conversely.

2.2 Main theorem: refinements of continuity

First, we reformulate continuity of functionals in terms of how much information is used to compute their values. The core of our problem is to replace "finite" by "at most k".

Definition 2.8. *Let $\Phi : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ be a total functional.*
1. A modulus of continuity for Φ is any covering π such that Φ is constant on $[u]$ for every $u \in \pi$.
2. Let $\mathcal{P}$ be any combination of the following properties: k-bounded, bounded, unambiguous, deterministic, computably enumerable, computable. We say that Φ uses finite (resp. $\mathcal{P}$) information if Φ has a (resp. $\mathcal{P}$) modulus of continuity.

Care! To say that Φ uses k-bounded information does not mean that the computation of $\Phi(f)$ queries only k values of f. It is merely an assumption about the existence of a particular modulus of continuity of Φ. A priori, it does not help the computation: though we know that the sole restriction of f to k critical points does matter, we have no clue to get these points. The natural way is to enumerate this k-bounded modulus of continuity for Φ, say $(u_i)_{i\in\mathbb{N}}$, and look for the first i such that f extends u_i. We know it does exists since π is a covering. Then to get $\Phi(f)$ we can replace f by the extension of u_i which is 0 outside the domain of u_i and compute Φ on this function. But this process queries f on all $\mathtt{Dom}(u_j)$'s for $j \leq i$. Which is a finite set but a priori arbitrary large. In contrast, a *deterministic k-bounded modulus of continuity allows to compute $\Phi(f)$ with at most k queries to f.*

Finite information is a mere reformulation of continuity.

Proposition 2.9. *Let $\Phi : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ be a total functional. The following conditions are equivalent.*

1. *Φ is continuous,*
2. *Φ uses finite information,*
3. *Φ uses unambiguous finite information,*
4. *Φ uses deterministic finite information.*

Moreover, these equivalences hold effectively: just add "computable" in all points .

Proof. Since (4) $\Rightarrow$ (3) $\Rightarrow$ (2) $\Leftrightarrow$ (1) are trival, it suffices to prove (2) $\Rightarrow$ (4). Let $(u_i)_{i\in\mathbb{N}}$ be an enumeration of a modulus of continuity for Φ. Let $\widetilde{\pi}$ be the family of partial functions $v : \mathbb{N} \to \mathbb{N}$ such that, for some i, v has domain $\bigcup_{j\leq i} \mathtt{Dom}(u_j)$ and $v \restriction \mathtt{Dom}(u_j) = u_j$ if and only if $j = i$. It is easy to see that $\widetilde{\pi}$ is a modulus of continuity for Φ. We claim that it is a deterministic covering. Indeed, consider $\mathcal{X} : \mathbb{N}^{\mathbb{N}} \to \mathcal{P}_{<\omega}(\mathbb{N})$ such that $\mathcal{X}(f)$ is computed as follows: query f on the elements in $\mathtt{Dom}(u_i)$ for $i = 0, 1, \ldots$ until the least i such that f extends u_i is found and then output $\bigcup_{j\leq i} \mathtt{Dom}(u_j)$. This computation queries f exactly on the set it outputs. To conclude, observe that $\widetilde{\pi}$ is the range of $\mathcal{X}$. □

Our main theorem shows that, modulo a quadratic blow-up, Proposition 2.9 extends with "at most k" in place of "finite".

Theorem 2.10. *Let $\Phi : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ be a total functional which uses k-bounded information. Then*
1. Φ uses unambiguous k^2-bounded information.
2. The $k \mapsto k^2$ blow-up in Point 1 is optimal.
3. Φ uses deterministic k^2-bounded information.
4. Point 3 has an effective version: if Φ is computable and has a modulus of continuity which is k-bounded and computably enumerable then it has one which is $(2k^2 - k)$-bounded, computable and deterministic.

The proof of this theorem is given in §2.3 for Point 2, in §2.4 for Point 3 (which subsumes Point 1) and in §2.5 for Point 4. Let us cite simple corollaries of the above theorem.

Corollary 2.11. *The following conditions are equivalent:*

1. *Φ uses bounded information,*
2. *Φ uses unambiguous bounded information,*
3. *Φ uses deterministic bounded information,*

A less trivial corollary is as follows.

Corollary 2.12. *Let us say that a covering π' refines a covering π if, for every $u \in \pi$, the clopen $[u]$ is a union $\bigcup_{v\in X}[v]$ for some subset X of π'.*
Every unambiguous k-bounded (resp. computably enumerable) covering can be refined to a deterministic k^2-bounded (resp. $(2k^2-k)$-bounded computable) covering.

Proof. Let $(u_i)_{i\in\mathbb{N}}$ be an enumeration of an unambiguous k-bounded covering π. Define $\Phi : \mathbb{N}^\mathbb{N} \to \mathbb{N}$ such that Φ takes value i on the clopen $[u_i]$. This definition makes sense because π is unambiguous. Obviously, π is a modulus of continuity for Φ. Since it is k-bounded, by Theorem 2.10 we can find a deterministic k^2-bounded modulus of continuity π' for Φ. Since π' is a modulus of continuity for Φ, the set $\Phi^{-1}(i)$ is a union $\bigcup_{v\in X}[v]$ for some subset X of π'. To conclude, observe that $\Phi^{-1}(i) = [u_i]$ by definition. □

A last result along these lines.

Proposition 2.13. *Let $\Phi : \mathbb{N}^\mathbb{N} \to \mathbb{N}$ be total computable and $\mathcal{M}_\Phi = \{u \in \mathcal{F}_{<\omega}^{\mathbb{N}\to\mathbb{N}} \mid \Phi$ is constant on $[u]\}$.*
1. $\mathcal{M}_\Phi$ is the largest modulus of continuity of Φ. It is Π^0_1 and, in general, not computable, though Φ has a computable modulus of continuity (cf. Proposition 2.9).
2. If Φ has a k-bounded modulus of continuity then the largest one is $\mathcal{M}_\Phi \cap \mathcal{F}_{\leq k}^{\mathbb{N}\to\mathbb{N}}$ It is Π^0_1 and, in general, not computable.

Proof. Let $0 \oplus v$ be the total function $\mathbb{N} \to \mathbb{N}$ which extends v and takes value 0 outside $\mathtt{Dom}(v)$. Observe that u is not in $\mathcal{M}_\Phi$ if and only if u admits two finite extensions v, w such that $\Phi(0\oplus v) \neq \Phi(0\oplus w)$ and the computation of Φ on $0 \oplus v$ (resp. $0 \oplus w$) queries only $\mathtt{Dom}(v)$ (resp. $\mathtt{Dom}(w)$).

For an example for which $\mathcal{M}_\Phi$ and $\mathcal{M}_\Phi \cap \mathcal{F}_{\leq 1}^{\mathbb{N}\to\mathbb{N}}$ are not computable, let $\varphi : \mathbb{N} \to \mathbb{N}$ be computable with non computable range and let $\Phi(f)$ be 0 if $\varphi(z) \neq f(0)$ for all $z \leq f(1)$ else 1. Clearly, Φ is computable and $\mathbb{N}^{\{0,1\}}$ is a computable 2-bounded modulus of continuity for Φ. Let u_y be the function with domain $\{0\}$ such that $u_y(0) = y$. Then $u_y \in \mathcal{M}_\Phi$ if and only if y is not in the range of φ. □

Remark 2.14. We do not know if there exists some computable Φ admitting a bounded modulus of continuity but no computable bounded modulus.

2.3 Proof of Theorem 2.10: The quadratic blow-up is optimal

Fix some $k \geq 2$. Let $\Phi : \mathbb{N}^\mathbb{N} \to \mathbb{N}$ be defined as follows:

$$\Phi(f) = \begin{cases} 0 & \textit{if } \exists i < k\ \forall j < k\ f(ik + j) = 0 \\ 1 & \textit{otherwise} \end{cases}$$

Fact 2.15. *There exists a k-bounded modulus of continuity for Φ and an unambiguous k^2-bounded one but no unambiguous $(k^2 - 1)$-bounded one.*

Proof. Since $\Phi(f)$ depends only on the values of $f(x)$ for $x < k^2$, it is trivial that $\mathbb{N}^{[0,\ldots,k^2[}$ constitutes an unambiguous k^2-bounded modulus of continuity for Φ. Let us get a k-bounded modulus. For $i = 0, \ldots, k-1$, let v_i be the constant function on domain $[ik, (i+1)k[$ with value 0. For $\vec{j} = (j_0, \ldots, j_{k-1}) \in [0, k[^k$ and $\vec{m} = (m_0, \ldots, m_{k-1}) \in (\mathbb{N} \setminus \{0\})^k$, let $w_{\vec{j},\vec{m}}$ be the partial function with domain $\{ik + j_i \mid i < k\}$ such that $w_{\vec{j},\vec{m}}(ik + j_i) = m_i$. It is easy to check that $\Phi^{-1}(0)$ is the union of the $[v_i]$'s whereas $\Phi^{-1}(1)$ is the union of the $[w_{\vec{j},\vec{m}}]$'s. Thus, the v_i's together with the $w_{\vec{j},\vec{m}}$'s constitute a k-bounded modulus of continuity for Φ.

By way of contradiction, suppose that π is an unambiguous (k^2-1)-bounded modulus of continuity for Φ. Since $\Phi(f)$ depends only on $(x \mapsto \min(f(x), 1)) \upharpoonright [0, k^2[$, the idea of the proof is to go from π to a "finitary" family of partial functions $[0, k^2[\to \{0, 1\}$ having domains with exactly $k^2 - 1$ points and then use a counting argument to get a contradiction.

For $u \in \pi$, let $[\![u]\!]$ be the set of total functions $[0, k^2[\to \{0, 1\}$ compatible with $x \mapsto \min(u(x), 1)$. Problem: $[u] \cap [v] = \emptyset$ does not imply $[\![u]\!] \cap [\![v]\!] = \emptyset$. This is why we consider

$$\pi' = \{u \in \pi \mid u(x) = 0 \textit{ for all } x \geq k^2 \textit{ in } \mathtt{Dom}(u) \\ \textit{and } u(x) \leq 1 \textit{ for all } x < k^2 \textit{ in } \mathtt{Dom}(u)\}\ .$$

Clearly, the $[\![u]\!]$'s, $u \in \pi'$, are pairwise disjoint (as are the $[u]$'s since π is unambiguous) and constitute a partition of $\{0, 1\}^{[0,k^2[}$. For $\varepsilon = 0, 1$, let $F_\varepsilon = \{f \upharpoonright [0, k^2[\mid f \in \Phi^{-1}(\varepsilon)\}$. Since $\Phi^{-1}(\varepsilon)$ is a union of some $[u]$'s, $u \in \pi'$, we see that F_ε is a union of some $[\![u]\!]$, $u \in \pi'$.

Since π is (k^2-1)-bounded, for each $u \in \pi'$, we can choose $S \subset [0, k^2[$ with $k^2 - 1$ points such that $\mathtt{Dom}(u) \cap [0, k^2[\subseteq S$. Replace u by all its $\{0, 1\}$-valued extensions to S. In this way, we get a family π'' such that F_0 and F_1 are disjoint unions of some $[\![v]\!]$'s with $v \in \pi''$. To conclude, let us do some counting. A total function $\alpha : [0, k^2[\to \{0, 1\}$ is in F_1 if and only if, for every $i = 0, \ldots, k-1$, $\alpha \upharpoonright [ik, (i+1)k[$ is not the constant function with value 0. Thus, the cardinality of F_1 is $(2^k - 1)^k$ and that of F_0 is $2^{k^2} - (2^k - 1)^k$. Observe that these numbers are odd. However, each $[\![v]\!]$, for $v \in \pi''$, contains exactly 2 elements because there is only one point in $[0, k^2[\setminus \mathtt{Dom}(v)$. Thus, as a disjoint union of some $[\![v]\!]$'s with $v \in \pi''$, both F_0, F_1 contain an even number of elements. Contradiction! □

2.4 Proof of Theorem 2.10: deterministic bounded modulus

First, two simple results, the second being the key for an inductive proof.

Proposition 2.16. *If $u, v \in \mathcal{F}^{\mathbb{N} \to \mathbb{N}}_{<\omega}$ (cf. Notation 2.1) agree on* $\mathtt{Dom}(u) \cap$

$\mathtt{Dom}(v)$ *(in particular, if v extends u or if they have disjoint domains) and* $\Phi : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ *is constant on* $[u]$ *and on* $[v]$ *then* Φ *takes the same value on* $[u]$ *and* $[v]$.

Proof. Consider $f \in \mathbb{N}^{\mathbb{N}}$ which extends both u and v. □

Proposition 2.17. *Let* $\Phi : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ *be a total functional. Suppose* π *is a* k*-bounded modulus of continuity for* Φ*. If* Φ *is not constant then there exists a subset* S *of* $\mathbb{N}$ *with at most* $2k-1$ *elements which meets the domain of every* $u \in \pi$.

Proof. Let $f, g \in \mathbb{N}^{\mathbb{N}}$ be such that $\Phi(f) \neq \Phi(g)$ and let $u, v \in \pi$ be such that $f \in [u]$ and $g \in [v]$. Set $S = \mathtt{Dom}(u) \cup \mathtt{Dom}(v)$. Applying Proposition 2.16, we see that $\mathtt{Dom}(u)$ and $\mathtt{Dom}(v)$ are not disjoint so that S has at most $2k-1$ elements. Also, if $w \in \pi$ then the value of Φ on $[w]$ is different from at least one of those on $[u]$ and $[v]$ hence $\mathtt{Dom}(w)$ meets S. □

A convenient notation.

Notation 2.18. *If* $\varphi, \psi : A \to B$ *are partial functions, we let* $\varphi \oplus \psi : A \to B$ *be the partial function such that*

- $\mathtt{Dom}(\varphi \oplus \psi) = \mathtt{Dom}(\varphi) \cup \mathtt{Dom}(\psi)$,
- $\varphi \oplus \psi$ *extends* ψ,
- $(\varphi \oplus \psi)(x)$ *is equal to* $\varphi(x)$ *if* $x \in \mathtt{Dom}(\varphi) \setminus \mathtt{Dom}(\psi)$.

(Intuition: the one who is right is the last one who spoke.)

We can now come to the wanted proof.

Proof of point 3 *in Theorem 2.10.*
We argue by induction on k. The case Φ is constant is trivial. In particular, this solves the initial case $k = 0$ of the induction.

Suppose $k \geq 1$ and Φ is not constant and there exists a k-bounded modulus of continuity π for Φ. Let S be as in Proposition 2.17 and let $s : \mathbb{N} \to \mathbb{N}$ have domain S. Let

$$\pi_S^{(s)} = \{u \restriction (\mathtt{Dom}(u) \setminus S) \mid u \in \pi \textit{ and } u \textit{ is compatible with } s\}.$$

Let us see that $\pi_S^{(s)}$ is a covering. If $f \in \mathbb{N}^{\mathbb{N}}$ then $f \oplus s$ is in $[u]$ for some $u \in \pi$ which is compatible with s. Hence $(f \oplus s) \oplus (f \restriction S) = f$ is in $[u \restriction (\mathtt{Dom}(u) \setminus S)]$.

Proposition 2.17 insures that all functions in $\pi_S^{(s)}$ have domains with $\leq k-1$ elements. Thus, $\pi_S^{(s)}$ is $(k-1)$-bounded.

Let $\Phi^{(s)} : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ be such that $\Phi^{(s)}(f) = \Phi(f \oplus s)$ for all $f \in \mathbb{N}^{\mathbb{N}}$. Observe that $\pi_S^{(s)}$ is a modulus of continuity for $\Phi^{(s)}$. Indeed, if u compatible with s and f, g are in $[u \restriction (\mathtt{Dom}(u) \setminus S)]$ then $f \oplus s$ and $g \oplus s$ both extend u. Since $u \in \pi$ we have $\Phi(f \oplus s) = \Phi(g \oplus s)$, i.e. $\Phi^{(s)}(f) = \Phi^{(s)}(g)$.

Applying the induction hypothesis, there exists a deterministic $(k-1)^2$-bounded modulus of continuity π_s for $\Phi^{(s)}$. Then the family $\pi' = \{u \oplus s \mid s \in \mathbb{N}^S \textit{ and } u \in \pi_s \textit{ and } u \textit{ is compatible with } s\}$ is a covering: if $f \restriction S = s$ and $f \in [u]$ with $u \in \pi_s$ then u is compatible with s and $f \in [u \oplus s]$. Since $(k-1)^2 + (2k-1) = k^2$, π' is k^2-bounded. Let us see that π' is a modulus of continuity for Φ. If $f, g \in [u \oplus s]$ with $u \in \pi_s$ and u is compatible with s then $f, g \in [u]$ and f, g extend s hence $\Phi^{(s)}(f) = \Phi^{(s)}(g)$, i.e. $\Phi(f \oplus s) = \Phi(g \oplus s)$. Since $f \oplus s = f$ and $g \oplus s = g$, we get $\Phi(f) = \Phi(g)$. Finally, we show that π' is deterministic. Since π_s is deterministic, it is given as in Proposition 2.6 by some $\alpha^{(s)} : \mathcal{F}_{<\omega}^{\mathbb{N} \to \mathbb{N}} \times \mathbb{N} \to \mathbb{N}$. Define α as follows: $\alpha(v, 0), \ldots, \alpha(v, 2k-2)$ enumerate S and $\alpha(v, 2k-1+i) = \alpha^{(v \restriction S)}(v, i)$. Then the covering given by α is exactly π'. Hence π' is deterministic. ∎

Remark 2.19. The above proof considers as known whether Φ is constant or not. Of course, this is an undecidable question. Since the inductive proof goes from Φ to all the $\Phi^{(s)}$'s, this undecidable question has to be answered infinitely many times. Thus, the above proof is hopelessly non effective.

2.5 Proof of Theorem 2.10: computable deterministic modulus

The next Definition introduces a bounded version of the loop mentioned in §1.1 to get the least u_i such that f is in the clopen $[u_i]$. Instead of querying f on the sequence of all $\mathtt{Dom}(u_i)$'s until getting i such that f extends u_i, we query f on a conveniently extracted short subsequence.

Definition 2.20. *Given $\ell \in \mathbb{N}$ and an enumeration $(u_i)_{i \in \mathbb{N}}$ of $\mathbb{N}^{\mathbb{N}}$ of a covering π, we define a functional $\mathcal{I}_{\pi,\ell} : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{<\omega}$ such that, for any $f \in \mathbb{N}^{\mathbb{N}}$, $\mathcal{I}_{\pi,\ell}(f)$ is the strictly increasing sequence of integers $(i_0, \ldots, i_m)$, with $m \leq \ell$, defined by the following clauses.*

1. *$i_0 = 0$.*
2. *Suppose i_p is defined. Then i_{p+1} is defined if and only if $p < \ell$ and f does not extend any u_i for $i \leq i_p$.*
3. *If defined, i_{p+1} is the least $j > i_p$ such that u_j is compatible with $f \restriction \bigcup_{q \leq p} \mathtt{Dom}(u_{i_q})$ and $\mathtt{Dom}(u_j) \not\subseteq \bigcup_{q \leq p} \mathtt{Dom}(u_{i_q})$.*

We also let $\mathcal{V}_{\pi,\ell} : \mathbb{N}^{\mathbb{N}} \to \mathcal{F}^{\mathbb{N}\to\mathbb{N}}_{<\omega}$ *be such that* $\mathcal{V}_{\pi,\ell}(f) = f \restriction \bigcup_{p\leq m} \mathtt{Dom}(u_{i_p})$ *where* $\mathcal{I}_{\pi,\ell}(f) = (i_0, \ldots, i_m)$*.*

Fact 2.21. *1. There exists an algorithm which uses the enumeration of* π *as an oracle and which computes functionals* $\mathcal{I}_{\pi,\ell}$ *and* $\mathcal{V}_{\pi,\ell}$ *in such a way that, on input* f*, it queries the sole values of* f *on the domain of* $\mathcal{V}_{\pi,\ell}$*.*
2. The range of the functional $\mathcal{V}_{\pi,\ell}$ *is a deterministic covering which is computable in oracle the enumeration of* π*. In particular, if* π *is computably enumerable then the range of the functional* $\mathcal{V}_{\pi,\ell}$ *is a computable deterministic covering.*

Proof. Point 1: consider the algorithm given by Definition 2.20. Points 2: apply Proposition-Definition 2.4. □

Fact 2.22. *Let* $\Phi : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ *be a total computable functional. If* Φ *admits a computably enumerable* k*-bounded modulus of continuity then it also admits a computable deterministic* $(2k^2 - k)$*-bounded modulus of continuity.*

Proof. Suppose π is a computably enumerable k-bounded modulus of continuity for Φ and $(u_i)_{i\in\mathbb{N}}$ is a computable enumeration of π. Let $\widetilde{\pi}^{(\ell)}$ be the range of $\mathcal{V}_{\pi,\ell}$. We first show that, for $\ell \geq 2k-1$, $\widetilde{\pi}^{(\ell)}$ is a modulus of continuity for Φ. Let $v = \mathcal{V}_{\pi,2k-1}(f)$ where $f \in \mathbb{N}^{\mathbb{N}}$. We have to show that Φ is constant on the clopen $[v]$. Let $\mathcal{I}_{\pi,\ell}(f) = (i_0, \ldots, i_m)$, so that v is the restriction of f to $\bigcup_{p\leq m} \mathtt{Dom}(u_{i_p})$.

In case v extends some u_j, it is clear that Φ is constant on $[v]$ (recall π is a modulus of continuity for Φ).

Suppose now that v does not extend any u_j. Then the construction of the i_p's halts because $m = \ell$. By way of contradiction, suppose Φ is not constant on $[v]$. Let $g, h \in [v]$ be such that $\Phi(g) \neq \Phi(h)$. Since π is a modulus of continuity for Φ, there exists $\xi, \eta \in \mathbb{N}$ such that $g \in [u_\xi]$ and $h \in [u_\eta]$. For each $p = 0, \ldots, m$, u_ξ and u_η cannot be both compatible with u_{i_p} else (by Proposition 2.16) Φ would take the same value on $[u_\xi]$ and $[u_{i_p}]$ and the same value on $[u_\eta]$ and $[u_{i_p}]$. Hence Φ would take the same value on $[u_\xi]$ and $[u_\eta]$, hence also on $g \in [u_\xi]$ and on $h \in [u_\eta]$), a contradiction.

Thus, there exists $a_p \in \mathtt{Dom}(u_{i_p}) \cap (\mathtt{Dom}(u_\xi) \cup \mathtt{Dom}(u_\eta))$ such that $u_{i_p}(a_p)$ is different from $u_\xi(a_p)$ or from $u_\eta(a_p)$. Now, condition (3) in Definition 2.20 insures that, for $q \geq 1$, $u_{i_q} \restriction \bigcup_{r<q} \mathtt{Dom}(u_{i_r})$ is compatible with f hence with v (since f extends $v = \mathcal{V}_{\pi,2k-1}(f)$) hence with g and h (since their restriction to $\bigcup_{r<q} \mathtt{Dom}(u_{i_r})$ is that of v) hence with u_ξ and u_η (since they are restrictions of g and h). Thus, a_q is in $\mathtt{Dom}(u_{i_p}) \setminus (\bigcup_{r<p} \mathtt{Dom}(u_{i_r}))$. In particular, $a_q \neq a_r$ for all $r < q$. This proves that the a_p's, $p = 0, \ldots, m$ are pairwise distinct.

Since the a_p's are in $\mathtt{Dom}(u_\xi) \cup \mathtt{Dom}(u_\eta)$ and we have already seen that $m = \ell$, we get

(†) $\mathtt{Dom}(u_\xi) \cup \mathtt{Dom}(u_\eta)$ has at least $\ell + 1$ elements.

Now, since $\Phi(g) \neq \Phi(h)$ and $g \in [u_\xi]$ and $h \in [u_\eta]$, u_ξ and u_η cannot be compatible. In particular, the intersection $\mathtt{Dom}(u_\xi) \cap \mathtt{Dom}(u_\eta)$ is not empty. Since π is k-bounded, $\mathtt{Dom}(u_\xi)$ and $\mathtt{Dom}(u_\eta)$ have at most k elements. Thus, $\mathtt{Dom}(u_\xi) \cup \mathtt{Dom}(u_\eta)$ has at most $2k-1$ elements. Using (†), we get $\ell+1 \leq 2k-1$, which contradicts the assumed inequality $\ell \geq 2k-1$.

Let us majorize the cardinality of the domain of $v \in \widetilde{\pi}^{(\ell)}$. Since $\mathtt{Dom}(v)$ is the union of the $\mathtt{Dom}(u_{i_p})$'s, $p = 0, \ldots, \ell$, and each $\mathtt{Dom}(u_{i_p})$ has at most k elements (again, π is k-bounded), $\mathtt{Dom}(v)$ contains at most $k(\ell+1)$ elements. Letting $\ell = 2k-1$, this gives a $2k^2$ upper bound.

Thus, $\widetilde{\pi}^{(2k-1)}$ is a $2k^2$-bounded modulus of continuity for Φ. Point 2 of Fact 2.21 insures that $\widetilde{\pi}^{(\ell)}$ is computable. By Point 1 of Fact 2.21, there is an algorithm which, on input $f \in \mathbb{N}^{\mathbb{N}}$, computes $v = \mathcal{V}_{\pi,2k-1}(f)$ querying f solely on $\mathtt{Dom}(v)$. Thus, $\widetilde{\pi}^{(2k-1)}$ is a computable deterministic $2k^2$-bounded modulus of continuity for Φ.

Let us improve the bound $2k^2$ to $2k^2-k$. To get the $2k^2$ bound, for every $p \leq 2k-1$, we majorized by k the cardinal of $\mathtt{Dom}(u_{i_p}) \setminus (\bigcup_{r<p} \mathtt{Dom}(u_{i_r}))$. Let t be maximum such that $\mathtt{Dom}(u_{i_t}) \setminus (\bigcup_{r<t} \mathtt{Dom}(u_{i_r}))$ has k elements. Then $\mathtt{Dom}(u_{i_t})$ is disjoint from all the $\mathtt{Dom}(u_{i_r})$'s for $r < t$, hence Φ takes the same value on all the $[u_{i_q}]$'s for $q \leq t$. This value is different from the value of Φ on $[u_\xi]$ or on $[u_\eta]$. Let it be on $[u_\xi]$. Then all a_q's, for $q \leq t$, can be taken in $\mathtt{Dom}(u_\xi)$. Since the a_q's are pairwise distinct and $\mathtt{Dom}(u_\xi)$ has at most k elements, we get $t+1 \leq k$. Now, for $p > t$, $\mathtt{Dom}(u_{i_p}) \setminus (\bigcup_{r<p} \mathtt{Dom}(u_{i_r}))$ has at most $k-1$ elements. Thus, the cardinal of $\mathtt{Dom}(v) = \bigcup_{q \leq 2k-1} \mathtt{Dom}(u_{i_q})$ is bounded by $(t+1)k + (2k-1-t)(k-1) = t + 2k^2 - 2k + 1 \leq 2k^2 - k$ since $t+1 \leq k$. □

3 Topology and bounded information

This section relies on a simple idea: k-bounded modulus of continuity looks like uniform continuity. Indeed, it is uniform continuity for an appropiate uniformity on the Baire space. Observe that this cannot be uniform continuity with respect to the usual metric on $\mathbb{N}^{\mathbb{N}}$: for instance, in Example 1.1, one has to know f on some arbitrarily large point (namely $\alpha(f(0), f(1))$) to get the value of $\Phi(f)$.

3.1 The deterministic bounded information uniformity

For the notion of uniformity on a space and the related classical results, we refer to classical textbooks (Bourbaki or Kelley's [4, 13]) and to the short review in the Appendix.

Proposition-Definition 3.1. *To any unambiguous bounded covering π of $\mathbb{N}^{\mathbb{N}}$, associate*

$$\begin{aligned} U_\pi &= \{(f,g) \in \mathbb{N}^{\mathbb{N}} \times \mathbb{N}^{\mathbb{N}} \mid f,g \textit{ extend a same} \\ &\qquad\qquad\qquad\qquad \textit{partial function } u \textit{ in } \pi\} \\ &= \bigcup_{u\in\pi}[u] \times [u] \qquad \textit{(cf. Notation 2.1)} \end{aligned}$$

1. The family $\mathcal{B}^{unamb}$ of all such U_π's is the basis of a transitive uniformity on $\mathbb{N}^{\mathbb{N}}$.
2. The family $\mathcal{B}^{det}$ of all such U_π's, when π varies over deterministic bounded coverings is a basis of the same uniformity.
This uniformity is called the deterministic bounded information uniformity.

Proof. Straightforward consequences of classical result (cf. Proposition .7 in the Appendix to the paper) and of Corollary 2.12 supra. □

Proposition 3.2. *1. The deterministic bounded information uniformity on $\mathbb{N}^{\mathbb{N}}$*

1. *generates the Baire topology on $\mathbb{N}^{\mathbb{N}}$,*
2. *is a transitive uniformity,*
3. *is a proper refinement of the uniformity of the usual Baire metric.*
4. *admits no countable basis hence is not metrizable.*

Proof. 1. By definition (cf. Proposition .3), a basis of the topology associated to a uniformity is obtained by taking sections $U|f = \{g \mid (f,g) \in U\}$ where U varies in a basis of the uniformity and f varies in $\mathbb{N}^{\mathbb{N}}$. In our case, the basis consists of the U_π's, π varying among unambiguous bounded coverings. Now, if $f \in [u]$ and $u \in \pi$ then $U_\pi|f = [u]$ which is a clopen in the basis of the Baire topology. To conclude, observe that every clopen $[u]$ is so obtained because any partial function $u : \mathbb{N} \to \mathbb{N}$ with finite domain belongs to some unambiguous bounded covering π, for instance the covering consisting of all functions with domain $\mathtt{Dom}(u)$.
2. We show $U_\pi \circ U_\pi = U_\pi$. Suppose (f,g) and (g,h) are both in $U_\pi = \bigcup_{u\in\pi}[u \times [u]$. Since π is unambiguous there is a unique $u \in \pi$ such that

$g \in [u]$. Thus, (f,g) and (g,h) are both in $[u]\times[u]$ hence so is (f,h) which is therefore in U_π.
3. Recall the usual Baire distance: $d(f,g)$ is 0 if $f=g$ and $2^{-\min\{n \mid f(n)\neq g(n)\}}$ otherwise. The basic entourages are the $E_n = \{(f,g) \mid d(f,g) < 2^{-n}\}$'s. Observe that $E_n = U_{\pi_n}$ where π_n is the unambiguous n-bounded covering $\mathbb{N}^{\{0,\dots,n-1\}}$ (all functions with domain $\{0,\dots,n-1\}$). Finally, the set $\pi = \{(0,0)\} \cup \{\{(0,x),(x,y)\} \mid x,y \in \mathbb{N}, x \neq 0\}$ is a deterministic 2-bounded covering which contains no E_n.
4. Let $(\pi_n)_{n\in\mathbb{N}}$ be a sequence of deterministic bounded coverings. Let π_n be k_n-bounded. By Proposition 2.6, there exist total functions $\alpha_n : \mathcal{F}^{\mathbb{N}\to\mathbb{N}}_{<k_n} \times \mathbb{N} \to \mathbb{N}$ such that $\pi_n = \{u \mid \mathtt{Dom}(u) = \{a_{n,i}(u) \mid i < k_n\}\}$ with $a_{n,i}(u) = \alpha_n(u \restriction \{a_{n,j}(u) \mid j < i\}, i)$. The set of u's in π_n with range included in $\{0,\dots,n\}$ is finite and non empty. Let $\theta(n)$ be the maximum element in some $\mathtt{Dom}(u)$ for such u's. Then $\pi = \{\{(0,n),(\theta(n)+1,y)\} \mid n,y \in \mathbb{N}\}$ is a deterministic 2-bounded covering such that U_π contains no U_{π_n}. □

The deterministic bounded information uniformity is the pertinent topological tool to characterize functionals using bounded information.

Recall that the discrete uniformity on $\mathbb{N}$ is that for which the diagonal $\{(x,x) \mid x \in \mathbb{N}\}$ is an entourage and it is associated to the distance $d(x,y) = 0$ if $x=y$ and 1 otherwise.

Theorem-Definition 3.3. *Endow $\mathbb{N}$ with the discrete uniformity and $\mathbb{N}^\mathbb{N}$ with the deterministic bounded information uniformity. For a total functional $\Phi : \mathbb{N}^\mathbb{N} \to \mathbb{N}$ the following conditions are equivalent:*

1. *Φ has a bounded modulus of continuity,*
2. *Φ is uniformly continuous.*

We say that Φ is effectively uniformly continuous in case it has a computably enumerable bounded modulus of continuity (hence also a computable one by Corollaries 2.11, 2.12).

Proof. Since we consider the discrete uniformity on the range set $\mathbb{N}$, Φ is uniformly continuous if and only if the inverse image of the diagonal $(\Phi,\Phi)^{-1}(\{(x,x) \mid x \in \mathbb{N}\})$ is an entourage of our uniformity on $\mathbb{N}^\mathbb{N}$, i.e. contains an entourage U_π where π is a deterministic bounded covering. This means that, for all $f,g \in B^A$, if $(f,g) \in U_\pi$ then $\Phi(f) = \Phi(g)$. Now, $(f,g) \in U_\pi$ exactly means $f,g \in [u]$ for some $u \in \pi$. In other words, π is a deterministic bounded modulus of continuity for Φ. □

3.2 Curryfication and linear uniform continuity

In view of applications to the dynamics of algorithms, we look at functionals between function spaces.

First, a preliminary notion.

Definition 3.4. *The unambiguous (resp. deterministic) degree of an entourage E is defined as the least ℓ such that E contains some U_π with π an unambiguous (resp. deterministic) ℓ-bounded covering.*

Due the quadratic blow-up when we go to deterministic coverings (cf. Corollary 2.12), we have to distinguish two cases in the above Definition. However, this will have no serious incidence, cf. Theorem-Definition 3.7.

Proposition 3.5. *If the unambiguous degrees of entourages E_1 and E_2 are k_1 and k_2 then that of their intersection is at most $k_1 + k_2$. Idem with deterministic degrees.*

Proof. If the covering π_ε, $\varepsilon = 1, 2$ is k_ε-bounded then

$$\pi_1 \oplus \pi_2 = \{u_1 \oplus u_2 \mid u_1 \in \pi_1,\ u_2 \in \pi_2,\ u_1, u_2 \textit{ compatible}\}$$

(cf. Notation 2.18) is a (k_1+k_2)-bounded covering such that $U_{\pi_1 \oplus \pi_2} \subseteq U_{\pi_1} \cap U_{\pi_2}$. Finally, it is easy to check that this $\oplus$ operation preserves unambiguity and determinim. □

As we shall use Curryfication, to distinguish sources from targets, we argue with function spaces B^A and D^C.

Definition 3.6 (Curryfication)**.** *1. Let $\Psi : B^A \to D^C$ be a total functional. We denote by $\partial\Psi$ the total functional $\partial\Psi : (B^A \times C) \to D$ such that $(\partial\Psi)(f, c) = \Psi(f)(c)$ for all $f \in B^A$ and $c \in C$.*
2. Let $\top$ be an element outside A, B, C, D. We identify C with $\{\top\} \to C$ and $B^A \times C$ with $B^A \times C^{\{\top\}}$, and $\partial\Psi$ with a total functional $\partial\Psi : B^A \times C^{\{\top\}} \to D$.

As is well-known, $\Psi : B^A \to D^C$ is continous (with respect to the Baire topology) if and only if so is its Curryfication $\partial\Psi : B^A \times C^{\{\top\}} \to D$.

Surprising as it may be, it turns out that this is no more true with uniform continuity. A strengthening of uniform continuity is necessary.

Theorem-Definition 3.7. *Endow B^A and D^C with the deterministic bounded information uniformities. Let $\Psi : B^A \to D^C$ be a total functional. The following conditions are equivalent.*

1. *$\partial\Psi : B^A \times C^{\{\top\}} \to D$ is uniformly continuous,*

2. *There exists some fixed k such that the inverse image $(\Psi,\Psi)^{-1}(E)$ of an entourage E of unambiguous degree ℓ is an entourage of unambiguous degree at most $k\ell$*

3. *Idem as (2) with deterministic degrees.*

Also, the above equivalences effectivize with effective uniform continuity for $\partial\Psi$ (cf. Theorem-Definition 3.3) and entourages associated to computably enumerable coverings.

When these conditions hold (resp. in the effective version), we say that $\Psi : B^A \to D^C$ is (resp. effectively) linearly uniformly continuous.

Proof. (1) $\Rightarrow$ (2). Suppose $\partial\Psi$ is uniformly continuous and let τ be an unambiguous k-bounded $\partial\Psi$-covering of $B^A \times C^{\{\top\}}$. Lifting k to $k+1$, we can suppose that every $u \in \tau$ is defined on $\top$. Let π be a be an unambiguous ℓ-covering of D^C. For each $v = \{(c_1,d_1),\ldots,(c_p,d_p)\} \in \pi$ with $p \leq \ell$, let π'_v be the family of all $u_1 \oplus \ldots \oplus u_p : A \to B$ such that $u_1,\ldots,u_p$ are pairwise compatible and $u_i \oplus (\top, c_i) \in \tau$ and the value of $\partial\Psi$ on $[u_i \oplus (\top, c_i)]$ is d_i for all $i = 1,\ldots,p$. Set $\pi' = \bigcup_{v\in\pi} \pi'_v$. Since each $v \in \pi$ has a size $\leq \ell$ domain and each $u_i \oplus (\top, c_i) \in \tau$ has a size k domain, we see that each function in π' has a size $\leq k\ell$ domain. Let $f \in B^A$. There exists a unique $v = ((c_1,d_1),\ldots,(c_\ell,d_p)) \in \pi$ such that $\Psi(f) \in [v]$. Observe that $d_i = \Psi(f)(c_i)$ for all i. For each $i = 1,\ldots,p$ there exists a unique u_i such that $u_i \oplus (\top, c_i) \in \tau$ and $f \oplus (\top, c_i) \in [u_i \oplus (\top, c_i)]$. Since $f \in [u_i]$ for all i, we see that $u_1 \oplus \ldots \oplus u_p$ is in π'_v hence in π'. Thus, $f \in [u_1 \oplus \ldots \oplus u_p]$. This proves that π' is an unambiguous $k\ell$-covering of B^A. Finally $\Psi(g) \in [v]$ for every $g \in [u_1 \oplus \ldots \oplus u_p]$. Thus, $(\Psi,\Psi)(U_{\pi'}) \subseteq U_\pi$ and Ψ is $(\ell \mapsto k\ell)$-uniformly continuous hence linearly uniformly continuous.

(2) $\Rightarrow$ (1). Suppose (2) holds with the constant k. For $c \in C$, let π_c be the unambiguous 1-bounded covering of all partial functions $C \to D$ with domain $\{c\}$. Condition (1) insures that there exists a unambiguous k-bounded covering π'_c of B^A such that, for all $u \in \pi'_c$, $f \mapsto \Psi(f)(c)$ is constant on $[u]$. Then $\partial\Psi$ is constant on $[u \oplus (\top, c)]$. The family τ of all $u \oplus (\top, c)$, where $c \in C$ and $u \in \pi'_c$, is then an unambiguous $(k+1)$-bounded $\partial\Psi$-covering. By Theorem-Definition 3.3, $\partial\Psi$ is uniformly continuous.

(1) $\Rightarrow$ (3) and (3) $\Rightarrow$ (1) are analogous.

Effectivization of this proof is routine. □

We shall need the above notions extended to finite products of function spaces in the source. Things are the same, only the notations become heavier.

Proposition 3.8. *Let I be a finite set and the A_i's, B_i's be countable discrete spaces. Let $T = \prod_{i \in I} {B_i}^{A_i}$ and $\Psi : T \to D^C$ be a total functional. Theorem-Definition 3.7 extends as expected: replace B^A by T and the condition on $\partial\Psi$ by the conjunction of the same conditions for the $\partial(proj_j \circ \Psi)$'s.*

4 Characterizing the dynamics of algorithms

We apply our results about functionals using bounded information to the theory of algorithms. More precisely, we characterize their dynamics: the way the environment evolves.

First, we recall the problem and the ASM solution.

4.1 Capturing the notion of algorithm

Since a long time there has been a huge amount of work around algorithms. However, up to the emergence of ASM ca 1984, there was no mathematical answer to the question "*what is an algorithm?*".

One can look at algorithms in two ways:

- *(Denotational)* What do they do? What is their input/output behaviour?
- *(Operational)* How do they work? What is their step by step behaviour? How does the environment evolve?

Church's Thesis asserts that the classical mathematical formalization of computability fully captures the denotational side of algorithms. However, *Church's Thesis does not say anything about the operational side.* Indeed, the numerous computation models which have been proved to denotationally coincide are far apart from one another as concerns operationality. For instance, palindrome recognition can be done in linear time with a two-tape Turing machine whereas it requires quadratic time with a one-tape Turing machine (cf. Hennie, 1966 [12], see also [1]). Thus, one-tape and two-tape Turing machines constitute computation models which are denotationally equivalent (both are Turing complete: they capture all computable functions) but not operationally equivalent.

This leads to a natural question: among the numerous Turing complete computation models is there one which captures all possible algorithms? i.e. is *operationally complete*?

Kolmogorov-Uspensky Machines, ca 1958 [14], can be viewed as the first trial to answer the question. As pointed by Yuri Gurevich [10], though they do not state it explicitely, there is a thesis underlying their paper, namely "every computation, performing only one restricted local action at a time, can be viewed as (not only being simulated by, but actually being) the computation of an appropriate KU machine (in the more general form)". Schönhage's Storage Modification Machines, ca 1970 [15], is an extension of KU machines which can be seen as the second trial towards a model capturing all algorithms. As is now known, cf. Gurevich, 1997 [6], both KU and SSM trials fail but are rather close to a solution…

As it is, the question is about an intuitive notion with an excessively wide range: algorithms may be non deterministic, more or less parallel, distributed,… Let us be less ambitious and consider algorithms with the following features:

1. They are deterministic transition systems which run in discrete time and manipulate items in countable data structures.
2. In a single transition they can perform only a uniformly bounded read/write action.
3. They tell how recursive calls (if any) are managed.

Condition (2) allows for vector assignments of items or cells in arrays of items. But it excludes any global assignment of a (variable length) array and any parallelism such as that in cellular automata. We added condition (3) to cut short the discussion about the nature of recursive calls.

The first convincing model to capture all such algorithms is that of Abstract State Machines [11] from which we extracted conditions (1) and (2) supra, cf. also [11, 5].

4.2 Abstract State Machines

A key point in the ASM solution is to realize that the notion of algorithm is not an absolute notion but, on the contrary, it is intrinsically oracular. No algorithm works from scratch: it uses primitive operations "given for free", somehow "atomic". Whatever elementary they may seem, it is just fair to explicit them. For instance, a Turing machine reads a cell, moves its head and changes state: how these operations are performed is ignored, we take them as given. However, if we want to write a program in some programming language to simulate a Turing machine, we see that all these operations are not for free, they need pieces of code! In other algorithms, primitive operations may be non trivial ones. For instance, Euclid's algorithm for the gcd takes as primitive the zero test and the remainder in Euclidean division on $\mathbb{N}$. For Strassen's algorithm to multiply matrices, scalar multiplication is considered as primitive!

A (deterministic) ASM is a construct as follows.

- A multisort logical functional structure (i.e. an algebra) $\mathfrak{M} = (M_1, \ldots, M_n; f_1, \ldots, f_p)$ on a so-called static functional vocabulary $(\mathtt{f_1}, \ldots, \mathtt{f_p})$. This static vocabulary is typed: types are of the form M_j or $M_{j_1} \times \ldots \times M_{j_q} \to M_r$. The interpretations f_i's respect the types of the $\mathtt{f_i}$'s.

- A so-called dynamic functional vocabulary $(\mathtt{g_1}, \ldots, \mathtt{g_q})$ which is similarly typed.

- A program consisting of assignments $\mathtt{g_i}(\mathtt{t_1}, \ldots, \mathtt{t}_\ell) := \mathtt{u}$ (where $\mathtt{g_i}$ is a dynamic symbol and the $\mathtt{t_j}$'s and $\mathtt{u}$ are ground terms built with both vocabularies), conditionals and finite sets of such instructions.

Thus, an ASM $\mathfrak{M}$ is a transition system in which a state is the logical structure obtained by expanding $\mathfrak{M}$ with an interpretation of the dynamic vocabulary. A run of $\mathfrak{M}$ is a sequence of states obtained from the first one (the "initial" state) by iterated applications of the ASM program. There is no loop in ASM programs: the run is the sole (meta) loop!

The base sets of the structure and the interpretations of the static symbols are fixed. They represent the background of the algorithm. In particular, the static functions are tools given for free. For instance, the tape of a Turing machine and the moves of the head become the sort $\mathbb{Z}$ (all integers) with the successor and predecessor functions. As for Euclid's algorithm, its static background is the algebra $(\mathbb{N}; 0, \mathrm{mod})$.

There is no constraint on static functions in ASMs: they may even be non computable.

The interpretations of the dynamic symbols vary from state to state. They represent the foreground of the algorithm: in programming, this is the dynamic environment.

Finally, what Church's Thesis asserts about the denotational level can be lifted to the operational level:

ASM Thesis [11]: Every computation obeying conditions (1), (2), (3) of §4.1 can be viewed as (not only being simulated by, but actually being) the computation of an appropriate ASM.

Exactly as with Church's Thesis, the ASM Thesis has been extensively positively checked with all possible computation models, cf. the ASM web page [16]. It also holds in a "second-order" form: all usual computation models correspond to classes of ASMs associated to a fixed background with no constraint on ASM programs, cf. [8, 9].

The ASM Thesis implies Church's thesis, cf. [5]. It is also a tool to prove algorithmic completeness of computation models: it suffices to match any ASM run. For instance, lambda calculus, suitably augmented with constants (to represent elements of the sorts and static functions) and reductions of lambda free ground terms, is so proved to be algorithmically complete [7].

4.3 Algorithms viewed as functionals

Following the ASM analysis but forgetting ASM programs, an algorithm appears as a static background plus a functional Ψ mapping the interpretations of the dynamic symbols in some state into the interpretations in the next state (obtained by application of the ASM program).

We shall now consider two restrictions to algorithms which we add to conditions (1), (2), (3) from §4.1.

(4) The interpretations of the dynamic symbols are total functions.

(5) The algorithm never gets stucked so that the associated functional Ψ is total.

In general, algorithms do not obey these conditions. They manipulate partial functions and sometimes they get stucked. Indeed, ASMs do not assume conditions (4), (5).

However, the method developed in this paper applies only to those *total* algorithms. Assuming conditions (4), (5), the functional Ψ associated to the algorithm – which we call the dynamics of the algorithm – is a total functional of the form

$$(\diamond) \qquad \Psi : T \to T \quad \textit{with} \quad T = \prod_{i=1}^{i=q} {B_i}^{A_i}$$

where q is the cardinal of the dynamic vocabulary, the A_i's are finite products of the M_j's, the B_i's are among the M_j's, the M_j's are sorts and $A_i \to B_i$ is the type of the i-th dynamic symbol. Moreover, as mentioned in the Introduction §1.2, following [11], the algorithms we consider manipulate their environment in quite a gentle way: for some fixed k, the transition step leads from $\vec{f}$ to $\Psi(\vec{f})$ obeying the following rules: for some k

$(1)_k$ *Bounded effect.* $\vec{f}$ and $\Psi(\vec{f})$ differ on at most k points.

$(2)_k$ *Bounded cause.* These $\leq k$ points and the values of $\Psi(\vec{f})$ on them depend on the values of $\vec{k}$ on at most k points.

$(3)_k$ *Bounded query.* The computation of $\Psi(\vec{f})$ has queried f on at most k points.

Remark 4.1. What is the bound k in an ASM? It is the number of occurrences of dynamic symbols in the terms of the ASM program. Cf. also Remark 2.5.

Applying our results on functionals with bounded modulus of continuity, we obtain the following results.

Theorem 4.2 (Characterization of ASM total functionals)**.** *Let T and $\Psi : T \to T$ be a total functional as in ($\diamond$). Endow T with the deterministic bounded information uniformity and $\mathcal{F}_{\leq k}^{T \to T}$ with the discrete uniformity. The following conditions are equivalent*

1. *Ψ satisfies rules $(1)_k$, $(2)_k$ for some k,*
2. *Ψ is the transition functional of an ASM (with possibly non computable static background) which, moreover, satisfies rule $(3)_{O(k^2)}$.*
3. *Ψ is of the form $\Psi = Id \oplus \psi$ (cf. Notation 2.18) where, $\psi : T \to \mathcal{F}_{\leq k}^{T \to T}$ is a total functional which is uniformly continuous (with a bound $O(k^2)$ for the modulus of continuity).*

Remark 4.3. The k and $O(k^2)$ parameters may seem to contradict the fact that Ψ is the step by step behaviour of an algorithmic procedure (with possible non computable background). However, recall that a bounded degree of parallelism is allowed as long as the associated work is bounded. These parameters are, in fact, relevant to the close analysis of the step by step of algorithms done in [2].

Theorem 4.4 (Characterization of the dynamics of algorithms)**.** *Keep the notations of Theorem 4.2. The following conditions are equivalent*

(1)' *Ψ is the dynamics of an algorithm (among those satisfying conditions (1), (2), (3) of §4.1),*

(2)' *Ψ is the transition functional of a computable ASM,*

(3)' *Ψ is of the form $\Psi = Id \oplus \psi$ (cf. Notation 2.18) where, $\psi : T \to \mathcal{F}_{\leq k}^{T \to T}$ is a total computable functional which is effectively uniformly continuous.*

Proof. Proof of Theorem 4.2. (1) $\Rightarrow$ (2). The form $\Psi = Id \oplus \psi$ is a restatement of $(1)_k$. Rule $(2)_k$ insures that $\partial\Psi$ uses k-bounded information, hence, by Point 3 of Theorem 2.10, we see that $(3)_{O(k^2)}$ holds, i.e. $\partial\Psi$ uses deterministic $O(k^2)$-bounded information. Adding $\mathcal{F}^{T\to T}_{<\omega}$ to the sorts (or coding this sort together with its basic operations inside the given sorts constituting T) and the function α from Proposition 2.6 associated to the deterministic bounded modulus of continuity of $\partial\Psi$, we get an ASM with Ψ as transition functional.
(2) $\Rightarrow$ (3) $\Rightarrow$ (1) is straightforward.
Proof of Theorem 4.4. $(1)' \Rightarrow (2)'$. This is the ASM Thesis. $(2)' \Rightarrow (3)'$. Straightforward since an ASM functional is ruled by an ASM program hence has a computable bounded modulus of continuity (cf. Remark 4.1). $(3)' \Rightarrow (1)'$. Straightforward. □

5 Conclusion and Perspectives

We have shown that the notion of functional using bounded information is rooted in that of algorithm and allows for a mathematical formalization using advanced topological tools.

However, we have treated the case of algorithms manipulating completely defined parameters (basically, total functions $\mathbb{N} \to \mathbb{N}$). And we also supposed the algorithm never gets stucked so that its transition functional is total. Of course, these assumptions do not exhaust the reality of algorithms. An extension of our work using an adequate Scott domain in place of the Baire space is a perspective future work.

Finally, parallel algorithms (the ASM theory being done in [3]) seem well suited to an analogous characterization: just drop rule $(1)_k$ and consider computable effectively linearly uniformly continuous functionals.

[Uniformity and topology in a nutshell] Cf. Bourbaki or Kelley's classical textbooks [4, 13].

Definition .1. *A uniformity $\mathcal{U}$ on a space S is a family of subsets of $S \times S$ (called entourages) such that*

1. *Every entourage $U \in \mathcal{U}$ contains the diagonal $\Delta = \{(x, x) \mid x \in S\}$.*
2. *$\mathcal{U}$ is a filter: every superset of an entourage is an entourage and the intersection of finitely many entourages is an entourage.*
3. *If U is in $\mathcal{U}$ then so is $U^{\mathtt{sym}} = \{(y, x) \mid (x, y \in U\}$.*

4. For every entourage U there exists an entourage V such that $V \circ V \subseteq U$ where $V \circ W = \{(x,z) \mid (\exists y\ ((x,y) \in V \text{ and } (y,z) \in W)\}$. In particular, since V contains the diagonal, $V = V \circ \Delta \subseteq V \circ V \subseteq U$.

$\mathcal{U}$ is transitive in case condition (4) can be strengthened to

(4') For every entourage U there exists an entourage V such that $V \circ V = U$. One can then suppose V is symmetric.

A basis $\mathcal{B}$ of $\mathcal{U}$ is a family of entourages such that every entourage contains an entourage in $\mathcal{B}$

Example .2. 1. Suppose $d : S \times S \to [0, +\infty]$ is a metric on S. For any $r > 0$ let $U_r = \{(x,y) \in S \times S \mid d(x,y) \leq r\}$. Then the family of supersets of the U_r's is a uniformity $\mathcal{U}_d$ on S (and the U_r's constitute a basis of $\mathcal{U}_d$).
2. If d is an ultrametric, i.e. $d(x,z) \leq \max(d(x,y), d(y,z))$ for all $x, y, z \in S$, then $U_r \circ U_r = U_r$ and therefore $\mathcal{U}_d$ is a transitive uniformity.
3. The finest uniformity is the discrete one: it which contains the diagonal (hence all its supersets).

To every uniformity is associated a topology.

Proposition .3. *Suppose $\mathcal{U}$ is a uniformity on S. For every $x \in S$ let $\mathcal{N}_{\mathcal{U}}(x)$ be the family of sets of the form $\{y \in T \mid (x,y) \in U\}$ for some entourage $U \in \mathcal{U}$. Let $\mathcal{T}_{\mathcal{U}}$ be the family of sets $X \subseteq T$ such that $X \in \mathcal{N}_{\mathcal{U}}(x)$ for every $x \in X$. Then $\mathcal{T}_{\mathcal{U}}$ is the family of open sets of a topology on X and $\mathcal{N}_{\mathcal{U}}(x)$ is the family of neighborhoods of x in the topology $\mathcal{T}_{\mathcal{U}}$.*

In the same way that topology is the right framework to deal with continuity, the notion of uniformity is the right one to deal with uniform continuity.

Definition .4. *Suppose $\mathcal{U}$ and $\mathcal{V}$ are uniformities on the respective spaces S and T. A map $F : S \to T$ is uniformly continuous with respect to $\mathcal{U}$ and $\mathcal{V}$ if $(F \times F)^{-1}(V) \in \mathcal{U}$ for every $V \in \mathcal{V}$, i.e. the inverse image by $F \times F$ of an entourage of $\mathcal{V}$ is an entourage of $\mathcal{U}$. In other words, for every entourage $V \in \mathcal{V}$ there exists an entourage $U \in \mathcal{U}$ such that if x, y are U-close (i.e. $(x,y) \in U$) then $F(x), F(y)$ are V-close.*

This definition extends the usual one with metric spaces.

Proposition .5. *If (S, d_S) and (T, d_T) are metric spaces then a map $F : S \to T$ is uniformly continuous with respect to the uniformities $\mathcal{U}_{d_S}$ and $\mathcal{U}_{d_T}$ (cf. Example .2 supra) if and only if F is uniformly continuous in the sense of metric spaces: for every $\varepsilon > 0$ there exists $\eta > 0$ such that, for all $x, y \in S$, if $d_S(x,y) \leq \eta$ then $d_T(F(x), F(y)) \leq \varepsilon$.*

As expected, uniform continuity implies continuity.

Proposition .6. *If $F : S \to T$ is uniformly continuous with respect to $\mathcal{U}$ and $\mathcal{V}$ then F is continuous with respect to the associated topologies $\mathcal{T}_{\mathcal{U}}$ and $\mathcal{T}_{\mathcal{V}}$.*

Transitive uniformities can be characterized via partitions.

Proposition .7. *To any partition $\sigma = (S_\alpha)_{\alpha \in A}$ of a space S we associate the set $V_\sigma = \bigcup_{\alpha \in A} S_\alpha \times S_\alpha$. Let $\mathcal{U}$ be a uniformity on S and Σ be the family of all partitions σ such that $U_\sigma \in \mathcal{U}$. The uniformity $\mathcal{U}$ is transitive if and only if $\{V_\sigma \mid \sigma \in \Sigma\}$ is a basis of $\mathcal{U}$.*

Proof. Suppose $V \subseteq S \times S$ contains the diagonal and is symmetric. Observe that V satisfies $V \circ V = V$ if and only V is an equivalence relation on S if and only if $V = U_\sigma$ for some partition σ (namely the partition constituted by equivalence classes of V). □

Finally, a property of transitive uniformities.

Proposition .8. *Suppose $\mathcal{U}$ is a transitive uniformity on S. A basis of the topology associated to $\mathcal{U}$ is the family of all pieces S_α of all partitions σ of S such that U_σ is in $\mathcal{U}$.*

Proof. Easy consequence of Proposition .3. □

References

[1] T. Biedl, J.F. Buss, E.D. Demaine, M.L. Demaine, M. Hajiaghayi and T. Vinař. Palindrome recognition using a multidimensional tape. *Theoretical Computer Science*, 302(1-3):475–480, 2003.

[2] Andreas Blass, Nachum Dershowitz and Yuri Gurevich: Exact Exploration and Hanging Algorithms. Computer Science Logic 2010, Lecture Notes in Computer Science 6247, ed. Anuj Dawar, Helmut Veith, 140-154, 2010.

[3] Andreas Blass and Yuri Gurevich. Abstract state machines capture parallel algorithms. *ACM Trans. on Computational Logic*, 4(4):578–651, 2003. Correction and extension. *ACM Trans. on Comp. Logic*, 9(3), 32 p., 2008.

[4] Nicolas Bourbaki. *General topology, Chapters 1–4*. Springer, 1998.

[5] Dershowitz and Gurevich. A natural axiomatization of computability and proof of Church's Thesis. *Bull. of Symbolic Logic*, 14(3):299–350, 2008.

[6] Dexter, Boyle and Yuri Gurevich. Gurevich Abstract State Machines and Schönage Storage Modification Machines. *JUCS*, 3(4): 279–303, 1997.

[7] Marie Ferbus and Serge Grigorieff. ASM and operational algorithmic completeness of Lambda Calculus. *LNCS 6300*, 301–327, 2010.

[8] Serge Grigorieff and Pierre Valarcher. Evolving MultiAlgebras unify all usual sequential computation models. *STACS 2010*, 2010.

[9] Serge Grigorieff and Pierre Valarcher. Classes of Algorithms: Formalization and Comparison. *Bulletin EATCS*, June 2012, in press.

[10] Yuri Gurevich. On Kolmogorov machines and related issues. *Bulletin EATCS*, 35: 71–82, June 1988. Reprinted in *Current Trends in Theoretical Computer Science*, 225–234, World Scientific, 1993.

[11] Yuri Gurevich. Sequential Abstract State Machines capture Sequential Algorithms. *ACM Trans. on Computational Logic*, 1(1):77–111, 2000.

[12] Fred Hennie and Richard Stearns. Two-Tape Simulation of Multitape Turing Machines. *Journal of ACM*, 13(4):533–546, October 1966.

[13] John L. Kelley. *General topology.* Springer-Verlag.

[14] Kolmogorov, A.N.: On the definition of algorithm. Uspekhi Mat. Nauk 13(4), pp. 3–28, 1958. Translations AMS 29, pp. 217–245 (1963)

[15] Schönhage Arnold: *Storage modification machines. SIAM J. on Comp.*, 9:490–508, 1980.

[16] `www.eecs.umich.edu/gasm/subjects/models.html`. This is the ASM web page on models of computation.