Lógica e Computação

Editor Convidado: Fernando Ferreira

Towards recursion schemata for the probabilistic class PP

Reinhard Kahle, *Isabel Oitavem*
DM e CMA Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
e-mail: kahle@mat.uc.pt
oitavem@fct.unl.pt

Resumo: Usando um esquema de recursão com ponteiros, apresentamos a primeira abordagem recursiva à classe de complexidade PP.

Abstract We propose a recursion-theoretic characterization of the probabilistic class PP, using recursion schemata with pointers.

palavras-chave: Complexidade implícita; classe PP; esquemas de recursão com ponteiros.

keywords: Implicit complexity; the class PP; recursion schemes with pointers.

1 Introduction

Our goal is to explore the potential of *pointers* in recursion-theoretic contexts as a tool to characterize probabilistic classes of computational complexity. In this work we study PP, the class of decision problems solvable by probabilistic Turing machines in polynomial time with an error probability of less than $\frac{1}{2}$ for all instances.

It is well-known that PP contains NP and that it is contained in Pspace; it is open whether these inclusions are proper or not.

In previous work of the second author, the use of recursion schemes with pointers lead to characterizations of NP and $\mathsf{FPspace}$ (the class of *functions* corresponding to Pspace), [Oit11, Oit08]. On this base, our objective consists in extending/restricting the recursion schemes for NP and $\mathsf{FPspace}$, respectively, in an appropriate way to capture exactly the power of the class PP. As a result of the *work in progress*, reported here, we get a purely recursion-theoretic characterization of the probabilistic class PP.

The characterization follows closely the one for NP, given in [Oit11]. It comes in two stages, $\mathbf{ST}_{\mathsf{P}}$ and $\mathbf{ST}_{\mathsf{PP}}$, where $\mathbf{ST}_{\mathsf{P}}$ characterizes the *functions* computable in polynomial time by deterministic Turing machines [BC92]. $\mathbf{ST}_{\mathsf{PP}}$ results then from "strengthening" $\mathbf{ST}_{\mathsf{P}}$ with a scheme designed to characterize the decision problems of PP.

2 Notation

Let us consider the word algebra $\mathbb{W}$, i.e. the algebra generated by one nullary and two unary constructors, respectively, ϵ, $\mathbf{S}_0$ and $\mathbf{S}_1$. $\mathbb{W}$ can be interpreted over the set of all 0-1 words. As usually in word algebra contexts, one considers a destructor (or predecessor) symbol of arity one, $\mathbf{P}$. One also introduces a symbol $\mathbf{C}$, of arity 4, for the conditional function of the algebra. They are defined as follows: $\mathbf{P}(\epsilon) = \epsilon$, $\mathbf{P}(\mathbf{S}_i x) = x$ and $\mathbf{C}(\epsilon, x, y_0, y_1) = x$, $\mathbf{C}(\mathbf{S}_i z, x, y_0, y_1) = y_i$, $i \in \{0, 1\}$. $\vec{x}$ abbreviates $x_1, \cdots, x_n$ for some natural number n.

We introduce some, polynomial time functions to be used later on the paper. The function *read* is supposed to read the last bit of its input. For technical reasons at the reading stage the bit 1 is coded by 10, so $read(w)$ returns 10 if the last bit of w is 1 and it returns 0 otherwise. $+$ denotes the binary addition. We extend $+$ to all 0-1 words by considering that ϵ and words starting with 0 correspond to $0 \in \mathbb{N}$. We use infix notation for $+$. Moreover, $+_{read}(w_0, w_1)$ abbreviates $read(w_0) + read(w_1)$ and we use infix notation for $+_{read}$ too. Finally, let $2^{|\epsilon|} = 1$ and $2^{|\mathbf{S}_i z|} = \mathbf{S}_0(2^{|z|})$, we define $\#_{1/2}(z, w) = \begin{cases} 1 & \text{if } w > 2^{|z|} \\ 0 & \text{otherwise} \end{cases}$.

3 The term systems $\mathbf{ST}_{\mathcal{P}}$ and $\mathbf{ST}_{\mathcal{PP}}$

For the definition of the classes $\mathbf{ST}_{\mathcal{P}}$ and $\mathbf{ST}_{\mathcal{PP}}$, we adopt the framework introduced by Bellantoni-Cook in [BC92]. Thus, functions terms have two sorts of input positions, *normal* and *safe*. As usual, we write normal and safe input positions by this order, separated by semicolon: $f(\vec{x}; \vec{y})$.

Definition 1 *Let $\mathcal{I}$ be the class of function terms composed of the constructors of $\mathbb{W}$: ε, $\mathbf{S}_0$, $\mathbf{S}_1$; the destructor $\mathbf{P}$, the conditional $\mathbf{C}$, and the projection functions (over both input sorts).*

1. *$\mathbf{ST}_{\mathcal{P}}$ is the closure of $\mathcal{I}$ under $\mathrm{SC}_{\mathcal{P}}$ and $\mathrm{SR}_{\mathbb{W}}$;*
2. *$\mathbf{ST}_{\mathcal{PP}}$ is the closure of $\mathbf{ST}_{\mathcal{P}}$ under $\mathrm{SC}_{\mathcal{P}}$ and $\mathrm{STR}_{\mathbb{W}}[\dot{+}]$, where*

$\mathrm{SC}_{\mathcal{P}}$ — *input-sorted composition:*

$$f(\vec{x}; \vec{y}) = h(\vec{r}(\vec{x};); \vec{s}(\vec{x}; \vec{y})) \ , \quad \vec{r}, \vec{s} \in \mathbf{ST}_{\mathcal{P}} \ ,$$

$\mathrm{SR}_{\mathbb{W}}$ — *input-sorted recursion on notation:*

$$f(\epsilon, \vec{x}; \vec{y}) = g(\epsilon, \vec{x}; \vec{y})$$
$$f(\mathbf{S}_i(z;), \vec{x}; \vec{y}) = h(\mathbf{S}_i(z;), \vec{x}; \vec{y}, f(z, \vec{x}; \vec{y})), \quad i \in \{0, 1\}.$$

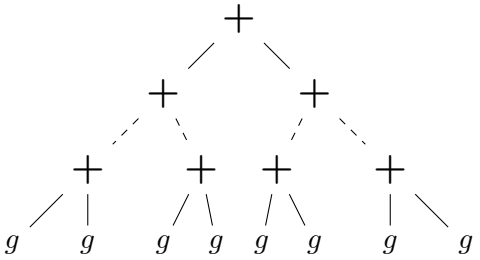
$\mathrm{STR}_{\mathbb{W}}[\dashv\!\vdash]$ — *input-sorted tree recursion with step function* $\dashv\!\vdash$*:*

$$f(p, \epsilon, \vec{x};) = g(p, \epsilon, \vec{x};)$$
$$f(p, \boldsymbol{S}_i(z;), \vec{x};) = \dashv\!\vdash(p, \boldsymbol{S}_i(z;), f(\boldsymbol{S}_0(p;), z, \vec{x};), f(\boldsymbol{S}_1(p;), z, \vec{x};);), \quad i \in \{0, 1\}.$$

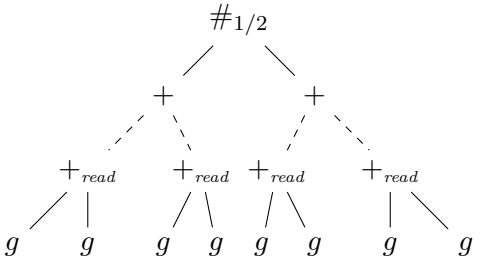
with g *and* $\dashv\!\vdash$ *in* $\mathbf{ST}_P$ *such that* $\dashv\!\vdash$ *satisfies:*

$$\begin{aligned}
\dashv\!\vdash(\epsilon, i, w_0, w_1;) &= \#_{1/2}(i, w_0 +_{read} w_1;), & \\
\dashv\!\vdash(p, i, w_0, w_1;) &= w_0 +_{read} w_1, & p \neq \epsilon, \\
\dashv\!\vdash(\epsilon, zi, w_0, w_1;) &= \#_{1/2}(zi, w_0 + w_1;), & z \neq \epsilon, \\
\dashv\!\vdash(p, zi, w_0, w_1;) &= w_0 + w_1, & p, z \neq \epsilon,
\end{aligned}$$

The motivation for this tree recursion scheme can be summarized as follows: the underline structure of $\mathrm{STR}_{\mathbb{W}}[\dashv\!\vdash]$ corresponds to the tree



where the pointer p and the recursion input z are omitted. By using them, one is able to obtain different outputs for $\dashv\!\vdash$ depending on its level in the tree above. Therefore, accordingly to the definition of $\dashv\!\vdash$ one may obtain the following labelling:



Once more, inputs are omitted. One reads from the leaves, and by performing the binary addition at all internal nodes one brings up to the root of the tree the information about how many times 1 occurs at the leaves.

Notice that $\#_{1/2}$ after performing the binary addition operation, returns 1 if the sum meets the threshold (i.e. if strictly more than half of the leaves are labelled by 1) and 0 otherwise.

One should notice that:

Remark 2 *1)* $\mathbf{ST}_{\mathsf{P}}$ *characterizes* FPtime*, the class of functions computable in deterministic polynomial time — see [BC92]; 2) Under this framework, recursive calls are usually placed into safe input positions of the step function. However, that becomes irrelevant when the recursion scheme imposes a fixed step function — like in* $\mathrm{STR}_{\mathbb{W}}[+]$*; 3) The tree recursion scheme used in [Oit08] to characterize* FPspace*, the class of functions computable in polynomial space, is (*$i \in \{0,1\}$*)*

$$f(p,\epsilon,\vec{x};\vec{y}) = g(p,\epsilon,\vec{x};\vec{y})$$
$$f(p,\boldsymbol{S}_i(z;),\vec{x};\vec{y}) = h(p,\boldsymbol{S}_i(z;),\vec{x};\vec{y},f(\boldsymbol{S}_0(p;),z,\vec{x};\vec{y}),f(\boldsymbol{S}_1(p;),z,\vec{x};\vec{y}))$$

$\mathrm{STR}_{\mathbb{W}}[+]$ *can be seen as a restriction of it. Thus,* $\mathbf{ST}_{PP} \subseteq$ FPspace*.*

4 $\mathbf{ST}_{PP}$ characterizes *PP*

The proofs of the upper and lower bound should now follow from an adaptation of the corresponding proofs for the recursion-theoretic characterization of *NP* in [Oit11]. Namely, by $\mathbf{ST}_{PP}$ characterizes *PP* we mean that *PP* coincides with the boolean part of $\mathbf{ST}_{PP}$. However, there are specific technical issues of this characterization which are not shared with the mentioned characterization of *NP*, and which become visible in the proofs (not given in this summary).

Referências

[BC92] S. Bellantoni and S. Cook. A new recursion-theoretic characterization of the poly-time functions. *Computational Complexity*, 2:97–110, 1992.

[Oit08] Isabel Oitavem. Characterizing Pspace with pointers. *Mathematical Logic Quarterly*, 54(3):317–323, 2008.

[Oit11] Isabel Oitavem. A recursion-theoretic approach to NP. *Annals of Pure and Applied Logic*, 162:661–666, 2011.

Research supported by the projects PTDC/FIL-FCI/109991/2009, PTDC/MHC-FIL/5363/2012 and PEst OE/MAT/UI0209/2011 (CMAF-UL), from FCT/MEC.

Calculabilidade do comportamento assimptótico de sistemas dinâmicos

Daniel S. Graça
CEDMES/FCT, Universidade do Algarve, C. Gambelas
8005-139 Faro, Portugal
& SQIG/Instituto de Telecomunicações, Lisboa, Portugal
e-mail: dgraca@ualg.pt

A noção de *algoritmo* é uma noção bastante antiga em matemática. Por exemplo, o algoritmo de Euclides já é mencionado nos *Elementos* por volta do ano 300 ano a.C. Um problema muito comum em matemática é querer ter um algoritmo que nos permita calcular uma certa quantidade. Por exemplo, na escola primária todos nós aprendemos algoritmos (regras) para adicionar ou multiplicar dois números.

Portanto, dado um problema matemático que sabemos ter solução, é natural querer saber se é possível calcular (ou computar) essa solução através de um algoritmo. Para mostrar que um problema é resolúvel por meio de um algoritmo, basta apresentar um algoritmo que garantidamente o resolva. Mas e como mostramos que um problema não é resolúvel desta forma? O principal problema parece ter a ver com a falta de uma definição rigorosa da noção intuitiva de algoritmo. Sem sabermos exatamente o que é um algoritmo, não é possível demonstrar que nenhum algoritmo pode resolver um determinado problema.

Este problema da definição rigorosa do que é um algoritmo só foi resolvida na década de 1930 com os trabalhos de Alonzo Church e Alan Turing. Utilizando modelos, como por exemplo as máquinas de Turing, passou a ser possível dizer quando é que, por exemplo, uma função inteira é computável. Essa teoria, baseada em número inteiros (ou equivalentemente em determinados tipos de estruturas discretas) tem tido imenso sucesso, formando as bases da teoria da computação, e potenciou desenvolvimentos que permitiram dar inicio à revolução informática a que se assistiu nos anos seguintes.

Apesar do sucesso da teoria da computação em lidar com estruturas discretas, muito menos tem sido estudado sobre problemas que envolvem estruturas contínuas, como por exemplo os números reais. No entanto, muitos problemas na prática envolvem números reais, pelo que parece importante considerar esse caso. Aliás, Turing já aborda esse caso no seu artigo seminal [Tur36].

Hoje em dia é possível trabalhar com a computabilidade de números e funções reais utilizando a teoria da *análise computável*. A ideia é *codificar*

esses números e funções através de números inteiros ou funções sobre os números inteiros, que já sabemos computar (ou mostrar que não são computáveis) através da teoria clássica da computabilidade.

Seguidamente, e de forma muito resumida, apresentamos algumas definições básicas (para mais detalhes, aconselhamos o artigo [BHW08], onde são apresentadas muitas outras definições alternativas, mas equivalentes às utilizadas em baixo).

Uma sequência de números racionais $\{q_n\}_{n\in\mathbb{N}}$ pode ser codificada por meio de uma função $\phi : \mathbb{N} \to \mathbb{N}^3$, com componentes $\phi_1, \phi_2, \phi_3 : \mathbb{N} \to \mathbb{R}$, desde que se tenha $q_n = (\phi_1(n) - \phi_2(n))/2^{\phi_3(n)}$. Da mesma forma se pode codificar uma dupla sequência de racionais $\{q_{i,j}\}_{i,j\in\mathbb{N}}$ através de uma função $\phi : \mathbb{N}^2 \to \mathbb{N}^3$.

Definição 1 *Um* nome *para um número real $x \in \mathbb{R}$ é uma função $\phi : \mathbb{N} \to \mathbb{N}^3$, que codifica uma sequência de racionais $\{q_n\}_{n\in\mathbb{N}}$ tal que $|q_n - x| \leq 2^{-n}$ para todo o $n \in \mathbb{N}$. Um* nome *para uma sequência de números reais $\{x_n\}_{n\in\mathbb{N}}$ é uma função $\phi : \mathbb{N}^2 \to \mathbb{N}^3$ que codifica uma dupla sequência de racionais $\{q_{n,i}\}_{n,i\in\mathbb{N}}$ tal que $|x_i - q_{n,i} \leq 2^{-i}|$ para todo o $i, n \in \mathbb{N}$.*

Definição 2 *Um número real (sequência de números reais) é* computável *se admite um nome computável.*

De forma intuitiva, um número é computável se pode ser aproximado com uma precisão arbitrária através de uma máquina de Turing (algoritmo). Os números racionais, $\pi, e, \sqrt{2}$ são exemplos de números computáveis.

Definição 3 *Uma função $f : \mathbb{R} \to \mathbb{R}$ é* computável *se existe uma máquina de Turing tal que, dado um nome ρ de $x \in \mathbb{R}$ e algum $n \in \mathbb{N}$, então a máquina de Turing com input n e oráculo ρ calcula $\phi(n)$, onde $\phi : \mathbb{N} \to \mathbb{N}^3$ é um nome de $f(x)$.*

Pode-se mostrar que todas as funções usuais da análise como e^x, cos, etc. são computáveis. Naturalmente pode-se estender esta definição para funções em $\mathbb{R}^n$.

Definição 4 *Um conjunto fechado $F \subseteq \mathbb{R}^n$ é* computável *se a função distância*

$$d_F(x) = min_{y\in F}\|x - y\|$$

é computável. Um conjunto aberto é computável *se o seu complementar é computável.*

Por exemplo $\varnothing, \mathbb{R}^2$ ou o disco $D^2 = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$ são exemplos de conjuntos computáveis. Na teoria clássica da computação, existe a noção de conjuntos recursivamente enumeráveis, que são conjuntos cujo elementos podem ser enumerados por uma função computável. Esta noção pode ser estendida para conjuntos de números reais como se segue.

Definição 5 *Um conjunto aberto $A \subseteq \mathbb{R}$ é* recursivamente enumerável *se existem sequências computáveis de números racionais $\{q_n\}_{n\in\mathbb{N}}$ e $\{r_n\}_{n\in\mathbb{N}}$ tais que $A = \cup_{n\in\mathbb{N}} B(q_n, r_n)$, onde $B(x, r) = \{y \in \mathbb{R} : |x - y| < r\}$. Um conjunto fechado F é* recursivamente enumerável *se existe uma sequência computável de números reais $\{x_n\}_{n\in\mathbb{N}}$ tal que $\cup_{n\in\mathbb{N}}\{x_n\}$ é denso em F.*

As definições anteriores podem ser estendidas de forma natural para subconjuntos em $\mathbb{R}^n$. Tal como nos resultados clássicos, pode-se também mostrar que um conjunto é computável sse esse conjunto e o seu complemento são recursivamente enumeráveis (r.e). Note-se que também poderíamos ter definido os conjuntos r.e. utilizando a função distância (ver por exemplo [BHW08]).

Nos nossos trabalhos, temos estado particularmente interessados em problemas que envolvem sistemas dinâmicos e o seu comportamento de longo prazo. Os sistemas dinâmicos aparecem em inúmeros contextos e são importantes em muitas aplicações. Devido à sua grande complexidade matemática, em muitas aplicações têm sido utilizados computadores para tentar obter informação sobre sistemas dinâmicos. O nosso objetivo foi o de compreender se os computadores podem ser utilizados com sucesso nessa tarefa. Apresentamos de seguida um resumo de alguns resultados que obtivemos (a referência [BGPZ13] apresenta uma descrição mais detalhada desses resultados, indicando as referências onde os resultados forma originalmente obtidos):

Teorema 6 *Dado como input uma função analítica f, o problema de decidir o número de pontos de equilíbrio de $y' = f(y)$ não é computável, mesmo em conjuntos compactos. O mesmo problema, mas considerando órbitas periódicas em vez de pontos de equilíbrio, também não é computável. No entanto, em sistemas estruturalmente estáveis, o conjunto formado por todos os pontos de equilíbrio é recursivamente enumerável, assim como o conjunto formado por todas as órbitas periódicas hiperbólicas.*

Teorema 7 *O atractor (geométrico) de Lorenz é computável, assim como a ferradura de Smale.*

Teorema 8 *A bacia de atração de um atractor é, em geral, não computável, mesmo que o atractor seja um ponto de equilíbrio hiperbólico e o sistema seja definido por uma função analítica computável. Essa não-computabilidade é persistente: pode-se perturbar o sistema e mesmo assim a bacia de atração pode continuar a ser não-computável. No entanto, a bacia de atração é recursivamente enumerável.*

Teorema 9 (Versão computável do Teorema de Hartman-Grobman) *Perto de um ponto de equilíbrio hiperbólico, é possível encontrar um homeomorfismo computável que transforma a solução do sistema não-linear na solução do seu sistema linearizado.*

Teorema 10 *A variedade estável/instável de um ponto de equilíbrio hiperbólico é computável localmente, mas não globalmente.*

Agradecimentos. O trabalho descrito neste artigo foi parcialmente suportado pelo EU FEDER POCTI/POCI e pela *Fundação para a Ciência e a Tecnologia* via SQIG - Instituto de Telecomunicações através do projeto FCT PEst-OE/EEI/LA0008/2013.

Referências

[BGPZ13] Olivier Bournez, Daniel S. Graça, Amaury Pouly, and N. Zhong. Computability and computational complexity of the evolution of nonlinear dynamical systems. In P. Bonizzoni, V. Brattka, and B. Löwe, editors, *Proc. 9th Conference on Computability in Europe, CiE 2013: The Nature of Computation—Logic, Algorithms, Applications*, LNCS 7921, pages 12–21. Springer, 2013.

[BHW08] V. Brattka, P. Hertling, and K. Weihrauch. A tutorial on computable analysis. In S. B. Cooper, , B. Löwe, and A. Sorbi, editors, *New Computational Paradigms: Changing Conceptions of What is Computable*, pages 425–491. Springer, 2008.

[Tur36] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, (Ser. 2–42):230–265, 1936.

O-minimality and sheaf cohomology

Mário J. Edmundo
Universidade Aberta
Campus do Tagus Park, Edifício Inovação I
Av. Dr. Jaques Delors
2740-122 Porto Salvo, Oeiras, Portugal
and
CMAF Universidade de Lisboa
Av. Prof. Gama Pinto 2
1649-003 Lisboa, Portugal
e-mail: edmundo@cii.fc.ul.pt

Luca Prelli
CMAF Universidade de Lisboa
Av. Prof. Gama Pinto 2
1649-003 Lisboa, Portugal
e-mail: lmprelli@fc.ul.pt

Resumo: O objectivo deste trabalho é contribuir para o desenvolvimento da teoria de feixes o-minimais definindo as seis operações neste contexto. Estas construções, para além do interesse que apresentam em si, fornecem as ferramentas essenciais para a obtenção de novas provas gerais e uniformes a alguns problemas (e.g. as conjecturas de Pillay) da geometria o-minimal sobre grupos definivelmente compactos definidos em estruturas o-minimais arbitrárias.

Abstract The aim of our work is to contribute to the development of the theory of o-minimal sheaf cohomology by defining the six Grothendieck operations in this setting. Beside their own interest, these constructions will provide the main missing ingredients to obtain general and unified proofs of some problems (e.g. Pillay's conjectures) of o-minimal geometry about definably compact definable groups in arbitrary o-minimal structures.

palavras-chave: Estruturas o-minimais; algebra homológica; teoria de feixes.

keywords: O-minimal structures; homological algebra; sheaf theory

1 Introduction

The recent impact of model theoretic techniques in algebra, algebraic geometry, number theory, has been remarkable. O-minimality is the analytic part

of model theory and deals with theories of ordered, hence topological, structures satisfying certain tameness properties. It generalizes semi-algebraic and subanalytic geometry and it is claimed to be the formalization of Grothendieck's notion of tame topology (topologie modérée).

The geometry of definable sets has had an impact in the theory of definable groups. For example: the triangulation theorem allows the development of an o-minimal singular (co)homology with Hurewicz theorem, Künneth formula, Poincaré duality and degree theory. These are essential ingredients in the proof of Pillay's conjecture in the field case (a non-standard analogue of Hilbert's 5^{th} problem for locally compact topological groups).

A natural question arises: Can we construct o-minimal cohomology for general o-minimal structures? The aim of this work is to give a positive answer to this question (under some suitable hypothesis). In order to make the required constructions we will need to develop o-minimal sheaf theory and define the Grothendieck operations in this setting.

2 Preliminaries

O-minimal structures. An ordered structure

$$\mathcal{M} = (M, <, (c)_{c\in\mathcal{C}}, (f)_{f\in\mathcal{F}}, (R)_{R\in\mathcal{R}})$$

is o-minimal if every definable subset of M in the structure is already definable in $(M, <)$, i.e. is a finite union of points and intervals. Examples of o-minimal structures are:

- $\mathcal{M} = (\mathbb{R}, <, 0, 1, +)$ PL-geometry
- $\mathcal{M} = (\mathbb{R}, <, 0, 1, +, \cdot)$ semi-algebraic geometry
- $\mathcal{M} = (\mathbb{R}, <, 0, 1, +, \cdot, (f)_{f\in\text{an}})$ subanalytic geometry

Other interesting examples can be obtained by adding the exponential exp or replacing $\mathbb{R}$ with $\mathbb{R}((t^{\mathbb{Q}}))$ (non-standard o-minimal structures).

O-minimal sheaves. Let X be a definable space (example: M^n with the order topology) and let $\widetilde{X}$ be the associated o-minimal spectrum. Let k be a field. Let $\text{Op}(\widetilde{X})$ be the family of open subsets of $\widetilde{X}$. A presheaf of k-vector

spaces is the data of:

$$\begin{array}{rcl} \mathrm{Op}(\widetilde{X}) & \to & \{k\text{-vector spaces}\} \\ U & \mapsto & \Gamma(U;F) \quad (= F(U)) \\ (V \subset U) & \mapsto & (F(U) \to F(V)) \quad \text{(restriction)} \\ & & s \mapsto s|_V \end{array}$$

The restriction is compatible with the inclusions (i.e. $U \subset V \subset W$, $s \in F(W)$, then $s|_V|_U = s|_U$ and $s|_W = s$). That is, a contravariant functor $F : \mathrm{Op}(\widetilde{X}) \to \mod(k)$. A presheaf is a sheaf if satisfies the following gluing conditions: let $U \in \mathrm{Op}(\widetilde{X})$ and let $\{U_j\}_{j\in J}$ be a covering of U in $\mathrm{Op}(\widetilde{X})$, then we have the exact sequence

$$0 \to F(U) \to \prod_{j\in J} F(U_j) \to \prod_{j,k\in J} F(U_j \cap U_k)$$

3 O-minimal sheaf cohomology

Operations. Let X be a definable space. We still denote by X its o-minimal spectrum to lighten notations. One can define, for sheaves and their derived category the operations $R\mathcal{H}om$ and RHom (hom-functors), $\otimes$ (tensor product) and, given a continuous definable map $f : X \to Y$, the functors Rf_* (direct image) and f^{-1} (inverse image).

One can define sheaf cohomology as $H^*a_{X*}F = H^*(X;F)$ (F sheaf, X definable manifold, a_X projection to a point). One can prove the following results ([3]):

- Vanishing theorems
- Vietoris-Beagle theorem
- Eilenberg-Steenrod axioms

Proper direct image. A definable set K is definably compact if every definable curve has a limit (in M^n iff definable closed and bounded). A definable map is definably proper if the inverse image of a definably compact is definably compact. Hence K is compact iff a_K is definably proper.

In order to study cohomology with definably compact support we need the functor of proper direct image $f_!$

$$f_!F(U) = \{s \in F(f^{-1}(U)),\ f \text{ definably proper on } supp(s)\}$$

When $f = a_X$ (in the derived category) we get cohomology with compact support. The following formulas hold ([4]):

- Projection formula
- Base change formula
- Künneth formula

Duality. In the derived category one can construct an extraordinary inverse image $f^!$. When $f = a_X$ we get the Poincaré-Verdier duality for o-minimal sheaves.

Cohomology computations. Once we define the six Grothendieck operations and the above fundamental formulas we are able to compute cohomology for general o-minimal structures as follows:

$$\begin{aligned} H^*(X,\mathbb{Q}) &= H^*(a_{X*}{a_X}^{-1}\mathbb{Q}) \text{ (cohomology)} \\ H_c^*(X,\mathbb{Q}) &= H^*(a_{X!}{a_X}^{-1}\mathbb{Q}) \text{ (cohomology with compact support)} \\ H_*(X,\mathbb{Q}) &= H^*(a_{X!}a_X^!\mathbb{Q}) \text{ (homology)} \\ H_*^{\mathrm{BM}}(X,\mathbb{Q}) &= H^*(a_{X*}a_X^!\mathbb{Q}) \text{ (Borel-Moore homology)} \end{aligned}$$

Remark. Some conditions ([4]) on the category of definable sets are needed. Examples of categories satisfying the conditions include: (i) regular, locally definably compact definable spaces in o-minimal expansions of real closed fields; (ii) Hausdorff locally definably compact definable spaces in o-minimal expansions of ordered groups with definably normal completions; (iii) locally closed definable subspaces of cartesian products of a given definably compact definable group in an arbitrary o-minimal structure.

Referências

[1] L. van den Dries *Tame topology and o-minimal structures* London Math. Soc. Lecture Note Series 248, Cambridge University Press, Cambridge 1998.

[2] M. Kashiwara and P. Schapira, *Sheaves on Manifolds*, Grundlehren der Math. 292, Springer-Verlag, Berlin 1990.

[3] M. Edmundo, G. Jones and N. Peatfield, "Sheaf cohomology in o-minimal structures", *J. Math. Logic*, Vol. 6 N.2 (2006), pp. 163-179.

[4] M. Edmundo and L. Prelli, "The six Grothendieck operations on o-minimal sheaves", *C. R. Acad. Sci. Paris, Ser. I*, Vol.352 (2014), pp. 455-458.