Descriptive Complexity of Probabilistic Complexity Classes through Second Order Generalized Quantifiers

Thiago Alves Rocha[1,2], **Ana Teresa Martins**[2]

[1]Departamento de Computação
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Maracanaú – CE – Brazil

[2]Departamento de Computação – Universidade Federal do Ceará (UFC)
Caixa Postal 6.021 – 60.455-970 – Fortaleza – CE – Brazil

thiago.alves@ifce.edu.br, ana@lia.ufc.br

Abstract. *Descriptive Complexity deals with the relationship between logical definability and computational complexity on finite structures. As an example in the case of probabilistic complexity classes, we have that BPP is equivalent to the class of problems definable by a randomised inflationary fixed-point logic with counting $BPIFP(C)$. In this paper, we show that we can define logics with generalized second order quantifiers equivalent to probabilistic complexity classes. This quantifiers are used to simulate the behavior of probabilistic Turing machines.*

Resumo. *Complexidade Descritiva lida com a relação entre definibilidade lógica e complexidade computational em estruturas finitas. Como exemplo no caso de classes de complexidade probabilísticas, temos que BPP é equivalente à classe de problemas definíveis por uma versão randômica da lógica de ponto-fixo inflacionário com contagem $BPIFP(C)$. Neste artigo, nós mostramos que podemos definir lógicas com quantificadores generalizados de segunda ordem equivalentes à classes de complexidade probabilísticas. Estes quantificadores são usados para simular o comportamento de máquinas de Turing probabilísticas.*

1. Introduction

Descriptive Complexity [Grädel et al. 2005] characterize the complexity of a problem based on a logical language needed to express it rather than physical measures such as time and space. We say that a logic $\mathcal{L}$ captures a complexity class $\mathcal{C}$ if $\mathcal{L}$ express all and only the problems of this class. In [Eickmeyer 2011], probabilistic logics $BPFO$ and $BPIFP(C)$ are defined and they capture probabilistic complexity classes $BPAC^0$ and BPP, respectively.

Our contribution is to define logics with generalized second order quantifiers and without randomisation and show that they are strong enough to capture probabilistic complexity class. The approach is to use second order quantifiers to simulate the behavior of randomised algorithms for problems in these classes. We demonstrate this by showing that our logics with generalized second order quantifiers are equivalent to randomised logics defined in the framework of [Eickmeyer 2011].

2. Probabilistic Complexity Classes

We assume basic knowledge of computational complexity classes as P and NP, Turing Machines and languages. In order to investigate probabilistic algorithms, we may formally introduce probabilistic Turing machines as in [Arora and Barak 2009].

Definition 1 (Probabilistic Turing Machine)**.** *A probabilistic Turing machine (PTM) is a Turing machine with two transition functions* δ_0*,* δ_1*. To execute a PTM* M *on an input* x*, we choose in each step with probability* $1/2$ *which transition function to use. For a function* $T : N \to N$*, we say that* M *runs in* $T(n)$*-time if for any input* x*,* M *halts on* x *within* $T(|x|)$ *steps regardless of the random choices it makes.*

Bellow, we have the class BPP that aims to capture efficient probabilistic computation.

Definition 2 (BPP)**.** *The class* BPP *(Bounded Error Probability) contains all languages* L *for which there is a probabilistic Turing machine* M *polynomially bounded in time with the following property: for all inputs* w*, if* $w \in L$*, then* $\Pr[M(x) = 1] \geq 2/3$ *and if* $w \notin L$*, then* $\Pr[M(x) = 0] \geq 2/3$*.*

Note that the PTM in the previous definition satisfies a very strong property: For every input, it either accepts it with probability at least $2/3$ or rejects it with probability at least $2/3$. Since a deterministic TM is a special case of a PTM (where both transition functions are equal), we have that $P \subseteq BPP$ while it is not known whether $BPP \subseteq NP$.

3. Randomised Logics

For a τ-structure $\mathcal{A}$, we denote the class of all $(\tau \cup \rho)$-expansions of $\mathcal{A}$ by $X(\mathcal{A}, \rho)$. We can view $X(\mathcal{A}, \rho)$ as a probability space with the uniform distribution. A structure $\mathcal{B} \in X(\mathcal{A}, \rho)$ can be seen as a $(\tau \cup \rho)$-structure such that for all k-ary $R \in \rho$ and all tuples $(a_1, ..., a_k)$ in the domain of $\mathcal{A}$ we can decide whether $(a_1, ..., a_k) \in R$ with probability $\frac{1}{2}$.

Definition 3. *Let* $\mathcal{L}$ *be a logic and* $0 \leq \alpha \leq \beta \leq 1$*. A formula* $\phi \in \mathcal{L}[\tau \cup \rho]$ *has a* $(\alpha, \beta]$*-gap if for all* τ*-structures* $\mathcal{A}$

$$\Pr_{\mathcal{B} \in X(\mathcal{A},\rho)}(\mathcal{B} \models \phi) \leq \alpha \text{ or } \Pr_{\mathcal{B} \in X(\mathcal{A},\rho)}(\mathcal{B} \models \phi) > \beta.$$

Definition 4. *Let* $\mathcal{L}$ *be a logic and* $0 \leq \alpha \leq \beta \leq 1$*. The logic* $P_{(\alpha,\beta]}\mathcal{L}$ *is defined as follows: for each vocabulary* τ*,*

$$P_{(\alpha,\beta]}\mathcal{L}[\tau] = \bigcup_\rho \{\phi \in \mathcal{L}[\tau \cup \rho] \mid \phi \text{ has a } (\alpha, \beta]\text{-gap}\},$$

where the union ranges over all vocabularies ρ *disjoint from* τ*. Let* $\phi \in P_{(\alpha,\beta]}\mathcal{L}[\tau]$*, the semantics is defined bellow:*

$$\mathcal{A} \models \phi \text{ if and only if } \Pr_{\mathcal{B} \in X(\mathcal{A},\rho)}(\mathcal{B} \models \phi) > \beta.$$

To finish, let $BP\mathcal{L} = P_{(1/3,2/3]}\mathcal{L}$ and define the randomised logics $BPFO$ and $BPIFP(C)$ using Definition 4 such that FO stands for First-Order Logic and $IFP(C)$ stands for Inflationary Fixed-Point with Counting [Eickmeyer 2011].

4. Descriptive Complexity

The Descriptive Complexity [Grädel et al. 2005] deals with the relationship between logical definability and computational complexity on finite structures. While the computational complexity is interested with the cost of computational resources, as time and space, to decide if structures in a class have a certain property, the Descriptive Complexity is concerned with the logical expression of this property.

We are concerned with the results which states that a logic $\mathcal{L}$ captures a complexity class $\mathcal{C}$ on class of structures $\mathcal{D}$ (notation $\mathcal{L} = \mathcal{C}$). It means that the $\mathcal{L}$-definable properties of structures in $\mathcal{D}$ are precisely those that are decidable in $\mathcal{C}$. The definition of $\mathcal{L} = \mathcal{C}$ is introduced by the following two definitions [Grädel et al. 2005].

Definition 5 ($\mathcal{L} \subseteq \mathcal{C}$)**.** *Let $\mathcal{L}$ be a logic, $\mathcal{D}$ a domain of finite structures, and $\mathcal{C}$ a computational complexity class. The logic $\mathcal{L}$ is in $\mathcal{C}$ on $\mathcal{D}$ if, for every fixed vocabulary τ and fixed sentence $\phi \in \mathcal{L}(\tau)$, the complexity of evaluating ϕ on $\mathcal{D}(\tau)$ is a problem in $\mathcal{C}$.*

Definition 6 ($\mathcal{C} \subseteq \mathcal{L}$)**.** *Let $\mathcal{L}$ be a logic, $\mathcal{D}$ a domain of finite structures, and $\mathcal{C}$ a computational complexity class. $\mathcal{C}$ is in $\mathcal{L}$ on $\mathcal{D}$ if, for every model class $\mathcal{K} \subseteq \mathcal{D}(\tau)$ such that the membership problem is in $\mathcal{C}$, there is a sentence $\phi \in \mathcal{L}(\tau)$ such that $\mathcal{K} = \{\mathcal{A} \in \mathcal{D}(\tau) | \mathcal{A} \models \phi\}$.*

Below, we state and briefly explain some results of Descriptive Complexity that will be used in this work. The next result is the seminal theorem of the Descriptive Complexity area.

Theorem 1 (Fagin's Theorem)**.** $SO\exists = NP$.

Using this result, we can define any NP problem in a formula of $SO\exists$. For instance, the formula

$$\phi_{3color} = \exists R \exists B \exists G \forall x((R(x) \vee G(x) \vee B(x)) \wedge \forall y(E(x,y) \rightarrow (\neg(R(x) \wedge R(y)) \wedge \neg(G(x) \wedge G(y)) \wedge (B(x) \wedge B(y))))).$$

define the class of graphs that are 3-colorable, i.e., a graph G satisfies ϕ_{3color} if and only if G is 3-colorable. Three colorability of graphs is an NP-complete problem. For the case of probabilistic complexity classes, [Eickmeyer 2011] has the following results:

Theorem 2 ([Eickmeyer 2011])**.** $BPFO = BPAC^0$.

where AC^0 is a complexity class of problems that are recognized by a family of Boolean circuits $(C_n)_{n \geq 1}$ such that each circuit C_n has n inputs, one output and the total number of gates is polynomially bounded by n. Besides that, there is a $d > 1$ such that all circuits C_n have depth at most d and we can test whether a circuit $C \in (C_n)_{n \geq 1}$ in $dlogtime$. $BPAC^0$ is defined similarly as BPP.

Theorem 3 ([Eickmeyer 2011])**.** $BPIFP(C) = BPP$.

5. Logics with Generalized Quantifiers

Generalized quantifiers are generalizations of standard quantifiers $\exists$ and $\forall$. A general way of defining generalized quantifiers that quantify over second order variables is introduced in [Andersson 2002]. For example, we can define a new quantifier $\mathcal{Q}^k_{>\frac{1}{2}}$:

$$\mathcal{Q}^k_{>\frac{1}{2}} = \{\langle A, P\rangle \, | P \subseteq \mathcal{P}(A^k) \text{ and } |P| > 2^{|A|^k - 1}\}.$$

Now we can define a logic $\mathcal{L}(\mathcal{Q}^k_{>\frac{1}{2}})$ adding formulas of the form $\mathcal{Q}^k_{>\frac{1}{2}} X^k \varphi(X)$ for $\varphi(X) \in \mathcal{L}$. For instance, we can define $FO(\mathcal{Q}^k_{>\frac{1}{2}})$ adding the following case in the semantics of FO:

$$\mathcal{A} \models \mathcal{Q}^k_{>\frac{1}{2}} X^k \varphi(X) \text{ if and only if } |\{R \in \mathcal{P}(A^k) | \mathcal{A} \models \varphi(R)\}| > 2^{|A|^k - 1}.$$

Now we show that we can define logics with generalized second order quantifiers equivalent to randomized logics $P_{\alpha,\beta}\mathcal{L}$. Let $\mathcal{L}$ be a logic. First, we define $G\mathcal{L}((\mathcal{Q}^k_{\leq\alpha})_{k\geq 1}, (\mathcal{Q}^k_{>\beta})_{k\geq 1})$ as

$$G\mathcal{L}((\mathcal{Q}^k_{\leq\alpha})_{k\geq 1}, (\mathcal{Q}^k_{>\beta})_{k\geq 1}) = \{\mathcal{Q}^k_{>\beta}X\varphi(X) \mid k \geq 1, \varphi(X) \in \mathcal{L} \text{ and for all structures } \mathcal{A}, \text{ we have } \mathcal{A} \models \mathcal{Q}^k_{>\beta}X\varphi(X) \text{ or } \mathcal{A} \models \mathcal{Q}^k_{\leq\alpha}X\varphi(X)\}.$$

Now we can show the equivalence:

Theorem 4. *Let* $0 \leq \alpha \leq \beta \leq 1$. $G\mathcal{L}((\mathcal{Q}^k_{\leq\alpha})_{k\geq 1}, (\mathcal{Q}^k_{>\beta})_{k\geq 1})$ *is equivalent to* $P_{(\alpha,\beta]}\mathcal{L}$.

Proof. Let τ be a vocabulary, $\mathcal{Q}^k_{>\beta}X\varphi(X) \in G\mathcal{L}((\mathcal{Q}^k_{\leq\alpha})_{k\geq 1}, (\mathcal{Q}^k_{>\beta})_{k\geq 1})$ and $\mathcal{A}$ a τ-structure. We have that $\varphi(X) \in \mathcal{L}[\tau \cup \{X\}]$, thus

$$P_{\mathcal{B}\in\mathcal{X}(\mathcal{A},\rho)}(\mathcal{B} \models \varphi(X)) = \frac{|\{\mathcal{B}\in\mathcal{X}(\mathcal{A},\rho)|\mathcal{B}\models\varphi(X)\}|}{\{\mathcal{B}|\mathcal{B}\in\mathcal{X}(\mathcal{A},\rho)\}} = \frac{|\{X\in\mathcal{P}(A^k)|(\mathcal{A},X)\models\varphi(X)\}|}{2^{|A|^k}}.$$

$\mathcal{A} \models \mathcal{Q}^k_{>\beta}X\varphi(X)$ iff $|\{R \in \mathcal{P}(A^k) \mid (\mathcal{A}, R) \models \varphi(R)\}| > \beta \times 2^{|A|^k}$ iff $\frac{|\{R\in\mathcal{P}(A^k)|(\mathcal{A},R)\models\varphi(R)\}|}{2^{|A|^k}} > \beta$ iff $P_{\mathcal{B}\in\mathcal{X}(\mathcal{A},\rho)}(\mathcal{B} \models \varphi(X)) > \beta$.

$\mathcal{A} \models \mathcal{Q}^k_{\leq\alpha}X\varphi(X)$ iff $|\{R \in \mathcal{P}(A^k) \mid (\mathcal{A}, R) \models \varphi(R)\}| \leq \alpha \times 2^{|A|^k}$ iff $\frac{|\{R\in\mathcal{P}(A^k)|(\mathcal{A},R)\models\varphi(R)\}|}{2^{|A|^k}} \leq \alpha$ iff $P_{\mathcal{B}\in\mathcal{X}(\mathcal{A},\rho)}(\mathcal{B} \models \varphi(X)) \leq \alpha$.

□

6. Capturing Results and Future Work

From Theorem 4, we can obtain the following results:

Theorem 5. $GFO((\mathcal{Q}^k_{\leq 1/3})_{k\geq 1}, (\mathcal{Q}^k_{>2/3})_{k\geq 1}) = BPAC^0$.

Theorem 6. $GIFP(C)((\mathcal{Q}^k_{\leq 1/3})_{k\geq 1}, (\mathcal{Q}^k_{>2/3})_{k\geq 1}) = BPP$.

For future work, we want to define logics with second order generalized quantifiers to capture probabilistic exponential time classes.

Acknowledgement

We are grateful to Francicleber Martins Ferreira for his valuable contributions to this paper.

References

Andersson, A. (2002). On Second-order Generalized Quantifiers and Finite Structures. *Ann. Pure Appl. Logic*, 115(1-3):1–32.

Arora, S. and Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition.

Eickmeyer, K. (2011). *Randomness in Complexity Theory and Logics.* PhD thesis, Humboldt University of Berlin. http://d-nb.info/1015169163.

Grädel, E., Kolaitis, P. G., Libkin, L., Marx, M., Spencer, J., Vardi, M. Y., Venema, Y., and Weinstein, S. (2005). *Finite Model Theory and Its Applications (Texts in Theoretical Computer Science. An EATCS Series).* Springer-Verlag New York, Inc., Secaucus, NJ, USA.