About the relationship between formal logic and complexity classes

Working paper

Comments welcome; my email: armandobcm@yahoo.com

Armando B. Matos

October 20, 2013

1 Introduction

We analyze a particular result of descriptive complexity, namely Fagin's Theorem – the work that started the field!. It relates a syntactic class of logic sentences, the existential second order logic, with a complexity class, the class NP.

This relation is surprising. How is it possible to represent, in terms of a class of "pure" logic, concepts like machines, computational times and polynomials?

Without further assumptions, I think it isn't possible. Yet, if reasonable assumptions are used for encoding the logic relations, and given the robustness of the "polynomially related" concept (definition in page 1), we can almost forget about the computational questions.

Those "reasonable assumptions" are related to the size of the representation of a logic relation r and to the access time to its properties. The representation should neither be "too large" nor "too small" (when this is possible!), and the access to its elements (does $r(x, y)$ hold?) should be relatively efficient.

Being a personal working paper we felt free to include when appropriate parts or adaptations of [3, 2, 5].

2 Basic concepts

Some basic concepts are assumed, namely first order logic (FOL), FOL with equality, language (or signature) $\mathcal{L}$, relation symbols, constants and function symbols, arity, model, satisfiable sentence, valid sentence, $\Sigma \models \sigma$.

Definition 1 *Two functions* $f(n) : \mathbb{N} \to \mathbb{N}$ *and* $g(n) : \mathbb{N} \to \mathbb{N}$ *are* polynomially related *if there are positive polynomials* p *and* p' *such that* $|f(n)| \leq p(|g(n)|)$ *and* $|g(n)| \leq p'(|f(n)|)$.

3 Differences between the finite and infinite model theories

3.1 Gödel's completeness theorem

Theorem 1 (Gödel's completeness theorem) *For a particular proof system of first order logic, we have* $\Sigma \models \sigma$ *iff* $\Sigma \vdash \sigma$.

This is remarkable: a universal search over all possible structures $\mathcal{A}$ (there are uncountable many!), checking for each one whether every structure $\mathcal{A}$ that satisfies Σ also satisfies σ, is equivalent to the existence of a finite object (a proof of σ from Σ).

3.2 Recursiveness of sets of valid formulas

Corollary 1 *The set of valid first-order sentences is r.e. but not recursive.*

Theorem 2 (Church's theorem) *Assume that the language* $\mathcal{L}$ *contains some relation symbol that is not unary. Then the set of valid first-order sentences over* $\mathcal{L}$ *is not recursive.*

Corollary 2 *The set of valid first-order sentences is r.e. but not recursive.*

Theorem 3 (Finite structures, Trakhtenbrot 1994) *Assume that the language* $\mathcal{L}$ *contains some relation symbol that is not unary. The set of valid first-order sentences over* $\mathcal{L}$ *is co-r.e. but not r.e.*

Note the difference between the general case (r.e. but not co-r.e.) and the finite case (co-r.e. but not r.e.).

Definition 2 *A sentence is* finitely controllable *if it is either unsatisfiable or finitely satisfiable.*

Theorem 4 *Let* $\mathcal{L}$ *be a recursive, finitely controllable set of FOL sentences. Then the decision problem "is* $\sigma \in \mathcal{L}$ *satisfiable?" is decidable.*

Proof. Run in parallel: (i) a search for a finite structure that satisfies σ; (ii) a search for a proof of unsatisfiability. One and only one of these processes finishes. □

3.3 Compactness theorem

Theorem 5 (Compactness theorem) *Let* Σ *be a set of first-order sentences. If every finite subset of* Σ *is satisfiable, then* Σ *is satisfiable.*

Proof. This proof uses twice Gödel's completeness theorem. If Σ is not satisfiable, then $\Sigma \models$ FALSE, where FALSE is some logically false sentence such as $p \wedge \neg p$. By the completeness theorem, $\Sigma \vdash$ FALSE. Thus, there is a proof of FALSE from Σ in Gödel's proof system. Since every proof has finite length in Gödel's proof system, only a finite subset $\Sigma' \subseteq \Sigma$ is used in the proof. So, $\Sigma' \vdash$ FALSE. By the completeness theorem again, $\Sigma' \models$ FALSE. So, Σ' is not satisfiable. □

Definition 3 *Two structures $\mathcal{A}$ and $\mathcal{B}$ over the same language are said to be elementarily equivalent if, for every first-order sentence σ in this language, $\mathcal{A} \models \sigma$ iff $\mathcal{B} \models \sigma$.*

The compactness theorem is not true for finite structures. An example: for each positive integer k, define σ_k to be a first-order sentence that says "there are at least k points". For example, we can express σ_3 as

$$\exists x_1 \exists x_2 \exists x_3 : (x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$$

Let $\Sigma = \{\sigma_1, \sigma_2, \ldots\}$. It is easy to see that Σ is not finitely satisfiable, although every finite subset of Σ is finitely satisfiable.

Theorem 6 *For every finite $\mathcal{L}$-structure A, there is a first-order sentence σ_A such that an arbitrary $\mathcal{L}$-structure $\mathcal{B}$ isomorphic to $\mathcal{A}$ iff $\mathcal{B} \models \sigma_A$.*

Comment. Thus, each finite structure is characterized up to isomorphism by a first-order sentence. Of course, no such theorem is true, in general, about infinite structures; consider, for example, non-standard models of arithmetic.

4 Spectra and generalized spectra (structures)

Definition 4 *The* spectrum *of a first order sentence is the set of cardinalities of the universes of its finite models.*

Example 1 *Consider the FO sentence $\forall x : (f(x) \neq x) \wedge (\forall x \forall y : (f(x) = y \Leftrightarrow f(y) = x))$. The spectrum is the set of even positive integers.*

Example 2 *Consider a FO sentence that is the conjunction of the field axioms plus conditions to establish that the ternary relations "$+$" and "$\times$" are functions $\mathbb{N}^2 \to \mathbb{N}$. Example of one element of the conjunction: $\forall x \, \forall y \, \forall x : (x \cdot (y \cdot z)) = ((x \cdot y) \cdot z)$. The spectrum is the set of powers of primes.*

Open problem 1 (Asser problem) *Is the class of spectra closed under complement?*

Notation. The expression $\sigma(P, Q, \ldots, R)$ denotes a first-order sentence over the language $\{P, Q, \ldots, R\}$. The arity of each relation is usually clear from the context. If not, we write for instance $R^{(a)}$ if R has arity a.
An example of a sentence of the form $\sigma(P, Q, R)$: $\forall x \exists y : (P(x) \wedge Q(y)) \Rightarrow R(x, y)$. □

Definition 5 *A* generalized spectra *or* Σ_1^1 *is a class over finite structures, where some, but not necessarily all, of the relation symbols in the language are existentially quantified. Thus, a* Σ_1^1 *sentence has the form* $\exists Q_1 \ldots Q_k : \sigma(P_1, \ldots, P_s, Q_1, \ldots Q_k)$*, where* $\sigma(P_1, \ldots, P_s, Q_1, \ldots Q_k)$ *is a first-order sentence and where the* Q_i*'s are relation symbols (these are referred to as the extra relation symbols).*

Comment. FOL sentences are not very expressive.

Comment. Σ_1^1 sentences are more expressive than FOL. They can, for instance, describe 3COL: there is a FOL sentence σ of the form

$$\exists\, Q_1\, Q_2\, Q_3 : \sigma(P, Q_1, Q_2, Q_3)$$

that states "the graph represented by the binary relation P (edges) is 3-colorable. A possible FOL sentence is

$$\begin{aligned}
&\sigma(P, Q_1, Q_2, Q_3) = \\
&\quad [\forall x : Q_1(x) \vee Q_2(x) \vee Q_3(x)] \wedge && (1)\\
&\quad [\forall x : \neg(Q_1(x) \wedge Q_2(x)) \wedge \neg(Q_2(x) \wedge Q_3(x)) \wedge \neg(Q_3(x) \wedge Q_1(x))] \wedge && (2)\\
&\quad [\forall x, y : P(x, y) \Rightarrow \\
&\qquad \neg(Q_1(x) \wedge Q_1(y)) \wedge \neg(Q_2(x) \wedge Q_2(y)) \wedge \neg(Q_3(x) \wedge Q_3(y))] && (3)
\end{aligned}$$

(Term (2) is not needed.)

Theorem 7 (Decidability of spectra) *Given a FOL expression* $\sigma(Q_1, \ldots, Q_k)$ *with* $k \geq 1$*, the value of* n*, and the relations* $Q_1, \ldots, Q_k$*, it is decidable in polynomial time if* $\sigma(Q_1, \ldots, Q_k)$ *is satisfied.*

Proof. Use the given relations to evaluate σ. This can be done very efficiently. □

Definition 6 (SPECTRA **class of decision problem)** *For each problem in the class there is a fixed FOL expression* $\sigma(Q_1, \ldots, Q_k)$
Instance: *A positive integer* n.
Question: *Is* n *in the spectra of* σ*?*
Comment. The language associated with a SPECTRA *problem is a set of integers*
Comment. One way to view the spectrum of $\sigma(Q_1, \ldots, Q_k)$ *is as the set of finite models of* $\exists Q_1, \ldots, Q_k : \sigma(Q_1, \ldots, Q_k)$*. Since all of the relation symbols in the language are quantified, and a model is simply a structure with universe of size* n *over the empty language, we can identify such a structure with the natural number* n*.*

Proof. The number of tuples $\langle Q_1, \ldots, Q_k \rangle$ in a universe of cardinality n is finite; for all those tuples evaluate σ and return TRUE iff for some tuple the value of σ is TRUE. □

Important comment. This algorithm is in NE (non-deterministic exponential) only because the representation of the instance has very small length, $|n| \approx \log n$.

5 Encoding and decision problems

In complexity, the efficiency of an algorithm is measured relatively to the *length of the input*. This implies that different input encodings usually correspond to different efficiencies. As an example, suppose that for some algorithm the input is n is represented in unary (1^n) and that the execution time is $t = n^2$. The algorithm is clearly polynomial. However, if the input is written in binary, its length is about $m = |n| \approx \log n$, and the same execution time t, expressed in terms of the input length (as it should be!), is exponential, namely $t = n^2 \approx (2^{|n|})^2 = (2^{2|n|}) = 2^{2m}$.

Consider now the following example related to the *output* length. The input is n and the output is $2^{(2^n)}$. Although the computation is simple, the output, written in binary, has length $1 + 2^n$, which is exponential in terms of n, and super-exponential in terms of $|n|$ (about $2^{2^{|n|}}$ bits). In this example the execution time of any algorithm is at least exponential, because *the time needed to write the result* is at least exponential, no matter what input encoding is used.

5.1 Encoding: very small inputs

When possible, that is, for specific forms of input, a logarithmically short coding can transform a polynomial algorithm in an exponential algorithm. The execution time is the same, but the measuring units are not.

Quoting [3]:

> ... generalized spectra are exponentially simpler (in terms of complexity) than spectra. Intuitively, it corresponds to the fact that the size of the input is exponentially bigger for generalized spectra than for spectra. For example, it takes around n^2 bits to encode a graph on n points, whereas it takes only around $\log n$ bits to encode the number n.

To avoid logarithmically (in terms of the universe) short encodings of structures, the following convention (or "trick") is used in [5]:

> In the special case where τ includes no input relation symbols, we pretend there is a unary relation symbol that is always false. For example, if $\tau = \emptyset$, then $\text{bin}(\mathcal{A}) = 0^{\|\mathcal{A}\|}$ [$\text{bin}(\mathcal{A})$ is the representation

of the structure by a string of 0's and 1's, and $n = \|\mathcal{A}\|$ is the size of the universe]. We do this to ensure that the size of $\text{bin}(\mathcal{A})$ is at least as large as $\|\mathcal{A}\|$.

5.2 Encoding: very large inputs

Very large inputs could also cause "complexity problems". For instance, if the input size is exponential in $|n|$, any algorithm would take an exponentially large time only to read the input. But this situation (exponentially large inputs) does not seem to occur "naturally". We would have for instance, to repeat 2^n times each "0" or "1"....

5.3 The G-SPECTRA (or FO$\exists$) decision problem

Definition 7 (G-SPECTRA **or** FO$\exists$ **decision problem)** *For each problem there is a fixed FOL expression*[1] $\sigma(P_1, \ldots, P_s, Q_1, \ldots Q_k)$.

Instance: *A positive integer n, the cardinality of the universe and relations P_1, P_2,..., P_s.*

Question: *For the given n, are there relations Q_1,..., Q_k such that $\sigma(P_1, \ldots, P_s, Q_1, \ldots Q_k)$ is* TRUE*?*

Comment. The language associated with a SPECTRA problem is the set of tuples of structures $\langle P_1, \ldots, P_s \rangle$ (for all integers $n \in \mathbb{N}$) for which the answer is YES.
As an example, the language associated with the problem 3COL (see page 4) is the set of graphs that are 3-colorable.

6 Finite Σ_1^1 equals NP

We now look specifically to Theorem 8 of [3] (page 6) that essentially states that the class of Σ_1^1 languages (generalized spectra) and NP are identical. In order to represent logic in Turing machines we need to encode relations as strings. Conversely, to represent configurations and computation histories of Turing machines we need to represent them as logic relations.

Theorem 8 *Let $\mathcal{L}$ be a nonempty language, and let $\mathcal{C}$ be a set of finite $\mathcal{L}$-structures that is closed under isomorphism. Then $\mathcal{C}$ is a generalized spectrum iff* $\text{enc}(\mathcal{C})$ *is in NP.*

The proof of Theorem 8 involves two directions. First consider a G-SPECTRA question,

$$\exists Q_1, \ldots, Q_k : \sigma(P_1, \ldots, P_s, Q_1, \ldots Q_k)?$$

[1] Recall the notation in page 4.

Denote by n the size of the universe. Assuming a "reasonable" encoding, relation with arity a can be represented by n^a bits, so that: (i) the tuple of relations $P_1,\ldots,\ P_s$ (instance of the problem) is represented by a string with length polynomially related to n (assuming $k \geq 1$!), (ii) similarly, the length of tape where the NDTM writes a possible representation of the tuple of relations $Q_1,\ldots,\ Q_k$ can always have length polynomially related to n. The non-deterministic computation is now clear. First (non-deterministic phase), write a string z with the appropriate length. Then (deterministic phase), verify that z represents, according to the pre-established encoding schema, the relations $Q_1,\ldots,\ Q_k$; then compute σ (represented by a string) in polynomial time Thus, G-SPECTRA is in NP.

Conversely, consider any non-deterministic, polynomial time, computation by a Turing machine M. Without entering in details, note that we have to define appropriate logical relations, and a logical statement that specifies things like

- The initial contents of the tape is, say, x.
- M in the non-deterministic polynomial time phase writes a string y.
- The transitions in the deterministic phase correspond to the description of M.
- By time n^m, M has halted and accepted w.

There are detailed proofs of Theorem 8 for instance in [2] (Theorem 6, pages 53–58) and [5] (Chapter 7). These proofs are not unlike those of Cook's Theorem ([4], pages 39–44), or of the theorem 11.2 in [1] (Chapter 4, pages 126–132).

6.1 Encoding structures by strings

A relation with arity a can be represented by n^a bits, and, to represent a tuple of relations $\langle R_1,\ldots,R_k\rangle$, we can use an additional symbol # and obtain

$$w = \text{enc}(R_1)\#\text{enc}(R_2)\#\ldots\#\text{enc}(R_k)$$

where $|w| = 2^{a_1} + 2^{a_2} + \ldots + 2^{a_k} + (k-1)$ bits.

Any other polynomially related representation (in terms of length) is equally satisfying. In Sections 5.2 and 5.1 (pages 6 and 5 respectively) we discussed codifications that *are not* polynomially related – but also not reasonable.

As an example of other representation, suppose that each entry in a relation is represented by its coordinates (row and column, written in binary). We get a larger number of bits than $|w|$, but a representation that is polynomially related with $|w|$. Even if the coordinates are written in unary, we get a polynomially related representation, because the total length is bounded by $2n \times n^a$.

Example of a problematic representation

Consider the following problem.

Instance: A graph with n nodes and no more than 10 edges.

Question: Does the graph have a path with length 5?

Suppose that the graph is represented by n and a list of no more then 10 edges, each one represented by a pair ⟨row, column⟩:

$$g = n\#i_1\#j_1\#i_2\#j_2 \ldots i_k\#j_k$$

where $k \leq 10$ and all integers are represented in binary. The length of the representation is

$$|g| = 2k + (2k+1)\log n \leq (2k+2)\log n \leq 22 \log n$$

for n sufficiently large. It follows that if an algorithm is polynomial time in terms of n, it is exponential in terms of the input length $|g|$.

6.2 Representing strings by structures

For simplicity assume a string s of 0's and 1's. We can encode s using a language $\mathcal{L} = \{U, <\}$ where U and $<$ are a unary and a binary relation symbols, respectively. If the length of s is n, the universe is $\{0, 1, \ldots, n-1\}$ and each integer of the universe represents a position in the string.

$$U(i) = \begin{cases} \text{TRUE} & \text{if } b[i] = 1 \\ \text{FALSE} & \text{if } b[i] = 0 \end{cases}$$

For instance, the relation U that corresponds to the string `011` is

$$U = \{\text{FALSE}, \text{TRUE}, \text{TRUE}\}$$

The relation "<" corresponds to the usual integer comparison.

7 How can logic represent the execution time?

The concepts of generalized spectrum or of FO∃ sentence seem completely unrelated to computing machines or execution time. Yet, Theorem 8 (page 6) describes the class of languages that can be executed by non-deterministic Turing machines (NDTM) in polynomial time in terms of a syntactic logic language (generalized spectra). How can a logic concept, the generalized spectrum (or FO∃), be related with machines, execution times, or polynomials? It isn't. The relationship mentioned in Theorem 8 holds only with some reasonable (but sometimes hidden) assumptions:

– [Non-deterministic computation.] Consider the quantified relations $Q_1, \ldots, Q_k$. For almost every reasonable method of representing these relations as

a string in a NDTM (the non-deterministic part of the computation), the length of the string (and the non-deterministic computation time) is *polynomially related with* n. This non-deterministic computation can precede all the deterministic computation. See however the sections 5.1 and 5.2 (pages 5 and 6 respectively), particularly the quotation of [3] in 5.1.

- [Representing quantified relations.] Similarly, the non-quantified relations $P_1,\ldots,\ P_k$, that are part of the NDTM input, should be represented by a string with a length polynomially related with n.
- [Efficient TM access to relations.] The computational access to the (representation of the) logic relations $Q_1,\ldots,\ Q_k,\ P_1,\ldots,\ P_s$ must be efficient (polynomial time in terms of the input length); this excludes for instance a situation where, perhaps with the goal of compressing the representation, the encoding schema is such that the information is very difficult (exponential time) to extract from the representation. This item is independent of the previous one, which is related with the length of the representation.
- [Deterministic computation.] The computation of a FO logic expression $\sigma(P_1,\ldots,P_s,Q_1,\ldots Q_k)$, where the relations Q_i and P_i are represented by a string with length polynomially related with n, can be done in polynomial time.

Relatively to the encoded length, we quote [4]:

> Since any two reasonable encoding encoding schemes for a problem Π will yield polynomially related input lengths, all our results will carry through for any such function ["length functions"] that meets the above conditions [polynomially related].

References

[1] George S. Boolos, John P. Burgess, and Richard C. Jeffrey. *Computability and Logic*. Cambridge University Press, 2007. Fifth Edition.

[2] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *SIAM-AMS*, 7:3–31, 1974. also in "Complexity and Computation", ed. R. Karp, pages 43–73.

[3] Ronald Fagin. Finite-model theory – a personal perspective. *Theoretical Computer Science*, 116:3–31, 1993.

[4] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[5] Neil Immerman. *Descriptive Complexity*. Springer, 1999.