Evaluating Formulas on Sparse Graphs

Part 1

Anuj Dawar

University of Cambridge Computer Laboratory

PhDOpen, Warsaw 1 October 2010

Complexity Theory

The study of *Complexity Theory* began in the 1960s and 1970s as an attempt to explain what makes certain computational tasks *inherently intractable*.

The main outcome of the study was the theory of *NP-completeness*.

Thousands of individual problems have been identified as *NP-complete*.

We have a strong informal understanding of what makes problems NP-complete (such as an *exponential, unstructured* search space).

> We do not have a theory of what kind of *structure* on the search space allows for tractable solution.

Structure and Specification

Many classical intractable problems (including many on Karp's original list of NP-complete problems) are decision problems on graphs.

Graphs serve as a very general form of *structure*.

The decision problem asks whether they satisfy a *specification*.

Graph Problems

1. Independent Set: *Given*: a graph G and a positive integer k

Decide: does G contain k vertices that are pairwise distinct and non-adjacent?

2. Dominating Set: *Given*: a graph G and a positive integer k

Decide: does G contain k vertices such that every vertex is among them or adjacent to one of them?

3. 3-Colourability: *Given*: a graph G

Decide: is there an assignment of three colours r, b, g to the vertices of G so that the endpoints of every edge are distinctly coloured?

4. Hamiltonicity: *Given*: a graph G

Decide: does G contain a cycle that visits every vertex exactly once?

Formalising the Specification

To talk of the *complexity of the specification* of the problem, we have to formalise the language in which the problems are specified.

Consider *first-order predicate logic*.

A collection X of variables, and formulas:

$$E(x, y) \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \neg\varphi \mid \exists x\varphi \mid \forall x\varphi$$

where $x, y \in X$.

In addition, we may sometimes allow *colours* $R(x)$ and constants $E(c, x)$.

Specifications in First-Order Logic

A formula φ without free variables specifies a property of graphs.

$$\exists x \exists y \exists z (x \neq y \wedge y \neq z \wedge x \neq z \wedge \neg E(x, y) \wedge \neg E(x, z) \wedge \neg E(y, z))$$

defines the graphs that have an independent set of size 3.

$$\exists x \exists y \exists z \forall w (x = w \vee y = w \vee z = w \vee E(x, w) \vee E(y, w) \vee E(z, w))$$

defines the graphs that have a dominating set of size 3.

More generally, we can write, for each k, formulas γ_k, δ_k that define, respectively the graphs with an independent set of size k and those with a dominating set of size k.

Complexity of First-Order Logic

What is the complexity of deciding, for a given graph G and formula φ whether or not $G \models \varphi$?

The straightforward algorithm proceeds recursively on the structure of φ:

- Atomic formulas by direct lookup.
- Boolean connectives are easy.
- If $\varphi \equiv \exists x\, \psi$ then for each v in G check whether

$$(G, x \mapsto v) \models \psi.$$

This shows that the problem can be solved in time $O(ln^m)$ and $O(m \log n)$ space, where l is the *length* of φ and n the *order* of G.
m is the nesting depth of quantifiers in φ (or by a more careful accounting, the number of distinct variables occurring in φ)

Complexity of First-Order Logic

The problem of deciding whether $G \models \varphi$ for a first-order φ is in time $O(ln^m)$ and $O(m \log n)$ space.

So, is in PSPACE and for a fixed φ, the problem of deciding membership in the class

$$\mathrm{Mod}(\varphi) = \{G \mid G \models \varphi\}$$

is in *logarithmic space* and *polynomial time*.

QBF—satisfiability of quantified Boolean formulas can be easily reduced to the FO satisfaction problem with G a fixed two-vertex graph.

Thus, the problem is PSPACE-complete, even for fixed G.

Weakness of First-Order Logic

For any fixed φ, the class of graphs G such that $G \models \varphi$ is decidable in *polynomial time* and *logarithmic space*.

There are computationally easy classes that are not defined by any first-order sentence.

- The class of graphs with an even number of vertices.
- The class of graphs (V, E) that are connected.

Both of these are known to be computable in LOGSPACE.

The latter by a celebrated result of Reingold.

Second-Order Logic

Second-order logic is obtained by adding to the defining rules of first-order logic two further clauses:

atomic formulae – $X(t_1, \ldots, t_a)$, where X is a *second-order variable*

second-order quantifiers – $\exists X \varphi$, $\forall X \varphi$

Second-order logic can express evenness and connectivity as well as properties that are deemed not to be feasibly computable, such as *graph 3-colourability*.

Indeed, it can express every *NP-complete* problem.

Examples

Evenness.

$$\exists B \exists S \quad \forall x \exists y B(x,y) \wedge \forall x \forall y \forall z B(x,y) \wedge B(x,z) \rightarrow y = z$$
$$\forall x \forall y \forall z B(x,z) \wedge B(y,z) \rightarrow x = y$$
$$\forall x \forall y S(x) \wedge B(x,y) \rightarrow \neg S(y)$$
$$\forall x \forall y \neg S(x) \wedge B(x,y) \rightarrow S(y)$$

Examples

Connectivity

$$\forall S(\exists x\, Sx \land (\forall x \forall y\, (Sx \land Exy) \rightarrow Sy)) \rightarrow \forall x\, Sx$$

$$\begin{aligned}
\forall a \forall b \exists P \quad & \forall x \forall y\, P(x,y) \rightarrow E(x,y) \\
& \exists x P(a,x) \land \exists x P(x,b) \land \neg\exists x P(x,a) \land \neg\exists x P(b,x) \\
& \forall x \forall y (P(x,y) \rightarrow \forall z (P(x,z) \rightarrow y = z)) \\
& \forall x \forall y (P(x,y) \rightarrow \forall z (P(z,x) \rightarrow y = z)) \\
& \forall x ((x \neq a \land \exists y P(x,y)) \rightarrow \exists z P(z,x)) \\
& \forall x ((x \neq b \land \exists y P(y,x)) \rightarrow \exists z P(x,z))
\end{aligned}$$

Examples

3-Colourability

$$\begin{aligned}\exists R \exists B \exists G \quad & \forall x(Rx \lor Bx \lor Gx) \land \\ & \forall x(\quad \neg(Rx \land Bx) \land \neg(Bx \land Gx) \land \neg(Rx \land Gx)) \land \\ & \forall x \forall y(Exy \rightarrow (\quad \neg(Rx \land Ry) \land \\ & \qquad \neg(Bx \land By) \land \\ & \qquad \neg(Gx \land Gy)))\end{aligned}$$

Descriptive Complexity

Theorem (Fagin)

A class of graphs is definable in *existential second-order logic* if, and only if, it is in the class NP.

A major open problem in the field of *Descriptive Complexity* has been to establish whether there is a similar descriptive characterisation of P—the class of computational problems decidable in polynomial time.

Is there any extension of first-order logic in which one can express all and only the feasibly computable problems?

Can the class P be “built up from below” by finitely many operations?

Monadic Second-Order Logic

Monadic Second-Order Logic (MSO) is the restriction of second-order logic where the second-order quantifiers are only over *sets* of vertices—not arbitrary relations.

3-colourability is MSO but not Hamiltonicity.

Guarded Second-Order Logic (or MS_2) is the restriction of second-order logic where the second-order quantifiers range over *sets of vertices* or *sets of edges*.

Hamiltonicity is MS_2.

Exercise: Show this

These restricted languages are well-behaved in many situations.

Complexity of MSO

A naïve algorithm along the lines we saw for first-order logic for evaluating MSO formulas would add the rule:

- If $\varphi \equiv \exists X\, \psi$ then for each $A \subseteq V(G)$ check whether

$$(G, X \mapsto A) \models \psi.$$

The problem of deciding whether $G \models \varphi$ for φ in MSO is in time $O(l2^{nm})$ and $O(mn)$ space.

So, the problem is in PSPACE (and therefore PSPACE-complete) but, *even for fixed* φ it can take exponential time.

We have seen that some NP-complete problems can be expressed by a fixed MSO formula φ.

Is FO contained in an initial segment of P?

Question posed in the title of a paper by **Stolboushkin and Taitslin**.

> Is there a fixed c such that for every first-order φ, $\mathrm{Mod}(\varphi)$ is decidable in time $O(n^c)$?

If P $=$ PSPACE, then the answer is yes, as the satisfaction relation is then itself decidable in time $O(n^c)$ and this bounds the time for all formulas φ.

> Thus, though we expect the answer is no, this would be difficult to prove.

A more uniform version of their question is:

> Is there a constant c and a computable function f so that the satisfaction relation for first-order logic is decidable in time $O(f(l)n^c)$?

In this case we say that the satisfaction problem is *fixed-parameter tractable* (FPT) with the formula length as parameter.

Parameterized Problems

Independent Set: *Given*: a graph G and a positive integer k

Decide: does G contain k vertices that are pairwise distinct and non-adjacent?

Dominating Set: *Given*: a graph G and a positive integer k

Decide: does G contain k vertices such that every vertex is among them or adjacent to one of them?

Here the input consists of a graph and an *integer parameter*.

For each fixed value of k, there is a first-order sentence φ_k such that $G \models \varphi_k$ if, and only if, G contains an independent set of k vertices.

Similarly for dominating set.

Parameterized Complexity

FPT—the class of problems of input size n and *parameter* l which can be solved in time $O(f(l)n^c)$ for some computable function f and constant c.

There is a hierarchy of *intractable* classes.

$$\text{FPT} \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq \text{AW}[\star]$$

Independent Set is $W[1]$-complete.

Dominating Set is $W[2]$-complete.

Parameterized Complexity of First-Order Satisfaction

Writing Π_t for those formulas which, in prenex normal form have t alternating blocks of quantifiers starting with a universal block:

> The satisfaction problem restricted to Π_t formulas (parameterized by the length of the formula) is hard for the class $W[t]$.

The satisfaction relation for first-order logic ($G \models \varphi$), parameterized by the length of φ is AW$[\star]$-complete.

Thus, a positive answer to the question of Stolboushkin and Taitslin would collapse the edifice of parameterized complexity theory.

Restricted Classes

One way to get a handle on the complexity of first-order satisfaction is to consider restricted graph classes.

Given: a first-order formula φ and a graph $G \in \mathcal{C}$

Decide: if $G \models \varphi$

For many interesting classes $\mathcal{C}$, this problem has been shown to be FPT, even for formulas of MSO.

We say that satisfaction of FO (or MSO) is *fixed-paramter tractable on* $\mathcal{C}$.

Finding Structure in Graph Classes

This course of lectures is about when we can find sufficient structure in the class $\mathcal{C}$ to make FO (MSO) satisfaction fixed-parameter tractable.

We will concentrate classes $\mathcal{C}$ of *sparse graphs* (i.e. graphs where the number of edges is much smaller than n).

We will look at proofs showing FO satisfaction is FPT on *graphs of bounded treewidth*, *planar graphs*, *classes of graphs that exclude a minor* and conclude with some conjectures that generalize all of these.

We start with a digression from graphs to look at *words*.

Logic on Words

Fix a finite alphabet Σ.

We consider formulas (of FO or SO) with atomic formulas

$$a(x)\text{(for each } a \in \Sigma) \quad and \quad x \leq y.$$

Then each formula defines a *language* in Σ^*.

Any language in NP can be defined in existential second-order logic.

Theorem (Büchi-Elgot-Trakhtenbrot)

A language L is defined by a formula of MSO if, and only if, L is regular.

Games

There are several different ways of proving this theorem.

Here we look at a proof of one direction (every MSO definable language is regular) and express it in terms of *Ehrenfeucht-style games*.

We first define these for first-order logic.

We drop, for the moment, the language of graphs, and consider any structures in a *relational* vocabulary.

$\mathbb{A}$ and $\mathbb{B}$ are structures over the same vocabulary, and A and B are their universes.

Quantifier Rank

The *quantifier rank* of a first-order formula φ, written $\mathrm{qr}(\varphi)$ is defined inductively as follows:

1. if φ is atomic then $\mathrm{qr}(\varphi) = 0$,
2. if $\varphi = \neg\psi$ then $\mathrm{qr}(\varphi) = \mathrm{qr}(\psi)$,
3. if $\varphi = \psi_1 \vee \psi_2$ or $\varphi = \psi_1 \wedge \psi_2$ then $\mathrm{qr}(\varphi) = \max(\mathrm{qr}(\psi_1), \mathrm{qr}(\psi_2))$.
4. if $\varphi = \exists x\psi$ or $\varphi = \forall x\psi$ then $\mathrm{qr}(\varphi) = \mathrm{qr}(\psi) + 1$

More informally, $\mathrm{qr}(\varphi)$ is the *maximum depth of nesting of quantifiers* inside φ.

Formulas of Bounded Quantifier Rank

Note: We assume that our signature consists only of relation and constant symbols. That is, there are *no function symbols of non-zero arity*.

With this proviso, it is easily proved that in a finite vocabulary, for each q, there are (up to logical equivalence) only finitely many sentences φ with $\mathrm{qr}(\varphi) \leq q$.

To be precise, we prove by induction on q that for all m, there are only finitely many formulas of quantifier rank q with at most m free variables.

Formulas of Bounded Quantifier Rank

If $\mathrm{qr}(\varphi) = 0$ then φ is a Boolean combination of atomic formulas. If it is has m variables, it is equivalent to a formula using the variables $x_1, \ldots, x_m$. There are finitely many formulas, *up to logical equivalence*.

Suppose $\mathrm{qr}(\varphi) = q + 1$ and the *free variables* of φ are among $x_1, \ldots, x_m$. Then φ is a Boolean combination of formulas of the form

$$\exists x_{m+1} \psi$$

where ψ is a formula with $\mathrm{qr}(\psi) = q$ and free variables $x_1, \ldots, x_m, x_{m+1}$.

By induction hypothesis, there are only finitely many such formulas, and therefore finitely many Boolean combinations.

Equivalence Relation

For two structures $\mathbb{A}$ and $\mathbb{B}$, we say $\mathbb{A} \equiv_q \mathbb{B}$ if for any sentence φ with $\mathrm{qr}(\varphi) \leq q$,

$$\mathbb{A} \models \varphi \text{ if, and only if, } \mathbb{B} \models \varphi.$$

More generally, if $\mathbf{a}$ and $\mathbf{b}$ are m-tuples of elements from $\mathbb{A}$ and $\mathbb{B}$ respectively, then we write $(\mathbb{A}, \mathbf{a}) \equiv_q (\mathbb{B}, \mathbf{b})$ if for any formula φ with m free variables $\mathrm{qr}(\varphi) \leq q$,

$$\mathbb{A} \models \varphi[\mathbf{a}] \text{ if, and only if, } \mathbb{B} \models \varphi[\mathbf{b}].$$

Types

We write $\text{Type}_q(\mathbb{A}, \mathbf{a})$ for the set of all formulas φ with $\mathrm{qr}(\varphi) \leq q$ such that $\mathbb{A} \models \varphi[\mathbf{a}]$.

$(\mathbb{A}, \mathbf{a}) \equiv_q (\mathbb{B}, \mathbf{b})$ is equivalent to $\text{Type}_q(\mathbb{A}, \mathbf{a}) = \text{Type}_q(\mathbb{B}, \mathbf{b})$.

There is a formula $\theta_{\mathbb{A},\mathbf{a}} \in \text{Type}_q(\mathbb{A}, \mathbf{a})$ such that:

if $\mathbb{B} \models \theta_{\mathbb{A},\mathbf{a}}[\mathbf{b}]$ then $(\mathbb{A}, \mathbf{a}) \equiv_q (\mathbb{B}, \mathbf{b})$. *Exercise:* Why?

We sometimes identify $\theta_{\mathbb{A},\mathbf{a}}$ with $\text{Type}_q(\mathbb{A}, \mathbf{a})$.

Partial Isomorphisms

A map f is a partial isomorphism between structures $\mathbb{A}$ and $\mathbb{B}$, if

- the domain of $f = \{a_1, \ldots, a_l\} \subseteq A$, including the interpretation of all constants;
- the range of $f = \{b_1, \ldots, b_l\} \subseteq B$, including the interpretation of all constants; and
- f is an isomorphism between its domain and range.

Note that if f is a partial isomorphism taking a tuple $\mathbf{a}$ to a tuple $\mathbf{b}$, then for any *quantifier-free* formula θ

$$\mathbb{A} \models \theta[\mathbf{a}] \text{ if, and only if, } \mathbb{B} \models \theta[\mathbf{b}].$$

Ehrenfeucht-Fraïssé Games

The q-round Ehrenfeucht game on structures $\mathbb{A}$ and $\mathbb{B}$ proceeds as follows:

- There are two players called Spoiler and Duplicator.
- At the ith round, Spoiler chooses one of the structures (say $\mathbb{B}$) and one of the elements of that structure (say b_i).
- Duplicator must respond with an element of the other structure (say a_i).
- If, after q rounds, the map

$$\{a_i \mapsto b_i \mid 1 \leq i \leq q\} \cup \{c^{\mathbb{A}} \mapsto c^{\mathbb{B}} \mid c \text{ a constant.}\}$$

is a partial isomorphism, then Duplicator has won the game, otherwise Spoiler has won.

Equivalence and Games

Write $\mathbb{A} \sim_q \mathbb{B}$ to denote the fact that *Duplicator* has a *winning strategy* in the q-round Ehrenfeucht game on $\mathbb{A}$ and $\mathbb{B}$.

The relation $\sim_q$ is, in fact, an *equivalence relation*. *Exercise: prove it.*

Theorem (Fraïssé; Ehrenfeucht)

$\mathbb{A} \sim_q \mathbb{B}$ if, and only if, $\mathbb{A} \equiv_q \mathbb{B}$

We give a proof for one direction $\mathbb{A} \sim_q \mathbb{B} \Rightarrow \mathbb{A} \equiv_q \mathbb{B}$ in some detail.

Proof

To prove $\mathbb{A} \sim_q \mathbb{B} \Rightarrow \mathbb{A} \equiv_q \mathbb{B}$, it suffices to show that if there is a sentence φ with $\mathrm{qr}(\varphi) \leq q$ such that

$$\mathbb{A} \models \varphi \quad \text{and} \quad \mathbb{B} \not\models \varphi$$

then *Spoiler* has a winning strategy in the q-round Ehrenfeucht game on $\mathbb{A}$ and $\mathbb{B}$.

Assume that φ is in *negation normal form*, i.e. all negations are in front of atomic formulas.

Note that this does not involve an increase in quantifier rank.

Proof

We prove by induction on q the stronger statement that if φ is a formula with $\mathrm{qr}(\varphi) \leq q$ and $\mathbf{a} = (a_1, \ldots, a_m)$ and $\mathbf{b} = (b_1, \ldots, b_m)$ are tuples of elements from $\mathbb{A}$ and $\mathbb{B}$ respectively such that

$$\mathbb{A} \models \varphi[\mathbf{a}] \quad \text{and} \quad \mathbb{B} \not\models \varphi[\mathbf{b}]$$

then *Spoiler* has a winning strategy in the q-round Ehrenfeucht game which starts from a position in which $a_1, \ldots, a_m$ and $b_1, \ldots, b_m$ have *already been selected*.

Proof

When $q = 0$, φ is a quantifier-free formula. Thus, if

$$\mathbb{A} \models \varphi[\mathbf{a}] \quad \text{and} \quad \mathbb{B} \not\models \varphi[\mathbf{b}]$$

there is an *atomic* formula θ that distinguishes the two tuples and therefore the map taking $\mathbf{a}$ to $\mathbf{b}$ is not a *partial isomorphism*.

When $q = p + 1$, φ is a Boolean combination for atomic formulas and formulas of the form $\exists x\psi$ or $\forall x\psi$ such that $\mathrm{qr}(\psi) \leq p$. Suppose there is such a subformula θ with

$$\mathbb{A} \models \theta[\mathbf{a}] \quad \text{and} \quad \mathbb{B} \not\models \theta[\mathbf{b}]$$

If $\theta = \exists x\psi$, *Spoiler* chooses a witness for x in $\mathbb{A}$.

If $\theta = \forall x\psi$, $\mathbb{B} \models \exists x\neg\psi$ and *Spoiler* chooses a witness for x in $\mathbb{B}$.

Proof (*sketch of converse*)

The proof of the converse ($\mathbb{A} \equiv_k \mathbb{B} \Rightarrow \mathbb{A} \sim_q \mathbb{B}$ again proceed by induction on q.

If $(\mathbb{A}, \mathbf{a}) \not\sim_q (\mathbb{B}, \mathbf{b})$ and this is witnessed by *Spoiler* choosing $a \in A$, then take the formula

$$\exists x \theta_{\mathbb{A}, \mathbf{a}a}$$

.

MSO Game

The m-round monadic Ehrenfeucht game on structures $\mathbb{A}$ and $\mathbb{B}$ proceeds as follows:

- At the ith round, *Spoiler* chooses one of the structures (say $\mathbb{B}$) and plays either a point move or a set move.

  In a point move, it chooses one of the elements of the chosen structure (say b_i) – *Duplicator* must respond with an element of the other structure (say a_i).

  In a set move, it chooses a subset of the universe of the chosen structure (say S_i) – *Duplicator* must respond with a subset of the other structure (say R_i).

MSO Game

- If, after m rounds, the map

$$a_i \mapsto b_i$$

is a partial isomorphism between

$$(\mathbb{A}, R_1, \ldots, R_q) \text{ and } (\mathbb{B}, S_1, \ldots, S_q)$$

then *Duplicator* has won the game, otherwise *Spoiler* has won.

MSO Game

If we define the *quantifier rank* of an MSO formula by adding the following inductive rule to those for a formula of FO:

if $\varphi = \exists S \psi$ or $\varphi = \forall S \psi$ then $\mathrm{qr}(\varphi) = \mathrm{qr}(\psi) + 1$

then, we have

Duplicator has a winning strategy in the m-round monadic Ehrenfeucht game on structures $\mathbb{A}$ and $\mathbb{B}$ if, and only if, for every sentence φ of MSO with $\mathrm{qr}(\varphi) \leq m$

$$\mathbb{A} \models \varphi \quad \text{if, and only if,} \quad \mathbb{B} \models \varphi$$

MSO Types

We write $\text{Type}_m^{\text{MSO}}(\mathbb{A}, \mathbf{a})$ to denote the set of all MSO formulas of quantifier rank at most m satisfied by $(\mathbb{A}, \mathbf{a})$.

We write $(\mathbb{A}, \mathbf{a}) \equiv_m^{\text{MSO}} (\mathbb{B}, \mathbf{b})$ to denote

$$\text{Type}_m^{\text{MSO}}(\mathbb{A}, \mathbf{a}) = \text{Type}_m^{\text{MSO}}(\mathbb{B}, \mathbf{b})$$

Just as for FO, there are only finitely many formulas of MSO with quantifier rank m and s free variables.

There is a single formula $\theta_{\mathbb{A},\mathbf{a}}$ that characterizes $\text{Type}_m^{\text{MSO}}(\mathbb{A}, \mathbf{a})$.

MSO on Words

Theorem (Büchi-Elgot-Trakhtenbrot)

For any sentence φ of MSO, the language $L_\varphi = \{w \mid s \text{ a word and } w \models \varphi\}$ is regular.

Suppose u_1, u_2, v_1, v_2 are words over an alphabet Σ such that

$$u_1 \equiv_m^{\text{MSO}} u_2 \quad \text{and} \quad v_1 \equiv_m^{\text{MSO}} v_2$$

then $u_1 \cdot v_1 \equiv_m^{\text{MSO}} u_2 \cdot v_2$.

Dulpicator has a winning strategy on the game played on the pair of words $u_1 \cdot v_1, u_2 \cdot v_2$ that is obtained as a composition of its strategies in the games on u_1, u_2 and v_1, v_2.

Myhill-Nerode Theorem

Theorem (Myhill-Nerode)

A language L is regular *if, and only if,* there is an equivalence relation $\sim$ on strings such that:

1. $\sim$ has finite index on the set of all strings;
2. $\sim$ is a congruence for string concatenation, i.e.

$$s_1 \sim t_1 \text{ and } s_2 \sim t_2 \quad \Rightarrow \quad s_1 \cdot s_2 \sim t_1 \cdot t_2;$$

   and

3. L is the union of some number of $\sim$-equivalence classes.

MSO Languages

φ—an MSO sentence of quantifier rank m.

- $\equiv_m^{\text{MSO}}$ has finite index since there are, up to logical equivalence, only finitely many MSO sentences of quantifier rank at most m.
- $\equiv_m^{\text{MSO}}$ is a congruence for concatenation by an easy argument using *Ehrenfeucht-Fraïssé games* (a special case of the *Feferman-Vaught theorem*).
- It is immediate that L_φ is closed under $\equiv_m^{\text{MSO}}$.