Descriptive Polynomial Complexity

Anuj Dawar

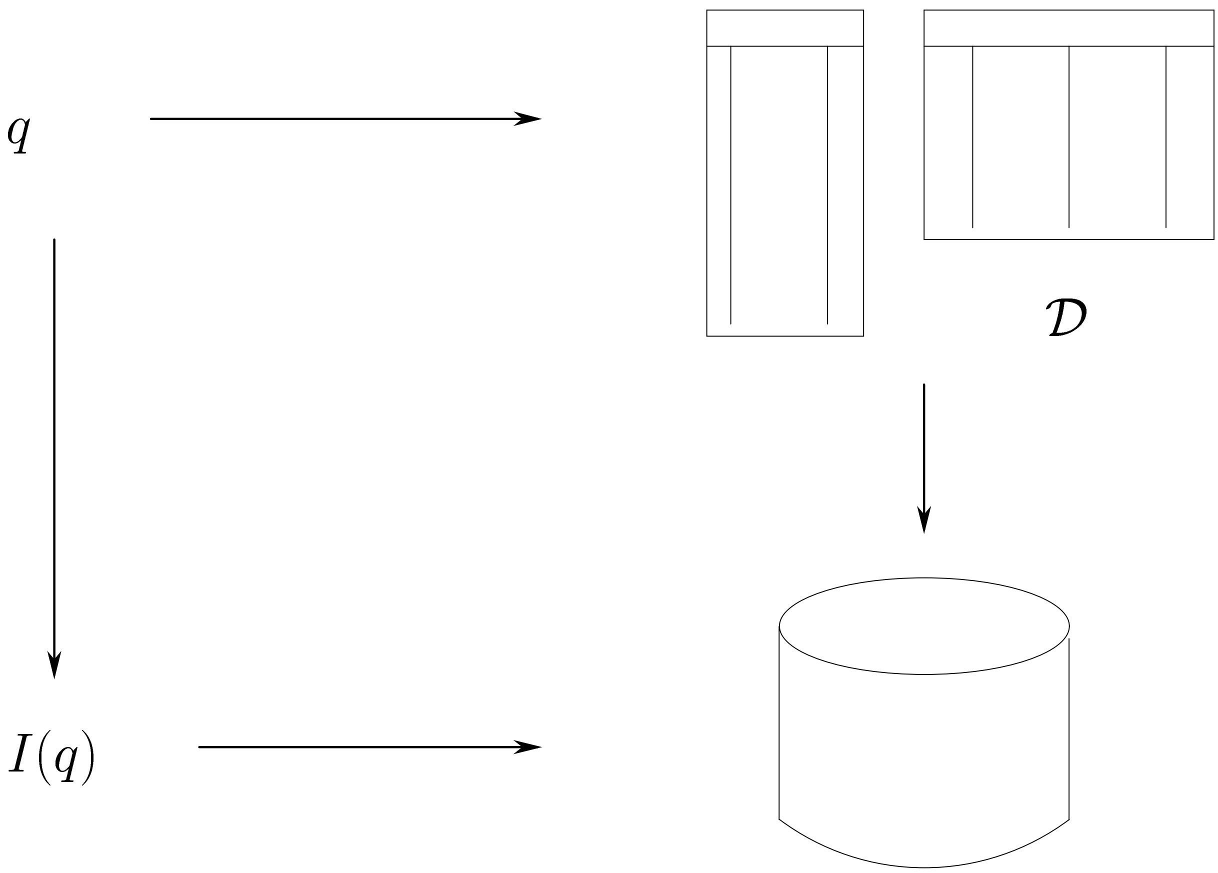
Computer Laboratory
University of Cambridge

Cambridge
February 1999

Question:

Is there a database query language in which one can express exactly the feasible, generic queries?

(Chandra-Harel 1980)



Descriptive Complexity

Computational Complexity:

space, time or other resources on a machine model of computation

complexity of a language - i.e. a set of strings

Descriptive Complexity:

classes of structures - *e.g.* a collection of graphs

the complexity of describing the collection

Relational Structures

In general, finite relational structures

$$\mathcal{A} = (A, R_1, \ldots, R_m, c_1, \ldots, c_n)$$

The description language is Predicate Logic

atomic formulas – $R(t_1, \ldots, t_m)$, $t_1 = t_2$

boolean operations – $\varphi \wedge \psi$, $\varphi \vee \psi$, $\neg\varphi$

first-order quantifiers – $\exists x \varphi$, $\forall x \varphi$

Other extensions

operators – *e.g.* fixed-point operators

higher order quantifiers

infinitary connectives

First-Order Logic

Mod(φ) – the collection of structures that satisfy the sentence φ.

If φ is first-order, then Mod(φ) $\in$ DSPACE$[\log n]$.

$\psi(x_1, \ldots, x_k)$ is a sub-formula of φ:

there are at most n^k tuples satisfying this formula

they can be represented using $k \log n$ bits

Second-Order Quantifiers

Existential Second-Order Quantification:

$$\exists P_1 \ldots \exists P_m \varphi$$

A structure $\mathcal{A}$ satisfies $\exists P \varphi$ if there is a relation R on the universe of $\mathcal{A}$ such that $(\mathcal{A}, R)$ satisfies φ.

Σ_1^1 – existential second order logic

$$\Sigma_1^1 \subseteq \mathsf{NP}$$

An existential second order quantifier represents a polynomial amount of non-determinism.

Example

3-colourability

$$
\begin{aligned}
\exists R \exists B \exists G \;\; & \forall x (Rx \vee Bx \vee Gx) \wedge \\
& \forall x (\;\; \neg(Rx \wedge Bx) \wedge \neg(Bx \wedge Gx) \wedge \\
& \qquad\quad \neg(Rx \wedge Gx)) \wedge \\
& \forall x \forall y (Exy \to (\;\; \neg(Rx \wedge Ry) \wedge \\
& \qquad\qquad\qquad\qquad \neg(Bx \wedge By) \wedge \\
& \qquad\qquad\qquad\qquad \neg(Gx \wedge Gy))
\end{aligned}
$$

Theorem (Fagin) $\Sigma^1_1 = \mathsf{NP}$

Corollary (Stockmeyer) $\mathsf{SO} = \mathsf{PH}$

$\Sigma^1_n = \Sigma^p_n$

Is there a logic that expresses exactly P?

Connectivity

There is no sentence φ of first-order logic such that a graph satisfies φ if and only if it is connected

If R is the smallest relation on a graph (V, E), such that

$$R(x, y) \equiv x = y \vee \exists z(E(x, z) \wedge R(z, y))$$

then, $\forall x \forall y R(x, y)$ expresses connectivity.

Inductive Logic

Let $\varphi(R, x_1, \ldots, x_k)$ *be a first-order formula in the vocabulary* $\sigma \cup \{R\}$

Associated operator Φ*:*

$$\Phi(R^{\mathcal{A}}) = \{(a_1, \ldots a_k) | (\mathcal{A}, R^{\mathcal{A}}) \models \varphi(R, x_1, \ldots, x_k)\}$$

Φ *is monotone if for any relations* R *and* S *such that* $R \subseteq S$*,* $\Phi(R) \subseteq \Phi(S)$

If Φ *is monotone, it has a least fixed point*

The least fixed point of Φ is obtained by iterating it

$$\Phi^0 = \emptyset$$

$$\Phi^{m+1} = \Phi\big(\Phi^m\big)$$

Then, for some m, $\Phi^{m+1} = \Phi^m =$ the least fixed point of Φ and $m \leq n^k$, where n is the size of $\mathcal{A}$

A sufficient syntactic condition for the formula φ to define a monotone map on all structures is that φ be positive in R

LFP

The language LFP is obtained by closing first order logic under an operation for forming the least fixed points of positive formulas:

LFP(σ)

- if φ is first-order formula over σ, then $\varphi \in$ LFP(σ)
- if φ is formed from formulas in LFP(σ) by conjunction, disjunction, negation and first-order quantification, then $\varphi \in$ LFP(σ), and
- if $\varphi \in$ LFP$(\sigma \cup \{R\})$, φ is positive in R and $x_1, \ldots, x_k$ are distinct variables, where k is the arity of R, then $\mathbf{lfp}(R, x_1 \ldots x_k)\varphi(t_1 \ldots t_k) \in$ LFP(σ) for any terms $t_1, \ldots, t_k$.

Example:

Let $\varphi(R, x, y)$ be $x = y \vee \exists z(E(x, z) \wedge R(z, y))$

Then, $\mathbf{lfp}(R, x, y)\varphi(x, y)$ is a formula in two free variables that expresses the transitive closure of E.

$\forall x \forall y \mathbf{lfp}(R, x, y)\varphi(x, y)$ expresses connectivity

Consider only structures where a linear order on the domain is given as a logical relation.

Theorem (Immerman, Vardi)
On ordered structures LFP = P

In the absence of order, LFP cannot express some simple properties like "the size of a graph is even."

Unordered Structures

Is there a natural logic for the Polynomial time properties of all structures?

Or more broadly:

Are the P-time properties of all structures recursively enumerable?

Consider graphs – structures over the signature $\{E\}$.

A graph of size n can be encoded as a binary string of length n^2.

$n!$ ways of encoding a graph.

$M_0, \ldots, M_i, \ldots$ – an enumeration of polynomially-clocked Turing machines.

Can we enumerate those that compute graph properties?

Alternatively:

$\varphi_0 \cdots \varphi_i \cdots$
enumeration of sentences of LFP$(\{E, <\})$.

Can we provide a recursive indexing, up to equivalence, of the order-invariant sentences?

The class P *is said to be recursively indexable, if there is a recursive set* $\mathcal{I}$ *and a Turing machine* M *such that:*

- on input $i \in \mathcal{I}$, M produces the code for a machine $M(i)$ and a polynomial p_i
- the class of structures accepted by $M(i)$, is a class in P
- $M(i)$ runs in time bounded by p_i
- for each class of structures $C \in$ P, there is an i such that $M(i)$ accepts C

Suppose M – a machine that canonically labels graphs:

$$M(G) \cong G \quad \text{and } G_1 \cong G_2 \Rightarrow M(G_1) = M(G_2)$$

Enumerate $M \to M_i$

It is not known if canonical labeling can be done in polynomial time.

If the polynomial time graph properties are not recursively enumerable, then canonical labeling cannot be done in P, and P $\neq$ NP.

Other descriptive characterisations

Other machine independent characterisations of polynomial time complexity.

- **Cobham** — functions defined through restricted forms of primitive recursion.
- **Buss** — provably total functions in weak proof systems.
- **Leivant** — extending to functions on terms.
- **McAllister** — term structures.

These characterise polynomial-time computation over numbers, terms, lists or other *inherently ordered* objects.

Extending LFP

What do we need to add to the language LFP in order to capture all of P?

Immerman conjectured that it would suffice to add a *counting* mechanism.

That is, for any formula $\varphi(x)$, we have a term $\#x\varphi(x)$, which denotes the *number* of elements that satisfy φ, and we have arithmetic operations on terms.

Cai, Fürer, Immerman showed that this is not enough.

However, LFP+counting is the best known approximation, form below to generic P.

Generalised Quantifiers

Hella *generalised the result of Cai et al. to the following*

> For any set of generalised quantifiers $\mathbf{Q}$, of bounded arity, there is a polynomial time property not definable in LFP($\mathbf{Q}$)

A generalised quantifier Q is associated with some class of structures C.

$$Q\bar{x}\bar{y}(\varphi(\bar{x}), \psi(\bar{y}))$$

asks if the structure defined by the formulas φ, ψ is in the class C.

In general, for any regular logic $\mathcal{L}$, $\mathcal{L}(Q)$ is the minimal regular logic that can express C.

Vectorisations

For any quantifier Q, we can form its vectorisations:

$$Q^k \quad \text{for all } k$$

where

$$Q^k \bar{x}\bar{y}(\varphi(\bar{x}), \psi(\bar{y}))$$

asks if the structure defined by the formulas φ, ψ on k-tuples is in the class C.

Example

If C is the class of connected graphs, then:

$$Qxy(\varphi(x,y))$$

is true if the graph defined by the formula φ is connected.

$$Q^k\bar{x}\bar{y}(\varphi(\bar{x},\bar{y}))$$

is true if the graph on k-tuples defined by φ is connected.

The following are equivalent:

- P is recursively indexable.
- There is a logic capturing P of the form FO($\mathbf{Q}$), where $\mathbf{Q}$ is the collection of vectorisations of a single quantifier.
- There is a complete problem in P under first order reductions.

(D. 1995)

Classes of graphs

In general, one can construct a logic which expresses all polynomial time properties of some restricted class of graphs, if the canonical labeling problem for that class is in polynomial time.

- Planar graphs
- Graphs of bounded genus
- Graphs of bounded tree width
- Graphs of bounded degree

In the first three cases, LFP+counting suffices.

(Grohe 1998)

Weaker Equivalences

Coinsider relations, weaker than isomorphism, which may be polynomial time decidable.

$$\equiv^k$$

Canonical labeling is still hard **(Grohe 1998)**

Positive results

- $\equiv^2$
- bisimulation

(Otto, 1997-98)

Summary

Chandra and Harel's question whether there is a logic corresponding exactly to P remains open.

Is there is no such logic, then P$\neq$NP.

The question is equivalent, in general, to asking if there is a problem complete for P with respect to first-order reductions.

The study yields insight into the trade-off between feasibility and nice structural properties.