Constraint Satisfaction Problems and Descriptive Complexity

Anuj Dawar

University of Cambridge

AIM workshop on Universal Algebra, Logic and CSP

1 April 2008

Descriptive Complexity

Descriptive Complexity seeks to classify computational problems on finite structures (i.e. *queries*) according to their definability in different logics.

Investigate the connection between

the *descriptive complexity* of queries (i.e. the logical resources required to describe them) and

their *computational complexity* (measured in terms of space, time, etc. on a suitable model of computation).

Logic and Complexity

Recall:

For a logic L

The *combined complexity* of L is the complexity of determining, given a structure $\mathbf{A}$ and a sentence $\varphi \in L$, whether or not $\mathbf{A} \models \varphi$.

The *data complexity* of L is in the complexity class C, if every query definable in L is in C.

We say L *captures* the complexity class C if the data complexity of L is in C, and every query whose complexity is in C is definable in L.

Fagin's Theorem

A formula of *Existential Second-Order Logic* (ESO) of vocabulary σ is of the form:

$$\exists R_1 \cdots \exists R_m \psi$$

- $R_1, \ldots, R_m$ are relational variables
- ψ is FO formula in the vocabulary $\sigma \cup \{R_1, \ldots, R_m\}$.

Theorem (Fagin 1974)

ESO captures NP.

Corollary

For each $\mathbf{B}$, CSP$(\mathbf{B})$ is definable in ESO.

3-Colourability

The following formula is true in a graph (V, E) if, and only if, it is 3-colourable.

$$
\begin{aligned}
\exists R \exists B \exists G \quad & \forall x (R(x) \vee B(x) \vee G(x)) \wedge \\
& \forall x (\neg (R(x) \wedge B(x)) \wedge \neg (B(x) \wedge G(x)) \wedge \neg (R(x) \wedge G(x))) \wedge \\
& \forall x \forall y (E(x, y) \rightarrow ((R(x) \wedge B(y)) \vee (B(x) \wedge R(y)) \vee \\
& \qquad\qquad\qquad (B(x) \wedge G(y)) \vee (G(x) \wedge B(y)) \vee \\
& \qquad\qquad\qquad (G(x) \wedge R(y)) \vee (R(x) \wedge G(y))))
\end{aligned}
$$

This example easily generalises to give a direct translation from $\mathbf{B}$ to a formula of ESO expressing CSP$(\mathbf{B})$.

CSPs in ESO

For a fixed structure $\mathbf{B}$, take a monadic second-order variable P_b for each element b in $\mathbf{B}$.

$$\begin{aligned} \exists P_{b_1} \cdots \exists P_{b_m} \quad & \forall x (\bigvee_{b \in \mathbf{B}} P_b(x)) \wedge \\ & \forall x (\bigwedge_{b \neq b'} \neg (P_b(x) \wedge P_{b'}(x))) \\ & \forall \bar{x} (\bigvee_{R \in \sigma} R(\bar{x}) \rightarrow (\bigvee_{\bar{b} \in R^{\mathbf{B}}} \bigwedge_i P_{b_i}(x_i))) \end{aligned}$$

This formula defines CSP$(\mathbf{B})$.

The formulas of ESO we obtain this way are of a special syntactic form:

- the first-order quantifiers are all *universal*;
- the second-order quantifiers are all *monadic*;
- all occurrences of unquantified relations are *negative*; and
- the equality symbol $=$ is not used.

MMSNP

Monotone, Monadic SNP without inequalities (MMSNP) was defined by Feder and Vardi as a syntactic fragment of ESO

SNP consists of formulas of ESO in which the first-order part is *universal*.

MMSNP consists of those formulas of SNP in which

- second-order quantifiers are monadic;
- unquantified relations are either all positive or all negative;
- the equality symbol does not appear in the scope of a negation.

MMSNP can express every query of the form CSP$(\mathbf{B})$, and more.

Dichotomy Results

Theorem (Feder-Vardi, Kun)

For every query Q in MMSNP, there is a $\mathbf{B}$ such that Q is equivalent to CSP$(\mathbf{B})$ under polynomial-time reductions.

Dichotomy Conjecture: Every query in MMSNP is either in P or NP-complete.

Let C be a class of queries obtained by dropping one of the three syntactic restrictions in the definition of MMSNP.

Theorem (Feder-Vardi)

For every query in NP, there is a query in C which is equivalent under polynomial-time reductions.

Courcelle's Theorem

One consequence of the syntactic restriction is that it enables us to deploy powerful *algorithmic meta-theorems*.

Theorem (Courcelle)

For any sentence φ of monadic second-order logic and every k, there is a polynomial time algorithm which, given a structure $\mathbf{A}$ of *treewidth* at most k will decide whether $\mathbf{A} \models \varphi$.

Corollary

CSP$(\mathbf{B})$ is tractable when restricted to inputs of bounded treewidth.

More on *treewidth* in Martin Grohe's talk.

Classifying Tractable CSPs

We next consider the definability of CSPs in logics whose *data complexity* is in P.

Recall:

- Data complexity of FO is in L, but it cannot express 2-colourability.
- Data complexity of Datalog is in P.
- For CSPs, Datalog is strictly more expressive than FO.

Datalog cannot express solvability of linear equations over $\mathbb{Z}_2$. (Feder-Vardi)

Ehrenfeucht-Fraïssé Games

Games provide a useful method for proving that certain properties are not definable in a logic.

There are many variations for different logics.

Two players (*Spoiler* and *Duplicator*) play on structures $\mathbf{A}$ and $\mathbf{B}$ with k pairs of pebbles $(a_1, b_1), \ldots, (a_k, b_k)$ for m rounds.

- at each move, *Spoiler* chooses a pebble a_i or b_i and places it on an element of the corresponding structure;
- *Duplicator* places the matching pebble on an element of the other structure;
- *Spoiler* wins the game if the partial map defined by $a_i \mapsto b_i$ is not a *partial isomorphism*.

Games and Equivalence

- *Duplicator* has a strategy to survive m rounds of the game on a pair of structures $\mathbf{A}$ and $\mathbf{B}$ if, and only if, $\mathbf{A}$ and $\mathbf{B}$ agree on all first-order sentences with *quantifier rank* at most m and using at most k distinct variables.

- *Duplicator* has a strategy to survive m rounds of the game against a *Spoiler* who only plays on $\mathbf{A}$ if, and only if, every *existential* sentence with *quantifier rank* at most m and using at most k distinct variables that is true in $\mathbf{A}$ is also true in $\mathbf{B}$.

- *Duplicator* has a strategy to maintain a *partial homomorphism* against a *Spoiler* who only plays on $\mathbf{A}$ if, and only if, every *existential positive* sentence with *quantifier rank* at most m and using at most k distinct variables that is true in $\mathbf{A}$ is also true in $\mathbf{B}$.

Proof (by example)

Suppose $\theta(x, y, z)$ is quantifier free, such that: $\mathbf{A} \models \exists x \forall y \exists z \theta$ and $\mathbf{B} \models \forall x \exists y \forall z \neg\theta$.

round 1: *Spoiler* chooses $a_1 \in A$ such that $\mathbf{A} \models \forall y \exists z \theta[a_1]$.

Duplicator responds with $b_1 \in B$.

round 2: *Spoiler* chooses $b_2 \in B$ such that $\mathbf{B} \models \forall z \neg\theta[b_1, b_2]$
Duplicator responds with $a_2 \in A$.

round 3: *Spoiler* chooses $a_3 \in A$ such that $\mathbf{A} \models \theta[a_1, a_2, a_3]$
Duplicator responds with $b_3 \in B$

Spoiler wins, since $\mathbf{B} \models \neg\theta[b_1, b_2, b_3]$.

Infinitary Logic

$L_{\infty\omega}$—extension of FO with infinitary conjunctions and disjunctions.

$\exists L^{+}_{\infty\omega}$—existential positive fragment of $L_{\infty\omega}$.

$\exists L^{k,+}_{\infty,\omega}$—fragment of $\exists L^{+}_{\infty\omega}$ using only k variables.

$\exists L^{\omega,+}_{\infty,\omega} = \bigcup_k \exists L^{k,+}_{\infty,\omega}$

$L^{k}_{\infty,\omega}$—fragment of $L_{\infty\omega}$ using only k variables.

$L^{\omega}_{\infty,\omega} = \bigcup_k L^{k}_{\infty,\omega}$

We have seen, every query in k-Datalog is in $\exists L^{k,+}_{\infty,\omega}$.

Consequently, every query in Datalog is in $\exists L^{\omega,+}_{\infty,\omega}$.

Infinite Games

Play with k pairs of pebbles on a pair of structures $\mathbf{A}$ and $\mathbf{B}$ with no limit on the number of rounds.

Duplicator has a strategy to maintain a partial homomorphism forever while *Spoiler* plays only in $\mathbf{A}$ if, and only if, every $\exists L^{k,+}_{\infty,\omega}$ sentence true in $\mathbf{A}$ is true in $\mathbf{B}$.
Existential k-pebble game

Duplicator has a strategy to maintain a partial isomorphism forever with *Spoiler* allowed to play in either structure if, and only if, $\mathbf{A}$ and $\mathbf{B}$ agree on every $L^{k}_{\infty,\omega}$ sentence.
k-pebble game

For *finite* $\mathbf{A}$ and $\mathbf{B}$, this is true if, and only if, $\mathbf{A}$ and $\mathbf{B}$ agree on all sentences of FO^k.
(Kolaitis-Vardi)

Using Games

To show that a query Q is not definable in FO, we find, for every m, a pair of structures $\mathbf{A}_m$ and $\mathbf{B}_m$ such that

- $\mathbf{A}_m \in Q$, $\mathbf{B}_m \in \overline{Q}$; and
- *Duplicator* wins an m round game with m pairs of pebbles on $\mathbf{A}_m$ and $\mathbf{B}_m$.

To show that a query Q is not definable in $\exists L^{\omega,+}_{\infty,\omega}$ (and hence, not in Datalog), we find, for every k, a pair of structures $\mathbf{A}_k$ and $\mathbf{B}_k$ such that

- $\mathbf{A}_k \in Q$, $\mathbf{B}_k \in \overline{Q}$; and
- *Duplicator* wins the *existential k-pebble* game on $\mathbf{A}_k$ and $\mathbf{B}_k$.

2-Colourability

C_n—a cycle of length n.

Duplicator wins the m round game on C_{2^m} and C_{2^m+1}.

2-Colourability is not definable in FO.

K_n—clique on n vertices.

Duplicator wins the infinite k-pebble game on K_k and K_{k+1}.

Even cardinality is not definable in $L^{\omega}_{\infty,\omega}$.

LFP

LFP is a logic that extends FO by allowing recursive definitions.

Like Datalog but in rules

$$R(\bar{x}) : - \varphi,$$

φ need not be a conjunctive query. Allow any formula of FO as long as IDBs only appear positively.

Alternating Reachability:

$$\begin{aligned} R(x) &\; : - \quad F(x) \\ R(x) &\; : - \quad \mathsf{Exi}(x) \land \exists y(E(x,y) \land R(x)) \\ R(x) &\; : - \quad \mathsf{Uni}(x) \land \forall y(E(x,y) \to R(x)) \end{aligned}$$

LFP

Datalog is just the existential positive fragment of LFP, i.e. those formulas of LFP in which negation and universal quantification do not appear.

The *data complexity* of LFP is P-complete.

If we consider only structures that include a *linear order* of their universe, LFP *captures* P. (Immerman; Vardi)

LFP $\subseteq L^{\omega}_{\infty,\omega}$.

Homomorphism Preservation

For any $\mathbf{B}$ if CSP$(\mathbf{B})$ is in FO, $\neg$CSP$(\mathbf{B})$ is in $\exists$FO^{+}. (Atserias)

If a Boolean query Q is in FO and closed under homomorphisms, then it is in $\exists$FO^{+}. (Rossman)

If CSP$(\mathbf{B})$ is in LFP, is $\neg$CSP$(\mathbf{B})$ in Datalog?

There is a Q in LFP, closed under homomorphisms but not definable in Datalog (D.-Kreutzer)

Picture

$$
\begin{array}{ccccc}
\mathsf{FO} & \subset & \mathsf{LFP} & \subset & L^{\omega}_{\infty,\omega} \\
\cup & & \cup & & \cup \\
\exists\mathsf{FO}^{+} & \subset & \mathsf{Datalog} & \subset & \exists L^{\omega,+}_{\infty,\omega}
\end{array}
$$

Note: Every CSP in FO is in $\exists$FO^{+} and every CSP in $\exists L^{\omega,+}_{\infty,\omega}$ is in Datalog.

We know that there are CSPs in Datalog not in FO.

The strictness of other inclusions in the above picture (*for CSPs*) remains open.

Adding Counting

LFP cannot express simple counting properties.

Immerman proposed an extension of LFP with a mechanism for counting.

LFPC

- two sorts of variables: x ranging over the elements of the structure $\mathbf{A}$ and ν ranging over numbers $\{0, \ldots, |A|\}$;
- for a formula φ, $\#x\,\varphi$ is a term denoting the number of elements satisfying φ;
- terms 0 and $\tau + 1$ for τ a term of numeric sort.

It was once conjectured that LFPC captures P.

Datalog with Counting

We could add counting to Datalog in a similar way.

> We allow terms $\#x\,\varphi$, $\tau + 1$ and equalities $\tau_1 = \tau_2$ between terms on the right-hand side of rules.

Datalog with counting has the same expressive power as LFPC.

(Grädel-Otto)

We can combine counting and recursion to simulate negation.

Infinitary Logic with Counting

LFPC can be translated into $C^{\omega}_{\infty\omega}$—an *infinitary logic with counting*.

$C^{\omega}_{\infty\omega}$ is obtained from first-order logic by allowing:

- *infinitary* conjunctions and disjunctions.
- *counting quantifiers*: $\exists^i x\, \varphi$
- only finitely many distinct variables in any formula.

$C^{k}_{\infty\omega}$ is the fragment of $C^{\omega}_{\infty\omega}$ where each formula has at most k variables.

LFPC is the P-uniform fragment of $C^{\omega}_{\infty\omega}$ (Otto).

Bijection Games

$C^k_{\infty\omega}$ is characterised by a k-pebble *bijection game*. (Hella).

The game is played on structures $\mathbf{A}$ and $\mathbf{B}$ with k pairs of pebbles.

- Spoiler chooses a pair of pebbles a_i and b_i;
- Duplicator chooses a bijection $h : A \rightarrow B$ such that for pebbles a_j and $b_j (j \neq i)$, $h(a_j) = b_j$;
- Spoiler chooses $a \in A$ and places a_i on a and b_i on $h(a)$.

Duplicator loses if the partial map $a_i \mapsto b_i$ is not a partial isomorphism.

Duplicator has a strategy to play forever if, and only if, $\mathbf{A}$ and $\mathbf{B}$ agree on all sentences of $C^k_{\infty\omega}$.

Counting is Not Enough

Theorem

There are polynomial-time queries on graphs that are not definable in $C^{\omega}_{\infty\omega}$.

(Cai-Fürer-Immerman)

Indeed, CSP$(\mathbb{Z}_2)$ is not definable.

Theorem

If CSP$(\mathbf{B})$ is definable in $C^{\omega}_{\infty\omega}$ then the variety of the algebra of $\mathbf{B}$ omits types 1 and 2.

(Atserias, Bulatov, D.)

Limits of $C^{\omega}_{\infty\omega}$

$$\begin{array}{ccccc}
\text{FOC} & \subset & \text{LFPC} & \subset & C^{\omega}_{\infty\omega} \\
\cup & & \cup & & \cup \\
\text{FO} & \subset & \text{LFP} & \subset & L^{\omega}_{\infty,\omega} \\
\cup & & \cup & & \cup \\
\exists\text{FO}^{+} & \subset & \text{Datalog} & \subset & \exists L^{\omega,+}_{\infty,\omega}
\end{array}$$

Is there a $\mathbf{B}$ such that

- CSP$(\mathbf{B}) \in C^{\omega}_{\infty\omega}$; and
- $\neg$CSP$(\mathbf{B}) \notin$ Datalog.

A Dichotomy Conjecture

Conjecture: For each $\mathbf{B}$, $\neg$CSP$(\mathbf{B})$ is either definable in Datalog or undefinable in $C^{\omega}_{\infty\omega}$.

This would be a consequence of the *Bounded Width* Conjecture of Larose-Zádori.

Fixed-Point Logic with Rank

We can define a logic LFPR that extends LFP with an operator for *matrix rank*.

LFPR properly extends the expressive power of LFPC.

CSP$(\mathbb{Z}_2)$ is definable in LFPR.

The data complexity of LFPR is contained in P.

Is every tractable CSP definable in LFPR?

More generally, is there a logic whose data complexity is in P and which expresses all tractable CSPs?