Descriptive Complexity and Polynomial Time
A Tutorial

Anuj Dawar
University of Cambridge

WoLLIC, Edinburgh, 2 July 2008
Descriptive Complexity

*Descriptive Complexity* provides an alternative perspective on Computational Complexity.

**Computational Complexity**

- Measure use of resources (space, time, *etc.*) on a machine model of computation;
- Complexity of a language—i.e. a set of strings.

**Descriptive Complexity**

- Complexity of a class of structures—e.g. a collection of graphs.
- Measure the complexity of describing the collection in a formal logic, using resources such as variables, quantifiers, higher-order operators, *etc.*

There is a fascinating interplay between the views.
First-Order Logic

For a first-order sentence $\varphi$, we ask what is the \textit{computational complexity} of the problem:

Given: a structure $\mathcal{A}$

Decide: if $\mathcal{A} \models \varphi$

In other words, how complex can the collection of finite models of $\varphi$ be?

In order to talk of the complexity of a class of finite structures, we need to fix some way of representing finite structures as strings.
Encoding Structures

We use an alphabet $\Sigma = \{0, 1, \#, -\}$.

For a structure $A = (A, R_1, \ldots, R_m, f_1, \ldots, f_l)$, fix a linear order $<$ on $A = \{a_1, \ldots, a_n\}$.

$R_i$ (of arity $k$) is encoded by a string $[R_i]<$ of 0s and 1s of length $n^k$.

$f_i$ is encoded by a string $[f_i]<$ of 0s, 1s and $-$s of length $n^k \log n$.

$$\langle A \rangle < = \underbrace{1 \cdots 1}_{n} \# [R_1]< \# \cdots \# [R_m]< \# [f_1]< \# \cdots \# [f_l]<$$

The exact string obtained depends on the choice of order.
Naïve Algorithm

The straightforward algorithm proceeds recursively on the structure of $\varphi$:

- Atomic formulas by direct lookup.
- Boolean connectives are easy.
- If $\varphi \equiv \exists x \psi$ then for each $a \in A$ check whether
  
  $$(A, c \mapsto a) \models \psi[c/x],$$

  where $c$ is a new constant symbol.

This runs in time $O(\ln^m)$ and $O(m \log n)$ space, where $m$ is the nesting depth of quantifiers in $\varphi$.

$$\text{Mod}(\varphi) = \{A \mid A \models \varphi\}$$

is in logarithmic space and polynomial time.
Second-Order Logic

There are computationally easy properties that are not definable in first-order logic.

- There is no sentence $\varphi$ of first-order logic such that $A \models \varphi$ if, and only if, $|A|$ is even.

- There is no formula $\varphi(E, x, y)$ that defines the transitive closure of a binary relation $E$.

Consider second-order logic, extending first-order logic with relational quantifiers

$\exists X \varphi$
Examples

Evenness

This formula is true in a structure if, and only if, the size of the domain is even.

\[ \exists B \exists S \quad \forall x \exists y B(x, y) \land \forall x \forall y \forall z B(x, y) \land B(x, z) \rightarrow y = z \]
\[ \forall x \forall y \forall z B(x, z) \land B(y, z) \rightarrow x = y \]
\[ \forall x \forall y S(x) \land B(x, y) \rightarrow \neg S(y) \]
\[ \forall x \forall y \neg S(x) \land B(x, y) \rightarrow S(y) \]
Examples

Transitive Closure

This formula is true of a pair of elements $a, b$ in a structure if, and only if, there is an $E$-path from $a$ to $b$.

\[
\exists P \forall x \forall y \ P(x, y) \rightarrow E(x, y)
\]

\[
\exists x P(a, x) \land \exists x P(x, b) \land \neg \exists x P(x, a) \land \neg \exists x P(b, x)
\]

\[
\forall x \forall y (P(x, y) \rightarrow \forall z (P(x, z) \rightarrow y = z))
\]

\[
\forall x \forall y (P(x, y) \rightarrow \forall z (P(z, x) \rightarrow y = z))
\]

\[
\forall x ((x \neq a \land \exists y P(x, y)) \rightarrow \exists z P(z, x))
\]

\[
\forall x ((x \neq b \land \exists y P(y, x)) \rightarrow \exists z P(x, z))
\]
Examples

3-Colourability

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

\[
\exists R \exists B \exists G \ \forall x (Rx \lor Bx \lor Gx) \land \\
\forall x (\neg (Rx \land Bx) \land \neg (Bx \land Gx) \land \neg (Rx \land Gx)) \land \\
\forall x \forall y (Exy \rightarrow (\neg (Rx \land Ry) \land \\
\neg (Bx \land By) \land \\
\neg (Gx \land Gy)))
\]
Fagin’s Theorem

**Theorem (Fagin)**

A class $\mathcal{C}$ of finite structures is definable by a sentence of *existential second-order logic* if, and only if, it is decidable by a *nondeterministic machine* running in polynomial time.

$$\text{ESO} = \text{NP}$$

One direction is easy: Given $A$ and $\exists P_1 \ldots \exists P_m \varphi$.

a nondeterministic machine can guess an interpretation for $P_1, \ldots, P_m$

and then verify $\varphi$. 
Fagin’s Theorem

Given a machine $M$ and an integer $k$, there is an ESO sentence $\varphi$ such that $\mathbb{A} \models \varphi$ if, and only if, $M$ accepts $[\mathbb{A}]_<$, for some order $<$ in $n^k$ steps.

$$\exists < \ \exists \text{State} \ \exists \text{Head} \ \exists \text{Tape}$$

$<$ is a linear order $\wedge$

$\wedge \text{State}(t+1, s_1) \rightarrow \text{State}(t, s) \lor \ldots$

$\wedge \text{State}(t+1, s_2) \rightarrow \ldots$ \hspace{1cm} encoding

$\wedge \text{Tape}(t+1, p) \leftrightarrow \text{Head}(t, p) \ldots$ \hspace{1cm} transitions

$\wedge \text{Head}(t+1, h+1) \leftrightarrow \ldots$ \hspace{1cm} of $M$

$\wedge \text{Head}(t+1, h-1) \leftrightarrow \ldots$

$\wedge$ at time $0$ the tape contains a description of $\mathbb{A}$

$\wedge \text{State}(\text{max}, s)$ for some accepting $s$
Fagin’s Theorem

State, Tape and Head are $2^k$-ary relations, that use the lexicographic order on $k$-tuples.

To state that Tape encodes the input structure:

$$\forall x \ x < n \rightarrow \text{Tape}(0, x)$$
$$x < n^a \rightarrow (\text{Tape}(0, x + n) \leftrightarrow R_1(x|_a))$$
$$\ldots$$

where,

$$x < n^a : \bigwedge_{i \leq (k-a)} x_i = 0$$
Is there a logic for P?

The major open question in *Descriptive Complexity* (first asked by Chandra and Harel in 1982) is whether there is a logic $\mathcal{L}$ such that

for any class of finite structures $\mathcal{C}$, $\mathcal{C}$ is definable by a sentence of $\mathcal{L}$ if, and only if, $\mathcal{C}$ is decidable by a deterministic machine running in polynomial time.

Formally, we require $\mathcal{L}$ to be a *recursively enumerable* set of sentences, with a computable map taking each sentence to a Turing machine $M$ and a polynomial time bound $p$ such that $(M, p)$ accepts a *class of structures*.

(Gurevich 1988)
Enumerating Queries

For a given structure $\mathcal{A}$ with $n$ elements, there may be as many as $n!$ distinct strings $[\mathcal{A}]_<$ encoding it.

Given $(M_0, p_0), \ldots, (M_i, p_i), \ldots$—an enumeration of polynomially-clocked Turing machines.

Can we enumerate a subsequence of those that compute graph properties, i.e. are encoding invariant, while including all such properties?
Inductive Definitions

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

Associate an operator $\Phi$ on a given structure $\mathbb{A}$:

$$\Phi(R^\mathbb{A}) = \{a \mid (\mathbb{A}, R^\mathbb{A}, a) \models \varphi(R, x)\}$$

We define the *increasing* sequence of relations on $\mathbb{A}$:

$$\Phi^0 = \emptyset$$

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

The *inflationary fixed point* of $\Phi$ is the limit of this sequence.

On a structure with $n$ elements, the limit is reached after at most $n^k$ stages.
The logic **IFP** is formed by closing first-order logic under the rule:

If $\varphi$ is a formula of vocabulary $\sigma \cup \{R\}$ then $[\text{ifp}_{R,t} \varphi](t)$ is a formula of vocabulary $\sigma$.

The formula is read as:

the tuple $t$ is in the inflationary fixed point of the operator defined by $\varphi$

**LFP** is the similar logic obtained using *least fixed points* of *monotone* operators defined by *positive* formulas.

**LFP** and **IFP** have the same expressive power (*Gurevich-Shelah; Kreutzer*).
Transitive Closure

The formula

$$\begin{align*}
\text{ift}_{T,xy}(x = y \lor \exists z (E(x, z) \land T(z, y)))\end{align*}$$

defines the *transitive closure* of the relation $E$

The expressive power of IFP properly extends that of first-order logic.

On structures which come equipped with a linear order IFP expresses exactly the properties that are in PTime.

*(Immerman; Vardi)*
Immerman-Vardi Theorem

\[ \exists < \exists \text{State} \exists \text{Head} \exists \text{Tape} \]

\[ < \text{ is a linear order } \wedge \]

\[ \begin{array}{l}
\text{State}(t + 1, s_1) \rightarrow \text{State}(t, s) \lor \ldots \\
\wedge \text{State}(t + 1, s_2) \rightarrow \ldots \\
\wedge \text{Tape}(t + 1, p) \leftrightarrow \text{Head}(t, p) \ldots \\
\wedge \text{Head}(t + 1, h + 1) \leftrightarrow \ldots \\
\wedge \text{Head}(t + 1, h - 1) \leftrightarrow \ldots \\
\end{array} \]

\[ \wedge \text{at time } 0 \text{ the tape contains a description of } A \]

\[ \wedge \text{State}(\text{max}, s) \text{ for some accepting } s \]

With a deterministic machine, the relations State, Tape and Head can be define \emph{inductively}. 

Anuj Dawar July 2008
IFP vs. Ptime

The order cannot be built up inductively.

It is an open question whether a canonical string representation of a structure can be constructed in polynomial-time.

If it can, there is a logic for PTime.
If not, then PTime $\not\subset$ NP.

All PTime classes of structures can be expressed by a sentence of IFP with $<$, which is invariant under the choice of order. The set of all such sentences is not r.e.

IFP by itself is too weak to express all properties in PTime.

Evenness is not definable in IFP.
Finite Variable Logic

We write $L^k$ for the first order formulas using only the variables $x_1, \ldots, x_k$.

$$(A, a) \equiv^k (B, b)$$

denotes that there is no formula $\varphi$ of $L^k$ such that $A \models \varphi[a]$ and $B \not\models \varphi[b]$

If $\varphi(R, x)$ has $k$ variables all together, then each of the relations in the sequence:

$$\Phi^0 = \emptyset; \Phi^{m+1} = \Phi^m \cup \Phi(\Phi^m)$$

is definable in $L^{2k}$.

Proof by induction, using *substitution* and *renaming* of bound variables.
Pebble Game

The $k$-pebble game is played on two structures $\mathbb{A}$ and $\mathbb{B}$, by two players—*Spoiler* and *Duplicator*—using $k$ pairs of pebbles $\{(a_1, b_1), \ldots, (a_k, b_k)\}$.

*Spoiler* moves by picking a pebble and placing it on an element ($a_i$ on an element of $\mathbb{A}$ or $b_i$ on an element of $\mathbb{B}$).

*Duplicator* responds by picking the matching pebble and placing it on an element of the other structure.

*Spoiler* wins at any stage if the partial map from $\mathbb{A}$ to $\mathbb{B}$ defined by the pebble pairs is not a partial isomorphism.

If *Duplicator* has a winning strategy for $q$ moves, then $\mathbb{A}$ and $\mathbb{B}$ agree on all sentences of $L^k$ of quantifier rank at most $q$. 

(Barwise) $\mathbb{A} \equiv^k \mathbb{B}$ if, for every $q$, *Duplicator* wins the $q$ round, $k$ pebble game on $\mathbb{A}$ and $\mathbb{B}$.

Equivalently (on finite structures) *Duplicator* has a strategy to play forever.
Evenness

To show that *Evenness* is not definable in IFP, it suffices to show that:

for every $k$, there are structures $A_k$ and $B_k$ such that $A_k$ has an even number of elements, $B_k$ has an odd number of elements and

$$A \equiv^k B.$$  

It is easily seen that *Duplicator* has a strategy to play forever when one structure is a set containing $k$ elements (and no other relations) and the other structure has $k + 1$ elements.
Immerman proposed $\text{IFP} + \mathbb{C}$—the extension of IFP with a mechanism for counting.

Two sorts of variables:

- $x_1, x_2, \ldots$ range over $|A|$—the domain of the structure;
- $\nu_1, \nu_2, \ldots$ which range over numbers in the range $0, \ldots, |A|$.

If $\varphi(x)$ is a formula with free variable $x$, then $\nu = \#x \varphi$ denotes that $\nu$ is the number of elements of $A$ that satisfy the formula $\varphi$.

We also have the order $\nu_1 < \nu_2$, which allows us (using recursion) to define arithmetic operations.
Counting Quantifiers

$C^k$ is the logic obtained from first-order logic by allowing:

- allowing counting quantifiers: $\exists^i x \varphi$; and
- only the variables $x_1, \ldots, x_k$.

Every formula of $C^k$ is equivalent to a formula of first-order logic, albeit one with more variables.

For every sentence $\varphi$ of IFP + C, there is a $k$ such that if $A \equiv^{C^k} B$, then

$$A \models \varphi \text{ if, and only if, } B \models \varphi.$$
Cai-Fürer-Immerman Graphs

There are polynomial-time decidable properties of graphs that are not definable in IFP + C. \(^{(Cai, Fürer, Immerman, 1992)}\)

More precisely, we can construct a sequence of pairs of graphs \(G_k, H_k (k \in \omega)\) such that:

- \(G_k \equiv^{C^k} H_k\) for all \(k\).
- There is a polynomial time decidable class of graphs that includes all \(G_k\) and excludes all \(H_k\).

Still, IFP + C is a \textit{natural} level of expressiveness within PTime.
Constructing $G_k$ and $H_k$

Given any graph $G$, we can define a graph $X_G$ by replacing every edge with a pair of edges, and every vertex with a gadget.

The picture shows the gadget for a vertex $v$ that is adjacent in $G$ to vertices $w_1$, $w_2$, and $w_3$. The vertex $v^S$ is adjacent to $a_{vw_i}(i \in S)$ and $b_{vw_i}(i \not\in S)$ and there is one vertex for all even size $S$. The graph $\tilde{X}_G$ is like $X_G$ except that at one vertex $v$, we include $V^S$ for odd size $S$. 
Properties

If $G$ is connected and has treewidth at least $k$, then:

1. $X_G \neq \tilde{X}_G$; and

2. $X_G \equiv^{C^k} \tilde{X}_G$.

(1) allows us to construct a polynomial time property separating $X_G$ and $\tilde{X}_G$.

(2) is proved by a game argument.

The original proof of (Cai, F"urer, Immerman) relied on the existence of balanced separators in $G$. The characterisation in terms of treewidth is from (D., Richerby 07).
Bijection Games

\[ \equiv^{C^k} \] is characterised by a \( k \)-pebble bijection game. \( \text{(Hella 96)} \).

The game is played on structures \( \mathbb{A} \) and \( \mathbb{B} \) with pebbles \( a_1, \ldots, a_k \) on \( \mathbb{A} \) and \( b_1, \ldots, b_k \) on \( \mathbb{B} \).

- **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, \( A \equiv^{C^k} B \).
TreeWidth

The \textit{treewidth} of a graph is a measure of how tree-like the graph is.

A graph has treewidth $k$ if it can be covered by subgraphs of at most $k + 1$ nodes in a tree-like fashion.
TreeWidth

Formal Definition:

For a graph \( G = (V, E) \), a tree decomposition of \( G \) is a relation \( D \subseteq V \times T \) with a tree \( T \) such that:

- for each \( v \in V \), the set \( \{ t \mid (v, t) \in D \} \) forms a connected subtree of \( T \); and

- for each edge \( (u, v) \in E \), there is a \( t \in T \) such that \( (u, t), (v, t) \in D \).

The treewidth of \( G \) is the least \( k \) such that there is a tree \( T \) and a tree-decomposition \( D \subseteq V \times T \) such that for each \( t \in T \),

\[
|\{ v \in V \mid (v, t) \in D \}| \leq k + 1.
\]
Cops and Robbers

A game played on an undirected graph $G = (V, E)$ between a player controlling $k$ cops and another player in charge of a robber.

At any point, the cops are sitting on a set $X \subseteq V$ of the nodes and the robber on a node $r \in V$.

A move consists in the cop player removing some cops from $X' \subseteq X$ nodes and announcing a new position $Y$ for them. The robber responds by moving along a path from $r$ to some node $s$ such that the path does not go through $X \setminus X'$.

The new position is $(X \setminus X') \cup Y$ and $s$. If a cop and the robber are on the same node, the robber is caught and the game ends.
Strategies and Decompositions

Theorem (Seymour and Thomas 93):
There is a winning strategy for the cop player with $k$ cops on a graph $G$ if, and only if, the tree-width of $G$ is at most $k - 1$.

It is not difficult to construct, from a tree decomposition of width $k$, a winning strategy for $k + 1$ cops.

Somewhat more involved to show that a winning strategy yields a decomposition.
Cops, Robbers and Bijections

If $G$ has treewidth $k$ or more, than the robber has a winning strategy in the $k$-cops and robbers game played on $G$.

We use this to construct a winning strategy for Duplicator in the $k$-pebble bijection game on $X_G$ and $\tilde{X}_G$.

- A bijection $h : X_G \rightarrow \tilde{X}_G$ is good bar $v$ if it is an isomorphism everywhere except at the vertices $v^S$.
- If $h$ is good bar $v$ and there is a path from $v$ to $u$, then there is a bijection $h'$ that is good bar $u$ such that $h$ and $h'$ differ only at vertices corresponding to the path from $v$ to $u$.
- Duplicator plays bijections that are good bar $v$, where $v$ is the robber position in $G$ when the cop position is given by the currently pebbled elements.
Restricted Graph Classes

If we restrict the class of structures we consider, IFP + C may be powerful enough to express all polynomial-time decidable properties.

- IFP + C captures PTime on any class of graphs of bounded treewidth. (Grohe and Mariño 1999).
- IFP + C captures PTime on the class of planar graphs. (Grohe 1998).

In each case, the proof proceeds by showing that for any $G$ in the class, a canonical, ordered representation of $G$ can be interpreted in $G$ using IFP + C.
Graph Minors

We say that a graph $G$ is a minor of graph $H$ (written $G \prec H$) if $G$ can be obtained from $H$ by repeated applications of the operations:

- *delete an edge*;

- *delete an vertex* (and all incident edges); and

- *contract an edge*
Graph Minors

Alternatively, $G = (V, E)$ is a minor of $H = (U, F)$, if there is a graph $H' = (U', F')$ with $U' \subseteq U$ and $F' \subseteq F$ and a surjective map $M : U' \to V$ such that

- for each $v \in V$, $M^{-1}(v)$ is a connected subgraph of $H'$; and
- for each edge $(u, v) \in E$, there is an edge in $F'$ between some $x \in M^{-1}(u)$ and some $y \in M^{-1}(v)$.

![Graph Minors Diagram](image-url)
Facts about Graph Minors

- \( G \) is planar if, and only if, \( K_5 \not\preceq G \) and \( K_{3,3} \not\preceq G \).
- If \( G \preceq H \) then \( G \prec H \).
- The relation \( \prec \) is transitive.
- If \( G \prec H \), then \( \text{tw}(G) \leq \text{tw}(H) \).
- If \( \text{tw}(G) < k - 1 \), then \( K_k \not\preceq G \).

Say that a class of structures \( C \) excludes \( H \) as a minor if \( H \not\preceq GA \) for all \( A \in C \).

\( C \) has excluded minors if it excludes some \( H \) as a minor (equivalently, it excludes some \( K_k \) as a minor).
- \( T_k \) excludes \( K_{k+2} \) as a minor.
More Facts about Graph Minors

**Theorem (Robertson-Seymour)**
In any infinite collection \( \{G_i \mid i \in \omega\} \) of graphs, there are \( i, j \) with \( G_i \prec G_j \).

**Corollary**
For any class \( \mathcal{C} \) **closed under minors**, there is a finite collection \( \mathcal{F} \) of graphs such that \( G \in \mathcal{C} \) if, and only if, \( F \not\prec G \) for all \( F \in \mathcal{F} \).

**Theorem (Robertson-Seymour)**
For any \( G \) there is an \( O(n^3) \) algorithm for deciding, given \( H \), whether \( G \prec H \).

**Corollary**
Any class \( \mathcal{C} \) closed under minors is decidable in **cubic time**.
Ptime on Minor-Closed Classes

**Conjecture**  
(Grohe)  

IFP + C captures PTime on every proper minor-closed class of graphs.

**Theorem**  
(Grohe 2008)  

IFP + C captures PTime on the class of graphs that exclude $K_5$ as a minor.

The Cai-Fürer-Immerman construction cannot be used to refute Grohe’s conjecture.

If $C$—a class of graphs contains $X_G$ and $\tilde{X}_G$ for graphs $G$ of unbounded treewidth, then $C$ does not exclude any graph as a minor.

(D., Richerby 2007)
Logics with Choice

Extending IFP with a choice operator allows us to define all polynomial-time decidable classes.

This is akin to adding order to the logic. It also allows sentences whose interpretation is dependent on choices, and therefore not determined by the structure alone.

The sentences that are invariant under choices express all polynomial-time properties, but do not form an r.e. set.

Gire and Hoang considered a method of restricting the choice operator to ensure that the interpretation of sentences was invariant.
Non-deterministic Choice

Given two formulas $\varphi(R, X, x); \psi(R, X, y)$ and a structure $\mathcal{A}$, we define the following sequence of pairs of relations.

$$\Phi^0 = \emptyset \quad \Psi^0 = \emptyset;$$

$$\Phi^{i+1} = \Phi^i \cup \varphi^\mathcal{A}(\Phi^i / R, \Psi^i / X);$$

$$\Psi^{i+1} = \{a\} \text{ for some } a \text{ such that } \mathcal{A} \models \psi(\Phi^i / R, \Psi^i / X)[a].$$

The sequence $\Phi^i$ converges to a limit.

We say that the pair of formulas $\varphi; \psi$ is choice-invariant if the limit does not depend on the choice of the sequence $\Psi^i$.

The collection of choice-invariant formulas captures $\text{PTime}$, but is not an $r.e.$ set.
Symmetric Choice

Alter the definition of the sequence so that:

\[
\begin{align*}
\Phi^0 &= \emptyset \quad \Psi^0 = \emptyset; \\
\Phi^{i+1} &= \Phi^i \cup \varphi^A(\Phi^i / R, \Psi^i / X); \\
\Psi^{i+1} &= \begin{cases} \\
\{a\} & \text{for } A \models \psi(\Phi^i / R, \Psi^i / X)[a] \\
\emptyset & \text{otherwise} \\
\end{cases}
\end{align*}
\]

Now, the limit of the sequence \( \Phi^i \) is independent of the choices.

However, it is not clear that a pair of formulas can be evaluated in \textit{polynomial time}.

The semantics involves an \textit{automorphism test}. 
Specified Symmetric Choice

The logic of *specified symmetric choice* (SSC-IFP) defines fixed points for triples \( \varphi; \psi; \theta \) of formulas.

\[ \Psi^{i+1} = \{ a \} \] only if \( \psi(\Phi^i/R, \Psi^i/X) \) defines an automorphism class and this is witnessed by \( \theta \) (i.e. this formula defines, for each pair of tuples satisfying \( \psi \), an automorphism mapping one to the other).

Any formula of SSC-IFP can be evaluated in polynomial time.

The Cai-Fürer-Immerman property can be expressed in SSC-IFP.

*(Gire-Hoang 98)*

**Open Question:** Is there a polynomial-time decidable property that cannot be expressed in SSC-IFP?
Choiceless Polynomial Time

*Choiceless Polynomial Time* (ĈPT) is a class of computational problems defined by Blass, Gurevich and Shelah.

It is based on a *machine model (Gurevich Abstract State Machines)* that works directly on a relational structure (rather than on a string representation).

The machine can access the collection of hereditarily finite sets over the universe of the structure.

ĈPT is the polynomial time and space restriction of the machines.

ĈPT is strictly more expressive than IFP, but still cannot express counting properties.

Consider ĈPT(Card)—the extension of ĈPT with counting.

Does it express all properties in PTime?
Choiceless Polynomial Time

\[ \tilde{\text{CPT}} \text{ can express the property of } \text{Cai, F"urer and Immerman.} \]

Any program of \( \tilde{\text{CPT}}(\text{Card}) \) that expresses the CFI property must use sets of \textit{unbounded rank}.

\[ \text{IFP} + C \text{ can be translated to programs of } \tilde{\text{CPT}}(\text{Card}) \text{ of bounded rank.} \]

\textit{(D., Richerby and Rossman 2008)}
Ongoing Research

Is there a $\text{PTime}$ decidable class that cannot be expressed in $\text{SSC-IFP}$?

Is there $\text{PTime}$ decidable class that is not in $\tilde{\text{CPT}}(\text{Card})$?

How does the expressive power of $\text{SSC-IFP}$ compare with that of $\tilde{\text{CPT}}(\text{Card})$?

Are there other natural extensions of $\text{IFP}$ that might capture $\text{PTime}$?