

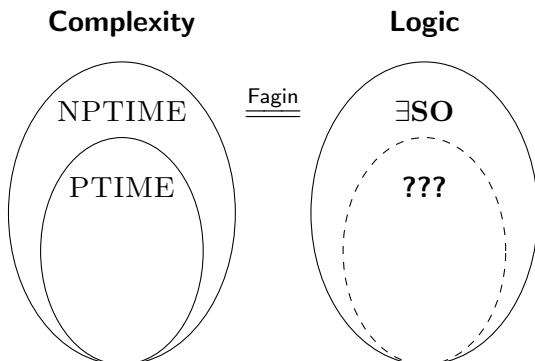
# Game comonads & generalised quantifiers

Adam Ó Conghaile  
joint with *Anuj Dawar*

March 2020

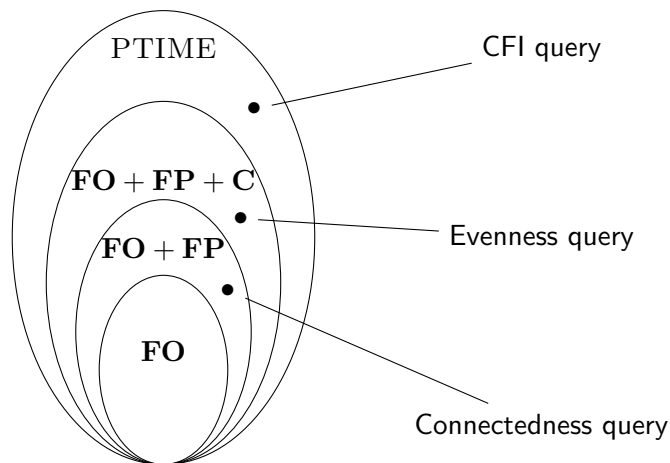
- Quantifiers & descriptive complexity
- Generalised quantifiers & Hella's game
- Game comonads & (co)resources
- Quantifier arity as a resource: a comonad for Hella's game
- Coalgebras and generalised tree-decomposition
- Future work

# The search for a logic for PTIME



- Since Fagin showed that  $\exists\text{SO}$  captures  $\text{NPTIME}$ , finite model theorists have tried to find a logic that *captures*  $\text{PTIME}$ .
- It was noticed early on that  $\text{FO}$  is not enough (e.g. graph connectedness is not in  $\text{FO}$ )
- To gain more power we need to add new types of computation to the logic. This can be done through quantifiers

# Computation and quantifiers



# Computation and quantifiers

## Fixed points

### Inflationary fix-point quantifiers

For  $\mathcal{A}$  a  $\tau$ -structure,  $\mathbf{a}, \mathbf{b}$  tuples of elements of  $\mathcal{A}$  of lengths  $n$  and  $m$  and some logic  $\mathcal{L}$ , let  $X$  be a second-order variable standing for an  $m$ -ary and  $\mathbf{x}, \mathbf{y}$  be a tuples of  $m$  and  $n$  first-order variables, respectively then any formula  $\phi(X, \mathbf{x}, \mathbf{y})$  in  $\mathcal{L}[\tau \cup \{X\}]$  defines a sequence

$$X_0^{\mathbf{a}} = \emptyset; \quad \dots \quad X_{i+1}^{\mathbf{a}} = \{\mathbf{b} \mid \mathcal{A}, \mathbf{a}, \mathbf{b} \models \phi(X_i^{\mathbf{a}}, \mathbf{x}, \mathbf{y})\}$$

and if  $\mathbf{b} \in \bigcup_i X_i^{\mathbf{a}}$  then we write

$$\mathcal{A}, \mathbf{a}, \mathbf{b} \models \text{ifp}_{\mathbf{x}, \mathbf{X}}(\phi(X, \mathbf{x}, \mathbf{y}))[\mathbf{z}]$$

### Counting quantifiers

For  $\mathcal{A}$  a  $\tau$ -structure,  $\mathbf{a}$  a tuple of elements of  $\mathcal{A}$  of lengths  $n$  and some logic  $\mathcal{L}$ , any formula  $\phi(\mathbf{x}, \mathbf{y})$  in  $\mathcal{L}[\tau]$  with  $n + m$  free variables defines a set

$$S_{\phi}^{\mathbf{a}} = \{\mathbf{b} \mid \mathcal{A}, \mathbf{a}, \mathbf{b} \models \phi(\mathbf{x}, \mathbf{y})\}$$

and for any  $m \in \mathbb{N}$ , if  $|S_{\phi}^{\mathbf{a}}| \geq m$  we write

$$\mathcal{A}, \mathbf{a} \models \exists^{\geq m} \mathbf{y}. \phi(\mathbf{x}, \mathbf{y})$$

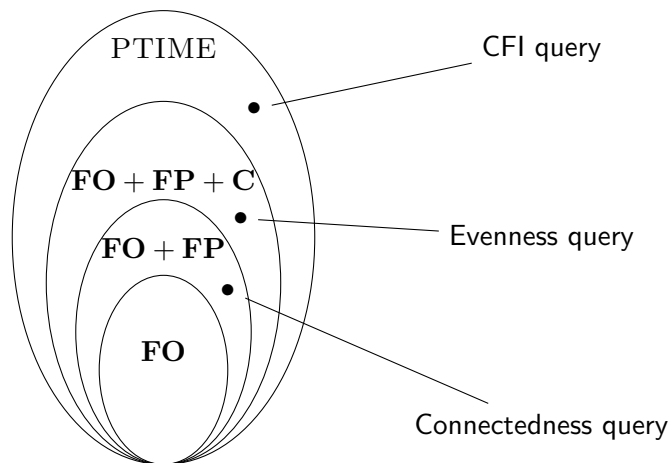
# Games for judging the strength of quantifiers

Spoiler-duplicator games are used to test the strength of new quantifiers we may add to **FO**. This is usually done by containing the logic **FO + Q** in some logic which is graded by some syntactic resource such as variables or quantifier depth.

Table below shows logics and each of the gradings and games use. Each of these can witness an example showing the **FO + Q** in question does not capture PTIME

Logic		Hierarchy	Corresponding Game
<b>FO</b>	=	$\bigcup_n \mathcal{L}_n$	$n$ -round E-F game
<b>FO + FP</b>	$\subset$	$\bigcup_k \mathcal{L}_{\infty\omega}^k$	$k$ -pebble game
<b>FO + FP + C</b>	$\subset$	$\bigcup_k \mathcal{C}_{\infty\omega}^k$	$k$ -pebble bijection game

# Computation and quantifiers





# Generalised quantifiers

So quantifiers can add expressive power to a logic in a similar way to how oracles add computational power to some model of computation.

Given a query  $q$  (i.e. an isomorphism-closed class of structures) which can't be expressed in **FO** how do we add  $q$  to **FO**

This is the idea behind generalised (or Lindström) quantifiers.

## Lindström generalised quantifiers (Syntax)

Given a logic  $\mathcal{L}[\sigma]$  and an isomorphism-closed class  $q$  of  $\{R_1, \dots, R_m\}$ -structures the logic  $\mathcal{L}(\mathcal{Q}_q)[\sigma]$  is the smallest extension  $\mathcal{L}'$  of  $\mathcal{L}[\sigma]$  closed under the normal syntactic rules for  $\mathcal{L}$  and the following construction: given  $\{\psi_i\}_{1 \leq i \leq m} \in \mathcal{L}'$  with  $\{\mathbf{x}_i\}_{1 \leq i \leq m}$  tuples of free variables of length  $n_i = \text{arity}(R_i)$  among the free variables of  $\psi_i$  then

$$\mathcal{Q}_q \mathbf{x}_1 \dots \mathbf{x}_m (\psi_1, \dots, \psi_m) \in \mathcal{L}'$$

Where the variables  $\mathbf{x}_1 \dots \mathbf{x}_m$  are bound in this formula

## Lindström generalised quantifiers (Semantics)

For a  $\sigma$ -structure  $\mathcal{A}$  and formulas  $\psi_i(\mathbf{x}_i, \mathbf{y}_i)$ , define the sets

$$\psi_i^{\mathcal{A}}(\cdot, \mathbf{b}_i) := \{\mathbf{a} \mid \mathcal{A}, \mathbf{a}, \mathbf{b}_i \models \psi_i(\cdot, \mathbf{y}_i)\} \subset A^{n_i}$$

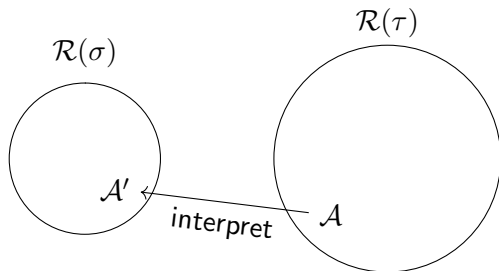
then for  $q, \tau$  and  $\psi_i$  as above we say

$$\mathcal{A} \mathbf{b}_1, \dots, \mathbf{b}_m \models \mathcal{Q}_q \mathbf{x}_1 \dots \mathbf{x}_m (\psi_1, \dots, \psi_m) [\mathbf{y}_1, \dots, \mathbf{y}_m]$$

if and only if

$$\langle \mathcal{A}, \psi_1^{\mathcal{A}}(\cdot, \mathbf{b}_1), \dots, \psi_m^{\mathcal{A}}(\cdot, \mathbf{b}_m) \rangle \in q$$

# Generalised quantifiers



# Generalised quantifiers

These can be weird

## Henkin quantifiers

$$\mathcal{A} \models \left\{ \begin{array}{ll} \forall x & \exists y \\ \forall u & \exists v \end{array} \right\} R(x, y, u, v) \iff \exists f, g : A \rightarrow A \text{ s.t. } \forall x, u \in A \\ (x, f(x), u, g(u)) \in R^{\mathcal{A}}$$

And very strong

## 3-COL quantifier

$q = 3\text{COL}$  or worse  $q$  could be undecidable!

but they are everything when it comes to the search for a logic for PTIME

## Dawar 1995

If there is a logic (in the sense of Gurevich 1988) which captures PTIME then there is such a logic of the form **FO** + **FP** + **Q** where **Q** is a collection of generalised quantifiers.

## Definition

For a Boolean query  $q$  on  $\sigma$ -structures, the arity of the quantifier  $Q_q$  is the maximum arity of the relations of  $\sigma$

Let  $Q_n$  denote the collection of all Lindström quantifiers of arity  $\leq n$  and  $\mathcal{L}_{\infty\omega}^k(Q_n)$  denote  $k$  variable infinitary first order logic extended by all quantifiers in  $Q_n$

Extending  $\mathcal{L}_{\infty\omega}^k$  by all arity  $n$  quantifiers can be seen as adding oracles for deciding all queries over signatures with arity  $\leq n$ .

The arity of generalised quantifiers gives us a new way of grading the strength of some extension of **FO**. In particular,  $\bigcup_n \bigcup_k \mathcal{L}_{\infty\omega}^k(Q_n)$  is a grading of **FO** extended with **all** Lindström quantifiers.

# A game for generalised quantifiers

Hella introduced a game to test the expressive power given by this new resource.

## $k$ -pebble $n$ -bijective game between $\mathcal{A}$ and $\mathcal{B}$

With game configurations as in the  $k$  pebble game, at the start of each round

- Duplicator provides a bijection  $f : A \rightarrow B$
- Spoiler moves pebbles  $i_1, \dots, i_n$  to  $a_1, \dots, a_n$
- Duplicator's pebbles  $i_1, \dots, i_n$  move to  $f(a_1) \dots f(a_n)$
- If the subset of  $A \times B$  given by the span  $A \leftarrow [k] \rightarrow B$  is not a partial homomorphism Spoiler wins.

## Theorem (Hella 1996)

The following are equivalent:

- Duplicator has a winning strategy for the  $k$ -pebble  $n$ -bijjective game between  $\mathcal{A}$  and  $\mathcal{B}$
- $\mathcal{A} \equiv \mathcal{B}$  (over  $\mathcal{L}_{\infty\omega}^k(Q_n)$ )
- $\mathcal{A} \equiv \mathcal{B}$  (over  $\mathbf{FO}^k(Q_n)$ )



Logic		Hierarchy	Corresponding Game
<b>FO</b>	=	$\bigcup_n \mathcal{L}_n$	$n$ -round E-F game
<b>FO + FP</b>	$\subset$	$\bigcup_k \mathcal{L}_{\infty\omega}^k$	$k$ -pebble game
<b>FO + FP + C</b>	$\subset$	$\left\{ \begin{array}{l} \bigcup_k \mathcal{C}_{\infty\omega}^k \\ \bigcup_k \mathcal{L}_{\infty\omega}^k(Q_1) \end{array} \right.$	$k$ -pebble bijection game
<b>FO + ALL</b>	$\subset$	$\bigcup_n \bigcup_k \mathcal{L}_{\infty\omega}^k(Q_n)$	$n$ -bijjective $k$ -pebble game

# Game comonads, recap

Game comonads give us a categorical semantics relating the gradings of syntactic structure studied by finite model theorists to the power of the query languages they represent.

Grading	Game	Comonad
$\mathbf{FO} = \bigcup_n \mathcal{L}_n$	$n$ round EF game	$\mathbb{E}_n$
$\mathbf{FO} + \mathbf{FP} \subset \bigcup_n \mathcal{L}_{\infty\omega}^k$	$k$ pebble game	$\mathbb{P}_k$
$\mathbf{ML} = \bigcup_n \mathbf{ML}_n$	$n$ round bisimulation game	$\mathbb{M}_n$

In our work we wanted to relate the two tiered hierarchy used by Hella to this language of game semantics

## Aim: Arity as a resource

Find a graded comonad for which the Kleisli category captures entailment and equivalence of structures in (fragments of)  $\mathcal{L}_{\infty\omega}^k(Q_n)$  in the same way that previous examples do for other resources.

But this had some initial challenges

## Challenges

- Multiple pebble moves per round
- Duplicator has new restriction in giving a bijection
- Hella's game generalises the bijection game not the pebble game. What's the approximation of homomorphism?

# Relaxing Hella's game

We chose a game that resembled Hella's game, except in every round Duplicator gives a *function*  $f : A \rightarrow B$  instead of a bijection.

## $k$ pebble $n$ -function game

With game configurations as in the  $k$  pebble game, at the start of each round

- Duplicator provides a bijection  $f : A \rightarrow B$
- Spoiler moves pebbles  $i_1, \dots, i_n$  to  $a_1, \dots, a_n$
- Duplicator's pebbles  $i_1, \dots, i_n$  move to  $f(a_1) \dots f(a_n)$
- If the subset of  $A \times B$  given by the span  $A \leftarrow [k] \rightarrow B$  is not a partial homomorphism Spoiler wins.

We prove that this game corresponds to an important fragment of  $\mathcal{L}_{\infty\omega}^k(Q_n)$  namely  $k$  variable infinitary logic extended with the arity  $n$  generalised quantifiers for **homomorphism-closed** queries,  $Q_n^H$

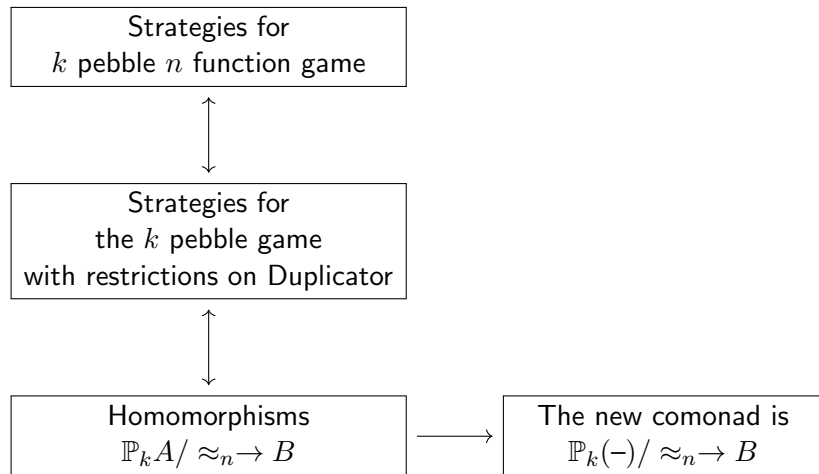
## Theorem 1

TFAE:

- Duplicator wins the  $k$ -pebble  $n$ -function game from  $\mathcal{A}$  to  $\mathcal{B}$
- $\mathcal{A} \equiv \mathcal{B}$  (over  $\exists + \mathcal{L}_{\infty\omega}^k(Q_n^H)$ ) i.e.

$$\forall \phi \in \mathcal{L}_{\infty\omega}^k(Q_n^H) \mathcal{A} \models \phi \implies \mathcal{B} \models \phi$$

# Comonadifying the $n$ function game



# Step 1: Moving to the $k$ pebble game

## Observation

Duplicator strategies for  $k$ -pebble  $n$ -function game can be seen as a restricted set of strategies for the  $k$  pebble game

Indeed, a Duplicator with a strategy for the  $k$ -pebble  $n$ -function game would play the  $k$ -pebble game as follows:

- At the end of turn  $mn$ , (for  $m \geq 0$ ), Duplicator writes looks at the game configuration (history of Spoiler moves) and writes down the function  $f_m$  they would have played on round  $m$  of the  $n$ -function game
- For rounds  $nm + 1, \dots, n(m + 1)$ , if Spoiler places a pebble on  $a \in A$ , Duplicator responds with  $f_m(a) \in B$
- Repeat.

It is not hard to see the translated strategy is winning of the  $k$ -pebble game if and only if the original strategy was winning for the  $n$ -function  $k$ -pebble game.



## Step 2: Quotienting $\mathbb{P}_k$

However, it is clear that the translation cannot go the other way. So to identify the homomorphisms which do indeed come from a winning strategy to the  $n$ -function  $k$  pebble game we need to identify Spoiler histories  $s \in \mathbb{P}_k \mathcal{A}$  which will be treated the same by any such strategy. Noting that a choice of  $f_m$  depending on the Spoiler moves up to (and including) Round  $nm$ , determines Duplicators responses in these strategies for rounds  $nm + 1, \dots, n(m + 1)$  we chose the equivalence relation

$\approx_n$

For any  $n \geq 1$  we define  $\approx_n$  to be the equivalence relation on  $\mathbb{P}_k \mathcal{A}$  where if  $s = [(a_1, p_1), \dots, (a_{nm+l}, p_{nm+l})]$  and  $t = [(b_1, q_1), \dots, (b_{nm+r}, q_{nm+r})]$  for some  $m \in \mathbb{N}$  and  $0 < l, r \leq n$  then

$$s \approx_n t \iff a_{nm+l} = b_{nm+r} \text{ and } \forall 1 \leq j \leq nm, a_j = b_j \text{ and } p_j = q_j$$

## Step 2: Quotienting $\mathbb{P}_k$

### Lemma 2

Given a “natural” equivalence relation  $\approx_A$  on  $\mathbb{P}_k A$  then choosing relations on  $\mathbb{P}_k A / \approx$  such that the quotient map reflects relations, we have that if the counit and comultiplication are well-defined w.r.t.  $\approx$  then  $\mathbb{P}_k(-) / \approx$  is a comonad

## Theorem 2 (Power Theorem)

- (i)  $\mathbb{P}_{n,k}\mathcal{A} \rightarrow B \iff \mathcal{A} \rightrightarrows B$  over  $\exists + \mathcal{L}^k(\mathbf{Q}_n^H)$
- (ii)  $\mathcal{A} \leftrightarrow_{\mathcal{K}(\mathbb{P}_{n,k})} B \iff \mathcal{A} \equiv B$  over  $\mathcal{L}^k(\mathbf{Q}_n^H)$
- (iii)  $\mathcal{A} \cong_{\mathcal{K}(\mathbb{P}_{n,k})} B \iff \mathcal{A} \equiv B$  over  $\mathcal{L}_{\infty\omega}^k(\mathcal{Q}_n)$

## Theorem 3 (Structure Theorem)

Coalgebras of  $\mathbb{P}_{n,k}$ ,  $\alpha : \mathcal{A} \rightarrow \mathcal{A} \rightarrow \mathbb{P}_{n,k}\mathcal{A}$  correspond to  $n$ -ary generalised tree-decompositions of  $\mathcal{A}$  with width  $< k$

## Definition (Generalised Treewidth)

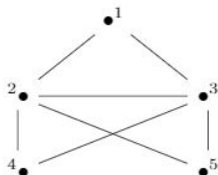
An  $n$ -ary tree-decomposition of a  $\sigma$ -structure  $A$  is defined by  $(T, \beta_1, \beta_2)$  where:

- $T$  is a tree
- $\beta_1, \beta_2 : T \rightarrow 2^A$  identify the *active* and *floating* bags at each node
- $\beta(t) = \beta_1(t) \cup \beta_2(t)$

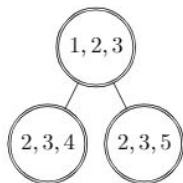
such that:

- 1 For every  $x \in A$ , there is at most one  $t \in T$  s.t.  $x \in \beta_2(t)$ .
- 2 For every  $x \in A$ , the set  $\{t \in T \mid x \in \beta(t)\}$  is a subtree
- 3 For any  $t_1 \prec t_2$  in  $T$  it must be the case that
  - 1  $|\beta_1(t_2) \cap \beta_2(t_1)| \leq r$  and
  - 2  $\beta_2(t_2) \cap \beta(t_1) = \emptyset$
- 4 For any  $(a_1, \dots, a_m) \in R^A$  there is a  $t \in T$  s.t.  $\{a_1, \dots, a_n\} \subset \beta(t)$  and  $|\{a_1, \dots, a_n\} \cap \beta_2(t)| \leq n$

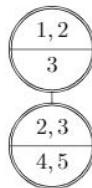
# Generalised tree-decomposition



Example of a graph  $\mathcal{G}$  of treewidth 2



A standard tree decomposition of  $\mathcal{G}$



1-ary tree decomposition of  $\mathcal{G}$

Figure 1: Two types of tree decompositions of a graph

# Conclusions & future work

- We've demonstrated that  $\mathbb{P}_k$  can be generalised to give categorical semantics to games for generalised quantifiers.
- We've come up with new methods of building new game comonads from old ones.
- Next we'd like to do the same for games with more restricted forms of generalised quantifiers e.g. Dawar, Grädel and Pakusa's  $\mathbf{LA}^\omega(Q)$  ( $\mathcal{L}_{\infty\omega}^\omega$  extended with all **linear algebraic** quantifiers over  $\mathbb{F}_p$  for each  $p \in Q$ )