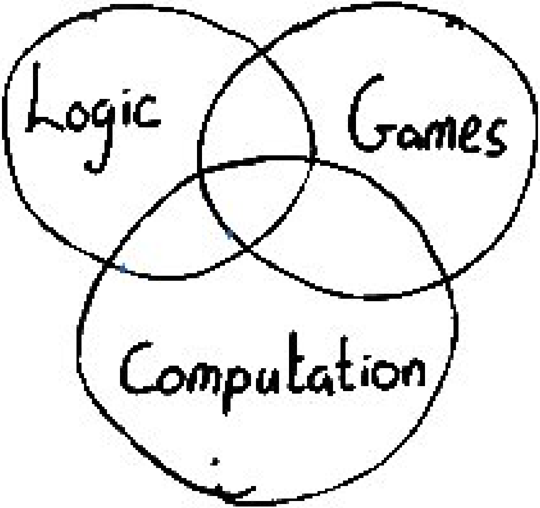
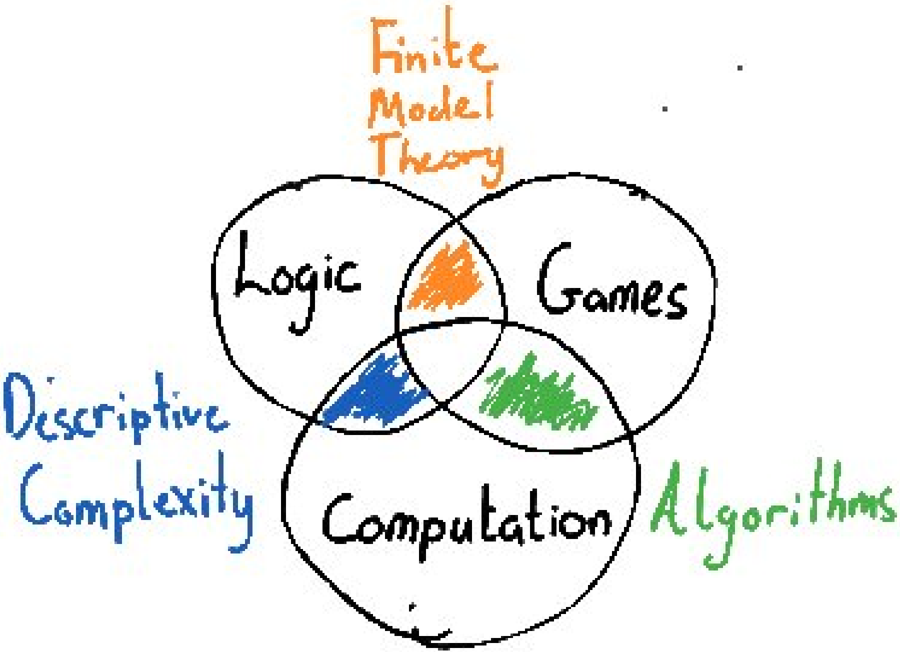
Game comonads & generalised quantifiers

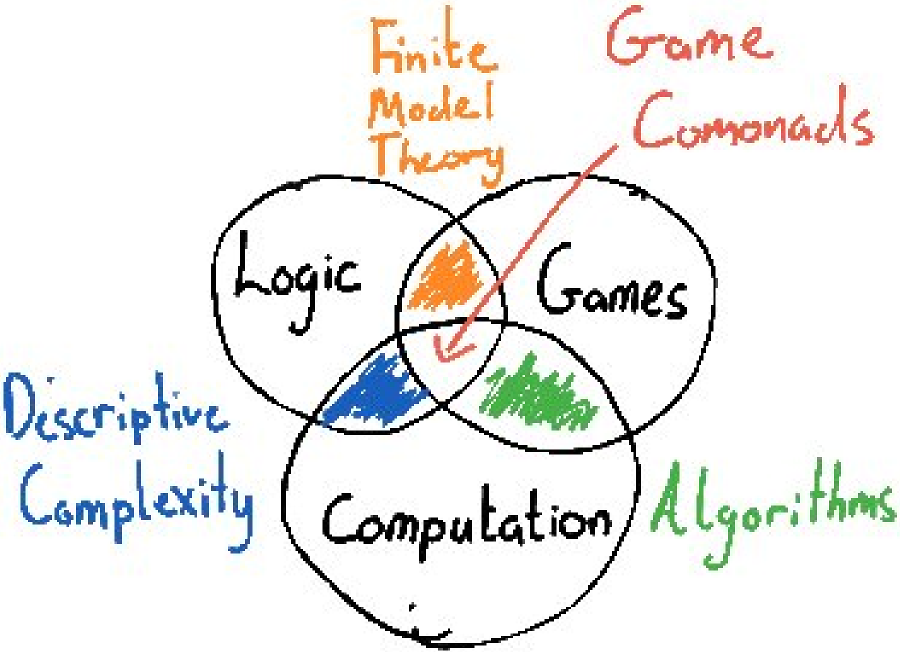
Adam Ó Conghaile
joint with *Anuj Dawar*

BCTCS 2020



Logic
Games
Computation


Finite Model Theory
Logic
Games
Descriptive Complexity
Computation
Algorithms


Finite Model Theory
Game Comonads
Logic
Games
Descriptive Complexity
Computation
Algorithms

Content

- Logic & Computation: Descriptive Complexity
- Logic & Games: Finite Model Theory
- Games & Computation: Algorithms
- Game Comonads at the intersection of all three
- Generalised quantifiers & my work

Preliminaries

We will look at computation, logic and games through **relation structures**

Signature σ has:

- relational symbols $R, T, E, \ldots$
- $\texttt{arity} : \sigma \to \mathbb{N}$

$\mathcal{R}(\sigma)$ has

- objects

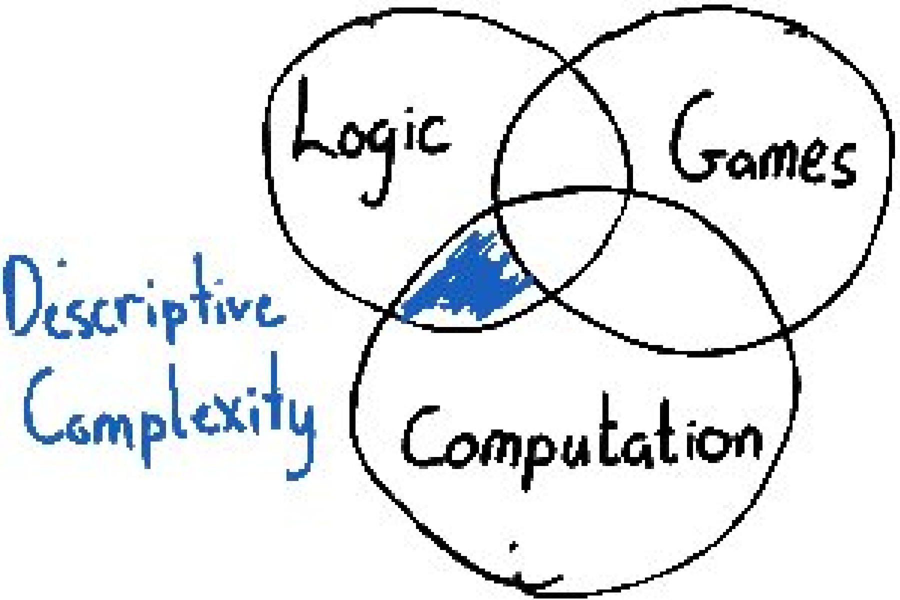
$$\mathcal{A} = \langle A, (R^A)_{R \in \sigma} \rangle \text{ where } R^A \subset A^{\texttt{arity}(R)} \text{ for each } R$$

- maps $\mathcal{A} \to \mathcal{B}$ are homomorphisms.

First order logic has syntax:

$$\mathbf{FO} = \top \mid \bot \mid R(x_1, \ldots x_m) \mid \neg\phi \mid \phi \vee \psi \mid \phi \wedge \psi \mid \exists x.\ \phi(x) \mid \forall x.\ \phi(x)$$

And usual semantics for the relation $\mathcal{A} \models \phi$
The "logics" ($\mathcal{L}$) we will talk about will be fragments/extensions of this.



Logic
Games
Computation
Descriptive Complexity

Logic & Computation: Classes and Queries

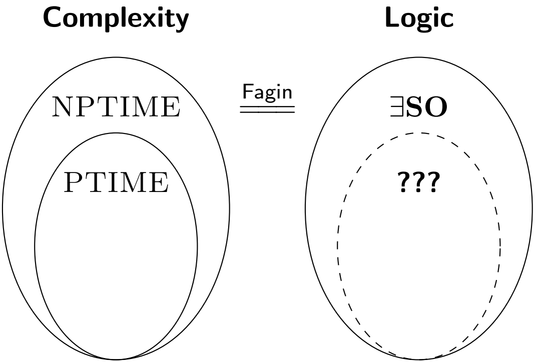
Given some class of (resource-limited) Turing machines $\mathcal{T}$ the **complexity class** associated to $\mathcal{T}$ is the collection of classes of finite relational structures (given a suitable encoding) recognised by a machine in $\mathcal{T}$

Given some logic $\mathcal{L}$, the **query class** associated to $\mathcal{L}$ is the collection of classes of finite relational structures which model some sentence ϕ in $\mathcal{L}$.

Descriptive complexity studies links of the form

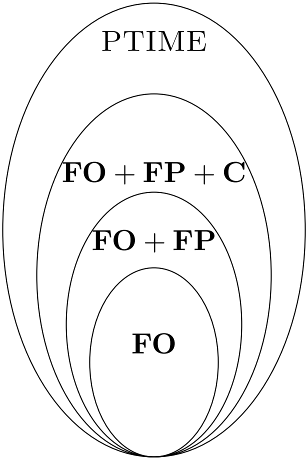
$$\mathbf{CC}(\mathcal{T}) = \mathbf{QC}(\mathcal{L})$$

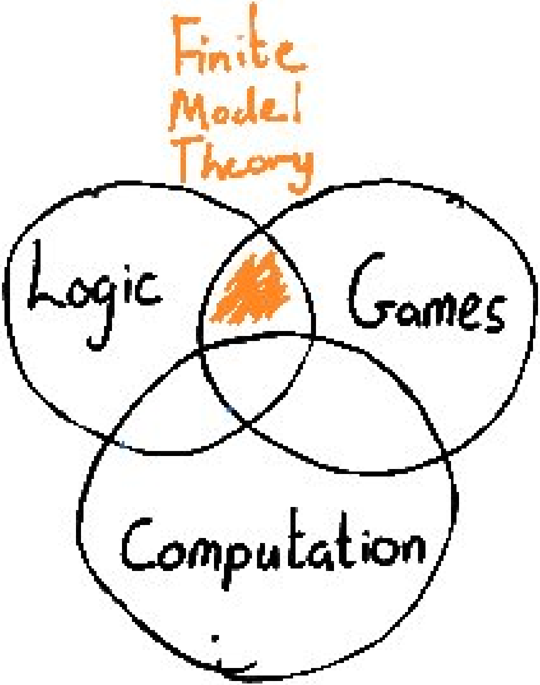
Logic & Computation: The search for a logic for PTIME



- Since Fagin showed that $\exists\mathbf{SO}$ captures NPTIME, finite model theorists have tried to find a logic that *captures* PTIME.
- $\mathbf{FO}$ is not enough (as we will see)
- To gain more power we need to add new types of computation to the logic. This can be done through quantifiers

Logic & Computation: Power and quantifiers




Finite Model Theory
Logic
Games
Computation

Logic & Games: Finite Model Theory

Want to determine if two structures agree on a certain logic, i.e.

$$\forall \phi \in \mathcal{L},\ \mathcal{A} \models \phi \iff \mathcal{B} \models \phi$$

To do this we use games!

Example *Ehrenfeucht-Fraïssé Game*
Two players: Spoiler and Duplicator. In round i

- Spoiler chooses A or B and then picks an element a_i or b_i
- Duplicator responds by choosing an element in the other structure

After each round we say Spoiler wins if the partial function $a_i \rightharpoonup b_i$ is a partial isomorphism between the two structures.

A Duplicator strategy which prevents Spoiler from ever winning will imply that $\mathcal{A}$ and $\mathcal{B}$ agree over a logic $\mathcal{L}$, additional rules on the game will determine exactly which logic.

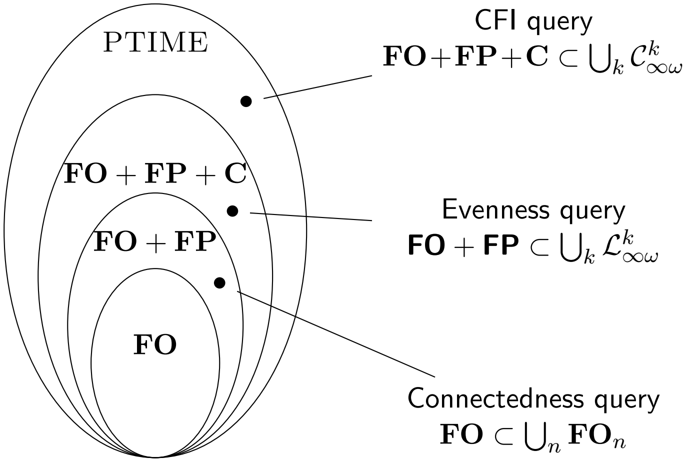
Logic & Games: Limits on spoiler $\iff$ syntactic restrictions

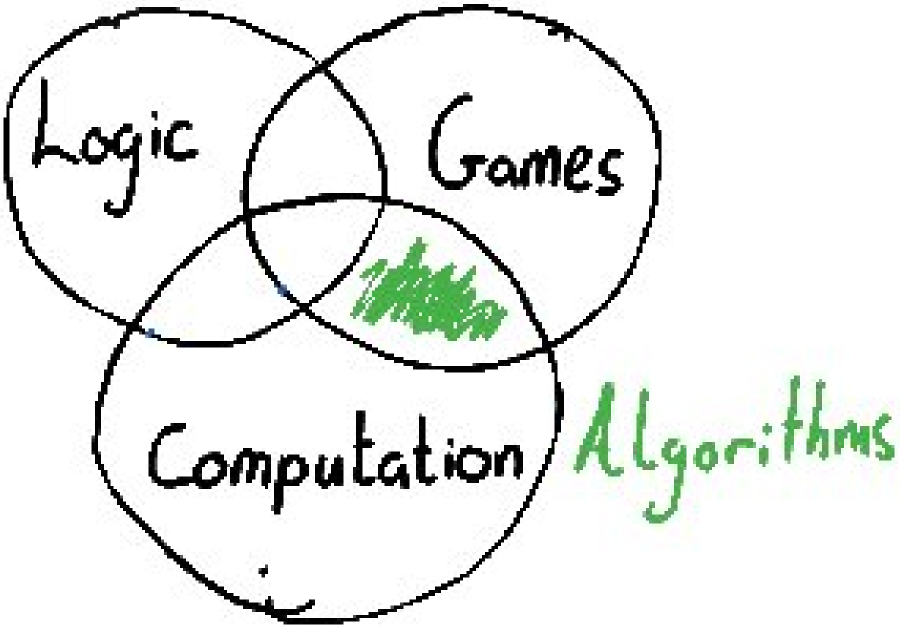
Rules	**Logic**
Play for n rounds	$\mathbf{FO}_n$
Limit to k pebbles	$\mathbf{FO}^k$
"One-way game"	$\exists + \mathbf{FO}$ $(\mathcal{A} \prec_{\exists+\mathcal{L}} \mathcal{B})$

Logic & Games: Limits on Duplicator $\Longleftrightarrow$ syntactic expansions

Rules	**Logic**
Play forever (with k pebbles)	$\mathcal{L}^k_{\infty\omega}$
Duplicator responds with a bijection	$\mathcal{L}^k_{\infty\omega} + \#$
Same bijection for n rounds	$\mathcal{L}^k_{\infty\omega} + \mathcal{Q}_{\mathbf{n}}$

Logic & Computation: Power and quantifiers




Logic
Games
Computation
Algorithms

Games & Computation: Approximations to Homomorphism & Isomorphism

Many computational tasks can be described as searching for homomorphism or isomorphism: e.g

- CSP: $\mathcal{X} \to \mathcal{D}$?
- Graph isomorphism: $\mathcal{G} \cong \mathcal{H}$?

Duplicator winning strategies for the various games discussed can be seen as approximations to homomorphism (one-way games) and approximations to isomorphism (two-way games)
Some of these correspond to known algorithms for approximating CSP and GI.

Games & Computation

Game	Algorithm
k pebble one-way game	k-local consistency for CSP
k pebble bijection game	k Weisfeiler-Lehman for GI

For certain special structures these approximations imply full homomorphism/isomorphism. In the case of the examples above the "special" property is a tree decomposition of width $\leq k+1$

A unified perspective: game comonads

So far, we have seen that spoiler-duplicator games:

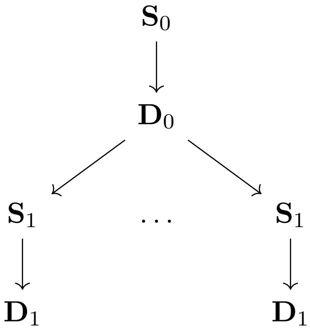
- help us evaluate expressiveness of different logics
- give us tractable algorithms for CSP/GI via approximations to homomorphism/isomorphism

Question: How can we realise these approximations to homomorphism/isomorphisms categorically?

Answer: Game comonads.

Game comonads: idea

Given some Spoiler-Duplicator game $\mathcal{G}(\mathcal{A}, \mathcal{B})$, can see (deterministic) duplicator strategies as trees:



$\mathbb{G}\mathcal{A} = \{\mathbf{S}_0 \dots \mathbf{S}_m \mid \mathbf{S}_i$ a valid Spoiler move in round i of $\mathcal{G}\}$

Goal: Choose a relational structure for $\mathbb{G}\mathcal{A}$ s.t.

$f : \mathbb{G}\mathcal{A} \to \mathcal{B}$ is a hom $\iff$ f is a winning strategy for Duplicator

Example: k-pebbling comonad

$$\mathbb{P}_k\mathcal{A} := (A \times [k])^+$$

$$\epsilon_A([(a_1, p_1), \ldots (a_n, p_n)]) = a_n$$

$$\delta_A([(a_1, p_1), \ldots (a_n, p_n)]) = [(s_1, p_1), \ldots (s_n, p_n)]$$

Relational structure chosen appropriately.

Example: k-pebbling comonad

Results (Abramsky, Dawar, Wang '17)

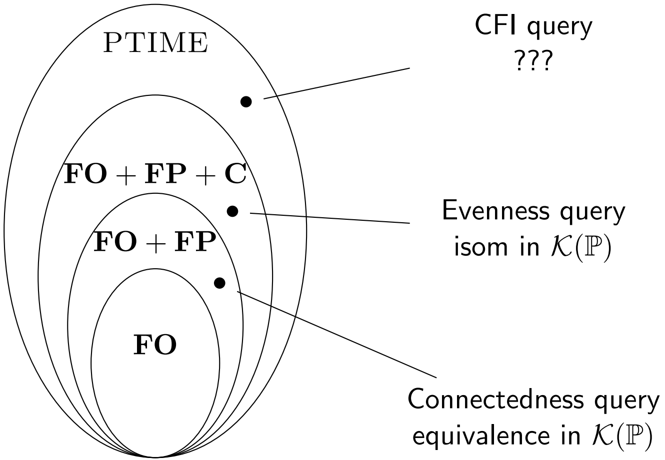
- $(\mathbb{P}_k, \epsilon, \delta)$ defines a comonad
- Kleisli homs $\mathbb{P}_k\mathcal{A} \to \mathcal{B}$ are k-local homs
- Kleisli isoms $A \cong_{\mathcal{K}(\mathbb{P}_k)} \mathcal{B}$ are proofs of k-WL equivalence
- The coalgebras, $\mathcal{A} \to \mathbb{P}_k\mathcal{A}$ are proofs of treewidth $\leq k+1$

Game comonads the story so far

Comonad $\mathbb{G}$	**Kleisli Homs** $\mathbb{G}\mathcal{A} \to \mathcal{B}$	**Kleisli Isoms** $\mathcal{A} \cong_{\mathcal{K}(\mathbb{G})} \mathcal{B}$	**Coalgebras** $\mathcal{A} \to \mathbb{G}\mathcal{A}$
k pebbling $\mathbb{P}_k$	$A \prec_{\exists + \mathcal{L}^k_{\infty\omega}} \mathcal{B}$	$A \equiv_{\mathcal{L}^k_{\infty\omega}(\#)} \mathcal{B}$	treewidth $\leq k+1$
n-round E-F $\mathbb{E}_n$	$A \prec_{\exists + \mathbf{FO}_n} \mathcal{B}$	$A \equiv_{\mathbf{FO}_n(\#)} \mathcal{B}$	treedepth $\leq k+1$
n-round bisim. $\mathbb{M}_n$	$A \prec_{\exists + \mathbf{ML}_n} \mathcal{B}$	$A \equiv_{\mathbf{ML}_n} \mathcal{B}$	modal depth $\leq k+1$

Others forthcoming for guarded fragment (Marsden et al.), pathwidth (Shah et al.)

Limits of this framework



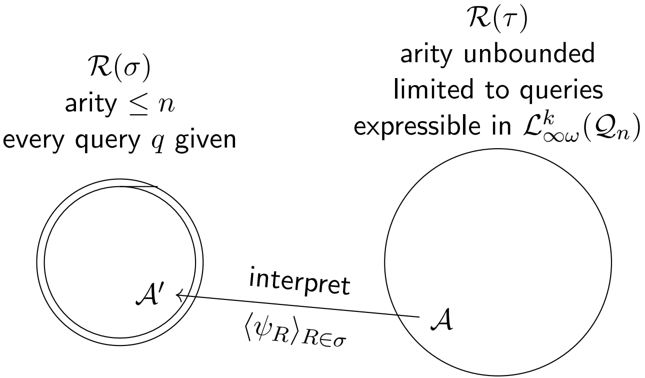
Generalised quantifiers

Recall from before that adding new quantifiers to our logic amounted to adding more computational power (getting us closer to PTIME for example)

This leads us to thinking of quantifiers as a logical version of an oracle in some sense.

The notion of generalised (or Lindeström) quantifiers makes this precise.

Generalised quantifiers: idea



A game for generalised quantifiers

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

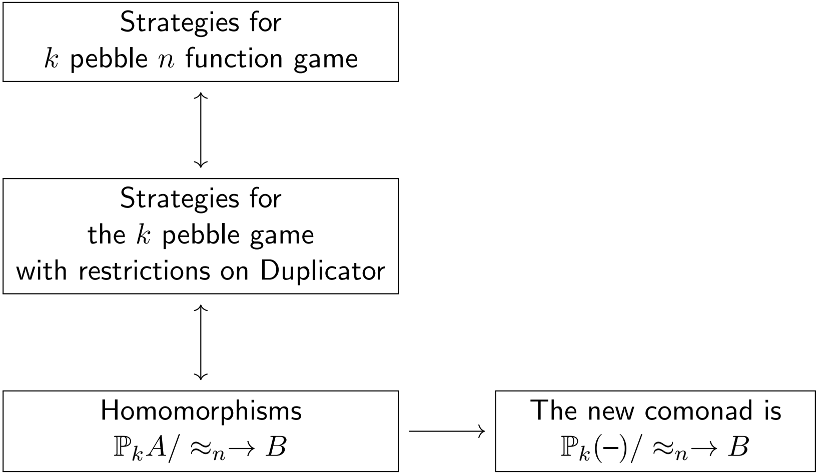
Rules	**Logic**
Play forever (with k pebbles)	$\mathcal{L}^k_{\infty\omega}$
Duplicator responds with a bijection	$\mathcal{L}^k_{\infty\omega} + \#$
Same bijection for n rounds	$\mathcal{L}^k_{\infty\omega} + \mathcal{Q}_\mathbf{n}$

Relaxing Hella's game

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

Rules	**Logic**
Play forever (with k pebbles)	$\mathcal{L}^k_{\infty\omega}$
Duplicator responds with a bijection	$\mathcal{L}^k_{\infty\omega} + \#$
Same bijection for n rounds	$\mathcal{L}^k_{\infty\omega}(\mathcal{Q}_\mathbf{n})$
Same function for n rounds	$\exists + \mathcal{L}^k_{\infty\omega}(\mathcal{Q}^\mathbf{H}_\mathbf{n})$

Comonadifying the n function game



Our results

Comonad $\mathbb{G}$	Kleisli Homs $\mathbb{G}\mathcal{A} \to \mathcal{B}$	Kleisli Isoms $\mathcal{A} \cong_{\mathcal{K}(\mathbb{G})} \mathcal{B}$	Coalgebras $\mathcal{A} \to \mathbb{G}\mathcal{A}$
k pebbling $\mathbb{P}_k$	$\mathcal{A} \prec_{\exists+\mathcal{L}^k_{\infty\omega}} \mathcal{B}$	$\mathcal{A} \equiv_{\mathcal{L}^k_{\infty\omega}(\#)} \mathcal{B}$	treewidth $\leq k+1$
n-round E-F $\mathbb{E}_n$	$\mathcal{A} \prec_{\exists+\mathbf{FO}_n} \mathcal{B}$	$\mathcal{A} \equiv_{\mathbf{FO}_n(\#)} \mathcal{B}$	treedepth $\leq k+1$
n-round bisim. $\mathbb{M}_n$	$\mathcal{A} \prec_{\exists+\mathbf{ML}_n} \mathcal{B}$	$\mathcal{A} \equiv_{\mathbf{ML}_n} \mathcal{B}$	modal depth $\leq k+1$
k-pebble n-function $\mathbb{P}_{n,k}$	$\mathcal{A} \prec_{+\mathcal{L}^k_{\infty\omega}(\mathcal{Q}^{\mathbf{H}}_{\mathbf{n}})} \mathcal{B}$	$\mathcal{A} \equiv_{\mathcal{L}^k_{\infty\omega}(\mathcal{Q}_{\mathbf{n}})} \mathcal{B}$	gen. treedepth $\leq k+1$

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 $\mathbf{LA}^{\omega}(\mathcal{Q})$ ($\mathcal{L}^{\omega}_{\infty\omega}$ extended with all **linear algebraic** quantifiers over $\mathbb{F}_p$ for each $p \in \mathcal{Q}$)