Everything Everywhere All in One

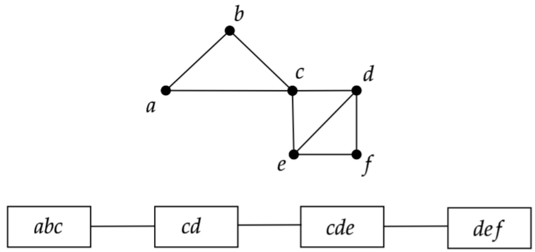
Yoàv Montacute

University of Cambridge

Joint work with Nihil Shah
Resources in Computation 2022

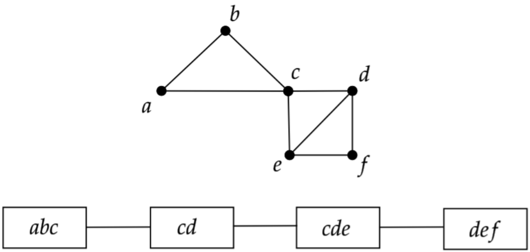
Pathwidth

Path decomposition (Robertson and Seymour)



Pathwidth

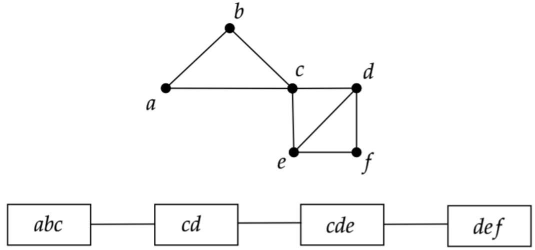
Path decomposition (Robertson and Seymour)



We define a **coalgebra number** $^{\mathsf{PR}}(A)$ of a finite structure A to be the least k such that there exists a coalgebra $: A \; ! \; \mathsf{PR}_k A$.

Pathwidth

Path decomposition (Robertson and Seymour)



We define a **coalgebra number** ${}^{\mathrm{PR}}(A)$ of a finite structure A to be the least k such that there exists a coalgebra $: A \; ! \; \mathrm{PR}_k A$.

Theorem (coalgebraic characterisation of pathwidth)

For all -structures A, $\mathrm{pw}(A) = {}^{\mathrm{PR}}(A) \quad 1$.

The comonad PR_k

Given a relational structure $\mathcal{A} = (A; R_1; \dots; R_m)$, we define $\mathsf{PR}_k \mathcal{A}$ as follows:

The comonad PR_k

Given a relational structure $A = (A, R_1, \ldots, R_m)$, we define $\mathsf{PR}_k A$ as follows:

The universe $\mathsf{PR}_k A$ consists of all pairs of non-empty sequences and indices

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

where $p_i \in [k]$, $a_i \in A$ and $i \in [n]$.

The comonad PR_k

Given a relational structure $\mathcal{A} = (A, R_1, \ldots, R_m)$, we define $\mathsf{PR}_k\mathcal{A}$ as follows:

The universe $\mathsf{PR}_k A$ consists of all pairs of non-empty sequences and indices

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

where $p_i \in [k]$, $a_i \in A$ and $i \in [n]$.

For each relation R (for simplicity let R be binary), we define $((s, i), (t, j)) \in R^{\mathsf{PR}_k\mathcal{A}}$ if

1. $s = t$;
2. If $j > i$ (resp. $i > j$), then the i-th pebble of s does not appear in $s(i, j]$;
3. $R^{\mathcal{A}}(\varepsilon_{\mathcal{A}}(s, i), \varepsilon_{\mathcal{A}}(t, j))$, where $\varepsilon_{\mathcal{A}}([(p_1, a_1), \ldots, (p_n, a_n)], i) = a_i$.

The coKleisli category

Consider the category $K(\mathsf{PR}_k)$ which is the coKleisli category over the comonad PR_k. Its objects are the same as $R(\quad)$ and morphisms from A to B in the category are homomorphisms $f : \mathsf{PR}_k A \to B$.

The coKleisli category

Consider the category $K(\mathsf{PR}_k)$ which is the coKleisli category over the comonad PR_k. Its objects are the same as $R(-)$ and morphisms from A to B in the category are homomorphisms $f : \mathsf{PR}_k A \to B$.

Composition of morphisms $g \circ_{K(\mathsf{PR}_k)} f = g \circ f^*$, where

$$f^* : ([(p_1, a_1), \ldots, (p_n, a_n)], i) \mapsto ([(p_1, f(s_1)), \ldots, (p_n, f(s_n))], i)$$

and $s_j = ([(p_1, a_1), \ldots, (p_n, a_n)], j)$, for all $j \in [n]$.

All-in-one k-pebble game

Let $9^{+} \mathrm{f} L^{k}$ denote the fragment of $9^{+} L^{k}$ with restricted conjunction. i.e. every conjunction has at most one quantified formula with free variables.

All-in-one k-pebble game

Let $9^{+} \mathsf{f} L^{k}$ denote the fragment of $9^{+} L^{k}$ with restricted conjunction. i.e. every conjunction has at most one quantified formula with free variables.

Consider the all-in-one k-pebble game $9\mathsf{P}\,\mathsf{Peb}_{k}(A; B)$. The game is played in one round during which:

1. Spoiler provides a sequence of pebble placements $[(p_1; a_1); :::; (p_n; a_n)]$.
2. Duplicator answers with a sequence $[(p_1; b_1); :::; (p_n; b_n)]$.

If every prefix induces a partial homomorphism from A to B, then Duplicator wins the game.

All-in-one k-pebble game

Let $9^{+} \mathsf{f} L^{k}$ denote the fragment of $9^{+} L^{k}$ with restricted conjunction. i.e. every conjunction has at most one quantified formula with free variables.

Consider the all-in-one k-pebble game $9\mathsf{PPeb}_k(A; B)$. The game is played in one round during which:

1. Spoiler provides a sequence of pebble placements $[(p_1; a_1); : : : ; (p_n; a_n)]$.
2. Duplicator answers with a sequence $[(p_1; b_1); : : : ; (p_n; b_n)]$.

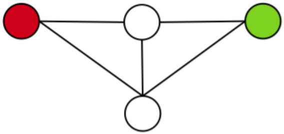
If every prefix induces a partial homomorphism from A to B, then Duplicator wins the game.

Theorem (morphism power theorem)

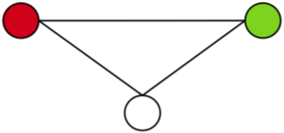
Given two -structures A and B, the following are equivalent:

- *Duplicator has a winning strategy in $9\mathsf{PPeb}_k(A; B)$.*
- $A \ \mathrm{V}^{9^{+} \mathsf{f} L^{k}} \ B$.
- *There exists a coKleisli morphism $f : \mathsf{PR}_k A \ ! \ B$.*

2-pebble game (standard vs. all-in-one)

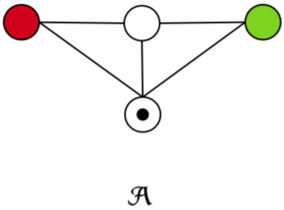


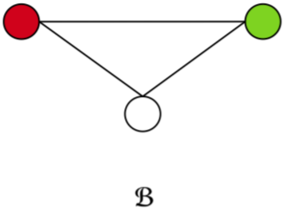
$\mathcal{A}$



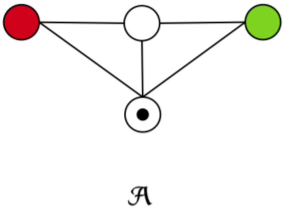
$\mathfrak{B}$

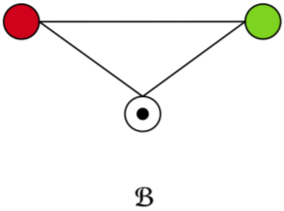
2-pebble game (standard vs. all-in-one)



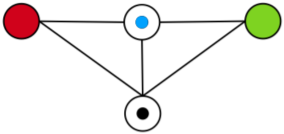


2-pebble game (standard vs. all-in-one)

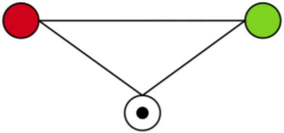




2-pebble game (standard vs. all-in-one)

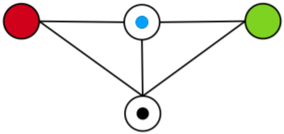


$\mathcal{A}$

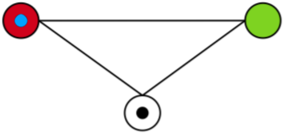


$\mathcal{B}$

2-pebble game (standard vs. all-in-one)

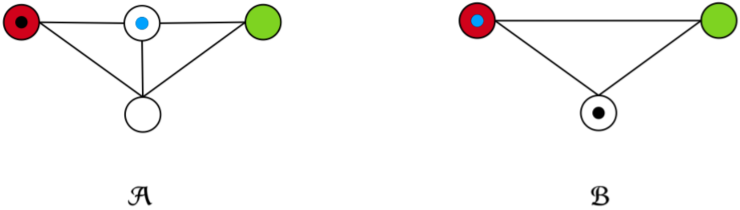


$\mathcal{A}$

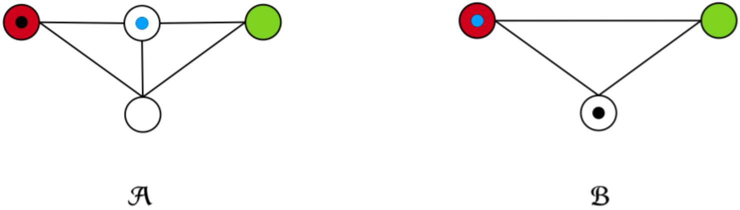


$\mathfrak{B}$

2-pebble game (standard vs. all-in-one)



2-pebble game (standard vs. all-in-one)



In this example Duplicator loses the 2-pebble game but wins the all-in-one 2-pebble game.

All-in-one bijective k-pebble game

Consider the all-in-one bijective k-pebble game $\#\mathsf{PPeb}_k(A; B)$. The game is played in one round during which:

1. Spoiler provides a sequence of pebble placements with one pebble placement hidden $[(p_1; a_1); : : : ; (p_j; \_); : : : ; (p_n; a_n)]$.
2. Duplicator answers with a sequence $[(p_1; \;\; _1); : : : ; (p_n; \;\; _n)]$ of pebble placements and bijections $\;\; _i : A \; ! \; B$.

If every prefix induces a partial isomorphism from A to B, then Duplicator wins the game.

All-in-one bijective k-pebble game

Consider the all-in-one bijective k-pebble game $\#\mathsf{PPeb}_k(A, B)$. The game is played in one round during which:

1. Spoiler provides a sequence of pebble placements with one pebble placement hidden $[(p_1, a_1), \ldots, (p_j, \_), \ldots, (p_n, a_n)]$.
2. Duplicator answers with a sequence $[(p_1, \;_1), \ldots, (p_n, \;_n)]$ of pebble placements and bijections $\;_i : A \to B$.

If every prefix induces a partial isomorphism from A to B, then Duplicator wins the game.

Theorem (isomorphism power theorem)

Given two -structures A and B, the following are equivalent:

- *Duplicator has a winning strategy in* $\#\mathsf{Peb}_k(A, B)$.
- $A \; {}^{\#\mathsf{f}\,L^k} \; B$.
- *There exists a coKleisli isomorphism* $f : \mathrm{PR}_k A \to B$.

Bijective 2-pebble game (standard vs. all-in-one)

Bijective 2-pebble game (standard vs. all-in-one)

The $\#L^k$-formula

$$\exists x \;\; \exists y \;\; Exy \wedge \exists_{\geq 2} x(Eyx \wedge Rx) \;\; \wedge \exists y \;\; Exy \wedge \exists_{\geq 2} x(Eyx \wedge Bx)$$

is true in A but not in B.

Lovász-type theorem for pathwidth

Theorem (Dawar, Jakl and Reggio)

Given a locally finite category $\mathcal{C}$ with pushout and proper factorisation system, for all $A, B \in \mathcal{C}$,

$$A \cong B \iff |\hom_{\mathcal{C}}(C, A)| = |\hom_{\mathcal{C}}(C, B)|, \forall C \in \mathcal{C}.$$

Lovász-type theorem for pathwidth

Theorem (Dawar, Jakl and Reggio)

Given a locally finite category $\mathcal{C}$ with pushout and proper factorisation system, for all $A, B \in \mathcal{C}$,

$$A \cong B \iff |\hom_{\mathcal{C}}(C, A)| = |\hom_{\mathcal{C}}(C, B)|, \forall C \in \mathcal{C}.$$

Theorem (Lovász-type theorem)

For every finite Σ-structures A and B:

$$A \equiv^{\#\mathsf{f}L^k} B \iff |\hom_{\Sigma_f}(C, A)| = |\hom_{\Sigma_f}(C, B)|,$$

for every finite Σ-structure C with pathwidth at most k.

Computational complexity

Ongoing work with Anuj Dawar and Nihil Shah.

Figure: Bisimulation vs. trace-equivalence

Computational complexity

Theorem (Balcázar, Gabarró and Sántha)

Deciding bisimulation is P-complete.

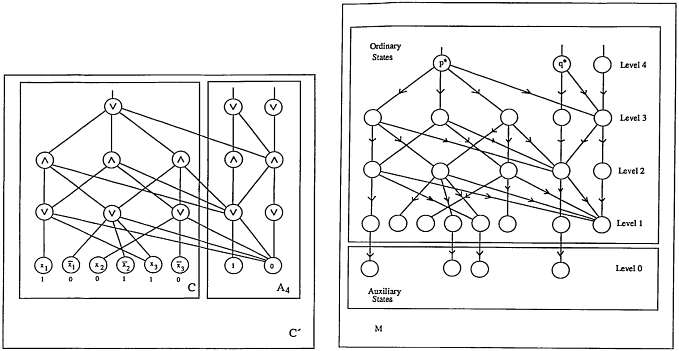


Figure: Balcazar, Gabarro and Santha (1992)

Computational complexity

Theorem (Kolaitis and Panttaja)

Determining the winner of the k-pebble game for a xed k is P-complete

Theorem (Kolaitis and Panttaja)

Determining the winner of the k-pebble game with k as an input is $EXPTIME$-complete.

Computational complexity

Theorem (Kolaitis and Panttaja)

Determining the winner of the k-pebble game for a xed k is P-complete

Theorem (Kolaitis and Panttaja)

Determining the winner of the k-pebble game with k as an input is $EXPTIME$-complete.

Figure: Kolaitis and Panttaja (2003)

Computational Complexity

Theorem (Chandra and Stockmeyer)

Deciding trace-equivalence is $PSPACE$-complete

Theorem (Chandra and Stockmeyer)

Deciding trace-equivalence is $PSPACE$-complete

Figure: Kanellakis and Smolka (1990)

Computational complexity

Conjecture

Determining the winner of the all-in-one k-pebble game for a xed k is $PSPACE$-complete.

Conjecture

Determining the winner of the all-in-one k-pebble game with k as an input is $EXPSPACE$-complete.

Thank you