Relational Colour Refinement for Non-Relational Signatures

Theodor Jurij Teslia

September 5, 2025

RWTH Aachen University

Introduction

- Colour Refinement is an important and interesting algorithm
- Applied in modern isomorphism solvers
- Can be characterised logically and combinatorially
- Extension to more than graphs seems desirable
- Scheidt and Schweikardt bibliography introduced Relational Colour Refinement
- Conceptually similar to classical Colour Refinement
- Also has a logical and a combinatorial characterisation

Contents of this presentation

1. Classical Colour Refinement
2. Relational Colour Refinement
3. Relational Colour Refinement for Structures With Functions
4. RCR on Subclasses of Relational Structures
5. Sketch of a Proof
6. Conclusion

Classical Colour Refinement

Colour Refinement

- Also called *CR* or 1-*dimensional Weisfeiler-Leman* algorithm
- Iterative graph algorithm
- Constructs colour for every vertex, based on colours of neighbours

Definition (Colour Refinement)

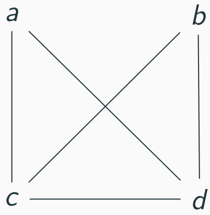
For graph $G = (V, E)$, for every $v \in V$ and $i \in \mathbb{N}$:

- Initial colour: $C_0(v) := 0$
- Next rounds:

$$C_{i+1}(v) := (C_i(v), \{\!\{ C_i(u) : \{v, u\} \in E \}\!\})$$

Example for CR

G:



- $C_0(a) = C_0(b) = C_0(c) = C_0(d) = 0$

- $C_1(a) = C_1(b) = (0, \{\!\{0, 0\}\!\})$
- $C_1(c) = C_1(d) = (0, \{\!\{0, 0, 0\}\!\})$

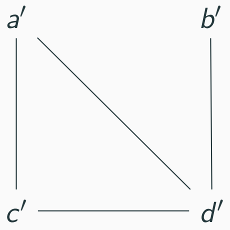
Distinguished graphs

- CR distinguishes two graphs G and H, if
- there exists $C_i(v)$ in colouring of G or H, such that the number of vertices with colour $C_i(v)$ is different in G than in H

Distinguished graphs

- CR distinguishes two graphs G and H, if
- there exists $C_i(v)$ in colouring of G or H, such that the number of vertices with colour $C_i(v)$ is different in G than in H

H:



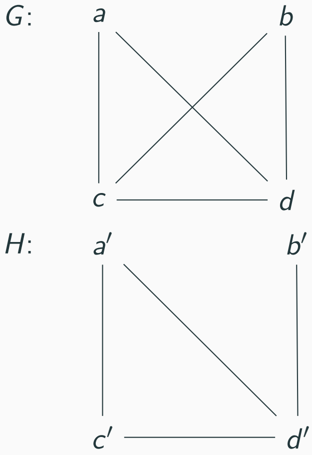
- Colours in first round equal
- $C_1(b') = (0, \{\!\{0\}\!\})$ does not appear in G

$\Rightarrow$ Colour Refinement distinguishes G and H

Characterisations of CR

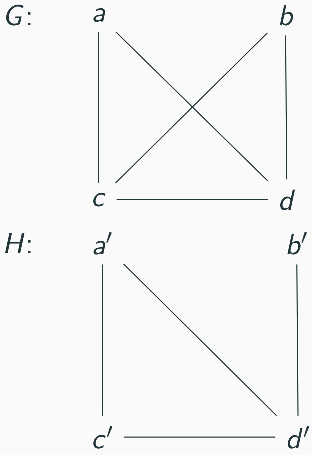
- There are equivalent characterisations for CR
- Due to bibliography:
  CR distinguishes G and H if, and only if, there exists $\varphi \in \mathsf{C}_2$, such that $G \models \varphi$ and $H \not\models \varphi$
- Due to bibliography:
  CR distinguishes G and H if, and only if, there exists tree T, such that $\hom(T, G) \neq \hom(T, H)$

Application of Characterisations to Example



- Used existence of colour $(0, \{\!\{0\}\!\})$ in colouring of H to distinguish G and H
- From colour it follows that vertex with degree 1 exists
- $\exists^{\geq 1} x \,.\, \exists^{=1} y \,.\, E(x, y)$ distinguishes G and H

Application of Characterisations to Example



- Used existence of colour $(0, \{\!\{0\}\!\})$ in colouring of H to distinguish G and H
- From colour it follows that vertex with degree 1 exists
- $\exists^{\geq 1} x \, . \, \exists^{=1} y \, . \, E(x, y)$ distinguishes G and H

- There are 5 edges in G but only 4 in H
- Tree $T := (\{v, u\}, \{\{v, u\}\})$ has 10 homomorphisms to G and 8 to H

Relational Colour Refinement

Relational Colour Refinement

- Called *RCR* for short
- Applies variant of classical Colour Refinement on tuples of structure
- Uses atomic type as part of initial colouring
- Uses pairs of indices as edges to mark shared elements of tuples
- Formally:

$$\mathrm{atp}(\mathbf{a}) = \{R \in \sigma : \mathbf{a} \in R\}$$

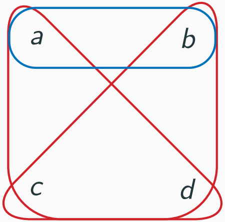
  and

$$\mathrm{stp}(\mathbf{a}, \mathbf{b}) = \{(i, j) \in [n] \times [m] : a_i = b_j\}$$

The Algorithm

- For relational structure $\mathfrak{A}$ and all tuples $\mathbf{a} \in \mathbf{A}$:
- Initial colour: $\varrho_0(\mathbf{a}) = (\mathrm{atp}(\mathbf{a}), \mathrm{stp}(\mathbf{a}, \mathbf{a}))$
- For the next rounds: $\varrho_{i+1}(\mathbf{a}) = (\varrho_i(\mathbf{a}), \{\!\{(\mathrm{stp}(\mathbf{a}, \mathbf{b}), \varrho_i(\mathbf{b})) : \mathrm{set}(\mathbf{a}) \cap \mathrm{set}(\mathbf{b}) \neq \emptyset\}\!\})$

An Example for RCR

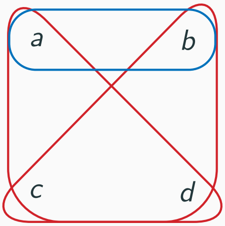


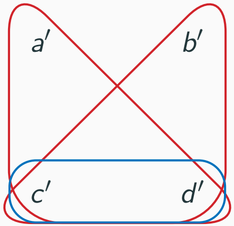
- Structure $\mathfrak{A} = (A, R^{\mathfrak{A}}, T^{\mathfrak{A}})$
- $A = \{a, b, c, d\},\ R^{\mathfrak{A}} = \{(a, b)\},\ T^{\mathfrak{A}} = \{(a, c, d), (b, c, d)\}$

- $\varrho_0((a, b)) = (\{R\}, \{(1, 1), (2, 2)\})$ and
  $\varrho_0((a, c, d)) = \varrho_0((b, c, d)) = (\{T\}, \{(1, 1), (2, 2), (3, 3)\})$
- $\varrho_1((a, c, d)) = (\varrho_0((a, c, d)), \{\!\{(\{(1, 1)\}, \varrho_0((a, b))), \dots\}\!\})$ and
  $\varrho_1((b, c, d)) = (\varrho_0((b, c, d)), \{\!\{(\{(1, 2)\}, \varrho_0((a, b))), \dots\}\!\})$

Distinguishing Relational Structures with RCR

- RCR distinguishes, if some colour appears differently often in the structures





- $\varrho_1((a, c, d))$ appears in colouring of left structure but not in right

Relational Colour Refinement

Logical Characterisation of RCR

Guarded Fragment of Counting Logic

- C_2 characterises CR on graphs
- Guarded fragment of counting logic GF(C) characterises RCR

Guarded Fragment of Counting Logic

- Everything except for quantifiers defined as in classical counting logic
- For atomic formula $\Delta \in \mathrm{GF(C)}$ and formula $\varphi \in \mathrm{GF(C)}$, we call Δ a guard for φ, if $\mathrm{free}(\Delta) \supseteq \mathrm{free}(\varphi)$
- Quantifiers appear only in form $\exists^{\geq i}\mathbf{v}\,.(\Delta \wedge \varphi)$, where Δ is guard for φ and $\mathrm{set}(\mathbf{v}) \subseteq \mathrm{free}(\Delta)$

- Examples:
  - $\exists^{\geq 2}(x, y)\,.(E(x, y) \wedge T(y)) \in \mathrm{GF(C)}$
  - $\exists^{\geq 3}(x, y, z)\,.(E(x, y) \wedge E(y, z) \wedge E(z, x)) \notin \mathrm{GF(C)}$

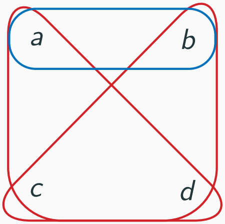
Characterising RCR Using Logic

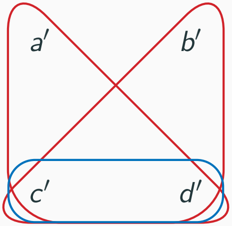
Theorem B from bibliography

Let $\mathfrak{A}$ and $\mathfrak{B}$ be two relational structures. Then the two following statements are equivalent.

1. RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$
2. There exists a sentence in GF(C) that is satisfied by $\mathfrak{A}$, but not by $\mathfrak{B}$

Example for Logical Characterisation of RCR





- We have seen RCR distinguishes the structures
- Formula $\exists^{\geq 1}(x, y, z).\,\big(T(x, y, z) \wedge \exists^{\geq 1}(y).\,(R(x, y))\big)$ satisfied by left and not by right structure

Relational Colour Refinement

Combinatorial Characterisation of RCR

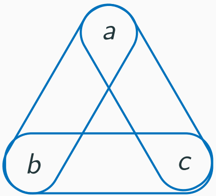
Acyclic Structures

- Counting homomorphisms from trees characterises CR on graphs
- Abstraction from trees to relational structures is needed: α-acyclic structures (in the following only acyclic structures)

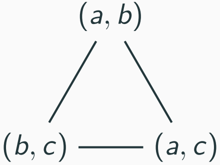
Acyclic Structures

- Let $\mathfrak{C}$ be relational structure
- Join tree J for $\mathfrak{C}$ is tree with $V(J) = \mathbf{C}$ and fulfils join-tree-property:
  - For every $e \in C$, the set $\{\mathbf{x} \in V(J) : e \in \text{set}(x)\}$ induces a connected subtree
- We call $\mathfrak{C}$ acyclic, if it has a join tree

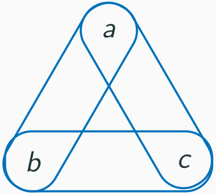
Examples for Acyclic Structures



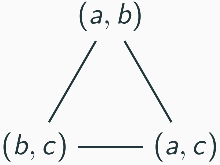
No:

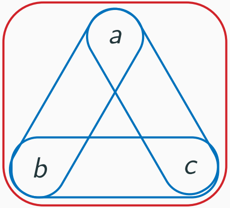


Examples for Acyclic Structures

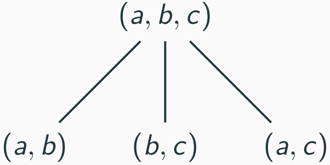


No:





Yes:



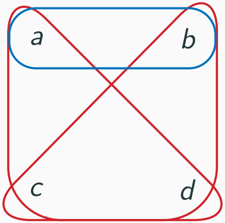
Characterising RCR Using Homomorphism Counting

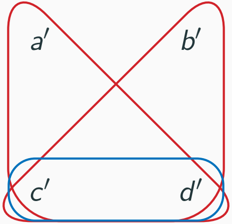
Theorem A from bibliography

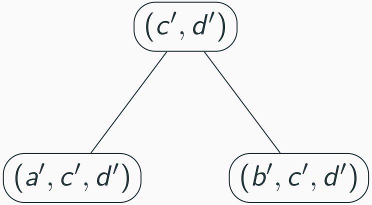
Let $\mathfrak{A}$ and $\mathfrak{B}$ be relational structures. Then the two following statements are equivalent.

1. RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$
2. There exists an acyclic relational structure $\mathfrak{C}$, such that it has a different number of homomorphisms to $\mathfrak{A}$ than to $\mathfrak{B}$

Example for Combinatorial Characterisation of RCR







- Right tree is join tree for middle structure, therefore middle structure is acyclic
- Identity is homomorphism, so middle structure has at least one homomorphism to itself
- Middle structure has no homomorphisms to left structure

Relational Colour Refinement for Structures With Functions

Relational Colour Refinement for Structures With Functions

- Many interesting structures use functions
- Colour Refinement algorithm for such structures seems desirable
- Will use the results of Scheidt and Schweikardt bibliography and investigate how robust they are
- Following structure:
    1. Presentation of two approaches for Colour Refinement for non-relational signatures
    2. Logical characterisation of both approaches
    3. Discussion on combinatorial characterisation

Naive RCR

- Goal: Encode non-relational structures and signatures as relational ones
- Functions can directly be interpreted as relations:

$$f(\mathbf{x}) = y \iff (\mathbf{x}y) \in R_f$$

- For non-relational signature σ define relational signature σ':
  - Relation symbol $R \in \sigma$ of arity $n \to$ introduce $R \in \sigma'$ of arity n
  - Function symbol $f \in \sigma$ of arity $n \to$ introduce $R_f \in \sigma'$ of arity $n+1$
- Encode σ-structure $\mathfrak{A}$ as σ'-structure $\mathfrak{A}'$:
  - For relation symbol $R \in \sigma$: $R^{\mathfrak{A}'} := R^{\mathfrak{A}}$
  - For function symbol $f \in \sigma$: $R_f^{\mathfrak{A}'} := \{(\mathbf{x}y) : f^{\mathfrak{A}}(\mathbf{x}) = y\}$
- We say naive RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$, if RCR distinguishes the encodings

Idea of the Transitive Expansion

- Approach is only defined for unary function symbols
- Encoding emulates the nesting of function applications
- Encode function f as family of relations $R_{f^1}, R_{f^2}, \dots,$ where $(x, y) \in R_{f^i}$ if $\underbrace{f(f(\dots f(}_{i \text{ times}} x))) = y$
- In the following: $f^i(x)$ written for $\underbrace{f(f(\dots f(}_{i \text{ times}} x)))$
- For multiple functions, also encode alternations, for example R_{fg} or $R_{g^2 f^3}$

Transitive Expansion i

Alternations of Function Applications

- Let σ be signature with unary function symbols
- Define set of all allowed function application alternations Alters_n^k as $\mathrm{Alters}_n^0(\sigma) = \{\mathrm{id}\}$ and

$$\begin{aligned}\mathrm{Alters}_n^k(\sigma) \coloneqq \mathrm{Alters}_n^{k-1}(\sigma) \cup \{f_1^{m_1} \dots f_k^{m_k} : & f_1, \dots, f_k \in \sigma_{\mathrm{Func}} \\ & \wedge \forall i \in [k] \,.\, m_i \in [n] \\ & \wedge \forall i \in [k-1] \,.\, f_i \neq f_{i+1}\}.\end{aligned}$$

- Example:
  - $\sigma = \{f/1, g/1\}$
  - $\mathrm{Alters}_2^2(\sigma) = \underbrace{\{\mathrm{id}\}}_{k=0} \cup \underbrace{\{f, f^2, g, g^2\}}_{k=1} \cup \underbrace{\{fg, fg^2, f^2g, f^2g^2, gf, \dots\}}_{k=2}$

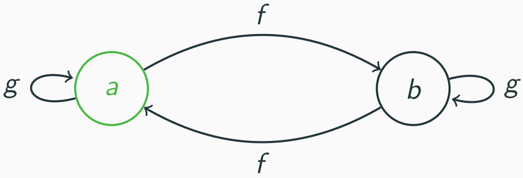
Transitive Expansion ii

Transitive Expansion

- For alternation depth k and σ-structure $\mathfrak{A}$ with $|\mathfrak{A}| = n$ define transitive expansion $\widetilde{\mathfrak{A}}$ as a $\widetilde{\sigma}$-structure
- For $\alpha, \beta, \alpha_1, \ldots, \alpha_\ell \in \mathrm{Alters}_n^k(\sigma)$ and relation symbol $R \in \sigma$ of arity ℓ, insert relation symbol $\mathrm{Eq}_{\alpha,\beta}$ of arity 2 and relation symbol $R_{\alpha_1,\ldots,\alpha_\ell}$ of arity ℓ into $\widetilde{\sigma}$
- Define $\mathrm{Eq}_{\alpha,\beta}^{\widetilde{\mathfrak{A}}} := \{(x, y) : \alpha^{\mathfrak{A}}(x) = \beta^{\mathfrak{A}}(y)\}$ and $R_{\alpha_1,\ldots,\alpha_\ell}^{\widetilde{\mathfrak{A}}} := \{(x_1, \ldots, x_\ell) : (\alpha_1^{\mathfrak{A}}(x_1), \ldots, \alpha_\ell^{\mathfrak{A}}(x_\ell)) \in R^{\mathfrak{A}}\}$

- For $k \in \mathbb{N}$ we say that RCR_k distinguishes structures $\mathfrak{A}$ and $\mathfrak{B}$, if RCR distinguishes the transitive expansions with alternation depth k

Example for the Transitive Expansion



- Structure $\mathfrak{A} = (A, R^{\mathfrak{A}}, f^{\mathfrak{A}}, g^{\mathfrak{A}})$
- $k = 1$ and $n = 2$: $\mathrm{Alters}_2^1(\sigma) = \{\mathrm{id}, f, f^2, g, g^2\}$
- $\widetilde{\sigma} = \{R_{\mathrm{id}}, R_f, R_{f^2}, R_g, R_{g^2}, \mathrm{Eq}_{\mathrm{id},\mathrm{id}}, \mathrm{Eq}_{\mathrm{id},f}, \mathrm{Eq}_{\mathrm{id},f^2}, \ldots, \mathrm{Eq}_{g^2,g^2}\}$
- Examples:
  - $R_f^{\widetilde{\mathfrak{A}}} = \{b\}$
  - $\mathrm{Eq}_{f^2,\mathrm{id}}^{\widetilde{\mathfrak{A}}} = \{(a, a), (b, b)\}$
  - $\mathrm{Eq}_{g,f}^{\widetilde{\mathfrak{A}}} = \{(a, b), (b, a)\}$

Relational Colour Refinement for Structures With Functions

Logical Characterisation of Naive RCR

Nesting-Free Guarded Fragment of Counting Logic

nfGF(C)

- Extends given definition of GF(C) for non-relational signatures
- Allow atomics of the following forms
  - Relation symbols and variable equations like in GF(C)
  - For function symbol f of arity ℓ and variables $x_1, \ldots, x_\ell, y$:
    $f(x_1, \ldots, x_\ell) = y \in \text{nfGF(C)}$

- Forbid nesting of terms, for example $f(g(x), y) = z$
- Informally: Usage of function symbols like relation symbols

Characterising Naive RCR Logically

Logical Characterisation of Naive RCR

Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures. Then the two following statements are equivalent.

1. Naive RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$
2. There exists a sentence $\varphi \in \mathrm{nfGF(C)}$ which is fulfilled by $\mathfrak{A}$, but not by $\mathfrak{B}$

Proof idea:

- Naive RCR distinguishes structures iff. RCR distinguishes encodings iff. there exists a sentence in GF(C) that distinguishes the encodings
- Define translation of sentences in GF(C) over signature σ' to and from sentences in nfGF(C) over signature σ
  - $R_f(\mathbf{x}y) \leftrightarrow f(\mathbf{x}) = y$

Relational Colour Refinement for Structures With Functions

Logical Characterisation of RCR_k

GF(C) **with alternation depth** k

GF(C)$_k$

- Fixate $k \in \mathbb{N}$
- Atomics are defined like in natural extension to non-relational signatures, with one restriction
- For every formula in GF(C)$_k$ and every term t that appears in it, there must exist a $n \in \mathbb{N}$, such that $t = \alpha$ for a $\alpha \in \mathrm{Alters}_n^k(\sigma)$

- Restrict number of alternations of function applications to k
- No restriction of number of application of same function in series
- Examples:
  - $f^2(g(h^3(x))) = y \notin \mathrm{GF(C)}_2$, but in $\mathrm{GF(C)}_3$
  - $f^i(x) = y \in \mathrm{GF(C)}_1$ for all $i \in \mathbb{N}$

Characterising RCR_k Logically ii

Logical Characterisation of RCR_k

Let $k \in \mathbb{N}$ and let $\mathfrak{A}$ and $\mathfrak{B}$ be two structures. Then the two following statements are equivalent.

1. RCR_k distinguishes $\mathfrak{A}$ and $\mathfrak{B}$
2. There exists a sentence in $\mathrm{GF(C)}_k$ that is fulfilled by $\mathfrak{A}$, but not by $\mathfrak{B}$

Characterising RCR_k Logically iii

Proof idea

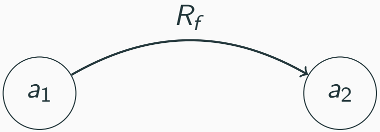
- 1. to 2.: Like, before sentence in GF(C) over signature $\widetilde{\sigma}$ can easily be translated into sentence in $\mathrm{GF(C)}_k$ over signature σ
- 2. to 1.:
  - Assume $n = |\mathfrak{A}| = |\mathfrak{B}|$
  - Translate and replace all atomic subformulae by formula that:
    - is equivalent for all structures with n elements
    - only contains terms $f^i(s(x))$ with $i \leq n$
  - Rearrange resulting formula to get valid $\mathrm{GF(C)}_k$-sentence
  - Results in equivalent formula for structures with n elements and for every term t there exists an $\alpha \in \mathrm{Alters}_n^k(\sigma)$, such that $t = \alpha$
  - Can easily be translated into sentence in GF(C) of signature $\widetilde{\sigma}$

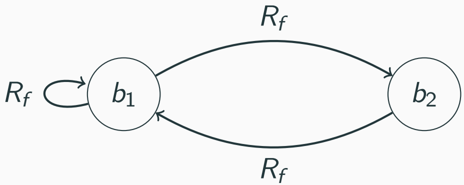
Relational Colour Refinement for Structures With Functions

Discussion on the Combinatorial Characterisation

Total and Functional Structures

- Let σ be a signature, σ' its naive encoding and $\mathfrak{A}'$ a σ'-structure
- We call $\mathfrak{A}'$ total if for every n-ary function symbol $f \in \sigma$ and every n-tuple $\mathbf{x}$ there is a y, such that $(\mathbf{x}y) \in R_f^{\mathfrak{A}'}$
- We call $\mathfrak{A}'$ functional if for every n-ary function symbol f there are no two $n+1$-tuples $(\mathbf{x}y), (\mathbf{x}z) \in R_f^{\mathfrak{A}'}$





Non-Relational Acyclic Structures

- Will define acyclicity w.r.t. the naive encoding

Non-Relational Acyclic Structures

- Let $\mathfrak{A}$ be a non-relational structure
- We call $\mathfrak{A}$ acyclic, if its naive encoding $\mathfrak{A}'$ is acyclic

Total and Functional Structures as Encodings

- Desired equivalence:

Non-relational, acyclic structure distinguishes $\mathfrak{A}$ and $\mathfrak{B}$ by homomorphism count

?

Naive RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$

- Result: Forward direction holds, backwards does not
- First step: Reformulate first statement:

Some non-relational, acyclic structure dist. $\mathfrak{A}$ and $\mathfrak{B}$ by hom. count

iff.

Some total, functional and acyclic structure dist. encodings $\mathfrak{A}'$ and $\mathfrak{B}'$ by hom. count

Enforcing Functionality

- We can show:

$$\text{Acyclic } \sigma'\text{-structure dist. } \mathfrak{A}' \text{ and } \mathfrak{B}' \text{ by hom. count}$$
$$\text{iff.}$$
$$\text{Functional and acyclic } \sigma'\text{-structure dist. } \mathfrak{A}' \text{ and } \mathfrak{B}' \text{ by hom. count}$$

Proof idea:

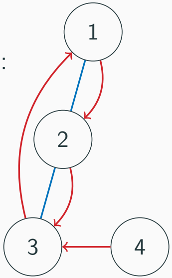
- Backwards direction is obvious
- Forwards direction eliminates collisions of the form $(\mathbf{x}y), (\mathbf{x}z) \in R_f$ by contracting y and z
- This can be done while maintaining the homomorphisms and acyclicity and can be repeated until no collisions remain

Non-Enforceability of Totality

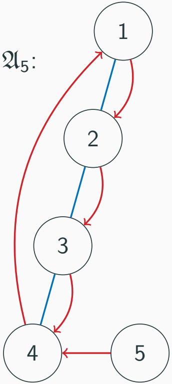
- There are structures that are distinguished by naive RCR, but there is no acyclic and total structure that distinguishes the encodings by homomorphism count
- Define signature $\sigma = \{E/2, f/1\}$
- Two families of σ-structures $(\mathfrak{A}_i)_{i \in \mathbb{N}_{\geq 4}}$ and $(\mathfrak{B}_i)_{i \in \mathbb{N}_{\geq 4}}$
- For all $i \in \mathbb{N}_{\geq 4}$: Naive RCR distinguishes $\mathfrak{A}_i$ and $\mathfrak{B}_i$, but no acyclic and total structure can distinguish the encodings by hom. count

Structures that are distinguished by nRCR but not total structures i

$\mathfrak{A}_4$:

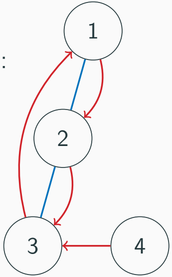


$\mathfrak{A}_5$:

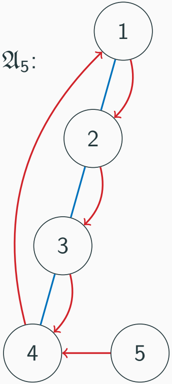


Structures that are distinguished by nRCR but not total structures i

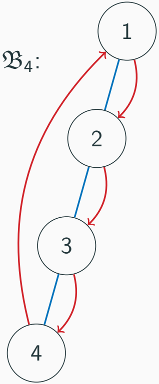
$\mathfrak{A}_4$:



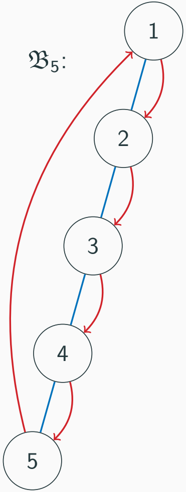
$\mathfrak{A}_5$:



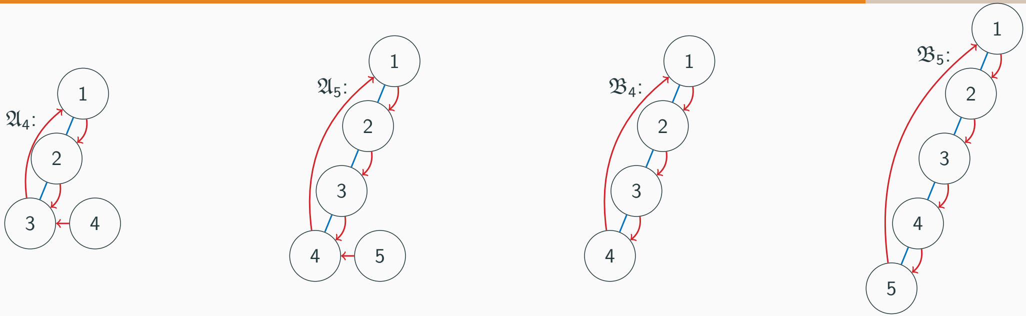
$\mathfrak{B}_4$:



$\mathfrak{B}_5$:



Structures that are distinguished by nRCR but not total structures ii



- Obviously distinguished by naive RCR
- If structure has R_f-loops or R_f-2-cycles, then no homomorphisms to either structure
- Because total, it has to contain larger R_f-cycles, but then cannot be acyclic

Results of combinatorial characterisation of naive RCR

We have the following results:

Naive RCR distinguishes $\mathfrak{A}$ and $\mathfrak{B}$

$$\Updownarrow$$

There exists acyclic structure that dist. encodings $\mathfrak{A}'$ and $\mathfrak{B}'$ by hom. count

$$\Updownarrow$$

There exists acyclic and functional structure that dist. encodings by hom. count

$\Uparrow$, but $\not\Downarrow$

There exists acyclic, total and functional structure that dist. encodings by hom. count

$$\Updownarrow$$

There exists acyclic, non-relational structure that dist. $\mathfrak{A}$ and $\mathfrak{B}$ by hom. count

RCR on Subclasses of Relational Structures

Restricting the Class of Structures

- For what subclass $\mathcal{S}$ of relational structures do we have the following equivalence:

  Two structures from $\mathcal{S}$ get distinguished by RCR

  iff.

  There exists an acyclic structure from $\mathcal{S}$ that dist. the structures by hom. count
- Does not hold for class of total structures
  - Encodings of classes of structures from before are total, but no total and acyclic structure dist. them by hom. count
- Another class to investigate: Class of symmetric structures

Restriction to Symmetric Structures

- Relational Structure is symmetric, if for every k-ary relation R and for every k-tuple $\mathbf{x} \in R$, every permutation of the elements in $\mathbf{x}$ is also in R

Restriction to Symmetric Structures

- Relational Structure is symmetric, if for every k-ary relation R and for every k-tuple $\mathbf{x} \in R$, every permutation of the elements in $\mathbf{x}$ is also in R
- For two symmetric structures we can show

$$\text{There exists acyclic structure dist. the structures by hom. count}$$
$$\text{iff.}$$
$$\text{There exists acyclic, symmetric structure dist. the structure by hom. count}$$

- From this, restriction to symmetric structures is possible

Sketch of a Proof

Description of the Lemma

- Lemma for translating $f^m(x_1) = x_2$ to a formula with a bounded number of applications of f in series
- Used in proof of logical characterisation of RCR_k

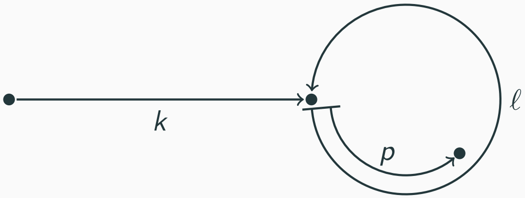
Lemma 4.6

A formula ψ of the form $f^m(x_1) = x_2 \in \mathrm{GF(C)}_1$ can be translated to a formula $\vartheta(x_1, x_2) \in \mathrm{GF(C)}_1$, such that:

1. They are equivalent for structures with n elements
2. There does not appear a term f^i with $i > n$ in ϑ
3. ϑ is of the form $\bigvee \Phi$ and if ϑ is fulfilled, then there exists exactly one $\varphi \in \Phi$ which is satisfied

Proof Idea

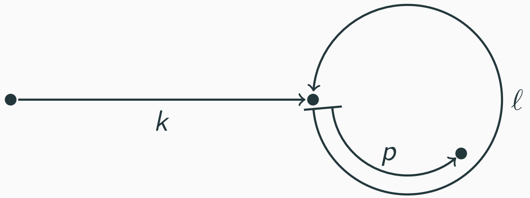
- Elements $f^0(x), f^1(x), \dots, f^m(x)$ describe a path through a structure
- If $m > n$, there have to be $i, j \leq n$ such that $f^i(x) = f^j(x)$
- Path can be decomposed into:
  1. Path to a cycle
  2. A cycle
  3. A last part of the cycle
- Define set $\mathcal{I}(n, m)$ as set of all such decomposition (k, ℓ, p)



Sketch of the Proof i

- Define $\vartheta(x_1, x_2) \coloneqq \bigvee_{(k,\ell,p)\in\mathcal{I}(n,m)} \zeta_{(k,\ell,p)}(x_1, x_2)$ where

$$\begin{aligned}\zeta_{(k,\ell,p)}(x_1, x_2) \coloneqq & f^{k+p}(x_1) = x_2 \wedge f^k(x_1) = f^{k+\ell}(x_1) \\ & \wedge f^{k-1}(x_1) \neq f^{k-1+\ell}(x_1) \\ & \wedge \bigwedge_{0<\ell'<\ell} f^k(x_1) \neq f^{k+\ell'}(x_1)\end{aligned}$$



Sketch of the Proof ii

- $f^{k+p}(x_1) = x_2 \wedge f^k(x_1) = f^{k+\ell}(x_1)$ ensures that (k, ℓ, p) decomposes the path into a path to a cycle and the cycle itself
- $f^{k-1}(x_1) \neq f^{k-1+\ell}(x_1) \wedge \bigwedge_{0<\ell'<\ell} f^k(x_1) \neq f^{k+\ell'}(x_1)$ ensures that only the lexicographically smallest decomposition is satisfied

- If ψ is satisfied, a smallest decomposition (k, ℓ, p) exists that decomposes the path of f
- Then it can be shown that $\zeta_{(k,\ell,p)}$ is satisfied, and because only the lexicographically smallest (k, ℓ, p) is satisfied, it is the only one

Sketch of the Proof iii

- If ϑ is satisfied, some $\zeta_{(k,\ell,p)}$ is satisfied
- This means that (k, ℓ, p) decomposes the path of f, therefore ψ is also satisfied

Conclusion

Conclusion

- We presented classical CR and Scheidt's and Scheikardt's RCR algorithm
- We defined two possible ways to apply their algorithm to non-relational signatures
  - Naive RCR
  - RCR_k
- We showed our results for the logical characterisations
  - Naive RCR gets characterised by the nesting free fragment of counting logic
  - RCR_k gets characterised by the natural extension of GF(C) to non-relational signatures where terms have a maximal alternation depth of k
- We disproved the characterisation by homomorphism counting
  - Functionality can be enforced
  - Totality cannot
- We showed results for the restriction to two subclasses of the relational structures
  - The restriction to total structures does not preserve the characterisation by hom. counting
  - The restriction to symmetric structures does preserve it

Equality between terms t and alternations α

- For a term t and a $\alpha \in \mathsf{Alters}_n^k(\sigma)$ we say $t = \alpha$, if:
- If $t = f^i(x)$, the i-times application of one function symbol f, and $\alpha = f^i$
- If $t = f^i(g^j(s(x)))$, where f and g are function symbols and s is a term, and $\alpha = f^i\alpha'$ and $s = \alpha'$
- Informally, if t is written using $\circ$, i.e. $f^i \circ g^j(x)$ instead of $f^i(g^j(x))$, the $\circ$ are omitted and then this equals α