Ptime Canonization for Two Variables with Counting

M. Otto
Mathematische Grundlagen der Informatik
RWTH Aachen
52074 Aachen, Germany

Abstract

We consider infinitary logic with two variable symbols and counting quantifiers, C^2, and its intersection with PTIME *on finite relational structures. In particular we exhibit a* PTIME *canonization procedure for finite relational structures which provides unique representatives up to equivalence in C^2. As a consequence we obtain a recursive presentation for the class of all those queries on arbitrary finite relational structures which are both* PTIME *and definable in C^2. The proof renders a succinct normal form representation of this non-trivial semantically defined fragment of* PTIME*. Through specializations of the proof techniques similar results apply with respect to the logic L^2, infinitary logic with two variable symbols, itself.*

1 Introduction and main results

Canonization addresses problems of the following kind: Given a notion of equivalence between relational structures of a certain type, is there an algorithm that determines unique representatives for each equivalence class? To be more explicit fix a relational vocabulary τ and let $\mathfrak{fin}[\tau] := \{\mathfrak{A} \mid \mathfrak{A} \text{ a finite } \tau\text{-structure}\}$. Denote the equivalence relation under consideration by $\sim$. Then we are looking for a functor which maps structures in $\mathfrak{fin}[\tau]$ to structures in $\mathfrak{fin}[\tau]$ over suitable standard domains:

$$H : \mathfrak{fin}[\tau] \longrightarrow \{\mathfrak{A} \in \mathfrak{fin}[\tau] \mid \mathrm{dom}(\mathfrak{A}) = n, n \geqslant 1\}^1$$

$$\text{satisfying:} \quad \begin{array}{l} \forall \mathfrak{A} \quad H(\mathfrak{A}) \sim \mathfrak{A} \quad \text{and} \\ \forall \mathfrak{A}, \mathfrak{A}' \quad \mathfrak{A} \sim \mathfrak{A}' \Rightarrow H(\mathfrak{A}) = H(\mathfrak{A}'). \end{array}$$

The standard case is that of *canonization up to isomorphism.* For finite graphs say, one such canonization is provided by the functor that maps each finite graph (V, E) of size $|V| = n$ to that graph with node set n that has the lexicographically least adjacency matrix among all those isomorphic with (V, E). While this canonization can obviously be done in PSPACE – in fact it is in the second level of the polynomial hierarchy Δ_2^p – it remains open whether canonization up to isomorphism can be achieved in PTIME. On the other hand, PTIME canonization bears a close relationship with recursive presentability of semantically defined fragments of PTIME on finite relational structures. The class of all PTIME boolean *queries* for finite τ-structures is such a fragment: It can be identified with the class $\{Q \subseteq \mathfrak{fin}[\tau] \mid Q \in \text{PTIME},\ Q \simeq\text{-closed}\}$. For an example consider the class of all PTIME graph properties. Closure with respect to isomorphism is the obvious requirement that a graph property may not evaluate to true on one graph and to false on an isomorphic copy of that same graph. This closure condition makes the class of PTIME boolean queries a typical *semantically defined class.* It is one of the major open problems in finite model theory whether this class admits a recursive presentation. Recursive presentability of this class is in fact equivalent with the existence of a *logic for* PTIME in the precise sense formulated e.g. in [7].

The following simple observation concerns the connection between PTIME canonization up to $\sim$ and recursive presentations of related fragments of PTIME.

Fact 1.1 *If there is a* PTIME *functor H that provides canonization for finite τ-structures up to $\sim$, then the following class of boolean queries is recursively presentable:*

$$\text{PTIME}^{\sim}_{\tau} := \{Q \subseteq \mathfrak{fin}[\tau] \mid Q \in \text{PTIME},\ Q \sim\text{-}closed\}.$$

In fact this class then coincides with the class of all those queries that can be evaluated through an application of a PTIME *algorithm to the representatives provided by H. We write* PTIME $\circ H$ *for this latter class.*

It is important to note that in PTIME $\circ H$, PTIME is safely represented by the class of all polynomi-

[1] For ease of notation we employ the standard set theoretic convention to identify n with the set $\{0, \ldots, n-1\}$.

ally clocked Turing machines, since there are no additional semantic restraints. In the situation described one may further obtain a logical characterization of $\mathrm{PTIME}_\tau^\sim$ as an immediate consequence of the well-known theorem of Immerman [9] and Vardi [16]: The class of queries in $\mathrm{PTIME} \circ H$ coincides with the class of queries that are definable in fixed-point logic in terms of the naturally ordered images under H.

All known positive results concerning the capturing of standard complexity classes below NP concern queries for linearly ordered structures, i.e. they assume the presence of a linear order in the input structures. The choice of a unique representative for any linearly ordered structure $\mathfrak{A}$ is trivial: The natural representative is determined by the requirement that the given ordering on $\mathfrak{A}$ translates into the natural ordering on the standard domain. While this kind of presentation usually is made part of the input conventions, it is instructive to note that canonization up to isomorphism is in LOGSPACE for linearly ordered structures. The difficulties, on the other hand, in capturing any of the standard complexity classes below NP for arbitrary not necessarily ordered structures is due to the inability so far to isolate the isomorphism preserving fragment of these classes when applied to unordered structures. In view of Fact 1.1, canonization is a means to *introduce order* in this case; if canonization is with respect to a coarser equivalence relation than isomorphism, however, we do not capture the entire class of all PTIME queries but only the subclass of correspondingly coarser queries within PTIME. In this paper we show that this approach is successful for the equivalence relations derived from the two-variable fragments of infinitary logic. To our knowledge this is the first result concerning recursive presentations for a semantically defined fragment of any complexity class below NP in the absence of order.

We write $L^k := L^k_{\infty\omega}$ for the infinitary variant of first-order logic that allows conjunctions and disjunctions over arbitrary sets of formulae, with the restriction that only k different variable symbols occur in each formula. Counting quantifiers, i.e. quantifiers of the type $\exists^{\geqslant m}$ asserting the existence of at least m elements possessing some property, are of course first-order definable. However, with only k variable symbols, no counting quantifier $\exists^{\geqslant m}$ for $m > k$ is expressible. Hence the following extensions of the L^k are natural. Let $C^k := L^k_{\infty\omega}(\exists^{\geqslant m})_{m\in\omega}$ be the extension obtained from L^k if instead of the ordinary existential quantifier $\exists$ all counting quantifiers $\exists^{\geqslant m}$ are permitted, with restrictions on the variable symbols just as for L^k.

These logics are important in finite model theory, since they comprise as sublogics the most prominent extensions of first-order logic for the purposes of modelling recursion on finite structures, the extensions by various fixed-point operators. These are all sublogics of $L^\omega := \bigcup_k L^k$, infinitary logic with finitely many variables in each formula. Similarly, C^ω, the union of the C^k comprises the further extensions of these fixed-point logics that incorporate counting in the natural manner, cf. [10, 6, 13]. These containments are particularly useful since the expressive power of the logics L^k and C^k can be analyzed very effectively through Ehrenfeucht-Fraïssé style games.

We write $\equiv_{L^k}$ and $\equiv_{C^k}$ for the associated notions of equivalence: $\mathfrak{A} \equiv_{L^k} \mathfrak{A}'$ if $\mathfrak{A}$ and $\mathfrak{A}'$ are indistinguishable in the logic L^k, i.e. if they satisfy exactly the same sentences of L^k; similarly for $\equiv_{C^k}$ and C^k. Since e.g. C^k is closed under countable disjunctions and conjunctions, it is immediate that a query $Q \subseteq \mathfrak{fin}[\tau]$ is closed w.r.t. $\equiv_{C^k}$ if and only if it is definable in C^k. Thus the class $\mathrm{PTIME}^{\equiv_{C^k}}$, defined according to Fact 1.1 above, coincides with the class of all those PTIME boolean queries that are C^k-definable. The same applies to L^k:

Definition 1.2 *Let* $\mathrm{PTIME}\cap C^k$ *and* $\mathrm{PTIME}\cap L^k$ *stand for the classes of those* PTIME *boolean queries that are* C^k*- or* L^k*-definable (closed w.r.t.* $\equiv_{C^k}$ *or* $\equiv_{L^k}$*).*

Observe that these classes are indeed non-trivial semantic classes. Definability in L^k or C^k is completely at odds with complexity considerations: Even L^2 suffices to define non-recursive classes of graphs; we give an example in Section 5.

Definition 1.3 *We say that* C^k admits PTIME canonization *if there is, for each finite relational type* τ*, a* PTIME *functor* H *for canonization up to* $\equiv_{C^k}$*:* H *sends finite* τ*-structures to* τ*-structures over a standard universe* n*,* $n \geqslant 1$*, such that:*

$$\begin{array}{l} \forall \mathfrak{A} \quad H(\mathfrak{A}) \equiv_{C^k} \mathfrak{A} \ \ \textit{and} \\ \forall \mathfrak{A}, \mathfrak{A}' \quad \mathfrak{A} \equiv_{C^k} \mathfrak{A}' \Rightarrow H(\mathfrak{A}) = H(\mathfrak{A}'). \end{array}$$

Similarly, L^k *admits* PTIME *canonization if there exists a functor* K *with analogous properties for* L^k*.*

Theorem 1.4 *Let* C^k *or* L^k *admit* PTIME *canonization through* H *or* K*, respectively. Then*

$$\begin{array}{rcl} \mathrm{PTIME} \cap C^k & \equiv & \mathrm{PTIME} \circ H\,, \ \ \textit{or} \\ \mathrm{PTIME} \cap L^k & \equiv & \mathrm{PTIME} \circ K\,, \ \ \textit{respectively.} \end{array}$$

On the right-hand sides, $\mathrm{PTIME} \circ H$ and $\mathrm{PTIME} \circ K$ stand for the class of *all* PTIME algorithms applied to images under H or K, respectively. In particular, the right-hand classes of queries are recursively enumerable. The present paper gives proof that indeed at least for $k = 2$ there are such canonizations H and K, so that we find that the C^2- and L^2-fragments of PTIME are recursively enumerable and admit a normal form of the indicated kind. We do get a little more than that even, owing to the manner in which we obtain these canonizations. Through the work of Abiteboul and Vianu, [1], L^k is known to possess concise *invariants*, that characterize structures up to $\equiv_{L^k}$. These PTIME functors I^k map finite τ-structures to linearly ordered structures of suitable type, such that for any two finite τ-structures $\mathfrak{A}$ and $\mathfrak{A}'$, $I^k(\mathfrak{A}) = I^k(\mathfrak{A}') \Leftrightarrow \mathfrak{A} \equiv_{L^k} \mathfrak{A}'$. Furthermore, fixed-point queries exactly correspond to the PTIME properties of the images under the I^k. See [1], and in particular [4] and [5] for the logical interpretation and the connection with L^k. For C^k analogous invariants I^k_C are introduced in [13] and applied to the analysis of fixed-point logic with counting, see also [6]. In either case, the invariants can be pictured as ordered quotients of the k-th power of $\mathfrak{A}$ with respect to equality of L^k- or C^k-types.

Comparing the properties of the invariants with the requirements of Definition 1.3 we see that one way to realize canonization is to provide *inverses to the invariants* in the sense of providing, e.g. in the case of C^k, a PTIME functor F such that for all finite $\mathfrak{A}$: $F\big(I^k_C(\mathfrak{A})\big) \equiv_{C^k} \mathfrak{A}$. With such an inverse it follows that $H := F \circ I^k_C$ gives PTIME canonization. Our main result yields such inverses for $k = 2$:

Main Theorem *There are* PTIME *functors yielding*

- *for any $\mathfrak{I}$ in the range of I^2_C a τ-structure $\mathfrak{A}$ such that $I^2_C(\mathfrak{A}) = \mathfrak{I}$.*
- *for any $\mathfrak{I}$ in the range of I^2 a τ-structure $\mathfrak{A}$ such that $I^2(\mathfrak{A}) = \mathfrak{I}$.*

In particular, the ranges of the invariants I^2_C and I^2 are PTIME *recognizable.*

Corollary *C^2 and L^2 admit* PTIME *canonization.*

Corollary PTIME $\cap\, L^2$ *and* PTIME $\cap\, C^2$ *are recursively enumerable and admit the following presentations:*

$$\begin{aligned}\text{PTIME} \cap C^2 &\equiv \text{PTIME} \circ I^2_C\\ \text{PTIME} \cap L^2 &\equiv \text{PTIME} \circ I^2.\end{aligned}$$

These normal forms indicate that in computing C^2- and L^2-definable queries, Turing machines can make only very limited computational use of the ordering they get in their input representation. Consider e.g. a query Q in PTIME $\cap\, L^2$: On the one hand there is a PTIME Turing machine that accepts exactly all ordered representations of structures in Q; on the other hand Q is L^2-definable, which in itself does not have any relevance for complexity matters. The combination of these two conditions, however, forces the existence of a PTIME algorithm recognizing $\big\{I^2(\mathfrak{A}) \,\big|\, \mathfrak{A} \in Q\big\}$. In a sense that is apparent from the precise definition of the invariants, this latter algorithm uses no more linear order than is L^2-definably available in the input structures!

We point out that both the canonization problem and the related inversion problem for the invariants were outlined as open problems for L^k in A. Dawar's thesis [4].

The problem of finding a structure $\mathfrak{A}$ that fits a given value of the invariants is here reduced to the purely combinatorial problem of constructing certain *colourings of square grids*. An analysis of their combinatorial properties leads to necessary conditions on the invariants which ultimately allow to reduce the construction to modular sub-problems reminiscent of magic square puzzles. The flavour of these is exemplified by the following:

For even n, colour the $n \times n$ square with n colours in such a way that one of the diagonals is monochromatic, the entire colouring is mirror symmetric with respect to this diagonal, and in each row and each column, each colour occurs exactly once.

As we have indicated above, the issues of finding PTIME canonizations for L^k or C^k, or of inverting the invariants I^k or I^k_C, make sense for arbitrary k. The specialization to $k = 2$ in the results of this paper reflects the fact that this is the only case we can cope with combinatorially. It remains a challenging open problem to settle the corresponding questions for arbitrary k.

We close this introductory section with some remarks concerning the special nature of the two-variable case. As noted above, L^2 is sufficiently expressive to define non-trivial – and even non-recursive – classes of finite structures . In some other respects, however, the case of only two variables is very special: The satisfiability problem for first-order sentences with at most two variable symbols over relational languages is decidable [12, 15]. In fact Mortimer shows in [12] that first-order logic with only two variables has the finite model property: Any satisfiable finitary sentence of L^2 has a finite model [12]. Neither of these facts remains true for more than 2 variables. Obviously, 3 variables suffice to axiomatize linear orderings without endpoints. Similarly 3 variables allow to axiomatize rectangular grid graphs which form the basis for a reduction of the domino problem to the satisfiability problem for finitary L^3, which is thus proved undecidable. We note that Mortimer's results guarantee recursive canonization up to $\equiv_{L^2}$. While a closer analysis of the size bound in [12] yields canonization in EXPTIME, the more explicit model constructions presented here seem to be essential for PTIME canonization. The present model constructions for L^2 also imply that any L^2-theory with finite models has a model of size that is linearly bounded by the number of distinct L^2-types required. This is yet another phenomenon peculiar to the case of only two variables.

C^2 is a much more powerful logic than L^2. It is not difficult to see that a first-order sentence using 2 variables together with just the counting quantifiers $\exists^{\geqslant 1}$ and $\exists^{\geqslant 2}$ suffices to axiomatize classes of necessarily infinite models: Take the sentence asserting that a binary relation is the graph of an injective but non-surjective function. It is not known whether the satisfiability problem for first-order sentences of C^2 is decidable. The expressive power of C^2 goes far beyond that of L^2 also for finite structures. Most notably, the *stable colouring of graphs* is C^2-definable [10]. Natural derived classes of graphs that are definable in C^2 are the classes of colour class size l for each natural number $l \geqslant 1$. By results of Babai and Kučera, [2], the class of colour size 1 graphs is dense in the class of all finite graphs, which further implies that almost all finite graphs are characterized even up to isomorphism by their C^2-theory.

In this condensed version we explicitly treat the case with counting quantifiers, that also is the technically more interesting. The results for L^2 are largely obtained through a specialization of the methods applied to C^2, and will only be summed up in Section 5. In the development of the problem in the following section we shall, however, appeal to the case of arbitrary k were this is natural.

2 Basic definitions and preliminaries

2.1 Games and invariants for the C^k

The expressive power of the logics L^k and C^k can very elegantly be characterized in Ehrenfeucht-Fraïssé style through appropriate games. The game for L^k is the usual *k-pebble game*, that goes back (in the equivalent but different formulation in terms of back-and-forth systems) to Barwise [3], and in the game formulation is due to Immerman [8], see also [14]. The variant of these games to incorporate counting, the *k-pebble game with counting* was introduced by Immerman and Lander in [10]. We here relate the games to PTIME invariants that classify finite relational structures up to L^k- or C^k-equivalence, respectively. The fundamental concept of such invariants was put forward in the work of Abiteboul and Vianu [1], where objects essentially equivalent with the L^k-invariants were introduced in the framework of relational computation and put to great use in understanding fixed-point and partial fixed-point logic. In the present logical framework they were formalized for L^k in [4] and introduced for C^k in [6, 13].

We fix a finite relational vocabulary τ; $\mathfrak{fin}[\tau]$ is the class of all finite τ-structures. For $\mathfrak{A} \in \mathfrak{fin}[\tau]$ and a finite tuple $\overline{a}$ from its universe A, let $\mathrm{atp}_{\mathfrak{A}}(\overline{a})$ denote the *atomic type* of $\overline{a}$ in $\mathfrak{A}$: The set of all atomic or negated atomic formulae satisfied by $\overline{a}$ in $\mathfrak{A}$. Let $\mathrm{tp}_{\mathfrak{A}}^{C^k}(\overline{a})$ denote the full C^k*-type* of $\overline{a}$ in $\mathfrak{A}$: The class of all C^k-formulae satisfied by $\overline{a}$ in $\mathfrak{A}$. As usual we extend the notation $\equiv_{C^k}$ to apply to tuples over structures: $(\mathfrak{A}, \overline{a}) \equiv_{C^k} (\mathfrak{A}', \overline{a}')$ is shorthand for the equality $\mathrm{tp}_{\mathfrak{A}}^{C^k}(\overline{a}) = \mathrm{tp}_{\mathfrak{A}'}^{C^k}(\overline{a}')$.

We explicitly treat the case of C^k. For the details of the game we refer to [10] or [6, 13]; here we can only provide a brief reminder intended to give some intuition for the treatment of the invariants. The C^k-game is played by two players, **I** and **II**, on two structures $\mathfrak{A}$ and $\mathfrak{A}'$. In a typical position k elements in each structure are marked by pebbles numbered 1 to k. Let $\overline{a}$ and $\overline{a}'$ denote the corresponding tuples of currently pebbled elements in $\mathfrak{A}$ and $\mathfrak{A}'$, respectively. The entire position is denoted $(\mathfrak{A}, \overline{a}; \mathfrak{A}', \overline{a}')$. The task of player **II** is to maintain *partial isomorphism* between the currently pebbled tuples of elements: $\mathrm{atp}_{\mathfrak{A}}(\overline{a}) = \mathrm{atp}_{\mathfrak{A}'}(\overline{a}')$. In each round of the game, player **I** challenges this equality in an exchange of moves that involve choices of equal size sets (these serve to incorporate cardinality restraints) and lead to the relocation of exactly one pair of corresponding pebbles. More precisely, a single round consists of 4 steps:
— **I** chooses $j \in \{1, \dots, k\}$ and a subset of one of the structures, say $B \subseteq A$;
— **II** has to designate a subset of the other structure of exactly the same size, here $B' \subseteq A'$;
— now **I** moves pebble j in $\mathfrak{A}'$ to any point within B';
— finally **II** moves the corresponding pebble to a point within B.

We say that **II** has a *winning strategy* in the game if, for any choice of moves for the opponent, **II** can maintain partial isomorphism indefinitely. The rules for the individual rounds are such that **II** has a winning strategy in position $(\mathfrak{A},\overline{a};\mathfrak{A}',\overline{a}')$ if and only if $(\mathfrak{A},\overline{a}) \equiv_{C^k} (\mathfrak{A}',\overline{a}')$, i.e. if the current pebble positions are indistinguishable in C^k. A strategy for **II** therefore must consist in maintaining equality of C^k-types.

Theorem 2.1 (Immerman, Lander)
*Player **II** has a winning strategy in the game starting from $(\mathfrak{A},\overline{a};\mathfrak{A}',\overline{a}')$ if and only if $(\mathfrak{A},\overline{a}) \equiv_{C^k} (\mathfrak{A}',\overline{a}')$.*

The crucial property of the C^k-types, also reflected in the game, is the following: $(\mathfrak{A},\overline{a}) \equiv_{C^k} (\mathfrak{A}',\overline{a}')$ holds if $\mathrm{atp}_{\mathfrak{A}}(\overline{a}) = \mathrm{atp}_{\mathfrak{A}'}(\overline{a}')$ and for all C^k-types α:

$$\begin{aligned}&\left|\left\{b \in A \mid \mathrm{tp}^{C^k}_{\mathfrak{A}}(\overline{a}\tfrac{b}{j}) = \alpha\right\}\right| \\ &= \left|\left\{b' \in A' \mid \mathrm{tp}^{C^k}_{\mathfrak{A}'}(\overline{a}'\tfrac{b'}{j}) = \alpha\right\}\right|. \quad ([2])\end{aligned}$$

Also observe that $\mathfrak{A} \equiv_{C^k} \mathfrak{A}'$ if and only if $\mathfrak{A}$ and $\mathfrak{A}'$ realize exactly the same C^k-types. The invariants for C^k can be abstracted from the C^k-game. They represent concisely the complete C^k-theory of structures – equivalently, all the structural information that is relevant in the C^k-game. Let $I^k_C(\mathfrak{A})$ stand for the invariant of $\mathfrak{A}$ w.r.t. C^k. $I^k_C(\mathfrak{A})$ is a presentation of the quotient of $\mathfrak{A}$ w.r.t. the equivalence relation $\equiv_{C^k}$ on the k-th power of the universe. The backbone of $I^k_C(\mathfrak{A})$ is formed by a *linearly ordered* representation of the equivalence classes in $A^k/\equiv_{C^k}$. The crucial point is that this quotient along with its linear ordering $\prec$ can inductively be generated in polynomially many steps. This ordered quotient $(A^k/\equiv_{C^k},\prec)$ is endowed with sufficient additional structure so that the entire C^k-theory of $\mathfrak{A}$ can be retrieved. The extra data are:
— Reachability: For each j, $1 \leqslant j \leqslant k$, a binary predicate E_j containing exactly those pairs (α_1,α_2) from $A^k/\equiv_{C^k}$ for which $\overline{a} \in \alpha_1 \Rightarrow \exists b\, \overline{a}\frac{b}{j} \in \alpha_2$.
— Symmetries: For each permutation $\sigma \in S_k$ a unary function mapping α to the equivalence class of $\sigma(\overline{a})$, for $\overline{a} \in \alpha$. We also denote this function by σ.
— Atomic types: A mapping $\Theta\colon A^k/\equiv_{C^k} \to \mathfrak{atp}$, that associates with each class $\alpha \in A^k/\equiv_{C^k}$ the atomic type of its members.
— Multiplicities: For each j, $1 \leqslant j \leqslant k$, a weight function ν_j from $A^k/\equiv_{C^k}$ to positive natural numbers which sends α to $\left|\left\{b \in A \mid \overline{a}\frac{b}{j} \in \alpha\right\}\right|$, for $\overline{a} \in \alpha$.

It is checked that these definitions are sound, i.e. that they are independent of choices of representatives $\overline{a} \in \alpha$. We eliminate some redundancy in these data in retaining only one of the E_j, $E := E_k$ say, and only one of the multiplicities ν_j, $\nu := \nu_k$. E.g. ν_1 can be recovered as $\nu_1 = \nu_k \circ (1,k)$, where $(1,k)$ is the transposition exchanging 1 and k.

Definition 2.2 *Let for each k and each fixed finite relational signature τ, the C^k-invariant I^k_C be the functor which sends a finite τ-structure $\mathfrak{A}$ to the weighted linearly ordered structure*

$$I^k_C(\mathfrak{A}) \;=\; \Big(A^k/\equiv_{C^k},\prec,E,(\sigma)_{\sigma\in S_k};\Theta;\nu\Big).$$

Observe that the values of ν over $A^k/\equiv_{C^k}$ are bounded by $|A|$. $I^k_C(\mathfrak{A})$ can thus in a canonical way be encoded as a relational structure over the numerical universe $\{0,\dots,|A|\}$. In this sense $|A|+1$ is regarded as the size of $I^k_C(\mathfrak{A})$. The following is proved in [6, 13]:

Theorem 2.3 *I^k_C is* Ptime *computable, in fact it can uniformly be defined in a fragment of fixed-point logic with counting. I^k_C classifies finite relational structures up to $\equiv_{C^k}$: $\quad I^k_C(\mathfrak{A}) = I^k_C(\mathfrak{A}')$ iff $\mathfrak{A} \equiv_{C^k} \mathfrak{A}'$.*

Recall from the introduction the inversion problem for I^k_C. We say that I^k_C *admits* Ptime *inversion* if there is a Ptime functor F such that for all $\mathfrak{A}$: $F\big(I^k_C(\mathfrak{A})\big) \equiv_{C^k} \mathfrak{A}$. The following is a restatement of Fact 1.1 in the particular case in which canonization is through inversion of the invariant.

Theorem 2.4 *If I^k_C admits* Ptime *inversion through F, then C^k admits* Ptime *canonization through $F \circ I^k_C$ and* Ptime $\cap\, C^k$ *has the following normal form:*

$$\textsc{Ptime} \cap C^k \;\equiv\; \textsc{Ptime} \circ I^k_C.$$

3 Game tableaux and inversion for I^2_C

3.1 Reduction to game tableaux

Consider the format of the C^k-invariant for $k = 2$. Of the action of the symmetric group only the transposition that exchanges first and second components remains to be considered. Let T stand for both, this transposition itself and its operation on the $\equiv_{C^2}$-classes. Recall that reachability for the first component can be recovered from E as E^T: $(\alpha,\alpha') \in E^T :\Leftrightarrow (T\alpha,T\alpha') \in E$. Similarly, multiplicity in the first component is recovered from ν with $\nu^T := \nu \circ T$. We now separate the equality type from the rest of the relational atomic type in order to isolate the basic combinatorial aspects of the game. As there are only two different equality types in two variables we introduce just one unary predicate Δ associated with the equality type $x_1 = x_2$ of the diagonal: $\alpha \in \Delta$ if

[2] We write $\overline{a}\frac{b}{j}$ for the tuple $\overline{a}$ with its j-th component replaced by b.

$(x_1 = x_2) \in \Theta(\alpha)$. So rather than the above presentation, we now use the following equivalent format:

$$I_C^2(\mathfrak{A}) \;=\; \Big(A^2/{\equiv_{C^2}}, \prec, E, T, \Delta; \Theta; \nu\Big).$$

The following definition of *weighted game tableaux* abstracts the combinatorial core of structures like the C^2-invariants: We disregard the association with relational atomic types. The conditions in the definition state some simple facts about real invariants.

Definition 3.1 *A* game tableau *is a finite structure* $\mathcal{Q} = (Q, \prec, E, T, \Delta)$, *linearly ordered by* $\prec$, *and satisfying the following:*

(i) E *is an equivalence relation on* Q.

(ii) T *is an involution on* Q, *i.e.* $T \circ T = \mathrm{id}_Q$.

(iii) $\Delta \subseteq Q$ *consists of fixed points of* T *and each* E*-class contains exactly one element from* Δ.

A weighted game tableau *is a pair* $(\mathcal{Q}; \nu)$ *consisting of a game tableau together with a weight function* ν *to positive natural numbers* $\nu : Q \to \omega \setminus \{0\}$.

We use greek letters $\alpha, \beta, \delta, \ldots$ for the elements of Q. Each C^2-invariant induces a weighted game tableau:

Lemma 3.2 *Let* $\mathfrak{I} = I_C^2(\mathfrak{A})$ *be a* C^2*-invariant. Then the reduct of* $\mathfrak{I}$ *to the vocabulary consisting of* $\prec, E, T$ *and* Δ *is a game tableau.*

We may therefore look at invariants as certain 'decorated' game tableaux; we isolate some obvious conditions for those decorations that can actually occur:

Definition 3.3 *Call a function* $\Theta\colon Q \to \mathsf{atp}$ good *for a tableau* $\mathcal{Q}$, *if it is compatible with the intended meaning of* T *and* Δ *in* $\mathcal{Q}$. *More precisely, for all* $\alpha \in Q$: $\alpha \in \Delta \Leftrightarrow (x_1 = x_2) \in \Theta(\alpha)$; $\Theta(T\alpha) = \Theta(\alpha)^T$, *where* θ^T *stands for the type* θ *with all occurrences of* x_1 *and* x_2 *exchanged; if* δ *is the unique colour in* Δ *with* $(\alpha, \delta) \in E$, *then* $\Theta(\delta) = \Theta(\alpha) \restriction x_1$, *where* $\theta \restriction x_1$ *stands for the restriction of* θ *to its first variable.*

Definition 3.4 *A* realization *of a weighted game tableau* $\mathcal{Q}$ *over* n *is a surjective mapping* $h\colon n \times n \longrightarrow Q$ *satisfying the following for all* $m_1, m_2 \in n$:

(i) $h(m_1, m_2) \in \Delta \;\Leftrightarrow\; m_1 = m_2$.

(ii) $h(m_2, m_1) = T(h(m_1, m_2))$.

(iii) $[h(m_1, m_2)]^E = \big\{ h(m_1, m_2') \;\big|\; m_2' \in n \big\}$.

(iv) $\big|\{ m_2' \mid h(m_1, m_2') = h(m_1, m_2) \}\big|$
$= \nu(h(m_1, m_2))$.

It is immediately checked that any $\mathfrak{A}$ with universe n yields a realization of the weighted game tableau corresponding to $I_C^2(\mathfrak{A})$. The canonical choice for h is: $h(m_1, m_2) := \mathrm{tp}_{\mathfrak{A}}^{C^2}(m_1, m_2)$. Conversely, let Θ be good for $\mathcal{Q}$. Any realization h of $(\mathcal{Q}; \nu)$ on n induces a unique τ-structure with universe n through the stipulation that for all $m_1, m_2 \in n$: $\mathrm{atp}(m_1, m_2) := \Theta(h(m_1, m_2))$; the conditions on good Θ guarantee that these are consistent.[3] Let $\mathfrak{A}(h, \Theta)$ stand for the structure obtained in this manner. The following is crucial for the reduction of the inversion problem to the realization problem for tableaux.

Lemma 3.5 *Let* Θ *be good for* $(\mathcal{Q}; \nu)$ *and let* h_1 *and* h_2 *be two realizations of the weighted tableau* $(\mathcal{Q}; \nu)$. *Then* $\mathfrak{A}(h_1, \Theta) \equiv_{C^2} \mathfrak{A}(h_2, \Theta)$.

The proof is an application of the game characterization of $\equiv_{C^2}$: The weighted game tableau fixes the combinatorial setup for the game to the extent that gives player **II** a strategy to maintain *equal colour w.r.t. the* h in a game on any two realizations. This suffices to give a strategy in the game on the expansions according to Θ.

Proposition 3.6 *The following are equivalent for any structure* $\mathfrak{I} = \big(\mathcal{Q}; \Theta; \nu\big)$ *of the format of a* C^2*-invariant:*

(i) $\mathfrak{I}$ *occurs as a* C^2*-invariant, i.e.* $\mathfrak{I} = I_C^2(\mathfrak{A})$ *for some finite* τ*-structure* $\mathfrak{A}$.

(ii) Θ *is good for* $\mathcal{Q}$ *and for some realization* h *of* $(\mathcal{Q}; \nu)$ *with associated* $\mathfrak{A}(h, \Theta)$: $I_C^2(\mathfrak{A}(h, \Theta)) = \mathfrak{I}$.

(iii) Θ *is good for* $\mathcal{Q}$, $\mathcal{Q}$ *admits at least one realization, and for all realizations* h *of* $(\mathcal{Q}, \nu)$ *with associated* $\mathfrak{A}(h, \Theta)$: $I_C^2(\mathfrak{A}(h, \Theta)) = \mathfrak{I}$.

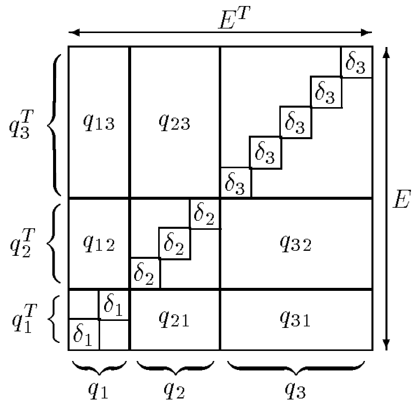
The proof is straightforward on the basis of Lemma 3.5. As a consequence of this proposition, Ptime inversion of I_C^2 is directly obtained from a Ptime algorithm that constructs realizations for the associated tableaux: Suppose that there is a Ptime algorithm $\mathcal{A}$ that constructs a realization for any realizable weighted game tableau. Given $\mathfrak{I}$ of the format of a C^2-invariant, abstract the associated weighted game tableau and apply $\mathcal{A}$. If this weighted tableau has no realization or Θ is not good, then $\mathfrak{I}$ is not in the range of I_C^2. If $\mathcal{A}$ produces a realization, then the canonical expansion of this realization to a τ-structure according to Θ is constructed. It only remains to test whether this structure yields the given $\mathfrak{I}$ as its C^2-invariant: If so, we have found the desired representative in this structure; otherwise $\mathfrak{I}$ is not in the range of I_C^2.

[3] For definiteness, let all relations of arity greater than 2 that might occur in τ contain no tuples with more than two distinct components.

3.2 Realizations of weighted tableaux

The solution of the realization problem is facilitated by a modularity phenomenon which allows to break the problem down to simpler sub-problems. In particular one can achieve a reduction to the case of a trivial reachability relation E. We introduce some notation for this approach. Think of a fixed game tableaux $\mathcal{Q}$. Let $q_1, \ldots, q_l$ be an enumeration of the E-classes in Q, sorted with respect to their $\prec$-least elements. Let $\delta_i \in q_i$ be the unique Δ-element in q_i, $1 \leqslant i \leqslant l$. Let $q_i^T := \{T\alpha \mid \alpha \in q_i\}$, $1 \leqslant i \leqslant l$. Observe that the q_i^T are the classes with respect to E^T, reachability in the first component. It follows that δ_i is the unique Δ-element also in q_i^T. Put $q_{ij} := q_i \cap q_j^T$ for $1 \leqslant i, j \leqslant l$. Observe that $T(q_{ij}) = q_{ji}$ and that for $i \neq j$, $q_{ij} \cap q_{ji} = \emptyset$. Note that the δ_i correspond to the distinct C^2-types of single elements. Figure 1 gives a graphic illustration for these definitions as well as for the following development; in the figure $\mathcal{Q}$ is decomposed according to the above stipulations and projected onto a square array corresponding to a potential realization.

Figure 1



It turns out that – with the exception of atomic information – the C^2-type of a pair is completely determined by the C^2-type of its components. For any realization h of $\mathcal{Q}$:

$$h(m_1, m_2) \in q_{ij} \Leftrightarrow h(m_1, m_1) = \delta_i \wedge h(m_2, m_2) = \delta_j.$$

The following is obtained with a counting argument applied to realizations.

Lemma 3.7 *If $(\mathcal{Q}, \nu)$ admits a realization, then there are positive integers $d_1, \ldots, d_l$ such that for all i, j: $\sum_{\alpha \in q_{ij}} \nu(\alpha) = \sum_{\alpha \in q_{jj}} \nu(\alpha) = d_j$ is independent of i.*

To make modularity explicit we introduce the following components of a tableau:

Definition 3.8 *Let $(\mathcal{Q}; \nu)$ be a weighted game tableau. Define the q_{ij} as above and define the restrictions of $(\mathcal{Q}, \nu)$ to the q_{ij} as follows:*

$$(\mathcal{Q}; \nu)_{ij} := \begin{cases} (q_{ii}, \prec, T \restriction q_{ii}, \delta_i, \nu \restriction q_{ii}) & \text{for } i = j \\ (q_{ij}, \prec, \nu \restriction q_{ij}, \nu^T \restriction q_{ij}) & \text{for } i \neq j. \end{cases}$$

Observe that E is trivial in restriction to the q_{ij}: $E \restriction q_{ij} = q_{ij} \times q_{ij}$. Similarly, $\Delta \restriction q_{ii} = \{\delta_i\}$ and $\Delta \restriction q_{ij} = \emptyset$ for $i \neq j$. For $i \neq j$, T restricts to a bijection between q_{ij} and q_{ji} which becomes external to the restrictions considered; we retain, however, the multiplicity information in $\nu^T = \nu \circ T$ in this off-diagonal case. With these restrictions and the d_j as defined in Lemma 3.7 we find the following reduction:

Lemma 3.9 *A mapping $h: n \times n \to Q$ is a realization of $(\mathcal{Q}, \nu)$ over n, if and only if $n = \sum d_i$ can be identified with the disjoint union of the d_i such that the restrictions $h_{ij}: d_i \times d_j \to q_{ij}$ of h to the $d_i \times d_j$ are surjective, compatible with T in the sense that $h_{ij} = T \circ h_{ji} \circ T$, and further satisfy the following conditions in terms of the individual $(\mathcal{Q}; \nu)_{ij}$:*

(i) $\forall m, m' \in d_i$:
$h_{ii}(m, m') = \delta_i \leftrightarrow m = m'$.

(ii) $\forall m \in d_i, \alpha \in q_{ij}$:
$\left|\{m' \in d_j \mid h_{ij}(m, m') = \alpha\}\right| = \nu(\alpha)$.

(iii) $\forall m' \in d_j, \alpha \in q_{ij}$:
$\left|\{m \in d_i \mid h_{ij}(m, m') = \alpha\}\right| = \nu^T(\alpha)$.

The proof is straightforward, when we identify d_i with the subset of n determined by the condition that $h(m, m) = \delta_i$. With this, the construction of a realization for $(\mathcal{Q}; \nu)$ reduces to the construction of sub-realizations for the individual E-trivial boxes corresponding to the $(\mathcal{Q}; \nu)_{ij}$. These sub-problems are easy for the off-diagonal boxes; for the diagonal boxes they reduce in a non-trivial manner to the type of magic square puzzle mentioned in the introduction. This reduction yields the following necessary and sufficient conditions for realizability of a tableaux – sufficiency is a consequence of Lemma 4.1 on constructible solutions for those puzzles, which we give in Section 4.

Theorem 3.10 *The weighted game tableau $(\mathcal{Q}, \nu)$ has a realization if and only if the following conditions are satisfied:*

(i) $\nu(\delta_i) = 1$ *for* $1 \leqslant i \leqslant l$.

(ii) For all $1 \leqslant i, j \leqslant l$: $\sum_{\alpha \in q_{ij}} \nu(\alpha) = d_j$.

(iii) For all $\alpha \in q_{ij}$, $1 \leqslant i, j \leqslant l$: $\frac{\nu(\alpha)}{\nu(T\alpha)} = \frac{d_j}{d_i}$.

(iv) If $d_i \equiv 1 \bmod 2$, $\alpha \in q_{ii} \setminus \Delta$ *and* $T\alpha = \alpha$, *then* $\nu(\alpha) \equiv 0 \bmod 2$.

There is a PTIME *algorithm that verifies these conditions and if they are satisfied produces a realization over* $n = \sum_{1 \leqslant i \leqslant l} d_i$.

Corollary 3.11 I^2_C *admits* PTIME *inversion. It follows that* C^2 *admits* PTIME *canonization and that* PTIME $\cap\, C^2$ *is recursively enumerable.*

4 Colourings of squares grids

This section gives some indication of the essential constructive step in the generation of realizations for weighted game tableaux. This concerns the solution of the sub-problems for the diagonal boxes in the modular decomposition according to Lemma 3.9. It makes use of some of the conditions stated as necessary and sufficient in Theorem 3.10.

Lemma 4.1 *Let* $(\mathcal{Q}, \nu)_{ii}$ *satisfy* $\nu = \nu \circ T$, $\nu(\delta_i) = 1$. *In case* $d_i = \sum_{\alpha \in q_{ii}} \nu(\alpha)$ *is odd, assume that for all* α *such that* $T\alpha = \alpha \neq \delta_i$, $\nu(\alpha)$ *is even. Then there is a surjective mapping* $h_{ii}\colon d_i \times d_i \longrightarrow q_{ii}$ *satisfying* $T \circ h_{ii} = h_{ii} \circ T$ *and conditions (i),(ii) for* $j = i$ *of Lemma 3.9:*

(i) $\forall m, m' \in d_i\colon h_{ii}(m, m') = \delta_i \leftrightarrow m = m'$,

(ii) $\forall m \in d_i\colon \left|\{m' \in d_i \mid h(m, m') = \alpha\}\right| = \nu(\alpha)$.

Proof

The proof splits into two separate cases, according to whether or not d_i is odd.

Case 1, d_i odd: Enumerate q_{ii} as $\delta_i, \alpha_1, \ldots, \alpha_s, \beta_1, \ldots, \beta_t, T\beta_1, \ldots, T\beta_t$, where the α_k are such that $T\alpha_k = \alpha_k$; for definiteness in the corresponding algorithm choose the enumeration such that the α_k are increasing with respect to $\prec$, and such that $\beta_k \prec \beta_{k'}, T\beta_{k'}$ for $1 \leqslant k < k' \leqslant t$. By assumption, we have that $\nu(\alpha_r) = 2m_r$, $\nu(\beta_k) = \nu(T\beta_k) = n_k$, for suitable m_r, n_k, $1 \leqslant r \leqslant s$, $1 \leqslant k \leqslant t$. It follows that $d_i = 1 + 2m$, where $m := \sum_r m_r + \sum_k n_k$. Put $D_0 := \{(u, v) \in d_i \times d_i \mid (v - u) \bmod d_i \in \{1, \ldots, m\}\}$, such that $d_i \times d_i$ is the disjoint union of the diagonal $\{(u, u) \mid u \in d_i\}$ with D_0 and $T(D_0)$. Identifying m with the ordered disjoint union of the m_r and the n_k, we can now define h_{ii} on D_0 as follows:

$$h_{ii}(u, v) := \begin{cases} \alpha_r & \text{if } (v - u) \bmod d_i \in m_r \\ \beta_k & \text{if } (v - u) \bmod d_i \in n_k. \end{cases}$$

On the diagonal, h_{ii} is set to δ_i; h_{ii} is extended to $T(D_0)$, finally, in accordance with the requirement that $T \circ h_{ii} = h_{ii} \circ T$. It remains to be checked that this h_{ii} gets all occurrence numbers right, which is easy.

Case 2, d_i even: We reduce the construction of h_{ii} to the following claim that will be proved in a separate lemma below:

Fact (i): There is a PTIME constructible mapping $f\colon d_i \times d_i \rightarrow d_i$, such that:

(i) $f(u, u) = 0$ for all $u \in d_i$,

(ii) $f \circ T = f$,

(iii) for all $u \in d_i$: $\{f(u, v) \mid v \in d_i\} = d_i$, and

(iv) for all $v \in d_i$: $\{f(u, v) \mid u \in d_i\} = d_i$.

Figure 2 illustrates such an f on the 6×6 grid:

Figure 2

3	5	4	2	1	0
4	2	5	3	0	1
5	4	1	0	3	2
2	3	0	1	5	4
1	0	3	4	2	5
0	1	2	5	4	3

As above, let q_{ii} be enumerated as $\delta_i, \alpha_1, \ldots, \alpha_s, \beta_1, \ldots, \beta_t, T\beta_1, \ldots, T\beta_t$, with $T\alpha_k = \alpha_k$. Let $\nu(\alpha_r) = m_r$, $1 \leqslant r \leqslant s$ and $\nu(\beta_k) = \nu(T\beta_k) = n_k$, $1 \leqslant k \leqslant t$. Thus $d_i = 1 + \sum_r m_r + 2\sum_k n_k$. Put $m := 1 + \sum_r m_r$, $n := \sum_k n_k$, so that $d_i = m + 2n$. Identifying m with the disjoint ordered union of 1 and the m_r, we put $h_{ii}(u, v) := \alpha_r$ if $0 < f(u, v) < m$ and $f(u, v) \in m_r$. It is immediately checked that this is compatible with the conditions on T and gets the occurrence numbers for the α_r in rows and columns of $d_i \times d_i$ right. On the diagonal h_{ii} is put equal to δ_i. It remains to allocate colours $\beta_k, T\beta_k$ consistently to those pairs (u, v) that are mapped to the $2n$-element subset $\{m, \ldots, d_i - 1\} \subseteq d_i$ by f. After a canonical identification of this subset with the disjoint union over the sets $2 \times n_k$ we want to put

$$h_{ii}(u, v) \in \{\beta_k, T\beta_k\} \quad \text{if} \quad f(u, v) \in 2 \times n_k.$$

In order to show that this can be done in accordance with the conditions on T and with the correct occurrence numbers for β_k and $T\beta_k$ separately, it suffices to show the following, which will also be part of the next lemma:

Fact (ii): Consider a subset $D \subseteq d_i \times d_i$ of the form $D = f^{-1}(a) \cup f^{-1}(b)$ for two different non-zero images a and b. Observe that in each row and in each column of $d_i \times d_i$ there are exactly two elements from D, and that D is disjoint from the diagonal. Then it is possible (in PTIME) to split this set into disjoint subsets in the form $D = D_0 \dot\cup T(D_0)$, such that in each row and in each column there is exactly one element from D_0.

Applying this splitting to each of the n pairs of f-values determined by the above identification with the $2 \times n_k$, it is checked that fixing the β-values of h_{ii} through

$$h_{ii}(u,v) = \begin{cases} \beta_k & \text{if } f(u,v) \in 2\times n_k, (u,v) \in D_0 \\ T\beta_k & \text{if } f(u,v) \in 2\times n_k, (u,v) \in T(D_0). \end{cases}$$

makes h_{ii} satisfy all requirements. Q.E.D.

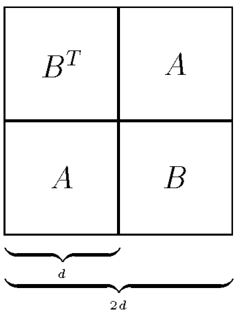
The following lemma contains the two facts about colourings of squares used in the above proof.

Lemma 4.2
(i) For even d there is a colouring of the $(d\times d)$-square with d colours, such that the diagonal is monochromatic, each colour occurs exactly once in each row and in each column, and the entire colouring is symmetric with respect to T.
(ii) Let $D \subseteq d \times d$ be disjoint from the diagonal, symmetric with respect to T, and with exactly two elements in each row and in each column of $d \times d$ belonging to D. Then there is a subset $D_0 \subseteq D$, such that in each row and column of $d \times d$ exactly one element of D_0 occurs, and such that $D = D_0 \dot\cup T(D_0)$.
In both cases, solutions are constructible in time polynomial in d.

Proof
The first claim is proved inductively. Observe, that evenness of d is a necessary condition, since each off-diagonal colour is symmetric with respect to T and contains no fixed points under T. The claim is obvious for $d = 2$. Inductively we show that from a colouring that is good for d we get one for $2d$ and one for $2d-2$.

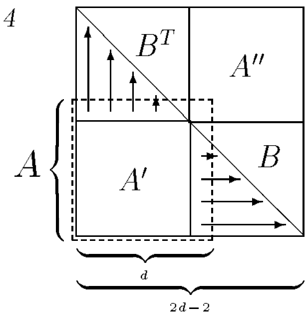
The $2d$ construction is rather obvious: Let A and B be d-squares coloured as required with two disjoint sets of d colours each. Then the composition of these as depicted in figure 3 results in a colouring that is good for $2d$.

Figure 3



Now for the $(2d-2)$-square: Let A be representation of of a correctly coloured d-square, with rightmost column coloured (from top to bottom) $0, \ldots, d-1$, and therefore top row (from left to right) $d-1, \ldots, 0$. Let A' be the resulting $(d-1)$-square when top row and rightmost column of A are removed, and let A'' be the mirror image of A' across the second diagonal (top left to bottom right). Let B be a colouring of the $d-1$-square of the following kind: the second diagonal is coloured from top left to bottom right $1, \ldots, d-1$; the other places are coloured with colours $d, \ldots, d-3$ in such a way that each of these colours occurs exactly once in each row and in each column of B (such colouring is easily obtained with cyclic permutations generating the rows). From A', A'', B and B^T we form the composition depicted in figure 4, where it is also indicated how the colours from the top row and rightmost column of A are integrated into B and B^T. It only remains to check, guided by that sketch, that the proposed composition represents a valid colouring of the $2d-2$-square.

Figure 4



For the second claim let D be as proposed. Consider the relation of belonging to the same row or same column in restriction to D. Since each row and column contain exactly 2 elements in D, D must be the disjoint union of even-length cycles with respect to this relation. Since this relation is T-invariant, for each such cycle C: $T(C) \cap C = \emptyset$ or $T(C) = C$.

Take a single such cycle C. We can put every second element along the cycle into D_0 (starting, for definiteness, with the lexicographically least member). In case $T(C) = C$, it has to be shown that this is in agreement with the operation of T, i.e. that no two T-related positions in C get into D_0. Let C be enumerated as $c_0, c_1, \ldots, c_{2n} = c_0$, c_0 and c_1 in the same row, c_1 and c_2 in the same column, etc. Assume for contradiction that $Tc_0 = c_{2m}$. Then c_1, the element of D in the same row as c_0 must be T-related with c_{2m-1}, the element of D that is in the same column with c_{2m}, since D is symmetric with respect to T. Proceeding in this manner we would find $Tc_m = c_m$, which is impossible, since D is disjoint from the diagonal. The overall splitting of D is obtained by going through all cycles in some fixed order (by lexicographically least elements): If the T-image of a cycle has already been split, then it is split in the way enforced by T, otherwise its D_0-part is obtained according to the above procedure. Q.E.D.

5 Specialization to L^2

The case of L^2 largely appears as a specialization of what we have done for C^2. The invariants related to L^k are introduced in complete analogy with the above. We retain the same additional data on the ordered quotient with respect to $\equiv_{L^k}$ on A^k, with the exception of the numerical information, that was encoded by the weight function. Therefore, the L^k-invariant of $\mathfrak{A}$, $I^k(\mathfrak{A})$ can be encoded as a relational structure of size $|A^k/\equiv_{L^k}| \leqslant |A|^k$. In analogy with Theorem 2.3 it is also PTIME computable, in fact definable in fixed-point logic, and classifies finite relational structures exactly up to $\equiv_{L^k}$: $I^k(\mathfrak{A}) = I^k(\mathfrak{A}')$ iff $\mathfrak{A} \equiv_{L^k} \mathfrak{A}'$.

The crucial systematic difference between the I^k_C and the I^k is that I^k may collapse the size of the given structure, whereas I^k_C cannot. In [4] and [5] give typical examples of tree structures for which the image under I^k is logarithmically small in terms of the size of the original structure; it should be noted that these examples apply to $k \geqslant 3$. Inversion for the I^k can therefore not be achieved in PTIME for $k > 2$. The appropriate modification for the general case would therefore be inversion in time polynomial *in the size of a minimal representative.* With this relaxed notion, it can be checked that still PTIME inversion of the I^k induces PTIME canonization up to $\equiv_{L^k}$. Again, the general problem remains open. It is even unclear whether the range of I^k is recursive. For $k = 2$, however, we get even more: We exhibit PTIME inversion for I^2 in terms of the size of the invariants! Thus it turns out that the potential collapse in size that may occur in the passage from $\mathfrak{A}$ to $I^k(\mathfrak{A})$ cannot be forced by any L^2-theory. Another sense therefore, in which the case of two variables proves to be very special.

The inversion problem for I^2 reduces the the problem of constructing realizations of *game tableaux* without weights. Again, the problem of finding realizations can be split to modular sub-problems. From these one obtains necessary and sufficient conditions for realizability and PTIME constructions of realizations as in the case of C^2. We state these much simpler conditions for a game tableaux $\mathcal{Q}$ and with q_{ij}, the δ_i and the d_i defined as above:

Theorem 5.1 *A game tableau $\mathcal{Q}$ has a realization if and only if all q_{ij} are non-empty, and for all i: $|q_{ii}| = 1 \Rightarrow |q_{ij}| = 1$ for $j = 1, \ldots, l$. If this condition is satisfied then a realization – in fact one of exactly minimal size – can be obtained in time polynomial in the size of $\mathcal{Q}$.*

Corollary 5.2 *I^2 admits PTIME inversion in the strict sense: There is a PTIME functor F which, applied to any $I^2(\mathfrak{A})$, yields a structure $F(I^2(\mathfrak{A})) \equiv_{L^2} \mathfrak{A}$. It follows that L^2 admits PTIME canonization, and that PTIME $\cap\, L^2$ is recursively enumerable, in fact PTIME $\cap\, L^2 \equiv$ PTIME $\circ\, I^2$.*

5.1 Non-recursive queries in L^2

As claimed in the introduction, even L^2 is sufficiently expressive to formalize arbitrarily complex and even non-recursive queries. To exemplify this, we describe a class of *layered rooted directed graphs* corresponding to the images of trees under strict homomorphisms. Let the vocabulary consist of a binary edge relation E. Inductively define

$$\begin{aligned}\varphi_0(x) &= \neg\exists y\, Eyx\\ \varphi_{i+1}(x) &= \neg\varphi_i(x) \wedge \exists y\, Eyx \wedge \forall y(Eyx \rightarrow \varphi_i(y)).\end{aligned}$$

Here $\varphi_i(y)$ is the result of exchanging all occurrences of x and y in φ_i. Consider the L^2-sentence $\exists^{=1} x\varphi_0(x) \wedge \forall x \bigvee_{i \geqslant 0} \varphi_i(x)$. $\exists^{=1} x\varphi_0(x)$ is shorthand for $\exists x\varphi_0(x) \wedge \forall x \forall y(\varphi_0(x) \wedge \varphi_0(y) \rightarrow x = y)$. χ axiomatizes directed graphs with a single source, that are connected, acyclic and have the additional property that any two paths connecting the source to the same point are of the same length. In particular, distance from the source defines a stratification of the graph or a well-behaved depth function.

$$\chi_m = \exists^{=1} x\varphi_0(x) \wedge \forall x \bigvee_{i \leqslant m} \varphi_i(x) \wedge \exists x \varphi_m(x)$$

axiomatizes those of these graphs that have overall depth m. For any set U of natural numbers, the disjunction $\bigvee_{m \in U} \chi_m$ defines the class of those layered rooted directed graphs whose depth is in U. Also cf. [11] for similar examples using, however, 3 variables.

6 Outlook and open problems

The major question that remains open is, whether similar results can be obtained for arbitrary k instead of the still rather special case $k = 2$. More precisely, we have the following problems for arbitrary k:

(1) Do C^k or L^k admit PTIME canonization?

(2) Can the invariant I_C^k be inverted in PTIME?

(3) Can I^k be inverted in time polynomial in the size of a minimal representative?

(2) and (3), respectively, imply (1) for C^k and L^k, respectively. Suppose I_C^k admitted PTIME inversion for all k. This would imply that $\text{PTIME} \cap C^\omega \equiv \bigcup_k \text{PTIME} \circ I_C^k$, which is particularly interesting since $\bigcup_k \text{PTIME} \circ I_C^k$ exactly is FP+C, *fixed-point logic with counting*, see [6, 13]. The analogous consequences for the L^k are slightly less smooth because of the possible collapse in size of the I^k. One weakened formulation is the following: Call a class $Q \subseteq \text{fin}[\tau]$ *bounded for the* I^k if for some k and some polynomial p: $|\mathfrak{A}| \leqslant p(|I^k(\mathfrak{A})|)$ for all $\mathfrak{A} \in Q$. The point is that on bounded classes I^k-inversion in the sense of (3) becomes PTIME inversion in the usual sense. If the I^k admitted inversion in the sense of (3) for all k, then we would have $\text{PTIME} \cap L^\omega \equiv \text{FP}$ on all classes that are bounded for the I^k.

The combinatorial tools developed in the present paper do not seem to be powerful enough for the general case. As pointed out above, two variables are very special in many respects. New results indicate that the case of three variables settles the general case. For instance we can show that $\text{PTIME} \cap C^\omega \equiv \text{FP+C}$ if I_C^3 admits PTIME inversion.

References

[1] S. Abiteboul, V. Vianu, *Generic Computation and Its Complexity*, Proc. 23rd ACM Symp. on Theory of Computing (1991), 209-219.

[2] L. Babai, L. Kučera, *Canonical Labellings of Graphs in Linear Average Time*, IEEE Symp. on Foundations of Computer Science (1980), 39-46.

[3] J. Barwise, *On Moschovakis Closure Ordinals*, Journal of Symbolic Logic **42** (1977), 292-296.

[4] A. Dawar, *Feasible Computation through Model Theory*, doctoral dissertation, University of Pennsylvania (1993).

[5] A. Dawar, S. Lindell, S. Weinstein, *Infinitary Logic and Inductive Definability over Finite Structures*, to appear in Information and Computation; a preliminary version appeared as a research report, University of Pennsylvania (1991).

[6] E. Grädel, M. Otto, *Inductive Definability with Counting on Finite Structures*, in: E. Börger et al (ed.), Computer Science Logic, Selected Papers from CSL '92, LNCS **702**, Springer (1993), 231-247.

[7] Y. Gurevich, *Logic and the Challenge of Computer Science*, in: E. Börger (ed), Trends in Theoretical Computer Science, Computer Science Press (1988), 1-57.

[8] N. Immerman, *Upper and Lower Bounds for First-Order Expressibility*, Journal of Computer and System Sciences **25** (1982), 76-98.

[9] N. Immerman, *Relational Queries Computable in Polynomial Time*, Information and Control **68** (1986), 86-104.

[10] N. Immerman, E. Lander, *Describing Graphs: A First Order Approach to Graph Canonization*, in: A. Selman (ed.), Complexity Theory Retrospective, Springer (1990), 59-81.

[11] Ph. Kolaitis, M. Vardi, *Infinitary Logic and* 0-1 *Laws*, Information and Computation **98** (1992), 258-294.

[12] M. Mortimer, *On Languages with Two Variables*, Zeitschrift für mathematische Logik und Grundlagen der Mathematik **21** (1975), 135-140.

[13] M. Otto, *The Expressive Power of Fixed-Point Logic with Counting*, to appear in Journal of Symbolic Logic (1994).

[14] B. Poizat, *Deux ou trois choses que je sais de L_n*, Journal of Symbolic Logic **47** (1982) 641-658.

[15] D. Scott, *A Decision Method for Validity of Sentences in Two Variables*, Journal of Symbolic Logic **27** (1962) 477.

[16] M. Vardi, *Complexity of Relational Query Languages*, Proc. 14th ACM Symp. on Theory of Computing (1982), 137-146.