ELSEVIER Annals of Pure and Applied Logic 85 (1997) 243–282

ANNALS OF PURE AND APPLIED LOGIC

Canonization for two variables and puzzles on the square

Martin Otto*

Mathematische Grundlagen der Informatik, RWTH Aachen, 52074 Aachen, Germany

Received 18 December 1995; revised 29 October 1996
Communicated by Y. Gurevich

Abstract

We consider infinitary logic with only two variable symbols, both with and without counting quantifiers, i.e. $L^2 := L^2_{\infty\omega}$ and $C^2 := L^2_{\infty\omega}(\exists^{\geqslant m})_{m\in\omega}$. The main result is that finite relational structures admit PTIME canonization with respect to L^2 and C^2: there are polynomial time computable functors mapping finite relational structures to unique representatives of their equivalence class with respect to indistinguishability in either of these logics. In fact we exhibit PTIME inverses to the natural PTIME invariants that characterize structures up to L^2- or C^2-equivalence, respectively. As a corollary we obtain recursive presentations of the classes of all PTIME boolean queries that are closed with respect to L^2- or C^2-equivalence.

AMS classification codes: 03C13, 03D15, 68Q15, 68Q25

Keywords: Finite model theory; Canonization; Descriptive complexity

1. Introduction

We give some basic definitions, an introduction to the problem, as well as a rough statement of the main results and an overview of the entire paper.

Definition 1.1. 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 (bound or free). Let L^ω be the union of the L^k, consisting of those formulae of infinitary logic that involve finite numbers of variables.

Counting quantifiers, i.e. quantifications 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 are natural.

* E-mail: otto@informatik.rwth-aachen.de.

PII S0168-0072(96)00047-4

Definition 1.2. $C^k := L^k_{\infty\omega}(\exists^{\geqslant m})_{m\in\omega}$ is 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 as for L^k. C^ω stands for the corresponding union $\bigcup_k C^k$.

In the case of L^k or C^k the restrictions to finite formulae corresponding to the first-order fragments of these logics, will be described as *finitary* L^k or *finitary* C^k, respectively.

The importance of these logics in finite model theory can be attributed to the fact that the most prominent extensions of first-order logic for the purposes of modelling recursion on finite structures, the extensions by various fixed-point operators, all are sublogics of L^ω. See [23] for more on the rôle of the L^k and [24] on their relation with fixed-point logics.

Similarly C^ω comprises all the expressive power to model the further extensions of these fixed-point logics that incorporate counting in the natural manner, cf. [13, 28]. These containments are particularly useful in proofs of non-expressibility, since the expressive power of the logics L^k and C^k can be analyzed very effectively through Ehrenfeucht-Fraïssé style games. These will concern us in some detail in the following sections.

Let us 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.

Definition 1.3. We say that C^k admits PTIME *canonization* if, for each finite relational type τ, there is a PTIME functor H sending each finite τ-structure $\mathfrak{A}$ to a finite τ-structure $H(\mathfrak{A})$ over some standard universe $\{0,\dots,n-1\}$, $n \geqslant 1$, such that

$$\begin{array}{ll} \forall \mathfrak{A}: & H(\mathfrak{A}) \equiv_{C^k} \mathfrak{A}, \\ \forall \mathfrak{A}, \mathfrak{A}': & \mathfrak{A} \equiv_{C^k} \mathfrak{A}' \Rightarrow H(\mathfrak{A}) = H(\mathfrak{A}'). \end{array}$$

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

Thus canonization functors K and H pick unique representatives from each class with respect to $\equiv_{L^k}$ or $\equiv_{C^k}$. Similar notions of canonization can be considered with respect to any other equivalence relation for finite structures. The standard example in fact is that of *canonization up to isomorphism*, which we shall refer to as *normalization*.

Normalization as such is of interest in descriptive complexity, and in finite model theory in general, because it addresses one of the central problems in the study of algorithms on structures. Any standard model of computation, like the Turing machine model, really deals with representations of input structures as ordered strings. That a Turing machine recognizes a class of finite structures, therefore, implicitly requires this machine to produce an answer independent of the choice of representation: whenever there are two different admissible representations of one and the same structure (or indeed of two isomorphic structures) then obviously the algorithm must either accept or reject both of them.

This is no problem at all with linearly ordered structures, since these admit canonical representations as strings. E.g. we can first identify (in a unique way!) the linearly ordered universe with the standard universe $\{0,\dots,n-1\}$ of the appropriate size n with its natural order. This convention in fact constitutes a trivial normalization procedure for ordered structures. On the basis of this standard representative of an input structure, all structural information is encoded in a binary string associated with some fixed order in which we enumerate the instantiations of the relevant atoms over this standard universe – like the canonical encoding of a graph with standard universe $\{0,\dots,n-1\}$ through its $n \times n$ adjacency matrix.

Therefore the class of all PTIME recognizable classes of finite linearly ordered structures is recursively enumerable in a straightforward manner. It is represented by the class of all polynomially clocked Turing machines taking these canonical representations as inputs. But much more is known, of course. By the well-known theorem due to Immerman [20] and Vardi [32] this class coincides with the class of all fixed-point definable classes; i.e. there is a natural logic for PTIME over ordered structures, in the sense of descriptive complexity.

It is customary in this area of finite model theory to apply the term boolean *query* to algorithms satisfying the invariance condition with respect to representations, or to the classes of finite structures recognized by them. Logically, a boolean query is an isomorphism-closed class of finite structures – or simply a class of finite structures, since in logic one rarely considers structures more closely than up to isomorphism and closure under isomorphisms usually goes without saying.

It is one of the major open problems in this area, whether the class of all PTIME queries is recursively enumerable (more precisely: whether it admits a recursively enumerable system of representatives through PTIME algorithms). In a suitable formalization of the issue, cf. [16], it turns out that recursive enumerability of the PTIME recognizable classes in this sense is equivalent with the existence of a logic for PTIME. The connection between the existence of efficient canonization or normalization procedures for finite structures and a recursive presentation for the class of all PTIME queries is already pointed out by Chandra and Harel in their fundamental paper [8]. The following is indeed obvious along the lines of the argument sketched above for the ordered case. Suppose there were a PTIME normalization algorithm for finite structures (or equivalently, just for finite graphs). Then the class of all PTIME queries would be recursively presentable through the set of all polynomially clocked PTIME algorithms as applied to the result of the given PTIME normalization procedure.

Whether the class of all finite graphs admits PTIME normalization (PTIME canonization up to isomorphism) is open. It is clear that graph normalization must be at least as hard as the graph isomorphism problem. The latter is not known to be in PTIME. It can be shown that graph normalization is in the second level of the polynomial hierarchy, in Δ_2^{pol} in fact, using a variant of the subgraph isomorphism problem for an oracle [27].

Since canonization functors as considered above provide unique representatives up to L^k- or C^k-equivalence rather than up to isomorphism, it is not surprising that these functors have to do with the following fragments of PTIME.

Definition 1.4. Let PTIME $\cap\, C^k$ and PTIME $\cap\, L^k$ stand for the classes of those PTIME boolean queries [1] that are closed with respect to C^k- or L^k-equivalence, respectively.

Note that closure of a class of finite structures with respect to C^k- or L^k-equivalence is equivalent with definability of that class in C^k or L^k. The proof is straightforward on the grounds that these logics are closed under negations and with respect to infinitary disjunctions and conjunctions. These closure properties imply that the C^k- or L^k-theory of any finite τ-structure $\mathfrak{A}$ is isolated by a single sentence $\varphi_{\mathfrak{A}}$ (of C^k or L^k, respectively). If Q is a class of finite τ-structures which is closed under C^k- or L^k-equivalence, then a disjunction over $\varphi_{\mathfrak{A}}$ for representatives of $\mathfrak{A} \in Q$ provides a definition of Q.

PTIME canonization for L^k or C^k bears the following relation with capturing the L^k- or C^k-fragments of PTIME.

Theorem 1.1. *Let C^k and L^k admit* PTIME *canonization through H and K. Then*

$$\begin{aligned} \text{PTIME} \cap C^k &\equiv \text{PTIME}(H) \\ \text{PTIME} \cap L^k &\equiv \text{PTIME}(K). \end{aligned}$$

On the right-hand sides, PTIME(H) *and* PTIME(K) *stand for the class of all* PTIME *algorithms applied to images under H or K, respectively. Since here* PTIME *once more is representable by the class of all polynomially clocked Turing machines, 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 kind indicated in Theorem 1.1.

We do get a little more than that even, owing to the manner in which these canonizations are obtained. Through the work of Abiteboul and Vianu [2] L^k is known to possess concise *invariants* that characterize structures up to $\equiv_{L^k}$. These are PTIME functors I^k mapping 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. We refer to A. Dawar's dissertation [9] and [10] for the logical interpretation and the connection with L^k. For C^k analogous invariants I^k_C are introduced

[1] For expository purposes we deal with boolean queries in the entire presentation. It should be stressed, however, that canonical modifications allow to extend all results to r-ary queries, where $r \leqslant k$ for obvious reasons. A detailed treatment in this extended setting is presented in [27].

and applied to the analysis of fixed-point logic with counting in [13, 28]. 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 PTIME [2] functors F and G such that for all finite $\mathfrak{A}$:

$$F(I_C^k(\mathfrak{A})) \equiv_{C^k} \mathfrak{A}, \qquad G(I^k(\mathfrak{A})) \equiv_{L^k} \mathfrak{A}.$$

With such inverses, it follows that $H := F \circ I_C^k$ and $K := G \circ I^k$ are PTIME canonizations according to Definition 1.3. Our main result in fact provides such inverses for $k=2$.

Main theorem. The ranges of the invariants I_C^2 and I^2 are PTIME recognizable, and there are PTIME constructions yielding

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

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

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

Corollary. PTIME $\cap\, L^2$, *the class of all* PTIME *boolean queries that are also L^2-definable, and* PTIME $\cap\, C^2$, *the class of all* PTIME *boolean queries that are also C^2-definable, are recursively enumerable and admit the following presentations*:

$$\text{PTIME} \cap C^2 \equiv \text{PTIME}(I_C^2),$$
$$\text{PTIME} \cap L^2 \equiv \text{PTIME}(I^2).$$

These normal forms indicate that in restriction to C^2- and L^2-definable queries, Turing computations can make only very limited computational use of the ordering they get in their input representation. Consider a query Q in PTIME $\cap\, L^2$. On the one hand there is a PTIME Turing machine that accepts exactly the 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 $\{I^2(\mathfrak{A}) \mid \mathfrak{A} \in Q\}$. In a sense, that will become 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 Dawar's dissertation [9].

The problem of finding a structure $\mathfrak{A}$ that fits a given value of the invariants, as well as the associated decision problem for the ranges of I_C^2 and I^2, will be reduced to the purely combinatorial problem of constructing certain colourings of square grids; hence

[2] We shall see that in the inversion problem for I^k with $k > 2$, PTIME computability in the usual sense has to be replaced by a weaker notion, because possibly $I^k(\mathfrak{A})$ is much smaller than $\mathfrak{A}$. The appropriate modification is given in Definition 3.1.

'puzzles on the square'. The flavour of these puzzles is exemplified by the following, which I recommend for a try with paper and pencil (a solution can be found as an example in the sequel):

> Colour the 6×6 square with 6 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 corresponding issues of finding PTIME canonizations for L^k or C^k, or of inverting the invariants I^k or I^k_C, perfectly well make sense for the general case of arbitrary k. The specialization to $k=2$ in the crucial sections of this paper first of all reflects the fact that this is the only case we can cope with combinatorially. There are, however, indications that the two-variable case really plays a special rôle on logical grounds.

L^2 is known to be very special in that the satisfiability problem for finitary L^2-sentences over relational languages is decidable [31, 25]. For a new proof of Mortimer's result see [12, 6]. In fact, Mortimer shows in [25] that first-order logic with only two variables has the finite model property: any satisfiable finitary sentence of L^2 has a finite model [25]. Neither of these facts remains true for more than 2 variables. Obviously, 3 variables suffice to axiomatize linear orderings without endpoints. Similarly 3 variables (and in fact also several weak extensions of first-order logic with two variables, see [14]) 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.

That finitary C^2 also is decidable for satisfiability, although it does not have the finite model property (see below), is shown in [15].

Note, however, that even L^2 is sufficiently expressive to define non-trivial – and even non-recursive – classes of finite structures. This has been observed in [18]; we give an example based on Poizat's [30], Exercise 1: there is for every $n \geqslant 1$ a finitary sentence χ_n in L^2 that characterizes linear orderings of length n.

The basis for this axiomatization is formed by formulae $\varphi_i(x)$, such that $\varphi_i[a]$ asserts that there is no sequence of points $a_l < a_{l-1} < \cdots < a_1 < a_0 = a$ of length $l > i$. For $i=0$ one simply takes $\varphi_0(x) = \neg\exists y(y < x)$. If $\varphi_i(x)$ is as required, let $\varphi_i(y)$ be the result of exchanging x and y throughout $\varphi_i(x)$, and put $\varphi_{i+1} = \forall y(y < x \rightarrow \varphi_i(y))$. Let now

$$\psi_n = \forall x \forall y(x = y \vee x < y \vee y < x) \wedge \forall x \varphi_{n-1}(x).$$

Finally put $\chi_1 = \psi_1$ and $\chi_n = \psi_n \wedge \neg\psi_{n-1}$ for $n > 1$.

Now the models of ψ_n exactly are the linear orderings with at most n elements. That these satisfy ψ_n is obvious. Conversely, observe that for all i, $\forall x \varphi_i(x)$ implies the following:

(1) $\forall x \neg(x < x)$,

(2) $\forall x \forall y \neg(x < y \wedge y < x)$,

(3) $\forall x \forall y \forall z(x < y \wedge y < z \rightarrow (\neg x = z \wedge \neg z < x))$.

This is because any $<$-loop or $<$-cycle immediately yields arbitrarily long descending $<$-sequences. Thus each ψ_n implies that $<$ is a linear ordering: it is irreflexive by (1); antisymmetric by (2); connex by the first conjunct in ψ_n; and transitive by (3) together with the first conjunct in ψ_n. Clearly a linear ordering with more than n elements does not satisfy $\forall x \varphi_{n-1}(x)$. Therefore the ψ_n and χ_n are as desired.

For any set U of positive natural numbers, the disjunction $\bigvee_{n \in U} \chi_n$ defines the class of those linear orderings whose length is in U. Therefore L^2 defines arbitrarily complex queries.

A straightforward extension of the given argument shows that for instance any class of linearly ordered graphs is definable in L^2. This is because over linearly ordered domains, the $\varphi_i(x)$ can also be used to define the individual elements: for $n > 1$, a is the n-th element in the ordering $(A, <)$ if it satisfies $\varphi_{n-1} \wedge \neg\varphi_{n-2}$. This may directly be used to characterize up to isomorphism any linearly ordered finite structure in a relational vocabulary with arities no greater than 2.

Observation 1.1. *Let τ contain a symbol $<$ for a linear ordering and otherwise consist of relations whose arities are at most two. Then any class of finite linearly ordered τ-structures is definable in L^2.*

While L^2 might still be regarded as rather trivial, the situation is different for C^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 $\exists^{\leqslant 1}$ suffices to axiomatize classes of necessarily infinite models. E.g. take the sentence that asserts of a binary relation R that it is the graph of an injective but non-surjective function. Finally, the expressive power of C^2 goes far beyond that of L^2 also in the finite. For instance the *stable colouring of graphs* is C^2-definable by results of Immerman and Lander [22] (see Theorem 2.1 below). By results of Babai et al. [3] (compare Theorem 2.2) this implies that almost all finite graphs are characterized even up to isomorphism by their C^2-theory.

Here is an overview over the organization of the paper.

In the following section we shall review some technical preliminaries, in particular, the Ehrenfeucht–Fraïssé style game characterizations for L^k- and C^k-equivalence, and the introduction of the invariants I^k and I^k_C. In this connection we also review the colour refinement technique and look at the stable colouring of graphs as an example for the expressive power of C^2. Section 3 states the problem of inverting the invariants and its connection with canonization in some greater detail. Specializing to $k = 2$, we then introduce *game tableaux* in Section 4 to obtain an abstract combinatorial reformulation of the inversion problem for I^2_C and I^2: the realizations of game tableaux, Section 4.1. The more technical parts in Sections 4.2 and 4.3 isolate necessary and sufficient conditions for realizability of game tableaux. In proving sufficiency we informally develop PTIME algorithms that yield realizations of game tableaux and thereby provide inverses for the invariants. We first treat the case of C^2 in Section 4.2, then – largely through specialization – obtain analogous results for L^2 in Section 4.3. Section 5 gives a very short summary of results, a discussion of the relation of our present work for L^2 with

Mortimer's results concerning the finite model property of the first-order fragment of L^2. Section 5 also contains some remarks concerning the difficulties in the case of arbitrary k and the rôle of the three-variable case as exhibited in [27, 29].

2. Colour refinement, games and invariants

The main concern of this section is a review of the Ehrenfeucht–Fraïssé style characterization of $\equiv_{C^k}$ and $\equiv_{L^k}$ by the associated pebble games. An analysis of these games allows to introduce the invariants I^k_C and I^k using the ideas of colour refinement and stable colouring. Since this also provides an important illustration of the expressive power of C^2, we begin with an outline of the stable colouring of graphs.

For technical purposes we want to use the terminology of *pre-orderings*, hence the following definition.

Definition 2.1. A *pre-ordering* $\preccurlyeq$ is a binary relation that is transitive, reflexive and connex: $\forall xyz(x \preccurlyeq y \wedge y \preccurlyeq z \rightarrow x \preccurlyeq z)$, $\forall x(x \preccurlyeq x)$ and $\forall xy(x \preccurlyeq y \vee y \preccurlyeq x)$.

We write $\prec$ for the associated strict pre-ordering:

$$x \prec y :\leftrightarrow (x \preccurlyeq y \wedge \neg y \preccurlyeq x),$$

and $\sim$ for the associated equivalence relation:

$$x \sim y :\leftrightarrow (x \preccurlyeq y \wedge y \preccurlyeq x).$$

$\sim$ should be seen as associated with the discriminating power of $\preccurlyeq$. The pre-ordering $\preccurlyeq$ itself may be thought of as a linear ordering of the quotient with respect to $\sim$.

2.1. *Colour refinement and the stable colouring of graphs*

Let (V,E) be a graph. A colouring of (V,E) with finitely many colours $1,\dots,r$ is a function $c: V \rightarrow \{1,\dots,r\}$. We regard the set of colours as ordered, and can therefore equivalently formalize this colouring as a pre-ordering on $V : v_1 \preccurlyeq v_2 :\leftrightarrow c(v_1) \leqslant c(v_2)$. The associated $\sim$ is the relation of having the same colour. A refinement of c is induced by the following mapping:

$$v \mapsto (c(v), |\{w \mid Evw \wedge c(w) = 1\}|, \dots, |\{w \mid Evw \wedge c(w) = r\}|),$$

which adds to the information about the present colour of the vertex itself the information about numbers of direct neighbours in each colour. Ordering these new colours lexicographically with dominating first component (former colour), and enumerating the new colours as $1,\dots,r'$ accordingly, we obtain a colouring $c': V \rightarrow \{1,\dots,r'\}$. c' is the *colour refinement* of c. It is a refinement in the sense that for the associated strict pre-orderings $\prec$ and $\prec'$ we have: $\prec \subseteq \prec'$. In other words, the discriminating power of the colouring is enhanced in the passage from c to c'.

If (V,E) is finite, repeated colour refinement must terminate in a stationary colouring after at most $|V|$ steps. The limit colouring obtained in this way from the trivial monochromatic colouring (corresponding to $\prec_0 = \emptyset$) is called the *stable colouring* of the graph. At the level of the associated strict pre-orderings, the stable colouring is the *least fixed point* of the monotone operator (on strict pre-orderings) corresponding to the single colour refinement step sending $\prec$ to $\prec'$.

For definability in C^2, it is sufficient to show that each level $\prec_i$ in this fixed-point process is definable by some C^2-formula $\varphi_i(x,y)$. Then the limit of the sequence $\emptyset = \prec_0 \subseteq \prec_1 \subseteq \cdots$ is defined by

$$\varphi(x,y) := \bigvee_{i \in \omega} \varphi_i(x,y).$$

Suppose that $\varphi_i(x,y)$ defines $\prec_i$. Let as usual $\varphi_i(y,x)$ be the result of exchanging all occurrences of x and y in $\varphi_i(x,y)$. Then for each $j \geqslant 1$ there is a formula $\psi_{i,j}(x)$ in C^2 which defines the j-th equivalence class with respect to $\sim_i$ in the sense of the ordering $\prec_i$. Inductively, first generate formulae $\chi_{i,j}(x)$ defining the union of the first j classes: $\chi_{i,0}(x) := \neg x = x$ defines the empty set, and $\chi_{i,1}(x) := \neg\exists y \varphi_i(y,x)$ defines the $\prec_i$-least $\sim_i$-class. Inductively, let $\chi_{i,j+1}(x) := \forall y(\varphi_i(y,x) \to \chi_{i,j}(y))$. Finally, $\psi_{i,j}(x) := \chi_{i,j}(x) \wedge \neg\chi_{i,j-1}(x)$ is as desired.

Definability of the $\prec_i$ is established by an induction with respect to i. $\varphi_0(x,y) := \neg x = x$ defines $\prec_0 = \emptyset$. For the single refinement step the crucial property, that gives rise to the lexicographic ordering of the new colours, is the following:

> there is some t, such that x and y have exactly the same numbers of neighbours in each colour $1,\ldots,t-1$, but y has more neighbours of colour t.

This can be expressed as follows, using the formulae $\psi_{i,j}$ to define the vertices of colour j:

$$\bigvee_{t \geqslant 1} \begin{pmatrix} \bigwedge_{j<t} |\{u | Exu \wedge \psi_{i,j}(u)\}| = |\{u \,|\, Eyu \wedge \psi_{i,j}(u)\}| \\ \wedge |\{u \,|\, Exu \wedge \psi_{i,t}(u)\}| < |\{u \,|\, Eyu \wedge \psi_{i,t}(u)\}| \end{pmatrix}.$$

In this formula the cardinality equalities and inequalities must be dissolved into infinite disjunctions according to the following pattern:

$$|\{u|\chi(x,u)\}| < |\{u|\chi(y,u)\}| \text{ is equivalent with } \bigvee_{m<n} \exists^{=m} y\, \chi(x,y) \wedge \exists^{=n} x\, \chi(y,x).$$

Note that the quantifiers $\exists^{=n}$ are definable from the $\exists^{\geqslant m}$ without introducing new variables. We have thus proved the first part of the following theorem from [22].

Theorem 2.1 (Immerman and Lander [22]). (i) *The stable colouring of finite graphs is C^2-definable: there is a C^2-formula $\eta(x,y)$ defining on all finite graphs the pre-ordering $\prec$ associated with the stable colouring.*

(ii) *Two vertices receive the same colour in the stable colouring of a finite graph if and only if they satisfy exactly the same formulae of C^2, i.e. the equivalence relation $\sim$ associated with the stable colouring is indistinguishability of vertices in C^2.*

The second statement already connects the colour refinement technique with classification up to equivalence in C^k; for both C^k and L^k this connection is carried further in the construction of the invariants for C^k and L^k, that we shall review in Section 2.3.

The following result from [3] was further used in [4] to obtain a graph normalization algorithm in *average linear time*. The 'almost all' statement of the theorem is to say that the proportion of graphs of size n satisfying the statement tends to 1 as n tends to infinity.

Theorem 2.2 (Babai, Erdös and Selkow [3]). *For almost all finite graphs the stable colouring gives different colours to any two distinct vertices. In other words, almost all finite graphs are in fact linearly ordered by the pre-ordering $\prec$ associated with the stable colouring. It follows that almost all finite graphs are characterized up to isomorphism by their C^2-theories.*

Regarding the technical treatment of the colour refinement reviewed above notice that a completely analogous procedure applies if the single edge relation E is replaced by any finite number of edge relations $E_1, \ldots, E_k$. The refinement step is simply adapted so as to account for all numbers of neighbours for each kind of edge individually. It is in this extended form that the colour refinement technique will serve as a tool in the game analysis and in the generation of the invariants in the following.

2.2. The games for C^k and L^k

In this section we review the appropriate variants of the Ehrenfeucht–Fraïssé game for L^k and C^k. For C^k these are the counting k-pebble games introduced by Immerman and Lander in [22]. The ordinary k-pebble games for L^k go back (in the equivalent but different formulation in terms of back-and-forth systems) to Barwise [5], and in their game formulation are also due to Immerman [19], see also [30]. We shall here treat the L^k-games as specializations of the C^k-games.

Note: All vocabularies are finite and purely relational throughout this paper.

Definition 2.2. We use the following terminology for *types*. Let $\mathfrak{A}$ be a structure, $\bar{a}$ a finite tuple from A. $\mathrm{atp}_{\mathfrak{A}}(\bar{a})$, the atomic type of $\bar{a}$ in $\mathfrak{A}$, is the set of all atomic or negated atomic formulae satisfied by $\bar{a}$ in $\mathfrak{A}$. We shall write $\mathfrak{atp}$ for the finite set of all atomic types in a specified number of variables and fixed relational vocabulary.

$\mathrm{tp}^{C^k}_{\mathfrak{A}}(\bar{a})$ and $\mathrm{tp}^{L^k}_{\mathfrak{A}}(\bar{a})$ denote the classes of all formulae in C^k or L^k that $\bar{a}$ satisfies in $\mathfrak{A}$; here we assume that the length of $\bar{a}$ is at most k. These are referred to as the C^k- or L^k-types of $\bar{a}$ in $\mathfrak{A}$, respectively.

As usual the notation $\equiv_{C^k}$ and $\equiv_{L^k}$ is extended to apply not only to structures, but also to tuples over these: $(\mathfrak{A}, \bar{a}) \equiv_{C^k} (\mathfrak{A}', \bar{a}')$ is shorthand for the equality $\mathrm{tp}^{C^k}_{\mathfrak{A}}(\bar{a}) = \mathrm{tp}^{C^k}_{\mathfrak{A}'}(\bar{a}')$, similarly for L^k.

The C^k-game is played by two players **I** and **II** on two structures $\mathfrak{A}$ and $\mathfrak{A}'$ of the same fixed relational type. There are k pebbles associated with each of the two structures,

$$\begin{array}{ll} p_1,\ldots,p_k & \text{for } \mathfrak{A}, \\ p'_1,\ldots,p'_k & \text{for } \mathfrak{A}'. \end{array}$$

In any position in the game, the pebbles $p_1,\ldots, p_k$ are positioned on elements $a_1,\ldots,a_k$ of the universe A of $\mathfrak{A}$, and similarly $p'_1,\ldots, p'_k$ on elements $a'_1,\ldots,a'_k$ of the universe A' of $\mathfrak{A}'$. Putting $\overline{a} := (a_1,\ldots,a_k)$ and $\overline{a}' := (a'_1,\ldots,a'_k)$ this position is denoted by $(\mathfrak{A},\overline{a};\mathfrak{A}',\overline{a}')$.

A typical move in the game, in position $(\mathfrak{A},\overline{a};\mathfrak{A}',\overline{a}')$, consists of the following four steps which lead to the relocation of a pair of corresponding pebbles over the respective universes:

(i) **I** chooses one of the structures, say $\mathfrak{A}$, a pair of corresponding pebbles (p_j, p'_j) and a non-empty subset of the universe of the chosen structure, here $B \subseteq A$.

(ii) **II** has to designate a subset of the same size in the opposite structure, here $B' \subseteq A'$, $|B'| = |B|$.

(iii) **I** places the pebble p'_j on an element b' within B'.

(iv) **II** places the corresponding pebble p_j on some b in B.

The resulting position is $(\mathfrak{A},\overline{a}\frac{b}{j};\mathfrak{A}',\overline{a}'\frac{b'}{j})$. We write $\overline{a}\frac{b}{j}$ for the tuple $\overline{a}$ with b substituted for the jth component. The game may continue as long as player **II** can maintain the following condition:

(W) The mapping associating the pebbled elements in $\mathfrak{A}$ with those in $\mathfrak{A}'$ must be a partial isomorphism, i.e. $\mathrm{atp}_{\mathfrak{A}}(\overline{a}) = \mathrm{atp}_{\mathfrak{A}'}(\overline{a}')$ for the current positions $\overline{a}$ and $\overline{a}'$.

I wins the game as soon as **II** violates this condition.

Intuitively, player **I** tries to spot a difference in the game positions and strives to force **II** into some position in which this difference is apparent at the atomic level. In the single move **I** challenges **II** through the statement that in one of the structures there are a certain number of replacements of a particular component in the current position of a certain kind. If **II** cannot produce the same numbers of replacements of respective kinds over the other structure, then, in the second phase of the move, **I** may force **II** into a resulting position violating this notion of 'kind'. It turns out that 'kind' really is C^k-type, whence the game is adequate for C^k in the sense of the next theorem.

We say that player **II** has a winning strategy in the game if, for any choice of moves for **I**, **II** can maintain the partial isomorphism condition (W) indefinitely.

Theorem 2.3 (Immerman and Lander [22]). *Player* **II** *has a winning strategy in the game starting from* $(\mathfrak{A},\overline{a};\mathfrak{A}',\overline{a}')$ *if and only if* $\overline{a}$ *over* $\mathfrak{A}$ *and* $\overline{a}'$ *over* $\mathfrak{A}'$ *cannot be distinguished in* C^k, *i.e. if and only if*

$$\mathrm{tp}^{C^k}_{\mathfrak{A}}(\overline{a}) = \mathrm{tp}^{C^k}_{\mathfrak{A}'}(\overline{a}').$$

Here is a review of the argument leading to the theorem:

Proof (*sketch*). First assume that $(\mathfrak{A},\overline{a}) \equiv_{C^k} (\mathfrak{A}',\overline{a}')$. Then **II** has a strategy to maintain this indistinguishability as follows. As there are at most $|A^k|+|A'^k|$ many different types over $\mathfrak{A}$ or $\mathfrak{A}'$, any of these types can be distinguished from all the others by means of some C^k-formula. It therefore follows from the assumption $(\mathfrak{A},\overline{a}) \equiv_{C^k} (\mathfrak{A}',\overline{a}')$ that exactly the same types are realized over $\mathfrak{A}$ and $\mathfrak{A}'$. Let these types be enumerated as $\alpha_1,\ldots,\alpha_N$, with associated C^k-formulae $\chi_1,\ldots,\chi_N$ that single them out over $\mathfrak{A}$ and $\mathfrak{A}'$. Therefore, for each α_i and each j, the number $|\{b \in A \mid \mathrm{tp}^{C^k}_{\mathfrak{A}}(\overline{a}\tfrac{b}{j}) = \alpha_i\}| = |\{b \in A \mid \mathfrak{A} \models \chi_i[\overline{a}\tfrac{b}{j}]\}|$ is determined by $\mathrm{tp}^{C^k}_{\mathfrak{A}}(\overline{a})$. The same holds of $(\mathfrak{A}',\overline{a}')$, whence $(\mathfrak{A},\overline{a}) \equiv_{C^k} (\mathfrak{A}',\overline{a}')$ also implies that for all α_i and all j, the corresponding numbers must be equal for $(\mathfrak{A},\overline{a})$ and $(\mathfrak{A}',\overline{a}')$:

$$|\{b \in A \mid \mathrm{tp}^{C^k}_{\mathfrak{A}}(\overline{a}\tfrac{b}{j}) = \alpha_i\}| = |\{b' \in A' \mid \mathrm{tp}^{C^k}_{\mathfrak{A}'}(\overline{a}'\tfrac{b'}{j}) = \alpha_i\}|.$$

Suppose **I** chooses to play in the j-th component and proposes $B \subseteq A$ as a challenge. By the above equality, **II** can choose $B' \subseteq A'$ such that for all α_i:

$$|\{b \in B \mid \mathrm{tp}^{C^k}_{\mathfrak{A}}(\overline{a}\tfrac{b}{j}) = \alpha_i\}| = |\{b' \in B' \mid \mathrm{tp}^{C^k}_{\mathfrak{A}'}(\overline{a}'\tfrac{b'}{j}) = \alpha_i\}|,$$

from which it follows that, no matter which $b' \in B'$ **I** chooses, **II** can make sure to answer with some $b \in B$ such that the resulting tuples $\overline{a}\tfrac{b}{j}$ and $\overline{a}'\tfrac{b'}{j}$ again realize the same types so that $\equiv_{C^k}$ is maintained.

Secondly, suppose that not $(\mathfrak{A},\overline{a}) \equiv_{C^k} (\mathfrak{A}',\overline{a}')$. This means that there is a formula φ in C^k such that $\mathfrak{A} \models \varphi[\overline{a}]$ but $\mathfrak{A}' \models \neg\varphi[\overline{a}']$. Let ξ be the quantifier rank of φ; $\xi > 0$ unless **I** has won already. We prove that **I** can in one move force a position in which the new tuples can be distinguished by a formula of quantifier rank $\zeta < \xi$. This suffices to give **I** a strategy since by repeated application of such moves the quantifier rank of the distinguishing formula must reach 0 in finitely many steps – a win for **I**. Assume w.l.o.g. that φ is of the form $\exists^{\geqslant m} x_j \psi(\overline{x})$ (other cases reduce to this one through the symmetry of the claim and through replacing φ by one of its boolean constituents if necessary). If **I** chooses pebble pair j and proposes $B := \{b \in A \mid \mathfrak{A} \models \psi[\overline{a}\tfrac{b}{j}]\}$ of cardinality at least m, then **II** cannot help but including at least one element b' in the response B' such that $\mathfrak{A}' \models \neg\psi[\overline{a}'\tfrac{b'}{j}]$ simply because by assumption on φ there are less than m examples available over $(\mathfrak{A}',\overline{a}')$. **I** need only choose such a b' from B' to force a resulting position in which ψ of quantifier rank less than ξ distinguishes the two tuples. $\square$

Remark. Usually the game is introduced in order to characterize $\equiv_{C^k}$ for structures rather than for k-tuples over these. This is achieved through the following modification. Let the first exchange of moves, starting from the naked structures $\mathfrak{A}$ and $\mathfrak{A}'$, be played as follows: **I** places one of the sets of k pebbles on elements of the corresponding structures, **II** places the other set in the other structure; then the game continues as above. That this is an appropriate modification follows from the observation that $\mathfrak{A} \equiv_{C^k} \mathfrak{A}'$ if and only if $\mathfrak{A}$ and $\mathfrak{A}'$ realize exactly the same C^k-types.

In the following we shall concentrate on finite structures even though some of the arguments may be adapted to the infinite case. The goal of our present considerations, however, is to give a concise finitary description of finite structures up to C^k-equivalence. To this end let us look at finite approximations of the infinite C^k-game and introduce corresponding equivalence relations $\sim_i$ for the classification of game positions:

$$(\mathfrak{A},\bar{a}) \sim_i (\mathfrak{A}',\bar{a}') \quad \text{iff} \quad \begin{array}{l}\textbf{II} \text{ has a strategy for at least } i \text{ moves}\\ \text{in the game on } (\mathfrak{A},\bar{a};\mathfrak{A}',\bar{a}').\end{array}$$

The following are easily checked:

(i) $\sim_i$ is an equivalence relation.

(ii) $\sim_0$ is equality of atomic types.

(iii) $\sim_{i+1}$ is a refinement of $\sim_i$.

Lemma 2.1. *Suppose that* $\sim_i = \sim_{i+1}$ *for all positions over two fixed structures* $\mathfrak{A}$ *and* $\mathfrak{A}'$, *i.e. that* $\forall \bar{a} \in A\ \forall \bar{a}' \in A'$:

$$(\mathfrak{A},\bar{a}) \sim_i (\mathfrak{A}',\bar{a}') \Rightarrow (\mathfrak{A},\bar{a}) \sim_{i+1} (\mathfrak{A}',\bar{a}').$$

Then, $\forall \bar{a} \in A\ \forall \bar{a}' \in A'$:

$$(\mathfrak{A},\bar{a}) \sim_i (\mathfrak{A}',\bar{a}') \Leftrightarrow (\mathfrak{A},\bar{a}) \equiv_{C^k} (\mathfrak{A}',\bar{a}').$$

The reason is that the agreement between $\sim_i$ and $\sim_{i+1}$ immediately gives **II** a strategy to maintain $\sim_i$-equivalence of positions: $\sim_{i+1}$-equivalence guarantees the existence of a response to the first player's first move which leads to $\sim_i$-equivalence. This strategy is obviously good for the infinite game.

Lemma 2.2. *If* $\mathfrak{A}$ *and* $\mathfrak{A}'$ *are finite, then there is some* $i \leqslant |A^k \times A'^k|$ *for which* $\sim_i = \sim_{i+1}$ *for all positions over* $\mathfrak{A}$ *and* $\mathfrak{A}'$.

This is a consequence of monotonicity: $\sim_j \supseteq \sim_{j+1}$ for all j. In restriction to $A^k \times A'^k$ this implies termination of the successive refinements as desired.

Let us now specialize to L^k. The appropriate game for L^k is obtained with a simplified rule for the moves leaving out the intermediate choices of sets. Player **I** chooses a pebble in one of the structures and relocates it to some element of that structure, **II** has to respond by relocating the corresponding pebble over the other structure. The winning conditions remain the same. It is readily proved that with this L^k-game we get the following.

Theorem 2.4 (Barwise [5] and Immerman [19]). *Player* **II** *has a winning strategy in the* L^k*-game on* $(\mathfrak{A},\bar{a};\mathfrak{A}',\bar{a}')$ *if and only if* $\mathrm{tp}^{L^k}_{\mathfrak{A}}(\bar{a}) = \mathrm{tp}^{L^k}_{\mathfrak{A}'}(\bar{a}')$.

2.3. The invariants

The analysis of the games gives rise to invariants that give very concise abstractions of the complete L^k- or C^k-theory of finite relational structures. In fact the invariants correspond to quotients with respect to C^k- or L^k-equivalence of k-tuples, respectively. The fundamental concept of such invariants was put forward in the work of Abiteboul and Vianu [2], where objects essentially equivalent with our 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 formulated for L^k in [9] and introduced for C^k in [13, 28].

As above we first treat C^k, then specialize to L^k. We now use the game to analyze a single finite structure. Fixing $\mathfrak{A}$ we regard $\sim_i$ as an equivalence relation on A^k. In slightly improper notation let us take $\overline{a} \sim_i \overline{a}'$, for $\overline{a}, \overline{a}'$ both from A, to mean $(\mathfrak{A}, \overline{a}) \sim_i (\mathfrak{A}, \overline{a}')$. Looking at the rules for a single move in the game, it is seen that the individual refinement step leading from $\sim_i$ to $\sim_{i+1}$ is governed by the following:

$$\begin{aligned} \overline{a} \sim_{i+1} \overline{a}' \quad \text{iff} \quad & \overline{a} \sim_i \overline{a}' \text{, and} \\ & \text{for all } 1 \leqslant j \leqslant k \text{ and for all } \sim_i\text{-classes } \alpha \in A^k/\sim_i\text{:} \\ & |\{b \in A \mid \overline{a}\tfrac{b}{j} \in \alpha\}| = |\{b' \in A \mid \overline{a}'\tfrac{b'}{j} \in \alpha\}|. \end{aligned}$$

Together with the initial stage

$$\overline{a} \sim_0 \overline{a}' \quad \text{iff} \quad \mathrm{atp}_{\mathfrak{A}}(\overline{a}) = \mathrm{atp}_{\mathfrak{A}}(\overline{a}')$$

this provides a uniform inductive definition of the $\sim_i$ that is very similar to those that occurred with the colour refinement technique studied in the last section. Let us therefore introduce a fixed linear ordering on the finite set $\mathfrak{atp}$ of all atomic types in variables $x_1, \ldots, x_k$ over the given relational vocabulary. We thus transform the initial colouring of A^k by atomic types into a global pre-ordering $\preccurlyeq_0$ with associated equivalence relation $\sim_0$ over the kth power of the universe. We can apply the colour refinement step as follows.

The classes of $\sim_i$, now linearly ordered through $\preccurlyeq_i$ or the associated strict $\prec_i$, serve as given colours; instead of the single edge relation in the graph we use the k many reachability relations for tuples in the game: $\overline{a}_1 E_j \overline{a}_2$ if $\overline{a}_1$ and $\overline{a}_2$ differ at most in the j-th component. Then the resulting new colours correspond to the $\sim_{i+1}$-classes, as desired.

The resulting stable colouring on the kth power of the universe provides a globally defined pre-ordering whose associated equivalence relation is equality of C^k-types. In other words, we obtain a linearly ordered representation of $A^k/\equiv_{C^k}$. This ordered quotient forms the backbone of the desired invariants which we shall now define.

Let $\preccurlyeq$ and $\prec$ stand for this pre-ordering with respect to C^k-types on the kth power of the universe, $\sim$ for the associated equivalence relation, which is equality of C^k-types. It is obvious from the way in which it has been obtained that $\prec$ is PTIME computable, in fact it is definable in the extension of fixed-point logic with the Rescher quantifier (cardinality comparison quantifier), see [28]. We endow the ordered quotient $(A^k/\sim, \prec)$ with additional structure so that the entire C^k-theory of $\mathfrak{A}$ can be retrieved. The extra data to be incorporated are:

1. *Atomic types*: For each atomic type $\theta \in \mathrm{atp}$ a unary predicate P_θ is introduced containing exactly those $\alpha \in A^k/\sim$, for which $\overline{a} \in \alpha \Rightarrow \mathrm{atp}_{\mathfrak{A}}(\overline{a}) = \theta$.

2. *Reachability*: For each j, $1 \leqslant j \leqslant k$, a binary predicate E_j is introduced containing exactly those pairs (α_1, α_2) from $A^k/\sim$ for which $\overline{a} \in \alpha_1 \Rightarrow \exists b\, \overline{a}\frac{b}{j} \in \alpha_2$.

3. *Symmetries*: For each σ in the symmetric group S_k a unary function F_σ is introduced which maps α to the equivalence class of $\sigma(\overline{a})$, for $\overline{a} \in \alpha$.

4. *Multiplicities*: For each j, $1 \leqslant j \leqslant k$, a weight function v_j from $A^k/\sim$ to natural numbers is introduced, which sends α to $|\{b \in A \mid \overline{a}\frac{b}{j} \in \alpha\}|$, for $\overline{a} \in \alpha$.

It is readily checked that the given definitions are sound, i.e. that they are independent of choices of representatives $\overline{a} \in \alpha$ where such choices occur, so that the F_σ and v_j are well-defined functions. But also in (1) and (2), the defining conditions are uniformly satisfied for all $\overline{a}$ from the respective class, or for none.

There is of course some redundancy in these data. In the presence of the F_σ it suffices for instance to encode just one of the relations E_j and one of the multiplicities v_j. For example, $v_j = v_k \circ F_{(j,k)}$, where (j,k) is the transposition exchanging j and k.

Definition 2.3. Let for each k and each fixed finite relational signature τ, the C^k-invariant I_C^k be the functor which sends a finite τ-structure $\mathfrak{A}$ to the weighted linearly ordered structure

$$I_C^k(\mathfrak{A}) = (A^k/\sim, \prec, E, (F_\sigma)_{\sigma \in S_k}, (P_\theta)_{\theta \in \mathrm{atp}}; v),$$

where atp stands for the finite set of atomic τ-types in k variables, the P_θ and F_σ are interpreted as above, E is the above E_k, and v the weight function v_k from above.

The relational part of $I_C^k(\mathfrak{A})$ with linearly ordered universe $A^k/\sim$ can in a canonical way be encoded over the ordered standard universe of size $|A|$ if k-tuples (in an initial segment with respect to the lexicographic ordering) are used to represent the elements of $A^k/\sim$. The weight function v takes values in $\{1, \ldots, |A|\}$ so that also for this function there is a canonical relational encoding over the ordered standard universe of size $|A|$. It is useful to standardize the values under I_C^k in this fashion and to regard $|A|$ as the size of $I_C^k(\mathfrak{A})$. The choice of $|A|$ for the size is a matter of convention since sizes of relational encodings are determined only up to polynomial transformations. What goes beyond mere convention is the fact that, because of the attached weights v, a relational encoding of $I_C^k(\mathfrak{A})$ is polynomially coupled in its size to the size of $\mathfrak{A}$ and not to the size of the quotient $A^k/\sim$.

Remark 2.1. Formally we identify $I_C^k(\mathfrak{A})$ with some canonical relational encoding of the ordered weighted structure $I_C^k(\mathfrak{A})$ over the ordered standard universe $\{1, \ldots, |A|\}$ and regard $|A|$ as the size of $I_C^k(\mathfrak{A})$.

Observe that I_C^k is PTIME computable.

Theorem 2.5. *$I_C^k(\mathfrak{A})$ fully determines the C^k-theory of $\mathfrak{A}$. In other words, I_C^k classifies finite relational structures exactly up to $\equiv_{C^k}$:*

$$I_C^k(\mathfrak{A}) = I_C^k(\mathfrak{A}') \quad \textit{iff} \quad \mathfrak{A} \equiv_{C^k} \mathfrak{A}'.$$

Proof (*sketch*). Inductively we show that for each element α of the invariant we can determine the C^k-type $\mathrm{tp}(\alpha)$ that it represents. The atomic part of $\mathrm{tp}(\alpha)$ is directly encoded through the P_θ. Suppose now that for some $\varphi \in C^k$ we have already determined the set $\Phi := \{\alpha \mid \varphi \in \mathrm{tp}(\alpha)\}$. Then $\exists^{\geqslant m} x_k \varphi \in \alpha$ if and only if

$$\sum_{(\alpha,\alpha') \in E, \alpha' \in \Phi} v(\alpha') \geqslant m\,.$$

If the counting quantifier is not applied to the kth variable, a suitable permutation is used to reduce it to this case. □

The invariants related to L^k are introduced in complete analogy with the above. The finite approximations $\sim_i$ can be introduced for positions with respect to the L^k-game in exactly the same way as above, and the reader is invited to check that the analogues of Lemmas 2.1 and 2.2 hold. The colour refinement techniques can be applied to grasp the refinement step from $\sim_i$ to $\sim_{i+1}$ with the following essential difference. No longer is it important to consider the number of neighbours of certain kind; only the one bit information whether or not there are any neighbours of that kind is recorded in the new colours. These new colours, with boolean entries for all but the first component, can be ordered lexicographically just as before. The resulting stable colouring gives rise to a pre-ordering whose equivalence relation is equality of L^k-types. Passing to the corresponding invariants, I^k, we retain the same additional data on the quotient with the exception of the numerical information that was encoded by the weight function.

Definition 2.4. Let for each k and each fixed finite relational signature τ, the L^k-invariant I^k be the functor which sends a finite τ-structure $\mathfrak{A}$ to the linearly ordered structure

$$I^k(\mathfrak{A}) = (A^k/{\sim}, \prec, E, (F_\sigma)_{\sigma \in S_k}, (P_\theta)_{\theta \in \mathrm{atp}}),$$

where now, of course, $\sim$ and $\prec$ are the result of the colour refinement appropriate for L^k.

Observe that the invariant $I^k(\mathfrak{A})$ is a structure of size $|A^k/{\sim}| \leqslant |A|^k$. In fact $I^k(\mathfrak{A})$ is naturally interpreted over A^k itself as a relational structure (there are no additional weights to be represented here). As with I_C^k, we may identify $I^k(\mathfrak{A})$ with its canonical representation over an ordered standard domain $\{1, \ldots, |A^k/{\sim}|\}$. Clearly, I^k also is PTIME computable. The analogue of Theorem 2.5 is the following.

Theorem 2.6. *$I^k(\mathfrak{A})$ determines the complete L^k-theory of $\mathfrak{A}$. In other words, I^k classifies finite relational structures exactly up to $\equiv_{L^k}$:*

$$I^k(\mathfrak{A}) = I^k(\mathfrak{A}') \quad \textit{iff} \quad \mathfrak{A} \equiv_{L^k} \mathfrak{A}'.$$

Some background information about the rôle of the invariants in descriptive complexity is in order. The importance of the invariants is due to the following theorem underlying the main results in [2]. Compare [10] in the case of L^k and see [13, 28] for the extensions dealing with counting. For the following remarks we assume some familiarity with the fixed-point logics: ordinary fixed-point logic FP (formalized as first-order logic with an operation for inductive fixed-points) and partial fixed-point logic PFP (first-order logic with an operation for general fixed-points), compare for instance [11]. We shall not explicitly need these or their counting extensions for the main results of this paper so that the reader not interested in the connection with fixed-point logics can safely skip the rest of this section.

Theorem 2.7. *The following are equivalent for all classes Q of finite τ-structures:*
(i) *Q is* FP*-definable.*
(ii) *For some k, $I^k(Q) = \{I^k(\mathfrak{A}) \mid \mathfrak{A} \in Q\}$ is in* PTIME.
(iii) *For some k, $I^k(Q)$ is* FP*-definable.*
The same applies with PFP *in the place of* FP *and* PSPACE *in the place of* PTIME.

The extension of FP that allows for counting is informally introduced by Immerman e.g. in [21]. The formalization chosen in [13], which is implicit in Immerman's work, uses a two-sorted framework in which ordinary relational structures get enriched by an additional second sort consisting of a linearly ordered domain of the same size as the given structure. The two sorts are linked by means of counting terms that take values in the second sort; they evaluate the cardinalities of definable subsets of the first sort. In this two-sorted framework ordinary inductive or partial fixed-point operators are allowed for the generation of fixed points of mixed sorts. We thus obtain logics FP + C and PFP + C which simultaneously enrich first-order logic by mechanisms for dealing with cardinalities and mechanisms for relational recursion. The relation of FP + C and PFP + C with $C^\omega = \bigcup_k C^k$ is analogous to that of FP and PFP with $L^\omega = \bigcup_k L^k$. In extension of Theorem 2.7 above we get the following.

Theorem 2.8. *The following are equivalent for all classes Q of finite τ-structures:*
(i) *Q is definable in fixed-point logic with counting* FP + C.
(ii) *For some k, $I^k_C(Q) = \{I^k_C(\mathfrak{A}) \mid \mathfrak{A} \in Q\}$ is in* PTIME.
(iii) *For some k, $I^k_C(Q)$ is definable in* FP.
Again, the same holds if we replace FP + C *by* PFP + C, FP *by* PFP *and* PTIME *by* PSPACE.

The proof of the statements of Theorems 2.7 and 2.8 rests on the fact that a fixed-point generation of the respective kind can be simulated over the respective invariants.

The essential reason behind this observation being that tuples enter fixed points class-wise. The fundamental insight into the rôle of such invariants in the analysis of fixed-point logics is due to Abiteboul and Vianu [2] and Theorem 2.7 provides the basis for the first main result in [2], that partial fixed-point collapses to fixed-point if and only if PSPACE collapses to PTIME. The equivalences between (ii) and (iii) in both theorems are consequences of the fact that the invariants are linearly ordered structures; PTIME is captured by FP over these [20, 32], and PSPACE by PFP [1, 32].

The crucial systematic difference between the I^k_C and the I^k is that a collapse in size may be involved in the passage from $\mathfrak{A}$ to $I^k(\mathfrak{A})$, whereas this cannot happen with I^k_C. Because of the numerical weights, the size of $I^k_C(\mathfrak{A})$ is strictly coupled to the size of $\mathfrak{A}$ itself, see Remark 2.1 and comments there. Dawar [9] and Dawar et al. [10] 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 $k \geqslant 3$ (compare [27]). It is this collapse which leads to the surprising second main result of Abiteboul and Vianu in [2], that each partial fixed point whose evaluation terminates in a polynomial number of iterations is equivalently expressible as a fixed point, if and only if PSPACE equals PTIME. For the extensions of partial fixed-point logic and fixed-point logic with counting, on the other hand, and owing to the fact that I^k_C does not collapse size, it follows from the above theorem, that indeed each partial fixed point with counting whose evaluation terminates in a polynomial number of iterations is equivalently expressible as a fixed point with counting, [13, 28].

3. Inversion of the invariants and canonization

Before specializing to $k=2$ and solving the inversion problem for $k=2$ in the next section, we discuss in some more detail the connections between the inversion for the I^k_C and I^k, canonization, and presentations of the query classes PTIME $\cap\, C^k$ and PTIME $\cap\, L^k$.

Consider the problem of inverting the functor I^k_C in the following sense. We want to find a functor F which when applied to any image $I^k_C(\mathfrak{A})$ yields a structure $\mathfrak{B}$, with an initial segment of the natural numbers as its universe, that is C^k-equivalent with $\mathfrak{A}$.

Note that for I^k_C the existence of a recursive inverse is trivial. Any class of C^k-equivalent finite structures consists of structures of the same size n, whence this class contains only finitely many structures up to isomorphism, and really only finitely many members with standard universe $\{0,\dots,n-1\}$. Therefore a unique representative can be chosen by picking the one with the lexicographically least encoding with respect to some fixed encoding scheme. In fact this procedure immediately gives rise to a PSPACE computable functor F. The interesting question, however, is whether or not there is a PTIME computable F.

Let us write G for a functor which similarly for L^k produces for each image $I^k(\mathfrak{A})$ a structure $\mathfrak{B}$ that is L^k-equivalent with $\mathfrak{A}$. Here the existence even of a recursive functor of this kind is not obvious for $k>2$. For $k=2$, however, Mortimer's result [25] that the first-order part of L^2 has the finite model property, implies in particular the existence of a recursive G. L^2-invariants can be translated into finitary Scott sentences for L^2. [25] gives recursive bounds on the size of minimal models of finitary L^2-sentences, whence a model of the Scott sentence can be found through exhaustive search. We refer to the last section of this paper for a more detailed account and for a comparison with what we shall here do in the case of L^2.

Note that the potential collapse in size in the transition to the L^k-invariant implies that at least for $k \geqslant 3$ there cannot be a PTIME G for the trivial reason that the smallest possible representative for $I^k(\mathfrak{A})$ might be of size exponential in the size of $I^k(\mathfrak{A})$. To state a reasonable variant of the problem for arbitrary k, we can therefore only require polynomiality in a bound on the size of the required representative.

Definition 3.1. We say that

(i) I^k_C admits PTIME inversion if there is a PTIME functor F such that $F \circ I^k_C$ preserves the complete C^k-theory, i.e. such that for all $\mathfrak{A}$: $F\big(I^k_C(\mathfrak{A})\big) \equiv_{C^k} \mathfrak{A}$.

(ii) I^k admits PTIME inversion if there is a functor G such that $G \circ I^k$ preserves the complete L^k-theory, and for any input $I^k(\mathfrak{A})$, $G\big(I^k(\mathfrak{A})\big)$ is computable in time polynomial in the minimal size of a structure L^k-equivalent with $\mathfrak{A}$.

Notice, that the requirements of Definition 3.1 (ii) for L^k can in particular be satisfied if there is an algorithm which upon input $(n, I^k(\mathfrak{A}))$ produces in time polynomial in n a representative of size n if such exists. With the help of an algorithm like this one can search for the minimal size of a representative and produce one in time still polynomial in the size of this resulting minimal representative. In this modified form of Definition 3.1 (ii) the problem of inverting I^k is posed as an open problem in [9]. Compare Definition 1.3 for the following.

Theorem 3.1. *If I^k_C admits* PTIME *inversion, then C^k admits* PTIME *canonization; the same holds for I^k and L^k.*

Proof. Let e.g. F be as in Definition 3.1(i). Put $H := F \circ I^k_C$. Then, for all $\mathfrak{A}$, $H(\mathfrak{A}) \equiv_{C^k} \mathfrak{A}$, since by definition $H(\mathfrak{A}) = F(I^k_C(\mathfrak{A})) \equiv_{C^k} \mathfrak{A}$. The condition that H is constant on $\equiv_{C^k}$-classes is obvious, since this holds already true of I^k_C. Hence H yields PTIME canonization with respect to C^k.

For L^k, and because of the possible collapse in size, one has to pay attention to polynomiality of $K := G \circ I^k$, with G as in Definition 3.1(ii). Observe, however, that G, applied to an invariant $I^k(\mathfrak{A})$ must in particular yield a resulting structure in time polynomial in $|\mathfrak{A}|$. $\square$

Recall that PTIME $\cap\, C^k$ and PTIME $\cap\, L^k$ stand for the classes of those PTIME queries that are closed with respect to $\equiv_{C^k}$ and $\equiv_{L^k}$ – or definable in C^k and L^k – respectively; cf. Definition 1.4. We restate Theorem 1.1 of the introduction in order to sketch its proof.

Theorem 3.2. *Let C^k and L^k admit* PTIME *canonization through H and K. Then*

$$\text{PTIME} \cap C^k \equiv \text{PTIME}(H)$$
$$\text{PTIME} \cap L^k \equiv \text{PTIME}(K).$$

In particular these fragments are recursively enumerable.

Proof. We sketch the proof for L^k. Assume first that Q is in PTIME $\cap L^k$. Let M be a PTIME algorithm for Q. Since Q is closed with respect to L^k-equivalence, $\mathfrak{A} \in Q$ iff $K(\mathfrak{A}) \in Q$ iff M accepts $K(\mathfrak{A})$, whence Q is in PTIME(K).

Let now M be any PTIME algorithm. Let Q be the class of all structures $\mathfrak{A}$ for which M accepts $K(\mathfrak{A})$. Then Q is in PTIME, since the size of $K(\mathfrak{A})$ is polynomially bounded in the size of $\mathfrak{A}$ as K is in PTIME. Q is also closed with respect to L^k-equivalence, since membership in Q only depends on $K(\mathfrak{A})$ and K is constant on $\equiv_{L^k}$-classes. □

If a canonization H is obtained through inversion of I^k_C, then the normal form of the last theorem can also be formulated in terms of the invariants.

Theorem 3.3. *Let I^k_C admit a* PTIME *inverse F. Then*

$$\text{PTIME} \cap C^k \equiv \text{PTIME}(I^k_C).$$

Proof. This reduces to the proof of the last theorem, if we notice that in the case of C^k and for $H = F \circ I^k_C$, PTIME(H) = PTIME(I^k_C), since H and I^k_C are PTIME computable from each other: $H = F \circ I^k_C$ and $I^k_C = I^k_C \circ H$. □

This latter argument is not available for L^k in general, in fact the examples of L^k-theories that enforce a logarithmic collapse in the passage to I^k for $k > 2$ show that PTIME(K) $\neq$ PTIME(I^k) for $k > 2$. For $k = 2$, however, we shall find an inverse G polynomial in its input in the usual sense, so that here after all we shall get a presentation of PTIME $\cap L^2$ as PTIME(I^2).

In the general case, however, we only have a weaker variant of Theorem 3.3 for the L^k. If I^k admits a PTIME inverse K, then PTIME $\cap L^k$ is recursively enumerable through the following: a boolean query Q is in PTIME $\cap L^k$ if membership of $\mathfrak{A}$ in Q is decidable in terms of $I^k(\mathfrak{A})$ by an algorithm whose time complexity is polynomial in the size of $K(I^k(\mathfrak{A}))$.

Let us say that I^k *is bounded* on a class of structures Q if there is a polynomial p such that $|\mathfrak{A}| \leqslant p(|I^k(\mathfrak{A})|)$ for all $\mathfrak{A} \in Q$. Boundedness of I^k on Q implies that also $I^{k'}$ is bounded on Q for all $k' > k$. Suppose that I^k is bounded on Q and that I^k admits a PTIME inverse K. Then K must in fact be a PTIME functor in the usual sense: $K(I^k(\mathfrak{A}))$ is PTIME computable in terms of $|A|$ by definition, and $|A|$ is polynomial in the size of $I^k(\mathfrak{A})$ for bounded I^k. In the case of bounded I^k, therefore, a complete analogy with Theorem 3.3 is obtained. If I^k admits a PTIME inverse F, then

$$\text{PTIME} \cap L^k \equiv \text{PTIME}(I^k) \quad \text{where } I^k \text{ is bounded.}$$

We conclude this section with some speculative theorems concerning the impact of PTIME inversion of the I^k_C or I^k for all k – speculative since the existence of such inverses for $k > 2$ remains open. Combining Theorems 2.8 and 3.3 we get the following in the case with counting.

Corollary 3.1. *If the I^k_C admit* PTIME *inversion for all k, then*

$$\text{PTIME} \cap C^{\omega} \equiv \text{FP} + \text{C}.$$

A weaker version is obtained in the absence of counting.

Corollary 3.2. *If the I^k admit* PTIME *inversion for all k, then* $\text{PTIME} \cap L^{\omega} \equiv \text{FP}$ *over all classes on which some I^k is bounded.*

It is remarkable in connection with Corollary 3.1 that all known examples for the separation $\text{FP} + \text{C} \subsetneq \text{PTIME}$ do indeed separate $\text{PTIME} \cap C^{\omega}$ from PTIME. This is simply because all known constructions are based on (rather sophisticated) applications of the game characterization provided by Theorem 2.3. Actually there seem to be essentially two known constructions that imply the separation in question. The original construction of Cai et al. [7] directly addresses this separation issue. Secondly, a more recent construction by Gurevich and Shelah [17], although originating in a different context, also gives rise to this separation result. In any case, it remains a tempting conjecture that FP + C indeed captures "PTIME in the world of C^{ω}" in the sense of Corollary 3.1.

4. Game tableaux and inversion for two variables

We now specialize to the case of two variables. It should be noted that the notion of a realization of a game tableau, as well as its application to obtain a combinatorial reformulation of the inversion problem, are easily adapted to the case of $k > 2$ at the expense of a slightly more involved notation. Since the feasibility of the present approach for $k > 2$ remains doubtful, however, we choose this point to give up all pretense at generality with respect to k. The following section introduces *game tableaux* and reduces the inversion problem for two variables to a problem concerning these tableaux.

4.1. Reduction to game tableaux

Recall the format of the L^k- and C^k-invariants for $k=2$. From Definitions 2.3 and 2.4 we get:

$$I^2(\mathfrak{A}) = (A^2/{\sim}, \prec, E, (F_\sigma)_{\sigma \in S_2}, (P_\theta)_{\theta \in \mathrm{atp}})$$
$$I^2_C(\mathfrak{A}) = (A^2/{\sim}, \prec, E, (F_\sigma)_{\sigma \in S_2}, (P_\theta)_{\theta \in \mathrm{atp}}; \nu),$$

where atp stands for the finite set of atomic τ-types in 2 variables. Note that both $\sim$ and $\prec$ here stand for the different interpretations appropriate for L^2 and C^2, respectively.

Of the action of the symmetric group only the transposition that exchanges first and second component remains to be considered. Let T stand both for this transposition $(1,2)$ itself and for the function $F_{(1,2)}$ describing its operation on the $\sim$-classes. Recall that the reachability relation E of the invariants is reachability in the second component, or E_2: $(\alpha, \alpha') \in E$ if for each pair (a_1, a_2) whose type is α there is some a_2' such that the type of (a_1, a_2') is α'. E_1, the corresponding reachability relation for the first component can be recovered as E^T, defined through $(\alpha, \alpha') \in E^T :\leftrightarrow (T\alpha, T\alpha') \in E$. Similarly, v is the weight function v_2 for the second component: $v(\alpha)$ is the number of substitutions for the second component that do not lead out of α. Recall that the analogous function v_1 for the first component is definable as v^T, $v^T = v \circ T$.

We now want to separate the equality type from the rest of the relational atomic type in order to isolate the basic combinatorial aspects of the games. There are only two different equality types in two variables, whence we introduce just one unary predicate symbol Δ associated with the equality type $x_1 = x_2$ of the diagonal.

Proviso 1. *For the purposes of the remaining technical development we use the following modified formats for the two-variable invariants*:

$$I^2(\mathfrak{A}) = (A^2/{\sim}, \prec, E, T, \Delta, (P_\theta)_{\theta \in \mathrm{atp}}),$$
$$I^2_C(\mathfrak{A}) = (A^2/{\sim}, \prec, E, T, \Delta, (P_\theta)_{\theta \in \mathrm{atp}}; v).$$

Here $\prec$ and E are binary predicates as before, T is a unary function, Δ and the P_θ are unary predicates, v a weight function to positive natural numbers.

Clearly, the new and the old formats are interdefinable even in first-order.

The following definition of game tableaux and weighted game tableaux is intended to abstract the combinatorial core of structures like the L^2- and C^2-invariants; essentially we disregard the association with relational atomic types. The conditions in the definition state some simple facts about real invariants – as we shall see in Lemma 4.1.

Definition 4.1. 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 a unary involution on Q, i.e. $T \circ T = \mathrm{id}_Q$.

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

A *weighted game tableau* is a pair $(\mathcal{Q}, v)$ consisting of a game tableau and a weight function $v : Q \to \omega \setminus \{0\}$.

Observe that the class of game tableaux is first-order definable and hence decidable in Ptime (even Logspace). We use Greek letters $\alpha, \beta, \delta, \ldots$ for the elements of Q and call them *colours*.

Each L^2-invariant induces a game tableau, simply through forgetting the P_θ and retaining only Δ, i.e. through neglecting the relational information in the atomic types. Similarly, each C^2-invariant induces a weighted game tableaux.

Lemma 4.1. *Let $\mathfrak{I}=I_C^2(\mathfrak{A})$ or $\mathfrak{I}=I^2(\mathfrak{A})$ be a C^2- or L^2-invariant (in the new format according to Proviso* 1). *Then the reduct of $\mathfrak{I}$ to the vocabulary consisting of $\prec, E, T$ and Δ is a game tableau.*

Of the conditions (i)–(iii) in Definition 4.1, (i) and (ii) are obvious. For (iii), the implication $\delta \in \Delta \rightarrow T\delta = \delta$ is also obvious; so is the fact that there must be an element of Δ in each E-class: let $\alpha \in A^2/\sim$, $\alpha = \mathrm{tp}_{\mathfrak{A}}(a_1, a_2)$; then $\delta := \mathrm{tp}_{\mathfrak{A}}(a_1, a_1) \in \Delta$ must be E-related with α. Uniqueness of δ within each E-class follows from the fact that for each pair α, α' of E-related types, there are elements a_1, a_2, a_2' in A such that $\alpha = \mathrm{tp}_{\mathfrak{A}}(a_1, a_2)$ and $\alpha' = \mathrm{tp}_{\mathfrak{A}}(a_1, a_2')$.

So invariants are 'decorated' game tableaux; we isolate some obvious conditions on those decorations that can actually occur.

Definition 4.2. Call an expansion $(\mathcal{Q}, (P_\theta)_{\theta\in\mathrm{atp}})$ of a game tableau *good* if each colour $\alpha \in Q$ is contained in P_θ for exactly one element $\theta =: \Theta(\alpha) \in \mathrm{atp}$, and if the induced mapping $\Theta : Q \rightarrow \mathrm{atp}$ satisfies for all α:

(i) $\alpha \in \Delta \Leftrightarrow (x_1 = x_2) \in \Theta(\alpha)$,

(ii) $\Theta(T\alpha) = \Theta(\alpha)^T$, where θ^T stands for the type θ with all occurrences of x_1 and x_2 exchanged, and

(iii) if δ is the unique colour in Δ with $(\alpha, \delta) \in E$, then $\Theta(\delta) = \Theta(\alpha)|_{x_1}$, where $\theta|_{x_1}$ stands for the restriction of θ to its first variable (regarded as a diagonal type in two variables containing $x_1 = x_2$).

These conditions are first-order, whence they can be checked in LOGSPACE. Necessity of these conditions for invariants is immediate along the lines of Lemma 4.1. To ease notation for sets of natural numbers we adopt the standard set-theoretic convention to identify a natural number n with the set $\{0, \ldots, n-1\}$.

Definition 4.3. A realization of a game tableau $\mathcal{Q}$ over n is a surjective mapping

$$h : n \times n \rightarrow Q,$$

satisfying the following:

(i) For all $m_1, m_2 \in n$: $h(m_1, m_2) \in \Delta \Leftrightarrow m_1 = m_2$.

(ii) For all $m_1, m_2 \in n$: $h(m_2, m_1) = T(h(m_1, m_2))$. Equivalently: $h \circ T = T \circ h$.

(iii) For all $m_1, m_2 \in n$, for all $\alpha \in Q$: $(h(m_1, m_2), \alpha) \in E \Leftrightarrow \exists m_2' h(m_1, m_2') = \alpha$.

If $(\mathcal{Q}, \nu)$ is a weighted game tableau, then a realization has to satisfy additionally:

(iv) For all $m_1, m_2 \in n$: $|\{m_2' \in n \mid h(m_1, m_2') = h(m_1, m_2)\}| = \nu(h(m_1, m_2))$.

The following is immediately checked:

Lemma 4.2. *For any $\mathfrak{A}$ of size $|A| = n$, $\mathfrak{A}$ itself yields a realization of the weighted game tableau corresponding to $I^2_C(\mathfrak{A})$. After identifying A with n, there is a canonical choice for h:*

$$\begin{aligned} &h : n \times n \to Q,\\ &(m_1, m_2) \mapsto \mathrm{tp}^{C^2}_{\mathfrak{A}}(m_1, m_2). \end{aligned}$$

Similarly for I^2.

The following lemma and proposition show that the problem of finding inverses for the invariants reduces to the problem of finding realizations for the associated game tableaux.

Lemma 4.3. *Let $(\mathcal{Q}, (P_\theta)_{\theta \in \mathfrak{atp}})$ be a good expansion of the game tableau $\mathcal{Q}$, $\Theta : Q \to \mathfrak{atp}$ the mapping corresponding to the expansion $(P_\theta)_{\theta \in \mathfrak{atp}}$. Then any realization h of $\mathcal{Q}$ on n induces a unique τ-structure $\mathfrak{A}(h, \Theta)$ with universe n through the stipulation that*

- *for all $m_1, m_2 \in n$: $\mathrm{atp}(m_1, m_2) = \Theta(h(m_1, m_2))$, and*
- *if τ has relations of arities greater than two, then no relation over $\mathfrak{A}(h, \Theta)$ contains tuples that have more than two distinct components.*

The mapping $(h, \Theta) \mapsto \mathfrak{A}(h, \Theta)$ is in PTIME *and satisfies the following.*

(i) *If h and h' are realizations of $\mathcal{Q}$ on n and n', respectively, then $\mathfrak{A}(h, \Theta) \equiv_{L^2} \mathfrak{A}(h', \Theta)$.*

(ii) *If h and h' are realizations of the same weighted version $(\mathcal{Q}, v)$ of $\mathcal{Q}$ on n and n', respectively, then $\mathfrak{A}(h, \Theta) \equiv_{C^2} \mathfrak{A}(h', \Theta)$.*

Proof (*sketch*). Existence and uniqueness are obvious; the conditions on good expansions are just such that the obvious way of attributing relational edges to pairs over n leads to no clashes. Clearly, $\mathfrak{A}(h, \Theta)$ is constructible in PTIME.

For claims (i) and (ii) we apply the appropriate games to show that player **II** has a winning strategy in the game on $(\mathfrak{A}(h, \Theta), (m_1, m_2); \mathfrak{A}(h', \Theta), (m'_1, m'_2))$ whenever $h(m_1, m_2) = h'(m'_1, m'_2)$. Observe that it suffices for **II** to maintain the equal colour condition $h(m_1, m_2) = h'(m'_1, m'_2)$, since this in particular implies equality of atomic types. The criteria for realizations are precisely such that this condition can be maintained for any move of **I** (for the C^2-case compare the strategy sketched in the proof of Theorem 2.3 above). □

Proposition 4.1. *The following are equivalent for any $\mathfrak{I} = (\mathcal{Q}, (P_\theta)_{\theta \in \mathfrak{atp}})$ of the standardized format of an L^2-invariant:*

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

(ii) *$\mathfrak{I}$ is a good expansion of $\mathcal{Q}$ and for some realization h of $\mathcal{Q}$ with associated τ-structure $\mathfrak{A}(h, \Theta)$: $I^2(\mathfrak{A}(h, \Theta)) = \mathfrak{I}$.*

(iii) $\mathfrak{I}$ *is a good expansion of* $\mathcal{Q}$, *there is a realization of* $\mathcal{Q}$, *and for all realizations* h *of* $\mathcal{Q}$ *with associated* τ*-structure* $\mathfrak{A}(h,\Theta) : I^2(\mathfrak{A}(h,\Theta)) = \mathfrak{I}$.

Similarly, the following are equivalent for any $\mathfrak{I} = (\mathcal{Q}, (P_\theta)_{\theta\in\mathrm{atp}}, \nu)$ *of the standardized format of a* C^2*-invariant*:

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

(ii) $(\mathcal{Q}, (P_\theta)_{\theta\in\mathrm{atp}})$ *is a good expansion of* $\mathcal{Q}$ *and for some realization* h *of* $(\mathcal{Q}, \nu)$ *with associated* τ*-structure* $\mathfrak{A}(h,\Theta) : I^2_C(\mathfrak{A}(h,\Theta)) = \mathfrak{I}$.

(iii) $(\mathcal{Q}, (P_\theta)_{\theta\in\mathrm{atp}})$ *is a good expansion of* $\mathcal{Q}$, $(\mathcal{Q}, \nu)$ *has a realization, and for all realizations* h *of* $(\mathcal{Q}, \nu)$ *with associated* τ*-structure* $\mathfrak{A}(h,\Theta)$: $I^2_C(\mathfrak{A}(h,\Theta)) = \mathfrak{I}$.

In both cases, a Ptime *inversion of the invariants is directly obtained from a* Ptime *algorithm that constructs realizations for the associated tableaux.*

Proof. The equivalence of (ii) and (iii) follows from the previous lemma. (ii) $\Rightarrow$ (i) is obvious. So finally assume (i), for example in the C^2-case. Let $\mathfrak{I} = I^2_C(\mathfrak{A})$, Θ the mapping corresponding to the good expansion in $\mathfrak{I}$. W.l.o.g. the universe A of $\mathfrak{A}$ is equal with $n = |A|$. This immediately gives a canonical realization h of $(\mathcal{Q}, \nu)$ on n as in Lemma 4.2. Let $\mathfrak{A}(h,\Theta)$ be the τ-structure associated with this realization. It follows that $\mathfrak{A}(h,\Theta) \equiv_{C^2} \mathfrak{A}$, in fact $\mathfrak{A}(h,\Theta) = \mathfrak{A}$ if there are no relation symbols of arity greater than 2 in τ. Thus (ii) follows.

Now, for the Ptime inverses for the invariants. Suppose e.g. that there is a Ptime algorithm $\mathscr{A}$ that constructs a realization for any realizable weighted game tableau. Given $\mathfrak{I}$ of the format of a C^2-invariant of a τ-structure, abstract the associated weighted game tableau and apply $\mathscr{A}$. If this weighted tableau has no realization, then $\mathfrak{I}$ is not in the range of I^2_C. If $\mathscr{A}$ produces a realization, then the canonical expansion of this realization to a τ-structure according to Lemma 4.3 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; if not, then again, $\mathfrak{I}$ is not in the range of I^2_C. $\square$

Proposition 4.1 shows that in order to get Ptime functors G and F that serve as inversions for I^2 and I^2_C in the sense of Definition 3.1, it suffices to supply Ptime constructions for realizations of (weighted) game tableaux.

We introduce some more notation. Let $\mathcal{Q}$ be a game tableau.

1. Let $q_1, \ldots, q_l$ be an enumeration of the E-classes in Q, sorted with respect to their $\prec$-least elements.

2. Let $\delta_i \in q_i$ be the unique Δ-element in q_i, $1 \leqslant i \leqslant l$.

It follows that $\Delta = \{\delta_i |\ 1 \leqslant i \leqslant l\}$.

3. 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 (or reachability in the first component). It follows that δ_i is the unique Δ-element also in q_i^T.

4. 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 $i \neq j$, $q_{ij} \cap q_{ji} = \emptyset$.

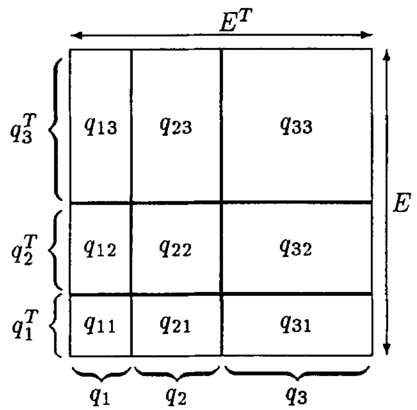

Fig. 1.

Fig. 1 illustrates these definitions. We picture E as vertical reachability, Δ as the main diagonal, and T as the transpose with respect to this diagonal.

Lemma 4.4. *Let h be a realization of $\mathscr{Q}$. Then*

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

Proof (*sketch*). Observe that $h(m_1,m_2) \in q_{ij}$ implies $(h(m_1,m_2),\delta_i)\in E$, whence there must be an m_2' such that $h(m_1,m_2')=\delta_i$, i.e. $h(m_1,m_1) = \delta_i$; similarly for the second component and E^T.

For the converse implication note that $h(m_1,m_1)=\delta_i$ implies $h(m_1,m_2) \in q_i$ for all m_2, and similarly $h(m_2,m_2) = \delta_j$ implies $h(m_1,m_2) \in q_j^T$ for all m_1 (compare conditions (ii) and (iii) of Definition 4.3). It follows that $h(m_1,m_2) \in q_i \cap q_j^T = q_{ij}$. □

Lemma 4.4 is crucial since it forms the basis for splitting the problem of finding a realization to simpler subproblems over the individual q_{ij}. This modularity will be exploited in the following sections, for C^2 and L^2 separately, to state necessary and sufficient conditions for the realizability of (weighted) game tableaux that also suffice to provide a PTIME construction of the realizations.

4.2. C^2 : Realizations of weighted game tableaux

4.2.1. A necessary condition and modularity

Definition 4.4. For a weighted game tableau $(\mathscr{Q}, \nu)$, with E-classes $q_1,\ldots,q_l$, and with the $q_{ij} = q_i \cap q_j^{\mathrm{T}}$ defined as above, define numbers d_i for $i=0,\ldots,l$ according to

$$d_i = \sum_{\alpha\in q_{ii}} \nu(\alpha).$$

Lemma 4.5. *If $(\mathcal{Q}, \nu)$ admits a realization, then for all $1 \leqslant i,j \leqslant l$ the sum $\sum_{\alpha \in q_{ij}} \nu(\alpha)$ is positive and independent of i:*

$$\sum_{\alpha \in q_{ij}} \nu(\alpha) = \sum_{\alpha \in q_{jj}} \nu(\alpha) = d_j.$$

Moreover, if h is any realization of $(\mathcal{Q}, \nu)$ over n, then $d_j = |\{m \in n \mid h(m,m) = \delta_j\}|$, whence it follows that $n = \sum_i d_i$.

Proof. Let h be a realization of $(\mathcal{Q}, \nu)$. By Lemma 4.4: $h(m_1, m_2) \in q_{ij} \Leftrightarrow h(m_1, m_1) = \delta_i \wedge h(m_2, m_2) = \delta_j$. The sum $\sum_{\alpha \in q_{ij}} \nu(\alpha)$, therefore, for any particular i is equal to the number of elements in

$$\{m \in n \mid h(m_1, m) \in q_j^T\}$$

for any m_1 with $h(m_1, m_1) \in \delta_i$. But since $h(m_1, m) \in q_j^T$ if and only if $h(m,m) = \delta_j$, this cardinality is also equal to $|\{m \in n \mid h(m,m) = \delta_j\}|$, independent of i and therefore equal to d_j. The last statement of the lemma follows directly. □

Now for the crucial modularity property of the problem of finding a realization for $(\mathcal{Q}, \nu)$. In a well-defined manner this problem breaks into separate problems corresponding to its restrictions to the individual q_{ij}. In fact the only interrelationship between these sub-problems is already expressed in the condition derived in the last lemma. The next lemma makes modularity explicit, but first we formally introduce these restrictions to the q_{ij}.

Definition 4.5. 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:

$$(\mathcal{Q}, \nu)_{ij} := \begin{cases} (q_{ii}, \prec |q_{ii}, T|q_{ii}, \delta_i, \nu|q_{ii}) & \text{for the diagonal case } j = i \\ (q_{ij}, \prec |q_{ij}, \nu|q_{ij}, \nu^T|q_{ij}) & \text{for the off-diagonal case } i \neq j. \end{cases}$$

We write '$|q_{ij}$' for 'restricted to q_{ij}'. Observe that E is no longer mentioned since in restriction to the q_{ij} it becomes trivial: $E|q_{ij} = q_{ij} \times q_{ij}$. Similarly, $\Delta|q_{ii} = \{\delta_i\}$ and $\Delta|q_{ij} = \emptyset$ for $i \neq j$ lead to corresponding simplifications in the formats of the restrictions. 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.

Lemma 4.6. *Let h be a realization of $(\mathcal{Q}, \nu)$ over $n = \sum d_i$, the d_i as defined in Definition 4.4. Then n can be identified with the disjoint union of the d_i in such a way that the restrictions $h_{ij} := h|(d_i \times d_j)$ of h to the $d_i \times d_j$, $1 \leqslant i,j \leqslant l$, satisfy*

$$\text{image}(h_{ij}) = q_{ij},$$

and in terms of the $(\mathcal{Q}, \nu)_{ij}$:

(1) *On the diagonal, for $j = i$:*

(i) *for all* $m, m' \in d_i$: $h_{ii}(m,m') = \delta_i \leftrightarrow m = m'$,
(ii) $h_{ii} \circ T = T \circ h_{ii}$,
(iii) *for all* $m \in d_i$, $\alpha \in q_{ii}$: $|\{m' \in d_i \mid h_{ii}(m,m') = \alpha\}| = v(\alpha)$.

(2) *In the off-diagonal boxes, for* $i < j$:
(i) *for all* $m \in d_i$, $\alpha \in q_{ij}$: $|\{m' \in d_j \mid h_{ij}(m,m') = \alpha\}| = v(\alpha)$,
(ii) *for all* $m' \in d_j$, $\alpha \in q_{ij}$: $|\{m \in d_i \mid h_{ij}(m,m') = \alpha\}| = v^T(\alpha)$.

Note that h_{ji} *for* $i < j$ *is determined by* h_{ij} *through the condition* $h_{ji} \circ T = T \circ h_{ij}$.

Proof (*sketch*). The claims are obvious after embedding d_i into that subset of n that is determined by the condition that $h(m,m) = \delta_i$. That this subset has size d_i is shown in Lemma 4.5. The rest follows using Lemma 4.4 and the conditions on realizations in Definition 4.3. □

Note that conditions (1)(i)–(iii) and (2)(i),(ii) really are in terms of the data in $(\mathcal{Q}, v)_{ii}$ and $(\mathcal{Q}, v)_{ij}$, respectively. That this decomposition is useful is shown by the next lemma.

Lemma 4.7. *Let* $(\mathcal{Q}, v)$ *be a weighted game tableau, the* d_i *defined according to Definition* 4.4. *Assume that for* $1 \leqslant i \leqslant j \leqslant l$ *there are functions* $h_{ij} : d_i \times d_j \rightarrow q_{ij}$ *satisfying the conditions* (1)(i)–(iii) (*for* $i = j$) *and* (2)(i) *and* (ii) (*for* $i < j$) *of the last lemma. Then there is a realization of* $(\mathcal{Q}, v)$ *on* $n := \sum_i d_i$.

Proof. Represent n as the ordered disjoint union of the d_i. Then a realization h is obtained simply by putting $h \mid (d_i \times d_j) := h_{ij}$ for $i \leqslant j$ and $h \mid (d_i \times d_j) := T \circ h_{ji} \circ T$ for $i > j$. The properties of a realization are checked without difficulties. □

4.2.2. Necessary and sufficient conditions

We now make use of the modularity expressed in the last lemma and split the construction of realizations into the two sub-cases corresponding to the restrictions to diagonal boxes q_{ii} and to off-diagonal boxes q_{ij}, $i < j$, respectively.

Off-diagonal boxes: This is the easier of the two cases, since there are fewer restrictions. In particular, there is no internal symmetry with respect to T.

Suppose that h_{ij} satisfies conditions (2)(i) and (ii) of Lemma 4.6. The crucial additional necessary condition is derived simply through counting the number of pairs that are coloured by $\alpha \in q_{ij}$. Counting column-wise over the d_i columns of $d_i \times d_j$ and, alternatively, row-wise over the d_j rows, we get for each $\alpha \in q_{ij}$:

$$d_i v(\alpha) = |\{(m,m') \in d_i \times d_j \mid h_{ij}(m,m') = \alpha\}| = d_j v^T(\alpha).$$

So necessity of the condition

$$\frac{v(\alpha)}{v(T\alpha)} = \frac{d_j}{d_i} \quad \text{for all } \alpha \in q_{ij}$$

is obvious. The next little lemma shows that conversely this condition allows to construct a mapping h_{ij} satisfying conditions (2)(i) and (ii) of Lemma 4.6.

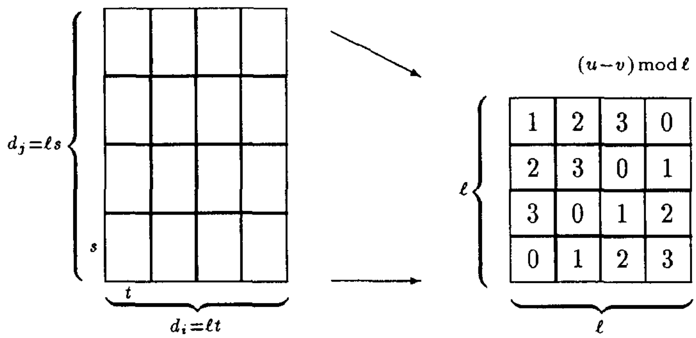

Fig. 2.

Lemma 4.8. *Let $(\mathcal{Q}, v)_{ij}$ for $i<j$ be such that all $\alpha \in q_{ij}$ satisfy*

$$\frac{v(\alpha)}{v(T\alpha)} = \frac{d_j}{d_i},$$

where $d_i = \sum_{\alpha \in q_{ij}} v^T(\alpha)$ and $d_j = \sum_{\alpha \in q_{ij}} v(\alpha)$. Then there is a surjective mapping $h_{ij}: d_i \times d_j \to q_{ij}$ such that (conditions (2)(i) and (ii) of Lemma 4.6 above):

(i) *for all $m \in d_i$: $|\{m' \in d_j \mid h(m,m') = \alpha\}| = v(\alpha)$,*

(ii) *for all $m' \in d_j$: $|\{m \in d_i \mid h(m,m') = \alpha\}| = v^T(\alpha)$.*

Proof. Let $\frac{s}{t}$ be the reduced presentation of $\frac{d_j}{d_i}$. Let $\alpha_1, \ldots, \alpha_r$ be an enumeration of q_{ij} (as ordered by $\prec$). The assumptions of the lemma say that there exist numbers m_k, for $1 \leqslant k \leqslant r$, such that $v(\alpha_k) = m_k s$, $v^T(\alpha_k) = m_k t$. With $\ell := \sum_k m_k$ it follows that $d_j = \ell s$, $d_i = \ell t$ (cf. Fig. 2).

Identify ℓ with the ordered disjoint union of the m_k, d_j with $\ell \times s$, d_i with $\ell \times t$. Then h_{ij} may be defined on $d_i \times d_j = (\ell \times t) \times (\ell \times s)$ by

$$h_{ij}((u,x),(v,y)) := \alpha_k \quad \text{for that } k \text{ with } (u-v) \bmod \ell \in m_k;$$

here u, v run through ℓ, x through t and y through s. Let us check the occurrence condition for columns. For $(u,x) \in \ell \times t$:

$$\{(v,y) \in \ell \times s \mid h_{ij}((u,x),(v,y)) = \alpha_k\} = \{(v,y) \in \ell \times s \mid y \in s, (u-v) \bmod \ell \in m_k\},$$

whence the cardinality of this set is $m_k s = v(\alpha_k)$ as required for (i). Condition (ii) on the occurrence numbers in rows is verified analogously. $\square$

Diagonal boxes. Consider now some diagonal box q_{ii} and first assume that for $(\mathcal{Q}, v)_{ii}$ there is a surjective mapping $h_{ii}: s \times s \to q_{ii}$ satisfying the requirements in (1)(i)–(iii) of Lemma 4.6. The most obvious implication is that $v(\delta_i)$ must be 1.

As above counting the elements in $\{(m,m') \mid h_{ii}(m,m') = \alpha\}$ for each $\alpha \in q_{ii}$ both row-wise and column-wise yields:

$$|\{(m,m') \mid h_{ii}(m,m') = \alpha\}| = sv(\alpha) = sv(T\alpha).$$

It follows that $\nu(\alpha) = \nu(T\alpha)$ for all $\alpha \in q_{ii}$. Summing the above numbers over all $\alpha \in q_{ii}$, we find $s = \sum_{\alpha \in q_{ii}} \nu(\alpha) = d_i$.

There is one more important condition to be derived. Suppose $\alpha \neq \delta_i$ but $T\alpha = \alpha$. Then the set of pairs over $d_i \times d_i$ coloured α does not intersect the diagonal, therefore T is a fixed-point free involution of this set, whence it must have an even number of elements. Since this number equals $d_i \nu(\alpha)$ this gives the condition that – in case d_i is odd – for α as described, $\nu(\alpha)$ must be even.

To sum up, we have seen the necessity of the following conditions:

$$\nu(\alpha) = \nu(T\alpha) \quad \text{for all } \alpha \in q_{ii}, \qquad \nu(\delta_i) = 1,$$

and in case d_i is odd:

$$\nu(\alpha) \equiv 0 \bmod 2 \quad \text{for all } \alpha \in q_{ii} \setminus \{\delta_i\} \text{ such that } T\alpha = \alpha.$$

These conditions also suffice to construct a realization for the diagonal box q_{ii}.

Lemma 4.9. *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 all $\nu(\alpha)$ for $T\alpha = \alpha \neq \delta_i$ are even.*

Then there is a surjective mapping $h_{ii}: d_i \times d_i \longrightarrow q_{ii}$ such that (conditions (1) (i)–(iii) of Lemma 4.6 above):

(i) *for all $m, m' \in d_i$: $h_{ii}(m, m') = \delta_i \leftrightarrow m = m'$,*

(ii) $T \circ h_{ii} = h_{ii} \circ T$,

(iii) *for all $m \in d_i$: $|\{m' \in d_i \mid h(m, m') = \alpha\}| = \nu(\alpha)$.*

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

Case 1: *d_i odd.* Let q_{ii} be enumerated as $\delta_i, \alpha_1, \ldots, \alpha_s, \beta_1, \ldots, \beta_t, T\beta_1, \ldots, T\beta_t$, 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. It follows that $d_i = 1 + 2\ell$, where $\ell := \sum_r m_r + \sum_k n_k$. Put $D_0 := \{(u, v) \in d_i \times d_i \mid (v - u) \bmod d_i \in \{1, \ldots, \ell\}\}$. Observe 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 ℓ with the ordered disjoint union of the m_r and the n_k, we may 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.

(a)

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

(b)

	•		○		
	○	•			
○					•
•				○	
				•	○
		○	•		

Fig. 3.

Fact (i): There is a (PTIME constructible) mapping $f: d_i \times d_i \to d_i$, 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 $v \in d_i : \{f(u,v) \mid u \in d_i\} = d_i$.

An example for such an f, for $d_i = 6$, is shown in Fig. 3(a). A general proof of existence is given in Lemma 4.10.

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 $v(\alpha_r) = m_r$ and $v(\beta_k) = v(T\beta_k) = n_k$. Thus $d_i = 1 + \sum_r m_r + 2\sum_k n_k$. Put $\ell := 1 + \sum_r m_r$, $n := \sum_k n_k$, so that $d_i = \ell + 2n$. Identifying ℓ with the ordered disjoint union of 1 and the m_r, we shall put $h_{ii}(u,v) := \alpha_r$ if $0 < f(u,v) < \ell$ 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 $\{\ell, \ldots, d_i - 1\} \subseteq d_i$ 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. This is illustrated in the example of Fig. 3: in 3(b) the set of points coloured 2 or 5 in 3(a) is split in the required manner.

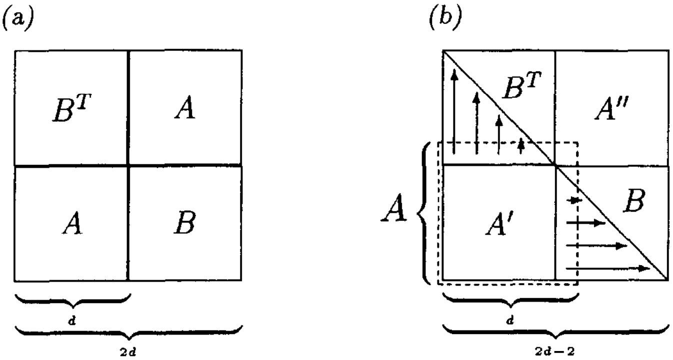

Fig. 4.

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. $\square$

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

Lemma 4.10.

(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 under T (i.e. $T(D) = D$), and such that D contains exactly two elements in each row and in each column of $d \times d$. 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 Fig. 4(a) results in a colouring that is good for $2d$.

Now for the $(2d-2)$-square. Let A be a representation 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 $1,\dots,d-1$; the other places are coloured with colours $d,\dots,2d-3$ 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 Fig. 4(b), 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.

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 and the horizontally next). 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 $c_0,c_1,\dots,c_{2n}=c_0$, 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. Also compare the example in Fig. 3(b). □

Let us finally sum up the necessary and sufficient conditions for a weighted game tableau to have a realization.

Theorem 4.1. *Let $(\mathcal{Q},v)$ be a weighted game tableau. Let the q_i, $1\leqslant i\leqslant l$, be the E-classes, the q_{ij} for $1\leqslant i,j\leqslant l$ defined as $q_{ij}=q_i\cap T(q_j)$. Let δ_i, $1\leqslant i\leqslant l$, be the unique element of $\Delta\cap q_i$. Let further $d_i:=\sum_{\alpha\in q_{ii}}v(\alpha)$ for $1\leqslant i\leqslant l$.*
Then $(\mathcal{Q},v)$ has a realization if and only if the following conditions are satisfied:

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

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

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

(iv) *For all* $1\leqslant i\leqslant l$ *such that* $d_i\equiv 1 \bmod 2$:
If $\alpha\in q_{ii}\setminus\Delta$ *and* $T\alpha=\alpha$, *then* $v(\alpha)\equiv 0 \bmod 2$.

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

Combining this with Proposition 4.1 and Theorems 3.1–3.3 we thus have the following.

Corollary 4.1. *I^2_C admits* PTIME *inversion: there is a* PTIME *functor mapping any $I^2_C(\mathfrak{A})$ to a structure over the universe $\{0,\dots,|A|-1\}$ that is C^2-equivalent with $\mathfrak{A}$. It follows that C^2 admits* PTIME *canonization and that* PTIME $\cap\, C^2$ *is recursively enumerable, in fact* PTIME $\cap\, C^2 =$ PTIME(I^2_C).

4.3. L^2: Realizations of game tableaux

The case of L^2 here largely appears as a specialization of what we have done for C^2. As in the last section the problem of finding a realization for a game tableau $\mathcal{Q}$ can be split into the sub-problems corresponding to the q_{ij}. These now take the following simpler form.

Definition 4.6. Let $\mathcal{Q}$ be a game tableau. With the q_{ij} as usual, the restrictions of $\mathcal{Q}$ to the q_{ij} are the following:

$$\mathcal{Q}_{ij} := \begin{cases} (q_{ii}, \prec |q_{ii}, T|q_{ii}, \delta_i) & \text{for the diagonal case } j = i, \\ (q_{ij}, \prec |q_{ij}) & \text{for the off-diagonal case } i \neq j. \end{cases}$$

The following is a direct specialization of Lemma 4.6.

Lemma 4.11. *Let h be a realization of $\mathcal{Q}$ over $n = \sum d_i$, where $d_i = |\{m \in n \mid h(m,m) = \delta_i\}|$. Then n can be identified with the disjoint union of the d_i in such a way that the restrictions $h_{ij} := h|(d_i \times d_j)$ of h to the $d_i \times d_j$, $1 \leqslant i,\ j \leqslant l$, satisfy*

$$\text{image}(h_{ij}) = q_{ij},$$

and in terms of the $\mathcal{Q}_{ij}$:

(1) *On the diagonal, for $j = i$:*
  (i) *for all $m, m' \in d_i$: $h_{ii}(m,m') = \delta_i \leftrightarrow m = m'$,*
  (ii) $h_{ii} \circ T = T \circ h_{ii}$,
  (iii) *for all $m \in d_i$: $\{h_{ii}(m,m') \mid m' \in d_i\} = q_{ii}$.*

(2) *In the off-diagonal boxes, for $i < j$:*
  (i) *for all $m \in d_i$: $\{h_{ij}(m,m') \mid m' \in d_j\} = q_{ij}$, and*
  (ii) *for all $m' \in d_j$: $\{h_{ij}(m,m') \mid m \in d_i\} = q_{ij}$.*

Let us call any function $h : n \times n \longrightarrow q_{ii}$ satisfying the conditions (1)(i)–(iii) a *realization* of $\mathcal{Q}_{ii}$ over n, and any function $h : n_1 \times n_2 \longrightarrow q_{ij}$ satisfying the conditions (2)(i) and (ii) a realization of $\mathcal{Q}_{ij}$ over $n_1 \times n_2$.

The off-diagonal boxes really cause no problems whatsoever.

Lemma 4.12. *Let $i \neq j$. If $q_{ij} \neq \emptyset$, then $\mathcal{Q}_{ij}$ has realizations over $n_1 \times n_2$ for any $n_1, n_2 \geqslant |q_{ij}|$.*

To see this, simply produce a realization for $n_1 = n_2 = |q_{ij}|$ through cyclic permutation of a row that contains each colour once; any greater values of n_1 or n_2 can be obtained by adding a corresponding number of repetitions of one of the columns or one of the rows.

The diagonal boxes are not quite as trivial.

Lemma 4.13. *For each $\mathcal{Q}_{ii}$:*

(i) *If $q_{ii} = \{\delta_i\}$, then the only realization of $\mathcal{Q}_{ii}$ is the trivial one-element realization.*

(ii) *If $|q_{ii}| > 1$, then there are realizations of $\mathcal{Q}_{ii}$ over any $n \geqslant n_0$, where the optimal value for n_0 is determined as follows: $n_0 = |q_{ii}|$ if there are no $\alpha \in q_{ii} \backslash \{\delta_i\}$ with $T\alpha = \alpha$, and the least even number greater than or equal to $|q_{ii}|$, otherwise.*

Proof (*sketch*). For the first claim observe that off-diagonal elements in $n \times n$ cannot be coloured with δ_i.

The second claim follows with the observation that $v = 1$ constant would yield a minimal realization. By the remarks preceding Lemma 4.9, however, this cannot be achieved in case $|q_{ii}|$ is odd and there are $\alpha \neq \delta_i$ in q_{ii} with $T\alpha = \alpha$ – in this case $|q_{ii}| + 1$ is therefore best possible. Lemma 4.9 yields a realization over the even number $|q_{ii}| + 1$ in this case, if we choose $v(\alpha) = 2$ for exactly one of those $\alpha \neq \delta_i$ that are fixed by T.

If $|q_{ii}| > 1$, any realization can (in a way that is compatible with T) be enlarged through repetition of the off-diagonal part of the top row and last column, and any choice of a T-compatible colouring for the remaining off-diagonal positions with colours from $q_{ii} \setminus \{\delta_i\}$. See Fig. 5 for an example, where a realization over some $n > 1$ is extended to a realization over $n + 2$. □

In analogy with Lemma 4.5 above we get the following very weak cardinality restraint.

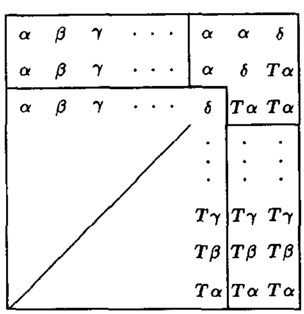

Fig. 5.

Lemma 4.14. *If the game tableau $\mathcal{Q}$ admits a realization, then for all $1 \leqslant i \leqslant l$:*

$$|q_{ii}| = 1 \;\Rightarrow\; |q_{ij}| = 1 \quad \textit{for } j = 1, \ldots, l.$$

We shall immediately see that all $\mathcal{Q}$ with non-empty q_{ij} that satisfy this last condition admit realizations. Unless all q_{ii} are singletons (in which case there is a unique realization), there is a continuous spectrum of sizes of realizations. In order to express the minimal size of a realization precisely, we abstract the following numbers from $\mathcal{Q}$. Let n_{ij}, for $1 \leqslant i, j \leqslant l$, be the optimal bound for a realization of the box $\mathcal{Q}_{ij}$ as found in Lemmas 4.12 and 4.13:

$$n_{ij} := \begin{cases} |q_{ij}| & \text{if } i \neq j, \\ |q_{ii}| & \text{if } i = j \text{ and } \forall \alpha \in q_{ii} \backslash \{\delta_i\} : T\alpha \neq \alpha, \\ 2\lceil \frac{1}{2}|q_{ii}| \rceil & \text{if } i = j \text{ and } \exists \alpha \in q_{ii} \backslash \{\delta_i\} : T\alpha = \alpha. \end{cases}$$

It follows that the smallest possible values for $|\{m \mid h(m,m) = \delta_i\}|$ in any realization are

$$(*) \qquad n_i := \max\{n_{ij} \mid 1 \leqslant j \leqslant l\}.$$

From the modularity of the problem and the results about realizations of the restrictions $\mathcal{Q}_{ij}$ in Lemmas 4.12 and 4.13 it follows that, provided the condition derived in Lemma 4.14 is satisfied, there is indeed a realization of $\mathcal{Q}$ on $n = \sum_i n_i$. This realization is obtained from realizations $h_{ij} : n_i \times n_j \rightarrow q_{ij}$ of the $\mathcal{Q}_{ij}$, $1 \leqslant i \leqslant j \leqslant l$ as follows. Identify n with the ordered disjoint union of the n_i and define $h : n \times n \rightarrow Q$ such that $h \mid (n_i \times n_j) = h_{ij}$ and $h \mid (n_j \times n_i) = T \circ h_{ij} \circ T$ for $i \leqslant j$. Observe that the bound obtained for a minimal realization is *linear* in the size of $\mathcal{Q}$.

Putting these facts together we finally obtain:

Theorem 4.2. *Let $\mathcal{Q}$ be a game tableau. Let the q_i, $1 \leqslant i \leqslant l$, be the E-classes, the q_{ij} for $1 \leqslant i, j \leqslant l$ defined as $q_{ij} = q_i \cap T(q_j)$. Then $\mathcal{Q}$ has a realization if and only if all q_{ij} are non-empty, and for $1 \leqslant i \leqslant l$:*

$$|q_{ii}| = 1 \;\Rightarrow\; |q_{ij}| = 1 \quad \textit{for } j = 1, \ldots, l.$$

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 4.2. *I^2 admits* PTIME *inversion in the following stronger sense: there is a* PTIME *functor G, which applied to any $I^2(\mathfrak{A})$ yields a structure $G(I^2(\mathfrak{A})) \equiv_{L^2} \mathfrak{A}$ – i.e. here we get inverses in polynomial time in the size of the invariant. It follows that L^2 admits* PTIME *canonization, and that* PTIME $\cap\, L^2$ *is recursively enumerable. In fact even* PTIME $\cap\, L^2 \equiv$ PTIME(I^2).

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. An L^2-theory that states that only

s different L^2-types are realized in its models must have a model of size at most $2s$. Another sense therefore, in which the case of two variables proves to be very special.

As a corollary from the proof of the theorem and with the n_i as defined in $(*)$ above we get:

Corollary 4.3. *If $\mathcal{Q}$ has any realization, then it has a realization over $n = \sum_i n_i$. In case not all q_{ii} are singletons, $\mathcal{Q}$ admits realizations exactly over all $n \geqslant \sum_i n_i$. In case all q_{ii} are singletons, $\mathcal{Q}$ has a unique realization (up to isomorphism).*

Corollary 4.4. *The spectrum of any complete L^2-theory that is satisfiable in the finite is either a singleton or the set of all cardinalities greater than n for some n.*

5. Conclusions and remarks

5.1. Summary

We have shown that both L^2 and C^2 admit PTIME canonization. This canonization is achieved through the PTIME inversion of the concise invariants that represent the complete L^2- or C^2-theories of finite structures. For the L^2-invariants the inversion is even PTIME in the size of the invariant, a feature that is known to be impossible for L^k with $k > 2$. An interesting consequence in descriptive complexity is that the L^2- and C^2-fragments of PTIME are recursively enumerable. Even more, they coincide with the classes of queries that are PTIME in terms of the associated invariants.

5.2. Relationship with Mortimer's work on L^2

Our results for L^2 would seem to be closely related with the work of Mortimer's on the finite model property of the first-order part of L^2 [25], compare also [11, 12, 6]. [25] gives a doubly exponential bound on the size of a minimal model for any satisfiable finitary sentence of L^2 (doubly exponential in the length of the sentence that is), while [12] derive an exponential bound. The syntactic analogues of our L^2-invariants I^2 are the corresponding variants of Scott sentences for L^2. We refer the reader to the exposition of the L^k Scott sentences and their relations with our type of invariants in A. Dawar's dissertation [9] and in [10]. These Scott sentences are usually defined inductively from an analysis of the game (or the equivalent back and forth systems); the intermediate formulae of this induction serve to isolate the classes with respect to $\sim_i$, indistinguishability of positions in up to i moves of the game. The resulting Scott sentence for $\mathfrak{A}$ expresses the fact that – for the appropriate closure index i as in Lemmas 2.1, 2.2 – $\sim_i$ is equal to the common refinement or limit of these indistinguishability relations and lists the descriptions of all types at level i that are realized in $\mathfrak{A}$. Its information content, therefore, is exactly that of the invariant $I^k(\mathfrak{A})$. In the usual syntactic induction, the resulting Scott sentence turns out to be of exponential length in the closure index i; its quantifier rank is of the order of the closure index. The invariants indicate, however, how this description can be condensed very much. The Scott sentence for $\mathfrak{A}$ with respect to L^k is logically equivalent with the complete L^k-theory of $\mathfrak{A}$, which is equivalent with any complete description of the invariant $I^k(\mathfrak{A})$ regarded as interpreted within $\mathfrak{A}$ itself. Let $\alpha_1, \ldots, \alpha_s$ be an enumeration of those L^2-types that occur over $\mathfrak{A}$, enumerated in the ordering imposed by $\prec$. Let us extend the vocabulary – much in the spirit of Mortimer's paper – by new binary predicates $R_1, \ldots, R_s$ associated with the α_i. Then a complete L^2-description of the invariant

$$I^2(\mathfrak{A}) = (A^2/{\sim}, \prec, E, T, \Delta, (P_\theta)_{\theta \in \mathrm{atp}})$$

can obviously be given in a sentence of the form

$$\begin{aligned}&\forall x \forall y \bigvee_i R_i xy \wedge \forall x \forall y \bigwedge_{i \neq j} \neg(R_i xy \wedge R_j xy) \wedge \bigwedge_i \forall x \forall y (R_i xy \rightarrow \theta_i(x, y)) \\ &\wedge \bigwedge_i \forall x \forall y (R_i xy \leftrightarrow R_{Ti} yx) \wedge \bigwedge_{i \in \underline{\Delta}, (i,j) \in \underline{E}} \forall x (R_i xx \leftrightarrow \exists y R_j xy),\end{aligned}$$

where we have put:

$$\begin{aligned}&\theta_i := \text{ the atomic component of } \alpha_i, \\ &\underline{\Delta} := \{ i \mid 1 \leqslant i \leqslant s, \alpha_i \in \Delta \}, \\ &\underline{E} := \{ (i, i') \mid 1 \leqslant i, i' \leqslant s, (\alpha_i, \alpha_{i'}) \in E \}, \\ &T : i \mapsto Ti \text{ such that } T\alpha_i = \alpha_{Ti}.\end{aligned}$$

This sentence is of quantifier rank 2 and of quadratic length in $s = |I^2(\mathfrak{A})|$. A closer look at the treatment in [25] shows that for this format Mortimer's bound really becomes polynomial in s, as it should for our purposes. However, his proof of this bound does not yield a PTIME construction of a model. For decidability purposes, of course, the mere existence of a recursive bound is sufficient. There is a model if one can be found through exhaustive search up to this bound. It seems therefore that even for L^2 the elaborate constructions described in the previous sections have some extra benefit. It should be noted, however, that our present constructions essentially depend on the availability of the complete L^2-theory as input; they do not directly apply to the construction of models for arbitrary finitary sentences of L^2.

5.3. Outlook

The main question obviously is whether similar techniques apply to $k > 2$. More broadly one may ask whether C^k or L^k admit PTIME canonization for arbitrary k. It is conceivable that PTIME $\cap L^\omega$ or PTIME $\cap C^\omega$ are recursively enumerable – and for all we know this possibility need not rely on the stronger question whether there is a logic for PTIME as a whole. PTIME $\cap L^\omega$ and PTIME $\cap C^\omega$ are indeed very interesting fragments of PTIME at any rate. Their recursive enumerability is an important and natural open problem that promises to be more tractable than that of a logic for PTIME, since here we have for tools the games, the invariants abstracted from these and the comparatively well-understood relation with natural fixed-point logics. The most obvious individual open problem raised by the present investigation is whether fixed-point logic with counting captures PTIME $\cap C^{\omega}$.

What is special about $k = 2$ in our approach? The notions of game tableaux, good extensions and realizations of game tableaux all generalize to $k > 2$ in an obvious way, with only some extra provisions for the more complex operation of the symmetric group S_k in place of the single non-trivial transformation T. The analogue of Proposition 4.1 continues to hold, so that also in this case everything is reduced to the construction of PTIME realizations of the k-dimensional versions of game tableaux. In the two-dimensional case the crucial property that facilitated our constructions is the modularity that is expressed in Lemmas 4.6 and 4.7. If we want to hold on to the concept of such modular 'boxes' then these acquire more complex internal structure for $k > 2$; in particular, these boxes no longer trivialize the reachability relation E.

It is interesting that new results in [27, 29] indicate that actually the case of three variables is not just the next in line but rather settles the general case as far as the main issues discussed here are concerned. There is a reduction procedure that would allow to employ canonization procedures for L^3 or C^3 (and similarly inversion procedures for I^3 or I^3_C) in the capturing of PTIME $\cap L^{\omega}$ and PTIME $\cap C^{\omega}$, respectively. In particular the conclusions of Corollaries 3.2 and 3.1 may be obtained on the basis of the existence of PTIME inverses for I^3 or I^3_C, respectively.

References

[1] S. Abiteboul and V. Vianu, Fixpoint extensions of first-order logic and Datalog-like languages, in: Proc. 4th IEEE Symp. on Logic in Computer Science (1989) 71–79.

[2] S. Abiteboul and V. Vianu, Generic computation and its complexity, Proc. 23rd ACM Symp. on Theory of Computing (1991) 209–219.

[3] L. Babai, P. Erdös and M. Selkow, Random graph isomorphism, SIAM J. Comput. 9 (1980) 628–635.

[4] L. Babai and L. Kučera, Canonical labellings of graphs in linear average time, IEEE Symp. on Foundations of Computer Science (1980) 39–46.

[5] J. Barwise, On Moschovakis closure ordinals, J. Symbolic Logic 42 (1977) 292–296.

[6] E. Börger, E. Grädel and Y. Gurevich, The Classical Decision Problem, Perspectives in Mathematical Logic (Springer, Berlin, 1996).

[7] J. Cai, M. Fürer and N. Immerman, An optimal lower bound on the number of variables for graph identification, Combinatorica 12 (1992) 389–410.

[8] A. Chandra and D. Harel, Structure and complexity of relational queries, J. Comput. System Sci. 25 (1982) 99–128.

[9] A. Dawar, Feasible computation through model theory, Doctoral dissertation, University of Pennsylvania (1993).

[10] A. Dawar, S. Lindell and S. Weinstein, Infinitary logic and inductive definability over finite structures, Inform. and Comput. 119 (1995) 160–175.

[11] H.-D. Ebbinghaus and J. Flum, Finite Model Theory, Perspectives in Mathematical Logic (Springer, Berlin, 1995).

[12] E. Grädel, Ph.G. Kolaitis and M.Y. Vardi, On the decison problem for two-variable first-order logic, Bulletin of the ASL, to appear.

[13] E. Grädel and M. Otto, Inductive definability with counting on finite structures, in: E. Börger et al., ed., Computer Science Logic, Selected Papers from CSL '92, Lecture Notes in Computer Science, vol. 702 (Springer, Berlin, 1993) 231–247.

[14] E. Grädel, M. Otto and E. Rosen, Undecidability results on two-variable logics, Preprint, 1996.
[15] E. Grädel, M. Otto and E. Rosen, Two-variable logic with counting is decidable, Preprint, 1996.
[16] Y. Gurevich, Logic and the challenge of computer science, in: E. Börger, ed., Trends in Theoretical Computer Science (Computer Science Press, Rockville, MD, 1988) 1–57.
[17] Y. Gurevich and S. Shelah, On finite rigid structures, J. Symbolic Logic 61 (1996) 549–562.
[18] I. Hodkinson, Finite variable logics, Bull. EATCS 51 (1993) 111–140.
[19] N. Immerman, Upper and lower bounds for first-order expressibility, J. Comput. System Sci. 25 (1982) 76–98.
[20] N. Immerman, Relational queries computable in polynomial time, Inform. and Control 68 (1986) 86–104.
[21] N. Immerman, Expressibility as a complexity measure: results and directions, Proc. 2nd Structure in Complexity Conf. (1987) 194–202.
[22] N. Immerman and E. Lander, Describing graphs: a first-order approach to graph canonization, in: A. Selman, ed., Complexity Theory Retrospective (Springer, Berlin, 1990) 59–81.
[23] Ph. Kolaitis and M. Vardi, Infinitary logic and 0-1 laws, Inform. and Comput. 98 (1992) 258–294.
[24] Ph. Kolaitis and M. Vardi, Fixpoint logic vs. infinitary logic in finite-model theory, Proc. 7th IEEE Symp. on Logic and Computer Science (1992) 46–57.
[25] M. Mortimer, On languages with two variables, Zeitschrift für mathematische Logik und Grundlagen der Mathematik 21 (1975) 135–140.
[26] M. Otto, Ptime canonization for two variables with counting, Proc. 10th IEEE Symp. on Logic and Computer Science (1995) 342–352.
[27] M. Otto, Bounded variable logics and counting – a study in finite models, Habilitation thesis RWTH Aachen (1995); revised: Lecture Notes in Logic, vol. 9 (Springer, Berlin, 1997).
[28] M. Otto, The expressive power of fixed-point logic with counting, J. Symbolic Logic 61 (1996) 147–176.
[29] M. Otto, Bounded-variable logics: two, three, and more, Preprint, 1996.
[30] B. Poizat, Deux ou trois choses que je sais de L_n, J. Symbolic Logic 47 (1982) 641–658.
[31] D. Scott, A decision method for validity of sentences in two variables, J. Symbolic Logic 27 (1962) 377.
[32] M. Vardi, Complexity of Relational Query Languages, Proc. 14th ACM Symp. on Theory of Computing (1982) 137–146.