Symmetry and First-Order: Explicitly Presentation-Invariant Circuits

Martin Otto *

Preliminary version, October 94

Abstract

We investigate circuit complexity under the additional assumption of full combinatorial symmetry with respect to permutations of the input representation. With this approach we derive a very natural and simple characterization of first-order logic and its infinitary variant with bounded numbers of variables over finite structures. Extending these results to not necessarily acyclic boolean networks, we derive corresponding characterizations of partial fixed-point logic (or the relational query language WHILE) and ordinary fixed-point logic.

Keywords: Finite model theory, descriptive complexity, generic computation, circuit complexity

1 Introduction

Circuit complexity classes provide one paradigm for measuring algorithmical complexity. In particular they are naturally adapted to capture some aspects of parallelism. We are here interested in the algorithmical complexity of problems related to finite relational structures as inputs, e.g. decision problems for finite graphs. One of the central topics in finite model theory concerns connections between logical definability and algorithmical complexity of such structural properties, i.e. of relational queries. Turning to circuit complexity we are interested in statements relating the computability of relational queries in certain kinds of circuits to their definability in certain logical systems.

To treat finite relational structures of a given type as inputs to boolean circuits, one usually considers families $(C_n)_{n\geq 1}$, where C_n is of a format to allow representations of size n structures as inputs. To obtain from a structure of size n a presentation as a binary string, we identify its universe with the standard domain $\{0,\ldots,n-1\}$. The input gates of C_n are labelled by the set of all instances of atoms (in the given type) over this standard domain. In this way the specification of a concrete input structure of size n corresponds to the obvious allocation of truth values 1 and 0 to the labelled input gates of C_n.

*Mathematische Grundlagen der Informatik, RWTH Aachen, 52074 Aachen, Germany,
Email: otto@mephisto.informatik.rwth-aachen.de

For the present investigation we stress two important points:

- A priori circuits provide a completely *non-uniform* measure of complexity. Without further restrictions, even non-recursive problems are contained in low circuit complexity classes (e.g. purely numerical properties of the size of the universe can be dealt with in families of trivial circuits). Therefore, various uniformity conditions are usually imposed, e.g. in terms of constructibility of the sequence within some (standard) complexity bound.
  Definability in a natural logical system, on the other hand, provides some uniformity in giving a single characterization across all cardinalities.

- One of the problems pervading finite model theory has to do with the arbitrariness of input representations in the case of not necessarily linearly ordered structures. There is no canonical labelling of the input nodes by the instances of the atoms over a given input structure. Hence, the condition that each circuit C_n really computes a boolean function of structures in the intended sense, is an essential implicit restriction: If $\mathfrak{A}_1$ and $\mathfrak{A}_2$, $\mathfrak{A}_i = \big(\{0, \ldots, n-1\}, \ldots\big)$, are isomorphic — i.e. if they correspond to different labellings of the elements of one and the same structure as $0, \ldots, n-1$ — then C_n must compute the same value on both inputs.
  For logical definability, invariance of the semantics under isomorphisms is one of the most basic conditions.

We here want to investigate circuits which satisfy the crucial invariance condition explicitly. The circuits considered below are *explicitly symmetric* in the sense that *any permutation of the universe of an input structure is reflected in an automorphism of the entire circuit.* This is of course the natural and straightforward combinatorial condition to guarantee isomorphism-invariant performance. We combine this requirement with a particular uniformity condition. For this notion of uniformity we present two conceptually different views, which turn out to be equivalent in our framework:

(i) Either we require the sequence $(C_n)_{n \geq 1}$ to be *coherent* in the sense that for $m > n$, C_n can be embedded into C_m in some very uniform way.

(ii) Or, the sequence of circuits is replaced by a single circuit providing input nodes for structures of any finite size. The input nodes are labelled to correspond to atoms over the natural countably infinite domain ω, thus providing a potentially infinite set-up. Now in every concrete computation on a finite input structure, only a finite (but arbitrarily located) portion of this infinite input field is used.

Towards our main result we consider a particular kind of size restriction that is well adapted to the required symmetry. Owing to the symmetry requirement, all permutations of the input field act as automorphisms on the circuit. Restricting these permutations to finite subsets, we require that the *orbits of nodes under these localized symmetry operations grow only polynomially* with the size of the localization. We shall call symmetric circuits that satisfy this local criterion of polynomiality *local polynomial.* The main result establishes that a boolean query on finite structures can be evaluated in an explicitly symmetric, locally polynomial circuit if and only if it is definable in infinitary logic with bounded number of variables, $L^{\omega}_{\infty\omega}$. Further restricting to circuits of finite depth we obtain a corresponding characterization of the expressive power of first-order logic. Waiving the acyclicity requirement for circuits, and similarly considering explicitly symmetric networks with a finite number of types of nodes, we obtain characterizations of partial and ordinary fixed-point logic. We sum up these

Main Results *For explicitly symmetric and locally polynomial boolean circuits, resp. networks, there are the following strict correspondences:*

Circuits	$\equiv$	*Infinitary Logic with Bounded Number of Variables*
Circuits of Finite Depth	$\equiv$	*First-Order Logic*
Finitary Networks	$\equiv$	*Partial Fixed-Point Logic*
Positive Finitary Networks	$\equiv$	*Fixed-Point Logic*

The interesting direction in these matches lies in the passage from circuit or network computability to logical definability. The common key ingredient is a combinatorial lemma to the effect that *locally polynomial predicates* are necessarily definable in terms of a finite tuple of parameters; here a predicate is locally polynomial if its restrictions to finite subsets have only polynomial size orbits in terms of the size of the restrictions.

The main statement among the above is the fact that any node in an explicitly symmetric and locally polynomial circuit computes a truth value that is definable in $L^{\omega}_{\infty\omega}$. The result concerning first-order really is just a consequence of restriction to finite depth and finite quantifier-rank in $L^{\omega}_{\infty\omega}$, respectively. The results concerning networks rather than circuits are obtained in direct analogy with those for circuits.

In fact, explicitly symmetric and locally polynomial networks without further restriction still correspond to $L^{\omega}_{\infty\omega}$. A finiteness requirement on the network, however, gives first-order plus the kind of recursion inherent in network evaluation: Partial fixed-point logic, or the relational query language While.

Symmetric circuits guarantee invariance with respect to the input representation through their graphical layout. Note that the symmetry condition is purely algebraic in terms of the underlying graph of the circuit. The class of these circuits provides an explicitly isomorphism-preserving model of computation, in the sense that isomorphic inputs give rise to isomorphic computations. Models of computation satisfying this invariance condition are called *generic* in the literature. They have been studied in the context of foundational issues, e.g. [GL 81], and also in the context of finite model theory and database theory, e.g. [AV 91]. There is considerable interest in complexity analysis related to natural generic models of computation for several reasons:

(i) The classical complexity classes, e.g. based on Turing machines, were obviously intended to deal with strings, i.e. with linearly ordered structures for inputs.

(ii) Logics *are* 'generic'. Correspondingly, the failure – so far – to equate the standard complexity classes below NP with logical systems in the absence of order, can be attributed to a more systematic misfit. In fact major open problems, like finding a logic for PTime, directly concern the difficulty of isolating the isomorphism-preserving fragment within the non-generic framework.

(iii) In many areas, such as database theory, it is important to have a-priori guarantees of presentation-invariance.

Note that the framework of explicitly symmetric circuits provides a universal model of isomorphism-preserving circuit recognition: It is a trivial observation that exactly all boolean queries on finite relational structures, i.e. all isomorphism preserving boolean functions, can be evaluated by explicitly symmetric circuits. However, for typical circuits, the straightforward symmetrization procedure leads to an exponential blow-up in size. If symmetrization is possible without being forced to non-polynomial width, then we are back in the realm of locally polynomial and explicitly symmetric circuits.

It is surprising that this weak size restriction turns out to be sufficiently powerful in the proposed framework to isolate some of the most important logical systems considered in finite model theory.

Concerning the connections with other work on logical definability in relation to circuit complexity classes, it is important to distinguish two different approaches. Most of the literature concerns the standard case of circuit computability on strings, i.e. ordered structures; on the logical side this is reflected in the assumed presence of linear order, and often of other built-in constructs that appeal to ordered standard domains. These characterizations would violate isomorphism- or presentation-invariance (genericity) if applied to computations on not necessarily ordered input structures.

Our results involving finite depth circuits and first-order logic should be compared to a corresponding characterization of first-order logic in terms of certain restricted families of circuits in a paper by Denenberg, Gurevich and Shelah [DGS 86], concerned with the genericity of circuit computations. The circuits exhibited there essentially also satisfy our criteria. In the crucial direction, however, we think that the present investigation yields new insights even for the first-order case, since the criteria proposed here are not directly related to the syntactic structure of first-order logic. On the contrary, they are derived quite independently on the basis of the invariance and uniformity considerations outlined above.

There are several logical characterizations of circuit or parallel complexity classes which concern the standard case of computations on strings. We mention the work of Immerman, Barrington, Straubing, Compton and Laflamme [I 89, BIS 86, CL 86], and also of Gurevich and Lewis, [GL 84], where low circuit complexity classes are equated with suitable logics over standard domains.

The central logic in our treatment is $L^{\omega}_{\infty\omega}$, infinitary logic with finitely many variables in each formula. We write $L^{k}_{\infty\omega}$ for that fragfment of infinitary logic $L_{\infty\omega}$ that consists of those formulae in which only k variable symbols occur, free or bound. $L^{\omega}_{\infty\omega}$ is the union of these, $L^{\omega}_{\infty\omega} = \bigcup_k L^{k}_{\infty\omega}$. These logics plays a central rôle in finite model theory. Technically this is due to the fact that they provide a common frame containing the most popular extensions of first-order logic that model relational recursion, fixed-point logic and partial fixed-point logic. Unlike these logics themselves, the $L^{k}_{\infty\omega}$ have a very neat and applicable characterization of their expressive power in terms of pebble games, which provide suitable variants of the Ehrenfeucht-Fraïssé characterization of elementary equivalence of structures. On the other hand, the $L^{k}_{\infty\omega}$ are sufficiently rich to define even non-recursive queries, thus being at odds with standard complexity measures. Intuitively, however, it is clear that some notion of polynomiality is involved in these logics, especially as considered as fragments of full-fledged infinitary logic without bounds on the number of variables. In the pebble games this polynomiality is reflected in the fact that positions in the k-pebble game correspond to the designation of up to k elements at a time over the structures involved; in particular there are only polynomialy many different positions in the game over a given finite structure. The point is that $L^k_{\infty\omega}$ can only assess properties of k-tuples, not of longer sequences of elements. But there is another scale of complexity inherent in the structure of the particular formula which cuts across all levels of standard complexity, thus giving rise to the definability of even non-recursive properties.

It is therefore appealing that here we obtain a characterization of $L^k_{\infty\omega}$ that is of a complexity theoretic flavour, and exactly accounts for that kind of polynomiality we observe in $L^\omega_{\infty\omega}$: It turns out to be precisely captured in the notion of local polynomiality of explicitly symmetric circuits.

To finish this introduction, here is a short outline of the organization of the rest of the paper:

Section 2 introduces the basic definitions and facts concerning relational structures, their representations and the kinds of circuits we want to consider. It also contains the formal definition of the logics $L^k_{\infty\omega}$ and some remarks concerning these.

In Section 3 we state and prove the main theorem dealing with circuits and $L^\omega_{\infty\omega}$.

The extension of these results to networks is given in Section 4, where we also give a short introduction to the relevant fixed-point logics.

The reader who is only interested in the circuit case can thus just ignore Section 4. For Section 3, some of the technicalities are avoidable if one only deals with the uniform circuits of infinitary format, and leaves aside the parallel track concerning coherent sequences of circuits of finitary format. I have attempted to separate the corresponding lines of development so that the reader who is comfortable with very infinite circuits can isolate this more elegant treatment and ignore the sequences of circuits.

2 Basic definitions, preliminaries

2.1 Finite structures and their representations

All vocabularies are finite and relational. Usually we think of the predicates in a vocabulary τ as enumerated in some fixed order as $R_1, \ldots, R_l$. Let then r_i stand for the arity R_i.

Definition 2.1 *The class of all finite τ-structures is denoted by $\mathfrak{str}_{\mathrm{fin}}[\tau]$. We write $\mathfrak{str}_n[\tau]$ for the class of τ-structures of size n. $\mathfrak{str}[n,\tau]$ is the class of all τ-structures over the standard universe $n = \{0, \ldots, n-1\}$.* [1]

We shall also consider structures over the standard countably infinite domain $\omega = \{0, 1, 2, \ldots\}$ of the natural numbers. $\mathfrak{str}[\omega, \tau]$ stands for the class of all τ-structures over the universe ω.

[1] For notational convenience, we apply the standard set-theoretic convention of identifying n with the set $\{0, \ldots, n-1\}$.

We write $\mathfrak{A} = \big(A, R_1^{\mathfrak{A}}, \ldots, R_l^{\mathfrak{A}}\big)$ for a τ-structure with universe A. The cardinality of A, $|A|$, is regarded as the size of $\mathfrak{A}$.

For computational purposes, finite structures are encoded or represented as binary strings that can for instance be written down on an input tape of a Turing machine, or applied to the input gates of circuits. The standard procedure is the following: A with $|A| = n$ is identified with the standard domain $n = \{0, \ldots, n-1\}$. In other words we pass from $\mathfrak{A} \in \mathfrak{str}_n[\tau]$ to a structure $\widehat{\mathfrak{A}} \in \mathfrak{str}[n, \tau]$, that is isomorphic with $\mathfrak{A}$. For structures in $\mathfrak{str}[n, \tau]$, canonical representations as strings are induced by canonical enumerations of all instantiations of τ-atoms over n: For each R_i the instantiations of R-atoms $R[\overline{m}]$, $\overline{m} \in n^{r_i}$ can be listed in lexicographic order. The sequence of boolean values corresponding to these atoms over $\widehat{\mathfrak{A}}$ describes $\widehat{\mathfrak{A}}$ completely.

The places in the encoding string can naturally be identified with the union $n^{r_1} \,\dot\cup\, \ldots \,\dot\cup\, n^{r_l}$. Let us introduce the abbreviating notation $\overline{r} = (r_1, \ldots, r_l)$ for the tuple of arities in τ, and $n^{\overline{r}}$ for this set that serves as a canonical parameterization of all instantiations of τ-atoms over the domain n. The encodings of structures in $\mathfrak{str}[n, \tau]$, and thereby indirectly of all structures in $\mathfrak{str}_n[\tau]$ are binary strings labelled by $n^{\overline{r}}$.

In general, however, there is no preferred bijection between A and the standard domain n. The passage from $\mathfrak{A} \in \mathfrak{str}_n[\tau]$ to an isomorphic representative $\widehat{\mathfrak{A}} \in \mathfrak{str}[n, \tau]$ is not well-defined. The main exception is in the case of structures that are linearly ordered themselves: There is an obvious uniquely determined correspondence that translates the given ordering on A into the natural ordering of the standard domain. In the general case, it is an essential implicit condition that computations on different encodings of the same structure – or just as well, of isomorphic structures – lead to the same outcome: *Invariance with respect to representation.* For $\mathfrak{A}$ as above, the ambiguity in the encoding is exactly described by the operation of the symmetric group on n elements, S_n: Any two different encodings are induced by two different bijections between A and n, hence related by a permutation of $n = \{0, \ldots, n-1\}$. S_n acts naturally not only on n, but also on all powers of n, on $n^{\overline{r}}$, and thereby also on strings labelled by $n^{\overline{r}}$. We thus get: Any two binary representations of the same or isomorphic structures of size n are related by permutations of the binary entries corresponding to the natural action of S_n on $n^{\overline{r}}$.

> Invariance with respect to representations of structures in $\mathfrak{str}_n[\tau]$ corresponds to invariance with respect to the action of S_n on binary strings labelled by $n^{\overline{r}}$.

It will be convenient to consider finite τ-structures of arbitrary finite size as embedded into the standard countably infinite domain ω. This can be formalized as follows. Extend τ by a new unary predicate symbol U to obtain $\tau_U := \tau \,\dot\cup\, \{U\}$. With the class of all finite τ-structures associate the following class of countably infinite τ_U-structures with universe ω:

$$\mathfrak{str}[\omega, \tau_U]^* := \big\{ \big(\omega, U, R_1, \ldots, R_l\big) \bigm| U \subset \omega \text{ finite }, R_i \subset U^{r_i} \text{ for } i = 1, \ldots, l \big\} \subset \mathfrak{str}[\omega, \tau_U].$$

The universe of any structure $\big(\omega, U, R_1, \ldots, R_l\big)$ in $\mathfrak{str}[\omega, \tau_U]^*$ is the set ω. The interpretation $U \subset \omega$ of the unary predicate U is a finite subset of ω, and the interpretations of the predicates in τ are restricted to this finite subdomain U. Thus, $\mathfrak{str}[\omega, \tau_U]^*$ provides a *uniform* *representation* of the members of $\mathfrak{str}_{\text{fin}}[\tau]$ as structures embedded into the standard domain ω. For a given finite τ-structure $\mathfrak{A}$ a representation in $\mathfrak{str}[\omega, \tau_U]^*$ is obtained from any injective mapping ρ of its universe A into ω: $\rho(\mathfrak{A}) = \big(\omega, \rho(A), \rho(R)_{R \in \tau}\big) \in \mathfrak{str}[\omega, \tau_U]^*$. Note that these representations are isomorphism-preserving: If f is an isomorphism between $\mathfrak{A}$ and $\mathfrak{A}'$, and ρ and ρ' are injections of A and A' into ω, then the corresponding representations $\rho(\mathfrak{A})$ and $\rho'(\mathfrak{A}')$ are related by the isomorphism $\rho' f \rho^{-1}$, which maps $\rho(A)$ to $\rho'(A')$ according to $\rho' \circ f \circ \rho^{-1}$ and, conversely, $\rho'(A')$ to $\rho(A)$ through $\rho \circ f^{-1} \circ \rho'^{-1}$. Note that this isomorphism corresponds to a permutation of ω with *finite support*, i.e. it only shifts finitely many elements. Isomorphisms between structures in $\mathfrak{str}_{\text{fin}}[\tau]$ are represented as finitary permutations of ω relating their representations in $\mathfrak{str}[\omega, \tau_U]^*$.

Representing structures in $\mathfrak{str}[\omega, \tau_U]^*$ as infinite binary strings corresponding to the canonical enumeration of instantiations of τ_U-atoms over ω, we get binary strings labelled by $\omega^{r_1} \dot\cup \ldots \dot\cup \omega^{r_l} \dot\cup \omega^1$. The last copy of ω is for the unary U-atoms, i.e. its entries correspond to the designation of the finite embedded universe. Similar to the above, let us write $\omega^{\overline{r}+1}$ for this set. As pointed out, the ambiguity in the representation corresponds to the action of finitary permutations of ω. We write S_ω for the group of all permutations of ω, and S_{fin} for the subgroup of those permutations that have finite support.

> Invariance with respect to representation of $\mathfrak{str}_{\text{fin}}[\tau]$-structures in $\mathfrak{str}[\omega, \tau_U]^*$ corresponds to invariance with respect to the natural action of S_{fin} on binary strings labelled by $\omega^{\overline{r}+1}$.

2.2 The logics

The logics we shall consider in connection with circuits are first-order logic $L_{\omega\omega}$ and its infinitary variants with bounded number of variables. Infinitary logic $L_{\infty\omega}$ has the first-order rules of construction of formulae augmented with disjunctions and conjunctions over arbitrary sets of formulae. Restrictions are obtained by limiting the number of variables that may occur (bound and free). $L^k_{\infty\omega}$ has variable symbols $x_1, \ldots, x_k$ only. $L^\omega_{\infty\omega}$ is the union of these restrictions, $L^\omega_{\infty\omega} = \bigcup_k L^k_{\infty\omega}$. The following examples serve to illustrate the expressive power of the $L^k_{\infty\omega}$:

— Over linear orderings $(A, <)$ two different variable symbols suffice to produce first-order formulae $\varphi_i(x)$, for $i \geq 1$, expressing that x is the i-th element with respect to $<$. We use variable symbols x and y. The formula $\varphi_1(x) := \neg\exists y(y < x)$ defines the first element. Inductively, $\varphi_{i+1}(x) := \neg\varphi_i(x) \wedge \forall y\big(y < x \rightarrow \bigvee_{j \leq i} \varphi_j(y)\big)$ is as desired, where $\varphi_j(y)$ is the result of exchanging x and y throughout the formula φ_j. Since three variables suffice to axiomatize linear orderings, we find that sentences in $L^3_{\infty\omega}$ can have arbitrarily complex and even non-recursive classes of finite models: Let χ be an axiomatization of linear orderings in $L^3_{\infty\omega}$. For any set $W \subset \omega \setminus \{0\}$, the sentence $\varphi_W := \chi \wedge \bigvee_{i \in W} \exists x\big(\varphi_i(x) \wedge \neg\exists y(x < y)\big)$ describes exactly those linear orderings, whose size is in W. We mention that actually two variables suffice to define some non-recursive properties.

— To indicate in another example how clever re-use of the finite supply of variables can capture some relational recursion, consider the definition of the transitive closure of a binary relation E. The formula $\psi_1(x, y) := x = y \vee Exy$ describes the pairs of E-distance 1.

Inductively, $\psi_{i+1}(x,y) := \psi_i(x,y) \vee \exists z\big(\psi_i(x,z) \wedge Ezy\big)$ defines those pairs (x,y), whose E-distance is at most $i+1$. Thus $\xi(x,y) := \bigvee_{i\geq 1}\psi(x,y)$ defines the transitive closure of E.

The last example is typical in the sense that in fact all relational recursion can be defined in $L^{\omega}_{\infty\omega}$. This is the main reason that these bounded variable fragments of infinitary logic play such an important rôle in finite model theory: They provide a common frame for the several logics extending first-order by fixed-point operations. These, and in particular partial fixed-point logic, equivalent with the relational language WHILE, will be introduced in connection with the analysis of certain boolean networks in Section 4.

The formula rank for $L^{\omega}_{\infty\omega}$ is defined by induction as follows:

$$\begin{aligned}
&\mathrm{rk}(\varphi) = 0 \quad \text{for atomic } \varphi;\\
&\mathrm{rk}(\neg\varphi) = \mathrm{rk}(\exists x\varphi) = \mathrm{rk}(\forall x\varphi) = \mathrm{rk}(\varphi) + 1;\\
&\mathrm{rk}(\textstyle\bigwedge_{i\in I}\varphi_i) = \mathrm{rk}(\bigvee_{i\in I}\varphi_i) = \sup_{i\in I}\big(\mathrm{rk}(\varphi_i) + 1\big).
\end{aligned}$$

Obviously, the usual quantifier rank is bounded by the formula rank, and over finite vocabularies, it is easily shown that any formula of $L^{k}_{\infty\omega}$ of finite quantifier rank is equivalent with a finitary formula in $L^{k}_{\infty\omega}$, i.e. with a first-order formula using only k variable symbols.

Lemma 2.1 *The following are equivalent in expressive power with respect to finite vocabularies: The finite quantifier rank fragment of $L^{k}_{\infty\omega}$, the finite formula rank fragment of $L^{k}_{\infty\omega}$, and first-order logic with only k variable symbols.*

Let $Q \subset \mathfrak{str}_{\mathrm{fin}}[\tau]$ be a class of finite τ-structures, closed under isomorphism of course. Recall how the structures in $\mathfrak{str}_{\mathrm{fin}}[\tau]$ relate to their representations in $\mathfrak{str}[\omega,\tau_U]^*$, and let $Q^U \subset \mathfrak{str}[\omega,\tau_U]^*$ be the class of all the representations thus obtained from structures in Q. It is not difficult to see that, for natural logics, definability of Q over $\mathfrak{str}_{\mathrm{fin}}[\tau]$ is equivalent with definability of Q^U over $\mathfrak{str}[\omega,\tau_U]^*$. We formally state this fact and indicate its proof for $L^{k}_{\infty\omega}$. The analogous result for first-order logic (with k variable symbols) is an immediate consequence of the proof.

Lemma 2.2 *Let $Q \subset \mathfrak{str}_{\mathrm{fin}}[\tau]$ and $Q^U \subset \mathfrak{str}[\omega,\tau_U]^*$ as above. Then the following are equivalent:*

(i) There is a sentence $\varphi \in L^{k}_{\infty\omega}[\tau]$ such that $Q = \big\{\mathfrak{A} \in \mathfrak{str}_{\mathrm{fin}}[\tau] \;\big|\; \mathfrak{A} \models \varphi\big\}$.

(ii) There is a sentence $\varphi^* \in L^{k}_{\infty\omega}[\tau_U]$ such that $Q^U = \big\{\mathfrak{A} \in \mathfrak{str}[\omega,\tau_U]^* \;\big|\; \mathfrak{A} \models \varphi^*\big\}$.

Sketch of proof

The direction from (i) to (ii) is an application of the *relativization property* for $L^{k}_{\infty\omega}$. Inductively, the relativization of $\chi(\overline{x})$ to U, χ^U, is defined as follows: $\chi^U = \chi$ for atomic formulae, $\cdot^U$ trivially commutes with boolean connectives, and $(\exists x\chi)^U = \exists x(Ux \wedge \chi^U)$, $(\forall x\chi)^U = \forall x(Ux \rightarrow \chi^U)$. It is immediate through syntactic induction, that for all formulae $\chi(\overline{x})$, for all $\overline{a}$ in a structure $\mathfrak{A} \in \mathfrak{str}_{\mathrm{fin}}[\tau]$, and for any injection ρ that gives a representation $\rho(\mathfrak{A})$ of $\mathfrak{A}$ in $\mathfrak{str}[\omega,\tau_U]^*$:

$$\mathfrak{A} \models \chi[\overline{a}] \;\text{ iff }\; \rho(\mathfrak{A}) \models \chi^U[\rho(\overline{a})].$$

If φ defines Q according to (i), then $\varphi^* := \varphi^U$ defines Q^U according to (ii).

The converse really amounts to saying that any formula that, as in (ii), defines a class Q_U must be equivalent with the relativization of a formula $\varphi \in L^k_{\infty\omega}[\tau]$. Then the equivalence displayed above shows that φ is as desired for (i).

The crucial equivalence with a relativization can either be derived from Gaifman's Theorem, or be proved in an ad-hoc manner inductively as follows. Let $\chi(\overline{x}) \in L^k_{\infty\omega}[\tau_U]$. We claim that over $\mathfrak{str}[\omega, \tau_U]^*$, χ is equivalent with a disjunction of formulae of the form $\varphi^U(\overline{x}_0) \wedge \eta(\overline{x})$, where $\varphi \in L^k_{\infty\omega}[\tau]$, $\overline{x}_0 \subset \overline{x}$, η is quantifier-free and involves only U and equality. To this end, split $\chi(\overline{x})$ into a disjunction corresponding to cases with respect to which of the free variables are interpreted as being outside the U-part. Let $\eta(\overline{x})$ be a quantifier-free type in the language consisting of equality and U, $\overline{x}_0$ the tuple of those variables from $\overline{x}$ of which η asserts that they are in the U-part, $\overline{x}_1$ the rest. Obviously χ is equivalent with the disjunction over the $\chi \wedge \eta$ for all such types η. In an easy induction over χ one shows that, over structures in $\mathfrak{str}[\omega, \tau_U]^*$, i.e. on structures that are trivial outside the U-part, $\chi \wedge \eta$ is equivalent with a formula of the form $[\psi(\overline{x}_0)]^U \wedge \eta$. Since φ^* of (ii) is a sentence without free variables, this gives its direct equivalence with a relativization φ^U over all structures in $\mathfrak{str}[\omega, \tau_U]^*$. Q.E.D.

2.3 Boolean circuits

We generally consider circuits of unbounded fan-in corresponding to the boolean connectives $\bigwedge$ (for arbitrary conjunctions), $\bigvee$ (for arbitrary disjunctions), and $\neg$ (for negation). Thus a *boolean circuit* is a connected and acyclic directed graph, whose nodes are labelled as follows:

- Nodes of in-degree 0 are of two types: Either they are regarded as *input nodes* and are unambiguously labelled by a set of *input labels*, or they correspond to boolean constants and carry labels T or F.
- Nodes of in-degree 1 may carry the label $\neg$ (for negation) or no label (for identity).
- Nodes of in-degree greater than 1 carry labels $\bigwedge$ or $\bigvee$ (for conjunction or disjunction).
- Usually we consider a single marked node as the *output node*.

Due to acyclicity there is a well defined depth function. The depth of the input nodes is 0, the depth of any other node is the supremum (in the strict sense) of the depths of all its direct predecessors.

After setting the input nodes to certain boolean values, truth values are propagated throughout the circuit in the standard fashion (the formal definition is by induction on depth). The labellings of input nodes that arise in our considerations are indicated in the following. Recall that $n = \{0, \dots, n-1\}$, and that $\overline{r}$ is the tuple of arities of the predicates in τ.

Definition 2.2 *(i) A boolean circuit is* formatted for $\mathfrak{str}[n, \tau]$ *if its input nodes are injectively labelled by the set* $n^{\overline{r}} = n^{r_1} \dot{\cup} \dots \dot{\cup} n^{r_l}$. *For short, we talk of an* $[n, \tau]$-circuit.

(ii) A boolean circuit is formatted for $\mathfrak{str}[\omega, \tau]$, *or is an* $[\omega, \tau]$-circuit *for short, if its input nodes are injectively labelled by the set* $\omega^{\overline{r}} = \omega^{r_1} \dot{\cup} \dots \dot{\cup} \omega^{r_l}$.

*(iii) A sequence $(C_n)_{n\geq 1}$ of circuits, where C_n is formatted for $\mathfrak{str}[n,\tau]$, will be termed a $[n,\tau]$-*sequence.

Note that a circuit formatted for $\mathfrak{str}[\omega,\tau]$ is necessarily infinite, whereas circuits formatted for $\mathfrak{str}[n,\tau]$ may or may not be finite. The input fields, in all cases, are adapted to taking the binary representations of respective structures, as outlined above, as inputs: The sets of input labels parameterize the set of all instances of atoms over the corresponding standard universe. We are interested in circuits that are appropriate for the evaluation of boolean queries for finite τ-structures. To this end we consider
(a) single infinite circuits formatted for $\mathfrak{str}[\omega,\tau_U]$, and meant to evaluate $Q^U \subset \mathfrak{str}[\omega,\tau_U]^*$, or
(b) $[n,\tau]$-sequences that evaluate Q in the usual sense that the n-th member of the sequence evaluates $Q_n = Q \cap \mathfrak{str}_n[\tau]$.
This leads to the following semantic interpretations of circuits:

Definition 2.3 *(i) An $[\omega,\tau_U]$-circuit C* computes the boolean query *Q on $\mathfrak{str}_{\rm fin}[\tau]$, if its output node computes the characteristic function of Q for all binary representations of finite τ-structures via $\mathfrak{str}[\omega,\tau_U]^*$-structures.*

(ii) An $[n,\tau]$-circuit C_n computes the boolean query *Q_n on $\mathfrak{str}_n[\tau]$, if its output node computes the characteristic function of Q_n on all representations of finite τ-structures via $\mathfrak{str}[n,\tau]$.*

(iii) An $[n,\tau]$-sequence $(C_n)_{n\geq 1}$ of circuits computes the boolean query *Q on $\mathfrak{str}_{\rm fin}[\tau]$, if, for each n, C_n computes the query $Q_n := Q \cap \mathfrak{str}_n[\tau]$.*

Observe, that the underlying boolean functions are functions on the domain $\{0,1\}^{\omega^{\overline{r}+1}}$ in (i), on $\{0,1\}^{n^{\overline{r}}}$ in (ii), and on $\bigcup_{1\geq 1}\{0,1\}^{n^{\overline{r}}}$ in (iii).

We are interested in symmetric circuits. Symmetry is expressed in terms of the automorphism group of the circuit. Over the input field we want to consider those permutations of labels as *automorphisms*, that are induced by permutations of the standardized universes, n or ω. Recall the action of S_n and S_ω, respectively, on $n^{\overline{r}}$, or on $\omega^{\overline{r}}$ and $\omega^{\overline{r}+1}$.

Definition 2.4 *Let C be an $[n,\tau]$- or an $[\omega,\tau]$-circuit. An* automorphism *of C is an automorphism of the underlying graph which respects the labelling in the following sense:
The labels of all non-input nodes must be preserved, on the input nodes an automorphism must correspond to an automorphism of the labelling; the output node has to be fixed.*

The above considerations about the ambiguity of the input representation showed that isomorphisms between finite τ-structures translate to automorphisms of the input labellings. In the case of representations in $\mathfrak{str}[\omega,\tau_U]^*$, furthermore we only need consider those permutations of ω that fix all but finitely many points, i.e. we restrict attention to the action of the subgroup $S_{\rm fin} \subset S_\omega$ of permutations of finite support.

Definition 2.5 *Let C be an $[n,\tau]$- or an $[\omega,\tau]$-circuit. C is* explicitly symmetric*, or* symmetric *for short, if the action of S_n or $S_{\rm fin}$ on the respective input field extends to an automorphic action on the entire circuit C.*

Note that symmetry of a circuit is not a semantic notion here! It is a purely combinatorial requirement concerning the graphical layout of circuits. It is the natural notion of this type, however, that guarantees invariance of the computed function with respect to representation or encoding. The latter, by contrast, is a semantic notion.

In fact, symmetry implies that the entire evaluation of the circuit becomes invariant with respect to representation: *Isomorphic inputs yield isomorphic computations.* This requirement is the essence of so-called *generic* models of computation. The following is obvious, but deserves explicit mention:

Proposition 2.1 *The following are equivalent for a boolean function f on $\{0,1\}^{\omega^{\overline{r}+1}}$:*

- *(i) f is the characteristic function of a boolean query $Q \subset \mathfrak{str}_{\text{fin}}[\tau]$ (in the sense of Definition 2.3 (i).*
- *(ii) f can be computed by an explicitly symmetric $[\omega, \tau_U]$-circuit.*

Similarly, the following are equivalent for a boolean function f on $\bigcup_{1 \geq 1}\{0,1\}^{n^{\overline{r}}}$:

- *(i) f is the characteristic function of a boolean query $Q \subset \mathfrak{str}_{\text{fin}}[\tau]$ (in the sense of Definition 2.3 (iii).*
- *(ii) f can be computed by an $[n, \tau]$-sequence $(C_n)_{n \geq 1}$ of explicitly symmetric circuits C_n.*

For the implications (i) $\Rightarrow$ (ii) observe that any boolean circuit computing f can be symmetrized, i.e. made explicitly symmetric, at the expense of introducing a host of new nodes. Essentially, the original circuit is superposed with all its images under the operation of the required symmetry group. Finally a new output node is added in such a way that it computes the disjunction (or conjunction) over all the images of former output nodes that were produced in the symmetrization. If f computes the characteristic function of a query, then all these copies of former output nodes must compute the same value, whence their conjunction and disjunction gives the value of f itself. Note that this ad-hoc symmetrization in general leads to an exponential blow-up in size — or in width, since we only add copies of nodes at the same depth.

In general the action of S_{fin} or S_n on C need not be unique, i.e. there could be different lifts of the group operation on the input field to the rest of the circuit. It is easily shown, however, that this can only be due to a redundancy in the layout of the circuit that can be eliminated through suitable factorization.

Proviso 1 *We shall assume without loss of generality that the explicitly symmetric circuits under consideration satisfy the following condition. For any automorphism π of C and any node $v \in C$: If π fixes the set of direct predecessors of v, then π fixes v itself. This implies in particular that any automorphism of C is induced by a unique automorphism of the input labelling.*

As pointed out above, symmetrization can in general be expected to lead to exponential numbers of isomorphic copies of the same node. Turning this expected trade-off between explicit symmetry and orbit size of nodes under the automorphism group into a criterion, we consider circuits that can be *symmetrized within polynomial bounds*, or symmetric circuits for which the orbits under automorphisms grows only polynomially with the size of the support of these automorphisms. For a formal definition of this notion of *local polynomiality* we introduce some notation concerning the group actions.

For finite $s \subset \omega$, put $S_s := \{\pi \in S_{\text{fin}} \mid \text{supp}(\pi) \subset s\} \subset S_{\text{fin}} \subset S_\omega$. Note that this in agreement with the notation S_n for the permutation group on $n = \{0, \ldots, n-1\}$. If H is any subgroup of S_ω that operates on C and v is a node of C, then $v^H := \{\pi(v) \mid \pi \in H\}$ stands for the orbit of v under H.

Definition 2.6 *Let C be an explicitly symmetric $[\omega, \tau]$-circuit. C is* locally polynomial *if there is a polynomial p such that for all finite $s \subset \omega$ and for all nodes v of C: $|v^{S_s}| \leq p(|s|)$.*

Similarly, an $[n, \tau]$-sequence of symmetric circuits $(C_n)_{n \geq 1}$ is locally polynomial *if there is one polynomial p giving a uniform bound on the orbit size across the whole sequence in the sense that for all n, for all $s \subset n$, and for all nodes v of C_n: $|v^{S_s}| \leq p(|s|)$.*

Note that, when we speak of locally polynomiality of circuits, explicit symmetry will always be understood.

Finally we come to considerations about *uniformity* in our framework of circuit computation. It is clear that computation of a query by a symmetric infinite $[\omega, \tau_U]$-circuit as in Definition 2.3 (i) induces some kind of uniformity: There is one infinite and extremely homogenous circuit for the computation of the query. The following definition isolates a corresponding uniformity criterion at the level of sequences of symmetric circuits.

The criterion is in terms of *embeddings* of one circuit into another. An embedding of C into C' is an injective mapping $h: C \to C'$, that preserves labels and edges: For $v \in C$, $h(v)$ must carry the same label in C' as v in C; for $v_1, v_2 \in C$, there is an edge (wire) connecting v_1 to v_2 in C if and only if there is an edge from $h(v_1)$ to $h(v_2)$ in C'.

Consider two members of an $[n, \tau]$-sequence of symmetric circuits, C_n and C_m, $n < m$. Note that the input field of C_n is a subset of the input field of C_m. Roughly, we want to consider a node of C_m as belonging to the smaller input field if it is fixed by all permutations of $m \setminus n$, i.e. fixed by $S_{m \setminus n}$. We call an embedding of C_n into C_m *complete* if its range consists of all nodes of C_m that are fixed by $S_{m \setminus n}$.

Definition 2.7 *A $[n, \tau]$-sequence $(C_n)_{n \geq 1}$ of symmetric circuits is called* coherent *if for all n, there is a complete embedding of C_n into C_m for all sufficiently large $m > n$.*

The following correspondence between the two notions of uniformity — a single infinite locally polynomial symmetric circuit vs. a coherent locally polynomial sequence of symmetric circuits — will be a consequence of our main theorem, i.e. we shall prove it via a detour involving the logical characterization through $L^\omega_{\infty\omega}$:

Proposition 2.2 *The following are equivalent for any boolean query $Q \subset \mathfrak{str}_{\text{fin}}[\tau]$:*

- *(i) Q can be computed by a coherent and locally polynomial $[n, \tau]$-sequence $(C_n)_{n \geq 1}$ of symmetric circuits.*
- *(ii) Q can be computed by a single symmetric and locally polynomial $[\omega, \tau_U]$-circuit.*

3 Boolean circuits and $L^{\omega}_{\infty\omega}$

Here is our main theorem dealing with the equivalence between $L^{\omega}_{\infty\omega}$ and locally polynomial symmetric circuits.

Theorem 3.1 *The following are equivalent for any boolean query $Q \subset \mathfrak{str}_{\mathrm{fin}}[\tau]$:*

- *(i) Q can be computed in a coherent locally polynomial $[n, \tau]$-sequence of explicitly symmetric circuits $(C_n)_{n\geq 1}$.*
- *(ii) Q can be computed in a locally polynomial symmetric $[\omega, \tau_U]$-circuit.*
- *(iii) Q is the class of finite models of a sentence of $L^{\omega}_{\infty\omega}$, i.e. Q is definable in $L^{\omega}_{\infty\omega}$.*

The proof will immediately yield the following in restriction to finite depth:

Theorem 3.2 *The following are equivalent for any boolean query $Q \subset \mathfrak{str}_{\mathrm{fin}}[\tau]$:*

- *(i) Q can be computed in a coherent locally polynomial $[n, \tau]$-sequence of explicitly symmetric finite circuits $(C_n)_{n\geq 1}$ of constant depth.*
- *(ii) Q can be computed in a locally polynomial symmetric $[\omega, \tau_U]$-circuit of finite depth.*
- *(iii) Q definable in first-order logic.*

Most of this section is devoted to the proof of the main theorem. The easy part consists of showing that the natural circuit expansions of $L^{\omega}_{\infty\omega}$-formulae lead to symmetric and locally polynomial circuits.

Locally polynomial symmetric circuits for $L^{\omega}_{\infty\omega}$

By induction over formulae in $L^{\omega}_{\infty\omega}$, we show the following:

Lemma 3.1 *Let $\varphi(\overline{x})$ be a formula in $L^{k}_{\infty\omega}[\tau]$ of formula rank α. Then there are:*

- *(a) A locally polynomial $[\omega, \tau]$-circuit, whose nodes correspond in a one-to-one fashion to the instantiations of atomic formulae or subformulae of φ over ω. Two nodes of this circuit are related by an automorphism of the circuit if and only if they correspond to S_{fin}-related instantiations of the same subformula. The node corresponding to the instantiation $\psi[\overline{m}]$ computes, on each input $\mathfrak{A} \in \mathfrak{str}[\omega, \tau]$, the boolean value of $\mathfrak{A} \models \psi[\overline{m}]$. The depth of this circuit is equal to the formula rank α of φ.*
- *(b) A locally polynomial coherent $[n, \tau]$-sequence of circuits of depth α, $(C_n)_{n\geq 1}$, such that the nodes of C_n correspond to the instantiations of atomic formulae and the subformulae of φ over n. Again, two nodes of C_n are related by an automorphism of C_n if and only if they correspond to S_n-related instantiations of the same subformula. The node corresponding to $\psi[\overline{m}]$ in C_n computes the boolean value of $\mathfrak{A} \models \psi[\overline{m}]$ on each input $\mathfrak{A} \in \mathfrak{str}[n, \tau]$.*

Note that an application of these statements to sentences of $L^{\omega}_{\infty\omega}$ proves the implications (iii) $\Rightarrow$ (i) and (iii) $\Rightarrow$ (ii) in the main theorem: This is immediate in the statement involving the sequences, since these already have the appropriate format. For (iii) $\Rightarrow$ (ii), involving $\mathfrak{str}[\omega, \tau_U]^*$, we have to apply (i) of the lemma to the relativization φ^U of the sentence φ that defines Q. Lemma 2.2 shows that the resulting $[\omega, \tau_U]$-circuit is adequate for Q.

Since any formula of $L^{\omega}_{\infty\omega}$ of finite formula or quantifier rank is equivalent with a first-order formula and vice versa, cf. Lemma 2.1, the corresponding implications in Theorem 3.2 follow as well.

Proof
The constructions of the circuits are by induction over formulae. We treat (i) first. The claim is obvious for atomic formulae φ. The desired circuits consist just of the input nodes. Note that the claim carries over to boolean combinations of formulae immediately. It only remains to treat the existential step.

Let $\psi(\overline{x}) = \exists z \chi(z, \overline{x})$. Assume that C is a circuit for $\chi(z, \overline{x})$ meeting the requirements, in particular its depth is equal to the formula rank of χ. The circuit for ψ is obtained trough the addition of one new level to C. Let the nodes in the new level be disjunctive nodes labelled by the instantiations $\psi[\overline{m}]$ of ψ over ω. The node labelled $\psi[\overline{m}]$ is connected to the top-level nodes of C that correspond to the instantiations $\chi[m, \overline{m}]$ of χ, for all $m \in \omega$. It is easy to check that the resulting circuit is as desired.

We turn to the coherent sequence required in (ii). It is easiest to obtain its members as restrictions of the infinite circuit obtained for the evaluation of φ over $\mathfrak{str}[\omega, \tau]$ according to (i). Let C be this $[\omega, \tau]$-circuit that is constructed according to the proof of (i). The desired coherent sequence of circuits consists of suitable restrictions of this single circuit C. To obtain C_n, restrict C to nodes whose associated instantiations of subformulae exclusively involve parameters from n. It is immediate by induction over the subformulae of φ that the nodes of C_n compute what they should. Local polynomiality and symmetry carry over from C. From the way all C_n are embedded into C it follows directly that the coherence condition is satisfied. Q.E.D.

$L^{\omega}_{\infty\omega}$-definability for locally polynomial symmetric circuits

In preparation for the proof of the important direction in the main theorem, we first isolate the crucial combinatorial core of the matter. The following lemma connects local polynomiality with definability over few parameters in an abstract setting.

Some notation to facilitate the statement and proof of the lemma: Let, for any tuple $\overline{m}$ the set of its components be denoted by $[\overline{m}]$. Put $G := S_{\mathrm{fin}}$, the group of permutations of ω with finite support. For any object O that can be subject to the action of G, let $G(O)$ be the subgroup that fixes O. We apply this notation in particular to predicates R over ω and to tuples $\overline{m}$ or their component sets $[\overline{m}]$. Recall that S_s, for finite $s \subset \omega$, is the group of permutations of ω whose support is contained in s.

Lemma 3.2 *Let $R \subset \omega^r$ be an r-ary predicate over ω. Assume that R is locally polynomial in the sense that for all finite $s \subset \omega$, the orbit of R under S_s is bounded by a polynomial in* *the size of s. I.e. there is a polynomial p of degree k, such that for all s: $\left|R^{S_s}\right| \leq p(|s|)$. Let k be the degree of p.*

Then there is a tuple of at most k distinct parameters $\overline{m}$ from ω such that R is definable over these: There is a quantifier free formula $\eta(x_1, \ldots, x_r, \overline{z})$ involving only equality such that $R = \{\overline{b} \in \omega^r \mid \eta[\overline{b}, \overline{m}]\}$. The parameter tuple $\overline{m}$ can be chosen with pairwise distinct components and such that

$$G(\overline{m}) \subset G(R) \subset G([\overline{m}]).$$

The following is a simplified proof. A preliminary version of this paper contained an unnecessarily complicated argument. I am grateful to Martin Grohe for pointing that out to me.

Proof

Generally $G(\overline{m}) \subset G([\overline{m}]) \subset G$. Observe that definability of R relative to parameters $\overline{m}$ is indeed equivalent with $G(\overline{m}) \subset G(R)$. To indicate the argument for the non-trivial implication, assume that $G(\overline{m}) \subset G(R)$. Let η be the disjunction over all equality types of tuples $(\overline{b}, \overline{m})$, $\overline{b} \in R$. A simple automorphism argument shows that $R = \{\overline{b} \mid \eta[\overline{b}, \overline{m}]\}$.

The proof of the lemma is given in two steps. Under the assumptions of the lemma, we show:

(i) There is a finite tuple $\overline{m}'$ such that R is fixed relative to $\overline{m}'$: $G(\overline{m}') \subset G(R)$.

(ii) For $\overline{m}'$ as in (i) there is a subtuple $\overline{m}$ such that $G(\overline{m}) \subset G(R) \subset G([\overline{m}])$.[2]

Note that a tuple $\overline{m}$ as in (ii) satisfies the claims of the lemma: Definability follows from $G(\overline{m}) \subset G(R)$; $G(R) \subset G([\overline{m}])$, on the other hand, implies that R is shifted by any permutation that does not preserve $\overline{m}$ setwise. As there are $\binom{|s|}{|[\overline{m}]|}$ choices for $[\overline{m}]$ over s, local polynomiality with p of degree k implies that the size of $[\overline{m}]$ is at most k.

The proof of (i) is easy. Assume to the contrary that there is no finite parameter set that fixes R. This implies the existence of a sequence of finite subsets $s_1 \subset s_2 \subset \ldots \subset \omega$ such that $|s_i| \leq 2ri$ together with permutations $\pi_i \in G$, such that π_i has support in s_{i+1}, fixes s_i pointwise, but does not fix $R|s_{i+1}$: Since R is not fixed relative to s_i, it suffices to choose an r-tuple that is in R and can be mapped to a tuple outside R without moving points in s_i. Let π_i be the permutation exchanging these two tuples, and join the components of these two r-tuples to s_i to obtain s_{i+1}. But now the π_i commute, so that for each subset $I \subset \{1, \ldots, d\}$, R is mapped to a different isomorphic copy of itself by $\pi_I := \prod_{i \in I} \pi_i$. Thus the orbit of R under $S_{s_{d+1}}$ has at least 2^d elements. In other words, the orbits of R grow exponentially, contradicting local polynomiality.

In order to prove (ii), it suffices to prove the following:

$(*)$ If R is fixed by all permutations that fix one of two finite sets, s_1 or s_2, pointwise, then R is fixed by all permutations that fix $s_1 \cap s_2$ pointwise.

[2] Note that we have to admit the empty tuple for $\overline{m}$, which occurs for fully S_ω-invariant R.

This suffices to reduce a finite tuple $\overline{m}'$ such that $G(\overline{m}') \subset G(R)$ until a tuple $\overline{m}$ with $G(\overline{m}) \subset G(R) \subset G([\overline{m}])$ is obtained: If $\pi \in G(R)$, then also $G(\pi(\overline{m}')) \subset G(R)$ (by conjugation with π), i.e. R is fixed by all permutations that fix $\overline{m}'$ and also by those that fix $\pi(\overline{m}')$. Thus, by $(*)$, R is fixed by all permutations that fix $[\overline{m}'] \cap \pi([\overline{m}'])$ pointwise. If $\pi \in G(R) \setminus G([\overline{m}'])$, then $[\overline{m}'] \cap \pi([\overline{m}'])$ is a proper subset of $[\overline{m}']$.

For the proof of $(*)$, let $s_i = [\overline{m}_i]$ and assume that $G(\overline{m}_i) \subset G(R)$ for $i = 1, 2$. W.l.o.g. let the $\overline{m}_i$ be tuples of pairwise distinct components, of the same length and such that $\overline{m}_i = (\overline{m}_0, \overline{u}_i)$ where $[\overline{u}_1] \cap [\overline{u}_2] = \emptyset$. So $s_1 \cap s_2 = [\overline{m}_0]$. Let $\overline{u}_0$ be any tuple of pairwise distinct components of the same length as the $\overline{u}_i$, $i = 1, 2$, but disjoint from these and from $\overline{m}_0$. Let $(\overline{u}_i, \overline{u}_j)$ denote the permutation that exchanges $\overline{u}_i$ and $\overline{u}_j$ and fixes all other points. It is obvious that $(\overline{u}_0, \overline{u}_1)$ fixes $\overline{m}_2$, and $(\overline{u}_0, \overline{u}_2)$ fixes $\overline{m}_1$, whence these permutations both fix R. Through conjugation with $(\overline{u}_0, \overline{u}_2)$ we find that $G(\overline{m}_0, \overline{u}_0)$ fixes R. Repeated application of this argument shows that $G(\overline{m}_0, \overline{u}) \subset G(R)$ for any tuple $\overline{u}$ of pairwise distinct components that is of the same length as $\overline{u}_0$ and disjoint from $\overline{m}_0$. Similarly $(\overline{u}, \overline{u}')$ fixes R for any two such tuples.

Let now $\rho \in G(\overline{m}_0)$. Choose $\overline{u} := \rho^{-1}(\overline{u}_1)$. Then $\rho \circ (\overline{u}_1, \overline{u})$ fixes $\overline{m}_1 = (\overline{m}_0, \overline{u}_1)$, therefore fixes R. Since $(\overline{u}_1, \overline{u})$ also fixes R, it follows that ρ fixes R. Q.E.D.

Note that in the proof we do not really need the infinite domain ω but could do with a sufficiently large finite domain for R. Sufficiently large, here, is to be understood in terms of the polynomial p and the arity r of R.

Corollary 3.1 *If $R \subset n^r$ is locally polynomial in terms of a polynomial p of degree k, and if n is sufficiently large in relation to r and p, then there is a tuple of at most k distinct parameters $\overline{m}$ from n such that R is definable over these. This tuple $\overline{m}$ can be chosen with pairwise distinct components and such that $G(\overline{m}) \subset G(R) \subset G([\overline{m}])$.*

We are now in a position to prove $L^{\omega}_{\infty\omega}$-definability of the boolean values computed by locally polynomial circuits. We first treat single circuits C, where C is either an $[\omega, \tau]$-circuit or an $[n, \tau]$-circuit. In the latter case assume that n is sufficiently large in relation to the polynomial that bounds the orbit size so that at the crucial step in the proof Corollary 3.1 applies.

Fix a locally polynomial $[\omega, \tau]$- or $[n, \tau]$-circuit C. Let the polynomial which bounds the orbit size be of degree k. Let G be the appropriate symmetry group for C, S_{fin} or S_n. Let $\{w_j \mid j \in J\}$ be a system of representatives of all nodes in C up to the operation of G, i.e. this set contains exactly one member of each orbit under G. Let $O_j = w_j^G$ be the orbit of w_j under G. As above, we write $G(\overline{m})$ and $G([\overline{m}])$ for the subgroups that fix $\overline{m}$ or the set of its components, similarly $G(v)$ for the subgroup fixing a node v of C.

Claim 1

(i) For every node $v \in C$ there is a tuple $\overline{m} \in \omega$ (if C is of $[\omega, \tau]$-format) or $\overline{m} \in n$ (if C is of $[n, \tau]$-format) of at most k components, pairwise distinct, such that

$$G(\overline{m}) \subset G(v) \subset G([\overline{m}]).$$

We shall call such tuple a base *for v.*

It follows from symmetry, that then also, for all $\pi \in G$: $G(\pi(\overline{m})) \subset G(\pi(v)) \subset G(\pi([\overline{m}]))$, i.e. that $\pi(\overline{m})$ is a base for $\pi(v)$.

(ii) Fix bases $\overline{m}_j$ for the w_j for each $j \in J$. Then the following predicates, that describe the edge relation between nodes, are quantifier-free equality definable over ω or n for each pair of indices $j, j' \in J$:

$$R_{jj'} := \big\{(\pi(\overline{m}_j), \pi'(\overline{m}_{j'})) \bigm| \pi, \pi' \in G, \pi(w_j) \in \mathrm{pred}(\pi'(w_{j'}))\big\}.$$

Note that we have to admit arity 0, or empty tuples, for the bases $\overline{m}_j$. If both, $\overline{m}_j$ and $\overline{m}_{j'}$ are empty, then $R_{jj'}$ is to be regarded as a boolean value (and we stretch the notion of quantifier free definability accordingly).

Proof

Note that the definability claimed in (ii) is almost trivial as a consequence of (i), symmetry and the definition of the $R_{jj'}$: It suffices to check that the $R_{jj'}$ are invariant under the operation of G, i.e. invariant under finitary permutations. Any predicate with this invariance property is definable in terms of equality types, as is shown by a standard argument similar to that at the very beginning of the proof of Lemma 3.2.

The explicit statement of (ii), however, is justified by its rôle in the proof. (i) and (ii) are proved simultaneously in an induction over the depth of nodes.

— For depth 0 we are either dealing with an input node, which corresponds to an instantiation of an atom, or with a boolean constant in a node labelled T or F. As a base we take the parameters of the instantiation in the first case, and the empty tuple in the case of a node labelled by a constant.

— Consider now v of depth $\alpha > 0$. W.l.o.g. assume that $v = w_{j_0}$ for some $j_0 \in J$. Note that all predecessors of v must be in orbits O_j whose depth is strictly less than α. Let J_0 be the set of indices j for which the orbit of w_j contains direct predecessors of v. We assume that statement (i) of the claim holds for all nodes of depth less then α, and that tuples $\overline{m}_j$ satisfying (i) for the w_j, $j \in J_0$ are fixed.

Define a predicate

$$R_j := \big\{\pi(\overline{m}_j) \bigm| \pi \in G, \pi(w_j) \in \mathrm{pred}(v)\big\}.$$

With this definition we get, for all $\sigma \in G$, $j \in J_0$:

$$\sigma(w_j) \in \mathrm{pred}(v) \;\Leftrightarrow\; \sigma(\overline{m}_j) \in R_j.$$

Only the direction from right to left is of interest. So assume that $\sigma(\overline{m}_j) \in R_j$, i.e. for some π: $\pi(\overline{m}_j) \in \mathrm{pred}(v)$ and $\pi(\overline{m}_j) = \sigma(\overline{m}_j)$. It follows that $\sigma^{-1} \circ \pi$ fixes $\overline{m}_j$, whence it fixes w_j (by the inductive hypothesis and the choice of $\overline{m}_j$ for w_j). But this implies that $\sigma(w_j) = \pi(w_j) \in \mathrm{pred}(v)$.

From this equivalence we also get, for any $\rho \in G$, $j \in J_0$:

$$\begin{aligned}
\rho(R_j) = R_j \quad &\Leftrightarrow \quad \Big(\forall \sigma \colon \sigma(\overline{m}_j) \in R_j \Leftrightarrow \rho \circ \sigma(\overline{m}_j) \in R_j\Big)\\
&\Leftrightarrow \quad \Big(\forall \sigma \colon \sigma(w_j) \in \mathrm{pred}(v) \Leftrightarrow \rho \circ \sigma(w_j) \in \mathrm{pred}(v)\Big)\\
&\Leftrightarrow \quad \rho(O_j \cap \mathrm{pred}(v)) = O_j \cap \mathrm{pred}(v).
\end{aligned}$$

It thus follows, with Proviso 1, that for all $\rho \in G$:

$$\begin{array}{rcl} \rho(v) = v & \Leftrightarrow & \rho(\mathrm{pred}(v)) = \mathrm{pred}(v) \\ & \Leftrightarrow & \rho(O_j \cap \mathrm{pred}(v)) = O_j \cap \mathrm{pred}(v) \text{ for all } j \in J_0 \\ & \Leftrightarrow & \rho(R_j) = R_j \text{ for all } j \in J_0. \end{array}$$

Hence $\left|R_j^H\right| \leq \left|v^H\right|$ for all $j \in J_0$ and all subgroups $H \subset G$. We apply Lemma 3.2 in the case of an $[\omega, \tau]$-circuit C, and Corollary 3.1 in the case of an $[n, \tau]$-circuit with sufficiently large n. Observe that 'sufficiently large' is in relation to p and the arity of the R_j, which is at most k by the inductive hypothesis. It follows that each R_j is quantifier free equality definable from a tuple of at most k parameters $\overline{p}_j$. Choose $\overline{p}_j$ as in Lemma 3.2/Corollary 3.1 such that $G(\overline{p}_j) \subset G(R_j) \subset G([\overline{p}_j])$.

Consequently, all the R_j for $j \in J_0$ are fixed by $G(\overline{m})$ for any (not, at first, necessarily finite) tuple $\overline{m}$ such that $[\overline{m}] = \bigcup_{j \in J_0}[\overline{p}_j]$. Hence, also v itself is fixed by $G(\overline{m})$. This $\overline{m}$ is also minimal in the sense that any permutation which does not fix $[\overline{m}]$ must move v: Any such permutation must move at least one $[\overline{p}_j]$, therefore R_j, hence also v. Thus, by local polynomiality, $[\overline{m}]$ is after all finite, in fact $\overline{m}$ can have at most k different components. This establishes the first part of Claim 1 for v itself.

Let now $\overline{x}$ be a tuple of variables of the appropriate arity for the base $\overline{m}$ just obtained for v. Let, for $j \in J_0$, $\overline{y}_j$ be a tuple of variables of the arity of R_j (i.e. of the same arity as $\overline{m}_j$), disjoint from $\overline{x}$. By Lemma 3.2/Corollary 3.1, R_j is quantifier-free equality definable from $\overline{p}_j$. Since $\overline{p}_j$ is a subtuple of $\overline{m}$, R_j is in particular quantifier free equality definable as $R_j = \left\{\overline{y}_j \mid \eta_j(\overline{y}_j, \overline{m})\right\}$, for a suitable formula η_j, for each $j \in J_0$. Recall that $v = w_{j_0}$ so that $\overline{m}$ may serve as the $\overline{m}_{j_0}$ in the sense of the statement (ii) of the claim. Note that then, by symmetry, $R_{jj_0} = \left\{\pi(\overline{y}_j, \overline{m}) \mid \pi \in G, \overline{y}_j \in R_j\right\}$ for all $j \in J_0$. For $j \notin J_0$, R_{jj_0} is empty. This proves (ii) for all pairs (j, j') with $j \in J$ and $j' = j_0$: For the non-trivial case, $j \in J_0$, we find

$$R_{jj'} = \left\{(\overline{y}_j, \overline{x}) \mid \eta_j(\overline{y}_j, \overline{x})\right\}.$$

Q.E.D.

Towards the logical definability of the values computed in the nodes, it turns out that a base for v corresponds to the tuple of instantiations for the free variables in the defining formula for that node.

Let C be a locally polynomial $[\omega, \tau]$- or $[n, \tau]$-circuit as above, let the polynomial p which bounds the orbit size be of degree k, and assume that n is sufficiently large in relation to p in the $[n, \tau]$-case. Let G be the appropriate symmetry group for C, S_{fin} or S_n. Let also the system of representatives $\left\{w_j \mid j \in J\right\}$ of all nodes in C with respect to the operation of G be fixed as above. For each w_j fix a base $\overline{m}_j$ according to the last claim.

Claim 2 *For any node v of depth α and with base $\overline{m}$ as in Claim 1, there is a formula $\varphi(\overline{x}) \in L^{2k}[\tau]$ of quantifier rank at most $\max(1, k\alpha)$ such that $\varphi[\overline{m}]$ defines the boolean value computed at v:*
If C is an $[\omega, \tau]$-circuit, then for all inputs $\mathfrak{A} \in \mathfrak{str}[\omega, \tau]$, the boolean value computed by C at v over input $\mathfrak{A}$ is the boolean value of $\mathfrak{A} \models \varphi[\overline{m}]$.

If C is an $[n,\tau]$-circuit, then for all inputs $\mathfrak{A} \in \mathfrak{str}[n,\tau]$, the boolean value computed by C at v over input $\mathfrak{A}$ is the boolean value of $\mathfrak{A} \models \varphi[\overline{m}]$.

It follows from symmetry, that then also, for all $\pi \in G$, the boolean value computed at $\pi(v)$ is defined by $\varphi[\pi(\overline{m})]$.

Before giving the proof, let us see how Claim 2 applies to the implication (ii) $\Rightarrow$ (iii) in Theorem 3.1. Let Q be a query over $\mathfrak{str}_{\mathrm{fin}}[\tau]$, C a locally polynomial $[\omega,\tau_U]$-circuit that computes Q (cf. Definition 2.3 (i)). Apply the above claim to the output node of C, to obtain a sentence $\varphi \in L^{2k}[\tau_U]$ that defines the value computed by C over any input in $\mathfrak{str}[\omega,\tau_U]$. With Lemma 2.2, φ can be transformed into a sentence $\varphi' \in L^{2k}[\tau]$ that is equivalent with φ over all structures in $\mathfrak{str}[\omega,\tau_U]^*$. Thus, Q is definable in L^{2k} as claimed in the main theorem. Since finite depth of C yields finite quantifier rank, an application of Lemma 2.1 proves the corresponding implication in Theorem 3.2.

Proof

The proof is by induction over depth, in parallel with the proof of the last claim. Depth 0 is trivial: Either v is an input node, then for φ we take the atom corresponding to that input node, or v corresponds to a boolean constant in which case we can take any universally false, respectively valid, sentence of quantifier rank 1 for φ.

So assume that $v = w_{j_0}$ is of depth $\alpha > 0$, and that the claim holds for all nodes of lesser depth. Let v for instance be a $\bigwedge$-node. Let $\overline{m} = \overline{m}_{j_0}$ be the chosen base at v. Let J_0 be the set of indices j for which w_j is related to a direct predecessor of v. By the inductive hypothesis, there is a formula $\varphi_j(\overline{y}_j)$ for each $j \in J_0$, satisfying the requirements of the claim for w_j and its fixed base $\overline{m}_j$. From Claim 1 we know that the predicates $R_{jj'}$ are quantifier free equality definable by formulae $\eta_{jj'}(\overline{y}_j,\overline{y}_{j'})$. Recall from the definition of the $R_{jj'}$ that the immediate predecessors of $v = w_{j_0}$ in the orbit of w_j are exactly those nodes $\pi(w_j)$ for which $(\pi(\overline{m}_j),\overline{m}) \in R_{jj_0}$. It follows that the formula

$$\varphi(\overline{x}) := \bigwedge_{j\in J} \forall \overline{y}_j\big(\eta_{jj_0}(\overline{y}_j,\overline{x}) \rightarrow \varphi_j(\overline{y}_j)\big)$$

is again in $L^{2k}_{\infty\omega}$ and defines the value computed at v. Observe that the quantifier rank of this new formula is bounded by the supremum of the quantifier ranks of the φ_j, each increased by k. Inductively, this leads to the bound formulated in the claim. Q.E.D.

We have now established the implications (ii) $\Rightarrow$ (iii) in both, Theorem 3.1 and its restriction to finite depth, Theorem 3.2. It remains to construct the defining formula for a coherent sequence of circuits. This is possible on the basis of the defining formulae for each individual member C_n.

Let $(C_n)_{n\geq 1}$ be a coherent and locally polynomial $[n,\tau]$-sequence of circuits. We begin with some preparatory remarks that exploit the coherence of this sequence.

Recall that C_n is embedded into C_m for $m > n$, m sufficiently large. Owing to Proviso 1 and the completeness condition, this embedding is in fact unique: The range of the embedding is fixed, since by completeness it must consist exactly of those nodes of C_m that are fixed by all permutations of $m \setminus n$. There cannot be any internal automorphisms of C_n that fix the input field of C_n. But the images of the input nodes are uniquely prescribed for any embedding.

We shall therefore identify nodes across the C_n in the following manner: $v \in C_n$ and $v' \in C_{n'}$ are identified if they are mapped to the same node of C_m under the complete embeddings of C_n and $C_{n'}$ into C_m for sufficiently large m. Saying that a node v occurs in C_n means that there is a node in C_n which is identified with v in this precise sense.

For the symmetries we regard S_n as a subgroup of S_m for $n < m$, and all the S_n as subgroups of S_{fin}. We say that v is S_{fin}-related to v', or that v is in the same S_{fin}-orbit as v', if v is S_n-related to v' in C_n for all sufficiently large n. Speaking of S_{fin}-orbits in this sense, we can again introduce a system of representatives of nodes $\{w_j \mid j \in J\}$, such that all nodes belong to one of the orbits O_j of a w_j under S_{fin}.

Let p of degree k be the polynomial that bounds orbit size in the C_n.

Claim 3 *For any node $v \in (C_n)_{n\geq 1}$ there is a tuple $\overline{m} \in \omega$ of at most k components, pairwise distinct, such that:*

(i) For all C_n in which v occurs:

$$S_n(\overline{m}) \subset S_n(v) \subset S_n([\overline{m}]).$$

As above $\overline{m}$ is called a base for v.

(ii) v is in C_n exactly for those n that contain the base $\overline{m}$.

(iii) The boolean value computed at v in C_n is defined by a formula $\varphi(\overline{x}) \in L^{2k}[\tau]$: For all inputs $\mathfrak{A} \in \mathfrak{str}[n,\tau]$ the boolean value computed by C_n at v over input $\mathfrak{A}$ is the boolean value of $\mathfrak{A} \models \varphi[\overline{m}]$.

Proof

The proof is an adaptation of Claims 1 and 2. Some points are immediate:

1) Each node v has a base in each C_n for sufficiently large n. This was proved in Claim 1; sufficiently large means large enough that v occurs in C_n and, more importantly, large enough in relation to the polynomial p so that Claim 1 applies.

2) Let $\overline{m}$ be a base for v in C_m, m sufficiently large. We claim that then $\overline{m}$ is a base for v in each C_n that contains v, and that these are exactly the C_n with $\overline{m} \in n$.

We first show that whenever m is sufficiently large in relation to n, and $\overline{m}$ is a base in C_m, then v occurs in C_n if and only if $\overline{m} \in n$: From the complete embedding of C_n into C_m we have

$$\begin{array}{rcl} v \in C_n & \Leftrightarrow & S_{m\setminus n} \subset S_m(v) \\ & \Leftrightarrow & S_{m\setminus n} \subset S_m([\overline{m}]) \quad \Leftrightarrow \quad [\overline{m}] \subset n. \end{array}$$

For the last equivalence, assume that m is at least $n + |[\overline{m}]| + 1$.

We turn to the uniformity of bases across all n. It is obvious that $S_m(\overline{m}) \subset S_m(v) \subset S_m([\overline{m}])$ in C_m, m sufficiently large, implies that $S_n(\overline{m}) \subset S_n(v) \subset S_n([\overline{m}])$ in C_n for all $n < m$ that contain $\overline{m}$. The crucial point, therefore, is upward agreement of bases as is expressed in the following observation: If $S_n(\overline{m}) \subset S_n(v) \subset S_n([\overline{m}])$ in C_n, $S_m(\overline{m}') \subset S_m(v) \subset S_m([\overline{m}'])$ in C_m, $n > |[\overline{m}]| + 1$, m sufficiently large with respect to n, then $[\overline{m}] = [\overline{m}']$. This follows with a complete embedding of C_n into C_m. First observe that $[\overline{m}']$ must be contained in

n, because v in C_m cannot be affected by $S_{m\setminus n}$, by completeness. That two bases within n must agree setwise is obvious.

We can thus choose bases that satisfy (i) and (ii) of the claim for all the representatives w_j, $j \in J$, of orbits with respect to S_{fin}. Let $\overline{m}_j$ be such base for w_j. Note that bases are compatible with the operation of S_{fin}: If $v = \pi(w_j)$, then $\pi(\overline{m}_j)$ is a base for v.

Recall that O_j is the orbit of w_j under S_{fin}. It follows from the above that $O_j \cap C_n$, the set of nodes in C_n that are S_{fin}-related to w_j, is the same as the set $\{\pi(w_j) \mid \pi \in S_{\text{fin}}, \pi(\overline{m}_j) \in n\}$; note however, that w_j need not occur in C_n or that $\overline{m}_j$ need not be in n.

Consider now the predicates that describe the edge relation. Put, just as in Claim 1 (ii),

$$R_{jj'} := \{(\pi(\overline{m}_j), \pi'(\overline{m}_{j'})) \mid \pi, \pi' \in S_{\text{fin}}, \pi(w_j) \in \text{pred}(\pi'(w_{j'})) \text{ in } C_n,\ n \text{ sufficiently large }\}.$$

We claim that these predicates uniformly describe the edge relations in all the C_n. More precisely: Their restrictions to n are the correct edge predicates for C_n. Let $\pi(\overline{m}_j), \pi'(\overline{m}_{j'}) \in n$. Then $\pi(w_j), \pi'(w_{j'}) \in C_n$ and, through embeddability, there is an edge connecting $\pi(w_j)$ to $\pi'(w_{j'})$ in C_n if and only if this is the case in all C_m in which these nodes both occur.

It follows from the arguments in Claims 1 and 2, that the corresponding predicates $R_{jj'}$ are uniformly definable by the formulae $\eta_{jj'}$ in all C_n. We thus find inductively, that also the formulae φ_j defining the values computed at w_j and constructed as in the proof of Claim 2 are uniformly applicable in all C_n. This finishes the proof of Claim 3. Q.E.D.

Strict matches for each $L^k_{\infty\omega}$

A last remark in this section concerns the apparent mismatch between the degree k of the polynomial that is responsible for local polynomiality, and the number of variables in formulae of $L^{\omega}_{\infty\omega}$. In Lemma 3.1, local polynomiality of degree k was established for circuits evaluating formulae in $L^k_{\infty\omega}$, but only definability in $L^{2k}_{\infty\omega}$ could be shown in Claims 1 and 3 for symmetric circuits whose orbit size is bounded by a polynomial of degree k.

Observe, however, that up to $2k$ variables were essentially needed in the description of the links between nodes in the proofs of Claims 1, 2 and 3: The formulae $\eta_{jj'}$ defining the edge relations $R_{jj'}$ used up to $2k$ variables. This is not surprising, however: Single nodes require a tuple of at most k parameters for their identification within their orbit in a circuit whose orbit sizes are bounded by a degree k polynomial. Edges correspond to pairs of nodes and thus may require up to $2k$ parameters. This also indicates a way to obtain an exact match, for each k separately, in Theorem 3.1: We have to redefine local polynomiality – or rather its degree – in terms of bounds on the orbit size of edges or wires, rather than in terms of nodes! To obtain this refinement we now define the following, cf. Definition 2.6:

Definition 3.1 *An explicitly symmetric $[\omega,\tau]$- or $[n,\tau]$-circuit C is* locally polynomial of degree k *if there is a polynomial p of degree k such that for all finite $s \subset \omega$ and for all edges e of C: $|e^{S_s}| \leq p(|s|)$. An analogous definition applies to $[n,\tau]$-sequences of symmetric circuits.*

Note that a circuit is locally polynomial in the sense of a polynomial bound on the orbits of nodes, if it is locally polynomial in the sense of a polynomial bound on the orbits of edges.

Thus, this new definition of the degree of local polynomiality really just is a refinement. The point is that in this way we obtain matches at each level k:

Theorem 3.3 *The following are equivalent for any query $Q \subset \mathfrak{str}_{\mathrm{fin}}[\tau]$ and for each k:*

(i) Q can be computed in a coherent $[n,\tau]$-sequence of explicitly symmetric circuits, that is locally polynomial of degree k.

(ii) Q can be computed in a single symmetric $[\omega,\tau_U]$-circuit that is locally polynomial of degree k.

(iii) Q is definable in $L^k_{\infty\omega}$.

Sketch of proof
We refer to the corresponding parts of the proofs of Lemma 3.1 and Claims 1, 2 and 3.

First consider a boolean query that is definable in $L^k_{\infty\omega}$. We claim that the circuits constructed in Lemma 3.1 are locally polynomial of degree k. The construction of the circuits was by induction over the formula rank. The operation of the symmetry group on any node is the operation on the parameters that instantiate the subformula belonging to that node. Thus, the former proof led to tuples of at most k parameters in each node. Now we have to consider edges. These occur between nodes that link an instantiation of a subformula $\varphi[\overline{m}]$ to nodes belonging to certain instantiations of its direct constituents. If φ is a boolean combination of subformulae, then the parameter set $[\overline{m}]$ is the union of the parameter sets at the immediate predecessor nodes. Therefore, in this case, all edges into the node of $\varphi[\overline{m}]$ are fixed relative to $\overline{m}$. Consider now the existential step. Let $\varphi(\overline{x}) = \exists z\psi(z,\overline{x})$. Edges now link nodes belonging to instantiations $\psi[m,\overline{m}]$ to the node associated with $\varphi[\overline{m}]$: Again, since ψ has only k variables, the tuple $(m,\overline{m})$ has at most k components, and the edge under consideration is fixed relative to these.

Now for the opposite direction, i.e. the analysis of the corresponding steps in the proofs of Claims 1, 2 and 3. Assume now local polynomiality of degree k. The defining formulae are obtained in the proof of Claim 2 by induction on the depth of nodes. In the construction of the formula $\varphi(\overline{x})$ defining the the value at w_{j_0}, those formulae $\varphi_j(\overline{y}_j)$ are used, that define the values at direct predecessor nodes of v. The $\eta_{jj_0}(\overline{y}_j,\overline{y}_{j_0})$ define the R_{jj_0} that exactly describe the occurrences of edges between nodes. If C is locally polynomial of degree k, then each η_{jj_0} must enforce sufficiently many equalities between variables in $(\overline{y}_j,\overline{y}_{j_0})$ so that these consist in fact of at most k distinct components. But these equalities can be exploited in the construction of the new formula φ, that in the case of a conjunctive node w_{j_0} was of the form

$$\varphi(\overline{y}_{j_0}) := \bigwedge_{j\in J} \forall\overline{y}_j\left(\eta_{jj_0}(\overline{y}_j,\overline{y}_{j_0}) \rightarrow \varphi_j(\overline{y}_j)\right).$$

Contraction of different variable symbols that are equated by η_{jj_0} into the same symbol therefore yields $\varphi \in L^k_{\infty\omega}$ if the φ_j are in $L^k_{\infty\omega}$ by inductive hypothesis. Q.E.D.

4 Boolean networks

4.1 Basic definitions

Boolean circuits generalize to a certain kind of computational networks if the condition of acyclicity is dropped. A *boolean network* is a connected directed graphs with a labelling completely analogous to that required for circuits. The *computation* of a network N on some input is described in terms of a sequence of assignments of truth values to the nodes of N, $T_t: N \rightarrow \{0,1\}$, for $t \geq 0$. Intuitively these assignments corresponds to a description of the time-dependent evolution of the network under stepwise flow of information. The input corresponds to an assignment to the input nodes. The initial assignment T_0 on all of N is obtained through setting the values of all non-input nodes to 0. Inductively a sequence $(T_t)_{t\geq 0}$ is generated as follows. On all nodes of in-degree 0, the initial assignment is kept throughout. For any node v of in-degree at least 1, the assignment $T_{t+1}(v)$ is determined in the obvious way on the basis of the assignments $T_t(u)$ of all direct predecessors u of v. The computation *terminates* if this sequence of assignments becomes ultimately constant, i.e. if for some t we have $T_{t+1} = T_t$. The value computed by N on an input, over which the computation terminates, is the boolean value given to the output node in the terminating truth assignment.

Definition 4.1 *A network N is formatted for $\mathfrak{str}[\omega, \tau]$ if the input nodes are labelled injectively by $\omega^{\overline{r}}$, $\overline{r}$ the tuple of arities in τ. It is formatted for $\mathfrak{str}[n, \tau]$ if its input nodes are labelled by $n^{\overline{r}}$. We talk of $[\omega, \tau]$- and $[n, \tau]$-networks. A sequence of networks $(N_n)_{n\geq 1}$ is an $[n, \tau]$-sequence if N_n is an $[n, \tau]$-network for each n.*

Computation of a query Q over $\mathfrak{str}_{\mathrm{fin}}[\tau]$, either by a single $[\omega, \tau_U]$-network or by a $[n, \tau]$-sequence of networks, is defined in complete analogy with Definition 2.3 above. For instance, an $[\omega, \tau_U]$-network computes Q if its output node computes the characteristic function of Q for all encodings of finite τ-structures via representations in $\mathfrak{str}[\omega, \tau_U]^*$.

The notion of automorphisms of networks is the same as for circuits. Also, the criterion of explicit symmetry carries over unchanged, cf. Definition 2.5:

Definition 4.2 *Let N be an $[n, \tau]$- or $[\omega, \tau]$-network. N is* explicitly symmetric, *or* symmetric, *if the action of S_n or S_{fin} on the respective input field extends to an automorphic action on the entire circuit C.*

In analogy with Proviso 1 we may restrict attention to networks whose automorphisms are not due to trivial redundancies:

Proviso 2 *For symmetric boolean networks N we assume the following:*
For any automorphism π of N and any node $v \in N$: If π fixes the direct predecessors of v setwise, then π fixes v itself.

Definition 4.3 *Let N be a symmetric $[\omega, \tau_U]$-network.*

(i) *N is* locally polynomial *if, as for circuits, the size of orbits is locally polynomial.*

(ii) N is finitary *if the number of orbits of nodes under the full automorphism group $aut(N)$ is finite, i.e. if there are only finitely many isomorphism types of nodes in N.*

An embedding between symmetric networks is defined just as for circuits as a label preserving isomorphic embedding of the underlying graphs. Such an embedding is complete if its image consists of those nodes that are fixed by all permutations that fix all images of input nodes.

Definition 4.4 *Let $(N_n)_{n\geq 1}$ be a $[n, \tau]$-sequence of symmetric networks.*

(i) $(N_n)_{n\geq 1}$ is locally polynomial *if there is a uniform polynomial bound on the orbit size across all n.*

(ii) $(N_n)_{n\geq 1}$ is coherent, *if for all n, there is a complete embedding of N_n into N_m for all sufficiently large $m > n$.*

(iii) A coherent sequence $(N_n)_{n\geq 1}$ is finitary, *if there is a uniform finite bound on the number of orbits of nodes under the full automorphism group $aut(N_n)$ in N_n.*

Just as for circuits, it will turn out that coherent locally polynomial $[n, \tau]$-sequences of networks are equivalent with locally polynomial $[\omega, \tau_U]$-networks as far as the evaluation of boolean queries over $\mathfrak{str}_{\mathrm{fin}}[\tau]$ is concerned.

Definition 4.5 *A network is* positive *if all nodes labelled $\neg$ (for negation) are immediate successors to input nodes.*

Positive networks obviously have the special property that the truth assignments T_t are monotone in the sense that $T_t(v) = 1$ implies that $T_{t'}(v) = 1$ for all later stages $t' > t$.

In connection with networks we shall mainly consider the most expressive of the fixed-point extensions of first-order logic: *Partial fixed-point logic*, PFP.

Let $\varphi = \varphi(X, \overline{x})$ be a formula with a free second-order variable X of some arity r matching the arity of $\overline{x} = (x_1, \ldots, x_r)$; φ may have other free variables, which then are regarded as parameters in what follows. With φ associate a global operator F_φ on r-ary predicates: Over a structure $\mathfrak{A}$, which interprets φ up to the designated free variables, $F_\varphi^{\mathfrak{A}}$ is defined through $F_\varphi^{\mathfrak{A}} : R \mapsto \{\overline{a} \mid \mathfrak{A} \models \varphi[R, \overline{a}]\}$, for all r-ary predicates $R \subset A^r$. Iteration of this operator on the empty predicate generates a sequence of predicates $(F_\varphi^{\mathfrak{A}})^i(\emptyset)$, where $(F_\varphi^{\mathfrak{A}})^i$ stands for the i-fold iteration of the operation $F_\varphi^{\mathfrak{A}}$, with $(F_\varphi^{\mathfrak{A}})^0 = \mathrm{id}_{A^r}$. The first elements in the sequence $(F_\varphi^{\mathfrak{A}})^i(\emptyset)$ are

$$\emptyset, F_\varphi^{\mathfrak{A}}(\emptyset), F_\varphi^{\mathfrak{A}}(F_\varphi^{\mathfrak{A}}(\emptyset)), \ldots$$

The *partial fixed point* of φ is defined to be either the empty set, if this sequence is ultimately non-trivially periodic, or the predicate determined as the fixed point reached in this sequence, if it exists. The predicates $(F_\varphi^{\mathfrak{A}})^i(\emptyset)$ are called the *stages* of the fixed-point generation.

In the logic PFP we have, in addition to the usual first-order rules for the formation of formulae, the following rule: With $\varphi(X, \overline{x})$ as above in PFP, the formula $\psi(\overline{z}) := [\mathrm{PFP}_{X,\overline{x}}\varphi]\,\overline{z}$ is also a formula of PFP. Its semantic is such that $\mathfrak{A} \models \psi[\overline{a}]$ if and only if $\overline{a}$ is in the partial fixed point determined by F_φ on $\mathfrak{A}$. It is well known that on linearly ordered structures, PFP captures PSpace, [V 82, AV 89]. On the class of all finite structures PFP is also equivalent with the language WHILE, see [AV 89, G 92].

Standard fixed-point logic, FP, can be obtained as a restriction of PFP as follows. Instead of allowing the above fixed-point generation for all formulae $\varphi(X, \overline{x})$ only admit formulae φ that are *positive* in all second-order variables (equivalently, negations may only occur in front of atoms not involving second-order variables). More intuitive descriptions of this logic are in terms of *least-fixed-point* or *inductive fixed-point* operators, cf. [G/S 86]. The great importance of FP in finite model theory is due to the result of Immerman and Vardi that on ordered structures FP coincides with PTime, [V 82, I 86].

It is not difficult to prove that both, FP and PFP, satisfy the property expressed in Lemma 2.2 for $L^{\omega}_{\infty\omega}$. Definability of a query $Q \subset \mathfrak{str}_{\mathrm{fin}}[\tau]$ in PFP or FP over $\mathfrak{str}_{\mathrm{fin}}[\tau]$ is equivalent with definability of $Q^U \subset \mathfrak{str}[\omega, \tau_U]^*$ in PFP or FP over $\mathfrak{str}[\omega, \tau_U]^*$:

Lemma 4.1 *Let $Q \subset \mathfrak{str}_{\mathrm{fin}}[\tau]$ and let $Q^U \subset \mathfrak{str}[\omega, \tau_U]^*$ be the class consisting of all representations of structures from Q over $\mathfrak{str}[\omega, \tau_U]^*$. Then the following are equivalent:*

(i) There is a sentence of PFP$[\tau]$ *(resp.* FP$[\tau]$*) that defines Q over $\mathfrak{str}_{\mathrm{fin}}[\tau]$.*

(ii) There is a sentence of PFP$[\tau_U]$ *(resp.* FP$[\tau_U]$*) that defines Q^U over $\mathfrak{str}[\omega, \tau_U]^*$.*

The following remarks about FP and PFP concern simultaneous fixed-point generations. Instead of a single $\varphi(X, \overline{x})$ one may consider systems of formulae $\varphi_j(X_1, \ldots, X_l, \overline{x}^{(j)})$, $1 \leq j \leq l$, where the arity of X_j matches that of the tuple $\overline{x}^{(j)}$ for each j. Regarded as operators for the simultaneous transformation of an l-tuple of predicates, this system induces stages according to:

$$\begin{array}{rcl} X_j^0 & = & \emptyset \\ X_j^{t+1} & = & \left\{\overline{x}^{(j)} \mid \varphi_j[X_1^t, \ldots, X_l^t, \overline{x}^{(j)}]\right\}, \ \ 1 \leq j \leq l. \end{array}$$

Again, this may or may not lead to a stationary assignment to the X_j, and the partial fixed point of the system is defined as in the standard case, with $\emptyset$ as the default value for all the fixed-point predicates if no stationary assignment is reached. We state without proof some technical facts that will be useful in our applications. The proofs are straightforward coding arguments.

Remark 4.1 *(i) Each component of the simultaneous partial fixed point of a system is definable in ordinary* PFP*.*

(ii) (Simultaneous) partial fixed points that are generated from some initial assignment other than $\emptyset$, that is PFP*-definable itself, are also definable in standard* PFP*.*

(iii) It is possible and often natural to admit 0-ary fixed-point variables in systems for the generation of simultaneous partial fixed-points, corresponding to boolean fixed-point variables. Modelling these through unary indicator predicates that are either full or empty, one sees that standard PFP *is rich enough to comprise this variation, too.*

A system for the simultaneous generation of partial fixed points is regarded as positive if the constituent formulae are positive in all fixed-point variables. Under the assumption of positivity all the closure properties stated for PFP above also apply to FP.

4.2 Boolean networks and fixed-point logics

Theorem 4.1 *For a boolean query Q on $\mathfrak{str}_{\text{fin}}[\tau]$, the following are equivalent:*

(i) Q can be computed by a finitary coherent locally polynomial $[n,\tau]$-sequence of symmetric networks.

(ii) Q is computable by a finitary locally polynomial symmetric $[\omega,\tau_U]$-network.

(iii) Q is definable in PFP.

The following will be an immediate corollary of the proof of Theorem 4.1:

Theorem 4.2 *For a boolean query Q on $\mathfrak{str}_{\text{fin}}[\tau]$, the following are equivalent:*

(i) Q can be computed by a finitary coherent locally polynomial $[n,\tau]$-sequence of positive symmetric networks.

(ii) Q is computable by a finitary locally polynomial symmetric positive $[\omega,\tau_U]$-network.

(iii) Q is definable in FP.

The proofs can largely be reduced to the treatment of circuits. We give sketches in the two following sections that deal with the passages form formulae to networks and from networks to defining formulae separately.

Finitary networks for PFP

We apply a normal form theorem for PFP: Any PFP-sentence is equivalent with one of the form $\chi\big([\mathrm{PFP}_{X,\overline{x}}\psi]\big)$, for first-order $\chi(X)$ and $\psi(X,\overline{x})$, where $\psi(X,\overline{x})$ is such that the fixed point induced by F_ψ always exists. See [G 92] for a presentation of this normal form theorem. The reduction to only one application of the partial fixed-point operator is standard, the guaranteed termination of the iteration is achieved through an intrinsic check for termination.

For network evaluation with its time dependent flow of information, the distances between nodes along different paths obviously matter. It is easy to find examples of circuits that compute different boolean functions, depending on whether they are evaluated as circuits or as networks. When we use circuits as building blocks for networks, some care has to be taken. We call a boolean circuit *synchronized* if it has the following property: Any two paths from a node v_1 to a node v_2 must have the same length. This immediately implies that the distance of a node from the sources (depth 0, or in-degree 0) is path independent and equals its depth.

It is easy to see, that synchronized circuits can be evaluated in the time dependent network manner without affecting the result.

The easiest way to obtain synchronized circuits for first-order formulae is to manipulate formulae prior to the application of Lemma 3.1. Any first-order formula is equivalent with one in which any two subformulae that are connected by $\wedge$ or $\vee$ have the same formula rank. This can be achieved, for instance, through semantically vacuous repetition of identical conjuncts or disjuncts. Since the depth of nodes in the constructions of Lemma 3.1 corresponds to formula rank of subformulae, these rank-balanced formulae lead to synchronized circuits automatically.

From Lemma 3.1, we thus obtain the following for any first-order formula $\varphi(X, \overline{x})$, which is regarded as a formula over the vocabulary $\tau \dot{\cup} \{X\}$:

(i) A locally polynomial $[\omega, \tau \dot{\cup} \{X\}]$-circuit C, with the following particular properties: C is synchronized and of finite depth d; the nodes at depth d correspond in a one-to-one fashion to the instantiations for $\overline{x}$ in $\varphi(X, \overline{x})$ over ω; two such nodes are related by an automorphism if they correspond to S_{fin}-related instantiations; the node corresponding to the instantiation $\overline{m}$ for $\overline{x}$ computes the boolean value of $\varphi[X, \overline{m}]$ over any input in $\mathfrak{str}[\omega, \tau \dot{\cup} \{X\}]$.

(ii) A locally polynomial coherent $[\omega, \tau \dot{\cup} \{X\}]$-sequence of circuits of constant finite depth d, $(C_n)_{n \geq 1}$, such that for all n: C_n is synchronized; the nodes at depth d of C_n correspond to the instantiations for $\overline{x}$ in $\varphi(X, \overline{x})$ over n; two of these nodes are related by an automorphism if they correspond to S_n-related instantiations; the node corresponding to the instantiation $\overline{m}$ computes the boolean value of $\varphi[X, \overline{m}]$ over any input in $\mathfrak{str}[n, \tau \dot{\cup} \{X\}]$.

In both cases we further obtain natural networks that 'compute the partial fixed-point' $\mathrm{PFP}_{X,\overline{x}}\varphi(X, \overline{x})$ as follows. Connect the depth d node corresponding to the instantiation $\overline{m}$ to the input node corresponding to the same instantiation of the atom $X\overline{x}$. The resulting network is formatted for $\mathfrak{str}[\omega, \tau]$ or $\mathfrak{str}[n, \tau]$, since the X-atoms no longer are input nodes. Symmetry is preserved, since the new links connect nodes which are fixed relative to each other by all automorphisms of the original circuit. For the same reason, local polynomiality is not affected. Let N be the resulting network. Let $u_{\overline{m}}$ denote the former input node for the instantiation $\overline{m}$ of $X\overline{x}$.

We claim that in a computation over input $\mathfrak{A} \in \mathfrak{str}[\omega, \tau]$ or $\mathfrak{A} \in \mathfrak{str}[n, \tau]$, with truth assignments T_t, these nodes $u_{\overline{m}}$ compute the stages of the partial fixed-point evaluation in the following sense: At time steps t with $(d+1)i \leq t < (d+1)(i+1)$, $T_t(u_{\overline{m}}) = 1$ if and and only if $\overline{m} \in (F_\varphi^{\mathfrak{A}})^i(\emptyset)$. This is clear for $i = 0$: By our conventions on network computation, the $u_{\overline{m}}$ are all assigned 0 during $0 \leq t \leq d$. Inductively, assume that the nodes $u_{\overline{m}}$ carry the correct truth values for $\left(F_\varphi^{\mathfrak{A}}\right)^i(\emptyset)$ at time steps $(d+1)i \leq t < (d+1)(i+1)$. With respect to the former circuit, this corresponds to the input $\left(\mathfrak{A}, (F_\varphi^{\mathfrak{A}})^i(\emptyset)\right)$. Therefore, the next $d+1$ steps in the network computation simulate the stepwise evaluation of the boolean circuit C on this input; synchronization is essential to allow step-wise evaluation instead of the standard evaluation for circuits. It follows that the claim holds for the consecutive time interval.

If the partial fixed-point evaluation reaches a stationary stage, then the entire network computation becomes stationary and terminates. The resulting values in the $u_{\overline{m}}$ represent the predicate $\mathrm{PFP}_{X,\overline{x}}\varphi(X, \overline{x})$.

Networks for $\chi\left(\left[\mathrm{PFP}_{X,\overline{x}}\psi\right]\right)$ are constructed by extending N with an appropriate circuit C for $\chi(X)$, as obtained from another application of Lemma 3.1 to the first-order formula

$\chi(X)$. N for $\mathrm{PFP}_{X,\overline{x}}\psi$ and C for $\chi(X)$ are joined to produce a network for $\chi\big([\mathrm{PFP}_{X,\overline{x}}\psi]\big)$ as follows. The nodes $u_{\overline{m}}$ in N take the place of the input nodes for the X-atoms in C. All other input nodes of C are merged with the corresponding input nodes of N. It is easy to see that the resulting networks computes the right thing, given that the PFP-evaluation always reaches a fixed point. Also, this join of N and C preserves symmetry and local polynomiality for reasons similar to those given above.

In order to obtain the desired $[\omega, \tau_U]$-network for the query Q defined by $\chi\big([\mathrm{PFP}_{X,\overline{x}}\psi]\big)$, we apply the entire argument to the relativization $\big[\chi\big([\mathrm{PFP}_{X,\overline{x}}\psi]\big)\big]^U$, i.e. to the PFP-sentence sentence that defines Q^U in the sense of Lemma 4.1.

To obtain the coherent sequence, we apply the above procedure to each member of a coherent sequence for φ as given in (ii) and join the resulting network in the manner described with the corresponding member of a coherent sequence for χ. It remains to check that coherence is also preserved in these manipulations. We omit the details.

For the variation concerning FP rather than PFP, we point out that an analogous normal form for FP allows to focus on FP-sentences of the form $\chi\big([\mathrm{FP}_{X,\overline{x}}\psi]\big)$, where $\chi(X)$ and $\psi(X, \overline{x})$ are first-order and ψ positive in X. W.l.o.g. one may assume that negation signs occur in ψ only in front of atoms involving equality or predicates other than X. The circuits that are obtained for such first-order formulae in Lemma 3.1 have negation gates only as immediate successors to input nodes and thus immediately lead to positive networks if the above constructions are applied.

PFP-definability for finitary locally polynomial networks

It remains to prove that any query computable in a finitary locally polynomial network, or in a finitary coherent locally polynomial sequence of networks, can be defined in PFP. The argument will be given in some detail for the single networks. It can be adapted to coherent sequences in close analogy with what has been done for circuits in the last section – the details, however, are omitted.

We associate with a network N a canonical unfolding as a circuit, which represents the entire time dependent computation of the network statically.

Definition 4.6 *Let N be a boolean network. The* canonical unfolding *of N is the boolean circuit defined as follows: The set of nodes of C is $\omega \times N$. For nodes v of depth 0 in N we introduce edges from (t, v) to $(t + 1, v)$ for all t. For an internal node $v \in N$, the node (t, v) is joined to the node $(t+1, v')$ if v is joined to v' in N. Labels carry over in the obvious way, with the following stipulation for nodes at depth 0:*
The labelling of a node v at depth 0 in N is transferred to the node $(0, v)$. All other nodes $(0, v)$, v of depth greater than 0 in N, receive the label F. Nodes (t, v) for v of depth 0 in N and $t > 0$ have in-degree 1 in C and receive no label.

It is not difficult to see that for all $t \in \omega$, the node (t, v) of C computes the truth value $T_t(v)$ of the network evaluation at v at time step t.

For each node v of N we call the set of nodes (t, v), $t \in \omega$, the *fibre of v* in C. As regards automorphisms, any $\pi \in \mathrm{aut}(N)$ induces an automorphism $\pi' \in \mathrm{aut}(C)$, which is uniquely determined with the property that it preserves fibres and acts as π on these. In fact $\mathrm{aut}(N)$ is naturally isomorphic with the subgroup $\mathrm{aut}_0(C)$ of $\mathrm{aut}(C)$ consisting of *fibre-preserving automorphisms*. The strategy to adapt our previous results to networks consists in replacing ordinary circuits with these fibred circuits. Rather than give a formal definition, we shall talk of fibred circuits just to stress that we replace the full automorphism group $\mathrm{aut}(C)$ of the circuit by its subgroup $\mathrm{aut}_0(C)$. Few facts about the origin of $\mathrm{aut}_0(C)$ will actually matter. These are:

(i) For symmetric $[n, \tau]$- or $[\omega, \tau_U]$-networks, there is a unique extension of the action of S_n or S_{fin} to $\mathrm{aut}_0(C)$.

(ii) Instead of Proviso 1 the circuits considered now satisfy the corresponding property only for automorphisms in $\mathrm{aut}_0(C)$: If $\pi \in \mathrm{aut}_0(C)$ fixes the set of direct predecessors of a node v then it fixes v itself.

It is easy to see that not only symmetry but also local polynomiality carries over from the network to its unfolding. The same is true of the coherence of sequences: Embeddability is obviously preserved, completeness of the embedding as well. We employ the results about circuits with the slight modification in the proof to take care of (ii) above, in order to get the following as a corollary to Claim 1 of the last section.

Fix a locally polynomial $\mathfrak{str}[\omega, \tau]$- or $\mathfrak{str}[n, \tau]$-network N. Let C be its canonical unfolding. Let the polynomial which bounds the orbit size be of degree k. Let G be the appropriate symmetry group for N and C, S_{fin} or S_n. Note that G acts in $\mathrm{aut}_0(C)$ according to (i) above.

Corollary 4.1 *For every node $v \in N$ there are a tuple $\overline{m} \in \omega$ (for $[\omega, \tau]$-format) or $\overline{m} \in n$ (for $[n, \tau]$-format) of at most k components, pairwise distinct, such that for each node (t, v) in the fibre of v in C:*

$$G(\overline{m}) \subset G((t, v)) \subset G([\overline{m}])$$

with respect to the action of G on C. It follows that $G(\overline{m}) \subset G(v) \subset G([\overline{m}])$ with respect to the action of G on N.

Let $\{w_j \mid j \in J\}$ be a set of representatives for all orbits of nodes under G, O_j the orbit of w_j. Fix a base $\overline{m}_j$ for each w_j according to the above corollary. Let s_j be the arity of $\overline{m}_j$. Note that $J \times \omega$ is a system of representatives for the orbits in C under $\mathrm{aut}_0(C)$.

Let us introduce the notation $[\![\omega]\!]^s$ for the set of all s-tuples with pairwise distinct components over ω, similarly for $[\![n]\!]^s$ over n. If N is an $[\omega, \tau]$-network, then the set $\bigcup_{j \in J} \{j\} \times [\![\omega]\!]^{s_j}$ parameterizes the nodes of N: $(j, \overline{m})$ is taken as an address of the node $\pi(w_j)$, where π is chosen such that $\pi(\overline{m}_j) = \overline{m}$. Similarly for an $[n, \tau]$-network, the parameter set is $\bigcup_{j \in J} \{j\} \times [\![n]\!]^{s_j}$. Note, however, that this parameterization is not in general injective, owing to the possibility that $G(v) \neq G(\overline{m})$ for bases $\overline{m}$ of v. The parameterization extends to the associated unfolding C of N, simply through adding a component ω for the depth in C.

Just as in the proof of Claim 1, we further find that the connections between nodes are quantifier-free equality definable in terms of this parameterization. Consider a node $(t+1, v) \in C$. Its direct predecessors are of the form (t, w), with w a direct predecessor of v

in N. In C we find that the following predicates are quantifier-free equality definable:

$$\begin{aligned} R_{(t,j)(t+1,j')} &= \{(\pi(\overline{m}_j)\pi'(\overline{m}_{j'})) \mid (t,\pi(w_j)) \in \mathrm{pred}(t+1,\pi'(w_{j'})) \text{ in } C\} \\ &= \{(\pi(\overline{m}_j)\pi'(\overline{m}_{j'})) \mid \pi(w_j) \in \mathrm{pred}(\pi'(w_{j'})) \text{ in } N\} \qquad =: \quad R_{jj'}. \end{aligned}$$

The independence of t that is indicated in the second equality is due to the definition of the canonical unfolding. Just as for circuits, these $R_{jj'}$ encode the edge relation of the network. Applying Claim 1 (ii) to C therefore yields quantifier-free equality formulae $\eta_{jj}(\overline{y}_j, \overline{y}_{j'})$ that define the sets $R_{jj'}$ appropriate for N.

We turn to the logical description of the computation of finitary N by PFP-formulae. From now on we deal entirely with the networks themselves once more.

As we assume N is finitary, the set J is finite. Introduce predicate variables X_j of arity s_j for each $j \in J$. Let $\overline{x}_j$ be a tuple of distinct element variables of arity s_j, $j \in J$. The intention is to use the $\overline{x}_j$ as addresses for the nodes in O_j and the X_j, or rather their stages in a PFP-evaluation, to describe the collections of those nodes in O_j that are assigned the truth value 1 in the corresponding stage of the network evaluation. For j with empty base, we appeal to the remarks about boolean fixed-point variables in Remark 4.1: Think of the corresponding X_j as boolean values that are modelled by unary indicator predicates.
For explicit notation let $J = \{1, \ldots, l\}$.

Claim 4 *Let N be of $[\omega, \tau]$-format. There are first-order formulae $\varphi_j(X_1, \ldots, X_l, \overline{x}_j)$ and $\chi_j(\overline{x}_j)$, for $1 \le j \le l$, such that the following are true in the computation of N on any input $\mathfrak{A} \in \mathfrak{str}[\omega, \tau]$:*

(i) For all $\overline{n} \in [\![\omega]\!]^{s_j}$: $\mathfrak{A} \models \chi_j[\overline{n}]$ if and only if the input node determined by $(j, \overline{m})$ is set to 1 in N on input $\mathfrak{A}$.

(ii) The sequence of predicates $\left(P_1^t, \ldots, P_l^t\right)_{t\in\omega}$ which is inductively generated through

$$\begin{aligned} P_j^0 &:= \{\overline{m} \mid \mathfrak{A} \models \chi_j[\overline{m}]\} \\ P_j^{t+1} &:= \{\overline{m} \mid \mathfrak{A} \models \varphi_j[P_1^t, \ldots, P_l^t, \overline{m}]\}, \ 1 \le j \le l \end{aligned}$$

indicates the truth assignments in the computation of N on $\mathfrak{A}$:
$\overline{m}$ is in P_j^t if and only if the node determined by $(j, \overline{m})$ evaluates to 1 under T_t in the computation on $\mathfrak{A}$.

The same applies to computations of $[n, \tau]$-networks over $\mathfrak{A} \in \mathfrak{str}[n, \tau]$.

Observe that this suffices to prove that any query $Q \subset \mathfrak{str}[\omega, \tau_U]^*$ that can be evaluated by a finitary locally polynomial $[\omega, \tau_U]$-network is PFP-definable over $\mathfrak{str}_{\mathrm{fin}}[\tau]$: Apply the claim to the given $[\omega, \tau_U]$-network. The system for a simultaneous partial fixed point from (ii) can be transformed into a single PFP$[\tau_U]$-formula for the definition of each component X_j of the simultaneous partial fixed point. Termination of the computation is equivalent with the existence of a fixed-point for the system, and the eventual value computed in the output node v of N is encoded in that component X_j of the resulting fixed point, that belongs to the orbit of v – actually this component is the encoding of a boolean fixed-point variable, since v has empty base, cf. Remark 4.1.

We thus find that $Q^U \subset \mathfrak{str}[\omega, \tau_U]^*$ is PFP-definable, but this is enough to obtain PFP-definability of Q itself by Lemma 4.1.

Sketch of proof
The existence of the χ_j is obvious, in fact these are chosen to be atomic formulae corresponding to the rôle of the different input nodes, or universally valid or false sentences in the case of source nodes labelled T or F.

Consider the requirements on the φ_j. Let $(j_0, \overline{m})$ be an address for v. The direct predecessors of v in O_j are those with addresses $\{\overline{y}_j \mid \eta_{jj_0}(\overline{y}_j, \overline{m})\}$. Suppose that v is a $\bigwedge$-node. $\varphi_{j_0}(X_1, \ldots, X_l, \overline{x})$ can then be chosen as

$$\varphi_{j_0}(X_1, \ldots, X_l, \overline{x}) := \bigwedge_{j=1\ldots l} \forall \overline{y}_j \big(\eta_{jj_0}(\overline{y}_j, \overline{x}) \rightarrow X_j \overline{y}_j\big).$$

That these formulae satisfy the requirements is proved in an easy induction. Q.E.D.

Note also that positivity of the network leads to formulae φ_j that are positive in all X_j with the possible exception of indices j of orbits of input nodes. But the predicates X_j for these orbits can be dispensed with anyway: Input nodes keep their initial truth assignment throughout. These initial assignments are defined by the corresponding χ_j, and these can completely replace X_j in the fixed-point system. This shows, that for positive N we can get a positive system and finally a defining formula in FP.

Acknowledgments

I am grateful to Erich Grädel for suggesting the extension to networks. Thanks are due to Martin Grohe for his very careful reading of a preliminary version of this paper, and for suggesting a substantial simplification in the proof of Lemma 3.2.

References

[AV 89] S. Abiteboul, V. Vianu, *Fixpoint Extensions of First-Order Logic and Datalog-Like Languages*, Proc. 4th IEEE Symp. on Logic in Computer Science (1989), 71–79

[AV 91] S. Abiteboul, V. Vianu, *Generic Computation and Its Complexity*, Proc. 23rd ACM Symp. on Theory of Computing (1991), 209–219

[BIS 86] D.A.M. Barrington, N. Immerman, H. Straubing *On Uniformity within* NC^1, Journal of Computer and System Sciences **41** (1990), 274–306

[CL 86] K. Compton, C. Laflamme *An Algebra and a Logic for* NC^1, Information and Computation **87** (1990), 241–263

[DGS 86] L. Denenberg, Y. Gurevich, S. Shelah *Definability by Constant-Depth Polynomial-Size Circuits*, Information and Control **70** (1986), 216–240

[GL 81] P. Gacs, L.A. Levin *Causal Nets or What is a Deterministic Computation*, Information and Control **51** (1981), 1–19

[G 92] M. Grohe,
Diplomarbeit, Universität Freiburg (1992)

[GL 84] Y. Gurevich, H.R. Lewis *A Logic for Constant-Depth Circuits*, Information and Control **61** (1984), 65–74

[G/S 86] Y. Gurevich, S. Shelah, *Fixed Point Extensions of First Order Logic* Annals of Pure and Applied Logic **32** (1986), 265–280

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

[I 89] N. Immerman, *Expressibility and Parallel Complexity*, SIAM Journal of Computation **18** (1989), 625–638

[V 82] M. Vardi, *Complexity of Relational Query Languages*, Proc. 14th ACM Symp. on Theory of Computing (1982), 137–146

[W 87] I. Wegener, *The Complexity of Boolean Functions*, Teubner (1987)