INFORMATION AND CONTROL 70, 216–240 (1986)

Definability by Constant-Depth Polynomial-Size Circuits

LARRY DENENBERG*

Harvard University, Cambridge, Massachusetts

YURI GUREVICH†

University of Michigan, Ann Arbor, Michigan

AND

SAHARON SHELAH‡

Hebrew University, Tel-Aviv, Israel

A function of boolean arguments is *symmetric* if its value depends solely on the number of 1's among its arguments. In the first part of this paper we partially characterize those symmetric functions that can be computed by constant-depth polynomial-size sequences of boolean circuits, and discuss the complete characterization. (We treat both uniform and non-uniform sequences of circuits.) Our results imply that these circuits can compute functions that are not definable in first-order logic. In the second part of the paper we generalize from circuits computing symmetric functions to circuits recognizing first-order structures. By imposing fairly natural restrictions we develop a circuit model with precisely the power of first-order logic: a class of structures is first-order definable if and only if it can be recognized by a constant-depth polynomial-time sequence of such circuits.

INTRODUCTION

A function of boolean arguments is *symmetric* if its value depends solely upon the number of 1's among its arguments; for example, the parity function (whose value is 1 iff an even number of its arguments are 1) and the counting function (whose value is the binary representation of the number of arguments that are 1) are symmetric. In [7] it is shown that there

* Partially supported by NSF Grants MCS 80-05386-A01 and 82-03482.
† Partially supported by NSF Grants MCS 83-01022 and DCR 85-03275.
‡ Partially supported by the Israel–U.S. Binational Science Foundation.

0019-9958/86 $3.00

exists no constant-depth polynomial-size (i.e., having polynomially bounded size) sequence of circuits computing the parity function. In this paper we generalize this result, partially characterizing the symmetric functions that can be so computed and discussing the complete characterization. We also consider the relationship between constant-depth polynomial-size circuits and logical formulas, and find a class of circuits which precisely captures the expressive power of first-order logic. This paper presents material originally appearing in [3] together with several new results.

Our circuits are built with three kinds of gates: inverters (negation gates) with one input and one output as usual, plus $\wedge$ and $\vee$ gates with an unbounded number of inputs and one output. The output from any gate may be connected to an arbitrary number of other gates, that is, there is no bound on fan-out. A circuit of *order n* is one with n inputs, usually denoted by $x_1, x_2, \ldots, x_n$. Without loss of generality we assume that all circuits have a single boolean output. A circuit is *positive* (*negative*) if there are an even (odd) number of inverters on every path from every input to the output; in particular, any circuit without inverters is positive. A positive (negative) circuit defines a monotone increasing (decreasing) boolean function. The *size* of a circuit is the number of its wires, and the *depth* of a circuit is the number of gates on the longest path between an input and the output. We sometimes use "low" and "high" as synonyms for the boolean circuit values 0 and 1, respectively.

Let C be a circuit of order n. If C computes a symmetric function of its inputs we say that C itself is symmetric and that it *recognizes* the set $\{i \mid C$ gives output 1 when exactly i inputs are $1\}$. A function K such that $K(n) \subseteq \{0, 1, \ldots, n\}$ is *circuit definable* if there exists a family $\{C_n\}$ of circuits, a integer d, and a polynomial π such that for every i, C_i has order i, recognizes $K(i)$, and has depth at most d and size at most $\pi(i)$. In addition, K is *uniformly circuit definable* if there exists a log-space bounded Turing machine that outputs an encoding for C_i given the unary notation for i as input.

We can regard K as a function from positive integers to bit strings, letting $K(n)$ correspond to a string of length $n+1$ whose ith member is 1 if $i \in K(n)$ and 0 otherwise. Our first results state that a function K is not circuit definable unless the central portions of these string contain only 0's or only 1's, where the "center" includes all of $K(n)$ except for a segment of length $o(n^e)$ on either end. If a larger center is similarly homogeneous—all but segments of length $O(\log^k n)$—then K is in fact circuit definable. Using the notation $[a, b]$ to denote the set of integers between a and b inclusive we can state the precise results as follows:

THEOREM 1. *Let K be a function assigning a subset of $\{0, 1, \ldots, n\}$ to each positive integer n, such that for some k each set $K(n)$ either includes or is dis-*

joint from the set $[\log^k n, n - \log^k n]$. *Then K is circuit definable. Moreover, if K is log-space computable it is uniformly circuit definable, and if K is closed upward or downward it is definable by positive or negative circuits, respectively.*

THEOREM 5. *For every real* $\varepsilon > 0$, *polynomial* π, *and integer* $d > 0$ *there exists N such that the set recognized by every symmetric circuit with* $n > N$ *inputs, depth d, and size at most* $\pi(n)$ *either contains or is disjoint from* $[n^\varepsilon, n - n^\varepsilon]$.

These results, proved in Sections 1 and 2 respectively, extend both the negative results of [7, 2] and some of the positive results of [1] for symmetric circuits. For example, let K_1 represent the majority function: then $K_1(n)$ contains $\lceil n/2 \rceil$ but not $\lceil n/2 \rceil - 1$, so for large n it is neither contained in nor disjoint from $[n^{1/2}, n - n^{1/2}]$. Hence given any polynomial size bound and constant depth bound Theorem 5 implies that for n sufficiently large no circuit so bounded can compute $K_1(n)$. On the other hand, $K_2(n) = \{\lceil \log^k n \rceil\}$ is circuit definable for any k.

Theorems 1 and 5 do not completely classify all possible functions—consider, for example, $K_3(n) = \{\lceil (\log n)^{\log \log n} \rceil\}$. In fact, none of the remaining functions is circuit definable; that is, the circuit definable functions are exactly those that qualify under Theorem 1. This result follows along lines explained in [6] (in which a nonuniform version of Theorem 1 and a stronger version of Theorem 5 are independently obtained) by combining a lemma of that paper with more recent work of [13] on lower bounds for the sizes of circuits computing parity.

Our research began with the conjecture that a class of structures is definable by a constant-depth polynomial-size bounded sequence of symmetric circuits if and only if it is definable by a first-order formula (we define in Sect. 3 the notion of definability of a class of structures). Although the "if" implication of the conjecture is obvious, Theorem 1 disproves the "only if" implication: circuit definability does not imply first-order definability even in the case of structures with a single unary predicate. Motivated by the work of Myers in [11], we introduce in Section 3 a stronger notion of circuit definability which allows us to revive (and prove) the conjecture in the case of structures containing unary predicates only. We also construct an example showing that the conjecture fails (for our stricter notion of circuit definability) in the case of structures containing even a single binary predicate; the same example allows us to disprove a conjecture of Myers concerning file machines. In the final section we define an even more restricted version of circuit definability and prove it precisely equivalent to definability of sets of structures by first-order formulas.

Notation. log refers to logarithm base 2, ln to natural logarithms. K is

always a function from positive integers to sets of natural numbers such that $K(n) \subseteq \{0, 1,..., n\}$. U_n (the canonical universe of n elements) is equal to $\{0, 1,..., n-1\}$. If C is a symmetric circuit we use $C(k)$ to denote the output of C (either 0 or 1) when k inputs are 1.

1. Circuit Definability of $\log^k n$

The main result of this section is the circuit definability of any function K whose "center" does not vary, as defined in

THEOREM 1. *Let K be a function assigning a subset of $\{0, 1,..., n\}$ to each positive integer n, such that for some k each set $K(n)$ either contains or is disjoint from $[\log^k n, n - \log^k n]$. Then K is circuit definable. Moreover,*

1. *If K is closed downward (that is, if $i \in K(n)$ implies $j \in K(n)$ for all $0 \leqslant j \leqslant i$) then it is definable by negative circuits,*

2. *If K is closed upward (that is, if $i \in K(n)$ implies $j \in K(n)$ for all $i \leqslant j \leqslant n$) then it is definable by positive circuits,*

3. *If K is log-space computable then it is uniformly circuit definable,*

4. *If K is log-space computable and closed upward (downward) then it is uniformly definable by positive (negative) circuits.*

(A function K is log-space computable if there exists a log-space bounded Turing machine that, given 1^n and $k \in \{0, 1,..., n\}$, decides whether $k \in K(n)$.)

We begin by giving a correspondence between sequences of circuits and sentences of first-order logic. Recall that U_n is $\{0, 1,..., n-1\}$, the canonical universe of size n.

LEMMA 2. *Let $\phi(X)$ be a first-order sentence with a sequence σ of predicate and function symbols and an additional unary predicate symbol X. Let $S_1, S_2,...$ be σ-structures such that the universe of S_i is U_i for each i. Suppose that for some N and for all $n \geqslant N$ the sentence $\phi(X)$ expresses $|X| \in K(n)$ in S_n. (That is, for every interpretation X^n of X on U_n the sentence $\phi(X)$ is true in S_n enriched by X^n if and only if $|X^n| \in K(n)$.) Then K is circuit definable. Moreover, if the sequence $\{S_i\}$ of structures is uniform in the sense that there exists an algorithm deciding atomic σ-statements about S_n in space $\log n$, then K is uniformly circuit definable, and if ϕ is positive (negative) in X then all circuits constructed can be made positive (negative).*

Proof. We remark that the converse of the first part of this lemma is proved in [8].

Since functions can be defined in terms of predicates, we may assume

without loss of generality that every member of σ is a predicate symbol. For $n < N$ special case circuits can be constructed to recognize $K(n)$. Given $n \geqslant N$ we build a circuit C_n as follows: First, construct a new sentence ϕ_n by replacing all quantifiers in ϕ by conjunctions and disjunctions. (For example, the formula $\forall x \chi(x)$ is replaced by $\chi(0) \wedge \chi(1) \wedge \cdots \wedge \chi(n-1)$.) Second, replace every atomic σ-formula in ϕ_n by its truth-value. The resulting formula can be converted directly into a circuit; all remaining atomic formulas are of the form Xi, which represents the boolean value of input number i. The depth of the circuit depends only upon the syntactic complexity of ϕ and so is constant with respect to n. The size is polynomial in n since each expansion of a quantifier can at worst multiply the formula size by n, and the number of quantifiers is constant. If the truth values of the atomic formulas can be computed uniformly, then the circuits can be constructed uniformly. Finally, since each input corresponds to an occurrence of X the last statement of the lemma is obvious. ▮

With an eye to the future we give an alternative description of the construction in this lemma: Given ϕ and n, build C_n starting at its single output, labelling that "wire" with the formula ϕ. If ϕ is of the form $\phi_1 \wedge \phi_2$ then make it the output of an $\wedge$-gate with inputs from the wires labelled ϕ_1 and ϕ_2; the cases $\phi = \phi_1 \vee \phi_2$ and $\phi = \neg\phi_1$ are handled similarly. If ϕ is of the form $\forall x\ \chi(x)$ then make it the output of an $\wedge$-gate with inputs from n wires labelled $\chi(0), \chi(1), \ldots, \chi(n-1)$, respectively. Continue building C_n in this manner, expanding and labelling each wire. Since ϕ is a sentence we eventually have wires whose labels are atomic formulas Xi, and these of course are the inputs. The circuit so constructed is identical to the circuit constructed in the lemma.

To build the formula that will express "$|X| \in K(n)$" we need a function that maps a small set X, which is spread out over U_n, into a small initial segment of U_n—and which does so in a 1–1 manner. The next lemma shows that the residue function can be so used although in a nonconstructive way.

Notation. Let $\mathrm{res}(i, j)$ denote the residue of i modulo j and let $\mathrm{res}(i, j, k)$ abbreviate $\mathrm{res}(\mathrm{res}(i, j), k)$. We also define $L(n) = (\log n)^{1/3}$ and $L^k(n) = (\log n)^{k/3}$.

LEMMA 3. *There exists n_0 such that for every $n \geqslant n_0$ and every $X \subseteq U_n$ with $|X| \leqslant L(n)$ there are positive integers $u < n$ and $v < \log n$ such that if $x, y \in X$ and $x \neq y$, then $\mathrm{res}(x, u, v) \neq \mathrm{res}(y, u, v)$.*

Proof. The function $\psi(x) = \sum \{k \ln p \mid p \text{ is prime and } p^k \leqslant x < p^{k+1}\}$ is well known in number theory. In particular, it is known that $\psi(x) \to x$ as $x \to \infty$ [9]. Choose n_0 such that $\psi(x) > x \ln 2$ whenever $x \geqslant 1 + \log\log n_0$.

Now suppose that $n \geqslant n_0$. Let m be $L(n)$ and let $X \subseteq U_n$ be given with $|X| \leqslant m$. Let u be the smallest integer such that for $x, y \in X$, $x \neq y$ implies $\text{res}(x, u) \neq \text{res}(y, u)$—obviously $u < n$. If $u < \log n$ take $v = u$ and the lemma is proved. Otherwise, we have $\psi(u-1) > (u-1)\ln 2$.

Set A to the least common multiple of $\{x - y \mid x, y \in X \text{ and } x > y\}$ and set B to the product of $\{p^k \mid p \text{ is prime and } p^k < u \leqslant p^{k+1}\}$. We note that B divides A: given any factor p^k of B with p prime and k maximal we have $p^k < u$, so the definition of u ensures that there exist x and y in X such that $x - y$ is divisible by p^k. Now B is less than $n^{m(m-1)}$ since A is. Moreover, $\ln B = \psi(u-1) > (u-1)\ln 2$, so $B > 2^{u-1}$. It follows that $2^{u-1} < n^{m(m-1)}$, and finally we get $u - 1 < m(m-1)(\log n)$ which implies that $u < m^2 \log n$.

Let X' be the set of residues modulo u of the members of X, and let v be the smallest integer such that for every distinct $x', y' \in X'$ we have $\text{res}(x', v) \neq \text{res}(y', v)$. If $v < \log u \leqslant \log n$ then we are done. Otherwise, repeat the argument of the preceding paragraph with u substituted for n and with X' substituted for X; n_0 has been chosen large enough to make this possible. The final inequality becomes $v < m^2 \log u < m^2(\log(m^2 \log n)) < \log n$, and the proof is complete. ∎

We are now ready to begin writing formulas. For each k, $n > 0$ let S_{kn} be a structure with universe U_n and with interpretations for:

- binary function symbols $+$, $*$, and "res," plus a binary predicate symbol $<$, all of whose interpretations in S_{kn} are the standard ones,
- individual constants $L^1, L^2, \ldots, L^k$, where for each i the interpretation of L^i in S_{kn} is the value $L^i(n)$,
- a binary predicate symbol whose interpretation in S_{kn} expresses "the xth bit in the binary notation of y is 1,"
- a binary predicate sumbol whose interpretation in S_{kn} expresses "x is the number of 1's in the binary notation for y," and
- binary predicate symbols $E_2, E_3, \ldots, E_k$ whose interpretations in S_{kn} will be explained later.

LEMMA 4. *For each positive integer k there exists a first-order formula $\phi_k(a, X)$ containing the predicate and function symbols of S_{kn} plus an individual constant symbol a and a monadic predicate symbol X, such that for all sufficiently large n and all interpretations X^n of X and a^n of a over U_n:*

(i) *if $\phi_k(a, X)$ is true in S_{kn} enriched with a^n and X^n, then $a^n = |X^n|$,*

(ii) *if $a^n = |X^n| \leqslant L^k(n)$, then $\phi_k(a, X)$ is true in S_{kn} enriched with a^n and X^n.*

Proof. The proof is by induction on k. The sentence ϕ_1 says that there exist u, v, w satisfying the conjunction of the following formulas:

- $\forall x,\ y \in X[x \neq y \to \mathrm{res}(x, u, v) \neq \mathrm{res}(y, u, v)]$,
- $\forall y\ [\exists x \in X(y = \mathrm{res}(x, u, v)) \leftrightarrow$ the yth bit of w is 1],
- a is the number of 1's in the binary notation for w.

This formula states that for some u and v the elements of X are mapped one-to-one onto a set of size a, which is counted by a bit vector w having the correct number of 1's. Obviously ϕ_1 cannot be satisfied in S_{1n} if the cardinality of X is other than a. On the other hand, when $|X| = a \leqslant (\log n)^{1/3}$ and n is sufficiently large, Lemma 3 ensures that values of u and v exist to make ϕ_1 true. In particular, $\phi_1(L^1, X)$ means that X has exactly $(\log n)^{1/3}$ elements. (A technical point: Lemma 3 also guarantees that v can be chosen less than $\log n$, which ensures that the corresponding bit vector w will be a member of the universe U_n.)

We begin the proof of the inductive case by explaining the interpretations of the predicates E_k. Let $V_{kn} = \{0, 1, \ldots, L^{2k+3}(n)\}$ and let P_{kn} be the set of all subsets of V_{kn} with at most $L(n)$ elements. For sufficiently large n, the size of P_{kn} is less than n. (This is obvious if we take logarithms: the size of P_{kn} is clearly less than $|V_{kn}|^{L(n)}$, whose logarithm is $(\log n)^{1/3}((2k+3)/3)(\log\log n) = o(\log n)$.) The interpretation of $E_k(x, y)$ is that x is an elements of the yth member of P_{kn}, using (say) lexicographic ordering of the elements of P_{kn}.

So assume that the lemma has been proved up to $k-1$. We write an auxiliary formula $\psi_k(a, Y)$ which uses an additional monadic predicate letter Y. In the following informal description of ψ_k we use A_y to denote the yth element of P_{kn}; of course, all references to A_y are actually translated using E_k. The formula $\psi_k(a, Y)$ says that there exist b, c, and z satisfying the conjunction of the following clauses:

- $b \leqslant L^1$, $c \leqslant L^{k-1}$,
- $a = b * L^{k-1} + c$,
- $b = |A_z|$,
- if $A_z \neq \varnothing$ then there are exactly L^{k-1} elements of Y that are less than the smallest element of A_z,
- for every two successive elements $u < v$ of A_z there are exactly L^{k-1} elements of Y in the interval $[u, v)$,
- there are exactly c elements of Y that are greater than every element of A_z. (In particular, $|Y| = c$ if A_z is empty.)

Each of these clauses is easily expressible using ϕ_1 and ϕ_{k-1}, and it is easy to see that $\psi_k(a, Y)$ expresses $a = |Y|$ in S_{kn} for $Y \subseteq V_{kn}$ and sufficiently large n.

Finally, we write $\phi_k(a, X)$, which says that there exists u satisfying the conjunction of the following clauses:

- $u \leqslant L^{2k+3}$,
- $\forall x, y \in X\ [x \neq y \rightarrow \mathrm{res}(x, u) \neq \mathrm{res}(y, u)]$,
- $\psi_k(a, \{y: \exists x \in X(y = \mathrm{res}(x, u))\})$,

where the final clause is actually written by expanding $\psi_k(a, Y)$ and eliminating Y in the usual way.

It is obvious that the truth of $\phi_k(a, X)$ in S_{kn} enriched with interpretations a^n and X^n implies that $a^n = |X^n|$, when n is sufficiently large (as required by ψ_k). On the other hand, suppose that the interpretations a^n and X^n are given with $a^n = |X^n| \leqslant L^k(n)$. We claim that if n is sufficiently large there is $u \leqslant L^{2k+3}(n)$ such that the function $f(x) = \mathrm{res}(x, u)$ is 1–1 on X^n. (The proof is the same as the first part of the proof of Lemma 3, but with m now equal to $L^k(n)$.) Thus $\phi_k(a, X)$ will be true, and the lemma is proved. ▮

Now we are ready to prove Theorem 1. First of all, suppose that K assigns a subset of $\{0, 1, ..., n\}$ to each positive integer n and that $K(n)$ is contained in $[0, \log^k n]$ for all n. Just let $\phi(X)$ be the formula

$$\exists a[\phi_k(a, X) \wedge Ka]$$

and apply Lemma 2 to $\phi(X)$ and the structures S_{kn} enriched by the interpretation "$a \in K(n)$" for atomic formulas Ka. Next observe that if $K(n)$ is circuit definable then so are the functions $K'(n) = \{i \mid n - i \in K(n)\}$ (simply negate each input in the circuits defining K) and $\bar{K}(n) = \{0, 1, ..., n\} - K(n)$ (simply negate the output of each circuit defining K). We also note that if two functions $K_1(n)$ and $K_2(n)$ are circuit definable, then so is the function $K(n) = K_1(n) \cup K_2(n)$, since we can use an $\vee$-gate to combine the circuits. These operations suffice to establish the circuit definability of any function K satisfying the conditions of the theorem.

Next suppose that K is closed downward, and again assume to begin with that $K(n) \subseteq \{0, 1, ..., \log^k n\}$ for each n. Replace the "$\leftrightarrow$" in ϕ_1 with "$\rightarrow$." Now ϕ_1 is negative in X and expresses "$|X| \leqslant a$" in the structures S_{1n}. For each $k > 1$ replace "exactly" by "at most" everywhere in the definition of ψ_k; the sentences $\phi_k(a, X)$ all become negative in X and express "$|X| \leqslant \alpha$" in the structures S_{kn}. The formula used to define K in the previous paragraph is thus negative in X, and therefore so are the circuits constructed by Lemma 2. Given a negative circuit computing $K(n)$ the circuits computing $K'(n)$ and $\bar{K}(n)$ (as constructed in the preceding paragraph) are positive. These constructions suffice to define any K that is closed upward or closed downward, since (for example) if K is closed upward then $\bar{K}$ is closed downward, and if $K(n)$ is closed downward but not contained in $\{0, 1, ..., \log^k n\}$ then $\bar{K}'(n)$ is closed downward and is so contained.

Finally, all of the interpretations of functions and predicates used in this section can obviously be computed in log space. If K is also log-space computable then by Lemma 2 every circuit family constructed here can be constructed uniformly, independently of its positive or negative character. ▮

2. Non-definability of n^ε

We say that a circuit C of order n *accepts* (resp. *rejects*) a natural number $p \leqslant n$ if its output is 1 (resp. 0) whenever exactly p inputs are 1, and we write $C(p)=1$ (resp. $C(p)=0$) just as in the case where C is symmetric. Note that a circuit of order n is symmetric if and only if it either accepts or rejects each p for $0 \leqslant p \leqslant n$. For $0 \leqslant p, q \leqslant n$ we say that C *distinguishes* p *from* q if it accepts one of p, q, and rejects the other. In this notation, Theorem 5 follows immediately from the following slightly stronger result:

Lemma 5. *Given a polynomial* π, *an integer* $d>0$, *and a real* $\varepsilon>0$, *there exists* N *such that for all* $n \geqslant N$ *and all* p, q *such that* $n^\varepsilon < p$, $q < n - n^\varepsilon$, *no circuit with order* n, *size bounded by* $\pi(n)$, *and depth at most* d *can distinguish* p *from* q.

The proof uses the technique of [7], slightly modified to obtain this more general result. In addition, our theorem is somewhat stronger in that it is phrased in terms of individual circuits, not of circuit families.

We call a circuit of order n *center discriminating* if it distinguishes $\lfloor n/2 \rfloor$ from $\lfloor n/2 \rfloor + 1$. If in any such circuit we fix any number of inputs at 0 and an equal number at 1 the resulting circuit is also center discriminating; this is the key property to be used in our proofs. We begin by proving that center discriminating circuits of large order and small depth and size cannot exist. At the end of this section we prove Lemma 5 by showing that any circuit distinguishing p from q as in the statement of the lemma could be transformed into such a center discriminating circuit.

To start, we restrict our attention to circuits consisting of alternating stages of $\wedge$- and $\vee$-gates, where each gate's inputs come exclusively from gates of the preceding stage and each gate's output connects to (any number of) gates of the next stage. The inputs all connect to gates of the first stage, which is always a stage of $\vee$-gates; thus a circuit of depth 2 is always in conjunctive normal form. We allow no inverters, but the negations of the inputs are available at no cost. (Converting a conventional circuit to one conforming to these restrictions entails at most an increase in size of a constant factor and no increase in depth [7].) All circuits in the remainder of this section are assumed to have this special form.

LEMMA 6. *For every polynomial π there exists N such that no depth 2 circuit of order $n \geqslant N$ and size at most $\pi(n)$ is center discriminating.*

Proof. We prove instead the following fact, of which the lemma is an obvious consequence: If C has order n and depth 2 and distinguishes k from $k+1$, then C has size at least $\binom{n}{k}(n-k)$.

In the trivial cases $k=0$ and $k=n-1$ we must show that C has size at least n, but this is obvious since any circuit which depends on all its inputs has at least n wires. Let C be a depth 2 circuit distinguishing k from $k+1$. Now C is the conjunction of an arbitrary number of $\vee$-gates, each of which has an arbitrary number of connections to inputs $x_1, x_2, \ldots, x_n$ and negations of inputs $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n$. We may assume that no such gate contains duplicates among its inputs.

Suppose we discard from C each $\vee$-gate whose output is 1 whenever exactly k inputs or exactly $k+1$ inputs are 1. The resulting circuit (still called C) still distinguishes k from $k+1$ since the new circuit is equivalent to the old one whenever exactly k or $k+1$ inputs are 1. For example, we discard any $\vee$-gate containing both x_i and $\bar{x}_i$ for any i.

Consider an $\vee$-gate of C with p positive inputs and q negative inputs. Since at this point we have $p+q \leqslant n$, every such gate falls into exactly one of the following classes:

I. $q > k+1$, and hence $p < n-k-1$

II. $p > n-k$, and hence $q < k$

III. $q < k+1$, $p < n-k$

IV. $p = n-k$, and hence $q \leqslant k$

V. $q = k+1$, and hence $p \leqslant n-k-1$.

Now C no longer has gates of class I, for if at most $k+1$ inputs are high and the rest low then at least one of the q negative inputs is low and the gate's output is 1; hence gates of class I have output 1 whenever k or $k+1$ inputs are high. Similarly, C has no gates of class II. (This argument fails for $k=0$ or $k=n-1$ but those cases were handled separately.)

We next show that C can have no gate of class III. For let A be such a gate. Suppose that the q inputs connected negatively to A are high and the p inputs connected positively to A are low. There are $n-p-q \geqslant (k+1)-q$ inputs left. We can set $k-q$ of these inputs high (and the rest low) to establish a situation where A has output 0 with k inputs high. But we can also set another one high, and A's output will be 0 with $k+1$ inputs high. Since C's output is 0 whenever A's output is 0 we have $C(k) = C(k+1) = 0$ contrary to assumption.

Now C cannot contain gates of both of the remaining classes, for if C has a gate of class IV we can set its $n-k$ positive inputs low and the rest high to see that $C(k) = 0$ and hence $C(k+1) = 1$. But a similar argument shows

that if C has a gate of class V we must have $C(k+1)=0$ and hence $C(k)=1$. So suppose that C is built entirely out of gates in class IV. Since $C(k)=0$, for every set of k inputs there must be a gate that is low when the members of the set are all high. But a class IV gate can be low for only a single choice of k inputs, since if any of the gate's $n-k$ positive inputs are high then the gate is high. Thus C has at least $\binom{n}{k}$ such gates, each with at least $n-k$ wires in and one out, for a total of $\binom{n}{k}(n-k+1)$ wires at least.

If C has only class V gates similar reasoning shows that it contains at least $\binom{n}{k+1}$ such gates each with at least $k+2$ wires. The minimum number of wires is therefore $\binom{n}{k}(n-k)(k+2)/(k+1)$, which exceeds $\binom{n}{k}(n-k)$. ∎

We next extend this result to arbitrary depth.

LEMMA 7. *For all polynomials π and integers $d \geqslant 2$ there exists N such that there is no center discriminating circuit of order $n \geqslant N$, size at most $\pi(n)$, and depth d.*

Proof. The proof is by induction on d. The base case $d=2$ is settled by the previous lemma. To prove the inductive case, we show how to construct from any center discriminating circuit of order n (with n sufficiently large), depth d, and size $\pi(n)$ another center discriminating circuit of size $a \cdot \pi(n)$, depth $d-1$, and order $\Omega(n^s)$ for constants a and s independent of n. The lemma then follows by a straightforward argument which we omit.

The construction is very similar to that of [7]. A *projection* ρ of a circuit C is a mapping from the inputs of C to the set $\{0, 1, *\}$. The projected circuit C^ρ has the same behavior as C, but with some inputs fixed at 0 or 1 as specified by ρ (inputs of C mapped to $*$ by ρ are the only ones that remain inputs in C^ρ). Given a circuit C as described above, we first project C to produce a center discriminating circuit C' of the same size and depth and order at least $n' = \Omega(n^s)$, in which every first level $\vee$-gate depends on at most c inputs where c depends only on the degree of π. We then project C' to get a center discriminating circuit C'' of the same size and depth and order at least $n'' = \Omega(n'^s)$ in which every second level $\wedge$-gate depends on at most b inputs, b a constant dependent only on c. We then rewrite the second level $\wedge$-gates of C'' explicitly as sums of products, merge the second and third levels (both now composed of $\vee$-gates) of the resulting circuit, and take the duals of all the gates to build a circuit of depth $d-1$ and size increased at most by a constant factor.

The only difference between our construction and that of [7] is that the projections involved must be such that the resulting circuits are still center discriminating. The construction in [7] requires only that the projected circuits compute parity, and obviously *any* projection of a circuit computing parity does so. To satisfy our stronger requirement it suffices to insure that ρ assigns the same number of 0's as 1's.

Consider a random projection ρ with the following probability distribution, which differs only slightly from the one in [7]:

$$\Pr[\rho(x_i) = *] = n^{-1/3}$$
$$\Pr[\rho(x_i) = 0] = \Pr[\rho(x_i) = 1] = \tfrac{1}{2}(1 - n^{-1/3}).$$

Let $\mu = \frac{1}{2}(1 - n^{-1/3})$, the expected number of 0's assigned by the projection. Of course, μ is also the expected number of 1's assigned. We say that such a projection *fails* if any of the following are true:

(i) The number of $*$'s assigned by ρ is less than $\frac{1}{2}n^{2/3}$.

(ii) The number of 0's assigned by ρ differs from μ by more than $n^{2/3}/8$.

(iii) The number of 1's assigned by ρ differs from μ by more than $n^{2/3}/8$.

(iv) Some first-level $\vee$-gate of C^ρ depends on more than $8k$ inputs, where k is the degree of π.

To bound the probability that ρ fails, we bound separately the probabilities that ρ satisfies (i)–(iv) and then add. Now $\Pr[\rho$ satisfies (iv)$]$ is $o(n^{-k})$ by an argument identical to that of [7]: any first level $\vee$-gate of C with more than $8k \ln n$ inputs contains (with high probability) some input assigned 1 by ρ, and hence does not depend on any of its inputs, while any such gate with fewer inputs is with high probability assigned fewer than $8k$ $*$'s. The other three probabilities are computed by simple applications of Chebyshev's theorem. (For example, the probability that (ii) holds is the probability that a random variable binomially distributed with parameter $p = \frac{1}{2}(1 - n^{-1/3})$ differs from its mean by more than $d = n^{2/3}/8$, which is at most $np(1-p)/d^2 = o(n^{-1/3})$.) For sufficiently large n, the probability that ρ fails can be shown to be strictly less than 1. Thus, there is some ρ that does *not* fail.

Once we have such a ρ, we proceed as follows: Let ρ assign p 0's and q 1's. Assume $p > q$; a dual argument handles the other case. Set $r = p - q$. By (ii) and (iii) above, r is at most $n^{2/3}/4$. Select any r inputs of C^ρ and fix them at 1, thus producing C'. By (i) and the bound on the size of r, C' has at least $n^{2/3}/4$ inputs. Condition (iv) is still under control since in passing from C^ρ to C' the number of inputs to any gate can only decrease. Finally, C' is center discriminating because it is the result of applying to a center discriminating circuit a projection with an equal number of 0's and 1's.

The argument that produces C'' from C' is similar: Take a projection chosen from the same distribution and say that it fails if it satisfies any of (i)–(iii) or if any second level $\wedge$-gate depends on more than b inputs. The arguments that bound the probability of failure due to (i)–(iii) are iden-

tical, and the argument that bounds the probability of failure due to the new condition is exactly the claim of [7, p. 20]: For every c there is a constant b_c such that if A is any depth 2 circuit all of whose $\vee$-gates are of size at most c, the probability that A^ρ depends on more than b_c inputs is $o(n^{-k})$. ∎

Finally, we prove the main lemma. The idea is that if C distinguishes between p and q which are far enough away from the "endpoints" 0 and n, we can fix some of the inputs of C to obtain a center discriminating circuit, which is impossible (for C with large enough order) by Lemma 7.

Proof of Lemma 5. Given π, ε, and d, let $g(n) = (n/2)^{1/\varepsilon}$ and let $\pi'(n)$ be $\pi(g(n))$. Choose N' by the previous lemma such that no circuit of order $n' \geqslant N'$, size at most $\pi'(n')$, and depth at most d is center discriminating, and let $N = g(N')$. We claim that this N fulfills the conditions of the lemma.

For assume C has order $n \geqslant N$, size at most $\pi(n)$, depth at most d, and that it distinguishes p from q for some $n^\varepsilon < p$, $q < n - n^\varepsilon$. Then there must exist r such that $n^\varepsilon < r < n - n^\varepsilon$ and C distinguishes r from $r+1$. If $r \leqslant n/2$ fix $n-2r$ inputs of C at 0; otherwise, fix $2r-n$ inputs of C at 1. In either case, the resulting circuit has order $n' \geqslant 2n^\varepsilon > N'$ and is center discriminating. It also has size at most $\pi(n) = \pi(g(2n^\varepsilon)) = \pi'(n')$ and depth d, and so it cannot exist. ∎

3. Strict Circuit Definability

In this section we study a new kind of definability of classes of structures by circuits and give a counterexample to a conjecture of Myers [11]. Throughout the section we consider only signatures with a finite number of predicate symbols and no function symbols, and structures whose universe is contained in U_n for some n (thus all structures are finite). As before, all circuits are assumed to have a single boolean output.

We begin with a few definitions. When σ is a signature, let L_σ^n be the set of atoms $P(a_1, a_2, \ldots, a_r)$, where P is a predicate symbol of σ and each a_i is an element of U_n. A circuit C *formatted for* (σ, n) is a circuit together with a surjection from the inputs of C onto L_σ^n; such a C has *order* n, and we say that each input of C is *labelled* with the corresponding atom in L_σ^n. We also say that a circuit is formatted for σ if it is formatted for (σ, n) for some $n > 0$.

A circuit formatted for (σ, n) can be thought of as an acceptor for σ-structures of order n: given such a structure S we place on each input of C the value (in S) of the atom with which it is labelled, and C accepts S if and only if its output is 1. We say that such a circuit is *symmetric* if the set

of structures that it accepts is closed under isomorphism; that is, C is symmetric if for any isomorphic structures S_1 and S_2 (each with universe U_n) C accepts S_1 if and only if C accepts S_2. A class Σ of σ-structures is *circuit definable* if there is a constant-depth, polynomial-size bounded sequence $C_1, C_2, \ldots$ of circuits, where each C_n is formatted for (σ, n), is symmetric, and accepts a structure S on U_n if and only $S \in \Sigma$. (Our previous definition of circuit definability is exactly the special case in which σ contains only a single monadic predicate symbol.) The symmetry condition implies that every circuit definable class of structures is closed under isomorphism; we consider only such classes in the remainder of this paper.

A class Σ of structures is called *elementary* if it is definable by a first-order sentence; that is, if there exists a sentence ϕ whose models are exactly the structures in Σ. It might seem plausible that circuit definability would exactly correspond to definability by first-order sentences. Unfortunately, the definition just proposed is too strong, as we have already seen: A circuit family recognizing the function $K(n) = \{\lceil \log n \rceil\}$ as in Section 1 defines the set $\Sigma_{\log}$ of structures containing interpretations for a single monadic predicate symbol X in which X is true of exactly $\lceil \log n \rceil$ members of the universe; and it can be shown by a quantifier elimination argument (see [10]) that $\Sigma_{\log}$ is not elementary. On the other hand, a construction like that of Lemma 2 easily establishes that every elementary class is circuit definable.

The following alternative scheme is motivated by the file machines of Myers [11]. Given a signature σ, let σ^+ be σ enriched with a new monadic predicate symbol U. If S is a σ^+-structure let $S|U^S$ be the σ-substructure of S whose universe is $\{x: Ux \text{ is true in } S\}$. A class Σ of σ-structures is *strictly circuit definable* if the class $\{S: S \text{ is a } \sigma^+\text{-structure and } S|U^S \in \Sigma\}$ is circuit definable.

Informally the idea is this: a circuit C_n of a family strictly defining a class of σ-structures looks like a circuit formatted for (σ, n) except that it comes equipped with another set of n inputs for U which we call "universe" inputs. Such a circuit can accept as input any structure S whose universe is contained in U_n: we simply place the truth-values of the atoms of S on the inputs of C_n as usual, use the universe inputs to specify which members of U_n are in the universe of S, and place arbitrary values on the remaining inputs of C_n (which are those inputs labelled $P(a_1, \ldots, a_r)$, where at least one of the a_j is not in the universe of S). For example, a circuit of order n strictly defining the class $\Sigma_{\log}$ described above would have $2n$ inputs arranged in pairs—for each i it has a universe input corresponding to U_i and an associated "predicate" input corresponding to Xi. Such a circuit has output 1 if and only if k of its universe inputs are 1 and exactly $\lceil \log k \rceil$ of the predicate inputs associated with those inputs are 1.

In this section we show the equivalence of strict circuit definability and

first-order definability in the case of languages with monadic predicate letters only. Lemma 2 of Section 1 provides this connection between the circuits of the original scheme and first-order formulas, for languages with a single monadic predicate but where formulas are allowed to have other, interpreted, predicates and functions. Our new scheme captures precisely the defining power of such first-order sentences where no auxiliary interpreted predicates or functions are permitted.

As before, we easily show that strict circuit definability is at least as powerful as first-order definability:

THEOREM 8. *Every elementary class of structures is strictly circuit definable.*

Proof. Let Σ be such a class and let ϕ be a first-order sentence whose models are exactly the structures in Σ. Let U be a new monadic predicate, and let $\phi \mid U$ be the result of restricting the quantifiers of ϕ to U; that is, replacing $\exists x\, \psi$ by $\exists x\, (Ux \wedge \psi)$ and replacing $\forall x\, \psi$ by $\forall x(Ux \rightarrow \psi)$. Finally, let ϕ^* be the sentence $\exists x\, Ux \wedge \phi \mid U$. The class of models of ϕ^* is by definition elementary and thus circuit definable, and any sequence of circuits defining the models of ϕ^* strictly defines the models of ϕ. ∎

The next result shows that elementary definability and strict circuit definability exactly coincide when only monadic predicate symbols are allowed:

THEOREM 9. *If σ is a signature containing only monadic predicate symbols then any class of σ-structures is strictly circuit definable if and only if it is elementary.*

Half of this theorem follows immediately from Theorem 8. We prove the converse via Lemma 10, in which we show that any strictly circuit definable class of structures with only monadic predicates must satisfy extremely severe restrictions which can easily be expressed in a first-order formula.

Suppose σ is a signature with p monadic predicate letters and with no other function or predicate letters. Any structure S appropriate to σ is fully described, up to isomorphism, by specifying the size of each of the 2^p subsets into which the predicates of S partition the universe. That is, we can think of S as a 2^p-tuple $\langle S_{00\cdots0}, S_{00\cdots1}, \ldots, S_{11\cdots1} \rangle$, where each subscript is a member of $\{0, 1\}^p$: $S_{00\cdots0}$ is the number of elements of S of which no predicate is true, $S_{00\cdots1}$ is the number of which only the pth predicate is true, and so forth. The sum of the S_w is $|S|$, the size of the universe of S. We can now perform arithmetic (pointwise) on structures of the same signature. For each w in $\{0, 1\}^p$ define $\mathbf{e}_w$ as the "basis" structure with a single element which satisfies predicates w; that is, $\mathbf{e}_w$ is the tuple

$\langle 0, 0, ..., 1, ..., 0 \rangle$, where the 1 occurs at coordinate w. We say that Σ distinguishes between two structures if exactly one of them is a member of Σ.

LEMMA 10. *Let σ be a signature with p monadic predicate letters and let Σ be a strictly circuit definable class of σ-structures. Then there exists k such that, for all w, Σ distinguishes between S and $S + \mathbf{e}_w$ only if $S_w < k$.*

Proof. Suppose to the contrary that Σ is strictly defined by the sequence $\{C_n\}$ of circuits whose size is bounded by a polynomial π and whose depth is at most d. By Theorem 5 there exists N such that no symmetric circuit of order $n \geqslant N$, size at most $\pi(n)$, and depth at most d can distinguish between q and $q+1$ for any $n/2^{p+1} \leqslant q \leqslant n/2$ (but recall that the circuits in $\{C_n\}$ are not necessarily symmetric). Since we assume the falsehood of the lemma, there must exist some σ-structure S such that for some $w \in \{0, 1\}^p$ both $S_w \geqslant N$ and Σ distinguishes between S and $S + e_w$. Choose S and w such that $|S|$ is as small as possible and S_w is as large as possible; that is, for no S' and w' such that Σ distinguishes between S' and $S' + e_{w'}$ do we have either $|S'| < |S|$ or both $|S'| = |S|$ and $S'_{w'} > S_w$.

We claim that $|S| \leqslant N \cdot 2^p$. For otherwise let u be such that S_u is maximal, thus $S_u > N \cdot 2^p$ and $u \neq w$. Since $|S|$ is minimal Σ does not distinguish between $S - e_u$ and S, nor between $S - e_u$ and $S - e_u + e_w$. Thus it distinguishes between $S - e_u + e_w$ and $S + e_w$, contradicting the choice of w.

Recall that the inputs of a circuit formatted for (σ^+, n) come in n groups, one group for each element of the input structure, with $p+1$ inputs in each group. Each group has a universe input which if false makes irrelevant the values of the other inputs in the group. When the universe input is true the other p inputs specify which predicates are true of the associated element. Take circuit $C_{2|S|}$, and for each $v \neq w$ set the predicate inputs to v in each of S_v groups, also setting the universe input high. Set the predicate inputs of the remaining $|S| + S_w$ inputs to w but do not set the universe inputs. The resulting circuit has $n = |S| + S_w$ inputs. It is symmetric and distinguishes between S_w and $S_w + 1$. But $n/2^{p+1} \leqslant |S|/2^p \leqslant N \leqslant S_w \leqslant n/2$, contradicting the choice of N. ∎

Now suppose Σ is a strictly circuit definable class of σ-structures as in Lemma 10 and let S be any σ-structure with a component S_w that is at least k. By Lemma 10, Σ contains S if and only if it contains all structures S' such that $S'_w \geqslant k$ and $S'_v = S_v$ for all $v \neq w$. Therefore, we can completely describe Σ by listing its structures whose components are all at most k, letting each structure in the list represent the class of all structures obtainable by replacing every occurrence of k by an arbitrary integer greater than or equal to k. But there are only a finite number of such sets to be considered, and each can be described by a first-order formula since we can easily express that at least k elements have a given property. Thus

Σ is an elementary set of structures, and we have completed the proof of Theorem 9.

We can further remark that since Σ is elementary it is strictly circuit definable by Theorem 8, thus proving the converse of Lemma 10. If we now consider the case where σ contains a single monadic predicate (as in Sects. 1 and 2) we obtain a characterization of the strictly circuit definable functions:

COROLLARY 11. *A function K from positive integers to sets of integers such that $K(n) \subseteq \{0, 1, \ldots, n\}$ for all n is strictly circuit definable if and only if there exist k and N such that both of the following conditions hold:*

- *Either $K(n)$ contains $[k, n-k]$ for every n, or $K(n)$ is disjoint from $[k, n-k]$ for every n.*
- *For every $n \geqslant N$ and every $i \in [0, k-1] \cup [n-k+1, n]$ we have $i \in K(n) \Leftrightarrow i \in K(N)$.*

Unfortunately, the correspondence between elementary definability and strict circuit definability holds only when signatures are restricted to monadic predicate symbols. If we allow other symbols, strict circuit definability is strictly more powerful:

THEOREM 12. *Let σ be a signature containing a single binary predicate symbol. There is a strictly circuit definable class of σ-structures that is not elementary.*

Proof. The idea is to construct a class of structures that is not definable by any first-order σ-sentence but which can be defined using auxiliary predicate symbols, which are then given specific interpretations as we convert from formulas to circuits (as in the proof of Lemma 2). The problem is that because the circuits will have universe inputs added, the additional predicates must have interpretations sufficiently robust to survive restriction to arbitrary substructures. For example, the predicate "x is the number of ones in the binary representation of y" can't be used because a substructure may contain y but not x. Instead we use the predicate $\leqslant$, since a total order on the universe remains a total order on any subset of the universe.

Let ε be the binary predicate symbol of σ and write $x \varepsilon y$ instead of εxy. For any σ-structure S define $P^S = \{p: \exists c\, p \varepsilon c\}$ and define C^S as the complement of P^S. Now let Σ be the class of all σ-structures S in which

1. for every subset X of P^S there is a unique $c \in C^S$ such that $X = \{p: p \varepsilon c\}$, and
2. P^S contains an even number of elements.

Intuitively, P^S is a set of points, C^S is the power set of P^S, ε^S is the con-

tainment relation, and Σ contains all structures with an even number of points. Condition (2) can be expressed with a first-order formula: just write that there is a unique empty set and that for every set c and every point p there is a unique set $c \cup \{p\}$. Nevertheless, Σ is not an elementary class.

FACT. *For every first-order sentence ψ in the language of boolean algebras there exists a positive integer n such that if A and B are finite boolean algebras each with at least n atoms, then ψ does not distinguish between A and B. (This fact follows immediately from the known classification of boolean algebras by elementary properties* [12, 5]. *It also can easily be proved directly using Ehrenfeucht games* [4].)

To prove that Σ is not elementary, suppose to the contrary that ϕ is a first-order sentence with a single binary predicate ε that defines Σ. It is easy to translate ϕ into a formula in the language of boolean algebras that asserts on each finite boolean algebra that the number of atoms is even: for example, replace every variable x with a pair (x_1, x_2) of variables, and then replaces each atomic formula $(u_1, u_2)\, \varepsilon(x_1, x_2)$ by a formula saying that u_1 is an atom and $u_1 \leqslant x_2$. A finite boolean algebra satisfies the resulting sentence if and only if it has an even number of atoms, contradicting the fact.

To show that Σ is strictly circuit definable, let σ' be σ enriched with a binary predicate symbol $\prec$. Let ϕ' be a first-order sentence saying that $\prec$ is a linear order, that condition (1) above is satisfied by ε, and that there is a $c \in C$ satisfying the following conditions:

1. c contains the smallest point of P with respect to $\prec$,
2. c does not contain the largest point of P with respect to $\prec$,
3. of any two points of P that are adjacent with respect to $\prec$, c contains one but not the other.

Clearly these conditions are met if and only if P has an even number of elements. Now by Theorem 8, the class of models of ϕ' is strictly circuit definable by a sequence of circuits $\{C_n\}$ formatted for σ'. In each circuit, fix each input labelled $\prec(i, j)$ at 1 if $i < j$ and at 0 otherwise. The resulting sequence of circuits is formatted for σ and strictly defines Σ. This completes the proof of Theorem 12. ∎

The class Σ used in this proof has another application: it constitutes a counterexample to Myer's conjecture [11]. We briefly recapitulate the background for the conjecture here, referring the reader to [11] for a full treatment. A *file* is a random access data structure consisting of *records*, strings of uniform length from a given alphabet, which are indexed by *addresses*, also strings of uniform length from a possibly different alphabet. A *file machine* is a Turing machine, possibly nondeterministic or alternating, with a write-only *address* tape, a read-only *record* tape, and any

number of read/write worktapes. The input to a file machine is a file, which the machine accesses as follows: whenever the machine writes an input address on the address tape the corresponding record of the input file is placed (in one step) on the record tape. The time and space requirements of file machines are measured in terms of the number of records in the input file (thus a file machine can accept a language in sublinear time). We consider only file acceptors here; that is, every file machine either accepts or rejects its input file, and the language accepted by the file machine is the class of files that it accepts.

A file can be used to represent a first-order structure as follows: We let q be the maximum number of argument places of any relation or function symbol of the language. Then an "address" of the file is a list of (encodings of) elements $x_1,..., x_q$ from the universe of the structure, and the associated record contains the truth values of the predicates $R(x_1,..., x_i)$ and encodings of the values of the functions $f(x_1,..., x_j)$. The class of finite models of any sentence is thus a file language, called an *elementary* language; obviously a file language is elementary if and only if it represents an elementary class of structures. A language is *invariant* if whenever it contains one of two files representing isomorphic structures, it also contains the other.

Myers asserts in [11] that the elementary languages definable by quantifier-free formulas are exactly the invariant deterministic log-time file languages and that the elementary languages definable by existential formulas are exactly the invariant nondeterministic log-time file languages. He conjectures that the elementary languages are exactly the invariant bounded-alternation log-time languages. Consider, however, the class Σ of structures defined in the proof of Theorem 12. We have shown that this class (and therefore also the corresponding file language) is not elementary. But Σ can be recognized by a file machine, as follows: Obviously, a file machine can recognize any elementary file language by simply checking the truth or falsity of the defining sentence ϕ in the structure represented by the input file. (The machine uses universal and existential branching to instantiate all quantifiers, and checks the truth-value of the resulting formula by querying the file for the values of the atoms.) Consider a file machine that recognizes the file language consisting of the models of ϕ' above, and modify the machine so that instead of querying the input file for the values of the atoms $\prec(i, j)$ it simply compares i and j. The resulting file machine recognizes the class Σ. This construction is purely analogous with the circuit construction above; in both cases we simply delete $\prec$ from the signature by computing its values directly.

The remainder of this section is something of an aside, but may be independently interesting.

In the definition of circuit definability it is important that any circuit family include a circuit of order n for each n, otherwise there are structures

upon which the family cannot operate. That restriction could obviously be relaxed with strict circuit definability, since in order to check a structure of size n it suffices that there be a circuit in the family with order n or more; that is, an infinite sequence of circuits suffices. We have assumed in the proof of Lemma 10 that every family has a circuit for each n. The proof needs only minor modifications if we assume instead that for each family there exists some polynomial π such that whenever the family contains a circuit of order i it has another of order between i and $\pi(i)$—that is, that there are at most "polynomially large" gaps in the sequence. This seems a reasonable restriction; a customer wishing to check a structure could be asked to buy a circuit of somewhat greater order, but justifiably would not wish to be forced to switch to a super-polynomially larger model.

If on the other hand we allow circuit families to have arbitrarily large gaps, then Lemma 10 fails. For example, let K be defined as follows:

$$K(n) = \begin{cases} \varnothing & \text{if } \log^* n \text{ is even;} \\ \{0, 1, \ldots, n\}, & \text{if } \log^* n \text{ is odd.} \end{cases}$$

where $\log^* n$ is the number of times base 2 logarithm must be taken to reduce n to 1. We can construct an infinite sequence of circuits $\{C_n\}$ each of which strictly recognizes K. We do so by constructing circuits of order 2, 2^2, 2^{2^2}, and so forth. The action of C_n when $\log^* n$ is odd is as follows: C_n ignores its predicate inputs (since they are irrelevant) and simply counts the number of 1's among its universe inputs. If more than $\log n$ of these are high C_n simply returns 1. If not, but more than $\log \log n$ of them are high, C_n returns 0. If not, but more than $\log \log \log n$ are high, C_n returns 1, and so forth. (If $\log^* n$ is even we simply exchange 0 and 1 as outputs of C_n.) Only $\log^* n$ counting subcircuits are needed, and by Theorem 1 each can be constructed in polynomial size and constant depth overall. The point is that $\log^* i$ is the same for all i between $\log n$ and n for these carefully chosen n, and thus C_n needs only a very rough count. Contrast this situation with that of standard circuit definability, where the classes we have shown not circuit definable are not definable even by an infinite sequence of circuits.

The function $K(n) = \{\log^* n\}$ can be handled in a similar manner. We can show, however, that allowing arbitrary gaps between the circuits does not give the power of our original notion of circuit definability. For example, no function of the form $K(n) = \{\lceil \log \log \cdots \log n \rceil\}$ can be defined in this new manner for any fixed number of logs.

4. Circuit Definability for First-Order Logic

We saw in the last section that every first-order definable class of structures is strictly circuit definable, but that the converse does not hold. We desire a circuit model that exactly captures the definitional power of first-order logic. In this final section we place a restriction on our circuits that gives precisely that result.

In Section 1 we explained how to build a circuit representing a sentence ϕ over universe $U_n = \{0, 1,..., n-1\}$: each wire of the circuit is labelled with a (possibly instantiated) subformula of the sentence and the input wires are labelled with variable-free atomic sentences. We now repeat that construction adding a second label to each wire. Specifically, suppose ϕ and n are given and let Γ_n be the alphabet $\{0, 1,..., n-1, \$_L, \$_R\}$. Each wire of the circuit under construction will be labelled with a formula in the language of ϕ and a string over Γ_n. (The formulas are exactly as in Sect. 1 but are repeated here for clarity.) The single output wire is labelled with ϕ and with the empty string. Now recursively: when a wire is labelled with a formula ψ and a string x our action depends on the form of ψ. If $\psi = \psi_1 \wedge \psi_2$ we let the wire be the output of an $\wedge$-gate whose two input wires are labelled with formulas ψ_1, ψ_2 respectively and with strings $x\$_L$, $x\$_R$, respectively. The case $\psi = \psi_1 \vee \psi_2$ is handled analogously, and for $\psi = \neg\psi_1$ we use an inverter whose single input wire has labels ψ_1 and $x\$_L$. If $\psi = \forall x\psi_1(x)$ we use an $\wedge$-gate with n inputs labelled with formulas $\psi(0)$, $\psi(1)$,..., $\psi(n-1)$ respectively and with strings $x0, x1,..., x(n-1)$, respectively; the case $\psi = \exists x\psi_1(x)$ is analogous. Note that the formulas labelling the inputs are such that the circuit is symmetric and formatted for (σ, n).

Obviously, every wire of the circuit so constructed has a distinct string label (although the formula labels need not be distinct); hence we identify wires with their string labels. If x is such a wire let ϕ_x be its formula label and let

$$\text{range}(x) = \{i \mid 0 \leqslant i < n \text{ and } i \text{ appears in } x\}.$$

For any wire x it is obvious that the signature of ϕ_x is contained in $\sigma \cup \text{range}(x)$. Moreover, C is a tree in the sense that the output of any gate (except for the final output) is the input to exactly one gate; we can speak of the *children* of each wire, where the leaves are the inputs to the circuit. Finally, note that the length of the string labels of C is bounded independently of n—the length of the longest labels is exactly the logical depth of the original formula.

Let f be a permutation of $\{0, 1,..., n-1\}$. Extend f to a permutation (also called f) of Γ_n^* by setting $f(\$_L) = \$_L$, $f(\$_R) = \$_R$, and $f(a_1 a_2 \cdots a_k) = b_1 b_2 \cdots b_k$, where $f(a_i) = b_i$ for each i. The following statements are obviously true for all $x, y \in \Gamma_n^*$ and all permutations f:

1. x is a wire of C if and only if $f(x)$ is a wire of C;

2. if x is a wire of C then x and $f(x)$ are outputs from gates of the same type ($\vee$-gate, $\wedge$-gate, inverter) or are both input wires;

3. if y is a child of x then $f(y)$ is a child of $f(x)$;

4. if x is an input wire whose formula label is an atomic formula Py then the formula label of $f(x)$ is $Pf(y)$.

We can now formalize the highly symmetric qualities of the circuits we have been building: Let C be any symmetric circuit formatted for (σ, n) as in Section 3. Such a circuit is *regular* if each wire is assigned a distinct string over Γ_n, if C forms a tree as above, and if every permutation f of U_n satisfies condition 1–4 above.

LEMMA 13. *Let C be a regular circuit formatted for (σ, n) and suppose that each string label of C has length less than $n-1$. Then*

(i) *The range of the root of C is empty.*

(ii) *If y is a child of x then $range(x) \subseteq range(y)$.*

(iii) *If x has children, then the intersection of the ranges of the children is contained in $range(x)$.*

(iv) *If x is a leaf, then each i that appears as an argument in the atomic sentence assigned to x is contained in $range(x)$.*

Proof. (i) If $\text{range}(r) \neq \varnothing$ then there is a permutation f such that $f(r) \neq r$. But then $f(r)$ must be another root of C.

(ii) If $\text{range}(x) \not\subseteq \text{range}(y)$ we can pick a permutation f such that $f(y) = y$ but $\text{range}(f(x)) \neq \text{range}(x)$. Then y is a child of both x and $f(x)$.

(iii) Suppose to the contrary that b belongs to $\text{range}(y)$ for each y but $b \notin \text{range}(x)$. Pick an arbitrary child y of x and pick a permutation f such that $f(x) = x$ but $f(b) \notin \text{range}(y)$. Then $f(y)$ is a child of x whose range does not contain b, contradicting the choice of b.

(iv) If not, pick f such that $f(x) = x$ but $f(y) \neq y$. Then two atomic formulas Py and $Pf(y)$ are assigned to the same input wire, which is impossible. ∎

This final form of circuit enjoys all the properties that we need to capture the expressive power of first-order logic. We begin the proof of this result by showing how to construct a formula given such a circuit.

THEOREM 14. *Let C be a regular circuit formatted for (σ, n). Then there is a first-order σ-sentence ϕ which holds on exactly those σ-structures with universe U_n that are accepted by C. Moreover, the quantifier depth of ϕ is at most equal to the length of the longest string labelling a wire of C.*

Proof. By induction we construct for each wire x of C a formula ϕ_x whose signature is contained in $\sigma \cup \text{range}(x)$.

If x is an input assigned the formula Py we let $\phi_x = Py$. If x is the output of an inverter whose input wire is y we let $\phi_x = \neg\phi_y$.

Now suppose x is the output of an $\vee$-gate. For every child y of x and every b in $\text{range}(y) - \text{range}(x)$, replace b by the variable v_b in ϕ_y; call the resulting formula ϕ_y^*. Two children y and z of x will be called *twins* if there is a permutation of U_n that is the identity on $\text{range}(x)$ and takes y to z. Twinning is an equivalence relation, and the number of equivalence classes is bounded independently of n. (*Proof.* Each class is characterized by a string label consisting of $\$_L$, $\$_R$, members of the range of x, and "wild cards" representing constants that take on all values outside the range of x. The number of these symbols is independent of n since the range of x is limited by the length of the string label of x.) The disjunction of all formulas ϕ_z, where z is a twin of y, is equivalent to the formula $\psi_y = \exists\vec{v}(\phi_y^* \wedge P)$, where $\vec{v}$ is the string of free variables in ϕ_y^* and P states that none of these variables is equal to any of the constants in $\text{range}(x)$. Take a collection of children of x that has exactly one representative from each twinning class and set ϕ_x equal to the disjunction of all the ψ_y with y in the collection. The case of $\wedge$-gates is handled similarly.

It is clear that for each wire x of C the subcircuit whose output is x accepts a σ-structure S with universe U_n if and only if S satisfies ϕ_x. In particular, C accepts S if and only if S satisfies the formula ϕ assigned to the output of C. The quantifier depth of ϕ is bounded since quantifiers can be added only as the range of a wire is reduced to that of its father, and the largest range of any wire is equal to the maximum label length. ∎

Call a sequence of regular circuits *label-bounded* if there exists k such that no wire of any circuit has string label exceeding k. The main result of this section is then

THEOREM 15. *A class of σ-structures is elementary if and only if it is strictly definable by a label-bounded sequence of regular circuits. That is, given a class Σ of σ-structures the following are equivalent:*

- *There exists a first-order σ-formula ψ such that Σ consists of exactly the models of ψ,*
- *There exists a label-bounded sequence $\{C_n\}$ of regular circuits formatted for $\sigma^+ = \sigma \cup \{U\}$ such that, for each n, circuit C_n accepts a σ^+-structure S if and only if $S|U^S \in \Sigma$.*

Proof. Half of this theorem was established as Theorem 8 since (as noted above) all circuits constructed in Section 3 were in fact regular. We now turn to the proof of the other half.

Let $\{C_n\}$ be a sequence of regular circuits formatted for σ^+ that strictly defines the set Σ of σ-structures, and suppose that no wire in the entire sequence has a string label longer than k. By Theorem 14 there corresponds to each circuit C_n (with n sufficiently large) a σ^+-sentence ϕ_n of quantifier depth at most k that holds on exactly those σ^+-structures accepted by C_n. If we do not distinguish between logically equivalent formulas there are only a finite number of such formulas (since the possible choices for ϕ_n do not depend on n), and thus there exists a sentence ϕ that occurs infinitely often as ϕ_n.

Now consider an arbitrary σ-structure S with universe U_m. Let S^+ be the σ^+-structure with universe $U_m \cup \{*\}$, with U_m as the interpretation of U, and with atomic σ-sentences true exactly when they are true in S—in particular, any such sentence with $*$ in any argument position is false in S^+. The desired sentence ψ will express the fact that S^+ satisfies ϕ. We need to show how to construct ψ and why such a ψ satisfies the requirements of the theorem.

The latter question is easily handled: let S and S^+ be as above, and choose any $n > m$ such that $\phi_n = \phi$. Let T be the structure with universe U_n obtained from S^+ by replacing $*$ by elements $m, m+1, \ldots, n-1$ that are indistinguishable from $*$. Then T satisfies ϕ if and only if S^+ satisfies ϕ, if and only if S satisfies ψ.

Finally, we show how to express that S^+ satisfies ϕ. The idea is to code the universe of S^+ with ordered pairs, where (i, i) encodes i and (i, j) encodes $*$ whenever $i \neq j$. Without loss of generality ϕ is prenex and its matrix is in U-developed disjunctive normal form; that is, the matrix is DNF and each disjunct contains as a conjunct either Uv or $\neg Uv$ for each variable v in ϕ. Replace every variable v with a pair of variables v, v', as follows: replace $\exists v$ with $\exists v, v'$; replace $\forall v$ with $\forall v, v'$; replace Uv with $v = v'$; and for each atomic formula that contains v in some argument position leave it untouched when it occurs in a disjunct containing Uv and replace it with FALSE when it occurs in a disjunct containing $\neg Uv$. Then the following lemma (whose inductive proof we omit) gives the desired result:

LEMMA 16. *Let ϕ be any prenex σ^+-formula whose matrix is in U-developed DNF and let ψ be constructed from ϕ as above.* (Note that we are discussing formulas rather than sentences: ψ will have twice as many free variables as ϕ.) *Let f assign elements of $U_m \cup \{*\}$ to the free variables of ϕ and let g assign elements of U_m to the free variables of ψ. Call f and g* corresponding *if for every free variable v of ϕ,*

1. $f(v) = * \leftrightarrow g(v) = g(v')$, *and*
2. *if* $f(v) \neq *$ *then* $f(v) = g(v) \neq g(v')$.

Then S^+ satisfies ϕ under f if and only if S satisfies ψ under g. In particular, when ϕ and hence ψ are sentences, ψ expresses the sentence "S^+ satisfies ϕ."

This completes the proof of Theorem 15. ∎

ACKNOWLEDGMENTS

We wish to thank Andreas Blass and Harry Lewis for very useful comments and helpful discussion. We also than Dale Myers for his patient correspondence, including his criticisms of earlier versions of this paper.

REFERENCES

1. CHANDRA, A. K., FORTUNE, S., AND LIPTON, R. (1983), Unbounded fan-in circuits and associative functions, *in* "15th ACM Sympos. Theory of Computing," pp. 52–60.
2. CHANDRA, A. K., STOCKMEYER, L. J., AND VISHKIN, U. (1982), Constant depth reducibility, *in* "23rd IEEE Sympos. Found. of Computer Science," pp. 1–13.
3. DENENBERG, L., GUREVICH, Y., AND SHELAH, S. (1983), "Cardinalities Definable by Constant Depth Polynomial Size Circuits," Technical Report TR-26-83, Harvard University.
4. EHRENFEUCHT, A. (1960), An application of games to the completeness problem for formalized theories, *Fund. Math.* **49**, No. 2, 128–141.
5. ERSHOV, Y. L. (1964), Decidability of the elementary theory of distributive lattice with relative complements and the theory of filters, *Algebra and Logic* **3**, No. 3, 17–38.
6. FAGIN, R., KLAWE, M., PIPPENGER, N. J., AND STOCKMEYER, L. (1985), Bounded depth polynomial size circuits for symmetric functions, *Theoret. Comput. Sci.* **36**, No. 2/3, 239–250.
7. FURST, M., SAXE, J., AND SIPSER, M. (1984), Parity, circuits, and the polynomial time hierarchy, *Math. Systems Theory* **17**, No. 1, 13–27.
8. GUREVICH, Y., AND LEWIS, H. R. (1984), A logic for constant-depth circuits, *Inform. Control* **61**, 65–74.
9. HARDY, G. H., AND WRIGHT, E. M. (1979), "An Introduction to the Theory of Numbers," Oxford Univ. Press, London/New York.
10. LYNDON, R. C. (1966), "Notes on Logic," Van Nostrand, Princeton, N. J.
11. MYERS, D. (1983), The random access hierarchy, *in* "15th ACM Sympos. Theory of Computing," pp. 355–364.
12. TARSKI, A. (1949), Arithmetical classes and types of boolean algebras, *Bull. Amer. Math. Soc.* **55** No. 1, 63.
13. YAO, A. C.-C. (1985), Separating the polynomial-time hierarchy by oracles, *in* "26th IEEE Sympos. Foundations of Computer Science," pp. 1–10.