The Monadic Quantifier Alternation Hierarchy over Grids and Pictures

Nicole Schweikardt
Institut für Informatik, Universität Mainz, D-55099 Mainz
email: nisch@informatik.uni-mainz.de

July 30, 1997

Abstract

The subject of this paper is monadic second-order logic over two-dimensional grids. We give a game-theoretical proof for the strictness of the monadic second-order quantifier alternation hierarchy over grids. Additionally, we can show that monadic second-order logic over coloured grids is expressive enough to define complete problems for each level of the polynomial time hierarchy.

1 Introduction

Grids are "finite graphs with two edge relations" whose elements are arranged as the elements of a matrix. The two edge relations are successor relations connecting each element with the proper element in the following row, respectively column. *Pictures* are coloured grids, i.e. grids with some additional unary relations, and can be viewed as "two-dimensional words".

Monadic second-order logic is the fragment of second-order logic in which second-order quantifiers may only range over *monadic* predicates, i.e. over sets. Σ_k^1 is the set of all second-order formulas having a prefix of k alternating blocks of second-order quantifiers (where the second-order predicates may range over relations of arbitrarily high arity), starting with an existential block, and followed by a first-order formula.

In section 2 we will provide the notation necessary to formulate the main results which are presented in section 3 and explained in the subsequent parts of the paper. Detailed proofs can be found in [SchN97].

Fagin and Stockmeyer ([Fag74, Sto77]) showed that the kth level of the polynomial time hierarchy is exactly the class of problems definable by a Σ_k^1-formula. Thus the question whether for growing k, *monadic* Σ_k^1-formulas (which in the following we will also denote by Σ_k) allow to describe more and more problems, can be viewed as the monadic analogue to the question, whether the polynomial time hierarchy (in short: PH) is strict.

Ajtai, Fagin and Stockmeyer ([AFS97]) obtained the result that for each $k \geqslant 1$ there is a problem which is both, definable by a *monadic* Σ_k^1-formula and complete for the kth level of the PH. Their result is unpublished yet. In section 4 of the present paper we give a proof of this fact which was found by Oliver Matz, Wolfgang Thomas, and the author. More precisely, we show that *monadic* Σ_k^1 over *coloured grids* is expressive enough to define complete problems for the kth level of the PH.

From automata theory ([Bue60, TW68]) we know that over *words* and *trees* all monadic second-order properties are in fact Σ_1-properties, i.e. the monadic second-order quantifier alternation hierarchy (in short: the monadic hierarchy) over words and trees collapses within its first level.

Following a technique of [MT97], in sections 5 and 6 we will show that the monadic hierarchy over grids is *strict* (– and thus answer a question posed by Matz and Thomas). In [MT97], Matz and Thomas proved that the monadic quantifier alternation hierarchy is

strict over graphs and infinite over grids.[1] To establish this, they investigated sets of grids in which for each height the set contains exactly one grid of that height, i.e. sets of grids where the width of the grids is a function of their height. Using concepts from automata theory, they showed that functions which are Σ_k-definable (in the sense that the corresponding sets of grids are Σ_k-definable), are at most k-fold exponential. In section 5 we present a different method, namely we analyse a special Ehrenfeucht game, which leads to the same upper bound on the growth rate of Σ_k-definable functions.
Finally in section 6 we show that some k-fold exponential function is definable by a Σ_k- and by a Π_k-formula, rather than just by a Σ_{2k+3}-formula as presented in [MT97].

Thus monadic second-order logic induces a strict hierarchy of picture-problems such that each level of this hierarchy is a subclass of the same level of the polynomial time hierarchy and contains a problem which is complete for that level of the polynomial time hierarchy. Let us mention that the problems witnessing the strictness of the monadic hierarchy over grids are all within NP, i.e. within the first level of the PH.

I thank Oliver Matz and Wolfgang Thomas for their kind permission to include Theorem 6, Thomas Schwentick for helpful remarks on the Ehrenfeucht game presented in section 5, and Clemens Lautemann for the help and advice he gave me during the preparation of this paper.

2 Notation

$\mathbb{N} = \{1, 2, \ldots\}$ is the set of all natural numbers, not including 0. Indices m, n, i, j etc. always refer to natural numbers. With $[m, n] := \{1, \ldots, m\} \times \{1, \ldots, n\}$ we denote the set of all tuples (i, j) for $1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n$.

2.1 Grids and Pictures

Definition 1 *Let* $m, n \geqslant 1$.
The grid $\underline{[m,n]}$ *of height* m *and width* n *is the structure* $\underline{[m,n]} = ([m,n], S_1^{m,n}, S_2^{m,n})$, *where* $S_1^{m,n} = \{((i,j),(i+1,j)) : 1 \leqslant i < m,\ 1 \leqslant j \leqslant n\}$ *is the row-successor relation, and* $S_2^{m,n} = \{((i,j),(i,j+1)) : 1 \leqslant i \leqslant m,\ 1 \leqslant j < n\}$ *is the column-successor relation.*

The elements in $[m, n]$ are called *vertices*. The expressions *row, column, top, bottom* etc. are understood in the usual way, e.g. the leftmost column contains exactly the vertices $(i, 1)$ for $1 \leqslant i \leqslant m$, and the top row consists of all vertices $(1, j)$ for $1 \leqslant j \leqslant n$.

Pictures are coloured grids in which each vertex is coloured by exactly one out of finitely many colours.

Definition 2 *Let* $t \geqslant 0$.
A t-bit picture $\underline{P}$ *is a structure* $\underline{P} = (\underline{[m,n]}, X_1^P, \ldots, X_t^P)$, *where* $\underline{[m,n]}$ *is a grid of height* m *and width* n, *and* $X_i^P \subseteq [m,n]$ *is a set of vertices for every* $i \leqslant t$.

The colour of a vertex $v \in [m,n]$ is given by the sets $X_1^P, ..., X_t^P$ as follows: For each $i \leqslant t$ let $b_i(v) = 1$ in case $v \in X_i^P$, and $b_i(v) = 0$ in case $v \notin X_i^P$. Then v is coloured by the string $b_1(v) \cdots b_t(v) \in \{0,1\}^t$, i.e. the ith set X_i^P determines the ith bit $b_i(v)$ of the colour-string $b_1(v) \cdots b_t(v)$ of each vertex v. Hence a t-bit picture is a coloured grid in which every vertex is coloured by one out of 2^t possible colours. Note that (uncoloured) grids are 0-bit pictures.

2.2 Monadic Second-Order Logic over Grids

Monadic second-order formulas over grids are built of two binary relation symbols S_1, S_2 (for row- and column-successor), the equaltiy symbol $=$, the logical connectives $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$, individual variables (i.e. first-order variables) $x, y, x_1, x_2, \ldots$, set variables (i.e. monadic second-order variables) $X, Y, X_1, X_2, \ldots$ and quantifiers $\exists, \forall$, which may quantify both individual and set variables.

[1] Note that over uncoloured grids, the infinity of the monadic hierarchy does not trivially imply the strictness.

With MSO we denote the set of all monadic second-order formulas over grids. FO-formulas are MSO-formulas in which no set-quantifier occurs. With free(φ) we denote the set of all individual and set variables occuring free in $\varphi \in$ MSO.
Let $\varphi \in$ MSO with free$(\varphi) = \{X_1, ..., X_t, x_1, ..., x_s\}$, $\underline{P} = (\underline{[m,n]}, X_1^P, ..., X_t^P)$ a t-bit picture and $x_1^P, ..., x_s^P$ vertices in $\underline{P}$. We write $(\underline{P}, x_1^P, ..., x_s^P) \models \varphi$ if φ is satisfied when interpreting the free occurrences of the variables $X_1, ..., X_t, x_1, ..., x_s$ with the values $X_1^P, ..., X_t^P, x_1^P, ..., x_s^P$, and interpreting S_1 and S_2 as the row- and the column-successor relation of the grid, respectively.

Example *left*$(x) := \forall y \neg S_2 yx$ is a FO-formula asserting that vertex x is in the leftmost column of a grid. Let *right*(x), *top*(x), *bottom*(x), *topleft*(x) etc. be similar FO-formulas asserting that vertex x is in the rightmost column, top row, bottom row, on the top left position, respectively.

Definition 3 *Let $t \geqslant 0$, $\varphi \in MSO$.*
If free$(\varphi) \subseteq \{X_1, \ldots, X_t\}$, with $MOD_{t\text{-pic}}(\varphi) = \{\underline{P} : \underline{P} \models \varphi$ and $\underline{P}$ is a t-bit picture$\}$ we denote the class of all t-bit pictures satisfying φ.
If free$(\varphi) \nsubseteq \{X_1, \ldots, X_t\}$, we define $MOD_{t\text{-pic}}(\varphi) = \emptyset$.
In case $t = 0$ we write $MOD_{\text{grid}}(\varphi) = MOD_{0\text{-pic}}(\varphi)$ for the class of all (uncoloured) grids satisfying φ.

A class $\mathcal{C}$ of t-bit pictures is called MSO-definable if there is a MSO-formula φ such that $\mathcal{C} = \text{MOD}_{t\text{-pic}}(\varphi)$.

Definition 4 *(The Monadic Quantifier Alternation Hierarchy)*

1. *We define the following fragments of MSO for every $k \geqslant 1$:*
   $\Sigma_0 = FO$, $\Sigma_k = \{\exists Y_1, ..., Y_r \neg\varphi : \varphi \in \Sigma_{k-1}, r \geqslant 0, Y_1, ..., Y_r$ *are set variables*$\}$
   $\Pi_k = \{\neg\varphi : \varphi \in \Sigma_k\}$
   $B(\Sigma_k)$ = the set of all boolean combinations of Σ_k-formulas.
   Thus Π_k is the set of all negated Σ_k-formulas, and Σ_k contains all MSO-formulas having a prefix of k alternating blocks of set quantifiers, starting with an existential block, and followed by a first-order formula.

2. *The kth level $\underline{\Sigma_k}$ of the monadic quantifier alternation hierarchy over* grids *is the class of all sets of (uncoloured) grids definable by a Σ_k-sentence, i.e.*
   $\underline{\Sigma_k} = \{MOD_{\text{grid}}(\varphi) : \varphi \in \Sigma_k$ *and* free$(\varphi) = \emptyset\}$.
   The classes $\underline{\Pi_k}$ and $\underline{B(\Sigma_k)}$ are defined analogously.
   $\underline{\Delta_k} = \underline{\Sigma_k} \cap \underline{\Pi_k}$ denotes the class of all sets of grids definable both by a Σ_k-formula and by a Π_k-formula.
   In the same way, for every $t \geqslant 0$, $\underline{\Sigma_k}[t\text{-pic}] = \{MOD_{t\text{-pic}}(\varphi) : \varphi \in \Sigma_k\}$ denotes the kth level of the monadic quantifier alternation hierarchy over t-bit pictures.
   The classes $\underline{\Pi_k}[t\text{-pic}]$, $\underline{B(\Sigma_k)}[t\text{-pic}]$ and $\underline{\Delta_k}[t\text{-pic}]$ are defined analogously.

2.3 Definability and Growth Rate of Functions

A function $f : \mathbb{N} \longrightarrow \mathbb{N}$ is called Σ_k-definable if there is a sentence $\varphi \in \Sigma_k$ such that $\text{MOD}_{\text{grid}}(\varphi) = \{\underline{[m, f(m)]} : m \geqslant 1\}$, i.e. for each $m \geqslant 1$ there is exactly one grid of height m, namely the grid of width $f(m)$, which satisfies φ.

Definition 5 *Let $s_0(m) = m$, $s_k(m) = 2^{s_{k-1}(m)}$ for every $k, m \geqslant 1$.*
A function $f : \mathbb{N} \longrightarrow \mathbb{N}$ is called
- at most k-fold exponential *if $f(m) \leqslant s_k(\mathcal{O}(m))$, i.e. there is a constant $c > 0$ such that $f(m) \leqslant s_k(cm)$ for all $m \geqslant 1$.*
- k-fold exponential *if $f(m) \leqslant s_k(\mathcal{O}(m))$ and $f(m) \nleqslant s_{k-1}(\mathcal{O}(m))$.*

3 Main Results

We can now state our results formally.
With $\underline{\Sigma_k^P}$ we denote the kth level of the polynomial time hierarchy.
In section 4 we will show the following

Theorem 6 *For each $k \geqslant 1$ there exists a set of 3-bit pictures which is both complete for the kth level of the polynomial time hierarchy (with respect to polynomial time reductions) and definable by a Σ_k-formula, i.e. $\underline{\Sigma_k}$ [3-pic] contains a $\underline{\Sigma_k^P}$-complete problem.*

In particular, monadic second-order logic over 3-bit pictures allows to define complete problems for each level of the polynomial time hierarchy.

Our second result is the strictness of the monadic hierarchy over grids and over t-bit pictures. In order to prove it, we will follow the strategy of Matz and Thomas in [MT97]. First we will reprove their following result, replacing their automata-theoretic proof by an argument based on Ehrenfeucht games.

Theorem 7 *(Matz, Thomas)*
Σ_k-definable functions are at most k-fold exponential.

Matz and Thomas then showed that some k-fold exponential function is Σ_{2k+3}-definable, concluding that the monadic hierarchy over grids is infinite. In section 6 we will improve this to

Theorem 8 *Let $f_1(m) = 2^m$, $f_{k+1}(m) = f_k(m)2^{f_k(m)}$ for every $k, m \geqslant 1$.*
For each $k \geqslant 1$, the k-fold exponential function f_k is Σ_k-definable and Π_k-definable.

Thus $\underline{\Sigma_k} \not\ni \{\underline{[m, f_{k+1}(m)]} : m \geqslant 1\} \in \underline{\Delta_{k+1}}$, and we directly obtain

Theorem 9 *$\underline{\Sigma_k} \subsetneq \underline{\Delta_{k+1}}$ and $\underline{\Sigma_k}$ [t-pic] $\subsetneq \underline{\Delta_{k+1}}$ [t-pic] for all $k, t \geqslant 1$.*
In particular, the monadic second-order quantifier alternation hierarchy over grids and t-bit pictures is strict, i.e. $\underline{\Sigma_k} \subsetneq \underline{\Sigma_{k+1}}$ and $\underline{\Sigma_k}$ [t-pic] $\subsetneq \underline{\Sigma_{k+1}}$ [t-pic] for all $k, t \geqslant 1$.

4 Monadic Second-Order Logic and the Polynomial Time Hierarchy

To prove Theorem 6, we encode quantified boolean formulas of a certain form into pictures. The main idea is to represent a CNF-formula φ by a picture which has a row for each variable and a column for each clause, such that the vertex in row i and column j has colour P (resp. N) if variable x_i occurs unnegated (resp. negated) in the jth clause of the formula. Then, an assignment to the variables in φ corresponds to a colouring of the rows of the corresponding picture; and satisfyability of φ can be expressed by a Σ_1-formula.

Let BF denote the set of all Boolean formulas over the set $V = \{x_{i,j} : i, j \geqslant 1\}$. For $i \geqslant 1$, an assignment A_i to the variables in $V_i = \{x_{i,j} : j \geqslant 1\}$ is a mapping $A_i : V_i \longrightarrow \{0, 1\}$. Let BF_k denote the set of all Boolean formulas over $V_1 \cup \cdots \cup V_k$. For a formula $\varphi \in \text{BF}_k$ we write $\exists A_1 \forall A_2 ... Q_k A_k(\varphi = 1)$ as an abbreviation for "there exists an assignment A_1 to V_1 such that for all assignments A_2 to V_2 ... such that under the assignments $A_1, ..., A_k$ the formula φ becomes true".
φ is in conjunctive (resp. disjunctive) normal form if $\varphi = \bigwedge_\alpha(\bigvee_\beta l_{\alpha,\beta})$ (resp. $\varphi = \bigvee_\alpha(\bigwedge_\beta l_{\alpha,\beta})$), where $l_{\alpha,\beta} \in \{x_{i,j}, \neg x_{i,j} : i, j \geqslant 1\}$. Let CNF (resp. DNF) denote the set of all Boolean formulas over V, which are in conjunctive (resp. disjunctive) normal form.

Definition 10 *Let $k \geqslant 1$.*
$CNF_{k,\exists} = \{\varphi \in BF_k : \varphi$ is in CNF and $\exists A_1 \forall A_2 ... Q_k A_k(\varphi = 1)\ \}$

$CNF_{k,\forall} = \{\varphi \in BF_k : \varphi$ *is in CNF and* $\forall A_1 \exists A_2 ... Q'_k A_k(\varphi = 1)\ \}$.
$DNF_{k,\exists}$ *and* $DNF_{k,\forall}$ *are defined analogously.*

From Stockmeyer ([Sto77]) it follows that for odd k, $\mathrm{CNF}_{k,\exists}$ is (logspace) complete in $\underline{\Sigma_k^P}$; and for even k, $\mathrm{DNF}_{k,\exists}$ is (logspace) complete in $\underline{\Sigma_k^P}$.
The following lemma shows how to translate CNF-formulas into 3-bit pictures.

Lemma 11 *There is a mapping* $\mathcal{P} : CNF \longrightarrow \{$*3-bit pictures*$\}$ *such that for all odd* k *there is a* Σ_k*-formula* Φ_k *and for all even* k *there is a* Π_k*-formula* Ψ_k *such that*
$\varphi \in CNF_{k,\exists} \iff \mathcal{P}(\varphi) \models \Phi_k$ *(if* k *is odd),*
$\varphi \in CNF_{k,\forall} \iff \mathcal{P}(\varphi) \models \Psi_k$ *(if* k *is even).*
This mapping can be computed by a deterministic polynomial time Turing machine.

Proof Let $\varphi \in$ CNF. We construct $\mathcal{P}(\varphi)$. Let k' be minimal such that $\varphi \in \mathrm{BF}_{k'}$. For $i \leqslant k'$ let W_i denote the set of all V_i-variables in φ. W.l.o.g. $W_i = \{x_{i,1}, ..., x_{i,s_i}\}$ for some $s_i \geqslant 1$. To write down things more succinctly, for $i \leqslant k'$, $r \leqslant s_i$, $\alpha := s_1 + \cdots + s_{i-1} + r$ we define $y_\alpha := x_{i,r}$.

Let $m := s_1 + \cdots + s_{k'}$ be the number of variables in φ, and $\varphi := \bigwedge_{j=1}^{n} C_j$, where the C_j are disjunctions of unnegated and negated variables.
The picture $\mathcal{P}(\varphi)$ has a row for each variable and a column for each clause C_j in φ, thus we choose $\mathcal{P}(\varphi) := (\underline{[m,n]}, P, N, B)$.
The set $B := \{(\alpha, j) : j \leqslant n,\ y_\alpha \in W_i$ for an odd $i \leqslant k'\}$ indicates that the top s_1 rows correspond to the W_1-variables, the following s_2 rows correspond to the W_2-variables, etc. The sets $P := \{(\alpha, j) : j \leqslant n,\ y_\alpha$ occurs unnegated in $C_j\}$ and $N := \{(\alpha, j) : j \leqslant n,\ y_\alpha$ occurs negated in $C_j\}$ encode the clauses $C_1, ..., C_n$.
Clearly $\mathcal{P}(\varphi)$ can be computed by a deterministic polynomial time Turing machine.

In the following, we construct the formulas Φ_k and Ψ_k (for odd, resp. even $k \geqslant 1$). A set $X \subseteq [m,n]$ containing either none or all vertices of a row encodes an assignment A to the variables in φ, and vice versa (by $A(y_\alpha) = 1$ iff row α belongs to X). In this notion, $assign(X) := \forall x, y(S_2xy \to (Xx \leftrightarrow Xy))$ is an FO-formula asserting that X encodes an assignment. Scanning $\mathcal{P}(\varphi)$ from the top to the bottom and watching the set B, we can construct an FO-formula $blocks(B_1, ..., B_k)$ (see the appendix) asserting that for all $i \leqslant k$ the set B_i consists of all rows α such that y_α is a W_i-variable. Under an assignment $A_1, ..., A_k$, φ becomes true if in each clause C_j at least one literal is true, i.e. there is a variable y_{α_j} which occurs unnegated in C_j and is assigned to 1, or occurs negated in C_j and is assigned to 0. Let α_j be the minimal index for which this holds (and $\alpha_j = \infty$ if in clause C_j no literal becomes true). Scanning $\mathcal{P}(\varphi)$ from the top to the bottom and watching the blocks $B_1, ..., B_k$, the assignments $X_1, ..., X_k$ and the sets P, N, we can construct an FO-formula $fulfill(F)$ (see the appendix) asserting that within each column j, F exactly contains the vertex (α_j, j) and all vertices below. Then φ becomes true under the assignment $X_1, ..., X_k$ if and only if all bottom vertices belong to F.
Thus if k is odd, for the formula $\Phi_k :=$

$$\exists X_1 \forall X_2 \cdots \exists X_k \exists B_1, ..., B_k \exists F \ (\ (\textstyle\bigwedge_{i odd} assign(X_i)) \ \wedge\ blocks(B_1, ..., B_k) \ \wedge\ fulfill(F) \ \wedge \\ ((\textstyle\bigwedge_{i even} assign(X_i)) \to \forall x(bottom(x) \to Fx))\ )$$

we derive $\mathcal{P}(\varphi) \models \Phi_k \iff \exists A_1 \forall A_2 ... \exists A_k(\varphi = 1)$, i.e. $\varphi \in \mathrm{CNF}_{k,\exists}$.
Analogously, for even k we obtain $\Psi_k \in \Pi_k$ (see the appendix). □

Proof of Theorem 6 Let $\mathcal{P}$ be the mapping obtained in Lemma 11. Clearly, for <u>odd k</u>, $\mathcal{P}$ provides a polynomial time reduction from the $\underline{\Sigma_k^P}$-complete problem $\mathrm{CNF}_{k,\exists}$ to the Σ_k-definable problem $\mathrm{MOD}_{\text{3-pic}}(\Phi_k)$. Thus $\mathrm{MOD}_{\text{3-pic}}(\Phi_k)$ is $\underline{\Sigma_k^P}$-complete and (monadic) Σ_k-definable.
For <u>even k</u>, $\mathrm{DNF}_{k,\exists}$ is $\underline{\Sigma_k^P}$-complete. We can reduce this problem to $\mathrm{MOD}_{\text{3-pic}}(\neg\Psi_k)$ as follows: Given a formula $\varphi \in$ DNF, we first compute the conjunctive normal form of its negation $\neg\varphi$ and then construct the corresponding 3-bit picture $\mathcal{P}(\neg\varphi)$. Then the following holds:

$\varphi \in \text{DNF}_{k,\exists} \iff \neg\varphi \notin \text{CNF}_{k,\forall} \iff \mathcal{P}(\neg\varphi) \not\models \Psi_k \iff \mathcal{P}(\neg\varphi) \models \neg\Psi_k$.
As $\neg\Psi_k$ is a Σ_k-formula, $\text{MOD}_{\text{3-pic}}(\neg\Psi_k)$ is $\underline{\Sigma_k^P}$-complete and Σ_k-definable. □

5 An Ehrenfeucht Game on Pictures

In this section we give a new, game-theoretic proof of Theorem 7, which was proved by O. Matz and W. Thomas in [MT97] by means of automata theory and tiling-systems.

Ehrenfeucht games (see e.g. [EF95]) are played on two structures (e.g. pictures) by two players, called Spoiler and Duplicator. Spoiler intends to show a difference between both structures while Duplicator tries to hide such a difference. In order to prove Theorem 7, we introduce the following kind of Ehrenfeucht game.

The (k, q, r)-game on two t-bit pictures $\underline{A}$ and $\underline{B}$

k is the number of second-order rounds, q is the number of first-order rounds and r is the number of sets available in each second-order round to colour both pictures.
The (k, q, r)-game is divided into two parts:
Part 1 k second-order rounds:
In each round Spoiler chooses one of the two pictures and colours it by choosing r sets of vertices in this picture. Duplicator responds by choosing r sets of vertices in the other picture.
Part 2 q first-order rounds:
In each round Spoiler chooses a vertex in one of the two pictures. Duplicator responds by choosing a vertex in the other picture.

At the beginning the t-bit pictures $\underline{A}$ and $\underline{B}$ are already coloured by sets $X_1^A, ..., X_t^A$ and $X_1^B, ..., X_t^B$. Let $X_{t+1}^A, ..., X_{t+kr}^A$ be the sets chosen to colour $\underline{A}$ during the second-order rounds, and $X_{t+1}^B, ..., X_{t+kr}^B$ the sets chosen to colour $\underline{B}$, indexed with respect to the order in which they were chosen.
For each $j \in \{1, ..., q\}$ let x_j^A be the vertex chosen in $\underline{A}$ and x_j^B the vertex chosen in $\underline{B}$ in the jth first-order round.

Duplicator wins iff the following three conditions hold:

1. Considering only the vertices $x_1^A, ..., x_q^A$ and $x_1^B, ..., x_q^B$, the pictures $\underline{A}$ and $\underline{B}$, extended by the new colours chosen in the second-order rounds, look the same. More precisely:
   - $x_j^A \in X_i^A \iff x_j^B \in X_i^B$ for every $j \leqslant q,\ i \leqslant t + kr$,
   - $x_j^A = x_{j'}^A \iff x_j^B = x_{j'}^B$ for all $j, j' \leqslant q$
   - $(x_j^A, x_{j'}^A) \in S_i^A \iff (x_j^B, x_{j'}^B) \in S_i^B$ for every $j, j' \leqslant q,\ i \in \{1, 2\}$,
     i.e. within the grid-structure $x_{j'}^A$ is the row- (respectively column-) successor of x_j^A if and only if $x_{j'}^B$ is the row- (respectively column-) successor of x_j^B.
2. The last 2^q columns of $\underline{A}$ and $\underline{B}$, extended by the new colours chosen during the second-order rounds, are coloured identically. In particular, $\underline{A}$ and $\underline{B}$ must have the same height.
   More precisely: If $\underline{A}$ is of height m^A and width n^A and $\underline{B}$ of height m^B and width n^B, then
   - $m^A = m^B =: m$ and
   - For every $i \leqslant t + kr$ and for each row $\alpha \in \{1, ..., m\}$ and each distance $\beta \in \{0, ..., 2^q - 1\}$ from the rightmost column of the grid, in $\underline{A}$ the vertex in row α and column $(n^A - \beta)$ belongs to X_i^A if and only if in $\underline{B}$ the vertex in row α and column $(n^B - \beta)$ belongs to X_i^B.

3. On the last 2^q columns Duplicator played *column-consistent*, i.e. in each first-order round Duplicator chose a vertex within the last 2^q columns if and only if Spoiler had chosen a vertex within the last 2^q columns; and whenever Spoiler had chosen a vertex within the last 2^q columns, then Duplicator chose a vertex having the same distance from the rightmost column.
   More precisely: If $\underline{A}$ is of width n^A and $\underline{B}$ of width n^B and if x_j^A is in column $(n^A - \beta_j^A)$ and x_j^B is in column $(n^B - \beta_j^B)$, then
   - $\beta_j^A < 2^q \iff \beta_j^B < 2^q$ and
   - if $\beta_j^A < 2^q$ then $\beta_j^A = \beta_j^B$.

We write $\underline{A} \underset{k,q,r}{\equiv} \underline{B}$ if Duplicator has a winning strategy in the (k, q, r)-game on $\underline{A}$ and $\underline{B}$, i.e. if, no matter which sets and vertices Spoiler chooses, Duplicator can always respond in such a way that she wins the game. Note that $\underset{k,q,r}{\equiv}$ is an equivalence relation on the set of all pictures.

Definition 12

1. *With $e_k^{q,r,t}(m)$ we denote the number of equivalence classes in the (k, q, r)-game on t-bit pictures of height m.*
2. *Let $\underline{A}$, $\underline{B}$ be two t-bit pictures of equal height m. With $\underline{AB}$ we denote the t-bit picture obtained by concatenating the first column of $\underline{B}$ to the rightmost column of $\underline{A}$.*
3. *With $\Sigma_k^{q,r}$ we denote the set of all Σ_k-formulas φ such that the first-order part of φ has quantifier depth at most q and each block of set quantifiers in φ consists of at most r set variables. $\Pi_k^{q,r}$ is defined analogously.*
   With $B(\Sigma_k^{q,r})$ we denote the set of all boolean combinations of $\Sigma_k^{q,r}$-formulas.

The following proposition provides a characterization of the (k, q, r)-game which enables us to prove Theorem 7.
Winning condition 1 is necessary to obtain part 1, winning conditions 2 and 3 are essentially needed to establish part 2 and thus part 3 of this proposition.

Proposition 13 *(Characterization of the (k, q, r)-game)*

1. *If $\underline{A} \underset{k,q,r}{\equiv} \underline{B}$ then $\underline{A}$ and $\underline{B}$ cannot be distinguished by formulas in $B(\Sigma_k^{q,r})$, i.e. for every formula $\varphi \in B(\Sigma_k^{q,r})$ it holds that $\underline{A} \models \varphi \iff \underline{B} \models \varphi$.*
2. *Let $\underline{A}$,$\underline{B}$,$\underline{C}$ be t-bit pictures of equal height. If $\underline{A} \underset{k,q,r}{\equiv} \underline{B}$ then $\underline{AC} \underset{k,q,r}{\equiv} \underline{BC}$.*
3. *To each t-bit picture $\underline{A}$ of height m there exists a t-bit picture $\underline{B}$ of height m and width $\leqslant e_k^{q,r,t}(m)$ such that $\underline{A} \underset{k,q,r}{\equiv} \underline{B}$.*
4. *$e_k^{q,r,t}(m) \leqslant s_{k+1}(\mathcal{O}(m))$ for every $k, q, r, t \geqslant 0$.*

Before giving a proof for Proposition 13 we will first apply it to prove Theorem 7.

Proof of Theorem 7 Let $f : \mathbb{N} \longrightarrow \mathbb{N}$ be a function defined by a Σ_k-sentence φ. Let $q, r, t \in \mathbb{N}$ such that φ is of the form $\exists X_1, ..., X_t \psi$ with $\psi \in \Pi_{k-1}^{q,r}$.
For each $m \in \mathbb{N}$ we will show that $f(m) \leqslant e_{k-1}^{q,r,t}(m)$; and thus part 4 of Proposition 13 implies $f(m) \leqslant s_k(\mathcal{O}(m))$ and Theorem 7.
As f is defined by the formula φ, we have $\mathrm{MOD}_{\mathrm{grid}}(\varphi) = \{\underline{[m, f(m)]} : m \geqslant 1\}$. Hence $\underline{[m, f(m)]} \models \varphi$, and thus there exist sets $X_1^A, ..., X_t^A \subseteq \underline{[m, f(m)]}$ such that $\underline{A} := (\underline{[m, f(m)]}, X_1^A, ..., X_t^A) \models \psi$. By part 3 of Proposition 13 there exists a t-bit picture $\underline{B} = (\underline{[m, n]}, X_1^B, ..., X_t^B)$ of width $n \leqslant e_{k-1}^{q,r,t}(m)$ such that $\underline{A} \underset{k-1,q,r}{\equiv} \underline{B}$. By part 1 of Proposition 13, $\underline{A}$ and $\underline{B}$ cannot be distinguished by $\Pi_{k-1}^{q,r}$-formulas. Thus $\underline{B} \models \psi$ and $\underline{[m, n]} \models \exists X_1, ..., X_t \psi$, i.e. $\underline{[m, n]} \models \varphi$ and $\underline{[m, n]} \in \mathrm{MOD}_{\mathrm{grid}}(\varphi)$. As φ defines f, we obtain

$n = f(m)$ and $f(m) \leqslant e_{k-1}^{q,r,t}(m)$. □

Proof of Proposition 13

1. By contradiction. Let $\varphi \in \mathrm{B}(\Sigma_k^{q,r})$ such that $\underline{A} \models \varphi$ and $\underline{B} \not\models \varphi$. W.l.o.g. $\varphi = \bigwedge_i(\bigvee_j \psi_{i,j})$ for some $i, j \geqslant 0$, $\psi_{i,j} \in (\Sigma_k^{q,r} \cup \Pi_k^{q,r})$. Obviously, there must be indices i, j such that $\underline{A} \models \psi_{i,j}$, and $\underline{B} \not\models \psi_{i,j}$. Spoiler obtains a winning strategy for the (k, q, r)-game on $\underline{A}$ and $\underline{B}$ by playing according to the formula $\psi_{i,j}$ in the way described e.g. in [EF95]. □

2. Duplicator's winning strategy on $\underline{AC}$, $\underline{BC}$ works as follows:
In the second-order rounds, on the $\underline{C}$-part of $\underline{AC}$, $\underline{BC}$, Duplicator chooses the same colouring as Spoiler; on the $\underline{A}$-part and the $\underline{B}$-part of $\underline{AC}$, $\underline{BC}$ Duplicator colours according to her winning strategy on $\underline{A}$, $\underline{B}$.
Playing the first-order rounds according to T. Schwentick's Extension Theorem ([SchT96]), Duplicator directly obtains the following winning strategy on $\underline{AC}$, $\underline{BC}$:
Let $\underline{A} = \underline{A_1 A_2}$ (and $\underline{B} = \underline{B_1 B_2}$) where $\underline{A_2}$ (and $\underline{B_2}$, respectively) consists of the last 2^q columns of $\underline{A}$ (and $\underline{B}$, respectively).
Note that winning condition 2 ensures that $\underline{A_2}$ and $\underline{B_2}$ are coloured identically. Thus on $\underline{A_2 C}$, $\underline{B_2 C}$ Duplicator can win by simply choosing the same vertex as Spoiler did. On the other hand, on $\underline{A_1 A_2}$, $\underline{B_1 B_2}$ Duplicator can win according to her strategy on $\underline{A}$, $\underline{B}$, and winning condition 3 asserts that Duplicator can even play column-consistent on $\underline{A_2}$, $\underline{B_2}$.
At the beginning of the game we view $\underline{A_1}$ and $\underline{B_1}$ as the *strategy area*, $\underline{C}$ as the *identity area* and $\underline{A_2}$ and $\underline{B_2}$ as the *buffer area*. Every time Spoiler chooses a vertex x in the *buffer area*, the three disjoint areas are modified:

- If x is closer to the *identity area* (with respect to the number of columns between x and this area), then the *identity area* is extended by all columns lying between x (inclusively) and the former *identity area*.
- If x is closer to the *strategy area*, then the *strategy area* is extended by all columns lying between x (inclusively) and the former *strategy area*.

Now Spoiler's vertex is either in the *strategy area* or in the *identity area*. In the *identity area* Duplicator responds with the identity, i.e. she chooses the vertex in the same row with the same distance from the rightmost column as Spoiler's vertex; in the *strategy area* Duplicator responds according to her strategy on $\underline{A}$, $\underline{B}$.
One can easily see that after the jth first-order round the width of the *buffer area* is $\geqslant 2^{q-j}$. In particular, after q first-order rounds no vertex chosen in the *strategy area* is adjacent to any vertex chosen in the *identity area*. Thus one can easily see that winning condition 1 is satisfied. As obviously winning conditions 2 and 3 are also satisfied, we have found a winning strategy for Duplicator in the (k, q, r)-game on $\underline{AC}$, $\underline{BC}$, i.e. $\underline{AC} \underset{k,q,r}{\equiv} \underline{BC}$. □

3. We consider the prefixes of $\underline{A}$ and make use of part 2 of this proposition.
If $\underline{A}$ has width $> e_k^{q,r,t}(m)$ then we can divide $\underline{A}$ into three parts $\underline{A_1}$, $\underline{A_2}$, $\underline{A_3}$ such that $\underline{A} = \underline{A_1 A_2 A_3}$, $\underline{A_2}$ has width $\geqslant 1$ and $\underline{A_1} \underset{k,q,r}{\equiv} \underline{A_1 A_2}$. By part 2 of this proposition we obtain $\underline{A_1 A_3} \underset{k,q,r}{\equiv} \underline{A_1 A_2 A_3}$, i.e. $\underline{A_1 A_3} \underset{k,q,r}{\equiv} \underline{A}$ and $\underline{A_1 A_3}$ has smaller width than $\underline{A}$. Hence by induction on the width of $\underline{A}$ we obtain a picture $\underline{B}$ of width $\leqslant e_k^{q,r,t}(m)$ such that $\underline{B} \underset{k,q,r}{\equiv} \underline{A}$. □

4. By induction on the number k of second-order rounds.
$\underline{k = 0}$: It can easily be shown that $e_0^{q,r,t}(m) \leqslant C 2^{t 2^q m} \leqslant s_1(\mathcal{O}(m))$, where $2^{t 2^q m}$ is the number of t-bit colourings of the last 2^q columns of a grid of height m, and C is a constant not depending on m (but only on t and the number q of first-order rounds).
$\underline{k > 0}$: The equivalence type of a t-bit picture $\underline{A}$ in the (k, q, r)-game is determined by the set $\{[(\underline{A}, X_1^A, ..., X_r^A)]_{k-1}^{q,r,(t+r)} : X_i^A \subseteq A\}$ of equivalence types in the $(k-1, q, r)$-game on $(t + r)$-bit pictures, reachable from $\underline{A}$ by choosing r sets in $\underline{A}$.
Thus $e_k^{q,r,t}(m) \leqslant 2^{e_{k-1}^{q,r,(t+r)}(m)} \leqslant 2^{s_k(\mathcal{O}(m))} = s_{k+1}(\mathcal{O}(m))$. □

6 The Strictness of the Monadic Hierarchy over Grids

In this section we prove Theorem 8. Basically our proof is a refinement of the method discribed in [MT97]. Note that in [MT97] a k-fold exponential function (namely the function $(\tilde{f}_k+1)$, where $\tilde{f}_1(m)=m2^m$, $\tilde{f}_{k+1}(m)=\tilde{f}_k(m)2^{\tilde{f}_k(m)}$) was found to be Σ_{2k+3}-definable. In this section we show that for every $k \geqslant 1$ the k-fold exponential function f_k (where $f_1(m)=2^m$, $f_{k+1}(m)=f_k(m)2^{f_k(m)}$) is Σ_k-definable and Π_k-definable, i.e. we construct sentences $\Phi_k \in \Sigma_k$ and $\Psi_k \in \Pi_k$ such that $\text{MOD}_{\text{grid}}(\Phi_k)=\text{MOD}_{\text{grid}}(\Psi_k) = \{\underline{[m, f_k(m)]} : m \geqslant 1\}$.

To explain how our formulas are constructed, we need some more notation. Let $m,n \geqslant 1$, $x=(i,j) \in [m,n]$, $X \subseteq [m,n]$. By $\text{Block}(X,x)=\{y=(i,j') : j' \geqslant j$ and $(i,j'') \notin X$ for all $j<j'' \leqslant j'\}$ we denote the set containing x and all vertices y lying to the right of x and in the same row as x, up to, but not including, the first vertex in X. A vertex y is called *the X-successor of x* if $x,y \in X$, y lies to the right of x and in the same row as x and no vertex between x and y is in X.

Lemma 14 *There are Σ_1-formulas $Block(X,x,y)$ and $Succ(X,x,y)$ such that for every grid $\underline{R}$, $x,y \in R$ and $X \subseteq R$ the following holds:*
1. $(\underline{R},X,x,y) \models Block(X,x,y) \iff y \in Block(X,x)$.
2. $(\underline{R},X,x,y) \models Succ(X,x,y) \iff y$ *is the X-successor of x.*

For a proof see the appendix.

The *characteristic (m,n)-matrix of $X \subseteq [m,n]$* is the matrix $c_{m,n}(X)$ of height m and width n with entries in $\{0,1\}$ which has 1's exactly at the positions in X, i.e. $c_{m,n}(X)_{i,j}=1$ if $(i,j) \in X$, and $c_{m,n}(X)_{i,j}=0$ if $(i,j) \notin X$.
If $x=(i,j) \in [m,n]$, $0 \leqslant l \leqslant n-j+1$, we write $X(x,l)$ to denote the number represented by the binary string of length l which is in $c_{m,n}(X)$ on the l positions to the right of x (inclusively). (We consider binary representations where the lowest bit is at the rightmost position.)

Definition 15 *Let $m,n \geqslant 1$, $C,X,Y \subseteq [m,n]$ and $f:\mathbb{N} \longrightarrow \mathbb{N}$.*

1. *Let $c_{i,j}=c_{m,n}(C)_{i,j}$. For $j \leqslant n$ let $r_j < 2^m$ be the number represented by the binary string $c_{1,j}...c_{m,j}$ standing in the jth column of $c_{m,n}(C)$, when read from the top to the bottom.*
   We call C a column-numbering, *if $r_{j+1} \equiv r_j+1 \pmod{2^m}$ for all $j<n$ (i.e. successive columns have successive C-numbers (modulo 2^m)).*
   A column-numbering C is called complete, *if 0^m is the C-number of the leftmost column and 1^m is the C-number of the rightmost column. (Note that a complete column-numbering of $\underline{[m,n]}$ exists iff n is a multiple of 2^m.)*

2. *We call X an f-*marking, *if X only contains top-row vertices and the top row of $c_{m,n}(X)$ is of the form $(10^{f(m)-1})^r$ for some $r \geqslant 1$, i.e. X divides the top row of $\underline{[m,n]}$ into blocks of width $f(m)$. (In particular, an f-marking of $\underline{[m,n]}$ exists iff n is a multiple of $f(m)$.)*

3. *We call Y an f-*numbering, *if in the top row of the grid every two successive blocks of width $f(m)$ have successive Y-numbers. More precisely: Let X be an f-marking. Then Y is an f-numbering iff $Y(y,f(m)) \equiv Y(x,f(m))+1 \pmod{2^{f(m)}}$ for all $x,y \in X$ such that y is the X-successor of x.*
   An f-numbering Y is called complete, *if $0^{f(m)}$ is the Y-number of the leftmost block of width $f(m)$, and $1^{f(m)}$ is the Y-number of the rightmost block of width $f(m)$. (Note that a complete f-numbering of $\underline{[m,n]}$ exists iff n is a multiple of $f(m)2^{f(m)}$.)*

Applied to a grid $\underline{[m,n]}$, a formula which defines f_k must check whether $n=f_k(m)$. We construct such a formula according to the following strategy:

Step 0:
Let C be a column-numbering such that the leftmost column is numbered by 0^m. Then n is

a multiple of $2^m = f_1(m)$ iff C is complete. If C is not complete, then $n \neq f_k(m)$; else X_1 is an f_1-marking iff X_1 consists of all top row vertices whose column-number is 0^m.
Step i: $(1 \leqslant i < k)$
Let Y_i be an f_i-numbering such that the leftmost block is numbered by $0^{f_i(m)}$. Then n is a multiple of $f_i(m)2^{f_i(m)} = f_{i+1}(m)$ iff Y_i is complete. If Y_i is not complete, then $n \neq f_k(m)$; else X_{i+1} is an f_{i+1}-marking iff X_{i+1} consists of all vertices $x \in X_i$ such that the Y_i-number of Block(X_i, x) is $0^{f_i(m)}$.
Step k:
If X_k is an f_k-marking, then $n = f_k(m)$ iff $|X_k| = 1$.

One can easily see that this strategy leads to a polynomial time algorithm deciding whether for an input grid $\underline{[m,n]}$ it holds that $n = f_k(m)$.
Our formula Φ_k (defining f_k) will be of the following form:

$$\begin{array}{lllll}\exists C, X_1, ..., X_k, Y_1, ..., Y_{k-1} \; (& C \text{ is a compl. column-num.} & \wedge & X_1 \text{ is an } f_1\text{-marking} & \wedge \\ & Y_1 \text{ is a compl. } f_1\text{-num.} & \wedge & X_2 \text{ is an } f_2\text{-marking} & \wedge \\ & \ldots & \wedge & \ldots & \wedge \\ & Y_{k-1} \text{ is a compl. } f_{k-1}\text{-num.} & \wedge & X_k \text{ is an } f_k\text{-marking} & \wedge \\ & |X_k| = 1 & & &).\end{array}$$

The following remark is used both to do the column-numbering and to do the f_k-numbering. It provides a simple formalization of what we do when adding 1 to a q-ary number.

Remark 16 Let $q \geqslant 2,\ r \geqslant 1$. Let $\overline{a} = a_{r-1}...a_0,\ \overline{b} = b_{r-1}...b_0 \in \{0, ..., q-1\}^r$ be the q-ary representations of length r of two natural numbers $a, b < q^r$. Then

$$\begin{array}{rl} a + 1 \equiv b \pmod{q^r} \iff & a_0 + 1 \equiv b_0 \pmod q, \text{ and for all } i < r-2: \\ & \text{If } a_i = q-1 \text{ and } b_i = 0, \text{ then } a_{i+1} + 1 \equiv b_{i+1} \pmod q, \\ & \text{else } a_{i+1} = b_{i+1}. \end{array}$$

The condition "$a_i = q-1$ and $b_i = 0$" means that at bit $i+1$ a carry of 1 must be added. In a context where we can quantify over positions in strings and where $c(x)$ denotes the letter at position x, we can formalize this to

$a + 1 \equiv b \pmod{q^r} \iff$

$$\begin{array}{ll} \forall x_0, x_1, y_0, y_1 \; (\; \{ & (x_1, x_0 \text{ are successive positions in } \overline{a}) \;\wedge \\ & (y_1, y_0 \text{ are successive positions in } \overline{b}) \;\wedge \\ & (x_0 \text{ denotes the same bit in } \overline{a} \text{ as } y_0 \text{ in } \overline{b}) \; \} \;\rightarrow \\ \{ & ((x_0 \text{ is the lowest bit in } \overline{a}) \rightarrow \; c(x_0) + 1 \equiv c(y_0) \pmod q) \;\wedge \\ & ((c(x_0) = q-1 \;\wedge\; c(y_0) = 0) \;\rightarrow\; c(x_1) + 1 \equiv c(y_1) \pmod q) \;\wedge \\ & ((\neg(c(x_0) = q-1 \;\wedge\; c(y_0) = 0)) \;\rightarrow\; c(x_1) = c(y_1)) \; \} \;). \end{array}$$

Interpreting $\overline{a}$ and $\overline{b}$ as the C-colouring of two successive columns of a grid and applying Remark 16 in the binary case (i.e. $q = 2$), we obtain the following lemma.

Lemma 17 *(The column-numbering)*
There are FO-formulas col-num(C) and complete-col-num(C) such that for every grid $\underline{R}$ and every $C \subseteq R$ we have
1. $(\underline{R}, C) \models$ *col-num*$(C) \iff C$ *is a column-numbering of* $\underline{R}$,
2. $(\underline{R}, C) \models$ *complete-col-num*$(C) \iff C$ *is a complete column-numbering of* $\underline{R}$.

For a proof see the appendix.

Lemma 18 *(The f_k-marking)*
There is a FO-formula f_1-marking(X, C) and a Π_1-formula marking(Z, X, Y) such that for every grid $\underline{R}$ and all sets $C, X, Y, Z \subseteq R$ we have:
1. *If C is a complete column-numbering of $\underline{R}$, then*
$(\underline{R}, X, C) \models f_1$-*marking*$(X, C) \iff X$ *is an f_1-marking of* $\underline{R}$.
2. *For all $k \geqslant 1$: If X is an f_k-marking and Y a complete f_k-numbering of $\underline{R}$, then*
$(\underline{R}, X, Y, Z) \models$ *marking*$(Z, X, Y) \iff Z$ *is an f_{k+1}-marking of* $\underline{R}$.

Proof We follow the strategy discribed on page 9.

1. To identify the columns with C-number 0^m, it suffices to look at the highest bits of the column-numbers: At position x the column-number switches from 1^m to 0^m iff the highest bit of the column-number switches from 1 to 0.

$f_1\text{-}marking(X, C) := \forall x\ (Xx \leftrightarrow\ (top(x) \wedge (topleft(x) \vee \exists y(S_2yx \wedge Cy \wedge \neg Cx)))\ ).$

2. We have to assert that Z consists exactly of those $x \in X$ for which the Y-number of Block(X, x) is $0^{f_k(m)}$. Again it suffices to look at the highest bits of the Y-numbers.

$$marking(Z, X, Y) := \forall x, y\ (\ (Zx \rightarrow Xx)\ \wedge\ (topleft(x) \rightarrow Zx)\ \wedge$$
$$(Succ(X, y, x) \rightarrow (Zx \leftrightarrow (Yy\ \wedge\ \neg Yx)))\ ).$$

We can replace the implication "$Succ(X, y, x) \rightarrow \cdots$" by "$\neg Succ(X, y, x) \vee \cdots$".
As $Succ(X, y, x)$ is in Σ_1 (cf. Lemma 14), $marking(Z, X, Y)$ is in Π_1. □

Lemma 19 *(How to compare f_k-numbers)*
Let $\overline{X}_k$ be an abbreviation for $X_1, ..., X_k$, and $\overline{Y}_{k-1}$ an abbreviation for $Y_1, ..., Y_{k-1}$.
For all $k \geqslant 1$ there are Π_k-formulas $equal_k(x, y, Y, \overline{X}_k, \overline{Y}_{k-1})$ and $inc_k(x, y, Y, \overline{X}_k, \overline{Y}_{k-1})$ such that for every grid $\underline{R}$ the following holds:
If X_i is an f_i-marking for all $i \leqslant k$, Y_i is a complete f_i-numbering for all $i < k$, and $x, y \in X_k$, then

1. $(\underline{R}, x, y, Y, \overline{X}_k, \overline{Y}_{k-1}) \models equal_k(x, y, Y, \overline{X}_k, \overline{Y}_{k-1}) \iff Y(x, f_k(m)) = Y(y, f_k(m))$,
   i.e. Block(X_k, x) and Block(X_k, y) have the same Y-number.
2. $(\underline{R}, x, y, Y, \overline{X}_k, \overline{Y}_{k-1}) \models inc_k(x, y, Y, \overline{X}_k, \overline{Y}_{k-1}) \iff$
   $Y(y, f_k(m)) \equiv Y(x, f_k(m)) + 1 \pmod{2^{f_k(m)}}$,
   i.e the Y-number of Block(X_k, y) is one larger than the Y-number of Block(X_k, x).

Proof By induction on k. Let m be the height of the grid $\underline{R}$.

1. The formula $equal_k$: $\underline{k = 1}$: We compare two blocks of width 2^m bit by bit. Note that under a fix column-numbering, all columns within a block of width 2^m have distinct numbers. Two positions in Block(X_1, x) and Block(X_1, y) denote corresponding bits iff they have the same column-number; i.e. iff there exists a column-numbering under which they both have column-number 0. Let $equal\text{-}col\text{-}num(x_0, y_0)$ be such a Σ_1-formula asserting that two top row positions x_0 and y_0 are in corresponding columns (see the appendix). Then we obtain a Π_1-formula $equal_1(x, y, Y, X_1)$ which ensures that every two positions within Block(X_1, x) and Block(X_1, y) which are of equal column-number, have the same Y-colouring. (For the formula, see the appendix.)
$\underline{k > 1}$: We compare the Y-colouring of two blocks of width $f_k(m)$ by comparing all corresponding sub-blocks of width $f_{k-1}(m)$. As Y_{k-1} is a complete f_{k-1}-numbering, we can use the Y_{k-1}-number of a sub-block as its address within the block. We use the formula $equal_{k-1}$ both to compare addresses and Y-colourings of blocks of width $f_{k-1}(m)$. Then the formula $equal_k$ ensures that all sub-blocks which have the same Y_{k-1}-address, have the same Y-colouring, and thus it asserts that the two blocks right to x and to y of width $f_k(m)$ have the same Y-colouring. (For the formula, see the appendix.)

2. The formula inc_k: $\underline{k = 1}$: We apply Remark 16 in the binary case and interpret $\overline{a}$ and $\overline{b}$ as the Y-colourings of Block(X_1, x) and Block(X_1, y) which both have width 2^m. Again we use the Σ_1-formula $equal\text{-}col\text{-}num(x_0, y_0)$ to identify corresponding bits in the two blocks. (For further details see the appendix.)
$\underline{k > 1}$: To construct inc_k we apply Remark 16, where the bits a_j in $\overline{a}$ correspond to the Y-colourings of the sub-blocks of width $f_{k-1}(m)$ lying within Block(X_k, x). Again, we interpret the Y_{k-1}-colourings of these sub-blocks as their addresses within Block(X_k, x); and we use the formulas $equal_{k-1}$ and inc_{k-1} to compare the Y-numbers of these sub-blocks. (For further details see the appendix.) □

Using the Π_k-formula inc_k, we can construct a Π_k-formula asserting that Y is an f_k-numbering:

Lemma 20 *(The f_k-numbering)*
For all $k \geqslant 1$ there are Π_k-formulas f_k-num$(Y, \overline{X}_k, \overline{Y}_{k-1})$ and complete-f_k-num$(Y, \overline{X}_k, \overline{Y}_{k-1})$ such that for every grid $\underline{R}$ the following holds:
If X_i is an f_i-marking for all $i \leqslant k$, and Y_i is a complete f_i-numbering for all $i < k$, then

1. $(\underline{R}, Y, \overline{X}_k, \overline{Y}_{k-1}) \models f_k\text{-}num(Y, \overline{X}_k, \overline{Y}_{k-1}) \iff$ *Y is an f_k-numbering.*
2. $(\underline{R}, Y, \overline{X}_k, \overline{Y}_{k-1}) \models complete\text{-}f_k\text{-}num(Y, \overline{X}_k, \overline{Y}_{k-1}) \iff$
   Y is a complete f_k-numbering.

For a proof see the appendix.

Now, for every $k \geqslant 1$ we can give formulas $\Phi_k \in \Sigma_k$ and $\Psi_k \in \Pi_k$ which both define f_k:

Proof of Theorem 8

$$\Phi_k := \exists C, X_1, ..., X_k, Y_1, ..., Y_{k-1} \ (\ complete\text{-}col\text{-}num(C) \ \wedge \ f_1\text{-}marking(X_1, C) \ \wedge \\ \bigwedge_{i=1}^{k-1} (\ complete\text{-}f_i\text{-}num(Y_i, \overline{X}_i, \overline{Y}_{i-1}) \ \wedge \ marking(X_{i+1}, X_i, Y_i) \ ) \ \wedge \\ Singleton(X_k) \ ),$$

where $Singleton(X_k)$ is a FO-formula which asserts that $|X_k| = 1$.
The correctness of $\Phi_k \in \Sigma_k$ follows directly from the preceding lemmas.
Similarly, we also obtain a Π_k-formula Ψ_k which defines f_k. (See the appendix.)
Hence the proof of Theorem 8 is complete. □

7 Remarks

It might be of interest to consider other kinds of grids, e.g. grids with built-in row- or column-*order* relations which are the transitive closure of the successor relations S_1 and S_2, respectively.

Our game-theoretical proof of Theorem 7 can directly be transferred to $(\leqslant_1, S_2)$-grids which have built-in row-order and column-successor relations. Thus we additionally obtain the strictness of the monadic hierarchy over $(\leqslant_1, S_2)$-grids.

With a slight modification of the proof of Theorem 8, one can see that for every $k \geqslant 1$ the function f_k is Σ_1-definable over $(\leqslant_1, \leqslant_2)$-grids, which have built-in row- and column order relations. The infinity (resp. strictness) of the monadic hierarchy over (coloured or uncoloured) $(\leqslant_1, \leqslant_2)$-grids still is an open problem. But as Theorem 6 can directly be transferred from grids to $(\leqslant_1, \leqslant_2)$-grids, the monadic hierarchy over coloured $(\leqslant_1, \leqslant_2)$-grids is strict unless the polynomial time hierarchy collapses.

References

[AFS97] M. Ajtai, R. Fagin, L. Stockmeyer. private communication, 1997.

[Bue60] J.R. Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grundl. Math.* 6 (1960), 66-92.

[EF95] H. Ebbinghaus, J. Flum. Finite Model Theory. *Springer-Verlag, New York*, 1995.

[Fag74] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of Computation* (R.M. Karp, Ed.), *SIAM-AMS Proceedings* 7 (1974), 43-73.

[MT97] O. Matz and W. Thomas. The monadic quantifier alternation hierarchy over graphs is infinite. *Proceedings of the 12th IEEE Symposium on Logic in Computer Science* (1997), to appear.

[SchN97] N. Schweikardt. The Monadic Second-Order Quantifier Alternation Hierarchy over Grids and Pictures. *Diploma thesis,* Johannes Gutenberg-Universität Mainz, 1997.

[SchT96] T. Schwentick. On winning Ehrenfeucht games and monadic NP. *Annals of Pure and Applied Logic* 79 (1996), 61-92.

[Sto77] L. Stockmeyer. The Polynomial Time Hierarchy. *Theoretical Computer Science* 3 (1977), 1-22.

[TW68] J.W. Thatcher, J.B. Wright. Generalized finite automata with an application to a decision problem of second order logic. *Math. Syst. Theory* 2 (1968), 57-82.

Appendix

We list the formulas and proofs that were omitted in sections 4 and 6.

Section 4

Formulas in Lemma 11

Let $(\oplus_{i=1}^k \varphi_i) \;:=\; (\bigvee_{i=1}^k \varphi_i) \;\wedge\; \neg(\bigvee_{i\neq j}(\varphi_i \wedge \varphi_j))$ denote the "exclusive or".

$$\begin{aligned} \mathit{blocks}(B_1,...,B_k) := \forall x,y\; (\quad & (\oplus_{i=1}^k B_i x) \;\wedge\; (\mathit{top}(x) \to B_1(x)) \wedge \\ & (S_1xy \to (\quad ((Bx \leftrightarrow By) \;\to\; \textstyle\bigvee_{i=1}^k (B_ix \wedge B_iy)) \wedge \\ & \qquad\qquad ((Bx \leftrightarrow \neg By) \;\to\; \textstyle\bigvee_{i=1}^{k-1} (B_ix \wedge B_{i+1}y))\;))\;) \end{aligned}$$

$$\begin{aligned} \mathit{fulfill}(F) := \forall x,y\; (\quad & (\mathit{top}(x) \to (\quad Fx \leftrightarrow ((Px \wedge X_1x) \vee (Nx \wedge \neg X_1x))\;))\wedge \\ & (S_1xy \to (\quad Fy \leftrightarrow \\ & \qquad (Fx \;\vee\; \textstyle\bigvee_{i=1}^k (B_iy \wedge ((Py \wedge X_iy) \vee (Ny \wedge \neg X_iy)))\;)\;))\;) \end{aligned}$$

If k is even, then for the Π_k-formula

$$\begin{aligned} \Psi_k := \forall X_1 \exists X_2 \cdots \exists X_k \exists B_1,...,B_k \exists F\; (\quad & (\textstyle\bigwedge_{i\,even} \mathit{assign}(X_i)) \wedge \\ & \mathit{blocks}(B_1,...,B_k) \;\wedge\; \mathit{fulfill}(F) \wedge \\ & ((\textstyle\bigwedge_{i\,odd} \mathit{assign}(X_i)) \;\to\; \forall x(\mathit{bottom}(x) \to Fx))\;) \end{aligned}$$

we derive $\quad \mathcal{P}(\varphi) \models \Psi_k \iff \forall A_1 \exists A_2 ... \exists A_k (\varphi = 1)$, i.e. $\varphi \in \mathrm{CNF}_{k,\forall}$.

Section 6

Proof of Lemma 14

1. We have to make sure that y is an element in the set $B := \mathrm{Block}(X,x)$ which contains x and exactly all vertices to the right of x, up to, but not including, the first vertex in X. B can be defined by a FO-formula which checks local neighbourhoods.

$$\begin{aligned} \mathit{Block}(X,x,y) := \exists B\; (\quad & By \;\wedge \\ \forall u \quad & \{\, \mathit{left}(u) \;\to\; (Bu \leftrightarrow u{=}x) \,\} \wedge \\ \forall u,v \quad & \{\, S_2uv \;\to\; (Bv \leftrightarrow (v{=}x \vee (Bu \wedge \neg Xv)))\,\}\,. \end{aligned}$$

2. $\quad \mathit{Succ}(X,x,y) := Xx \;\wedge\; Xy \;\wedge\; \exists z(\mathit{Block}(X,x,z) \;\wedge\; S_2zy)$

Obviously, this formula is equivalent to a Σ_1-formula. 2

Proof of Lemma 17

We apply Remark 16 in the binary case (i.e. $q = 2$) and interpret $\overline{a}$ and $\overline{b}$ as the C-colouring of two successive columns of the grid. Then

$S_1x_1x_0$	corresponds to	"x_1, x_0 are successive positions in $\overline{a}$",
$S_2x_0y_0$	corresponds to	"x_0 denotes the same bit as y_0",
$\mathit{bottom}(x_0)$	corresponds to	"x_0 is the lowest bit",
Cx_0 (resp. $\neg Cx_0$)	corresponds to	"$c(x_0) = q-1$ (resp. $c(x_0) = 0$)"
$Cx_0 \leftrightarrow Cy_0$	corresponds to	"$c(x_0) \;=\; c(y_0)$",
$Cx_0 \leftrightarrow \neg Cy_0$	corresponds to	"$c(x_0)+1 \;\equiv\; c(y_0) \pmod 2$",

Thus we obtain the following formula:

$$col\text{-}num(C) := \forall x_0, x_1, y_0, y_1 \ (\ \{ \ S_1 x_1 x_0 \ \wedge \ S_1 y_1 y_0 \ \wedge S_2 x_0 y_0 \ \} \ \rightarrow \\ \{ \ (bottom(x_0) \rightarrow (Cx_0 \leftrightarrow \neg Cy_0)) \ \wedge \\ ((Cx_0 \wedge \neg Cy_0) \rightarrow (Cx_1 \leftrightarrow \neg Cy_1)) \wedge \\ ((\neg(Cx_0 \wedge \neg Cy_0)) \rightarrow (Cx_1 \leftrightarrow Cy_1)) \ \} \ ),$$

and

$$complete\text{-}col\text{-}num(C) := col\text{-}num(C) \wedge (\forall x(left(x) \rightarrow \neg Cx)) \wedge (\forall x(right(x) \rightarrow Cx))$$

additionally asserts that the leftmost column of $c_{m,n}(C)$ only contains 0's and that the rightmost column consists of 1's. 2

Formulas in Lemma 19

1. *equal-col-num*:

For two top row vertices x_0, y_0, the formula *equal-col-num* must assert that there exists a column-numbering such that x_0 and y_0 both are in columns of number 0.

$$equal\text{-}col\text{-}num(x_0, y_0) \ := \ \exists C \ (\ col\text{-}num(C) \wedge zero\text{-}col(x_0, C) \wedge zero\text{-}col(y_0, C) \ ),$$

where

$$zero\text{-}col(x_0, C) \ := \ (\exists z \ (S_2 z x_0 \wedge Cz \wedge \neg Cx_0)) \ \vee \ (left(x_0) \wedge (\forall z \ (left(z) \rightarrow \neg Cz)) \ )$$

asserts that either at point x_0 the highest bit of the C-number switches from 1 to 0, or x_0 is in the leftmost column and the leftmost column has C-number 0, i.e. $zero\text{-}col(x_0, C)$ asserts that x_0 is in a column of C-number 0.

2. $equal_1$:

$$equal_1(x, y, Y, X_1) \ := \forall x_0, y_0 \ (\ \ (Block(X_1, x, x_0) \ \wedge Block(X_1, y, y_0) \ \wedge \ equal\text{-}col\text{-}num(x_0, y_0)) \rightarrow \\ (Yx_0 \leftrightarrow Yy_0) \ ).$$

Note that this formula is equivalent to a Π_1-formula.

3. $equal_k$:

$$equal_k(x, y, Y, \overline{X}_k, \overline{Y}_{k-1}) \ := \ \ \forall x_0, y_0 \ (\ \ (\ X_{k-1} x_0 \ \wedge \ Block(X_k, x, x_0) \ \wedge \\ X_{k-1} y_0 \ \wedge \ Block(X_k, y, y_0) \ \wedge \\ equal_{k-1}(x_0, y_0, Y_{k-1}, \overline{X}_{k-1}, \overline{Y}_{k-2}) \ ) \ \ \rightarrow \\ equal_{k-1}(x_0, y_0, Y, \overline{X}_{k-1}, \overline{Y}_{k-2}) \ \ ).$$

4. inc_1:

$S_2 x_1 x_0 \ \wedge \ Block(X_1, x, x_1) \ \wedge \ Block(X_1, x, x_0)$	corresponds to	"x_1, x_0 are successive positions in $\overline{a}$ ",
$equal\text{-}col\text{-}num(x_0, y_0)$	corresponds to	"x_0 denotes the same bit in $\overline{a}$ as y_0 in $\overline{b}$ ",
$right(x_0) \ \vee \ \exists z(S_2 x_0 z \wedge X_1 z)$	corresponds to	"x_0 is the lowest bit in $\overline{a}$ ",
Cx_0 (resp. $\neg Cx_0$)	corresponds to	"$c(x_0) = q - 1$ (resp. $c(x_0) = 0$)"
$Cx_0 \leftrightarrow Cy_0$	corresponds to	"$c(x_0) \ = \ c(y_0)$",
$Cx_0 \leftrightarrow \neg Cy_0$	corresponds to	"$c(x_0) + 1 \ \equiv \ c(y_0) \ (\mathrm{mod}\ 2)$",

Thus we obtain the following formula:

$$inc_1 := \forall x_0, x_1, y_0, y_1 \ (\ \ \{ \ S_2 x_1 x_0 \ \wedge \ Block(X_1, x, x_1) \ \wedge \ Block(X_1, x, x_0) \ \wedge \\ S_2 y_1 y_0 \ \wedge \ Block(X_1, y, y_1) \ \wedge \ Block(X_1, y, y_0) \ \wedge \\ equal\text{-}col\text{-}num(x_0, y_0) \ \} \ \rightarrow \\ \{ \ ((right(x_0) \ \vee \ \exists z(S_2 x_0 z \wedge X_1 z)) \ \rightarrow \ (Cx_0 \leftrightarrow \neg Cy_0)) \ \wedge \\ ((Cx_0 \wedge \neg Cy_0) \ \rightarrow \ (Cx_1 \leftrightarrow \neg Cy_1)) \ \wedge \\ ((\neg(Cx_0 \wedge \neg Cy_0)) \ \rightarrow \ (Cx_1 \leftrightarrow Cy_1)) \ \} \ ).$$

As $Block(X_1, x, x_0)$ and $equal\text{-}col\text{-}num(x_0, y_0)$ are in Σ_1, one can easily see that inc_1 is in Π_1 (note that implications "$u \rightarrow v$" can be replaced by "$\neg u \vee v$").

5. inc_k:

Let $\ max(Y, x_0, X_{k-1}) \ := \ \forall z(Block(X_{k-1}, x_0, z) \rightarrow Yz)$ be a Π_1-formula which asserts that the Y-colouring of Block(X_{k-1}, x_0) consists of 1's.

Similarly $min(Y, x_0, X_{k-1}) := \forall z(Block(X_{k-1}, x_0, z) \rightarrow \neg Yz)$ is a Π_1-formula which asserts that the Y-colouring of $\mathrm{Block}(X_{k-1}, x_0)$ consists of 0's.

$Succ(X_{k-1}, x_1, x_0) \wedge Block(X_k, x, x_1) \wedge Block(X_k, x, x_0)$	cor. to	"x_1, x_0 are successive positions in $\overline{a}$",
$equal_{k-1}(x_0, y_0, Y_{k-1}, \overline{X}_{k-1}, \overline{Y}_{k-2})$	cor. to	"x_0 denotes the same bit in $\overline{a}$ as y_0 in $\overline{b}$",
$max(Y_{k-1}, x_0, X_{k-1})$	cor. to	"x_0 is the lowest bit in $\overline{a}$",
$max(Y, x_0, X_{k-1})$ (resp. $min(Y, x_0, X_{k-1})$)	cor. to	"$c(x_0) = q - 1$ (resp. $c(x_0) = 0$)"
$equal_{k-1}(x_0, y_0, Y, \overline{X}_{k-1}, \overline{Y}_{k-2})$	cor. to	"$c(x_0) = c(y_0)$",
$inc_{k-1}(x_0, y_0, Y, \overline{X}_{k-1}, \overline{Y}_{k-2})$	cor. to	"$c(x_0) + 1 \equiv c(y_0) \pmod q$",

Thus we obtain the following formula:

$$
\begin{aligned}
inc_k := \forall x_0, x_1, y_0, y_1 \ (\quad & \{ Succ(X_{k-1}, x_1, x_0) \wedge Block(X_k, x, x_1) \wedge Block(X_k, x, x_0) \wedge \\
& \quad Succ(X_{k-1}, y_1, y_0) \wedge Block(X_k, y, y_1) \wedge Block(X_k, y, y_0) \wedge \\
& \quad equal_{k-1}(x_0, y_0, Y_{k-1}, \overline{X}_{k-1}, \overline{Y}_{k-2}) \} \rightarrow \\
& \{ (max(Y_{k-1}, x_0, X_{k-1}) \rightarrow inc_{k-1}(x_0, y_0, Y, \overline{X}_{k-1}, \overline{Y}_{k-2})) \wedge \\
& \quad ((max(Y, x_0, X_{k-1}) \wedge min(Y, y_0, X_{k-1})) \rightarrow inc_{k-1}(x_1, y_1, Y, \overline{X}_{k-1}, \overline{Y}_{k-2})) \wedge \\
& \quad ((\neg(max(Y, x_0, X_{k-1}) \wedge min(Y, y_0, X_{k-1}))) \rightarrow equal_{k-1}(x_1, y_1, Y, \overline{X}_{k-1}, \overline{Y}_{k-2})) \}).
\end{aligned}
$$

As $equal_{k-1}$ and inc_{k-1} are in Π_{k-1}, one can easily see that inc_k is in Π_k (note that implications "$u \rightarrow v$" can be replaced by "$\neg u \vee v$").

Proof of Lemma 20

$$f_k\text{-}num(Y, \overline{X}_k, \overline{Y}_{k-1}) := \forall x, y \ (Succ(X_k, x, y) \rightarrow inc_k(x, y, Y, \overline{X}_k, \overline{Y}_{k-1}))$$

is a Π_k-formula asserting that every two successive blocks of width $f_k(m)$ (where m is the height of the grid) have successive Y-numbers.

To obtain a complete f_k-numbering, we additionally have to make sure that the leftmost block has Y-number 0 and that the rightmost block has Y-number 1. We obtain the following formula:

$$
\begin{aligned}
complete\text{-}f_k\text{-}num(Y, \overline{X}_k, \overline{Y}_{k-1}) := \quad & f_k\text{-}num(Y, \overline{X}_k, \overline{Y}_{k-1}) \wedge \\
& (\forall x(topleft(x) \rightarrow min(Y, x, X_k))) \wedge \\
& (\forall x, y((topright(y) \wedge X_k x \wedge Block(X_k, x, y)) \rightarrow max(Y, x, X_k))),
\end{aligned}
$$

where $(topright(y) \wedge X_k x \wedge Block(X_k, x, y))$ makes sure that x is the starting point of the rightmost f_k-block. 2

The Π_k-formula Ψ_k which defines f_k

Ψ_k is constructed according to the strategy explained on page 9 and is of the following form:

$\forall C, X_1, ..., X_k, Y_1, ..., Y_{k-1}$ (
($col\text{-}num(C)$ $\wedge$ the leftmost column has C-number 0) $\rightarrow$ (C is complete $\wedge$ (
($f_1\text{-}marking(X_1, C) \wedge f_1\text{-}numbering(Y_1...) \wedge$ leftmost block has Y_1-number 0) $\rightarrow$ (Y_1 complete $\wedge$ (
...
($mark.(X_{k-1}, ...) \wedge f_{k-1}\text{-}num.(Y_{k-1}...) \wedge$ leftm. block: Y_{k-1}-number 0) $\rightarrow$ (Y_{k-1} complete $\wedge$ (
($marking(X_k, ...) \rightarrow |X_k| = 1$)) $\cdots$)).

This formula is equivalent to a Π_k-formula.