Analogs of Fagin's Theorem for Small Nondeterministic Finite Automata

Christos A. Kapoutsis⋆ and Nans Lefebvre

LIAFA, Université Paris VII, France

Abstract. Let 1N and SN be the classes of families of problems solvable by families of polynomial-size *one-way* and *sweeping nondeterministic finite automata*, respectively. We characterize 1N in terms of families of polynomial-length formulas of *monadic second-order logic with successor*. These formulas existentially quantify two local conditions in disjunctive normal form: one on cells polynomially away from the two ends of the input, and one more on the cells of a fixed-width window sliding along it. We then repeat the same for SN and for slightly more complex formulas.

1 Introduction

The 'Sakoda-Sipser analogy' suggests that, parallel to the standard *complexity theory* that measures *time* on *Turing machines*, one can build a robust complexity theory measuring *size* in *two-way finite automata* [10]. An updated suggested outline of such a theory was given in [6], and the name '*minicomplexity theory*' was proposed soon later. One premise behind such research is that many phenomena of standard complexity theory emerge already in much weaker devices, and that their study at such early level may deepen our understanding.

Here we test this premise relative to *descriptive complexity theory*, the logical parallel of complexity theory where, instead of the Turing machines that solve a problem, we study the logical formulas that specify it [5]. Does minicomplexity theory have such a parallel? For example, consider Fagin's Theorem, the logical characterization of NP which inaugurated descriptive complexity [4]: Is there an analogous theorem for the minicomplexity counterpart of NP, the class 2N of problems solvable by polynomial-size two-way nondeterministic finite automata?

We answer this question for the *one-way* and *sweeping* restrictions of 2N, the subclasses 1N and SN corresponding to automata whose heads move only forward (1NFAs) or reverse only on end-markers (SNFAs). We start at Büchi's Theorem, which translates between 1NFAs and formulas of *monadic second-order logic with successor* (MSO[S]) [3]. There, the tempting guess that polynomial-size 1NFAs correspond to polynomial-length MSO[S] formulas is valid only from automata to formulas; in contrast, polynomial-size formulas may translate to 1NFAs of non-elementary size [9]. We thus refine Büchi's proof, to find suitably restricted formulas where polynomial length indeed corresponds to polynomial 1NFA size.

⋆ Supported by a Marie Curie Intra-European Fellowship (PIEF-GA-2009-253368) within the European Union Seventh Framework Programme (FP7/2007-2013).

We arrive at ‘*existential anchor-slide* DNFs’ (EAS/DNFs), formulas which quantify existentially two ‘local’ conditions in disjunctive normal form: an ‘*anchor*’, which describes cells that are ‘anchored’ relative to the two ends of the input; and a ‘*slide*’, which describes the cells of a window that ‘slides’ along the input. Our Theorem 1 is that the desired correspondence indeed holds when the anchored cells lie polynomially near the two ends and the width of the sliding window is constant. Then, our Theorem 2 generalizes this correspondence to SNFAs and to EAS/DNFs of a ‘*multi-core*’ variant of many anchors/slides with limited variable access; our argument naturally involves *rotating* automata (SNFAs with only forward passes) and the corresponding class RN, actually reproving RN = SN [7].

2 Preparation

2.1 Nondeterministic Finite Automata

A *sweeping nondeterministic finite automaton* (SNFA) is a tuple $N = (S, \Sigma, \delta, q_0)$ of a set of *states* S, an *alphabet* Σ, a *special state* $q_0 \in S$, and a set of *transitions* $\delta \subseteq S \times (\Sigma \cup \{\vdash, \dashv\}) \times S$, where $\vdash, \dashv \notin \Sigma$ are two *end-markers*. A word $w \in \Sigma^*$ is presented to N between the end-markers (Fig. 1a). The computation starts at q_0 on $\vdash$. At every step, the next state may be any of those derived from δ and the current state and symbol. The next tape cell is always the adjacent one in the direction of motion; except if the current symbol is $\dashv$ and the next state is not q_0 or if the current symbol is $\vdash$, in which two cases the next cell is the adjacent one towards the other end-marker. So, each branch of the resulting computation performs a number of alternating forward and backward *passes* over $\vdash w \dashv$, and eventually loops, hangs, or falls off $\dashv$ into q_0. In the last case, we say N *accepts* w.

We say N is *layered* if S can be split into ρ *layers* $S_1, \dots, S_\rho$ such that all accepting computations perform exactly ρ passes and every r-th pass ($1 \leq r \leq \rho$) uses only transitions departing from states in S_r. Pictorially, the state diagram consists of ρ sub-diagrams, each visited exactly once and only through transitions on $\vdash$ or $\dashv$ (Fig. 1c). With a small increase in size, every SNFA can be made layered.

Lemma 1. *Every s-state* SNFA *has a $O(s^2)$-state equivalent with $< 2s$ layers.*

A *rotating nondeterministic finite automaton* (RNFA) is a SNFA that performs only forward passes (Fig. 1b). Formally, we just change how we pick the next cell:

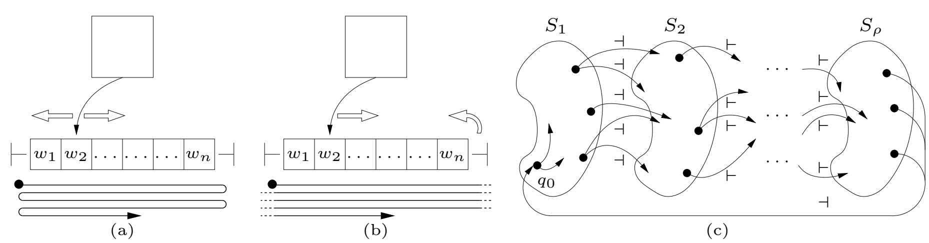

Fig. 1. Schematic of (a) a SNFA, (b) a RNFA, (c) the state diagram of a layered SNFA.

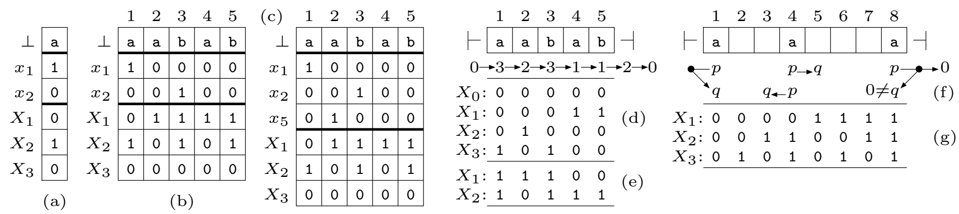

Fig. 2. (a) A column of $\Sigma|V_1|V_2$, if $\Sigma = \{\mathtt{a},\mathtt{b}\}$, $V_1 = \{x_1,x_2\}$, $V_2 = \{X_1,X_2,X_3\}$. (b) A well-formed $\hat{w}$ over $\Sigma|V_1|V_2$; here $\hat{w}(\perp) = \mathtt{aabab}$, $\hat{w}(x_2) = 3$, $\hat{w}(X_2) = \{1,3,5\}$. (c) The word $\hat{w}[x_5/2]$. (d,e) Encoding a computation of a 4-state 1NFA, with 1 variable per state (d), or per bit in the codes of states (e). (f) Defining $\alpha^{\vdash}_{pq}$, α_{pq}, $\alpha^{\dashv}_{p}$, and $\alpha^{\dashv}_{pq}$. (g) Checking that a word has length 8, by implementing a 3-bit counter.

it is always the adjacent one to the right; except if the current symbol is $\dashv$ and the next state is not q_0, in which case the next cell is that of $\vdash$. *Layered* RNFAs are defined similarly, and satisfy Lemma 1 with 'RNFA' and '$\leq s$' instead of 'SNFA' and '$< 2s$'. A *one-way nondeterministic finite automaton* (1NFA) is a RNFA that performs only 1 pass. Formally, we just insist that every $(\,.\,,\dashv,.\,) \in \delta$ is of the form $(\,.\,,\dashv, q_0)$. *Deterministic* 1NFAs (1DFAs) obey the usual restriction.

A (*promise*) *problem* over Σ is a pair $\mathfrak{L} = (L, \tilde{L})$ of disjoint subsets of Σ^*. A machine *solves* $\mathfrak{L}$ if it accepts all $w \in L$ but no $w \in \tilde{L}$. A family of machines $\mathcal{M} = (M_h)_{h\geq 1}$ *solves* a family of problems $(\mathfrak{L}_h)_{h\geq 1}$ if every M_h solves $\mathfrak{L}_h$. The machines of $\mathcal{M}$ are *small* if every M_h has $\leq p(h)$ states, for some polynomial p.

2.2 Monadic Second-Order Logic with Successor

In *monadic second-order logic with successor* over Σ ($\text{MSO}_\Sigma[\mathsf{S}]$), formulas are built from a list of *first-order variables* $x_1, x_2, \ldots$, a list of *monadic second-order variables* $X_1, X_2, \ldots$, one predicate $a(\,.\,)$ for each $a \in \Sigma$, the *successor* predicate $\mathsf{S}(\,.\,,.\,)$, the connectives $\wedge,\vee,\neg$, and the quantifiers $\exists,\forall$.[1] Each formula φ is either an *atom*, of the form $a(x)$, $X(x)$, or $\mathsf{S}(x, y)$; or *compound*, of the form $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\exists x\phi$, $\forall x\phi$, $\exists X\phi$, or $\forall X\phi$, where x,y two f.o. variables, X a s.o. variable, $a \in \Sigma$, and ϕ,ψ two simpler formulas. The *length* $|\varphi|$ of φ is the number of occurences of symbols in it, ignoring punctuation and counting each x_i, X_i, and a as 1 symbol. An atom or negation of an atom is called *literal*; a conjunction (resp., disjunction) of literals is called $\wedge$-*clause* ($\vee$-*clause*); a disjunction (conjunction) of $\leq m$ such clauses is called an m-DNF (m-CNF).[2]

Formulas of $\text{MSO}_\Sigma[\mathsf{S}]$ are interpreted on words over alphabets that extend Σ, as follows. For V_1,V_2 two sets of f.o. and s.o. variables respectively, let $\Sigma|V_1|V_2$ be the alphabet of all functions $u : \{\perp\}\cup V_1\cup V_2 \to \Sigma\cup\{\mathtt{0},\mathtt{1}\}$ that map $\perp$ into Σ and variables into $\{\mathtt{0},\mathtt{1}\}$: $u(\perp) \in \Sigma$ and $u[V_1\cup V_2] \subseteq \{\mathtt{0},\mathtt{1}\}$. Intuitively, every such u is a column of $1+|V_1|+|V_2|$ cells, labelled by the elements of $\{\perp\}\cup V_1\cup V_2$ and filled

[1] The *equality* predicate $\,.=.\,$ may also be used, but we will not need it.

[2] Note that in standard complexity the meaning of "2-CNF", "3-CNF", etc. is different.

by the respective u-values (Fig. 2a). Likewise, every $\hat{w} = \hat{w}_1 \cdots \hat{w}_n \in (\Sigma|V_1|V_2)^*$ is a table of n columns, and $1+|V_1|+|V_2|$ rows: one labelled $\perp$, hosting an n-long word over Σ; the rest labelled by variables, hosting n-long bitstrings (Fig. 2b). We say $\hat{w}$ is *well-formed* if $n \neq 0$ and each f.o. variable row hosts exactly one $\texttt{1}$. Then $\hat{w}(\perp)$ is the $\perp$-row word $\hat{w}_1(\perp) \cdots \hat{w}_n(\perp) \in \Sigma^*$; $\hat{w}(x)$ is the index i of the unique $\hat{w}_i$ hosting $\texttt{1}$ in the row of $x \in V_1$; and $\hat{w}(X)$ is the set $\{i \mid \hat{w}_i(X){=}\texttt{1}\}$ of indices of columns hosting $\texttt{1}$ in the row of $X \in V_2$ (Fig. 2b). If $y \notin V_1$ and $1 \leq i \leq n$, then $\hat{w}[y/i]$ is the well-formed $\hat{w}'$ over $\Sigma|V_1{\cup}\{y\}|V_2$ derived from $\hat{w}$ by adding a row with label y and bits such that $\hat{w}'(y) = i$ (Fig. 2c); similarly for $\hat{w}[Y/I]$, when $Y \notin V_2$ and $I \subseteq \{1, \ldots, n\}$.

Given a well-formed n-long $\hat{w}$ over $\Sigma|V_1|V_2$ and a formula $\varphi(\overline{x}, \overline{X})$ with its free variables $\overline{x}$ and $\overline{X}$ in $V_1 \cup V_2$, we say $\hat{w}$ *satisfies* φ, in symbols $\hat{w} \models \varphi$, if:

$$\text{for } \varphi \equiv a(x): \quad \hat{w}_{\hat{w}(x)}(\perp) = a \tag{1}$$
$$\text{for } \varphi \equiv X(x): \quad \hat{w}(x) \in \hat{w}(X) \tag{2}$$
$$\text{for } \varphi \equiv \mathsf{S}(x,y): \quad \hat{w}(x) + 1 = \hat{w}(y) \tag{3}$$
$$\text{for } \varphi \equiv \exists x \phi: \quad \text{there exists } i \in \{1, \ldots, n\} \text{ such that } \hat{w}[x/i] \models \phi$$
$$\text{for } \varphi \equiv \exists X \phi: \quad \text{there exists } I \subseteq \{1, \ldots, n\} \text{ such that } \hat{w}[X/I] \models \phi\,,$$

and similarly or in obvious ways for $\varphi \equiv \neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\forall x \phi$, or $\forall X \phi$.

We introduce an extension of $\text{MSO}_\Sigma[\mathsf{S}]$, called $\text{MSO}^+_\Sigma[\mathsf{S}, \mathbb{Z}^*]$. The '+' means that, instead of predicates $a(\,.\,)$ for $a \in \Sigma$, we use predicates $\alpha(\,.\,)$ for $\alpha \subseteq \Sigma$. The '$\mathbb{Z}^*$' means that we now use constants from $\mathbb{Z}^* := \{\pm 1, \pm 2, \ldots\}$ to refer to specific columns. So, now a *term* is any f.o. variable x or constant $c \in \mathbb{Z}^*$, and an *atom* has the form $\alpha(t)$, $X(t)$, or $\mathsf{S}(t, t')$, where $\alpha \subseteq \Sigma$ and t,t' are terms. The *length* of a formula φ is extended so that each α and c count as 1 symbol, too. The *margin* of φ is $\max\{|c| \mid c \in \mathbb{Z}^* \text{ occurs in } \varphi\}$; or 0, if φ uses no constants.

On a well-formed n-long $\hat{w}$ over $\Sigma|V_1|V_2$, the meaning $\hat{w}(c)$ of a constant c is just c, if $1 \leq c \leq n$; or $n{+}c{+}1$, if $-n \leq c \leq -1$; or undefined, otherwise. So, positive (resp., negative) constants refer to a column by its offset from the left (right) end of $\hat{w}$. Then, the definition of $\hat{w} \models \varphi$ is modified in cases (1)-(3):

$$\text{for } \varphi \equiv \alpha(t): \quad \hat{w}_{\hat{w}(t)}(\perp) \in \alpha \tag{1$'$}$$
$$\text{for } \varphi \equiv X(t): \quad \hat{w}(t) \in \hat{w}(X) \tag{2$'$}$$
$$\text{for } \varphi \equiv \mathsf{S}(t,t'): \quad \hat{w}(t) + 1 = \hat{w}(t')\,; \tag{3$'$}$$

in addition, we declare $\hat{w} \models \varphi$ automatically false if φ uses any constant $> n$.

The next lemma says that $\text{MSO}^+_\Sigma[\mathsf{S}, \mathbb{Z}^*]$ is as expressive as $\text{MSO}_\Sigma[\mathsf{S}]$, but more concise. Still, the savings in formula length are negligible, if we ignore polynomial differences and if alphabet size, margin, and length are polynomially related.

Lemma 2. *Every* $\text{MSO}_\Sigma[\mathsf{S}]$ *formula of length* l *has an equivalent in* $\text{MSO}^+_\Sigma[\mathsf{S}, \mathbb{Z}^*]$ *of margin* 0 *and length* $\leq l$. *Conversely, every* $\text{MSO}^+_\Sigma[\mathsf{S}, \mathbb{Z}^*]$ *formula of margin* τ *and length* l *has an equivalent in* $\text{MSO}_\Sigma[\mathsf{S}]$ *of length* $O(\tau{+}\sigma l)$*, where* $\sigma := |\Sigma|$.

A formula $\varphi(\overline{x}, \overline{X})$ *solves* a problem $\mathfrak{L} = (L, \tilde{L})$ over $\Sigma|\overline{x}|\overline{X}$ if $\hat{w} \models \varphi$ for all well-formed $\hat{w} \in L$ but no well-formed $\hat{w} \in \tilde{L}$. A family of formulas $\mathcal{F} = (\varphi_h)_{h \geq 1}$

solves a family of problems $(\mathfrak{L}_h)_{h\geq 1}$ if every φ_h solves $\mathfrak{L}_h$. The formulas of $\mathcal{F}$ are *small* if every φ_h has length $\leq p(h)$, for some polynomial p.

3 Existential Anchor-Slide Sentences

A formula is *local* if it is free of $\mathsf{S}(\,.\,,.\,)$ and quantifiers; so, it is built just by applying $\wedge$,$\vee$,$\neg$ to atoms of the form $\alpha(t)$ and $X(t)$. E.g., if $\tilde{\mathtt{a}} := \{\mathtt{a}\}$ then

$$\begin{aligned} &\psi_*(X) := \tilde{\mathtt{a}}(+1) \wedge X(+1) \\ \text{and} \qquad &\phi_*(x,y,X) := [\ \tilde{\mathtt{a}}(x) \wedge X(x) \wedge \neg X(y)\ ] \vee [\ \neg X(x) \wedge X(y)\ ] \end{aligned} \tag{4}$$

are two local formulas. A local formula is *anchored* if all its terms are constants (e.g., as in ψ_*); it is *floating* if all its terms are f.o. variables (e.g., as in ϕ_*).

Now let $\phi(x_1,\dots,x_k,\overline{X})$ be a floating local, for some $k \geq 1$. Then the formula

$$\forall x_1 \cdots \forall x_k [\ \mathsf{S}(x_1,x_2) \wedge \cdots \wedge \mathsf{S}(x_{k-1},x_k) \ \rightarrow\ \phi(x_1,\dots,x_k,\overline{X})\ ]$$

claims that ϕ is true on every k successive cells; or, more intuitively, that ϕ holds at every stop of a window of width k which slides along the word. We call this a *sliding* formula, we represent it more succinctly with the shorthand notation

$$\forall \widehat{x_1 \cdots x_k}\, \phi(x_1,\dots,x_k,\overline{X})\,,$$

and refer to k and ϕ as its *width* and *float*. (For $k = 1$, this is just $\forall x_1 \phi(x_1,\overline{X})$.)

We are interested in sentences that are existentially quantified conjunctions of an anchored local and a sliding formula; that is, sentences of the form

$$\exists X_1 \dots \exists X_d [\ \psi(\overline{X})\ \wedge\ \forall \widehat{x_1 \cdots x_k}\, \phi(\overline{x},\overline{X})\ ]\,, \tag{5}$$

where ψ is anchored local of some margin τ; ϕ is floating local; and $\overline{X},\overline{x}$ are short for $X_1,\dots,X_d$, $x_1,\dots,x_k$. We call (5) an *existential anchor-slide sentence* (EAS) of *depth* d, *margin* τ, and *width* k, having *anchor* ψ, *float* ϕ, *slide* $\forall\widehat{\overline{x}}\,\phi$, and *core* $\psi \wedge \forall\widehat{\overline{x}}\,\phi$. We say it is in m-DNF (resp., m-CNF), an EAS/DNF (EAS/CNF), if both ψ and ϕ are m-DNFs (m-CNFs). E.g., for the ψ_*,ϕ_* of (4), here is an EAS in 2-DNF

$$\exists X [\ \psi_*(X)\ \wedge\ \forall \widehat{xy}\, \phi_*(x,y,X)\ ]$$

of depth 1, margin 1, and width 2 (satisfied iff all odd-indexed cells host an $\mathtt{a}$).

Our first theorem says that polynomial-size 1NFAs are equivalent to EAS/DNFs of polynomial length, polynomial margin, and constant width; and that this holds already when the depth is logarithmic, the margin is 1, and the width is 2.

Theorem 1. *The following are equivalent, for every family of problems $\mathcal{L}$:*
1. *$\mathcal{L}$ has small* 1NFAs*.*
2. *$\mathcal{L}$ has small* EAS/DNFs *of logarithmic depth, margin* 1*, and width* 2*.*
3. *$\mathcal{L}$ has small* EAS/DNFs *of small margin and fixed width.*

Proof. [(1)⇒(2)] By Lemma 3. [(2)⇒(3)] Trivial. [(3)⇒(1)] By Lemma 10. $\square$

Our next theorem generalizes Theorem 1 to SNFAs and sentences of the form

$$\exists\overline{X}_1 \ldots \exists\overline{X}_\rho \bigwedge_{r=1}^{\rho}[\ \psi_r(\overline{X}_r, \overline{X}_{r+1}) \ \wedge\ \forall\widehat{x_1 \cdots x_k}\, \phi_r(\overline{x}, \overline{X}_r)\ ]\,, \tag{6}$$

where each ψ_r is anchored local of some margin τ; each ϕ_r is floating local; each $\overline{X}_r$ is short for $X_{r,1}, \ldots, X_{r,d}$ for some d; and $\overline{x}$ is short for $x_1, \ldots, x_k$.[3] Note how the $X_{r,j}$ are split into ρ groups so that the r-th core uses only groups r and $r{+}1$ in its anchor and only group r in its float. We call (6) an *existential multicore anchor-slide sentence* (EMAS) of *multiplicity* ρ, *depth* d, *margin* τ, and *width* k. We say it is in m-DNF, an EMAS/DNF, if all anchors and floats are m-DNFs.

Theorem 2. *The following are equivalent, for every family of problems $\mathcal{L}$:*
1. *$\mathcal{L}$ has small* RNFAs.
2. *$\mathcal{L}$ has small* SNFAs.
3. *$\mathcal{L}$ has small* EMAS/DNFs *of logarithmic depth, margin* 1*, and width* 2.
4. *$\mathcal{L}$ has small* EMAS/DNFs *of small margin and fixed width.*

Proof. [(1)⇒(2),(3)⇒(4)] Trivial. [(2)⇒(3),(4)⇒(1)] By Lemmas 4 and 13. □

4 From Automata to Formulas

The standard construction of an MSO[S] sentence for an s-state 1NFA uses, for each state p, a variable X_p for the set of cells where p is used along an accepting computation (Fig. 2d) [3]. The result can be cast into an EAS/DNF of depth s and length $O(s^3)$. A trick of [11] reduces the depth to 1 but increases the length to quasi-polynomial. The next lemma finds a EAS/DNF of logarithmic depth and polynomial length. Then Lemma 4 generalizes this to SNFAs and EMAS/DNFs.

Lemma 3. *Every s-state* 1NFA *has an* EAS *in s^2*-DNF*, of depth $\lceil \log s \rceil$, margin* 1*, width* 2*, and length $O(s^2 \log s)$.*

Proof. Pick any s-state 1NFA N. Without loss of generality, say $N = ([s], \Sigma, \delta, 0)$, where $[s] := \{0, \ldots, s{-}1\}$. Let $d := \lceil \log s \rceil$. For $j = 1, \ldots, d$, let variable X_j be the set of cells where an accepting computation uses a state p whose binary code has `1` as its j-th most significant bit. Pictorially, a cell's 'bits of membership' to $X_1, \ldots, X_d$ encode the state used on it (Fig. 2e). Under this representation, the claim "the state used on cell z is p" is expressed by the floating local $\wedge$-clause:

$$\xi_p(z, \overline{X}) := \bigwedge_{j=1}^{d} \overset{p,j}{\neg}\, X_j(z)\,, \tag{7}$$

where "$\overset{p,j}{\neg}$" means either "$\neg$" or nothing, depending on whether the j-th most significant bit of the code of p is respectively `0` or `1`. We also introduce, for each $p, q \in [s]$, the set of symbols of Σ that allow a transition from p to q, and the set of symbols that allow together with $\dashv$ a transition from p to 0 (Fig. 2f):

$$\begin{aligned} \alpha_{pq} &:= \{a \in \Sigma \mid (p, a, q) \in \delta\}\,, \\ \alpha_p^{\dashv} &:= \{a \in \Sigma \mid (\exists p')[(p, a, p'), (p', \dashv, 0) \in \delta]\}\,. \end{aligned} \tag{8}$$

[3] When $1 \le r \le \rho$, we assume "$r{+}1$" for $r = \rho$ means 1; and "$r{-}1$" for $r = 1$ means ρ.

Then, our slide says that "on every two successive cells, two states p, q are used such that the symbol of the first cell allows a transition from p to q":

$$\forall\widehat{xy}\,\phi(x,y,\overline{X}) := \quad \forall\widehat{xy}\bigvee_{p,q\in[s]}[\ \xi_p(x,\overline{X}) \wedge \alpha_{pq}(x) \wedge \xi_q(y,\overline{X})\ ]\,. \tag{9}$$

Our anchor says that "on the two outer cells, two states p, q are used such that (i) 0 can reach p on $\vdash$ and (ii) the last symbol and $\dashv$ allow q to reach 0":

$$\psi(\overline{X}) := \bigvee_{(0,\vdash,p)\in\delta,\,q\in[s]}[\ \xi_p(+1,\overline{X}) \wedge \xi_q(-1,\overline{X}) \wedge \alpha_q^{\dashv}(-1)\ ]\,. \tag{10}$$

Easily, the resulting $\text{MSO}^+_\Sigma[\mathsf{S},\mathbb{Z}^*]$ sentence $\varphi := \exists\overline{X}[\psi(\overline{X}) \wedge \forall\widehat{xy}\,\phi(x,y,\overline{X})]$ is an EAS in s^2-DNF, of depth d, margin 1, width 2, and length $O(s^2 d)$. Moreover, one easily verifies that N accepts w iff $w \models \varphi$, for all non-empty $w \in \Sigma^*$. $\square$

Lemma 4. *Every s-state* SNFA *has an* EMAS *in $O(s^4)$-*DNF*, of multiplicity $<2s$, depth $O(\log s)$, margin 1, width 2, and length $O(s^5 \log s)$.*

Proof. Pick any s-state SNFA N. Without loss of generality, say $N = ([s], \Sigma, \delta, 0)$. By Lemma 1, there is an equivalent ρ-layer SNFA $\tilde{N} = ([\tilde{s}], \Sigma, \tilde{\delta}, 0)$, for $\rho < 2s$ and $\tilde{s} = O(s^2)$. Generalizing Lemma 3, we build a sentence for $\tilde{N}$. Let $d := \lceil \log \tilde{s} \rceil$.

For each $r = 1, \dots, \rho$, we use the variables $\overline{X}_r := X_{r,1}, \dots, X_{r,d}$ to describe (the binary codes of) the states along the r-th pass of an accepting computation of $\tilde{N}$. (So, $X_{r,j}$ is the set of cells where the r-th pass uses a state whose binary code has $\mathtt{1}$ as its j-th bit.) The claim "the state used by the r-th pass on cell z is p" is now expressed by $\xi_p(z, \overline{X}_r)$, the floating local $\wedge$-clause of (7) with each X_j replaced by $X_{r,j}$. Generalizing (8), we also define for each $p, q \in [\tilde{s}]$ the sets of symbols that allow (alone, with $\vdash$, or with $\dashv$) a transition from p to q (Fig. 2f):

$$\begin{aligned} \alpha_{pq} &:= \{a \in \Sigma \mid (p,a,q) \in \tilde{\delta}\}\,, \\ \alpha_{pq}^{\vdash} &:= \{a \in \Sigma \mid (\exists p')[(p,a,p'),(p',\vdash,q) \in \tilde{\delta}]\}\,, \\ \alpha_{pq}^{\dashv} &:= \{a \in \Sigma \mid (\exists p')[(p,a,p'),(p',\dashv,q) \in \tilde{\delta}]\}\,. \end{aligned} \tag{8s}$$

Then, the r-th float generalizes that of (9) to describe a step of the r-th pass:

$$\phi_r(x,y,\overline{X}_r) := \begin{cases} \bigvee_{p,q\in[\tilde{s}]}[\ \xi_p(x,\overline{X}_r) \wedge \alpha_{pq}(x) \wedge \xi_q(y,\overline{X}_r)\ ] & \text{if } r \text{ odd,} \\ \bigvee_{p,q\in[\tilde{s}]}[\ \xi_q(x,\overline{X}_r) \wedge \alpha_{pq}(y) \wedge \xi_p(y,\overline{X}_r)\ ] & \text{if } r \text{ even.} \end{cases} \tag{9s}$$

The r-th anchor describes either the last two steps of the r-th pass, if $r < \rho$:

$$\psi_r(\overline{X}_r, \overline{X}_{r+1}) := \begin{cases} \bigvee_{\substack{p,q\in[\tilde{s}] \\ q\neq 0}}[\ \xi_p(-1,\overline{X}_r) \wedge \alpha_{pq}^{\dashv}(-1) \wedge \xi_q(-1,\overline{X}_{r+1})\ ] & \text{if } r \text{ odd,} \\ \bigvee_{p,q\in[\tilde{s}]}[\ \xi_p(+1,\overline{X}_r) \wedge \alpha_{pq}^{\vdash}(+1) \wedge \xi_q(+1,\overline{X}_{r+1})\ ] & \text{if } r \text{ even;} \end{cases}$$

or the first and the last step of the entire computation, if $r = \rho$:

$$\psi_\rho(\overline{X}_\rho, \overline{X}_1) := \bigvee_{(0,\vdash,p)\in\tilde{\delta},\,q\in[\tilde{s}]}[\ \xi_p(+1,\overline{X}_1) \wedge \xi_q(-1,\overline{X}_\rho) \wedge \alpha_{q0}^{\dashv}(-1)\ ]\,. \tag{10s}$$

The final sentence $\exists\overline{X}_1 \cdots \exists\overline{X}_\rho \bigwedge_r[\psi_r(\overline{X}_r,\overline{X}_{r+1}) \wedge \forall\widehat{xy}\,\phi_r(x,y,\overline{X}_r)]$ is an EMAS in $\tilde{s}^2$-DNF, of multiplicity ρ, depth d, margin 1, width 2, and length $O(\rho\tilde{s}^2 d)$. $\square$

The next lemma says that small EAS/CNFs can be more powerful than small EAS/DNFs: indeed, even small SNFAs can be simulated by them (with just 1 core).

Lemma 5. **(i)** *Every s-state* 1NFA *has an* EAS *in* $O(s^2)$-CNF*, of depth* $\lceil \log s \rceil$*, margin* 1*, width* 2*, and length* $O(s^2 \log s)$*.* **(ii)** *Every s-state* SNFA *has an* EAS *in* $O(s^5)$-CNF*, of depth* $O(s \log s)$*, margin* 1*, width* 2*, and length* $O(s^5 \log s)$*.*

5 From Formulas to Automata

Fix an alphabet Σ and two sets of f.o. and s.o. variables V_1 and V_2. We assume all formulas in this section are over Σ and draw their variables from $V_1 \cup V_2$.

Lemma 6. *Every floating local $\wedge$-clause has a* 1*-state* 1DFA*.*

Proof. Pick any floating local $\wedge$-clause $\kappa(\overline{x}, \overline{X}) = \bigwedge_j \lambda_j$. Note that each λ_j is of the form $\alpha(x)$, $X(x)$, $\neg\alpha(x)$, or $\neg X(x)$, for some $x \in V_1$, $\alpha \subseteq \Sigma$, $X \in V_2$. Say a column $u \in \Sigma|V_1|V_2$ *passes* (*the test of*) λ_j if either $u(x)$=0 or $u(x)$=1 $\wedge\, u \models \lambda_j$, for x the one f.o. variable of λ_j. Say u *passes* $\kappa(\overline{x}, \overline{X})$ if it passes all λ_j.

Claim. For every well-formed $\hat{w} \in (\Sigma|V_1|V_2)^*$: $\hat{w} \models \kappa$ iff every $\hat{w}_i$ passes κ.

Proof. [$\Rightarrow$] Suppose $\hat{w} \models \kappa$. Pick any column $\hat{w}_i$. Pick any λ_j, and let x be its one f.o. variable. If $\hat{w}_i(x) = 0$ then $\hat{w}_i$ passes λ_j, by definition. If $\hat{w}_i(x) = 1$ then $\hat{w}_i$ passes λ_j, since $\hat{w} \models \lambda_j$ and so $\hat{w}_i \models \lambda_j$. So, $\hat{w}_i$ passes all λ_j, and thus also κ.

[$\Leftarrow$] Suppose every $\hat{w}_i$ passes κ. Pick any λ_j, and let x be its one f.o. variable. Let $i^* := \hat{w}(x)$ be the unique i with $\hat{w}_i(x) = 1$. Since $\hat{w}_{i^*}$ passes κ (as all $\hat{w}_i$ do), it passes λ_j. Since $\hat{w}_{i^*}(x) = 1$, this means $\hat{w}_{i^*} \models \lambda_j$; that is, $\hat{w}_{\hat{w}(x)} \models \lambda_j$. Hence $\hat{w} \models \lambda_j$. Since λ_j was arbitrary, we conclude $\hat{w} \models \kappa$. ⊡

Therefore, a 1DFA $M = ([1], \Sigma|V_1|V_2, \,.\,, 0)$ simply scans its input $\hat{w}$ checking that every column $\hat{w}_i$ passes κ. If any of them does not, then M just hangs. ☐

Lemma 7. *Every local $\wedge$-clause of margin τ has a $(\tau{+}1)^2$-state* 1NFA*.*

Proof. Pick any local $\wedge$-clause $\kappa(\overline{x}, \overline{X})$ of margin τ. Note that each literal of κ is of the form $\alpha(t)$, $X(t)$, $\neg\alpha(t)$, or $\neg X(t)$, for some $t \in V_1 \cup \{\pm 1, \ldots, \pm\tau\}$, $\alpha \subseteq \Sigma$, $X \in V_2$. Hence, κ is the conjunction of three smaller $\wedge$-clauses,

$$\kappa(\overline{x}, \overline{X}) = \kappa_{\mathrm{L}}(\overline{X}) \wedge \kappa_{\mathrm{f}}(\overline{x}, \overline{X}) \wedge \kappa_{\mathrm{R}}(\overline{X}) \,,$$

whose terms are all in $\{+1, \ldots, +\tau\}$, in V_1, and in $\{-1, \ldots, -\tau\}$, respectively. We know (Lemma 6) that κ_{f} has a 1-state 1DFA M_{f}, and we show (below) that κ_{L} has a $(\tau{+}1)$-state 1DFA M_{L} and κ_{R} has a $(\tau{+}1)$-state 1NFA N_{R}. Hence, the standard cartesian product of M_{L}, M_{f}, N_{R} is a $(\tau{+}1)^2$-state 1NFA for κ.

To build M_{L}, we first assume that κ_{L} contains at least one occurence of every $c \in \{+1, \ldots, +\tau\}$ (if some c is missing, just replace κ_{L} with $\kappa_{\mathrm{L}} \wedge \Sigma(c)$). Then κ_{L} is a conjunction of exactly τ smaller $\wedge$-clauses,

$$\kappa_{\mathrm{L}}(\overline{X}) = \kappa_1(\overline{X}) \wedge \kappa_2(\overline{X}) \wedge \cdots \wedge \kappa_\tau(\overline{X}) \,,$$

where the only term in κ_c is c. Easily then, $M_{\rm L} := ([\tau{+}1], \Sigma|V_1|V_2, ., 0)$ simply checks that the first τ input columns "satisfy" respectively $\kappa_1, \dots, \kappa_\tau$.

To build $N_{\rm R}$, we similarly write $\kappa_{\rm R}$ as a conjunction of τ smaller $\wedge$-clauses,

$$\kappa_{\rm R}(\overline{X}) = \kappa_{-\tau}(\overline{X}) \wedge \cdots \wedge \kappa_{-2}(\overline{X}) \wedge \kappa_{-1}(\overline{X}) \, ,$$

where again the only term in κ_c is c. Easily then, $N_{\rm R} := ([\tau{+}1], \Sigma|V_1|V_2, ., \tau)$ starts by consuming input columns until it nondeterministically guesses when it has reached the τ-th rightmost one. Then it checks that the next τ columns "satisfy" respectively $\kappa_{-\tau}, \dots, \kappa_{-1}$, and are indeed followed by $\dashv$. ☐

Lemma 8. *Every local m-DNF of margin τ has an $m(\tau{+}1)^2$-state 1NFA.*

Proof. On $\vdash$, a 1NFA $N = ([m]\times[\tau{+}1]\times[1]\times[\tau{+}1], \Sigma|V_1|V_2, ., (0,0,0,0))$ guesses which of the m $\wedge$-clauses will be satisfied, and goes on to verify it by simulating the corresponding $(\tau{+}1)^2$-state cartesian 1NFA given by Lemma 7. ☐

Lemma 9. *Every sliding m-DNF of width k has an $(m{+}1)^{k-1}$-state 1NFA.*

Proof. Pick any floating local m-DNF $\phi(\overline{x}, \overline{X}) = \bigvee_{j=1}^m \kappa_j$, where $\overline{x} = x_1, \dots, x_k$ and each κ_j is a floating local $\wedge$-clause. We may assume each κ_j contains at least one occurence of every x_r (if some x_r is missing, just replace κ_j with $\kappa_j \wedge \Sigma(x_r)$) and is thus the conjunction of exactly k smaller $\wedge$-clauses,

$$\kappa_j(\overline{x}, \overline{X}) = \kappa_{j,1}(x_1, \overline{X}) \wedge \kappa_{j,2}(x_2, \overline{X}) \wedge \cdots \wedge \kappa_{j,k}(x_k, \overline{X}) \, ,$$

where x_r is the only term in $\kappa_{j,r}$. Hence, an n-long well-formed word $\hat{w}$ satisfies

$$\forall \widehat{x_1 \cdots x_k}\, \phi(\overline{x}, \overline{X}) \;\; = \;\; \forall \widehat{x_1 \cdots x_k} \bigvee_{j=1}^m \bigwedge_{r=1}^k \kappa_{j,r}(x_r, \overline{X})$$

if at every stop $i = 1, \dots, n{-}k{+}1$ of a sliding k-wide window there is a clause κ_j such that each individual column $\hat{w}_{i+r-1}$ in the window "satisfies" the respective sub-clause $\kappa_{j,r}$ (in the formal sense that $\hat{w}_{i+r-1}[x_r/1] \models \kappa_{j,r}(x_r, \overline{X})$). In other words, we ask for a sequence $j_1, j_2, \dots, j_{n-k+1}$ of choices of clauses such that each

	$\hat{w}_1$	$\hat{w}_2$	$\hat{w}_3$	$\hat{w}_4$	$\hat{w}_5$	$\hat{w}_6$	$\hat{w}_7$	$\hat{w}_8$
1	$j_1, 1$	$j_1, 2$	$j_1, 3$	$j_1, 4$				
2		$j_2, 1$	$j_2, 2$	$j_2, 3$	$j_2, 4$			
3			$j_3, 1$	$j_3, 2$	$j_3, 3$	$j_3, 4$		
4				$j_4, 1$	$j_4, 2$	$j_4, 3$	$j_4, 4$	
5					$j_5, 1$	$j_5, 2$	$j_5, 3$	$j_5, 4$

(e.g., when $n{=}8$, $k{=}4$)

column $\hat{w}_i$ (now $i = 1, \dots, n$) "satisfies" every relevant sub-clause $\kappa_{j_t,r}$ that we get by ranging $r = 1, \dots, k$ and keeping $t + (r{-}1) = i$ (as well as $1 \le t \le n{-}k{+}1$, if $\hat{w}_i$ is among the first $k{-}1$ or last $k{-}1$ columns).

To check this condition, a 1NFA $N = ([m{+}1]^{k-1}, \Sigma|V_1|V_2, ., (0,\dots,0))$ guesses the choices j_i one by one, remebering only the last $k{-}1$ of them at every step. Specifically, N reads $\hat{w}_i$ in state $(j_{i-k+1}, \dots, j_{i-2}, j_{i-1})$; it then guesses j_i and checks that $w_i[x_r/1] \models \kappa_{j_t,r}$ for every $r = 1, \dots, k$ and $t = i{-}r{+}1$; if any check fails, N hangs; otherwise, it moves to $\hat{w}_{i+1}$ in state $(j_{i-k+2},\dots,j_{i-1},j_i)$. Special care is needed on the first $k{-}1$ columns: there, N uses states with 0s in ≥ 1 of the leftmost components to denote that there is no corresponding sub-clause to check. Likewise, during the last $k{-}1$ columns, N uses states with 0s in ≥ 1 of the rightmost components. Of course, N cannot know when the $k{-}1$-st rightmost column has been reached; so, at every step it spawns an extra branch, which guesses that the time is right and expects to read $\dashv$ after exactly $k{-}1$ steps. $\square$

Lemma 10. *Every* EAS m-DNF *of margin* τ *and width* k *has an equivalent* 1NFA *with* $O(m^k\tau^2)$ *states.*

Proof. Take the cartesian product N of the two 1NFAs for the anchor (Lemma 8) and the slide (Lemma 9). Then, for the existential quantification, just drop all s.o. variable information from the transitions of N (see also Lemma 11). $\square$

For EMAS, we need a restriction of RNFAs which interact well with existential quantifiers. We first define this restriction and prove the associated interaction.

Let $N=(S,\Sigma|\overline{X},\delta,\,.\,)$ be a RNFA. A transition $(p,u,q)\in\delta$ *ignores* X_j if "it does not read it": either $u\in\{\vdash,\dashv\}$; or $u\in\Sigma|\overline{X}$ and also $(p,\tilde{u},q)\in\delta$, where $\tilde{u}$ the column derived from u by complementing $u(X_j)$. We say N is *stratified* if "each X_j is read in at most one pass": (i) N is layered, and (ii) for ρ the number of layers, there is a partition $\overline{X}_1,\dots,\overline{X}_\rho$ of $\overline{X}$ such that every transition between states of layer r ignores all $\overline{X}_t$ with $t\neq r$, for all $r=1,\dots,\rho$.

Lemma 11. *If* $\varphi(\overline{X})$ *has a stratified* s*-state* RNFA, *then* $\exists\overline{X}\varphi(\overline{X})$ *has a layered* s*-state* RNFA.

We now continune our build-up towards multicore existential anchor-slides.

Lemma 12. *If every* $\psi_r(\overline{X}_r,\overline{X}_{r+1})$ *is an anchored local* m-DNF *of margin* τ, *then* $\bigwedge_{r=1}^{\rho}\psi_r(\overline{X}_r,\overline{X}_{r+1})$ *has a* $\rho m^3(\tau{+}1)^2$*-state* RNFA *stratified by* $\overline{X}_1,\dots,\overline{X}_\rho$.

Proof. Let $\varphi(\overline{X}):=\bigwedge_{r=1}^{\rho}\psi_r(\overline{X}_r,\overline{X}_{r+1})$. To check $\hat{w}\models\varphi$, a ρ-layer RNFA may use its r-th pass to check $\hat{w}\models\psi_r$ by simulating the $m(\tau{+}1)^2$-state 1NFA given for ψ_r by Lemma 8. But this easy RNFA is not stratified, so we must work more.

We know $\psi_r=\bigvee_{j=1}^{m}\kappa_{r,j}(\overline{X}_r,\overline{X}_{r+1})$, where each $\kappa_{r,j}$ is an anchored local $\wedge$-clause of margin τ. Since every literal uses ≤ 1 s.o. variable, we can split $\kappa_{r,j}$

$$\kappa_{r,j}(\overline{X}_r,\overline{X}_{r+1}) \;=\; \mu_{r,j}(\overline{X}_r)\;\wedge\;\nu_{r,j}(\overline{X}_{r+1})$$

into two sub-clauses which use only one group of variables each. Therefore,

$$\varphi(\overline{X})=\textstyle\bigwedge_{r=1}^{\rho}\bigvee_{j=1}^{m}[\,\mu_{r,j}(\overline{X}_r)\;\wedge\;\nu_{r,j}(\overline{X}_{r+1})\,]\,.$$

So, $\hat{w}\models\varphi$ iff for each r there is a choice j such that $\hat{w}\models\mu_{r,j}\wedge\nu_{r,j}$. Viewed differently, $\hat{w}\models\varphi$ iff there exists a sequence of choices $j_1,\dots,j_\rho$ such that the following conjunction on the left becomes true:

$$\begin{array}{l} \hat{w}\models\mu_{1,j_1}(\overline{X}_1)\wedge\nu_{1,j_1}(\overline{X}_2)\\ \&\ \hat{w}\models\mu_{2,j_2}(\overline{X}_2)\wedge\nu_{2,j_2}(\overline{X}_3)\\ \&\ \quad\vdots\\ \&\ \hat{w}\models\mu_{\rho,j_\rho}(\overline{X}_\rho)\wedge\nu_{\rho,j_\rho}(\overline{X}_1) \end{array} \quad\Longleftrightarrow\quad \begin{array}{l} \hat{w}\models\mu_{1,j_1}(\overline{X}_1)\wedge\nu_{\rho,j_\rho}(\overline{X}_1)\\ \&\ \hat{w}\models\mu_{2,j_2}(\overline{X}_2)\wedge\nu_{1,j_1}(\overline{X}_2)\\ \&\ \quad\vdots\\ \&\ \hat{w}\models\mu_{\rho,j_\rho}(\overline{X}_\rho)\wedge\nu_{\rho-1,j_{\rho-1}}(\overline{X}_\rho) \end{array}$$

Now, this conjunction is equivalent to the one on the right, which just "cyclically shifts down" the column of the ν_{r,j_r} to align the groups of s.o. variables. Hence, $\hat{w} \models \varphi$ iff there exist $j_1, \ldots, j_\rho$ such that $\hat{w} \models (\nu_{r-1,j_{r-1}} \wedge \mu_{r,j_r})(\overline{X}_r)$ for all r.

Our stratified RNFA N uses this last condition. Also, for each $r = 1, \ldots, \rho$ and $j, j' = 1, \ldots, m$, it uses the $(\tau{+}1)^2$-state 1NFA $N[r, j, j']$ over $\Sigma|\overline{X}_r$ given by Lemma 7 for the margin-τ anchored local $\wedge$-clause $(\nu_{r-1,j} \wedge \mu_{r,j'})(\overline{X}_r)$. The machine starts by guessing and storing j_ρ. It then performs ρ passes. The r-th pass starts by recalling j_{r-1} from the previous pass (or j_ρ from the starting guess, if $r = 1$) and guessing j_r (or recalling j_ρ from the starting guess, if $r = \rho$). Then, N simulates $N[r, j_{r-1}, j_r]$ to check $\hat{w} \models (\nu_{r-1,j_{r-1}} \wedge \mu_{r,j_r})(\overline{X}_r)$. If at the end of the last pass all simulations have accepted, then N accepts. This algorithm can be implemented with states of the form $(j^*; r, j, j'; p)$ where $1 \leq j^*, j, j' \leq m$, $1 \leq r \leq \rho$, and $p \in [\tau{+}1]^2$, meaning that: the starting guess for j_ρ was j^*; the guesses for j_{r-1},j_r were j,j'; and the current r-th pass is at state p in simulating $N[r, j_{r-1}, j_r]$. This is indeed a stratified RNFA, with $\rho{\cdot}m^3{\cdot}(\tau{+}1)^2$ states. □

Lemma 13. *Every* EMAS m-DNF *of multiplicity* ρ*, margin* τ *and width* k *has a* ρ*-layer* RNFA *with* $O(\rho{\cdot}m^{k+2}\tau^2)$ *states.*

Proof. Let $\exists\overline{X}_1 \cdots \exists\overline{X}_\rho \bigwedge_{r=1}^{\rho}[\psi_r(\overline{X}_r, \overline{X}_{r+1}) \wedge \forall\widehat{x_1 \cdots x_k}\phi_r(\overline{X}_r)]$ be the given EAS. Easily, this is equivalent to $\exists\overline{X}\varphi(\overline{X})$, where $\varphi := [\bigwedge_{r=1}^{\rho}\psi_r] \wedge \bigwedge_{r=1}^{\rho}[\forall\widehat{\overline{x}}\phi_r]$. Let N_{a} be the stratified RNFA with $\rho m^3(\tau{+}1)^2$ states given by Lemma 12 for $\bigwedge_{r=1}^{\rho}\psi_r$. For each $r = 1, \ldots, \rho$, let N_r be the 1NFA of $(m{+}1)^{k-1}$ states given by Lemma 9 for $\forall\widehat{\overline{x}}\phi_r$. Now, a RNFA N for φ can just simulate all of $N_{\mathrm{a}}, N_1, \ldots, N_\rho$ and accept if they all do. The simulation is possible because N_{a} is stratified by $\overline{X}_1, \ldots, \overline{X}_\rho$ and each N_r is defined over $\Sigma|\overline{X}_r$; so, each N_r can be simulated during the r-th pass of the simulation of N_{a}. Essentially, we build N by replacing each layer of N_{a} by its cartesian product with the corresponding N_r: each state is of the form $(j^*; r, j, j'; p; q)$, meaning that the current r-th pass is at state $(j^*; r, j, j'; p)$ in simulating N_{a} (cf. proof of Lemma 12) and at state q in simulating N_r. Easily, N is also stratified by $\overline{X}_1, \ldots, \overline{X}_\rho$ and uses $\rho m^3(\tau{+}1)^2{\cdot}(m{+}1)^{k-1}$ states. □

6 Conclusion

Refining Büchi's Theorem, we established analogs of Fagin's Theorem for small *one-way*, *rotating*, and *sweeping* nondeterministic finite automata. We thus took a first step towards what one could call a '*descriptive minicomplexity theory*'.

We are still missing a descriptive chracterization of 2N. Similarly, one can ask for such characterizations for all other major minicomplexity classes (cf. [6]).

More broadly, one can ask for other tests of the premise of minicomplexity, that many phenomena of standard complexity theory emerge already at this level. E.g., complexity theory has parallels studying *function problems* [8, §10.3] and *real computation* [2]: are there such parallels for minicomplexity as well?

Finally, we suggest some notation that may facilitate discussions like ours. For three classes of functions $\mathcal{D}$,$\mathcal{T}$,$\mathcal{K}$, let the class EAS/DNF$[\mathcal{D}, \mathcal{T}, \mathcal{K}]$ consist of every family of problems solvable by a family $(\varphi_h)_{h\geq 1}$ of small EAS/DNFs of depth $d(h)$, margin $\tau(h)$, and width $k(h)$, for some $d \in \mathcal{D}$, $\tau \in \mathcal{T}$, $k \in \mathcal{K}$. Define similarly the classes $\mathsf{EAS/CNF}$, $\mathsf{EMAS/DNF}$, $\mathsf{EMAS/CNF}$. Then Theorems 1 and 2 are:

$$\mathsf{1N} = \mathsf{EAS/DNF}[\mathsf{log}, 1, 2] = \mathsf{EAS/DNF}[*, \mathsf{poly}, \mathsf{const}]$$
$$\mathsf{RN} = \mathsf{SN} = \mathsf{EMAS/DNF}[\mathsf{log}, 1, 2] = \mathsf{EMAS/DNF}[*, \mathsf{poly}, \mathsf{const}]\,,$$

for the obvious meaning of $1, 2, \mathsf{const}, \mathsf{log}, \mathsf{poly}$, and for '$*$' denoting 'maximum possible' (here: poly). Moreover, for 2^{1N} the class for exponential-size 1NFAs [6] and for the obvious meaning of exp, we can prove the relationships

$$\mathsf{1N} \left\{ \begin{array}{ccc} \subsetneq & \mathsf{EAS/CNF}[\mathsf{log}, 1, 2] & \subseteq \\ {\scriptstyle (a)} & & \\ {\scriptstyle (b)} & & {\scriptstyle (c)} \\ \subsetneq & \mathsf{RN} = \mathsf{SN} & \subsetneq \end{array} \right\} \mathsf{EAS/CNF}[*, 1, 2] \subseteq \mathsf{EAS/CNF}[*, \mathsf{exp}, *] \overset{(d)}{\subseteq} 2^{\mathsf{1N}}$$

where (d) uses easy variants of Lemmas 6–10; (b) is known [7]; and (a), (c) use Lemma 5. The strictness of (a) uses the problem "Given two sets $\alpha, \beta \subseteq [h]$, check that $\alpha \subseteq \beta$", which is in $\mathsf{EAS/CNF}[0, 1, 0]$ (easy) but not in $\mathsf{1N}$ (a 'fooling set' argument). The strictness of (c) uses the problem "Given $w \in \{\mathtt{a}\}^*$, check that $|w| = 2^h$", which is in $\mathsf{EAS/CNF}[*, 1, 2]$ (just use s.o. variables as in Fig. 2g, to increment an h-bit counter from 0 to 2^h-1) but not in $\mathsf{2N}$ [1, Fact 5.2].

Acknowledgment Many thanks to Thomas Colcombet and Achim Blumensath for several very helpful discussions during the preparation of this work.

References

1. J.-C. Birget. Two-way automata and length-preserving homomorphisms. *Mathematical Systems Theory*, 29:191–226, 1996.
2. L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation*. Springer, 1997.
3. R. J. Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 6(1-6):66–92, 1960.
4. R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of Computation*, volume VII of *AMS-SIAM Symposia in Applied Mathematics*, pages 43–73, 1974.
5. N. Immerman. *Descriptive complexity*. Springer-Verlag, 1998.
6. C. Kapoutsis. Size complexity of two-way finite automata. In *Proceedings of DLT*, pages 47–66, 2009.
7. C. Kapoutsis, R. Královič, and T. Mömke. Size complexity of rotating and sweeping automata. *Journal of Computer and System Sciences*, 78(2):537–558, 2012.
8. C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
9. K. Reinhardt. The complexity of translating logic to finite automata. In E. Grädel, W. Thomas, and T. Wilke, editors, *Automata, logics, and infinite games*, volume 2500 of *Lecture Notes in Computer Science*, pages 231–238. Springer-Verlag, 2002.
10. W. J. Sakoda and M. Sipser. Nondeterminism and the size of two-way finite automata. In *Proceedings of STOC*, pages 275–286, 1978.
11. W. Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25(3):360–376, 1982.