

Satisfiability via Smooth Pictures

Mateus de Oliveira Oliveira^(✉)

Institute of Mathematics - Czech Academy of Sciences,
Žitná 25, 115 67 Praha 1, Czech Republic
mateus.oliveira@math.cas.cz

Abstract. A picture over a finite alphabet Γ is a matrix whose entries are drawn from Γ . Let $\pi : \Sigma \rightarrow \Gamma$ be a function between finite alphabets Σ and Γ , and let $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ . Given a picture N over Γ , the picture satisfiability problem consists in determining whether there exists a picture M over Σ such that $\pi(M_{ij}) = N_{ij}$, and such that the constraints imposed by V and H on adjacent vertical and horizontal positions of M are respectively satisfied. This problem can be easily shown to be NP-complete. In this work we introduce the notion of s -smooth picture. Our main result states the satisfiability problem for s -smooth pictures can be solved in time polynomial on s and on the size of the input picture. With each picture N , one can naturally associate a CNF formula $F(N)$ which is satisfiable if and only if N is satisfiable. In our second result, we define an infinite family of unsatisfiable pictures which intuitively encodes the pigeonhole principle. We show that this family of pictures is polynomially smooth. In contrast we show that the formulas which naturally arise from these pictures are hard for bounded-depth Frege proof systems. This shows that there are families of pictures for which our algorithm for the satisfiability for smooth pictures performs exponentially better than certain classical variants of SAT solvers based on the technique of conflict-driven clause-learning (CDCL).

Keywords: Smooth pictures · Bounded frege proof systems · Pigeonhole principle

1 Introduction

A picture over an alphabet Γ is a matrix whose elements are drawn from Γ . Let $\pi : \Sigma \rightarrow \Gamma$ be a function between finite alphabets Σ and Γ , and let $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ . In the picture satisfiability problem we are given an $m \times n$ picture N over Γ , and the goal is to determine whether there exists an $m \times n$ picture M over Σ such that the following conditions are satisfied. First, $N_{i,j} = \pi(M_{i,j})$ for each $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$; second, each two consecutive vertical entries of M belong to V ; and third, each two consecutive horizontal entries of M belong to H . If such a picture M exists, we say that M is a (π, V, H) -solution for N . Variations of the picture satisfiability problem have been studied since the seventies in the context of pattern recognition [21, 23], image

processing [5, 21], tiling systems [14, 22] and formal language theory [8, 11, 15, 21]. In this work, we introduce the notion of s -smooth picture. Our main result states that one can determine whether an s -smooth picture N has a (π, V, H) -solution in time $O(|\Sigma|^{e(\pi, V)} \cdot s^{e(\pi, V)} \cdot m \cdot n)$. Here, $e(\pi, V) \leq |\Sigma \times \Sigma|$ is a parameter that does not depend on the size of the picture. As an implication, we have that if \mathcal{F} is a family of pictures such that each $m \times n$ picture in \mathcal{F} is $poly(m, n)$ -smooth, then the picture satisfiability problem for this family can be solved in polynomial time.

The pigeonhole principle states that if m pigeons are placed into $m - 1$ holes, then at least one hole contains two pigeons. In a influential work, Haken showed that a family of propositional formulas H_m encoding the pigeonhole principle requires resolution refutations of exponential size [2, 9]. Following Haken's work, the pigeonhole principle and many of its variants have played a central role in propositional proof complexity theory [3, 19]. In particular, it has been shown that refutations of the formulas H_m in constant-depth proof systems must have exponential size [12, 13, 18]. In our second result, we define an infinite family of pictures $P^{m, m-1}$ encoding the pigeonhole principle. Subsequently, we show that this family of pictures is $poly(m)$ -smooth. This implies that our algorithm for smooth pictures is able to detect the unsatisfiability of the pictures $P^{m, m-1}$ in polynomial time.

For each fixed triple (π, V, H) and each picture N one can derive a natural constant-width CNF formula $F(N)$ which is satisfiable if and only if N has a (π, V, H) -solution. Our third result states that the family of formulas $F(P^{m, m-1})$ derived from the pigeonhole pictures is still hard for constant depth Frege proof systems. The proof of this result follows by application of routine techniques to show that small refutations of $F(P^{m, m-1})$ imply small refutations of the formulas H_m . This last result establishes a point of comparison between our algorithm for the satisfiability of smooth pictures, and SAT solvers based on the technique of conflict-driven clause-learning (CDCL) [7, 16]. Indeed, it has been shown that certain variants of CDCL-based SAT solvers, such as those introduced in [7, 16], are equivalent in power to resolution-based proof systems [1, 4, 10, 17]. Since bounded-depth Frege is stronger than the resolution proof system, our third result implies that the formulas $F(P^{m, m-1})$ derived from the pigeonhole pictures are hard for such variants of CDCL SAT solvers.

The remainder of the paper is organized as follows. Next, in Sect. 2 we introduce some notation and some basic results concerning leveled finite automata. Subsequently, in Sect. 3 we formally define the picture satisfiability problem and introduce the notion of s -smooth picture. In Sect. 4 we state and prove our main theorem, namely, that the satisfiability problem for pictures can be solved in time polynomial on its smoothness. In Sect. 5 we define the family of pigeonhole pictures and show that this family is polynomially smooth. In Sect. 6 we define a natural translation from pictures to constant-width CNF formulas, and in Sect. 7 we show that CNF formulas derived from the pigeonhole pictures according to our translation require exponential bounded-depth Frege proofs.

2 Preliminaries

A *leveled nondeterministic finite automaton* (LNFA) over an alphabet Σ is a triple $\mathcal{A} = (Q, \Sigma, \mathfrak{R})$ where Q is a set of states partitioned into subsets Q_0, \dots, Q_n and $\mathfrak{R} \subseteq \bigcup_{i=1}^n Q_{i-1} \times \Sigma \times Q_i$ is a transition relation. The states in Q_0 are the initial states of \mathcal{A} , while Q_n is the set of final states of \mathcal{A} . For each $i \in \{0, \dots, n\}$, we say that Q_i is the i -th level of \mathcal{A} . The size of \mathcal{A} is defined as $|\mathcal{A}| = |Q| + |\mathfrak{R}|$. We say that a string $w \in \Sigma^n$ is accepted by \mathcal{A} if there exists a sequence of transitions

$$q_0 \xrightarrow{w_1} q_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} q_n$$

such that $q_i \in Q_i$ for each i in $\{0, 1, \dots, n\}$, and $(q_{i-1}, w_i, q_i) \in \mathfrak{R}_i$ for each i in $\{1, \dots, n\}$. We denote by $\mathcal{L}(\mathcal{A})$ the set of all strings accepted by \mathcal{A} . We note that all strings accepted by \mathcal{A} have size n , i.e., $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^n$. We say that \mathcal{A} is a *leveled deterministic finite automaton* (LDFA) if Q_0 has a unique state q_0 , and for each state $q \in Q$, and each symbol $a \in \Sigma$ there exists at most one state $q' \in Q$ such that $(q, a, q') \in \mathfrak{R}$. Let $\pi : \Sigma \rightarrow \Gamma$ be a function and let $w = w_1 w_2 \dots w_n$ be a string in Σ^* . We denote by $\pi(w) = \pi(w_1) \pi(w_2) \dots \pi(w_n)$ the image of w under π .

Lemma 2.1 (Synchronized Product of Automata). *Let \mathcal{A} and \mathcal{A}' be LNFA over Σ accepting strings of size n . Let $V \subseteq \Sigma \times \Sigma$ be a binary relation over Σ . Then one can construct in time $|\mathcal{A}| \cdot |\mathcal{A}'|$ an LNFA $\mathcal{A} \otimes_V \mathcal{A}'$ accepting the following language over $\Sigma \times \Sigma$.*

$$\mathcal{L}(\mathcal{A} \otimes_V \mathcal{A}') = \{(w_1, w'_1)(w_2, w'_2) \dots (w_n, w'_n) \mid w \in \mathcal{L}(\mathcal{A}), w' \in \mathcal{L}(\mathcal{A}'), (w_i, w'_i) \in V\}.$$

3 Pictures

An (m, n) -picture over a finite set of symbols Σ is an $m \times n$ matrix whose entries are drawn from Σ . Let $\pi : \Sigma \rightarrow \Gamma$ be a function between finite sets of symbols Σ and Γ , and let $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ . Finally, let N be an (m, n) -picture over Γ . We say that an (m, n) -picture M over Σ is a (π, V, H) -solution for N if the following conditions are satisfied.

1. $N_{i,j} = \pi(M_{i,j})$ for each $i \in \{1, \dots, m\}$ and each $j \in \{1, \dots, n\}$.
2. $(M_{i,j}, M_{i,j+1})$ belongs to H for each $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n-1\}$.
3. $(M_{i,j}, M_{i+1,j})$ belongs to V for each $i \in \{1, \dots, m-1\}$ and $j \in \{1, \dots, n\}$.

Intuitively, the symbols in Σ may be regarded as colored versions of symbols in Γ . For each symbol $a \in \Gamma$, the set $\pi^{-1}(a) \subseteq \Sigma$ is the set of colored versions of a . Thus M is a (π, V, H) -solution for N if M is a colored version of N and the entries in M respect the vertical and horizontal constraints imposed by V and H respectively. If N admits a (π, V, H) -solution, then we say that N is *satisfiable* (with respect to (π, V, H)). Otherwise, we say that N is *unsatisfiable*.

Definition 3.1 (Picture Satisfiability Problem). *Let $\pi : \Sigma \rightarrow \Gamma$ be a function and $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ . Given an (m, n) -picture N over Γ , is N satisfiable with respect to (π, V, H) ?*

3.1 Smooth Pictures

Let $[n] = \{1, \dots, n\}$. We assume that the set $[m] \times [n] = \{(i, j) \mid i \in [m], j \in [n]\}$ is endowed with a lexicographic ordering $<$, which sets $(i, j) < (i', j')$ if either $i < i'$, or $i = i'$ and $j < j'$. We write $(i, j) \leq (i', j')$ to denote that $(i, j) = (i', j')$ or $(i, j) < (i', j')$. For each $(i, j) \in [m] \times [n]$, we let

$$S(m, n, i, j) = \{(i', j') \mid (i', j') \leq (i, j)\}$$

be the set of all positions in $[m] \times [n]$ that are (lexicographically) smaller than or equal to (i, j) .

Let $\pi : \Sigma \rightarrow \Gamma$ be a function, and $V, H \subseteq \Sigma \times \Sigma$ be a binary relation over Σ . We say that a function $M : S(m, n, i, j) \rightarrow \Sigma$ is an (i, j) -partial (π, V, H) -solution for N if the following conditions are satisfied.

1. $(M_{i',j'}, M_{i',j'+1}) \in H$ for each $(i', j'), (i', j' + 1)$ in $S(m, n, i, j)$.
2. $(M_{i',j'}, M_{i'+1,j'}) \in V$ for each $(i', j'), (i' + 1, j')$ in $S(m, n, i, j)$.

Note that for simplicity we write $M_{i,j}$ in place of $M(i, j)$ to designate an entry of M . Intuitively, an (i, j) -partial (π, V, H) -solution for N is a function that colors the positions of N up to the entry (i, j) with elements from Σ in such a way that the vertical and horizontal constraints imposed by V and H respectively are respected. If (i, j_1) and (i, j_2) are positions in $S(m, n, i, j)$ with $j_1 < j_2$, then we let $M_{i,[j_1,j_2]} = M_{i,j_1} \dots M_{i,j_2}$ be the string formed by all entries at the i -th row of M between positions j_1 and j_2 . Now let $(i, j) \in S(m, n, i, j)$ with $(i, j) \geq (1, n)$. The (i, j) -boundary of M is defined as follows.

$$\partial_{i,j}(M) = \begin{cases} M_{i,[1,n]} & \text{if } j = n. \\ M_{i,[1,j]} \cdot M_{i-1,[j+1,n]} & \text{if } j < n. \end{cases} \quad (1)$$

In other words, if $j = n$, then $\partial(M)$ is the string consisting of all entries in the i -th row of M . On the other hand, if $j < n$, then $\partial(M)$ is obtained by concatenating the string corresponding to the first j entries of row i with the last $(n - j)$ entries of row $(i - 1)$. The notion of boundary of a partial solution is illustrated in Fig. 1.

| | | | | | | |
|-----|---|---|-----|---|---|---|
| | | | j | | | |
| | a | b | c | a | c | b |
| | c | b | a | b | c | a |
| i | c | a | b | c | | |
| | | | | | | |
| | | | | | | |

Fig. 1. An (i, j) -partial solution M where $i = 3$ and $j = 4$. The grey entries form the boundary of M . Therefore $\partial_{i,j}(M) = cabccab$.

Below, we define the (i, j) -feasibility boundary of a picture N over Γ as the set of (i, j) -boundaries of partial solutions of N .

Definition 3.2 (Feasibility Boundary). Let $\pi : \Sigma \rightarrow \Gamma$ be a function, $V, H \subseteq \Sigma \times \Sigma$, and N be a (m, n) -picture over Γ . The (i, j) -feasibility boundary of N with respect to (π, V, H) , denoted by $\partial_{i,j}(N, \pi, V, H)$, is defined as follows.

$$\partial_{i,j}(N, \pi, V, H) = \{\partial_{i,j}(M) \mid M \text{ is an } (i, j)\text{-partial } (\pi, V, H)\text{-solution for } N\}. \quad (2)$$

Note that N has a (π, V, H) -solution if and only if its (m, n) -feasibility boundary $\partial_{m,n}(N, \pi, V, H)$ is non-empty. Below we define the notion of *smooth picture*.

Definition 3.3 (Smooth Picture). Let $\pi : \Sigma \rightarrow \Gamma$ be a function, and $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ . We say that an (m, n) -picture N over Γ is *s-smooth* if for each $(i, j) \geq (1, n)$, the set $\partial_{i,j}(N, \pi, V, H)$ can be represented by an LDFA of size at most s .

Intuitively, a picture N is smooth if each of its feasibility boundaries can be efficiently represented. The main goal of this work is to show that the automata representing the boundaries of feasibility of a picture N can actually be constructed in time polynomial on the smoothness parameter s and on the size of N . Additionally, once these automata are constructed, one can proceed to actually construct a solution for N if such a solution exists.

4 Satisfiability of Smooth Pictures in Polynomial Time

In this section we show that the satisfiability problem for smooth pictures can be solved in time polynomial on the size of the picture and on its smoothness parameter. Let $\pi : \Sigma \rightarrow \Gamma$ be a function and $V \subseteq \Sigma \times \Sigma$ be a binary relation over Σ . We let $e(\pi, V) = \max_{a \in \Gamma, b \in \Sigma} |\{c \in \pi^{-1}(a) \mid (b, c) \in V\}|$ be the extension number of V . Below we state our main theorem.

Theorem 4.1 (Main Theorem). Let $\pi : \Sigma \rightarrow \Gamma$ be a function and $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ . Let N be an (m, n) -picture over Γ . There is an algorithm that works in time $O(|\Sigma|^{e(\pi, V)} \cdot s^{e(\pi, V)} \cdot m \cdot n)$ and either constructs a (π, V, H) -solution for N , or correctly determines that no such solution exists.

Note that as an application, we have that if \mathcal{F} is a family of pictures such that for each (m, n) -picture N in \mathcal{F} , N is *poly* (m, n) -smooth, then the picture satisfiability problem for \mathcal{F} can be solved in polynomial time.

We dedicate this section to the proof of Theorem 4.1. For $(i, j) \geq (1, n)$, let $\mathcal{A}_{i,j}(N, \pi, V, H)$ be the LDFA with minimum number of states accepting the set of strings $\partial_{i,j}(N, \pi, V, H)$. The next Lemma will be used in the construction of the automaton $\mathcal{A}_{1,n}(N, \pi, V, H)$. Note that the language accepted by this automaton is simply the set of all colored versions of the first row of N which satisfy constraints imposed by the horizontal relation H .

Lemma 4.2. Let $\pi : \Sigma \rightarrow \Gamma$ be a function, and let $H \subseteq \Sigma \times \Sigma$ be a binary relation over Σ . Let $w = w_1 w_2 \dots w_n$ be a string in Γ^n . Then one can construct

in time $O(|\Sigma|^2 \cdot n)$ an LDFA $\mathcal{A}(w, H)$ of size at most $O(|\Sigma|^2 \cdot n)$ accepting the following language.

$$\mathcal{L}(\mathcal{A}(w, H)) = \{u \in \Sigma^n \mid \pi(u) = w, (u_i, u_{i+1}) \in H \text{ for } i \in \{1, \dots, n-1\}\}.$$

Proof. First, we construct an automaton $\mathcal{A} = (Q, \Sigma, \mathfrak{R})$ as follows. We set $Q = Q_0 \dot{\cup} Q_1 \dot{\cup} \dots \dot{\cup} Q_n$ and $\mathfrak{R} = \bigcup_{i=1}^n \mathfrak{R}_i$ where $Q_0 = \{q_0\}$, $Q_i = \{q_{i,a} \mid a \in \Sigma\}$ for each $i \in \{1, \dots, n\}$, $\mathfrak{R}_1 = \{(q_0, a, q_{1,a}) \mid \pi(a) = w_1\}$, and for each $i \in \{1, \dots, n-1\}$, $\mathfrak{R}_i = \{(q_{i,a}, b, q_{i+1,b}) \mid (a, b) \in H, \pi(b) = w_i\}$. Note that Q has $|\Sigma| \cdot n + 1$ states and at most $|\Sigma|^2 \cdot n$ transitions. Now it is straightforward to check that for each string $u \in \Sigma^n$ there exists an accepting path $q_0 \xrightarrow{u_1} q_{1,u_1} \xrightarrow{u_2} \dots \xrightarrow{u_n} q_{n,u_n}$ in \mathcal{A} if and only if $\pi(u_1)\pi(u_2)\dots\pi(u_n) = w_1w_2\dots w_n$ and for each $i \in \{1, \dots, n-1\}$, $(u_i, u_{i+1}) \in H$. Finally, we let $\mathcal{A}(w, H) = \text{Min}(\mathcal{A})$ be the minimum LDFA which accepts the same language as \mathcal{A} . Since \mathcal{A} is acyclic, minimization can be performed in time linear on the size of \mathcal{A} [20]. So the overall time to construct $\mathcal{A}(w, H)$ is still $O(|\Sigma|^2 \cdot n)$. \square

As a corollary of Lemma 4.2, the LDFA $\mathcal{A}_{1,n}(N, \pi, V, H)$ can be constructed in time $O(|\Sigma|^2 \cdot n)$.

Corollary 4.3. *Let $\pi : \Sigma \rightarrow \Gamma$ be a function and $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ . Then the LDFA $\mathcal{A}_{1,n}(N, \pi, V, H)$ has size $O(|\Sigma|^2 \cdot n)$ and can be constructed in time $O(|\Sigma|^2 \cdot n)$.*

Proof. Set $\mathcal{A}_{i,j}(N, \pi, V, H) = \mathcal{A}(w, H)$, where $w = N_{1,[1,n]}$ and $\mathcal{A}(w, H)$ is the automaton of Lemma 4.2. \square

For each $(i, j) \in [m] \times [n]$ with $(i, j) < (m, n)$, let $\text{Suc}(i, j) = (i, j + 1)$ if $j < n$ and $\text{Suc}(i, j) = (i + 1, 1)$ if $j = n$. In other words, $\text{Suc}(i, j)$ is the smallest pair in $[m] \times [n]$ which is greater than (i, j) according to the lexicographical ordering $<$. Our next step consists in showing that if $(i', j') = \text{Suc}(i, j)$, then the automaton $\mathcal{A}_{i',j'}(N, \pi, V, H)$ can be constructed in time polynomial on the size of $\mathcal{A}_{i,j}(N, \pi, V, H)$. Towards this construction we will need to introduce some notation concerning leveled finite automata.

Let $\mathcal{A} = (Q, \Sigma, \mathfrak{R})$ be a leveled finite automaton over Σ and let q be a state in Q . The *non-deterministic degree* of q , denoted $d(q)$, is defined as the maximum number of states that can be reached from q when reading some fixed symbol $a \in \Sigma$.

$$d(q) = \max_{a \in \Sigma} |\{q' \mid (q, a, q') \in \mathfrak{R}\}| \quad (3)$$

We say that \mathcal{A} has non-deterministic degree d if the following conditions are satisfied.

1. All states of \mathcal{A} have non-deterministic degree at most d .
2. All states of \mathcal{A} of non-deterministic degree greater than one belong to the same level.

The following lemma states that a leveled nondeterministic finite automaton \mathcal{A} of non-deterministic degree d can be transformed into a leveled deterministic finite automaton $\det(\mathcal{A})$ of size at most $|\mathcal{A}|^{O(d)}$ which accepts the same language as \mathcal{A} .

Lemma 4.4. *Let \mathcal{A} be an LNFA of non-deterministic degree d . Let $\det(\mathcal{A})$ be the minimum LDFA accepting $\mathcal{L}(\mathcal{A})$. Then $\det(\mathcal{A})$ has size at most $|\mathcal{A}|^d$ and can be constructed in time $O(|\mathcal{A}|^d)$.*

Proof. Let $\mathcal{A} = (Q, \Sigma, \mathfrak{R})$ where $Q = Q_0 \dot{\cup} Q_1 \dot{\cup} \dots \dot{\cup} Q_n$. Assume that all states of \mathcal{A} with non-deterministic degree greater than one belong to level Q_i . Then the sub-automaton of \mathcal{A} induced by the states $Q_0 \dot{\cup} \dots \dot{\cup} Q_{i-1}$ is deterministic, in the sense that all of its states have non-deterministic degree at most one. Therefore we just need to determinize the sub-automaton of \mathcal{A} induced by the states $Q_i \dot{\cup} \dots \dot{\cup} Q_n$. This determinization process will be achieved by an adaptation of the traditional subset construction, but with the caveat that only subsets of states of size at most d need to be considered.

Let $S \subseteq Q$ be a subset of states of \mathcal{A} , and let $a \in \Sigma$. Then, we define $Q(S, a) = \{q' \mid \exists q \in S, (q, a, q') \in \mathfrak{R}\}$ as the set of all states reachable from some state in S through a transition labeled with a . We note that for each $j \in \{0, \dots, n-1\}$, if $S \subseteq Q_j$ then $Q(S, a) \subseteq Q_{j+1}$. We note that since each state $q \in Q_i$ has non-deterministic degree at most d , we have that $|Q(\{q\}, a)| \leq d$ for each symbol $a \in \Sigma$. Additionally, for each $j \in \{i+1, \dots, n-1\}$, the non-deterministic degree of each state in Q_j is at most one. Therefore, for each subset $S \subseteq Q_j$, we have that $|Q(S, a)| \leq |S|$. Therefore, to construct a deterministic version of \mathcal{A} we only need to consider sets of states of size at most d . The construction is as follows. Consider the automaton $\mathcal{A}' = (Q', \Sigma, \mathfrak{R}')$ where $Q' = Q'_0 \cup Q'_1 \cup \dots \cup Q'_n$ and $\mathfrak{R}' = \mathfrak{R}'_0 \dot{\cup} \dots \dot{\cup} \mathfrak{R}'_n$. For each $j \in \{0, \dots, i\}$ we let $Q'_j = \{q_{j, \{q\}} \mid q \in Q_j\}$. In other words, Q'_j has one state $q_{j, \{q\}}$ for each state $q \in Q_j$. Now for $j \in \{i+1, \dots, n\}$, $Q'_j = \{q_{j, S} \mid S \subseteq Q_j, |S| \leq d\}$. In other words, Q'_j has one state $q_{j, S}$ for each subset of Q_j of size at most d . Now for each $j \in \{1, \dots, n\}$, we set $\mathfrak{R}'_j = \{(q_{j-1, S}, a, q_{j, S'}) \mid q_{j-1, S} \in Q'_{j-1}, q_{j, S'} \in Q'_j, S' = Q(S, a)\}$. Clearly, the automaton \mathcal{A}' is deterministic, and has size at most $|\mathcal{A}|^d$. Additionally we have that $w_1 w_2 \dots w_n$ is accepted by \mathcal{A} if and only if there exists an accepting sequence

$$q_{0, S_0} \xrightarrow{w_1} q_{1, S_1} \xrightarrow{w_2} \dots \xrightarrow{w_n} q_{n, S_n}$$

in \mathcal{A}' where $S_0 = \{q_0\}$ and for each $j \in \{0, \dots, n-1\}$, $S_{j+1} = Q(S_j, w_{j+1})$. This implies that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$. Since \mathcal{A}' is deterministic and acyclic, the minimum leveled deterministic finite automaton $\det(\mathcal{A})$ accepting $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$ can be constructed in time linear on the size of \mathcal{A}' [20], i.e., in time $O(|\mathcal{A}|^d)$. \square

Let $(i', j') = \text{Suc}(i, j)$. The following proposition, whose proof is immediate, establishes a way of defining the boundary set $\partial_{i', j'}(N, \pi, V, H)$ in terms of $\partial_{i, j}(N, \pi, V, H)$.

Proposition 4.5. *Let $\pi : \Sigma \rightarrow \Gamma$ be a function, $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ , and N be an $(m \times n)$ -picture over Γ . Let $\partial_{i,j}(N, \pi, V, H)$ be the set of (i, j) -boundaries of N where $(i, j) < (m, n)$.*

1. *If $j = n$, then*

$$\partial_{i+1,1}(N, \pi, V, H) = \{aw_2\dots w_n \mid a \in \Sigma, (w_1, a) \in V\}.$$

2. *If $j < n$, then*

$$\partial_{i,j+1}(N, \pi, V, H) = \{w_1\dots w_jaw_{j+2}\dots w_n \mid a \in \Sigma, (w_j, a) \in H, (w_{j+1}, a) \in V\}.$$

Using Proposition 4.5, we will prove the following theorem.

Theorem 4.6. *Let $\pi : \Sigma \rightarrow \Gamma$ be a function, $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ , and N be a (m, n) -picture over Γ . Let $(i', j') = \text{Suc}(i, j)$. Then $\mathcal{A}_{i',j'}(N, \pi, V, H)$ can be constructed in time*

$$O(|\Sigma|^{e(\pi, V)} \cdot |\mathcal{A}_{i,j}(N, \pi, V, H)|^{e(\pi, V)}).$$

Proof. Let $\mathcal{A} = \mathcal{A}_{i,j}(N, \pi, V, H) = (Q, \Sigma, \mathfrak{R})$ be the minimum LDFA accepting the set of boundaries $\partial_{i,j}(N, \pi, V, H)$ where $Q = Q_0 \dot{\cup} Q_1 \dot{\cup} \dots \dot{\cup} Q_n$ and $\mathfrak{R} = \mathfrak{R}_1 \dot{\cup} \dots \dot{\cup} \mathfrak{R}_n$. Now let $\mathcal{A}' = (Q', \Sigma, \mathfrak{R}')$ be the LDFA obtained from \mathcal{A} as follows. First set $Q' = Q'_0 \dot{\cup} Q'_1 \dot{\cup} \dots \dot{\cup} Q'_n$ where $Q'_0 = Q_0$ and $Q'_i = \{q^a \mid q \in Q_i, a \in \Sigma\}$. Subsequently set $\mathfrak{R}' = \mathfrak{R}_1 \dot{\cup} \dots \dot{\cup} \mathfrak{R}_n$ where $\mathfrak{R}'_1 = \{(q_0, a, q^a) \mid (q_0, a, q) \in \mathfrak{R}_1\}$ and for each $i \in \{2, \dots, n\}$, $\mathfrak{R}'_i = \{(q^a, b, r^b) \mid (q, b, r) \in \mathfrak{R}_i, a \in \Sigma\}$. Then \mathcal{A}' is still deterministic, and $q_0 \xrightarrow{w_1} q_1 \xrightarrow{w_2} \dots \xrightarrow{w_n} q_n$ is an accepting path in \mathcal{A} if and only if $q_0 \xrightarrow{w_1} q_1^{w_1} \xrightarrow{w_2} \dots \xrightarrow{w_n} q_n^{w_n}$ is an accepting path in \mathcal{A}' . In other words, \mathcal{A}' accepts precisely the same language as \mathcal{A} . Note that $|\mathcal{A}'| \leq |\Sigma| \cdot |\mathcal{A}|$. Now using \mathcal{A}' we will construct a non-deterministic automaton $\mathcal{A}'' = (Q'', \Sigma, \mathfrak{R}'')$ of non-deterministic degree at most $e(\pi, V)$ which accepts the language $\partial_{i',j'}(N, \pi, V, H)$ where $(i', j') = \text{Suc}(i, j)$. Note that the set of states of \mathcal{A}'' is a copy of the set of states of \mathcal{A}' . Now to define the set of transitions \mathfrak{R} of \mathcal{A}'' we need to consider two cases.

1. In the first case, $j = n$, and therefore $i' = i + 1$ and $j' = 1$. In this case we set $\mathfrak{R}''_k = \mathfrak{R}'_k$ for each $k \in \{2, \dots, n\}$, and

$$\mathfrak{R}''_1 = \{(q_0, a, r^b) \mid a \in \pi^{-1}(N_{i+1,1}), (q_0, b, r^b) \in \mathfrak{R}'_1, (b, a) \in V\}.$$

Therefore, we have that $q_0 \xrightarrow{w_1} q_1^{w_1} \xrightarrow{w_2} q_2^{w_2} \dots \xrightarrow{w_n} q_n^{w_n}$ is an accepting path of \mathcal{A}' if and only if for each a with $(w_1, a) \in V$, $q_0 \xrightarrow{a} q_1^{w_1} \xrightarrow{w_2} q_2^{w_2} \dots \xrightarrow{w_n} q_n$ is an accepting path of \mathcal{A}'' . Stated otherwise,

$$\mathcal{L}(\mathcal{A}'') = \{aw_2\dots w_n \mid a \in \Sigma, (w_1, a) \in V, w \in \mathcal{L}(\mathcal{A}')\}.$$

By Proposition 4.5, we have that $\mathcal{L}(\mathcal{A}'') = \partial_{i',j'}(N, \pi, V, H)$.

2. In the second case, $j < n$, and therefore $i' = i$ and $j' = j + 1$. In this case, we set $\mathfrak{R}_k = \mathfrak{R}'_k$ for $k \neq j + 1$. We define \mathfrak{R}''_{j+1} as follows.

$$\mathfrak{R}''_{j+1} = \{(q^c, a, r^b) \mid (q^c, b, r^b) \in \mathfrak{R}_k, a \in \pi^{-1}(N_{i,j+1}), (c, a) \in H, (b, a) \in V\}.$$

Then we have that $q_0 \xrightarrow{w_1} q_1^{w_1} \xrightarrow{w_2} q_2^{w_2} \dots \xrightarrow{w_n} q_n^{w_n}$ is an accepting path of \mathcal{A} if and only if for each $a \in \Sigma$ with $(w_j, a) \in H$ and $(w_{j+1}, a) \in V$, the path

$$q_0 \xrightarrow{w_1} \dots \xrightarrow{w_j} q_j^{w_j} \xrightarrow{a} q_{j+1}^{w_{j+1}} \dots \xrightarrow{w_n} q_n^{w_n}$$

is an accepting path of \mathcal{A}'' . In other words,

$$\mathcal{L}(\mathcal{A}'') = \{w_1 \dots w_j a w_{j+2} \dots w_n \mid w \in \mathcal{L}(\mathcal{A}'), a \in \Sigma, (w_j, a) \in H, (w_{j+1}, a) \in V\}.$$

Since $\mathcal{L}(\mathcal{A}') = \partial_{i,j}(N, \pi, V, H)$, by Proposition 4.5 we have that $\mathcal{L}(\mathcal{A}'') = \partial_{i',j'}(N, \pi, V, H)$.

In both cases considered above, \mathcal{A}'' has as many states and transitions as \mathcal{A}' . Nevertheless, contrary to \mathcal{A}' , \mathcal{A}'' is non-deterministic. Fortunately, the non-deterministic degree of \mathcal{A}'' is upper bounded by $e(\pi, V)$. This implies that by Lemma 4.4, the minimum leveled deterministic finite automaton accepting the same language as \mathcal{A}'' can be constructed in time

$$O(|\mathcal{A}''|^{e(\pi, V)}) = O(|\Sigma|^{e(\pi, V)} \cdot |\mathcal{A}_{i,j}(N, \pi, V, H)|^{e(\pi, V)}).$$

□

Now, for each $(i, j) \geq (1, n)$ we know how to construct the automaton $\mathcal{A}_{i,j}(N, \pi, V, H)$ accepting the set $\partial_{i,j}(N, \pi, V, H)$. Note that since by assumption, the picture N is s -smooth, we have that the collection of all automata $\mathcal{A}_{i,j}(N, \pi, V, H)$ can be constructed in time $O(|\Sigma|^{e(\pi, V)} \cdot s^{e(\pi, V)} \cdot m \cdot n)$. If the last automaton $\mathcal{A}_{m,n}(N, \pi, V, H)$ accepts the empty language, then the picture N has no (π, V, H) -solution. On the other hand, if this language is not empty, then we still need to construct a solution. Let $\mathcal{A}_i = \mathcal{A}_{i,n}(N, \pi, V, H)$ be the automaton accepting the boundary set $\partial_{i,n}(N, \pi, V, H)$. Let $\gamma : \Sigma \times \Sigma \rightarrow \Sigma$ be a projection which sets $\gamma(a, b) = a$ for each pair $(a, b) \in \Sigma \times \Sigma$. In other words, γ erases the second coordinate of each pair $(a, b) \in \Sigma \times \Sigma$. For a string $u = (a_1, b_1)(a_2, b_2) \dots (a_n, b_n) \in (\Sigma \times \Sigma)^n$, we let $\gamma(u) = a_1 a_2 \dots a_n$. Also, for a string $w \in \Sigma^n$, let $\mathcal{A}(w)$ be the minimum LDFA that accepts w , and no other string. We will construct a (π, V, H) -solution M for N row by row, starting from the last row of M and finishing at the first. The construction is inductive. More precisely, we construct a sequence w^n, w^{n-1}, \dots, w^1 of strings in Σ^n as follows.

1. Let w^n be an arbitrary string in $\mathcal{L}(\mathcal{A}_n)$.
2. For $i = n - 1, \dots, 1$:
 - (a) Let w^i be an arbitrary string in $\mathcal{L}(\gamma(\mathcal{A}_i \otimes_V \mathcal{A}(w^{i+1})))$.
3. Set M to be the (m, n) -picture over Σ such that w^i is the i -th row of M .

We note that the string w^i selected from $\mathcal{L}(\gamma(\mathcal{A}_i \otimes_V \mathcal{A}(w^{i+1})))$ is simply a string in the boundary $\partial_{i,n}(N, \pi, V, H)$ satisfying the property that $(w_k^i, w_k^{i+1}) \in V$ for each $k \in \{1, \dots, n\}$. In other words, w^i is vertically compatible with w^{i+1} . Additionally, since w^i by definition belongs to $\partial_{i,n}(N, \pi, V, H)$ we have that for each $k \in \{1, \dots, n-1\}$, $(w_k^i, w_{k+1}^i) \in H$. In other words, the horizontal constraints imposed by the relation H are satisfied by each w^i . This shows that the picture M obtained by setting, for each $i \in \{1, \dots, m\}$, w^i as the i -th row of M is a valid (π, V, H) -solution for N . We note that the time to select each string w^i is of the order of $O(|\mathcal{A}_i|)$. Therefore, the total algorithm still runs in the time $O(|\Sigma|^{e(\pi, V)} \cdot s^{e(\pi, V)} \cdot m \cdot n)$. This proves Theorem 4.1. \square

5 Pigeonhole Pictures

In this section we define a family of pictures formalizing the pigeonhole principle. We call these pictures the pigeonhole pictures. Subsequently, we will show that this family is polynomially smooth. This implies that our algorithm for the satisfiability of smooth pictures is able to determine whether a given element of this family has a solution, and in the case it has, the algorithm is able to construct it. On the other hand, we will show in Sect. 7 that CNF formulas derived from the pigeonhole pictures require constant-depth Frege proofs of exponential size.

Our primary alphabet $\Gamma = \{\oplus, \circ, \odot\}$ has a left symbol \oplus , used to fill all entries of the first column of the picture, a right symbol \odot , used to fill all entries of the last column of the picture, and a middle symbol \circ , used to fill all entries in between the first and last columns. Our colored alphabet is defined as $\Sigma = \{b, g, bb, bg, gb, gg, rr\}$. Finally, the projection $\pi : \Sigma \rightarrow \Gamma$ is such that $\pi(\{bb, bg, gb, gg, rr\}) = \{\circ\}$, $\pi(b) = \oplus$ and $\pi(g) = \odot$. Intuitively, the letters r, b, g stand for “red”, “blue” and “green” respectively. The symbols in $\{bb, bg, gb, gg, rr\}$ are called double colors. The left symbol \oplus can only be colored with b , then right symbol \odot can only be colored with g , and the middle symbol can be colored with any double color in $\{bb, bg, gb, gg, rr\}$. The color red serves to indicate presence of a pigeon. The colors blue and green serve to mark the “footprint” of a pigeon. The vertical and horizontal relations are defined as follows:

$$\begin{aligned}
 V &= \{ (xb, yb), (xb, rr), (rr, xg), (xg, yg) \mid x, y \in \{b, g\} \} \cup \{ (b, b), (g, g) \} \\
 H &= \{ (bx, by), (bx, rr), (rr, gx), (gx, gy) \mid x, y \in \{b, g\} \} \\
 &\quad \cup \\
 &\quad \{ (b, bz), (b, rr), (rr, g), (gz, g) \mid z \in \{b, g\} \}
 \end{aligned} \tag{4}$$

We define the (m, n) -pigeonhole picture, $P^{m,n}$, as the $m \times (n+2)$ picture, where all entries in the first column are filled with the left symbol \oplus , all entries in the last column are filled with the right symbol \odot , and all other entries are filled with the middle symbol \circ .

Definition 5.1 (Pigeonhole Picture). *The (m, n) -pigeonhole picture $P^{m,n}$ is the $m \times (n + 2)$ picture over Γ defined as follows.*

$$P_{ij}^{m,n} = \begin{cases} \oplus & \text{if } j = 1. \\ \odot & \text{if } j = n + 2. \\ \circ & \text{otherwise.} \end{cases} \quad (5)$$

Intuitively, the rows of $P^{m,n}$ correspond to pigeons, while the middle columns correspond to holes. Pigeons are not allowed to occupy the first nor the last columns. If M is a (π, V, H) -solution for the pigeonhole picture then the fact that $M_{i,j} = rr$ indicates that the i -th pigeon is placed at the hole represented by column j . The fact that $M_{i,j} = bx$ for some $x \in \{g, b\}$ indicates that the i -th pigeon is placed in some column greater than j , while the fact that $M_{i,j} = gx$ for some $x \in \{b, g\}$ indicates that the i -th pigeon is placed in some column smaller than j . Analogously, if $M_{i,j} = xb$ for some $x \in \{b, g\}$, then the pigeon that is placed at the j -th hole is greater than i , while if $M_{i,j} = xg$, then the pigeon that is placed at the j -th hole is smaller than i .

Note also that the horizontal relation guarantees that a pigeon must occur in each row of a satisfying assignment. This is because there is no allowed pair $(xy, x'y')$ where x is blue and x' is green. Therefore in a satisfying assignment, any row must have at least one position colored with rr . Now for the columns we note that if some pigeon occurs in a position (i, j) then the second color in each entry below (i, j) must be green, while the second color of each entry above (i, j) must be blue. Therefore no two pigeons are allowed to appear on the same column of a satisfying assignment.

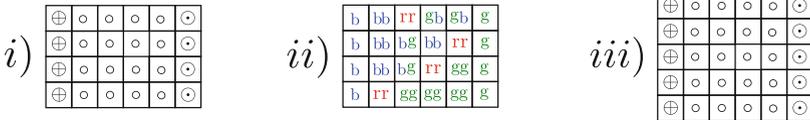


Fig. 2. (i) The pigeonhole picture $P^{4,4}$. (ii) A solution for $P^{4,4}$. (iii) The pigeonhole picture $P^{5,4}$ is unsatisfiable. (Color figure online)

We note that if $m > n$, then pigeonhole picture $P^{m,n}$ is unsatisfiable. The following theorem says that the family of pigeonhole pictures is polynomially smooth. This implies that our algorithm for the satisfiability of smooth pictures can be used to decide the unsatisfiability of the pigeonhole pictures in polynomial time.

Theorem 5.2. *Let $P^{m,n}$ be the (m, n) -pigeonhole picture. Then for each $i, j \in \{1, \dots, m\} \times \{1, \dots, n + 1\}$, the boundary $\partial_{i,j}(P^{m,n}, \pi, V, H)$ is accepted by an LDFA of size at most $O(m \cdot n)$.*

Proof. Let M be a solution for $P^{m,n}$. If $M_{i,j} = y_1 y_2$ where $i \in \{2, \dots, n + 1\}$, then we say that y_1 is the first coordinate of $M_{i,j}$ while y_2 is the second coordinate of $M_{i,j}$.

We sketch the proof in the case in which $j = n + 2$. In other words we will show how to construct the boundaries corresponding to full rows. The generalization for smaller values of j is straightforward. Thus let M be an $(i, n + 2)$ partial solution. Then the border of M is simply its i -th row. Since M is a partial solution, there exists a unique k such that $M_{i,k} = rr$. Additionally, if $k' < k$ then the first coordinate of $M_{i,k'}$ is necessarily blue, indicating that a pigeon has not occurred in that row up to position k' . On the other hand, if $k' > k$, then the first coordinate of $M_{i,k'}$ is green, indicating that a pigeon has already occurred at row i . Now, we also have that precisely i pigeons must be present in M , one for each row. Therefore, there must exist precisely $i - 1$ entries of row i whose second coordinate is green. Indeed, for an entry $M_{i,k'}$ with $k' \neq k$, if the second coordinate of $M_{i,k'}$ is green, then we know that a pigeon has already occurred at column k' . On the other hand, if $M_{i,k'}$ is blue then we know that no pigeon occurred yet at column k' . It turns out that the converse also holds. Namely, if each two consecutive entries of row i satisfy the horizontal constraints imposed by the horizontal relation H and there exists a unique entry which is equal to rr and precisely $i - 1$ entries whose second coordinate is green, then we know that all other entries of M can be filled in such a way that it is an $(i, n + 2)$ solution. In other words, the strings $w \in \Sigma^{n+2}$ belonging to the border of an $(i, n + 2)$ partial solution are characterized by the following properties.

1. The first entry of w is b and the last entry is g .
2. $(w_k, w_{k+1}) \in H$ for each $k \in \{1, \dots, n + 1\}$
3. There exists a unique k such that $w_k = rr$
4. w has precisely $i - 1$ entries whose second coordinate is green.

Now one can implement a leveled deterministic finite automaton with $O(i \cdot n)$ transitions and levels Q_0, Q_1, \dots, Q_{n+1} which accepts a string $w \in \Sigma^{n+2}$ if and only if the conditions above are satisfied. Note that the three first conditions are immediate to verify using such an automaton. The fourth condition can be implemented by considering that each level Q_k is split into subsets of states Q_k^r for $r \in \{1, \dots, i\}$, where the states in Q_k^r indicate that from the k first read symbols of w , r of them have the second coordinate green. \square

Corollary 5.3. *Let $P^{m,n}$ be the pigeonhole picture. Then the algorithm devised in the proof of Theorem 4.1 determines in time $O(m^4 \cdot n^4)$ whether $P^{m,n}$ has a (π, V, H) -solution. In case such a solution exists the algorithm constructs it.*

Proof. By Theorem 5.2, $P^{m,n}$ is $O(m \cdot n)$ -smooth. Additionally, the extension number of (π, V) is $e(\pi, V) = 3$. Therefore from Theorem 4.6, we can construct each automaton $\mathcal{A}_{i,j}(P^{m,n}, \pi, V, H)$ in time at most $O(m^3 \cdot n^3)$. Since there are $m \cdot n$ such automata, the whole algorithm takes time at most $O(m^4 \cdot n^4)$. \square

6 From Pictures to Constant Width CNF Formulas

Let $\pi : \Sigma \rightarrow \Gamma$ be a function, $V, H \subseteq \Sigma \times \Sigma$ be binary relations over Σ , and M be an $m \times n$ -picture over Γ . Next we define a constant width CNF formula $F(M)$ that is satisfiable if and only if M is (π, V, H) -satisfiable.

Let $S_{ij} = \pi^{-1}(M_{ij})$ be the set of colored versions of the symbol M_{ij} . The formula $F(M)$ has a variable x_{ija} for each $(i, j) \in [m] \times [n]$, and each symbol $a \in S_{ij}$. Intuitively, the variable x_{ija} is true if the position (i, j) of a solution picture is set to a . The following set of clauses specifies that in a satisfying assignment, precisely one symbol of S_{ij} occupies the position (i, j) .

$$\text{OneSymbol}(M, i, j) \equiv \bigvee_{s \in S_{ij}} x_{ijs} \wedge \bigwedge_{s, s' \in S_{ij}, s \neq s'} (\bar{x}_{ijs} \vee \bar{x}_{ijs'}) \quad (6)$$

The next set of clauses expresses the fact that no pair of symbols $(a, a') \notin H$ occur in consecutive horizontal positions at row i .

$$\text{Horizontal}(M, i) \equiv \bigwedge_{(a, a') \notin H, j \in \{1, \dots, n-1\}} (\bar{x}_{ija} \vee \bar{x}_{i(j+1)a'}) \quad (7)$$

Similarly, the following set of clauses expresses the fact that no pair of symbols $(a, a') \notin V$ occurs in consecutive vertical positions at column j .

$$\text{Vertical}(M, j) \equiv \bigwedge_{(a, a') \notin V, i \in \{1, \dots, m-1\}} (\bar{x}_{ija} \vee \bar{x}_{(i+1)ja'}) \quad (8)$$

Finally, we set the formula $F(M)$ as follows.

$$F(M) \equiv \bigwedge_{i=1}^m \text{Horizontal}(M, i) \wedge \bigwedge_{j=1}^n \text{Vertical}(M, j) \wedge \bigwedge_{ij} \text{OneSymbol}(M, i, j) \quad (9)$$

7 Lower Bound for Bounded Depth Frege Proofs

Let $P^{m, m-1}$ be the pigeonhole pictures as defined in Sect. 5. In this section we will show that bounded-depth Frege refutations of the family of formulas $\{F(P^{m, m-1})\}_{m \in \mathbb{N}}$ require exponential size. Recall that a Frege system is specified by a finite set of rules of the form

$$\frac{\varphi_0(q_1, \dots, q_m)}{\varphi_1(q_1, \dots, q_m), \dots, \varphi_r(q_1, \dots, q_m)} \quad (10)$$

where q_1, \dots, q_m are variables and $\varphi_0, \varphi_1, \dots, \varphi_r$ are formulas in the language $\vee, \wedge, \neg, 0, 1$, and q_1, \dots, q_m . The only requirement is that the rules are sound and complete [6]. An instance of the rule is obtained by substituting particular formulas ψ_1, \dots, ψ_m (in the language $\vee, \neg, 0, 1, y_{ij}$) for the variables q_1, \dots, q_m . A rule in which $r = 0$ is called an axiom scheme. The size of a formula F is the number of symbols $\{\vee, \wedge, \neg\}$ in it. The depth of a formula is the size of the longest path from the root of F to one of its leaves. We say that a proof has depth d if all formulas occurring in it have depth at most d .

Now consider the family of pigeonhole formulas $\{H_m\}_{m \in \mathbb{N}}$. For each $m \in \mathbb{N}$, H_m has variables y_{ij} for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, m-1\}$ and the following clauses.

1. $(y_{i,1} \vee \dots \vee y_{i,m-1})$ for each $i \in \{1, \dots, m\}$.
2. $(\neg y_{i,j} \vee \neg y_{k,j})$ for each $i, k \in \{1, \dots, m\}, j \in \{1, \dots, m-1\}$.

Intuitively, when set to true, the variable $y_{i,j}$ indicates that pigeon i sits at hole j . Clauses of the first type specify that each pigeon has to sit in at least one hole, clauses of the second type specify that no two distinct pigeons sit in the same hole. Clearly, the formula H_m is unsatisfiable for each $m \in \mathbb{N}$. The following theorem states that H_m is hard for bounded depth Frege systems.

Theorem 7.1 ([13, 18]). *For each Frege proof system F there exists a constant c such that for each d , and each sufficiently large m , every depth- d refutation of H_m must have size at least $2^{m^{c-d}}$.*

Now let $F(P^{m,m-1})$ be the formula derived from the pigeonhole picture $P^{m,m-1}$. This formula has the following variables¹:

$$x_{i,j,bb}, \quad x_{i,j,bg}, \quad x_{i,j,gb}, \quad x_{i,j,gg} \quad i \in \{1, \dots, m\}, j \in \{2, \dots, m\} \quad (11)$$

$$x_{i,1,b}, \quad x_{i,m+1,g}, \quad \text{for } i \in \{1, \dots, m\} \quad (12)$$

We will consider a substitution of variables that transforms the formula $F(P^{m,m-1})$ into a formula F' in the variables y_{ij} used by the formula H_m . Intuitively, the variable $x_{i,j,rr}$ which expresses that the pigeon i is being placed at column j in a solution for the picture $P^{m,m-1}$, is mapped to the variable $y_{i,j-1}$ which expresses that the pigeon i is placed at hole $j-1$. Note that the j -th column of a solution for the picture $P^{m,m-1}$ is the $(j-1)$ -th hole. The other variables are then mapped to a conjunction of disjunctions. For instance, the variable $x_{i,j,bb}$ is true in an hypothetical satisfying assignment of $F(P^{m,m-1})$ if and only if the pigeon at row i occurs after the j -th entry of this row, and the pigeon at column j appears after the i -th entry of this column. Analogue substitutions can be made with respect to the other variables. These substitutions are formally specified below.

1. For $i \in \{1, \dots, m\}$:
 - (a) $x_{i,1,b} \rightarrow (y_{i,1} \vee \dots \vee y_{i,m-1})$
 - (b) $x_{i,(m+1),g} \rightarrow (y_{i,1} \vee \dots \vee y_{i,m-1})$
2. For $i \in \{1, \dots, m\}, j \in \{2, \dots, m\}$
 - (a) $x_{i,j,rr} \rightarrow y_{i,(j-1)}$
 - (b) $x_{i,j,bb} \rightarrow (y_{i,j} \vee \dots \vee y_{i,m-1}) \wedge (y_{i+1,j} \vee \dots \vee y_{m,j})$
 - (c) $x_{i,j,bg} \rightarrow (y_{i,j} \vee \dots \vee y_{i,m-1}) \wedge (y_{1,j} \vee \dots \vee y_{i-1,j})$
 - (d) $x_{i,j,gb} \rightarrow (y_{i,1} \vee \dots \vee y_{i,j-2}) \wedge (y_{i+1,j} \vee \dots \vee y_{m,j})$
 - (e) $x_{i,j,gg} \rightarrow (y_{i,1} \vee \dots \vee y_{i,j-2}) \wedge (y_{1,j} \vee \dots \vee y_{i-1,j})$

¹ Recall that the first and last columns of the pigeonhole picture do not correspond to holes.

Let F' be the formula that is obtained from $F(P^{m,m-1})$ by replacing its variables according to the substitutions defined above. Then the formula F' has only variables y_{ij} . Additionally, the implication $H_m \Rightarrow F'$ can be proved by a bounded depth Frege proof of polynomial size. Now suppose that Π is a depth- d Frege refutation of the formula $F(P^{m,m-1})$ of size S . Then, if we replace all variables occurring in formulas of Π according to the substitutions above, we get a depth $d + 2$ Frege refutation Π' of the formula F' whose size is at most $O(m) \cdot S$. But since the implication $H_m \Rightarrow F'$ has a Frege proof of size $\text{poly}(m)$, we have that Π' also can be used to construct a refutation of H_m of size $\text{poly}(m) \cdot S$. Therefore by Theorem 7.1, the size of S must be at least $2^{m^{c'-d}}$ for some constant c' independent on m . \square

Acknowledgments. This work was supported by the European Research Council, grant number 339691, in the context of the project Feasibility, Logic and Randomness (FEALORA). The author thanks Pavel Pudlák and Neil Thapen for enlightening discussions on Frege proof systems.

References

1. Beame, P., Kautz, H., Sabharwal, A.: Towards understanding and harnessing the potential of clause learning. *J. Artif. Intell. Res.* **22**, 319–351 (2004)
2. Beame, P., Pitassi, T.: Simplified and improved resolution lower bounds. In: *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pp. 274–282. IEEE (1996)
3. Buss, S.R., et al.: Resolution proofs of generalized pigeonhole principles. *Theoret. Comput. Sci.* **62**(3), 311–317 (1988)
4. Buss, S.R., Hoffmann, J., Johannsen, J.: Resolution trees with lemmas: resolution refinements that characterize DLL algorithms with clause learning. *Logical Meth. Comput. Sci.* **4**, 1–18 (2008)
5. Cherubini, A., Rehgizzi, S.C., Pradella, M., San, P.: Picture languages: Tiling systems versus tile rewriting grammars. *Theoret. Comput. Sci.* **356**(1), 90–103 (2006)
6. Cook, S.A., Reckhow, R.A.: The relative efficiency of propositional proof systems. *J. Symbol. Logic* **44**(01), 36–50 (1979)
7. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) *SAT 2003*. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004)
8. Giammarresi, D., Restivo, A.: Recognizable picture languages. *Int. J. Pattern Recogn. Artif. Intell.* **6**(2&3), 241–256 (1992)
9. Haken, A.: The intractability of resolution. *Theoret. Comput. Sci.* **39**, 297–308 (1985)
10. Hertel, P., Bacchus, F., Pitassi, T., Van Gelder, A.: Clause learning can effectively P-simulate general propositional resolution. In: *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI 2008)*, pp. 283–290 (2008)
11. Kim, C., Sudborough, I.H.: The membership and equivalence problems for picture languages. *Theoret. Comput. Sci.* **52**(3), 177–191 (1987)
12. Krajíček, J.: Lower bounds to the size of constant-depth propositional proofs. *J. Symbol. Logic* **59**(01), 73–86 (1994)

13. Krajčiček, J., Pudlák, P., Woods, A.: An exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Struct. Algorithms* **7**(1), 15–39 (1995)
14. Latteux, M., Simplot, D.: Recognizable picture languages and domino tiling. *Theoret. Comput. Sci.* **178**(1), 275–283 (1997)
15. Maurer, H.A., Rozenberg, G., Welzl, E.: Using string languages to describe picture languages. *Inf. Control* **54**(3), 155–185 (1982)
16. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: engineering an efficient SAT solver. In: *Proceedings of the 38th Annual Design Automation Conference*, pp. 530–535. ACM (2001)
17. Pipatsrisawat, K., Darwiche, A.: On the power of clause-learning SAT solvers as resolution engines. *Artif. Intell.* **175**(2), 512–525 (2011)
18. Pitassi, T., Beame, P., Impagliazzo, R.: Exponential lower bounds for the pigeonhole principle. *Comput. Complex.* **3**(2), 97–140 (1993)
19. Raz, R.: Resolution lower bounds for the weak pigeonhole principle. *J. ACM (JACM)* **51**(2), 115–138 (2004)
20. Revuz, D.: Minimisation of acyclic deterministic automata in linear time. *Theoret. Comput. Sci.* **92**(1), 181–189 (1992)
21. Rosenfeld, A.: *Picture Languages: Formal Models for Picture Recognition*. Academic Press (2014)
22. Simplot, D.: A characterization of recognizable picture languages by tilings by finite sets. *Theoret. Comput. Sci.* **218**(2), 297–323 (1999)
23. Stromoney, G., Siromoney, R., Krithivasan, K.: Abstract families of matrices and picture languages. *Comput. Graph. Image Process.* **1**(3), 284–307 (1972)



<http://www.springer.com/978-3-319-40969-6>

Theory and Applications of Satisfiability Testing – SAT
2016

19th International Conference, Bordeaux, France, July
5-8, 2016, Proceedings

Creignou, N.; Le Berre, D. (Eds.)

2016, XXIV, 564 p. 119 illus., Softcover

ISBN: 978-3-319-40969-6