The $\#a = \#b$ Pictures are Recognizable

Klaus Reinhardt

Wilhelm-Schickhard Institut für Informatik, Universität Tübingen
Sand 13, D-72076 Tübingen, Germany
e-mail: reinhard@informatik.uni-tuebingen.de

Abstract. We show that the language of pictures over $\{a, b\}$ (with a reasonable relation between height and width), where the number of a's is equal to the number of b's, is recognizable using a finite tiling system. This means that counting in rectangular arrays is definable in existential monadic second-order logic.

Classification: Automata and formal languages, Logic

1 Introduction

In [GRST94] pictures are defined as two-dimensional rectangular arrays of symbols of a given alphabet. A set (language) of pictures is called recognizable if it is recognized by a finite tiling system. It was shown in [GRST94] that a picture language is recognizable iff it is definable in existential monadic second-order logic. In [Wil97] it was shown that star-free picture expressions are strictly weaker than first-order logic. The context-sensitive languages are characterized in [LS97a] as frontiers of picture languages. In the same spirit a link to computational complexity is established in [Bor99], where NP is characterized with the notion of recognizability by padding 1-dimensional words with blanks to form an n-dimensional cube.

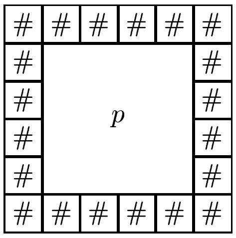
A comparison to other regular and context-free formalisms to describe picture languages can be found in [Mat97,Mat98]. Characterizations of the recognizable picture languages by automata can be found in [IN77] and [GR96], where also the subclasses, which are defined by a restriction from nondeterminism to determinism or unambiguity are considered.

In the one-dimensional case counting is some kind of a prototype concept for non recognizability ($\equiv$ non regularity) but spending one extra dimension easily enables counting (see Section 2) for one line. But it was so far conjectured that counting can not be done for 2 dimensions without having an extra third dimension available. In [Rei98] the author could only find a nonuniform method for simulating a counter along a picture, which just showed why the attempts, which had been made to disprove Theorem 1, had failed.

1.1 Preliminaries

Definition 1. *[GRST94] A* picture *over Σ is a two-dimensional array of elements of Σ. The set of pictures of size (m, n) is denoted by $\Sigma^{m,n}$. A* picture language *is a subset of* $\Sigma^{*,*} := \bigcup_{m,n\geq 0} \Sigma^{m,n}$. *For a* $p \in \Sigma^{m,n}$, *we define* $\hat{p} \in \Sigma^{m+2,n+2}$ *adding a frame of symbols* $\# \notin \Sigma$.

Let $T_{m,n}(p)$ *be the set of all sub-pictures of* p *with size* (m, n).

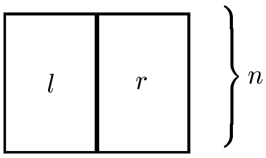
$\hat{p} :=$ 

A picture language $L \subseteq \Gamma^{*,*}$ *is called* local *if there is a* Δ *with* $L = \{p \in \Gamma^{*,*} | T_{2,2}(\hat{p}) \subset \Delta\}$. *A picture language* $L \subseteq \Gamma^{*,*}$ *is called* hv-local *if there is a* Δ *with* $L = \{p \in \Gamma^{*,*} | T_{1,2}(\hat{p}) \cup T_{2,1}(\hat{p}) \subset \Delta\}$. *A picture language* $L \subseteq \Sigma^{*,*}$ *is called* recognizable *if there is a mapping* $\pi : \Gamma \to \Sigma$ *and a local language* $L' \subset \Gamma^{*,*}$ *with* $L = \pi(L')$.

This means in order to recognize a picture, we have to (non-deterministically) find a pre-image in the local pre-image language. According to [LS97b] L is recognizable if and only if there is a mapping $\pi : \Gamma \to \Sigma$ and a hv-local language $L' \subset \Gamma^{*,*}$ with $L = \pi(L')$, this means we can use a hv-local pre-image language as well.

A necessary condition for recognizability, reflecting that at most an exponential amount of information can get from one half of the picture to another, is the following:

Lemma 1. *[Mat98] Let* $L \subseteq \Gamma^{*,*}$ *be recognizable and* $(M_n \subseteq \Gamma^{n,*} \times \Gamma^{n,*})$ *be sets of pairs with* $\forall n, \forall (l, r) \in M_n$ $lr \in L$ *and* $\forall (l, r) \neq (l', r') \in M_n$ $lr' \notin L$ *or* $l'r \notin L$, *then* $|M_n| \in 2^{O(n)}$.

$lr =$ 

Let L be the set of pictures over $\{a, b\}$, where the number of a's is equal to the number of b's. Considering pictures where the width is $f(n) \in 2^{\omega(n)}$ for the height n, we can find $f(n)$ pairs $(l_1, r_1), (l_2, r_2), ..., (l_{f(n)}, r_{f(n)})$, such that l_i has i more a's as b's and r_i has i more b's as a's for all $i \leq f(n)$, thus $l_i r_i$ is in L but all the $l_i r_j$ with $i \neq j$ have a different number of a's and b's and are thus not in L. By contradiction of $f(n) \in 2^{\omega(n)}$ with Lemma 1 we get the following:

Corollary 1. *The language of pictures over* $\{a, b\}$, *where the number of* a*'s is equal to the number of* b*'s (and where sizes* (n, m) *might occur, which do not follow the restriction* $m \leq f(n)$ *or* $n \leq f(m)$ *for a function* $f \in 2^{O(n)}$*) is not recognizable.*

The main result of the paper is the proof of the following theorem:

Definition 2. *The picture language* $L_=$ *(resp.* $L^c_=$*) is the set of pictures over* $\{a, b\}$ *(resp.* $\{a, b, c\}$*), where the number of* a*'s is equal to the number of* b*'s and having a size* (n, m) *with* $m \leq 2^n$ *and* $n \leq 2^m$.

Remark: We could as well use any other constant base k instead of 2, which means that there is no gap to Corollary 1.

Theorem 1. *The languages $L_=$ and $L_=^c$ are recognizable.*

1.2 Overview

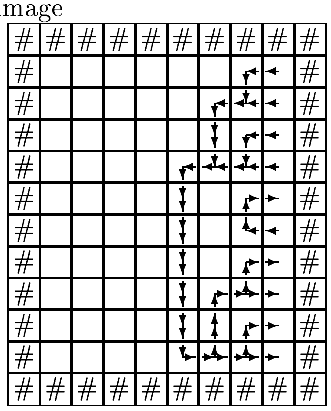
The next section will show how easy it is, if we only have to count the difference of a's and b's in the bottom line. The problem which arises in the general case, is that the counter is not able to accept simple increments at any local position. Section 3 reduces to the problem of counting only a's and b's in odd columns and rows.

The essential idea of the proof of Theorem 1, in order to overcome this problem, is to construct some 'counting flow', which has small constant capacity at each connection leading from one position to its neighbor. The connections have different orders, which are powers of 4. For example a flow of 7 in a connection of order 4^i represents a total flow of $7 \cdot 4^i$. The number of occurrences of a connection with order 4^i is exponentially decreasing with i similar to the counter used in [Für82]. Since the order can not be known on the local level (the alphabet is finite but not the order), some 'skeleton structure' must describe a hierarchy of orders, which gives the information, when some counted value can be transfered from one order to the next. If for example a row having the order 4^i crosses a column having the order 4^{i+1}, the flow in the row may be decreased by 4 simultaneously increasing the flow in the column by 1, which preserves the total flow. This skeleton-language is described in Section 4. Section 5 describes the counting flow for squares of the power of 2 using a variation of the Hilbert-curve. Section 6 shows the generalization to exponentially many appended squares by combining the techniques of Sections 2 and 4 and the generalization to odd cases by folding.

2 Simple counting for one line

This section can be viewed as an exercise for Section 5. Here we consider the language of pictures over $\Sigma = \{a, b, c\}$ with an equal number of a's and b's at the rightmost column and the rest filled with c's (Like for example $\pi(p)$ below).

We use a local pre-image language describing a flow. It is defined over the the alphabet $\Gamma = \{-1, 0, 1\}^4$, where the numbers in (l, r, u, d) describe the flow going out of the position to the left, the right, up and down. In the graphical representation we describe this by arrows. The sources of the flow correspond to the a's, which is described by $\pi((1, 0, 0, 0)) = \pi(\boxed{\leftarrow}) = a$. Analogously the b's are the sinks: $\pi((-1, 0, 0, 0)) = \pi(\boxed{\rightarrow}) = b$. Everywhere else the flow has to be continued, which is expressed by $\pi((l, r, u, d)) = c$ if $2l + r + u + d = 0$. The main point is that the flow to the left side has the double order. This means that flows from the rightmost column to the second rightmost column and flows within the second rightmost column have order 1 and in general flows between the i-th rightmost column and the $i+1$-th rightmost column and flows within the $i+1$-th rightmost column have the order 2^{i-1}. For example we may use the pre-image

$\hat{p} =$  for $\pi(\hat{p}) =$

$$\begin{array}{|c|c|c|c|c|c|c|c|c|c|}\hline
\# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \hline
\# & c & c & c & c & c & c & c & a & \# \\ \hline
\# & c & c & c & c & c & c & c & a & \# \\ \hline
\# & c & c & c & c & c & c & c & a & \# \\ \hline
\# & c & c & c & c & c & c & c & a & \# \\ \hline
\# & c & c & c & c & c & c & c & b & \# \\ \hline
\# & c & c & c & c & c & c & c & a & \# \\ \hline
\# & c & c & c & c & c & c & c & b & \# \\ \hline
\# & c & c & c & c & c & c & c & b & \# \\ \hline
\# & c & c & c & c & c & c & c & b & \# \\ \hline
\# & c & c & c & c & c & c & c & b & \# \\ \hline
\# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \hline
\end{array}$$

Although there might be several possible pre-images, one of them can be obtained in the following way: For a and b there is only one possible pre-image. The pre-image (l, r, u, d) for c is chosen by taking $r := l'$ for the right neighbor $(l', , ,)$ and $u := d'$ for the upper neighbor $(, , , d')$ ($u := 0$ if the upper neighbor is #). If $r + u = 2$ resp. -2, let $l := -1$ resp 1 and $d := 0$, if $r + u = 1$ resp. -1, let $d := -1$ resp 1 and $l := 0$, else $d := l := 0$.

In this way the flow from one row down to the next row corresponds to the binary representation of the difference in the number of a's and b's so far. A flow to the left corresponds to a carry-bit.

The formal definition for Δ is

$$\begin{aligned}
\Delta := \; & \{\begin{array}{|c|}\hline \# \\ \hline \gamma \\ \hline\end{array} \mid \gamma = (l, r, 0, d)\} \cup \{\begin{array}{|c|}\hline \gamma \\ \hline \# \\ \hline\end{array} \mid \gamma = (l, r, u, 0)\} \\
\cup \; & \{\begin{array}{|c|c|}\hline \# & \gamma \\ \hline\end{array} \mid \gamma = (0, r, u, d)\} \cup \{\begin{array}{|c|c|}\hline \gamma & \# \\ \hline\end{array} \mid \gamma = (l, 0, 0, 0), l \in \{1, -1\}\} \\
\cup \; & \{\begin{array}{|c|}\hline \delta \\ \hline \gamma \\ \hline\end{array} \mid \delta = (l, r, u, d), \gamma = (l', r', d, d')\} \\
\cup \; & \{\begin{array}{|c|c|}\hline \delta & \gamma \\ \hline\end{array} \mid \delta = (l, r, u, d), \gamma = (r, r', u', d')\}.
\end{aligned}$$

Remark: We could as well use $\Gamma = \{1-k, ..., k-1\}^4$ and $\pi((l, r, u, d)) = c$ if $k \cdot l + r + u + d = 0$ to be able to treat pictures of size (n, m) with $m \leq k^n$.

3 Reduction to odd positions

We view a mapping $e : \Sigma \mapsto \Sigma_e^{i,j}$ as lifted to map a picture of size (m, n) over Σ to a picture of size (im, jn) over Σ_e.

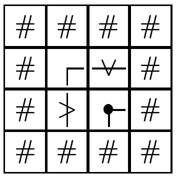
Lemma 2. *A picture language L over Σ is recognizable if $e(L)$ is recognizable and e is injective.*

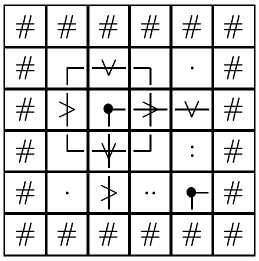
Proof. Let $e(L)$ be recognizable by a mapping $\pi_e : \Gamma_e \mapsto \Sigma_e$ and a tiling $\Delta_e \subseteq \Gamma_e^{2,2}$. Construct $\Gamma := \{g \in \Gamma_e^{i,j} \mid \exists s \in \Sigma\ e(s) = \pi_e(g)\}$, $\pi : \Gamma \mapsto \Sigma$ with $\pi(g) = s$ for $e(s) = \pi_e(g)$ (e injective) and $\Delta := \{p \in (\Gamma_e \cup \{\#\})^{2i,2j} \mid p \in (\Gamma \cup \{\#\})^{2,2}, T_{2,2}(p) \subseteq \Delta_e \cup \{\begin{array}{|c|c|}\hline \# & \# \\ \hline \# & \# \\ \hline \end{array}\}\}$, where for simplicity we identify $\#$ with the picture of size (i,j) consisting only of $\#$.

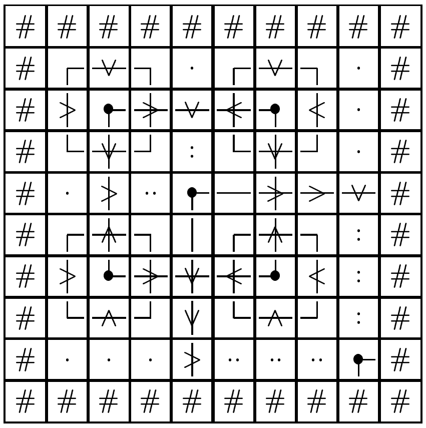
We use $e : \{a,b,c\} \mapsto \{a,b,c,d\}^{2,2}$ with $e(x) = \begin{array}{|c|c|}\hline x & d \\ \hline d & d \\ \hline \end{array}$ for $x \in \{a,b,c\}$. In order to show that $L_{=}^{c}$ and thus $L_{=} = L_{=}^{c} \cap \{a,b\}^{*,*}$ is recognizable, it remains to show in Lemma 6 that $e(L_{=}^{c})$ is recognizable, which means that we only have to care about a's and b's on positions with a odd row and column number by intersecting with the recognizable language $\{\begin{array}{|c|c|}\hline x & d \\ \hline d & d \\ \hline \end{array} \mid x \in \{a,b,c\}\}^{*,*}$.

4 The skeleton for squares of the power of 2

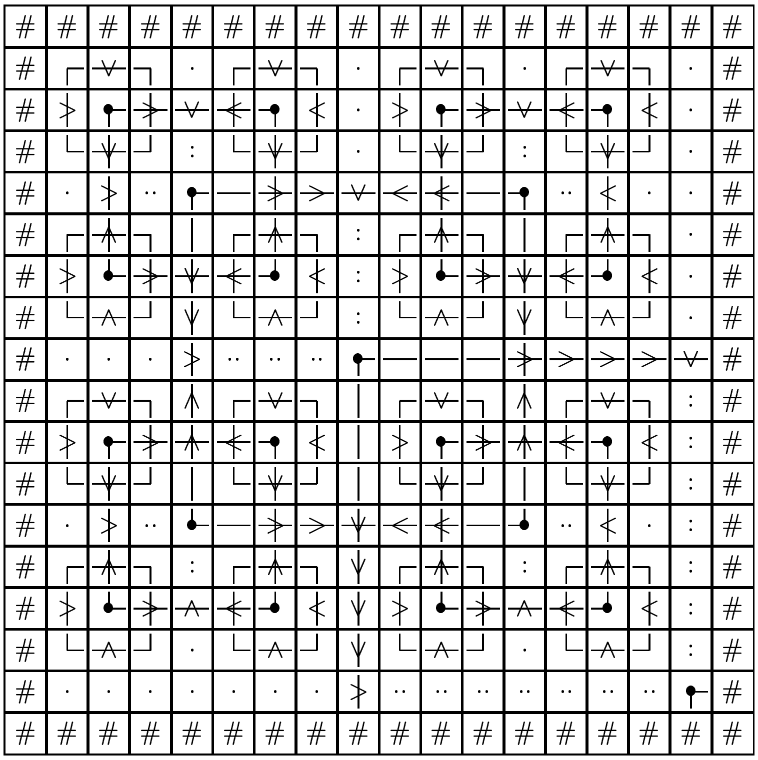
The *skeleton* describes a square, where each corner is surrounded by a square of half the size. The skeleton is described as a hv-local language L_S over the alphabet Σ_S = {⌝, ⌜, ⌞, ⌟, ⫯, ⫯, ⫰, ⫰, ·, ··, :, ⩛, ⩚, ⪥, ⪤, ⩛, ⩚, ⪥, ⪤, |, ⩛, ⩚, —, <, >}. Since the last section we are particularly interested in the intersection $L_S \cap L_R := \{p_1, p_2, ...\}$ with the recognizable language $L_R = \{\begin{array}{|c|c|}\hline x & y \\ \hline z & w \\ \hline \end{array} \mid x \in$ {⌝, ⌜, ⌞, ⌟}, $y, z, w \in$ {⫯, ⫯, ⫰, ⫰, ·, ··, :, ⩛, ⩚, ⪥, ⪤, ⩛, ⩚, ⪥, ⪤, |, ⩛, ⩚, —, <, >}$\}^{*,*}$. The first 4 examples are

$\hat{p}_1 =$ 

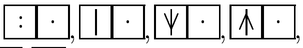
$\hat{p}_2 =$ 

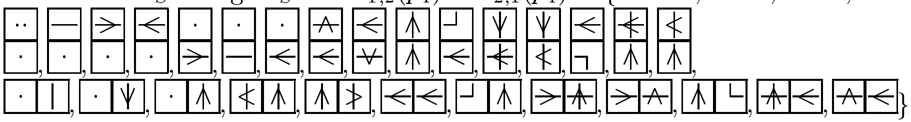
$\hat{p}_3 =$ 

and



$\hat{p}_4 =$

We define L_S using $\Delta_S := T_{1,2}(\hat{p}_4) \cup T_{2,1}(\hat{p}_4) \cup \{$ 

 $\}$

First we will show by induction that for every i a picture p_i of size $(2^i, 2^i)$ is in $L_S \cap L_R$:

Consider p_3 as the base of induction and p_4 as an example for a step of induction. Except from the right and lower edge, where the picture meets the #'s of the frame, Δ_S has all symmetries of a square. The upper left quarter of p_{i+1} is exactly p_i. Furthermore the 3 sub-pictures of size $(2^i - 1, 2^i - 1)$ starting with $p_{i+1}(1, 2^i + 1) =$ ┌, $p_{i+1}(2^i + 1, 1) =$ ┌ and $p_{i+1}(2^i + 1, 2^i + 1) =$ ┌ are rotations of the sub-picture of size $(2^i - 1, 2^i - 1)$ starting with $p_i(1, 1) =$ ┌ around $p_{i+1}(2^i, 2^i) =$ ┳. Now Δ_S allows to continue the 2^i-th row after ┳ with —'s until the ⇸ at the column with the only ⩛ at the lower edge of the upper right sub-picture. From now on continue with ⇾'s until the ⩛ at the last column, which had .'s so far and continues with :'s until the ┳ in the lower right corner. (Column 2^i and last row analogously).

The opposite direction, that for every i exactly one picture p_i of size $(2^i, 2^i)$ is in $L_S \cap L_R$, follows (considering the only possibility for the right and lower edge) from the following lemma:

Lemma 3. *For every picture $p \in L_S \cap L_R$, for every $i > 1$, $r, c \geq 0$ the sub-picture q of size $(2^i - 1, 2^i - 1)$ starting at $p(c2^i + 1, r2^i + 1)$ has up to 90^o-rotation the following shape at the outside: The upper row is periodicly filled by $q(t, 1) =$ ┌ (resp. ⩛, ┐, ·) if $t Mod 4 = 1$ (resp. 2,3,0) and $t < 2^i$. The same holds in 90^o-rotation symmetric manor except that $q(2^{i-1}, 2^i - 1) =$ ⩔ and $q(2^i - 1, 2^{i-1}) =$ ⋗ if $i > 2$ and $q(2^{i-1}, 2^i - 1) =$ ↡ and $q(2^i - 1, 2^{i-1}) =$ ↠ if $i = 2$.*

On http://www-fs.informatik.uni-tuebingen.de/~reinhard/picgen.html is a Java program, which demonstrates that in most cases the symbol is determined by its left and upper neighbors; a wrong choice in the remaining cases will sooner or later result in an unsolvable case at another position. The program is interactive and could help the reader to understand the following proof.

Proof. Base of induction for $i = 2$: Assume furthermore by induction on r and c, that the left neighbor of $q(1, 1)$ is not in {⩚, ↟, ⩛, ↡} (it is # for $c = 0$) and that the upper neighbor of $q(1, 1)$ is not in {⇤, ↞, ⇥, ↠} (it is # for $r = 0$). Since $q(1, 1) =$ ┐ would by Δ_S require the left neighbor in {⩛, ↟}, $q(1, 1) \in$ {└, ┘} would by Δ_S require the upper neighbor in {⇤, ↞, ⇥, ↠} and $q(1, 1) \in$ {┐, ┌, └, ┘}, we have $q(1, 1) =$ ┌. Consequently $q(2, 1) \in$ {⩛, ↟}, $q(1, 2) \in$ {⇥, ↞}, each of the 4 combinations forcing $q(2, 2) =$ ┬, ├, ┴ or ┤ furthermore $q(3, 1) =$ ┐, $q(1, 3) =$ └, thus one of the 4 combinations of $q(2, 3) \in$ {⩚, ↡} and $q(3, 2) \in$ {⇤, ↠} and at last $q(3, 3) =$ ┘. From $q(3, 1) =$ ┐ it follows that its right neighbor is not in {⩚, ↟, ⩛, ↡} which is needed for $c + 1$ and analogously the lower neighbor of $q(1, 3) =$ └ is not in {⇤, ↞, ⇥, ↠} which is sufficient for $r + 1$. The two shanks of $q(2, 2)$ point to the exception of the outside shape.

Step from i to $i + 1$: Assume furthermore by induction on r and c, that the left neighbors of $q(1, 2^{i-1})$ and $q(1, 2^i + 2^{i-1})$ are in {#, ·, :, |, ⩔, ⩓} and the left neighbor of $q(1, 2^i)$ is in {#, ·, :, |, ⩔, ⩓, ⩚, ↟, ⩛, ↡, ⋗} (it is # for $c = 0$) and that the upper neighbors of $q(2^{i-1}, 1)$ and $q(2^i + 2^{i-1}, 1)$ are in {#, ·, ··, —, ⋖, ⋗} and the upper neighbor of $q(2^i, 1)$ is in {#, ·, ··, —, ⋖, ⋗, ⇤, ↞, ⇥, ↠, ⩔} (it is # for $r = 0$). For $i = 2$ this forces by induction $q(1, 2^i + 2^{i-1}) = q(1, 6) = q(1, 2^{i-1}) = q(1, 2) =$ ⇥ and thus $q(2^i - 1, 2^i + 2^{i-1}) = q(3, 6) = q(2^i - 1, 2^{i-1}) = q(3, 2) =$ ↠. For $i > 2$ this forces $q(1, 2^i + 2^{i-1}) = q(1, 2^{i-1}) = \cdot$ and thus by induction $q(2^i - 1, 2^i + 2^{i-1}) = q(2^i - 1, 2^{i-1}) =$ ⋗. Consequently $q(2^i, 2^i + 2^{i-1}), q(2^i, 2^{i-1}) \in$ {⩛, ↡, ⩚, ↟, ⋗}. For $i = 2$ this forces by induction $q(2^i + 1, 2^i + 2^{i-1}) = q(5, 6) = q(2^i + 1, 2^{i-1}) = q(5, 2) =$ ↞ and thus $q(2^{i+1} - 1, 2^i + 2^{i-1}) = q(7, 6) = q(2^{i+1} - 1, 2^{i-1}) = q(7, 2) =$ ⇤. For $i > 2$ this forces $q(2^i + 1, 2^i + 2^{i-1}) = q(2^i + 1, 2^{i-1}) =$ ⋖ and thus by induction $q(2^{i+1} - 1, 2^i + 2^{i-1}) = q(2^{i+1} - 1, 2^{i-1}) = \cdot$. Thus their right neighbors are in {·, :, |, ⩔, ⩓} which is sufficient for $c+1$. Together with the analogous direction exchanging the coordinates this means that each of the 4 sub i-pictures of the $i + 1$-sub-picture q has its exceptional side hidden inside q. Since $q(1, 2^i - 1) =$ └, considering the possible left neighbors leads to $q(1, 2^i) \in$ {·, ⋗, ⋖}. The periodical contents of the rows $2^i - 1$ and $2^i + 1$ only allows to continue row 2^i with the same symbol until column 2^{i-1}, where $q(2^{i-1},2^i) \in$ { ⧐ , ⇸, ⧏ , ⇷}. This allows only the combination $q(1,2^i) = ... = q(2^{i-1}-1,2^i) = \cdot$ and $q(2^{i-1},2^i) =$ ⧐ resp. $q(1,2^i) = ... = q(2^{i-1}-1,2^i) =$ ← and $q(2^{i-1},2^i) =$ ⇷, which has to be continued with $q(2^{i-1}+1,2^i) = ... = q(2^i-1,2^i) = \cdot\cdot$ resp. —. Depending on the analogous column 2^i we get $q(2^i,2^i) \in$ { ┏, ┗} resp. $q(2^i,2^i) \in$ {┓ , ┛ } and further have to continue with $q(2^i+1,2^i) = ... = q(2^i+2^{i-1}-1,2^i) =$ — resp. $\cdot\cdot$, $q(2^i+2^{i-1},2^i) =$ ⇸ resp. ⧏ and with $q(2^i+2^{i-1}+1,2^i) = ... = q(2^{i+1}-1,2^i) =$ → resp. $\cdot$. Together with the analogous column 2^i this completes the description of q. The right (resp. lower) neighbor of $q(2^{i+1}-1,2^i) \in$ {→, $\cdot$} (resp. $q(2^i,2^{i+1}-1) \in$ { ⩛ , $\cdot$}) must be in {$\cdot$, $:$, $|$, ⩛ , ⩚ , ⊼, ⇞, ⊻, ⇟, →} (resp. {$\cdot$, $\cdot\cdot$, —, ←, →, ⧏ , ⇷, ⧐ , ⇸, ⩛ }), which is sufficient for $c+1$ (resp. $r+1$).

5 The counting flow for squares of the power of 2

Lemma 4. *$e(L_{=}^{c}) \cap \bigcup_i \Sigma^{2^i,2^i}$ is recognizable.*

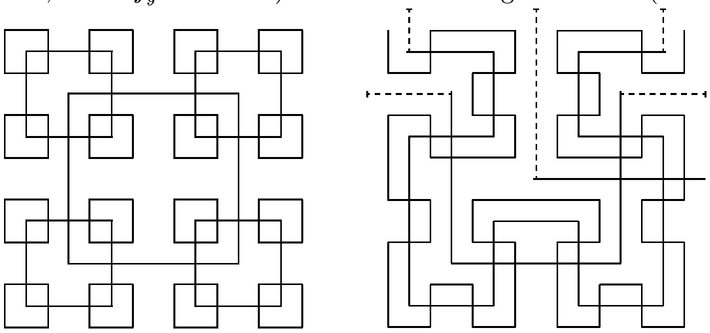
Proof. We define a language L_F in the following and show $e(L_{=}^{c}) \cap \bigcup_i \Sigma^{2^i,2^i} = L_F \cap e(\{a,b,c,\}^{*,*})$. We give each flow from one cell to its neighbor a capacity from -9 to 9 by defining $\Sigma_F := \Sigma_S \times \{-9,-8,...,9\}^4$. Furthermore we allow only those symbols $(x,l,r,u,d) \in \Sigma$ fulfilling:
$\pi(x,l,r,u,d) := a$ if $x \in$ {┐ , ┌, └, ┘ } $\wedge\, l+r+u+d = 1$,
$\pi(x,l,r,u,d) := b$ if $x \in$ {┐ , ┌, └, ┘ } $\wedge\, l+r+u+d = -1$,
$\pi(x,l,r,u,d) := c$ if $x \in$ {┐ , ┌, └, ┘ } $\wedge\, l+r+u+d = 0$,
$\pi(x,l,r,u,d) := d$ if $x \in$ {┓ , ┏, ┗, ┛ } $\wedge\, l+r+u+d = 0$,
or $x \in$ {$\cdot$, $\cdot\cdot$, $:$, $|$, ⩛ , ⩚ , —, ←, →} $\wedge\, l=-r \wedge u=-d$,
or $x \in$ {⊻, ⊼, ⇟, ⇞} $\wedge\, l+r+4u+4d = 0$,
or $x \in$ { ⧏ , ⧐ , ⇷, ⇸} $\wedge\, 4l+4r+u+d = 0$. The tiling

$$\Delta_F := \{\begin{array}{|c|}\hline f\\\hline g\\\hline\end{array} \mid \begin{array}{|c|}\hline f_1\\\hline g_1\\\hline\end{array} \in \Delta_S, f=(f_1,l,r,u,d), g=(g_1,l',r',d,d')\}$$
$$\cup\ \{\begin{array}{|c|c|}\hline f & g\\\hline\end{array} \mid \begin{array}{|c|c|}\hline f_1 & g_1\\\hline\end{array} \in \Delta_S, f=(f_1,l,r,u,d), g=(g_1,r,r',u',d')\}.$$

takes care of the continuation of the flow (additionally we have the tiles with # having flow 0). The symbols ┐ , ┌, └, ┘ allow sources and sinks of the flow; they only occur in odd rows and columns and therefor have the order 1; ┓ , ┏, ┗, ┛ occur where the flow in the column and the row have the same order; $\cdot$, $\cdot\cdot$, $:$, $|$, ⩛ , ⩚ , —, ←, → occur where the flow in the column and the row have a completely different order and ⊻, ⊼, ⇟, ⇞, ⧏ , ⧐ , ⇷, ⇸ occur where a rectangle or its elongation meets a rectangle of half the size which means that a carry of the counter can take place here. In general for any j and i the $2^{i-1}+j\cdot 2^i$ -th row resp. column have the order 4^i (if they are in the picture). Thus from the existence of a flow follows that the number of sources and sinks (a's and b's) must be equal. A picture in $e(\{a,b,c,\}^{*,*})$ has its pre-images in L_R and thus Lemma 3 makes sure that the projection to the first component in the 5-tuples has the correct structure, which means that $L_F \cap e(\{a,b,c,\}^{*,*}) \subseteq e(L_{=}^{c}) \cap \bigcup_i \Sigma^{2^i,2^i}$.

For the other direction we have to show that for any picture of size $(2^i, 2^i)$ with an equal number of sources and sinks (w.l.o.g. on positions on odd rows and columns), we can construct at least one corresponding flow:

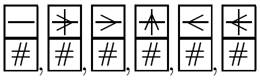
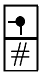
Herefor we use the Hilbert-curve, where each corner point $(2^{i-1}+j_x\cdot 2^i, 2^{i-1}+j_y\cdot 2^i)$ of the curve having order 4^i is surrounded by 4 corner points $(2^{i-1}+j_x\cdot 2^i \pm 2^{i-2}, 2^{i-1}+j_y\cdot 2^i \pm 2^{i-2})$ of the curve having order 4^{i-1} (see also [NRS97]).



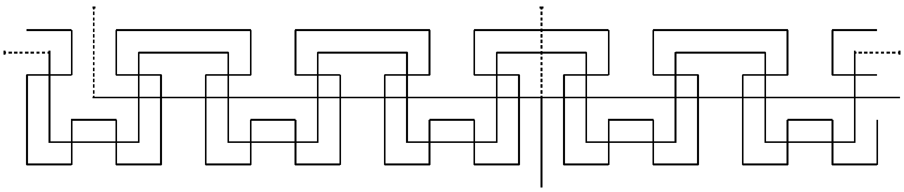
A curve of each order uses 3 lines of a square and then one elongation line to get to the next square. Hereby at least one of the 3 lines crosses the curve of the next higher order. (If it crosses it a second time, we ignore the second crossing in the following consideration.) Now we construct the flow along the curve according to the following rules: If a flow of more than 3 resp. less than -3 crosses a flow of the next higher order or if a flow of more than 0 resp. less than 0 crosses a negative resp. positive flow of the next higher order, it is decreased resp. increased by 4 and the flow of the next higher order is increased resp. decreased by one. We may assume by induction that a curve when it is entering a square, has a flow $\in [-6, 6]$. After at most 3 times crossing the curve of the next lower order (which could bring the flow for example to -9 or 9), it will cross the curve of the next higher order bringing the flow to $[-5, 5]$. Since we have to consider 4 crossings in the square, the first condition of the rule makes sure that the curve also leaves the square with a flow between -6 and 6 and thus never exceeds its capacity. The second condition of the rule makes sure that a small total flow will find itself represented in curves with low order, which is important towards the end of the picture.

6 The generalization

Lemma 5. $e(L^c_{=}) \cap \bigcup_{i,j} \Sigma^{j2^i,2^i}$ *is recognizable.*

Proof. Adding the tiles  and  to Δ_S allows skeleton-pictures of size $(j2^i, 2^i)$ for any $j > 0$.

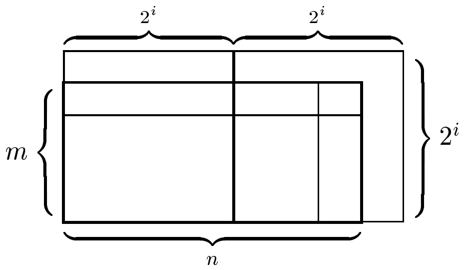
The counting flow described in the previous section can be continued from one square to the next like the following picture illustrates:



But what can we do with a flow (of order 4^i) reaching the bottom line? The idea is to combine the method with the method of Section 2 by using $\Sigma_e := \Sigma_F \times \{-1, 0, 1\}^4$ as new alphabet designing Δ_e in such way that a transfer of the flow from one system to the other is allowed at those symbols and , which occur at the bottom line. The r-th row (from the bottom) can now have flows of the order $4^i \cdot 2^{r-1}$. This allows to represent numbers up to $4^i \cdot 2^{2^i}$ and thus recognize pictures of size $(j2^i, 2^i)$ for any $0 < j < 2^{2^i}$ (resp. k^{2^i}) with the same number of a's and b's.

Lemma 6. *$e(L_{=}^{c})$ is recognizable.*

Proof. For the general case of pictures of size (m, n), where we assume w.l.o.g. $m > n$ we choose the smallest i with $2^i \geq n$ and the smallest j with $j2^i \geq m$. Then since $2n > 2^i$ and $2m > j2^i$ a picture of size $(j2^i, 2^i)$, which was recognized with the method described so far, can be folded twice and get the size (m, n). This folding can be simulated using an alphabet $\Sigma_g := \Sigma_e^4$, where the first layer corresponds to the picture and the other 3 layers may contain the simulated border consisting of # and parts of the flow but no sinks and sources (this means only c's and d's). Δ_g simulates Δ_e on each layer by additionally connecting layer 1 with 4 and 2 with 3 at the top border and 1 with 2 and 3 with 4 at the right border.



Remark: In the same way we can furthermore generalize this counting method to n-dimensional recognizable 'picture'-languages by folding $L_{=}$ into n dimensions like in [Bor99].

7 Outlook

It remains the question for a deterministic counting method on pictures. Obviously this can not be done using the deterministic version of on-line tessalation acceptors [IN77] as model, since the automaton can not handle the number occurring in the last line. But a good candidate is the notion of deterministic recognizability in [Rei98]. At least in the case of squares of the power of 2, a construction of the skeleton along the Hilbert-curve should be possible, but working out the details will be hard.

References

[Bor99] B. Borchert. A formal languages characterization of NP. Manuscript at http://math.uni-heidelberg.de/logic/bb/papers/NP-char.ps, 1999.

[Für82] Martin Fürer. The tight deterministic time hierarchy. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pages 8–16, San Francisco, California, 5–7 May 1982.

[GR96] D. Giammarresi and A. Restivo. Two-dimensional languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Language Theory*, volume III. Springer-Verlag, New York, 1996.

[GRST94] Dora Giammarresi, Antonio Restivo, Sebastian Seibert, and Wolfgang Thomas. Monadic second-order logic over pictures and recognizability by tiling systems. In P. Enjalbert, E.W. Mayr, and K.W. Wagner, editors, *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science, STACS 94 (Caen, France, February 1994)*, LNCS 775, pages 365–375, Berlin-Heidelberg-New York-London-Paris-Tokyo-Hong Kong-Barcelona-Budapest, 1994. Springer-Verlag.

[IN77] K. Inoue and A. Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, 13:95–121, 1977.

[LS97a] M. Latteux and D. Simplot. Context-sensitive string languages and recognizable picture languages. *Information and Computation*, 138(2):160–169, 1 November 1997.

[LS97b] M. Latteux and D. Simplot. Recognizable picture languages and domino tiling. *Theoretical Computer Science*, 178(1-2):275–283, 1997. Note.

[Mat97] Oliver Matz. Regular expressions and context-free grammars for picture languages. In *14th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1200 of *lncs*, pages 283–294, Lübeck, Germany, 27 February–March 1 1997. Springer.

[Mat98] Oliver Matz. On piecewise testable, starfree, and recognizable picture languages. In Maurice Nivat, editor, *Foundations of Software Science and Computation Structures*, volume 1378 of *Lecture Notes in Computer Science*, pages 203–210. Springer, 1998.

[NRS97] R. Niedermeier, K. Reinhardt, and P. Sanders. Towards optimal locality in mesh-indexings. In L. Czaja B.S. Chlebus, editor, *Proceedings of the FCT'97*, LNCS 1279, pages 364–375. Springer, sept. 1997.

[Rei98] K. Reinhardt. On some recognizable picture-languages. In L. Brim, editor, *Proceedings of the 23th Conference on Mathematical Foundations of Computer Science*, number 1450 in Lecture Notes in Computer Science, pages 760–770. Springer-Verlag, August 1998.

[Wil97] Thomas Wilke. Star-free picture expressions are strictly weaker than first-order logic. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *Automata, Languages and Programming*, volume 1256 of *Lect. Notes Comput. Sci.*, pages 347–357, Bologna, Italy, 1997. Springer.