# Regular Expressions and Context-Free Grammars for Picture Languages

Oliver Matz

Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Univertität, Olshausenstraße 40, D-24098 Kiel
oma@informatik.uni-kiel.de

**Abstract.** We introduce a new concept of regular expression and context-free grammar for picture languages (sets of matrices over a finite alphabet) and compare and connect these two formalisms.
Keywords: formal languages, pictures languages, grammars, regular expressions.

## 1 Introduction

Many attempts have been made to generalize the definitions and results of the theory of formal word languages to other, more complex objects than words, e.g. traces, graphs, and trees. One possible generalization are *pictures* (matrices over a finite alphabet, e.g. two-dimensional words). Sets of pictures will be called *picture languages* or simply *languages*. Picture languages have been investigated by many authors; a comprehensive survey is [GR96]; another collection of references can be found for example in [Sir87].

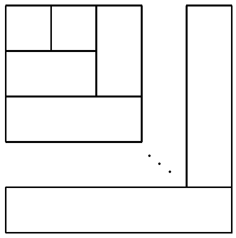
It is a not finished subject to transfer results of the theory of word languages to picture languages. Here we present a new approach based on regular expressions and context-free grammars.

In [GRST96] the concept of *tiling systems* as a device for recognizing picture languages is investigated. Tiling systems are a possible generalization of the concept of finite automata for word languages. In [GRST96] it is shown that their expressive power is equal to that of formulas of existential monadic second order theory over pictures, so the close relation of automata and logic carries over from the theory of word languages to picture languages and gives a somewhat "robust" class of picture languages – the *recognizable picture languages*.

Regular expressions are a device for the definition of word languages that is more difficult to transfer to picture languages. A straightforward adaption of regular expressions to pictures — we will call them "simple" — is studied in [GR92,GRST96,GR96]. Simple expressions use two partial concatenations named row- and column concatenation (denoted by $\ominus$ and $\oplus$), which put their arguments vertically above each other (horizontally next to each other, resp.), provided they have the same width (height, resp.). Additionally, both of these concatenations may be iterated. But the expressive power of such expressions is much weaker than that of tiling systems, so the Kleene theorem cannot be carried over to the theory of picture languages.

In this paper we suggest a more powerful type of regular expressions for picture languages, the so-called *regular expressions with operators*. Inside these expressions we allow the Kleene star to range over more complex combinations of juxtapositions, unions, and even intersections of concatenations.

As an informal example, let us assume we have two expressions $r$ and $s$ that generate all columns, i.e. pictures of width 1, and all rows, i.e. pictures of height 1, resp. We consider the column-concatenation with $r$ (let us denote it by $(\varobar r)$) as an individual object. It will enlarge a picture that it is applied to by one column. The row-concatenation with $s$ (denoted $(\ominus s)$) will enlarge its argument by one row. Now, if we allow a Kleene star to iterate the juxtaposition of these two so-called operators, we get another operator, $((\varobar r)(\ominus s))^*$, which enlarges its argument alternatingly by one row and one column a finite number of times. If this operator is applied to the expression generating all $1 \times 1$-squares, we obtain a regular expression with operators that generates all squares, as illustrated by the figure.



We will distinguish different classes of expressions with operators, depending on how operators may be constructed. Given appropriate constraints, these expressions do not exceed the expressive power of tiling systems; but they are more powerful than simple regular expressions (the set of squares is not definable by a simple regular expression, see [GR92,GRST96,GR96]). For one particular class of expressions with operators it remains open whether it exhausts the class of recognizable languages.

We will also try to transfer the concept of context-free grammar from word languages to picture languages. Our approach for this differs very much from the one in [Sir87]. In our grammars, sentential forms are terms built by the two binary concatenation symbols $\ominus$, $\varobar$ as well as terminals and non-terminals, which are used as constant symbols. A rule has a non-terminal on the left hand side and a sentential form on the right hand side. The derivation proceeds as follows: One starts with the start symbol and replaces repeatedly a non-terminal $A$ by the right hand side of an $A$-rule until a sentential form is reached that consists entirely of terminal symbols. If this can be evaluated to a picture, this picture is generated.

We characterize the expressive power of certain regular expressions with operators by context-free grammars with a certain constraint concerning the way recursion is allowed. This result corresponds in a way to the classical equivalence of regular word expressions to right-linear word grammars. This is why we think that our definition of "regular picture languages" by expressions with operators gives another natural and robust class of picture languages that is worth studying.

More detailed proofs can be found in [Mat95].

## 2 Basic Notions

Throughout the paper we consider a fixed finite alphabet $\Sigma$. A *picture* over $\Sigma$ of *size* $(m, n)$ (where $m, n \geq 1$) is a $m \times n$-matrix over $\Sigma$. For a picture $P$ of size $(m, n)$, we define $\underline{\overline{P}} = m$ and $|P| = n$. We denote the set of all pictures over $\Sigma$ by $\Sigma^+_+$.

Next we will define a row- and a column-concatenation for pictures. Let $P, Q$ be pictures of size $(k, l)$, $(m, n)$ respectively.

If $k = m$, then the column concatenation $P \varobar Q$ of the two pictures is the $k \times (n+l)$-picture obtained by appending $Q$ to the right of $P$.

Analogously, in case $l = n$ their row concatenation $P \ominus Q$ is defined as the $(k+m) \times l$-picture of obtained by appending $Q$ to the bottom of $P$.

These partial concatenations can be extended to languages as usual, i.e. for $L, M \subseteq \Sigma^+_+$ we define $L \varobar M = \{P \varobar Q \mid P \in L, Q \in M\}$. These concatenations can be iterated: For a language $L \subseteq \Sigma^+_+$ we set $L^{\varobar 1} := L$ and $L^{\varobar(i+1)} := L^{\varobar i} \varobar L$. Now the *column closure* of $L$ is defined as $L^{\varobar +} := \bigcup_{i \geq 1} L^{\varobar i}$. The *row closure* is defined analogously.

If no ambiguity arises, we denote the column concatenation $L \varobar M$ of two languages $L$, $M$ by $(L\, M)$, and similarly use $\begin{pmatrix} L \\ M \end{pmatrix}$ instead of $L \ominus M$. The iterated column concatenation may be written as $L^i$ and $L^+$ instead of $L^{\varobar i}$ and $L^{\varobar +}$, resp., whereas $L_i$ and $L_+$ denote $L^{\ominus i}$ and $L^{\ominus +}$, resp. The latter notion will only be used in case no conflict with indices occurs.

## 3 Regular Expressions with Operators

The set $\cap$-$REG(\Sigma)$ of *simple regular expressions over* $\Sigma$ with typical element $r$ is defined by the following BNF-style rules:

$$r ::= a \mid (r_1 \cup r_2) \mid (r_1 \ominus r_2) \mid (r_1 \varobar r_2) \mid (r_1 \cap r_2) \mid r^{\varobar +} \mid r^{\ominus +}$$

Here $a$ stands for an arbitrary letter from $\Sigma$. The language generated by such an expression is defined in a straightforward way: For all $a \in \Sigma$ let $\mathcal{L}(a) = \{a\}$ (the singleton of the $1 \times 1$-picture $a$), and for two expressions $r$ and $s$ we define $\mathcal{L}(r \ominus s) = \mathcal{L}(r) \ominus \mathcal{L}(s)$ an so on. The subset of $\cap$-$REG(\Sigma)$ of *monotonic* expressions (i.e. expressions without intersection symbol) will be denoted by $REG(\Sigma)$. The classes of languages definable by such expressions will be denoted by the corresponding calligraphic notations $\cap$-$\mathcal{REG}(\Sigma)$ and $\mathcal{REG}(\Sigma)$.

In these and similar cases we will omit the explicit mentioning of $\Sigma$ if possible. We will omit brackets inside expressions following the usual conventions, i.e. ${}^{\varobar +}$ and ${}^{\ominus +}$ bind stronger than concatenation symbols, which bind in turn stronger than union and intersection symbols.

*Example 1.* Consider the language $L$ that consists of the set of all pictures such that there is one row and one column (both not at the border) that hold $b$'s and the remainder of the pictures is filled with $a$'s. $L$ is generated by the expression

$$\begin{pmatrix} a_+^+ \\ b^+ \\ a_+^+ \end{pmatrix} b_+ \begin{pmatrix} a_+^+ \\ b^+ \\ a_+^+ \end{pmatrix} \cap \begin{pmatrix} (a_+^+ \, b_+ \, a_+^+) \\ b^+ \\ (a_+^+ \, b_+ \, a_+^+) \end{pmatrix}.$$

The fact that the set of squares over a one-letter alphabet is not generated by a simple expression is an immediate consequence of the following characterization of the class $\cap$-$\mathcal{REG}(\Sigma)$.

**Theorem 2.** *(see [Mat95].) A language $L \subseteq \{a\}_+^+$ is in $\cap$-$\mathcal{REG}$ iff it is in $\mathcal{REG}$ iff the set $\{(m,n) \mid a_n^m \in L\}$ is a finite union of Cartesian products of ultimately periodic subsets of $\mathbb{N}_{\geq 1}$.*

The above theorem is an analogue to the known fact that a set $N$ of integers is ultimately periodic iff the set $\{a^n \mid n \in N\}$ is a regular word language. (Note that this theorem remains true even when we allow also complementation symbols inside regular expressions.)

Theorem 2 shows that the expressive power of *REG* is very limited. But note that on the other hand any picture language that is recognizable by tiling systems in the sense of [GRST96,GR92] is the projection of a picture language that is generated by an expression from $\cap$-*REG* over a possibly larger alphabet. But we think that the use of intersection and projections disturb in a way the "assembling character" of regular expressions.

That is why we investigate another type of regular expressions whose main idea is to allow the Kleene star to range over more complex juxtapositions of concatenations. Before we give the syntax and semantics of these expressions, let us consider again the example of the introduction. The crucial point was to consider terms such as $(\varobar a_+)$, the concatenation with a column of $a$'s to the right, as individual objects, which may be either applied to expressions or composed with each other by juxtaposition, iteration, and union.

With this intuitive idea of the class of regular expressions with operators, we will give a more formal definition. For the sake of maximal generality we will allow the intersection and union symbols as well, both in "expressions" and in "operators".

**Definition 3.** The set $\cap$-$REG^{UOP}(\Sigma)$ of regular expressions with typical element $r$ and the set $\cap$-$UOP(\Sigma)$ of unrestricted regular operators with typical element $\varrho$ are defined by the following BNF-style rules:

$$r ::= a \mid (r_1 \ominus r_2) \mid (r_1 \varobar r_2) \mid r_1{}^{\ominus+} \mid r_1{}^{\varobar+} \mid (r_1 \cup r_2) \mid (r_1 \cap r_2) \mid r\varrho$$
$$\varrho ::= (\ominus r) \mid (\varobar r) \mid (r\ominus) \mid (r\varobar) \mid (\varrho_1 \varrho_2) \mid \varrho^* \mid \varrho_1 \cup \varrho_2 \mid \varrho_1 \cap \varrho_2$$

Again, $a$ stands for an arbitrary element from $\Sigma$.

As before, we drop the "$\cap$-" prefix in the respective notation to denote the classes of *monotonic* expressions and operators, i.e. ones that do not have intersection symbols in it.

Before we give the semantics of these type of expressions, let us consider another example. The expression $ab((a\oplus)(\oplus b))^*$ describes the set of all words (= pictures with one single row) that result from $ab$ by appending repeatedly one $a$ to the left and one $b$ to the right, i.e. the language $\{a^ib^i \mid i \geq 1\}$. Since this word language is non-regular, we make the disappointing observation that we leave the class of recognizable languages if we allow the iteration in such an unrestricted way. Since our aim was to find suitable extensions of the concept of regular expressions to pictures, we shall put a certain constraint on the way operators may be juxtaposed in order to ensure that the resulting language is recognizable by tiling systems.

The crucial point for this constraint is that we make sure that an operator that "works to the right" (like $(\oplus b)$) is never juxtaposed, united, or intersected with another operator that "works to the left" (like $(a\oplus)$), and similarly top- and bottom-operators are seperated. On the other hand, we allow the combination of, say, bottom and right, so our example regular expression $a((\oplus a_+)(\ominus a^+))^*$ for the set of squares is still o.k. We will call operators that meet this constraint *restricted*. This is put formal by the following definition, in which the decoration symbols $r$, $l$, $b$ and $t$ (for "right", "left", "bottom" and "top", resp.) are used in order to indicate in which direction an operator potentially enlarges a picture that it is applied to.

**Definition 4.** The set $\cap$-$REG^{ROP}(\Sigma)$ (set of expression with restricted operators) with typical element $r$ and the sets $ROP_{\mathfrak{d}}(\Sigma)$ for every $\mathfrak{d} \in \{br, bl, tl, tr\}$ (set of restricted $\cap$-regular operators for direction $\mathfrak{d}$) with typical element $\varrho^{\mathfrak{d}}$ are defined by the following BNF-style rules:

$$\begin{aligned}
r &::= a \mid (r_1 \ominus r_2) \mid (r_1 \oplus r_2) \mid r^{\ominus+} \mid r^{\oplus+} \mid (r_1 \cup r_2) \mid (r_1 \cap r_2) \mid r\varrho \\
\varrho &::= \varrho^{br} \mid \varrho^{tr} \mid \varrho^{bl} \mid \varrho^{tl} \\
\varrho^{br} &::= (\ominus r) \mid (\oplus r) \mid (\varrho_1^{br}\varrho_2^{br}) \mid \varrho^{br\,*} \mid (\varrho_1^{br} \cup \varrho_2^{br}) \mid (\varrho_1^{br} \cap \varrho_2^{br}) \\
\varrho^{tr} &::= (r\ominus) \mid (\oplus r) \mid (\varrho_1^{tr}\varrho_2^{tr}) \mid \varrho^{tr\,*} \mid (\varrho_1^{tr} \cup \varrho_2^{tr}) \mid (\varrho_1^{tr} \cap \varrho_2^{tr}) \\
\varrho^{bl} &::= (\ominus r) \mid (r\oplus) \mid (\varrho_1^{bl}\varrho_2^{bl}) \mid \varrho^{bl\,*} \mid (\varrho_1^{bl} \cup \varrho_2^{bl}) \mid (\varrho_1^{bl} \cap \varrho_2^{bl}) \\
\varrho^{tl} &::= (r\ominus) \mid (r\oplus) \mid (\varrho_1^{tl}\varrho_2^{tl}) \mid \varrho^{tl\,*} \mid (\varrho_1^{tl} \cup \varrho_2^{tl}) \mid (\varrho_1^{tl} \cap \varrho_2^{tl})
\end{aligned}$$

Again, $a$ stands for a letter from $\Sigma$. The elements in $ROP_{\mathfrak{d}}(\Sigma)$ are called $\mathfrak{d}$-*regular operators* for every *direction* $\mathfrak{d} \in \{br, bl, tl, tr\}$. We also define $\mathfrak{d}$-regular operators for $\mathfrak{d} \in \{r, l, t, b\}$: For any $r \in \cap$-$REG^{ROP}$, the operator $(\oplus r)$ (or $(r\oplus)$, or $(\ominus r)$, or $(r\ominus)$, resp.) is an $r$-regular (or $l$-regular, or $b$-regular, or $t$-regular, resp.) elementary operator. For any $\mathfrak{d} \in \{r, l, t, b\}$, more complex $\mathfrak{d}$-regular operators are built by union, intersection, concatenation, or Kleene-star the way as above.

Now the definition of the semantics of expressions and operators is straightforward:

**Definition 5.** Two functions $[\![]\!] : \cap$-$REG^{UOP} \rightarrow 2^{\Sigma^+_+}$ and $[\![]\!] : \cap$-$UOP \rightarrow 2^{(\Sigma^+_+ \times \Sigma^+_+)}$ are defined simultaneously by induction over the structure of $\cap$-regular expressions and operators:

For $r, s \in REG^{UOP}$ and $\sigma, \varrho \in UOP$ let:

- $[\![a]\!] = \{a\}$ for all $a \in \Sigma$,
- $[\![(r \cup s)]\!] = [\![r]\!] \cup [\![s]\!]$, (similarly for $\cap$, $\oplus$ or $\ominus$ instead of $\cup$,)
- $[\![r^{\oplus+}]\!] = [\![r]\!]^{\oplus+}$, (similarly for $^{\ominus+}$ instead of $^{\oplus+}$,)
- $[\![r\varrho]\!] = \{R \in \Sigma^{+}_{+} \mid \exists P \in [\![r]\!] : (P, R) \in [\![\varrho]\!]\}$,
- $[\![(\oplus r)]\!] = \{(P, R) \mid P \in \Sigma^{+}_{+} \wedge R \in P \oplus [\![r]\!]\}$,
- $[\![(r\oplus)]\!] = \{(P, R) \mid P \in \Sigma^{+}_{+} \wedge R \in [\![r]\!] \oplus P\}$, (similarly for $\ominus$ instead of $\oplus$,)
- $[\![(\sigma\varrho)]\!] = [\![\sigma]\!] \circ [\![\varrho]\!]$, where $\circ$ denotes the usual relational product,
- $[\![(\sigma \cup \varrho)]\!] = [\![\sigma]\!] \cup [\![\varrho]\!]$, (similarly for $\cap$ instead of $\cup$,)
- $[\![\sigma^{*}]\!] = \bigcup_{i \in \mathbb{N}} [\![\sigma]\!]^{i}$.

Binary relations on $\Sigma^{+}_{+}$ will be referred to as *operations*. For an expression $r$ we denote the *pictures language generated by* $r$ also with $\mathcal{L}(r)$ instead of $[\![r]\!]$. We we will denote the class of languages (operations, resp.) generated by elements from a class of expressions (operators, resp.) by the corresponding calligraphic notation. Thus $\cap$-$\mathcal{REG}^{\mathcal{ROP}}(\Sigma)$ denotes the class of languages definable by expressions with restricted operators and so on.

In the context of regular expressions with operators, the symbols $^{\oplus+}$ and $^{\ominus+}$ become superfluous because for every regular expression $r$ one has $\mathcal{L}(r^{\oplus+}) = \mathcal{L}(r(\oplus r)^{*})$, and similarly for $\ominus$ instead of $\oplus$.

A picture language or an operation will be called *(monotonic)* $\cap$*-regular* if it is the denotation of a (monotonic) $\cap$-regular expression or operation. We make the following simple observation:

*Remark 6.* All of the above mentioned classes of languages are closed under rotation and reflection. The classes defined by monotonic expressions are closed under projection.

For the set of regular expressions with restricted operators one can show that every picture language generated by such an expression is recognizable by a finite tiling system, as stated in the following theorem.

**Theorem 7.** *Every language in* $\cap$-$\mathcal{REG}^{\mathcal{ROP}}(\Sigma)$ *is recognizable by tiling systems as defined in [GR92].*

We conjecture that the converse of Theorem 7 is not true, but we cannot show this.

### 3.1 Two More Examples

We use an example of [GRST96] to show that $\cap$-$\mathcal{REG}^{\mathcal{ROP}}$ is not closed under complement. In fact, the complement of a $\cap$-regular language need not even be recognizable:

*Example 8.* Let $\Sigma$ be a finite alphabet. Let $q = \Sigma((\ominus\Sigma^{+})(\oplus\Sigma_{+}))^{*}$, and

$$r = \bigcup_{a\neq b}\begin{pmatrix} a \\ \Sigma_{+} \\ b \end{pmatrix},\ s = \Sigma_{+}^{+} r \Sigma_{+}^{+},\ t = s \cap \begin{pmatrix} q \\ \Sigma^{+} \end{pmatrix},\ u = \begin{pmatrix} \Sigma_{+}^{+} \\ t \\ \Sigma_{+}^{+} \end{pmatrix} \cap \begin{pmatrix} q \\ q \end{pmatrix}.$$

$q$ generates the set of all squares; $s$ generates all pictures whose top and bottom row are different; $t$ all pictures which additionally have size $(n+1, n)$ for some $n$. Finally, $u$ generates all pictures of size $(2n, n)$ for some $n \geq 2$, in which there are two different rows with a square in between, i.e. all pictures of the form $\begin{matrix} P \\ Q \end{matrix}$ where $P$ and $Q$ are different squares larger than $1\times 1$. This language is not recognizable (see [GRST96]) and is thus (by Theorem 7) not in $\cap$-$\mathcal{REG}^{\mathcal{ROP}}$.

*Example 9.* Let $q$ as above. The expression $\Sigma((\oplus q)(\ominus(q \oplus q)))^{*}$ generates the set of all squares whose side length is a power of two.

To see this, note that $(\oplus q)(\ominus(q \oplus q))$ is an operator that will enlarge a $m\times n$-picture $P$, where $m+n$ is even, to an $(m+\frac{1}{2}(m+n))\times(n+m)$-picture with $P$ in its left upper corner. The expression $\Sigma$ generates all $1\times 1$-squares, so for all $i$, the expression $\Sigma((\oplus q)(\ominus(q \oplus q)))^{i}$ generates all $2^{i}\times 2^{i}$-pictures.

## 4 Context-Free Grammars

We will introduce a concept for context-free grammars that we consider a straightforward adaption to pictures languages. For the classes $\mathcal{REG}^{\mathcal{ROP}}$, $\mathcal{REG}$, and $\mathcal{REG}^{\mathcal{UOP}}$, we will find certain subclasses of context-free grammars of the same expressive power (*rank-$\{br, bl, tl, tr\}$-linear*, *rank-$\{r, l, t, b\}$-linear grammars*, or *rank-linear grammars resp.* The first and second of these classes yield — when restricted to words rather than pictures — a class of grammars that characterizes the regular word languages.

$$\begin{aligned} S &\to \begin{pmatrix} A \\ C \end{pmatrix} \Bigg|\ a \\ A &\to (S\,B) \\ B &\to \begin{pmatrix} b \\ B \end{pmatrix} \Bigg|\ b \\ C &\to (c\,C) \mid a \end{aligned}$$

Before we introduce our notion of context-free grammar for picture languages, we give a toy example that might be self explaining. Here, $S$ is the start symbol, and $A$, $B$, $C$ are other non-terminal symbols, and $a, b, c$ are the terminal symbols. $B$ yields the set of all columns of $b$'s, whereas $C$ yields the set of rows in $c^{*}a$. The alternating recursion of $S$ and $A$ makes sure that this grammar produces the set of squares over $\{a, b, c\}$ that have $a$'s on the diagonal, $c$'s above, and $b$'s below it. In order to obtain a grammar that produces all squares over the singleton $a$, one may replace all $b$'s and $c$'s by the letter $a$.

### 4.1 Sentential Forms, Grammars, Context-Free Languages

Trying to adapt the concept of (context-free) grammar from the theory of formal word languages to that of picture languages involves the following crucial problem: How can subpictures of a given picture be replaced by pictures of possibly different size? In order to avoid this problem, we do not use pictures as sentential forms, but terms built up by terminal and non-terminal symbols and the binary symbols $\ominus$ and $\varobar$.

**Definition 10.** Let $V$ be any alphabet. A *sentential form over $V$* is an element from $REG(V)$ in which only the connectives $\ominus$ and $\varobar$ (but not $\cup$, ${}^{\varobar+}$ and ${}^{\ominus+}$) occur. $SF(V)$ denotes the set of all sentential forms over $V$.

Note that the language generated by a sentential form $\alpha$ can have at most one element, which we will denote by $[\![\alpha]\!]$.

For example, possible sentential forms over $\{a, b, c, d\}$ are $(a \ominus b) \varobar (c \ominus d)$ and $(a \ominus (b \varobar c) \ominus d)$. The first one generates the the picture $\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}$, whereas the second does not generate a picture.

A context-free picture grammar will be defined very similarly to a word grammar. Derivation works the same way as for word grammars: One sentential form results from the preceding one by replacing a non-terminal with some corresponding right hand side, giving again a sentential form. The end of such a derivation is reached when there are only terminal symbols left. If this "terminal sentential form" can be evaluated to a picture, this picture is generated by the grammar.

**Definition 11.** A *context-free picture grammar* is a tuple $G = (N, \Sigma, \rightarrow, S)$, where $N$ is a finite set of *non-terminal symbols*, disjoint from the set $\Sigma$ of *terminal symbols*; $S \in N$ (the *start symbol*); and $\rightarrow \subseteq N \times SF(N \cup \Sigma)$ is a set of *rules*.

For a context-free grammar $G = (N, \Sigma, \rightarrow, S)$, we define the relation $\vdash_G$ on $SF(N \cup \Sigma)$ by $\beta \vdash_G \gamma$ iff there is some rule $(A, \alpha) \in \rightarrow$ such that $\gamma$ results from $\beta$ by replacing one occurrence of $A$ by $\alpha$. (We drop subscript $G$ if possible.) We denote the reflexive and transitive closure of $\vdash$ by $\overset{*}{\vdash}$.

Two grammars $G_1$, $G_2$ with the same terminal symbol set $\Sigma$ are called *strongly equivalent* iff for every terminal sentential form $\alpha \in SF(\Sigma)$ the equivalence $S_1 \overset{*}{\vdash}_{G_1} \alpha \iff S_2 \overset{*}{\vdash}_{G_2} \alpha$ holds.

We denote by $\mathcal{L}(G) = \{[\![\alpha]\!] \mid \alpha \in SF(\Sigma), S \overset{*}{\vdash} \alpha\}$ the picture language *generated by $G$*.

### 4.2 Limits of Context-Free Grammars

The following is proven similarly as the corresponding fact in the case of word languages:

*Remark 12.* Every context-free grammar is strongly equivalent to a context-free grammar in *Chomsky Normal Form (CNF)*, i.e. having only rules of the form $(A, B \ominus C)$, $(A, B \varobar C)$, and $(A, a)$, where $A, B, C$ are non-terminals and $a$ is a terminal.

The Chomsky Normal Form can be used to prove the following.

**Lemma 13.** *The language of Example 1 is not context-free.*

*Proof.* Consider a context-free grammar $G$ in CNF with $L \subseteq \mathcal{L}(G)$. We show $\mathcal{L}(G) \nsubseteq L$.

Provided that $n$ is sufficiently large, among the $n-2$ different $n\times n$-pictures $P_i$ over $\{a, b\}$ that have $b$'s exactly in the $i$-th row and the $i$-th column ($i \in \{2, \ldots, n-1\}$) there are two different ones, say $P_k$ and $P_l$, such that both of them can be derived from the same two-symbol sentential form $\alpha$, say $\alpha = C \mathbin{\text{⦶}} D$. (The case $\alpha = C \ominus D$ is analogous.)

Now we can choose decompositions $P_k = P \mathbin{\text{⦶}} R$ and $P_l = P' \mathbin{\text{⦶}} R'$ with $P, R, P', R' \in \Sigma^{+}_{+}$ such that $P, P'$ can be derived from $C$ and $R, R'$ can be derived from $D$. It is easy to see that $P \mathbin{\text{⦶}} R' \in \mathcal{L}(G) \setminus L$.

The above lemma shows that $\cap$-$\mathcal{REG} \nsubseteq \mathcal{CF}$ for two-letter alphabets. Together with Theorem 15 this shows that $\cap$-$\mathcal{REG}$ and $\mathcal{REG}^{\mathcal{ROP}}$ are incomparable.

### 4.3 Rank-Linear Grammars and Regular Expressions

Our aim is to find constraints for context-free grammars such that grammars with these constraints capture exactly the expressive power of monotonic expressions with unrestricted (or with restricted) operators.

These constraints are formalized in the definition of *rank-linear grammars*, which have some restriction on the way recursion is allowed. The main idea is to have some kind of ranking on the set of non-terminals and to require that the rank of the left hand side is larger than the rank of the non-terminals on the right hand side of a rule, except for at most one of them, which may have the same rank. If, additionally, this particular non-terminal is always "at the same place" – top-left, top-right, bottom-left or bottom-right – inside the right hand sides of all rules with "equally large" non-terminals on the left hand sides, then the grammar will be even $\{br, bl, tl, tr\}$-*rank-linear*. A corresponding definition applies for the directions from $\{r, l, t, b\}$.

**Definition 14.** Let $G = (N, \Sigma, \rightarrow, S)$ be a context-free grammar, $\leq$ be a preorder (i.e. a reflexive, transitive relation) on $N$. The equivalence relation $\leq \cap \geq$ is denoted by $\equiv$. Let $< := \leq \setminus \equiv$.

A rule $(A, \alpha)$ is *linear wrt.* $\leq$ if there is at most one occurrence of some non-terminal $B$ in $\alpha$ with $B \equiv A$ and all other non-terminals in $\alpha$ are $< A$.

If, additionally, there are $x \in \{l, r\}$ and $y \in \{t, b\}$ such that in subterms $\alpha_l \mathbin{\text{⦶}} \alpha_r$ of $\alpha$ the factor $\alpha_x$ does not contain this $B$ and analogously for $\ominus$ and $y$, then the rule $(A, \alpha)$ is $yx$-*linear wrt.* $\leq$. If all non-terminals in $\alpha$ are $< A$, then the rule $(A, \alpha)$ is $yx$-linear for any $x$ and $y$.

If a rule is $tx$- and $bx$-linear wrt. $\leq$, then it is $x$-*linear wrt.* $\leq$. If it is $yr$- and $yl$-linear wrt. $\leq$, then it is $y$-*linear wrt.* $\leq$.

The grammar $G$ is *rank-linear wrt.* $\leq$ if all rules are linear wrt. $\leq$. The grammar $G$ is called $\{br, bl, tl, tr\}$-*rank-linear wrt.* $\leq$ if for every $\equiv$-equivalence class $[A]$ there is a $\mathfrak{d} \in \{br, bl, tl, tr\}$ such that all the $B$-rules for $B \equiv A$ are $\mathfrak{d}$-linear.

$\{r, l, t, b\}$-*rank-linear grammars* are defined analogously.

A grammar is *rank-linear*, $\{br, bl, tl, tr\}$*-rank-linear*, or $\{r, l, t, b\}$-rank-linear iff there is a preorder $\leq$ such that it has the respective property wrt. $\leq$. A language is called *rank-linear* (or $\{br, bl, tl, tr\}$*-rank-linear* or $\{r, l, t, b\}$*-rank-linear*, resp.) if it is generated by some grammar with the respective property.

The notion on rank-linearity of context-free picture grammars compares to the linear word grammars as follows: Rank-linear grammars in which no $\ominus$-connective occurs may be viewed as word grammars; a word grammar is linear iff it is in this sense a rank-linear grammar *wrt. the universal relation on the set of nonterminals* as the preorder.

The example grammar from the beginning of this section is in CNF and $\{br, bl, tl, tr\}$-rank-linear wrt. a preorder with $S \equiv A$ and $B, C < A, S$. The $S$-rules and $A$-rules are $br$-linear wrt. this preorder, and the $B$-rules ($C$-rules) are $t$-linear ($l$-linear, resp.) wrt. this preorder.

Note the little incompatibility of notations that in our definition an $r$-linear rule wrt. the universal relation corresponds to what is known as a left-linear rule of a word grammar and vice versa.

Rank-linear word grammars can also generate non-linear word languages such as $\{a^i b^i \mid i \geq 1\}^+$. (Consider the grammar with the rules $S \to SA \mid A,\ A \to aAb \mid ab$ and a preorder such that $A < S$.)

The following theorem states that rank-linear grammars, $\{br, bl, tl, tr\}$-rank-linear grammars, and $\{r, l, t, b\}$-rank-linear grammars capture exactly the expressive power of $REG^{UOP}$, $REG^{ROP}$, and $REG$, resp.

**Theorem 15.** *For all languages $L \subseteq \Sigma^+_+$ we have*

- *$L$ is rank-linear iff $L \in \mathcal{REG}^{\mathcal{UOP}}$.*
- *$L$ is $\{br, bl, tl, tr\}$-rank-linear iff $L \in \mathcal{REG}^{\mathcal{ROP}}$.*
- *$L$ is $\{r, l, t, b\}$-rank-linear iff $L \in \mathcal{REG}$.*

The proof uses the concept of *generalized operator grammars (GOGs)* as an intermediate stage. These are, roughly speaking, grammars in which the right hand side of rules need not be sentential forms but may be arbitrary monotonic expressions from $REG^{UOP}$. In a derivation step of a GOG, one non-terminal $A$ is replaced by a sentential form $\alpha$ that is "below" a right hand side $\beta$ of an $A$-rule in the sense that $\alpha$ results from $\beta$ by replacing subexpressions in an outermost strategy, namely: Each union $\sigma \cup \varrho$ is replaced either with $\sigma$ or with $\varrho$, each iteration $\sigma^*$ is replaced with a finite number of compositions of $\sigma$, and so on.

The different notions of rank-linearity can be defined similarly for GOGs.

For a given monotonic regular expression $r$, there is the equivalent GOG with the only rule $(S, r)$. So the construction of a grammar amounts to reducing the complexity of the right hand sides inductively by introducing new rules until a grammar is reached, whose right hand sides contains only sentential forms.

Conversely, a given grammar with one of the rank-linearity properties is also a GOG with this property. Now the construction of an expression means to reduce the number of rules and to replace the recursion of non-terminals with the iteration of operators. Formally, the latter is done by induction over the maximal length of a strictly decreasing chain wrt. to the given preorder.

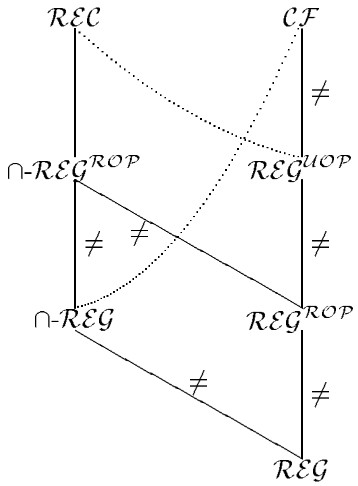
The term "generalized operator grammar" has been chosen because of the similarity to the "generalized transition graphs" used to pass from automata to regular word expressions.

## 5 Summary of (Non-)Inclusion Results

The following table shows the mentioned non-inclusion results that hold over alphabets with at least two letters, and a witness for each.

| | |
|---|---|
| $\cap$-$\mathcal{REG} \not\subseteq \mathcal{CF}$ | language of lemma 13 |
| $\mathcal{REG}^{\mathcal{ROP}} \not\subseteq \cap$-$\mathcal{REG}$ | set of squares |
| $\mathcal{REG}^{\mathcal{UOP}} \not\subseteq \mathcal{REC}$ | $\{a^i b^i \mid i \geq 1\}$ |

From the above facts, Theorem 7, and trivial inclusions like $\mathcal{REG} \subseteq \mathcal{REG}^{\mathcal{ROP}}$, one can infer the results presented in the figure for non-trivial alphabets. Here, the dotted lines indicate incomparability and the remaining lines show inclusions, which are marked by $\neq$ when known to be proper. It is open whether the inclusion $\mathcal{REG}^{\mathcal{ROP}} \subseteq \cap$-$\mathcal{REG}^{\mathcal{ROP}}$ is also proper in case of a one-letter alphabet, but it is known that the inclusion $\mathcal{REG}^{\mathcal{ROP}} \subseteq \mathcal{REC}$ is (see [Mat95]). Note that for a singleton alphabet both concatenations are commutative, from which one can deduce that any unrestricted operator can be transformed into an equivalent *br*-operator and therefore $\mathcal{REG}^{\mathcal{UOP}}(\{a\}) = \mathcal{REG}^{\mathcal{ROP}}(\{a\})$.




## 6 Conclusion

We have shown that besides the concept of tiling systems, which is an extension of recognizability to picture languages, there are interesting classes of regular expressions which allow to define a canonical analogon to regular word languages in the context of pictures. However, it seems to be impossible to define regular picture language in such a way that all characterization results and closure properties of the theory of words carry over to pictures.

Our characterization results may be viewed as a step towards a Kleene-like theorem for picture languages: In the same sense that a right-linear word grammar can be considered a non-deterministic finite automaton on words, a

$\{br, bl, tl, tr\}$-rank-linear grammar might be considered a kind of automaton on pictures (or rather on sentential forms). However, the translation into automata theoretic terms is not as immediate for $\{br, bl, tl, tr\}$-rank-linear grammars as for (right)-linear word grammars.

The following open problems are of particular interest:

- Do $\mathcal{CF}$ and $\mathcal{REG}^{\mathcal{ROP}}$ coincide in case of a one-letter alphabet? If so, this would be a nice correspondence to the situation of word languages.
- Do $\cap$-$\mathcal{REG}^{\mathcal{ROP}}$ and $\mathcal{REC}$ coincide? One candidate for a recognizable language not in $\cap$-$\mathcal{REG}^{\mathcal{ROP}}$ is the language of all pictures over $\{a\}$ for which $|P|$ is prime and $\underline{\overline{P}}$ is the length of an accepting run of a fixed LBA accepting the set of words of prime length.
- Is the emptiness problem for $REG^{ROP}$ decidable? Note that it is undecidable for $\cap$-$REG^{ROP}$, even for a fixed alphabet of size 1; see [Mat95].

Besides these questions dealing with the picture language definition formalisms of this paper, one could imagine to transfer other concepts from word languages to pictures in such a way that some of the known results of the theory of word languages remain true. For example: Is there a reasonable analogon to the monoid recognizability of word languages?

Moreover, future research may deal with the more general case of $n$-dimensional arrays for arbitrary integers $n$ instead of $n = 1$ or $n = 2$, e.g. words or pictures. It is easy and only a matter of notational inconvenience to redo all the definitions and proofs of this paper for arbitrary dimensions.

**Acknowledgments.** I thank Wolfgang Thomas for the help and advice he gave me during the preparation of both my diploma thesis and this paper.

## References

[GR92] D. Giammarresi and A. Restivo. Recognizable picture languages. In *Proceedings First International Colloqium on Parallel Image Processing 1991*, number 6 in International Journal Pattern Recognition and Artificial Intelligence, pages 241–256, 1992.

[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.

[GRST96] D. Giammarresi, A. Restivo, S. Seibert, and W. Thomas. Monadic second-order logic and recognizability by tiling systems. *Information and Computation*, 125:32–45, 1996.

[Mat95] Oliver Matz. *Klassifizierung von Bildsprachen mit rationalen Ausdrücken, Grammatiken und Logik-Formeln*. Diploma thesis, Christian-Albrechts-Universität Kiel, 1995. (German).

[Sir87] R. Siromoney. Advances in array languages. In Ehrig et al., editors, *Graph Grammars and Their Applications to Computer Science*, volume 291 of *Lecture Notes in Computer Science*, pages 549–563. Springer-Verlag, 1987.