Can. J. Math., Vol. XXXIX, No. 6, 1987, pp. 1413-1433

ESTIMATING THE SIZE OF CONTEXT-FREE TILING LANGUAGES

KENNETH HOLLADAY

Tilings and grammars. The problem of counting polyominoes motivates this paper. We will develop a general question for study that has counting polyominoes as a special case. We generalize in two ways. Polyominoes are shapes on the tiling made of square tiles. We will consider shapes on other tilings. The set of all polyominoes can be generated by a context-free array grammar, but the size of this set is estimated by counting the words of certain subsets and supersets that are generated by more convenient grammars. Our general question is the problem of counting the words of a context-free array language on a periodic tiling.

Counting polyominoes is a difficult problem that has not been completely solved yet. There are various techniques for roughly estimating the number of polyominoes of a given size. We will extend some of these techniques to our general question. In most cases, a given technique actually yields a sequence of increasingly accurate estimates at the cost of increasingly long calculations. Thus we could obtain better numbers by enlisting the aid of the computer. We will not do this here, but instead merely illustrate the techniques using straightforward examples.

A *tiling* is a locally finite cover of the plane by topological disks which we will call *tiles*. We require that the interiors of the tiles be disjoint and that the intersection of two or more tiles be empty or a connected subset of the boundary of each of them. A nonempty intersection of 3 or more tiles is necessarily a single point, which is called a *vertex*. A nonempty intersection of two tiles which consists of more than one point is called a *face* of each of the tiles and the two tiles are said to be *adjacent*. Two tiles intersecting in a single point (which will be a vertex) are not adjacent; instead, they are said to be *touching*. We will be interested in *polygonal tilings*, which means that all faces are line segments. Note that it is possible for two or more consecutive faces of a tile to be colinear. Any set of tiles of a tiling is associated with a graph, called its *face graph*. There is a node of the face graph for each tile of the set and arcs joining nodes whose corresponding tiles are adjacent. A set of tiles is said to be *connected* if and only if its face graph is connected. Two tilings are of the same *topological type* if and only if the face graph of the set of all tiles of

Received December 6, 1983 and in revised form April 14, 1986.

the one tiling is graph isomorphic to the face graph of the set of all tiles of the other.

Although the ideas of this paper can be easily extended to general periodic tilings, for simplicity, we will consider only the highly symmetrical isohedral tilings. The *symmetry group* of a tiling is the group of all Euclidean isometries that send tiles to tiles. We will usually regard the symmetries as being permutations on the set of tiles and the set of edges and the set of vertices of the tiling. A tiling is *periodic* if and only if its symmetry group contains two independent translations. A tiling is *isohedral* if and only if its symmetry group is transitive on the set of tiles. See [**4**] for a classification of isohedral polygonal tilings. For the remainder of this paper, tiling will mean isohedral polygonal tiling. There are 11 topological types of such tilings and we will use the notation in [**3**] for them.

Isotonic array grammars were defined in [**10**], see also [**11**]. The set of *symbols* of a grammar is a finite set given as three disjoint subsets: the *nonterminals*, N; the *terminals*, T; and a single symbol, the *blank*, $\#$. We write $V = N \cup T$. Given a fixed tiling, a *word* on these symbols is a map from the set of tiles to $V \cup \{\#\}$ such that the preimage of V is finite and connected. Two words are *congruent* if and only if they differ only by composition with a symmetry of the tiling. An *array rewriting rule* (or rule) consists of a *model*, M, and two maps α and β from M to $V \cup \{\#\}$. The model is a finite connected set of tiles. A rule $R = (M, \alpha, \beta)$ *rewrites* a word w_1 as a word w_2 if and only if the following holds. There is a translation, G, of the tiling's symmetry group such that for any tile, t, either $G(t) \notin M$ and $w_1(t) = w_2(t)$ or else $G(t) \in M$ and $w_1(t) = \alpha(G(t))$ and $w_2(t) = \beta(G(t))$. We will want rules to preserve the connectedness of the preimage of V. The conditions given below for a context-free rule will assure this. An *isotonic array grammar* (or grammar) is a triple

$$G = (V \cup \{\#\}, P, S)$$

composed of a set of symbols $V \cup \{\#\}$; a finite set, P, of rules for these symbols and a *start symbol* $S \in N$.

The array language generated by a grammar can now be defined. A word w_2 is *derivable by* G from a word w_1 if and only if either $w_1 = w_2$ or there is a finite sequence of words v_i (from $i = 0$ to $i = n$) such that $v_0 = w_1$, $v_n = w_2$ and for every i from 1 to n, there is a rule of G that rewrites v_{i-1} as v_i. We write $w_1 \rightarrow w_2(G)$. A word is terminal if and only if its range is $T \cup \{\#\}$ instead of all of $V \cup \{\#\}$. Given a start symbol S and a tile, t, the *start word at t* (written S_t) is the word with $S_t(t) = S$ and $S_t(t') = \#$ for all tiles $t' \neq t$. The *language generated by G starting from t* (written $L(G; t)$) is defined as

https://doi.org/10.4153/CJM-1987-066-4 Published online by Cambridge University Press

$$L(G;\, t) = \{w \text{ terminal word} \mid \ S_t \rightarrow w(G)\}.$$

If t' is translate of t then the words of $L(G;\, t')$ will be congruent to the words of $L(G;\, t)$. We will say that the languages are the same. If t' is not a translate of t, then $L(G;\, t')$ may or may not be the same as $L(G;\, t)$. We say that a word w is a *sentential form* of G starting from t if and only if $S_t \rightarrow w(G)$. If the set, T, of terminal symbols of a grammar has but one element, we say that the language generated by the grammar is a *language of shapes* since the terminal only delineates a shape from the background of blanks. In this paper, we are primarily concerned with languages of shapes. We will often slightly misuse the term language as defined above. Instead of referring to $L(G;\, t)$ as the language, we will speak the language as being the union of $L(G;\, t)$ over all tiles t. If we say that a grammar, G, generates such a language, we mean that the language is obtained by first forming $L(G;\, t)$ and then taking the union over all tiles t.

The fundamental example of a language of shapes is the set of all connected shapes on a tiling. There are two reasonable definitions for this set, depending on whether the shapes are required to be simply connected or not. Terminology is not standardized, but for shapes on the usual tiling by squares, the trend is to use the word animal for shapes that are simply connected and polyomino for shapes that need not be simply connected. Extending this to a general tiling, we will call the set of simply connected shapes the *language of animals* and the set of connected shapes the *language of polytiles*. The grammar for the polytiles is easier. There is one nonterminal, the start symbol, and, of course, one terminal. There is a group of rules for each translation orbit of tiles. For a given translation orbit of tiles there is one rule for each translation orbit of edges of the tiles. We pick one tile of the orbit and the model for a rule consists of the chosen tile and the tile on the other side of the chosen edge. In the α map, the chosen tile has the nonterminal and the other tile is blank. In the β map, both tiles have the nonterminal. There is also a rule for each translation orbit of tiles that rewrites the nonterminal as the terminal. This grammar for the tiling $P_5 - 21$ of type $[3^3, 4^2]$ is illustrated in figure 1.

The rules of the grammar for the animals of a tiling use blanks to assure that each addition of a tile preserves simple connectedness. There is one nonterminal, the start symbol, and one terminal. There is a group of rules for each translation orbit of tiles. As above, there is a rule for each edge of a sample tile from each orbit of tiles. Each rule tries to add the tile on the other side of the selected edge if it can do so while preserving simple connectedness. The rule's model will contain, in addition to the tile (call it x) with the nonterminal and the tile (call it y) to be added, several additional tiles that will get blanks in both the α and β maps. These are tiles that must not be part of the animal if the addition of y is to preserve simple connectedness. These tiles include all tiles adjacent to or touching y, with some exceptions. Of course x is not included. Also not included are any tiles that pass the following test. For any simply connected set of tiles containing the tile in question and x, the addition of y always leaves the set simply connected. The α map of the rule gives x the nonterminal and the other tiles blanks. The β map gives both x and y nonterminals and the rest blanks. In addition to the above rules, for each translation orbit of tiles there is a rule rewriting the nonterminal as the terminal. This grammar is illustrated in figure 2 using $P_5 - 21$.

The *size of a word* on some set of symbols is the number of tiles which are not blank. We define the *size of a language L*, written $|L|$, as follows. Let $L(n)$ be the number of incongruent words in L of size n. Then

$$|L| = \limsup_{n \to \infty} \sqrt[n]{L(n)}.$$

The size could also be defined as the reciprocal of the radius of convergence of the ordinary generating function of $L(n)$. Note that in the definition of size, we may use either $L(G; t)$ or the union over all tiles. Since $L(n)$ is the number of incongruent words and a periodic tiling has a

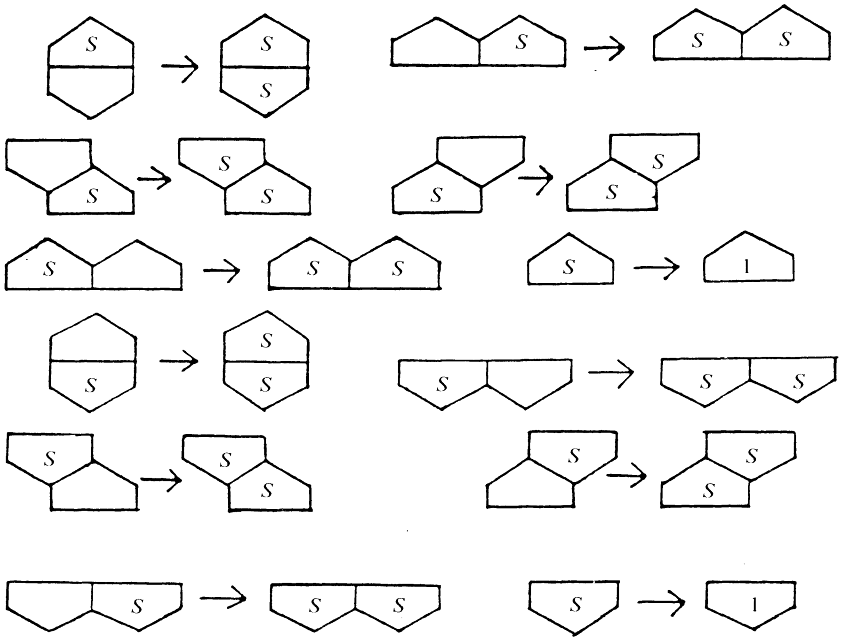

Figure 1. A weak context-free grammar for the polytiles on $P_5 - 21$ (of type $[3^3.4^2]$). The start symbol and only nonterminal is S. The terminal is 1. Cells with blanks are left blank.

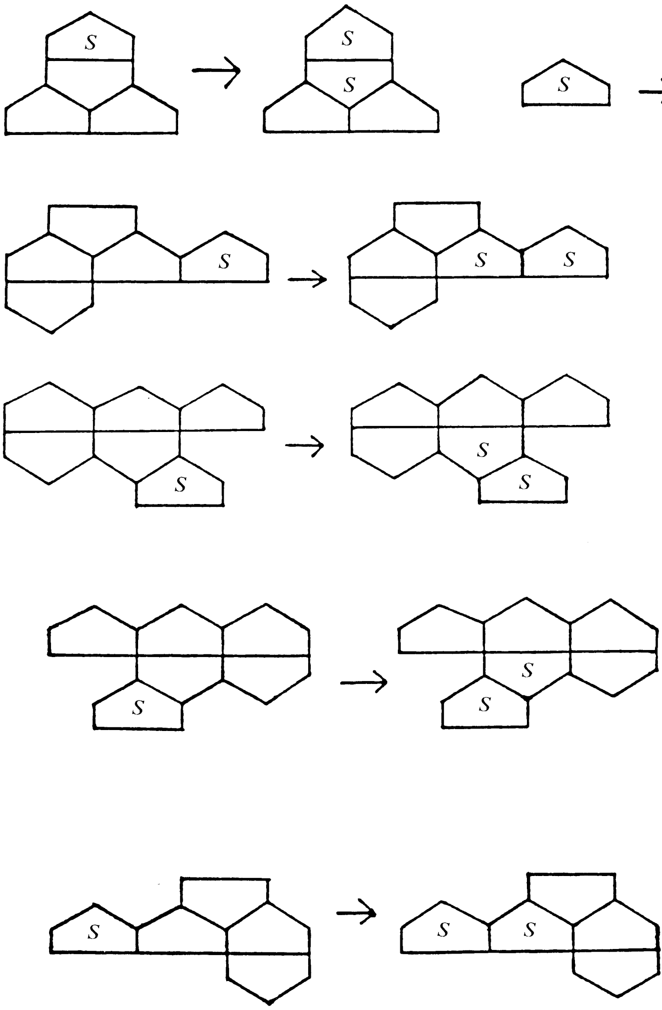

Figure 2. Half of a shape sensitive context-free grammar for the animals on $P_5 - 21$ (of type $[3^3.4^2]$). The other six rules are obtained by reflection in a horizontal axis. They apply to S in a pentagon of the other translation orbit. S is the start symbol and only nonterminal. 1 is the terminal. Tiles with blanks are left blank.

finite number of orbits of tiles, the size will be the same whether we use $L(G; t)$ or the union. The *Klarner constant* of a tiling T, written $K(T)$, is defined as the size of the language of polytiles on T. It is not difficult to see that two periodic tilings of the same topological type must have the same Klarner constant. An interesting question is whether the size of the language of animals on a tiling can be strictly less than the Klarner constant of the tiling. Note that a language with n terminals has size at most $nK(T)$. Thus size divided by terminal set size is a normalized measure that is independent of the number of terminals.

Context-free grammars. Isotonic array grammars are too general to really capture the notion of a purely locally defined shape, so we will now restrict our attention to the array analogs of context-free string grammars. The first such analog was defined in [**2**] and two others were defined in [**7**]. All array analogs restrict the α map of any rule to have a nonterminal value at one tile and blanks at all of the other tiles in the model. If this nonterminal value is the symbol A, we refer to the rule as an *A-rule*. The three analogs defined below differ in the extent to which blanks may be used as context. The freest use of blanks occurs with a *shape sensitive context-free grammar* (SSCFG). Each rule of a SSCFG must have $\beta^{-1}(V)$ connected and containing the tile with the nonterminal α value. This allows the use of a tile t where $\alpha(t) = \beta(t) = \#$. An A-rule might not be applicable to a tile with an A if the tile was not accompanied by the appropriate blanks. The grammar for the animals of a tiling is a SSCFG and it uses blanks as context to maintain simple connectedness. The language generated by a SSCFG is called a *shape sensitive context-free language* (SSCFL).

The next version of context-free is more restrictive. A rule may test a tile to see if it is blank, but it must then write a nonblank symbol into the tile. In a *weak context-free grammar* (WCFG) every rule has a connected model and all β values in V. Thus we can still use blanks as context, since an A-rule does not apply to an instance of A that is lacking the appropriate surrounding blanks. The grammar of the polytiles of a tiling is a WCFG. The polytiles of a tiling are a WCFL but the animals are not. The method of Proposition 2 of [**7**] can be used to show that no WCFG can generate exactly the animals of a tiling. Since every WCFG is also a SSCFG, the class of SSCFL's properly contains the class of WCFL's.

The last and most restrictive form of context-free array grammar does not use blanks as context at all. In these grammars, we want the mere presence of a nonterminal in a word to already guarantee the presence of any blanks needed to apply any rule. We do require that the tiles to be rewritten by a rule applied to another tile's nonterminal indeed be blank. A nonblank cell must not be rewritten by a rule applied to another tile's nonterminal, even if the rule would correctly reproduce the tile's previous value. To obtain a formally isotonic rule, we introduce a "don't care" symbol; to enforce the above restrictions, we require that a grammar of this type, a *strong context-free grammar* (SCFG), be validated before it is said to generate a *strong context-free language* (SCFL). Let

$\ast \notin (V \cup \{\#\})$ be a new "don't care" symbol that matches any symbol of $V \cup \{\#\}$. Write $V\ast = V \cup \{\ast\}$. For each rule of a SCFG, we require that the α map take a nonterminal value at one tile and the value $\ast$ at all remaining tiles of the model. The model must be connected and the β map must have all values in V. Since the $\ast$'s match anything, an A-rule can be applied to any word with a tile containing A that is a translate of the model's tile that takes α value A. Given an A-rule, a word w with a tile t such that $w(t) = A$ and a translation σ sending t to the tile of the rule's model M with α value A; we say that the application of this rule to this instance of A *does not conflict with* w if and only if

$$w(\sigma^{-1}(\{S \in M| \quad S \neq t\})) = \{\#\}.$$

A SCFG G is *valid* if and only if for any sentential form w of G and any nonterminal A of w, there is no A-rule of G whose application conflicts with w. A SCFL is the language generated by a valid SCFG. Thus, to show that a language is a SCFL, we must give a SCFG that generates it and a proof of the validity of this SCFG. The methods of Proposition 1 of [**7**] can be used to show that no valid SCFG can generate exactly the polytiles of a tiling. However, if the $\ast$'s of a SCFG are simply replaced by blanks, we will obtain a WCFG that must generate exactly the same language as the SCFG did. Thus the class of WCFL's properly contains the class of SCFL's.

At this point we prove a technical lemma that will be helpful in the next section. A *single production* is a rule whose β map takes a nonterminal value for the tile where the α map is a nonterminal and takes the value blank for any other tiles of the rule's model.

LEMMA 1. *A grammar of any of the three context-free types can be modified so as to eliminate all single productions while still generating the same language and remaining the same context-free type.*

Proof. For WCFG's and SCFG's, a single production must have a one tile model. Thus we can eliminate single productions from these types by the same method as in the context-free string grammar case. See, for example, Section 4.3 of [**6**]. For SSCFG's, a single production may have more than one tile in the model. The above method must keep track of the accumulated tiles of blanks used as context. Suppose a sequence of single productions produces the nonterminal A from the nonterminal B. In the above mentioned method, after removing all single productions from the grammar, we add some new rules back. For each A-rule we add a corresponding B-rule with the B replacing the A. We modify this procedure by enlarging (if necessary) the new B-rule's model by adding any tiles used as context in the derivation of A from B that do not have something written in them by the A-rule. Since there may be several ways of obtaining A from B that have different configurations of added tiles used as context, this means we may have to add several new B-rules for each A-rule.

As the last item in this section, we want to define what it means for a context-free grammar to be ambiguous. To do this we must define an analog of rightmost derivation. We will call them *priority derivations*, and they are defined as follows. For each rule of the grammar an arbitrary ordering is assigned to those tiles of the model given nonterminals by the β map of the rule. During a derivation, it is possible that more than one rule will be applied to the same nonterminal in a given tile. Therefore we must also assign an arbitrary order to the set of rules itself. This done, we may define the priority order for applying rules by using a stack or last-in-first-out storage device. Each time a rule is applied, we push onto the stack, in the order prescribed by the rule, tokens representing each tile given a nonterminal by the rule. The next rule to be applied will be applied to the nonterminal in the tile whose token is popped off the top of the stack. If more than one rule is to be applied to this nonterminal, we pick the highest priority rule. A priority derivation is one that follows these rules at every step. A context-free grammar is *ambiguous* if and only if there is a word of the language it generates that has at least two different priority derivations. Notice that whether or not a grammar is ambiguous is independent of the various arbitrary choices made in defining a priority derivation.

Linear grammars. A *linear* grammar is a context-free grammar in which every rule's β map gives at most one tile a nonterminal value. In this section we will give a method for counting the words of a given size in any linear unambiguous strong context-free grammar. The method gives upper bounds for the size of any linear context-free grammar. We actually count derivations rather than words. Thus an ambiguous grammar will be overcounted since different derivations can give the same final result. In a weak or shape sensitive context-free grammar, there may be an overcount because some derivations lead to conflicts and thus produce no result. The rules of a linear grammar produce a system of recursive equations relating the numbers of words of various sizes. We must then solve a homogeneous constant coefficients system of linear recursions. First we demonstrate how such systems can always be solved.

Let Seq be the set of sequences of real numbers. If addition and scalar multiplication are defined componentwise, Seq is a real vector space. A linear operator is defined on Seq by $(Ef)(n) = f(n + 1)$ for $n = 1, 2, 3, \ldots$. Let Seq(m) be defined as the real vector space of column vectors of

m elements of Seq. We define a system of linear recursions in the variables $x_j \in$ Seq for $j = 1$ to m as follows. The degree of the system in the variable x_j will be $d(j)$. For $i = 1$ to m, let $P_{ij}(E)$ be a polynomial in E with degree less than $d(j)$. The equation for x_j is

$$E^{d(j)}x_j = \sum_{i=1}^{m} P_{ij}(E)x_i.$$

This system of equations in the x_j can be put in matrix form by defining two matrices whose entries are polynomials in E. Let the ijth entry of the matrix P be P_{ij}. D is a diagonal $m \times m$ matrix with ith diagonal entry $E^{d(i)}$. Let $X \in \text{Seq}(m)$ have the x_j as its entries. The matrix form of the system is

(1) $DX = PX.$

The set of X satisfying (1) will be a $d = \sum d(i)$ dimensional subspace of Seq(m), since any solution is completely determined by the d values of $x_j(n)$ for $n = 1$ to $d(j)$ and $j = 1$ to m.

THEOREM 2. *There is a fixed recursion of degree d that is satisfied by every component x_j of every solution of* (1).

Proof. There is a matrix C (the transpose of the cofactor of $D - P$) whose entries are polynomials in E and that satisfies the equation

$$C(D - P) = (\det(D - P))I_m$$

where I_m is the $m \times m$ identity matrix [**1**]. Multiplying a solution vector X of (1) by both sides of this equation yields

$$0 = C(D - P)X = (\det(D - P))I_mX = (\det(D - P))X.$$

Thus $\det(D - P)$, a polynomial in E, annihilates every component of X. $\text{Det}(D - P)$ is not identically zero and is easily seen to be monic of degree d. We will use a corollary of this theorem. Let Q_j be a polynomial in E for $j = 1$ to m.

COROLLARY 3. *The sequence*

$$y = \sum_{j=1}^{m} Q_j(E)x_j$$

satisfies the same recursion as all of the x_i do.

Note that such a y could also satisfy other recursions of degree lower than d.

Counting derivations in a linear grammar produces a system like (1). Initially we assume a SCFG with no single productions. We define an element of Seq for each nonterminal of the grammar. If A is a nonterminal, the associated sequence $A(n)$ gives the number of sentential forms of size n containing A as the unique nonterminal. Each rule of the grammar with a nonterminal in the β map will contribute a term to the recursion for that nonterminal's sequence. For example, if an A-rule writes terminals into k tiles and the nonterminal B into one tile, the recursion for $B(n)$ will contain a term $A(n - k)$. If a nonterminal other than the start symbol never appears as a value of a β map of any rule, then that nonterminal will never appear in any derivation of the language and it may be dropped from the grammar. If the start symbol never appears as a value of a β map, then the start symbol's sequence is just $S(1) = 1$ and $S(n) = 0$ for $n > 1$. Otherwise all the sequences appear in nontrivial equations. Since there are no single productions, the equations will be proper recursions.

By Theorem 2, there will be a fixed recursion, which we will write as $P(E)A = 0$, satisfied by every nonterminal's sequence. $P(E)$ is a polynomial in E of degree d. The number of words of size n, which we write as $L(n)$, in the language generated by the grammar can be expressed in terms of the various nonterminal's sequences. Each rule with β map all terminals and blanks contributes a term to the formula for $L(n)$. If an A-rule writes terminals into k tiles, there is a term $A(n - k)$ in the formula for $L(n)$. Applying Corollary 3 to the sequence $L(n)$, we find that it too satisfies the recursion $P(E)L = 0$. Now, by the standard theory of solutions of a linear homogeneous constant coefficients recursion (see [**5**]), we can give the largest real root of $P(x) = 0$ as an upper bound for the size of the language. As noted after Corollary 3, it is possible that $L(n)$ might also satisfy a recursion of degree less than d. In this case the minimal recursion for $L(n)$ will be a divisor of $P(E)$ and the size of the language may be a smaller root of $P(x) = 0$. For WCFG's and SSCFG's this count of derivations overcounts the actual words; so again, the size of the language might be smaller than the largest root of $P(x) = 0$. But in any case we do obtain an upper bound.

As an application of this method, we consider a linear language of shapes that can be defined on any tiling. The *Hamiltonian language* of a tiling consists of all polytiles whose face graph contains a Hamiltonian path. A WCFG for this language can be easily obtained by modifying the grammar for polytiles given above. In a typical rule with a two tile model, we change the β map. The tile that had the nonterminal in the α map now gets a terminal in the new rule's β map. If each tile has S edges then this grammar gives the upper bound S for the size of the Hamiltonian language. The upper bound can be easily reduced to $S - 1$. The grammar to show this is a refinement of the grammar just given. There is a subset of nonterminals for each translation orbit of tiles. There is one nonterminal in the subset for each edge of a fixed tile in the given translation orbit. The edge indicated by the nonterminal is interpreted as the edge shared with a tile that joined the word at an earlier step in the derivation. There is a rule rewriting this nonterminal for each edge of the tile except the indicated edge. The rule's model has two tiles, the fixed tile and the tile on the other side of the rule's selected edge. In the α map, the fixed tile has its nonterminal and the other tile has a blank. In the β map, the fixed tile has a terminal and the other tile has the appropriate nonterminal. There are rules rewriting any nonterminal as the terminal. The start rules just write a suitable nonterminal into a blank tile.

Better upper bounds for the size of the Hamiltonian language of a tiling are obtained by two methods for extending the ideas used in the grammar of the previous paragraph. In one of these methods, each rule adds several new tiles instead of just one. This method increases the number of rules. In the other method, each nonterminal records several preceding tiles instead of just one. This method increases the number of nonterminals and implies an increase in the number of rules. This is because if the nonterminals record the preceding K tiles, we will want the rules to add at least K new tiles. Table 1 gives the upper bounds for some tilings that are obtained by

TABLE 1

Some upper bounds for the size of the Hamiltonian language of several isohedral tilings. The second column gives the number of previous tiles recorded by a nonterminal. The third column gives the number of tiles added by each rule. The fifth column gives the polynomial resulting from the recursion produced by the grammar. In the two cases marked by *, this polynomial factors over the integers.

1 Tiling type	2 Number recorded	3 Number added	4 Upper bound	5 Polynomial yielding upper bound	
$[3.12^2]$	2	2	1.76929	$x^6 - 4x^4 + 4x^2 - 4$	*
$[4.8^2]$	2	2	1.92595	$x^6 - 4x^4 + 4$	
$[6^3]$	3	3	1.96104	$x^6 - 7x^3 - 4$	
$[3.6.3.6]$	2	2	2.73205	$x^4 - 8x^2 + 4$	*
$[4^4]$	1	3	2.92402	$x^3 - 25$	
$[4^4]$	1	4	2.90278	$x^4 - 71$	
$[4^4]$	2	2	2.89118	$x^4 - 8x^2 - 3$	
$[3^6]$	1	2	4.79583	$x^2 - 23$	
$[3^6]$	1	3	4.68755	$x^3 - 103$	
$[3^6]$	2	2	4.63361	$x^6 - 22x^4 + 11x^2 + 8$	

TABLE 2

Some lower bounds for the size of the Hamiltonian language of several isohedral tilings. The third column gives the polynomial resulting from the recursion produced by the grammar. In the case marked by *, this polynomial factors over the integers.

1 Tiling type	2 Lower bound	3 Polynomial yielding lower bound
$[3.12^2]$	1.56302	$x^6 - 3x^2 - 4x - 1$
$[4.8^2]$	1.74067	$x^8 - 4x^5 - 2x^4 - 2$
$[6^3]$	1.73205	$x^2 - 3$
$[3.6.3.6]$	2.26953	$x^3 - x^2 - 2x - 2$
$[4^4]$	2.41421	$x^2 - 2x - 1$
$[3^6]$	3.30278	$x^4 - 3x^3 - 3x - 1$ *

various combinations of these methods. We see that increasing the number of tiles added does not yield as much improvement as increasing the memory of nonterminals. However, increasing the number of nonterminals greatly complicates finding the recursion. The number of variables can be reduced and the recursion simplified by combining nonterminal variables into more symmetric variables. As an example of this, we will consider the usual tiling by squares $P_4 - 56$. Let the nonterminals record the two preceding tiles and the rules add two new tiles. To find a recursion, let $A(n)$ be the sum of the variables for the 4 nonterminals where the last tile and the two preceding tiles form a straight row. Let $B(n)$ be the sum of the variables for the 8 nonterminals where the last tile and the two preceding tiles form an L shape. If $T(n) = A(n) + B(n)$ is the total, then

$$A(n + 2) = 3A(n) + 3B(n) = 3T(n),$$
$$B(n + 2) = 6A(n) + 5B(n) = 5T(n) + A(n),$$
$$T(n + 4) = 8T(n + 2) + 3T(n).$$

We can obtain lower bounds for the size of the Hamiltonian language of a tiling by constructing subsets that are unambiguous SCFL's. Since we can make an exact count for this kind of subset, we get a lower bound for the size of the whole language. To get a valid SCFG, words can not be allowed to grow freely in all directions. The grammar in figure 2 of [7] is an example. In that grammar words can grow left, right or up but not down. The languages generated by such grammars are reasonably large and give nontrivial lower bounds. Table 2 lists some lower bounds obtained in this fashion.

The lower bounds of Table 2 are also lower bounds for the Klarner constants of the respective tilings. Other linear languages larger than the

Hamiltonian languages give even bigger lower bounds. But it seems to be difficult to produce a really large (relative to the Klarner constant) linear language with a manageable set of rules. We now turn to the nonlinear boardpile languages, which have manageable grammars and large sizes.

Boardpile languages. A *stratification* of a tiling is a partition of the tiles into an infinite number of subsets called *layers*. Each layer is connected as a set of tiles. There is a translation symmetry of the tiling that simultaneously fixes all of the layers as sets. The symmetry group of the tiling is transitive on the set of layers. It is easy to see that any periodic tiling, isohedral or not, has many stratifications. We are mostly interested in *thin* stratifications, for which no layer contains as a proper subset a layer of another stratification. Given a stratification, we will visualize the tiling as oriented so that the translation fixing the layers is a horizontal translation. Then the layers will be stacked vertically. Two layers are *adjacent* if and only if a tile of one is adjacent to a tile of the other.

LEMMA 4. *Every layer is adjacent to exactly two other layers.*

Proof. By taking the union of its tiles, each layer can be considered as a set of points in the plane. We want to look at the boundaries of these sets. These boundaries are polygonal curves. None of these boundaries can contain a bounded component since layers are connected and a bounded set can not be fixed by a translation. The translation fixing the layers must send the components of the boundary of a layer into themselves. If we look at a boundary point and its translate, it must be possible to join the two points by a path in the layer on one side of the boundary and by a path in the layer on the other side. Thus the boundary also connects the two points. Each component of the boundary of a layer is the intersection of exactly two layers (which are adjacent). If not, there would be a point of the boundary that was contained in at least two other layers. By applying the translation that fixes all the layers, we obtain another point contained in the boundaries of the same three layers. There would be paths joining these two points that are internal to each of the three layers except at the endpoints. Since they do not cross except at the endpoints, one of the paths would be between the other two. The layers are supposed to be face connected, but the part of the layer containing the middle path would be either bounded or else cut off from the rest of the layer. Thus the boundaries are intersections of pairs of layers and they are all fixed by a certain translation. Since the layers are connected, it is clear they must each consist of the tiles in the strip between two consecutive boundaries.

The set of layers has a transitive symmetry group. How many translation orbits of boundaries can there be? Consider the symmetry sending a layer into the layer just above it. Two cases occur. Either the bottom boundary of the lower layer goes to the bottom of the upper layer and the top to the top. Or else the boundary between the layers is sent to itself and the bottom of the lower layer goes to the top of the upper. In the first case all boundaries will be translates of each other. If the second case occurs but the first does not, then there are two translation orbits of boundaries that are interlaced, each layer having one boundary from each orbit. If the first case occurs, we say the stratification has *symmetric* layers. If the first case does not occur (and so the second case applies), we say the stratification has *asymmetric* layers.

We can now define the boardpiles of a stratification. A *board* is a connected set of tiles contained in a single layer. A *boardpile* is a polytile whose intersection with every layer of the stratification is either empty or a board. The *boardpile language* for a stratification is the set of all boardpiles of the stratification. Note that boardpile languages are never linear. We can construct T shaped boardpiles with each of the three arms being arbitrarily long. However, boardpile languages can be given by valid unambiguous SCFG's. If the usual tiling by squares is given the obvious stratification into rows, we obtain the boardpile language studied by Klarner in **[8]**. Figure 3 gives a valid unambiguous SCFG for this case. To give a grammar of this type for any stratification on any periodic tiling, we would need a valid unambiguous linear SCFG for boards of the stratification. We will not give a general construction for such a grammar here, but note that it is easily accomplished for any specific instance that arises.

Boardpile languages are notable for having large sizes compared to other strong context-free languages. In **[8]**, Klarner showed that the boardpile language on squares mentioned above has size about 3.205569. He also showed that the obvious stratification of the regular hexagon tiling into straight rows gives a boardpile language with size about 3.863131. Klarner's formula can not be directly applied to the expressions counting boardpiles in most stratifications. In a few cases we can get upper or lower bounds. For example, the pentagonal tiling $P_5 - 21$ has a stratification with asymmetric layers where one boundary is like the boundary for squares and the other boundary is like that for hexagons (see figure 4). We conclude that the resulting boardpile language has size between the two numbers mentioned above. To obtain exact sizes of boardpile languages requires evaluating more complicated expressions than that of equation (7) of **[8]**. In that equation, each composition contributed one product to the sum. For most boardpile languages, each

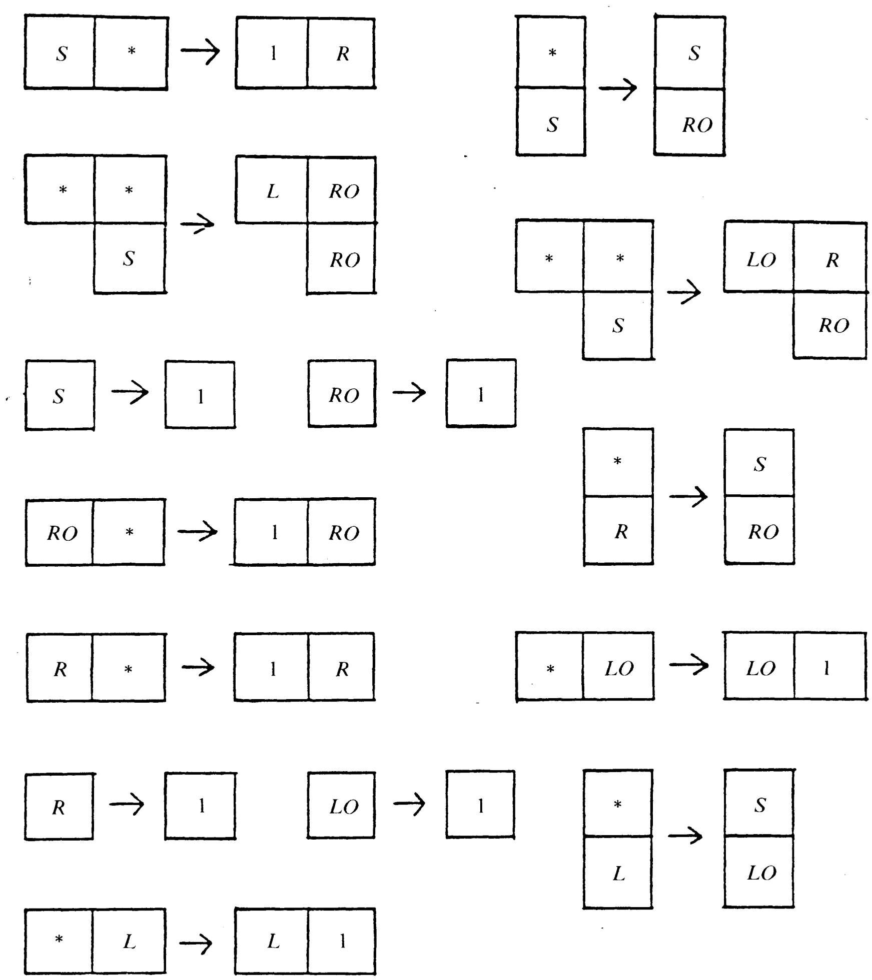

Figure 3. A strong context-free unambiguous grammar for a boardpile language on $P_4 - 56$ (Type $[4^4]$). S is the start symbol. $N = \{S, RO, R, LO, L\}$, $T = \{1\}$ and * is the "don't care" symbol.

composition will contribute several products to the sum. And for stratifications with asymmetric layers, instead of one function $f(m, n)$, there will be two such functions that alternate in each product.

Upper bounds. In this final section we will discuss a method for obtaining upper bounds on the Klarner constant of a tiling. This method, given in **[9]**, relies on a technique for estimating the radius of convergence

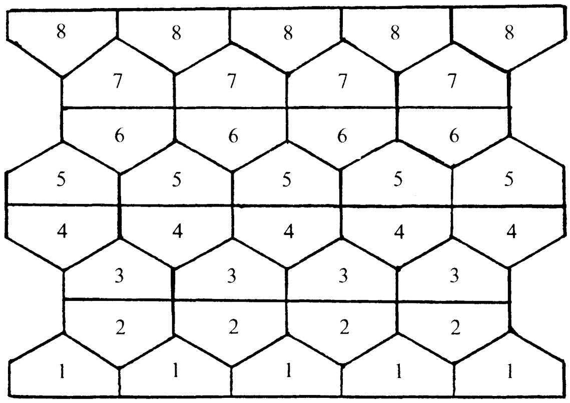

Figure 4. A thin stratification for $P_5 - 21$ (Type $[3^3.4^2]$). The numbers 1 to 8 in the tiles indicate layer number. Note that this is an asymmetric stratification since the two components of a layer's boundary are not the same.

of the diagonal of a double power series representing a rational function. In the simplest version of the method, we have a context-free grammar for the polytiles of a tiling that has only one nonterminal. Each rule, R, is given a weight, $w(R)$, that is a monomial with numerical coefficient 1 in the variables x and y. The power of x is the number of tiles that are blank in the α map of the rule and not blank in the β map. The power of y is the number of tiles given the terminal by the β map. Given a priority derivation of the grammar, we form a product with a factor of its weight for each time a rule is used in the derivation and also one additional factor of x to account for the original start symbol. A derivation producing a terminal word must have product $x^n y^n$, where n is the size of the word. Consider the double power series

$$(2) \quad x\left[1 - \sum_i w(R_i)\right]^{-1} = \sum_{k=0}^{\infty} x\left[\sum_i w(R_i)\right]^k$$

where the sums in i are over the rules of the grammar. Each contribution to the coefficient of $x^n y^n$ in this series comes from a sequence of rules that is a possible priority derivation of a terminal word of size n. Hence the coefficient is an upper bound on the number of such words and the reciprocal of the radius of convergence of the diagonal series gives an upper bound on the size of the language generated by the grammar. Since some sequences of rules give derivations with conflicts, we obtain in general only an upper bound and not an exact value. If needed, the technique of Section 3 of **[9]** is used to estimate the radius of convergence of the diagonal.

Our first upper bound is a general one. Suppose that each tile of our isohedral tiling has exactly S edges.

PROPOSITION 5. *The Klarner constant of the tiling is less than or equal to*

$$\frac{(S - 1)^{S-1}}{(S - 2)^{S-2}}.$$

Proof. To use the version of the method sketched above, we must extend our notion of symbol. We would need the symbol to record orientation information about any tile it was written in. In our case, the single nonterminal of the grammar would distinguish one edge of any tile it was written in. The distinguished edge would indicate the adjacent tile from which the nonterminal was first written into the tile. Given a tile with the nonterminal in it, there are rules of the grammar that write nonterminals into any subset of adjacent tiles that does not contain the distinguished adjacent tile. The tile itself is given a terminal. A rule writing nonterminals into k adjacent tiles would have weight $x^k y$.

This extended notion of symbol does not cause any real difficulties but we can avoid it by slightly elaborating our method. We have a nonterminal for each edge of one tile from each translation orbit of tiles. Again the nonterminal's edge distinguishes the adjacent tile that is not to be added by any rule. We use essentially the same rules for each nonterminal. A rule adds a subset of adjacent tiles at positions defined relative to the distinguished edge. A given rule determines which adjacent tiles will have the appropriate nonterminals written in them by counting edges proceeding clockwise from the distinguished edge. Then, instead of a priority derivation giving a sequence of exact rules, it would just give a sequence of relative rules.

For both approaches, the sum of weights in (2) is $y(1 + x)^{S-1}$. The coefficient of $x^n y^n$ is the binomial coefficient

$$\binom{(S - 1)n}{n - 1}$$

and the proposition follows by an easy calculation.

The upper bound of Proposition 5 also applies to tilings that are not isohedral if we define S as the maximum number of edges of any tile. In the proof, if a derivation calls for a rule to be applied to a tile with an insufficient number of edges, then that derivation produces no result. If the tiling has one orbit of tiles with many more edges than the other orbits of tiles, then the estimate above will be rather poor. In general, this estimate seems to get worse with increasing S. The upper bound of 4 for the Klarner constant of a triangulation seems to be the best case.

The upper bound of Proposition 5 can be reduced by a technique analogous to forming the sets of twigs $E(k)$ in [**9**]. For a fixed individual tiling, it is more efficient to find a better first grammar before beginning the construction of the higher order $E(k)$. For our first example, we treat the case of the regular hexagon tiling. So that the neighbors of a tile will be in specific directions, we regard the tiling as being oriented so that each hexagon has a pair of horizontal edges. In order to be able to make any polytile, a grammar must be able to write into any neighbor of a tile with a nonterminal. In the grammar of Proposition 5, this was assured since one neighbor had already been written in and there were rules that could write into any of the other five neighbors. However, we can get to all the neighbors without needing rules to directly write into all of them. We now give a grammar that can directly write into only three neighbors. We will describe the rules for a hexagon that was first written in from the hexagon directly below it; the other cases are rotations of this case and we are using relative rules again as in the proof of Proposition 5. As in that proof, we use either a single orientable nonterminal that can distinguish the neighbor it was written in from or else a whole set of non-orientable nonterminals that convey this information. There are rules writing the appropriate nonterminal into any subset of the three adjacent hexagons that are to the upper left, directly above and to the upper right of the given hexagon. We show that this set of rules is sufficient by induction on the length of the derivation from the start tile to our given tile. In this grammar, the start tile will always be a bottommost tile of the polytile generated. All polytiles have bottommost tiles so we have not lost any. For a start tile we use the nonterminal that distinguishes the tile directly below. Thus, since start tile was a bottommost tile, the three neighbors of the start tile that are not covered by the rules will be blanks in the final polytile. Proceeding to our given hexagon as the induction step; we note that, by induction, all neighbors of the hexagon below it can be written in. This includes all three lower neighbors of the given hexagon since a derivation would have reached them before reaching our given hexagon. Since the rules allow us to write into the remaining three upper neighbors, we are done. Computing the sum of weights for this grammar, we obtain the same sum as the set of twigs in figure 3 of [**9**]. Therefore, 6.75 is an upper bound for the Klarner constant of the regular hexagon tiling. Proposition 5 gives a bound of about 12.2 for this tiling.

For the remaining examples, the rules will treat small groups of tiles instead of single tiles as above. We use the following terminology. The tiles of a tiling will be partitioned into sets called *aggregate tiles*. If the original tile boundaries in the interior of the aggregate tiles are ignored, the aggregate tiles themselves form a tiling. In our examples this aggregate tiling will be one of the three regular tilings. The first example of this kind is the triangulation of type $[3.12^2]$. Here each aggregate tile is formed by three triangles whose 120° angles meet at the center of the aggregate tile. Each rule determines what symbols are to be written in all of the tiles of an aggregate tile. There is a nonterminal for each of the six translation orbits of original tiles. Each of these nonterminals has its version of a set of nine relative rules. A given rule will terminalize the tile with the α map nonterminal and possibly one or both of the other tiles in the aggregate tile. The rule may also write an appropriate nonterminal into the neighboring tile of either or both of the adjacent aggregate tiles that are not adjacent to the tile with the α map nonterminal. We need not write into the aggregate tile adjacent to the tile with α map nonterminal since, except at the start, a rule applied to this aggregate tile is what originally wrote the α map nonterminal. To complete the grammar, we need a start nonterminal and six rules to rewrite it as an appropriate nonterminal of the type defined above. We also need six additional start rules that rewrite the start symbol as an appropriate nonterminal and also write a terminal into the adjacent tile that is in a different aggregate tile. Without these additional start rules, we could not generate those polytiles whose boundary contained no segments that were aggregate tile boundaries. (There would be no place to start.) Since the start rules are only applied once, at the beginning, they may be ignored in the counting procedure and thus they do not effect the size estimate. This grammar gives as an upper bound for the size, the real root of $x^3 - 4x - 4$, about 2.382976.

The triangulation $P_3 - 10$ of type $[4.8^2]$ allows a similar construction. The aggregate tiles are the squares formed by the four triangles having a common vertex for their 90° angles. This time there are 25 relative rules for each of four nonterminals and eight start rules for the start symbol. We get an upper bound of about 3.187013 for the Klarner constant of the tiling.

The next example is the quadrilateral tiling of type [3.6.3.6]. The aggregate tiles are regular hexagons formed by groups of three quadrilaterals. There are three translation orbits of quadrilaterals but each quadrilateral could be entered from either of two faces, so we need six nonterminals. We will describe the set of fifteen relative rules for a quadrilateral with a horizontal base, leaning to the right, and entered from the tile directly below. This quadrilateral's aggregate tile is adjacent to six other aggregate tiles. The lower three are assumed already written in. In addition to terminalizing some quadrilaterals of the α map nonterminal's aggregate tile, the rules can write appropriate nonterminals into some of the adjacent quadrilaterals of the three upper neighboring aggregate tiles. Note that the quadrilateral above the α map nonterminal's tile is adjacent to two of these neighboring aggregate tiles but the quadrilateral to the left of the α map nonterminal's tile is adjacent to only one of these neighboring aggregate tiles. This grammar does not need additional start rules, six are sufficient. The resulting upper bound for the Klarner constant of the tiling is about 3.3357041.

In our last three examples, an aggregate tile is adjacent to two original tiles in each of the neighboring aggregate tiles. A rule that terminalizes tiles of a given aggregate tile needs to write in at most one original tile of any neighboring aggregate tile. One original tile with a nonterminal is sufficient for the entire aggregate tile. This allows us to omit rules that write into two different tiles of the same neighboring aggregate tile. One of these examples is the tiling $P_4 - 41$ of type [3.4.6.4]. An aggregate tile is an equilateral triangle composed of three quadrilaterals whose 120° angles share a vertex. There are six translation orbits of quadrilaterals and each quadrilateral can be entered from either of two faces. Each of the twelve nonterminals has a version of the set of twenty one relative rules. This grammar gives an upper bound of about 4.546892 for Klarner constant of the tiling. Another example is the equilateral triangle tiling itself. We group sets of four triangles into larger equilateral triangles. The aggregate tiling is of the same type as the original tiling. There are two translation orbits of tiles and a triangle could be entered from either of two faces. Note that the center triangle of an aggregate tile never gets a nonterminal. The sets of relative rules have twenty rules each. This grammar gives an upper bound of about 3.097276 for the Klarner constant of the tiling. Our last example is the triangulation of type [4.6.12]. An aggregate tile is an equilateral triangle composed of six triangles whose 60° angles meet at a common vertex. There is one nonterminal for each of the twelve translation orbits of original triangles. Each set of relative rules has 69 rules. The resulting upper bound for the Klarner constant is about 2.965690.

Acknowledgment. The methods discussed in this paper were directly inspired by the work of David A. Klarner on counting polyominoes. I would also like to thank my typist, Elaine, and my wife Wendy.

References

1. N. Bourbaki, *Algebra I* (Addison-Wesley, Reading, 1973), 532.

2. C. R. Cook and P. S. P. Wang, *A Chomsky hierarchy of isotonic array grammars and languages*, Computer Graphics and Image Processing, Vol. 8, (1978), 144-152.

3. B. Grünbaum and G. C. Shephard, *The eight-one types of isohedral tilings in the plane*, Math. Proc. Camb. Phil. Soc. *82* (1977), 177-196.

4. ——— *Isohedral tilings of the plane by polygons*, Comment. Math. Helvetici *53* (1978), 542-571.

5. M. Hall, *Combinatorial theory* (Blaisdell, Waltham, 1967), 22.

6. M. A. Harrison, *Introduction to formal language theory* (Addison-Wesley, Reading, 1978), 101.

7. K. W. Holladay, *On some classes of context-free array grammars*, Proc. 13th S.E. Conf. on Comb., Graph Theory and Computing, Boca Raton, Feb. 15-19, 1982, *Congressus numerantium 36* (1982), 325-332.

8. D. A. Klarner, *Cell growth problems*, Can. J. Math. *19* (1967), 851-863.

9. D. A. Klarner and R. L. Rivest, *A procedure for improving the upper bound for the number of n-ominoes*, Can. J. Math. *25* (1973), 585-602.

10. A. Rosenfeld, *Isotonic grammars, parallel grammars and picture grammars*, in *Machine intelligence 6* (Univ. of Edinburgh Press, Edinburgh, 1971), 281-294.

11. ——— *Picture languages* (Academic Press, New York, 1979).

University of New Orleans,
New Orleans, Louisiana