An Optimal Algorithm for Tiling the Plane with a Translated Polyomino

Andrew Winslow *

Abstract

We give a $O(n)$-time algorithm for determining whether translations of a polyomino with n edges can tile the plane. The algorithm is also a $O(n)$-time algorithm for enumerating all such tilings that are also regular, and we prove that at most $\Theta(n)$ such tilings exist.

1 Introduction

Motivated by applications in parallel computing, Shapiro [9] asked whether it could be decided if translations of a given polyomino could tile the plane. Beauquier and Nivat [1] proved that the problem was not only decidable, but solvable in polynomial time by testing a simple criterion called the *BN criterion*. Informally, a tile satisfies the BN criterion if it can be surrounded by instances of itself (see Figure 1). Such a surrounding corresponds to a *regular* or *isohedral* tiling where all tiles share an identical neighborhood.

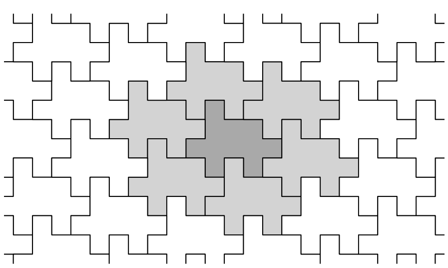

Figure 1: A polyomino tile (dark gray), a surrounding of the tile (gray), and the induced regular tiling (white).

Using a naive algorithm, the BN criterion can be applied to a polyomino with n edges in $O(n^4)$ time. Gambini and Vuillon [4] gave an improved $O(n^2)$-time algorithm and around the same time, Brlek, Provençal, and Fédou [2] achieved $O(n)$-time algorithms for two special cases: (1) the boundary contains no consecutive repeated sections larger than $O(\sqrt{n})$, and (2) testing a restricted version of the BN criterion (surroundable by just four instances). Provençal [8] further improved on the algorithm of Gambini and Vuillon for the general case, obtaining $O(n \log^3(n))$ running time, and in a recent survey, Blondin Massé, Brlek, and Labbé [7] conjecture that a $O(n)$-time algorithm exists. In this work, we confirm their conjecture by giving such an algorithm (Theorem 5).

Our algorithm also doubles as an algorithm for enumerating all surroundings (regular tilings) of the polyomino. As part of the proof of the running time of the algorithm, we prove a claim of Provençal [8] that the number of surroundings of a tile with itself is $O(n)$ (Theorem 3), complementing other tight bounds by Blondin Massé et al. [6] on a special class of surroundings.

2 Definitions

A *letter* is a symbol $x \in \{\mathbf{u}, \mathbf{d}, \mathbf{l}, \mathbf{r}\}$. The *complement* of a letter x, written $\bar{x}$, is defined by the bijection $\bar{\mathbf{u}} = \mathbf{d}$, $\bar{\mathbf{r}} = \mathbf{l}$, $\bar{\mathbf{d}} = \mathbf{u}$, and $\bar{\mathbf{l}} = \mathbf{r}$. A *word* W is a sequence of letters, and the ith letter of W is denoted $W[i]$. A *boundary word* is a word corresponding to the sequence of edge directions encountered during a clockwise traversal of a polyomino's boundary, e.g. the polyomino in Figure 1 has (circular) boundary word $\mathbf{uru^2rdr^2dr(dl)^2uldlul}$.

For a word W, $\widehat{W}$ denotes the sequence of complemented letters of W traversed in reverse order. A *factor of* W is a contiguous sequence X of letters in W. A factor X is *admissible* provided $W = XU\widehat{X}V$ with $|U| = |V|$, non-complementary first and last letters of U, and similarly of V. A *factorization* of W is a partition of W into consecutive (possibly length-0) factors F_1 through F_k, written $W = F_1 F_2 \dots F_k$.

3 BN Factorizations

Definition 1 *A factorization of a boundary word W is a* BN factorization *provided it is of the form $W = ABC\widehat{A}\widehat{B}\widehat{C}$.*

Lemma 1 *Let P be a polyomino with boundary word W. There exists a regular tiling of P with clockwise neighbor-shared boundary word factors $F_1, F_2, \dots, F_k$ if and only if $F_1 F_2 \dots F_k$ is a BN factorization.*

Observe that this mapping between between tilings and factorizations is a bijection.

Lemma 2 *A boundary word W has $O(|W|)$ BN factorizations.*

*Université Libre de Bruxelles, 1050 Bruxelles, Belgium, andrew.winslow@ulb.ac.be

Theorem 3 *A polyomino with n sides has $O(n)$ regular tilings.*

Provençal [8] observed that polyominoes with $\Omega(n)$ tilings exist, e.g. $\mathbf{ur}^i\mathbf{dl}^i$ with $i \geq 1$ has i regular tilings.

4 An Algorithm for Enumerating Factorizations

Lemma 4 *Let W be a polyomino boundary word. The BN factorizations of W can be enumerated in $O(|W|)$ time.*

Proof. Corollary 5 of [2] states that all factors of a BN factorization are admissible. The algorithm first computes all admissible factors, then enumerates factorizations consisting of them.

Computing admissible factors. Corollary 5 of [2] implies that there are at most $2|W|$ admissible factors, since admissible factor has a distinct center. For each center $W[i]$ or $W[i]W[i+1]$, the admissible factor with this center is LR, where R is the longest common prefix of WW starting at $WW[i+1]$ and $\widehat{W}\widehat{W}$ starting at $\widehat{W}\widehat{W}[|W|/2-(i+1)]$. Use a suffix-tree-based approach (see Theorem 9.1.1 of [5]) to preprocess these words in $O(|W|)$ time so that the longest common prefixes can be computed in $O(1)$ time each and $O(|W|)$ total time. The word L is defined and computed similarly.

Enumerating factorizations. Let $W = AY\widehat{A}Z$ with A an admissible factor and $|Y| = |Z|$. Let $B_1, B_2, \ldots, B_l$ be the admissible prefix factors of Y, with $|B_1| < |B_2| < \cdots < |B_l|$. Similarly, let $C_1, \ldots, C_m$ be the suffix factors with $|C_1| < \cdots < |C_m|$. A variation of Lemma C4 of [3] (omitted due to space) implies that for fixed A, there exist intervals $[b, l]$, $[c, m]$ such that the BN factorizations $AB_iC_j\widehat{A}\widehat{B_i}\widehat{C_j}$ are exactly those with $i \in [b, l]$ or $j \in [c, m]$.

First, construct a list of all admissible factors starting at each $W[k]$, sorted by length in $O(|W|)$ time using counting sort. Repeat for factors ending at each $W[k]$.

Next, use a two-finger scan to find, for each factor A that ends at $W[k]$, the longest factor B_l starting at $W[k+1]$ such that $|A| + |B_l| \leq |W|/2$. Then check whether C_j, the factor following B_l such that $|AB_lC_j| = |W|/2$, is admissible and report the factorization $AB_lC_j\widehat{A}\widehat{B_l}\widehat{C_j}$ if so. Checking whether C_j is admissible takes $O(1)$ time using an array mapping each center to the unique admissible factor with this center.

Enumerated additional BN factorizations containing A by checking factors B_i with $i = l-1, l-2, \ldots$ for an admissible following factor C_j. If C_j is admissible, report the factorization, otherwise stop the iteration, since $i = b-1$.

Finally, use a similar two-finger scan to find, for each factor A that starts at $W[k]$, the longest factor C_m that ends at $W[k+|W|/2-1]$ such that $|A|+|C_m| \leq |W|/2$, check whether B_i preceeding C_m such that $|AB_iC_m| = |W|/2$ is admissible, and report the possible BN factorization. Then check and report similar factorizations with C_j for $j = m-1, m-2, \ldots$ until $j = c-1$.

In total, the two-finger scans take $O(|W|)$ time plus $O(1)$ time to report each factorization. Each factorization is reported once per choice of A. Remove duplicate factorizations with a canonical factor labeling (A contains $W[0]$) and radix sorting the six-tuples of the first letter indices of the factors. Then by Lemma 2, reporting factorizations also takes $O(|W|)$ time. ☐

Theorem 5 *Let P be a polyomino with n sides. The regular tilings of P can be enumerated in $O(n)$ time.*

Acknowledgments

The author would like to thank Stefan Langerman for fruitful discussions that greatly improved the paper.

References

[1] D. Beauquier and M. Nivat. On translating one polyomino to tile the plane. *Discrete & Computational Geometry*, 6:575–592, 1991.

[2] S. Brlek, X. Provençal, and J.-M. Fédou. On the tiling by translation problem. *Discrete Applied Mathematics*, 157:464–475, 2009.

[3] Z. Galil and J. Seiferas. A linear-time on-line recognition algorithm for "Palstar". *Journal of the ACM*, 25(1):102–111, 1978.

[4] L. Gambini and L. Vuillon. An algorithm for deciding if a polyomino tiles the plane by translations. *RAIRO - Theoretical Informatics and Applications*, 41(2):147–155, 2007.

[5] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.

[6] A. B. Massé, S. Brlek, A. Garon, and S. Labbé. Every polyomino yields at most two square tilings. In *7th International Conference on Lattice Paths and Applications*, pages 57–61, 2010.

[7] A. B. Masseé, S. Brlek, and S. Labbé. Combinatorial aspects of Escher tilings. In *22nd International Conference on Formal Power Series and Algebraic Combinatorics*, pages 533–544, 2010.

[8] X. Provençal. *Combinatoire des mots, géométrie discrète et pavages.* PhD thesis, Université du Québec à Montréal, 2008.

[9] H. D. Shapiro. Theoretical limitations on the efficient use of parallel memories. *IEEE Transactions on Computers*, 27(5):421–428, 1978.