Non saturated polyhexes and polyiamonds

Alexandre Blondin Massé
Laboratoire de combinatoire et d'informatique mathématique
Université du Québec à Montréal
blondin_masse.alexandre@uqam.ca

Julien de Carufel
Université du Québec à Trois-Rivières
Trois-Rivières, Canada
Julien.de.carufel@uqtr.ca

Alain Goupil
Université du Québec à Trois-Rivières
Trois-Rivières, Canada
Alain.Goupil@uqtr.ca

Abstract

Induced subtrees of a graph G are induced subgraphs of G that are trees. Fully leafed induced subtrees of G have the maximal number of leaves among all induced subtrees of G. In this extended abstract we are investigating the enumeration of a particular class of fully leafed induced subtrees that we call non saturated. After an overview of this recent subject, we proceed to the enumeration of fixed non saturated polyhexes and polyiamonds.

1 Introduction

We recently presented a new family of combinatorial and graph theoretic structures called *fully leafed induced subtrees* of a simple graph G of size n which are induced subtrees of a graph G with a maximal number of leaves [BCGS17]. Fully leafed induced subtrees are realized as polyforms in two-dimensional regular lattices and polycubes in the three dimensional cubic lattice so that tree-like polyforms of size n are *fully leafed* when they contain the maximum number of leaves among all tree-like polyforms of size n. Recursive and exact expressions were given in [BCGS17, BCGLNV18] for the number of leaves in a number of particular cases which include graphs and polyforms.

We also showed in [BCGLNV18] that the problem of deciding if there exists an induced subtree with i vertices and ℓ leaves in a simple graph G of size n is NP-complete in general. We define the map $L_G : \mathbb{N} \rightarrow \mathbb{N}$ by the condition that $L_G(n)$ is the maximal number of leaves among all induced subtrees of G of size n. We call this map the *leaf function* of G. We have computed the values of the map L_G for some classical graphs and we have described in [BCGLNV18] a branch-and-bound algorithm that computes the function L_G for any simple graph G. In [BCGLNV18], we consider the problem of deciding wether a given sequence $(\ell_0, \ell_1, \ldots, \ell_n)$ of natural numbers is the leaf sequence $(L_G(n))_n$ of some graph G. We call this problem the *leaf realization problem*. In the particular case where G is a caterpillar, a bijection [BCGLNV18] was exhibited between the set of discrete derivatives of the leaf sequences $(L_G(i))_{3\leq i\leq |G|}$ and the set of prefix normal words introduced in [FL11] and investigated in [BFLRS14, BFLRS17].

In: L. Ferrari, M. Vamvakari (eds.): Proceedings of the GASCom 2018 Workshop, Athens, Greece, 18–20 June 2018, published at http://ceur-ws.org

We discuss the concept of saturated and non saturated polyforms and polycubes [BCGS17] in Sections 4 and 5 where we exhibit three bijections. The first bijection relates the set $\mathcal{T}_{\text{squ}}(k)$ of tree-like polyominoes of size k and the set $\mathcal{ST}_{\text{squ}}(4k+1)$ of saturated tree-like polyominoes of size $4k+1$. The second bijection maps the set $\mathcal{ST}_{\text{hex}}(n)$ of saturated polyhexes of size n to the set $\mathcal{ST}_{\text{tri}}(n)$ of saturated polyiamonds of size n. The third bijection is between the set $\mathcal{ST}_{\text{cub}}(41k+28)$ of saturated tree-like polycubes of size $41k+28$ and the set $4\mathcal{T}_i(3k+2)$ of 4-trees that are polycubes of size $3k+2$ recently introduced [BCG18].

In this extended abstract, we are interested with the enumeration of fixed fully leafed tree-like polyhexes and polyiamonds. These families of polyforms are induced subtrees of infinite regular lattices λ and their leaf function is denoted by $L_\lambda(n)$.

It was already shown in [BCGS17] that saturated tree-like polyhexes and polyiamonds are easy to enumerate. They both are caterpillars with a linear shape and there is in fact a bijective correspondance between these two sets. The description of non saturated tree-like polyhexes and polyiamonds is more intricate and we now focus on their enumeration.

2 Polyforms, Polycubes and Graphs

Let $G=(V,E)$ be a simple graph, $u \in V$ and $U \subseteq V$. For any subset $U \subseteq V$, the *subgraph induced by* U is the graph $G[U]=(U, E \cap \mathcal{P}_2(U))$, where $\mathcal{P}_2(U)$ is the set of 2-elements subsets of V. The *square lattice* is the infinite simple graph $\mathcal{G}_2=(\mathbb{Z}^2, A_4)$, where A_4 is the 4-*adjacency relation* defined by $A_4=\{(p,p') \in \mathbb{Z}^2 \mid \text{dist}(p,p')=1\}$ and dist is the Euclidean distance of $\mathbb{R}^2$. For any $p \in \mathbb{Z}^2$, the set $c(p)=\{p' \in \mathbb{R}^2 \mid \text{dist}_\infty(p,p') \leq 1/2\}$, where dist_∞ is the uniform distance of $\mathbb{R}^2$, is called the *square cell* centered in p. The function c is naturally extended to subsets of $\mathbb{Z}^2$ and subgraphs of $\mathcal{G}_2$. For any finite subset $U \subseteq \mathbb{Z}^2$, we say that $\mathcal{G}_2[U]$ is a *grounded polyomino* if it is connected. The set of all grounded polyominoes is denoted by $\mathcal{GP}$. Given two grounded polyominoes $P=\mathcal{G}_2[U]$ and $P'=\mathcal{G}_2[U']$, we write $P \equiv_t P'$ (resp. $P \equiv_i P'$) if there exists a translation $T: \mathbb{Z}^2 \to \mathbb{Z}^2$ (resp. an isometry I on $\mathbb{Z}^2$) such that $U'=T(U)$ (resp. $U'=I(U)$). A *fixed polyomino* (resp. *free polyomino*) is then an element of $\mathcal{GP}/\equiv_t$ (resp. $\mathcal{GP}/\equiv_i$). Clearly, any connected induced subgraph of $\mathcal{G}_2$ corresponds to exactly one connected set of square cells via the function c. Consequently, from now on, polyominoes will be considered as simple graphs rather than sets of edge-connected square cells.

All definitions of *cell*, *grounded polyomino*, *fixed polyomino* and *free polyomino* in the above paragraph are extended to the *hexagonal lattice* with the 6-*adjacency relation*, the *triangular lattice* with the 3-*adjacency relation* and the *cubic lattice* with the 6-*adjacency relation*. Grounded polyforms and polycubes are thus connected subgraphs of regular grids and the terminology of graph theory becomes available. Let $T=(V,E)$ be any finite simple non empty tree. A vertex $u \in V$ is a *leaf* of T when $\deg_T(u)=1$ and is an *inner vertex* otherwise. For any $d \in \mathbb{N}$, the number of vertices of degree d is denoted by $n_d(T)$ and $n(T)=|V|$ is the number of vertices of T which is also called the *size* of T.

A (grounded, fixed or free) *tree-like polyform* (resp. polycube) is therefore a (grounded, fixed or free) polyform (resp. polycube) whose associated graph is a tree. We now introduce rooted grounded tree-like polyforms and polycubes.

Definition 2.1. A *rooted grounded tree-like polyform or polycube* in a lattice λ is a triple $R=(T,r,\vec{u})$ such that

(i) $T=(V,E)$ is a grounded tree-like polyform or polycube of size at least 2;
(ii) $r \in V$, called the *root* of R, is a cell of T;
(iii) $\vec{u} \in \lambda$, called the *direction* of R, is a unit vector such that $r+\vec{u}$ is a cell of T adjacent to r.

When $r+\vec{u}$ is a leaf of T, we say that R is non-final. Otherwise R is called final.

If $R=(T,r,\vec{u})$ is a rooted, grounded, non-final tree-like polyform or polycube, a unit vector $\vec{v} \in \lambda$ is called a *free direction* of R whenever $r-\vec{v}$ is a leaf of T. We now introduce the operation called the *graft union* of tree-like polyforms and polycubes.

Definition 2.2 (Graft union)**.** Let $R=(T,r,\vec{u})$ and $R'=(T',r',\vec{u'})$ be rooted grounded non-final tree-like polyforms or polycubes in the lattice λ such that $\vec{u'}$ is a free direction of R. The *graft union* of R and R', whenever it exists, is the rooted grounded tree-like polyform or polycube

$$R \vartriangleleft R'=(\mathbb{Z}_3[V \cup \tau(V')], r, \vec{u}),$$

where V, V' are the respective sets of vertices of T, T' and τ is the translation with respect to the vector $\overrightarrow{r'r}-\vec{u'}$.

The graft union is naturally extended to fixed and free tree-like polyforms and polycubes.

3 Fully leafed polyforms and polycubes

The leaf function $L_\lambda(n)$, giving the maximal number of leaves in an induced subtree of size n, ihas been established for all planar regular grids λ i.e. for polyominoes, polyhexes and polyiamonds and also for polycubes.

Theorem 3.1 ([BCGS17])**.** *Let L_{squ}, L_{hex} and L_{tri} denote respectively the leaf functions of polyominoes, polyhexes and polyiamonds. Then we have*

$$L_{\text{squ}}(n) = \begin{cases} 2, & \text{if } n = 2; \\ n-1, & \text{if } n = 3,4,5; \\ \ell_{\text{squ}}(n-4)+2, & \text{if } n \geq 6. \end{cases} \tag{1}$$

$$\begin{aligned} L_{\text{hex}}(n) = L_{\text{tri}}(n) &= \begin{cases} 2, & \text{if } n = 2,3; \\ \ell_{\text{hex}}(n-2)+1, & \text{if } n \geq 4. \end{cases} \\ &= \left\lfloor \frac{n}{2} \right\rfloor + 1 \end{aligned} \tag{2}$$

and the asymptotic growth of L_λ is given by $L_\lambda(n) \sim n/2$ for the three families λ of tree-l9ke polyforms.

The proof that these expressions are exact is based on (i) the construction of families of polyforms that satisfy Equations (1) and (2); (ii) the elimination of all possible branches that would belong to a tree-like polyform of minimal size with more leafs than $L_\lambda(n)$.

In the case of three dimensional polycubes, the leaf function is more intricate.

Theorem 3.2 ([BCGS17])**.** *Let L_{cub} be the leaf-function on the cubic lattice. Then*

$$L_{\text{cub}}(n) = \begin{cases} f_{\text{cub}}(n)+1, & \text{if } n = 6,7,13,19,25; \\ f_{\text{cub}}(n), & \text{if } 2 \leq n \leq 40 \text{ and } n \neq 6,7,13,19,25; \\ f_{\text{cub}}(n-41)+28, & \text{if } 41 \leq n \leq 81; \\ \ell_{\text{cub}}(n-41)+28, & \text{if } n \geq 82. \end{cases} \tag{3}$$

$$\text{where} \qquad f_{\text{cub}}(n) = \begin{cases} \lfloor (2n+2)/3 \rfloor, & \text{if } 0 \leq n \leq 11; \\ \lfloor (2n+3)/3 \rfloor, & \text{if } 12 \leq n \leq 27; \\ \lfloor (2n+4)/3 \rfloor, & \text{if } 28 \leq n \leq 40. \end{cases}$$

The proof of this fact uses the same argument than in the two dimensional case but the set of possible branches in a tree-like polycube of size n that would have more leaves than $L_{\text{cub}}(n)$ is larger and it must be established with a computer program that exhausts all possibilities. The asymptotic growth of L_{cub} is still linear but it satisfies the surprising ratio $L_{\text{cub}}(n) \sim 28n/41$.

4 Saturated polyforms and polycubes

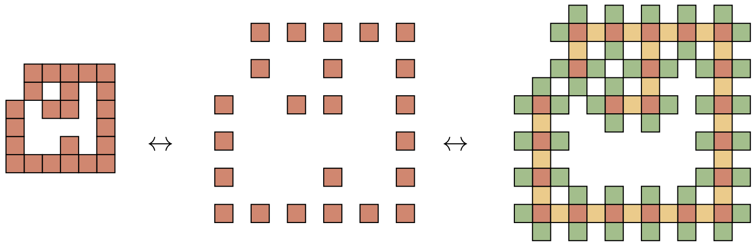
Let L_λ denote any of the four leaf functions described in (1), (2) and (3). Since $L_\lambda(n)$ satisfies a linear recurrence, it is immediate that there exists two parallel linear functions $\overline{L_\lambda}$, $\underline{L_\lambda}$ and a positive integer n_0 such that

$$\underline{L_\lambda}(n) \leq L_\lambda(n) \leq \overline{L_\lambda}(n), \quad \text{for } n \geq n_0,$$

if we add the constraint that there must exist infinitely many positive integers $n > 0$ for which $L_\lambda(n) = \overline{L_\lambda}(n)$ and $L_\lambda(n) = \underline{L_\lambda}(n)$, then the functions $\overline{L_\lambda}(n)$ and $\underline{L_\lambda}(n)$ become unique. *Saturated* tree-like polyforms and polycubes are defined as those tree-like polyforms and polycubes T for which $n_1(T) \geq \overline{L}(n(T))$.

Sets of saturated tree-like polyforms and polycubes possess structural properties that allow their bijective reduction to simpler polyforms and polycubes. These bijections are, to our actual knowledge, lattice dependent and are useful in the enumeration of saturated tree-like polyforms. We describe these bijections in the following paragraphs.

The upper bounding linear function of saturated polyominoes is $\overline{L}_{\text{squ}}(n) = (n+3)/2$. For integers $k \geq 1$, saturated tree-like polyominoes T have size $n(T) = 4k+1$ and $n_1(T) = 2k+2$ leaves. It is not difficult to show that saturated tree-like polyominoes are the iterated graft union of copies of a unique tile of size 5 made of one cell of degree 4 and four leaves that we call a cross because of its shape (see Figure 2).



(a) T (b) $I(T)$ (c) $\phi_{\text{squ}}(T)$

Figure 1: The bijection ϕ_{squ} for tree-like polyominoes

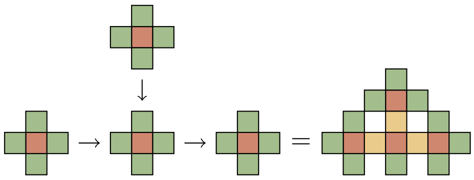


Figure 2: Saturated tree-like polyomino as graft union of crosses

Theorem 4.1 (Cross operator, [BCGS17])**.** *There exists a bijection ϕ_{squ} from the set $\mathcal{T}_{\text{squ}}(k)$ of tree-like polyominoes of size k and the set $\mathcal{ST}_{\text{squ}}(4k+1)$ of saturated tree-like polyominoes of size $4k+1$:*

$$\mathcal{T}_{\text{squ}}(k) \xrightarrow{\phi_{\text{squ}}} \mathcal{ST}_{\text{squ}}(4k+1).$$

The bijection ϕ_{squ} is illustrated in Figure 1 and it informs us that the complexity of counting saturated tree-like polyominoes of size $4k+1$ is identical to the complexity of the enumeration of tree-like polyominoes of size k.

Theorem 4.2 (Geometric shape of saturated polyhexes and polyiamonds, [BCGS17])**.** *Each saturated polyhex (resp. polyiamond) is the successive graft union of crosses in the hexagonal (resp. triangular) lattice.*

Proof. This result is immediate from the facts that graft union preserves degree distribution, that saturated polyhexes and polyiamonds have cells of degree 3 and 1 and that a cross is the only elementary polyform which contains a cell of degree 3. See Figures 3 and 4 for an illustration. □

Proposition 4.3 ([BCG18])**.** *There exists a bijection from, respectively, free and fixed saturated tree-like polyiamonds to free and fixed saturated polyhexes.*

Proof. (sketch). The correspondance is established by simply truncating the triangles to form hexagons, as shown in Figure 5. □

5 Non saturated polyhexes and polyiamonds

The description of nonsaturated polyforms is more intricate than the saturated case because there is more freedom in the choice of the position of "extra cells". At the moment, we are only able to enumerate polyhexes and polyiamonds. We leave the enumeration of fully leafed non saturated tree-like polyominoes and polycubes as open problems.

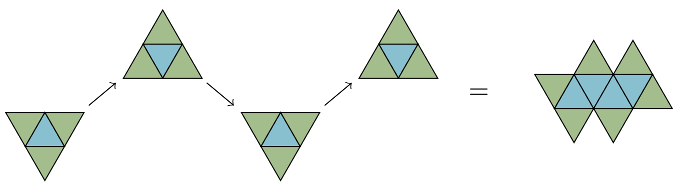


Figure 3: A saturated tree-like polyiamond as graft union of crosses

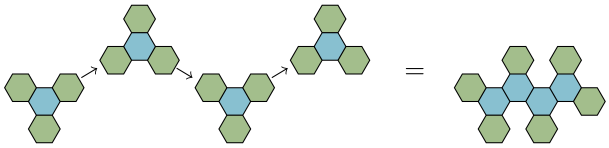


Figure 4: Saturated tree-like polyhexes as graft union of crosses

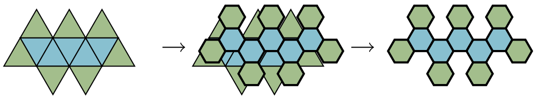


Figure 5: Bijection from satured polyiamonds to saturated polyhexes.

k	0	1	2	3	4	5	6	7	8	$k \geq 9$
$flhex_t(2k+3)$	9	18	45	102	180	246	327	426	516	$36 + 78(k-2)$

Proposition 5.1. *For $k \geq 0$, the number $flhex_t(2k+3)$ of fixed, fully leafed polyhex trees of odd size $2k+3$ is given by the following expressions*

Proof. The proof is the result of a case study where special structures appear until the size $n(T) = 19$ is reached. We know already that fully leafed polyhexes of odd size $n = 2k+3$ contain k cells of degree three and one cell of degree two, denoted x. These polyhexes are not saturated. As shown in Figure 6, the cell x partitions the k cells of degree three of a fixed fully leafed polyhex T in two disjoint connected sets T_1, T_2 of respective sizes $j \geq 0$ and $k - j \geq 0$. Both sets have one cell adjacent to x and both sets are, with one exception, forming a path. We denote by $C_t(j)$ the set of all paths of cells of degree 3 of length j. Furthermore we denote by $C_t(j)C_t(k-j)$ the set of fixed fully leafed polyhexes T which are the concatenation $T = T_1 x T_2$ with $T_1 \in C_t(j)$, $T_2 \in C_t(k-j)$ and we denote by $c_t(j)c_t(k-j)$ the cardinality of $C_t(j)C_t(k-j)$. Clearly we have

$$flhex_t(2k+3) = \sum_{j=0}^{\lfloor k/2 \rfloor} c_t(j)c_t(k-j)$$

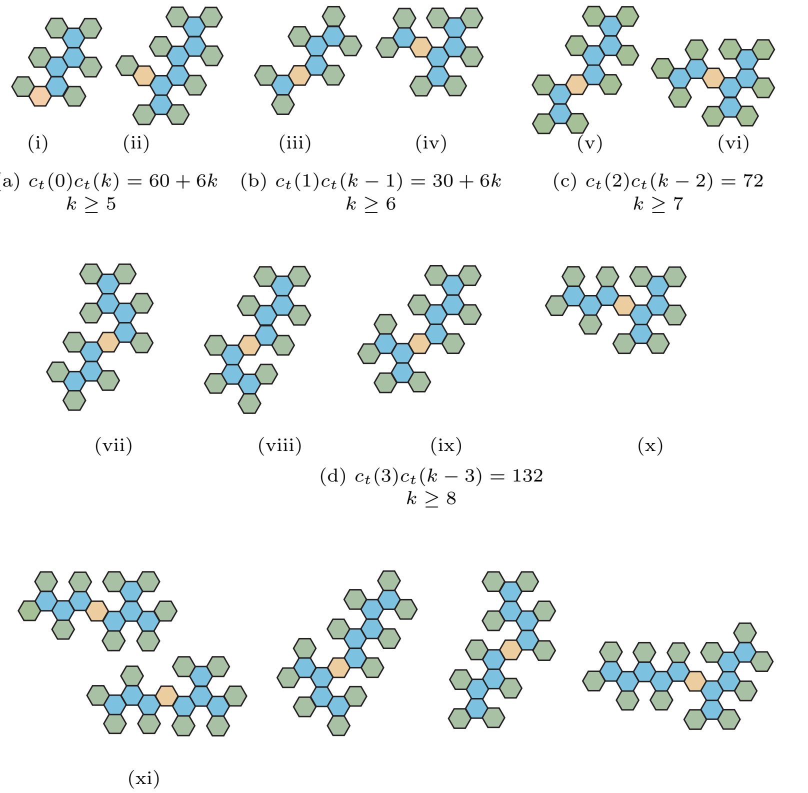
In order to obtain $flhex_t(2k+3)$, we evaluate each term $c_t(j)c_t(k-j)$ and provide a general expression when it exists.

First, we look at the case $C_t(0)C_t(k)$ shown in Figure 6(a). For all $k \geq 5$, there are 60 fixed polyhexes T where x is adjacent to a leaf of $T_2 \in C_t(k)$ (Figure 6 (a)(i)) and $6k$ polyhexes where x is adjacent to an inner cell of $T_2 \in C_t(k)$ (Figure 6 (a)(ii)). These cover all cases in the set $C_t(0)C_t(k)$ and we have $c_t(0)c_t(k) = 60 + 6k$ for $k \geq 5$.

In the case $C_t(1)C_t(k-1)$, there are 48 fixed polyhexes where a leaf of $T_2 \in C_t(k-1)$ is adjacent to x (Figure 6(b)(iii)) and $6(k-3)$ polyhexes where an inner cell of $T_2 \in C_t(k-1)$ is adjacent to x and $k \geq 6$ (Figure 6(b)(iv)). In the case $C_t(2)C_t(k-2)$, $k \geq 7$, there are 72 fixed polyhexes where either a leaf of $T_2 \in C_t(k-2)$ or a cell $y \in T_2$ next to a leaf is adjacent to x (Figure 6(c)(v) and (vi)). In the set $C_t(3)C_t(k-3)$, $k \geq 8$, a case study shows that there are $c_t(3)c_t(k-3) = 132$ polyhexes (Figure 6(d)(vii-x)). When $4 = j < k - j$, there are 180 fixed polyhexes (Figure 6(e) and (d)(xi)). This count includes a particular polyhex $T_1 \in C_t(4)$, shown in Figure 6(d)(xi), that is not a path. When $5 \leq j \leq k - j$, special configurations disappear and there are 66 polyhexes where $5 \leq j = k - j$ and 132 polyhexes with $5 \leq j < k - j$. Summing all the previous cases for $k \geq 9$, we obtain

$$\begin{aligned}\sum_{j=0}^{\lfloor (k-1)/2 \rfloor} c_t(j)c_t(k-j) &= [60 + 6k] + [48 + 6(k-3)] + [72] + [132] \\ &\quad + [180] + [132] + \left[\sum_{j=5}^{\lfloor (k-1)/2 \rfloor} 132\right] + 66\chi(k \text{ is even}) \\ &= 36 + 78(k-2), \quad k \geq 9.\end{aligned}$$

(*b*) For $k < 9$, the values $c_t(j)c_t(k-j)$ are different from the general cases described above for a number of reasons that forbid their inclusion in a general setting. There is no space here for this analysis. □



(a) $c_t(0)c_t(k) = 60 + 6k$, $k \geq 5$ (b) $c_t(1)c_t(k-1) = 30 + 6k$, $k \geq 6$ (c) $c_t(2)c_t(k-2) = 72$, $k \geq 7$

(d) $c_t(3)c_t(k-3) = 132$, $k \geq 8$

(e) $c_t(4)c_t(k-4) = 180$, $k \geq 9$ (f) $c_t(j)c_t(j) = 66$, $j \geq 5$ (g) $c_t(j)c_t(k-j) = 132$, $5 \leq j < k-j$

Figure 6: Non saturated fully leafed tree-like polyhexes

We now proceed to the enumeration of non-saturated tree-like polyiamonds.

Proposition 5.2. *For $k \geq 0$, the number $fltri_t(2k+3)$ of fixed, fully leafed polyiamonds trees of odd size $2k+3$ is given by the following expressions:*

k	0	1	2	3	4	5	6	7	8	$k \geq 9$
$fltri_t(2k+3)$	6	12	30	60	90	108	132	168	198	$24k-12$

Table 1: Number of non saturated polyiamonds of size $2k+3$

Proof. First notice that these polyiamonds have exactly one cell of degree 2 denoted x. We enumerate polyiamonds by exhibiting a set P of seven basic polyiamonds, each made of an inner set of cells of degree 3, to be adjacent to x. These 7 polyamonds types are shown in Figure 7. Three of these polyiamonds have a variable size and they appear in Figure 7 $(E), (F), (G)$. The black thicker segment appearing in each polyiamond of Figure 7 marks the cell of the polyiamond that is to be adjacent to the cell x of degree 2.

We claim that every fully leafed non saturated tree-like polyiamond is obtained by choosing a pair of structures and by positioning them adjacent to x. In order to avoid duplication of cases, we assume that structures F and G contain respectively at least three and four cells of degree three.

For instance, the pair $\{D, F\}$ generates 12 fixed polyiamonds of size $2k+3$ for any $k \geq 7$, one of which, of size $n(T) = 19$, is illustrated in Figure 8. The enumeration of these non saturated polyamonds is done by finding the number of fixed polyiamonds that are obtained with every pair $\{X, Y\}$. These values are presented in Table 2. The numbers $fltri_t(2k+3)$ of non saturated polyiamonds presented in Table 1 are obtained from Table 2. □

The enumeration of polyhexes and polyiamonds presented in Propositions 5.1 and 5.2 have been independently verified with computer programs that iteratively enumerate non saturated tree-like polyforms of size n by repeatedly grafting cells of degree three to a cell of degree two [deC18]. We know from the degree distribution of the inner cells of non saturated polyiamonds that all fully leafed tree-like polyhexes and polyiamonds can be

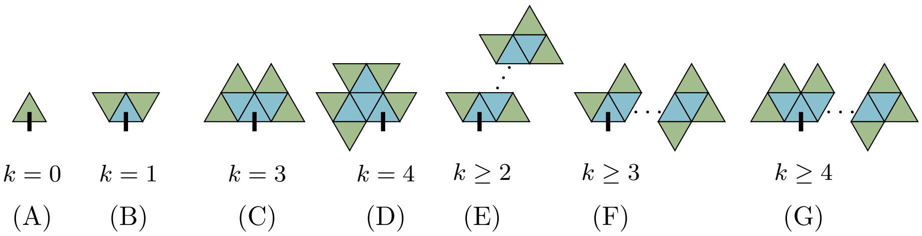


Figure 7: The set P of polyiamonds in non saturated fully leafed tree-like polyiamonds

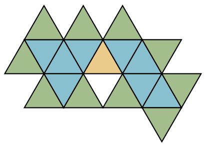


Figure 8: A fully leafed polyiamond of size $2k + 3$, $k \geq 7$.

	(A)	(B)	(C)	(D)	(E)	(F)	(G)
(A)	6 $k = 0$	12 $k = 1$	12 $k = 3$	12 $k = 4$	24 $k \geq 2$	12 $k \geq 3$	12 $k \geq 4$
(B)		6 $k = 2$	0	12 $k = 5$	12 $k \geq 3$	12 $k \geq 4$	0
(C)			0	0	0	0	0
(D)				6 $k = 8$	12 $k \geq 6$	12 $k \geq 7$	0
(E)					$6(k-3)$ $k \geq 4$	$12(k-4)$ $k \geq 5$	0
(F)						$6(k-5)$ $k \geq 6$	0
(G)							0

Table 2: Number of polyiamonds obtained from pairs in the set P

obtained this way. Indeed, since non saturated polyiamonds have one cell x of degree two and k cells of degree three, all these polyiamonds can be obtained by the graft of cells of degree three one at a time.

6 Concluding remarks

The next step in the enumeration of fully leafed polyforms and polycubes is the enumeration of non saturated polyominoes and polycubes and also of saturated polycubes. In the context of enumeration of families of polyforms and polycubes, we have shown that bijections can be used to give an evaluation of the complexity of the problems. For instance, we know that the enumeration of saturated polyominoes of size $4k + 1$ is precisely equivalent to the enumeration of tree-like polyominoes of size k. It would certainly be interesting to have a similar result for the enumeration of saturated tree-like polycubes.

References

[BCG18] A. Blondin Massé, J. de Carufel and A. Goupil. Saturated fully leafed tree-like polyforms and polycubes. At *https://arxiv.org/abs/1803.09181*, 2018.

[BCGLNV18] A. Blondin Massé, J. de Carufel, A. Goupil, M. Lapointe, É. Nadeau, and É. Vandomme. Leaf realization problem, caterpillar graphs and prefix normal words. *Theoretical Computer Science*, 732:1–13, 2018.

[BCGLNV18] A. Blondin Massé, J. de Carufel, A. Goupil, M. Lapointe, E. Nadeau and É Vandomme. Fully leafed induced subtrees. At *https://arxiv.org/abs/1709.09808*, 2018.

[BCGS17] A. Blondin Massé, J. de Carufel, A. Goupil and M. Samson. Fully Leafed Tree-Like Polyominoes and Polycubes. *In: L. Brankovic, J. Ryan, W. F. Smyth (Eds.): Combinatorial Algorithms / 28th International Workshop, IWOCA 2017, Newcastle, NSW, Australia*, *Lecture Notes in Computer Science*, 10765:206–218, 2017.

[BFLRS14] P. Burcsi, G. Fici, Z. Lipták, F. Ruskey and J. Sawada. On combinatorial generations on prefix normal words. In: *Proceedings of the 28th Annual Symposium on Combinatorial Pattern Matching (CPM 2014)*, *Lecture Notes in Computer Science*, 8486:60–69, 2014.

[BFLRS17] P. Burcsi, G. Fici, Z. Lipták, F. Ruskey and J. Sawada. On prefix normal words and prefix normal forms. *Theoretical Computer Science*, 659:1–13, 2017.

[deC18] J. de Carufel. Python programs to enumerate fully leafed tree-like polyhexes and polyiamonds. at *https://github.com/decaruju/polyhex-polyiamond-enumerator*, 2018.

[FL11] G. Fici and Z. Lipták. On prefix normal words. In: *Proceedings of the 15th International Conference on Developments in Language Theory (DLT 2011)*, *Lecture Notes in Computer Science*, 6795:228–238, 2011.