



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Extremal Degree-Based Topological Indices of General
Polyomino Chains via Dynamic Programming

Tesis presentada al

Colegio de Matemáticas

como requisito parcial para la obtención del grado de

LICENCIADO EN MATEMÁTICAS

por

Manuel Montes y Morales

Asesorado por

Dr. Hugo Adán Cruz Suárez

y

M.Sc. Saylé Caridad Sigarreta Ricardo

Puebla Pue.

Febrero de 2026



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Extremal Degree-Based Topological Indices of General
Polyomino Chains via Dynamic Programming

Tesis presentada al

Colegio de Matemáticas

como requisito parcial para la obtención del grado de

LICENCIADO EN MATEMÁTICAS

por

Manuel Montes y Morales

Asesorado por

Dr. Hugo Adán Cruz Suárez

y

M.Sc. Saylé Caridad Sigarreta Ricardo

Puebla Pue.

Febrero de 2026

Título: Extremal Degree-Based Topological Indices of General Polyomino Chains via Dynamic Programming

Estudiante: MANUEL MONTES Y MORALES

COMITÉ

Dr. Carlos Alberto López Andrade
Presidente

Dr. Rei Israel Ortega Gutiérrez
Secretario

Dr. Carlos Guillén Galván
Vocal

Dr. Hugo Adán Cruz Suárez
Asesor

M.Sc. Saylé Caridad Sigarreta Ricardo
Coasesora

Agradecimientos

La culminación de este trabajo de tesis ha sido una enorme experiencia en mi camino académico, la cual tengo que agradecerle al Dr. Hugo Adán y la M.SC. Saylé Sigarreta por haberme introducido al área y aceptado asesorarme e incluirme dentro de su grupo de investigación. En particular, debo dar las gracias a Saylé, la cual ha pasado a ser mi compañera de trabajo y con la que he compartido cientos de horas de audios discutiendo ideas, sin las cuales este trabajo no sería lo que es ahora.

Además, quiero agradecer a mis padres y abuelos que desde pequeño me han inculcado el gusto por la ciencia y la curiosidad por conocer. Ellos han sido mis mayores ejemplos de personas excelentes dentro de todas sus áreas y es a ellos a quienes les debo mi pasión por la resolución de problemas y muchos de mis valores como persona. Quiero agradecer también a mi novia María, a la que amo con todo mi ser y me ha acompañado desde el inicio de mi camino en las matemáticas, ella ha sido mi pilar fundamental y no puedo agradecerle suficiente lo mucho que me ha apoyado en todas mis metas.

Finalmente quiero agradecer a mis amigos Pablo, Abril y Lulú, los cuales siempre han estado ahí para mí y alegrado mis días.

Table of Contents

Abstract	5
1. Introduction	1
1.1. Introduction to Extremal Chemical Graph Theory	1
1.1.1. Introduction to Chemical Graph Theory	1
1.1.2. Topological Indices and Degree Based Topological Indices	2
1.1.3. Polyomino Chains and the Extremal Problem	4
1.2. Links and Classes of Polyomino Chains	6
1.2.1. Polyomino Chains and General Polyomino Chains	6
1.2.2. Links and Results in the Literature	9
2. Extremal Polyomino Chains	13
2.1. Actions	14
2.1.1. Neighborhoods of Impact and Actions	14
2.1.2. Recursive Construction and Connection with Links	17
2.1.3. Independence of f	18
2.2. Generating One Extremal Polyomino Chain	19
2.2.1. Is the Problem Greedy?	19
2.2.2. Dynamic Programming Approach to Build an Extremal Polyomino Chain .	20
2.2.3. Cyclic Behavior of Extremal Polyomino Chains	23
2.2.4. One Extremal Polyomino Chain of the Augmented Zagreb Index	26
2.3. Counting and Generating All Extremal Polyomino Chains	29
2.4. Long-Term Behavior of Extremal Polyomino Chains and their Bang-Bang Dynamic	35
2.5. Isomorphic Polyomino Chains and their Extremal Dynamics	40
2.5.1. The Flip of a Polyomino Chain	40
2.5.2. How Many Non-Isomorphic Polyomino Chains of n Squares are there? . . .	45
2.5.3. Extremal Isomorphic Polyomino Chains	47
3. Extremal General Polyomino Chains	51

3.1. General Polyomino Chains and General Actions	52
3.1.1. General Actions and the Problem of Realizability	52
3.1.2. Connection of General Actions with Links	55
3.2. Generating One Extremal General Polyomino Chain	56
3.2.1. Making a General Action Sequence Realizable	56
3.2.2. Dynamic Programming Approach to Build an Extremal General Polyomino Chain	61
3.3. Counting and Generating All Extremal General Polyomino Chains	65
3.3.1. Counting and Generating All Extremal General Action Sequences	65
3.3.2. Cyclic Behavior of Extremal General Polyomino Chains	69
3.3.3. All General Polyomino Chains for R_{-1}	73
3.4. Polyomino Chains vs General Polyomino Chains	78
3.4.1. Conditions such that All Extremal General Action Sequences are Action Sequences	78
3.4.2. Conditions such that there are Extremal General Action Sequences Which are not Action Sequences	82
Conclusion	85
Bibliography	87
Appendix	89

Abstract

Based on the results of our works *Maximum augmented Zagreb index on polyomino chains* [22], which was published in the journal *Applied Mathematics and Computation*, and *Extremal degree-based indices of general polyomino chains via dynamic programming* [21], which is already submitted in a peer reviewed journal, in this work, we focus on the problem of finding extremal graphs with respect to degree-based topological indices, which is a major area in chemical graph theory and plays a fundamental role in the design of chemical compounds, especially as part of the QSPR analysis. In particular, we center our study in polyomino chains, which is an important graph family in the context of the extremal problem.

Dividing the problem into a restricted setting and the general setting of the problem, we develop a dynamic programming framework, by means of a new encoding of polyomino chains based on local geometric descriptions which we call actions, for identifying extremal polyomino chains with respect to any degree-based topological index. Specifically, for the restricted version of the problem, our approach provides an explicit recurrence and a constructive algorithm that enable both the computation of an extremal restricted polyomino chain in linear time with respect to the number of squares and the enumeration of all extremal restricted polyomino chains in linear time with respect to their amount, being able to retrieve all non-isomorphic extremal restricted polyominoes in quadratic time with respect to the amount of extremal restricted chains. On the other hand, for the general problem, we generalize our approach and as a consequence obtain methods for both the computation of an extremal polyomino chain in linear time with respect to the number of squares and the enumeration of all extremal polyomino chains, with the complication the time complexity for the latter is no longer linear and depends on the multiple interpretations of the encodings.

As concrete application of both settings, we resolve two open problems posed in 2016 and 2015 respectively, by determining, for any given number of squares, all the restricted polyomino chains that maximize the Augmented Zagreb Index, AZI , and all the polyomino chains that maximize the generalized Randić index with parameter $\alpha = -1$, R_{-1} , which are both relevant indices due to their numerous connections with chemical characteristics.

Moreover, beyond the specific contributions of our work, we importantly establish conditions for cyclic patterns to appear in extremal constructions, conditions so that the linear and zig-zag chains (which are central in most of the existent results in the literature) appear as extremal and conditions so that solving the extremal problem can be reduced to solving it in the more simple restricted setting. In particular, as a consequence of all this theoretic development, we immediately retrieve most of the literature’s results and provide a constructive and systematic methodology for tackling extremal problems in graph theory, for which we provide a link to the code in the conclusion section.

Keywords: *Extremal Graph Theory, Degree-Based Topological Indices, Polyomino Chains, Augmented Zagreb Index, Generalized Randić Index, Dynamic Programming.*

Chapter 1

Introduction

1.1. Introduction to Extremal Chemical Graph Theory

1.1.1. Introduction to Chemical Graph Theory

The study of molecular structure is a central and crucial part of chemistry. Based on the fact that all biological functions depend on events that occur at the molecular level and that these themselves are directed by large or clusters of molecules, the constitution of molecular structure becomes the key to unlock nature's blueprint. The purpose of studying molecular structure is then to be able to explain and predict the properties and behavior of substances, giving us the capacity to comprehend biological systems, design new drugs and materials, and further develop technology in areas such as medicine and environmental science.

At the current time, quantum mechanics is the ultimate approach to the exact quantification of molecular structure. Given the coordinates and atomic numbers for a collection of atoms, the Schrödinger equation, in principle, can be solved for the eigenvalues and eigenvectors that describe the energy and electron distribution, from which other properties of the molecular aggregate can be then derived from the wave function and energy [19]. Hence, a complete quantum mechanical treatment may give us as a result our desired connection between molecular structure and physicochemical properties. Nevertheless, quantum mechanical approaches have a high computational complexity, not to mention the approximate calculation nature of all their computational versions result in a substantial loss of information, especially for large molecules and interacting systems [19]. Therefore, it is necessary to provide a substitute for another quantitative way of describing molecular structure, in such a way that their computational complexity or characterization is easy to afford and study. It is here that we find the force and importance of chemical graph theory, which is a branch of mathematical chemistry that, through graph theoretic concepts, is concerned with the analysis of all consequences of connectivity.

From its conception, graph theory has proved to be a very powerful mathematical tool used to model an immense variety of problems, both inside mathematics and in the natural sciences. In particular, out of all natural sciences, chemical systems are the ones which present as a common feature the presence of sites and connections between them; where sites may be atoms, electrons, molecules, molecular fragments or orbitals, and connections may represent bonds, elementary reaction steps or Van der Waals forces [29].

As a special type of chemical system, there is hardly any which is closer to the notion of a graph than the constitutional formula of a chemical compound; where individual atoms correspond to

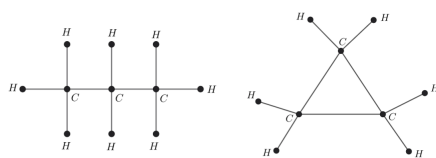


Figure 1.1: The Molecular Graphs for Propane and Cyclopropane.

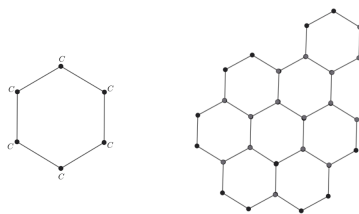


Figure 1.2: The Molecular Graphs for Benzene and Benzo[a]corene.

vertices and chemical bonds to edges between them. In Figure 1.1 we show the graph representation of both propane and cyclopropane, in which we leave at their vertices the chemical labels of the corresponding elements; while in Figure 1.2 we show a more simplified graph representation of molecules where we have removed the hydrogen bonds and depict benzene and benzo[a]corene respectively.

Having established this relation between chemical compounds and graphs, chemists and mathematicians have used it as an advantage in order to classify, systematize, enumerate and design chemical systems of interest, along with the generation of new concepts based on the underlying structure and the regularities revealed by chemical data [6].

It is as far back as 1843, that graph-theoretical results started to get introduced into the area, with Laurent and Gerhardt establishing that the number of atoms of odd valence in a molecule was always even and the many observations of homologous series made by Kekulé, who managed to visualize the formation of both double and triple bonds and give a structural formula for benzene. Later on, with the rising problem of isomer identification, mathematicians like Cayley were able to obtain isomer counts for alkanes by the use of generating functions, and after that, many combinatorial graph theoretic techniques like symmetry classes and permutation groups were incorporated into the study of chemical structures such as aldehydes, amines, ethers and organic acids [6]. By 1986 many areas within the study of chemical graph theory had been introduced, such as isomer enumeration, additivity studies, chemical bonding theory and the study of topological indices; having in that sole year, more than 500 papers published [6].

In the following section, we will introduce our area of interest, which is the study of topological indices.

1.1.2. Topological Indices and Degree Based Topological Indices

As a motivational example, we can take a look at the molecular graphs of butane and isobutane, shown in Figure 1.3, which despite being closely related, clearly have very different structures. However, how different are they? Is it possible to assign some numerical value to the graphs of molecules so that differences in structure could be quantified? Moreover, is it possible to differentiate, numerically, molecular structures sufficiently so that significant correlations with physical,

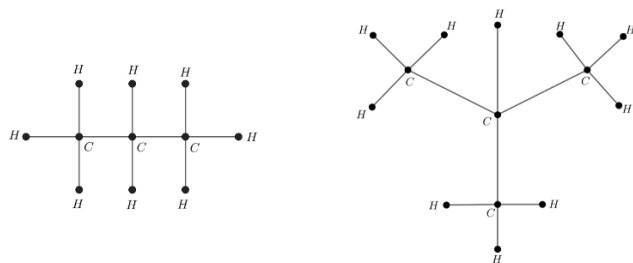


Figure 1.3: The Molecular Graphs for Butane and Isobutane.

chemical, and biological properties are established? The answer to both of these last questions is yes and the mathematical mechanism to do so is through the concept of topological indices, which are single numbers used to characterize the graph of a molecule regardless of the ways it could be drawn or labeled.

The interest in topological indices is then apparent to the organic and medicinal chemist, who can use them as a means to form quantitative structure-activity relationships (QSAR) and structure-property relationships (QSPR), which are essential in areas such as pharmacology, toxicology, environmental chemistry and drug design [29]. Furthermore, the non-empirical and low computational cost of topological indices, has made them an indispensable tool for studies where experimental testing and exact computation are unaffordable (such as analyzing the relative potency of drugs or studying intermolecular interaction), not to mention that their study has brought them to correlation with some physical properties to a point near the experimental limit of the values [19].

As examples of some of the properties topological indices can quantify we have: solubility, heat of atomization, stability, the refractive index, strain energy, heat of vaporization, the partition coefficient and π -electron energy [19]; which in turn give a wide range of applications, as are the optimization of pharmacological, biocidal and pesticidal activity, the designing out and prediction of toxicity, the selection of compounds with optimal pharmacokinetic properties and the optimization of drug activity, specially as a tool for filtering drug candidates out of large chemical databases [28].

Now, in order to distinguish topological indices by the way they capture the topological structure of graphs, we divide them into three main categories: Degree-based, Distance-based and Spectrum-based; from which we shall focus on the first for this work and present their formal definition in Definition 1.1.1.

Topological degree-based indices are especially useful in chemical graph theory. For instance, the Geometric-Arithmetic index (GA) has been successfully employed as a predictive tool in QSPR/QSAR studies, being one of its applications in the analysis of drugs used in blood cancer treatment [32]. Similarly, the Atom-Bond Connectivity index (ABC) has proven valuable in modeling molecular properties such as polarity and refractivity [13] and the Randić index (R), which helps quantify the notion of molecular branching, has shown strong correlations with the solubility of alkanes in water [26], as well as relevance in the study of polymers [9]. In addition, many indices such as the Zagreb indices, which estimate the π -electron energy, and the augmented Zagreb index (AZI), which has shown strong correlation with standard heats of formation, boiling points and the capture of structural variations among families of isomers [11, 14], have been successfully used in several studies, such as in the treatment of coronavirus (COVID-19) [30].

Definition 1.1.1. A *topological degree-based index* is a function TI_f defined by a general formula

of the form:

$$TI_f(G) = \sum_{uv \in E(G)} f(d_u, d_v)$$

where G is a graph, $E(G)$ is the set of edges of G , d_u and d_v are the degrees of the vertices u and v , respectively, and f is a real-valued symmetric function, that is, $f(x, y) = f(y, x)$ for all $x, y \in \{1, 2, \dots\}$.

Remark 1.1.2. Notice as an immediate consequence of asking function f symmetric, we have the topological degree-based index will be a graph invariant; that is $TI_f(G_1) = TI_f(G_2)$ whenever $G_1 \cong G_2$.

Over the years, more than 120 topological indices have been developed for the study of physico-chemical properties. In Table 1.1 we show the most important and central degree-based indices as discussed in [17], where we include indices such as the Generalized Atom-Bond Connectivity Index and the Generalized Randić Index, which depend on a parameter $r \in \mathbb{R}$ and have been widely studied due to the relevance of their special cases.

Degree-based Index	$f(x, y)$
Randić Index (R)	$\frac{1}{\sqrt{xy}}$
Sum Connectivity Index (χ)	$\frac{1}{\sqrt{x+y}}$
Geometric Arithmetic Index (GA)	$\frac{\sqrt{xy}}{\frac{1}{2}(x+y)}$
Atom Bond Connectivity Index (ABC)	$\sqrt{\frac{x+y-2}{xy}}$
Harmonic Index (H)	$\frac{2}{x+y}$
First Zagreb Index (M_1)	$x + y$
Second Zagreb Index (M_2)	xy
Augmented Zagreb Index (AZI)	$\left(\frac{xy}{x+y-2}\right)^3$
Generalized Randić Index (R_r)	$(xy)^r$
Generalized Sum Connectivity Index (χ_r)	$\left(\frac{1}{\sqrt{x+y}}\right)^r$
Generalized Atom Bond Connectivity Index (ABC_r)	$\left(\frac{x+y-2}{xy}\right)^r$

Table 1.1: Main degree-based topological indices and their associated functions $f(x, y)$.

1.1.3. Polyomino Chains and the Extremal Problem

As discussed in the preceding sections, the need is evident to start studying and characterizing the properties of concrete graph structures related to chemical compounds. In particular, in this work we'll focus on the study of polyomino chains, which are graph configurations relevant in chemical graph theory as they can be used to model a variety of molecular structures, including polymers, crystal lattices, polycyclic compounds, and certain classes of organic compounds [10]. In addition, thanks to their well structured geometric construction, it has been possible to treat them

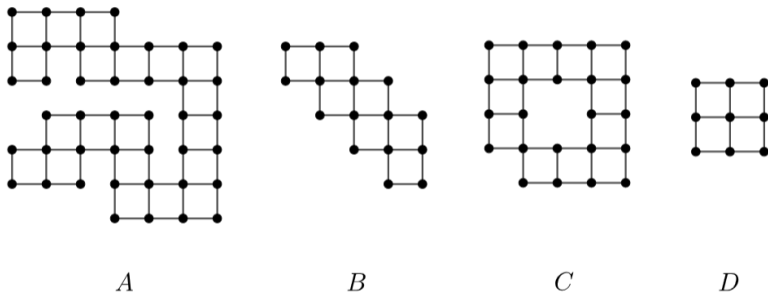


Figure 1.4: Examples of polyomino chains (graphs A and B) and of non-polyomino chains (graphs C and D).

as a meaningful example in the study of other related molecules, which has been the main reason for their extensive examination, specifically in the context of degree-based topological indices [1, 18].

Definition 1.1.3. Let G be a graph.

- a) G is a *polyomino system* if it is finite, 2-connected, and every interior face (also called a *cell*) is a unit square with a side length of one.
- b) The *inner dual graph* of a polyomino system G is the plane graph whose vertices are the cells; two vertices are adjacent if and only if the corresponding cells share an edge.
- c) A *polyomino chain* is a polyomino system G whose inner dual is a path. In particular, a polyomino chain can be constructed iteratively by adding unit squares one by one in the order induced by that path.

In figure 1.4 we show four graphs in which we give examples of polyomino chains (graphs A and B) and examples of others which aren't (C and D). In this case, notice the reason why graph C isn't a polyomino chain comes directly from the fact of not even being a polyomino system, since one of its interior faces isn't surrounded by a regular square. In the case of graph D, even though it is a polyomino system, it isn't a polyomino chain, since its dual graph isn't a path.

Now, with respect to the study of topological indices in concrete classes of graphs, there has been a lot of work towards their characterizations, focusing on their connection to other graph theoretical objects and their extremal values, which is the problem of finding for a specific topological index the graphs that maximize or minimize it, being one of the first to highlight their mathematical difficulty Paul Erdős and Béla Bollobás with their study on the Randić Index [5] in 1998. In particular, considerable attention has been devoted to characterizing extremal polyomino chains with respect to degree-based indices [3, 4, 23, 24, 27, 31]. In this problem, due to the fact polyomino chains can be thought of as paths made out of squares, the question can be phrased into: for a given natural number n and a topological degree-based index TI_f , which polyomino chain, G , with n squares maximizes (or minimizes) $TI_f(G)$?

As an example, in Figure 1.5 we show the seven non-isomorphic polyomino chains with 5 squares that exist, for which we have used the Randić index, $f(x, y) = \frac{1}{\sqrt{xy}}$. In this case, the graphs G_1 and G_7 correspond to the polyomino chains with 5 squares that maximize and minimize the expression $TI_f(G)$ respectively.

At this point, it is important to highlight without further tools, the problem of finding a extremal polyomino chain of a fixed number of squares for a topological degree-based index isn't computationally efficient. Naively, since we know polyomino chains must form a path, we could start placing one square after the other and begin to build every possible graph candidate by

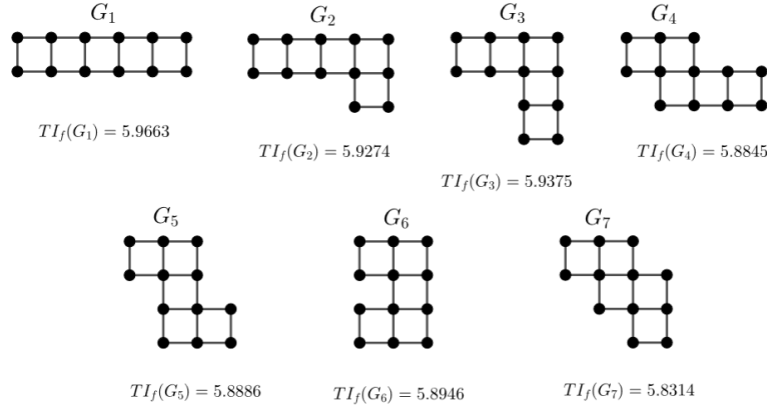


Figure 1.5: Finding the extremal polyomino chains of 5 squares with respect to the Randić index.

exhausting the different choices of placements for each square (which are three). However, the amount of these possible candidates is exponential with respect to the number of squares, not to mention that many of them will not even be polyomino chains, which forces us to have a way to tell them apart. Moreover, variations of the main problem add additional complications, like which is the set of all extremal polyomino chains? Which is the set of all non-isomorphic extremal polyomino chains? How do extremal polyomino chains change with the change in the number of squares? In the case of parametrized indices, how do extremal polyomino chains behave for different values of the parameters? Or, if we want a specific polyomino chain to be extremal, how must the topological degree index be?

In the course of this work we will come to study all of these questions.

1.2. Links and Classes of Polyomino Chains

1.2.1. Polyomino Chains and General Polyomino Chains

By now, we have described polyomino chains through their graph theoretic definition. However, we need a better way to encode polyomino chains if we are to work in algorithms that are able to generate them. Thus, in order to do so, we'll take advantage of both the path representation underlying a polyomino chain, as well as their strong geometric description when thought as embedded in $\mathbb{R}^2$. For what follows we refer as the *construction process of a polyomino chain* to the process of consecutively adding its squares to the graph, in the order given by its path representation.

Definition 1.2.1. Let PC_n be a polyomino chain with n unit squares embedded in $\mathbb{R}^2$.

- a) The chain is *horizontal*(resp. *vertical*) at step $i \geq 2$ of its construction process if the (i) -th square is placed to the right/left(resp. up/down) of the $(i - 1)$ -th. The corresponding *orientations* are encoded by the instructions set

$$\mathcal{I} = \{R, L, U, D\} \quad (\text{right, left, up, down}).$$

Although introduced for the chain, direction/orientation at step i depend only on how the (i) -th square is attached; thus we may regard the (i) -th square itself as carrying them. As a convention, since all two-square chains are isomorphic, we fix PC_2 to be horizontal with right orientation (see Figure 1.6)

Figure 1.6: The graph PC_2 .

- b) By part a) of this definition, any chain PC_n determines a sequence of instructions given by the orientations of its squares

$$(I_1, I_2, \dots, I_n), \quad I_k \in \mathcal{I}$$

where we adopt the convention $I_1 = I_2 = R$ and we write the chain as $PC(R, R, I_3, \dots, I_n)$. We call $(R, R, I_3, \dots, I_n)$ a *sequence of instructions*.

The use of sequences of instructions makes now straightforward the exhaustive search over all possible graph candidates for extremal polyomino chains. Nevertheless, the issue regarding the question of whether all of these candidates truly correspond to polyomino chains still remains, since a sequence of instructions may not necessarily correspond to one. For this purpose, observe that the graph theoretic definition of a polyomino chain does not give us any manageable criteria, especially when it comes to evaluate if a graph is a polyomino chain as we keep placing consecutive squares one by one. So, in order to establish a useful equivalence, we must first give a few additional definitions and elaborate on the definition of an interior face, which is used in Definition 1.1.3 to define a polyomino system.

Definition 1.2.2. Let $(R, R, I_3, \dots, I_n)$ be a sequence of instructions with $I_k \in \mathcal{I}$. The sequence is *realizable* if the full construction of the graph given by consecutively placing squares in the orientations of I_k , as stated in Definition 1.2.1, yields a polyomino chain; in this case we denote it by $PC(R, R, I_3, \dots, I_n)$.

Definition 1.2.3. When carrying out the (i) -th step of the constructive process, since the addition of the i -th square to the $(i-1)$ -th involves only one edge of the former, it follows two of its vertices remain unused in the connection. We refer to these vertices as *the ending vertices of the i -th square*.

Definition 1.2.4. An *interior face* of a graph is a region enclosed by the graph's edges when it is drawn on a plane without any edges crossing. More formally, an interior face of a graph is a region enclosed by a cycle of vertices of the graph, such that there doesn't exist any other cycle between a subset of those vertices.

Lemma 1.2.5. Let $(I_1, I_2, \dots, I_n)$ be a sequence of instructions. If at every step of its constructive process the last square has its ending vertices with degree exactly two, then the sequence is realizable.

Proof. We argue by induction on i , the step of the constructive process.

For $i = 1$ it is clear the graph consisting of only the first square is a polyomino chain. So assume for $i \geq 1$ the graph consisting of the first i squares is a polyomino chain and that when we attach the $(i+1)$ -th square to the structure by joining one of its edges to another of the (i) -th, its ending vertices have degree exactly two.

In order to bring in notation, let's label the vertices of the $(i+1)$ -th square as v_1, v_2, v_3 and v_4 ; where v_2 and v_3 are its ending vertices (see Figure 1.7).

Suppose there exists a cycle involving either v_2 or v_3 , namely $\mathcal{C}$ and assume without loss of generality $v_2 \in \mathcal{C}$. Then because any vertex in a cycle has an edge with at least two others, and v_2 has an edge only with v_1 and v_3 , it follows $v_1, v_3 \in \mathcal{C}$. Similarly, since $v_3 \in \mathcal{C}$ but it only has edges with v_2 and v_4 , we have $v_4 \in \mathcal{C}$. Therefore, we can conclude any cycle having either v_2 or v_3 , must have the vertices v_1, v_2, v_3 and v_4 , which in particular form a cycle, namely the $(i+1)$ -th square.

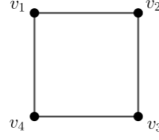


Figure 1.7: Labeled $(i + 1)$ -th square.

Thereby, the only interior face using v_2 or v_3 is the $(i + 1)$ -th square and since all other interior faces come from using solely vertices from the first i squares, which by the induction hypothesis must correspond to 1-unit squares, it follows the graph consisting of the first $i + 1$ squares is a polyomino chain.

By induction we conclude the sequence of instructions $(I_1, I_2, \dots, I_n)$ is realizable. $\square$

Remark 1.2.6. As a matter of fact it is also possible to show if G is a polyomino chain, then at every step of its constructive process the last square has ending vertices with degree exactly two. However, we won't prove this result here, as it needs the use of a theorem which is not relevant for the rest of our theory. Nevertheless, we provide the proof and required theoretical details to show Proposition 1.2.7 in the [Appendix](#).

Proposition 1.2.7. *Let $(I_1, I_2, I_3, \dots, I_n)$ be a sequence of instructions. The sequence is realizable iff at every step of its constructive process the last square has its ending vertices with degree exactly two.*

Remark 1.2.8. Notice by performing an exhaustive search over the set of all sequences of instructions $(R, R, I_3, \dots, I_n)$ and using an auxiliary matrix in order to geometrically realize the graph associated with the sequence, it is possible to check for the validity of Lemma 1.2.5 as well as compute the topological index value. This algorithm effectively gives us the exponential brute force search for the extremal polyomino chains.

By means of Lemma 1.2.5, the problem of knowing if a given sequence of instructions corresponds to a polyomino chain has been solved. However, in theory, it has been hard to manage techniques that deal with the issue of encodings not necessarily corresponding to our object of study. Therefore, many works have focused in finding extremal graphs for a specific family of polyomino chains which avoids the encoding misrepresentation [4, 15, 27, 31]. This family of polyomino chains can be expressed in terms of instructions, as those composed of only instructions R and D , and is called the family of *restricted polyomino chains*.

In the following proposition we show, that as desired, sequences of only instructions R and D correspond to polyomino chains. In order to do so, we'll apply Lemma 1.2.5 and instead of working with the placement of squares, we'll work in the dual graph which can be thought of been realized in $\mathbb{R}^2$, with the first vertex being placed at $(0, 0)$. Notice that in such scenario, the condition of the lemma is equivalent to the last vertex, with coordinates (x, y) , attached to the graph not having neighbors at positions $(x, y - 1)$, $(x + 1, y - 1)$, $(x + 1, y)$, $(x + 1, y + 1)$ and $(x, y + 1)$ when the last vertex was placed as a result of an R instruction (Figure 1.8, a)) or at positions $(x + 1, y)$, $(x + 1, y + 1)$, $(x, y + 1)$, $(x - 1, y + 1)$ and $(x - 1, y)$ when the last vertex was placed as a result of a D instruction (Figure 1.8, b)).

Proposition 1.2.9. *Any sequence of instructions $(R, R, I_3, \dots, I_n)$ in which $I_k \in \{R, D\}$, is realizable.*

Proof. We proceed by induction on n , the number of instructions, and help ourselves with the invariant if the last vertex in the construction process has coordinates (x, y) , then the number of instructions R and D equal x and y respectively.

For a sequence of 1 instruction, it is clear the graph consisting of only one vertex at $(0, 0)$ satisfies Lemma's 1.2.5 hypothesis and the number of R and D instructions is equal to 0.

Assume for any sequence of $n - 1$ instructions at each step of the constructive process the last vertex (x, y) attached satisfied Lemma's 1.2.5 hypothesis and the number of R and D instructions was equal to x and y respectively. Now, take any sequence of n instructions and assume without loss of generality the (n) -th instruction was R . We must show if the $(n + 1)$ -th vertex has coordinates (x, y) , that there aren't vertices at positions $(x, y - 1)$, $(x + 1, y - 1)$, $(x + 1, y)$, $(x + 1, y + 1)$ and $(x, y + 1)$, as show in Figure 1.8, a).

By the induction hypothesis applied to the (n) -th vertex, which we know has coordinates $(x - 1, y)$ (because the instruction used for the $(n + 1)$ -th vertex was R), it follows from the first $n - 1$ instructions in the sequence, exactly $x - 1$ of them are R and y are D . But this means it is impossible any of the first n vertices has positions $(x, y - 1)$, $(x + 1, y - 1)$, $(x + 1, y)$, $(x + 1, y + 1)$ or $(x, y + 1)$, for either they need more R instructions or more D instructions. Therefore, the $(n + 1)$ -th square corresponding to the $(n + 1)$ -th vertex in the dual graph, will have its ending vertices with degree exactly two.

The case in which the $(n + 1)$ -th instruction is D is completely analogous, taking into account the relevant positions to show are empty are the ones show in Figure 1.8, b). Hence, we conclude the sequence of instructions $(R, R, I_3, \dots, I_n)$ is realizable.

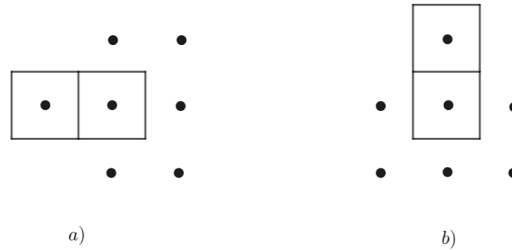


Figure 1.8: Cases depending whether the $(n + 1)$ -th instruction is R or D respectively.

□

Remark 1.2.10. Due to the relevance and depiction in the literature of restricted polyomino chains as only polyomino chains, we shall also in this work reserve the term polyomino chain for the restricted family, while we shall say *general polyomino chain* in the case the graph satisfies no other additional condition than those stated in Definition 1.1.3.

1.2.2. Links and Results in the Literature

At the current time, the study of extremal general polyomino chains has been centered around figuring out ways to capture the geometric structure of the graphs, with the intention of providing a closed formula to optimize. Examples of these structural encodings can be found in [27], which studies polyomino chains (the restricted class) through the polyomino's link representation and [3], which studies general polyomino chains through the polyomino's structural vectors.

In essence, both of these particular representations try to capture the geometry of how a given polyomino chain turns and twists in the plane which, as a matter of fact, capture it in such a way that from any of the two different encodings, it is possible to uniquely determine the associated chain. Due to their geometric simplicity (which comes from being defined only for restricted polyomino chains) and their useful comparison with our own study, we will now

introduce the concept of the link sequence for a polyomino chain in order to illustrate one such geometric-based representation.

Definition 1.2.11. Let PC_n be a polyomino chain encoded by the sequence of instructions $(R, R, I_3, \dots, I_n)$; $I_k \in \{R, D\}$. Its *sequence of links* is a sequence of the form:

$$(1, 1, L_3, \dots, L_n), \quad L_k \in \{1, 2\}$$

for $k > 2$, $L_k = 1$ if $I_k = I_{k-1}$ and $L_k = 2$ otherwise. We call each of the values L_k a link of the sequence.

Remark 1.2.12. Notice what links represent is the geometric notion of change of direction, which is possible because of the restricted growth of polyomino chains.

With the use of these geometric-based representations, it has been possible to calculate the value of topological degree-based indices for any general polyomino chain via a closed formula and for some of them even find extremal chains, which as it turns out have been found in their majority to be either linear or zig-zag chains (as defined in Definition 1.2.13). In Theorem 1.2.14 we summarize the main results presented in [3], where most of the general characterizations that exist are collected.

Definition 1.2.13. Let PC_n be a polyomino chain encoded by the sequence of instructions $(R, R, I_3, \dots, I_n)$.

- a) We call PC_n a *linear chain* if every instruction $I_k = R$; $1 \leq k \leq n$, and we denote it by Li_n (see Figure 1.9).
- b) We call PC_n a *zig-zag chain* if every instruction $I_k = R$ for k even and $I_k = D$ for k odd; $2 \leq k \leq n$, and we denote it by Z_n (see Figure 1.9).

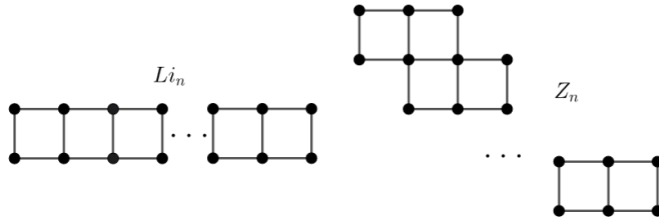


Figure 1.9: The linear and zig-zag polyomino chains.

Theorem 1.2.14. Let PC_n denote any general polyomino chain with n squares:

$$\begin{array}{ll} M_i(Li_n) \leq M_i(PC_n) \leq M_i(Z_n) & AZI(Li_n) \leq AZI(PC_n), \quad n \geq 6 \\ H(Z_n) \leq H(PC_n) \leq H(Li_n) & ABC(PC_n) \leq ABC(Z_n) \\ R(Z_n) \leq R(PC_n) \leq R(Li_n) & R_r(Li_n) \leq R_r(PC_n) \leq R_r(Z_n), \quad r > 0 \\ GA(Z_n) \leq GA(PC_n) \leq GA(Li_n) & ABC_r(Li_n) \leq ABC_r(PC_n) \leq ABC_r(Z_n), \quad r > 1 \\ \chi(Z_n) \leq \chi(PC_n) \leq \chi(Li_n) & \chi_r(Li_n) \leq \chi_r(PC_n) \leq \chi_r(Z_n), \quad r > 0. \end{array}$$

However, despite the full characterization shown in Theorem 1.2.14 of the extremal polyomino chains for many of the most important indices, the approach of geometric representations has found difficulties in finding extremal chains for indices such as the augmented Zagreb index (AZI),

the atom-bond index (ABC), and in general for parametrized versions. In particular, geometric representations have not been successful when the extremal chains aren't linear or zig-zag, not to mention the case in which multiple families of chains correspond to the extremal values. In the investigation made by R. Cruz and J. Rada [24], they worked precisely on this issue, and through the use of linearization techniques and a lot of careful analysis, they managed to find one family that minimizes the ABC index as well as others for more values (but not all) of the parameter in the generalized Randić index. Nevertheless, this case by case study has proven troublesome and limited to the appearance of concrete structural values, which ultimately lead to finding families of extremal chains one by one. Thus, there is a need for a new method which is able to study the extremal problem as a whole.

Chapter 2

Extremal Polyomino Chains

Based on the results of our work *Maximum augmented Zagreb index on polyomino chains* [22], which was published in the journal *Applied Mathematics and Computation*, in this chapter, as an important particular case of the problem, we center on finding the extremal chains for any degree-based topological index when we are restricted to using only polyomino chains. In order to do this, we first introduce a new way to encode the information behind polyomino chains, which we name their *action sequence* and is based on retrieving only their local geometric structure. With this tool, we then proceed to define a dynamic programming framework, through which we show that we can produce, for any fixed number of squares n , a polyomino chain that maximizes or minimizes the given topological index in linear time with respect to n . In particular, we employ this technique to find one polyomino chain with maximum augmented Zagreb index, which was until now an open problem.

Furthermore, by means of the dynamic programming design, we prove it is not only possible to obtain one extremal polyomino chain, but all of them (possibly isomorphic) in linear time with respect to their amount and as an application, we characterize all polyomino chains with maximum augmented Zagreb index. Also, based on the dynamic nature of the procedure, we study the impact of the values of the topological index in the behavior of its extremal chains, from which we establish the most extreme cases in the system lead precisely to extremal chains being either linear or zig-zag; thus retrieving as a consequence most of the characterizations that had been previously made in the literature and throwing some light to the question of why a certain family of polyominoes is extremal.

Finally, thanks to the restricted polyomino chain setting, we define the notion of the *flip* of a polyomino, with which we solve the problem of finding all non-isomorphic extremal polyomino chains for any degree-based topological index and give some combinatorial results as well as a further equivalence that connects the notion of extremal uniqueness with linear and zig-zag chains.

2.1. Actions

2.1.1. Neighborhoods of Impact and Actions

As we mentioned in Section 1.2.2, most of the encodings used to describe polyomino and general polyomino chains are based on capturing the graphs geometric structure as a whole. However, looking at the expression of the topological degree-based index we can see each edge is treated independently from the others, which gives us a feeling for a local effect:

$$TI_f(G) = \sum_{uv \in E(G)} f(d_u, d_v).$$

As a consequence, it is sufficient to study how the degrees associated to the endpoints of the edges of the graph change as we add a new square. Fortunately, these affected edges are not many, as can be appreciated from Figure 2.1 in which we depict the neighborhood around the last square in which edges might have the degree of their endpoints altered.

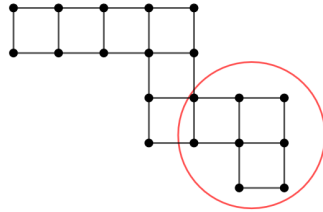


Figure 2.1: The neighborhood in which the evaluation of the edges under f could be altered.

Definition 2.1.1. Let PC_n be a general polyomino chain with n squares and let v_1 and v_2 be its two ending vertices. We define the *neighborhood of impact* produced by the last square of the polyomino as the subgraph G of PC_n which has as edges all of those in PC_n whose endpoints either connect with v_1 or v_2 or are one of them. The set of vertices of G is then the collection of all vertices in PC_n which are endpoints to an edge in G .

Also, we consider two neighborhoods of impact distinct iff they are not isomorphic or if the degrees of their vertices, taken from the respective chains they are subgraphs of, does not coincide under the isomorphism between them.

Having defined the notion of the neighborhood of impact produced by the last square, we notice that in general such a subgraph only takes edges out of the last three squares of the polyomino and vertices out of the last four, which motivates us to study the set of all possible neighborhoods. In Figure 2.2 we display all four non-isomorphic polyomino chains with four squares and color their respective neighborhoods of impact, through which we can observe their shape isn't related to the direction the last square was placed (as can be seen by comparing the subgraphs of *a*) and *d*) which had their last square placed to the right, or by comparing the subgraphs of *b*) and *c*) which had their last square placed down), but to the notion of how they locally changed direction.

Definition 2.1.2. Let PC_n be a general polyomino chain with instruction sequence $(R, R, I_3, \dots, I_n)$; $I_n \in \{R, L, D, U\}$. We say the last square of the chain *stayed in the same direction* if $I_n = I_{n-1}$ and that it *changed the direction* of the chain otherwise.

Analyzing the structure of the distinct neighborhoods of impact by means of the idea of change of direction, we can see that in both polyomino chains *a*) and *c*) in which the last square stayed in the same direction, their subgraphs were isomorphic; while in both polyomino chains *b*) and *d*) in which the last square changed the direction, their subgraphs were isomorphic. It is based

on this observation, we now assign an specific notation based on the notion of change in order to distinguish the four possible neighborhoods.

Notation 2.1.3. Let S denote a square stayed in the same direction and C that it changed, we shall refer to S and C as *states* and write E_1E_2 ; $E_1, E_2 \in \{S, C\}$ to denote a certain square is at state E_2 and the previous square is at state E_1 , thus giving us a way to describe a transition between states.

Under this notation we have written in the caption of Figure 2.2 the corresponding transitions for the last squares in each of the four graphs shown; where for example in $b)$, the transition corresponding to the last square is SC since it changed direction and the previous square maintained it.

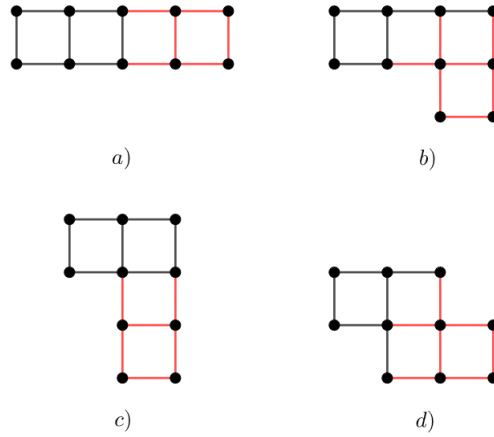


Figure 2.2: The neighborhoods of impact corresponding to transitions SS , SC , CS and CC respectively.

Now that we have defined notation for the different neighborhoods of impact that arise when adding a new square, we will proceed to specifically calculate how each of these distinct additions modify the value of the topological index; with the intention to later come up with an equality of the form $TI_f(PC_n) = TI_f(PC_{n-1}) +$ (the alteration caused by a transition produced by adding the last square). Below we show these modification values for transitions SS , SC , CS and CC .

$$\begin{aligned}
 SS : & -f(2,3) - f(2,3) - f(2,2) \\
 & + f(3,3) + f(3,3) + f(3,3) \\
 & + f(2,3) + f(2,3) + f(2,2) \\
 & = 3f(3,3) \\
 SC : & -f(3,3) - f(2,3) - f(2,2) - f(3,3) \\
 & + f(3,4) + f(3,4) + f(2,3) + f(3,4) \\
 & + f(2,4) + f(2,3) + f(2,2) \\
 & = 3f(3,4) + f(2,4) + f(2,3) - 2f(3,3) \\
 CS : & -f(2,4) - f(2,3) - f(2,2) \\
 & + f(3,4) + f(3,3) + f(3,3) \\
 & + f(2,3) + f(2,3) + f(2,2) \\
 & = f(3,4) - f(2,4) + f(2,3) + 2f(3,3) \\
 CC : & -f(3,4) - f(2,3) - f(2,3) - f(2,2) \\
 & + f(4,4) + f(2,4) + f(3,4) + f(2,3) \\
 & + f(2,4) + f(2,3) + f(2,2) \\
 & = f(4,4) + 2f(2,4).
 \end{aligned}$$

Remark 2.1.4. Even though we stated in general the neighborhoods of impact shown in Figure 2.2 where all of the possible neighborhoods, this is only true if the polyomino chain has more than three squares. Otherwise, special cases appear due to the fact there isn't enough space for the neighborhoods to completely develop as they would were there at least four squares. In Figure

2.3 we show the only three special cases, which we name special transitions and denote as T , SS^* and SC^* . Notice in particular T corresponds to the transition of a polyomino having zero squares passing to one, which is why its notation has only one letter. Also, we do not regard T as being a state and we consider the first square to have preserved the direction of the chain, that is, being in state S .

Below we give the respective modifications to the topological index value caused by the special transitions.

$$T : 4f(2, 2)$$

$$\begin{aligned} SS^* : & -f(2, 2) - f(2, 2) - f(2, 2) \\ & + f(2, 3) + f(2, 3) + f(3, 3) \\ & + f(2, 3) + f(2, 3) + f(2, 2) \\ = & f(3, 3) + 4f(2, 3) - 2f(2, 2) \end{aligned}$$

$$\begin{aligned} SC^* : & -f(2, 3) - f(3, 3) - f(2, 2) - f(2, 3) \\ & + f(2, 4) + f(3, 4) + f(2, 3) + f(3, 4) \\ & + f(2, 4) + f(2, 2) + f(2, 3) \\ = & 2f(2, 4) + 2f(3, 4) - f(3, 3). \end{aligned}$$

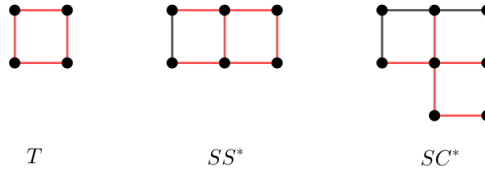


Figure 2.3: The special transitions T , SS^* and SC^* .

Finally, through the use of transitions, we are ready to introduce a new encoding for a given polyomino chain, where the main focus will now be in how the chain locally changes.

Definition 2.1.5. We will refer to the set $\{SS, SC, CS, CC\}$ as *the set of actions* of polyomino chains and to the set $\{T, SS^*, SC^*\}$ as *the set of special actions* of polyomino chains; where in particular, we will refer to the elements of any of the two sets as *actions* and specifically to those in the set of special actions as *special actions*.

Let PC_n be a polyomino chain with n squares and instruction sequence $(R, R, I_3, \dots, I_n)$; $I_k \in \{R, D\}$. We will define the *action sequence of the polyomino chain PC_n* to be a sequence $(a_1, a_2, a_3, \dots, a_n)$ where a_k is an action and satisfies the following:

- $a_1 = T, a_2 = SS^*$.
- if $k = 3$ and $I_3 = D$, then $a_3 = SC^*$.
- Otherwise:

$$\begin{aligned} & \text{if } I_{k-2} = I_{k-1} \text{ and } I_{k-1} = I_k; & a_k = SS \\ & \text{if } I_{k-2} = I_{k-1} \text{ and } I_{k-1} \neq I_k; & a_k = SC \\ & \text{if } I_{k-2} \neq I_{k-1} \text{ and } I_{k-1} = I_k; & a_k = CS \\ & \text{if } I_{k-2} \neq I_{k-1} \text{ and } I_{k-1} \neq I_k; & a_k = CC. \end{aligned}$$

Also, if $(a_1, a_2, a_3, \dots, a_n)$ is the action sequence associated to PC_n , we can refer to PC_n as $PC(a_1, a_2, a_3, \dots, a_n)$.

Remark 2.1.6. For example, the action sequence corresponding to the instruction sequence $(R, R, D, D, D, R, D, D, R, R)$ is $(T, SS^*, SC^*, CS, SS, SC, CC, CS, SC, CS)$. It is also worth mentioning that in terms of actions the linear and zig-zag polyomino chains are expressed as follows: $Li_n = PC(T, SS^*, SS, SS, \dots, SS)$ and $Z_n = PC(T, SS^*, SC^*, CC, CC, \dots, CC)$.

2.1.2. Recursive Construction and Connection with Links

Based on the discussion of Section 2.1.1, we can now define a real valued function g_f with domain equal to the set of all actions, such that it assigns them their effect on the TI_f function. As a consequence, we can immediately obtain the following lemma.

Lemma 2.1.7. *Let TI_f be a degree-based index and let $(T, SS^*, a_3, \dots, a_n)$ be the action sequence of the polyomino chain PC_n . Then*

$$TI_f(PC(T, SS^*, a_3, \dots, a_n)) = TI_f(PC(T, SS^*, a_3, \dots, a_{n-1})) + g_f(a_n),$$

where

$$\begin{aligned} g_f(T) &= 4f(2, 2) \\ g_f(SS^*) &= f(3, 3) + 4f(2, 3) - 2f(2, 2) & g_f(SC^*) &= 2f(2, 4) + 2f(3, 4) - f(3, 3) \\ g_f(SS) &= 3f(3, 3) & g_f(SC) &= 3f(3, 4) + f(2, 4) + f(2, 3) - 2f(3, 3) \\ g_f(CS) &= f(3, 4) - f(2, 4) + f(2, 3) + 2f(3, 3) & g_f(CC) &= f(4, 4) + 2f(2, 4). \end{aligned}$$

Remark 2.1.8. Observe that through the use of Lemma 2.1.7, it is not only possible to easily compute the value of the degree-based topological index with linear complexity, but it is also possible to compute it in constant time if the polyomino was described by the amount of each different action in its action sequence.

$$\begin{aligned} TI_f(PC(T, SS^*, a_3, \dots, a_n)) &= g_f(T) + g_f(SS^*) + g_f(a_3) + \dots + g_f(a_n) \\ &= n_1 g_f(T) + n_2 g_f(SS^*) + n_3 g_f(SC^*) + n_4 g_f(SS) \\ &\quad + n_5 g_f(SC) + n_6 g_f(CS) + n_7 g_f(CC). \end{aligned}$$

Working a little bit further into the use of Lemma 2.1.7, it is important to notice it makes explicit mention to a polyomino chain PC_n in order to obtain its action sequence. However, due to the fact we are working with polyomino chains (that is, the restricted case), it is possible to establish a sense of action sequence, independent of previously having a polyomino chain, such that action sequences always correspond to actual polyomino chains in the same way Lemma 1.2.5 established the same result for sequences of instructions.

Definition 2.1.9. We will say a sequence of the form $(a_1, a_2, a_3, \dots, a_n)$ is an *action sequence* with n squares if a_k is an action and the sequence satisfies:

- $a_1 = T, a_2 = SS^*$ and $a_3 \in \{SS, SC^*\}$
- For $k \geq 4$, $a_k \in \{SS, SC, CS, CC\}$ and if $a_k = E_1^k E_2^k$, then $E_2^{k-1} = E_1^k$.

Remark 2.1.10. It is straightforward from Definition 2.1.9 to deduce that an action sequence always corresponds to the action sequence of a polyomino chain. To do so, it is sufficient to translate the action sequence into an instruction sequence by performing a reverse process to the one described in Definition 2.1.5 and apply Lemma 1.2.5. Moreover, by the uniqueness of the translation into instructions, it follows different action sequences yield different instruction sequences.

Observing that action sequences of polyomino chains in particular satisfy the conditions of Definition 2.1.9 and taking into account Remark 2.1.10, we can apply Lemma 2.1.7 to action sequences in general. We shall then write $TI_f((a_1, a_2, a_3, \dots, a_n))$ in place of $TI_f(PC(a_1, a_2, a_3, \dots, a_n))$, which is relevant to our study since we will later design the dynamic programming framework around optimizing over the set of all action sequences, with no reference to polyomino chains.

Definition 2.1.11. Let $(a_1, a_2, a_3, \dots, a_n)$ be an action sequence with n squares. If $TI_f((a_1, a_2, a_3, \dots, a_n))$ has maximum value over the set of all possible action sequences with n squares, we'll say the action sequence $(a_1, a_2, a_3, \dots, a_n)$ is extremal.

It is important to highlight at this point, that our current encoding of polyomino chains through actions, although made with an emphasis on locality, isn't far away from extensions of other geometric-based representations, such as encodings using links. As a matter of fact, in [27] we can find a similar result to Lemma 2.1.7 using pairs of links in order to be able to denote what our actions retrieved as local changes. Moreover, in this restricted setting, sequences of actions, in the same way as geometric-based representations, still uniquely determine a polyomino chain, which means locality in restricted polyomino chains still encompasses the whole geometric information. However, as we'll come to realize in Sections 3.1.1 and 3.1.2, the generalization of actions to the general polyomino chain setting will no longer be equivalent to immediate descriptions using links, which will be a consequence of sequences of actions representing multiple chains as a result of the loss of geometric information through local descriptions.

2.1.3. Independence of f

Until this point, the function f through which the topological degree-based index was defined, has been a key element in our study. Nevertheless, now that we have the function g_f , we wonder how freely we could define its values without the need of explicitly having f . What this means, is that we want to know if giving arbitrary values to the function g_f allows us to recover a function f which is symmetric. Notice that the importance of such a result is that it would enable us to interpret the values of g_f through the actions to which they correspond and as an immediate consequence, it would allow us to design counterexamples based on desired behaviors.

So first of all, before examining g_f , it is important to notice since we are working with polyomino chains (even if it were the general case), the degrees of vertices cannot be less than 2 or greater than 4. Thus, it is only necessary to define f for the pairs of values shown in Figure 2.4, where in fact only half of them can be freely chosen, since the symmetry of f accounts for the other half of the values. As a convention, we'll choose to freely define $f(2, 2), f(2, 3), f(2, 4), f(3, 3), f(3, 4)$ and $f(4, 4)$.

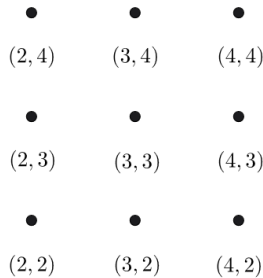


Figure 2.4: The pairs of values where f must be defined.

Returning to our main question, it is sufficient to show that by defining g_f we can retrieve the necessary values for f . In order to do so, we establish the system (2.1), where we have eliminated the

row corresponding to action SC^* since Remark 2.1.12 shows it is linearly dependent of the others, and where the columns correspond to $f(2, 2)$, $f(2, 3)$, $f(2, 4)$, $f(3, 3)$, $f(3, 4)$ and $f(4, 4)$ respectively.

Remark 2.1.12. We have we can write the effect of action SC^* in terms of the others, where we specifically choose to write this action through the others since it can only be used once in an action sequence.

$$g_f(SC^*) = g_f(SS) + g_f(SC) - g_f(CS)$$

$$\left[\begin{array}{cccccc|c} 0 & 0 & 0 & 3 & 0 & 0 & g_f(SS) \\ 0 & 1 & 1 & -2 & 3 & 0 & g_f(SC) \\ 0 & 1 & -1 & 2 & 1 & 0 & g_f(CS) \\ 0 & 0 & 2 & 0 & 0 & 1 & g_f(CC) \\ -2 & 4 & 0 & 1 & 0 & 0 & g_f(SS^*) \\ 4 & 0 & 0 & 0 & 0 & 0 & g_f(T) \end{array} \right] \quad (2.1)$$

Calculating the determinant of (2.1), we obtain it is none-zero, so solving the system yields the desired values to define f :

$$\begin{aligned} f(2, 2) &= \frac{g_f(T)}{4} \\ f(2, 3) &= \frac{g_f(SS^*)}{4} + \frac{g_f(T)}{8} - \frac{g_f(SS)}{12} \\ f(2, 4) &= \frac{g_f(SC)}{4} + \frac{5g_f(SS)}{8} + \frac{g_f(SS^*)}{8} + \frac{g_f(T)}{16} - \frac{3g_f(CS)}{4} \\ f(3, 3) &= \frac{g_f(SS)}{3} \\ f(3, 4) &= \frac{g_f(CS)}{4} + \frac{g_f(SC)}{4} + \frac{g_f(SS)}{24} - \frac{g_f(SS^*)}{8} - \frac{g_f(T)}{16} \\ f(4, 4) &= g_f(CC) + \frac{3g_f(CS)}{2} - \frac{g_f(SC)}{2} - \frac{5g_f(SS)}{4} - \frac{g_f(SS^*)}{4} - \frac{g_f(T)}{8}. \end{aligned}$$

2.2. Generating One Extremal Polyomino Chain

2.2.1. Is the Problem Greedy?

Now that we know by means of Remark 2.1.10 that working in the set of all action sequences is the same as working in the set of all polyomino chains, we wonder if optimizing the value of the topological degree-based index over the set of all action sequences follows a greedy strategy. That is, can an extremal action sequence be found by successively choosing the action that provides the immediate maximum value?

Here, in order to build a counterexample, sustaining ourselves on the results of Section 2.1.3, we'll conveniently define g_f as follows:

$$g_f(T) = 0, g_f(SS^*) = 1, g_f(SS) = 2, g_f(SC) = 3, g_f(CS) = 4, g_f(CC) = 100$$

where in particular, as a consequence of Remark 2.1.12, we have $g_f(SC^*) = 1$. Notice that the reason we define g_f in this way, is to make the decision of an immediate change of direction (represented by action SC^*) unattractive; while on the other hand making two consecutive changes of direction the best possible.

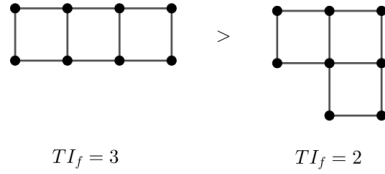


Figure 2.5: The polyomino chain with 3 squares and maximum TI_f value.

Calculating, by the use of Lemma 2.1.7, the TI_f value for the only two action sequences with three squares we have $TI_f(PC(T, SS^*, SS)) > TI_f(PC(T, SS^*, SC^*))$, as shown in Figure 2.5.

Assuming optimization of action sequences followed a greedy strategy, we would have that an extremal action sequence with four squares must be of the form (T, SS^*, SS, a_4) . Again, by calculating the TI_f value for the only two possible action sequences with that form, which are shown in Figure 2.6, we would conclude an action sequence with four squares and maximum TI_f value is (T, SS^*, SS, SC) . However, as shown in Figure 2.7, the action sequence (T, SS^*, SC^*, CC) has a bigger TI_f value, which shows optimization over action sequences does not follow a greedy approach.

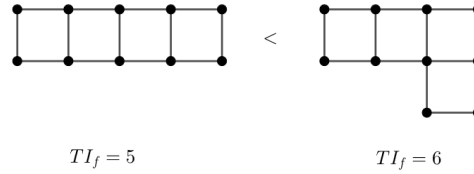


Figure 2.6: The polyomino chain with 4 squares and maximum TI_f value assuming the first three squares must correspond to the action sequence (T, SS^*, SS) .

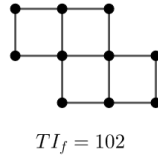


Figure 2.7: The actual polyomino chain with 4 squares and maximum TI_f value.

Also, even if the greedy approach did, by some peculiarity of the values of g_f , find an extremal action sequence, it would be unable to find any others. Take for example g_f as defined above, with the exception now $g_f(CC) = 1$. Then all of the above reasoning would still apply and the greedy approach would find the action sequence (T, SS^*, SS, SC) , which is in fact an extremal action sequence. However, the other extremal sequence (T, SS^*, SC^*, CS) would not be taken into account. In Figure 2.8 we show all polyomino chains with four squares with their respective TI_f values using the new g_f .

2.2.2. Dynamic Programming Approach to Build an Extremal Polyomino Chain

In Section 2.2.1 we observed, since optimization did not follow a greedy approach, that extremal polyomino chains with n squares could be built from a polyomino chain not extremal for $n - 1$

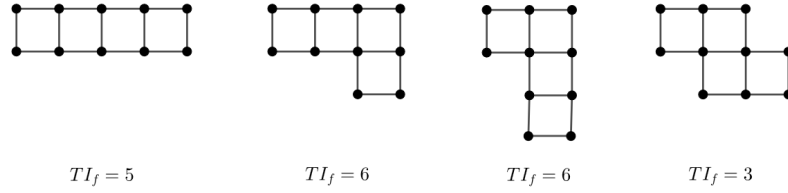


Figure 2.8: Multiple extremal polyomino chains.

squares. To overcome that difficulty, what we'll do in this section is build a maximum from all of the best routes possible exhaustively searching by using dynamic programming.

Dynamic programming is a classical optimization technique that solves complex problems by breaking them into overlapping subproblems that exhibit an optimal substructure property [16, 8]. More formally, a dynamic programming approach consists of a set of well-defined problems, a set of decisions or transitions and a function h ; where the intention is to establish the structure of a directed graph whose vertices correspond to the set of problems and the edges to the set of decisions, such that solving a problem is equivalent to solving the problems to which it has an edge and combining their results through h . In Figure 2.9 we illustrate a graph that could correspond to a dynamic programming framework, where the goal is to solve the problem marked in red.

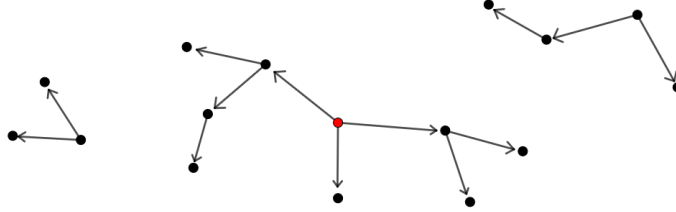


Figure 2.9: Example of a dynamic programming framework.

Given a degree-based index TI_f , we define $M_f(n, E)$ for $E = S, C$ and $n \geq 1$ as the *maximum value* of this index among all action sequences with n squares ending at state E ; that is, $a_n = E_1^n E_2^n$ with $E_2^n = E$. We shall refer to the values $M_f(n, E)$ as *problems* and to the action sequences that produce them as *maximal action sequences*. Motivated by the recursive nature of such indices in polyomino chains, the following result provides a dynamic programming formulation for computing $M_f(n, E)$ for all $n \geq 1$ and $E \in \{S, C\}$; where we'll employ actions as the natural transitions between problems and Lemma 2.1.7 as the function that combines the solutions of subproblems.

Remark 2.2.1. If there does not exist an action sequence with n squares ending at state E , we will define $M_f(n, E) = -\infty$, which notice happens only for problems $M_f(1, C)$ and $M_f(2, C)$.

Theorem 2.2.2. *Let TI_f be a degree-based index. Then for $n \leq 3$,*

$$\begin{array}{ll} M(1, S) = g_f(T) & M(1, C) = -\infty \\ M(2, S) = M(1, S) + g_f(SS^*) & M(2, C) = -\infty \\ M(3, S) = M(2, S) + g_f(SS) & M(3, C) = M(2, S) + g_f(SC^*). \end{array}$$

And for $n \geq 4$,

$$\begin{array}{l} M(n, S) = \max\{M(n-1, S) + g_f(SS), M(n-1, C) + g_f(CS)\} \\ M(n, C) = \max\{M(n-1, S) + g_f(SC), M(n-1, C) + g_f(CC)\}. \end{array}$$

Proof. The expressions for $n \leq 3$ are straightforward to check, so we'll only prove the expression for problem $M(n, S)$; $n \geq 4$. The proof for the expression of $M(n, C)$ is completely analogous.

By definition of $M_f(n-1, S)$ and $M_f(n-1, C)$, there exist at least one action sequence $(T, SS^*, \dots, a_{n-1})$ ending at state S and another $(T, SS^*, \dots, b_{n-1})$ ending at state C , such that

$$\begin{aligned} M_f(n-1, S) &= TI_f((T, SS^*, \dots, a_{n-1})) \\ M_f(n-1, C) &= TI_f((T, SS^*, \dots, b_{n-1})) \end{aligned}$$

Hence, by Lemma 2.1.7, we obtain

$$M_f(n-1, S) + g_f(SS) = TI_f((T, SS^*, \dots, a_{n-1}, SS))$$

and

$$M_f(n-1, C) + g_f(CS) = TI_f((T, SS^*, \dots, b_{n-1}, CS)).$$

Therefore, by the definition of $M_f(n, S)$, it follows

$$\max\{M_f(n-1, S) + g_f(SS), M_f(n-1, C) + g_f(CS)\} \leq M_f(n, S).$$

Conversely, there exists at least one action sequence $(T, SS^*, \dots, a_n)$ ending at state S such that

$$M_f(n, S) = TI_f((T, SS^*, \dots, a_n)).$$

Thus, by Lemma 2.1.7,

$$\begin{aligned} M_f(n, S) &= TI_f((T, SS^*, \dots, a_{n-1})) + g_f(E_1^n S) \\ &\leq M_f(n-1, E_2^{n-1}) + g_f(E_1^n S) \\ &= M_f(n-1, E_1^n) + g_f(E_1^n S) \\ &\leq \max\{M_f(n-1, S) + g_f(SS), M_f(n-1, C) + g_f(CS)\}. \end{aligned}$$

This concludes the proof. □

Now, observe that, due to the linearity of the degree-based indices (see Definition 1.1.1), the identity $TI_{-f}(G) = -TI_f(G)$ holds. This implies that minimizing TI_f is equivalent to maximizing TI_{-f} . Consequently, as a direct application of Theorem 2.2.2, we obtain a method for computing the *minimum value* of any degree-based index among all action sequences with n squares and ending at state E , denoted by $m_f(n, E)$, with $n \geq 1$ and $E \in \{S, C\}$.

Corollary 2.2.3. *Let TI_f be a degree-based index. Then, for $n \geq 1$ and $E \in \{S, C\}$*

$$m_f(n, E) = -M_{-f}(n, E).$$

According to Corollary 2.2.3, we may restrict our attention to the maximization problem. Moreover, it is important to notice that the method introduced in Theorem 2.2.2 not only allows us to determine the maximum value of a fixed degree-based index over all action sequences with n squares and ending at state E , but also provides a constructive way to generate at least one maximal action sequence for problem $M_f(n, E)$ by using the underlying directed graph implicit in the dynamic programming framework. This construction can be performed recursively by selecting each action a_k for $k \in \{n, \dots, 4\}$, according to the following rules:

- If we are at problem $M_f(k, S)$ and $M_f(k-1, S) + g_f(SS) \geq M_f(k-1, C) + g_f(CS)$, then assign $a_k = SS$ and move to problem $M_f(k-1, S)$. Otherwise, assign $a_k = CS$ and move to problem $M_f(k-1, C)$.

- If we are at problem $M_f(k, C)$ and $M_f(k-1, C) + g_f(CC) \geq M_f(k-1, S) + g_f(SC)$, then assign $a_k = CC$ and move to problem $M_f(k-1, C)$. Otherwise, assign $a_k = SC$ and move to problem $M_f(k-1, S)$.

Finally, if we are at problem $M_f(3, S)$ assign $a_1 = T, a_2 = SS^*, a_3 = SS$ or $a_1 = T, a_2 = SS^*, a_3 = SC^*$ otherwise. Hereafter, we refer to this procedure as the *constructive method* induced by the problem $M_f(n, E)$.

As an example, in Figure 2.10 we display a possible graph induced by applying Theorem 2.2.2, where each cell denotes a problem $M_f(n, E)$ and each arrow depicts the subproblem it was built from in the recursive process. For instance, performing the constructive method for problem $M_f(5, S)$ we would obtain the action sequence (T, SS^*, SC^*, CC, CS) .

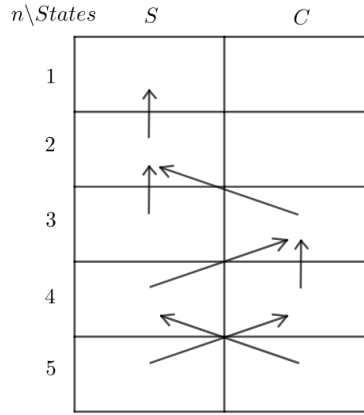


Figure 2.10: An example of a graph induced by performing Theorem 2.2.2.

At this point, it is worth emphasizing that our main objective is to determine the maximum value of a degree-based index among all polyomino chains with a fixed number of squares, say n . Therefore, the quantity of real interest is

$$M_f(n) := \max\{M_f(n, S), M_f(n, C)\}.$$

In any case, the problem reduces to analyzing $M_f(n, E)$, and thus all the previous discussion remains fully applicable.

Remark 2.2.4. Notice by applying Theorem 2.2.2 and the constructive method we can find a maximal action sequence with n squares and ending at state S or C in linear time with respect to n . Consequently, we can also find an extremal action sequence with n squares by taking a maximal action sequence of either $M_f(n, S)$ or $M_f(n, C)$ depending on which has a higher value.

2.2.3. Cyclic Behavior of Extremal Polyomino Chains

Now, thanks to Theorem 2.2.2 and the constructive method of problems $M_f(n, E)$, we are capable of not only computing the topological degree-based index for extremal polyomino chains but also retrieving one such chain. However, because both of these procedures are of a recursive nature, we face an important problem when trying to characterize the extremal polyomino chains corresponding to a given index. Since it is not possible to compute $M_f(n, E)$ for all n and all indices potentially behave differently, in this section we give results concerning the appearance of *cyclic patterns* in the construction of problems $M_f(n, E)$.

The key observation for this process is to notice that for the calculation of the expressions $\max\{M_f(n-1, S) + g_f(SE), M_f(n-1, C) + g_f(CE)\}$ it is not necessary to recall the values $M_f(n-1, S)$ and $M_f(n-1, C)$ but only their difference. For example,

$$\begin{aligned} M_f(n-1, S) + g_f(SE) &\geq M_f(n-1, C) + g_f(CE) \\ \Rightarrow M_f(n-1, S) - M_f(n-1, C) &\geq g_f(CE) - g_f(SE). \end{aligned}$$

In other words, if we know the advantage one problem has over the other, it is possible to know which expression in the $\max$ argument is maximum. Based on this, we search for conditions over advantages in order to establish patterns in the decisions behind the construction of the values of problems $M_f(n, E)$.

Definition 2.2.5. Let $n \geq 3$. If $M_f(n, S) \geq M_f(n, C)$, we define *the advantage of problem* $M_f(n, S)$, denoted by $V(n, S)$ as $M_f(n, S) - M_f(n, C)$ and *the advantage of problem* $M_f(n, C)$, denoted by $V(n, C)$ as 0. Similarly, if $M_f(n, C) \geq M_f(n, S)$, we define $V(n, S) = 0$ and $V(n, C) = M_f(n, C) - M_f(n, S)$.

In the case of $n = 1, 2$ we define as special cases $V(1, S) = M_f(1, S)$, $V(1, C) = -\infty$ and $V(2, S) = M_f(2, S)$, $V(2, C) = -\infty$.

Also, we'll denote by $x \in \mathbb{R}$ *the base value of problems* $M_f(n, S)$ and $M_f(n, C)$, which is defined as $x = M_f(n, S) - V(n, S) = M_f(n, C) - V(n, C)$.

Lemma 2.2.6. Let $n \geq 1$ and suppose for some $1 \leq j < n$ we have $V(n, S) = V(j, S)$ and $V(n, C) = V(j, C)$. If $M_f(j, H_1) + g_f(b_1) \geq M_f(j, H_2) + g_f(b_2)$; where either $H_1 = S, H_2 = C$ or $H_1 = C, H_2 = S$ and b_1, b_2 are actions coming from states H_1 and H_2 respectively, then $M_f(n, H_1) + g_f(b_1) \geq M_f(n, H_2) + g_f(b_2)$.

Proof. So suppose $M_f(j, H_1) + g_f(b_1) \geq M_f(j, H_2) + g_f(b_2)$. Then, because $M_f(j, H_1)$ and $M_f(j, H_2)$ are problems for the same number of squares, by the definition of advantage, for $x^* \in \mathbb{R}$ we can write them as:

$$M_f(j, H_1) = x^* + V(j, H_1), \quad M_f(j, H_2) = x^* + V(j, H_2).$$

From this it follows:

$$\begin{aligned} M_f(j, H_1) + g_f(b_1) &\geq M_f(j, H_2) + g_f(b_2) \\ \Rightarrow M_f(j, H_1) - M_f(j, H_2) &\geq g_f(b_2) - g_f(b_1) \\ \Rightarrow V(j, H_1) - V(j, H_2) &\geq g_f(b_2) - g_f(b_1) \end{aligned}$$

But then, since by hypothesis we know the advantages for the problems with j squares equal those of the problems with n squares, we have

$$V(n, H_1) - V(n, H_2) \geq g_f(b_2) - g_f(b_1),$$

and because by the definition of advantage, for $x \in \mathbb{R}$ we can write

$$M_f(n, H_1) = x + V(n, H_1), \quad M_f(n, H_2) = x + V(n, H_2),$$

we can conclude $M_f(n, H_1) + g_f(b_1) \geq M_f(n, H_2) + g_f(b_2)$. □

Lemma 2.2.7. Let $n \geq 1$ and suppose for some $1 \leq j < n$, $V(n, S) = V(j, S)$ and $V(n, C) = V(j, C)$. If problems $M_f(n+1, E)$ and $M_f(j+1, E)$ are of the form $\max\{M_f(i-1, S) + g_f(SE), M_f(i-1, C) + g_f(CE)\}$, where $i \in \{n+1, j+1\}$, $E \in \{S, C\}$ and we have $M_f(j+1, E) = M_f(j, H) + g_f(HE)$ where H is a state, then $M_f(n+1, E) = M_f(n, H) + g_f(HE)$.

Proof. Because $M_f(j+1, E) = \max\{M_f(j, S) + g_f(SE), M_f(j, C) + g_f(CE)\} = M_f(j, H) + g_f(HE)$, it follows $M_f(j, H) + g_f(HE) \geq M_f(j, H') + g_f(H'E)$ where H' is the remaining state. But then, by Lemma 2.2.6 we have:

$$M_f(n, H) + g_f(HE) \geq M_f(n, H') + g_f(H'E),$$

and since the problem $M_f(n+1, E)$ is of the form $\max\{M_f(n, S) + g_f(SE), M_f(n, C) + g_f(CE)\}$, this implies $M_f(n+1, E) = M_f(n, H) + g_f(HE)$. $\square$

Remark 2.2.8. It is important to highlight, that in Lemma 2.2.6 we left actions b_1 and b_2 as general as possible because we will later apply this lemma in a more bigger context in Section 3.3.2. Also, in Lemma 2.2.7 we make the now redundant statement that problems $M_f(n+1, E)$ and $M_f(j+1, E)$ are of the form $\max\{M_f(i-1, S) + g_f(SE), M_f(i-1, C) + g_f(CE)\}$. However, in Section 3.3.2 we will also apply this lemma in a more general setting, in which problems $M_f(n, E)$ need not be of that form.

Lemma 2.2.9. *Let $n \geq 1$ and suppose for some $1 \leq j < n$, $V(n, S) = V(j, S)$ and $V(n, C) = V(j, C)$. If problems $M_f(n+1, S)$, $M_f(n+1, C)$, $M_f(j+1, S)$ and $M_f(j+1, C)$ are of the form $\max\{M_f(i-1, S) + g_f(SE), M_f(i-1, C) + g_f(CE)\}$, where $i \in \{n+1, j+1\}$, $E \in \{S, C\}$, then $V(n+1, S) = V(j+1, S)$ and $V(n+1, C) = V(j+1, C)$.*

Proof. Suppose we had

$$M_f(j+1, S) = M_f(j, H_1) + g_f(H_1S) \quad \text{and} \quad M_f(j+1, C) = M_f(j, H_2) + g_f(H_2C),$$

where $H_1, H_2 \in \{S, C\}$. Then by the definition of advantage, for $x^* \in \mathbb{R}$ we can also write them as:

$$\begin{aligned} M_f(j+1, S) &= x^* + V(j, H_1) + g_f(H_1S), \\ M_f(j+1, C) &= x^* + V(j, H_2) + g_f(H_2C). \end{aligned}$$

Now, assuming without loss of generality that $M_f(j+1, S) \geq M_f(j+1, C)$, we have $V(j+1, S) = V(j, H_1) - V(j, H_2) + g_f(H_1S) - g_f(H_2C)$ and $V(j+1, C) = 0$.

On the other hand, by Theorem 2.2.7, it follows

$$M_f(n+1, S) = M_f(n, H_1) + g_f(H_1S) \quad \text{and} \quad M_f(n+1, C) = M_f(n, H_2) + g_f(H_2C)$$

and therefore, for $x \in \mathbb{R}$ we can also write them as:

$$\begin{aligned} M_f(n+1, S) &= x + V(n, H_1) + g_f(H_1S) \\ M_f(n+1, C) &= x + V(n, H_2) + g_f(H_2C), \end{aligned}$$

which by hypothesis yields

$$\begin{aligned} M_f(n+1, S) &= x + V(j, H_1) + g_f(H_1S) \\ M_f(n+1, C) &= x + V(j, H_2) + g_f(H_2C). \end{aligned}$$

It would only remain to show $M_f(n+1, S) \geq M_f(n+1, C)$ but this is immediate from the fact $V(j, H_1) + g_f(H_1S) \geq V(j, H_2) + g_f(H_2C)$. $\square$

Corollary 2.2.10. *Let $n \geq 1$ and suppose for some $1 \leq j < n$, $V(n, S) = V(j, S)$ and $V(n, C) = V(j, C)$. If problems $M_f(n+1, S)$, $M_f(n+1, C)$, $M_f(j+1, S)$ and $M_f(j+1, C)$ are of the form $\max\{M_f(i-1, S) + g_f(SE), M_f(i-1, C) + g_f(CE)\}$, where $i \in \{n+1, j+1\}$, $E \in \{S, C\}$ and $M_f(j+1) = M_f(j+1, E)$, then $M_f(n+1) = M_f(n+1, E)$.*

Finally, as a consequence of all the results in this section, we give conditions such that the computations of problems $M_f(n, E)$ and the actions chosen in the constructive method enter into a cyclic pattern.

Theorem 2.2.11. *Let $n \geq 1$ be the first integer such that there exists $1 \leq j < n$ where $V(n, S) = V(j, S)$, $V(n, C) = V(j, C)$ and problems $M_f(i, E)$ with $i \geq j + 1$ are of the form $\max\{M_f(i - 1, S) + g_f(SE), M_f(i - 1, C) + g_f(CE)\}$; $E \in \{S, C\}$. For $i \geq n + 1$ let $k = j + [i - n \bmod (n - j)]$ if $i - n \bmod (n - j) \neq 0$ and $k = n$ otherwise. If we have $M_f(k, E) = M_f(k - 1, H) + g_f(HE)$ with $H \in \{S, C\}$, then $M_f(i, E) = M_f(i - 1, H) + g_f(HE)$ and $V(i, E) = V(k, E)$. Also, if $M_f(k) = M_f(k, E)$, then $M_f(i) = M_f(i, E)$.*

2.2.4. One Extremal Polyomino Chain of the Augmented Zagreb Index

Provided Theorem 2.2.2, the constructive method introduced in Section 2.2.2 and Theorem 2.2.11, we are now ready to analyze the extremal behavior of degree-based indices when restricted to polyomino chains. As an application we shall carry out the study on the Augmented Zagreb Index with the aim of addressing the unresolved problem of finding its maximum extremal chains.

The Augmented Zagreb Index (*AZI*) is a molecular structure descriptor defined by Furtula, Graovac, and Vukićević in [2], inspired by the success of the Atom-Bond Connectivity (*ABC*) index and which is based on the function

$$f(x, y) = \left(\frac{xy}{x + y - 2} \right)^3; x, y \in \{1, 2, \dots\}.$$

Overall, many results have confirmed that the *AZI* is a highly effective and reliable descriptor for molecular property prediction, having been shown to outperform the *ABC* index in predicting the heats of formation of heptane and octane isomers, and to exhibit the highest reported correlations with the standard heats of formation and boiling points of octane isomers across multiple studies [14]. Also, comparative analyses with other degree-based topological indices further highlight the remarkable structural sensitivity and smoothness of the *AZI*, particularly in the context of isomeric compounds [12].

However, although several extremal results concerning the *AZI* have been reported [2], the specific problem of identifying the polyomino chains with maximal *AZI* for a fixed number of squares was posed in [3] and, to the best of our knowledge, remains unresolved.

Before presenting a first result regarding the *AZI*, we introduce a definition concerning the description of polyomino chains and notation that will simplify the characterization of the extremal chains under consideration.

Definition 2.2.12. A square of a polyomino chain is called *terminal* if it has exactly one adjacent square and a *kink* if it has two adjacent squares and contains a vertex of degree 2. A *segment* is then defined as a maximal linear chain together with an adjacent kink or terminal square, where its length is defined as its number of squares and the segment is called *external* if it has at least one terminal square and *internal* otherwise.

Notation 2.2.13. We define a polyomino chain with $m \geq 2$ segments as an *augmented zigzag of type 1* if all its segments have length 3; this polyomino chain will be denoted by AZ_m^1 (see Figure 2.11 a)). Similarly, a polyomino chain with $m \geq 3$ segments is called an *augmented zigzag of type 2* if all its segments are of length 3, except for exactly one internal segment of length 2; this family will be denoted by AZ_m^2 and an specific member is denoted by $AZ_m^2(i)$ where i denotes the index of the segment of length 2 (see Figure 2.11 b)).

For simplicity, throughout this section, we use the notation g , $M(n, E)$, and $M(n)$ instead of g_f , $M_f(n, E)$, and $M_f(n)$, respectively. By carrying out base computations, we obtain $g(T) = 32$,

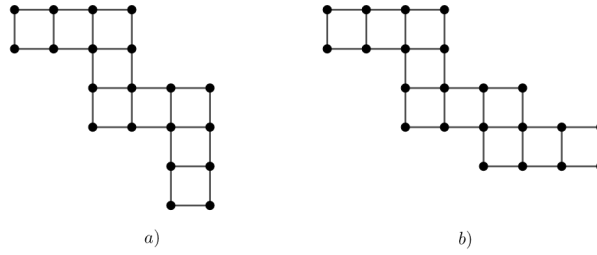


Figure 2.11: a) The polyomino chain AZ_4^1 , b) The polyomino chain $AZ_5^2(4)$.

$g(SS^*) \approx 27.39$, $g(SC^*) \approx 32.26$, $g(SS) \approx 34.17$, $g(SC) \approx 34.69$, $g(CS) \approx 36.6$, $g(CC) \approx 34.96$; which can serve the reader as a useful reference although the computations for the following proof need the use of the exact expressions.

Theorem 2.2.14. *For $n \geq 5$, polyomino chains that maximize the AZI are $AZ_{\frac{n-1}{2}}^1$ when n is odd and $AZ_{\frac{n}{2}}^2(\frac{n}{2} - 1)$ when n is even. Specifically,*

$$M(n) = \frac{4456}{125}n - \frac{26763}{2000} - \frac{2312}{3375}I_{\{n=2k\}}.$$

Proof. By Theorem 2.2.2 the values for $M(1)$ and $M(2)$ are clear. On the other hand, for $M(3)$ we need to compare the values obtained for $M(3, S)$ and $M(3, C)$ which equal $g(T) + g(SS^*) + g(SS)$ and $g(T) + g(SS^*) + g(SC^*)$ respectively. Through a straightforward algebraic manipulation we can conclude:

$$M(3, S) > M(3, C),$$

and thus $M(3) = M(3, S)$. Here it is important to say, for the following comparisons we will no longer remember the values of $M(3, S)$ or $M(3, C)$ but only their advantages, which in this case are $V(3, S) = g(SS) + g(SC^*)$ and $V(3, C) = 0$, since this will allow us to carry out the inequalities while only dragging a small representative term solely in terms of g and be able to spot the conditions in order to enter a cyclic pattern. To illustrate, we now compute $M(4)$ for which we must first obtain the values for $M(4, S)$ and $M(4, C)$. Observe that

$$M(3, C) + g(CS) > M(3, S) + g(SS), \quad \text{since } g(CS) > V(3, S) + g(SS)$$

and

$$M(3, S) + g(SC) > M(3, C) + g(CC), \quad \text{since } V(3, S) + g(SC) > g(CC)$$

From these inequalities, we can determine the values of $M(4, S)$ and $M(4, C)$ and by applying Remark 2.1.12, it follows that, $M(4) = M(4, S) = M(4, C)$, i.e $V(4, S) = V(4, C) = 0$.

For $n = 5$, the inequalities $g(SS) < g(CS)$ and $g(SC) < g(CC)$ imply that

$$M(5, S) = M(4, C) + g(CS) > M(4, S) + g(SS),$$

$$M(5, C) = M(4, C) + g(CC) > M(4, S) + g(SC).$$

Therefore,

$$M(5) = M(5, S) = M(4, C) + g(CS),$$

because $g(CS) > g(CC)$. Moreover, the advantages are given by $V(5, S) = g(CS) - g(CC)$ and $V(5, C) = 0$. Now, for $n = 6$, a similar argument yields:

$$M(6, S) = M(5, C) + g(CS) > M(5, S) + g(SS),$$

$$M(6, C) = M(5, S) + g(SC) > M(5, C) + g(CC).$$

Hence,

$$M(6) = M(6, S) = M(5, C) + g(CS),$$

and $V(6, S) = g(CC) - g(SC)$, $V(6, C) = 0$. Next, for $n = 7$, we observe that:

$$M(7, S) = M(6, C) + g(CS) > M(6, S) + g(SS), \quad \text{since } g(CC) + g(SS) < g(CS) + g(SC),$$

and

$$M(7, C) = M(6, S) + g(SC) = M(6, C) + g(CC),$$

that is, the computation of $M(7, C)$ yields a tie in the *max* argument. Following the rules given for the construction method, it then follows $M(7, C) = M(6, C) + g(CC)$. Consequently,

$$M(7) = M(7, S) = M(6, C) + g(CS),$$

since $g(CS) > g(CC)$, and $V(7, S) = g(CS) - g(CC)$, $V(7, C) = 0$. Before proceeding further, we summarize the key information obtained so far in Figure 2.12 which depicts the underlying graph of the dynamic programming approach. In it, we have included the values for the advantages $V(n, S)$ and $V(n, C)$, where notice we did not write function g in the expressions for the advantages in order to make the table more compact, although every action should be inside a $g()$ expression. Also, for visual simplicity, we have highlighted in red the cells corresponding to the problem with the highest index value, that is, the problem corresponding to $M(n)$, even though this could be retrieved by merely looking at the advantages' values.

According to the key information summarized in Figure 2.12, it is evident that for $n = 7$ and $j = 5$ we arrive at the conditions for Theorem 2.2.11, which will force problems for $n \geq 8$ to behave the same as those for $n = 6$ and $n = 7$. This indicates the onset of a cyclic pattern in the decision process, with a period of two. Therefore, based on the previous information and applying the constructive method for problem $M(n, S)$, we conclude the following:

When n is odd, the polyomino chain that maximizes the *AZI* index is defined by the action sequence

$$(T, SS^*, SS, \underbrace{SC, CS}_{\frac{n-3}{2} \text{ times}})$$

which corresponds to a polyomino chain composed of $\frac{n-1}{2}$ segments, each of length 3. In other words, it is the polyomino chain $AZ_{\frac{n-1}{2}}^1$.

On the other hand, when n is even, the polyomino chain that maximizes the *AZI* index is defined by the action sequence

$$(T, SS^*, SS, \underbrace{SC, CS}_{\frac{n-6}{2} \text{ times}}, SC, CC, CS)$$

which corresponds to a polyomino chain composed of $\frac{n}{2}$ segments, each of length 3 except for the penultimate which has length 2. In other words, it is the polyomino chain $AZ_{\frac{n}{2}}^2 (\frac{n}{2} - 1)$.

Finally, the value of $M(n)$ follows directly from Lemma 2.1.7.

□

Remark 2.2.15. For $n \leq 4$, based on Figure 2.12, we have the following are extremal action sequences:

- For $n = 1$: (T)

$n \backslash \text{States}$	S	C
1	<u>T</u>	$-\infty$
2	<u>$T + SS^*$</u>	$-\infty$
3	<u>$SS - SC^*$</u>	0
4	<u>0</u>	<u>0</u>
5	<u>$CS - CC$</u>	0
6	<u>$CC - SC$</u>	0
7	<u>$CS - CC$</u>	0

Figure 2.12: The dynamic programming table associated with Theorem 2.2.2 for the maximum AZI value.

- For $n = 2$: (T, SS^*)
- For $n = 3$: (T, SS^*, SS)
- For $n = 4$: Both (T, SS^*, SS, SS) and (T, SS^*, SS, SC)

2.3. Counting and Generating All Extremal Polyomino Chains

In Section 2.2.4 we managed to find at least one extremal polyomino chain for every n taking into account the AZI index. However, are these extremal polyomino chains found by Theorem 2.2.2 and the constructive method all of the extremal polyomino chains for the given index? Taking a closer look on the constructive method, we can see there exists a potential place where we lose information and that is when ties occur in the $\max$ arguments present in the computation of the $M_f(n, E)$ values, which can occur as it happened for problem $M(7, C)$ in Theorem 2.2.14. Thus, it becomes relevant to study what happens in the presence of ties and how do these relate to the set of maximal polyomino chains. In this section we address this problem and as an application we fully characterize all extremal polyomino chains for the AZI index.

From this point on, for a fixed integer $n \geq 1$ and $E \in \{S, C\}$, we denote by $n_t(n, E)$ the number of ties that occur during the recursive computation of $M_f(n, E)$. These ties represent situations where both arguments of the $\max$ expression have the same value, resulting in multiple equally optimal configurations. We define the base cases as: $n_t(i, E) = 0$ for all $i \in \{1, 2, 3\}$ and $E \in \{S, C\}$, while for $n \geq 4$, $n_t(n, E)$ is defined recursively as follows:

- a) If only one expression in the $\max$ argument of $M_f(n, E)$ is the maximum, then
 - if the maximum is an expression involving $M_f(n-1, S)$, $n_t(n, E) = n_t(n-1, S)$.

- if the maximum is an expression involving $M_f(n-1, C)$, $n_t(n, E) = n_t(n-1, C)$.

b) Else

$$n_t(n, E) = 1 + n_t(n-1, S) + n_t(n-1, C).$$

Remark 2.3.1. Notice by the way the values $n_t(n, E)$ are defined, we have established a new dynamic programming framework and along it a new directed graph which we shall call the *index tie graph*; where differently as in the graph associated with Theorem 2.2.2, a problem here can have multiple edges (which happens in the case $n_t(n, E) = 1 + n_t(n-1, S) + n_t(n-1, C)$).

Lemma 2.3.2. *Let $n \geq 3$, $E \in \{S, C\}$ and $(T, SS^*, a_3, \dots, a_n)$ be a maximal action sequence corresponding to problem $M_f(n, E)$. Then the following statements hold:*

a) *For all $k \in \{1, \dots, n\}$, it follows that*

$$M_f(k, E_2^k) = TI_f((T, SS^*, a_3, \dots, a_k)).$$

b) *There exists at least one distinct maximal action sequence $(T, SS^*, b_3, \dots, b_n)$ for problem $M_f(n, E)$ if and only if*

$$n_t(k+1, E_2^{k+1}) = 1 + n_t(k, S) + n_t(k, C),$$

for some $k \in \{3, \dots, n-1\}$.

c) *Given $k \in \{3, \dots, n\}$. The maximum number of consecutive actions, counting backward from the last one, that are shared by all maximal action sequences of the problem $M_f(n, E)$ is $n-k$ and these actions are precisely $a_{k+1}, \dots, a_{n-1}, a_n$ if and only if*

$$n_t(n, E) = n_t(n-1, E_2^{n-1}) = \dots = n_t(k, E_2^k) = 1 + n_t(k-1, S) + n_t(k-1, C).$$

Proof.

a) The first statement is immediate for the cases $k = 1, 2$. For $k \geq 3$, assume by way of contradiction that

$$TI_f((T, SS^*, a_3, \dots, a_k)) < M_f(k, E_2^k).$$

By the definition of $M_f(k, E_2^k)$, there exists at least an action sequence $(T, SS^*, b_3, \dots, b_k)$ such that

$$M_f(k, E_2^k) = TI_f((T, SS^*, b_3, \dots, b_k)).$$

Then, by Lemma 2.1.7, it follows that

$$TI_f((T, SS^*, a_3, \dots, a_k, a_{k+1}, \dots, a_n)) < TI_f((T, SS^*, a_3, \dots, b_k, a_{k+1}, \dots, a_n)),$$

which contradicts the maximality of the sequence $(T, SS^*, a_3, \dots, a_n)$.

b) Now, let us prove the forward implication of the second statement. Since the two action sequences are distinct, there must be at least one index at which they differ. Without loss of generality, assume that $k \in \{3, \dots, n\}$ is the first index (counting from right to left) such that $a_k \neq b_k$, that is, $a_j = b_j$ for all $j \geq k+1$. Notice this implies the state from which a_k comes from is different from the one b_k comes from, while the states they end at are equal.

Then, by Lemma 2.1.7, it follows that

$$TI_f((T, SS^*, a_3, \dots, a_k)) = TI_f((T, SS^*, b_3, \dots, b_k)).$$

Moreover, by the first statement of this lemma and because actions a_k and b_k end at the same state,

$$M_f(k, E_2^k) = TI_f((T, SS^*, a_3, \dots, a_k))$$

and

$$M_f(k, E_2^k) = TI_f((T, SS^*, b_3, \dots, b_k)).$$

Therefore, since we know the states from which a_k and b_k came from differ, we have:

$$M_f(k-1, S) + g_f(S, E_2^k) = M_f(k-1, C) + g_f(C, E_2^k).$$

which implies $n_t(k, E_2^k) = 1 + n_t(k-1, S) + n_t(k-1, C)$.

To prove the reverse implication, observe that,

$$M_f(n, E) = TI_f((T, SS^*, a_3, \dots, a_n)).$$

Then, applying the first statement and Lemma 2.1.7, we obtain

$$M_f(n, E) = M_f(k+1, E_2^{k+1}) + g_f(a_{k+2}) + \dots + g_f(a_n).$$

By our hypothesis, we then have

$$M_f(k, S) + g_f(SE_2^{k+1}) = M_f(k, C) + g_f(CE_2^{k+1}),$$

which leads to the identity,

$$\begin{aligned} M_f(n, E) &= M_f(k, S) + g_f(SE_2^{k+1}) + g_f(a_{k+2}) + \dots + g_f(a_n) \\ &= M_f(k, C) + g_f(CE_2^{k+1}) + g_f(a_{k+2}) + \dots + g_f(a_n). \end{aligned}$$

Now, without loss of generality, assume that the action a_{k+1} was equal to SE_2^{k+1} . Then, by the first statement,

$$M_f(k, S) = TI_f((T, SS^*, a_3, \dots, a_k)).$$

Moreover, there exists an action sequence $(T, SS^*, b_3, \dots, b_k)$ such that

$$M_f(k, C) = TI_f((T, SS^*, b_3, \dots, b_k)).$$

Therefore, the action sequences

$$(T, SS^*, a_3, \dots, a_k, a_{k+1}, a_{k+2}, \dots, a_n) \quad \text{and} \quad (T, SS^*, b_3, \dots, b_k, CE_2^{k+1}, a_{k+2}, \dots, a_n)$$

are distinct and both attain the value $M_f(n, E)$, as required.

- c) To prove the sufficiency direction, assume that all maximal action sequences associated with $M_f(n, E)$ share the same last $n-k$ actions, denoted by a_m for $m \in \{k+1, \dots, n\}$, and that at least two of these action sequences differ at the k^{th} action. Selecting two such sequences that differ at index k and applying the second part of this lemma, it follows that

$$n_t(k, E_2^k) = 1 + n_t(k-1, S) + n_t(k-1, C).$$

; where notice every maximal action sequence for $M_f(n, E)$ has its k^{th} action ending at state E_2^k since they all coincide in the $(k+1)^{\text{th}}$ action.

Now, to prove that $n_t(m, E_2^m) = n_t(m-1, E_2^{m-1})$ for all $m \in \{k+1, \dots, n\}$, suppose by way of contradiction that

$$n_t(m, E_2^m) = 1 + n_t(m-1, S) + n_t(m-1, C).$$

Applying the second part of the lemma, this implies the existence of a maximal action sequence for $M_f(n, E)$ whose m^{th} action differs from a_m , contradicting the assumption that all maximal action sequences share the same last $n - k$ actions.

For the reverse implication of the statement assume that

$$n_t(n, E) = n_t(n - 1, E_2^{n-1}) = \dots = n_t(k, E_2^k) = 1 + n_t(k - 1, S) + n_t(k - 1, C)$$

for some $k \in \{3, \dots, n\}$.

Let A denote the set of all maximal action sequences associated with $M_f(n, E)$. Then, every chain in A must share the last $n - k$ actions with the action sequence $(T, SS^*, a_3, \dots, a_k, a_{k+1}, \dots, a_n)$. To verify this, suppose that there exists a sequence in A that differs from it for the first time (from right to left) at some action of index j , with $j \in \{k + 1, \dots, n\}$. Then, by the second part of this lemma, it follows that

$$n_t(j, E_2^j) = 1 + n_t(j - 1, S) + n_t(j - 1, C),$$

which contradicts our initial assumption.

Finally, we need to show that there exist at least two action sequences in A that differ at the k^{th} action. This follows directly from the fact that all sequences in A have their $(k + 1)^{\text{th}}$ action equal, and since

$$n_t(k, E_2^k) = 1 + n_t(k - 1, S) + n_t(k - 1, C),$$

the second part of this lemma ensures the existence of two action sequences differing at the k^{th} action.

Thus, both conditions required in the statement are satisfied.

□

Theorem 2.3.3. *Let $n \geq 1$ and $E \in \{S, C\}$. Then $n_t(n, E) + 1$ equals the number of distinct maximal action sequences, for the problem $M_f(n, E)$, among all action sequences with n squares and ending at state E .*

Proof. The result is immediate for $n = 1, 2$. For $n \geq 3$, we proceed by induction on the number $m \geq 1$ of distinct maximal action sequences.

Base case. Let $m = 1$, and suppose that $n_t(n, E) \geq 1$. Then, by the recursive definition of $n_t(n, E)$, there must exist $k \in \{3, \dots, n\}$ such that

$$n_t(n, E) = n_t(n - 1, E_2^{n-1}) = \dots = n_t(k, E_2^k) = 1 + n_t(k - 1, S) + n_t(k - 1, C)$$

where the states $\{E_2^{n-1}, \dots, E_2^k\}$ are taken from one of the existing action sequences $((T, SS^*, a_3, \dots, a_k, \dots, a_n)$ corresponding to $M_f(n, E)$.

However, the second statement of Lemma 2.3.2 implies the existence of at least two distinct maximal action sequences, which contradicts the assumption that $m = 1$. Therefore, the base case holds.

Inductive step. Assume the result holds for all $m' < m$, with $m > 1$. By the constructive proof of the first implication in the second statement of Lemma 2.3.2, each pair of maximal action sequences is associated with some index $k \in \{3, \dots, n\}$. Among the m maximal action sequences, there must exist a pair for which this index attains its largest value, say $n^* \leq n$. As a consequence, the maximal number of consecutive links, starting from the last one, that are shared by all maximal action sequences is $n - n^*$. By the third statement of Lemma 2.3.2 it then follows that $n_t(n, E) =$

$n_t(n^*, E_2^{n^*}) = 1 + n_t(n^* - 1, S) + n_t(n^* - 1, C)$. On the other hand, we can partition the set of maximal action sequences into two subsets according to the value of the action at index n^* : one with $a_{n^*} = SE_2^{n^*}$ and another with $a_{n^*} = CE_2^{n^*}$, having cardinalities m_1 and m_2 , respectively; $m_1 + m_2 = m$ and $m_i \geq 1$ for $i = 1, 2$.

Now, by the first statement of Lemma 2.3.2, these m_1 and m_2 maximal sequences can be cut into maximal action sequences for the subproblems $M_f(n^* - 1, S)$ and $M_f(n^* - 1, C)$, respectively. Moreover, such sequences have to correspond to all maximal action sequences for these subproblems; otherwise, a distinct maximal sequence in either subset could be extended to produce a new maximal sequence for $M_f(n, E)$, contradicting the assumption that there are exactly m maximal action sequences. Thus, by the induction hypothesis, we have

$$n_t(n^*, S) = m_1 - 1 \quad \text{and} \quad n_t(n^*, C) = m_2 - 1,$$

which implies

$$n_t(n, E) = n_t(n^*, E_2^{n^*}) = 1 + (m_1 - 1) + (m_2 - 1) = m - 1.$$

This completes the inductive proof. □

Based on Theorem 2.3.3 and using the tie graph generated by the dynamic programming framework over problems $n_t(n, E)$, we can now determine all maximal action sequences corresponding to problem $M_f(n, E)$. In order to do this, we similarly define a construction method for $n_t(n, E)$ by recursively selecting each action a_k for $k \in \{n, \dots, 4\}$, according to the following rules:

- If we are at problem $n_t(k, E)$ and $n_t(k, E) = n_t(k - 1, S)$ or $n_t(k - 1, E) = n_t(k - 1, C)$, then assign $a_k = SE$ and move to problem $n_t(k - 1, S)$ or assign $a_k = CE$ and move to problem $n_t(k - 1, C)$, respectively.
- Otherwise, if $n_t(k, E) = 1 + n_t(k - 1, S) + n_t(k - 1, C)$, we have two options: Either assign $a_k = SE$ and move to problem $n_t(k - 1, S)$ or assign $a_k = CE$ and move to problem $n_t(k - 1, C)$.

Finally, if we are at problem $n_t(3, S)$ assign $a_1 = T, a_2 = SS^*, a_3 = SS$ or $a_1 = T, a_2 = SS^*, a_3 = SC^*$ otherwise.

Notice, that by performing every possible assignation in the constructive method associated to problem $n_t(n, E)$, we obtain all maximal action sequences for problem $M_f(n, E)$. Moreover, by computing all maximal action sequences for problems $M_f(n, S)$ and $M_f(n, C)$, as appropriate, we can obtain all of the extremal action sequences with n squares. As a special case concerning the cardinality of the set of all maximal action sequences, we obtain the following corollary.

Corollary 2.3.4. *For $n \geq 1$ and $E \in \{S, C\}$. If $n_t(n, E) = 0$, then the maximal action sequence retrieved by the constructive method of $M_f(n, E)$ is the unique maximal action sequence.*

Remark 2.3.5. Thanks to Remark 2.1.10, observe that obtaining the set of all extremal action sequences corresponds to obtaining the set of all extremal polyomino chains up to isomorphism (which is something that will be further improved in Section 2.5.1). Also, in regard to the time complexity of performing the constructive method for $n_t(n, E)$, notice this is exponential with respect to n since the amount of all maximal action sequences corresponding to problem $M_f(n, E)$ could possibly be the set of all action sequences ending at state E , whose cardinality is 2^{n-1} and could happen if for example g_f assigned the same value to every action.

Now that we have developed tools to identify all extremal polyomino chains for any given degree-based index, we proceed to characterize all of the extremal polyomino chains for the AZI .

Theorem 2.3.6. *For $n \geq 5$, the polyomino chains that maximize the AZI are $AZ_{\frac{n-1}{2}}^1$ when n is odd and the family $AZ_{\frac{n}{2}}^2$ when n is even. Specifically,*

$$M(n) = \frac{4456}{125}n - \frac{26763}{2000} - \frac{2312}{3375}I_{\{n=2k\}}.$$

Proof. Using the calculations done in Theorem 2.2.14, it is immediate to obtain the AZI's tie graph shown in Figure 2.13, where in particular, the corresponding cell to problem $n_t(7, C)$ has an edge to both problems $n_t(6, S)$ and $n_t(6, C)$ because we had previously obtained $M(7, C) = M(6, S) + g(SC) = M(6, C) + g(CC)$. Also, since for $n = 7$ and $j = 5$ we arrived at the conditions of Theorem 2.2.11, the behavior for problems $n_t(n, S)$ and $n_t(n, C)$ repeats itself in the same way by experiencing cycles of size two, which notice is justified because all of the results in Section 2.2.3 preserved equalities, meaning ties are preserved throughout cyclic patterns.

Therefore, based on the previous information and applying the constructive method for the problem $n_t(n, S)$, we conclude the following:

When n is odd, by Corollary 2.3.4, the unique polyomino chain that maximizes the AZI index is defined by the action sequence

$$(T, SS^*, SS, \underbrace{SC, CS}_{\frac{n-3}{2} \text{ times}}),$$

which corresponds to a polyomino chain composed of $\frac{n-1}{2}$ segments, each of length 3. In other words, it is the polyomino chain $AZ_{\frac{n-1}{2}}^1$.

On the other hand, when n is even, we can show by induction that the polyomino chains that maximize the AZI index are defined by action sequences of the form:

$$(T, SS^*, SS, \underbrace{SC, CS}_m, SC, CC, CS, \underbrace{SC, CS}_{\frac{n-6}{2}-m}); \quad 0 \leq m \leq \frac{n-6}{2}.$$

For $n = 6$, it is clear by performing the constructive method explicitly, that the only maximal action sequence is $(T, SS^*, SS, SC, CC, CS)$.

Now, for $n \geq 8$, following the tie graph dynamic, we have:

$$\begin{aligned} M(n, S) &= M(n-2, S) + g(SC) + g(CS) \\ &= M_f(n-3, S) + g_f(SC) + g_f(CC) + g_f(CS). \end{aligned}$$

which by the induction hypothesis implies every maximal action sequence for problem $M(n, S)$ can be written as

$$(T, SS^*, SS, \underbrace{SC, CS}_m, SC, CC, CS, \underbrace{SC, CS}_{\frac{n-8}{2}-m}, SC, CS) \quad \text{with } 1 \leq m \leq \frac{n-8}{2}$$

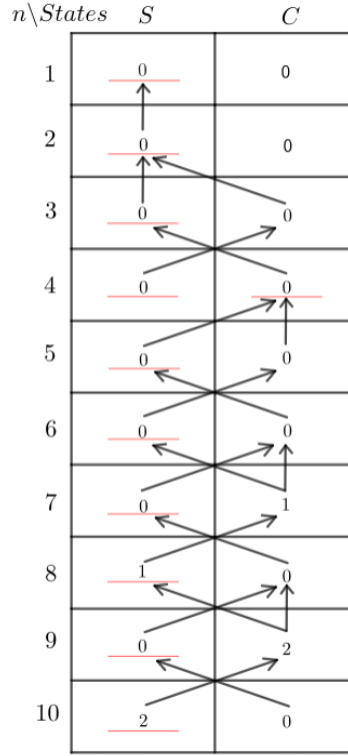
or

$$(T, SS^*, SS, \underbrace{SC, CS}_{\frac{n-5}{2} \text{ times}}, SC, CC, CS),$$

which together complete all cases of the desired form. In terms of segments, these sequences correspond to polyomino chains with $\frac{n}{2}$ segments, all of length 3 except for one internal segment of length 2. In other words, they belong to the family $AZ_{\frac{n}{2}}^2$.

Finally, the value of $M(n)$ follows directly from Lemma 2.1.7.

□


 Figure 2.13: The tie graph for the maximum AZI value.

Remark 2.3.7. Notice, as a direct application of Corollary 2.3.4, the maximum action sequences for $n \leq 4$ remain to be the ones stated in Remark 2.2.15. Furthermore, in this particular case of the AZI , we can find all maximum non-isomorphic polyomino chains. In order to do this, it is sufficient to observe that for $n \geq 6$ and n even, the polyomino chain given by the action sequence

$$(T, SS^*, SS, \underbrace{SC, CS}_{m \text{ times}}, SC, CC, CS, \underbrace{SC, CS}_{\frac{n-6}{2}-m \text{ times}})$$

is isomorphic to the one given by the action sequence

$$(T, SS^*, SS, \underbrace{SC, CS}_{\frac{n-6}{2}-m \text{ times}}, SC, CC, CS, \underbrace{SC, CS}_{m \text{ times}}).$$

Performing a simple counting argument, we can also conclude that the number of non-isomorphic polyomino chains for $n \geq 6$ and n even is $\lceil \frac{n}{4} - 1 \rceil$.

2.4. Long-Term Behavior of Extremal Polyomino Chains and their Bang-Bang Dynamic

As could be appreciated from the proofs of Theorems 2.2.14 and 2.3.6, the repetition of decision scenarios in the dynamic programming framework underlying problems $M_f(n, E)$ was crucial in order to pinpoint the long-term behavior of the maximal action sequences. Motivated by this observation, it now becomes relevant to study conditions for g_f under which the decisions made to compute $M_f(n, E)$ follow specific patterns.

Working on this line of thought, actions SS and CC stand out by the fact they can be repeatedly chosen, in contrast to actions SC and CS which need an intermediate step in order to be able to be used once again. In the following lemma we show that due to this property, conditions can be placed over the actions SS or CC so that they overwhelm the decision process and ultimately yield a specific pattern for extremal polyomino chains.

Lemma 2.4.1. *Let $g_f(SS) > \max\{g_f(SC), g_f(CC), \frac{g_f(SC)+g_f(CS)}{2}\}$. Then the following statements hold:*

- a) *If for $n \geq 4$ $M_f(n, S) = M_f(n-1, S) + g_f(SS)$, then $M_f(n+1, S) = M_f(n, S) + g_f(SS)$ without ties.*
Moreover, for $n \geq 4$ $M_f(n, S) = M_f(n-1, S) + g_f(SS)$ without ties.
- b) *For $n \geq 3$ if $M_f(n) = M_f(n, S)$, then $M_f(n+1) = M_f(n+1, S)$ and $M_f(n+1) \neq M_f(n+1, C)$.*
- c) *If for $n \geq 4$ $M_f(n) = M_f(n, C)$, then $M_f(n, C) = M_f(n-1, C) + g_f(CC)$ without ties.*
Moreover, for some $n^ \geq 3$ it must be $M_f(n) = M_f(n, S)$.*
- d) *For $n \geq 1$ if $M_f(n) = M_f(n, E)$ then $n_t(n, E) = 0$.*

Proof.

- a) Because $M_f(n, S) = M_f(n-1, S) + g_f(SS)$, we have:

$$\begin{aligned} M_f(n-1, S) + g_f(SS) &\geq M_f(n-1, C) + g_f(CS) \\ \Leftrightarrow M_f(n-1, C) - M_f(n-1, S) &\leq g_f(SS) - g_f(CS), \end{aligned}$$

which means the gain produced by using action SS is greater than or equal to the gain produced by building over problem $M_f(n-1, C)$. We shall now prove this is also the case for n , that is:

$$M_f(n, C) - M_f(n, S) \leq g_f(SS) - g_f(CS).$$

Since $M_f(n, C) = \max\{M_f(n-1, S) + g_f(SC), M_f(n-1, C) + g_f(CC)\}$, we have two cases:

Case 1: Assume $M_f(n, C) = M_f(n-1, S) + g_f(SC)$. In this case it follows

$$\begin{aligned} M_f(n, C) - M_f(n, S) &= M_f(n-1, S) + g_f(SC) - (M_f(n-1, S) + g_f(SS)) \\ &= g_f(SC) - g_f(SS) \\ &< g_f(SS) - g_f(CS) \quad \text{because } g_f(SS) > \frac{g_f(SC) + g_f(CS)}{2} \end{aligned}$$

Case 2: Assume $M_f(n, C) = M_f(n-1, C) + g_f(CC)$. In this case it follows

$$\begin{aligned} M_f(n, C) - M_f(n, S) &= M_f(n-1, C) + g_f(CC) - (M_f(n-1, S) + g_f(SS)) \\ &= M_f(n-1, C) - M_f(n-1, S) + g_f(CC) - g_f(SS) \\ &\leq g_f(SS) - g_f(CS) + g_f(CC) - g_f(SS) \\ &= g_f(CC) - g_f(CS) \\ &< g_f(SS) - g_f(CS) \end{aligned}$$

Therefore, in any case we can conclude $M_f(n, C) + g_f(CS) < M_f(n, S) + g_f(SS)$, from which it follows $M_f(n+1, S) = M_f(n, S) + g_f(SS)$ and by the strict inequality this happens without a tie.

For the second part, notice it is now sufficient to show $M_f(4, S) = M_f(3, S) + g_f(SS)$ without ties. Suppose by way of contradiction this wasn't the case, then we would have:

$$\begin{aligned}
 M_f(3, S) + g_f(SS) &\leq M_f(3, C) + g_f(CS) \\
 \Leftrightarrow g_f(T) + g_f(SS^*) + g_f(SS) + g_f(SS) &\leq g_f(T) + g_f(SS^*) + g_f(SC^*) + g_f(CS) \\
 \Leftrightarrow 2g_f(SS) &\leq g_f(SC^*) + g_f(CS) \\
 \Leftrightarrow 2g_f(SS) &\leq g_f(SS) + g_f(SC) - g_f(CS) + g_f(CS) \quad \text{by Remark 2.1.12} \\
 \Leftrightarrow g_f(SS) &\leq g_f(SC),
 \end{aligned}$$

which is a contradiction.

- b) Let $M_f(n) = M_f(n, S)$, which means $M_f(n, S) \geq M_f(n, C)$. Then by our main hypothesis it follows

$$\begin{aligned}
 M_f(n+1, S) &= \max\{M_f(n, S) + g_f(SS), M_f(n, C) + g_f(CS)\} \\
 &\geq M_f(n, S) + g_f(SS) \\
 &= \max\{M_f(n, S) + g_f(SC), M_f(n, C) + g_f(CC)\} \\
 &= M_f(n+1, C),
 \end{aligned}$$

from which we conclude $M_f(n+1) = M_f(n+1, S)$ and $M_f(n+1) \neq M_f(n+1, C)$.

- c) Suppose by way of contradiction for some $n \geq 4$ it was the case $M_f(n-1, C) + g_f(CC) \leq M_f(n-1, S) + g_f(SC)$. Then because $g_f(SS) > g_f(SC)$, we obtain:

$$M_f(n-1, C) + g_f(CC) \leq M_f(n-1, S) + g_f(SC) < M_f(n-1, S) + g_f(SS),$$

which implies $M_f(n) = M_f(n, S) > M_f(n, C)$. Thus we reach a contradiction.

For the second part, if for $n = 3$ we had $M_f(n) = M_f(n, S)$ (which happens iff $g_f(SS) \geq g_f(SC^*)$) we are done. So suppose $M_f(n) \neq M_f(n, S)$, that is, $M_f(n)$ is equal only to $M_f(n, C)$. We have this will remain true while:

$$\begin{aligned}
 M_f(n, S) &< M_f(n, C) \\
 \Rightarrow g_f(T) + g_f(SS^*) + g_f(SS) + (n-3)g_f(SS) &< g_f(T) + g_f(SS^*) + g_f(SC^*) + (n-3)g_f(CC) \\
 \Rightarrow g_f(SS) + (n-3)g_f(SS) &< g_f(SC^*) + (n-3)g_f(CC) \\
 \Rightarrow n &< \frac{g_f(SC^*) - g_f(SS)}{g_f(SS) - g_f(CC)} + 3
 \end{aligned}$$

Therefore, for $n^* = \left\lceil \frac{g_f(SC^*) - g_f(SS)}{g_f(SS) - g_f(CC)} + 3 \right\rceil$ it follows $M_f(n^*) = M_f(n^*, S)$.

- d) This is immediate from the fact by part a) of this lemma we have for every $n \geq 4$ $M_f(n, S) = M_f(n-1, S) + g_f(SS)$ without ties and by part c), that if $M_f(n) = M_f(n, C)$ then $M_f(n, C) = M_f(n-1, C) + g_f(CC)$ without ties. Also notice the result is trivial for $n \leq 3$.

□

Notice, that differently as it happened in previous patterns in the dynamic programming frameworks, the pattern established by Lemma 2.4.1 isn't one concerning the repetition of differences on the problems $M_f(n, S)$ and $M_f(n, C)$, but one concerning the nature of the states themselves. What this means, is that Lemma 2.4.1 has established a dynamic over the way we can transition from being in a state to another. In Figure 2.14 we show the graph representing this dynamic, from which it can be seen state S has become a recurrent state (that is a state that once entered



Figure 2.14: The bang-bang dynamic established over states S and C through Lemma 2.4.1.

is never left) while state C has become transitive (that is a state that once left is never entered). Altogether, this implies that the structure of the dynamic programming pattern follows what is called in dynamic systems as a bang-bang structure.

In the next theorem, thanks to the given bang-bang structure, we characterize the extremal polyomino chains for the established dynamic, which is a direct consequence of Lemma 2.4.1.

Theorem 2.4.2. *Let TI_f be a degree-based topological index such that*

$$g_f(SS) > \max \left\{ g_f(SC), g_f(CC), \frac{g_f(SC) + g_f(CS)}{2} \right\}.$$

Then:

- a) *If $g_f(SS) > g_f(SC^*)$, then among all polyomino chains with $n \geq 3$ squares, the linear chain Li_n uniquely maximizes TI_f .*
- b) *If $g_f(SS) = g_f(SC^*)$, then for all $n \geq 4$, Li_n uniquely maximizes TI_f ; and for $n = 3$, we have the two possible polyomino chains maximize it.*
- c) *If $g_f(SS) < g_f(SC^*)$, define the threshold $n^* := \left\lceil \frac{g_f(SC^*) - g_f(SS)}{g_f(SS) - g_f(CC)} + 3 \right\rceil$. Then:*
 - i) *For $3 \leq n < n^*$, Z_n uniquely maximizes TI_f .*
 - ii) *For $n = n^*$, Li_n achieves the maximum TI_f ; and possibly Z_n does as well.*
 - iii) *For $n > n^*$, Li_n uniquely maximizes TI_f .*

Remark 2.4.3. As a concrete example of Theorem 2.4.2, we can define g_f as follows:

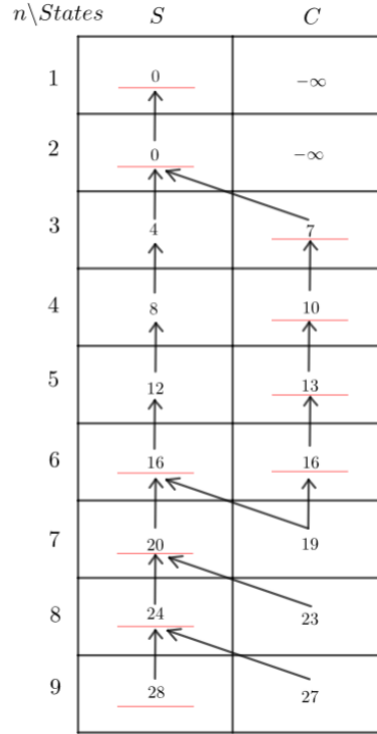
$$g_f(T) = 0, g_f(SS^*) = 0, g_f(SS) = 4, g_f(SC) = 3, g_f(CS) = 0, g_f(CC) = 3, g_f(SC^*) = 7$$

from which it is clear g_f satisfies the conditions given by the theorem and falls within case c) with $n^* = 6$. In particular, since the function has so simple numeric values, it is easy to compute its tie graph, see Figure 2.15; where in each cell we write the values for the problems $M_f(n, E)$.

Also, because action CC plays the role of a dual version of action SS , we can similarly give conditions under which the pattern over the dynamic programming framework has an analogous behavior.

Lemma 2.4.4. *Let $g_f(CC) > \max \{ g_f(CS), g_f(SS), \frac{g_f(SC) + g_f(CS)}{2} \}$. Then the following statements hold:*

- a) *If for $n \geq 4$ $M_f(n, C) = M_f(n-1, C) + g_f(CC)$, then $M_f(n+1, C) = M_f(n, C) + g_f(CC)$ without ties. Moreover, for $n \geq 4$ $M_f(n, C) = M_f(n-1, C) + g_f(CC)$ without ties.*
- b) *For $n \geq 3$ if $M_f(n) = M_f(n, C)$, then $M_f(n+1) = M_f(n+1, C)$ and $M_f(n+1) \neq M_f(n+1, S)$.*
- c) *If for $n \geq 4$ $M_f(n) = M_f(n, S)$, then $M_f(n, S) = M_f(n-1, S) + g_f(SS)$ without ties. Moreover, for some $n^* \geq 3$ it must be $M_f(n) = M_f(n, C)$.*


 Figure 2.15: The tie graph for the function g_f of Remark 2.4.3.

d) For $n \geq 1$ if $M_f(n) = M_f(n, E)$ then $n_t(n, E) = 0$.

Theorem 2.4.5. Let TI_f be a degree-based topological index such that

$$g_f(CC) > \max \left\{ g_f(CS), g_f(SS), \frac{g_f(SC) + g_f(CS)}{2} \right\}.$$

Then:

- a) If $g_f(SS) < g_f(SC^*)$, then among all polyomino chains with $n \geq 3$ squares, the linear chain Z_n uniquely maximizes TI_f .
- b) If $g_f(SS) = g_f(SC^*)$, then for all $n \geq 4$, Z_n uniquely maximizes TI_f ; and for $n = 3$, we have the two possible polyomino chains maximize it.
- c) If $g_f(SS) > g_f(SC^*)$, define the threshold $n^* := \left\lceil \frac{g_f(SS) - g_f(SC^*)}{g_f(CC) - g_f(SS)} + 3 \right\rceil$. Then:
 - i) For $3 \leq n < n^*$, Li_n uniquely maximizes TI_f .
 - ii) For $n = n^*$, Z_n achieves the maximum TI_f ; and possibly Li_n does as well.
 - iii) For $n > n^*$, Z_n uniquely maximizes TI_f .

Finally, as a consequence of Theorem 2.4.2 and Theorem 2.4.5, we can retrieve most of the literature's results. However, it is important to highlight we are working only over the set of polyomino chains, while the results compiled in Theorem 1.2.14 were obtained over the set of general polyomino chains. Even so, in Section 3.4.1 we'll show for these indices it is sufficient to work in the restricted polyomino chain setting.

Corollary 2.4.6. *Let PC_n denote any polyomino chain with n squares:*

$$\begin{array}{ll} M_i(Li_n) \leq M_i(PC_n) \leq M_i(Z_n) & GA(Z_n) \leq GA(PC_n) \leq GA(Li_n) \\ H(Z_n) \leq H(PC_n) \leq H(Li_n) & \chi(Z_n) \leq \chi(PC_n) \leq \chi(Li_n) \\ R(Z_n) \leq R(PC_n) \leq R(Li_n) & ABC(PC_n) \leq ABC(Z_n) \end{array}$$

Moreover, among all polyomino chains with n squares, the linear chain Li_n attains the minimum AZI when $n \geq 6$, while the zigzag chain Z_n achieves the minimum AZI for $n = 3, 4, 5$.

2.5. Isomorphic Polyomino Chains and their Extremal Dynamics

2.5.1. The Flip of a Polyomino Chain

By now, we have developed tools that allow us to obtain for any degree-based topological index, either one extremal polyomino chain in linear time or the set of all extremal polyomino chains in exponential time, in the worst case scenario. However, when we obtain the set of all extremal polyomino chains, it could still be many of them are isomorphic. Thus, in order to be able to further refine the set of all extremal polyomino chains obtained through the use of the constructive method of problems $n_t(n, E)$, we must now explore the relationship of isomorphic polyomino chains, actions and their TI_f value.

To begin our exploration, notice that the relationship of isomorphisms between polyomino chains and the actions that compose their representations is not direct. For example, take the action sequences $(T, SS^*, SC^*, CS, SS, SS)$ and $(T, SS^*, SS, SS, SS, SC)$, it is clear these sequences do not share the same amount of every action, nevertheless the polyomino chains they represent are isomorphic as can be seen from Figure 2.16 in which graphs *a)* and *b)* correspond to the first and second sequence respectively. Moreover, even if we had two action sequences with the same amount for every action (which immediately implies they have the same TI_f value), it could be the polyomino chains they represent are not isomorphic. As an example take the sequences $(T, SS^*, SS, SS, SC, CS, SS, SC)$ and $(T, SS^*, SS, SS, SS, SC, CS, SC)$, whose graphs are shown in Figure 2.17 in *a)* and *b)* respectively.

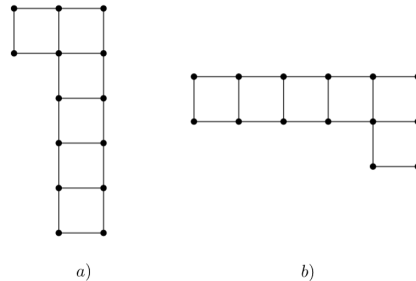


Figure 2.16: Two isomorphic polyomino chains that do not share the same amount of every action.

Continuing our exploration, we now go on to define our key concept in our study of isomorphic polyomino chains, which is the notion of the flip of an action sequence; where the name flip is motivated by its involution property displayed at Proposition 2.5.5.

Definition 2.5.1. Let a be an action in the set $\{SS, SC, CS, CC\}$. We define its *flip action*, denoted by a^F , as $a^F = E_2E_1$; $a = E_1E_2$.

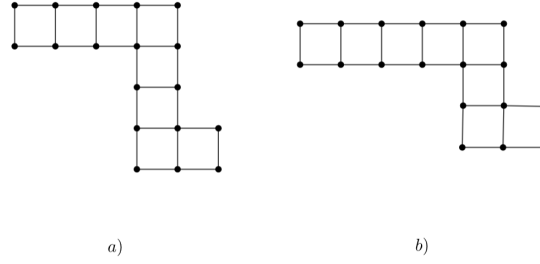


Figure 2.17: Two non-isomorphic polyomino chains that do share the same amount of every action.

Definition 2.5.2. Let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence with $n \geq 1$. We define its *flip action sequence* as $(T, SS^*, b_3, \dots, b_n)$ where:

- $b_3 = SS$ if a_n ends at state S and $b_3 = SC^*$ if a_n ends at state C .
- For $k \geq 4$, $b_k = a_{(n-k+4)}^F$.

And we denote this sequence $(T, SS^*, b_3, \dots, b_n)$ by $(T, SS^*, a_3, \dots, a_n)^F$.

Remark 2.5.3. For example, for the action sequence

$$(T, SS^*, SC^*, CS, SS, SC, CC, CC, CS, SS, SC, CS, SC, CC, CC, CC)$$

we have its flip action sequence is

$$(T, SS^*, SC^*, CC, CC, CC, CS, SC, CS, SS, SC, CC, CC, CS, SS, SC)$$

In Figure 2.18 we show their graphs, which are given by a) and b) respectively.

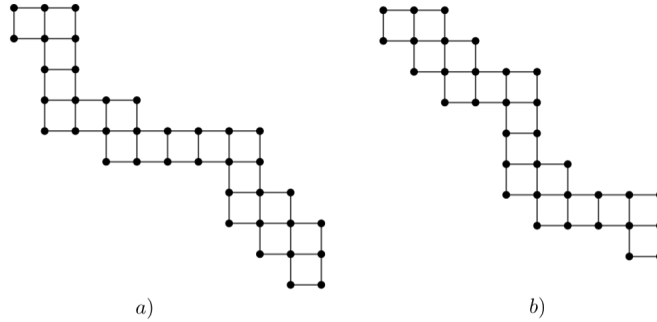


Figure 2.18: Example of the polyomino chains given by an action sequence and its flip action sequence.

Proposition 2.5.4. Let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence, then $(T, SS^*, a_3, \dots, a_n)^F$ is also an action sequence.

Proof. Let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence where we will denote the starting and ending states of one of its actions a_k as A_1^k and A_2^k respectively, and let $(T, SS^*, b_3, \dots, b_n)$ be its flip action sequence where we will denote the starting and ending states of one of its actions b_k as B_1^k and B_2^k respectively.

We have to show first of all $B_2^3 = B_1^4$. So suppose a_n ends at state S , then by Definition 2.5.2 it follows $b_3 = SS$ and thus $B_2^3 = S$. Also, by Definition 2.5.2 we have B_1^4 equals the ending state of action $a_{n-4+4} = a_n$, which implies $B_2^3 = B_1^4$. The case a_n ends at state C is analogous.

Now, for $4 \leq k < n$, let's show $B_2^k = B_1^{k+1}$. We have since $b_k = a_{(n-k+4)}^F$, that $B_2^k = A_1^{n-k+4}$. Also, because $b_{k+1} = a_{(n-k+3)}^F$, we can conclude $B_1^{k+1} = A_2^{n-k+3} = A_1^{n-k+4} = B_2^k$.

Therefore the flip of an action sequence is also an action sequence. $\square$

Proposition 2.5.5. *Let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence, then $(T, SS^*, a_3, \dots, a_n) = ((T, SS^*, a_3, \dots, a_n)^F)^F$.*

Proof. Let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence, $(T, SS^*, c_3, \dots, c_n) = (T, SS^*, a_3, \dots, a_n)^F$ and $(T, SS^*, b_3, \dots, b_n) = ((T, SS^*, a_3, \dots, a_n)^F)^F$. We'll show $(T, SS^*, b_3, \dots, b_n) = (T, SS^*, a_3, \dots, a_n)$.

We have $c_n = a_4^F = (E_1^4 E_2^4)^F = E_2^4 E_1^4 = E_2^4 E_2^3$. But then, since b_3 ends at the state c_n ended, it follows b_3 ends at the same state as a_3 and because both come from state S , we have $b_3 = a_3$.

Now, for $4 \leq k \leq n$, we have $b_k = c_{(n-k+4)}^F = (a_{(n-(n-k+4)+4)}^F)^F = (a_k^F)^F = ((E_1^k E_2^k)^F)^F = (E_2^k E_1^k)^F = E_1^k E_2^k = a_k$.

Thus $(T, SS^*, a_3, \dots, a_n) = ((T, SS^*, a_3, \dots, a_n)^F)^F$. $\square$

Because of Proposition 2.5.4 and Remark 2.1.10, it is clear that the flip of an action sequence represents a polyomino chain. However, it remains unclear whether this polyomino chain is isomorphic to the one represented by the original action sequence or even if it has the same TI_f value. In accordance to these questions, we now prove an important result with respect to the possible isomorphisms of polyomino chains, in which the concept of the flip of an action sequence is central.

Notation 2.5.6. Let PC_n be a polyomino chain, we denote by $PC_n(i, j)$ the polyomino chain consisting of the i^{th} up to the j^{th} square of PC_n ; where $i \leq j$ and $1 \leq i, j \leq n$.

Also, we refer as the *reverse interpretation* of PC_n to the instruction sequence of PC_n when its squares are considered in the reverse order of its path.

Theorem 2.5.7. *For $n \geq 3$ let $PC1_n$ and $PC2_n$ be polyomino chains with action sequences $(T, SS^*, a_3, \dots, a_n)$ and $(T, SS^*, b_3, \dots, b_n)$ respectively. Then $PC1_n \cong PC2_n$, if and only if*

$$(T, SS^*, b_3, \dots, b_n) = (T, SS^*, a_3, \dots, a_n)$$

or

$$(T, SS^*, b_3, \dots, b_n) = (T, SS^*, a_3, \dots, a_n)^F,$$

Proof. To proof the sufficiency direction, notice because $n \geq 3$, $PC1_n$ has two different squares, namely its first and its last, which have exactly two adjacent vertices with degree 2. Thus if $PC1_n$ is isomorphic to $PC2_n$, then it must be the first square of $PC1_n$ is mapped to the first of $PC2_n$ or to its last.

Based on the previous observation we break the proof into cases.

- 1) Suppose the first square of $PC1_n$ is mapped to the first of $PC2_n$. Notice since graph isomorphisms preserve adjacency, it follows the k^{th} square of $PC1_n$ must be mapped onto the k^{th} of $PC2_n$, which is straightforward to show by induction using the definition of polyomino chain (see Definition 1.1.3).

Let's show $(T, SS^*, b_3, \dots, b_n) = (T, SS^*, a_3, \dots, a_n)$. We'll proceed by induction on n .

For $n = 2$ it is clear their first two actions coincide and that $PC1_n(1, 2) \cong PC2_n(1, 2)$.

Now, assume for $2 \leq k < n$ it is true $(T, SS^*, b_3, \dots, b_k) = (T, SS^*, a_3, \dots, a_k)$ and that $PC1_n(1, k) \cong PC2_n(1, k)$. Then, for $k + 1$ suppose by way of contradiction $b_{k+1} \neq a_{k+1}$. Since the sequences $(T, SS^*, b_3, \dots, b_n)$ and $(T, SS^*, a_3, \dots, a_n)$ are action sequences, this means a_{k+1} and b_{k+1} must come from the same state but end at a different one.

Assume without loss of generality a_{k+1} ended at state S and b_{k+1} ended at state C . It then follows a_{k+1} maintained the direction of $PC1_n(1, k)$ while b_{k+1} changed the direction of $PC2_n(1, k)$. As a consequence, the degrees of the vertices of the $(k - 1)^{th}$ square in $PC1_n(1, k + 1)$ remained unchanged while those of the $(k - 1)^{th}$ square in $PC2_n(1, k + 1)$ changed (see Figure 2.19).

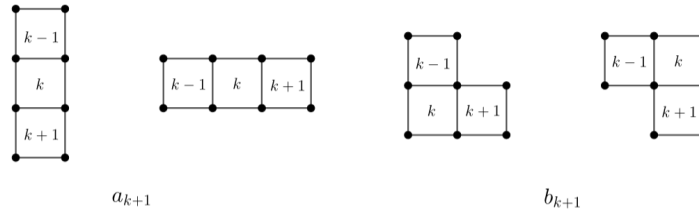


Figure 2.19: The possible graphs of the last three squares of the polyomino chains $PC1_n(1, k + 1)$ and $PC2_n(1, k + 1)$ in the case a_k ends at state S and b_k end at state C , respectively.

Moreover, since no square after the $(k + 1)^{th}$ can alter the degrees of the $(k - 1)^{th}$ squares and by the induction hypothesis prior to adding the $(k + 1)^{th}$ square the degrees of the $(k - 1)^{th}$ squares of $PC1_n(1, k)$ and $PC2_n(1, k)$ were equal, it follows in $PC1_n$ and $PC2_n$ the degrees of the vertices of their $(k - 1)^{th}$ squares do not match. Hence contradicting $PC1_n \cong PC2_n$. Therefore $b_{k+1} = a_{k+1}$ and it is immediate $PC1_n(1, k + 1) \cong PC2_n(1, k + 1)$.

We can thus conclude

$$(T, SS^*, b_3, \dots, b_n) = (T, SS^*, a_3, \dots, a_n)$$

- 2) Suppose the first square of $PC1_n$ was mapped to the last of $PC2_n$, which is equivalent to having the last square of $PC1_n$ mapped to the first of $PC2_n$. In this case, thanks again to the fact isomorphisms preserve adjacency, we can also show by induction the k^{th} square of $PC1_n$ must be mapped to the $(n - k + 1)^{th}$ square of $PC2_n$.

Let's show for $B := (T, SS^*, b_3, \dots, b_n)$ and $A := (T, SS^*, a_3, \dots, a_n)^F$, $B = A$. We'll proceed by induction on n .

For $n = 2$ it is clear the first two actions of B equal those of A and that $PC2_n(1, 2) \cong PC1_n(n - 1, n)$.

Now, for $n = 3$, if the action a_n ends at state S , it must be the first square in $PC1_n(n - 2, n)$ maintains the direction of the chain when read under the reverse interpretation. Otherwise, it would imply $a_n = SC$ which is a contradiction (for a graphic representation see Figure 2.20 a)). Performing then the same argument as in case 1), it has to be b_3 also maintains the direction of $PC2_n(1, 2)$. Hence $b_3 = SS$ which is precisely the the third action of A .

Similarly, if a_n ends at state C , we can show $b_3 = SC^*$ (see Figure 2.20 b)) which would be equal to the third action of A .

In either case we also conclude $PC2_n(1, 3) \cong PC1_n(n - 2, n)$.

Now, for $3 \leq k < n$ suppose the first k actions of B equal the first k actions of A and $PC2_n(1, k) \cong PC1_n(n - k + 1, n)$. We must show $b_k = a_{(n-k+4)}^F$. We'll perform the argument by cases.

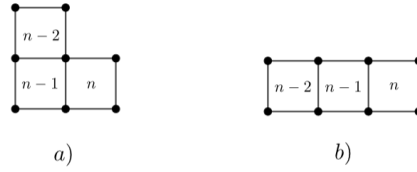


Figure 2.20: The graphs of the $PC1_n(n-2, n)$ depending on whether the first square changes of maintains the direction of the chain, respectively, when read under the reverse interpretation.

So suppose $b_{k+1} = SC$, then because an action for $k \geq 4$ completely determines (up to isomorphism) its impact neighborhood, we have the graph of $PC2_n(k-2, k+1)$ equals one of the graphs shown in Figure 2.21.

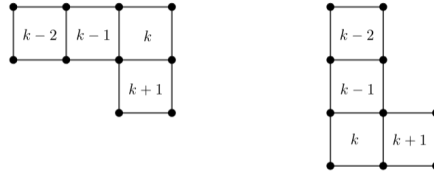


Figure 2.21: The graphs of the possible neighborhoods when $b_{k+1} = SC$.

But then, because by the induction hypothesis $PC2_n(1, k) \cong PC1_n(n-k+1, n)$, it follows the first three squares of $PC1_n(n-k+1, n)$ correspond to one of the graphs shown in Figure 2.22.

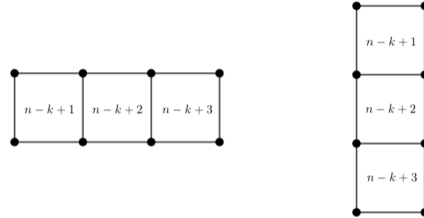


Figure 2.22: The possible graphs for the first three squares of $PC1(n-k+1, n)$ under the assumption $b_{k+1} = SC$.

Applying the same argument as in case 1), since action b_{k+1} changed the direction of $PC2_n(1, k)$, it must be the $(n-k)^{th}$ square of $PC1_n$ also changes the direction of $PC1_n(n-k+1, n)$ when read under the inverse interpretation. Thus the first four squares of $PC1_n(n-k, n)$ correspond to one of the graphs shown in Figure 2.23, where in both cases we have $a_{(n-k+3)} = CS$ and therefore $b_{k+1} = SC = (CS)^F = (a_{(n-(k+1)+4)})^F = a_{(n-k+3)}^F$.

The other cases follow similar arguments and in all of them we conclude $B = A$.

This finishes the proof of the first implication.

For the reverse implication of the statement, it is clear if $(T, SS^*, b_3, \dots, b_n) = (T, SS^*, a_3, \dots, a_n)$, then $PC1_n \cong PC2_n$ since they would be the exact same polyomino chain. It only remains to prove it for the case $(T, SS^*, b_3, \dots, b_n) = (T, SS^*, a_3, \dots, a_n)^F$.

In order to this, in light of the first implication of the theorem, it is sufficient to show there exists a polyomino chain PC_n such that there is an isomorphism between $PC1_n$ and PC_n by

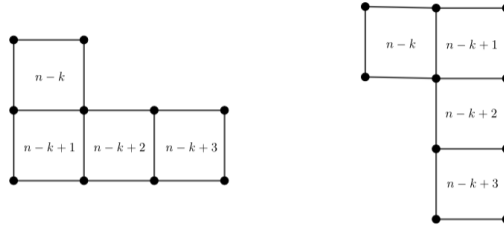


Figure 2.23: Possible graphs for $PC1_n(n-k, n)$.

sending the first square of the former to the last of the latter. But notice we can obtain this polyomino chain PC_n by performing rigid transformations on $PC1_n$. In particular, if the instruction corresponding to the n^{th} square of $PC1_n$ is D , we can transform $PC1_n$ into a polyomino chain starting at its n^{th} square by rotating it 90° clockwise and then reflecting it horizontally (for a graphic representation see Figure 2.24 in which this case corresponds to transforming $a)$ onto $c)$). Similarly, if the instruction corresponding to the n^{th} square of $PC1_n$ is R , we can transform $PC1_n$ into a polyomino chain starting at its n^{th} square by reflecting it vertically and the horizontally (for a graphic representation see Figure 2.24 in which this case corresponds to transforming $b)$ onto $c)$).

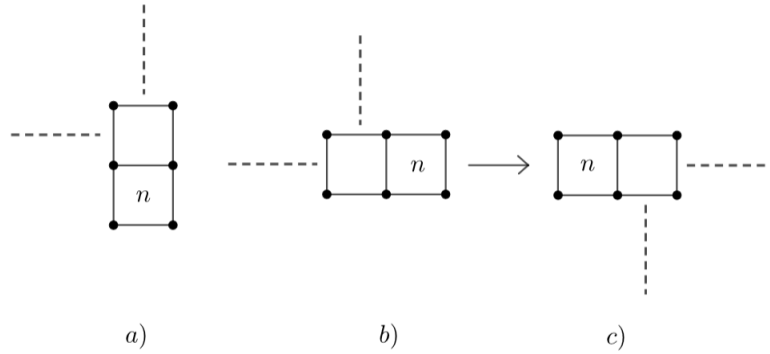


Figure 2.24: Transforming $PC1_n$ into the desired PC_n .

It is then clear in either case the desired polyomino chain PC_n exists and by case 2) of the first implication of this theorem, it follows the action sequence of PC_n is equal to $(T, SS^*, a_3, \dots, a_n)^F$. But then, as action sequences represent a unique polyomino chain (Remark 2.1.10), we have $PC2_n \cong PC_n \cong PC1_n$.

□

As a direct consequence of Theorem 2.5.7 and the fact topological degree-based indices are in particular graph invariants (Remark 1.1.2), the following corollary is immediate.

Corollary 2.5.8. *Let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence, then*

$$TI_f((T, SS^*, a_3, \dots, a_n)^F) = TI_f((T, SS^*, a_3, \dots, a_n)).$$

2.5.2. How Many Non-Isomorphic Polyomino Chains of n Squares are there?

Now that we have characterized through Theorem 2.5.7 the isomorphisms between polyomino chains, a first natural question concerning polyomino chains in general is how many non-isomorphic

polyomino chains of n squares are there. As a first guess, one would say their amount is $\frac{2^{n-2}}{2}$, since there are 2^{n-2} distinct action sequences with n squares having the first two fixed and for each of them we have they are isomorphic to their flip action sequence. However, this assumes $(T, SS^*, a_3, \dots, a_n) \neq (T, SS^*, a_3, \dots, a_n)^F$, which isn't the case as Figure 2.25 shows.

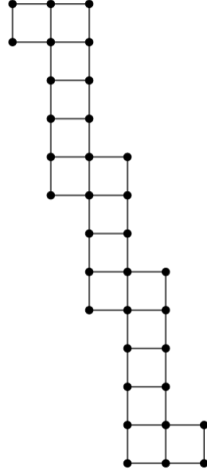


Figure 2.25: The polyomino chain corresponding to the action sequence $(T, SS^*, SC^*, CS, SS, SS, SC, CC, CS, SS, SC, CC, CS, SS, SS, SC)$, which is equal to its flip.

Definition 2.5.9. Let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence. We'll say the action sequence is symmetric if and only if

$$(T, SS^*, a_3, \dots, a_n) = (T, SS^*, a_3, \dots, a_n)^F.$$

Remark 2.5.10. As important special cases, notice the linear and zig-zag polyomino chains are symmetric.

Taking into account the possibility of symmetric polyomino chains, it is straightforward to deduce if Y_n is the set of all symmetric polyomino chains with n squares, then the number of non-isomorphic polyomino chains with n squares will be $|Y_n| + \frac{2^{n-2} - |Y_n|}{2} = \frac{|Y_n|}{2} + 2^{n-3}$. In the next theorem we go on and compute $|Y_n|$ for every $n \geq 1$.

Theorem 2.5.11. For $n \geq 1$, $|Y_n| = 2^{\lfloor \frac{n-3}{2} \rfloor + 1}$.

Proof. The result is easily verified for $n \leq 3$. We'll now divide the proof by cases depending on the parity of n .

a) Assume n is odd, with $n \geq 5$ and let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence.

In this case we have $\lfloor \frac{n-3}{2} \rfloor = \frac{n-3}{2}$ and by Definition 2.5.9 we have if $(T, SS^*, a_3, \dots, a_n)$ is to be symmetric, then it must be

$$a_{(3+\frac{n-3}{2}+1)}, \dots, a_n = a_{(3+\frac{n-3}{2})}^F, \dots, a_4^F$$

It would only remain to impose the last action ended at the same state as the third, but this is automatically true by the fact our previous condition guarantees the last action ends at the same state as the one the fourth comes from.

Thus, since we have two possible choices for the block of the first three actions, namely (T, SS^*, SS) or (T, SS^*, SC^*) , and since we can freely chose for each of the actions $a_4, \dots, a_{(3+\frac{n-3}{2})}$ two possible values, it follows there are

$$2 * 2^{(3+\frac{n-3}{2})-4+1} = 2^{\frac{n-1}{2}} = 2^{\lfloor \frac{n-3}{2} \rfloor + 1}$$

symmetric polyomino chains with n squares.

b) Assume n is even, with $n \geq 4$ and let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence.

In this case we have $\lfloor \frac{n-3}{2} \rfloor = \frac{n-4}{2}$ and by Definition 2.5.9 we have if $(T, SS^*, a_3, \dots, a_n)$ is to be symmetric, then it must be

$$a_{(3+\frac{n-4}{2}+2)}, \dots, a_n = a_{(3+\frac{n-4}{2})}^F, \dots, a_4^F$$

where $a_{(3+\frac{n-4}{2}+1)}$ must be equal to either SS or CC depending on the state action $a_{(3+\frac{n-4}{2})}$ ended at.

It would only remain to impose the last action ended at the same state as the third. If $n = 4$, we have the action sequence looks as follows $(T, SS^*, a_3, a_{(3+\frac{n-4}{2}+1)})$ and by the condition imposed on action $a_{(3+\frac{n-4}{2}+1)}$, it follows the state the last action ends at is the same the state the third does. Finally, if $n > 4$, the excat same argument as in a) yields the desired result.

Thus, since we have two possible choices for the block of the first three actions, namely (T, SS^*, SS) or (T, SS^*, SC^*) , and since we can freely chose for each of the actions $a_4, \dots, a_{(3+\frac{n-4}{2})}$ two possible values, it follows there are

$$2 * 2^{(3+\frac{n-4}{2})-4+1} = 2^{\frac{n-2}{2}} = 2^{\lfloor \frac{n-3}{2} \rfloor + 1}$$

symmetric polyomino chains with n squares.

□

Corollary 2.5.12. *For $n \geq 2$ the number of non-isomorphic polyomino chains is*

$$2^{\lfloor \frac{n-3}{2} \rfloor} + 2^{n-3}$$

2.5.3. Extremal Isomorphic Polyomino Chains

Coming back to the problem of refining the set of all extremal polyomino chains, notice that by Theorem 2.5.7, we now have the means to find all non-isomorphic extremal polyomino chains with n squares. To do this, it is sufficient to compare each of the extremal action sequences, found through the construction method of $n_t(n, E)$, with all of the others and eliminate their flip action sequences (in the case the action sequence was not symmetric). Nevertheless, it is important to consider that this methodology has a time complexity of $N^2 \times n$; where N is the amount of all extremal action sequences and n the number of squares. Thus, the established procedure leaves room for the search of a more efficient approach and it remains relevant to further study the behavior of extremal action sequences and their relation with isomorphisms.

To begin our exploration of the interaction between isomorphisms and extremal action sequences, notice first of all that if an action sequence is maximal, that is, corresponds to a problem $M_f(n, E)$, it need not be the case its flip action sequence is also maximal. However, we can place conditions such that we are able to attribute the flip action sequence the same properties.

Lemma 2.5.13. *If $(T, SS^*, a_3, \dots, a_n)$ is an extremal action sequence, then $(T, SS^*, a_3, \dots, a_n)^F$ is also an extremal action sequence.*

Proof. Because $(T, SS^*, a_3, \dots, a_n)$ is an extremal action sequence, it follows its TI_f value must equal that of $M_f(n)$. But then, by Corollary 2.5.8, the TI_f value of its flip action sequence must also be equal to $M_f(n)$. $\square$

Corollary 2.5.14. *Let $M_f(n) = M_f(n, S) = M_f(n, C)$. If $(T, SS^*, a_3, \dots, a_n)$ is a maximal action sequence, then $(T, SS^*, a_3, \dots, a_n)^F$ is also a maximal action sequence.*

Proof. Let $(T, SS^*, a_3, \dots, a_n)$ be a maximal action sequence corresponding to problem $M_f(n, E)$. Because $M_f(n) = M_f(n, E)$, it follows that the sequence is not only maximal but extremal and by Lemma 2.5.13 so is $(T, SS^*, a_3, \dots, a_n)^F$ which means it must correspond to a maximal action sequence of $M_f(n, S)$ or $M_f(n, C)$. $\square$

Remark 2.5.15. Observe, although Corollary 2.5.14 guarantees the preservation of maximality for flip action sequences, it does not guarantee maximal action sequences for one problem have their flip action sequences maximal for the same problem, thus making isomorphisms mix between states. In the next result we give conditions to avoid this mixture.

Corollary 2.5.16. *Let $M_f(n) = M_f(n, E)$ for some $E \in \{S, C\}$ without a tie. If $(T, SS^*, a_3, \dots, a_n)$ is a maximal action sequence corresponding to problem $M_f(n, E)$, then $(T, SS^*, a_3, \dots, a_n)^F$ is also a maximal action sequence corresponding to problem $M_f(n, E)$.*

Proof. Assume without loss of generality that $E = S$, then the proof is exactly the same as the one for Corollary 2.5.14 with the exception it now can't be $(T, SS^*, a_3, \dots, a_n)^F$ corresponds to a maximal action sequence for problem $M_f(n, C)$ since these are not extremal. $\square$

Remark 2.5.17. Notice as a direct consequence of Corollary 2.5.16, if $E = S$, then not only do all maximal action sequences of problem $M_f(n, S)$ have their flip action sequences also maximal for the same problem, but they all need to have their first three actions equal to T, SS^*, SS . Moreover, observe that the flip action sequences of maximal action sequences of problem $M_f(n, C)$ need not be maximal.

Similarly, if $E = C$, then all maximal action sequences of problem $M_f(n, C)$ must have their first three actions equal to T, SS^*, SC^* and the flip action sequences of maximal action sequences of problem $M_f(n, S)$ need not be maximal.

We now continue by showing one result connecting isomorphisms and the value of the problem $n_t(n, E)$.

Proposition 2.5.18. *If $M_f(n) = M_f(n, E)$ for some $E \in \{S, C\}$ without ties and $n_t(n, E) = 0$, then the unique extremal action sequence is symmetric.*

Proof. By Theorem 2.3.3 we know there exist a unique maximal action sequence for problem $M_f(n, E)$, which will also be the unique extremal action sequence because $M_f(n) = M_f(n, E)$ without ties. Moreover, by Corollary 2.5.16 it follows its flip action sequence is also a maximal action sequence for problem $M_f(n, E)$ and since there only exists one for that problem, it must be the unique extremal action sequence is symmetric. $\square$

As an example of a surprising application of Proposition 2.5.18, we have the following Corollary which in particular applies for $n = 4$ in the proof of Theorem 2.2.14 that dealt with the maximization of the AZI . The advantage now is we can not only obtain $M_f(4, S) = M_f(4, C)$ without the use of direct computation but we can do it even without explicit mention of f or g , depending now only on having the structure given by the tie graph.

Corollary 2.5.19. *If for $n \geq 1$ $n_t(n, E) = 0$ for both $E = S, C$ and the unique action sequences for problems $M_f(n, S)$ and $M_f(n, C)$ are not symmetric, then $M_f(n, S) = M_f(n, C)$.*

Finally, in the light of Proposition 2.5.18, it is important to point out that the linear and zig-zag chains are far more special beyond the fact they used actions that could be successively repeated, which was the reason they emerged as important extremal polyomino chains in Section 2.4. We now notice part of the reason they emerge so often as extremal is that they are symmetric, which is something necessary in order to be the unique extremal polyomino chains for a given index and some fixed number n of squares. Furthermore, in the next theorem we show if a polyomino chain is to be the unique extremal chain for all values $n \in \{5, \dots, k\}$, then it must be either the linear or the zig-zag chain, which makes the dynamic established on Theorem 2.4.2 become even more special.

Theorem 2.5.20. *Let $(T, SS^*, a_3, \dots, a_n)$ be an action sequence such that for every $5 \leq k \leq n$, $(T, SS^*, a_3, \dots, a_k)$ is the unique extremal action sequence. Then*

$$(T, SS^*, a_3, \dots, a_n) = (T, SS^*, SS, SS, \dots, SS)$$

or

$$(T, SS^*, a_3, \dots, a_n) = (T, SS^*, SC^*, CC, \dots, CC)$$

Proof. Let $5 \leq k \leq n$, because $(T, SS^*, a_3, a_4, \dots, a_k)$ is the unique extremal action sequence with k squares, it follows by Proposition 2.5.18 that the action sequence is symmetric. Then, by definition we have $a_k = a_4^F$. As a consequence, we can conclude the action sequence $(T, SS^*, a_3, a_4, \dots, a_n)$ is of the form

$$(T, SS^*, a_3, a_4, a_4^F, \dots, a_4^F)$$

It is only left to show a_4 is equal to SS or CC . But this follows from the fact the sequence in particular is an action sequence and hence the ending state of a_4 must equal the state action a_5 comes from, which we know is equal to the state a_4 ended. Also notice that the fact the first three actions are equal to T, SS^*, SS or T, SS^*, SC^* , in each respective case, is again true because the sequence is an action sequence.

□

Chapter 3

Extremal General Polyomino Chains

Based on the results of our work *Extremal degree-based indices of general polyomino chains via dynamic programming* [21], which is already submitted in a peer reviewed journal, in this chapter we focus on the general problem of finding extremal chains for any degree-based topological index in the complete setting of general polyomino chains. In order to do so, we will study the generalization of actions and the generalization of the dynamic programming approach, which were key elements in the study of extremal chains in the restricted polyomino chain setting. However, in dealing with general polyomino chains there is an important new complication which comes from the fact encodings need not be realizable and that general sequences of actions could have many distinct interpretations, not necessarily representing a unique graph. As a consequence, general sequences of actions are not by themselves enough to produce extremal chains efficiently.

To overcome the aforementioned difficulty, in this chapter we develop a series of algorithms that allow us to transform a general sequence of actions into another general sequence of actions which corresponds to a general polyomino chain and has the same TI_f value. With this transformation process, we then continue to truly generalize the dynamic programming approach from which we are able to obtain, for any fixed number of squares n , a general polyomino chain that maximizes or minimizes the given topological index in linear time with respect to n .

Furthermore, by means of the generalized dynamic programming design we prove, similarly as in the restricted case, that it is possible to obtain all extremal general action sequences in linear time with respect to their amount and that it is possible to obtain all extremal general polyomino chains (possibly isomorphic) through an exhaustive search over their interpretations. Also, to make the problem of interpretations more manageable, we give an important result concerning the interpretations of a special family of general action sequences and as an application we characterize all maximum general polyomino chains of the generalized Randić index with parameter -1 , which was until now an open problem.

Finally, because of the additional complexity of working with general polyomino chains, we give sufficient conditions so that the extremal chains of a given degree-based topological index can all be found by studying action sequences and sufficient conditions so that the use of general action sequences is necessary.

3.1. General Polyomino Chains and General Actions

3.1.1. General Actions and the Problem of Realizability

Motivated by the effectiveness of the encoding of polyomino chains through sequences of actions, we wonder if the encoding can be generalized to general polyomino chains. Taking advantage once again of the concept of neighborhoods of impact (Definition 2.1.1), we can observe that due to the use of instructions $\{R, L, D, U\}$ it is now possible, differently as in the restricted case, that the subgraph corresponding to the neighborhood of impact produced by the last square takes vertices out of the last five. Nevertheless, exhaustively checking the neighborhoods of impact produced by the last squares of all possible general polyomino chains with five squares, it turns out there exists only one new neighborhood of impact (see Figure 3.1) in addition to those shown in Figure 2.2.

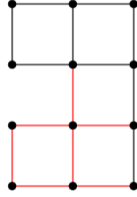


Figure 3.1: The new neighborhood of impact in general polyomino chains.

Notice that in accordance to the notion of change of direction, in this case it is also true that the new neighborhood of impact, which ends up changing the direction of the chain, is isomorphic to the neighborhoods of impact of actions SC and CC , which were also transitions based on change. Thus the new action naturally enters in one of our previous categories based on states. However, observe that although the new action has $E_1 = S$ and $E_2 = C$, it isn't equal to SC and therefore we'll choose to name it TT in reference to the shape similar to a tight turn of its neighborhood.

Below we calculate the modification in the value of the topological index produced by the use of action TT .

$$\begin{aligned}
 TT : & -f(3,4) - f(3,3) - f(2,3) - f(2,2) \\
 & + f(4,4) + f(3,4) + f(4,3) + f(2,3) \\
 & + f(2,4) + f(2,3) + f(2,2) \\
 = & f(2,3) + f(2,4) + f(4,3) + f(4,4) - f(3,3).
 \end{aligned}$$

Finally, because the neighborhoods of impact for $n \geq 4$ are given by actions SS, CS, SC, CC and TT , and since the neighborhoods of impact for $n \leq 3$ remain given by the special actions T, SS^* and SC^* , we can generalize the encoding of sequences of actions to general polyomino chains.

Definition 3.1.1. Let $I \in \{R, L, D, U\}$ be an instruction, we then refer to the instruction with the same direction as I but different orientation, as I 's inverse instruction and we denote it by $\bar{I}$.

$$\begin{aligned}
 R & \longleftrightarrow L \\
 D & \longleftrightarrow U
 \end{aligned}$$

Definition 3.1.2. We will refer to the set $\{SS, SC, CS, CC, TT\}$ as *the set of general actions* of general polyomino chains and to the set $\{T, SS^*, SC^*\}$ as *the set of special actions* of general polyomino chains; where in particular, we will refer to the elements of any of the two sets as *general actions* and specifically to those in the set of special actions as *special actions*.

Let PC_n be a general polyomino chain with n squares and instruction sequence $(R, R, I_3, \dots, I_n)$; $I_k \in \{R, L, D, U\}$. We will define the *general action sequence of the general polyomino chain* PC_n to be a sequence $(a_1, a_2, a_3, \dots, a_n)$ where a_k is a general action and satisfies the following:

- $a_1 = T, a_2 = SS^*$.
- if $k = 3$ and $I_3 = D$ or $I_3 = U$, then $a_3 = SC^*$.
- Otherwise:
If $n \geq 4$, I_{k-2} differs in direction from I_{k-3} , $I_{k-1} = I_{k-2}$ and $I_k = \overline{I_{k-3}}$, then $a_k = TT$

Else:

$$\begin{aligned} & \text{If } I_{k-2} = I_{k-1} \text{ and } I_{k-1} = I_k; & a_k &= SS \\ & \text{if } I_{k-2} = I_{k-1} \text{ and } I_{k-1} \text{ differs in direction from } I_k; & a_k &= SC \\ & \text{if } I_{k-2} \text{ differs in direction from } I_{k-1} \text{ and } I_{k-1} = I_k; & a_k &= CS \\ & \text{if } I_{k-2} \text{ differs in direction from } I_{k-1} \text{ and } I_k = I_{k-2}; & a_k &= CC \end{aligned}$$

Also, if $(a_1, a_2, a_3, \dots, a_n)$ is the general action sequence associated to PC_n , we can refer to PC_n as $PC(a_1, a_2, a_3, \dots, a_n)$.

Remark 3.1.3. Notice that in Definition 3.1.2 the patterns displayed for the translation of blocks of three consecutive instructions do not cover the 4^3 possible arrangements. However, since we know for a fact the sequence of instructions corresponds to a general polyomino chain, some arrangements of instructions are automatically impossible to find. In particular, if I_m, I_{m+1} and I_{m+2} are three consecutive instructions, it is immediate to see it can't be $I_{m+1} = \overline{I_m}$ or that I_{m+1} differs in direction from I_m and $I_{m+2} = \overline{I_m}$. Thus Definition 3.1.2 covers all possible cases.

For example, the general action sequence corresponding to the instruction sequence $(R, R, D, D, L, D, L, L, L, U, U, U, R, R, D)$ is:

$$(T, SS^*, SC^*, CS, TT, CC, CC, CS, SS, SS, SC, CS, SS, SC, CS, TT)$$

where it is important to emphasize that the instruction sequence corresponded to a general polyomino chain.

Now, in the same way we established a recursive construction for polyomino chains, we can similarly obtain the generalization of Lemma 2.1.7 for general polyomino chains.

Lemma 3.1.4. *Let TI_f be a degree-based index and let $(T, SS^*, a_3, \dots, a_n)$ be the general action sequence of the general polyomino chain PC_n . Then*

$$TI_f(PC(T, SS^*, a_3, \dots, a_n)) = TI_f(PC(T, SS^*, a_3, \dots, a_{n-1})) + g_f(a_n) \quad (3.1)$$

where

$$g_f(TT) = f(2, 3) + f(2, 4) + f(4, 3) + f(4, 4) - f(3, 3)$$

$$\begin{aligned} g_f(T) &= 4f(2, 2) & g_f(SC^*) &= 2f(2, 4) + 2f(3, 4) - f(3, 3) \\ g_f(SS^*) &= f(3, 3) + 4f(2, 3) - 2f(2, 2) & g_f(SC) &= 3f(3, 4) + f(2, 4) + f(2, 3) - 2f(3, 3) \\ g_f(SS) &= 3f(3, 3) & g_f(CC) &= f(4, 4) + 2f(2, 4) \\ g_f(CS) &= f(3, 4) - f(2, 4) + f(2, 3) + 2f(3, 3). \end{aligned}$$

Remark 3.1.5. Notice we can write the effect of the general action TT in terms of the others as

$$g_f(TT) = g_f(CS) + g_f(CC) - g_f(SS).$$

Continuing with our generalization, recall in the case of polyomino chains what was important of Lemma 3.1.4 wasn't its use over polyomino chains but over action sequences as independent objects. Thus, we must now define a sense of general action sequence independent of previously having a general polyomino chain, so that we can then establish a dynamic programming framework that optimizes the TI_f value over the set of all general action sequences.

Definition 3.1.6. We will say a sequence of the form $(a_1, a_2, a_3, \dots, a_n)$ is a *general action sequence* with n squares if a_k is a general action and the sequence satisfies:

- $a_1 = T, a_2 = SS^*$ and $a_3 \in \{SS, SC^*\}$
- For $k \geq 4$, $a_k \in \{SS, SC, CS, CC, TT\}$ and if $a_k = E_1^k E_2^k$, then $E_2^{k-1} = E_1^k$.
Also, if $a_k = TT$, then

$$a_{k-2} = SC \text{ and } a_{k-1} = CS$$

or

$$a_{k-2} = CC \text{ and } a_{k-1} = CS.$$

Remark 3.1.7. Notice in Definition 3.1.6 we ask actions TT must be preceded either by actions SC and CS or by actions CC and CS because although action TT comes from state S , the sequences (SS, TT) and (TT, CS, TT) cannot occur, either because it would not make sense or because it would automatically make the sequence not correspond to a general polyomino chain.

Taking into account the limitation present in the use of action TT , discussed in Remark 3.1.7, we now give an alternative encoding of general actions which will homogenize that every general action that comes from a certain state, can follow any other that ends in it.

Definition 3.1.8. Let ST^*, ST and CT represent the triads of actions (SC^*, CS, TT) , (SC, CS, TT) and (CC, CS, TT) respectively; where the states they start and end at are defined naturally. We will refer to the set $\{SS, SC, CS, CC, ST, CT\}$ as *the set of compressed actions* of general polyomino chains and to the set $\{T, SS^*, SC^*, ST^*\}$ as *the set of compressed special actions* of general polyomino chains; where in particular, we will refer to the elements of any of the two sets as *compressed actions*.

We will say a sequence of the form $(b_1, b_2, b_3, \dots, b_m)$ is a *compressed action sequence* if b_k is a compressed action and the sequence satisfies:

- $b_1 = T, b_2 = SS^*$ and $b_3 \in \{SS, SC^*, ST^*\}$
- For $k \geq 4$, $b_k \in \{SS, SC, CS, CC, ST, CT\}$ and if $b_k = E_1^k E_2^k$, then $E_2^{k-1} = E_1^k$.

Also, we say a compressed action sequence has n squares if the general action sequence obtained by substituting the compressed actions ST^*, ST and CT for the triads they represent, has n general actions.

Remark 3.1.9. Observe Lemma 3.1.4 can be also applied to compressed action sequences corresponding to general polyomino chains by simply stating:

$$\begin{aligned} g_f(ST^*) &= g_f(SC^*) + g_f(CS) + g_f(TT) \\ g_f(ST) &= g_f(SC) + g_f(CS) + g_f(TT) \\ g_f(CT) &= g_f(CC) + g_f(CS) + g_f(TT). \end{aligned}$$

It would now only remain to show, in the same way as it happened with action sequences, that general action sequences correspond to general polyomino chains. But as it turns out, just as sequences of instructions over the set $\{R, L, D, U\}$ are not necessarily realizable, so do general action sequences not necessarily correspond to a general polyomino chain. Moreover, general action sequences need not necessarily represent a unique sequence of instructions.

Definition 3.1.10. Let $(a_1, a_2, \dots, a_n)$ be a general action sequence and $(R, R, I_3, \dots, I_n)$ an instruction sequence. If the general action sequence corresponding to $(R, R, I_3, \dots, I_n)$ is $(a_1, a_2, \dots, a_n)$, we will say the instruction sequence is an *interpretation* of the general action sequence.

In particular, if for a general action sequence there exists an interpretation which corresponds to a general polyomino chain, we will say the sequence is *realizable*. Otherwise, we say the sequence is non-realizable.

Remark 3.1.11. In contrast to action sequences which had a one to one correspondence with polyomino chains, there exist general action sequences that are non-realizable, as is the case of $(T, SS^*, SS, SC, CS, SS, SS, SC, CS, TT, CS, SC, CS, SS, SC, CS, TT, CS, SS, SS)$ and general action sequences with multiple interpretations corresponding to general polyomino chains, as is the case of $(T, SS^*, SS, SS, SS, SS, SC, CS, SS, SS, SS, SC, CS, SS, SS, SC, CS, SS, SC, CS, TT)$ for which we show two interpretations in Figure 3.2.

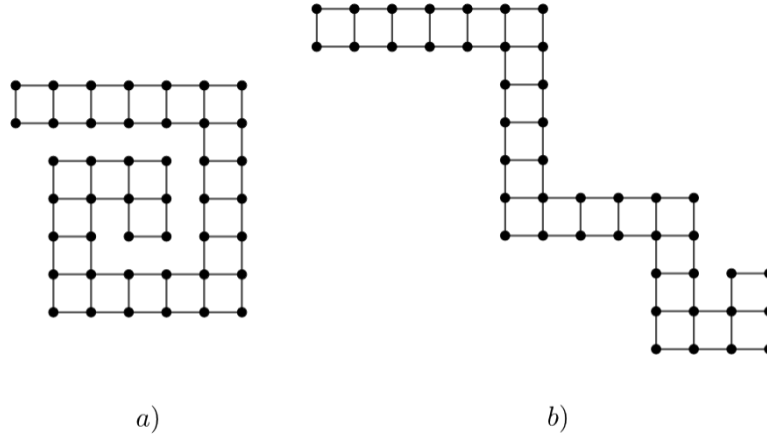


Figure 3.2: a) the general polyomino chain corresponding to the instruction sequence $(R, R, R, R, R, R, D, D, D, D, D, L, L, L, L, U, U, U, R, R, D)$. b) the general polyomino chain corresponding to the instruction sequence $(R, R, R, R, R, R, D, D, D, D, D, R, R, R, R, D, D, D, R, R, U)$.

Nevertheless, even though by the existence of non-realizable general action sequences we are not allowed to say the computation of $TI_f((a_1, a_2, \dots, a_n))$ equals the value of the topological index, at least we know if the sequence was realizable it would coincide with the topological index value of any of the general polyomino chains the sequence represents. Thus, generalizing the computation of the topological index to general action sequences, we will proceed to generalize the dynamic programming framework despite the fact sequences could be non-realizable and focus on Section 3.2.1 in transforming a general action sequence into another that is realizable and has the same TI_f value.

3.1.2. Connection of General Actions with Links

Before continuing with our study of finding extremal general polyomino chains, it is worth noticing that our current encoding through general action sequences, unlike action sequences, is no longer close to geometric-based representations, such as encodings using links.

In order to make the comparison more explicit, although links have been used in the literature only for polyomino chains, it is straightforward to generalize them to encode general polyomino chains.

Definition 3.1.12. Let PC_n be a general polyomino chain and $(R, R, I_3, \dots, I_n)$ its instruction sequence; $I_k \in \{R, L, D, U\}$. We define its *sequence of links* as a sequence of the form

$$(1, 1, L_3, \dots, L_n); \quad L_k \in \{1, 2, 3\}$$

which satisfies for $k \geq 3$:

1. $L_k = 1$ if $I_k = I_{k-1}$.
2. Otherwise, let $j = \max\{1 < \ell < k-1 : \text{direction of } I_\ell \neq \text{direction of } I_k\}$, whenever this maximum exists.
 - If this maximum does not exist, then $L_k = 2$.
 - If the maximum exists and the orientation of I_k matches that of I_j , then $L_k = 2$.
 - Otherwise, $L_k = 3$.

Also, if $(1, 1, L_3, \dots, L_n)$ is the sequence of links associated to PC_n , we can refer to PC_n as $PC(1, 1, L_3, \dots, L_n)$.

From Definition 3.1.12 it is clear sequences of links for general polyomino chains, as well as any other geometric-based representation, can only yield one instruction sequence by performing the translation process in reverse. On the other hand, general action sequences can now represent possibly many general chains, which is a result of them capturing only local geometric information and thus not being able to uniquely determine how each of the pieces glue together.

Moreover, results such as Lemma 3.1.4 are not as immediate to write in terms of links, where we must now use triads of links (instead of pairs as it happened for Lemma 2.1.7) in order to try to capture the way the TI_f changes. The reason for this is that many structures of links, although geometrically distinct, locally behave the same.

Lemma 3.1.13. Let TI_f be a degree-based index, and let $(1, 1, L_3, \dots, L_n)$ be the sequence of links of a general polyomino chain PC_n with $n \geq 4$. Then

$$TI_f(PC(1, 1, L_3, \dots, L_n)) = TI_f(PC(1, 1, L_3, \dots, L_{n-1})) + g_f(L_{n-2}, L_{n-1}, L_n),$$

where

$$g_f(L_{n-2}, L_{n-1}, L_n) = \begin{cases} 3f(3, 3), & (L_{n-2}, L_{n-1}, L_n) = (i, 1, 1), \\ 3f(3, 4) + f(2, 4) + f(2, 3) - 2f(3, 3), & (L_{n-2}, L_{n-1}, L_n) = (i, 1, 2) \text{ or } (1, 1, 3), \\ f(3, 4) - f(2, 4) + f(2, 3) + 2f(3, 3), & (L_{n-2}, L_{n-1}, L_n) = (i, 2, 1) \text{ or } (1, 3, 1), \\ f(4, 4) + 2f(2, 4), & (L_{n-2}, L_{n-1}, L_n) = (i, 2, 2) \text{ or } (1, 3, 2), \\ f(2, 3) + f(2, 4) + f(3, 4) + f(4, 4) - f(3, 3), & (L_{n-2}, L_{n-1}, L_n) = (2, 1, 3) \text{ or } (3, 1, 3), \end{cases}$$

with $i \in \{1, 2, 3\}$.

Therefore, we can conclude that locally geometric-based encodings of general polyomino chains present an inherent advantage in studying the value of degree-based indices, which will be further emphasized by the compact dynamic programming approach achieved in Section 3.2.2.

3.2. Generating One Extremal General Polyomino Chain

3.2.1. Making a General Action Sequence Realizable

Based on Section 3.1.1, optimizing degree-based indices over general polyomino chains could be achieved by working with general action sequences, as they encapsulate the essential information

required for computing such indices. However, in order to do so we must first solve the main challenge of this approach which lies in guaranteeing that we can obtain realizable general action sequences.

Notation. Let $(a_1, a_2, \dots, a_N)$ be a sequence of compressed actions. We define the cumulative number of direction changes up to and including a_i by

$$m_0 = 0, \quad m_i = m_{i-1} + \mathbb{I}_{\{a_i \in \{SC, CC, SC^*\}\}} + 2\mathbb{I}_{\{a_i \in \{ST, CT, ST^*\}\}}, \quad i \geq 1.$$

Each action a_i , where $i \geq 2$, consists of two letters; we denote by $F_i \in \{S, C\}$ its first letter and by $S_i \in \{S, C, T\}$ its second letter. These variables will be used in the construction and validation steps below.

Algorithm 1: Parity correction for tight turns (compressed actions)

Input: Compressed action sequence $(a_1, \dots, a_N)$
Output: Modified compressed action sequence $(a_1, \dots, a_N)$

```

1  $m \leftarrow 0$ 
2 for  $i = 2$  to  $N$  do
3   if  $a_i \in \{SC, CC, SC^*\}$  then
4      $m \leftarrow m + 1$ 
5   if  $a_i \in \{ST, CT, ST^*\}$  then
6      $m \leftarrow m + 2$ 
7   if  $S_i = T$  and  $m$  is odd then
8      $j \leftarrow \max\{1, \dots, i-1 : S_j = C\}$ 
9      $a_j \leftarrow F_j T; \quad a_i \leftarrow F_i C$ 

```

Lemma 3.2.1. *Let $(a_1, \dots, a_N)$ be a sequence of compressed actions. For every $i \in \{2, \dots, N\}$, after completing iteration i of Algorithm 1, the following invariants hold:*

- a) *For every $2 \leq j \leq i$ with $S_j = T$, m_j is even.*
- b) *The new sequence is indeed a sequence of compressed actions.*
- c) *The value of TI_f and the ending state of a_N are unchanged.*

Proof. We argue by induction on i .

a) For $i = 2$, if $S_2 = T$ then by definition $m_2 = 2$, hence even. Assume the claim holds up to i , and consider iteration $i + 1$. If $S_{i+1} \neq T$ or m_{i+1} is even, Algorithm 1 does nothing and the inductive hypothesis applies.

Suppose $S_{i+1} = T$ and m_{i+1} is odd. Set

$$\ell = \max\{j \in \{1, \dots, i\} : S_j = C\},$$

which exists because by the definition of m_{i+1} , we have that m_i must also be odd. By maximality of ℓ there is no index $j \in \{\ell+1, \dots, i\}$ with $S_j = C$.

Algorithm 1 updates $a_\ell : F_\ell C \mapsto F_\ell T$ and $a_{i+1} : F_{i+1} T \mapsto F_{i+1} C$. Since m_k changes parity only when passing a C (increments by $+1$) and not when passing a T (increments by $+2$), we have:

- Before the update, the parity of m_ℓ equals that of m_{i+1} , hence m_ℓ is odd.

- After the update, the only prefix up to ℓ whose number of C -endings changes is ℓ itself, decreasing by 1; thus m_ℓ flips to be even.

All other m_j with $j \leq \ell - 1$ are unchanged (which are even by the inductive hypothesis when $S_j = T$), and there is no $j \in \{\ell + 1, \dots, i\}$ with $S_j = T$; indeed, if such j existed, the parity of m_j would coincide with that of m_ℓ (no C between ℓ and j) prior to the update, contradicting the inductive hypothesis (m_j even). Finally, S_{i+1} becomes C , so there is nothing to check at $j = i + 1$. Hence, after iteration $i + 1$, every m_j with $S_j = T$ and $2 \leq j \leq i + 1$ is even.

b) Notice first that, since $(a_1, \dots, a_N)$ is a sequence of compressed actions, it follows that $a_1 = T, a_2 = SS^*$ and $a_3 \in \{SS, SC^*, ST^*\}$. Hence, regardless of the modifications performed by the algorithm, a_1 and a_2 remain unchanged and if a_3 is equal to ST^* or SS , it is also never changed. If $a_3 = SC^*$, it may be changed to ST^* , so in all cases the first three actions of the new compressed sequence still satisfy their required conditions. Moreover, since in the compressed notation we move actions TT together with preceding actions SC, CS or CC, CS , in the new sequence when uncompressed it holds these precede any action TT . It remains to show that consecutive actions are still compatible, that is, the state one compressed action ends at equals the state the next comes from. But for this it is sufficient to prove inductively that when a change is performed, the action following $F_\ell T$ starts with C and that the one, if it exists, following $F_{i+1} C$ starts with C ; but this is immediate from the fact before the change the sequence was a sequence of compressed actions and we had $F_\ell C$ and $F_{i+1} T$.

c) By the local recurrence of Equation (3.1), TI_f is the sum of local contributions $g_f(\cdot)$. Only positions ℓ and $i + 1$ change, so it suffices to verify

$$g_f(F_\ell C) + g_f(F_{i+1} T) = g_f(F_\ell T) + g_f(F_{i+1} C),$$

which follows from the identity $g_f(F T) = g_f(F C) + g_f(CS) + g_f(TT)$ and additivity in Equation (3.1). Therefore, the value of TI_f is unchanged.

Also, since it can only be compressed actions of the form $F_i T$ change to compressed actions of the form $F_i C$, it follows that no matter the case, the ending state of a_i remains unchanged for every $i \in \{1, \dots, N\}$. In particular, the ending state of a_N remains unchanged. $\square$

Before proceeding, note that part a) of Lemma 3.2.1 implicitly guarantees that Algorithm 1 is well defined. To reach our main goal in this section, which is ensuring realizability, we now need a bridge from the sequence of compressed actions produced by Algorithm 1 to a corresponding sequence of instructions. Algorithm 2 provides this translation.

Algorithm 2: General Actions $\rightarrow$ Instructions

Input: General action sequence $(T, SS^*, a_3, \dots, a_n)$

Output: Instruction sequence $(I_1, I_2, \dots, I_n)$

```

1  $j \leftarrow \max\{t \in \{1, \dots, n\} : a_1, \dots, a_t \notin \{SC^*, SC\}\}$ 
2 for  $k = 1$  to  $j$  do  $I_k \leftarrow R$ 
3 if  $j + 1 < n$  then
4    $I_{j+1} \leftarrow D$ 
5 for  $i = j + 2$  to  $n$  do
6    $\ell \leftarrow \max\{3 \leq \lambda \leq i - 1 : a_\lambda \notin \{SS, CS\}\}$ 
7   if  $a_i \in \{SS, CS\}$  then  $I_i \leftarrow I_{i-1}$ 
8   else if  $a_i \in \{SC, CC\}$  then  $I_i \leftarrow I_{\ell-1}$ 
9   else  $I_i \leftarrow \overline{I_{\ell-1}}$ 
```

Since the variable m_i was central for Algorithm 1, it is natural to introduce its analogue for general action sequences. Given the general action sequence $(a_1, a_2, \dots, a_n)$, define for $i \in \{1, \dots, n\}$

$$M_0 := 0, \quad M_i := M_{i-1} + \mathbf{1}_{\{a_i \in \{SC, CC, SC^*, TT\}\}}.$$

Thus M_i counts the number of direction changes up to the i^{th} square.

Because we do not work with an arbitrary general action sequence, it is useful to record the specific properties that Algorithm 1 enforces on the instructions $(I_1, I_2, \dots, I_n)$ produced by Algorithm 2.

Corollary 3.2.2. *Let $(b_1, b_2, \dots, b_N)$ be a compressed action sequence, $(a_1, a_2, \dots, a_n)$ be its uncompressed version after applying Algorithm 1 and $(I_1, I_2, \dots, I_n)$ the sequence of instructions generated by Algorithm 2. Then, for every $i \geq 1$:*

- a) *If M_i is even, the i^{th} square is horizontal. Moreover, if $a_i = TT$ then M_i and M_{i-3} are even.*
- b) *If M_i is odd, the i^{th} square is vertical.*
- c) *$(I_1, I_2, \dots, I_n)$ is an interpretation of $(a_1, a_2, \dots, a_n)$.*

Lemma 3.2.3. *Let $(a_1, a_2, \dots, a_n)$ be the uncompression of the compressed action sequence produced by Algorithm 1, and let $(I_1, \dots, I_n)$ be the instruction sequence obtained via Algorithm 2. Then:*

- a) *For every $i \in \{1, \dots, n\}$ one has $I_i \in \{R, L, D\}$; in particular, U never occurs.*
- b) *None of the following contiguous patterns appears in $(I_1, \dots, I_n)$: LR, RL, LDR, RDL .*

Proof. a) We argue by induction on i . By Algorithm 2, $I_1 = R \in \{R, L, D\}$, so the base case holds. Assume $I_k \in \{R, L, D\}$ for all $k \leq i$ and consider I_{i+1} . There are three possibilities:

1. If $a_{i+1} \in \{SS, CS, SS^*\}$, then $I_{i+1} = I_i \in \{R, L, D\}$ by the induction hypothesis.
2. If $a_{i+1} \in \{SC, CC, SC^*\}$, Algorithm 2 sets $I_{i+1} = D$ or $I_{i+1} = I_{\ell-1}$ where $\ell = \max\{3 \leq \lambda \leq i : a_\lambda \notin \{SS, CS\}\}$. Hence I_{i+1} is one of the previously constructed instructions and belongs to $\{R, L, D\}$.
3. If $a_{i+1} = TT$, because $(a_1, a_2, \dots, a_n)$ is a general action sequence, it follows (a_{i-1}, a_i, a_{i+1}) equals (SC, CS, TT) , (CC, CS, TT) or (SC^*, CS, TT) , so in every case $\ell = \max\{3 \leq \lambda \leq i : a_\lambda \notin \{SS, CS\}\} = i - 1$ and Algorithm 2 gives $I_{i+1} = \overline{I_{\ell-1}} = \overline{I_{i-2}}$. By Corollary 3.2.2 a) applied to $i + 1$, both M_{i+1} and M_{i-2} are even; hence the $(i - 2)^{\text{th}}$ square is horizontal and therefore $I_{i-2} \in \{R, L\}$. Since the inverse of a horizontal instruction is again horizontal, $I_{i+1} \in \{R, L\}$.

In all cases $I_{i+1} \in \{R, L, D\}$, completing the induction and part a).

Before b) we record the following mapping fact, immediate from Algorithm 2, Corollary 3.2.2 and part a) of this lemma: for every $i \geq 1$,

$$\begin{aligned} M_i \text{ even and } a_{i+1} \in \{SS, CS, SS^*\} &\implies I_{i+1} \in \{R, L\}, \\ M_i \text{ even and } a_{i+1} \in \{SC, CC, SC^*\} &\implies I_{i+1} = D, \\ M_i \text{ odd and } a_{i+1} \in \{SS, CS\} &\implies I_{i+1} = D, \\ M_i \text{ odd and } a_{i+1} \in \{SC, CC, TT\} &\implies I_{i+1} \in \{R, L\}. \end{aligned} \tag{3.2}$$

b) First, the pairs (L, R) and (R, L) cannot occur. Indeed, suppose $I_i \in \{R, L\}$. Then either M_{i-1} is even with $a_i \in \{SS, CS, SS^*\}$, or M_{i-1} is odd with $a_i \in \{SC, CC, TT\}$; in both situations M_i is even. By Corollary 3.2.2 a), it can't be $a_{i+1} = TT$. If $a_{i+1} \in \{SS, CS\}$, Algorithm 2 gives $I_{i+1} = I_i$; if $a_{i+1} \in \{SC, CC\}$, then by Equation (3.2) we have $I_{i+1} = D$. Thus I_{i+1} is never the inverse of I_i , ruling out LR and RL .

Next, the triples (L, D, R) and (R, D, L) cannot occur. Assume $I_i \in \{R, L\}$ and $I_{i+1} = D$. By Equation (3.2), this forces $a_{i+1} \in \{SC, CC, SC^*\}$ and hence M_{i+1} is odd. If $a_{i+2} \in \{SS, CS\}$, Algorithm 2 copies the previous instruction and $I_{i+2} = D$; if $a_{i+2} \in \{SC, CC\}$, then the reference index in Algorithm 2 is $\ell = i + 1$, so $I_{i+2} = I_{\ell-1} = I_i$. In neither subcase does I_{i+2} become the inverse of I_i , so LDR and RDL are excluded. □

We are now ready to prove the main theorem of this section. To this end, we introduce the following *safety zones* (see Figure 3.3).

Right-zone(i): This zone consists of all squares that are either above or to the left of the i^{th} square, as well as any square that is located at least two squares above and to the right of the i^{th} square. The i^{th} square itself is also included in this zone.

Down-zone(i): This zone consists of all squares that are above the i^{th} square. The i^{th} square itself is also included in this zone.

Left-zone(i): This zone consists of all squares that are either above or to the right of the i^{th} square, as well as any square that is located at least two squares above and to the left of the i^{th} square. The i^{th} square itself is also included in this zone.

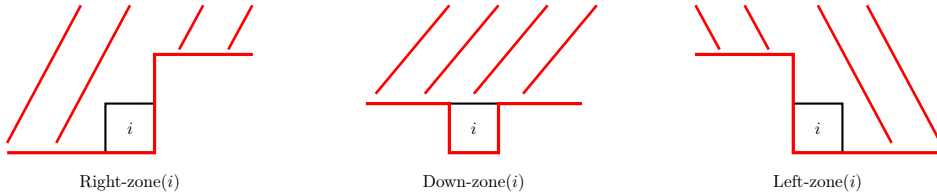


Figure 3.3: Graphical representation of the three different safety zones.

Theorem 3.2.4. *Let $(I_1, I_2, \dots, I_n)$ be the instruction sequence obtained by applying Algorithm 1 followed by Algorithm 2. Then $(I_1, I_2, \dots, I_n)$ is realizable and the degree-based index of the general polyomino chain it represents equals that of the compressed action sequence used to initialize Algorithm 1.*

Proof. We prove its realizability by induction on $i \geq 2$. In view of Lemma 1.2.5, it suffices to show that when the $(i+1)^{th}$ square is appended via I_{i+1} , the two ending vertices have degree 2. In fact, we will maintain the invariant that the first i squares lie entirely inside one of the three safety zones Right-zone(i), Left-zone(i), or Down-zone(i) introduced in Figure 3.3, according to the instruction I_i provided by Algorithm 2.

Base case $i = 2$. The two initial squares are horizontal; hence they lie in Right-zone(2) and the ending vertices of the second square have degree 2.

Inductive step. Assume the claim holds for some $i \geq 2$: the first i squares lie in the safety zone prescribed by I_i , and the ending vertices of the i^{th} square have degree 2. We prove the statement for $i + 1$. By Lemma 3.2.3 a), $I_i \in \{D, R, L\}$, and we split into cases.

Case $I_i = D$. By the induction hypothesis the first i squares lie at or above the row of the i^{th} square. Appending the $(i + 1)^{th}$ square below the i^{th} one places it strictly below all previous squares; hence no previously placed square can meet the two lower (ending) vertices of the new square, and these vertices have degree 2. Moreover, the first $i + 1$ squares lie in $\text{Down-zone}(i + 1)$.

Case $I_i = R$. By Lemma 3.2.3 b), the pair (L, R) is forbidden; hence $I_{i-1} \in \{R, D\}$. In either subcase the induction hypothesis localizes the first i squares in the corresponding safety zone.

Subcase $I_{i-1} = R$. All first i squares lie in $\text{Right-zone}(i)$: as there are no squares among the first i located below the i^{th} square or to its right and less than two squares up, it follows that the ending vertices of the $(i + 1)^{th}$ square must have degree two. Moreover, the first $i + 1$ squares lie in $\text{Right-zone}(i + 1)$. Otherwise, if there were a square outside this zone, it would contradict the assumption that the first i squares lie entirely in $\text{Right-zone}(i)$.

Subcase $I_{i-1} = D$. Here the first i squares lie in $\text{Down-zone}(i)$, and the new square is appended to the right. Thus, there are no earlier squares strictly to the right of the $(i + 1)^{th}$ square, nor directly below it. To force one of the two right (ending) vertices to have degree > 2 one would need an earlier square either immediately above the $(i + 1)^{th}$ square or one unit above and one unit to the right. But Lemma 3.2.3 b) excludes the short zig-zag patterns (L, D, R) and (R, D, L) , and combining this with the induction hypothesis (location of the first i squares) rules out those two possibilities. Hence, the ending vertices have degree 2, and again all $i + 1$ squares lie in $\text{Right-zone}(i + 1)$.

Case $I_i = L$. This is symmetric to the case $I_i = R$, using Left-zone and the forbidden patterns in Lemma 3.2.3 b).

This completes the induction and by Lemma 1.2.5 the instruction sequence is realizable. For the index identity, Corollary 3.2.2 c) shows that the instruction sequence is an interpretation corresponding to the compressed action sequence obtained through Algorithm 1, and Lemma 3.2.1 c) shows that Algorithm 1 preserves TI_f on compressed actions; chaining both equalities yields the second claim. $\square$

3.2.2. Dynamic Programming Approach to Build an Extremal General Polyomino Chain

After overcoming the main challenge of optimizing degree-based indices over general polyomino chains through the use of general actions, this section focuses on the generalization of the dynamic programming framework which we'll define over the set of compressed action sequences due to the fact these can break into pieces and built on top of each other without further considerations.

Let TI_f be a degree-based index, we define $M_f(n, E)$ for $E = S, C$ and $n \geq 1$ as the *maximum value* of this index among all compressed action sequences with n squares and ending at state E . We shall refer to the values $M_f(n, E)$ as problems and to the compressed action sequences that produce them as *maximal compressed action sequences*. Based on the recursive nature stemming from Equation (3.1), the following result provides a dynamic programming formulation for computing $M_f(n, E)$ for all $n \geq 1$ and $E \in \{S, C\}$.

Remark 3.2.5. If there does not exist a compressed action sequence with n squares ending at state E , we will define $M_f(n, E) = -\infty$, which notice happens only for problems $M_f(1, C)$ and $M_f(2, C)$.

Theorem 3.2.6. *Let TI_f be a degree-based index. Then for $n \leq 5$,*

$$\begin{aligned} M(1, S) &= g_f(T) & M(1, C) &= -\infty \\ M(2, S) &= M(1, S) + g_f(SS^*) & M(2, C) &= -\infty \\ M(3, S) &= M(2, S) + g_f(SS) & M(3, C) &= M(2, S) + g_f(SC^*) \end{aligned}$$

$$\begin{aligned} M_f(4, S) &= \max\{M_f(3, S) + g_f(SS), M_f(3, C) + g_f(CS)\}, \\ M_f(4, C) &= \max\{M_f(3, S) + g_f(SC), M_f(3, C) + g_f(CC)\}, \\ M_f(5, S) &= \max\{M_f(4, S) + g_f(SS), M_f(4, C) + g_f(CS)\}, \\ M_f(5, C) &= \max\{M_f(4, S) + g_f(SC), M_f(4, C) + g_f(CC), M_f(2, S) + g_f(ST^*)\}, \end{aligned}$$

And for $n \geq 6$,

$$M_f(n, S) = \max\{M_f(n-1, S) + g_f(SS), M_f(n-1, C) + g_f(CS)\},$$

and

$$\begin{aligned} M_f(n, C) &= \max\{M_f(n-1, S) + g_f(SC), M_f(n-1, C) + g_f(CC), \\ &\quad M_f(n-3, S) + g_f(ST), M_f(n-3, C) + g_f(CT)\}. \end{aligned}$$

Proof. The expressions for $n \leq 5$ are straightforward to check exhaustively, so we'll only prove the expression for problem $M_f(n, C)$; $n \geq 6$. The proof for the expression of $M_f(n, S)$ is completely analogous.

By definition of $M_f(\cdot, \cdot)$, there exist compressed action sequences

$$(a_1^1, \dots, a_{k_1}^1), \quad (a_1^2, \dots, a_{k_2}^2), \quad (a_1^3, \dots, a_{k_3}^3), \quad (a_1^4, \dots, a_{k_4}^4)$$

with $(n-1)$, $(n-1)$, $(n-3)$ and $(n-3)$ squares, respectively, such that

$$\begin{aligned} E_2^{k_1} &= S, & M_f(n-1, S) &= TI_f((a_1^1, \dots, a_{k_1}^1)), \\ E_2^{k_2} &= C, & M_f(n-1, C) &= TI_f((a_1^2, \dots, a_{k_2}^2)), \\ E_2^{k_3} &= S, & M_f(n-3, S) &= TI_f((a_1^3, \dots, a_{k_3}^3)), \\ E_2^{k_4} &= C, & M_f(n-3, C) &= TI_f((a_1^4, \dots, a_{k_4}^4)). \end{aligned}$$

By Equation (3.1), appending an admissible last compressed action yields

$$\begin{aligned} M_f(n-1, S) + g_f(SC) &= TI_f((a_1^1, \dots, a_{k_1}^1, SC)), \\ M_f(n-1, C) + g_f(CC) &= TI_f((a_1^2, \dots, a_{k_2}^2, CC)), \\ M_f(n-3, S) + g_f(ST) &= TI_f((a_1^3, \dots, a_{k_3}^3, ST)), \\ M_f(n-3, C) + g_f(CT) &= TI_f((a_1^4, \dots, a_{k_4}^4, CT)). \end{aligned}$$

and each of the four right-hand sides corresponds to a compressed action sequence with n squares and ending at state C . Since $M_f(n, C)$ is the maximum TI_f among such sequences, we obtain

$$\begin{aligned} M_f(n, C) &\geq \max\{M_f(n-1, S) + g_f(SC), M_f(n-1, C) + g_f(CC), \\ &\quad M_f(n-3, S) + g_f(ST), M_f(n-3, C) + g_f(CT)\}. \end{aligned}$$

For the reverse inequality, let $(a_1, \dots, a_k)$ be a compressed action sequence with n squares, $E_2^k = C$, and $M_f(n, C) = TI_f((a_1, \dots, a_k))$. There are two cases.

Case $S_k = C$. Then $a_k \in \{SC, CC\}$ and by Equation (3.1)

$$M_f(n, C) = TI_f((a_1, \dots, a_{k-1})) + g_f(a_k) \leq \max\{M_f(n-1, S) + g_f(SC), M_f(n-1, C) + g_f(CC)\}.$$

Case $S_k = T$. Then $a_k \in \{ST, CT\}$, which represents the compression of a three-step block; hence, the compressed action sequence $(a_1, \dots, a_{k-1})$ has $(n-3)$ squares. Using Equation (3.1),

$$M_f(n, C) = TI_f((a_1, \dots, a_{k-1})) + g_f(a_k) \leq \max\{M_f(n-3, S) + g_f(ST), M_f(n-3, C) + g_f(CT)\}.$$

Combining both cases yields the desired upper bound and proves the recurrence for $M_f(n, C)$. $\square$

Now observe that, exactly as it happened with action sequences, as a direct application of Theorem 3.2.6, we obtain a method for computing the *minimum value* of any degree-based index among all compressed action sequences with n squares and ending at state E , denoted by $m_f(n, E)$, with $n \geq 1$ and $E \in \{S, C\}$.

Corollary 3.2.7. *Let TI_f be a degree-based index. Then, for $n \geq 1$ and $E \in \{S, C\}$*

$$m_f(n, E) = -M_{-f}(n, E).$$

Remark 3.2.8. Notice that Corollary 3.2.7 would be valid no matter the representation used to establish the dynamic programming approach, since it only depends on the fact degree-based indices are linear.

According to Corollary 3.2.7, we may also restrict our attention in the general polyomino chain setting to the maximization problem. Moreover, it is important to notice that the method introduced in Theorem 3.2.6 not only allows us to determine the maximum value of a fixed degree-based index over all compressed action sequences with n squares and ending at state E , but also provides a constructive way to generate at least one maximal compressed action sequence for problem $M_f(n, E)$ by using the underlying directed graph implicit in the dynamic programming framework. This construction can be performed recursively by selecting a compressed action a , according to the following hierarchy:

$$ST, CT, ST^*, SS, CS, CC, SC, SS^*$$

where this means if for example we are at some problem $M_f(n, E)$ and both expressions $M_f(n-j_1, H_1) + g_f(b_1)$ and $M_f(n-j_2, H_2) + g_f(b_2)$ with $H_1, H_2 \in \{S, C\}$, b_1, b_2 compressed actions and j_1, j_2 dependent of b_1, b_2 , in its *max* argument are equal to $M_f(n, E)$, then we assign $a = b_1$ if and only if b_1 is to the left of b_2 in our hierarchy and we move to problem $M_f(n-j_1, H_1)$.

Finally, if we are at problem $M_f(1, S)$ we take $a = T$. Joining all compressed actions a for each step we obtain a maximal compressed action sequence corresponding to problem $M_f(n, E)$. Hereafter, we refer to this procedure as the *constructive method* induced by the problem $M_f(n, E)$; where it shall be always clear whether we refer to the dynamic programming using only actions or compressed actions.

As an example, in Figure 3.4 we display a possible graph induced by applying Theorem 3.2.6, where each cell denotes a problem $M_f(n, E)$ and each arrow depicts the subproblem it was built from in the recursive process. For instance, performing the constructive method for problem $M_f(11, S)$ we would obtain the maximal compressed action sequence $(T, SS^*, SS, ST, CC, CT, CS)$. However, it is important to highlight maximal compressed action sequences need not be realizable, as is the case in this example. By applying Algorithm 1 to the given maximal compressed action sequence we obtain the new compressed action sequence $(T, SS^*, SS, ST, CT, CC, CS)$ which we are guaranteed by Theorem 3.2.4 is realizable (which we

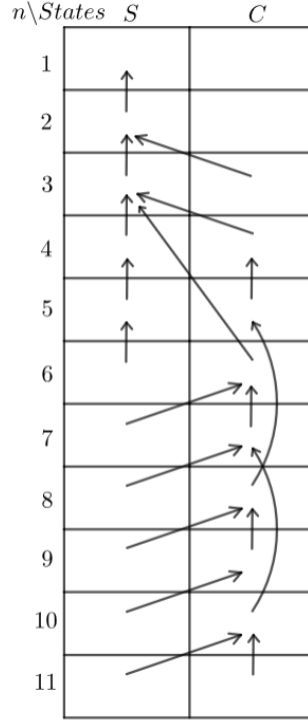


Figure 3.4: An example of a graph induced by performing Theorem 3.2.6.

can concretely verify by obtaining its associated instruction sequence through Algorithm 2) and in this case also corresponds to problem $M_f(11, S)$ since we are guaranteed it preserves the TI_f and ending state.

At this point, it is worth emphasizing that our main objective is to determine the maximum value of a degree-based index among all general polyomino chains with a fixed number of squares, say n . But since any general polyomino chain can be encoded by a compressed action sequence and we can always find maximal compressed action sequences which are realizable, it follows that the quantity of real interest is

$$M_f(n) := \max\{M_f(n, S), M_f(n, C)\}$$

In any case, the problem reduces to analyzing $M_f(n, E)$ and applying , and thus all the previous discussion remains fully applicable.

Remark 3.2.9. Analyzing the computational cost of our proposed method, notice by applying Theorem 3.2.6 and the constructive method, we can find a maximal compressed action sequence with n squares and ending at state S or C in linear time with respect to n . Consequently, we can also find an extremal compressed action sequence with n squares by taking the maximal compressed action sequence of either $M_f(n, S)$ or $M_f(n, C)$ depending on which has a higher value. Finally, the extremal compressed action sequence can be turned into a general polyomino chain having the same TI_f value by applying Algorithms 1 and 2, which run in linear time with respect to n .

Thus, altogether, our method finds an extremal general polyomino chain in linear time with respect to n .

3.3. Counting and Generating All Extremal General Polyomino Chains

3.3.1. Counting and Generating All Extremal General Action Sequences

Continuing with our generalization, we will now proceed to generalize the concept of number of ties to the new dynamic programming approach by compressed action sequences.

For a fixed integer $n \geq 1$ and $E \in \{S, C\}$, we denote by $n_t(n, E)$ the *number of ties* that occur during the recursive computation of $M_f(n, E)$; where these ties represent situations where more than one argument of the *max* expression have the same value, resulting in multiple equally optimal configurations. We define the base cases as: $n_t(i, E) = 0$ for all $i \in \{1, 2, 3\}$ and $E \in \{S, C\}$, while for $n \geq 4$, $n_t(n, E)$ is defined recursively as follows:

- a) If only one expression in the *max* argument of $M_f(n, E)$ is the maximum, say $M_f(n-j, E^*) + g_f(b)$ with $E^* \in \{S, C\}$, b a compressed action and j dependent of b , we define $n_t(n, E)$ as

$$n_t(n, E) = n_t(n-j, E^*)$$

- b) Else let W be equal to the sum of all values $n_t(n-j, E^*)$ such that the expression $M_f(n-j, E^*) + g_f(b)$ is part of the *max* argument of $M_f(n, E)$ and is equal to the maximum. We then define $n_t(n, E)$ as

$$n_t(n, E) = 1 + W$$

Remark 3.3.1. Notice by the way the values $n_t(n, E)$ are defined, we have established a new dynamic programming framework and along it a new directed graph which we shall call the *index tie graph*; where it shall always be clear whether we are using only actions or compressed actions. Furthermore, notice that differently as in the case of actions, here a problem can have up to four edges.

As it turns out, the proofs of the generalizations of Lemma 2.3.2 and Theorem 2.3.3 on compressed action sequences are completely analogous since they did not depend on the actual actions used but on their additivity through Lemma 2.1.7 and the maximality of problems $M_f(n, E)$. However, one must be more careful in dealing with indices, since compressed action sequences differently from action sequences, do not coincide in indexation with their number of squares. Also, the proof complicates in notation because of the many ways problem $n_t(n, E)$ can break down. Thus, for completeness, we at least give the generalized statements, where if $(T, SS^*, a_3, \dots, a_N)$ is a compressed action sequence with n squares, we denote by n_K the number of squares of its subsequence up to the compressed action a_K (in particular we write n instead of n_N).

Lemma 3.3.2. *Let $n \geq 3$, $E \in \{S, C\}$ and $(T, SS^*, a_3, \dots, a_N)$ be a maximal compressed action sequence corresponding to problem $M_f(n, E)$. Then the following statements hold:*

- a) *For all $K \in \{1, \dots, N\}$, it follows*

$$M_f(n_K, E_2^K) = TI_f((T, SS^*, a_3, \dots, a_K)).$$

- b) *There exists at least one distinct maximal compressed action sequence $(T, SS^*, b_3, \dots, b_L)$ for problem $M_f(n, E)$ if and only if*

$$n_t(n_{(K+1)}, E_2^{K+1}) \neq n_t(n_K, E_2^K),$$

for some $K \in \{3, \dots, N-1\}$.

- c) Given $K \in \{3, \dots, N\}$. The maximum number of consecutive compressed actions, counting backward from the last one, that are shared by all maximal compressed action sequences of the problem $M_f(n, E)$ is $n - K$ and these compressed actions are precisely $a_{K+1}, \dots, a_{N-1}, a_N$ if and only if

$$n_t(n, E) = n_t(n_{(N-1)}, E_2^{N-1}) = \dots = n_t(n_K, E_2^K) \neq n_t(n_{(K-1)}, E_2^{K-1}).$$

Theorem 3.3.3. *Let $n \geq 1$ and $E \in \{S, C\}$. Then $n_t(n, E) + 1$ equals the number of distinct maximal compressed action sequences for the problem $M_f(n, E)$, among all compressed action sequences with n squares and ending at state E .*

Based on Theorem 3.3.3 and using the tie graph generated by the dynamic programming framework over problems $n_t(n, E)$, we can now determine all maximal compressed action sequences corresponding to problem $M_f(n, E)$. In order to do this, we similarly define a construction method for $n_t(n, E)$ by recursively selecting a compressed action a , according to the following rules:

- If we are at problem $n_t(k, E)$ with $n \geq 3$, and it satisfies case a) of its definition, then we take $a = b$ and move to problem $n_t(n - j, E^*)$.
- Otherwise, if $n_t(k, E)$ with $n \geq 3$, satisfies case b) of its definition, we can take a equal to any action b , used in an expression $M_f(n - j, E^*) + g_f(b)$ for W , and move to problem $n_t(n - j, E^*)$.

Finally, if we are at problems $n_t(2, S), n_t(3, S)$ or $n_t(3, C)$ we can take the compressed action sequences of their unique maximal sequences.

Notice, that by performing every possible assignment in the constructive method associated to problem $n_t(n, E)$, we obtain all maximal compressed action sequences for problem $M_f(n, E)$. Moreover, by computing all maximal compressed action sequences for problems $M_f(n, S)$ and $M_f(n, C)$, as appropriate, we can obtain all of the extremal compressed action sequences with n squares.

Remark 3.3.4. For the same reasons stated in Remark 2.3.5, the time complexity of obtaining all maximal compressed action sequences of problem $M_f(n, E)$ is linear with respect to their amount, but possibly exponential with respect to n . However, in contrast to Remark 2.3.5, although we are able to generate all extremal compressed action sequences, we cannot guarantee that all such sequences are realizable and thus we could be possibly retrieving a lot of irrelevant information. Consequently, the combination of Algorithms 1 and 2 remains the only efficient approach for directly constructing at least an extremal general polyomino chain.

Furthermore, observe that applying Algorithm 1 to the entire collection of extremal compressed action sequences would indeed transform them into realizable but this process would only retain a subset of all realizable extremal compressed action sequences.

Helping with the problem of retrieving non-realizable extremal compressed action sequences, we now give the generalization of Corollary 2.3.4, which in this case provides criteria to guarantee we can obtain all extremal general polyomino chains efficiently.

Corollary 3.3.5. *If for $n \geq 1$, $n_t(n, E) = 0$ for some $E \in \{S, C\}$, then the sequence retrieved by the constructive method of $M_f(n, E)$ is the unique maximal compressed action sequence for problem $M_f(n, E)$ and is realizable. Moreover, if there exists a unique extremal compressed action sequence, then it is realizable.*

Proof. Suppose for $n \geq 1$, $n_t(n, E) = 0$ for some $E \in \{S, C\}$ and that $(a_1, a_2, \dots, a_N)$ was a maximal compressed action sequence corresponding to problem $M_f(n, E)$. Then, because $n_t(n, E) = 0$, by Theorem 3.3.3 it follows $(a_1, a_2, \dots, a_N)$ is the unique maximal sequence for that problem.

But, since applying Algorithm 1 to the sequence yields a maximal compressed action sequence corresponding to problem $M_f(n, E)$ which is realizable, it must be $(a_1, a_2, \dots, a_N)$ is realizable.

The second part follows from the exact same argument. $\square$

To address the problem raised in Remark 3.3.4, it is necessary to identify criteria and algorithms capable of distinguishing realizable compressed action sequences. To this end, Algorithm 3 describes how to generate, from a given general action sequence, all corresponding interpretations; where the translation between general actions and instructions is established by performing the reverse process to the one introduced in Definition 3.1.2. Then, by combining Algorithm 3 with a subsequent analysis of each resulting instruction sequence, carried out by applying Lemma 1.2.5, we obtain a systematic procedure to determine whether a given general action sequence is realizable (which observe is sustained on Proposition 1.2.7). We refer to this procedure as the *exhaustive analysis of the general action sequence*.

Algorithm 3: A general action sequence $\rightarrow$ all of its interpretations

Input: General action sequence $(a_1, \dots, a_n)$

Output: Instruction sequence $(I_1, \dots, I_n)$

```

1  $I_1 = R; \quad flag \leftarrow 0;$ 
2 for  $i = 2$  to  $n$  do
3   if  $a_i \in \{SS, CS, SS^*\}$  then  $I_i \leftarrow I_{i-1}$ 
4   else if  $a_i = CC$  then  $I_i \leftarrow I_{i-2}$ 
5   else if  $a_i = TT$  then  $I_i \leftarrow \overline{I_{i-3}}$ 
6   else if  $flag = 0$  and  $a_i \in \{SC, SC^*\}$  then  $I_i \leftarrow D$  and  $flag \leftarrow 1$ 
7   else if  $a_i = SC$  and  $a_{i-2} \in \{SC, CC, TT\}$  then  $I_i \leftarrow I_{i-3}$ 
8   else  $I_i \in \{R, L, D, U\} \setminus \{I_{i-1}, \overline{I_{i-1}}\}$ 

```

Remark 3.3.6. Observe that in Algorithm 3, the generation of multiple instructions can occur only through general actions of type SC . This is because SC is the unique action which appears in more than one scenario involving I_{k-1} and I_k , to which we asked in Definition 3.1.2 only to differ in direction. Consequently, in a general action sequence $(a_1, \dots, a_n)$, the occurrences of SC are precisely the positions at which one may possibly choose between two distinct corresponding instructions. We therefore refer to the SC general actions as the *pivots of the general action sequence*.

Also, in Algorithm 3 we decide to fix the instruction corresponding to the first pivot found in the sequence because the graphs given by the sequences $(R, R, \dots, R, D)$ and $(R, R, \dots, R, U)$ are isomorphic. And we fix as well the instruction corresponding to every pivot with $a_{i-2} \in \{SC, CC, TT\}$ because choosing it to be equal to $\overline{I_{i-3}}$ would turn it into a TT general action, which does not correspond to the effect it should have on the TI_f .

Now, with Algorithm 3, it is possible not only to obtain all extremal compressed action sequences but also all extremal general polyomino chains, for which we obtain their instruction sequences as a byproduct of the algorithm. Analyzing the computational complexity of this procedure, in light of Remark 3.3.6, it is straightforward to determine that the complexity of Algorithm 3 is $O(2^{\text{number of pivots}})$. Moreover, since the number of pivots in a given general action sequence can be typically comparable to $n/2$ (because each use of an action SC needs at least an intermediate action different from SC), it follows that the exhaustive analysis of a general action sequence has exponential time complexity in $n/2$. Consequently, the complexity of finding all extremal general polyomino chains, which needs the application of Lemma 1.2.5 to every obtained interpretation, is $O(2^{n/2} N n)$; N is the number of extremal general action sequences.

However, because of this possible branching of a general action sequence into an exponential amount of corresponding instruction sequences that will most likely be in their majority non-realizable or isomorphic, it still remains as an open problem to develop an efficient algorithm, or to identify purely necessary or sufficient conditions that enable the discrimination of realizable general action sequences and their set of realizable interpretations (preferentially avoiding isomorphisms). Working in this direction, we provide a lemma that for certain patterns will stop the branching and which will play a crucial role in the characterization of all extremal general polyomino chains of R_{-1} .

Lemma 3.3.7. *Let $(a_1, \dots, a_n)$ be a general action sequence and ℓ the index of the sequence's second pivot. Suppose a_i with $i \geq \ell$ is a pivot and $(I_1, \dots, I_{i-1})$ is a sequence of instructions obtained by applying Algorithm 3 up to action a_{i-1} , then*

- a) *If $a_{i-2} \in \{SC, CC, TT\}$, then the instruction assigned to the pivot a_i is fixed and equals I_{i-3} .*
- b) *If $i - 7 > 1$, $I_{i-7} = I_{i-6}$, $a_{i-5} = SC, a_{i-4} = CS, a_{i-3} = TT, a_{i-2} = CS$ and $a_{i-1} = SS$, then the instruction assigned to the pivot a_i is fixed and equals I_{i-5} , provided that $(I_{i-8}, I_{i-7}, I_{i-6}, I_{i-5}, I_{i-4}, I_{i-3}, I_{i-2}, I_{i-1}, I_i)$ is realizable.*
- c) *If $i - 8 > 1$, $I_{i-8} = I_{i-7}$, $a_{i-6} = SC, a_{i-5} = CS, a_{i-4} = TT, a_{i-3} = CS, a_{i-2} = a_{i-1} = SS, a_{i+1} = CS$ and $a_{i+2} = TT$, then the instruction assigned to the pivot a_i is fixed and equals I_{i-5} , provided that $(I_{i-8}, I_{i-7}, I_{i-6}, I_{i-5}, I_{i-4}, I_{i-3}, I_{i-2}, I_{i-1}, I_i, I_{i+1}, I_{i+2})$ is realizable.*
- d) *If $I_{i-5} = I_{i-4}$, $a_{i-3} = SC, a_{i-2} = CS, a_{i-1} = SS, a_{i+1} = CS, a_{i+2} = TT$ and $a_{i+3} = CS$, then the instruction assigned to the pivot a_i is fixed and equals I_{i-4} , provided that $(I_{i-5}, I_{i-4}, I_{i-3}, I_{i-2}, I_{i-1}, I_i, I_{i+1}, I_{i+2}, I_{i+3})$ is realizable.*
- e) *If $i - 9 > 1$, $I_{i-9} = I_{i-8}$, $a_{i-7} = SC, a_{i-6} = CS, a_{i-5} = TT, a_{i-4} = CS, a_{i-3} = a_{i-2} = a_{i-1} = SS, a_{i+1} = CS, a_{i+2} = TT$ and $a_{i+3} = CS$, then the instruction assigned to the pivot a_i is fixed and equals I_{i-6} , provided that $(I_{i-9}, I_{i-8}, I_{i-7}, I_{i-6}, I_{i-5}, I_{i-4}, I_{i-3}, I_{i-2}, I_{i-1}, I_i, I_{i+1}, I_{i+2}, I_{i+3})$ is realizable.*

Proof. Case a) is true directly by Algorithm 3. For the next cases we shall only prove b), as all others follow a similar argument.

So suppose $i - 7 > 1$, $I_{i-7} = I_{i-6}$, $a_{i-5} = SC, a_{i-4} = CS, a_{i-3} = TT, a_{i-2} = CS, a_{i-1} = SS$ and that the sequence of instructions $(I_{i-8}, I_{i-7}, I_{i-6}, I_{i-5}, I_{i-4}, I_{i-3}, I_{i-2}, I_{i-1}, I_i)$ (which exists because the instruction I_{i-7} is not the first) is realizable. Since the general action a_i is a pivot and it's not the first nor does it satisfy $a_{i-2} \in \{SC, CC, TT\}$, it follows Algorithm 3 possibly assigns it instructions $\{R, L, D, U\} \setminus \{I_{i-1}, \overline{I_{i-1}}\} = \{I_{i-5}, \overline{I_{i-5}}\}$.

Assume by way of contradiction it assigned it the instruction $\overline{I_{i-5}}$. Then, assuming without loss of generality that the first three squares corresponding to the instructions I_{i-8}, I_{i-7} and I_{i-6} laid horizontally, which we can do because $I_{i-7} = I_{i-6}$, we would have the sequence of instructions $(I_{i-8}, I_{i-7}, I_{i-6}, I_{i-5}, I_{i-4}, I_{i-3}, I_{i-2}, I_{i-1}, I_i)$ represents the graph shown in Figure 3.5, which is not a general polyomino chain. Thus, it must be a_i is assigned the instruction I_{i-5} . □

Proposition 3.3.8. *Let $(a_1, \dots, a_n)$ be a general action sequence and $(I_1, \dots, I_n)$ one of its interpretations such that every pivot of the general action sequence after the first satisfies a condition of Lemma 3.3.7, then its only possibly realizable interpretation is given by the unique sequence obtained through Algorithm 3 together with the fixed assignments of Lemma 3.3.7 to every such pivot.*

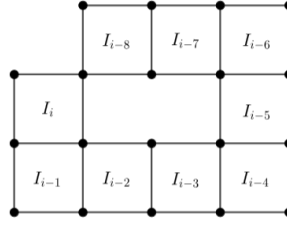


Figure 3.5: The graph corresponding to the sequence of instructions $(I_{i-8}, I_{i-7}, I_{i-6}, I_{i-5}, I_{i-4}, I_{i-3}, I_{i-2}, I_{i-1}, I_i)$ with $I_i = \overline{I_{i-5}}$.

3.3.2. Cyclic Behavior of Extremal General Polyomino Chains

Now, before trying to characterize the extremal general polyomino chains of degree-based indices, it is necessary to generalize the conditions for cyclic behaviors to appear. Here, differently as it happened in the case of polyomino chains (see Section 2.2.3), it is not sufficient to spot the reappearance of a pair of advantages, but of three consecutive pairs. In other words, if we know the advantages of three consecutive problems, we will show it is possible to know which expression in the *max* argument is maximum. In what follows we carry out the not so immediate generalization in order to establish patterns in the decisions behind the construction of the values of problems $M_f(n, E)$.

Lemma 3.3.9. *Let $n \geq 3$ and suppose for some $3 \leq j < n$*

$$\begin{aligned} V(n-2, S) &= V(j-2, S), & V(n-2, C) &= V(j-2, C) \\ V(n-1, S) &= V(j-1, S), & V(n-1, C) &= V(j-1, C) \\ V(n, S) &= V(j, S), & V(n, C) &= V(j, C), \end{aligned}$$

and problems $M_f(n, S), M_f(n-1, S), M_f(j, S)$ and $M_f(j-1, S)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SS), M_f(i-1, C) + g_f(CS)\},$$

where $i \in \{n, n-1, j, j-1\}$.

If problems $M_f(n+1, C)$ and $M_f(j+1, C)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SC), M_f(i-1, C) + g_f(CC), M_f(i-3, S) + g_f(ST), M_f(i-3, C) + g_f(CT)\},$$

where $i \in \{n+1, j+1\}$, and we have $M_f(j+1, C) = M_f(j+1-k, E) + g_f(b)$; where $k \in \{1, 3\}$, $E \in \{S, C\}$ and $b \in \{SC, CC, ST, CT\}$, then $M_f(n+1, C) = M_f(n+1-k, E) + g_f(b)$.

Proof. First of all, notice if problems $M_f(n+1, C)$ and $M_f(j+1, C)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SC), M_f(i-1, C) + g_f(CC), M_f(i-3, S) + g_f(ST), M_f(i-3, C) + g_f(CT)\}$$

then they can be rewritten as

$$\begin{aligned} &\max\{\max\{M_f(i-1, S) + g_f(SC), M_f(i-1, C) + g_f(CC)\}, \\ &\quad \max\{M_f(i-3, S) + g_f(ST), M_f(i-3, C) + g_f(CT)\}\} \end{aligned}$$

Now, let

$$\max\{M_f(j-2, S) + g_f(ST), M_f(j-2, C) + g_f(CT)\} = M_f(j-2, H_1) + g_f(b_1)$$

and

$$\max\{M_f(j, S) + g_f(SC), M_f(j, C) + g_f(CC)\} = M_f(j, H_2) + g_f(b_2)$$

where $H_1, H_2 \in \{S, C\}$, $b_1 \in \{ST, CT\}$ and $b_2 \in \{SC, CC\}$. Then, because by hypothesis

$$\begin{aligned} V(n, S) &= V(j, S) & V(n, C) &= V(j, C) \\ V(n-2, S) &= V(j-2, S) & V(n-2, C) &= V(j-2, C) \end{aligned}$$

by Lemma 2.2.6 it follows

$$\max\{M_f(n-2, S) + g_f(ST), M_f(n-2, C) + g_f(CT)\} = M_f(n-2, H_1) + g_f(b_1)$$

and

$$\max\{M_f(n, S) + g_f(SC), M_f(n, C) + g_f(CC)\} = M_f(n, H_2) + g_f(b_2).$$

Suppose without loss of generality, that $M_f(j-2, H_1) + g_f(b_1) \geq M_f(j, H_2) + g_f(b_2)$. Then, since we can always write the problems by means of their advantages and their base values, for $x_1^*, x_2^* \in \mathbb{R}$ we have:

$$\begin{aligned} x_1^* + V(j-2, H_1) + g_f(b_1) &\geq x_2^* + V(j, H_2) + g_f(b_2) \\ \Rightarrow x_1^* - x_2^* &\geq V(j, H_2) - V(j-2, H_1) + g_f(b_2) - g_f(b_1). \end{aligned}$$

On the other hand, in order to show $M_f(n-2, H_1) + g_f(b_1) \geq M_f(n, H_2) + g_f(b_2)$, writing it in terms of their advantages and base values $x_1, x_2 \in \mathbb{R}$, we would have to show:

$$x_1 + V(n-2, H_1) + g_f(b_1) \geq x_2 + V(n, H_2) + g_f(b_2),$$

which is equivalent by our hypothesis to showing

$$x_1 - x_2 \geq V(j, H_2) - V(j-2, H_1) + g_f(b_2) - g_f(b_1).$$

Thus, it only remains to prove $x_1 - x_2 \geq x_1^* - x_2^*$. For this, we will now use the hypothesis made on problems $M_f(n, S), M_f(n-1, S), M_f(j, S), M_f(j-1, S)$ and relate the values x_1^*, x_2^* with x_1, x_2 respectively.

So first of all, by the form of problem $M_f(j-1, S)$, we have $M_f(j-1, S) = M_f(j-2, Z_1) + g_f(a_1)$; where $Z_1 \in \{S, C\}$ and $a_1 \in \{SS, CS\}$. Then, writing it in terms of its advantage and base value, we have:

$$M_f(j-1, S) = x_1^* + V(j-2, Z_1) + g_f(a_1).$$

Also, by the form of problem $M_f(j, S)$, we have $M_f(j, S) = M_f(j-1, Z_2) + g_f(a_2)$; where $Z_2 \in \{S, C\}$ and $a_2 \in \{SS, CS\}$. To relate x_1^* with x_2^* we now divide into cases:

1) If $Z_2 = S$, then it follows:

$$\begin{aligned} M_f(j, S) &= x_1^* + V(j-2, Z_1) + g_f(a_1) + g_f(a_2) \\ \Rightarrow x_2^* + V(j, S) &= x_1^* + V(j-2, Z_1) + g_f(a_1) + g_f(a_2) \\ \Rightarrow x_1^* - x_2^* &= V(j, S) - V(j-2, Z_1) - g_f(a_1) - g_f(a_2) \end{aligned}$$

2) If $Z_2 = C$, then it follows:

$$\begin{aligned} M_f(j, S) &= M_f(j-1, S) - V(j-1, S) + V(j-1, C) + g_f(a_2) \\ \Rightarrow M_f(j, S) &= x_1^* + V(j-2, Z_1) + g_f(a_1) - V(j-1, S) + V(j-1, C) + g_f(a_2) \\ \Rightarrow x_2^* + V(j, S) &= x_1^* + V(j-2, Z_1) + g_f(a_1) - V(j-1, S) + V(j-1, C) + g_f(a_2) \\ \Rightarrow x_1^* - x_2^* &= V(j, S) - V(j-2, Z_1) - g_f(a_1) + V(j-1, S) - V(j-1, C) - g_f(a_2) \end{aligned}$$

Now, for problems $M_f(n-1, S)$ and $M_f(n, S)$, by Lemma 2.2.7 we have they behave exactly the same as their j indexed counterparts, therefore:

$$M_f(n-1, S) = M_f(n-2, Z_1) + g_f(a_1)$$

and

$$M_f(n, S) = M_f(n-1, Z_2) + g_f(a_2).$$

Working out the same process as the one done with problems $M_f(j-1, S)$ and $M_f(j, S)$ but now with the advantage and base value representation of problems $M_f(n-1, S)$, $M_f(n, S)$ we arrive at similar expressions for the difference $x_1 - x_2$.

1) If $Z_2 = S$, then:

$$x_1 - x_2 = V(n, S) - V(n-2, Z_1) - g_f(a_1) - g_f(a_2).$$

2) If $Z_2 = C$, then:

$$x_1 - x_2 = V(n, S) - V(n-2, Z_1) - g_f(a_1) + V(n-1, S) - V(n-1, C) - g_f(a_2).$$

However, since by hypothesis the advantages $V(n, S)$, $V(n-2, Z_1)$, $V(n-1, S)$ and $V(n-1, C)$ are equal to $V(j, S)$, $V(j-2, Z_1)$, $V(j-1, S)$ and $V(j-1, C)$ respectively, we can conclude in either case $x_1 - x_2 = x_1^* - x_2^*$ and hence $M_f(n-2, H_1) + g_f(b_1) \geq M_f(n, H_2) + g_f(b_2)$.

Therefore, the conclusion of the statement follows. □

Lemma 3.3.10. *Let $n \geq 3$ and suppose for some $3 \leq j < n$*

$$\begin{aligned} V(n-2, S) &= V(j-2, S), & V(n-2, C) &= V(j-2, C) \\ V(n-1, S) &= V(j-1, S), & V(n-1, C) &= V(j-1, C) \\ V(n, S) &= V(j, S), & V(n, C) &= V(j, C) \end{aligned}$$

and that problems $M_f(n, S)$, $M_f(n-1, S)$, $M_f(j, S)$, $M_f(j-1, S)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SS), M_f(i-1, C) + g_f(CS)\},$$

where $i \in \{n, n-1, j, j-1\}$.

If problems $M_f(n+1, S)$, $M_f(j+1, S)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SS), M_f(i-1, C) + g_f(CS)\},$$

and problems $M_f(n+1, C)$, $M_f(j+1, C)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SC), M_f(i-1, C) + g_f(CC), M_f(i-3, S) + g_f(ST), M_f(i-3, C) + g_f(CT)\},$$

where $i \in \{n+1, j+1\}$, then $V(n+1, S) = V(j+1, S)$ and $V(n+1, C) = V(j+1, C)$.

Proof. Because problems $M_f(j+1, S)$ and $M_f(j+1, C)$ have the forms

$$\max\{M_f(i-1, S) + g_f(SS), M_f(i-1, C) + g_f(CS)\}$$

and

$$\max\{M_f(i-1, S) + g_f(SC), M_f(i-1, C) + g_f(CC), M_f(i-3, S) + g_f(ST), M_f(i-3, C) + g_f(CT)\}$$

respectively, it follows we can write them as

$$\begin{aligned} M_f(j+1, S) &= M_f(j, H_1) + g_f(b_1) \\ &= x_1^* + V(j, H_1) + g_f(b_1) \end{aligned}$$

and

$$\begin{aligned} M_f(j+1, C) &= M_f(j+1-k, H_2) + g_f(b_2) \\ &= x_2^* + V(j+1-k, H_2) + g_f(b_2), \end{aligned}$$

where $H_1, H_2 \in \{S, C\}$, $k \in \{1, 3\}$, $b_1 \in \{SS, CS\}$ and $b_2 \in \{SC, CC, ST, CT\}$. Then, by our hypothesis, Lemma 2.2.7 and Lemma 3.3.9, we have problems $M_f(n+1, S)$ and $M_f(n+1, C)$ behave exactly the same:

$$\begin{aligned} M_f(n+1, S) &= M_f(n, H_1) + g_f(b_1) \\ &= x_1 + V(n, H_1) + g_f(b_1) \end{aligned}$$

$$\begin{aligned} M_f(n+1, C) &= M_f(n+1-k, H_2) + g_f(b_2) \\ &= x_2 + V(n+1-k, H_2) + g_f(b_2). \end{aligned}$$

Assume without loss of generality $M_f(j+1, S) \geq M_f(j+1, C)$. We now divide into cases depending on the value of k in order to determine the advantages.

- If $k = 1$, then it follows $x_1^* = x_2^*$ and $x_1 = x_2$. Thus

$$M_f(j+1, S) - M_f(j+1, C) = V(j, H_1) - V(j, H_2) + g_f(b_1) - g_f(b_2)$$

and since $V(n, H_1) = V(j, H_1)$, $V(n, H_2) = V(j, H_2)$, we have $M_f(n+1, S) - M_f(n+1, C) = M_f(j+1, S) - M_f(j+1, C)$.

- If $k = 3$, then by the proof of Lemma 3.3.9 we have $x_1^* - x_2^* = x_1 - x_2$ and therefore, similarly as in the previous case, it follows $M_f(n+1, S) - M_f(n+1, C) = M_f(j+1, S) - M_f(j+1, C)$.

Finally, because in either case $M_f(n+1, S) - M_f(n+1, C) = M_f(j+1, S) - M_f(j+1, C)$, we can conclude $V(n+1, S) = V(j+1, S)$ and $V(n+1, C) = V(j+1, C)$.

□

Corollary 3.3.11. *Let $n \geq 3$ and suppose for some $3 \leq j < n$*

$$\begin{aligned} V(n-2, S) &= V(j-2, S), & V(n-2, C) &= V(j-2, C) \\ V(n-1, S) &= V(j-1, S), & V(n-1, C) &= V(j-1, C) \\ V(n, S) &= V(j, S), & V(n, C) &= V(j, C) \end{aligned}$$

and that problems $M_f(n, S), M_f(n-1, S), M_f(j, S), M_f(j-1, S)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SS), M_f(i-1, C) + g_f(CS)\}$$

where $i \in \{n, n-1, j, j-1\}$.

If problems $M_f(n+1, S), M_f(j+1, S)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SS), M_f(i-1, C) + g_f(CS)\}$$

problems $M_f(n+1, C), M_f(j+1, C)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SC), M_f(i-1, C) + g_f(CC), M_f(i-3, S) + g_f(ST), M_f(i-3, C) + g_f(CT)\}$$

where $i \in \{n+1, j+1\}$ and $M_f(j+1) = M_f(j+1, E)$, then $M_f(n+1) = M_f(n+1, E)$; $E \in \{S, C\}$.

Finally, as a consequence of all the results in this section, we give conditions such that the computations of problems $M_f(n, E)$ and the general actions chosen in the constructive method enter into a cyclic pattern.

Theorem 3.3.12. *Let $n \geq 3$ be the first integer such that there exists $3 \leq j < n$*

$$\begin{aligned} V(n-2, S) &= V(j-2, S), & V(n-2, C) &= V(j-2, C) \\ V(n-1, S) &= V(j-1, S), & V(n-1, C) &= V(j-1, C) \\ V(n, S) &= V(j, S), & V(n, C) &= V(j, C) \end{aligned}$$

problems $M_f(i, S)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SS), M_f(i-1, C) + g_f(CS)\}$$

for $i \geq j-1$ and problems $M_f(i, C)$ are of the form

$$\max\{M_f(i-1, S) + g_f(SC), M_f(i-1, C) + g_f(CC), M_f(i-3, S) + g_f(ST), M_f(i-3, C) + g_f(CT)\}$$

for $i \geq j+1$.

Then, for $i \geq n+1$ let $k = j + [i - n \bmod (n-j)]$ if $i - n \bmod (n-j) \neq 0$ and $k = n$ otherwise. If we have $M_f(k, E) = M_f(k-\ell, H) + g_f(b)$ with $H \in \{S, C\}$, $\ell \in \{1, 3\}$ and b a general action, then $M_f(i, E) = M_f(i-\ell, H) + g_f(b)$ and $V(i, E) = V(k, E)$. Also, if $M_f(k) = M_f(k, E)$, then $M_f(i) = M_f(i, E)$.

Remark 3.3.13. Notice, in the same way as happened for the results in Section 2.2.3, since every result in this section preserved equalities, Theorem 3.3.12 can also be applied to problems $n_t(n, S)$ and $n_t(n, C)$, meaning ties are preserved throughout cyclic patterns.

3.3.3. All General Polyomino Chains for R_{-1}

In this section, by making use of Theorems 3.2.6, 3.3.3, 3.3.12, Algorithms 1, 2, 3, the constructive method of problems $n_t(n, E)$ and Proposition 3.3.8, we show as an application of our theory in the context of general polyomino chains, the characterization of all maximum general polyomino chains for the generalized Randić index with parameter equal to -1 , which was until now an open problem.

The Randić index was originally introduced by the chemist Milan Randić in 1975 [25] as

$$R(G) = \sum_{uv \in E(G)} \frac{1}{\sqrt{d_u d_v}},$$

which was later generalized by Bollobás and Erdős in 1998 [5] for any $r \in \mathbb{R}$ as

$$R_r(G) = \sum_{uv \in E(G)} (d_u d_v)^r.$$

Nowadays, the generalized Randić index has become one of the most extensively studied, widely applied, and well-recognized topological indices [20]. Its importance coming from its strong correlation with various chemical properties, including the boiling points, surface area, and solubility in water of alkanes [26]. In particular, R_{-1} , known as the second modified Zagreb index, is of special interest due to its connection with the eigenvalues of the normalized Laplacian matrix of the graph [7].

Regarding extremal general polyomino chains with n squares under the generalized Randić index (R_r), the following facts are known (see Theorems 2.1, 2.2 and 3.9 in [23]):

- a) If $r > 0$, then Li_n minimizes and Z_n maximizes R_r .
- b) If $-0.84313 < r < 0$, then Z_n minimizes and Li_n maximizes R_r .
- c) If $-1.23853 < r < -1$, then Z_n minimizes and Z_n^3 (defined below) maximizes R_r .
- d) If $-1 \leq r \leq -0.84313$, then Z_n minimizes R_r .

However, there are still many gaps concerning the extremal chains for the parameter's values. In what follows we shall solve this problem for $r = -1$, hoping that the proposed procedure serves as a guideline for the study of the remaining values.

Before stating the main result for R_{-1} , we fix a compact notation for the canonical families of general polyomino chains that will appear in the extremal configurations.

Notation 3.3.14. Let $m \geq 0$.

- We denote by $C_{\ell_1, \dots, \ell_q}^{\alpha_1, \dots, \alpha_q}$ the family of general polyomino chains in which all vertical segments have length three, and the total of $\sum_{i=1}^q \alpha_i$ horizontal segments, consists of α_j segments of length ℓ_j , for $j = 1, \dots, q$; where the change of vertical to horizontal is always given by a general action TT and the change of horizontal to vertical is always given by a general action SC . In particular:
 - C_3^{m+1} : exactly $m+1$ horizontal segments of length 3. See Figure 3.6 a).
 - $C_{3,4}^{m,1}$: m horizontal segments of length 3 and one horizontal segment of length 4. See Figure 3.6 b).
 - $C_{3,5}^{m,1}$: m horizontal segments of length 3 and one horizontal segment of length 5.
 - $C_{3,6}^{m,1}$: m horizontal segments of length 3 and one horizontal segment of length 6.
 - $C_{3,4}^{m-1,2}$: $m-1$ horizontal segments of length 3 and two horizontal segments of length 4.
 - $C_{3,4}^{m-2,3}$: $m-2$ horizontal segments of length 3 and three horizontal segments of length 4.
 - $C_{3,4,5}^{m-1,1,1}$: $m-1$ horizontal segments of length 3, one horizontal segment of length 4, one horizontal segment of length 5.
- We denote by $\bar{C}_3^{m+1}(i)$; $1 \leq i \leq m+1$, the general polyomino chain with $m+1$ horizontal segments and $m+1$ vertical segments, where all segments have length three and where we have up to the i^{th} vertical segment the change of horizontal to vertical is given by the general action SC and the change of vertical to horizontal is given by a general action TT , while from the i^{th} vertical segment onward, the changes work the other way around. As an example see Figure 3.6 c), which corresponds to $\bar{C}_3^3(2)$.
 We denote by $\bar{C}_3^{m+1}$ the family of general polyomino chains consisting of all $\bar{C}_3^{m+1}(i)$.
- We denote by $\bar{C}_{3,4}^{m,1}(i, j)$; $1 \leq i \leq m+1$ and $1 \leq j \leq i$, the general polyomino chain consisting of $m+1$ horizontal segments and $m+1$ vertical segments, where we have up to the i^{th} vertical segment the change of horizontal to vertical is given by the general action SC and the change of vertical to horizontal is given by a general action TT , while from the i^{th} vertical segment onward, the changes work the other way around. Also, all segments have length three, except for the j^{th} horizontal segment which has length four. As an example see Figure 3.6 b), which corresponds to $\bar{C}_{3,4}^{2,1}(3, 3)$.
 We denote by $\bar{C}_{3,4}^{m,1}$ the family of general polyomino chains consisting of all $\bar{C}_{3,4}^{m,1}(i, j)$.

- We denote by $C_{3,4}^{m,1}(i,j)$; $1 \leq i \leq m+1$ and $i \leq j \leq m+1$, the general polyomino chain consisting of $m+1$ horizontal segments and $m+1$ vertical segments, where we have up to the i^{th} vertical segment the change of horizontal to vertical is given by the general action SC and the change of vertical to horizontal is given by a general action TT , while from the i^{th} vertical segment onward, the changes work the other way around. Also, all segments have length three, except for the j^{th} vertical segment which has length four. As an example see Figure 3.6 d), which corresponds to $C_{3,4}^{2,1}(1,1)$.

We denote by $\underline{C}_{3,4}^{m,1}$ the family of general polyomino chains consisting of all $C_{3,4}^{m,1}(i,j)$.

- We use Z_n^3 to denote a general polyomino chain with n squares, in which all changes of horizontal to vertical are given by the general action SC and the changes of vertical to horizontal are given by a general action TT . Also, every segment has length three (except when n is even, in which case the last segment has length four).

In particular, $Z_n^3 = C_3^{m+1}$ for $n = 3 + 4m, 5 + 4m$, while $Z_n^3 \subset C_{3,4}^{m,1}$ for $n = 4 + 4m$. Note that, when $n = 6 + 4m$, the final segment of Z_n^3 is vertical. Moreover, the following inclusions hold: $C_3^{m+1} \subset \bar{C}_3^{m+1}$ for $n = 3 + 4m, 5 + 4m$, $C_{3,4}^{m,1} \subset \bar{C}_{3,4}^{m,1}$ for $n = 4 + 4m, 6 + 4m$ and $Z_n^3 \subset \underline{C}_{3,4}^{m,1}$ for $n = 6 + 4m$. Furthermore, each element of $\bar{C}_{3,4}^{m,1}$ is isomorphic to a unique element of $\underline{C}_{3,4}^{m,1}$, see Figure 3.6 b) and d).

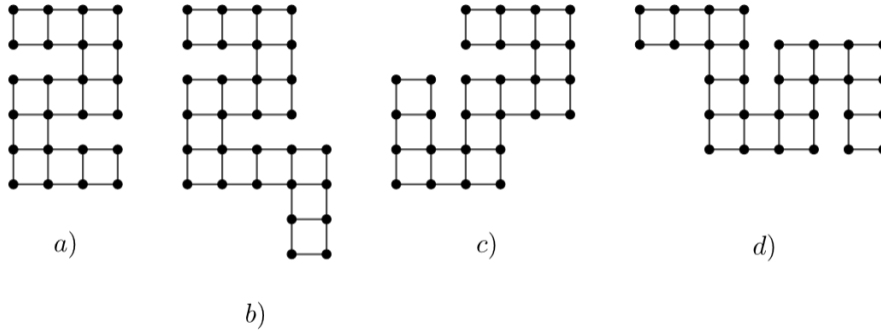


Figure 3.6: According to Notation 3.3.14, the graphs shown in a), b), c) and d) correspond, respectively, to $Z_{11}^3 = C_3^3$, $\bar{C}_{3,4}^{2,1}(3,3)$, $\bar{C}_3^3(2)$ and $\underline{C}_{3,4}^{2,1}(1,1)$

For simplicity, in this section, we will use the notation g , $M(n, S)$, $M(n, C)$, and $M(n)$ instead of g_f , $M_f(n, S)$, $M_f(n, C)$, and $M_f(n)$, respectively. Also, by carrying out base computations, we obtain $g(T) = 1$, $g(SS^*) \approx 0.277$, $g(SC^*) \approx 0.305$, $g(ST^*) \approx 0.979$, $g(SS) \approx 0.333$, $g(SC) \approx 0.319$, $g(CS) \approx 0.347$, $g(CC) \approx 0.312$, $g(ST) \approx 0.993$, $g(CT) \approx 0.986$; which can serve the reader as a useful reference although the computations for the following proof need the use of the exact expressions.

Theorem 3.3.15. *Let $n \geq 3$ and write $n = k + 4m$ with $k \in \{3, 4, 5, 6\}$ and $m \in \mathbb{Z}_{\geq 0}$. Then the general polyomino chains with n squares that maximize R_{-1} are as follows:*

- When $n = 3 + 4m$, the general polyomino chain C_3^{m+1} .
- When $n = 4 + 4m$, the maximizing family is $C_{3,4}^{m,1}$, with cardinality $m+1$.
- When $n = 5 + 4m$, the maximizing families are $\bar{C}_3^{m+1}$, $C_{3,5}^{m,1}$, and $C_{3,4}^{m-1,2}$, yielding a total of $\frac{(m+1)(m+4)}{2}$ maximum general polyomino chains.

d) When $n = 6 + 4m$, the maximizing families are $C_{3,6}^{m,1}$, $\bar{C}_{3,4}^{m,1}$, $C_{3,4}^{m,1}$, $C_{3,4,5}^{m-1,1,1}$, and $C_{3,4}^{m-2,3}$, yielding a total of $\frac{(m+1)(m+2)(2m+18)}{12}$ maximal general polyomino chains.

Specifically, for $k \in \{3, 4, 5, 6\}$,

$$M(k + 4m) = \frac{11}{18} + \frac{1}{3}k + \frac{143}{144}m.$$

Proof. For this proof we can begin by following an exact same reasoning as the one displayed on the proof of Theorem 2.2.14, with the only difference now problems $M(n, E)$ are given by Theorem 3.2.6.

We can continue that process up to $n = 10$, where we summarize the key information in Figure 3.7, which depicts the underlying tie graph of the dynamic programming approach. In it, we have included the values for the advantages $V(n, S)$ and $V(n, C)$, where notice we did not write function g in the expressions for the advantages in order to make the table more compact, although every action should be inside a $g()$ expression. Also, for visual simplicity, we have highlighted in red the cells corresponding to the problem with the highest index value, that is, the problem corresponding to $M(n)$.

According to the key information summarized in Figure 3.7, it is evident that for $n = 10$ and $j = 6$ we arrive at the conditions for Theorem 3.3.12, which together with Remark 3.3.13, will force problems for $n \geq 11$ to behave the same as those for $n = 7, 8, 9, 10$. This indicates the onset of a cyclic pattern in the decision process, with a period of four. Therefore, based on the previous information and applying the constructive method for problem $n_t(n, S)$, we obtain the following:

- When $n = 3 + 4m$ with $m \geq 0$, the extremal compressed action sequence is given by

$$(SS, \underbrace{ST, CS}_{m \text{ times}}).$$

and by applying Proposition 3.3.8 to its uncompressed general action sequence, we have its only realizable interpretation is given by the general polyomino chain C_3^{m+1} .

- When $n = 4 + 4m$ with $m \geq 0$, the extremal compressed action sequences are given by

$$(SS, \underbrace{ST, CS}_{m-i \text{ times}}, SS, \underbrace{ST, CS}_{i \text{ times}})$$

with $0 \leq i \leq m$. Then, by applying Proposition 3.3.8 to their uncompressed general action sequences, we have their only realizable interpretations correspond to the family of general polyomino chains $C_{3,4}^{m,1}$.

- When $n = 5 + 4m$ with $m \geq 0$, the extremal compressed action sequences are given by

$$(SS, \underbrace{ST, CS}_{m-i \text{ times}}, SC, CS, \underbrace{ST, CS}_{i \text{ times}})$$

and

$$(SS, \underbrace{ST, CS}_{m-i-j \text{ times}}, \underbrace{ST, CS}_{j \text{ times}}, SS, \underbrace{ST, CS}_{i \text{ times}})$$

with $0 \leq i \leq m$ and $0 \leq j \leq m-i$. Then, by applying Proposition 3.3.8 to their uncompressed general action sequences, we have their only realizable interpretations correspond to the families of general polyomino chains $\bar{C}_3^{m+1}$, $C_{3,5}^{m,1}$ and $C_{3,4}^{m-1,2}$.

- When $n = 6 + 4m$ with $m \geq 0$, the extremal compressed action sequences are given by

$$(SS, \underbrace{ST, CS}_{m-i-j \text{ times}}, SC, CS, \underbrace{ST, CS}_{j \text{ times}}, \underbrace{SS, ST, CS}_{i \text{ times}}),$$

$$(SS, \underbrace{ST, CS}_{m-i-j \text{ times}}, \underbrace{SS, ST, CS}_{j \text{ times}}, SC, CS, \underbrace{ST, CS}_{i \text{ times}})$$

and

$$(SS, \underbrace{ST, CS}_{m-i-j-k \text{ times}}, \underbrace{SS, ST, CS}_{k \text{ times}}, \underbrace{SS, ST, CS}_{j \text{ times}}, \underbrace{SS, ST, CS}_{i \text{ times}})$$

with $0 \leq i \leq m$, $0 \leq j \leq m - i$, and $0 \leq k \leq m - i - j$. Then, by applying Proposition 3.3.8 to their uncompressed general action sequences, we have their only realizable interpretations correspond to the families of general polyomino chains $C_{3,6}^{m,1}$, $C_{3,4}^{m,1}$, $C_{3,4}^{m,1}$, $C_{3,4,5}^{m-1,1,1}$ and $C_{3,4}^{m-2,3}$.

Finally, the concrete computations of $M(n)$ follow directly from Equation (3.1).

□

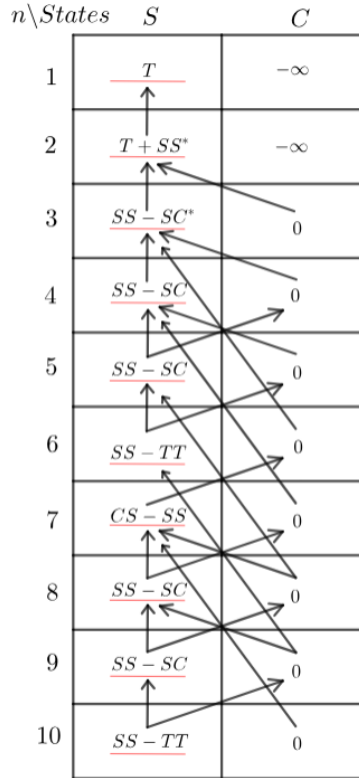


Figure 3.7: The tie graph for the maximum R_{-1} value.

Remark 3.3.16. In the proof of the previous result, once the form of all extremal compressed action sequences are determined for each case, the application of Algorithms 1 and 2 would immediately yield the following. The maximum of R_{-1} over general polyomino chains with n squares is attained by the following families:

- a) $n = 3 + 4m$: the family C_3^{m+1} .
- b) $n = 4 + 4m$: the family $C_{3,4}^{m,1}$.
- c) $n = 5 + 4m$: the families C_3^{m+1} , $C_{3,5}^{m,1}$, and $C_{3,4}^{m-1,2}$.
- d) $n = 6 + 4m$: the families $C_{3,6}^{m,1}$, $C_{3,4}^{m,1}$, Z_n^3 , $C_{3,4,5}^{m-1,1,1}$, and $C_{3,4}^{m-2,3}$.

This highlights the usefulness of Algorithms 1 and 2 as crucial tools which, once the dynamic programming analysis is completed, allow one to efficiently obtain at least one maximum structure. Even more, for $n = 3 + 4m, 4 + 4m$ the above resulting families coincide with all maximum families. In particular, in this case, after completing the dynamic programming analysis, the application of the exhaustive analysis to all extremal action sequences aimed at determining all maximum structures becomes manageable thanks to the information provided by Proposition 3.3.8, which effectively fixed all pivots in the obtained extremal general action sequences and left for each only one instruction sequence to check for realizability.

3.4. Polyomino Chains vs General Polyomino Chains

3.4.1. Conditions such that All Extremal General Action Sequences are Action Sequences

At this point we have studied how to find the extremal chains corresponding to a given degree-based topological index in either the restricted polyomino chain or in the general polyomino chain setting. However, the added complexity in the general polyomino chain setting coming from its bigger dynamic programming frameworks and the harder task to distinguish isomorphisms, make us wonder whether we can limit ourselves to the study of actions instead of general actions and still be able to find all extremal sequences. In this section we provide conditions in order for this to happen and we elaborate on some advantages of this simplification, which ultimately turn the theory developed for restricted polyomino chains not only useful for their generalization but by their direct application over the general polyomino chain setting.

Definition 3.4.1. Let TI_f be a topological degree-based index and $n \geq 1$, we denote by $M_f^1(n)$ the set of all extremal action sequences with n squares and we denote by $M_f^2(n)$ the set of all extremal general action sequences with n squares.

Proposition 3.4.2. Let TI_f be a topological degree-based index and $n \geq 1$, then either $M_f^1(n) \subseteq M_f^2(n)$ or $M_f^1(n) \cap M_f^2(n) = \emptyset$, where every general action sequence in $M_f^2(n) \setminus M_f^1(n)$ uses general actions TT .

Proof. Suppose there exists an extremal general action sequence A that used general actions TT and let B be an extremal action sequence. If $TI_f(B) < TI_f(A)$, then it follows since all extremal action sequences have the same TI_f value, that A has a greater TI_f value than any action sequence. But then, since the set of all action sequences is equal to the set of all general action sequences that do not use the general action TT , it follows all extremal general action sequences must use action TT . Thus $M_f^1(n) \cap M_f^2(n) = \emptyset$ and every general action sequence in $M_f^2(n) \setminus M_f^1(n)$ uses general actions TT .

Now, if $TI_f(B) = TI_f(A)$, it follows in particular all extremal action sequences are extremal general action sequences and hence $M_f^1(n) \subset M_f^2(n)$.

Finally, if there does not exist an extremal general action sequence that uses general actions TT , then it follows all extremal general action sequences are extremal action sequences and therefore we can conclude $M_f^1(n) = M_f^2(n)$.

□

Based on Proposition 3.4.2, in order to have the set of all extremal general action sequences equal to the set of all extremal action sequences, we must focus on giving conditions such that no maximal general action sequence uses the general action TT . However, since our optimization occurs over the set of compressed action sequences, this goal turns into giving conditions such that maximal compressed action sequences do not use general actions ST, CT or ST^* .

So to begin with, we will give conditions such that there is a choice of actions with greater TI_f value than choosing the general action ST^* for problem $M_f(5, C)$. To do this, we will ask exhaustively that some path of decisions using only actions outperforms the use of ST^* , in Figure 3.8 we show graphically all such paths. Also, for the algebraic simplification of the inequalities in Equation (C1) notice we use the uncompression of ST^* as well as Remark 2.1.12 and Remark 3.1.5. To the expression of Equation (C1) we name *Condition 1*.

$$\begin{aligned}
 & g_f(SS) + g_f(SS) + g_f(SC) > g_f(ST^*) & 2g_f(SS) > g_f(CS) + g_f(CC) \\
 & \vee g_f(SS) + g_f(SC) + g_f(CC) > g_f(ST^*) & \Leftrightarrow \vee g_f(SS) > g_f(CS) \\
 & \vee g_f(SC^*) + g_f(CC) + g_f(CC) > g_f(ST^*) & \vee g_f(SS) + g_f(CC) > 2g_f(CS) \\
 & \vee g_f(SC^*) + g_f(CS) + g_f(SC) > g_f(ST^*) & \vee g_f(SC^*) > g_f(CC)
 \end{aligned} \tag{C1}$$

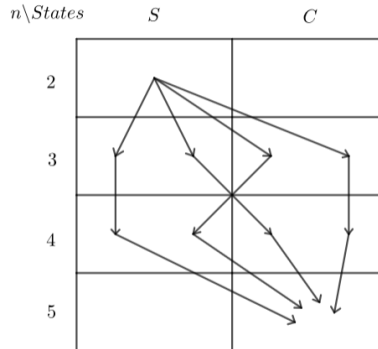


Figure 3.8: All action paths building problem $M_f(5, C)$ from problem $M_f(2, S)$.

Lemma 3.4.3. *Condition 1 is true if and only if no maximal compressed action sequence for problems $M_f(n, S)$ and $M_f(n, C)$ has the compressed action ST^* for all $n \geq 1$.*

Proof. Let us prove the forward implication. So suppose Condition 1 was true, then it is clear for $n < 5$ there cannot exist maximal compressed action sequences corresponding to the problems $M_f(n, S)$ or $M_f(n, C)$ that use the compressed action ST^* , since this action isn't even a possibility for such problems.

Now, for $n = 5$, due to the fact Condition 1 is true, we have there exists at least one compressed action sequence with 5 squares and without compressed actions ST^* such that its TI_f is strictly greater than the one corresponding to the unique compressed action sequence with 5 squares which uses the compressed action ST^* . Thus, it follows no maximal compressed action sequence for the problem $M_f(5, C)$ has the compressed action ST^* . Also, it is clear no maximal compressed action sequence for problem $M_f(5, S)$ uses the compressed action ST^* .

For $n > 5$ suppose there existed a maximal compressed action sequence $(a_1, a_2, \dots, a_N)$ corresponding to problem $M_f(n, E)$ where $E \in \{S, C\}$ and such that it used the compressed action

ST^* . Then, because the compressed action ST^* can only appear in the first 5 squares, we have

$$(a_1, a_2, a_3, \dots, a_N) = (T, SS^*, ST^*, \dots, a_N)$$

But then, by part *a*) of Lemma 3.3.2 it follows (T, SS^*, ST^*) is a maximal compressed action sequence corresponding to problem $M_f(5, C)$ which is a contradiction.

Now for the reverse implication of the statement, since for all $n \geq 1$ no maximal compressed action sequence for problems $M_f(n, S)$ or $M_f(n, C)$ have the compressed action ST^* , it follows in particular this will be true for $M_f(5, C)$. But then, this means there exists a compressed action sequence ending at state C with 5 squares and without the compressed action ST^* such that it has a greater TI_f value than the sequence (T, SS^*, ST^*) . Therefore, because Condition 1 compares the TI_f of every compressed action sequence with 5 squares ending at state C exhaustively with the one of (T, SS^*, ST^*) , it must be some expression in Equation (C1) evaluates true. $\square$

Motivated by the success of Condition 1, we now give conditions such that there are choices of actions with greater TI_f value than choosing the general actions ST or CT respectively, for problem $M_f(6, C)$. To do this, we again exhaustively ask that some path of decisions using only actions outperforms the use of ST or CT , where we graphically show these paths in Figure 3.9. Also, for the algebraic simplification of the inequalities in Equations (C2) and (C3) notice we use the uncompression of the general actions ST and CT as well as Remark 2.1.12 and Remark 3.1.5. To the expressions of Equations (C2) and (C3) we name *Condition 2* and *Condition 3* respectively.

$$\begin{aligned} & g_f(SS) + g_f(SS) + g_f(SC) > g_f(ST) & 3g_f(SS) > 2g_f(CS) + g_f(CC) \\ \vee & g_f(SS) + g_f(SC) + g_f(CC) > g_f(ST) & \vee g_f(SS) > g_f(CS) \\ \vee & g_f(SC) + g_f(CC) + g_f(CC) > g_f(ST) & \vee g_f(SS) + g_f(CC) > 2g_f(CS) \\ \vee & g_f(SC) + g_f(CS) + g_f(SC) > g_f(ST) & \vee g_f(SC^*) > g_f(CC) \end{aligned} \quad (C2)$$

$$\begin{aligned} & g_f(CC) + g_f(CC) + g_f(CC) > g_f(CT) & g_f(CC) + g_f(SS) > 2g_f(CS) \\ \vee & g_f(CC) + g_f(CS) + g_f(SC) > g_f(CT) & \vee g_f(SC^*) > g_f(CC) \\ \vee & g_f(CS) + g_f(SS) + g_f(SC) > g_f(CT) & \vee g_f(SS) + g_f(SC^*) > 2g_f(CC) \\ \vee & g_f(CS) + g_f(SC) + g_f(CC) > g_f(CT) & \end{aligned} \quad (C3)$$

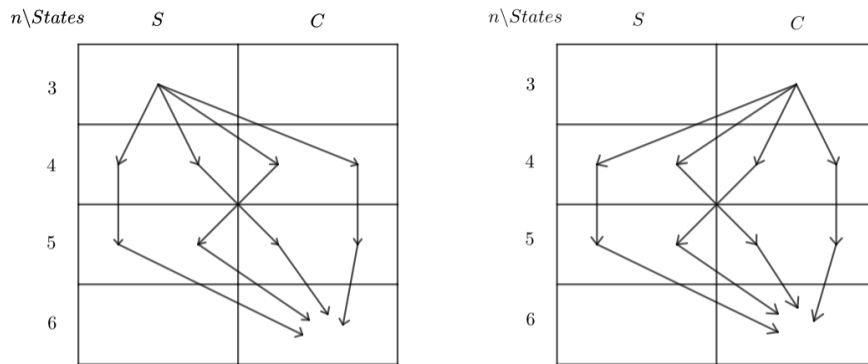


Figure 3.9: All action paths building problem $M_f(6, C)$ from problem $M_f(3, S)$ and problem $M_f(3, C)$ respectively.

Lemma 3.4.4. *If Condition 2 and Condition 3 are true, then no maximal compressed action sequence for problems $M_f(n, S)$ and $M_f(n, C)$ has the compressed actions ST or CT for all $n \geq 1$.*

Proof. Suppose Condition 2 and Condition 3 are true. Then it is clear for $M_f(6, S)$, $M_f(n, S)$ and $M_f(n, C)$ with $n \leq 5$, there exist no maximal compressed action sequences that use the compressed actions ST or CT .

Now, for $M_f(6, C)$ let (a_1, a_2, a_3, ST) be a compressed action sequence with 6 squares. Since in particular Condition 2 is true, it follows there exist compressed actions $\{b_1, b_2, b_3\}$ from which none are ST or CT , such that b_1 comes from state S and b_3 ends at state C , where $g_f(b_1) + g_f(b_2) + g_f(b_3) > g_f(ST)$. Therefore we obtain:

$$TI_f((a_1, a_2, a_3, b_1, b_2, b_3)) > TI_f((a_1, a_2, a_3, ST))$$

Thus no compressed action sequence using the compressed action ST can be maximal for $M_f(6, C)$.

Similarly, because Condition 3 is true we can also conclude no compressed action sequence using the compressed action CT can be maximal for $M_f(6, C)$. Therefore, no maximal compressed action sequence for the problem $M_f(6, C)$ uses compressed actions ST or CT . We'll now proceed by induction.

So suppose for $m \leq n$, with $n \geq 6$, there exist no maximal compressed action sequences for problems $M_f(m, S)$ and $M_f(m, C)$ that use the compressed actions ST or CT . Let $(a_1, a_2, \dots, a_K)$ be a maximal compressed action sequence corresponding to problem $M_f(n+1, S)$, then by part a) of Lemma 3.3.2 it follows $(a_1, a_2, \dots, a_{K-1})$ is a maximal compressed action sequence for the problem $M_f(n_{(K-1)}, E_2^{K-1})$; where $n_{(K-1)} \leq n$. But then, by the induction hypothesis no compressed action in $(a_1, a_2, \dots, a_{K-1})$ can be ST or CT and because a_K must end at state S , it follows it can't be ST or CT either. Hence no maximal compressed action sequence for problem $M_f(n+1, S)$ uses the compressed actions ST or CT .

Now, let $(a_1, a_2, \dots, a_K)$ be a maximal compressed action sequence corresponding to problem $M_f(n+1, C)$. By the same argument as before, we have the compressed action sequence $(a_1, a_2, \dots, a_{K-1})$ does not have compressed actions ST or CT . Assume by way of contradiction a_K was without loss of generality the compressed action ST . Then, because Condition 2 is true we can in the same way as before obtain the compressed actions $\{b_1, b_2, b_3\}$, none of which is ST or CT and satisfy

$$TI_f((a_1, a_2, \dots, a_{K-1}, b_1, b_2, b_3)) > TI_f((a_1, a_2, \dots, a_{K-1}, ST)),$$

where observe both sequences have $n+1$ squares and hence contradict the initial sequence was maximal for $M_f(n+1, C)$. Analogously, we can work out the case $a_K = CT$ by using the fact Condition 3 is true.

Thus we can conclude the statement holds for $n+1$. □

Finally, as a consequence of Lemmas 3.4.3 and 3.4.4, we can establish sufficient conditions such that searching for all extremal general action sequences reduces to finding all extremal action sequences.

Theorem 3.4.5. *If Condition 1, Condition 2 and Condition 3 are true, then no maximal compressed action sequence for problems $M_f(n, S)$ and $M_f(n, C)$ has the compressed actions $\{ST, CT, ST^*\}$ for all $n \geq 1$.*

Corollary 3.4.6. *If Condition 1, Condition 2 and Condition 3 are true, then $M_f^1(n) = M_f^2(n)$ for all $n \geq 1$.*

Notice as a result of Theorem 3.4.5, if for a given degree-based topological index we have Conditions 1, 2 and 3 are satisfied, then we can now solve the problems $M_f(n, S)$, $M_f(n, C)$, $n_t(n, S)$ and $n_t(n, C)$ by their more simpler constructions given by Theorem 2.2.2 and the definition of $n_t(n, E)$ given in Section 2.3 even though we were in the general polyomino chain setting. Moreover, since every result from Sections 2.2.3, 2.3, 2.4 and 2.5.1 depended only on the dynamic programming frameworks and the fact we dealt with action sequences, automatically if Conditions 1, 2 and 3 are true, these results can be also applied to the general setting; being the most important acquisitions the simpler conditions to arrive to cyclic behaviors (Theorem 2.2.11), the conditions for linear and zig-zag chains to be extremal (Theorems 2.4.2, 2.4.5), the results concerning isomorphisms and the characterization of linear and zig-zag chains as the only possible unique extremal chains which are extremal at every previous step (Theorem 2.5.20).

Remark 3.4.7. At this point it is relevant to mention, that although we are guaranteed by Corollary 3.4.6 that the set of all extremal general action sequences is equal to the set of all extremal action sequences, to obtain all general polyomino chains we still need to perform the exhaustive analysis to each action sequence by means of Algorithm 3, which notice is necessary because differently as in the case of polyomino chains, in the general setting we have the set of instructions $\{R, L, D, U\}$ and thus, pivots could still take on possible two values. Nevertheless, by the more simple structure of action sequences, it may be we can establish more results similar to Lemma 3.3.7 which help us fix the interpretations of the sequence. Also, helping ourselves by the results of isomorphisms in action sequences, we can at least get rid of about half the extremal action sequences, which in practice saves a lot of computational work since the application of Algorithm 3 is in the worst case exponential with respect to n . Notice however, in the case of linear and zig-zag chains, we can still assure they are unique and no further analysis is required since these sequences have either zero or only one pivot, meaning their interpretations are the same as in the restricted polyomino chain setting.

Taking Remark 3.4.7 into consideration and observing that the First Zagreb, Second Zagreb, Harmonic, Randić, Geometric Arithmetic, Sum Connectivity and Atom Bond Connectivity indices satisfy Conditions 1, 2 and 3 for both the maximization and minimization problems; while the Augmented Zagreb Index satisfies them just for the minimization problem, we can now extend Corollary 2.4.6 to general polyomino chains.

Corollary 3.4.8. *Let PC_n denote any general polyomino chain with n squares:*

$$\begin{array}{ll} M_i(Li_n) \leq M_i(PC_n) \leq M_i(Z_n) & GA(Z_n) \leq GA(PC_n) \leq GA(Li_n) \\ H(Z_n) \leq H(PC_n) \leq H(Li_n) & \chi(Z_n) \leq \chi(PC_n) \leq \chi(Li_n) \\ R(Z_n) \leq R(PC_n) \leq R(Li_n) & ABC(PC_n) \leq ABC(Z_n). \end{array}$$

Moreover, among all general polyomino chains with n squares, the linear chain Li_n attains the minimum AZI when $n \geq 6$, while the zigzag chain Z_n achieves the minimum AZI for $n = 3, 4, 5$.

3.4.2. Conditions such that there are Extremal General Action Sequences Which are not Action Sequences

In contrast with the previous section, there are cases where the search of all extremal general action sequences cannot be simplified and requires the use of the complete theory developed for general polyomino chains. In this section we proceed to give conditions such that the cases $M_f^1(n) \subset M_f^2(n)$ and $M_f^1(n) \cap M_f^2(n) = \emptyset$ happen, which notice based on Proposition 3.4.2 and the fact we optimize over the set of compressed action sequences, reduces to give conditions such that the compressed actions ST , CT or ST^* are used in maximal compressed action sequences.

Lemma 3.4.9. *Let $n \geq 6$. If for both problems $M_f(n, S)$ and $M_f(n, C)$ there exist maximal compressed action sequences that use the compressed actions ST, CT or ST^* , then for all $m \geq n$ there exist maximal compressed action sequences that use the compressed actions ST, CT or ST^* for both problems $M_f(m, S)$ and $M_f(m, C)$.*

Also, if for problems $M_f(n, S)$ and $M_f(n, C)$ all of their corresponding maximal compressed action sequences use compressed actions ST, CT or ST^ , then for all $m \geq n$ all of the maximal compressed action sequences for problems $M_f(m, S)$ and $M_f(m, C)$ will use compressed actions ST, CT or ST^* .*

Proof. Let $n \geq 6$ and suppose for both problems $M_f(n, S)$ and $M_f(n, C)$ there exist maximal compressed action sequences that use the compressed actions ST, CT or ST^* . Assume for $n \leq m' < m$ we have there exist maximal compressed action sequences for problems $M_f(m', S)$ and $M_f(m', C)$ that use the compressed actions ST, CT or ST^* .

Let $(a_1, a_2, \dots, a_K)$ be a maximal compressed action sequence corresponding to problem $M_f(m, S)$. Then, by part a) of Lemma 3.3.2 we know $(a_1, a_2, \dots, a_{K-1})$ is a maximal compressed action sequence for problem $M_f(n_{(K-1)}, E_2^{K-1})$; where $n_{(K-1)} < m$. Now, because a_K cannot be ST, CT or ST^* it follows $n_{(K-1)} = m-1 \geq n$. Thus, by the induction hypothesis $(a_1, a_2, \dots, a_{K-1})$ uses compressed actions ST, CT or ST^* , which implies so does $(a_1, a_2, \dots, a_K)$.

Now, let $(a_1, a_2, \dots, a_K)$ be a maximal compressed action sequence corresponding to problem $M_f(m, C)$. Then, by part a) of Lemma 3.3.2 we know $(a_1, a_2, \dots, a_{K-1})$ is a maximal compressed action sequence for problem $M_f(n_{(K-1)}, E_2^{K-1})$; where $n_{(K-1)} < m$. If a_K isn't equal to ST or CT , then the argument for the previous case works. Else, if a_K is equal to ST or CT , then it is clear $(a_1, a_2, \dots, a_K)$ uses compressed actions ST, CT or ST^* . Thus in either way the statement holds.

The proof for the second part of the statement is analogous. □

Corollary 3.4.10. *Let $n \geq 6$. If for both problems $M_f(n, S)$ and $M_f(n, C)$ there exist maximal compressed action sequences that use the compressed actions ST, CT or ST^* , then for all $m \geq n$ $M_f^1(m) \subset M_f^2(m)$.*

Also, if for problems $M_f(n, S)$ and $M_f(n, C)$ all of their corresponding maximal compressed action sequences use compressed actions ST, CT or ST^ , then for all $m \geq n$ $M_f^1(m) \cap M_f^2(m) = \emptyset$.*

Theorem 3.4.11. *If Condition 2 and Condition 3 are false, then for $n \geq 6$ there exist maximal compressed action sequences for problem $M_f(n, C)$ that use compressed actions ST, CT or ST^* . Moreover if $(a_1, a_2, \dots, a_N)$ is one of these sequences, then at least one from a_N, a_{N-1}, a_{N-2} must be ST, CT or ST^* .*

Also, if in Condition 2 and Condition 3 we make all inequalities include the equality and they are still false, then for $n \geq 6$ all maximal compressed action sequences for problem $M_f(n, C)$ use compressed actions ST, CT or ST^ . Moreover if $(a_1, a_2, \dots, a_N)$ is one of these sequences, then at least one from a_N, a_{N-1}, a_{N-2} must be ST, CT or ST^* .*

Proof. We will begin by proving the second part of the statement.

So suppose in Condition 2 and Condition 3 we made all inequalities include the equality and that both Condition 2 and Condition 3 were false. Let $(a_1, a_2, \dots, a_N)$ be a compressed action sequence such that no compressed action is equal to ST, CT or ST^* . Then, take its last three compressed actions a_{N-2}, a_{N-1}, a_N (which exist because $n \geq 6$), if $E_1^{N-2} = S$ we have the compressed actions $\{a_{N-2}, a_{N-1}, a_N\}$ were compared with ST in Condition 2 and since this condition is false, it follows

$$g_f(a_{N-2}) + g_f(a_{N-1}) + g_f(a_N) < g_f(ST)$$

which implies

$$TI_f((a_1, a_2, \dots, a_{N-3}, a_{N-2}, a_{N-1}, a_N)) < TI_f((a_1, a_2, \dots, a_{N-3}, ST)).$$

Thus the sequence $(a_1, a_2, \dots, a_N)$ cannot be maximal for problem $M_f(n, C)$. Similarly, if $E_1^{N-2} = C$ we can also conclude $(a_1, a_2, \dots, a_N)$ cannot be maximal for problem $M_f(n, C)$ and therefore all maximal compressed action sequences for $M_f(n, C)$ must use the compressed actions ST, CT or ST^* . Moreover, by the argument used for the proof, it is clear at least one from a_N, a_{N-1}, a_{N-2} must be ST, CT or ST^* .

The first part of the statement follows an analogous argument. □

Remark 3.4.12. Notice in this section we did not establish conditions in terms of Conditions 1, 2 or 3 so that the maximal compressed action sequences corresponding to problem $M_f(n, S)$ use compressed actions ST, CT or ST^* . However, based on Theorem 3.4.11 we at least have a notion of where to start looking for this to happen from which, by Lemma 3.4.9, we know it's sufficient to find at least on value m so that compressed actions ST, CT or ST^* are used and inherit from there onward.

Also, it is important to highlight Theorem 3.4.11 is stronger than just stating at some point there will always be maximal compressed action sequences with compressed actions ST, CT or ST^* , since it guarantees the continuous appearance of these actions in arbitrary large indices in the maximal compressed action sequences, thus giving us a sense of indetachability from the general approach.

As an application of the results of this section, it can be verified for the maximization problem on the Augmented Zagreb Index, that Condition 2 and Condition 3, where they are made to include the equality, are false. Thus for $n \geq 6$ we have all maximal compressed action sequences for problems $M_f(n, C)$ use the compressed actions ST, CT or ST^* . Also, it can be verified there exists a maximal compressed action sequence for problem $M_f(7, S)$ that uses compressed actions ST, CT or ST^* and has a greater TI_f value than the value of $M_f(7, S)$ when computed in the restricted polyomino chain setting. Hence, all maximal compressed action sequences for $M_f(7, S)$ use compressed actions ST, CT or ST^* and by Corollary 3.4.10 for all $m \geq 6$ $M_f^1(m) \cap M_f^2(m) = \emptyset$. As a consequence, the characterization of the extremal action sequences for the AZI shown in Theorem 2.3.6, remains only applicable in the restricted polyomino chain setting and a further analysis over compressed action sequences is needed in order to characterize all of its extremal general polyomino chains.

Conclusion

In this work, we have introduced a general framework for identifying extremal polyomino chains with respect to any degree-based topological index, based on a dynamic programming approach and an encoding of polyomino chains by means of a local geometric representation using the new concept of actions. As a consequence, throughout the work we’ve come to study and give solutions to the majority of the most important problems concerning extremal polyomino chains, retrieving in the way many of the results found in the literature and solving two concrete open problems concerning specific indices.

In particular, we have established methods to find, in the case of the restricted polyomino chain setting, one extremal polyomino chain in linear time with respect to its number of squares and to find all extremal polyomino chains in linear time with respect to their amount, being possible in this case to even find all non-isomorphic extremal polyomino chains with a time complexity of $O(nN^2)$; where N is the amount of extremal polyomino chains and n their number of squares. On the other hand, in the general polyomino chain setting we were also able to find one extremal general polyomino chain in linear time with respect to its number of squares, while being able to find all extremal general polyomino chains with a computational complexity of $O(2^{n/2}nN)$; where N is the number of extremal compressed action sequences and n the number of squares. As applications of these methods, we characterized all extremal restricted polyomino chains for the Augmented Zagreb Index and all extremal general polyomino chains for the generalized Randić Index with parameter equal to -1 , both of which were open problems in the literature.

Moreover, we have also established conditions for the onset of cyclic behaviors on extremal chains, findings concerning the appearance of linear and zig-zag chains, as well as their characterization as the only unique extremal chains which can also be unique and extremal for all previous sub-problems, and conditions in order to tell if solving the extremal polyomino chain problem can be done within the restricted setting, or if the general setting is necessary.

Taking into account the previous results, our dynamic programming framework by means of actions becomes relevant for the study of extremal polyomino chains, proving not only to be a useful tool for concrete computations but also for a more generalized approach to theoretical results on extremal problems in chemical graph theory. However, despite the success of the technique, there are still many problems in need of further study, such as the problem of dealing with parametrized indices as whole and not just for fixed values, the characterization of more special long-term dynamics similar to the way we established conditions for linear and zig-zag chains to appear, the design of more efficient algorithms or additional results similar to Lemma 3.3.7 for identifying realizable general action sequences, and the extension of the extremal problem to other graph families. In particular, the effort for solving the extremal problem for all values of a parametrized index is currently underway and we believe that the proposed methodology of this work provides a valuable foundation for tackling these and other related challenges in the analysis of extremal molecular structures under degree-based and topological indices.

Finally, we provide access to our implementations of all of the methods obtained from our work,

such as the constructive methods for problems $M_f(n, S)$, $n_t(n, E)$, the identification of linear and zig-zag chains as extremal, Algorithms [1](#), [2](#) and [3](#), the identification of non-isomorphic polyomino chains and the analysis of whether the extremal problem can be solved on the restricted polyomino chain setting or not. The code is available at: [Link to the Code](#).

Bibliography

- [1] Akbar Ali, Akhlaq Ahmad Bhatti, and Zahid Raza. Some vertex-degree-based topological indices of polyomino chains. *J. Comput. Theor. Nanosci.*, 12(9):2101–2107, 2015.
- [2] Akbar Ali, Boris Furtula, Ivan Gutman, and Damir Vukičević. Augmented Zagreb index: extremal results and bounds. *MATCH Commun. Math. Comput. Chem.*, 85:211–244, 2021.
- [3] Akbar Ali, Zahid Raza, and Akhlaq Ahmad Bhatti. Bond incident degree (BID) indices of polyomino chains: a unified approach. *Appl. Math. Comput.*, 287:28–37, 2016.
- [4] M. An and L. Xiong. Extremal polyomino chains with respect to general Randić index. *Journal of Combinatorial Optimization*, 31:635–647, 2016.
- [5] Bollobás B. and Erdős P. Graphs of extremal weights. *Ars Combinatoria*, 50:225–233, 1998.
- [6] D. Bonchev and D.H Rouvray. *Chemical Graph Theory, Introduction and Fundamentals*. CRC Press, 1991.
- [7] Michael Cavers. The normalized Laplacian matrix and general Randić index of graphs. *Faculty of Graduate Studies and Research, University of Regina*, 2010.
- [8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press, 2022.
- [9] Alejandro Estrada-Moreno and Juan A. Rodríguez-Velázquez. On the General Randić index of polymeric networks modelled by generalized Sierpiński graphs. *Discrete Applied Mathematics*, 263:140–151, 2019.
- [10] Adeel Farooq, M. Habib, A. Mahboob, et al. Zagreb polynomials and redefined Zagreb indices of dendrimers and polyomino chains. *Open Chem.*, 17(1):1374–1381, 2019.
- [11] Boris Furtula, Ante Graovac, and Damir Vukičević. Augmented Zagreb index. *J. Math. Chem.*, 48:370–380, 2010.
- [12] Boris Furtula, Ivan Gutman, and Matthias Dehmer. On structure-sensitivity of degree-based topological indices. *Appl. Math. Comput.*, 219(17):8973–8978, 2013.
- [13] Leena Rosalind Mary Gnanaraj, Deepa Ganesan, and Muhammad Kamran Siddiqui. Topological indices and QSPR analysis of NSAID drugs. *Polycycl. Aromat. Compd.*, 43(10):9479–9495, 2023.
- [14] Ivan Gutman and Jelena Tošović. Testing the quality of molecular structure descriptors: vertex-degree-based topological indices. *J. Serb. Chem. Soc.*, 78(6):805–810, 2013.
- [15] H.Deng, S. Balachandra, S.K. Ayyaswamy, and Y.B Venkatakrishnan. The harmonic indices of polyomino chains. *National Academy Science Letters*, 37:451–455, 2014.

- [16] Lawrence Hubert, Phipps Arabie, and Jacqueline Meulman. *Combinatorial data analysis: optimization by dynamic programming*. SIAM, 2001.
- [17] Gutman Ivan. Degree-based topological indices. *Croatica Chemica Acta*, 86:351–361, 2013.
- [18] Young Chel Kwun, Adeel Farooq, Waqas Nazeer, Zohaib Zahid, Saba Noreen, and Shin Min Kang. Computations of the M-polynomials and degree-based topological indices for dendrimers and polyomino chains. *Int. J. Anal. Chem.*, 2018(1):1709073, 2018.
- [19] Kier Lemont B. and Hall Lowell H. *Molecular Connectivity in Chemistry and Drug Research*. Academic Press, 1976.
- [20] Xueliang Li and Yongtang Shi. A survey on the Randic index. *MATCH Commun. Math. Comput. Chem*, 59(1):127–156, 2008.
- [21] Manuel Montes-y Morales, Saylé Sigarreta, and Hugo Cruz-Suárez. Extremal degree-based indices of general polyomino chains via dynamic programming. 2026. Manuscript submitted in a peer reviewed journal on January 19, 2026.
- [22] Manuel Montes-y Morales, Saylé Sigarreta, and Hugo Cruz-Suárez. Maximum augmented Zagreb index on polyomino chains. *Applied Mathematics and Computation*, 519:129929, 2026.
- [23] Cruz R. and Rada J. Extremal polyomino chains of VDB topological indices. *Applied Mathematical Sciences*, 9:5371–5388, 2015.
- [24] J. Rada. The linear chain as an extremal value of VDB topological indices of polyomino chains. *Applied Mathematical Sciences*, 8:5133–5143, 2014.
- [25] Milan Randić. Characterization of molecular branching. *Journal of the American Chemical Society*, 97(23):6609–6615, 1975.
- [26] V. S. Shigehalli and Austin Merwin Dsouza. Quantitative structure–property relationship (QSPR) analysis of general Randić index. *J. Algebr. Stat.*, 13:1957–1967, 2022.
- [27] Saylé C. Sigarreta, Saylı M. Sigarreta, and Hugo Cruz-Suárez. On degree-based topological indices of random polyomino chains. *Math. Biosci. Eng.*, 19:8760–8773, 2022.
- [28] Puzyn Thomasz, Leszczynski Jerzy, and Cronin Mark T. *Recent Advances in QSAR Studies: Methods and Applications*. Springer Science & Business Media, 2010.
- [29] Nenad Trinajstić. *Chemical Graph Theory*. CRC Press, 1936.
- [30] Jianxin Wei, Murat Cancan, Atiq Ur Rehman, Muhammad Kamran Siddiqui, Muhammad Nasir, Muhammad Tayyab Younas, and Muhammad Farhan Hanif. On topological indices of remdesivir compound used in treatment of coronavirus (COVID-19). *Polycycl. Aromat. Compd.*, 42(7):4300–4316, 2022.
- [31] Z. Yarahmadi, A.R. Ashrafia, and S. Moradi. Extremal polyomino chains with respect to Zagreb indices. *Applied Mathematics Letters*, 25:166–171, 2012.
- [32] Shahid Zaman, Hafiza Sana Arbab Yaqoob, Asad Ullah, and Mariam Sheikh. QSPR analysis of some novel drugs used in blood cancer treatment via degree-based topological indices and regression models. *Polycycl. Aromat. Compd.*, 44(4):2458–2474, 2024.
- [33] Xing Zhang. A proof of the Jordan Curve Theorem. *Natural Science*, 11:345–360, 2019.

Appendix

Proof of Proposition 1.2.7

In this section we prove Proposition 1.2.7, which was introduced in Section 1.2.1 and was only proved halfway through Lemma 1.2.5. In what follows we first introduce some new definitions concerning Jordan curves, which will be necessary for our approach, since the main challenge will be to show at some step of the constructive process an interior face is made.

Definition C.0.13. Let C be the unit circle $\{(x, y) | x^2 + y^2 = 1\}$, a *Jordan curve* Γ is the image of C under an injective continuous mapping γ into $\mathbb{R}^2$; that is, a simple closed curve on the plane. Let $t(\theta) = (\cos\theta, \sin\theta)$, $\theta \in [0, 2\pi]$ be the natural parametrization of C , then a Jordan curve Γ is said to be a *Jordan polygon* if there is a partition $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ of the interval $[0, 2\pi]$ such that

$$\gamma((\cos\theta, \sin\theta)) = (a_i\theta + b_i, c_i\theta + d_i)$$

for $\theta \in [\theta_i, \theta_{i+1}]$, $i \in \{1, 2, \dots, n\}$ and a_i, b_i, c_i, d_i real constants. We call the pair (γ, Θ) a *realization of the polygon* Γ .

In particular, if (γ, Θ) is a realization of a polygon Γ , we can associate to it a simple graph by naturally defining edges and vertices, and we write $\Gamma := v_1 E_1 v_2 \dots v_n E_n$; where the endpoints of E_i are v_i and v_{i+1} when $i < n$ and v_n and v_1 when $i = n$. Also, we define the notion of *previous* and *next* vertex of v_i as v_{i-1} and v_{i+1} respectively, where the previous vertex to v_1 is v_n and the next vertex of v_n is v_1 . Similarly, if G is a simple graph which is a cycle, we can embed it in $\mathbb{R}^2$ and obtain one of its realizations (γ, Θ) .

Definition C.0.14. Let $\Gamma := v_1 E_1 v_2 \dots v_n E_n$ be a Jordan polygon, p a point in $\mathbb{R}^2$ not in Γ and ℓ_p be a ray starting from p . If $v_i \in \ell_p$, then we define its *type on* ℓ_p as follows:

- If its previous vertex is also in ℓ_p , we say vertex v_i is of *type 3* (see Figure C.10 b), c)).
- Else, let v be the first vertex in ℓ_p intersected after it intersected v_i or v_i itself, such that its next vertex v' isn't in ℓ_p .
 - If the previous vertex of v_i and v' lie on different sides of the line obtained by extending the ray ℓ_p , we say vertex v_i is of *type 1* (see Figure C.10 a), f)).
 - Else, we say vertex v_i is of *type 2* (see Figure C.10 d), e)).

Also, to the edge having as endpoints v_{i-1} and v_i , and the edge having as endpoints v and v' , we name *the arms of vertex* v_i .

Remark C.0.15. Notice in Definition C.0.14 we are guaranteed to find the vertex v because we are dealing with a closed polygon curve and if from a point onward all next vertices were collinear, then the curve would never close.

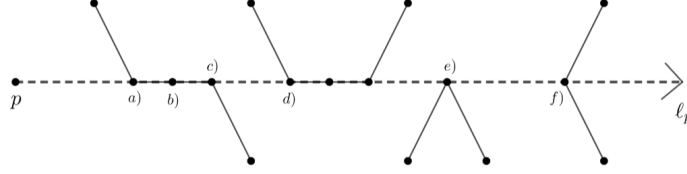


Figure C.10: Types of vertices.

Definition C.0.16. Let $\Gamma := v_1 E_1 v_2 \cdots v_n E_n$ be a Jordan polygon, p a point in $\mathbb{R}^2 \setminus \Gamma$ and ℓ_p a ray starting from p . We define the *intersection index* of p given by ℓ_p and denoted by $I(p)$ as:

$$I(p) := \sum_{q \in \Gamma \cap \ell_p} f(q) \bmod 2$$

where f is defined as $f(q) = 0$ if q is a vertex of type 3, type 2 or q is not a vertex and the endpoints of the edge it belongs to lie on ℓ_p . Else $f(q) = 1$.

We now enunciate the Jordan Curve Theorem for Jordan polygons, which will be the main tool for our proof of Proposition 1.2.7 and whose proof is shown in the following section for the interested reader.

Theorem C.0.17. Let $\Gamma := v_1 E_1 v_2 \cdots v_n E_n$ be a Jordan polygon, then the set $\mathbb{R}^2 \setminus \Gamma$ consists of two connected components, namely the interior and exterior of Γ .

Also, if p is a point in $\mathbb{R}^2 \setminus \Gamma$ and ℓ_p is a ray starting from p . We have if $I(p)$ is even then p lies in the outside of Γ and if $I(p)$ is odd, then p lies in the inside of Γ .

Proposition C.0.18. Let $(I_1, I_2, I_3, \dots, I_n)$ be a sequence of instructions. The sequence is realizable iff at every step of its constructive process the last square has its ending vertices with degree exactly two.

Proof. Since by Lemma 1.2.5 we have already shown if at every step of the graph's constructive process the last square has its ending vertices with degree exactly two then the sequence of instructions is realizable, it only remains to prove if at some point of the constructive process of the graph, the ending vertices have not both degree exactly two, then the graph described by the sequence of instructions is not a polyomino chain.

So because at some point of the constructive process of the graph its ending vertices have not both degree exactly two, there must exist a first square S_i , placed by the instruction I_i , such that when we attach it to the graph described by the sequence $(I_1, \dots, I_{i-1})$, its ending vertices have not both degree exactly two.

Denote the center of the square S_z , given by instruction I_z , as c_z . Suppose without loss of generality $I_i = R$ and that the coordinates of c_i , when the graph is embedded in $\mathbb{R}^2$, were (x, y) . Then, if its ending vertices have not both degree exactly two, it must be there exists $j \in \{1, \dots, i-1\}$ such that the coordinates of the center of square S_j , c_j , are $(x, y+1)$, $(x+1, y+1)$, $(x+1, y)$, $(x+1, y-1)$ or $(x, y-1)$ (see Figure C.11).

We now divide the proof into cases.

- a) If c_j has coordinates $(x, y+1)$, $(x+1, y)$ or $(x, y-1)$, then it follows in the dual graph of $(I_1, \dots, I_n)$, c_i would have an edge with c_j and thus the dual graph would contain a cycle, given by $(c_j, c_{j+1}, \dots, c_i)$. But then this means the dual graph is not a path and hence the graph represented by $(I_1, \dots, I_n)$ is not a polyomino chain.

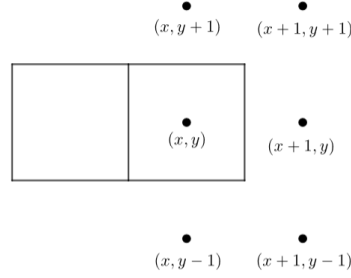


Figure C.11: Possible values for c_j .

- b) Else, suppose without loss of generality the coordinates of the center c_j were $(x+1, y+1)$, we will now define a Jordan polygon by $\Gamma := c_j E_j c_{j+1} \cdots c_i E_i$; where the edges E_z are the natural line segments joining vertices c_z and c_{z+1} when $z < i$ and E_i is the line segment joining vertices c_i and c_j .

Denoting $p := (x, y+1)$ and $q := (x, y-1)$, which we know are not vertices of Γ by case a), it follows $I(p) \neq I(q)$; where we choose ℓ_p and ℓ_q to be vertical rays. In order to see this, notice ℓ_q intersects Γ at the same points as ℓ_p with the exception ℓ_q also intersects c_i , which is of type 1 (see Figure C.12). Hence, we can conclude by Theorem C.0.17 one of either p or q lies within the interior of Γ and therefore we have one of either p or q must be inside an interior face of the graph described by $(I_1, \dots, I_n)$, because points in $\mathbb{R}^2$ can only belong to an edge of the graph, an interior face, the exterior face or be a vertex of the graph and it is clear neither p nor q can be in the edges or be a vertex of the graph described by $(I_1, \dots, I_n)$.

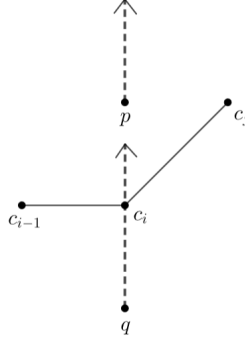


Figure C.12: Deducing $I(p) \neq I(q)$.

Assume without loss of generality p was in one interior face of the graph described by $(I_1, \dots, I_n)$. Then, if we denote by S the unit square in $\mathbb{R}^2$ to which p is center, we have since all points in S are path connected, that by Theorem C.0.17, they all belong to an interior face of the graph, again also arguing they cannot be vertices or belong to edges in the graph.

So if S was equal to the whole interior face to which p belongs, then it would follow in the dual graph, $(c_j, c_{j+1} \cdots, c_i, p)$ forms a cycle. Else, if S was properly contained in the interior face to which p belongs, it would follow the interior face is not equal to a unit square and in either case the sequence $(I_1, \dots, I_n)$ would not be realizable.

□

Jordan Polygon Theorem

In this section we realize the proof of Theorem C.0.17, which we divide into three main lemmas, based on the proof made in [33].

Lemma C.0.19. *Let $\Gamma := v_1E_1v_2 \cdots v_nE_n$ be a Jordan polygon, then $\mathbb{R}^2 \setminus \Gamma$ has at most two connected components.*

Proof. Let $\delta := \min\{d(E_i, E_j) \mid E_i \text{ and } E_j \text{ are not adjacent}\}$ and let $A_i := \{q \in \mathbb{R}^2 \mid d(q, E_i) < \delta\}$. It is clear, by definition, that no points of A_i belong to E_j ; where $j \neq i-1, i, i+1$. Also, notice since in particular the distance of E_i to its endpoints is an upper bound for δ , $A_i \cap \Gamma \subset E_{i-1} \cup E_i \cup E_{i+1}$.

Now, notice $A_i \setminus \Gamma$ has two connected components, which is immediate to see by analytically describing the sets on one side of the polygonal path $A_i \cap \Gamma$ and on the other, which are open and disjoint. We name these components A'_i and A''_i respectively.

Suppose this partition was in particular done for A_i , then for A_{i+1} we can similarly partition it into two connected components and because A_i contains an open ball with center at v_{i+1} , it follows A'_i and A''_i both intersect different components of $A_{i+1} \setminus \Gamma$. Hence, we can denote the components of $A_{i+1} \setminus \Gamma$ as A'_{i+1} and A''_{i+1} depending on which they intersected. Continuing the same process for all edges, we end up with the sets $A'_1, \dots, A'_n$ and $A''_1, \dots, A''_n$ where $A'_i \cap A'_{i+1} \neq \emptyset$ and $A''_i \cap A''_{i+1} \neq \emptyset$. Thus, since all sets A'_i and A''_i are connected, it follows $A'_1 \cup \dots \cup A'_n$ and $A''_1 \cup \dots \cup A''_n$ are connected as well.

Now, let $p \in \mathbb{R}^2 \setminus \Gamma$, we can obtain q a closest point in Γ to p (which exists because p and Γ are compact) and since q is a closest point to p , it follows the line segment from p to q doesn't intersect Γ elsewhere. Also, because q must belong to some A_i , it follows p is path connected to an element in A'_i or A''_i (which can be obtained by taking a point of the line segment from p to q inside a small open ball with center at q and such that the ball is contained in A_i).

Therefore, since every element $p \in \mathbb{R}^2 \setminus \Gamma$ is path connected to $A'_1 \cup \dots \cup A'_n$ or $A''_1 \cup \dots \cup A''_n$, it follows $\mathbb{R}^2 \setminus \Gamma$ has at most two connected components. □

Lemma C.0.20. *Let $\Gamma := v_1E_1v_2 \cdots v_nE_n$ be a Jordan polygon, then $\mathbb{R}^2 \setminus \Gamma$ has at least two connected components.*

Proof. Let p be any point in $\mathbb{R}^2 \setminus \Gamma$ and $m \in \mathbb{R}$. If for every p we fix the slope of ℓ_p to be m , it is clear we partitioned $\mathbb{R}^2 \setminus \Gamma$ into points with odd and even intersection indices. We will now show there is no continuous path between a point with even intersection index and a point with odd intersection index. In order to do so, we prove for any point $p \in \mathbb{R}^2 \setminus \Gamma$ there is a neighborhood around it such that any other point in it has the same intersection index value, where the slopes of the rays considered are equal to m .

So let $p \in \mathbb{R}^2 \setminus \Gamma$, $I(p)$ be its intersection index under ℓ_p with slope equal to m and ℓ_i denote the line passing through the vertex v_i with slope m . Define $\Gamma_1 := \Gamma \cup_{i=1}^n \ell_i$, and take $q \in B(p, \epsilon) \setminus \Gamma$, where $\epsilon = d(p, \Gamma_1)$. We divide the proof $I(q) = I(p)$ into cases depending on the intersection points, z , ℓ_p has with Γ , which we illustrate in Figure C.13:

- a) If z is a point on an edge E_i which isn't a vertex, then if E_i is parallel to ℓ_p , it follows ℓ_q won't intersect it so we have in this case ℓ_q doesn't count any crossing and ℓ_p adds $f(z) = 0$.

If on the contrary E_i is not parallel to ℓ_p , then it follows ℓ_q must also intersect it because ℓ_q has the same slope as ℓ_p , q is on the same side of E_i that p (since $\epsilon = d(p, \Gamma_1) < d(p, \Gamma) < d(p, E_i)$) and since the endpoints of E_i must be on different sides of ℓ_p , the one in the same side as

q must go past ℓ_q , otherwise it would contradict $q \in B(p, \epsilon) \setminus \Gamma$. Hence ℓ_q adds 1 to its intersection index as well as ℓ_p .

- b) If z is a vertex of type 1, it follows only one of z 's arms endpoints is on the same side of ℓ_p that q , and by the same argument as in a), ℓ_q will intersect the edge corresponding to the arm. Thus ℓ_q adds 1 to its intersection index in the same way as ℓ_p .
- c) If z is a vertex of type 2, it follows both z 's arms lie on the same side of ℓ_p . If this side is not the same the one q is in, then ℓ_q intersects none. Otherwise ℓ_q intersects both and in any case adds 0 to its intersection index in the same way as ℓ_p .
- d) If z is a vertex of type 3 and both of the edges that have it as endpoints are parallel to ℓ_p , then it is the same argument as the first part of case a). Else, if one of the edges that have it as endpoint is not parallel to ℓ_p , then no matter whether this edge is on the same side of ℓ_p as q or not, it follows a previously intersected vertex by ℓ_p is of type 1 or 2 and in either case ℓ_q by cases b) and c) adds the same amount as ℓ_p to the intersection index.

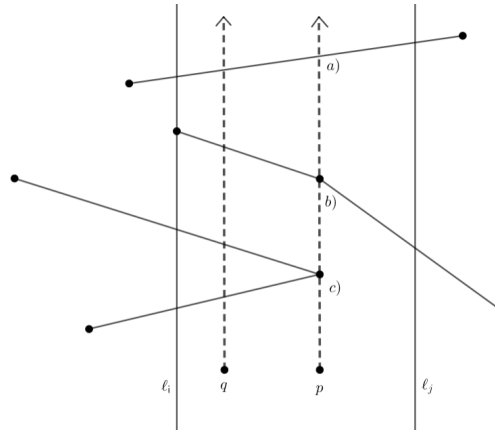


Figure C.13: Cases depending on the intersection points ℓ_p has with Γ .

Finally, we can similarly argue the edges ℓ_q intersects must correspond to equivalent cases on ℓ_p . Therefore we can conclude $I(q) = I(p)$.

So now, let $p, q \in \mathbb{R}^2 \setminus \Gamma$ such that $I(p)$ is even and $I(q)$ is odd. Suppose by way of contradiction that $g : [0, 1] \rightarrow \mathbb{R}^2 \setminus \Gamma$ was a continuous path from p to q (that is, $g(0) = p$ and $g(1) = q$). Then, because we have $g(1)$ is a point with odd intersection index, we can take:

$$t' = \inf\{t \in [0, 1] | g(t) \text{ is a point with odd intersection index}\}.$$

However, since the intersection index is preserved on small neighborhoods, we have there exists $\alpha > 0$ such that the intersection index of $g(t)$ has the same parity as $g(t')$ whenever $t \in (t' - \alpha, t' + \alpha)$ but this contradicts values in $(t' - \alpha, t')$ have even intersection indices and that there is at least one value in $[t', t' + \alpha)$ with odd intersection index. Thus continuous paths can only be made between points with intersection indices of the same parity.

It only remains to show there exist points with even intersection index and points with odd. To see this, let $p \in \mathbb{R}^2 \setminus \Gamma$ such that ℓ_p crosses an edge of Γ first and not through an endpoint. Then, it is clear we can find another point $q \in \ell_p$ immediately after the first crossing such that $q \in \mathbb{R}^2 \setminus \Gamma$ and ℓ_q has the same crossings as ℓ_p from there onward. Therefore, since crossing an edge adds 1 to the intersection index, it follows $I(q) \neq I(p)$ and hence one must be even and the other odd.

Finally, due to the fact points with even and odd intersection index cannot be in the same path connected component, we have $\mathbb{R}^2 \setminus \Gamma$ must have at least two path connected components, which in turn implies it must at least have two connected components. $\square$

Lemma C.0.21. *Let $\Gamma := v_1 E_1 v_2 \cdots v_n E_n$ be a Jordan polygon and $p \in \mathbb{R}^2 \setminus \Gamma$, $I(p)$ is well-defined. That is, for any ray chosen for p , $I(p)$ has the same value.*

Proof. Let $p \in \mathbb{R}^2 \setminus \Gamma$ and ℓ_p some ray starting at p . Since the number of vertices is finite in Γ , we can find an angle range $[-\theta, \theta]$; $\theta \in (0, \pi)$ such that the ray ℓ_p rotated by $\beta \in [-\theta, \theta]$, which we'll denote by $\ell_p(\beta)$, doesn't intersect any vertices of Γ . Then, by applying the exact same arguments as in the proof of Lemma C.0.20, we can show the intersection index of p under the ray $\ell_p(\beta)$ equals that of p under the ray ℓ_p .

Now, since we can clearly make a continuous path $g : [0, 1] \rightarrow [0, 2\pi]$, where the intersection index for p under $\ell_p(0)$ is $I(p)$, we have for all $\beta \in [0, 2\pi]$ the intersection index for p under $\ell_p(\beta)$ is $I(p)$. Otherwise, if this wasn't true for some $\beta \in (0, 2\pi]$, we could obtain:

$$t' = \inf\{t \in [0, 1] \mid \text{the intersection index for } p \text{ with } \ell_p(g(t)) \neq I(p)\}.$$

But then, because for $\ell_p(g(t'))$ we can find a neighborhood $[-\theta, \theta]$ such that $\ell_p(g(t') + \alpha)$ gives the same intersection index as $\ell_p(g(t'))$; $\alpha \in [-\theta, \theta]$, we can conclude t' is not the infimum and thus it must be for all $\beta \in [0, 2\pi]$ p has with $\ell_p(\beta)$ intersection index equal to $I(p)$. $\square$

Theorem C.0.22. *Let $\Gamma := v_1 E_1 v_2 \cdots v_n E_n$ be a Jordan polygon, then the set $\mathbb{R}^2 \setminus \Gamma$ consists of two connected components, namely the interior and exterior of Γ .*

Also, if p is a point in $\mathbb{R}^2 \setminus \Gamma$ and ℓ_p is a ray starting from p . We have if $I(p)$ is even then p lies in the outside of Γ and if $I(p)$ is odd, then p lies in the inside of Γ .

Proof. By Lemmas C.0.19 and C.0.20 it is immediate $\mathbb{R}^2 \setminus \Gamma$ has exactly two connected components and since by Lemma C.0.21 we are allowed to speak of $I(p)$ freely, we have one component corresponds of all points with even intersection index and the other of all points with odd intersection index. Moreover, since for v_i a vertex of Γ we can find a radius r such that $\Gamma \subset B(v_i, r)$, by taking $p \notin B(v_i, r)$ and ℓ_p as a ray starting from p and going in the same direction as $v_i \vec{p}$, we have ℓ_p doesn't intersect Γ and hence $I(p) = 0$. Thus we can conclude points with even intersection index lie outside Γ and points with odd intersection index lie inside. $\square$