

UNIWERSYTET ŚLĄSKI W KATOWICACH
WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH
INSTYTUT INFORMATYKI

MGR GRZEGORZ MOŚ

**Algorytmy agregujące modele o geometrii
heksagonalnej**

ROZPRAWA DOKTORSKA W DYSCYPLINIE INFORMATYKA
W DZIEDZINIE NAUK ŚCISŁYCH I PRZYRODNICZYCH

PROMOTOR:
PROF. DR HAB. MICHAŁ BACZYŃSKI

Katowice, 2026

Tytuł pracy

Algorytmy agregujące modele o geometrii heksagonalnej

Abstrakt

Głównym celem niniejszej dysertacji jest zaproponowanie nowych metod porównywania i agregacji struktur, których kształt oparty jest na siatce heksagonów, a także analizie właściwości otrzymanych metod oraz ich potencjalnych praktycznych zastosowań. Reprezentujemy ścieżki w rozważanej przestrzeni za pomocą ciągów binarnych, gdzie bit 0 oznacza kąt lewostronny między kolejnymi krawędziami, a bit 1 kąt prawostronny. Dzielimy badane struktury na ścieżki i organizujemy je w drzewo ternarne, które następnie reprezentujemy za pomocą rozszerzonych ciągów binarnych. Porównujemy tak reprezentowane obiekty, wykorzystując metryki oraz częściowe porządki, a następnie agregujemy, wyznaczając odpowiednio medoid oraz elementy najmniejsze i największe, minimalne i maksymalne, a także infimum i supremum. Analizujemy użyteczność zaproponowanych metod w praktycznych zastosowaniach, takich jak teoria gier, wiedza rozproszona czy chemia organiczna.

Słowa kluczowe

siatka heksagonów, ciągi binarne, agregacja, porównywanie grafów, porządki częściowe

Thesis title

Aggregation algorithms for models with hexagonal geometry

Abstract

The main goal of this dissertation is to propose new methods for comparing and aggregating structures whose shape is based on a hexagonal grid, as well as to analyze the properties of the resulting methods and their potential practical applications. We represent paths in the considered space using binary sequences, where the bit 0 denotes a left turn angle between consecutive edges, and the bit 1 denotes a right turn angle. We divide the examined structures into paths and organize them into a ternary tree, which is then represented using extended binary sequences. We compare the objects represented this way by employing metrics and partial orders and then aggregate them by determining, respectively, the medoid, as well as the smallest and largest, minimal and maximal elements, along with the infimum and supremum. We analyze the usefulness of the proposed methods in practical applications such as game theory, distributed knowledge, and organic chemistry.

Key words

hexagonal grid, binary sequences, aggregation, graph comparison, partial orders

Spis treści

| | |
|---|-----------|
| Wstęp | 9 |
| 1 Podstawowe zagadnienia matematyczne | 15 |
| 1.1 Punkt, odcinek oraz prosta | 15 |
| 1.2 Obrót punktu oraz kąt między odcinkami | 18 |
| 1.3 Odbicie lustrzane punktu | 25 |
| 1.4 Własności obrotu oraz odbicia lustrzanego | 29 |
| 1.5 Porządki częściowe | 34 |
| 2 (Rozszerzone) ciągi binarne | 37 |
| 2.1 Ciągi binarne | 37 |
| 2.2 Rozszerzenie pojęcia ciągu binarnego | 38 |
| 2.3 Rozszerzony ciąg binarny jako drzewo ternarne | 41 |
| 2.4 Operacje na rozszerzonych ciągach binarnych | 48 |
| 3 Heksastrykty | 55 |
| 3.1 Grafy | 55 |
| 3.2 Heksastrykty | 56 |
| 3.3 Właściwe pokrycia ścieżkowe heksastryktów | 62 |
| 3.4 Reprezentacja heksastryktów rozszerzonymi ciągami binarnymi | 67 |
| 4 Algorytmy agregujące heksastrykty | 73 |
| 4.1 Agregacja z wykorzystaniem metryk | 73 |
| 4.1.1 Odległość Hamminga | 78 |
| 4.1.2 Rozszerzona odległość Hamminga | 78 |
| 4.1.3 Rozszerzona ważona odległość Hamminga | 82 |
| 4.1.4 Odległość Levenshteina | 85 |
| 4.1.5 Medoid | 86 |
| 4.2 Agregacja z wykorzystaniem porządków | 87 |
| 4.2.1 Porządek leksykograficzny | 89 |
| 4.2.2 Porządek epaleksykograficzny | 92 |
| 5 Zastosowania | 97 |
| 5.1 Teoria gier oraz symulacje komputerowe | 100 |
| 5.2 Systemy wieloagentowe oraz wiedza rozproszona | 104 |

| | |
|--|------------|
| 5.3 Chemia organiczna oraz stereoizometria | 108 |
| 6 Zakończenie | 111 |
| A Wyniki pokrewne | 115 |
| A.1 Współrzędne wierzchołków heksastruktury | 115 |
| A.2 Walidacja ciągów binarnych jako reprezentantów heksastruktów | 127 |
| A.3 Zliczanie ścieżek rzędu drugiego na siatce heksagonów | 131 |
| B Implementacje w języku Python | 139 |
| B.1 Funkcja τ | 139 |
| B.2 Funkcja ς | 140 |
| B.3 Funkcja π | 141 |
| B.4 Rozszerzone ciągi binarne | 141 |
| B.5 Metryka dla rozszerzonych ciągów binarnych | 144 |
| B.6 Catanatron | 145 |
| B.7 Zliczanie ścieżek heksastruktów | 147 |
| Bibliografia | 154 |
| Skorowidz rysunków | 160 |
| Skorowidz implementacji | 162 |
| Skorowidz symboli | 163 |
| Skorowidz nazw | 167 |

Wstęp

Postęp technologiczny determinuje konieczność doskonalenia istniejących oraz implementacji nowatorskich metod przetwarzania danych, które mają na celu optymalizację procesów, zwiększenie efektywności obliczeń oraz redukcję złożoności czasowej oraz pamięciowej operacji. Rozwój technik agregacji danych w szczególnym rodzaju grafów odgrywa istotną rolę w tych badaniach ze względu na niewielki stopień zgłębienia tematu w literaturze oraz wyjątkowo skomplikowaną postać analizowanych obiektów. Agregacja struktur skonstruowanych na siatce heksagonów jest procesem wyznaczania reprezentanta dla ustalonego zbioru grafów, których geometria odzwierciedla fragmenty siatki utworzonej z sześciokątów.

Niniejsza dysertacja doktorska poświęcona jest zaproponowaniu nowych metod porównywania oraz agregacji struktur, których kształt oparty jest na siatce heksagonów, nazywanych w rozprawie heksastruktami, a także analizie właściwości otrzymanych metod oraz ich potencjalnych praktycznych zastosowań.

Agregacja, rozumiana jako wyznaczanie reprezentanta dla określonego zbioru obiektów tego samego rodzaju (np. liczb), jest znana od wieków, czego przykładem jest średnia arytmetyczna. Uniwersalność tej definicji uzasadnia jej stosowanie w licznych rozważaniach naukowych, takich jak konstrukcja sieci neuronowych (patrz [1], [14]), normy trójkątne w logice rozmytej (patrz [26]) oraz wielu innych obszarach (patrz [15], [16]). Dynamiczny postęp technologiczny napędza rozwój technik przetwarzania danych. W rezultacie agregacja obejmuje coraz bardziej złożone obiekty, takie jak teksty, zmienne losowe czy grafy (patrz [16], [11]).

Heksastrukty to spójne grafy, w których każda para krawędzi o wspólnym wierzchołku tworzy kąt 120° . Sześciokąt, charakterystyczny dla tych struktur, jest figurą geometryczną obecną w wielu dziedzinach naukowych, której kształt oraz właściwości są powszechnie wykorzystywane. W geometrii jest jednym z trzech wielokątów foremnych (obok trójkąta równobocznego oraz kwadratu), które umożliwiają utworzenie parkietażu foremnego. Parkietaż stanowi problem polegający na wypełnieniu płaszczyzny wielokątami, które są styczne oraz nie pokrywają się wzajemnie. Ponadto krata sześciokątna jest jedną z pięciu dwuwymiarowych krat Bravaisa, które są wykorzystywane m.in. jako:

- struktura algebraiczna do reprezentacji relacji binarnych między ustalonymi elementami,
- struktura porządkowa do reprezentacji zależności między ustalonymi elementami z wykorzystaniem częściowego porządku (patrz [7]),

- struktura krystalograficzna do analizy cienkich warstw materiałów (patrz [23]).

Kształt ten występuje także w chemii oraz fizyce, gdzie związki chemiczne oraz materiały, takie jak benzen czy grafen, przyjmują tę postać (patrz [40], [49]). Co więcej, wiedza rozproszona oraz sieci szkieletowe są zagadnieniami informatycznymi, w których wykorzystuje się ten wzorzec (patrz [17], [13]). Wreszcie sześciokąty znajdują szerokie zastosowanie w teorii gier, gdzie tworzą mapy oraz plansze gier, dla których poszukuje się optymalnych strategii zwycięstwa (np. gra planszowa Catan, patrz [45]).

Głównymi celami niniejszej dysertacji są:

- rozszerzenie pojęcia ciągu binarnego celem reprezentacji rozgałęzionej struktury grafów rozważanych w sposób prostszy, bez utraty informacji o wizualnej postaci obiektów,
- wyprowadzenie wzorów na obrót oraz odbicie lustrzane rozważanych struktur z wykorzystaniem rozszerzonych ciągów binarnych,
- zdefiniowanie metryk pozwalających na miarę niepodobieństwa rozważanych grafów,
- agregacja struktur poprzez zastosowanie otrzymanych metryk w konstrukcji medoidu oraz centroidu,
- zbadanie możliwości wykorzystania częściowego porządku do agregacji rozszerzonych ciągów binarnych,
- zaproponowanie zastosowań otrzymanych metod agregacji w różnych dziedzinach naukowych,
- zaimplementowanie rozszerzonych ciągów binarnych w postaci obiektów oraz otrzymanych metryk oraz metod agregacji w postaci funkcji w języku Python.

W badaniach podejmujemy również następujące zagadnienia pokrewne:

- wyprowadzenie wzoru na całkowitoliczbowe współrzędne wierzchołków rozważanych grafów,
- przedstawienie oraz zbadanie własności układu współrzędnych dla siatki sześciokątów,
- zbadanie możliwości walidacji rozważanych struktur na podstawie ich binarnej reprezentacji,
- zbadanie problemu zliczania rozważanych struktur z uwzględnieniem obrotów oraz odbić lustrzanych.

Niniejsza praca składa się z sześciu rozdziałów oraz dwóch dodatków. W rozdziale pierwszym, zatytułowanym „Podstawowe zagadnienia matematyczne”, prezentujemy teorię z zakresu geometrii oraz zbiorów częściowo uporządkowanych. Definiujemy precyzyjnie zagadnienia dotyczące punktów, odcinków, prostych oraz kątów na płaszczyźnie, które

wykorzystamy w dalszej części pracy. Przedstawiamy w całości konstrukcję macierzy rotacji oraz macierzy odbicia lustrzanego z wykorzystaniem podstawowych narzędzi geometrycznych, tj. funkcje trygonometryczne oraz twierdzenie cosinusów. Badamy własności otrzymanych wyników, m.in. określając zależności między rotacją a odbiciem lustrzanym. Ponadto przytaczamy niezbędne pojęcia oraz wyniki z tematyki zbiorów częściowo uporządkowanych. Teoria omówiona w niniejszym rozdziale została opracowana szczegółowo oraz umożliwia jej zastosowanie w pozostałej części rozprawy.

W rozdziale drugim, zatytułowanym „(Rozszerzone) ciągi binarne”, wprowadzamy nowe pojęcie, jakim jest rozszerzony ciąg binarny, który rozszerza pojęcie ciągu binarnego o możliwość wtrącenia nowego ciągu binarnego między bity istniejącego ciągu binarnego. Umożliwia to reprezentację docelowo badanych obiektów w sposób prostszy oraz łatwiejszy do prezentacji oraz umożliwia wykorzystanie nowego rodzaju metod porównawczych. Określamy, w jaki sposób można porównywać rozszerzone ciągi binarne za pomocą metryk zdefiniowanych na ciągach binarnych. Ponadto badamy przedstawienie struktury rozszerzonych ciągów binarnych w postaci ciągu mniej złożonych rozszerzonych ciągów binarnych oraz wyprowadzamy wzory na obrót oraz odbicie lustrzane rozszerzonych ciągów binarnych.

W rozdziale trzecim, zatytułowanym „Heksastrykty”, definiujemy precyzyjnie wybrane pojęcia z teorii grafów, aby w następnym kroku przejść do definicji właściwej dla tej dysertacji obiektów, czyli heksastryktów. Następnie ustalamy taką formę pokryć ścieżkowych heksastryktów, żeby można było ją wykorzystać do reprezentacji przy użyciu ciągów binarnych oraz rozszerzonych ciągów binarnych.

W rozdziale czwartym, zatytułowanym „Algorytmy agregujące heksastrykty”, wyprowadzamy wzór na metrykę dla rozszerzonych ciągów binarnych na podstawie metryki dla ciągów binarnych. W dalszej kolejności implementujemy ją w równaniach opisujących medoid oraz centroid, badając użyteczność istniejących oraz nowych metryk dla procesu agregacji.

W rozdziale piątym, zatytułowanym „Zastosowania”, proponujemy wykorzystanie otrzymanych metod agregacji w różnych dziedzinach naukowych. Pierwsze zastosowanie dotyczy teorii gier. Poszukujemy strategii wygrywających w grze planszowej Catan. Gra ta polega na budowaniu ścieżek oraz zamków oraz pozyskiwaniu surowców na planszy utworzonej z sześciokątów. Drugie zastosowanie dotyczy inżynierii agentowej. Agregujemy wiedzę rozproszoną w celu otrzymania informacji o rozważanej przestrzeni. Rozważamy zbiór heksastryktów, który odpowiada wiedzy pewnej grupy agentów dotyczącej np. mapy, a następnie agregujemy tę wiedzę, aby określić ogólne własności tej mapy. Trzecie, ostatnie zastosowanie dotyczy chemii organicznej oraz stereoizomerii. Pokazujemy, że wiele węglowodorów można reprezentować, porównywać oraz agregować metodami przedstawionymi w niniejszej pracy.

W rozdziale szóstym, zatytułowanym „Zakończenie”, podsumowujemy otrzymane wyniki oraz opisujemy kierunki dalszych badań.

W dodatku A, zatytułowanym „Wyniki pokrewne”, przedstawiamy rezultaty, które są pośrednio związane z agregacją heksastryktów. Zaczynamy od przedstawienia nowych oraz optymalnych metod obliczania współrzędnych w siatce sześciokątów. Następnie przed-

stawiamy sześciokątny układ współrzędnych. Na końcu zajmujemy się pojęciem walidacji ciągów binarnych, które mogą reprezentować heksastryki, co pozwala na zbadanie zagadnienia zliczania heksastryków o jednoelementowym pokryciu ścieżkowym.

W dodatku B, zatytułowanym „Implementacje w języku Python”, przedstawiamy rozważane obiekty oraz metody zaimplementowane w języku Python, tj. rozszerzone ciągi binarne, metryki, porządki, metody agregacji oraz symulacje.

Wyniki badań przedstawionych w niniejszej dysertacji doktorskiej zostały opublikowane w następujących pracach:

- „An overview of honeycomb-based graph aggregation functions” (patrz [29]) w podrozdziałach 3.2, 4.1 oraz A.3,
- „Honeycomb-Based Polygonal Chains Aggregation Functions” (patrz [31]) w podrozdziałach 3.2, 3.4 oraz 4.1,
- „Exploring Vertex Representation and Cardinality of Aggregation Functions in Honeycomb-based Polygonal Chains” (patrz [30]) w podrozdziałach 3.2, A.1, A.2 oraz A.3,
- „Similarity Functions and Medoids for Honeycomb-Based Structures” (patrz [32]) w podrozdziałach 3.3, 3.4, 4.1 oraz 5.1.
- „Uncertainty of Aggregation: Investigating Dissimilarity between Honeycomb-Based Polygonal Chains” (patrz [33]) w podrozdziałach 4.1, 4.2, 5.1 oraz 5.2.

Badania, rozważania oraz wyniki były prezentowane oraz dyskutowane podczas następujących międzynarodowych konferencji naukowych:

- The 19th World Congress of the International Fuzzy Systems Association and the 12th Conference of the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT 2021) we wrześniu 2021 roku – referat pt. „An overview of honeycomb-based graph aggregation functions” (wyniki przedstawione w podrozdziałach 3.2, 4.1 oraz A.3),
- The 19th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2022) w lipcu 2022 roku – referat pt. „Honeycomb-based polygonal chains aggregation functions” (wyniki przedstawione w rozdziałach 3.2, 3.4 oraz 4.1),
- The 13th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2023) we wrześniu 2023 roku – referat pt. „Exploring Vertex Representation and Cardinality of Aggregation Functions in Honeycomb-based Polygonal Chains” (wyniki przedstawione w podrozdziałach 3.2, A.1, A.2 oraz A.3),
- The 17th Conference on Fuzzy Set Theory and Applications (FSTA 2024) na przełomie stycznia i lutego 2024 roku – referat pt. „Similarity Functions and Medoids for Honeycomb-Based Structures” (wyniki przedstawione w podrozdziałach 3.3, 3.4, 4.1 oraz 5.1),

- The 6th Polish Conference on Artificial Intelligence (PP-RAI 2025) w kwietniu 2025 roku – referat pt. „Uncertainty of Aggregation: Investigating Dissimilarity between Honeycomb-Based Polygonal Chains” (wyniki przedstawione w podrozdziałach 4.1, 4.2, 5.1 oraz 5.2).

Ponadto były one wielokrotnie prezentowane na seminariach naukowych zespołu badawczego „Metody inteligencji obliczeniowej” na Uniwersytecie Śląskim w Katowicach oraz na międzyuczelnianych seminariach naukowych instytutów informatyki Uniwersytetu Śląskiego oraz Uniwersytetu Rzeszowskiego.

Rozdział 1

Podstawowe zagadnienia matematyczne

Rozprawę rozpoczynamy od rozdziału, w którym przedstawiamy fundamentalne zagadnienia oraz wyniki matematyczne, które wykorzystamy do konstrukcji siatki heksagonów oraz struktur projektowanych na tejże siatce, a następnie do ich uporządkowania oraz agregacji. W całej pracy przyjmujemy, że \mathbb{N} jest zbiorem liczb naturalnych bez 0, tzn.

$$\mathbb{N} = \{1, 2, 3, 4, 5, \dots\}.$$

Geometria jest działem matematyki szeroko zbadanym oraz przedstawionym w wielu publikacjach i podręcznikach. Naukowcy prowadzą analizy dotyczące punktów, odcinków, wektorów, prostych oraz innych obiektów geometrycznych oraz badają ich wzajemne relacje oraz zastosowania z różnych perspektyw (przede wszystkim w ramach geometrii obliczeniowej, ale również geometrii algebraicznej, geometrii różniczkowej czy geometrii wykreślnej). Przedstawiamy teorię, opierając się na fundamentalnych własnościach funkcji trygonometrycznych, w celu uniknięcia wprowadzania oraz stosowania skomplikowanych pojęć, które nie będą istotne w dalszych częściach dysertacji. Stanowią one zredukowane, szczególne lub autorsko rozszerzone (np. twierdzenia 1.2.1 oraz 1.3.1) zagadnienia teorii zaprezentowanej w standardowych podręcznikach teoretycznych (patrz [2]).

Następnie przedstawiamy podstawowe zagadnienia teorii mnogości związane z porządkami częściowymi (patrz [20]), które posłużą do uporządkowania wcześniej określonych struktur oraz do ich agregacji przy użyciu pojęć ograniczeń zbiorów, w szczególności infimum oraz supremum.

1.1 Punkt, odcinek oraz prosta

Uporządkowaną parę liczb rzeczywistych nazywamy **punktem**. Niech $p_0 = (x_0, y_0)$ oraz $p_1 = (x_1, y_1)$ będą punktami. Wprowadzamy porządek leksykograficzny na płaszczyźnie \mathbb{R}^2 , tj. taką relację \leq na zbiorze wszystkich punktów, że

$$p_0 \leq p_1 \iff x_0 < x_1 \vee (x_0 = x_1 \wedge y_0 \leq y_1).$$

Wprowadzamy ostry porządek leksykograficzny $<$ na zbiorze wszystkich punktów wzorem:

$$p_0 < p_1 \iff x_0 < x_1 \vee (x_0 = x_1 \wedge y_0 < y_1).$$

Para różnych punktów p_0 oraz p_1 jest końcami **odcinka** s , jeżeli tworzą zbiór punktów $\{\lambda p_0 + (1-\lambda)p_1 : \lambda \in [0, 1]\}$. Odcinek s oznaczamy symbolem $\overline{p_0 p_1}$. **Długość** $|s|$ odcinka s o końcach w punktach $p_0 = (x_0, y_0)$ oraz $p_1 = (x_1, y_1)$ wyrażamy wzorem:

$$|s| = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}.$$

Odcinek z uporządkowaną parą punktów końcowych nazywamy **wektorem** oraz oznaczamy $\overrightarrow{p_0 p_1}$. Niech $a, b, c \in \mathbb{R}$ oraz niech $a \neq 0$ lub $b \neq 0$. **Prostą** l nazywamy zbiór punktów $\{(x, y) : ax + by + c = 0\}$ i oznaczamy $l : ax + by + c = 0$.

Określamy położenie punktu $p' = (x', y')$ względem prostej $l : ax + by + c = 0$. Punkt p' **leży na prostej** l , gdy spełnia jej równanie. Załóżmy, że p' nie leży na niej oraz niech $b \neq 0$. Wówczas równanie prostej redukuje się do postaci $ax + c = 0$. Zatem otrzymujemy dla tej prostej zbiór punktów:

$$\left\{ \left(-\frac{c}{a}, y \right) : y \in \mathbb{R} \right\},$$

czyli wizualnie otrzymujemy pionową prostą przecinającą oś poziomą w punkcie $x = -\frac{c}{a}$. Wówczas możemy określić czy punkt p' jest po lewej stronie lub po prawej stronie prostej l . Jeżeli $a(ax' + c) > 0$, to punkt p' **leży na prawo od prostej** l . Jeżeli natomiast $a(ax' + c) < 0$, to punkt p' **leży na lewo od prostej** l .

Jeżeli $b \neq 0$, to

$$y = -\frac{a}{b}x - \frac{c}{b}.$$

Określamy, czy punkt p' jest powyżej lub poniżej prostej l . Jeżeli $b(ax' + by' + c) > 0$, to punkt p' **leży powyżej prostej** l . Jeżeli $b(ax' + by' + c) < 0$, to punkt p' **leży poniżej prostej** l .

Twierdzenie 1.1.1. *Prosta $l : ax + by + c = 0$ przechodzi przez różne punkty $p_0 = (x_0, y_0)$ oraz $p_1 = (x_1, y_1)$ wtedy i tylko wtedy, gdy spełnione jest równanie:*

$$(x - x_1)(y_0 - y_1) - (y - y_1)(x_0 - x_1) = 0. \quad (1.1.1)$$

Ponadto

$$\begin{cases} a = y_0 - y_1, \\ b = -(x_0 - x_1), \\ c = -x_1(y_0 - y_1) + y_1(x_0 - x_1). \end{cases}$$

Dowód. Niech p_0 oraz p_1 spełniają równanie prostej l . Wówczas otrzymujemy układ równań:

$$\begin{cases} ax_0 + by_0 + c = 0 \\ ax_1 + by_1 + c = 0. \end{cases}$$

Stąd $c = -ax_1 - by_1$ oraz

$$a(x_0 - x_1) + b(y_0 - y_1) = 0.$$

Założmy, że $x_0 \neq x_1$. Stąd $b \neq 0$ (ponieważ w przeciwnym przypadku również $a = 0$ i nie są spełnione warunki prostej) i dalej

$$a = -b \cdot \frac{y_0 - y_1}{x_0 - x_1}$$

oraz

$$c = b \cdot \left(\frac{x_1(y_0 - y_1)}{x_0 - x_1} - y_1 \right).$$

Podstawiając obliczone a oraz c do równania prostej l mamy:

$$-b \cdot \frac{y_0 - y_1}{x_0 - x_1} \cdot x + by + b \cdot \left(\frac{x_1(y_0 - y_1)}{x_0 - x_1} - y_1 \right) = 0.$$

Mnożąc obustronnie przez $x_0 - x_1$ i grupując powyższe równanie otrzymujemy równanie (1.1.1).

Z drugiej strony założmy, że $y_0 \neq y_1$. Stąd $a \neq 0$,

$$b = -a \cdot \frac{x_0 - x_1}{y_0 - y_1}$$

oraz

$$c = -ax_1 + a \cdot \frac{x_0 - x_1}{y_0 - y_1}.$$

Ponownie, podstawiając obliczone b oraz c do równania prostej l mamy:

$$ax - a \cdot \frac{x_0 - x_1}{y_0 - y_1} \cdot y - ax_1 + a \cdot \frac{x_0 - x_1}{y_0 - y_1} = 0.$$

Mnożąc obustronnie przez $y_0 - y_1$ i grupując powyższe równanie otrzymujemy równanie (1.1.1). □

Powiemy, że prosta $l: ax + by + c = 0$ **przechodzi przez punkty p_0 oraz p_1** , gdy spełnia równanie (1.1.1). Określmy teraz położenie punktu $p' = (x', y')$ względem wektora $v = \overrightarrow{p_0 p_1}$ dla różnych punktów $p_0 = (x_0, y_0)$ oraz $p_1 = (x_1, y_1)$. Zauważmy, że możemy określić położenie punktu względem odcinka tak samo jak względem prostej. Uporządkowanie końców odcinka pozwala określić, czy punkt znajduje się po lewej stronie, czy po prawej stronie wektora. Niech $l: ax + by + c = 0$ będzie prostą przechodzącą przez punkty p_0 oraz p_1 . Punkt p' **leży na przedłużeniu wektora v** , gdy spełnia równanie prostej l . Założmy, że punkt p' nie spełnia tego warunku. Co więcej, założmy, że $p_0 < p_1$ (później rozważymy warunek przeciwny). Jeżeli $x_0 = x_1$, tj. v leży na prostej pionowej, to punkt p' **leży na lewo (prawo) od wektora v** , gdy leży na lewo (prawo) od prostej l . Jeżeli $x_0 \neq x_1$, to punkt p' **leży na lewo (prawo) od wektora v** , gdy leży powyżej (poniżej) prostej l . Z drugiej strony, jeżeli $p_1 < p_0$, to wszystkie położenia lewostronne w powyższych definicjach zamieniają się na prawostronne, a położenia prawostronne zamieniają się na lewostronne.

W następującym twierdzeniu pokazujemy konsekwencję w określaniu położenia punktu względem wektora oraz wektora skierowanego przeciwnie:

Twierdzenie 1.1.2. *Jeżeli punkt p' leży na lewo (prawo) od wektora $\overrightarrow{p_1 p_0}$, to leży na prawo (lewo) od wektora $\overrightarrow{p_0 p_1}$.*

Dowód. Niech $p_1 < p_0$. Zauważmy, że

$$\begin{aligned}
& (x - x_1)(y_0 - y_1) - (y - y_1)(x_0 - x_1) \\
&= x(y_0 - y_1) - y(x_0 - x_1) - x_1y_0 + x_1y_1 + x_0y_1 - x_1y_1 \\
&= -x(y_1 - y_0) + y(x_1 - x_0) - x_1y_0 + x_0y_0 + x_0y_1 - x_0y_0 \\
&= -x(y_1 - y_0) + y(x_1 - x_0) - y_0(x_1 - x_0) + x_0(y_1 - y_0) \\
&= -(x - x_0)(y_1 - y_0) + (y - y_0)(x_1 - x_0) \\
&= -((x - x_0)(y_1 - y_0) - (y - y_0)(x_1 - x_0)).
\end{aligned}$$

□

Określenie położenia punktu względem wektora będzie kluczowe do ustalenia położenia kąta wklęsłego między uporządkowaną parą odcinków o wspólnym wierzchołku. Relacja ta zostanie następnie wykorzystana w reprezentacji struktur o kształcie opartym na siatce heksagonów.

1.2 Obrót punktu oraz kąt między odcinkami

W tej części omówimy pojęcie obrotu punktu względem punktu ustalonego oraz pojęcie kąta między dwoma odcinkami. Zaczniemy od motywacji definicji, przedstawiając wzór na wartość kąta tworzonego przez odcinki o wspólnym końcu, a następnie wyprowadzimy wzór opisujący zależność między różnymi końcami rozważanych odcinków. Zauważmy na początku, że przy rozważaniu kąta między dwoma odcinkami bierzemy pod uwagę kąt $\alpha \in [0, \pi]$. Natomiast w przypadku obrotu punktu rozważamy kąt $\alpha \in (-\pi, \pi]$.

Twierdzenie 1.2.1. *Niech p, p_0, p_1 będą takimi punktami, że $p_0 \neq p$ i $p_1 \neq p$ oraz niech $s_0 = \overline{pp_0}$ i $s_1 = \overline{pp_1}$. Wówczas kąt $\alpha \in [0, \pi]$ między wektorami s_0 oraz s_1 wynosi*

$$\alpha = \arccos \left(\frac{(x_0 - x)(x_1 - x) + (y_0 - y)(y_1 - y)}{|s_0| \cdot |s_1|} \right). \quad (1.2.2)$$

Ponadto spełniona jest równość:

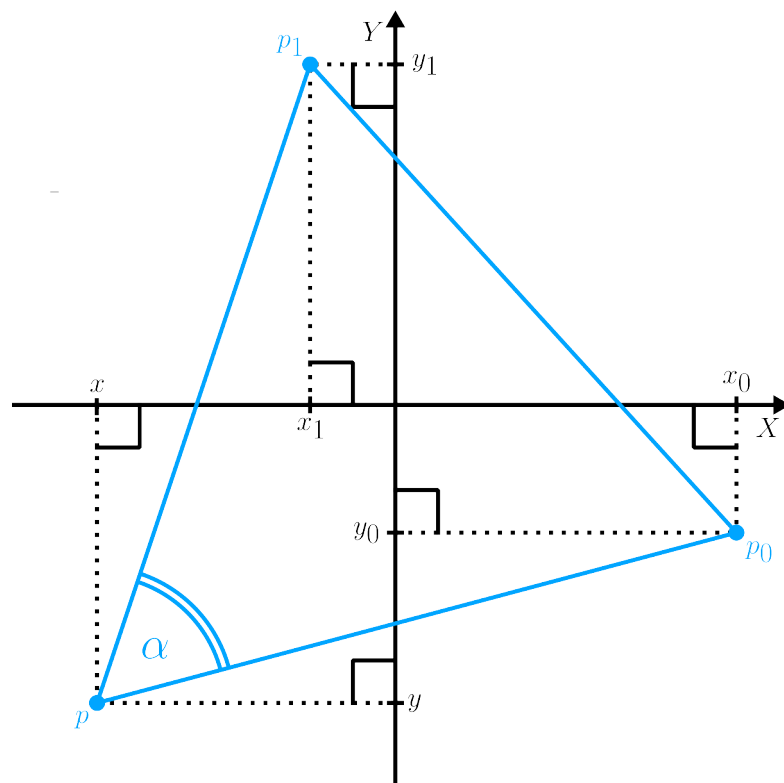
$$p_1 = \frac{|s_1|}{|s_0|} \cdot \begin{bmatrix} \cos(\alpha) & \mp \sin(\alpha) \\ \pm \sin(\alpha) & \cos(\alpha) \end{bmatrix} \cdot (p_0 - p) + p. \quad (1.2.3)$$

Dowód. Zauważmy, że jeżeli $x_0 = x_1$ oraz $y_0 = y_1$, to odcinki s_0 oraz s_1 tworzą kąt o wartości 0, co spełnia równości (1.2.2) oraz (1.2.3).

Niech $p_0 \neq p_1$. Rozważamy trójkąt o wierzchołkach w punktach p, p_0 oraz p_1 (rys. 1.1). W dalszej części dowodu dopuszczamy możliwość degeneracji rozważanego trójkąta. Wówczas twierdzenie cosinusów oraz wykorzystywane własności rozszerzamy na kąty 0° oraz 180° tego trójkąta.

Niech s będzie odcinkiem o końcach w punktach p_0 oraz p_1 . Z twierdzenia cosinusów mamy:

$$|s|^2 = |s_0|^2 + |s_1|^2 - 2 \cdot |s_0| \cdot |s_1| \cdot \cos(\alpha).$$



Rysunek 1.1: Trójkąt o wierzchołkach $p = (x, y)$, $p_0 = (x_0, y_0)$ oraz $p_1 = (x_1, y_1)$, gdzie $x_0, x_1 \geq x$ oraz $y_0, y_1 \geq y$, przedstawiony w kartezjańskim układzie współrzędnych z zaznaczonym kątem α utworzonym przez odcinki o końcach w punktach p i p_0 oraz p i p_1 .

Stąd

$$\cos(\alpha) = \frac{|s_0|^2 + |s_1|^2 - |s|^2}{2 \cdot |s_0| \cdot |s_1|}$$

Zauważmy, że

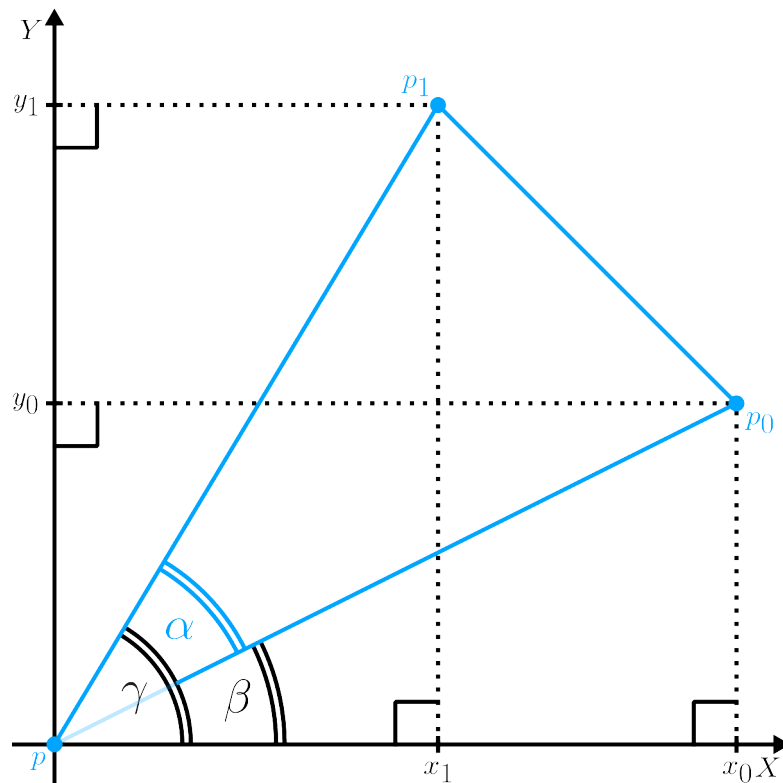
$$\begin{aligned} |s_0|^2 + |s_1|^2 - |s|^2 &= (x_0 - x)^2 + (y_0 - y)^2 + (x_1 - x)^2 + (y_1 - y)^2 - (x_1 - x_0)^2 - (y_1 - y_0)^2 \\ &= -2xx_0 - 2yy_0 - 2xx_1 - 2yy_1 + 2x_0x_1 + 2y_0y_1 + 2x^2 + 2y^2 \\ &= 2 \cdot ((x_0 - x)(x_1 - x) + (y_0 - y)(y_1 - y)). \end{aligned}$$

Ostatecznie otrzymujemy równanie (1.2.2).

Przechodzimy teraz do dowodu równości (1.2.3). Rozważamy trzy przypadki w zależności od wzajemnego położenia punktów p , p_0 oraz p_1 . W pierwszym przypadku rozważamy jedną z możliwości:

1. $x_0, x_1 \geq x$, $y_0, y_1 \geq y$;
2. $x_0, x_1 \geq x$, $y_0, y_1 \leq y$;
3. $x_0, x_1 \leq x$, $y_0, y_1 \geq y$;
4. $x_0, x_1 \leq x$, $y_0, y_1 \leq y$.

Oznacza to, że rozważamy przypadek, w którym punkty p_0 oraz p_1 leżą w tej samej ćwiartce układu współrzędnych względem punktu p . Wówczas kąt α możemy obliczyć jako różnicę kątów tworzonych przez punkty p_1 , p i (x_1, y) oraz przez punkty p_0 , p i (x_0, y) (patrz rys. 1.2).



Rysunek 1.2: Trójkąt o wierzchołkach $p = (x, y)$, $p_0 = (x_0, y_0)$ oraz $p_1 = (x_1, y_1)$, gdzie $x_0, x_1 \geq x$ oraz $y_0, y_1 \geq y$, przedstawiony w kartezjańskim układzie współrzędnych, gdzie punktem przecięcia się osi X oraz Y jest punkt p , z zaznaczonymi kątami: α utworzonym przez wierzchołki p_0 , p i p_1 , β utworzonym przez wierzchołki (x_0, y) , p i p_0 oraz γ utworzonym przez wierzchołki (x_1, y) , p i p_1 .

Założmy, że:

- jeżeli $y_0, y_1 \geq y$, to punkt p_1 leży powyżej prostej przechodzącej przez punkty p i p_0 ,
- jeżeli $y_0, y_1 \leq y$, to punkt p_1 leży poniżej prostej przechodzącej przez punkty p i p_0 .

Oznaczamy:

- kąt α – utworzony przez punkty p_0 , p i p_1 ,
- kąt β – utworzony przez punkty p_0 , p i (x_0, y) ,
- kąt γ – utworzony przez punkty p_1 , p i (x_0, y) .

Wówczas $\gamma = \alpha + \beta$. Stąd

$$\begin{aligned}\frac{|x_1 - x|}{|s_1|} &= \cos(\alpha + \beta) \\ &= \cos(\alpha) \cos(\beta) - \sin(\alpha) \sin(\beta) \\ &= \cos(\alpha) \cdot \frac{|x_0 - x|}{|s_0|} - \sin(\alpha) \cdot \frac{|y_0 - y|}{|s_0|} \\ &= \cos(\alpha) \cdot \frac{|x_0 - x|}{|s_0|} - \sin(\alpha) \cdot \frac{|y_0 - y|}{|s_0|}\end{aligned}$$

oraz

$$\begin{aligned}\frac{|y_1 - y|}{|s_1|} &= \sin(\alpha + \beta) \\ &= \sin(\alpha) \cos(\beta) + \cos(\alpha) \sin(\beta) \\ &= \sin(\alpha) \cdot \frac{|x_0 - x|}{|s_0|} + \cos(\alpha) \cdot \frac{|y_0 - y|}{|s_0|} \\ &= \sin(\alpha) \cdot \frac{|x_0 - x|}{|s_0|} + \cos(\alpha) \cdot \frac{|y_0 - y|}{|s_0|},\end{aligned}$$

co daje jedno z równań (1.2.3) w zależności od rozważanej ćwiartki. Podobne wyniki otrzymamy, gdy przyjmiemy założenia, że:

- jeżeli $y_0, y_1 \geq y$, to punkt p_1 leży poniżej prostej przechodzącej przez punkty p i p_0 ,
- jeżeli $y_0, y_1 \leq y$, to punkt p_1 leży powyżej prostej przechodzącej przez punkty p i p_0 .

W drugim przypadku rozważamy następujące możliwości:

1. $x_0, x_1 \geq x$ i $y_0 \leq y \leq y_1$ lub $y_1 \leq y \leq y_0$,
2. $x_0, x_1 \leq x$ i $y_0 \leq y \leq y_1$ lub $y_1 \leq y \leq y_0$,
3. $y_0, y_1 \geq y$ i $x_0 \leq x \leq x_1$ lub $x_1 \leq x \leq x_0$,
4. $y_0, y_1 \geq y$ i $x_0 \leq x \leq x_1$ lub $x_1 \leq x \leq x_0$.

Wówczas punkty p_0 i p_1 leżą w dwóch sąsiadujących ćwiartkach układu współrzędnych względem punktu p . Wtedy kąt α możemy obliczyć jako sumę kątów utworzonych przez punkty:

- $p_1, p, (x, y_1)$ oraz $p_0, p, (x, y_0)$, jeżeli $y_0, y_1 \geq y$ lub $y_0, y_1 \leq y$,
- $p_1, p, (x_1, y)$ oraz $p_0, p, (x_0, y)$, jeżeli $x_0, x_1 \geq x$ lub $x_0, x_1 \leq x$.

Rozważmy pierwszą możliwość. Oznaczamy (zgodnie z rysunkiem 1.3):

- kąt α – utworzony przez punkty p_0, p i p_1 ,
- kąt β – utworzony przez punkty p_0, p i (x, y_0) ,
- kąt γ – utworzony przez punkty p_1, p i (x, y_1) .

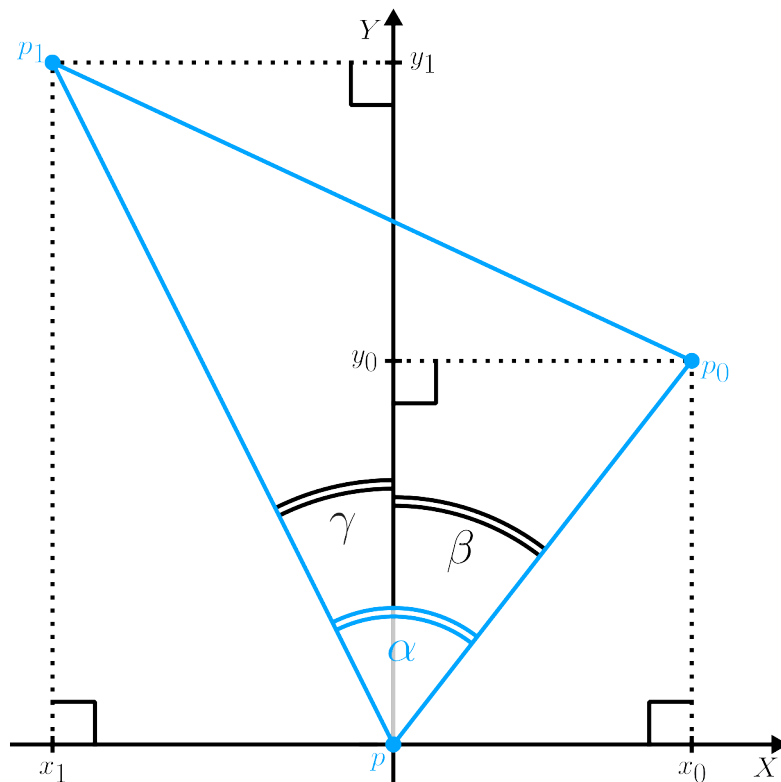
Mamy:

$$\begin{aligned}
 \frac{|x_1 - x|}{|s_1|} &= \sin(\gamma) \\
 &= \sin(\alpha - \beta) \\
 &= \sin(\alpha) \cos(\beta) - \cos(\alpha) \sin(\beta) \\
 &= \sin(\alpha) \cdot \frac{|y_0 - y|}{|s_0|} - \cos(\alpha) \cdot \frac{|x_0 - x|}{|s_0|}
 \end{aligned}$$

oraz:

$$\begin{aligned}
 \frac{|y_1 - y|}{|s_1|} &= \cos(\gamma) \\
 &= \cos(\alpha - \beta) \\
 &= \cos(\alpha) \cos(\beta) + \sin(\alpha) \sin(\beta) \\
 &= \cos(\alpha) \cdot \frac{|y_0 - y|}{|s_0|} + \sin(\alpha) \cdot \frac{|x_0 - x|}{|s_0|},
 \end{aligned}$$

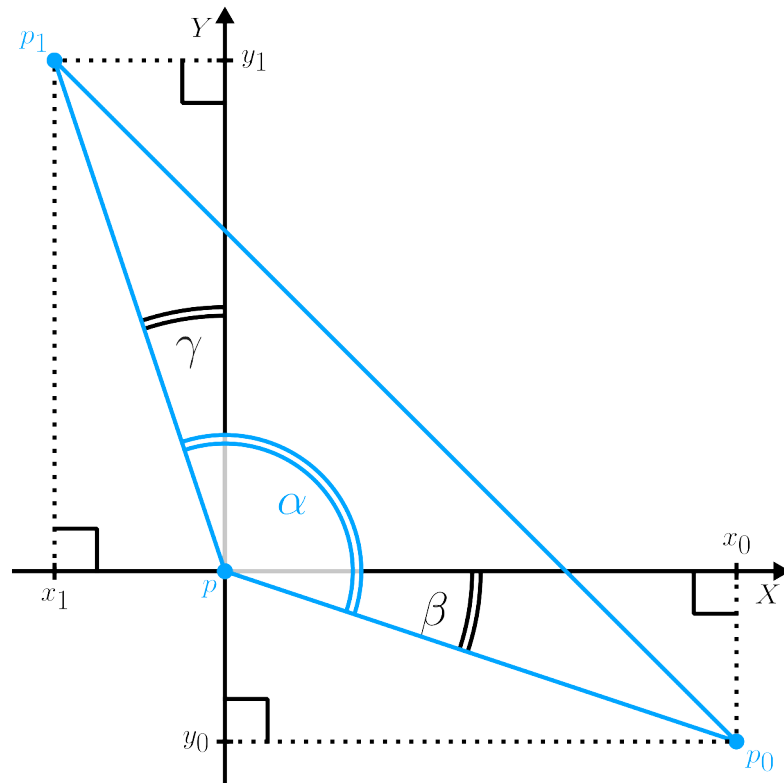
co daje jedno z równań (1.2.3) w zależności od rozważanych ćwiartek. Podobne wyniki otrzymamy przy rozważeniu drugiej możliwości.



Rysunek 1.3: Trójkąt o wierzchołkach $p = (x, y)$, $p_0 = (x_0, y_0)$ i $p_1 = (x_1, y_1)$, gdzie $x_0 \geq x$, $x_1 \leq x$ i $y_0, y_1 \geq y$, przedstawiony w kartezjańskim układzie współrzędnych, gdzie punktem przecięcia się poziomej osi i Y jest punkt p , z zaznaczonymi kątami: α utworzonym przez wierzchołki p_0 , p i p_1 , β utworzonym przez wierzchołki p_0 , p i (x, y_0) oraz γ utworzonym przez wierzchołki (x, y_1) , p i p_1 .

W ostatnim przypadku mamy jedną z możliwości:

1. $x_0 \leq x, x_1 \geq x, y_0 \leq y, y_1 \geq y,$
2. $x_0 \leq x, x_1 \geq x, y_0 \geq y, y_1 \leq y,$
3. $x_0 \geq x, x_1 \leq x, y_0 \leq y, y_1 \geq y,$
4. $x_0 \geq x, x_1 \leq x, y_0 \geq y, y_1 \leq y.$



Rysunek 1.4: Trójkąt o wierzchołkach $p = (x, y)$, $p_0 = (x_0, y_0)$ i $p_1 = (x_1, y_1)$, gdzie $x_0 \geq x, x_1 \leq x, y_0 \leq y$ i $y_1 \geq y$, przedstawiony w kartezjańskim układzie współrzędnych, gdzie punktem przecięcia się poziomej osi i Y jest punkt p , z zaznaczonymi kątami: α utworzonym przez wierzchołki p_0, p i p_1 , β utworzonym przez wierzchołki p_0, p i (x_0, y) oraz γ utworzonym przez wierzchołki $(x, y_1), p$ i p_1 .

Zatem punkty p_0 i p_1 leżą w różnych, niesąsiadujących ze sobą, ćwiartkach układu współrzędnych względem punktu p . Niech l będzie prostą przechodzącą przez punkty p i p_0 . Wówczas kąt α między odcinkami s_0 i s_1 możemy obliczyć jako sumę kątów β, γ i $\frac{\pi}{2}$, gdzie:

- jeżeli p_1 leży nad prostą l oraz $y_0 \leq y$ lub jeżeli p_1 leży pod prostą l oraz $y_0 \geq y$, to β jest wyznaczony przez punkty p_0, p i (x_0, y) oraz γ jest wyznaczony przez punkty p_1, p i (x, y_1) (rys. 1.4),

- jeżeli p_1 leży pod prostą l oraz $y_0 \leq y$ lub jeżeli p_1 leży nad prostą l oraz $y_0 \geq y$, to β jest wyznaczony przez punkty p_0, p , (x, y_0) oraz γ jest wyznaczona przez punkty p_1, p i (x_1, y) .

Dla pierwszego przypadku mamy:

$$\begin{aligned} \frac{|x_1 - x|}{|s_1|} &= \sin(\gamma) \\ &= \sin\left(\alpha - \beta - \frac{\pi}{2}\right) \\ &= -\cos(\alpha - \beta) \\ &= -\cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) \\ &= -\cos(\alpha) \cdot \frac{|y_0 - y|}{|s_0|} - \sin(\alpha) \cdot \frac{|x_0 - x|}{|s_0|} \end{aligned}$$

oraz:

$$\begin{aligned} \frac{|y_1 - y|}{|s_1|} &= \cos(\gamma) \\ &= \cos\left(\alpha - \beta - \frac{\pi}{2}\right) \\ &= \sin(\alpha - \beta) \\ &= \sin(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta) \\ &= \sin(\alpha) \cdot \frac{|x_0 - x|}{|s_0|} - \cos(\alpha) \cdot \frac{|y_0 - y|}{|s_0|}, \end{aligned}$$

co daje jedno z równań (1.2.3) w zależności od rozważanych ćwiartek. Podobne wyniki otrzymamy przy rozważeniu drugiej możliwości. \square

Uwaga 1.2.2. W twierdzeniu 1.2.1, odcinki s_0 i s_1 tworzą również kąt $2\pi - \alpha$, który nazywany jest kątem wklęsłym.

Oznaczamy macierz, która pojawia się w równaniu (1.2.3). **Macierzą rotacji** nazywamy taką funkcję $\text{Rot}: \mathbb{R} \rightarrow \mathbb{R}^{2 \times 2}$, że

$$\text{Rot}(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \quad (1.2.4)$$

Zauważmy następujący fakt wynikający bezpośrednio z własności funkcji trygonometrycznych:

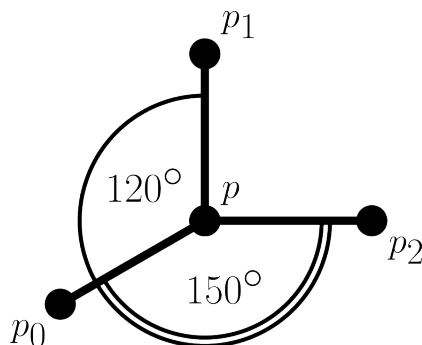
Wniosek 1.2.3. Macierz rotacji jest funkcją okresową o okresie 2π .

Twierdzenie 1.2.1 oraz przyjęte oznaczenie pozwalają nam zdefiniować obrót, czyli przekształcenie punktu p_0 poprzez jego obrót o kąt $\alpha \in \mathbb{R}$ względem ustalonego punktu p . Punkt p_1 nazywamy obrotem punktu p_0 względem punktu p o kąt $\alpha \in \mathbb{R}$, gdy spełnione jest równanie:

$$p_1 = \text{Rot}(\alpha) \cdot (p_0 - p) + p.$$

Odcinki $s_0 = \overline{pp_0}$ i $s_1 = \overline{pp_1}$ tworzą kąt $\alpha \in [0, \pi]$, gdy punkt p_1 jest obrotem punktu p_0 względem punktu p o kąt $\pm\alpha$. Ponadto kąt α jest **prawostronny (lewostronny)** względem odcinka s_0 , gdy punkt p_1 leży po prawej (lewej) stronie względem wektora $\overrightarrow{p_0p}$.

Przykład 1.2.4. Na rysunku 1.5 przedstawiliśmy odcinki $s_0 = \overline{pp_0}$, $s_1 = \overline{pp_1}$ i $s_2 = \overline{pp_2}$. Odcinki s_0 i s_1 tworzą lewostronny kąt 120° . Odcinki s_0 i s_2 tworzą prawostronny kąt 150° .



Rysunek 1.5: Heksastrukt o dwóch wierzchołkach stopnia pierwszego i pozostałych wierzchołkach stopnia drugiego o pokryciu ścieżką reprezentowaną przez ciąg binarny 11110000.

Wniosek 1.2.5. Jeżeli odcinek s_0 tworzy z odcinkiem s_1 kąt prawostronny, to odcinek s_1 tworzy z odcinkiem s_0 kąt lewostronny.

Wniosek 1.2.6. Niech punkt p_1 będzie obrotem punktu p_0 względem punktu p o kąt $\alpha \in \mathbb{R}$. Jeżeli $\alpha \bmod 2\pi < \pi$ ($\alpha \bmod 2\pi > \pi$), to odcinki $\overline{pp_0}$ i $\overline{pp_1}$ tworzą kąt prawostronny (lewostronny).

Przedstawione w tym podrozdziale zagadnienia są kluczowe dla zdefiniowania oraz reprezentacji struktur o kształcie opartym na siatce heksagonów. Jeżeli kolejne krawędzie figury tworzą kąt $\pm 120^\circ$, to określenie położenia kolejnych wierzchołków tej figury sprowadza się do ustalenia położenia kąta, jaki tworzą kolejne krawędzie. Taka właściwość pozwala uprościć analizę geometryczną i sprawia, że reprezentacja kształtu może być realizowana przez sekwencję kątów zamiast konieczności bezpośredniego określania współrzędnych każdego punktu.

1.3 Odbicie lustrzane punktu

Ostatnim przekształceniem geometrycznym, które omówimy w tym rozdziale, jest odbicie lustrzane punktu względem prostej. Odbicie lustrzane stanowi istotne przekształcenie, które koresponduje, np. z koncepcją chiralności w chemii. Pomimo wyraźnego podobieństwa strukturalnego, cząsteczki wykazują zróżnicowane właściwości fizyczne i chemiczne. Tak jak wcześniej rozpoczniemy od przedstawienia motywacji, a następnie wyprowadzimy wzór na współrzędne odbicia lustrzanego punktu względem zadanej prostej.

Twierdzenie 1.3.1. Niech $p = (x, y)$ i $p_0 = (x_0, y_0)$ będą punktami i niech l będzie prostą przechodzącą przez punkt p nachyloną do poziomej osi pod kątem α . Wówczas istnieje taki punkt p_1 , że odcinek o końcach w punktach p_0 i p_1 jest prostopadły do prostej l , punkty p_0 i p_1 są równo oddalone od prostej l oraz spełniona jest równość:

$$p_1 = \begin{bmatrix} \cos(2\alpha) & \sin(2\alpha) \\ \sin(2\alpha) & -\cos(2\alpha) \end{bmatrix} \cdot (p_0 - p) + p. \quad (1.3.5)$$

Dowód. Niech $\beta \in [0, \pi]$ będzie kątem utworzonym przez wierzchołki p_0 , p i (x_0, y) . Zauważmy, że trójkąt utworzony przez te wierzchołki jest prostokątny. Niech $\gamma \in [0, \frac{\pi}{2} + \alpha]$ będzie kątem utworzonym przez odcinek $s_0 = \overline{p_0 p}$ oraz prostą l . Wówczas konstruujemy punkt p_1 obracając punkt p_0 względem punktu p o kąt $\pm 2\gamma$ w taki sposób, aby odcinek $s_1 = \overline{p_1 p}$ tworzył kąt γ z prostą l . Zauważmy, że trójkąt o wierzchołkach p_0 , p , p_1 jest równoramienny, a prosta l dzieli go na dwa trójkąty przystające. Zatem odcinek $s = \overline{p_0 p_1}$ przecina prostą l pod kątem prostym. Ponadto punkty p_0 i p_1 są równo oddalone od prostej l .

Przejdźmy do dowodu równości (1.3.5). Podzielimy problem na kilka przypadków. Przede wszystkim rozważymy $\alpha \geq 0$, gdyż dla wartości ujemnych dowód przebiega analogicznie. Co więcej, zakładamy, że kąt γ jest utworzony przez półprostą utworzoną przez punkty prostej l niemniejsze niż p . Dowód drugiej części przebiega podobnie.

Mamy:

$$\sin(\beta) = \frac{|y_0 - y|}{|s_0|}$$

oraz:

$$\cos(\beta) = \frac{|x_0 - x|}{|s_0|}.$$

Ponadto mamy:

$$\sin(2\beta) = \frac{2 \cdot |x_0 - x| \cdot |y_0 - y|}{|s_0|^2}$$

oraz:

$$\cos(2\beta) = \frac{(x_0 - x)^2 - (y_0 - y)^2}{|s_0|^2}.$$

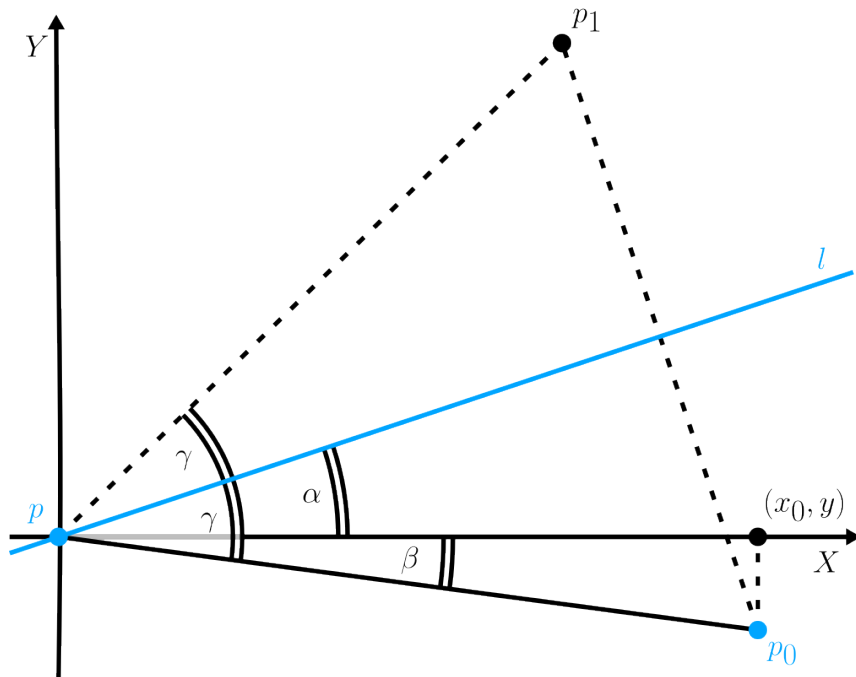
Przypadek 1. Niech $\gamma \in [\alpha, \frac{\pi}{2}]$ oraz niech p_0 leży pod prostą l . Wówczas punkt p_0 leży w czwartej ćwiartce względem punktu p (por. rys. 1.6). Zatem $x_0 \geq x$, $y_0 \leq y$ oraz $\gamma = \alpha + \beta$. Ze wzoru na obrót punktu mamy:

$$\begin{aligned} x_1 - x &= (x_0 - x) \cdot \cos(2\gamma) - (y_0 - y) \cdot \sin(2\gamma) \\ &= (x_0 - x) \cdot \cos(2\alpha + 2\beta) - (y_0 - y) \cdot \sin(2\alpha + 2\beta) \\ &= (x_0 - x) \cdot (\cos(2\alpha) \cos(2\beta) - \sin(2\alpha) \sin(2\beta)) \\ &\quad - (y_0 - y) \cdot (\sin(2\alpha) \cos(2\beta) + \cos(2\alpha) \sin(2\beta)) \\ &= \cos(2\alpha) \cdot ((x_0 - x) \cdot \cos(2\beta) - (y_0 - y) \cdot \sin(2\beta)) \\ &\quad + \sin(2\alpha) \cdot (-(x_0 - x) \cdot \sin(2\beta) - (y_0 - y) \cdot \cos(2\beta)) \\ &= \frac{1}{|s_0|^2} \cdot \cos(2\alpha) \cdot \left((x_0 - x)^3 - (x_0 - x)(y_0 - y)^2 + 2(x_0 - x)(y_0 - y)^2 \right) \\ &\quad + \frac{1}{|s_0|^2} \cdot \sin(2\alpha) \cdot \left(2(x_0 - x)^2(y_0 - y) - (y_0 - y)(x_0 - x)^2 + (y_0 - y)^3 \right) \\ &= \cos(2\alpha) \cdot (x_0 - x) + \sin(2\alpha) \cdot (y_0 - y) \end{aligned}$$

oraz:

$$\begin{aligned}
 y_1 - y &= (x_0 - x) \cdot \sin(2\gamma) + (y_0 - y) \cdot \cos(2\gamma) \\
 &= (x_0 - x) \cdot \sin(2\alpha + 2\beta) + (y_0 - y) \cdot \cos(2\alpha + 2\beta) \\
 &= (x_0 - x) \cdot (\sin(2\alpha) \cos(2\beta) + \cos(2\alpha) \sin(2\beta)) \\
 &\quad + (y_0 - y) \cdot (\cos(2\alpha) \cos(2\beta) - \sin(2\alpha) \sin(2\beta)) \\
 &= \sin(2\alpha) \cdot ((x_0 - x) \cdot \cos(2\beta) - (y_0 - y) \cdot \sin(2\beta)) \\
 &\quad + \cos(2\alpha) \cdot ((x_0 - x) \cdot \sin(2\beta) + (y_0 - y) \cdot \cos(2\beta)) \\
 &= \frac{1}{|s_0|^2} \cdot \sin(2\alpha) \cdot \left((x_0 - x)^3 - (x_0 - x)(y_0 - y)^2 + 2(x_0 - x)y_0 - y^2 \right) \\
 &\quad + \frac{1}{|s_0|^2} \cdot \cos(2\alpha) \cdot \left(-2(x_0 - x)^2(y_0 - y) + (x_0 - x)^2(y_0 - y) - (y_0 - y)^3 \right) \\
 &= \sin(2\alpha) \cdot (x_0 - x) - \cos(2\alpha) \cdot (y_0 - y),
 \end{aligned}$$

co daje równanie (1.3.5).



Rysunek 1.6: Układ współrzędnych z zaznaczoną prostą l pozytywnie nachyloną do poziomej osi i przechodzącą przez punkt p , który tworzy z punktami p_0 , p_1 trójkąt, zaznaczonym punktem (x_0, y) oraz oznaczonymi kątami: α utworzonym przez prostą l oraz poziomą oś, β utworzonym przez punkty p_0 , p , (x_0, y) oraz γ utworzonym przez punkty p_0 , p i prostą l oraz punkty p_1 , p i prostą l . Punkt p_0 leży w czwartej ćwiartce układu współrzędnych względem punktu p . Punkt p_1 powstał przez obrót punktu p_0 względem punktu p o kąt 2γ .

Przypadek 2. Niech $\gamma \in [0, \alpha]$ oraz niech p_0 leży pod prostą l . Wówczas p_0 leży w pierwszej ćwiartce względem punktu p (por. rys. 1.7). Zatem $x_0 \geq x$, $y_0 \geq y$ oraz $\gamma = \alpha - \beta$.

Ponownie wykorzystując wzór na obrót punktu mamy:

$$\begin{aligned}
x_1 - x &= (x_0 - x) \cdot \cos(2\gamma) - (y_0 - y) \cdot \sin(2\gamma) \\
&= (x_0 - x) \cdot \cos(2\alpha - 2\beta) - (y_0 - y) \cdot \sin(2\alpha - 2\beta) \\
&= (x_0 - x) \cdot (\cos(2\alpha) \cos(2\beta) + \sin(2\alpha) \sin(2\beta)) \\
&\quad - (y_0 - y) \cdot (\sin(2\alpha) \cos(2\beta) - \cos(2\alpha) \sin(2\beta)) \\
&= \cos(2\alpha) \cdot ((x_0 - x) \cdot \cos(2\beta) + (y_0 - y) \cdot \sin(2\beta)) \\
&\quad + \sin(2\alpha) \cdot ((x_0 - x) \cdot \sin(2\beta) - (y_0 - y) \cdot \cos(2\beta)) \\
&= \frac{1}{|s_0|^2} \cdot \cos(2\alpha) \cdot \left((x_0 - x)^3 - (x_0 - x)(y_0 - y)^2 + 2(x_0 - x)(y_0 - y)^2 \right) \\
&\quad + \frac{1}{|s_0|^2} \cdot \sin(2\alpha) \cdot \left(2(x_0 - x)^2(y_0 - y) - (y_0 - y)(x_0 - x)^2 + (y_0 - y)^3 \right) \\
&= \cos(2\alpha) \cdot (x_0 - x) + \sin(2\alpha) \cdot (y_0 - y)
\end{aligned}$$

oraz:

$$\begin{aligned}
y_1 - y &= (x_0 - x) \cdot \sin(2\gamma) + (y_0 - y) \cdot \cos(2\gamma) \\
&= (x_0 - x) \cdot \sin(2\alpha - 2\beta) + (y_0 - y) \cdot \cos(2\alpha - 2\beta) \\
&= (x_0 - x) \cdot (\sin(2\alpha) \cos(2\beta) - \cos(2\alpha) \sin(2\beta)) \\
&\quad + (y_0 - y) \cdot (\cos(2\alpha) \cos(2\beta) + \sin(2\alpha) \sin(2\beta)) \\
&= \sin(2\alpha) \cdot ((x_0 - x) \cdot \cos(2\beta) + (y_0 - y) \cdot \sin(2\beta)) \\
&\quad + \cos(2\alpha) \cdot (-(x_0 - x) \cdot \sin(2\beta) + (y_0 - y) \cdot \cos(2\beta)) \\
&= \frac{1}{|s_0|^2} \cdot \sin(2\alpha) \cdot \left((x_0 - x)^3 - (x_0 - x)(y_0 - y)^2 + 2(x_0 - x)y_0 - y^2 \right) \\
&\quad + \frac{1}{|s_0|^2} \cdot \cos(2\alpha) \cdot \left(-2(x_0 - x)^2(y_0 - y) + (x_0 - x)^2(y_0 - y) - (y_0 - y)^3 \right) \\
&= \sin(2\alpha) \cdot (x_0 - x) - \cos(2\alpha) \cdot (y_0 - y),
\end{aligned}$$

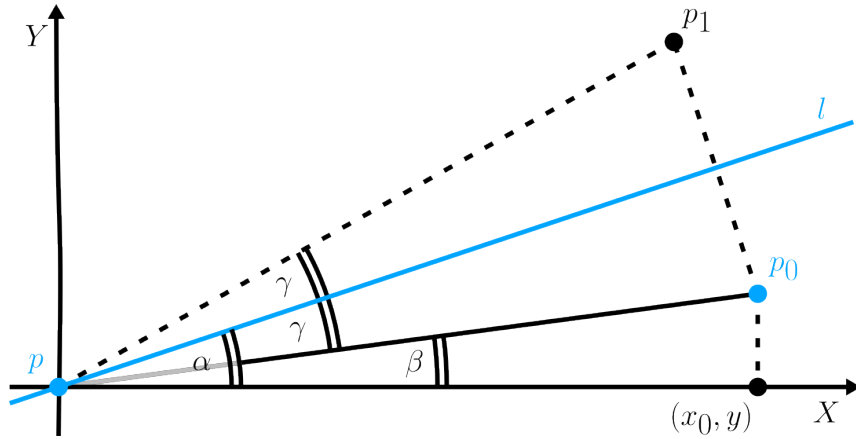
co również daje równanie (1.3.5).

Przypadek 3. Niech $\gamma \in [\alpha, \frac{\pi}{2} - \alpha]$ oraz niech punkt p_0 leży powyżej prostej l (por. rys. 1.8). Wówczas rozważamy kąt -2γ , ponieważ obracamy punkt p_1 w przeciwną stronę. Stąd $x_0 \geq x$, $y_0 \geq y$ oraz $-2\gamma = 2\alpha - 2\beta$. Otrzymujemy takie same równania jak w przypadku 2, co zamyka ten przypadek.

Przypadek 4. Niech $\gamma \in [\frac{\pi}{2} - \alpha, \frac{\pi}{2} + \alpha]$ oraz niech p_0 leży powyżej prostej l . Wówczas punkt p_0 leży w drugiej ćwiartce względem punktu p (por. rys. 1.9). Stąd $x_0 \leq x$, $y_0 \geq y$ oraz $\gamma = \pi - \alpha - \beta$. Ponownie rozważamy kąt -2γ , więc $-2\gamma = 2\alpha + 2\beta$. Otrzymujemy takie same równania jak w przypadku 1, co zamyka ten przypadek i kończy dowód. \square

Oznaczmy macierz, która pojawia się w równaniu (1.3.5). **Macierzą odbicia** nazywamy taką funkcję $\text{Ref} : \mathbb{R} \rightarrow \mathbb{R}^{2 \times 2}$, że

$$\text{Ref}(\alpha) = \begin{bmatrix} \cos(2\alpha) & \sin(2\alpha) \\ \sin(2\alpha) & -\cos(2\alpha) \end{bmatrix}. \quad (1.3.6)$$



Rysunek 1.7: Układ współrzędnych z zaznaczoną prostą l pozytywnie nachyloną do poziomej osi i przechodzącą przez punkt p , która tworzy z punktami p_0 , p_1 trójkąt, zaznaczonym punktem (x_0, y) oraz oznaczonymi kątami: α utworzonym przez prostą l oraz poziomą oś, β utworzonym przez punkty p_0 , p , (x_0, y) oraz γ utworzonym przez punkty p_0 , p i prostą l oraz punkty p_1 , p i prostą l . Punkt p_0 leży w pierwszej ćwiartce układu współrzędnych względem punktu p poniżej prostej l . Punkt p_1 powstał przez obrót punktu p_0 względem punktu p o kąt 2γ .

Punkt p_1 jest **odbiciem lustrzanym** punktu p_0 ze względu na prostą przechodzącą przez punkt p nachyloną do poziomej osi pod kątem $\alpha \in \mathbb{R}$, gdy spełnione jest równanie:

$$p_1 = \text{Ref}(\alpha) \cdot (p_0 - p) + p. \quad (1.3.7)$$

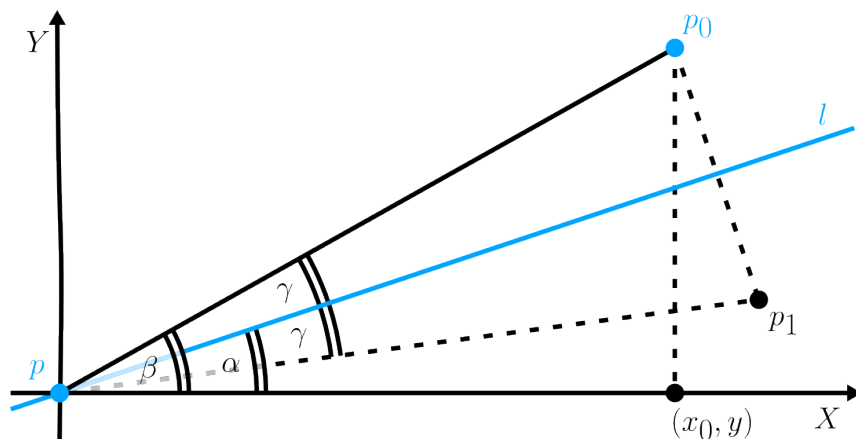
1.4 Własności obrotu oraz odbicia lustrzanego

Na zakończenie rozważań geometrycznych przedstawimy szereg własności spełnianych przez obroty oraz odbicia lustrzane. Zaczniemy od wyników dotyczących odwzorowań Rot oraz Ref , aby lepiej zrozumieć ich wpływ na badaną figurę geometryczną.

Twierdzenie 1.4.1. *Niech $\alpha, \beta \in \mathbb{R}$ i $n \in \mathbb{Z}$. Wówczas:*

1. $\text{Rot}(\alpha) \cdot \text{Rot}(\beta) = \text{Rot}(\alpha + \beta)$,
2. $\text{Ref}(\alpha) \cdot \text{Ref}(\beta) = \text{Rot}(2\alpha - 2\beta)$,
3. $\text{Rot}(\alpha) \cdot \text{Ref}(\beta) = \text{Ref}(\frac{1}{2}\alpha + \beta)$,
4. $\text{Ref}(\alpha) \cdot \text{Rot}(\beta) = \text{Ref}(\alpha - \frac{1}{2}\beta)$,
5. $\text{Rot}^n(\alpha) = \text{Rot}(n \cdot \alpha)$,
6. $\text{Ref}^{2n}(\alpha) = I$,
7. $\text{Ref}^{2n+1}(\alpha) = \text{Ref}(\alpha)$.

Dowód. Niech $\alpha, \beta \in \mathbb{R}$ i $n \in \mathbb{Z}$.



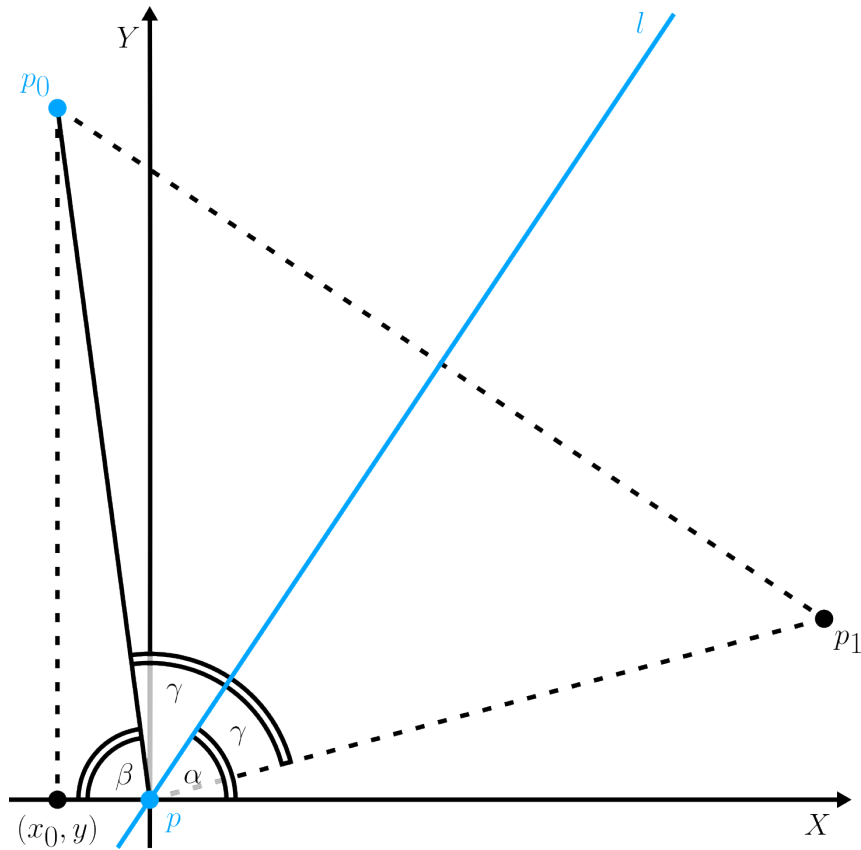
Rysunek 1.8: Układ współrzędnych z zaznaczoną prostą l pozytywnie nachyloną do poziomej osi i przechodzącą przez punkt p , który tworzy z punktami p_0, p_1 trójkąt, zaznaczonym punktem (x_0, y) oraz oznaczonymi kątami: α utworzonym przez prostą l oraz poziomą oś, β utworzonym przez punkty $p_0, p, (x_0, y)$ oraz γ utworzonym przez punkty p_0, p i prostą l oraz punkty p_1, p i prostą l . Punkt p_0 leży w pierwszej ćwiartce układu współrzędnych względem punktu p powyżej prostej l . Punkt p_1 powstał przez obrót punktu p_0 względem punktu p o kąt -2γ .

1. Mamy:

$$\begin{aligned} \text{Rot}(\alpha) \cdot \text{Rot}(\beta) &= \begin{bmatrix} \cos(\alpha) \cos(\beta) - \sin(\alpha) \sin(\beta) & -\cos(\alpha) \sin(\beta) - \sin(\alpha) \cos(\beta) \\ \sin(\alpha) \cos(\beta) + \cos(\alpha) \sin(\beta) & -\sin(\alpha) \sin(\beta) + \cos(\alpha) \cos(\beta) \end{bmatrix} \\ &= \begin{bmatrix} \cos(\alpha + \beta) & -\sin(\alpha + \beta) \\ \sin(\alpha + \beta) & \cos(\alpha + \beta) \end{bmatrix} \\ &= \text{Rot}(\alpha + \beta). \end{aligned}$$

2. Mamy:

$$\begin{aligned} \text{Ref}(\alpha) \cdot \text{Ref}(\beta) &= \begin{bmatrix} \cos(2\alpha) \cos(2\beta) + \sin(2\alpha) \sin(2\beta) & \cos(2\alpha) \sin(2\beta) - \sin(2\alpha) \cos(2\beta) \\ \sin(2\alpha) \cos(2\beta) - \cos(2\alpha) \sin(2\beta) & \sin(2\alpha) \sin(2\beta) + \cos(2\alpha) \cos(2\beta) \end{bmatrix} \\ &= \begin{bmatrix} \cos(2\alpha - 2\beta) & -\sin(2\alpha - 2\beta) \\ \sin(2\alpha - 2\beta) & \cos(2\alpha - 2\beta) \end{bmatrix} \\ &= \text{Rot}(2\alpha - 2\beta). \end{aligned}$$



Rysunek 1.9: Układ współrzędnych z zaznaczoną prostą l pozytywnie nachyloną do poziomej osi i przechodzącą przez punkt p , który tworzy z punktami p_0, p_1 trójkąt, zaznaczonym punktem (x_0, y) oraz oznaczonymi kątami: α utworzonym przez prostą l oraz poziomą oś, β utworzonym przez punkty $p_0, p, (x_0, y)$ oraz γ utworzonym przez punkty p_0, p i prostą l oraz punkty p_1, p i prostą l . Punkt p_0 leży w drugiej ćwiartce układu współrzędnych względem punktu p powyżej prostej l tworząc z nią kąt $\gamma \leq \frac{\pi}{2}$. Punkt p_1 powstał przez obrót punktu p_0 względem punktu p o kąt -2γ .

3. Mamy:

$$\begin{aligned}
 \text{Rot}(\alpha) \cdot \text{Ref}(\beta) &= \begin{bmatrix} \cos(\alpha) \cos(2\beta) - \sin(\alpha) \sin(2\beta) & \cos(\alpha) \sin(2\beta) + \sin(\alpha) \cos(2\beta) \\ \sin(\alpha) \cos(2\beta) + \cos(\alpha) \sin(2\beta) & \sin(\alpha) \sin(2\beta) - \cos(\alpha) \cos(2\beta) \end{bmatrix} \\
 &= \begin{bmatrix} \cos(\alpha + 2\beta) & \sin(\alpha + 2\beta) \\ \sin(\alpha + 2\beta) & -\cos(\alpha + 2\beta) \end{bmatrix} \\
 &= \begin{bmatrix} \cos(2(\frac{1}{2}\alpha + \beta)) & \sin(2(\frac{1}{2}\alpha + \beta)) \\ \sin(2(\frac{1}{2}\alpha + \beta)) & -\cos(2(\frac{1}{2}\alpha + \beta)) \end{bmatrix} \\
 &= \text{Ref}(\frac{1}{2}\alpha + \beta).
 \end{aligned}$$

4. Mamy:

$$\begin{aligned}
 \text{Ref}(\alpha) \cdot \text{Rot}(\beta) &= \begin{bmatrix} \cos(2\alpha) \cos(\beta) + \sin(2\alpha) \sin(\beta) & -\cos(2\alpha) \sin(\beta) + \sin(2\alpha) \cos(\beta) \\ \sin(2\alpha) \cos(\beta) - \cos(2\alpha) \sin(\beta) & -\sin(2\alpha) \sin(\beta) - \cos(2\alpha) \cos(\beta) \end{bmatrix} \\
 &= \begin{bmatrix} \cos(2\alpha - \beta) & \sin(2\alpha - \beta) \\ \sin(2\alpha - \beta) & -\cos(2\alpha - \beta) \end{bmatrix} \\
 &= \begin{bmatrix} \cos(2(\alpha - \frac{1}{2}\beta)) & \sin(2(\alpha - \frac{1}{2}\beta)) \\ \sin(2(\alpha - \frac{1}{2}\beta)) & -\cos(2(\alpha - \frac{1}{2}\beta)) \end{bmatrix} \\
 &= \text{Ref}\left(\alpha - \frac{1}{2}\beta\right).
 \end{aligned}$$

5. Równość wynika bezpośrednio z punktu 1. oraz z indukcji matematycznej.

6. Równość wynika bezpośrednio z punktu 2.

7. Równość wynika bezpośrednio z punktów 6.

□

Wniosek 1.4.2. *Funkcja Rot jest przemienna względem mnożenia, tj.*

$$\text{Rot}(\alpha) \cdot \text{Rot}(\beta) = \text{Rot}(\beta) \cdot \text{Rot}(\alpha)$$

dla $\alpha, \beta \in \mathbb{R}$.

Powyższe formuły potwierdzają intuicyjny wpływ obrotów oraz odbić lustrzanych na rozważaną figurę geometryczną. Składanie obrotów odpowiada sumowaniu kątów, co wyraża kolejne rotacje figury względem początku układu współrzędnych, przeciwnie do ruchu wskazówek zegara, o odpowiednie kąty. Natomiast składanie odbić lustrzanych uwzględnia zależność między prostymi, względem których dokonuje się odbicia badanego punktu. W efekcie otrzymujemy rotację punktu o dwukrotność różnicy kątów między tymi prostymi. Na koniec podkreślamy przemienność działania rotacji, co oznacza, że kolejność, w jakiej stosujemy kąty kolejnych rotacji badanego punktu, nie ma znaczenia i nie wpływa na wynik końcowy.

Kolejne wyniki są kluczowe dla definiowania struktur opartych na siatce heksagonalnej. Wykazujemy w nich, że kształt badanego obiektu pozostaje niezmienny zarówno przy jego obrocie (por. twierdzenia 1.4.3 i 1.4.4), jak i przy odbiciu lustrzanym (por. twierdzenia 1.4.5 i 1.4.6). W związku z tym skupimy się na badaniu struktury obiektu z uwzględnieniem jego kształtu, niezależnie od konkretnych współrzędnych w płaszczyźnie. Taka analiza pozwala ująć właściwości geometryczne obiektu w sposób bardziej uniwersalny.

Twierdzenie 1.4.3. *Niech p_0 i p_1 będą różnymi punktami. Niech s będzie odcinkiem o końcach w p_0 i p_1 . Wówczas długość odcinka s' o końcach w punktach p'_0 i p'_1 powstałych przez rotację punktów, odpowiednio, p_0 i p_1 , względem punktu $p = (x, y)$ o kąt $\alpha \in \mathbb{R}$, jest równa długości odcinka s .*

Dowód. Niech $p = (x, y)$, $p_0 = (x_0, y_0)$, $p_1 = (x_1, y_1)$, $p'_0 = (x'_0, y'_0)$ i $p'_1 = (x'_1, y'_1)$. Stąd

$$\begin{aligned}(x'_1 - x'_0)^2 &= ((x_1 - x) \cos(\alpha) - (y_1 - y) \sin(\alpha) - (x_0 - x) \cos(\alpha) + (y_0 - y) \sin(\alpha))^2 \\ &= ((x_1 - x_0) \cos(\alpha) - (y_1 - y_0) \sin(\alpha))^2 \\ &= (x_1 - x_0)^2 \cos^2(\alpha) + (y_1 - y_0)^2 \sin^2(\alpha) - 2(x_1 - x_0) \cdot (y_1 - y_0) \sin(\alpha) \cos(\alpha)\end{aligned}$$

oraz

$$\begin{aligned}(y'_1 - y'_0)^2 &= ((x_1 - x) \sin(\alpha) + (y_1 - y) \cos(\alpha) - (x_0 - x) \sin(\alpha) - (y_0 - y) \cos(\alpha))^2 \\ &= ((x_1 - x_0) \sin(\alpha) + (y_1 - y_0) \cos(\alpha))^2 \\ &= (x_1 - x_0)^2 \sin^2(\alpha) + (y_1 - y_0)^2 \cos^2(\alpha) + 2(x_1 - x_0) \cdot (y_1 - y_0) \sin(\alpha) \cos(\alpha).\end{aligned}$$

Zatem

$$|s'| = \sqrt{(x'_1 - x'_0)^2 + (y'_1 - y'_0)^2} = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} = |s|.$$

□

Twierdzenie 1.4.4. *Rozważamy różne punkty $p_0 = (x_0, y_0)$, $p_1 = (x_1, y_1)$ i $p_2 = (x_2, y_2)$. Niech s_0 będzie odcinkiem o końcach w punktach p_0 i p_1 oraz niech s_1 będzie odcinkiem o końcach w punktach p_1 i p_2 . Ponadto niech s'_0 i s'_1 będą odcinkami o końcach, odpowiednio, w punktach p'_0 i p'_1 oraz p'_1 i p'_2 , gdzie p'_0 , p'_1 i p'_2 powstały przez obrót punktów, odpowiednio, p_0 , p_1 i p_2 , o kąt α względem punktu $p = (x, y)$. Wówczas odcinki s'_0 i s'_1 tworzą ten sam kąt co odcinki s_0 i s_1 .*

Dowód. Wiemy z twierdzenia 1.4.3, że $|s_0| = |s'_0|$ oraz $|s_1| = |s'_1|$. Niech $p'_0 = (x'_0, y'_0)$, $p'_1 = (x'_1, y'_1)$ i $p'_2 = (x'_2, y'_2)$. Niech s_0 i s_1 tworzą kąt φ . Wówczas

$$\begin{aligned}\text{Rot}(\varphi) \cdot (p'_0 - p'_1) + p'_1 &= \text{Rot}(\varphi) \cdot \text{Rot}(\alpha) \cdot (p_0 - p_1) + \text{Rot}(\alpha) \cdot (p_1 - p) + p \\ &= \text{Rot}(\alpha) \cdot (\text{Rot}(\varphi) \cdot (p_0 - p_1) + p_1 - p) + p \\ &= \text{Rot}(\alpha) \cdot (p_2 - p) + p \\ &= p'_2.\end{aligned}$$

□

Twierdzenie 1.4.5. *Niech p_0 i p_1 będą różnymi punktami. Niech s będzie odcinkiem o końcach w p_0 i p_1 . Wówczas długość odcinka s' o końcach w punktach p'_0 i p'_1 powstałych przez odbicie lustrzane, odpowiednio, p_0 i p_1 , względem prostej l przechodzącej przez punkt p nachylonej do poziomej osi pod kątem $\alpha \in \mathbb{R}$, jest równa długości odcinka s .*

Dowód. Niech $p = (x, y)$, $p_0 = (x_0, y_0)$, $p_1 = (x_1, y_1)$, $p'_0 = (x'_0, y'_0)$, $p'_1 = (x'_1, y'_1)$ i l będzie prostą przechodzącą przez punkt p nachyloną do poziomej osi pod kątem $\alpha \in \mathbb{R}$. Stąd

$$\begin{aligned}(x'_1 - x'_0)^2 &= ((x_1 - x) \cos(2\alpha) + (y_1 - y) \sin(2\alpha) - (x_0 - x) \cos(2\alpha) - (y_0 - y) \sin(2\alpha))^2 \\ &= ((x_1 - x_0) \cos(2\alpha) + (y_1 - y_0) \sin(2\alpha))^2 \\ &= (x_1 - x_0)^2 \cos^2(2\alpha) + (y_1 - y_0)^2 \sin^2(2\alpha) + 2(x_1 - x_0)(y_1 - y_0) \sin(2\alpha) \cos(2\alpha)\end{aligned}$$

oraz

$$\begin{aligned}(y'_1 - y'_0)^2 &= ((x_1 - x) \sin(2\alpha) - (y_1 - y) \cos(2\alpha) - (x_0 - x) \sin(2\alpha) + (y_0 - y) \cos(2\alpha))^2 \\ &= ((x_1 - x_0) \sin(2\alpha) - (y_1 - y_0) \cos(2\alpha))^2 \\ &= (x_1 - x_0)^2 \sin^2(2\alpha) + (y_1 - y_0)^2 \cos^2(2\alpha) - 2(x_1 - x_0)(y_1 - y_0) \sin(2\alpha) \cos(2\alpha).\end{aligned}$$

Zatem

$$|s'| = \sqrt{(x'_1 - x'_0)^2 + (y'_1 - y'_0)^2} = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} = |s|.$$

□

Twierdzenie 1.4.6. *Rozważamy różne punkty $p_0 = (x_0, y_0)$, $p_1 = (x_1, y_1)$ i $p_2 = (x_2, y_2)$. Niech s_0 będzie odcinkiem o końcach w punktach p_0 i p_1 oraz niech s_1 będzie odcinkiem o końcach w punktach p_1 i p_2 . Ponadto niech s'_0 i s'_1 będą odcinkami o końcach, odpowiednio, w punktach p'_0 i p'_1 oraz p'_1 i p'_2 , gdzie p'_0 , p'_1 i p'_2 powstały przez odbicie lustrzane punktów, odpowiednio, p_0 , p_1 i p_2 , względem prostej l przechodzącej przez punkt p nachylonej do poziomej osi pod kątem $\alpha \in \mathbb{R}$. Wówczas odcinki s'_0 i s'_1 tworzą ten sam kąt co odcinki s_0 i s_1 .*

Dowód. Wiemy z twierdzenia 1.4.3, że $|s_0| = |s'_0|$ oraz $|s_1| = |s'_1|$. Niech $p'_0 = (x'_0, y'_0)$, $p'_1 = (x'_1, y'_1)$ i $p'_2 = (x'_2, y'_2)$. Niech s_0 i s_1 tworzą kąt φ . Wówczas

$$\begin{aligned} \text{Rot}(\varphi) \cdot (p'_0 - p'_1) + p'_1 &= \text{Rot}(\varphi) \cdot \text{Ref}(\alpha) \cdot (p_0 - p_1) + \text{Ref}(\alpha) \cdot (p_1 - p) + p \\ &= \text{Ref}(\alpha) \cdot (\text{Rot}(\varphi) \cdot (p_0 - p_1) + p_1 - p) + p \\ &= \text{Ref}(\alpha) \cdot (p_2 - p) + p \\ &= p'_2. \end{aligned}$$

□

Podsumowując, przedstawione własności oraz wyniki geometryczne są kluczowe dla zdefiniowania struktur o kształcie opartym na siatce heksagonów, z uwzględnieniem ich kształtu oraz dowolności położenia w układzie współrzędnych. Ponadto będziemy badali wpływ rotacji i odbić lustrzanych na ich porównywanie oraz agregację.

1.5 Porządki częściowe

Porządek częściowy \leq jest rodzajem relacji dwuargumentowej, która jest zwrotna, przechodnia oraz antysymetryczna (patrz [20]). Pozwala na porównywanie oraz ustalanie hierarchii elementów w rozważanym zbiorze, a tym samym na jego agregację przez wyznaczanie takich elementów jak element minimalny lub maksymalny, element najmniejszy lub największy oraz infimum lub supremum. W przypadku, gdy istnieje taka relacja dla każdej pary elementów rozważanego zbioru, taki porządek nazywamy **porządkiem liniowym**.

Niech \leq będzie ustalonym porządkiem na zbiorze obiektów \mathcal{A} i niech $\mathcal{A}_0 \subset \mathcal{A}$. Element $A \in \mathcal{A}_0$ nazywamy **elementem minimalnym (maksymalnym)** w \mathcal{A}_0 , gdy w \mathcal{A}_0 nie istnieje element mniejszy (większy) od A . Element $A \in \mathcal{A}_0$ nazywamy **elementem najmniejszym (największym)** w \mathcal{A}_0 , gdy jest jedynym elementem minimalnym (maksymalnym) w \mathcal{A}_0 . Element $A \in \mathcal{A}$ nazywamy **ograniczeniem dolnym (górnym)** zbioru \mathcal{A}_0 , gdy dla każdego elementu $A' \in \mathcal{A}_0$ spełniona jest zależność $A \leq A'$ (odpowiednio $A' \leq A$). Ponadto element $A \in \mathcal{A}$ nazywamy **infimum (supremum)** zbioru \mathcal{A}_0 , gdy jest największym ograniczeniem dolnym (najmniejszym ograniczeniem górnym) zbioru \mathcal{A}_0 .

Poniższe wyniki prezentujemy ze względu na ich istotny wpływ na proces agregacji z wykorzystaniem porządków. Wskazują, że w zależności od rozważanego porządku

oraz badanego zbioru, taka agregacja może być ograniczona lub nawet niemożliwa (por. twierdzenie 1.5.2). Z drugiej strony agregacja jest jednoznaczna, jeżeli porządek jest odpowiednio określony (por. twierdzenie 1.5.1).

Twierdzenie 1.5.1 ([20], strona 22). *Jeżeli infimum oraz supremum lub jedno z nich istnieją, to są wyznaczone jednoznacznie.*

Twierdzenie 1.5.2 ([20], strona 22). *Infimum (supremum) nie istnieje, gdy rozważany zbiór nie ma ograniczeń dolnych (ograniczeń górnych) lub gdy nie istnieje największe ograniczenie dolne (najmniejsze ograniczenie górne).*

Przykład 1.5.3. *Niech porządek \leq będzie słabą nierównością na zbiorze liczb całkowitych \mathbb{Z} . Jeżeli rozważymy $\mathcal{A} = \mathbb{Z}$, to w takim zbiorze nie istnieje ograniczenie dolne, a tym samym nie istnieje infimum.*

Twierdzenie 1.5.4 ([20], uwaga 6.11). *Jeżeli rozważany porządek jest liniowy oraz w rozważanym zbiorze istnieje element najmniejszy, to element najmniejszy, element minimalny i infimum są tym samym elementem dla tego zbioru. Podobnie, jeżeli w rozważanym zbiorze istnieje element największy, to element największy, element maksymalny i supremum również są tym samym elementem dla tego zbioru.*

Przykład 1.5.5. *Rozważamy zbiór $\mathcal{A} = (0, 1]$ oraz porządek liniowy \leq będący słabą nierównością na zbiorze liczb rzeczywistych. Wówczas nie istnieją elementy minimalny i najmniejszy w tym zbiorze, gdyż dla każdego $A \in \mathcal{A}$ istnieje element mniejszy, np. element $A' = \frac{1}{2}A$. Ponadto każdy element $A'' \leq 0$ jest ograniczeniem dolnym zbioru \mathcal{A} . Stąd, $A''' = 0$ jest największym ograniczeniem dolnym, czyli jest infimum dla zbioru \mathcal{A} . Z drugiej strony 1 jest elementem maksymalnym, elementem największym oraz supremum w zbiorze \mathcal{A} .*

Podsumowując, w omawianym podrozdziale przedstawiliśmy niezbędne aspekty związane z częściowymi oraz liniowymi porządkami na zbiorach, które mają istotne znaczenie w kluczowym aspekcie niniejszej pracy, którym jest porównywanie obiektów oraz ich agregacja. Przedstawione przykłady pozwalają na lepsze zrozumienie przyjętej koncepcji oraz planowanego jej wykorzystania w dalszych rozważaniach.

Rozdział 2

(Rozszerzone) ciągi binarne

2.1 Ciągi binarne

Liczby binarne są liczbami wyrażonymi w pozycyjnym systemie liczbowym o podstawie 2, czyli w systemie dwójkowym (patrz [12]). Są szeroko wykorzystywane w informatyce, elektronice (patrz [34]) oraz w teorii liczb (np. liczby adyczne p). W kolejnych rozdziałach wykorzystamy je do reprezentacji struktur o kształcie opartym na siatce heksagonów.

Niech $n \in \mathbb{N}$. Ciąg $B: \{0, 1, \dots, n-1\} \rightarrow \{0, 1\}$ nazywamy **ciągiem binarnym (o długości n)** i oznaczamy go:

$$B = b_0 b_1 \dots b_{n-1},$$

gdzie $b_k = B(k)$ dla $k \in \{0, 1, \dots, n-1\}$. W przeciwieństwie do liczb binarnych, dopuszczamy możliwość pojawienia się cyfry 0 na początkowych miejscach tego ciągu. Ponadto przyjmujemy, że jeżeli $n = 0$, to ciąg binarny B jest pusty i oznaczamy go:

$$B = \emptyset.$$

W literaturze pojawiają się również inne oznaczenia pustych ciągów, np. grecką literą ϵ (por. [22], strona 29). Rodzinę wszystkich ciągów binarnych oznaczamy \mathcal{B} . Niech $B, B' \in \mathcal{B}$, gdzie $B = b_0 b_1 \dots b_{n-1}$ oraz $B' = b'_0 b'_1 \dots b'_{n'-1}$ dla pewnych $n, n' \in \mathbb{N}$. Definiujemy **konkatenację** ciągów binarnych BB' jako ciąg binarny $B'' = b''_0 b''_1 \dots b''_{n''-1}$, gdzie $n'' = n + n'$, $b''_i = b_i$ dla $i \in \{0, 1, \dots, n-1\}$ oraz $b''_{n+j} = b'_j$ dla $j \in \{0, 1, \dots, n'-1\}$.

Zamiana bitów oraz odwrócenie ciągu są podstawowymi operacjami na ciągach binarnych. Niech $B = b_0 b_1 \dots b_{n-1}$ i $B' = b'_0 b'_1 \dots b'_{n-1}$ będą ciągami binarnymi o długości $n \in \mathbb{N}$. Odwzorowanie τ dane wzorem $\tau(B) = B'$, gdzie $b'_k = 1 - b_k$ dla każdego $k \in \{0, 1, \dots, n-1\}$, nazywamy **zamianą bitów** ciągu B . Odwzorowanie ς dane wzorem $\varsigma(B) = B'$, gdzie $b'_k = b_{n-1-k}$ dla każdego $k \in \{0, 1, \dots, n-1\}$, nazywamy **odwróceniem** ciągu binarnego B . Ponadto złożenie $\pi = \tau \circ \varsigma$ nazywamy **permutacją** ciągu binarnego B .

Przykład 2.1.1. Jeżeli $B = 00000101$, to $\tau(B) = 11111010$, $\varsigma(B) = 10100000$ oraz $\pi(B) = 01011111$.

Bezpośrednio z definicji funkcji τ , ς oraz π oraz konkatenacji wynika szereg własności, które przedstawiamy bez dowodu ze względu na ich prostotę.

Twierdzenie 2.1.2. *Prawdziwe są następujące zależności:*

- (i) $\tau \circ \tau \equiv \text{id}$, czyli τ jest involucją.
- (ii) $\varsigma \circ \varsigma \equiv \text{id}$, czyli ς jest involucją.
- (iii) $\pi \equiv \varsigma \circ \tau$, czyli złożenie funkcji τ oraz ς jest przemienne i daje π .
- (iv) $\pi \circ \varsigma \equiv \tau \equiv \varsigma \circ \pi$, czyli złożenie funkcji ς oraz π jest przemienne i daje τ .
- (v) $\pi \circ \tau \equiv \varsigma \equiv \tau \circ \pi$, czyli złożenie funkcji τ oraz π jest przemienne i daje ς .

Twierdzenie 2.1.3. Jeżeli $B, B' \in \mathcal{B}$, to:

- (i) $\tau(BB') = \tau(B)\tau(B')$,
- (ii) $\varsigma(BB') = \varsigma(B')\varsigma(B)$,
- (iii) $\pi(BB') = \pi(B')\pi(B)$,

Uwaga 2.1.4. W podrozdziałach B.1, B.2 oraz B.3 przedstawiamy implementacje w języku Python funkcji, odpowiednio, τ , ς oraz π .

2.2 Rozszerzenie pojęcia ciągu binarnego

Rozszerzamy pojęcie ciągu binarnego o możliwość wtrącenia w ciąg binarny kolejnego ciągu binarnego. Pozwoli nam to na utworzenie drzewa ternarnego, w którym liście są ciągami binarnymi, co wykorzystamy bezpośrednio w reprezentacji złożonych grafów o kształcie opartym na siatce heksagonów. Takie podejście wymaga wykorzystania w definicjach oraz w dowodach indukcji strukturalnej (patrz [22]). W informatyce istnieje wiele struktur zdefiniowanych w taki sposób, np. palindromy (patrz [22], sekcja 5.1.1) lub drzewa binarne (patrz [6], sekcja B.5.3).

Definicja 2.2.1. Rozszerzone ciągi binarne definiujemy następująco:

Przypadek bazowy: Jeżeli B_1, B_2 oraz B_3 są ciągami binarnymi, to trójka (B_1, B_2, B_3) jest rozszerzonym ciągiem binarnym.

Krok indukcyjny: Trójka (B_1, E_2, E_3) jest rozszerzonym ciągiem binarnym, gdy B_1 jest ciągiem binarnym oraz spełniony jest jeden z następujących warunków:

- E_2 jest ciągiem binarnym oraz E_3 jest rozszerzonym ciągiem binarnym,
- E_2 jest rozszerzonym ciągiem binarnym oraz E_3 jest ciągiem binarnym,
- E_2 oraz E_3 są rozszerzonymi ciągami binarnymi.

Przykład 2.2.2. Jeżeli $B_1 = 001$, $B_2 = 010$ oraz $B_3 = 000$, to $(B_1, B_2, B_3) = (001, 010, 000)$ jest rozszerzonym ciągiem binarnym.

Przykład 2.2.3. Jeżeli $B_1 = 110$, $E_2 = (01, 11, 01)$ oraz $E_3 = (011, 100, 101)$, to $(B_1, E_2, E_3) = (110, (01, 11, 01), (011, 100, 101))$ jest rozszerzonym ciągiem binarnym.

Rozszerzony ciąg binarny $E = (B_1, E_2, E_3)$ przedstawiamy też następująco:

$$E = B_1(E_2)E_3.$$

Przykład 2.2.4. Jeżeli $B_1 = 001$, $B_2 = 010$ oraz $B_3 = 000$, to $B_1(B_2)B_3 = 001(010)000$ jest rozszerzonym ciągiem binarnym.

Przykład 2.2.5. Jeżeli $B_1 = 110$, $E_2 = 01(11)01$ oraz $E_3 = 011(100)101$, to $B_1(E_2)E_3 = 110(01(11)01)011(100)101$ jest rozszerzonym ciągiem binarnym.

Ciągi B_1 , E_2 oraz E_3 nazywamy **podciągami** rozszerzonego ciągu binarnego E . Rodzinę wszystkich ciągów binarnych oraz rozszerzonych ciągów binarnych oznaczamy \mathcal{E} .

Niech $B \in \mathcal{B}$ oraz niech $\mathcal{E} \setminus \mathcal{B} \ni E = (B_1, E_2, E_3)$. Złożonością ciągów binarnych oraz rozszerzonych ciągów binarnych nazywamy odwzorowanie $\omega: \mathcal{E} \rightarrow \mathbb{N}$, które definiujemy rekurencyjnie:

- $\omega(B) = 1$,
- $\omega(E) = \omega(E_2) + \omega(E_3)$.

Rozszerzony ciąg binarny E jest **skończony**, gdy $\omega(E) < \infty$. W niniejszej dysertacji rozważamy wyłącznie skończone rozszerzone ciągi binarne.

Przykład 2.2.6. Rozważamy rozszerzony ciąg binarny $E = 110(01(11)01)011(100)101$. Wówczas

$$\begin{aligned} \omega(E) &= \omega(110(01(11)01)011(100)101) \\ &= \omega(01(11)01) + \omega(011(100)101) \\ &= \omega(11) + \omega(01) + \omega(100) + \omega(101) \\ &= 4. \end{aligned}$$

Bezpośrednio z definicji złożoności ciągów binarnych oraz rozszerzonych ciągów binarnych otrzymujemy następujące własności:

Wniosek 2.2.7. Jeżeli $\mathcal{E} \setminus \mathcal{B} \ni E = (B_1, E_2, E_3)$, to $\omega(E) > 1$, $\omega(E) > \omega(E_2)$ oraz $\omega(E) > \omega(E_3)$.

Mierzenie złożoności rozszerzonych ciągów binarnych będzie miało dla nas kluczowe znaczenie w badaniu własności tych ciągów określonych rekurencyjnie.

Rozwijamy pojęcie konkatenacji ciągów binarnych o rozszerzone ciągi binarne. Jeżeli $B, B'_1, B'_2, B'_3 \in \mathcal{B}$, to konkatenacją ciągu binarnego B oraz rozszerzonego ciągu binarnego $E' = B'_1(B'_2)B'_3$ nazywamy rozszerzony ciąg binarny (BB'_1, B'_2, B'_3) i oznaczamy go $BE' = BB'_1(B'_2)B'_3$. Z drugiej strony konkatenacją rozszerzonego ciągu binarnego E' oraz ciągu binarnego B nazywamy rozszerzony ciąg binarny (B'_1, B'_2, B'_3B) i oznaczamy go $E'B = B'_1(B'_2)B'_3B$. Rozwijamy tę definicję o dowolne rozszerzone ciągi binarne z wykorzystaniem indukcji strukturalnej. Jeżeli $B' \in \mathcal{B}$ oraz $E = (B_1, E_2, E_3) \in \mathcal{E}$, to konkatenacją

rozszerzonego ciągu binarnego E oraz ciągu binarnego B' nazywamy rozszerzony ciąg binarny $(B_1, E_2, E_3 B')$ i oznaczamy go $EB' = B_1(E_2)E_3 B'$. Jeżeli $E = (B_1, E_2, E_3) \in \mathcal{E}$ oraz $E' \in \mathcal{E}$, to konkatenacją rozszerzonych ciągów binarnych E oraz E' nazywamy rozszerzony ciąg binarny $(B_1, E_2, E_3 E')$ i oznaczamy go $EE' = B_1(E_2)E_3 E'$.

Przykład 2.2.8. Jeżeli $B_1 = 110$, $E_2 = 01(11)01$ oraz $E_3 = 011(100)101$, to $B_1 E_2 = 11001(11)01$ oraz $E_2 E_3 = 01(11)01011(100)101$.

Określanie złożoności połączonych ciągów binarnych oraz rozszerzonych ciągów binarnych prowadzi do następujących wyników, które przedstawiamy bez dowodu ze względu na ich prostotę:

Wniosek 2.2.9. Jeżeli $B \in \mathcal{B}$ oraz $E \in \mathcal{E}$, to $\omega(BE) = \omega(E) = \omega(EB)$.

Wniosek 2.2.10. Jeżeli $E, E' \in \mathcal{E}$, to $\omega(EE') = \omega(E) + \omega(E') - 1$.

Przykład 2.2.11. Jeżeli $B_1 = 110$, $E_2 = 01(11)01$ oraz $E_3 = 011(100)101$, to $\omega(B_1 E_2) = 2 = \omega(E_2)$, $\omega(E_3) = 2$ oraz

$$\begin{aligned} \omega(E_2 E_3) &= \omega(01(11)01011(100)101) \\ &= \omega(11) + \omega(01011(100)101) \\ &= 1 + \omega(100) + \omega(101) \\ &= 3 \\ &= \omega(E_2) + \omega(E_3) - 1. \end{aligned}$$

Pozostało określić równość dwóch rozszerzonych ciągów binarnych. Definicję przedstawimy z zachowaniem ścisłości formalnej, stosując indukcję strukturalną. Równość dwóch rozszerzonych ciągów binarnych będzie wynikała z równości ich składników. W uwadze 2.2.12 prezentujemy najprostsz taki przypadek.

Uwaga 2.2.12. Niech $B_1, B_2, B_3, B'_1, B'_2, B'_3 \in \mathcal{B}$. Wówczas następujące warunki są równoważne:

- $B_1 0 B_2 = B'_1 0 B'_2$ oraz $B_1 1 B_3 = B'_1 1 B'_3$,
- $B_1 = B'_1$, $B_2 = B'_2$ oraz $B_3 = B'_3$.

Dowód. Niech $B_1, B_2, B_3, B'_1, B'_2, B'_3$ będą ciągami binarnymi o długościach, odpowiednio,

$$n_1, n_2, n_3, n'_1, n'_2, n'_3 \in \mathbb{N}.$$

Jeżeli $n_1 < n'_1$, to $b'_{1,i} = 1$ oraz $b'_{1,i} = 2$ dla pewnego $i \in [n_1, n'_1 - 2]$, sprzeczność. Zatem $n_1 = n'_1$, $n_2 = n'_2$ oraz $n_3 = n'_3$. Stąd $B_1 = B'_1$, $B_2 = B'_2$ oraz $B_3 = B'_3$. Dowód w drugą stronę jest oczywisty. \square

Definicja 2.2.13. Niech $B_1, B'_1 \in \mathcal{B}$ i $E_2, E_3, E'_2, E'_3 \in \mathcal{E}$ oraz niech $E = (B_1, E_2, E_3)$ i $E' = (B'_1, E'_2, E'_3)$ będą rozszerzonymi ciągami binarnymi. Definiujemy rekurencyjnie zależność $E = E'$.

Przypadek bazowy: Jeżeli E_2, E_3, E'_2 oraz E'_3 są ciągami binarnymi, to $E = E'$ wtedy i tylko wtedy, gdy $B_1 = B'_1, E_2 = E'_2$ oraz $E_3 = E'_3$.

Krok indukcyjny: Rozważamy następujące przypadki:

- E_2 i E'_2 są ciągami binarnymi oraz E_3 i E'_3 są rozszerzonymi ciągami binarnymi,
- E_2 i E'_2 są rozszerzonymi ciągami binarnymi oraz E_3 i E'_3 są ciągami binarnymi,
- E_2, E'_2, E_3 oraz E'_3 są rozszerzonymi ciągami binarnymi.

W każdym przypadku korzystamy z tego samego podejścia, tj. $E = E'$ wtedy i tylko wtedy, gdy $B_1 = B'_1, E_2 = E'_2$ oraz $E_3 = E'_3$.

Bezpośrednio z definicji złożoności zwykłych oraz rozszerzonych ciągów binarny oraz z definicji równości rozszerzonych ciągów binarnych wynika następujący fakt:

Wniosek 2.2.14. Niech $E, E' \in \mathcal{E}$. Jeżeli $\omega(E) \neq \omega(E')$, to $E \neq E'$.

Ponadto E interpretujemy jako parę B_10E_2 oraz B_11E_3 , którą nazywamy **składowymi** rozszerzonego ciągu binarnego E . Taka interpretacja będzie miała uzasadnienie w reprezentacji rozważanych grafów, gdzie każda składowa będzie reprezentowała jeden z dwóch podgrafów badanej struktury, a 0 i 1 odpowiadają kątom, odpowiednio, lewostronnym oraz prawostronnym.

Uwaga 2.2.15. W podrozdziale B.4 przedstawiamy wdrożenie w języku Python rozszerzonych ciągów binarnych. W implementacjach B.4.2 oraz B.4.4 wdrażamy zwracanie, odpowiednio, podciągów oraz składowych z rozszerzonych ciągów binarnych. Ponadto w implementacji B.4.5 przedstawiamy wdrożenie w języku Python funkcji ω dla rozszerzonych ciągów binarnych.

2.3 Rozszerzony ciąg binarny jako drzewo ternarne

Zauważmy, że rozszerzony ciąg binarny ma strukturę drzewa ternarnego, gdzie węzły są ciągami binarnymi lub rozszerzonymi ciągami binarnymi, przy czym każdy węzeł będący ciągiem binarnym nie ma dzieci oraz każdy węzeł będący rozszerzonym ciągiem binarnym ma dokładnie troje dzieci (patrz „Ternary search tries” w [42], strona 746). Wówczas możemy iterować po kolejnych ciągach binarnych oraz rozszerzonych ciągach binarnych dla danego rozszerzonego ciągu binarnego. Niech C_0 będzie rozszerzonym ciągiem binarnym i niech odwzorowanie $\kappa: \mathbb{N}_0 \rightarrow \mathbb{N}$ będzie dane wzorem:

$$\kappa(i) = 3 \cdot \left\lfloor \frac{2i}{3} \right\rfloor + 1. \quad (2.3.1)$$

Rozważamy taką sekwencję $C = (C_i)_{i \in I}$ ciągów binarnych i rozszerzonych ciągów binarnych oraz taki zbiór indeksów $I \subset \mathbb{N}_0$, że:

- jeżeli $i \in I$ spełnia $i \bmod 3 = 1$, to C_i jest ciągiem binarnym,

- jeżeli C_i jest rozszerzonym ciągiem binarnym dla $i \in I$, tj. postaci:

$$C_i = C_{\kappa(i)}(C_{\kappa(i)+1})C_{\kappa(i)+2},$$

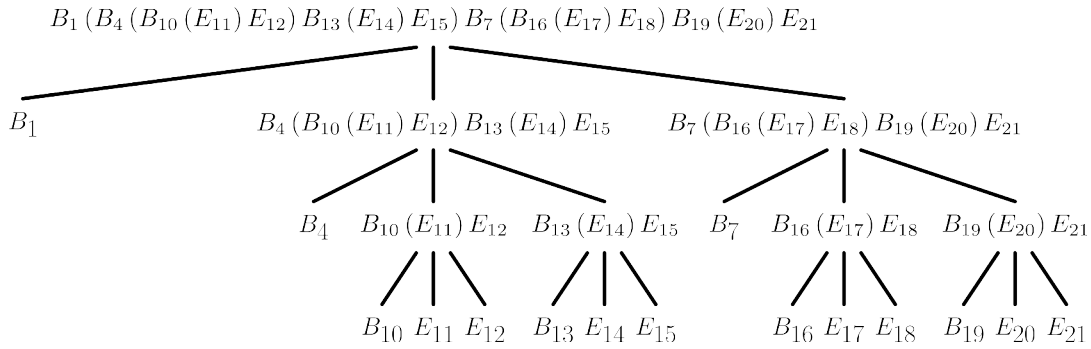
oraz

$$\{\kappa(i), \kappa(i) + 1, \kappa(i) + 2\} \subset I.$$

Sekwencję C nazywamy **rozwinieciem** rozszerzonego ciągu binarnego C_0 .

Przykład 2.3.1. Na rysunku 2.1 przedstawiliśmy rozwinięcie rozszerzonego ciągu binarnego E_0 w postaci drzewa ternarnego danego wzorem:

$$E_0 = B_1(B_4(B_{10}(E_{11})E_{12})B_{13}(E_{14})E_{15})B_7(B_{16}(E_{17})E_{18})B_{19}(E_{20})E_{21}.$$



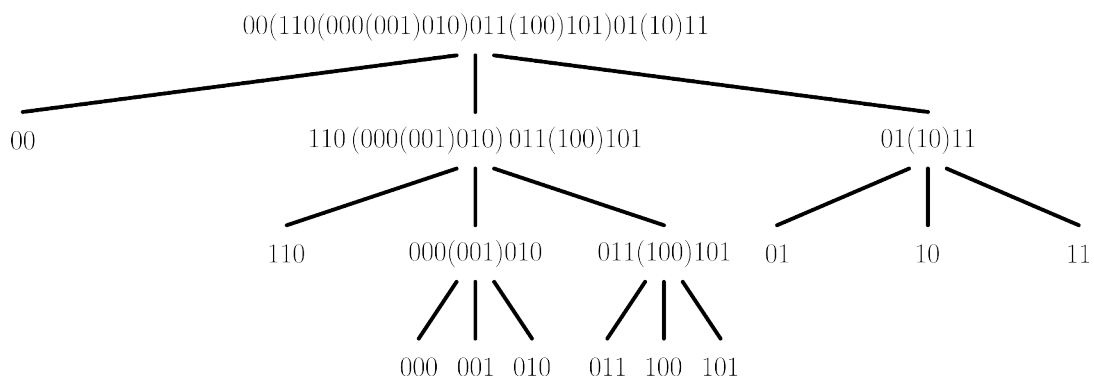
Rysunek 2.1: Drzewo ternarne o korzeniu będącym rozszerzonym ciągiem binarnym $E_0 = B_1(B_4(B_{10}(E_{11})E_{12})B_{13}(E_{14})E_{15})B_7(B_{16}(E_{17})E_{18})B_{19}(E_{20})E_{21}$, którego dziećmi są B_1 , $E_2 = B_4(B_{10}(E_{11})E_{12})B_{13}(E_{14})E_{15}$ i $E_3 = B_7(B_{16}(E_{17})E_{18})B_{19}(E_{20})E_{21}$. Ciągi B_4 , $E_5 = B_{10}(E_{11})E_{12}$ i $E_6 = B_{13}(E_{14})E_{15}$ są dziećmi E_2 oraz B_7 , $E_8 = B_{16}(E_{17})E_{18}$ i $E_9 = B_{19}(E_{20})E_{21}$ są dziećmi E_3 . Ciągi B_{10} , E_{11} i E_{12} są dziećmi E_5 , ciągi B_{13} , E_{14} i E_{15} są dziećmi E_6 , ciągi B_{16} , E_{17} i E_{18} są dziećmi E_8 oraz ciągi B_{19} , E_{20} , i E_{21} są dziećmi E_9 .

Przykład 2.3.2. Poniższa sekwencja ciągów jest rozwinięciem rozszerzonego ciągu binarnego $00(110(000(001)010)011(100)101)01(10)11$ (por. rys. 2.2):

- $C_0 = 00(110(000(001)010)011(100)101)01(10)11$,
- $C_1 = 00$,
- $C_2 = 110(000(001)010)011(100)101$,
- $C_3 = 01(10)11$,
- $C_4 = 110$,
- $C_5 = 000(001)010$,
- $C_6 = 011(100)101$,
- $C_7 = 01$,

- $C_8 = 10$,
- $C_9 = 11$,
- $C_{10} = 000$,
- $C_{11} = 001$,
- $C_{12} = 010$,
- $C_{13} = 011$,
- $C_{14} = 100$,
- $C_{15} = 101$.

Ciąg C_0 jest rozszerzonym ciągiem binarnym oraz $\kappa(0) = 1$, zatem istnieją ciągi o indeksach 1, 2 i 3. Ciągi C_2 i C_3 są rozszerzonymi ciągami binarnymi oraz $\kappa(2) = 4$ i $\kappa(3) = 7$, więc istnieją ciągi o indeksach 4, 5 i 6 oraz 7, 8 i 9. Ciągi C_5 i C_6 również są rozszerzonymi ciągami binarnymi oraz $\kappa(5) = 10$ i $\kappa(6) = 13$, więc istnieją ciągi o indeksach 10, 11 i 12 oraz 13, 14 i 15.



Rysunek 2.2: Drzewo ternarne przedstawiające rozwinięcie rozszerzonego ciągu binarnego $00(110(000(001)010)011(100)101)01(10)11$.

Poprawność definicji oraz intuicyjne zrozumienie rozwinięcia rozszerzonego ciągu binarnego przedstawimy przez badanie własności jego indeksów, co pozwoli jednocześnie na ścisłą analizę oraz przystępne wyjaśnienie charakterystyki tego ciągu. Głównym celem jest wykazanie, dla jakich argumentów funkcja κ jest różnowartościowa. Dzięki temu uzyskamy gwarancję, że rozważany indeks nie powtórzy się w kolejnych wierzchołkach drzewa. Bezpośrednio z definicji rozwinięcia rozszerzonego ciągu otrzymujemy następujący wynik:

Twierdzenie 2.3.3. *Jeżeli E_i jest rozszerzonym ciągiem binarnym, to $i \bmod 3 \neq 1$.*

Przechodzimy do wskazania argumentów, dla których odwzorowanie κ jest różnowartościowe. Dowód podzielimy na części, aby zachować przejrzystość. Dzięki monotoniczności funkcji podłoga otrzymujemy następujący rezultat:

Twierdzenie 2.3.4. *Odwzorowanie κ dane wzorem (2.3.1) jest niemalejące.*

Lemat 2.3.5. Niech $n \in \mathbb{Z}$. Jeżeli $n \bmod 3 \neq 1$, to $\lfloor \frac{2n}{3} \rfloor = \lfloor \frac{2n+1}{3} \rfloor$.

Dowód. Jeżeli istnieje takie $k \in \mathbb{Z}$, że $n = 3k$, to

$$\left\lfloor \frac{2n+1}{3} \right\rfloor = \left\lfloor \frac{6k+1}{3} \right\rfloor = \left\lfloor 2k + \frac{1}{3} \right\rfloor = \lfloor 2k \rfloor = \left\lfloor \frac{6k}{3} \right\rfloor = \left\lfloor \frac{2n}{3} \right\rfloor.$$

Jeżeli istnieje takie $k \in \mathbb{Z}$, że $n = 3k + 2$, to

$$\left\lfloor \frac{2n+1}{3} \right\rfloor = \left\lfloor \frac{6k+5}{3} \right\rfloor = \left\lfloor 2k + \frac{5}{3} \right\rfloor = \left\lfloor 2k + \frac{4}{3} \right\rfloor = \left\lfloor \frac{6k+4}{3} \right\rfloor = \left\lfloor \frac{2n}{3} \right\rfloor.$$

□

W twierdzeniu 2.3.6 pokazujemy zależność indeksów między rodzicem oraz jego dziećmi w rozwinięciu rozszerzonego ciągu binarnego z wykorzystaniem funkcji κ .

Twierdzenie 2.3.6. Niech $i \in \mathbb{N}_0$. Wówczas:

(i) jeżeli $i \bmod 3 = 0$, to $\kappa(i+2) = \kappa(i) + 3$,

(ii) jeżeli $i \bmod 3 = 2$, to $\kappa(i+1) = \kappa(i) + 3$.

Dowód. Jeżeli istnieje takie $j \in \mathbb{Z}$, że $i = 3j$, to

$$\kappa(i) = 3 \cdot \left\lfloor \frac{2 \cdot 3j}{3} \right\rfloor + 1 = 6j + 1$$

oraz, na podstawie lematu 2.3.5,

$$\kappa(i+2) = 3 \cdot \left\lfloor \frac{2(3j+2)}{3} \right\rfloor + 1 = 3 \cdot \left\lfloor \frac{2 \cdot 3j + 4}{3} \right\rfloor + 1 = 3 \cdot \left\lfloor \frac{2 \cdot 3j}{3} + \frac{4}{3} \right\rfloor + 1 = 3 \cdot \left\lfloor \frac{2 \cdot 3j}{3} \right\rfloor + 3 + 1 = 6j + 4.$$

Jeżeli istnieje takie $j \in \mathbb{Z}$, że $i = 3j + 2$, to, ponownie na podstawie lematu 2.3.5,

$$\kappa(i) = 3 \cdot \left\lfloor \frac{2(3j+2)}{3} \right\rfloor + 1 = 6j + 4$$

oraz

$$\kappa(i+1) = 3 \cdot \left\lfloor \frac{2(3j+3)}{3} \right\rfloor + 1 = 3 \cdot (2j+2) + 1 = 6j + 7.$$

□

W twierdzeniu 2.3.7 pokazujemy, że κ jest różnowartościowa dla takich indeksów rozwinięcia rozszerzonego ciągu binarnego, w których może się pojawić rozszerzony ciąg binarny, co potwierdza poprawność konstrukcji rozwinięcia rozszerzonego ciągu binarnego, zapewniając niepowtarzające się indeksy podciągów.

Twierdzenie 2.3.7. Odwzorowanie κ dane wzorem (2.3.1) jest różnowartościowe dla takich $i \in \mathbb{N}$, że $i \bmod 3 \neq 1$.

Dowód. Niech $i, i' \in \mathbb{N}$ spełniają $i \bmod 3 \neq 1$ oraz $i' \bmod 3 \neq 1$. Wówczas $\kappa(i) = \kappa(i')$ wtedy i tylko wtedy, gdy istnieje takie $n \in \mathbb{N}$, że

$$n \leq \frac{2i}{3} < n+1 \quad \text{oraz} \quad n \leq \frac{2i'}{3} < n+1,$$

co jest równoważne

$$\frac{3n}{2} \leq i < \frac{3n+3}{2} \quad \text{oraz} \quad \frac{3n}{2} \leq i' < \frac{3n+3}{2}.$$

Jeżeli $n = 2k$ dla pewnego $k \in \mathbb{N}$, to

$$3k \leq i < 3k + 1.5 \quad \text{oraz} \quad 3k \leq i' < 3k + 1.5,$$

tj. $i, i' \in \{3k, 3k + 1\}$. Wartość argumentów i i i' nie przystaje do 1 modulo 3, więc $i = i'$.
Jeżeli $n = 2k + 1$ dla pewnego $k \in \mathbb{N}$, to

$$3k + 1.5 \leq i < 3k + 3 \quad \text{oraz} \quad 3k + 1.5 \leq i' < 3k + 3,$$

tj. $i, i' \in \{3k + 2\}$, czyli $i = i'$. □

Następnie przedstawimy kilka interesujących, naszym zdaniem, wyników pokazujących strukturę indeksów rozwinięcia rozszerzonego ciągu binarnego. W lemacie 2.3.8 prezentujemy obserwację dotyczącą obrazu zbioru kolejnych indeksów ze względu na funkcję κ , aby następnie w twierdzeniu 2.3.9 przedstawić wzór na indeksy podciągów w określonym poddrzewie analizowanego drzewa.

Lemat 2.3.8. *Niech $i, i' \in \mathbb{N}_0$. Jeżeli $i \leq i'$ spełniają $i \bmod 3 \neq 1$ i $i' \bmod 3 \neq 1$ oraz*

$$I = \{j \in \{i, i + 1, \dots, i'\} : j \bmod 3 \neq 1\},$$

to

$$\{\kappa(i) + s : i \in I, s \in \{0, 1, 2\}\} = \{M, M + 1, \dots, M'\}, \quad (2.3.2)$$

gdzie $M = \kappa(i)$ oraz $M' = \kappa(i') + 2$.

Dowód. Dowód przeprowadzimy z wykorzystaniem indukcji matematycznej ze względu na różnicę między i i i' .

Przypadek bazowy: Jeżeli $i' - i = 0$, to $I = \{i\}$ oraz

$$\{\kappa(i) + s : i \in I, s \in \{0, 1, 2\}\} = \{\kappa(i), \kappa(i) + 1, \kappa(i) + 2\}.$$

Założenie indukcyjne: Załóżmy, że $i' - i > 0$ oraz, że spełnione jest równanie (2.3.2).

Krok indukcyjny: Jeżeli $i' \bmod 3 = 0$, to kolejnym indeksem jest $i' + 2$ i wówczas $I' = I \cup \{i' + 2\}$. Jeżeli $i' \bmod 3 = 2$, to kolejnym indeksem jest $i' + 1$ i wówczas $I' = I \cup \{i' + 1\}$. Na mocy twierdzenia 2.3.6 oraz na podstawie założenia indukcyjnego, dla obu przypadków mamy:

$$\begin{aligned} \{\kappa(i) + s : i \in I', s \in \{0, 1, 2\}\} &= \\ &= \{\kappa(i) + s : i \in I, s \in \{0, 1, 2\}\} \cup \{\kappa(i' + j), \kappa(i' + j) + 1, \kappa(i' + j) + 2\} \\ &= \{\kappa(i), \kappa(i) + 1, \dots, \kappa(i') + 2\} \cup \{\kappa(i' + j), \kappa(i' + j) + 1, \kappa(i' + j) + 2\} \quad (2.3.3) \\ &= \{\kappa(i), \kappa(i) + 1, \dots, \kappa(i') + 2\} \cup \{\kappa(i') + 3, \kappa(i') + 4, \kappa(i') + 5\} \\ &= \{\kappa(i), \kappa(i) + 1, \dots, \kappa(i') + 5\}, \end{aligned}$$

gdzie $j \in \{1, 2\}$. □

Twierdzenie 2.3.9. Niech $i \in \mathbb{N}_0$ i niech C będzie rozwinięciem pewnego rozszerzonego ciągu binarnego. Jeżeli $E_i \in C$ jest rozszerzonym ciągiem binarnym, to indeksy ciągów binarnych i rozszerzonych ciągów binarnych w poddrzewie o korzeniu w E_i należą do zbiorów:

$$I(i, M) = \{2^M \cdot (\kappa(i) + 2) - 2, 2^M \cdot (\kappa(i) + 2) - 1, \dots, 2^M \cdot (\kappa(i) + 5) - 3\}, \quad (2.3.4)$$

gdzie M jest dowolną nieujemną liczbą całkowitą.

Dowód. Dowód przeprowadzimy wykorzystując indukcję matematyczną ze względu na M . Niech $i \in \mathbb{N}_0$, niech sekwencja $C = (C_i)_{i \in I}$ będzie rozwinięciem rozszerzonego ciągu binarnego E i niech E_i będzie rozszerzonym ciągiem binarnym dla $i \in I$. Jeżeli $M = 0$, to, z definicji odwzorowania κ , rozważamy współczynniki dzieci rozszerzonego ciągu binarnego E_i :

$$I(i, 0) = \{\kappa(i), \kappa(i) + 1, \kappa(i) + 2\} = \{2^0 \cdot (\kappa(i) + 2) - 2, 2^0 \cdot (\kappa(i) + 2) - 1, 2^0 \cdot (\kappa(i) + 2)\},$$

Przypuśćmy, że równanie (2.3.4) jest spełnione dla pewnego $M \in \mathbb{N}$. Jeżeli istnieje rozszerzony ciąg binarny o indeksie $j = 2^M \cdot (\kappa(i) + 2) - 1$, to

$$\kappa(j) = 3 \left\lfloor \frac{2 \cdot (2^M \cdot (\kappa(i) + 2) - 1)}{3} \right\rfloor + 1 = 2^{M+1} \cdot (\kappa(i) + 2) - 2.$$

Podobnie dla ostatniego indeksu $j' = 2^M \cdot 3 \left\lfloor \frac{2i}{3} \right\rfloor + 2 - 3$ przedziału z tezy powiększonego o 2 celem otrzymania indeksu trzeciego ciągu binarnego lub rozszerzonego ciągu binarnego dla rozszerzonego ciągu binarnego o tym indeksie:

$$\kappa(j') + 2 = 3 \left\lfloor \frac{2(2^M \cdot 3 \left\lfloor \frac{2i}{3} \right\rfloor + 2) - 3}{3} \right\rfloor + 1 + 2 = 2^{M+1} \cdot 3 \left(\left\lfloor \frac{2i}{3} \right\rfloor + 2 \right) - 3.$$

Na mocy wniosku 2.3.4, odwzorowanie κ jest niemalejące, więc $\kappa(j)$ oraz $\kappa(j') + 2$ są skrajnymi wartościami nowego przedziału. Na mocy lematu 2.3.8 obraz kolejnych indeksów z danego przedziału daje ciąg kolejnych liczb, gdzie najmniejszą jest obraz najmniejszego indeksu, a największą jest obraz największego indeksu powiększonego o 2. Stąd

$$I(i, M + 1) = \left\{ 2^{M+1} \cdot 3 \left(\left\lfloor \frac{2i}{3} \right\rfloor + 1 \right) - 2, 2^{M+1} \cdot 3 \left(\left\lfloor \frac{2i}{3} \right\rfloor + 1 \right) - 1, \dots, 2^{M+1} \cdot 3 \left(\left\lfloor \frac{2i}{3} \right\rfloor + 2 \right) - 3 \right\}.$$

□

Przykład 2.3.10. Niech E_0 będzie rozszerzonym ciągiem binarnym. Przyjmujemy jego iterację zgodnie ze wzorem (2.3.1). Wówczas (por. rys. 2.1)

$$I(0, 0) = \{1, 2, 3\},$$

$$I(0, 1) = \{4, 5, \dots, 9\},$$

$$I(0, 2) = \{10, 11, \dots, 21\},$$

$$I(0, 3) = \{22, 23, \dots, 45\},$$

$$I(0, 4) = \{46, 47, \dots, 93\}.$$

Ponadto

$$\begin{aligned} I(2, 0) &= \{4, 5, 6\}, \\ I(2, 1) &= \{10, 11, \dots, 15\}, \\ I(2, 2) &= \{22, 23, \dots, 33\}, \\ I(2, 3) &= \{46, 47, \dots, 69\}. \end{aligned}$$

oraz

$$\begin{aligned} I(3, 0) &= \{7, 8, 9\}, \\ I(3, 1) &= \{16, 17, \dots, 21\}, \\ I(3, 2) &= \{34, 35, \dots, 45\}, \\ I(3, 3) &= \{70, 71, \dots, 93\}. \end{aligned}$$

Bezpośrednio z twierdzenia 2.3.9 otrzymujemy następujący rekurencyjny wzór na zbiór indeksów kolejnych dzieci w badanym drzewie ternarym:

Wniosek 2.3.11. *Jeżeli $i \bmod 3 \neq 1$ dla $i \in \mathbb{N}_0$, to dla każdego $M \in \mathbb{N}_0$ mamy:*

$$I(i, M + 1) = I(\kappa(i) + 1, M) \cup I(\kappa(i) + 2, M).$$

Na zakończenie podrozdziału przedstawiamy wzór na złożoność rozważanego rozszerzonego ciągu binarnego z wykorzystaniem jego rozwinięcia. Intuicyjnie wystarczy zliczyć ciągi binarne o indeksach, które nie są przystające do 1 modulo 3. Dzięki temu możemy szybko obliczyć tę złożoność na podstawie wizualnej prezentacji rozszerzonego ciągu binarnego.

Wniosek 2.3.12. *Jeżeli $(C_i)_{i \in I}$ jest rozwinięciem $E_0 \in \mathcal{E}$, to*

$$\omega(E_0) = |\{C_i : i \in I, i \bmod 3 \neq 1, \omega(C_i) = 1\}|. \quad (2.3.5)$$

Dowód. Niech $E_0 \in \mathcal{E}$ i niech $(C_i)_{i \in I}$ będzie rozwinięciem E_0 . Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na złożoność E_0 .

Przypadek bazowy: Jeżeli $\omega(E_0) = 1$, to E_0 jest ciągiem binarnym. Stąd

$$C = \{E_0\},$$

co daje

$$|\{C_i : i \in I, i \bmod 3 \neq 1, \omega(C_i) = 1\}| = |\{C_0\}| = 1.$$

Założenie indukcyjne: Niech $\omega(E) = n$ dla pewnego takiego $n \in \mathbb{N}$, że $n > 1$. Zakładamy, że równanie (2.3.5) jest spełnione dla takich $F \in \mathcal{E}$, że $\omega(F) < \omega(E) = n$.

Krok indukcyjny: Jeżeli $E_0 = B_1(E_2)E_3$, to $\omega(E_0) = \omega(E_2) + \omega(E_3)$. Na podstawie wniosku 2.3.11, istnieją takie zbiory indeksów:

$$I_2 = \{I(\kappa(0) + 1, M) : M \in \mathbb{N}_0\} \cap I$$

oraz:

$$I_3 = \{I(\kappa(0) + 2, M) : M \in \mathbb{N}_0\} \cap I,$$

że $I = I_2 \cup I_3 \cup \{0, 1\}$. Na mocy założenia indukcyjnego mamy:

$$\omega(E_2) = |\{C_i : i \in I_2, i \bmod 3 \neq 1, \omega(C_i) = 1\}|$$

oraz:

$$\omega(E_3) = |\{C_i : i \in I_3, i \bmod 3 \neq 1, \omega(C_i) = 1\}|.$$

Ponadto $C_0 = E_0$ nie jest ciągiem binarnym, więc

$$\begin{aligned} \omega(E) &= \omega(E_2) + \omega(E_3) \\ &= |\{C_i : i \in I_2, i \bmod 3 \neq 1, \omega(C_i) = 1\}| + |\{C_i : i \in I_3, i \bmod 3 \neq 1, \omega(C_i) = 1\}| \\ &= |\{C_i : i \in I_2 \cup I_3, i \bmod 3 \neq 1, \omega(C_i) = 1\}| \\ &= |\{C_i : i \in I, i \bmod 3 \neq 1, \omega(C_i) = 1\}|. \end{aligned}$$

□

Przykład 2.3.13. Rozważamy rozszerzony ciąg binarny:

$$E = 00(110(000(001)010)011(100)101)01(10)11.$$

Wówczas

$$\begin{aligned} \omega(00(110(000(001)010)011(100)101)01(10)11) &= \\ &= \omega(110(000(001)010)011(100)101) + \omega(01(10)11) \\ &= \omega(000(001)010) + \omega(011(100)101) + \omega(10) + \omega(11) \\ &= \omega(001) + \omega(010) + \omega(100) + \omega(101) + 2 \\ &= 6. \end{aligned}$$

Elementy rozwinięcia rozszerzonego ciągu binarnego E o indeksach 8, 9, 11, 12, 14 i 15 są ciągami binarnymi (por. rys. 2.2). Równanie (2.3.5) jest spełnione, ponieważ są to wszystkie indeksy, które nie przystają do 1 modulo 3 i są ciągami binarnymi.

2.4 Operacje na rozszerzonych ciągach binarnych

W niniejszym podrozdziale rozszerzymy pojęcia zamiany bitów, odwrócenia ciągu oraz permutacji na rozszerzone ciągi binarne. Zaproponowane wzory zachowują własności przedstawione dla ciągów binarnych (tj. okresowość względem iteracji). Ponadto pozwalają na otrzymanie alternatywnych reprezentacji heksastruktu o kształcie opartym na siatce heksagonów, co przedstawiamy w dalszej części pracy.

Niech $E = B_1(E_2)E_3$ będzie rozszerzonym ciągiem binarnym. Rozszerzamy pojęcie zamiany bitów w ciągu binarnym τ na zamianę bitów w rozszerzonym ciągu binarnym, definiowaną wzorem rekurencyjnym:

$$\tau(E) = \tau(B_1)(\tau(E_3))\tau(E_2).$$

Ponadto rozszerzamy pojęcie odwrócenia ciągu binarnego ς na rozszerzone ciągi binarne, definiowaną wzorem rekurencyjnym:

$$\varsigma(E) = \varsigma(E_3)(\tau(E_2))\varsigma(B_1).$$

Na końcu rozszerzamy operację permutacji ciągów binarnych π na rozszerzone ciągi binarne wzorem:

$$\pi = \zeta \circ \tau.$$

Przykład 2.4.1. *Rozważamy rozszerzony ciąg binarny $E = 001(010)000$. Wówczas*

$$\tau(E) = \tau(001)(\tau(000))\tau(010) = 110(111)101,$$

$$\zeta(E) = \zeta(000)(\tau(010))\zeta(001) = 000(101)100$$

oraz

$$\pi(E) = \zeta(\tau(E)) = \zeta(110(111)101) = \zeta(101)(\tau(111))\zeta(110) = 101(000)011.$$

Przykład 2.4.2. *Rozważamy rozszerzony ciąg binarny $E = 110(01(11)01)011(100)101$. Wówczas*

$$\begin{aligned} \tau(E) &= \tau(110)(\tau(011(100)101))\tau(01(11)01) \\ &= 001(\tau(011)(\tau(101))\tau(100))\tau(01)(\tau(01))\tau(11) \\ &= 001(100(010)011)10(10)00, \end{aligned}$$

$$\begin{aligned} \zeta(E) &= \zeta(011(100)101)(\tau(01(11)01))\zeta(110) \\ &= \zeta(101)(\tau(100))\zeta(011)(\tau(01)(\tau(01))\tau(11))011 \\ &= 101(011)110(10(10)00)011, \end{aligned}$$

oraz

$$\begin{aligned} \pi(E) &= \zeta(\tau(E)) \\ &= \zeta(001(100(010)011)10(10)00) \\ &= \zeta(10(10)00)(\tau(100(010)011))\zeta(001) \\ &= \zeta(00)(\tau(10))\zeta(10)(\tau(100)(\tau(011))\tau(010))100 \\ &= 00(01)01(011(100)101)100. \end{aligned}$$

Bezpośrednio z definicji π otrzymujemy wzór rekurencyjny przedstawiony w twierdzeniu 2.4.3. Mimo jego prostoty wyznaczenie obrazu badanego rozszerzonego ciągu binarnego może stanowić wyzwanie, co pokazujemy w dalszej części tego podrozdziału (por. przykład 2.4.11).

Twierdzenie 2.4.3. *Jeżeli $E = B_1(E_2)E_3$ jest rozszerzonym ciągiem binarnym, gdzie $B_1 \in \mathcal{B}$ oraz $E_2, E_3 \in \mathcal{E}$, to*

$$\pi(E) = \pi(E_2)(E_3)\pi(B_1). \quad (2.4.6)$$

Dowód. Niech $E = B_1(E_2)E_3$, gdzie $B_1 \in \mathcal{B}$ oraz $E_2, E_3 \in \mathcal{E}$. Wówczas

$$\begin{aligned} \pi(E) &= \zeta(\tau(B_1(E_2)E_3)) \\ &= \zeta(\tau(B_1)(\tau(E_3))\tau(E_2)) \\ &= \zeta(\tau(E_2))\left(\tau^{(2)}(E_3)\right)\zeta(\tau(B_1)) \\ &= \pi(E_2)(E_3)\pi(B_1). \end{aligned}$$

□

Odwzorowania τ , ς oraz π spełniają szereg istotnych własności, które wykorzystamy w dalszej części pracy. Inwolucja τ wynika bezpośrednio z definicji.

Twierdzenie 2.4.4. *Odwzorowanie τ jest involucją, tzn. $\tau \circ \tau \equiv \text{id}$.*

Przykład 2.4.5. *Jeżeli $E = 100(01(11)01)011(100)101$, to*

$$\begin{aligned}\tau(E) &= \tau(100)(\tau(011(100)101))\tau(01(11)01) \\ &= 011(\tau(011)(\tau(101))\tau(100))\tau(01)(\tau(01))\tau(11) \\ &= 011(100(010)011)10(10)00\end{aligned}$$

oraz

$$\begin{aligned}\tau^{(2)}(E) &= \tau(011(100(010)011)10(10)00) \\ &= \tau(011)(\tau(10(10)00))\tau(100(010)011) \\ &= 100(\tau(10)(\tau(00))\tau(10))\tau(100)(\tau(011))\tau(010) \\ &= 100(01(11)01)011(100)101 \\ &= E.\end{aligned}$$

Inwolucję ς wykazujemy indukcją strukturalną względem dzieci trzeciego podciągu rozważanego ciągu binarnego.

Twierdzenie 2.4.6. *Odwzorowanie ς jest involucją, tzn. $\varsigma \circ \varsigma \equiv \text{id}$.*

Dowód. Niech $E = B_1(E_2)E_3$ będzie rozszerzonym ciągiem binarnym, gdzie $B_1 \in \mathcal{B}$ oraz $E_2, E_3 \in \mathcal{E}$. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na złożoność E_3 .

Przypadek bazowy: Jeżeli $\omega(E_3) = 1$, to E_3 jest ciągiem binarnym, a stąd

$$\begin{aligned}\varsigma^{(2)}(E) &= \varsigma(\varsigma(E_3)(\tau(E_2))\varsigma(B_1)) \\ &= \varsigma^{(2)}(B_1)\left(\tau^{(2)}(E_2)\right)\varsigma^{(2)}(E_3) \\ &= B_1(E_2)E_3 \\ &= E.\end{aligned}\tag{2.4.7}$$

Założenie indukcyjne: Niech E_3 będzie rozszerzonym ciągiem binarnym o złożoności $n \in \mathbb{N}$ i niech $(C_i)_{i \in I}$ będzie rozwinięciem E_3 . Zakładamy, że $\varsigma^{(2)}(F) = F$ dla wszystkich rozszerzonych ciągów binarnych o złożoności mniejszej niż n .

Krok indukcyjny: Z definicji rozwinięcia rozszerzonego ciągu binarnego $E_3 = C_1(C_2)C_3$. Na mocy wniosku 2.2.7, ciąg C_3 jest złożoności mniejszej niż ciąg E_3 . Zatem, na mocy

założenia indukcyjnego, mamy:

$$\begin{aligned}
\varsigma^{(2)}(E) &= \varsigma(\varsigma(E_3)(\tau(E_2))\varsigma(B_1)) \\
&= \varsigma(\varsigma(C_1(C_2)C_3)(\tau(E_2))\varsigma(B_1)) \\
&= \varsigma(\varsigma(C_3)(\tau(C_2))\varsigma(C_1)(\tau(E_2))\varsigma(B_1)) \\
&= \varsigma(\varsigma(C_1)(\tau(E_2))\varsigma(B_1))\left(\tau^{(2)}(C_2)\right)\varsigma^{(2)}(C_3) \\
&= \varsigma^{(2)}(B_1)\left(\tau^{(2)}(E_2)\right)\varsigma^{(2)}(C_1)(C_2)C_3 \\
&= B_1(E_2)C_1(C_2)C_3 \\
&= B_1(E_2)E_3 \\
&= E.
\end{aligned}$$

□

Przykład 2.4.7. Jeżeli $E = 000(001(010)100)011(110)101$, to

$$\begin{aligned}
\varsigma(E) &= \varsigma(011(110)101)(\tau(001(010)100))\varsigma(000) \\
&= \varsigma(101)(\tau(110))\varsigma(011)(\tau(001)(\tau(100))\tau(010))000 \\
&= 101(001)110(110(011)101)000
\end{aligned}$$

oraz

$$\begin{aligned}
\varsigma^{(2)}(E) &= \varsigma(101(001)110(110(011)101)000) \\
&= \varsigma(110(110(011)101)000)(\tau(001))\varsigma(101) \\
&= \varsigma(000)(\tau(110(011)101))\varsigma(110)(110)101 \\
&= 000(\tau(110)(\tau(101))\tau(011))011(110)101 \\
&= 000(001(010)100)011(110)101 \\
&= E.
\end{aligned}$$

Dowód okresowości π względem iteracji poprzedzamy lematem, w którym pokazujemy, że wielokrotne złożenie odwzorowania π prowadzi do otrzymania rozszerzonego ciągu binarnego z przestawionymi składowymi.

Lemat 2.4.8. Jeżeli E, E' i E'' są rozszerzonymi ciągami binarnymi, to

$$\pi^{(\omega(E))}(\pi(E)(E')E'') = \pi(E')(E'')E.$$

Dowód. Niech E, E' i E'' będą rozszerzonymi ciągami binarnymi. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na złożoność ciągu E .

Przypadek bazowy: Jeżeli $\omega(E) = 1$, to E jest ciągiem binarnym, to

$$\pi(\pi(E)(E')E'') = \pi(E')(E'')E.$$

Założenie indukcyjne: Niech $\omega(E) = n$ dla pewnego $n \in \mathbb{N}$, niech $(C_i)_{i \in I}$ będzie rozwinięciem rozszerzonego ciągu binarnego E i niech $F, F', F'' \in \mathcal{E}$. Zakładamy, że jeżeli $\omega(F) < \omega(E) = n$, to

$$\pi^{(\omega(F))}(\pi(F)(F')F'') = \pi(F')(F'')F.$$

Krok indukcyjny: Z definicji rozwinięcia rozszerzonego ciągu binarnego $E = C_1(C_2)C_3$. Ponadto, na mocy wniosku 2.2.7, $\omega(C_2), \omega(C_3) < \omega(E)$. Stąd $\pi(E) = \pi(C_2)(C_3)\pi(C_1)$. Na mocy założenia indukcyjnego mamy:

$$\begin{aligned}
\pi^{(\omega(E))}(\pi(E)(E')E'') &= \pi^{(\omega(C_2)+\omega(C_3))}(\pi(C_2)(C_3)\pi(C_1)(E')E'') \\
&= \pi^{(\omega(C_3))}(\pi(C_3)(\pi(C_1)(E')E'')C_2) \\
&= \pi(\pi(C_1)(E')E'')(C_2)C_3 \\
&= \pi(E')(E'')C_1(C_2)C_3 \\
&= \pi(E')(E'')E.
\end{aligned}$$

□

Przykład 2.4.9. Lemat 2.4.8 zilustrujemy rozważając następujące ciągi: $E = 011(110)101$, $E' = 111$ i $E'' = 001(010)100$. Wówczas $\pi(E) = 100(101)001$, $\pi(E') = 000$ oraz $\omega(E) = 2$. Stąd

$$\begin{aligned}
\pi^{(\omega(E))}(\pi(E)(E')E'') &= \pi^{(2)}(100(101)001(111)001(010)100) \\
&= \pi(\pi(100(101)001(111)001(010)100)) \\
&= \pi(010(001(111)001(010)100)110) \\
&= \pi(001(111)001(010)100)(110)101 \\
&= 000(001(010)100)001(110)101 \\
&= \pi(E')(E'')E.
\end{aligned}$$

Twierdzenie 2.4.10. Jeżeli $E \in \mathcal{E}$, to $\pi^{(\omega(E)+1)}(E) = E$, tj. π jest okresowa względem iteracji.

Dowód. Niech $E \in \mathcal{E}$ oraz niech $(C_i)_{i \in I}$ będzie rozwinięciem rozszerzonego ciągu binarnego E . Na mocy lematu 2.4.8 mamy:

$$\begin{aligned}
\pi^{(\omega(E)+1)}(E) &= \pi^{(\omega(E)+1)}(C_1(C_2)C_3) \\
&= \pi^{(\omega(E))}(\pi(C_2)(C_3)\pi(C_1)) \\
&= \pi^{(\omega(C_2)+\omega(C_3))}(\pi(C_2)(C_3)\pi(C_1)) \\
&= \pi^{(\omega(C_3))}(\pi(C_3)(\pi(C_1))C_2) \\
&= C_1(C_2)C_3 \\
&= E.
\end{aligned}$$

□

Przykład 2.4.11. Twierdzenie 2.4.10 zilustrujemy dla rozszerzonego ciągu binarnego $E =$

000(001(010)100)011(110)101. Wówczas $\omega(E) = 4$, Wykorzystując wzór 2.4.6 mamy:

$$\begin{aligned}
 \pi^{(1)}(E) &= \pi(001(010)100)(011(110)101)\pi(000) \\
 &= \pi(010)(100)\pi(001)(011(110)101)111 \\
 &= 101(100)011(011(110)101)111, \\
 \pi^{(2)}(E) &= \pi(100)(011(011(110)101)111)\pi(101) \\
 &= 110(011(011(110)101)111)010, \\
 \pi^{(3)}(E) &= \pi(011(011(110)101)111)(010)\pi(110) \\
 &= \pi(011(110)101)(111)\pi(011)(010)100 \\
 &= \pi(110)(101)\pi(011)(111)001(010)100 \\
 &= 100(101)001(111)001(010)100, \\
 \pi^{(4)}(E) &= \pi(101)(001(111)001(010)100)\pi(100) \\
 &= 010(001(111)001(010)100)110, \\
 \pi^{(5)}(E) &= \pi(001(111)001(010)100)(110)\pi(010) \\
 &= \pi(111)(001(010)100)\pi(001)(110)101 \\
 &= 000(001(010)100)011(110)101 \\
 &= E.
 \end{aligned}$$

Uwaga 2.4.12. W implementacjach B.4.6, B.4.7 i B.4.8 przedstawiamy wdrożenie w języku Python funkcji, odpowiednio, τ , ζ i π , dla rozszerzonych ciągów binarnych.

Rozdział 3

Heksastrukty

3.1 Grafy

Centralnym tematem naszych badań jest szczególny rodzaj grafów. Teoria grafów stanowi niezwykle rozwiniętą dziedzinę nauki, przede wszystkim matematyki i informatyki, znajdującą zastosowanie w wielu obszarach, głównie dzięki intuicyjnej oraz estetycznej reprezentacji graficznej. Wykorzystanie grafów do modelowania złożonych problemów w matematyce, fizyce, informatyce oraz innych dziedzinach naukowych w sposób istotnie prostszy doprowadziło do intensyfikacji badań oraz znaczącego postępu w tych obszarach nauki. W rezultacie wiele zagadnień oraz terminów jest niejednoznacznie zdefiniowanych w podręcznikach i publikacjach naukowych (patrz [3], [46], [18] i [19]). Z tego powodu konieczny jest precyzyjny opis stosowanych przez nas terminów oraz zestawienie definicji, które będą używane w kontekście teorii grafów.

Grafem (nieskierowanym) nazywamy skończony i niepusty zbiór wierzchołków V wraz z przeciwzwrotną i symetryczną relacją $R \subseteq V \times V$, tzn. graf nie zawiera pętli i jest złożony z krawędzi nieskierowanych. Symetryczną parę (u, v) i (v, u) elementów relacji R nazywamy **krawędzią** e i oznaczamy ją

$$e = \{u, v\}.$$

Zbiór krawędzi oznaczmy literą E . Zauważmy, że zbiór krawędzi E całkowicie określa relację R . Zatem możemy przedstawiać graf za pomocą jego wierzchołków oraz krawędzi, tj.

$$G = (V, E).$$

Graf $G' = (V', E')$ nazywamy **podgrafem** grafu $G = (V, E)$, gdy $V' \subset V$ oraz

$$E' = \{\{v, w\} \in E : v, w \in V'\}.$$

Parę krawędzi grafu **nazywamy sąsiadami**, gdy mają dokładnie jeden ten sam wierzchołek. Niech $v, v' \in V$. **Stopień** $\deg v$ wierzchołka v jest liczbą różnych wierzchołków, z którymi wierzchołek v tworzy krawędź.

Twierdzenie 3.1.1 ([18], wniosek 2.1(a)). *W każdym grafie jest parzysta liczba wierzchołków stopnia nieparzystego.*

Niech $n \in \mathbb{N}$. **Ścieżką** łączącą wierzchołki $v, v' \in V$ nazywamy taki ciąg wierzchołków $p = (v_0, v_1, \dots, v_n)$, że $v = v_0, v' = v_n, \{v_i, v_{i+1}\} \in E$ oraz $\{v_i, v_{i+1}\} \neq \{v_j, v_{j+1}\}$ dla takich $i, j \in \{0, 1, \dots, n-1\}$, że $i \neq j$, i oznaczamy go:

$$v \xrightarrow{p} v'.$$

Jeżeli $n > 1$, to ciąg krawędzi $q = (\{v, v_1\}, \{v_1, v_2\}, \dots, \{v_{n-1}, v'\})$ jednoznacznie wyznacza ścieżkę (v, v_1, \dots, v') i wówczas taki ciąg q również nazywamy ścieżką łączącą wierzchołek v z v' . Wierzchołki v i v' nazywamy, odpowiednio, **początkiem** oraz **końcem** ścieżki p i oznaczamy, odpowiednio, \tilde{p} oraz \bar{p} . Ścieżkę p nazywamy **ścieżką rzędu drugiego**, gdy $\deg v_k = 2$ dla każdego $k \in \{1, 2, \dots, n-1\}$. Ścieżka p' jest **podścieżką** ścieżki p , jeżeli jest ciągiem kolejnych wierzchołków ścieżki p . **Cykl** to ścieżka, która ma początek i koniec w tym samym wierzchołku. Graf nazywamy **spójnym**, gdy każda para jego wierzchołków jest połączona ścieżką. **Pokryciem ścieżkowym** P grafu (V, E) nazywamy taki zbiór jego ścieżek, że dla każdej krawędzi $e \in E$ istnieje dokładnie jedna taka ścieżka $p \in P$, że $e \in p$.

Twierdzenie 3.1.2. *Niech $n \in \mathbb{N}$. Jeżeli ciąg (v_0, v_1, \dots, v_n) jest ścieżką, to ciąg $(v_n, v_{n-1}, \dots, v_0)$ również jest ścieżką.*

Niech $p = (v_0, v_1, \dots, v_n)$ będzie ścieżką grafu, gdzie $n \in \mathbb{N}$. Wówczas ścieżkę $(v_n, v_{n-1}, \dots, v_0)$ nazywamy odwrotnością ścieżki p i oznaczamy ją p^{-1} .

3.2 Heksastrykty

Przystępujemy do zdefiniowania obiektów, które stanowią główny przedmiot naszych badań. W tym celu musimy określić postać wierzchołków sześciokątów tworzących heksagonalną siatkę na płaszczyźnie oraz krawędzi, które te wierzchołki łączą. Załóżmy, że istnieje sześciokąt, którego centralny punkt ma współrzędne (x, y) , oraz że długość boków sześciokątów wynosi 1 (por. rys. 3.1). Wówczas współrzędne punktów centralnych sąsiadujących sześciokątów mają współrzędne: $(x - \sqrt{3}, y)$, $(x - \frac{\sqrt{3}}{2}, y + \frac{3}{2})$, $(x + \frac{\sqrt{3}}{2}, y + \frac{3}{2})$, $(x + \sqrt{3}, y)$, $(x + \frac{\sqrt{3}}{2}, y - \frac{3}{2})$ i $(x - \frac{\sqrt{3}}{2}, y - \frac{3}{2})$. Łatwo zauważyć, że współrzędne punktów centralnych takich sześciokątów mają postać:

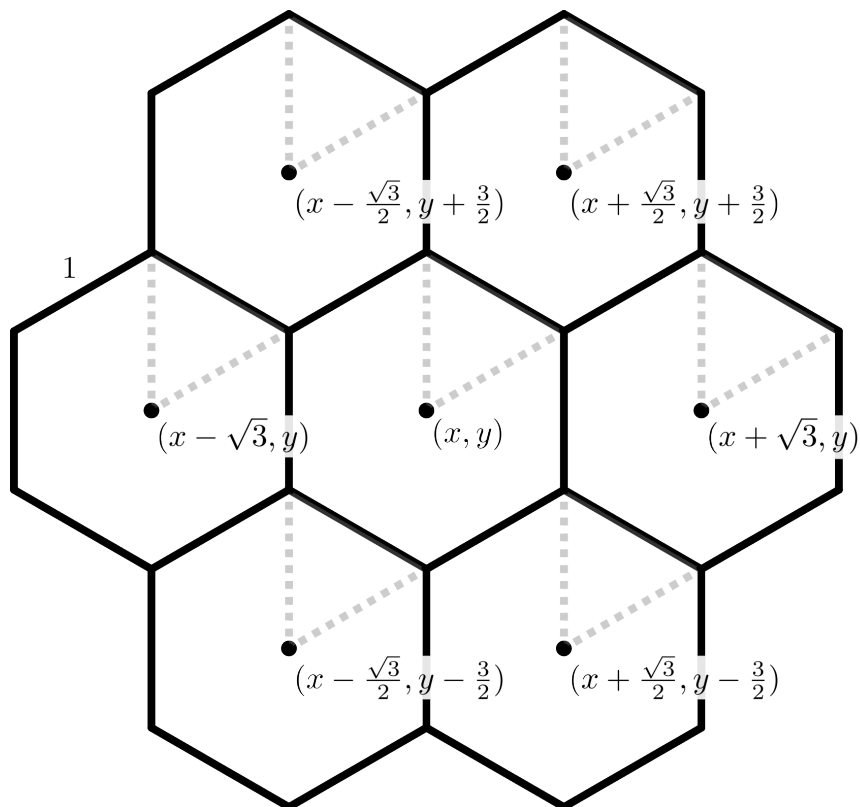
$$\left(i\sqrt{3} + \frac{j\sqrt{3}}{2}, \frac{3j}{2} \right),$$

gdzie $i, j \in \mathbb{Z}$. Stąd, dla każdego takiego punktu, możemy obliczyć współrzędne wierzchołków sześciokąta, którego jest środkiem (por. rys. 3.2). Wówczas wierzchołki sześciokątów tworzących siatkę heksagonalną mają współrzędne:

$$\left(i\sqrt{3} + \frac{j\sqrt{3}}{2} + \frac{k\sqrt{3}}{2}, \frac{3j}{2} + 1 - \frac{k}{2} \right), \quad (3.2.1)$$

gdzie $k \in \{0, 1\}$.

Graf spójny $H = (V, E)$, w którym wierzchołki są punktami na płaszczyźnie o współrzędnych postaci (3.2.1) oraz krawędzie istnieją wyłącznie między wierzchołkami, których



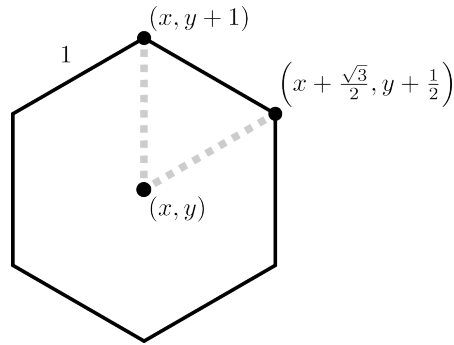
Rysunek 3.1: Sześciokąt foremny o środku w punkcie (x, y) otoczony przylegającymi sześciokątami foremnymi z zaznaczonymi środkami. Wszystkie sześciokąty mają boki długości 1.

odległość wynosi 1, nazywamy **heksastruktem**. Na mocy twierdzeń 1.4.3 oraz 1.4.4 obroty heksastrukturów nie wpływają na ich kształt, gdyż zachowane są długości krawędzi oraz miary kątów, które sąsiadujące krawędzie tworzą na rysunku grafu. Nasze zainteresowanie koncentruje się na geometrycznym kształcie rysunku grafu, zatem heksastruktem nazywamy również każde przekształcenie współrzędnych wierzchołków heksastruktury względem rotacji, odbić lustrzanych oraz skalowania. Dlatego bez straty ogólności przyjmujemy, że w każdym heksastrukcie istnieje krawędź o wierzchołkach w punktach $(1, 0)$ oraz $(0, 0)$. Zauważmy, że na rysunku heksastruktury każda para krawędzi o wspólnym wierzchołku tworzy kąt 120° .

Podgraf H' grafu H będący heksastruktem nazywamy **podheksastruktem** heksastruktury H . Powiemy, że podheksastrukty $H_1 = (V_1, E_1)$ oraz $H_2 = (V_2, E_2)$ heksastruktury H są **rozłączne**, gdy $E_1 \cap E_2 = \emptyset$.

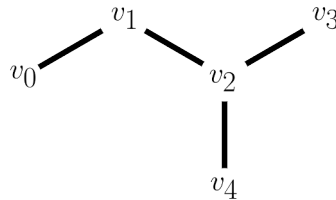
Wniosek 3.2.1. *Stopień wierzchołków w heksastrukcie może być równy 1, 2 lub 3 (por. rys. 3.3).*

Dowód. Przypuśćmy, że w heksastrukcie $H = (V, E)$ istnieje wierzchołek $v \in V$ stopnia $n \in \mathbb{N}$ większy od 3. Wówczas istnieją takie wierzchołki $v_1, v_2, \dots, v_n \in V$, że $\{v, v_k\} \in E$ dla $k \in \{1, 2, \dots, n\}$. Niech krawędź $\{v, v_i\}$ tworzy kąt prawostronny z krawędzią $\{v, v_{i-1}\}$ dla $i \in \{2, 3, \dots, n\}$. Z definicji heksastruktury, każda taka para tworzy kąt 120 stopni. Stąd, $\{v, v_1\}$ i $\{v, v_2\}$ tworzą kąt 120 stopni, $\{v, v_1\}$ i $\{v, v_3\}$ tworzą kąt 240 stopni oraz $\{v, v_1\}$



Rysunek 3.2: Sześciokąt foremny o środku w punkcie (x, y) i o bokach długości 1 z zaznaczonymi dwoma wierzchołkami o współrzędnych $(x, y + 1)$ i $(x + \frac{\sqrt{3}}{2}, y + \frac{1}{2})$.

i $\{v, v_4\}$ tworzą kąt 360 stopni. Na podstawie wniosku 1.2.3, współrzędne wierzchołków v_4 i v_1 pokrywają się. Czyli, taka krawędź już istnieje w E . Sprzeczność. \square



Rysunek 3.3: Heksastrukt złożony z pięciu wierzchołków, gdzie v_0, v_3 i v_4 są wierzchołkami stopnia pierwszego, v_1 jest wierzchołkiem stopnia drugiego oraz v_2 jest wierzchołkiem stopnia trzeciego.

Twierdzenie 3.2.2. Niech heksastrukt $H = (V, E)$ zawiera wierzchołki stopnia co najwyżej drugiego.

- (i) Jeżeli w H istnieją wierzchołki stopnia pierwszego, to są one dokładnie dwa (por. heksastrukt po lewej stronie na rysunku 3.4).
- (ii) Jeżeli wszystkie wierzchołki H są stopnia drugiego, to dla każdego z nich istnieje cykl (por. heksastrukt po prawej stronie na rysunku 3.4).
- (iii) Istnieje ścieżka złożona ze wszystkich krawędzi E .

Dowód.

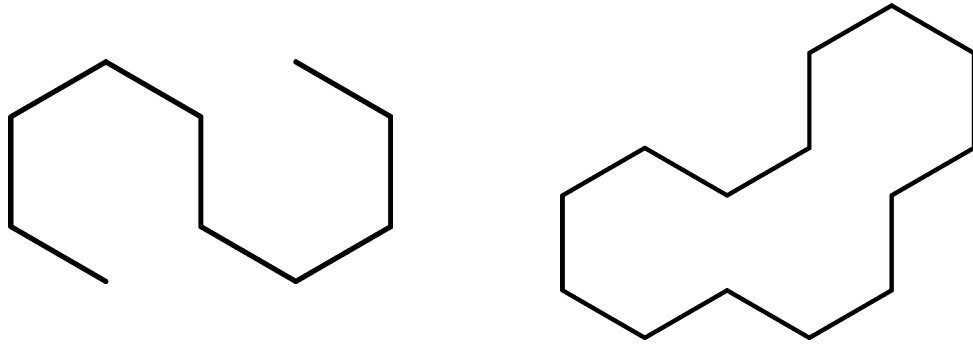
- (i) Niech $|V| = n$ dla takiego $n \in \mathbb{N}$, że $n \geq 2$, i niech $w_0 \in V$ będzie stopnia 1. Zatem istnieje taki $w_1 \in V$, że $\{w_0, w_1\} \in E$. Jeżeli $n = 2$, to w_1 jest szukanym wierzchołkiem, gdyż $\deg w_1 = 1$. W przeciwnym przypadku, jeżeli $\deg w_1 = 1$, to dla wszystkich wierzchołków $w \in V \setminus \{w_0, w_1\}$ nie istnieje taka krawędź $e \in E$, że $w_0 \in e$ lub $w_1 \in e$. Wówczas nie istnieje ścieżka łącząca w_0 z w , co przeczy spójności. Stąd $\deg w_1 = 2$. Niech $n > 2$, $k \in \{0, 1, \dots, n-2\}$ i $\{w_{k-1}, w_k\} \in E$ dla $w_{k-1}, w_k \in V$. Jeżeli $\deg w_k = 1$, to dla wszystkich $w \in V \setminus \{w_0, w_1, \dots, w_k\}$ nie istnieje taka krawędź $e \in E$, że $w_i \in e$

dla $i \in \{0, 1, \dots, k\}$. Wówczas nie istnieje ścieżka łącząca w_0 z w , co przeczy spójności. Stąd $\deg w_k = 2$.

Niech $n > 2$ oraz $\{w_{k-1}, w_k\} \in E$ dla $w_{k-1}, w_k \in V$ i $k \in \{0, 1, \dots, n-1\}$. Jeżeli $\deg w_{n-1} = 2$, to istnieje takie $w \in V \setminus \{w_0, w_1, \dots, w_{n-1}\}$, że $\{w_{n-1}, w\} \in E$. Sprzeczność, gdyż $V \setminus \{w_0, w_1, \dots, w_{n-1}\} = \emptyset$, tzn. nie istnieje takie w . Stąd $\deg w_{n-1} = 1$.

- (ii) Niech $\deg v = 2$ dla wszystkich $v \in V$ oraz niech $|V| = n$. Niech $w_0 \in V$. Wówczas istnieje takie $w_1 \in V$, że $\{w_0, w_1\} \in E$ oraz takie $w_2 \in V$, że $\{w_1, w_2\} \in E$. Stąd $\deg w_1 = 2$. Niech $(w_0, w_1, \dots, w_{n-1})$ będzie takim ciągiem wierzchołków V , że $\{w_k, w_{k+1}\} \in E$ dla $k \in \{0, 1, \dots, n-2\}$. Wówczas $\deg w_k = 2$ dla $k \in \{1, 2, \dots, n-2\}$. Stąd $\{w_0, w_{n-1}\} \in E$, tzn. istnieje $w_0 \rightsquigarrow w_0$.
- (iii) Jeżeli $v \stackrel{P}{\rightsquigarrow} v'$ dla takich $v, v' \in V$, że $\deg v = 1 = \deg v'$, to p jest szukaną ścieżką. W przeciwnym przypadku każdy cykl $v \rightsquigarrow v$, dla $v \in V$, jest szukaną ścieżką.

□



Rysunek 3.4: Heksastryk o dwóch wierzchołkach stopnia pierwszego i pozostałych wierzchołkach stopnia drugiego po lewej stronie oraz heksastryk o wierzchołkach stopnia drugiego po prawej stronie.

W kolejnych obserwacjach pokazujemy, w jaki sposób można podzielić heksastryk na podheksastryki, z których jeden jest ścieżką rzędu drugiego. Stosujemy podejście dziel i zwyciężaj, polegające na rozbiciu problemu na rozłączne podproblemy (w twierdzeniu 3.2.3 rozważamy heksastryk bez wierzchołków stopnia pierwszego, a w twierdzeniu 3.2.5, z co najmniej jednym takim wierzchołkiem), które następnie pozwalają na wydzielenie takiej ścieżki rzędu drugiego w dowolnym heksastryku.

Twierdzenie 3.2.3. Niech heksastryk $H = (V, E)$ zawiera wyłącznie wierzchołki stopnia co najmniej drugiego.

- (i) Liczba wierzchołków stopnia trzeciego w heksastryku H jest parzysta.
- (ii) Dla każdego wierzchołka stopnia trzeciego istnieje ścieżka rzędu drugiego łącząca go z wierzchołkiem stopnia trzeciego (por. heksastryk po lewej stronie na rysunku 3.5).

Dowód. Niech $H = (V, E)$ będzie heksastrykiem o wierzchołkach stopnia co najmniej drugiego.

- (i) Parzysta liczba wierzchołków stopnia trzeciego w heksastrukcie wynika bezpośrednio z twierdzenia 3.1.1.
- (ii) W H nie istnieją wierzchołki stopnia pierwszego, więc na podstawie spójności H oraz skończoności zbioru V wierzchołek stopnia trzeciego $v \in V$ jest połączony ścieżką rzędu drugiego z wierzchołkiem stopnia trzeciego $w \in V$. Zauważmy, że szukany wierzchołek w może być wierzchołkiem v , tzn. ta ścieżka jest cyklem.

□

Przykład 3.2.4. Na rysunku 3.5 po lewej stronie przedstawiliśmy heksastrukt rozważany w twierdzeniu 3.2.3, który zawiera ścieżkę rzędu drugiego łączącą dwa wierzchołki stopnia trzeciego (zaznaczoną kolorem czerwonym).

Twierdzenie 3.2.5. Jeżeli wierzchołek stopnia pierwszego jest połączony ścieżką rzędu drugiego p z wierzchołkiem stopnia drugiego lub trzeciego w heksastrukcie $H = (V, E)$, to istnieją takie dwa rozłączne podheksastrukty $H_1 = (V_1, E_1)$ oraz $H_2 = (V_2, E_2)$ heksastruktu H , że $E_1 = \{e \in E : e \in p\}$, $V_1 = \{v \in V : v \in e, e \in p\}$, $V_1 \cup V_2 = V$ oraz $E_1 \cup E_2 = E$.

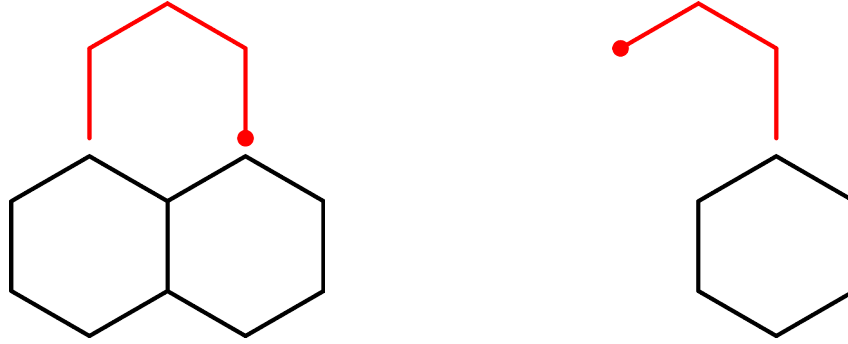
Dowód. Niech $H = (V, E)$ będzie takim heksastruktem, że istnieje ścieżka rzędu drugiego $u \xrightarrow{p} v$, gdzie $u \in V$ jest wierzchołkiem stopnia pierwszego oraz $v \in V$ jest wierzchołkiem stopnia drugiego lub trzeciego. Zauważmy, że graf $H_1 = (V_1, E_1)$, gdzie $E_1 = \{e \in E : e \in p\}$ oraz $V_1 = \{v \in V : v \in e, e \in E_1\}$, jest podheksastruktem heksastruktu H . Wystarczy dowieść, że graf $H_2 = (V_2, E_2)$, gdzie $E_2 = E \setminus E_1$ oraz $V_2 = \{v \in V : v \in e, e \in E_2\}$, jest heksastruktem. Niech $w_1 \xrightarrow{q} w_2$ w H , gdzie $w_1, w_2 \in V_2$. Załóżmy, że nie istnieje ścieżka $w_1 \rightsquigarrow w_2$ w H_2 . Wówczas istnieje taka krawędź $e \in p$, że $e \in q$. Każda ścieżka łącząca wierzchołki krawędzi e z w_1 i w_2 zawiera taką krawędź $f \in E$, że $v \in f$. Ponieważ p jest rzędu drugiego, więc $V_1 \cap V_2 = \{v\}$, a stąd ścieżka q zawiera dwukrotnie krawędź f . Sprzeczność. □

Przykład 3.2.6. Na rysunku 3.5 po prawej stronie przedstawiliśmy heksastrukt rozważany w twierdzeniu 3.2.5, który zawiera ścieżkę rzędu drugiego łączącą wierzchołek stopnia pierwszego z wierzchołkiem stopnia trzeciego (zaznaczoną kolorem czerwonym).

W kolejnych twierdzeniach pokazujemy sposób podziału heksastruktu H na dwa podheksastrukty ze względu na obecność cyklu w H .

Twierdzenie 3.2.7. Jeżeli p jest ścieżką rzędu drugiego będącą podścieżką cyklu w heksastrukcie $H = (V, E)$, to istnieją takie rozłączne podheksastrukty $H_1 = (V_1, E_1)$ i $H_2 = (V_2, E_2)$ heksastruktu H , że $E_1 = \{e \in E : e \in p\}$, $V_1 = \{v \in V : v \in e, e \in E_1\}$, $V_1 \cup V_2 = V$ oraz $E_1 \cup E_2 = E$.

Dowód. Niech $H = (V, E)$ będzie heksastruktem oraz niech $u \xrightarrow{p} v$ będzie ścieżką rzędu drugiego będącą podścieżką cyklu c w heksastrukcie H , gdzie $u, v \in V$. Zauważmy, że heksastrukt $H_1 = (V_1, E_1)$, gdzie $E_1 = \{e \in E : e \in p\}$ oraz $V_1 = \{v \in V : v \in e, e \in E_1\}$, jest podheksastruktem heksastruktu H . Wystarczy dowieść, że graf $H_2 = (V_2, E_2)$, gdzie $E_2 = E \setminus E_1$ oraz $V_2 = \{v \in V : v \in e, e \in E_2\}$, jest heksastruktem. Zauważmy, że jeżeli



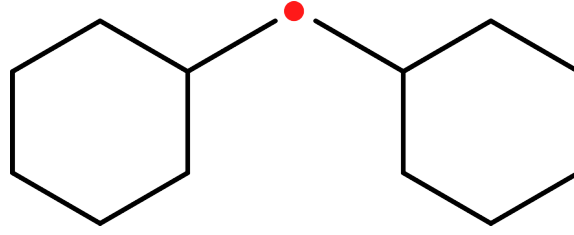
Rysunek 3.5: Heksastrukt niezawierający wierzchołków stopnia pierwszego z zaznaczoną kolorem czerwonym ścieżką rzędu drugiego łączącą wierzchołek stopnia trzeciego z wierzchołkiem stopnia trzeciego po lewej stronie. Heksastrukt z zaznaczoną kolorem czerwonym ścieżką rzędu drugiego łączącą wierzchołek stopnia pierwszego z wierzchołkiem stopnia trzeciego po prawej stronie.

ścieżka $w_1 \overset{q}{\rightsquigarrow} w_2$ łącząca wierzchołki $w_1, w_2 \in V_2$ istnieje w H oraz nie istnieje w H_2 , tzn. istnieje taka krawędź $e \in p$, że $e \in q$. Zatem p jest podścieżką q , ale p jest podścieżką cyklu, więc istnieje ścieżka $u \overset{r}{\rightsquigarrow} v$ różna od p . Stąd istnieje ścieżka $w_1 \rightsquigarrow w_2$. Sprzeczność. \square

Twierdzenie 3.2.8. *Jeżeli nie istnieje cykl zawierający wierzchołek stopnia drugiego v heksastruktu H , to istnieją takie rozłączne podheksastrukty $H_1 = (V_1, E_1)$ i $H_2 = (V_2, E_2)$ heksastruktu H , że $V_1 \cap V_2 = \{v\}$, $V_1 \cup V_2 = V$ oraz $E_1 \cup E_2 = E$.*

Dowód. Niech v będzie wierzchołkiem stopnia drugiego w heksastrukcie $H = (V, E)$, dla którego nie istnieje cykl zawierający v . Wówczas istnieją takie wierzchołki v' i v'' , że $\{v, v'\} \in E$ oraz $\{v, v''\} \in E$. Na mocy spójności H , dla każdego $u \in V$ istnieje ścieżka $u \overset{p}{\rightsquigarrow} v$. Wówczas istnieje taka krawędź $e \in p$, że $v' \in e$ lub $v'' \in e$. Ponadto, jeżeli istnieje taka krawędź $e \in p$, że $v' \in e$, to $v'' \notin e$ dla wszystkich krawędzi $e \in p$. Podobnie, jeżeli istnieje taka krawędź $e \in p$, że $v'' \in e$, to $v' \notin e$ dla wszystkich krawędzi $e \in p$. Niech $V_1 \subset V$ będzie zbiorem takich wierzchołków $u \in V$, że $u = v$ lub $v' \in e$ dla pewnej krawędzi $e \in p$, gdzie $u \overset{p}{\rightsquigarrow} v$. Ponadto niech $E_1 = \{\{u, w\} \in E : u, w \in V_1, u \neq w\}$. Wówczas dla każdej pary wierzchołków w V_1 istnieje ścieżka złożona z krawędzi zbioru E_1 . Zatem graf $H_1 = (V_1, E_1)$ tworzy heksastrukt. Analogicznie, niech $V_2 \subset V$ będzie zbiorem takich wierzchołków $u \in V$, że $u = v$ lub $v'' \in e$ dla pewnej krawędzi $e \in p$, gdzie $u \overset{p}{\rightsquigarrow} v$ oraz niech $E_2 = \{\{u, w\} \in E : u, w \in V_2, u \neq w\}$. Graf $H_2 = (V_2, E_2)$ tworzy heksastrukt. Stąd H_1 i H_2 są rozłącznymi podheksastruktami heksastruktu H spełniającymi tezę twierdzenia. \square

Przykład 3.2.9. *Na rysunku 3.6 przedstawiliśmy heksastrukt H rozważany w twierdzeniu 3.2.8, który zawiera wierzchołek nienależący do żadnego cyklu w H (zaznaczony kolorem czerwonym). Wówczas możemy ten heksastrukt podzielić na dwa rozłączne podheksastrukty względem tego wierzchołka.*



Rysunek 3.6: Heksastrukt H z wierzchołkiem zaznaczonym kolorem czerwonym, dla którego nie istnieje cykl w H podzielony na dwa podheksastryki względem tego wierzchołka.

3.3 Właściwe pokrycia ścieżkowe heksastruktów

W tej części zdefiniujemy specyficzny typ pokrycia ścieżkowego heksastruktu, w którym koniec danej ścieżki jest początkiem dokładnie dwóch ścieżek albo nie ma kontynuacji, oraz początek każdej ścieżki jest końcem dokładnie jednej ścieżki. Dzięki temu heksastrukt przedstawimy jako drzewo binarne, w którym wierzchołkami są początki i końce ścieżek rzędu drugiego.

Powiemy, że pokrycie ścieżkowe P heksastruktu H z ustaloną ścieżką $p \in P$ jest **właściwe**, gdy spełnione są następujące warunki:

- dla każdej ścieżki $q \in P \setminus \{p\}$ istnieje dokładnie jedna ścieżka, której koniec znajduje się w \bar{q} ,
- dla każdej ścieżki $q \in P$ spełniony jest dokładnie jeden z następujących warunków:
 - dla każdej ścieżki $r \in P \setminus \{p\}$ zachodzi nierówność $\bar{q} \neq \bar{r}$,
 - istnieją dokładnie dwie takie ścieżki $s_1, s_2 \in P \setminus \{q\}$, że $\bar{s}_1 = \bar{q} = \bar{s}_2$.

Ustaloną ścieżkę $p \in P$ nazywamy **początkiem** właściwego pokrycia ścieżkowego P . Krawędź $e \in p$ dla początku $p \in P$, taką, że $\bar{p} \in e$, nazywamy **krawędzią początkową** właściwego pokrycia ścieżkowego P .

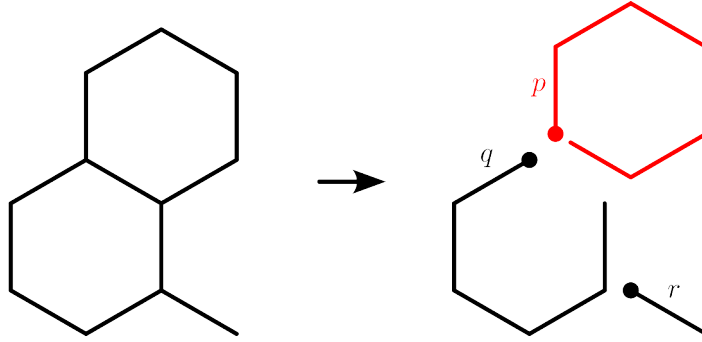
Przykład 3.3.1. Na rysunku 3.7 przedstawiliśmy pokrycie $\{p, q, r\}$ heksastruktu H , które nie jest właściwe. Nie istnieją dokładnie dwie ścieżki różne od p o początku w \bar{p} . Ponadto nie istnieje ścieżka o końcu w \bar{r} .

Właściwe pokrycie ścieżkowe istnieje dla każdego heksastruktu. Pokażemy to konstruując takie pokrycie z wykorzystaniem indukcji strukturalnej ze względu na liczbę wierzchołków stopnia trzeciego w rozważanym heksastrukcie.

Twierdzenie 3.3.2. *Dla każdego heksastruktu istnieje co najmniej jedno pokrycie właściwe.*

Dowód. Niech $H = (V, E)$ będzie heksastruktem. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na liczbę wierzchołków stopnia trzeciego w heksastrukcie H .

Przypadek bazowy: Jeżeli w H nie istnieje wierzchołek stopnia trzeciego, to zachodzi dokładnie jeden z następujących przypadków:



Rysunek 3.7: Pokrycie $\{p, q, r\}$ heksastruktury H , które nie jest właściwe, gdyż nie istnieją dokładnie dwie różne od p ścieżki o początku w \vec{p} oraz nie istnieje ścieżka o końcu w \vec{r} dla ustalonej ścieżki p .

- istnieją dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego (por. heksastrukt po lewej stronie na rysunku 3.4),
- wszystkie wierzchołki są stopnia drugiego (por. heksastrukt po prawej stronie na rysunku 3.4).

Na mocy twierdzenia 3.2.2, istnieje ścieżka złożona ze wszystkich wierzchołków heksastruktury H . Zatem ta ścieżka jest szukanym pokryciem tego heksastruktury.

Założenie indukcyjne: Niech H zawiera n wierzchołków stopnia trzeciego dla $n \in \mathbb{N}$.

Krok indukcyjny: Ze względu na skończoność zbioru wierzchołków heksastruktury istnieje taki wierzchołek v połączony ścieżką rzędu drugiego p z wierzchołkiem stopnia trzeciego w , że $\deg v = 1$ albo istnieje taki cykl c zawierający v w H , że p jest podścieżką c .

Na mocy twierdzeń 3.2.5 oraz 3.2.7, istnieją takie rozłączne podheksastruktury $H_1 = (V_1, E_1)$ i $H_2 = (V_2, E_2)$ heksastruktury H , że $E_1 = \{e \in E : e \in p\}$ i $V_1 = \{v \in V : v \in e, e \in E_1\}$ oraz $V_1 \cup V_2 = V$ i $E_1 \cup E_2 = E$.

Jeżeli w H_2 nie istnieje cykl dla wierzchołka w , to, na mocy twierdzenia 3.2.8, istnieją takie podheksastruktury $H'_2 = (V'_2, E'_2)$ oraz $H''_2 = (V''_2, E''_2)$ heksastruktury H_2 , że $V'_2 \cap V''_2 = \{w\}$, $V'_2 \cup V''_2 = V_2$, $E'_2 \cup E''_2 = E_2$ oraz w jest wierzchołkiem stopnia pierwszego zarówno w H'_2 jak i w H''_2 . H'_2 oraz H''_2 zawierają mniej niż n wierzchołków stopnia trzeciego. Na mocy założenia indukcyjnego, dla ustalonej ścieżki o początku w wierzchołku w , istnieją pokrycia P' oraz P'' heksastruktów, odpowiednio, H'_2 oraz H''_2 spełniające warunki tezy. Wówczas szukanym pokryciem jest $P = \{p\} \cup P' \cup P''$.

Jeżeli natomiast w H_2 istnieje cykl c zawierający wierzchołek w , to istnieje ścieżka rzędu drugiego q łącząca w z wierzchołkiem stopnia trzeciego w H_2 , będąca podścieżką cyklu c . Na mocy założenia indukcyjnego, dla ustalonej ścieżki q , istnieje pokrycie ścieżkowe P' heksastruktury H_2 , spełniające warunki tezy. Wówczas szukanym pokryciem jest $P = \{p\} \cup P'$. \square

Zauważmy, że pokrycia właściwe mają strukturę drzewa binarnego. Dowód tego faktu przeprowadzimy z wykorzystaniem indukcji ze względu na maksymalną liczbę kolejno połą-

czonych ścieżek pokrycia, tj. największą długość takiego ciągu ścieżek pokrycia właściwego, w którym początek każdej kolejnej ścieżki jest końcem poprzedniej ścieżki. Definiujemy funkcję $\vartheta: \mathcal{P} \times P \rightarrow \mathbb{N}$ zwracającą **maksymalną liczbę kolejno połączonych ścieżek** w pokryciu P , począwszy od ustalonej ścieżki $p \in P$, w następujący sposób:

- $\vartheta_P(p) = 1$, gdy w $P \setminus \{p\}$ nie istnieje ścieżka o początku w \vec{p} ,
- $\vartheta_P(p) = 1 + \max\{\vartheta_{P \setminus \{p\}}(p'), \vartheta_{P \setminus \{p\}}(p'')\}$, gdzie $p', p'' \in P \setminus \{p\}$ są różnymi ścieżkami o początkach w \vec{p} .

Ponadto, gdy p jest początkiem właściwego pokrycia P , to wartość funkcji ϑ jest maksymalną liczbą kolejno połączonych ścieżek w P .

Twierdzenie 3.3.3. *Jeżeli P jest takim właściwym pokryciem o początku w $p \in P$, że $|P| > 1$ oraz $p_1, p_2 \in P$ są różnymi ścieżkami o początkach w \vec{p} , to istnieją takie pokrycia właściwe $P_1, P_2 \subset P$ o początkach, odpowiednio, w p_1 i p_2 , że $P_1 \cup P_2 = P$ oraz $P_1 \cap P_2 = \emptyset$.*

Dowód. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na maksymalną liczbę kolejno połączonych ścieżek w pokryciu P . Niech p będzie początkiem właściwego pokrycia ścieżkowego P heksastruktury H . Jeżeli $|P| > 1$, to istnieją takie różne ścieżki $q, r \in P$, że $\vec{q} = \vec{p} = \vec{r}$.

Przypadek bazowy: Jeżeli $|P| > 1$, to najmniejsza możliwa maksymalna liczba kolejno połączonych ścieżek w P wynosi 2. Wówczas $|P| = 3$, $P = \{p, q, r\}$ oraz końce q i r nie są początkami innych ścieżek. Zatem szukane heksastrukty są utworzone przez te ścieżki, które jednocześnie są ich pokryciami.

Założenie indukcyjne: Niech $\vartheta_P(p) = n$ dla $n \in \mathbb{N}$.

Krok indukcyjny: Zauważmy, że ścieżka p należy do najdłuższego ciągu kolejno połączonych ścieżek, jako ścieżka której początek jest niewzględniony jako koniec jakiegokolwiek innej ścieżki. Wówczas $\vartheta_P(p) = 1 + \max\{\vartheta_P(q), \vartheta_P(r)\}$ Stąd $\vartheta_P(q) < \vartheta_P(p)$ oraz $\vartheta_P(r) < \vartheta_P(p)$, więc możemy skorzystać z założenia indukcyjnego. Zatem istnieją podheksastrukty H_1 i H_2 heksastruktury H o takich właściwych pokryciach, odpowiednio, P_1 i P_2 , o początkach, odpowiednio, w q i r , że $P_1 \cap P_2 = \emptyset$ oraz $P_1 \cup P_2 \cup \{p\} = P$. \square

Twierdzenie 3.3.4. *Niech P będzie właściwym pokryciem heksastruktury H dla ustalonej początkowej ścieżki $p \in P$. Jeżeli $q \in P$ jest taką ścieżką, że $\vec{q} \neq \vec{r}$ dla wszystkich $r \in P$, to istnieje takie pokrycie właściwe Q heksastruktury H dla ustalonej ścieżki początkowej q^{-1} , że dla każdej ścieżki $s \in Q$ mamy $s \in P$ lub $s^{-1} \in P$.*

Dowód. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na maksymalną liczbę kolejno połączonych ścieżek w pokryciu P . Niech P będzie właściwym pokryciem heksastruktury H dla ustalonej ścieżki początkowej $p \in P$.

Przypadek bazowy: Jeżeli $\vartheta_P(p) = 1$, to $|P| = 1$ oraz p jest jedyną ścieżką w P . Wówczas $Q = \{p^{-1}\}$ również jest właściwym pokryciem heksastruktury H o ustalonej

ścieżce początkowej p^{-1} .

Założenie indukcyjne: Niech $\vartheta_P(p) = n$ dla pewnego $n \in \mathbb{N}$ oraz niech $q \in P \setminus \{p\}$ będzie taką ścieżką, że $\vec{q} \neq \vec{r}$ dla wszystkich $r \in P$.

Krok indukcyjny: Zauważmy, że istnieje dokładnie jedna taka ścieżka $r \in P$, że $\vec{q} = \vec{r}$. Wówczas istnieje taka ścieżka $s \in P$ różna od q , że $\vec{s} = \vec{q}$. Stąd

$$\vec{q}^{-1} = \vec{r}^{-1} = \vec{s}.$$

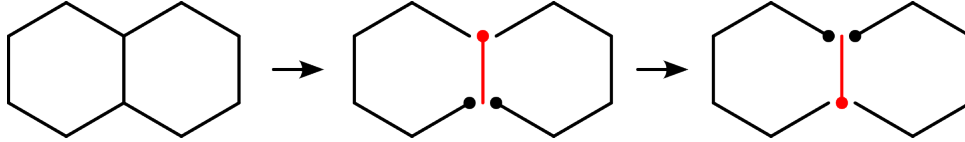
Na mocy twierdzenia 3.3.3 istnieją takie właściwe pokrycia $P_1, P_2 \subset P \setminus \{p\}$ podheksastruktów, odpowiednio, H_1 i H_2 heksastruktu H , że $P_1 \cup P_2 \cup \{p\} = P$ oraz $P_1 \cap P_2 = \emptyset$. Wówczas $q \in P_1$ lub $q \in P_2$. Załóżmy bez straty ogólności, że $q \in P_1$. Zauważmy, że istnieją takie $p', p'' \in P$, że $\vec{p}' = \vec{p} = \vec{p}''$ oraz $p' \in P_1$ i $p'' \in P_2$. Wówczas $\vartheta_P(p) > \vartheta_P(p')$. Na mocy założenia indukcyjnego istnieje takie pokrycie właściwe P'_1 heksastruktu H_1 o początku w q^{-1} , że dla każdej ścieżki $r \in P_1$ zachodzi $r \in P'_1$ lub $r^{-1} \in P'_1$. Zauważmy, że $p'^{-1} \in Q$, więc szukanym pokryciem właściwym heksastruktu H o początku w q^{-1} jest $P_2 \cup P'_1 \cup \{p^{-1}\}$. \square

Twierdzenie 3.3.5. *Niech P będzie właściwym pokryciem heksastruktu H z ustaloną ścieżką początkową $p \in P$. Jeżeli istnieją takie ścieżki $q_1, q_2 \in P$, że $\vec{q}_1 = \vec{p} = \vec{q}_2$, istnieją takie ścieżki $p_1, p_2 \in P$, że $\vec{p}_1 = \vec{p} = \vec{p}_2$ oraz istnieją takie pokrycia właściwe $P_1, P_2 \subset P$ o początkach w, odpowiednio, p_1 i p_2 , że $P_1 \cup P_2 = P$, $P_1 \cap P_2 = \emptyset$, $q_1 \in P_1$, $p_1 \in P_1$, $q_2 \in P_2$ oraz $p_2 \in P_2$, to istnieje takie pokrycie właściwe Q heksastruktu H o początku w p^{-1} , że dla każdej ścieżki $q \in Q$ zachodzi $q \in P$ lub $q^{-1} \in P$.*

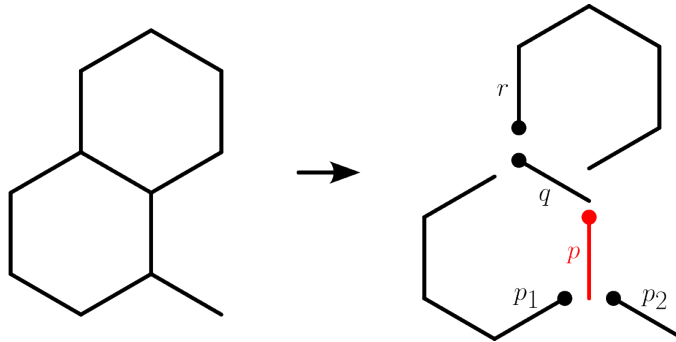
Dowód. Niech P będzie takim właściwym pokryciem heksastruktu H z ustaloną ścieżką początkową $p \in P$, że istnieją takie $q_1, q_2 \in P$, że $\vec{q}_1 = \vec{p} = \vec{q}_2$, istnieją takie ścieżki $p_1, p_2 \in P$, że $\vec{p}_1 = \vec{p} = \vec{p}_2$ oraz istnieją takie pokrycia właściwe $P_1, P_2 \subset P$ o początkach w, odpowiednio, p_1 i p_2 , że $P_1 \cup P_2 = P$, $P_1 \cap P_2 = \emptyset$, $q_1 \in P_1$, $p_1 \in P_1$, $q_2 \in P_2$ oraz $p_2 \in P_2$. Zauważmy, że $\vec{q}_1 \neq \vec{s}_1$ oraz $\vec{p}_1 \neq \vec{s}_1$ dla wszystkich $s_1 \in P_1$, a także że $\vec{q}_2 \neq \vec{s}_2$ oraz $\vec{p}_2 \neq \vec{s}_2$ dla wszystkich $s_2 \in P_2$. Na mocy twierdzenia 3.3.4 istnieją takie pokrycia właściwe Q_1 i Q_2 o początkach w, odpowiednio, q_1 i q_2 , że dla każdej ścieżki $s_1 \in Q_1$ zachodzi $s_1 \in P_1$ lub $s_1^{-1} \in P_1$ oraz dla każdej ścieżki $s_2 \in Q_2$ zachodzi $s_2 \in P_2$ lub $s_2^{-1} \in P_2$. Zatem $Q = \{p^{-1}\} \cup Q_1 \cup Q_2$ jest szukanym pokryciem właściwym heksastruktu H o początku w p^{-1} . \square

Przykład 3.3.6. *Heksastrukt oraz jego pokrycia przedstawione na rysunku 3.8 spełniają założenia twierdzenia 3.3.5.*

Przykład 3.3.7. *Rozważamy heksastrukt oraz jego pokrycie $P = \{p, p_1, p_2, q, r\}$ jak na rysunku 3.9. Jeżeli rozważymy pokrycie o początku w p^{-1} oraz o ścieżkach $s \in P$ lub $s^{-1} \in P$, to wówczas nie będzie ono właściwe, gdyż ani p_2 , ani p_2^{-1} nie będzie spełniało warunków definicji pokrycia właściwego, tj. p_2 będzie rozpoczynała się w wierzchołku, w którym rozpoczyna się inna ścieżka, lub nie będzie istniała ścieżka o końcu w \vec{p}_2 . Zatem warunek istnienia dwóch rozłącznych podpokryć w twierdzeniu 3.3.5 jest istotny.*



Rysunek 3.8: Heksastrukt złożony z dwóch sześciokątów o wspólnej krawędzi (po lewej stronie). Pokrycie właściwe tego heksastruktu o początku zaznaczonym kolorem czerwonym, w którym pierwszym wierzchołkiem jest pierwszy wierzchołek stopnia trzeciego (środkowy). Pokrycie właściwe tego samego heksastruktu o początku zaznaczonym kolorem czerwonym, w którym pierwszym wierzchołkiem jest drugi wierzchołek stopnia trzeciego (po prawej stronie).



Rysunek 3.9: Pokrycie właściwe $P = \{p, p_1, p_2, q, r\}$ (po prawej) heksastrukt (po lewej) o początku w p , dla którego nie istnieje pokrycie właściwe o początku w p^{-1} , zawierające ścieżki $s \in P$ lub $s^{-1} \in P$. Kropkami oznaczono początki poszczególnych ścieżek. Na czerwono zaznaczono początek pokrycia P .

Niech $p = (e_1, e_2, \dots, e_n)$, gdzie $n \in \mathbb{N}$, będzie ścieżką heksastrukt H . Ścieżkę $q = (e'_1, e'_2, \dots, e'_n)$ nazywamy odbiciem lustrzanym ścieżki p , gdy spełnione są następujące warunki:

- jeżeli e_k i e_{k+1} tworzą kąt lewostronny, to e'_k i e'_{k+1} tworzą kąt prawostronny,
- jeżeli e_k i e_{k+1} tworzą kąt prawostronny, to e'_k i e'_{k+1} tworzą kąt lewostronny,

dla każdego $k \in \{1, 2, \dots, n-1\}$. Jeżeli ścieżka p tworzy pokrycie ścieżkowe grafu H , to graf H' o pokryciu $\{q\}$ nazywamy odbiciem lustrzanym grafu H . Jeżeli zbiór $\{p_1, p_2, \dots, p_m\}$ tworzy pokrycie ścieżkowe heksastrukt H , to heksastrukt H' o pokryciu ścieżkowym $\{q_1, q_2, \dots, q_m\}$ nazywamy odbiciem lustrzanym grafu H , gdy:

- q_i jest odbiciem lustrzanym p_i ,
- jeżeli $\vec{p}_i = \vec{p}_j$, to $\vec{q}_i = \vec{q}_j$,

dla każdego $i, j \in \{1, 2, \dots, m\}$.

Poprawność powyższej definicji wynika bezpośrednio z twierdzeń 1.4.5 oraz 1.4.6.

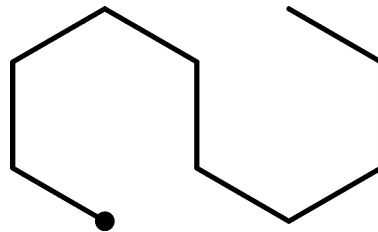
3.4 Reprezentacja heksastrukturów rozszerzonymi ciągami binarnymi

Heksastryki będziemy reprezentować przy użyciu ciągów binarnych oraz rozszerzonych ciągów binarnych. Chcemy zachować możliwość odtworzenia struktury grafu na podstawie danego reprezentanta. Intuicyjnie, krocząc po kolejnych krawędziach ustalonej ścieżki w heksastrukcie, każdy kąt lewostronny utworzony przez sąsiadujące ze sobą krawędzie będziemy oznaczać liczbą 0, natomiast każdy kąt prawostronny utworzony przez sąsiadujące ze sobą krawędzie będziemy oznaczać liczbą 1.

Twierdzenie 3.4.1. *Niech $n \in \mathbb{N} \setminus \{1, 2\}$. Jeżeli ciąg $p = (\{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{n-1}, v_n\})$ jest ścieżką heksastruktury, to istnieje taki ciąg binarny $B = b_0b_1 \dots b_{n-2}$, że $b_k = 0$ ($b_k = 1$) gdy krawędzie $\{v_k, v_{k+1}\}$ i $\{v_{k+1}, v_{k+2}\}$ tworzą kąt lewostronny (prawostronny), gdzie $k \in \{0, 1, \dots, n-2\}$.*

Niech p będzie ścieżką heksastruktury. Powiemy, że ścieżka p jest **reprezentowana przez ciąg binarny B** , gdy spełnia warunki twierdzenia 3.4.1. Jeżeli ścieżka p tworzy pokrycie heksastruktury H , to powiemy, że heksastruktura H jest **reprezentowana ciągiem binarnym B** . Krawędź początkową ścieżki tworzącej pokrycie heksastruktury reprezentowaną przez ciąg binarny B nazywamy **krawędzią początkową** dla ciągu binarnego B .

Przykład 3.4.2. *Na rysunku 3.10 przedstawiliśmy heksastrukturę, którego pokrycie składa się z jednej ścieżki reprezentowanej ciągiem binarnym 11110000, o początku zaznaczonym kropką.*



Rysunek 3.10: Heksastruktura o dwóch wierzchołkach stopnia pierwszego i pozostałych wierzchołkach stopnia drugiego, o pokryciu ścieżką reprezentowaną ciągiem binarnym 11110000, o początku zaznaczonym kropką.

Zauważmy, że jeżeli v jest wierzchołkiem stopnia trzeciego w heksastrukcie H , to istnieją trzy ścieżki rzędu drugiego o początku lub końcu w v oraz o końcu w wierzchołku stopnia pierwszego lub trzeciego. Na mocy twierdzenia 3.1.2, niech p_1 będzie ścieżką rzędu drugiego o końcu w v oraz niech p_2 i p_3 będą różnymi ścieżkami rzędu drugiego o początku w v . Zgodnie z powyższym twierdzeniem, ścieżki p_1 , p_2 i p_3 reprezentujemy ciągami binarnymi, odpowiednio, B_1 , B_2 i B_3 . Stąd taką trójkę ścieżek możemy reprezentować przy pomocy rozszerzonego ciągu binarnego $B_1(B_2)B_3$. Ustalamy dla jednoznaczności, że ostatnia krawędź ścieżki p_1 tworzy kąt lewostronny z pierwszą krawędzią ścieżki p_2 . Wówczas taki rozszerzony ciąg binarny jest wyznaczony jednoznacznie dla tak określonych ścieżek p_1 , p_2 i p_3 . Analogicznie definiujemy reprezentację dowolnego heksastruktury przy pomocy ciągów binarnych oraz rozszerzonych ciągów binarnych.

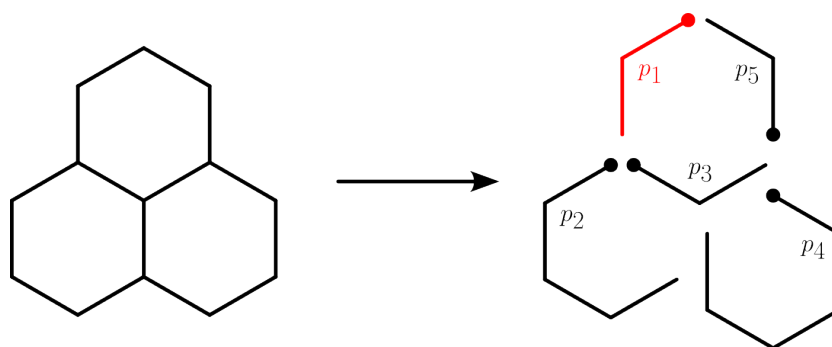
Definiujemy rekurencyjnie reprezentację heksastruktu H o pokryciu właściwym P o początku p z wykorzystaniem ciągów binarnych oraz rozszerzonych ciągów binarnych, ze względu na liczbę ścieżek w P .

Przypadek bazowy: Jeżeli $|P| = 1$, to H reprezentujemy ciągiem binarnym reprezentującym ścieżkę p .

Krok indukcyjny: Na mocy twierdzenia 3.3.3, jeżeli pokrycie właściwe heksastruktu zawiera więcej niż jedną ścieżkę, to możemy je podzielić na dwa rozłączne pokrycia właściwe P_1 i P_2 o początkach w , odpowiednio, q i r , podheksastruktów tego heksastruktu, ze względu na początek tego pokrycia. Wówczas szukaną reprezentacją heksastruktu H jest rozszerzony ciąg binarny $B_0(E_1)E_2$, gdzie B_0 jest ciągiem binarnym reprezentującym ścieżkę p , natomiast na podstawie założenia indukcyjnego E_1 jest reprezentacją pokrycia P_1 o początku w w q , a E_2 jest reprezentacją pokrycia P_2 o początku w w r .

Krawędź początkową pokrycia właściwego heksastruktu reprezentowanego przez rozszerzony ciąg binarny E nazywamy **krawędzią początkową** rozszerzonego ciągu binarnego E .

Przykład 3.4.3. Rozważamy heksastrukt H oraz jego właściwe pokrycie ścieżkowe $P = \{p_1, p_2, p_3, p_4, p_5\}$ przedstawione na rysunku 3.11. Ścieżka p_1 jest reprezentowana przez ciąg binarny 0, ścieżka p_2 przez 000, ścieżka p_3 przez 0, ścieżka p_4 przez 1111 oraz ścieżka p_5 przez 0. Dzielimy pokrycie P na ścieżkę p_1 oraz pokrycia $P_1 = \{p_2\}$ oraz $P_2 = \{p_3, p_4, p_5\}$ podheksastruktów, odpowiednio, H_1 oraz H_2 heksastruktu H . Analogicznie, dzielimy pokrycie P_2 na ścieżkę p_3 oraz pokrycia $P'_2 = \{p_4\}$ oraz $P''_2 = \{p_5\}$ podheksastruktów, odpowiednio, H'_2 oraz H''_2 heksastruktu H_2 . Wówczas heksastrukt H'_2 jest reprezentowany przez ciąg binarny 1111, heksastrukt H''_2 przez ciąg binarny 0, heksastrukt H_2 przez rozszerzony ciąg binarny 0(0)1111, heksastrukt H_1 przez 000 oraz heksastrukt H przez 0(0(0)1111)000.



Rysunek 3.11: Przykładowe właściwe pokrycie ścieżkowe (po prawej) heksastruktu złożonego z trzech sześciokątów o wspólnym wierzchołku (po lewej). Kolorem czerwonym zaznaczono ścieżkę początkową. Kropkami oznaczono pierwsze wierzchołki poszczególnych ścieżek.

Zbadajmy wpływ operacji τ , ζ i π na ciągi binarne oraz rozszerzone ciągi binarne reprezentujące heksastrukty. W każdym przypadku najpierw rozpatrzemy ścieżki heksa-

struktury, a następnie sam heksastrykt. Rozpocznijemy od zamiany bitów, czyli od zamiany kątów lewostronnych na prawostronne oraz prawostronnych na lewostronne.

Twierdzenie 3.4.4. *Jeżeli ciąg binarny B reprezentuje ścieżkę p , to ciąg binarny $\tau(B)$ reprezentuje ścieżkę p' będącą odbiciem lustrzanym ścieżki p .*

Dowód. Zamiana kątów lewostronnych na prawostronne oraz prawostronnych na lewostronne jest równoważna odbiciu lustrzanemu wierzchołków i krawędzi względem prostej l przechodzącej przez pierwszą krawędź ścieżki, na mocy twierdzeń 1.4.5 oraz 1.4.6. \square

Przykład 3.4.5. *Na rysunku 3.12 przedstawiliśmy heksastrykty będące ścieżkami reprezentowanymi przez ciągi binarne 01111 oraz $\tau(01111) = 10000$.*



Rysunek 3.12: Heksastrykty reprezentowane przez ciągi binarne, odpowiednio od lewej do prawej, 01111 oraz $\tau(01111) = 10000$.

Twierdzenie 3.4.6. *Jeżeli rozszerzony ciąg binarny E reprezentuje heksastrykt H , to rozszerzony ciąg binarny $\tau(E)$ reprezentuje heksastrykt H' będący odbiciem lustrzanym heksastryktu H .*

Dowód. Niech $E = B_0(E_1)E_2$ będzie rozszerzonym ciągiem binarnym reprezentującym heksastrykt H . Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na złożoność E .

Przypadek bazowy: Jeżeli $\omega(E) = 1$, to E_1 i E_2 są ciągami binarnymi. Na mocy twierdzeń 2.1.3 i 3.4.4, ciągi binarne $\tau(B_00E_1) = \tau(B_0)1\tau(E_1)$ i $\tau(B_01E_2) = \tau(B_0)0\tau(E_2)$ reprezentują heksastrykty będące odbiciami lustrzanymi heksastryktów reprezentowanych przez ciągi binarne, odpowiednio, B_00E_1 i B_01E_2 . Na mocy twierdzeń 1.4.5 oraz 1.4.6 $\tau(E)$ reprezentuje heksastrykt, który jest odbiciem lustrzanym heksastryktu H .

Założenie indukcyjne: Niech E_1 oraz E_2 będą rozszerzonymi ciągami binarnymi. Zakładamy, że $\tau(E_1)$ i $\tau(E_2)$ reprezentują heksastrykty, które są odbiciami lustrzanymi heksastryktów reprezentowanych przez ciągi, odpowiednio, E_1 i E_2 .

Krok indukcyjny: Na mocy twierdzenia 2.1.3 i założenia indukcyjnego ciągi $\tau(B_00E_1)$ oraz $\tau(B_01E_2)$ reprezentują heksastrykty, które są odbiciem lustrzanym heksastryktów

reprezentowanych przez ciągi, odpowiednio, B_00E_1 oraz B_01E_2 . Zatem $\tau(E)$ reprezentuje heksastrukt będący odbiciem lustrzanym heksastruktu H . \square

Przykład 3.4.7. Na rysunku 3.14 przedstawiliśmy heksastryki reprezentowane przez rozszerzone ciągi binarne $01(0)11$ i $\tau(01(0)11) = 10(00)1$.

Przechodzimy do odwzorowania π , które polega na odwróceniu oraz zamianie kątów lewostronnych na prawostronne oraz prawostronnych na lewostronne rozważanej ścieżki.

Twierdzenie 3.4.8. Jeżeli ciąg binarny B reprezentuje ścieżkę p , to ciąg binarny $\pi(B)$ reprezentuje ścieżkę p^{-1} .

Dowód. Teza twierdzenia wynika bezpośrednio z wniosku 1.2.5. \square

Przykład 3.4.9. Na rysunkach 3.12 i 3.13 przedstawiliśmy heksastryki będące ścieżkami reprezentowane przez ciągi binarne 00001 oraz $\pi(11110) = 00001$.



Rysunek 3.13: Heksastryki reprezentowane przez ciągi binarne, odpowiednio od lewej do prawej, 00001 oraz $\tau(01111) = 11110$.

Wniosek 3.4.10. Jeżeli ciąg binarny B reprezentuje heksastrukt H , to ciąg binarny $\pi(B)$ również reprezentuje heksastrukt H .

Twierdzenie 3.4.11. Jeżeli rozszerzony ciąg binarny E reprezentuje heksastrukt H , to rozszerzony ciąg binarny $\pi(E)$ również reprezentuje heksastrukt H .

Dowód. Niech $(C_i)_{i \in I}$ będzie rozwinięciem rozszerzonego ciągu binarnego E reprezentującego heksastrukt H o pokryciu właściwym P oraz niech $h(k) = g^{(k)}(0)$ i $g(k) = \kappa(k) + 1$ dla każdego $k \in \mathbb{N}$, przy czym $g^{(k)}$ rozumiemy jako k -te złożenie funkcji g . Ponadto niech P_1 , P_2 i P_3 będą pokryciami właściwymi podheksastruktów, odpowiednio, H_1 , H_2 i H_3 heksastruktu H reprezentowanych ciągami, odpowiednio, $C_{h(1)-1}$, $C_{h(1)}$ i $C_{h(1)+1}$. Wówczas $P_1 \cup P_2 \cup P_3 = P$ oraz istnieją takie $p_1 \in P_1$, $p_2 \in P_2$ i $p_3 \in P_3$, że $\vec{p}_1 = \vec{p}_2 = \vec{p}_3$. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na $n \in \mathbb{N}$, dla którego $C_{h(n)}$ jest ciągiem binarnym.

Przypadek bazowy: Jeżeli $n = 1$, to C_2 jest ciągiem binarnym oraz $P_1 = \{p_1\}$ i $P_2 = \{p_2\}$. Ciąg $\pi(C_1)$ reprezentuje ścieżkę $q_1 = p_1^{-1}$ oraz ciąg $\pi(C_2)$ reprezentuje ścieżkę $q_2 = p_2^{-1}$. Ponadto $\vec{q}_2 = \vec{q}_1$ oraz $\vec{q}_2 = \vec{p}_3$. Pokrycie $\{q_1, q_2\} \cup P_3$ również jest pokryciem

właściwym heksastruktu H , więc rozszerzony ciąg binarny $\pi(C_2)(C_3)\pi(C_1) = \pi(E)$ reprezentuje H .

Założenie indukcyjne: Niech $n \in \mathbb{N}$. Zakładamy, że dla wszystkich $k \in \{1, 2, \dots, n-1\}$, jeżeli $C_{h(k)}$ reprezentuje podheksastrukt $H_{h(k)}$ heksastruktu H , to $\pi(C_{h(k)})$ również reprezentuje $H_{h(k)}$. W szczególności zachodzi $\pi(C_{h(1)}) = \pi(C_{h(2)})(C_{h(2)+1})\pi(C_{h(2)-1})$, a rozszerzony ciąg binarny $\pi(C_{h(1)})$ reprezentuje heksastrukt H_2 .

Krok indukcyjny: Jeżeli $C_{h(2)-1}$ reprezentuje ścieżkę p_1 , to $\pi(C_{h(2)-1})$ reprezentuje $q_2 = p_1^{-1}$. Ponadto $\pi(C_{h(1)-1})$ reprezentuje ścieżkę $q_1 = p_1^{-1}$. Wówczas $\bar{q}_2 = \bar{q}_1 = \bar{p}_3$. Zatem $\pi(C_2)(C_3)\pi(C_1) = \pi(E)$ reprezentuje H . \square

Przykład 3.4.12. Na rysunkach 3.14 i 3.15 przedstawiliśmy heksastrukty reprezentowane przez rozszerzone ciągi binarne $01(0)11$, $\pi(01(0)11) = 1(11)01$ oraz $\pi^{(2)}(10(00)1) = 00(01)0$.

Ostatnim odwzorowaniem jest ζ , które polega na odwróceniu ciągu binarnego. Wyniki wynikają z definicji π oraz twierdzenia 2.4.4.

Twierdzenie 3.4.13. Jeżeli ciąg binarny B reprezentuje ścieżkę p , to ciąg binarny $\zeta(B)$ reprezentuje ścieżkę p' , która jest odbiciem lustrzanym ścieżki p .

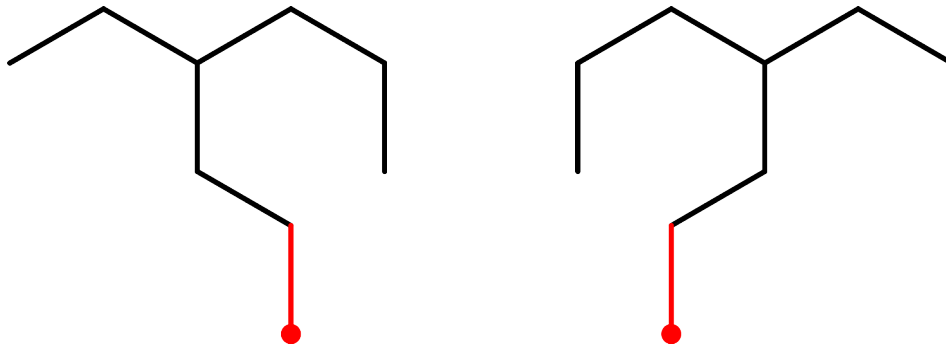
Przykład 3.4.14. Na rysunkach 3.12 i 3.13 przedstawiliśmy heksastrukty będące ścieżkami, reprezentowane przez ciągi binarne 01111 oraz $\zeta(01111) = 11110$.

Twierdzenie 3.4.15. Jeżeli rozszerzony ciąg binarny E reprezentuje heksastrukt H , to rozszerzony ciąg binarny $\zeta(E)$ reprezentuje heksastrukt będący odbiciem lustrzanym heksastruktu H .

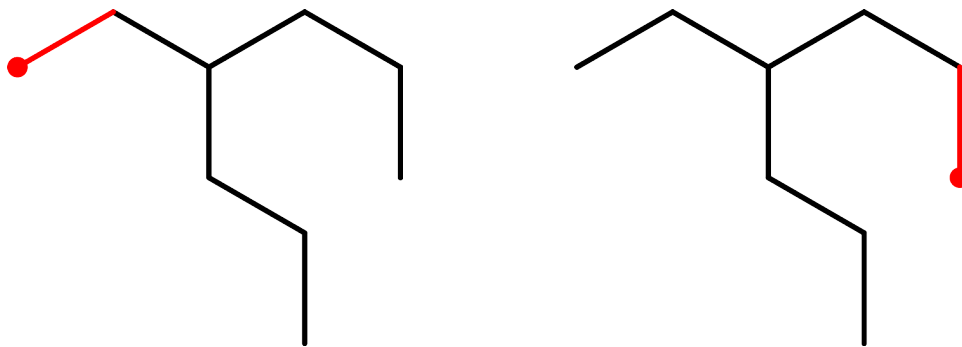
Przykład 3.4.16. Na rysunkach 3.14 i 3.16 przedstawiliśmy heksastrukty reprezentowane przez rozszerzone ciągi binarne $01(0)11$ oraz $\zeta(01(0)11) = 11(1)10$.

Podsumowując, jeżeli heksastrukt H jest reprezentowany przez rozszerzony ciąg binarny E , to $\pi(E)$ reprezentuje ten sam heksastrukt, lecz o pokryciu o początku w innym wierzchołku, $\tau(E)$ reprezentuje heksastrukt będący odbiciem lustrzanym heksastruktu H o tym samym początku co E oraz $\zeta(E)$ reprezentuje heksastrukt będący odbiciem lustrzanym heksastruktu H o innym początku niż E . Zauważmy na zakończenie rozdziału, że na mocy twierdzenia 2.4.10 istnieje maksymalna liczba reprezentacji heksastruktu H rozszerzonymi ciągami binarnymi pochodzącymi od rozszerzonego ciągu binarnego E .

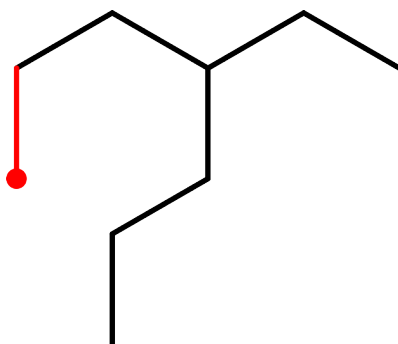
Wniosek 3.4.17. Jeżeli E reprezentuje heksastrukt H , to $\pi^{(n)}(E)$ również reprezentuje H , przy czym $n \in \{1, 2, \dots, \omega(E)\}$.



Rysunek 3.14: Heksastrukty reprezentowane przez rozszerzone ciągi binarne, odpowiednio od lewej do prawej, $01(0)11$ i $\tau(01(0)11) = 10(00)1$.



Rysunek 3.15: Heksastrukty reprezentowane przez rozszerzone ciągi binarne, odpowiednio od lewej do prawej, $1(11)01$ i $\pi(1(11)01) = 00(01)0$.



Rysunek 3.16: Heksastrukt reprezentowany przez rozszerzony ciąg binarny $\varsigma(01(0)11) = 11(1)10$.

Rozdział 4

Algorytmy agregujące heksastrukty

Ciągi binarne możemy porównywać, wykorzystując do tego m.in. metryki oraz miary odległości oraz częściowe porządki dla ciągów. W pierwszym przypadku będziemy rozważali liczbę działań niezbędnych do przekształcenia jednego ciągu w drugi, np. poprzez dodawanie oraz usuwanie elementów, zamianę elementów na inne oraz przestawianie elementów miejscami. Ponadto możemy preferować jedne różnice względem innych ze względu na pozycję bitów w ciągach, przy czym różnice na początku ciągów mają krytyczny wpływ na wygląd heksastruktów reprezentowanych przez rozważane ciągi binarne. W drugim przypadku popularne porządki bezpośrednio uwzględniają pozycję znaków w badanych ciągach i na tej podstawie porządkują je. W niniejszym rozdziale przedstawimy definicje popularnych metryk oraz porządków, zaproponujemy nowe metody uwzględniające wpływ zmiany bitów ciągu na różnice geometryczne heksastruktów oraz rozszerzymy stosowane metody porównywania rozszerzonych ciągów binarnych. Stanowi to odpowiedź na rozważania przedstawione w czwartym rozdziale pracy [16], w której profesor Marek Gągolewski podkreśla konieczność rozwijania metod agregacji grafów oraz innych typów danych.

4.1 Agregacja z wykorzystaniem metryk

Porównywanie obiektów za pomocą metryk stanowi powszechnie stosowaną metodę opartą na solidnych podstawach matematycznych (patrz [8] oraz [36]). W tym procesie wykorzystuje się odwzorowania dwuargumentowe spełniające trzy proste warunki. Na ich podstawie opracowano wiele metod agregacji, np. w klasyfikacji statystycznej (patrz [9] oraz [21]).

Przytaczamy definicję metryki dla dowolnego rodzaju obiektów. Niech X będzie niepustym zbiorem. Metryka $d: X^2 \rightarrow [0, +\infty)$ jest odwzorowaniem, które spełnia warunki:

- (i) identyczność nierozróżnialnych, tj. $d(a, b) = 0 \iff a = b$,
- (ii) symetrię, tj. $d(a, b) = d(b, a)$,
- (iii) nierówność trójkąta, tj. $d(a, c) \leq d(a, b) + d(b, c)$,

dla wszystkich $a, b, c \in X$.

Odległość między ciągami binarnymi oraz rozszerzonymi ciągami binarnymi nie wynika w oczywisty sposób z definicji metryki. Złożona postać rozszerzonych ciągów binarnych oraz sposób ich zdefiniowania w niniejszej pracy wymaga rozbudowania tego pojęcia oraz zbadania jego własności. Przyjmujemy, że d jest ustaloną dowolnie metryką na zbiorze ciągów binarnych.

Niech $E, E', E'' \in \mathcal{E}$ oraz niech $(C_i)_{i \in I}$, $(C'_i)_{i \in I'}$ i $(C''_i)_{i \in I''}$ będą rozwinięciami, odpowiednio, E , E' i E'' . Wówczas definiujemy rekurencyjnie odwzorowanie d_d^* , które jest rozszerzeniem miary odległości d dla ciągów binarnych na rozszerzone ciągi binarne.

$$d_d^*(E, E') = \begin{cases} d(E, E'), & E, E' \in \mathcal{B}, \\ \min\{d_d^*(E, C'_1 0 C'_2) + d_d^*(C'_1 1 C'_3, \emptyset), \\ \quad d_d^*(E, C'_1 1 C'_3) + d_d^*(C'_1 0 C'_2, \emptyset)\}, & E \in \mathcal{B}, E' \in \mathcal{E} \setminus \mathcal{B}, \\ d_d^*(E', E), & E \in \mathcal{E} \setminus \mathcal{B}, E' \in \mathcal{B}, \\ \min\{d_d^*(C_1 0 C_2, C'_1 0 C'_2) + d_d^*(C_1 1 C_3, C'_1 1 C'_3), \\ \quad d_d^*(C_1 0 C_2, C'_1 1 C'_3) + d_d^*(C_1 1 C_3, C'_1 0 C'_2)\}, & E, E' \in \mathcal{E} \setminus \mathcal{B}. \end{cases} \quad (4.1.1)$$

Pokażemy, że d_d^* jest metryką, dzieląc dowód na części, w których zbadamy poszczególne własności tego odwzorowania. Bezpośrednio z definicji otrzymujemy następujący rezultat:

Twierdzenie 4.1.1. *Odwzorowanie d_d^* jest symetryczne.*

Twierdzenie 4.1.2. *Odwzorowanie d_d^* jest nieujemne.*

Dowód. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na złożoność argumentów różnego typu:

- (i) $E, E' \in \mathcal{B}$,
- (ii) $E \in \mathcal{B}$ oraz $E' \in \mathcal{E} \setminus \mathcal{B}$,
- (iii) $E, E' \in \mathcal{E} \setminus \mathcal{B}$.

W przypadku (i) nieujemność wynika bezpośrednio z własności metryki d .

Przypadek bazowy (ii): Jeżeli $\omega(E') = 2$ oraz $(C'_i)_{i \in I'}$ jest rozwinięciem rozszerzonego ciągu binarnego E' , to $C'_2, C'_3 \in \mathcal{B}$, a stąd

$$d_d^*(E, C'_1 0 C'_2) + d_d^*(C'_1 1 C'_3, \emptyset) = d(E, C'_1 0 C'_2) + d(C'_1 1 C'_3, \emptyset) \geq 0 \quad (4.1.2)$$

oraz

$$d_d^*(E, C'_1 1 C'_3) + d_d^*(C'_1 0 C'_2, \emptyset) = d(E, C'_1 1 C'_3) + d(C'_1 0 C'_2, \emptyset) \geq 0. \quad (4.1.3)$$

Zatem $d_d^*(E, E') \geq 0$.

Założenie indukcyjne (ii): Niech $n \in \mathbb{N}$, niech $\omega(E') = n$ oraz niech $d_d^*(E, F')$ będzie nieujemne dla takich $F' \in (C'_i)_{i \in I'}$, że $\omega(F') < n$, tj. gdy $i \in I' \setminus \{0\}$.

Krok indukcyjny (ii): Na mocy założenia indukcyjnego spełnione są nierówności (4.1.2) oraz (4.1.3), a zatem $d_d^*(E, E') \geq 0$.

Przypadek bazowy (iii): Jeżeli $\omega(E) = \omega(E') = 2$, to $C_1, C_2, C_3, C'_1, C'_2, C'_3 \in \mathcal{B}$, a stąd

$$d_d^*(C_1 0 C_2, C'_1 0 C'_2) + d_d^*(C_1 1 C_3, C'_1 1 C'_3) = d(C_1 0 C_2, C'_1 0 C'_2) + d(C_1 1 C_3, C'_1 1 C'_3) \geq 0 \quad (4.1.4)$$

oraz

$$d_d^*(C_1 0 C_2, C'_1 1 C'_3) + d_d^*(C_1 1 C_3, C'_1 0 C'_2) = d(C_1 0 C_2, C'_1 1 C'_3) + d(C_1 1 C_3, C'_1 0 C'_2) \geq 0. \quad (4.1.5)$$

Ze względu na dwuargumentowość odwzorowania d_d^* oraz twierdzenie 4.1.1, dowód dzielimy na następujące przypadki:

(iii.a) $\omega(E) = 2$ i indukcja ze względu na $\omega(E') \in \mathbb{N}$,

(iii.b) $\omega(E) = n$ dla pewnego $n \in \mathbb{N}$ i indukcja ze względu na $\omega(E') \in \mathbb{N}$.

Założenie indukcyjne dla (iii.a): Niech $\omega(E) = 2$, niech $\omega(E') = n' \in \mathbb{N}$ oraz niech $d_d^*(C_1 0 C_2, F') \geq 0$ i $d_d^*(C_1 1 C_3, F') \geq 0$ dla każdego takiego $F' \in (C'_i)_{i \in I'}$, że $\omega(F') \in \{1, 2, \dots, n' - 1\}$, tj. gdy $i \in I' \setminus \{0\}$.

Krok indukcyjny dla (iii.a): Na mocy wniosku 2.2.7 oraz założenia indukcyjnego $d_d^*(C_1 0 C_2, C'_1 0 C'_2) \geq 0$, $d_d^*(C_1 0 C_2, C'_1 1 C'_3) \geq 0$, $d_d^*(C_1 1 C_3, C'_1 0 C'_2) \geq 0$ oraz $d_d^*(C_1 1 C_3, C'_1 1 C'_3) \geq 0$. Zatem spełnione są nierówności (4.1.4) oraz (4.1.5), a stąd $d_d^*(E, E') \geq 0$.

Założenie indukcyjne dla (iii.b): Niech $\omega(E) = n$ dla pewnego $n \in \mathbb{N}$, niech $\omega(E') = n' \in \mathbb{N}$ oraz niech $d_d^*(C_1 0 C_2, F') \geq 0$ i $d_d^*(C_1 1 C_3, F') \geq 0$ dla każdego takiego $F' \in (C'_i)_{i \in I'}$, że $\omega(F') \in \{1, 2, \dots, n' - 1\}$.

Krok indukcyjny dla (iii.b): Analogicznie do dowodu kroku indukcyjnego dla (iii.a), na mocy wniosku 2.2.7 oraz założenia indukcyjnego $d_d^*(C_1 0 C_2, C'_1 0 C'_2) \geq 0$, $d_d^*(C_1 0 C_2, C'_1 1 C'_3) \geq 0$, $d_d^*(C_1 1 C_3, C'_1 0 C'_2) \geq 0$ oraz $d_d^*(C_1 1 C_3, C'_1 1 C'_3) \geq 0$. Zatem spełnione są nierówności (4.1.4) oraz (4.1.5), a stąd $d_d^*(E, E') \geq 0$. \square

Twierdzenie 4.1.3. *Jeżeli E jest rozszerzonym ciągiem binarnym, to*

$$d_d^*(E, \emptyset) = d_d^*(C_1 0 C_2, \emptyset) + d_d^*(C_1 1 C_3, \emptyset). \quad (4.1.6)$$

Twierdzenie 4.1.4. *Jeżeli E jest rozszerzonym ciągiem binarnym, to $d_d^*(E, \emptyset) > 0$.*

Dowód. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na złożoność rozszerzonego ciągu binarnego E . Zauważmy, że gdy E jest rozszerzonym ciągiem binarnym, to na mocy twierdzenia 4.1.3 wzór (4.1.1) upraszcza się do postaci (4.1.6).

Przypadek bazowy: Jeżeli $\omega(E) = 2$, to $C_2, C_3 \in \mathcal{B}$. Wówczas $C_10C_2 \neq \emptyset$ oraz $C_11C_3 \neq \emptyset$, a stąd $d(C_10C_2, \emptyset) > 0$ oraz $d(C_11C_3, \emptyset) > 0$. Zatem $d_d^*(E, \emptyset) > 0$.

Założenie indukcyjne: Niech $\omega(E) = n$ dla pewnego $n \in \mathbb{N} \setminus \{1, 2\}$. Załóżmy, że $d_d^*(F, \emptyset) > 0$ dla każdego takiego $F \in \mathcal{E}$, że $\omega(F) < n$.

Krok indukcyjny: Z założenia indukcyjnego mamy $d_d^*(C_10C_2, \emptyset) > 0$ oraz $d_d^*(C_11C_3, \emptyset) > 0$. Teza wynika bezpośrednio z wykorzystania tych nierówności w twierdzeniu 4.1.3. \square

Uwaga 4.1.5. Niech $B \in \mathcal{B}$. Jeżeli $E' \in \mathcal{E} \setminus \mathcal{B}$, to $d_d^*(B, E') > 0$.

Uwaga 4.1.6. Niech $E, E' \in \mathcal{E}$. Jeżeli $\omega(E) \neq \omega(E')$, to $d_d^*(E, E') > 0$.

Dowód. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na złożoność ciągów E oraz E' . Na mocy symetrii d_d^* założmy bez straty ogólności, że $\omega(E) < \omega(E')$.

Przypadek bazowy: Jeżeli $\omega(E) = 1$, to teza wynika bezpośrednio z wniosku 4.1.5.

Założenie indukcyjne: Niech $\omega(E) = n$ oraz $\omega(E') = n'$ dla takich $n, n' \in \mathbb{N}$, że $n < n'$. Załóżmy, że $d_d^*(F, F') > 0$ dla takich $F, F' \in \mathcal{E}$, że $\omega(F) < n$, $\omega(F') < n'$ oraz $\omega(F) \neq \omega(F')$.

Krok indukcyjny: Na mocy założenia indukcyjnego istnieją takie $n_2, n_3, n'_2, n'_3 \in \mathbb{N}$, że $n = n_2 + n_3$, $n' = n'_2 + n'_3$, $\omega(C_2) = n_2$, $\omega(C_3) = n_3$, $\omega(C'_2) = n'_2$ oraz $\omega(C'_3) = n'_3$. Zauważmy, że spełniona jest jedna z nierówności $n_2 < n'_2$ oraz $n_3 < n'_3$ lub jedna z nierówności $n_2 < n'_3$ oraz $n_3 < n'_2$. Zatem na mocy założenia indukcyjnego spełniona jest jedna z nierówności $d_d^*(C_10C_2, C'_10C'_2) > 0$ oraz $d_d^*(C_11C_3, C'_11C'_3) > 0$ lub jedna z nierówności $d_d^*(C_10C_2, C'_11C'_3) > 0$ oraz $d_d^*(C_11C_3, C'_10C'_2) > 0$. Wówczas

$$d_d^*(C_10C_2, C'_10C'_2) + d_d^*(C_11C_3, C'_11C'_3) > 0$$

oraz

$$d_d^*(C_10C_2, C'_11C'_3) + d_d^*(C_11C_3, C'_10C'_2) > 0.$$

\square

Twierdzenie 4.1.7. Rozszerzone ciągi binarne E oraz E' są równe wtedy i tylko wtedy, gdy $d_d^*(E, E') = 0$.

Dowód. Przypomnijmy, że rozszerzone ciągi binarne E oraz E' są równe wtedy i tylko wtedy, gdy $C_10C_2 = C'_10C'_2$ oraz $C_11C_3 = C'_11C'_3$. Na mocy wniosku 2.2.14, jeżeli $E = E'$,

to $\omega(E) = \omega(E')$. Ponadto, na mocy wniosku 4.1.6, jeżeli $d_d^*(E, E') = 0$, to $\omega(E) = \omega(E')$. Zatem zakładamy, że $\omega(E) = \omega(E')$ w dowodach w obie strony. Dowód przeprowadzimy z wykorzystaniem indukcji strukturalnej ze względu na złożoność rozszerzonych ciągów binarnych E oraz E' .

Przypadek bazowy (\implies): Jeżeli $\omega(E) = 2 = \omega(E')$, to $C_1, C_2, C_3, C'_1, C'_2, C'_3 \in \mathcal{B}$. Stąd oraz z własności metryki d mamy:

$$d_d^*(C_1 0 C_2, C'_1 0 C'_2) + d_d^*(C_1 1 C_3, C'_1 1 C'_3) = d(C_1 0 C_2, C'_1 0 C'_2) + d(C_1 1 C_3, C'_1 1 C'_3) = 0. \quad (4.1.7)$$

Zatem $d_d^*(E, E') = 0$.

Założenie indukcyjne (\implies): Niech $\omega(E) = n = \omega(E')$ dla pewnego $n \in \mathbb{N}$. Załóżmy, że $d_d^*(F, F') = 0$ dla takich $F \in (C_i)_{i \in I}$ i $F' \in (C'_i)_{i \in I'}$, że $F = F'$, $\omega(F) < n$ oraz $\omega(F') < n$, tj. gdy $C_i = C'_i$ oraz $i \neq 0$.

Krok indukcyjny (\implies): Równość $d_d^*(E, E') = 0$ wynika bezpośrednio z równości (4.1.7) oraz z założenia indukcyjnego.

Przypadek bazowy (\impliedby): Jeżeli $\omega(E) = 2 = \omega(E')$, to, na mocy własności metryki d , mamy:

$$d(C_1 0 C_2, C'_1 0 C'_2) + d(C_1 1 C_3, C'_1 1 C'_3) = 0$$

wtedy i tylko wtedy, gdy $d(C_1 0 C_2, C'_1 0 C'_2) = 0$ oraz $d(C_1 1 C_3, C'_1 1 C'_3) = 0$. Stąd $C_1 0 C_2 = C'_1 0 C'_2$ oraz $C_1 1 C_3 = C'_1 1 C'_3$, a tym samym $E = E'$.

Założenie indukcyjne (\impliedby): Niech $\omega(E) = n = \omega(E')$ dla pewnego $n \in \mathbb{N}$. Załóżmy, że jeżeli $d_d^*(F, F') = 0$ dla takich $F, F' \in \mathcal{E}$, że $\omega(F) < n$ oraz $\omega(F') < n$, to $F = F'$.

Krok indukcyjny (\impliedby): Na mocy nieujemności odwzorowania d_d^* zachodzi, że $d_d^*(E, E') = 0$ wtedy i tylko wtedy, gdy $d_d^*(C_1 0 C_2, C'_1 0 C'_2) = 0$ oraz $d_d^*(C_1 1 C_3, C'_1 1 C'_3) = 0$. Na mocy założenia indukcyjnego, $C_1 0 C_2 = C'_1 0 C'_2$ oraz $C_1 1 C_3 = C'_1 1 C'_3$. Zatem $E = E'$. \square

Twierdzenie 4.1.8. *Odwzorowanie d_d^* jest metryką wtedy i tylko wtedy, gdy d jest metryką.*

Zauważmy jeszcze pewną własność tak zdefiniowanej metryki d_d^* , która wynika bezpośrednio z nierówności trójkąta oraz z twierdzenia 4.1.3, a mianowicie różnica między rozszerzonym ciągiem binarnym a pustym ciągiem binarnym nie jest mniejsza niż różnica między składowymi tego rozszerzonego ciągu binarnego.

Wniosek 4.1.9. *Jeżeli E jest rozszerzonym ciągiem binarnym, to $d_d^*(E, \emptyset) \geq d_d^*(C_1 0 C_2, C_1 1 C_3)$.*

Uwaga 4.1.10. *W podrozdziale B.5 prezentujemy implementację miary odległości dla ciągów binarnych oraz rozszerzonych ciągów binarnych.*

W kolejnych częściach będziemy badać konkretne metryki dla ciągów binarnych oraz ich wpływ na porównywanie heksastruktów.

4.1.1 Odległość Hamminga

Odległość Hamminga jest jedną z najprostszych metryk wykorzystywanych do porównywania ciągów znaków tej samej długości (patrz [8]). Informuje o liczbie pozycji, na których dane obiekty się różnią.

Niech $n \in \mathbb{N}$ i niech $B_0 = b_{0,0}b_{0,1} \dots b_{0,n-1}$ oraz $B_1 = b_{1,0}b_{1,1} \dots b_{1,n-1}$. Odległość Hamminga $d_H: \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbb{R}$ dana jest wzorem:

$$d_H(B_0, B_1) = |\{k \in \{0, 1, \dots, n-1\} : b_{0,k} \neq b_{1,k}\}|. \quad (4.1.8)$$

Ze względu na ograniczony alfabet rozważanych ciągów znaków (0 i 1) upraszczamy wzór (4.1.8) do następującej postaci:

Twierdzenie 4.1.11. *Jeżeli $n \in \mathbb{N}$, $B_0 = b_{0,0}b_{0,1} \dots b_{0,n-1}$ oraz $B_1 = b_{1,0}b_{1,1} \dots b_{1,n-1}$, to*

$$d_H(B_0, B_1) = \sum_{i=0}^{n-1} |b_{0,i} - b_{1,i}|.$$

Zbadamy użyteczność odległości Hamminga do porównywania heksastruktów reprezentowanych przez ciągi binarne oraz rozszerzone ciągi binarne. Najistotniejszym ograniczeniem tej metryki jest porównywanie ciągów tylko tej samej długości (por. przykłady 4.1.12 oraz 4.1.14). Zatem liczba potencjalnych zastosowań radykalnie się zmniejsza. Ponadto w przykładzie 4.1.12 pokazujemy, że ciągi binarne o różnicach na skrajnych bitach mogą być tak samo podobne, jak te o różnicach, które dzieli mniejsza liczba indeksów.

Przykład 4.1.12. *Rozważamy ciągi binarne 10000, 00100, 00001. Wówczas*

$$d_H(10000, 00100) = d_H(10000, 00001) = d_H(00100, 00001) = 2.$$

Zatem rozważane ciągi są tak samo różne od siebie.

Przykład 4.1.13. *Rozważmy ciągi binarne 1011001 oraz 0110111. Wówczas $d_H(1011001, 0110111) = 5$. Zatem należy zmienić bity na pięciu pozycjach, aby z ciągu binarnego 1011001 otrzymać ciąg binarny 0110111.*

Przykład 4.1.14. *Rozważamy rozszerzone ciągi binarne 00(01)11 oraz 10(10)10. Na mocy wzoru (4.1.1) mamy:*

$$\begin{aligned} d_{d_H}^*(00(01)11, 10(10)10) &= \min\{d_H(00001, 10010) + d_H(00111, 10110), d_H(00001, 10110) + d_H(00111, 10010)\} \\ &= \min\{3 + 2, 4 + 3\} \\ &= 5. \end{aligned}$$

4.1.2 Rozszerzona odległość Hamminga

Pierwszym krokiem zwiększania użyteczności miary Hamminga jest rozszerzenie jej wzoru na ciągi binarne o różnych długościach. Zaczniemy od ogólniejszej postaci, w której krótszy ciąg porównujemy z podciągiem dłuższego ciągu. Niech $n_0, n_1 \in \mathbb{N}_0$, $n =$

$\min\{n_0, n_1\}$, $N = \max\{n_0, n_1\}$, $p = \arg \min_{k \in \{0,1\}} n_k$ oraz

$$d_{H^*}^m(B_0, B_1) = \begin{cases} N, & n = 0, \\ N - n + \sum_{i=0}^{n-1} |b_{p,i} - b_{1-p,i+m}|, & n > 0. \end{cases}$$

dla $m \in \{0, 1, \dots, N - n\}$.

Twierdzenie 4.1.15. *Odwzorowanie $d_{H^*}^0$ jest metryką.*

Dowód. Niech B_0, B_1, B_2 będą ciągami binarnymi o długościach, odpowiednio, $n_0, n_1, n_2 \in \mathbb{N}$. Zauważmy, że symetria $d_{H^*}^0$ wynika bezpośrednio z symetrii funkcji \min , \max oraz wartości bezwzględnej różnicy.

Założmy, że $n_0 \leq n_1$. Jeżeli $n_0 = 0$, to $d_{H^*}^0(B_0, B_1) = 0$ wtedy i tylko wtedy, gdy $n_1 = 0$, czyli gdy oba ciągi są puste. Jeżeli $n_0 > 0$, to $d_{H^*}^0(B_0, B_1) = 0$ wtedy i tylko wtedy, gdy

$$n_1 - n_0 + \sum_{i=0}^{n_0-1} |b_{0,i} - b_{1,i}| = 0,$$

tj. $n_1 = n_0$ oraz $b_{0,i} = b_{1,i}$ dla $i \in \{0, 1, \dots, n_0 - 1\}$, tzn. $B_0 = B_1$.

Podzielimy dowód nierówności trójkąta na podproblemy ze względu na wzajemne relacje między n_0, n_1 oraz n_2 .

- Jeżeli $n_0 = n_1 = n_2 = 0$, to

$$d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) = 0 = d_{H^*}^0(B_0, B_2).$$

- Jeżeli $n_0 = n_1 = 0$ oraz $n_2 > 0$, to

$$d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) = 0 + n_2 = d_{H^*}^0(B_0, B_2).$$

- Jeżeli $n_0 = n_2 = 0$ oraz $n_1 > 0$, to

$$d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) = n_1 + n_1 \geq d_{H^*}^0(B_0, B_2).$$

- Jeżeli $n_1 = n_2 = 0$ oraz $n_0 > 0$, to

$$d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) = n_0 + 0 = d_{H^*}^0(B_0, B_2).$$

- Jeżeli $n_0 = 0$ oraz $0 < n_1 \leq n_2$, to

$$\begin{aligned} d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_1 + n_2 - n_1 + \sum_{i=0}^{n_1-1} |b_{1,i} - b_{2,i}| \\ &\geq n_2 \\ &= d_{H^*}^0(B_0, B_2). \end{aligned}$$

- Jeżeli $n_0 = 0$ oraz $0 < n_2 \leq n_1$, to

$$\begin{aligned} d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_1 + n_1 - n_2 + \sum_{i=0}^{n_2-1} |b_{1,i} - b_{2,i}| \\ &\geq n_2 \\ &= d_{H^*}^0(B_0, B_2). \end{aligned}$$

- Jeżeli $n_1 = 0$ oraz $0 < n_0 \leq n_2$, to

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_0 + n_2 \\
&\geq n_2 + \sum_{i=0}^{n_0-1} |b_{1,i} - b_{2,i}| \\
&\geq n_2 - n_0 + \sum_{i=0}^{n_0-1} |b_{1,i} - b_{2,i}| \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Jeżeli $n_1 = 0$ oraz $0 < n_2 \leq n_0$, to

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_0 + n_2 \\
&\geq n_0 + \sum_{i=0}^{n_2-1} |b_{1,i} - b_{2,i}| \\
&\geq n_0 - n_2 + \sum_{i=0}^{n_2-1} |b_{1,i} - b_{2,i}| \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Jeżeli $n_2 = 0$ oraz $0 < n_0 \leq n_1$, to

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_1 - n_0 + \sum_{i=0}^{n_0-1} |b_{1,i} - b_{2,i}| + n_1 \\
&\geq n_0 \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Jeżeli $n_2 = 0$ oraz $0 < n_1 \leq n_0$, to

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_0 - n_1 + \sum_{i=0}^{n_1-1} |b_{1,i} - b_{2,i}| + n_1 \\
&= n_0 + \sum_{i=0}^{n_1-1} |b_{1,i} - b_{2,i}| \\
&\geq n_0 \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Jeżeli $0 < n_0 \leq n_1 \leq n_2$, to mamy:

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_1 - n_0 + \sum_{i=0}^{n_0-1} |b_{0,i} - b_{1,i}| + n_2 - n_1 + \sum_{i=0}^{n_1-1} |b_{1,i} - b_{2,i}| \\
&\geq n_2 - n_0 + \sum_{i=0}^{n_0-1} |b_{0,i} - b_{2,i}| \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Ježeli $0 < n_0 \leq n_2 \leq n_1$, to mamy:

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_1 - n_0 + \sum_{i=0}^{n_0-1} |b_{0,i} - b_{1,i}| + n_1 - n_2 + \sum_{i=0}^{n_2-1} |b_{1,i} - b_{2,i}| \\
&\geq n_2 - n_0 + \sum_{i=0}^{n_0-1} |b_{0,i} - b_{2,i}| \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Ježeli $0 < n_1 \leq n_0 \leq n_2$, to mamy:

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_0 - n_1 + \sum_{i=0}^{n_1-1} |b_{0,i} - b_{1,i}| + n_2 - n_1 + \sum_{i=0}^{n_1-1} |b_{1,i} - b_{2,i}| \\
&\geq n_2 - n_1 + \sum_{i=n_1}^{n_0-1} 1 + \sum_{i=0}^{n_1-1} |b_{0,i} - b_{2,i}| \\
&\geq n_2 - n_0 + \sum_{i=0}^{n_0-1} |b_{0,i} - b_{2,i}| \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Ježeli $0 < n_1 \leq n_2 \leq n_0$, to mamy:

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_0 - n_1 + \sum_{i=0}^{n_1-1} |b_{0,i} - b_{1,i}| + n_2 - n_1 + \sum_{i=0}^{n_1-1} |b_{1,i} - b_{2,i}| \\
&\geq n_0 - n_1 + \sum_{i=n_1}^{n_2-1} 1 + \sum_{i=0}^{n_1-1} |b_{0,i} - b_{2,i}| \\
&\geq n_0 - n_2 + \sum_{i=0}^{n_2-1} |b_{0,i} - b_{2,i}| \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Ježeli $0 < n_2 \leq n_0 \leq n_1$, to mamy:

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_1 - n_0 + \sum_{i=0}^{n_0-1} |b_{0,i} - b_{1,i}| + n_1 - n_2 + \sum_{i=0}^{n_2-1} |b_{1,i} - b_{2,i}| \\
&\geq n_0 - n_2 + \sum_{i=0}^{n_2-1} |b_{0,i} - b_{2,i}| \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

- Ježeli $0 < n_2 \leq n_1 \leq n_0$, to mamy:

$$\begin{aligned}
d_{H^*}^0(B_0, B_1) + d_{H^*}^0(B_1, B_2) &= n_0 - n_1 + \sum_{i=0}^{n_1-1} |b_{0,i} - b_{1,i}| + n_1 - n_2 + \sum_{i=0}^{n_2-1} |b_{1,i} - b_{2,i}| \\
&\geq n_0 - n_2 + \sum_{i=0}^{n_2-1} |b_{0,i} - b_{2,i}| \\
&= d_{H^*}^0(B_0, B_2).
\end{aligned}$$

□

Przykład 4.1.16. Rozważmy ciągi binarne 000, 1000 oraz 01000. Wówczas

$$d_{H^*}^1(000, 1000) = 1,$$

$$d_{H^*}^1(1000, 01000) = 1$$

oraz

$$d_{H^*}^1(000, 01000) = 3.$$

Stąd

$$d_{H^*}^1(000, 1000) + d_{H^*}^1(1000, 01000) < d_{H^*}^1(000, 01000).$$

Zatem odwzorowanie $d_{H^*}^j$ nie musi spełniać warunku trójkąta dla $j \neq 0$, a tym samym nie jest metryką.

Twierdzenie 4.1.15 oraz przykład 4.1.16 pokazują, że przy takim sposobie rozszerzenia odległości Hamminga jedynym prawidłowym kierunkiem zdefiniowania odwzorowania mierzącego odległości między ciągami binarnymi różnych długości, które jest metryką, jest porównywanie kolejnych bitów obu ciągów, poczynając od pierwszych indeksów. Zatem **rozszerzoną odległością Hamminga** nazywamy odwzorowanie $d_{H^*} = d_{H^*}^0$.

Twierdzenie 4.1.17. Jeżeli B_0 i B_1 są niepustymi ciągami binarnymi o długościach, odpowiednio, n_0 i n_1 , oraz $n = \min\{n_0, n_1\}$ i $N = \max\{n_0, n_1\}$, to

$$d_{H^*}(B_0, B_1) = \sum_{0 \leq i < n} |b_{0,i} - b_{1,i}| + \sum_{n \leq i < N} 1.$$

Twierdzenie 4.1.17 prezentujemy w celu ułatwienia jego porównania z metryką przedstawioną w kolejnym podrozdziale.

4.1.3 Rozszerzona ważona odległość Hamminga

Kontynuujemy proces rozszerzania metryki Hamminga. Naszym kolejnym krokiem jest nadanie większej istotności różnicom między ciągami binarnymi na wcześniejszych indeksach. Niech $n_0, n_1 \in \mathbb{N}$ oraz niech $B_0 = b_{0,0}b_{0,1} \dots b_{0,n_0-1}$ i $B_1 = b_{1,0}b_{1,1} \dots b_{1,n_1-1}$ będą ciągami binarnymi oraz niech $w: \mathbb{N}_0 \rightarrow \mathbb{R}_+$. Rozważamy odwzorowanie d_{wH^*} dane wzorem:

$$d_{wH^*}(B_0, B_1) = \begin{cases} 0, & N = 0, \\ \sum_{0 \leq i < N} w(i), & n = 0, \\ \sum_{0 \leq i < n} w(i) \cdot |b_{0,i} - b_{1,i}| + \sum_{n \leq i < N} w(i), & n > 0. \end{cases} \quad (4.1.9)$$

Twierdzenie 4.1.18. Niech B_0 i B_1 będą ciągami binarnymi o długościach, odpowiednio, $n_0, n_1 \in \mathbb{N}$, niech $n = \min\{n_0, n_1\}$ i $N = \max\{n_0, n_1\}$ oraz niech $w: \mathbb{N} \rightarrow \mathbb{R}_+$. Wówczas odwzorowanie d_{wH^*} dane wzorem (4.1.9) jest metryką.

Dowód. Nieujemność oraz symetria odwzorowania d_{wH^*} wynika bezpośrednio z nieujemności i symetrii wartości bezwzględnej oraz z nieujemności funkcji w . Niech $n_0, n_1, n_2 \in \mathbb{N}$ oraz niech $B_0 = b_{0,0}b_{0,1} \dots b_{0,n_0-1}$, $B_1 = b_{1,0}b_{1,1} \dots b_{1,n_1-1}$ i $B_2 = b_{2,0}b_{2,1} \dots b_{2,n_2-1}$ będą ciągami binarnymi. Sprawdzamy warunek trójkąta dla wszystkich możliwych zależności między n_1 , n_2 i n_3 .

- Jeżeli $n_0 = n_1 = n_2 = 0$, to

$$d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) = 0 = d_{wH^*}(B_0, B_2).$$

- Jeżeli $n_0 = n_1 = 0$ oraz $n_2 > 0$ (lub $n_1 = n_2 = 0$ oraz $n_0 > 0$ z wykorzystaniem symetrii), to

$$d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) = 0 + \sum_{0 \leq i < n_2} w(i) = d_{wH^*}(B_0, B_2).$$

- Jeżeli $n_0 = n_2 = 0$ oraz $n_1 > 0$, to

$$\begin{aligned} d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) &= \sum_{0 \leq i < n_1} w(i) + \sum_{0 \leq i < n_1} w(i) \\ &\geq 0 \\ &= d_{wH^*}(B_0, B_2). \end{aligned}$$

- Jeżeli $n_0 = 0$ oraz $0 < n_1 \leq n_2$ (lub $n_2 = 0$ oraz $0 < n_0 \leq n_1$), to

$$\begin{aligned} d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) &= \sum_{0 \leq i < n_1} w(i) + \sum_{0 \leq i < n_1} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_1 \leq i < n_2} w(i) \\ &\geq \sum_{0 \leq i < n_2} w(i) \\ &= d_{wH^*}(B_0, B_2). \end{aligned}$$

- Jeżeli $n_0 = 0$ oraz $0 < n_2 \leq n_1$ (lub $n_2 = 0$ oraz $0 < n_1 \leq n_0$), to

$$\begin{aligned} d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) &= \sum_{0 \leq i < n_1} w(i) + \sum_{0 \leq i < n_2} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_2 \leq i < n_1} w(i) \\ &\geq \sum_{0 \leq i < n_1} w(i) \\ &\geq \sum_{0 \leq i < n_2} w(i) \\ &= d_{wH^*}(B_0, B_2). \end{aligned}$$

- Jeżeli $n_1 = 0$ oraz $0 < n_0 \leq n_2$, to

$$\begin{aligned} d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) &= \sum_{0 \leq i < n_0} w(i) + \sum_{0 \leq i < n_2} w(i) \\ &\geq \sum_{0 \leq i < n_2} w(i) \\ &\geq \sum_{0 \leq i < n_0} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_0 \leq i < n_2} w(i) \\ &= d_{wH^*}(B_0, B_2). \end{aligned}$$

- Jeżeli $n_1 = 0$ oraz $0 < n_2 \leq n_0$, to

$$\begin{aligned} d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) &= \sum_{0 \leq i < n_2} w(i) + \sum_{0 \leq i < n_0} w(i) \\ &\geq \sum_{0 \leq i < n_0} w(i) \\ &\geq \sum_{0 \leq i < n_2} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_2 \leq i < n_0} w(i) \\ &= d_{wH^*}(B_0, B_2). \end{aligned}$$

- Jeżeli $n_0 \leq n_1 \leq n_2$ (lub $n_2 \leq n_1 \leq n_0$ z wykorzystaniem symetrii), to

$$\begin{aligned}
& d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) = \\
&= \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{1,i}| + \sum_{n_0 \leq i < n_1} w(i) + \sum_{0 \leq i < n_1} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_1 \leq i < n_2} w(i) \\
&= \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{1,i}| + \sum_{0 \leq i < n_1} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_0 \leq i < n_2} w(i) \\
&\geq \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{1,i}| + \sum_{0 \leq i < n_0} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_0 \leq i < n_2} w(i) \\
&\geq \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{2,i}| + \sum_{n_0 \leq i < n_2} w(i) \\
&= d_{wH^*}(B_0, B_2).
\end{aligned}$$

- Jeżeli $n_0 \leq n_2 \leq n_1$ (lub $n_2 \leq n_0 \leq n_1$ z wykorzystaniem symetrii), to

$$\begin{aligned}
& d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) = \\
&= \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{1,i}| + \sum_{n_0 \leq i < n_1} w(i) + \sum_{0 \leq i < n_2} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_2 < i < n_1} w(i) \\
&\geq \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{1,i}| + \sum_{n_0 \leq i < n_1} w(i) + \sum_{0 \leq i < n_0} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_2 < i < n_1} w(i) \\
&\geq \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{2,i}| + \sum_{n_0 \leq i < n_1} w(i) + \sum_{n_2 \leq i < n_1} w(i) \\
&\geq \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{2,i}| + \sum_{n_0 \leq i < n_2} w(i) + \sum_{n_2 \leq i < n_1} w(i) \\
&= \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{2,i}| + \sum_{n_0 \leq i < n_2} w(i) \\
&= d_{wH^*}(B_0, B_2).
\end{aligned}$$

- Jeżeli $n_1 \leq n_0 \leq n_2$ (lub $n_1 \leq n_2 \leq n_0$ z wykorzystaniem symetrii), to

$$\begin{aligned}
& d_{wH^*}(B_0, B_1) + d_{wH^*}(B_1, B_2) = \\
&= \sum_{0 \leq i < n_1} w(i) \cdot |b_{0,i} - b_{1,i}| + \sum_{n_1 \leq i < n_0} w(i) + \sum_{0 \leq i < n_1} w(i) \cdot |b_{1,i} - b_{2,i}| + \sum_{n_1 \leq i < n_2} w(i) \\
&\geq \sum_{0 \leq i < n_1} w(i) \cdot |b_{0,i} - b_{2,i}| + \sum_{n_1 \leq i < n_0} w(i) + \sum_{n_1 \leq i < n_2} w(i) \\
&\geq \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{2,i}| + \sum_{n_1 \leq i < n_2} w(i) \\
&\geq \sum_{0 \leq i < n_0} w(i) \cdot |b_{0,i} - b_{2,i}| + \sum_{n_0 \leq i < n_2} w(i) \\
&= d_{wH^*}(B_0, B_2).
\end{aligned}$$

□

Uwaga 4.1.19. Jeżeli $w(i) = 1$ dla wszystkich $i \in \mathbb{N}_0$, to $d_{wH^*} = d_{H^*}$ (por. twierdzenie 4.1.17).

W niniejszej pracy rozważamy wagi będące odwrotnościami kolejnych potęg dwójki. Odwzorowanie d_{wH^*} dane wzorem (4.1.9), gdzie $w(i) = 2^{-i}$, nazywamy **rozszerzoną wagą odległością Hamminga**.

Przykład 4.1.20. Rozważamy rozszerzone ciągi binarne 1(111)11 oraz 01(0101)010101. Wówczas dla rozszerzonej ważonej odległości Hamminga mamy:

$$d_{wH^*}(10111, 0100101) = 1.92,$$

$$d_{wH^*}(1111, 011010101) = 1.25,$$

$$d_{wH^*}(10111, 011010101) = 1.68,$$

$$d_{wH^*}(1111, 0100101) = 1.48.$$

Stąd

$$d_{d_{wH^*}}^*(1(111)11, 01(0101)010101) = 2.16.$$

4.1.4 Odległość Levenshteina

Oprócz odległości Hamminga rozważamy również inną popularną metrykę. **Odległość Levenshteina** polega na przekształceniu jednego ciągu w drugi poprzez wstawianie, usuwanie oraz zamianę bitów (patrz [4]). Liczbę takich operacji zwraca odwzorowanie d_L , które jest dane wzorem rekurencyjnym:

$$d_L(B_0, B_1) = \begin{cases} n_1, & n_0 = 0, \\ n_0, & n_1 = 0, \\ d_L(B'_0, B'_1), & b_{0,0} = b_{1,0}, \\ 1 + \min \{d_L(B'_0, B_1), d_L(B_0, B'_1), d_L(B'_0, B'_1)\}, & \text{w pp.}, \end{cases} \quad (4.1.10)$$

gdzie $B'_0 = b_{0,1}b_{0,2} \dots b_{0,n_0-1}$ i $B'_1 = b_{1,1}b_{1,2} \dots b_{1,n_1-1}$.

Przykład 4.1.21. Rozważamy ciągi binarne 01 oraz 110. Obliczamy $d_L(01, 110)$ zgodnie ze wzorem (4.1.10). Mamy:

$$\begin{aligned} d_L(0, 1) &= 1 + \min \{d_L(\emptyset, 1), d_L(0, \emptyset), d_L(\emptyset, \emptyset)\} \\ &= 1 + \min \{1, 1, 0\} \\ &= 1 \end{aligned}$$

oraz:

$$\begin{aligned} d_L(01, 10) &= 1 + \min \{d_L(1, 10), d_L(01, 0), d_L(1, 0)\} \\ &= 1 + \min \{1, 1, 1\} \\ &= 2. \end{aligned}$$

Stąd

$$\begin{aligned} d_L(01, 110) &= 1 + \min \{d_L(1, 110), d_L(01, 10), d_L(1, 10)\} \\ &= 1 + \min \{2, 2, 1\} \\ &= 2. \end{aligned}$$

Przykład 4.1.22. *Rozważamy rozszerzone ciągi binarne 1(111)11 oraz 01(0101)010101. Wówczas*

$$d_L(10111, 0100101) = 3,$$

$$d_L(1111, 011010101) = 4,$$

$$d_L(10111, 011010101) = 4,$$

$$d_L(1111, 0100101) = 5.$$

Stąd

$$d_{d_L}^*(000(1100)111, 10(10)1010) = 7.$$

4.1.5 Medoid

Medoid jest obiektem matematycznym powszechnie wykorzystywanym, między innymi w analizie skupień (patrz [25]). Niech \mathfrak{C} będzie zbiorem wszystkich ciągów binarnych oraz rozszerzonych ciągów binarnych, niech $d: \mathfrak{C} \times \mathfrak{C} \rightarrow [0, +\infty)$ będzie metryką oraz niech \mathcal{E} będzie skończonym podzbiorem zbioru \mathfrak{C} . Element $E \in \mathcal{E}$, dla którego suma odległości od wszystkich elementów zbioru \mathcal{E} jest możliwie najmniejsza, tj. dany wzorem:

$$E = \arg \min_{E' \in \mathcal{E}} \sum_{F \in \mathcal{E}} d(E', F),$$

nazywamy medoidem zbioru \mathcal{E} .

Przykład 4.1.23. *Rozważamy metrykę Hamminga d_H oraz ciągi binarne $\{00, 01, 10\}$. Wówczas*

$$d_H(00, 01) = 1,$$

$$d_H(00, 10) = 1$$

oraz

$$d_H(01, 10) = 2.$$

Element 00 jest medoidem dla tego zbioru ciągów binarnych.

Następująca obserwacja jest kluczowa dla wykorzystania medoidu w analizie zbioru ciągów reprezentujących heksastrukty. Ponieważ medoid jest wybierany spośród elementów tego zbioru jako ciąg, dla którego suma odległości od pozostałych ciągów jest najmniejsza, gwarantuje to, że w wyniku agregacji otrzymamy również ciąg reprezentujący heksastrukt. Dzięki temu nie zachodzi obawa, czy medoid zachowuje właściwości heksastruktu, ponieważ jest to zapewnione przez konstrukcję metody.

Wniosek 4.1.24. *Jeżeli M jest medoidem dla zbioru ciągów binarnych oraz rozszerzonych ciągów binarnych reprezentujących heksastrukty, to M również reprezentuje heksastrukt.*

Zauważmy, że medoid nie zawsze jest wyznaczony jednoznacznie. W takich przypadkach, aby uzyskać dokładnie jednego reprezentanta zbioru, konieczne jest zastosowanie alternatywnej metryki lub dodatkowej metody agregacji dla otrzymanych kandydatów na medoid.

Przykład 4.1.25. *Rozważamy metrykę Hamminga d_H oraz ciągi binarne 0 i 1. Wówczas każdy z tych ciągów może być medoidem zbioru $\{0, 1\}$.*

Ogólniej możemy wyciągnąć następujący wniosek na podstawie przykładu 4.1.25:

Wniosek 4.1.26. *Jeżeli agregowany zbiór ciągów jest dwuelementowy, to każdy jego element jest medoidem.*

Przypadek opisany we wniosku 4.1.26 nie obejmuje wszystkich sytuacji, w których medoid jest wyznaczony niejednoznacznie.

Przykład 4.1.27. *Rozważamy metrykę Hamminga d_H oraz ciągi binarne 001, 010 i 100. Wówczas*

$$d_H(001, 010) = 2,$$

$$d_H(001, 100) = 2$$

oraz

$$d_H(010, 100) = 2.$$

Zatem każdy z tych ciągów może być medoidem.

4.2 Agregacja z wykorzystaniem porządków

Porządki częściowe dostarczają kolejnych metod agregacji heksastruktów (patrz [20] oraz [37]). Wykorzystanie porządków umożliwi bardziej intuicyjną interpretację oraz klasyfikację wyników w porównaniu z metrykami, a tym samym pozwala na głębsze zrozumienie zależności strukturalnych. W przypadku ciągów binarnych mamy do dyspozycji wiele istniejących porządków dla ciągów liczbowych oraz tekstowych, np. porządek leksykograficzny. Wówczas celem agregacji możemy wykorzystać ograniczenie dolne i górne oraz infimum i supremum rozważanego zbioru obiektów z wykorzystaniem porządków (por. podrozdział 1.5).

Porządkowanie ciągów jest bardziej intuicyjne, a tym samym prostsze ze względu na powszechne występowanie porządków oraz ich wykorzystanie w wielu dziedzinach nauki. Tym samym porządkowanie heksastruktów reprezentowanych przez ciągi binarne nie stanowi problemu. Bardziej interesującymi przypadkami są porównywanie ciągów binarnych z rozszerzonymi ciągami binarnymi oraz porównywanie samych rozszerzonych ciągów binarnych. Niech B będzie ciągiem binarnym, niech $E = B_0(E_1)E_2$ będzie rozszerzonym ciągiem binarnym oraz niech \leq będzie porządkiem na zbiorze ciągów binarnych. Rozszerzamy porządek \leq na ciągi binarne oraz rozszerzone ciągi binarne. Powiemy, że B jest nie większy niż E , gdy $B \leq B_0E_1$ lub $B \leq B_01E_2$, co oznaczamy $B \leq E$. Intuicyjnie ciąg binarny jest nie większy od rozszerzonego ciągu binarnego, gdy jest nie większy od jednego z ciągów binarnych zawartych w rozszerzonym ciągu binarnym.

Przykład 4.2.1. *Niech $B = 010101$ i niech $E = 01(01)01$. Wówczas $B \leq E$, gdy $010101 \leq 01001$ lub $010101 \leq 01101$.*

Zastosowanie analogicznego podejścia do porównywania ciągów binarnych oraz rozszerzonych ciągów binarnych nie jest możliwe. Stwierdzenie porządku na podstawie tylko jednego podciągu rozszerzonego ciągu binarnego doprowadzi do sytuacji, w której cała rodzina złożonych rozszerzonych ciągów binarnych będzie ograniczona przez prosty ciąg binarny.

Przykład 4.2.2. Niech $B = 001$ i niech $E = 0(0)0$. Jeżeli $B_00E_1 \leq B$, tj. $000 \leq 001$, i stwierdzamy, że $E \leq B$, to $B_0(E_1)E'_2 \leq B$ dla każdego $E'_2 \in \mathcal{E}$. Wówczas zarówno $0(0)11 \leq 001$, jak i $0(0)01010101010101 \leq 001$.

Stąd porównywanie ciągów binarnych oraz rozszerzonych ciągów binarnych w tę stronę silnie zależy od kontekstu oraz charakterystyki problemu, w jakim metoda ta ma zostać wykorzystana. Tym samym nie proponujemy ogólnej metody porównywania rozszerzonych ciągów binarnych z wykorzystaniem porządków, a jedynie jeden z wariantów, który zapewni spójność z porównywaniem ciągów binarnych oraz rozszerzonych ciągów binarnych.

Niech $E = B_0(E_1)E_2$ i $E' = B'_0(E'_1)E'_2$ będą rozszerzonymi ciągami binarnymi. Powiemy, że E jest nie większy niż E' , gdy $B_00E_1 \leq B'_00E'_1$ oraz $B_01E_2 \leq B'_01E'_2$, co oznaczamy $E \leq E'$.

Przykład 4.2.3. Niech $E = 0(0)0$ i $E' = 1(1)1$. Wówczas $E \leq E'$, gdy $000 \leq 101$ oraz $010 \leq 111$. Podobnie $E' \leq E$, gdy $101 \leq 000$ oraz $111 \leq 010$.

Poniżej prezentujemy ograniczenia tak zdefiniowanego porządku dla rozszerzonych ciągów binarnych.

Przykład 4.2.4. Niech $B_1, B_2, B_3, B'_1, B'_2$ oraz B'_3 będą różnymi ciągami binarnymi takimi, że $B_10B_2 \leq B'_10B'_2$ oraz $B'_11B'_3 \leq B_11B_3$ dla pewnego porządku \leq na zbiorze wszystkich ciągów binarnych. Wówczas ani $E \leq E'$, ani $E' \leq E$, gdzie $E = B_1(B_2)B_3$ oraz $E' = B'_1(B'_2)B'_3$.

Wniosek 4.2.5. Tak zdefiniowany porządek \leq na zbiorze ciągów binarnych oraz rozszerzonych ciągów binarnych nie jest porządkiem liniowym (por. rys. 4.5).

Wniosek 4.2.6. Jeżeli rozszerzone ciągi binarne są różnej złożoności, to są nieporównywalne względem tak zdefiniowanego porządku \leq .

Mimo wielu niedogodności związanych z porównywaniem rozszerzonych ciągów binarnych przy użyciu porządków podejście to nadal uważamy za perspektywiczne i obiecujące. Wykorzystanie porządków, takich jak porządek epaleksykograficzny, pozwala na uchwycenie istotnych różnic geometrycznych oraz topologicznych między heksastruktami, co może prowadzić do ciekawych i wartościowych wyników badań. Choć wiąże się to z większą złożonością analizy oraz wzrostem liczby nieporównywalnych par struktur, potencjalne korzyści płynące z głębszej i bardziej precyzyjnej klasyfikacji struktur czynią tę metodę atrakcyjną do dalszego rozwijania oraz zastosowania w praktyce.

4.2.1 Porządek leksykograficzny

Porządek leksykograficzny, powszechnie stosowany do sortowania danych tekstowych, stanowi efektywną i intuicyjną metodę organizacji informacji. Sortowanie odbywa się na podstawie porównań leksykalnych między poszczególnymi elementami ciągów. Prostota tego podejścia wynika z zastosowania jednoznacznych reguł porównywania znaków, co umożliwia łatwą implementację algorytmów sortowania.

Niech $B = b_0b_1 \dots b_{n-1}$ i $B' = b'_0b'_1 \dots b'_{n'-1}$ będą ciągami binarnymi o długościach, odpowiednio, $n, n' \in \mathbb{N}_0$. Niech $m = \min\{n, n'\}$. Porządek leksykograficzny \leq_L definiujemy w następujący sposób: ciąg binarny B jest nie większy niż ciąg binarny B' , co oznaczamy $B \leq_L B'$, gdy istnieje taki indeks $i \in \{0, 1, \dots, m-1\}$, że $b_i < b'_i$ oraz $b_j = b'_j$ dla wszystkich $j \in \{0, 1, \dots, i-1\}$ lub gdy $n \leq n'$ i $b_j = b'_j$ dla wszystkich $j \in \{0, 1, \dots, n-1\}$.

Przykład 4.2.7. *Rozważamy ciągi binarne 0, 1, 00 oraz 01. Wówczas*

$$0 \leq_L 00 \leq_L 01 \leq_L 1.$$

Twierdzenie 4.2.8. *Porządek leksykograficzny jest porządkiem liniowym na zbiorze ciągów binarnych.*

Na mocy twierdzeń 1.5.4 oraz 4.2.8 liczba metod agregacji dla porządku leksykograficznego z uwzględnieniem ciągów binarnych ogranicza się do infimum i supremum. Równocześnie wynikiem agregacji będzie jeden z elementów rozważanego zbioru.

Przykład 4.2.9. *Rozważamy porządek leksykograficzny \leq_L oraz zbiór ciągów binarnych $\{00, 01, 10\}$. Wówczas 00 jest infimum dla tego zbioru oraz 10 jest supremum dla tego zbioru.*

Interpretację oraz praktyczny potencjał porządku leksykograficznego w agregacji heksastruktów przedstawimy na poniższych przykładach.

Przykład 4.2.10. *Rozważamy zbiór wszystkich trzejelementowych ciągów binarnych wraz z odpowiadającymi im heksastruktami (por. rys. 4.1). Łatwo zauważyć, że im mniejsza jest struktura względem porządku leksykograficznego, tym bardziej jest skierowana w lewo względem krawędzi początkowej. W ogólnym przypadku porządek leksykograficzny pozwala posortować rozważane struktury od najkrótszych do najdłuższych, a następnie priorytetyzuje zakręty w lewo nad zakrętami w prawo (por. rysunki 4.1 oraz 4.2).*

Wniosek 4.2.11. *Ciąg binarny 0 jest najmniejszy w zbiorze ciągów binarnych reprezentujących heksastruktury względem porządku leksykograficznego.*

Wniosek 4.2.12. *Ciąg binarny 1111101111 jest największy w zbiorze ciągów binarnych reprezentujących heksastruktury względem porządku leksykograficznego.*

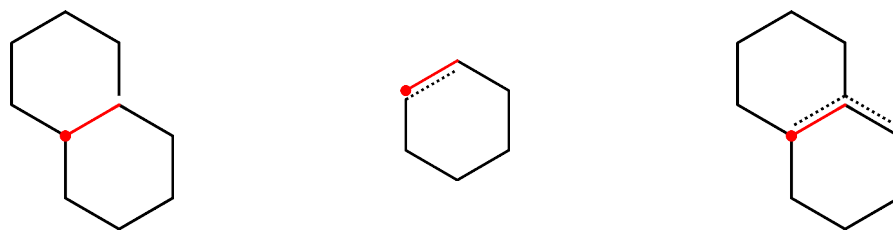
Dowód. Ciągi binarne $B_0 = 111111$, $B_1 = 11111011110$ oraz $B_2 = 11111011111$ są większe od 1111101111, ale nie reprezentują żadnych heksastruktów (por. rys. 4.3). Większe ciągi binarne możemy otrzymać poprzez dokładanie kolejnych bitów na końcach ciągów binarnych B_0 , B_1 oraz B_2 , jednak nie będą one reprezentowały żadnych heksastruktów, ponieważ B_0 , B_1 oraz B_2 nie reprezentują heksastruktów. \square

| | | | |
|-----|-----|-----|-----|
| | | | |
| 000 | 001 | 010 | 011 |
| | | | |
| 100 | 101 | 110 | 111 |

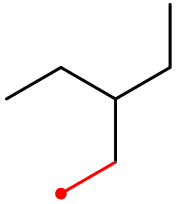
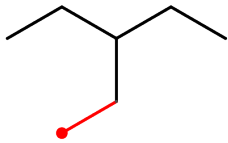
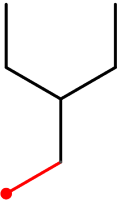
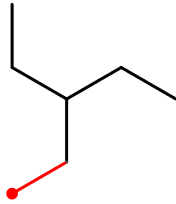
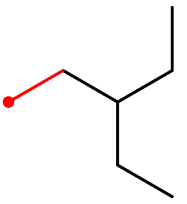
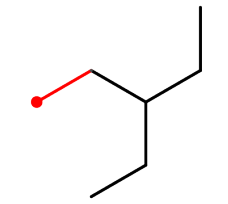
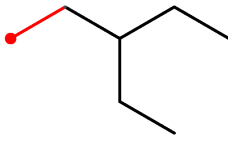
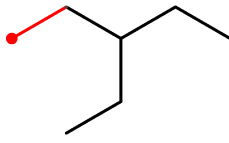
Rysunek 4.1: Struktury reprezentowane przez trzelementowe ciągi binarne, uporządkowane leksykograficznie wierszami od lewej górnej struktury do prawej dolnej struktury. Na czerwono zaznaczono krawędzie początkowe rozważanych ciągów binarnych.

| | | | |
|------|-----|-----|------|
| | | | |
| 0 | 00 | 000 | 0000 |
| | | | |
| 0001 | 001 | 01 | 1 |

Rysunek 4.2: Struktury reprezentowane przez ciągi binarne 0, 00, 000, 0000, 0001, 001, 01 oraz 1, uporządkowane leksykograficznie wierszami od lewej górnej struktury do prawej dolnej struktury. Na czerwono zaznaczono krawędzie początkowe rozważanych ciągów binarnych.



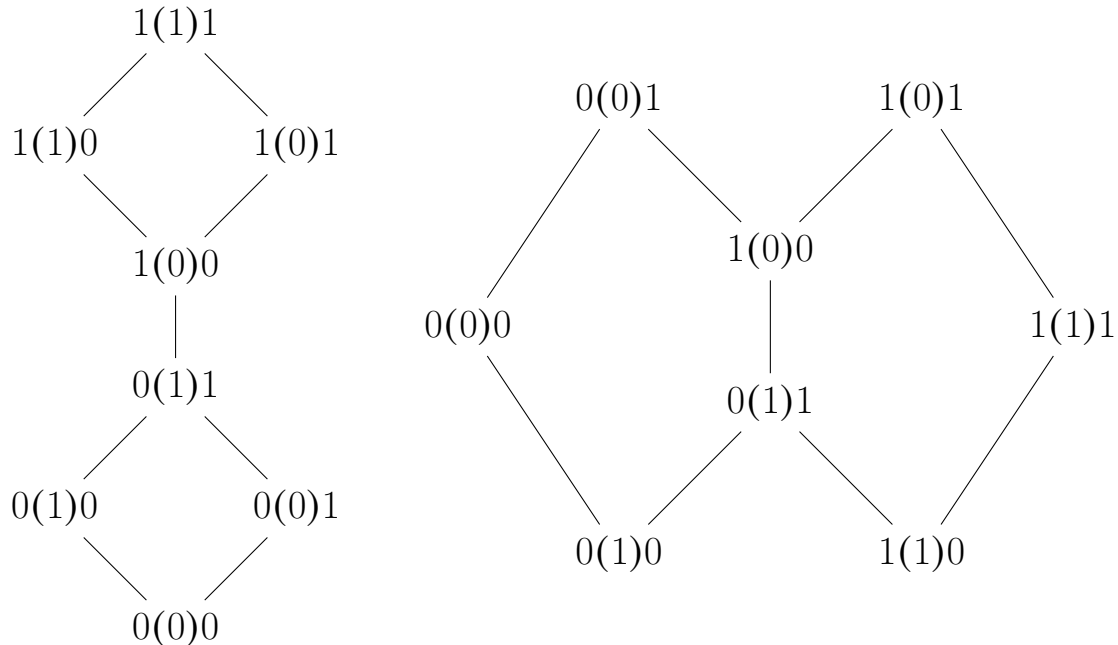
Rysunek 4.3: Po lewej stronie przedstawiono heksastrukt reprezentowany przez ciąg binarny 1111101111. Po prawej stronie pokazano nieprawidłowe heksastruktury powstałe z heksagonu przedstawionego po lewej stronie poprzez dodanie krawędzi lewostronnej lub krawędzi prawostronnej na końcu. Pośrodku przedstawiono nieprawidłowy heksastrukt, w którym ostatnia krawędź pokrywa się z pierwszą krawędzią, a jego reprezentacją binarną mógłby być ciąg 111111.

| | | | |
|---|---|--|---|
|  |  |  |  |
| 0(0)0 | 0(0)1 | 0(1)0 | 0(1)1 |
|  |  |  |  |
| 1(0)0 | 1(0)1 | 1(1)0 | 1(1)1 |

Rysunek 4.4: Wszystkie heksastruktury reprezentowane przez rozszerzone ciągi binarne złożone z jednoelementowych ciągów binarnych: 0(0)0, 0(0)1, 0(1)0, 0(1)1, 1(0)0, 1(0)1, 1(1)0 oraz 1(1)1. Na czerwono zaznaczono krawędzie początkowe rozważanych rozszerzonych ciągów binarnych.

Przykład 4.2.13. Rozważamy porządek leksykograficzny \leq_L oraz zbiór wszystkich rozszerzonych ciągów binarnych złożonych z jednoelementowych ciągów binarnych (por. rys. 4.4 oraz 4.5). Wówczas w tym zbiorze element $0(0)0$ jest infimum, a element $1(1)1$ jest supremum.

Przykład 4.2.14. Rozważamy porządek leksykograficzny \leq_L oraz rozszerzone ciągi binarne $0(1)0$, $0(0)1$, $1(1)0$ oraz $1(0)1$ (por. rys. 4.5). Wówczas taki zbiór nie zawiera elementu najmniejszego ani elementu największego.



Rysunek 4.5: Diagramy Hassego dla wszystkich rozszerzonych ciągów binarnych złożonych z jednoelementowych ciągów binarnych. Porządek leksykograficzny wykorzystano w diagramie po lewej stronie. Porządek epileksykograficzny wykorzystano w diagramie po prawej stronie.

Podsumowując, porządek leksykograficzny daje wygodną i prostą metodę agregacji heksastruktur kosztem ograniczonej i skromnej interpretacji jego wyników. Możemy na jego podstawie stwierdzić, który z heksastruktur jest złożony z krótszych ścieżek oraz który ma więcej kątów lewostronnych na pozycjach początkowych. Został wykorzystany w zastosowaniu przedstawionym w podrozdziale 5.2 do agregacji wiedzy rozproszonej, co przedstawiono na rysunku 5.8.

4.2.2 Porządek epileksykograficzny

Prostota porządku leksykograficznego niesie ze sobą określone konsekwencje formalne. Pozwala nam on jedynie określić, czy dany heksastruktur jest bardziej „zakręcony” w lewo w porównaniu do innego heksastrukturu oraz czy ma mniejszą długość. Aby uzyskać pełniejszy obraz i umożliwić badanie większej liczby aspektów geometrycznych tych struktur, proponujemy rozszerzenie tego porządku o dodatkowe kryterium, jakim jest liczba sąsiadujących ze sobą identycznych bitów. Wprowadzenie tego rozszerzenia umożliwi bardziej

szczegółową i precyzyjną analizę różnic między heksastruktami, co z kolei pozwoli lepiej uchwycić ich właściwości topologiczne i geometryczne.

Niech B i B' będą ciągami binarnymi oraz niech p i p' będą liczbami sąsiadujących ze sobą identycznych bitów w ciągach, odpowiednio, B i B' . Porządkiem epaleksykograficznym (gr. *epanálepsis*, „powtórzenie”) nazywamy taki porządek \leq_M , że $p < p'$ lub $p = p'$ oraz $B \leq_L B'$.

Przykład 4.2.15. *Rozważamy ciągi binarne: 0, 100, 0000 oraz 1010. Wówczas $0 \leq_M 100$, $0 \leq_M 0000$, $0 \leq_M 1010$, $100 \leq_M 0000$, $1010 \leq_M 100$ oraz $1010 \leq_M 0000$.*

Twierdzenie 4.2.16. *Porządek epaleksykograficzny jest porządkiem liniowym na zbiorze ciągów binarnych.*

Analizę interpretacji oraz praktycznego potencjału porządku epaleksykograficznego w agregacji heksastruktów przeprowadzimy analogicznie do analizy porządku leksykograficznego.

Przykład 4.2.17. *Rozważamy heksastrukty reprezentowane przez trzejelementowe ciągi binarne: 000, 001, 010, 011, 100, 101 oraz 111 (por. rys. 4.6). Wówczas możemy ułożyć te ciągi w ciągu niemalejącym względem porządku epaleksykograficznego w następujący sposób:*

$$010 \leq_M 101 \leq_M 001 \leq_M 011 \leq_M 100 \leq_M 110 \leq_M 000 \leq_M 111.$$

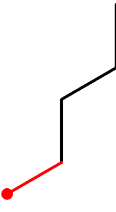
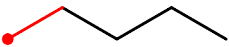
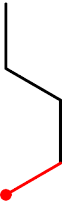
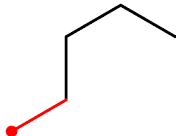
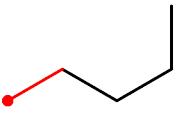
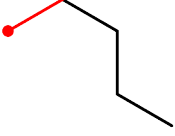
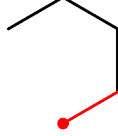
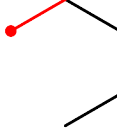
Geometrycznie struktury układamy od tych o mniej skomplikowanym kształcie, w których występuje mniejsza liczba sąsiadujących ze sobą kątów po tej samej stronie krawędzi, do tych o bardziej skomplikowanym kształcie, w których pojawia się większa liczba sąsiadujących ze sobą kątów po tej samej stronie krawędzi.

Przykład 4.2.18. *Rozważamy heksastrukty reprezentowane przez ciągi binarne: 0, 01, 1, 00, 001, 000, 0001 oraz 0000 (por. rys. 4.6). Wówczas możemy ułożyć te ciągi w ciągu niemalejącym względem porządku epaleksykograficznego w następujący sposób:*


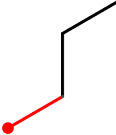
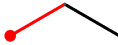

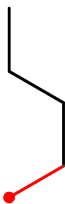
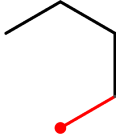
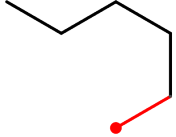
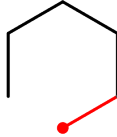
$$0 \leq_M 01 \leq_M 1 \leq_M 00 \leq_M 001 \leq_M 000 \leq_M 0001 \leq_M 0000.$$

Potwierdza to formalną definicję porządku epaleksykograficznego, zgodnie z którą heksastrukty uznaje się za mniejsze, jeżeli są reprezentowane przez ciągi binarne zawierające mniejszą liczbę par sąsiadujących identycznych bitów. W przypadku równej liczby takich par porządek ustala się na podstawie odchylenia w lewo, analogicznie do porządku leksykograficznego.

Przykład 4.2.19. *Na rysunku 4.5 przedstawiliśmy diagram Hassego, który pokazuje relacje porządku epaleksykograficznego dla ciągów binarnych: 0(0)0, 0(0)1, 0(1)0, 0(1)1, 1(0)0, 1(0)1, 1(1)0 oraz 1(1)1. W porównaniu z diagramem Hassego dla porządku leksykograficznego obserwujemy wzrost liczby par elementów, które są nieporównywalne względem siebie w nowym porządku. Ponadto zauważamy, że diagram Hassego dla porządku epaleksykograficznego powstał poprzez obrócenie diagramu leksykograficznego o 90 stopni zgodnie z ruchem wskazówek zegara, przy czym zachowano pionową relację między rozszerzonymi*

| | | | |
|---|---|---|---|
|  |  |  |  |
| 010 | 101 | 001 | 011 |
|  |  |  |  |
| 100 | 110 | 000 | 111 |

Rysunek 4.6: Struktury reprezentowane przez trzyelementowe ciągi binarne, posortowane rosnąco z wykorzystaniem porządku epaleksykograficznego wierszami od lewej górnej struktury do prawej dolnej struktury.

| | | | |
|---|---|---|---|
|  |  |  |  |
| 0 | 01 | 1 | 00 |
|  |  |  |  |
| 001 | 000 | 0001 | 0000 |

Rysunek 4.7: Struktury reprezentowane przez ciągi binarne 0, 00, 000, 0000, 0001, 001, 01 oraz 1, posortowane rosnąco z wykorzystaniem porządku epaleksykograficznego wierszami od lewej górnej struktury do prawej dolnej struktury.

ciągami binarnymi $0(1)1$ i $1(0)0$. Ta zmiana wizualna ilustruje istotną modyfikację relacji porządkowych między rozważanymi obiektami, wskazując na inny sposób hierarchizacji i porządkowania struktur wewnątrz zbioru. Dzięki temu obrazowi możemy intuicyjnie zrozumieć, w jaki sposób rozbudowa porządku wpływa na wzajemne zależności między ciągami.

Przykłady 4.2.15, 4.2.17, 4.2.18 i 4.2.19 pozwalają na wyciągnięcie następujących wniosków dotyczących porządku epaleksykograficznego:

Wniosek 4.2.20. *Ciąg binarny 0 jest najmniejszy w zbiorze ciągów binarnych reprezentujących heksastryki względem porządku epaleksykograficznego.*

Wniosek 4.2.21. *Nie istnieje element największy w zbiorze ciągów binarnych reprezentujących heksastryki względem porządku epaleksykograficznego.*

Dowód. Przypuśćmy, że $B \in \mathcal{B}$ jest elementem największym w zbiorze ciągów binarnych reprezentujących heksastryki względem porządku epaleksykograficznego. Niech B posiada $p \in \mathbb{N}$ par sąsiadujących ze sobą identycznych bitów. Wówczas ciąg binarny B' o długości $4p$, gdzie $b_{4i} = 0 = b_{4i+1}$ oraz $b_{4i+2} = 1 = b_{4i+3}$, dla $i \in \{0, 1, \dots, p-1\}$, ma $2p$ sąsiadujących ze sobą tych samych bitów. Istnienie heksastryki reprezentowanego przez taki ciąg binarny zostało pokazane w przykładzie A.1.13. Zatem $B \leq_M B'$, a więc B nie jest elementem największym w tym zbiorze. \square

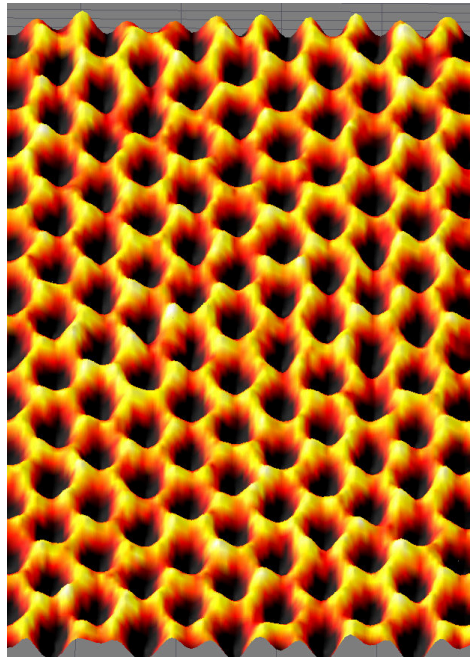
Podsumowując, kierunek rozwinięcia porządku leksykograficznego o uwzględnienie pojawiania się identycznych bitów obok siebie, prowadzący do porządku epaleksykograficznego, oceniamy jako bardzo dobry. Ten rozszerzony porządek dostarcza dodatkowych informacji o porównywanych heksastrykach, pozwalając na lepsze uwzględnienie ich geometrycznych cech. Jednocześnie nie ogranicza on z góry analizy do konkretnego ciągu binarnego, co zwiększa jej elastyczność. Naturalną konsekwencją wprowadzenia tego porządku jest wzrost liczby par heksastryków, które mogą okazać się bezpośrednio nieporównywalne, jednak zyski w postaci głębszej analizy różnic i lepszej reprezentacji struktur przeważają nad tym ograniczeniem.

Rozdział 5

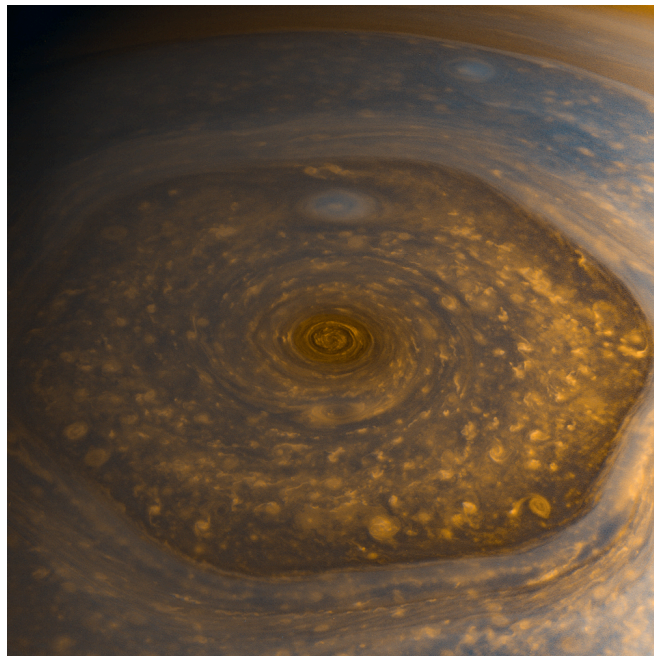
Zastosowania

Przykłady zastosowań wyników uzyskanych w niniejszej dysertacji ograniczamy do trzech wybranych przypadków ze względu na rozpowszechnienie struktury sześciokątnej w różnorodnych zagadnieniach i problemach naukowych. Powszechność występowania tej geometrii wynika z jej efektywności w wypełnianiu przestrzeni oraz z jej związków z fundamentalnymi strukturami przyrody i materiałów. W tym rozdziale rozwiniemy trzy zagadnienia naukowe o najwyższym potencjale wykorzystania heksastrukturów oraz przedstawionych w niniejszej pracy metod ich porównywania i agregacji. Niemniej kształt ten pojawia się, a tym samym istnieje możliwość wykorzystania agregacji heksastrukturów w następujących dziedzinach i zagadnieniach naukowych:

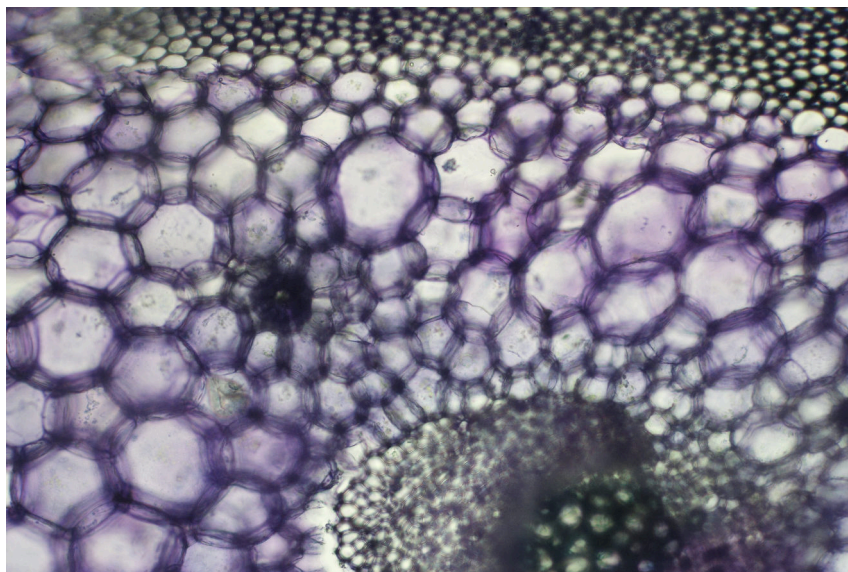
- informatyka – sieci szkieletowe (patrz [13]),
- matematyka – teoria grafów i kraty (patrz [23]),
- fizyka ciała stałego – struktura materiałów takich jak grafen (patrz [40], [49] oraz rys. 5.1),
- kosmologii – pierścień na północnym biegunie Saturna (patrz [48] oraz rys. 5.2),
- botanika – organizacja komórek w tkankach roślinnych (patrz [35] oraz rys. 5.3),
- nauki o Ziemi – bazaltowe kolumny (patrz [43] oraz rys. 5.4),
- geofizyka – siatka geodezyjna (patrz [39]),
- zoologia – analiza wzorców komórek siatkowych (patrz [28], [41]).



Rysunek 5.1: Obraz pojedynczej warstwy grafenu otrzymany przy użyciu mikroskopu elektronowego. Źródło: <https://www.flickr.com/photos/armymaterielcommand/6795812766>. Licencja: CC BY-ND 2.0.



Rysunek 5.2: Heksagonalna burza na Saturnie. Źródło: <https://www.flickr.com/photos/kevinmgill/24588767787>. Licencja: CC BY-ND 2.0.



Rysunek 5.3: Przekrój poprzeczny selera naciowego z widocznymi komórkami o kształcie sześciokątnym. Źródło: <https://www.flickr.com/photos/cbaek/6118835197>. Licencja: CC BY-ND 2.0.



Rysunek 5.4: Bazaltowe kolumny o kształcie sześciokątnym. Źródło: <https://www.geograph.org.uk/photo/1478562>. Licencja: CC BY-ND 2.0.

5.1 Teoria gier oraz symulacje komputerowe

Sześciokątna siatka jest powszechnie wykorzystywana w teorii gier oraz ludologii, ponieważ umożliwia tworzenie spójnych map złożonych z heksagonalnych żetonów, a jednocześnie pozwala na zdefiniowanie sześciu możliwych akcji odpowiadających każdej ze ścian pojedynczego żetonu (patrz [27] oraz [44]). Dzięki temu powstaje wiele gier o wysokim stopniu interaktywności, w których każda akcja gracza jest ściśle związana z topologią siatki sześciokątnej oraz wzajemnymi relacjami między poszczególnymi żetonami. Dobrymi przykładami takich gier planszowych są Hex, Rój (ang. *Hive*) oraz Catan, które wykorzystują unikalne właściwości struktury heksagonalnej do budowania złożonych i dynamicznych mechanik gry.

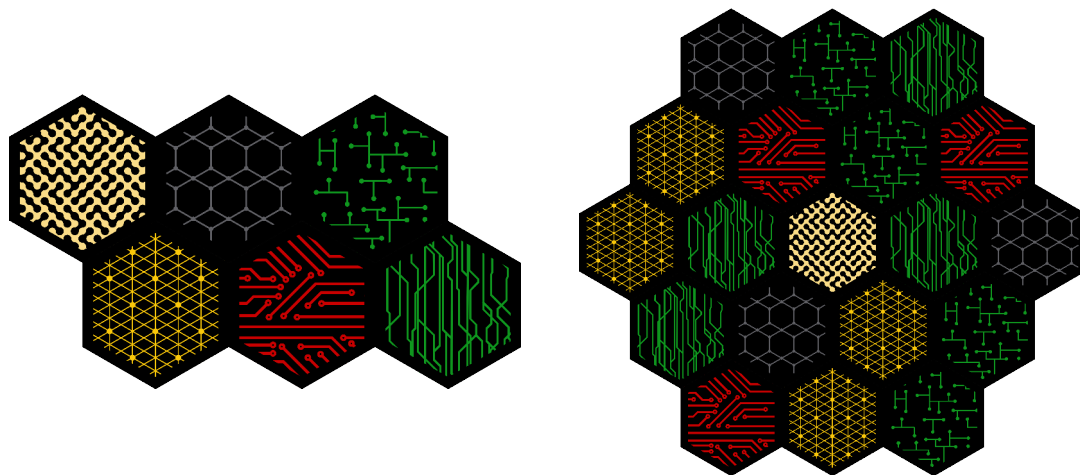
Catan, znany również jako „Osadnicy z Catanu” (ang. *Settlers of Catan*), jest popularną grą planszową zaprojektowaną przez Klausa Teubera, która zadebiutowała w 1995 roku. Jest to gra strategiczna przeznaczona dla 2–4 graczy, w której uczestnicy wcielają się w rolę osadników kolonizujących fikcyjną wyspę Catan. Głównym celem gry jest zdobywanie punktów zwycięstwa poprzez budowanie osad, miast oraz dróg, a także rozwijanie swojej kolonii i prowadzenie efektywnej strategii rozwoju. Plansza składa się z pól heksagonalnych reprezentujących różne rodzaje terenu (por. rys. 5.5), z których można pozyskać surowce:

- z gór – kamień,
- z pastwisk – wełnę,
- z pól uprawnych – zboże
- ze wzgórz – glinę,
- z lasów – drewno,

oraz z pustyni, która nie dostarcza żadnych surowców. Gracze zbierają te surowce na podstawie rzutów kostkami i wykorzystują je do rozbudowy swoich osiedli, stawiając drogi na krawędziach sześciokątów oraz budowie na ich wierzchołkach.

Gracze mogą tworzyć na mapie siatkę dróg, którą formalnie reprezentuje heksastrukt, struktura oparta na siatce sześciokątów, charakteryzująca się określoną topologią i geometrią. Dzięki temu możliwe jest systematyczne porównywanie poszczególnych heksastruktów oraz łączenie ich w ramach procesu agregacji, który wykorzystuje metody matematyczne i algorytmiczne opracowane wcześniej. Proces ten umożliwia analizę zbioru różnych konfiguracji dróg pod kątem ich własności oraz wzajemnych zależności. Głównym celem takiej agregacji jest identyfikacja heksastruktów, który maksymalizuje prawdopodobieństwo osiągnięcia zwycięstwa, co stanowi kluczowy element w strategii i optymalizacji rozgrywki. W ten sposób podejmowane decyzje stają się bardziej świadome i ukierunkowane na uzyskanie optymalnych rezultatów.

W trakcie naszych badań przeprowadziliśmy 1001 symulacji, w których gracz zbudował ustaloną drogę, odniósł zwycięstwo oraz zbudował spójną sieć dróg pozbawioną cykli.



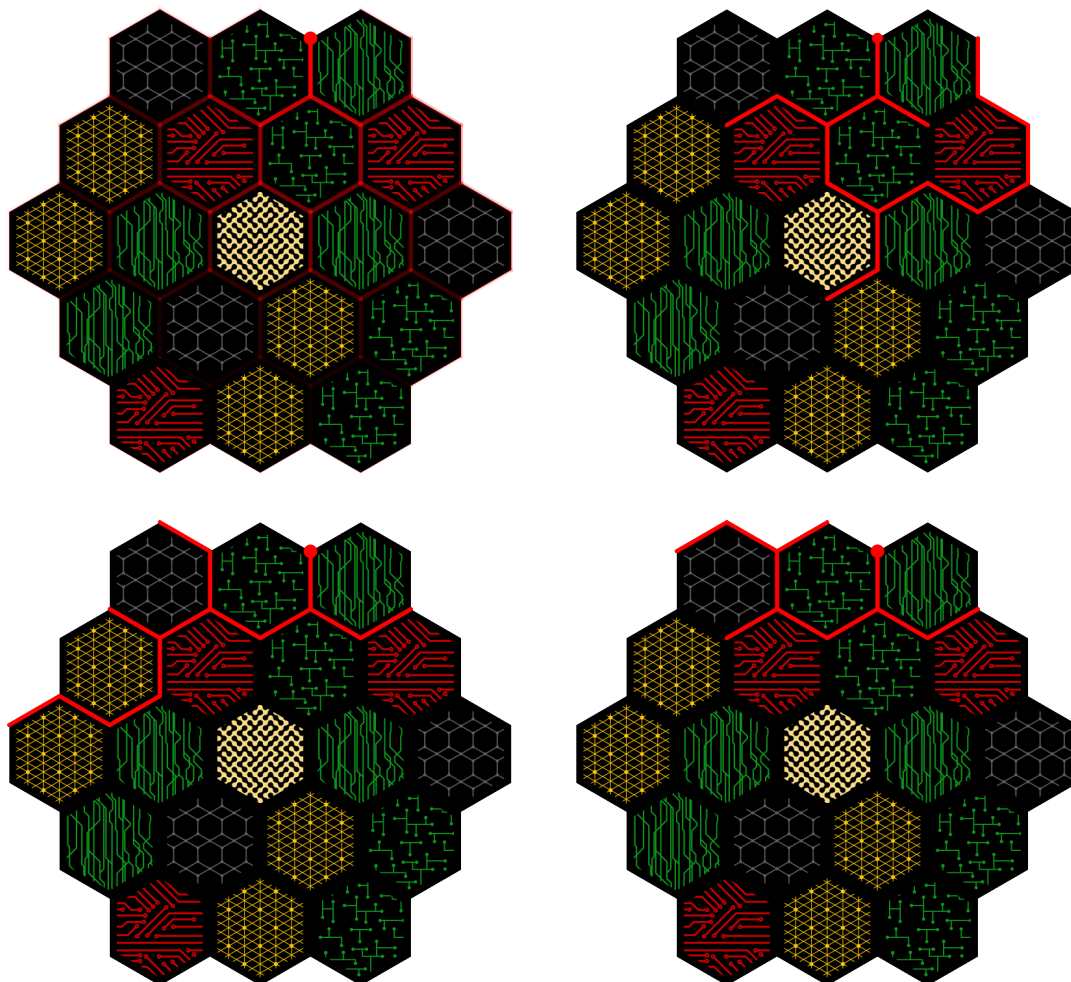
Rysunek 5.5: Kafelki w kształcie sześciokątów, z których układana jest mapa do gry w Catan (po lewej stronie). Żetony, odpowiednio od lewej góry do prawego dołu, reprezentują: pustynię, góry, pastwiska, pola uprawne, wzgórza i lasy. Podstawowa konfiguracja żetonów terenów w mapę Catan (po prawej stronie).

Dzięki takim założeniom możemy reprezentować każdą symulację dokładnie jednym heksastruktem, który przedstawia sieć dróg zbudowaną przez gracza (por. rys. 5.6). Możemy zamienić lub pominąć poszczególne założenia w zależności od celu badań. Pominięcie warunku wygranej gracza pozwoli nam na ogólną analizę heksastruktury budowanego przez graczy. Rozważenie heksastruktur z cyklami zwiększy złożoność obliczeniową ze względu na konieczność uwzględnienia kilku reprezentacji heksastruktur przy pomocy rozszerzonych ciągów binarnych.

Uwaga 5.1.1. *W repozytorium przedstawionym w dodatku B udostępniliśmy skrypt w języku Python, który umożliwia czytelnikowi przeprowadzenie symulacji gry Catan oraz agregację wyników otrzymanych z wykorzystaniem metod opisanych w niniejszym podrozdziale.*

Otrzymane heksastryki zagregowaliśmy wyznaczając medoidy z wykorzystaniem trzech miar odległości: Levenshteina, rozszerzonej odległości Hamminga oraz rozszerzonej ważonej odległości Hamminga. Na rysunku 5.6 przedstawiliśmy odpowiednie medoidy, medoid wyznaczony przy pomocy odległości Levenshteina umieściliśmy w prawym górnym rogu, medoid oparty na rozszerzonej odległości Hamminga w lewym dolnym rogu, natomiast medoid wykorzystujący rozszerzoną ważoną odległość Hamminga znajduje się w prawym dolnym rogu. W dalszej części analizujemy zalety oraz ograniczenia każdego z wyznaczonych medoidów, ze szczególnym uwzględnieniem wpływu tych medoidów na zwiększenie prawdopodobieństwa wygranej. Dzięki temu podejściu oceniamy efektywność poszczególnych metod agregacji w kontekście zastosowań praktycznych i wskazujemy najbardziej obiecujące kierunki optymalizacji modelu.

Medoid wyznaczony z wykorzystaniem odległości Levenshteina reprezentuje układ ścieżek rozgałęziony we wszystkich kierunkach mapy, co wskazuje na podejmowanie przez gracza decyzji bez jasno określonego planu budowy sieci dróg. Zdecydowaną zaletą tego medoidu w porównaniu do pozostałych jest obecność długiej ścieżki złożonej z dziesięciu odcinków dróg, co zwiększa szansę na zdobycie punktów za takie osiągnięcie. Ponadto wy-



Rysunek 5.6: Po lewej u góry przedstawiono 1001 heksastrukturów reprezentujących sieci dróg stworzone przez graczy, nałożone na siebie i przestawione na podstawowej mapie Catan, z początkiem zaznaczonym czerwonym punktem oraz krawędziami o odcieniach czerwieni o nasyceniu $1/1001$ dla każdej ścieżki. Kolejno, po prawej u góry, po lewej na dole oraz po prawej na dole, przedstawiono medoidy dla 1001 heksastrukturów otrzymanych w symulacji, zaprezentowane na podstawowej mapie Catan, z początkiem zaznaczonym czerwonym punktem oraz krawędziami o kolorze czerwonym, wyznaczone z wykorzystaniem odległości, odpowiednio, Levenshteina, rozszerzonej Hamminga oraz rozszerzonej ważonej Hamminga.

budowane drogi umożliwiają pozyskiwanie surowców z dziewięciu różnorodnych kafelków, co wpływa pozytywnie na potencjał rozwoju. Z drugiej strony identyfikujemy trzy drogi prowadzące wokół pustyni oraz kolejne trzy umiejscowione na brzegach mapy, które nie generują surowców, co obniża efektywność całego układu dróg. Dodatkowo łączna liczba dróg wynosi piętnaście, co może stanowić istotne wyzwanie dla graczy ze względu na konieczność rywalizacji z trzema przeciwnikami, którzy utrudniają realizację tak rozbudowanego planu.

Medoid wyznaczony z wykorzystaniem rozszerzonej odległości Hamminga przedstawia układ ścieżek, który jest wyraźnie skierowany ku zachodniej części mapy. Utworzony kształt sugeruje, że gracz planował strukturę sieci dróg i budował je według określonego, konsekwentnego kierunku. Cała sieć składa się z trzynastu dróg, przy czym najdłuższa pojedyncza ścieżka obejmuje osiem połączonych odcinków. Taki układ wskazuje na większe prawdopodobieństwo osiągnięcia podobnej konfiguracji przez graczy w porównaniu ze strukturami otrzymanymi poprzez agregację z wykorzystaniem miary Levenshteina. Medoid ten umożliwia pozyskiwanie surowców z dziewięciu różnych kafelków, co jest wynikiem równym do tego, jaki uzyskaliśmy dla medoidu opartego na odległości Levenshteina. Dodatkowo zauważamy, że dwie drogi przebiegają wzdłuż skraju mapy, natomiast pozostałe drogi łączą pola dostarczające surowce, co zwiększa efektywność sieci i wspiera rozwój gracza.

Ostatni medoid, wyznaczony z wykorzystaniem rozszerzonej ważonej odległości Hamminga, reprezentuje wyraźnie ograniczony układ ścieżek składający się z dziesięciu dróg, z których najdłuższa tworzy ścieżkę o długości sześciu odcinków. Taka konfiguracja pozwala na zdobywanie surowców jedynie z sześciu pól, co stanowi wynik znacząco gorszy w porównaniu z dwoma pozostałymi przypadkami. Ponadto ta struktura pomija pola uprawne, co ogranicza możliwości pozyskania zboża, jednego z kluczowych surowców. Dodatkowo trzy drogi tej sieci są usytuowane na obrzeżach mapy, co zmniejsza ich funkcjonalność. Mimo że istnieje wysoka szansa na ułożenie takiego układu dróg podczas rozgrywki, to jednak ograniczona liczba pozyskiwanych surowców oraz krótsza najdłuższa ścieżka mogą istotnie ograniczyć szanse na zdobycie wysokiej liczby punktów.

Analiza otrzymanych wyników wskazuje, że najbardziej obiecującą metryką spośród trzech rozważanych jest rozszerzona odległość Hamminga. Zaproponowany układ ścieżek charakteryzuje się wysokim prawdopodobieństwem realizacji podczas rozgrywki, zapewnia dostęp do wszystkich kluczowych surowców oraz zwiększa znacząco szansę na zdobycie punktów za najdłuższą ścieżkę.

Ponadto zmiana warunków początkowych, takich jak układ kafelków dostarczających surowce oraz wybór innej krawędzi początkowej, może prowadzić do sformułowania dodatkowych istotnych wniosków, które pozwolą na lepsze dopasowanie strategii do różnorodnych wariantów rozgrywki.

5.2 Systemy wieloagentowe oraz wiedza rozproszona

Systemy wieloagentowe (ang. *Multi-Agent Systems, MAS*) to stale rozwijająca się interdyscyplinarna dziedzina badań, łącząca m.in. informatykę, sztuczną inteligencję, ekonomię oraz socjologię, znajdująca zastosowanie w wielu obszarach naukowych i biznesowych, np. w robotyce, technologiach blockchain, systemach transportowych oraz zarządzaniu organizacjami (patrz [17], [24], [38], [47]). Wyjątkowa zdolność tych systemów do reprezentowania dynamicznych, rozproszonych oraz autonomicznych systemów, powszechnie spotykanych we współczesnych aplikacjach technologicznych, społecznych i ekonomicznych, wymaga ciągłego rozwoju narzędzi oraz algorytmów pozwalających na ich efektywne działanie.

Systemy wieloagentowe składają się z autonomicznych jednostek, nazywanych agentami, które współpracują w celu realizacji celów indywidualnych oraz wspólnych. Każdy agent ma możliwość podejmowania autonomicznych decyzji oraz interakcji z innymi agentami tworząc integralną część całego systemu. Agentem może być nie tylko system informatyczny, ale również człowiek, robot, organizacja, model matematyczny lub inne źródło informacji. Ich współpraca, poza wymianą informacji, obejmuje również koordynację działań oraz negocjacje.

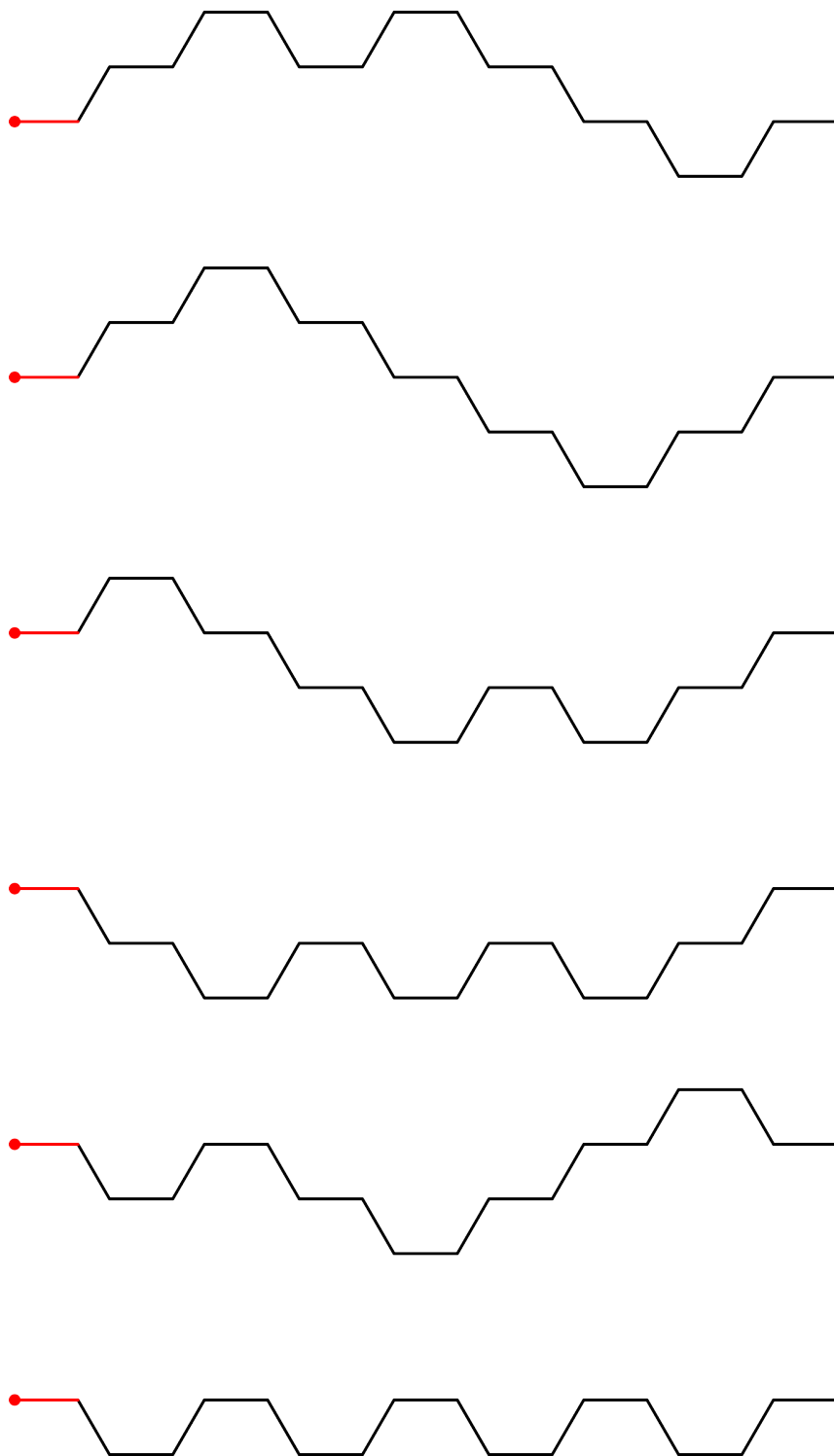
Jednym z aspektów systemów wieloagentowych jest możliwość modelowania wiedzy rozproszonej, czyli sytuacji, w której informacje są rozdzielone między agentów, a ostateczny wynik uzyskuje się poprzez ich agregację. Struktura takich systemów odpowiada matematycznie grafom, w których wierzchołki reprezentują agentów, a krawędzie odzwierciedlają zależności między tymi agentami. W konsekwencji heksastryktu, jako specyficzny typ grafów, mogą być wykorzystywane do reprezentacji oraz agregacji wiedzy rozproszonej.

Rozważamy system, w którym wyróżniamy agenta początkowego oraz agenta końcowego, a każdy agent stanowi część pewnej ścieżki będącej heksastryktem, która ma początek w agencie początkowym oraz koniec w agencie końcowym. Przyjmujemy, że system składa się z sześciu ścieżek, reprezentowanych przez ciągi binarne:

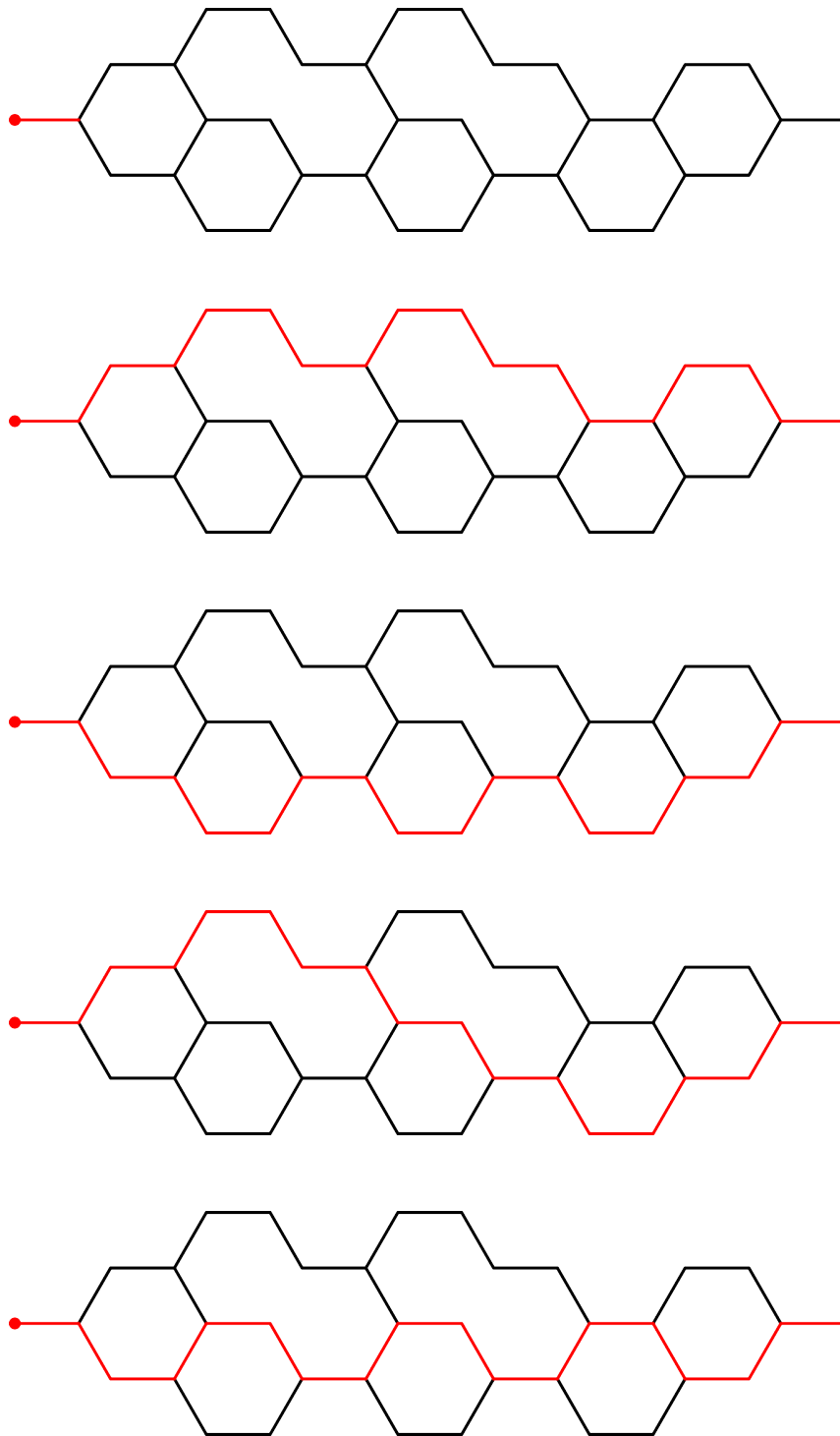
- 0101100110101001,
- 0101101010100101,
- 0110101001100101,
- 1010011001100101,
- 1001101001010110,
- 1001100110011001.

Wówczas możemy wykorzystać agregację heksastryktów z wykorzystaniem porządków częściowych w celu wyznaczenia takiej ścieżki, która będzie zapewniała optymalny przepływ informacji (por. rysunki 5.7 oraz 5.8). Otrzymujemy następujące wyniki agregacji powyższych ścieżek z wykorzystaniem porządku leksykograficznego:

- infimum – 0101100110101001 (por. czerwoną ścieżkę w drugim heksastrykcie na rysunku 5.8),



Rysunek 5.7: Heksastrykty reprezentowane przez ciągi binarne, z początkiem zaznaczonym kolorem czerwonym, odpowiednio od góry do dołu: 0101100110101001, 0101101010100101, 0110101001100101, 1010011001100101, 1001101001010110, 1001100110011001.



Rysunek 5.8: Heksastrykty reprezentowane przez ciągi binarne: 0101100110101001, 0101101010100101, 0110101001100101, 1010011001100101, 1001101001010110 oraz 1001100110011001, nałożone na siebie w taki sposób, aby ich krawędzie początkowe pokrywały się, tworząc sieć systemu wieloagentowego. Kolejno od góry do dołu przedstawiono sieć z krawędzią początkową zaznaczoną kolorem czerwonym oraz sieci z zaznaczoną kolorem czerwonym ścieżką będącą infimum oraz supremum względem porządku leksykograficznego, a także infimum oraz supremum względem porządku epaleksykograficznego.

- supremum – 1010011001100101 (por. czerwoną ścieżkę w trzecim heksastrukcie na rysunku 5.8).

Ponadto otrzymujemy następujące wyniki agregacji powyższych ścieżek z wykorzystaniem porządku epaleksykograficznego:

- infimum – 0101101010100101 (por. czerwoną ścieżkę w czwartym heksastrukcie na rysunku 5.8),
- supremum – 1001100110011001 (por. czerwoną ścieżkę w piątym heksastrukcie na rysunku 5.8).

Podsumowując, wykorzystanie agregacji heksastrukturów w sieciach systemów wieloagentowych opartych na siatce heksagonalnej wykazuje istotny potencjał zarówno w badaniach teoretycznych, jak i zastosowaniach praktycznych.

5.3 Chemia organiczna oraz stereoizometria

Stereoizomery to związki chemiczne, w których atomy są ułożone w takich samych sekwencjach, a które różnią się ułożeniem przestrzennym (patrz [5]). W zależności od kontekstu stereoizomery związków opartych na siatce sześciokątnej możemy reprezentować za pomocą heksastruktów, co umożliwi ich porównywanie oraz agregację na podstawie konfiguracji przestrzennej. Takie podejście pozwala na podjęcie dalszych badań nad strukturą oraz właściwościami chemicznymi i fizycznymi takich związków. Jednym z takich związków jest 3-metyloheksan, dla którego istnieją dwa enancjomery, czyli związki będące względem siebie odbiciami lustrzanymi (por. rys. 5.9). Wówczas (R)-3-metyloheksan reprezentujemy rozszerzonym ciągiem binarnym (01)1, a (S)-3-metyloheksan reprezentujemy rozszerzonym ciągiem binarnym (0)10 (por. rys. 5.10). Zauważmy, że stosując podstawowe operacje binarne na rozszerzonych ciągach binarnych, z jednego ciągu otrzymujemy drugi, tj.

$$\tau((01)1) = (0)10$$

oraz

$$\tau((0)10) = (01)1.$$

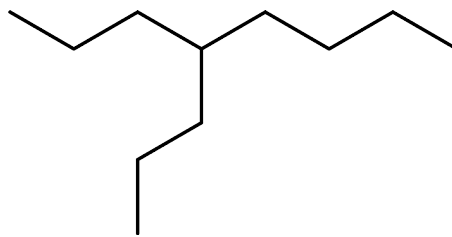


Rysunek 5.9: Odpowiednio od lewej do prawej przedstawiono wzory strukturalne (R)-3-metyloheksanu i (S)-3-metyloheksanu. Pogrubiona krawędź oznacza wiązanie wychodzące z płaszczyzny. Przerywana krawędź oznacza wiązanie chowające się za płaszczyzną.



Rysunek 5.10: Odpowiednio od lewej do prawej przedstawiono heksastrukty reprezentowane przez rozszerzone ciągi binarne (01)1 i (0)10 z początkami zaznaczonymi kolorem czerwonym.

W ogólnej postaci wszystkie węglowodory można reprezentować za pomocą rozszerzonych ciągów binarnych, co otwiera nowe potencjalne kierunki badań w chemii. 4-propyloktan to związek złożony z jedenastu atomów węgla oraz dwudziestu czterech



Rysunek 5.11: Wzór szkieletowy 4-propylooktanu.

atomów wodoru, w którym grupa propylowa jest przyłączona do czwartego atomu węgla głównego łańcucha oktanowego (por. rys. 5.11). Reprezentację tej cząsteczki można przedstawić przy użyciu trzech różnych rozszerzonych ciągów binarnych w zależności od przyjętego pierwszego wiązania: 10(01)101, 01(101)10 oraz 010(10)01. Stosując odpowiednie operacje binarne, można otrzymać z dowolnego z wymienionych ciągów pozostałe, np.

$$\pi(10(01)101) = 01(101)10$$

oraz

$$\pi(01(101)10) = 010(10)01.$$

Rozdział 6

Zakończenie

Celem niniejszej pracy jest zaproponowanie nowych metod porównywania oraz agregacji spójnych grafów, których struktura jest oparta na siatce heksagonalnej. Istotą przyjętego podejścia jest zachowanie pełnej informacji o strukturze tych grafów podczas procesu porównywania oraz agregacji, tak aby wynikiem tych operacji był również graf o takiej samej, heksagonalnej strukturze.

Kluczowym wynikiem naszych badań jest przedstawienie struktur o kształcie opartym na siatce heksagonów, które nazwaliśmy heksastruktami, przy pomocy rozszerzonych ciągów binarnych, co pozwala na odtworzenie rozważanych grafów oraz ułatwia ich porównywanie oraz agregację. Otrzymaliśmy go poprzez utworzenie pokrycia ścieżkowego będącego drzewem binarnym, co nazwaliśmy pokryciem właściwym, które istnieje dla każdego heksastrukt (por. twierdzenie 3.3.2). Następnie, reprezentując rekurencyjnie każdą ścieżkę heksastruktu ciągiem binarnym (por. twierdzenie 3.4.1), reprezentujemy heksastrukt o danym właściwym pokryciu ścieżkowym rozszerzonym ciągiem binarnym. Każdy ciąg binarny lub rozszerzony ciąg binarny reprezentuje dokładnie jeden heksastrukt, a przekształcenie rozważanego ciągu w heksastrukt polega na obliczaniu kolejnych współrzędnych wierzchołków heksastrukt z uwzględnieniem kątów $\pm 120^\circ$, więc najważniejsze założenie celu tej pracy zostało spełnione, nie tracimy informacji o strukturze grafu podczas jego reprezentacji ciągami binarnymi lub rozszerzonymi ciągami binarnymi.

Naszymi docelowymi wynikami są: zdefiniowanie metryki dla rozszerzonych ciągów binarnych (por. podrozdział 4.1 oraz wzór (4.1.1)) oraz zaproponowanie porównywania rozszerzonych ciągów binarnych z wykorzystaniem częściowych porządków (por. podrozdział 4.2).

Rozważyliśmy metrykę Hamminga oraz metrykę Levenshteina, a następnie rozszerzyliśmy pierwszą z nich na ciągi binarne dowolnej długości oraz ostatecznie wprowadziliśmy funkcję ważącą różnicę między dwoma porównywanymi ciągami na poszczególnych ich pozycjach. Rozszerzona ważona odległość Hamminga pozwala nam na uwzględnienie różnic między heksastruktami, gdzie różnica na początkowych pozycjach ma zdecydowanie większy wpływ na różnicę między obiektami niż różnica na końcowych indeksach. Metrykę Levenshteina wzięliśmy pod uwagę ze względu na brak znaczenia pozycji, na których ciągi się różnią, gdyż metryka ta określa liczbę następujących operacji: dodawanie znaków, usuwanie znaków, podstawienie znaków oraz zamiana pozycjami dwóch znaków. Wszystkie te

metryki przeanalizowaliśmy w praktycznym zastosowaniu agregując ścieżki w symulatorze Catanatron (por. podrozdział 5.1).

Porządek leksykograficzny to najpopularniejszy porządek pozwalający na sortowanie ciągu znaków. Konsekwencją jego prostoty jest ograniczona interpretacja wyniku porównania ciągów binarnych, gdyż informuje nas tylko o tym, który z badanych heksastruktów jest skręcony bardziej w prawo od drugiego. Dlatego proponujemy nowy porządek, który nazwaliśmy porządkiem epaleksykograficznym, który najpierw porównuje, który z rozważanych heksastruktów ma więcej par kątów znajdujących się po tej samej stronie krawędzi, a następnie korzysta z porządku leksykograficznego. Pozwala nam na zbadanie więcej różnic w strukturze heksastruktów niż porządek leksykograficzny kosztem zwiększenia się liczby nieporównywalnych par heksastruktów. W tym przypadku również przeanalizowaliśmy znaczenie obu porządków w praktycznych zastosowaniach agregując heksastrukty reprezentujące wiedzę rozproszoną (por. podrozdział 5.2).

Przedstawione wyniki zwięźczamy wskazaniem wybranych, aktualnych problemów badawczych w różnych dziedzinach nauki, które cechuje wysoki potencjał praktycznego zastosowania (por. rozdział 5). Szczegółowo badamy użyteczność zaproponowanych metryk oraz porządków w zagadnieniach: teoria gier i symulacje komputerowe oraz wiedza rozproszona. Ponadto proponujemy wykorzystanie wprowadzonych odwzorowań τ , ζ i π w chemii organicznej oraz stereoizomerii, gdzie cząsteczki złożone z węgla i wodoru o tym samym wzorze chemicznym tworzą struktury będące dla siebie odbiciem lustrzanym (por. podrozdział 5.3). Co więcej, w Dodatku A przedstawiamy wyniki pochodnych problemów, które pojawiły się w trakcie badań.

Poza tym prezentujemy szereg wyników pośrednich oraz pobocznych, które naszym zdaniem również są bardzo istotne i wytyczają kolejne kierunki badań naukowych. Zdefiniowane operacje τ , ζ oraz π (por. podrozdziały 2.1 oraz 2.4), które zwracają ciągi binarne powstałe przez, odpowiednio, zamianę bitów ciągu na przeciwne, odwrócenie ciągu oraz zamianę bitów na przeciwne i odwrócenie ciągu, mają bezpośrednio przełożenie na własności geometryczne heksastruktów (por. podrozdział 2.1). Zamiana bitów oraz odwrócenie ciągu prowadzą do powstania heksastruktów będącymi odbiciami lustrzanymi do rozważanego heksastruktu (por. twierdzenia 3.4.4 oraz 3.4.13). Natomiast odwzorowanie π prowadzi do otrzymania rotacji badanego heksastruktu (por. twierdzenie 3.4.8). Zaproponowane rozszerzenia tych funkcji dla rozszerzonych ciągów binarnych zachowują te własności, a tym samym pozwalają na szerszą analizę heksastruktów reprezentowanych przez rozszerzone ciągi binarne (por. podrozdział 2.4 oraz twierdzenia 3.4.6, 3.4.11 i 3.4.15).

W dodatku A prezentujemy wyniki dotyczące wyznaczania współrzędnych danego heksastruktu oraz zliczania heksastruktów reprezentowanych przez ciągi binarne bez kolizji o danej długości. W podrozdziale A.1 prezentujemy heksagonalny układ współrzędnych, gdzie wierzchołki przedstawiamy jako parę (ι, ρ) , która odpowiada współczynniki stojącemu przy macierzy, odpowiednio, I oraz $\text{Rot}(120^\circ)$ (por. wniosek A.1.9). Taka postać zdecydowanie upraszcza obliczanie kolejnych współrzędnych wierzchołków heksastruktu pomijając mnożenie macierzy oraz obliczanie pierwiastków (por. twierdzenie A.1.16), a tym samym walidację ciągów binarnych, które potencjalnie mogą reprezentować heksastrukty (por. podrozdział A.2). W ostatniej części dodatku A prezentujemy wyniki dotyczące

problemu zliczania ścieżek rzędu drugiego na siatce heksagonów. Przedstawiamy wyniki dla ogólnego przypadku zliczania ciągów binarnych z uwzględnieniem niezmienniczości ze względu na rotacje oraz z uwzględnieniem niezmienniczości ze względu na odbicia lustrzane i rotacje (por. twierdzenia A.3.6, A.3.9 oraz A.3.13).

Na końcu pracy, w Dodatku B przedstawiamy implementacje obiektów, narzędzi oraz metod opisanych w tej pracy w języku Python. Przedstawiamy implementacje funkcji τ , ς oraz π (por. podrozdziały B.1, B.2 i B.3), a następnie implementację ciągów binarnych oraz rozszerzonych ciągów binarnych w postaci klasy (por. podrozdział B.4) oraz metryki (por. podrozdział B.5). Następnie prezentujemy sposoby wykonania symulacji z wykorzystaniem Catanatronu (por. podrozdział B.6) oraz metodę zliczania ścieżek rzędu drugiego heksastruktów (por. podrozdział B.6).

Oczywistym kierunkiem dalszych badań jest rozważenie nowych metryk oraz analiza ich własności oraz użyteczności w praktycznych zastosowaniach. Ponadto w rozszerzonej ważonej odległości Hamminga wykorzystaliśmy funkcję ważącą $w(i) = 2^{-i}$. Warto rozważenia byłyby również inne funkcje ważące, a w szczególności te malejące, które przede wszystkim uwzględniają wpływ różnicy bitów na początkowych pozycjach na różnicę między heksastruktami.

Ponadto zaproponowany porządek dla rozszerzonych ciągów binarnych wymaga dalszych badań ze względu na ograniczony potencjał tego porządku w takiej postaci. Oceniamy, że ta definicja będzie rozszerzana oraz korygowana przede wszystkim w konkretnych problemach badawczych, a nie w ogólnym przypadku.

Rozszerzenia wymaga również proces wyznaczania medoidu, który możemy rozszerzyć do wyznaczania centroidu. Wówczas wynikowy heksastrukt nie będzie ograniczony do rozważanego zbioru obiektów, a może być dowolny. Ograniczeniem jest tutaj utworzenie przestrzeni wszystkich heksastruktów. Oprócz wygenerowania zbioru rozszerzonych ciągów binarnych konieczna jest również weryfikacja, czy istotnie istnieje heksastrukt reprezentowany przez taki zbiór.

Dodatek A

Wyniki pokrewne

W niniejszym rozdziale przedstawiamy wyniki, które odegrały istotną rolę w toku prowadzonych przez nas badań, mimo że nie są bezpośrednio związane z procesem agregacji heksastruktów. Rezultaty te dotyczą w szczególności metod obliczania współrzędnych wierzchołków heksastruktów, co stanowi podstawę dalszej analizy ich właściwości geometrycznych. Ponadto skupiamy się na zagadnieniu zliczania heksastruktów o określonej długości, przy czym w trakcie tych rozważań uwzględnialiśmy ich niezmienniczość względem odbić lustrzanych oraz obrotów, co pozwala wyeliminować powtarzające się przypadki i uzyskać bardziej precyzyjne statystyki dotyczące klasyfikacji tych struktur. Całość analizy kończymy omówieniem kluczowych wyników i sformułowaniem wniosków, które w naszej ocenie mają istotne znaczenie teoretyczne i praktyczne, a ich poznanie przyczynia się do pogłębienia zrozumienia problematyki heksastruktów oraz wskazuje kierunki dalszych badań.

A.1 Współrzędne wierzchołków heksastruktu

W tym podrozdziale przestawimy wyniki dotyczące obliczania współrzędnych wierzchołków heksastruktów. Wyprowadzimy wzory nierekurencyjne oraz zaproponujemy heksagonalny układ współrzędny, w którym wierzchołki heksastruktów są parą liczb całkowitych, a nie rzeczywistych. Przyjmujemy poniższe oznaczenia, aby ich nie powielać w treści twierdzeń. Niech $n \in \mathbb{N}$ i niech $p = (e_0, e_1, \dots, e_{n-1})$ jest taką ścieżką heksastruktu reprezentowaną przez ciąg binarny $B = b_0 b_1 \dots b_{n-2}$, że $v_i, v_{i+1} \in e_i$ dla $i \in \{0, 1, \dots, n-1\}$.

Bezpośrednio z definicji heksastruktu, z definicji obrotu punktu oraz z definicji o ciągu binarnym reprezentującym ścieżkę heksastruktu, otrzymujemy następujące, rekurencyjny i nierekurencyjny, wzory dla współrzędnych kolejnych wierzchołków ścieżki heksastruktu:

Twierdzenie A.1.1. *Jeżeli $k \in \{2, 3, \dots, n\}$, to*

$$v_k = \text{Rot}^{2b_{k-2}-1}(120^\circ) \cdot (v_{k-2} - v_{k-1}) + v_{k-1}.$$

Twierdzenie A.1.2. *Jeżeli $k \in \{2, 3, \dots, n\}$, to*

$$v_k = (I - M_{k-2}) \cdot v_1 + M_{k-2} \cdot v_0,$$

gdzie $M: \{0, 1, \dots, n-2\} \rightarrow \mathbb{R}^{2 \times 2}$ dana jest wzorem:

$$M_m = \sum_{i=0}^m (-1)^i \prod_{j=0}^i \text{Rot}^{2b_{j-1}}(120^\circ). \quad (1.1.1)$$

Dowód. Dowód przeprowadzimy indukcją strukturalną ze względu na indeks rozważanego wierzchołka ścieżki.

Przypadek bazowy: Jeżeli $k = 2$, to

$$\begin{aligned} v_2 &= R_0 \cdot (v_0 - v_1) + v_1 \\ &= (I - R_0) \cdot v_1 + R_0 \cdot v_0 \\ &= (I - M_0) \cdot v_1 + M_0 \cdot v_0. \end{aligned}$$

Założenie indukcyjne: Załóżmy, że

$$v_k = (I - M_{k-2}) \cdot v_1 + M_{k-2} \cdot v_0$$

dla wszystkich $k \in \{2, 3, \dots, n-1\}$.

Krok indukcyjny: Niech $R_k = \text{Rot}^{2b_{k-1}}(120^\circ)$ dla $k \in \{0, 1, \dots, n-2\}$. Na mocy twierdzenia A.1.1 mamy:

$$\begin{aligned} v_n &= R_{n-2} \cdot (v_{n-2} - v_{n-1}) + v_{n-1} \\ &= (I - R_{n-2}) \cdot v_{n-1} + R_{n-1} \cdot v_{n-2} \\ &= (I - R_{n-2}) \cdot ((I - M_{n-3}) \cdot v_1 + M_{n-3} \cdot v_0) + R_{n-2} \cdot ((I - M_{n-4}) \cdot v_1 + M_{n-4} \cdot v_0) \\ &= (I - M_{n-3} - R_{n-2} + R_{n-2} \cdot M_{n-3} + R_{n-2} - R_{n-2} \cdot M_{n-4}) \cdot v_1 \\ &\quad + (M_{n-3} - R_{n-2} \cdot M_{n-3} + R_{n-2} \cdot M_{n-4}) \cdot v_0. \end{aligned}$$

Zauważmy, że

$$M_{n-3} = M_{n-4} + (-1)^{n-3} \prod_{j=0}^{n-3} R_j,$$

a stąd

$$\begin{aligned} v_n &= \left(I - M_{n-3} + R_{n-2} \cdot (-1)^{n-3} \prod_{j=0}^{n-3} R_j \right) \cdot v_1 + \left(M_{n-3} - R_{n-2} \cdot (-1)^{n-3} \prod_{j=0}^{n-3} R_j \right) \cdot v_0 \\ &= \left(I - M_{n-3} - (-1)^{n-2} \prod_{j=0}^{n-2} R_j \right) \cdot v_1 + \left(M_{n-3} + (-1)^{n-2} \prod_{j=0}^{n-2} R_j \right) \cdot v_0 \\ &= (I - M_{n-2}) \cdot v_1 + M_{n-2} \cdot v_0. \end{aligned}$$

□

Wniosek A.1.3. Jeżeli $k \in \{2, 3, \dots, n\}$, to $v_k = v_1 + M_{k-2} \cdot (v_0 - v_1)$.

Przykład A.1.4. Rozważamy ścieżkę heksastruktu $p = \{v_0, v_1, \dots, v_7\}$ reprezentowaną przez ciąg binarny 111000 oraz wierzchołki początkowe $v_0 = [1 \ 1]^T$ oraz $v_1 = [1 \ 2]^T$. Obliczmy współrzędne wierzchołków tej ścieżki (por. rys. A.1).

Odwzorowanie Rot jest cykliczne, więc mamy następujące 3 wartości dla kąta 120° :

$$\text{Rot}(120^\circ) = \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}, \quad \text{Rot}^2(120^\circ) = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix}, \quad \text{Rot}^3(120^\circ) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I.$$

Wówczas otrzymujemy następujące wartości odwzorowania M danego wzorem (1.1.1):

$$M_0 = \sum_{i=0}^0 (-1)^i \prod_{j=0}^i \text{Rot}^{2b_j-1}(120^\circ) = \text{Rot}(120^\circ) = \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix},$$

$$M_1 = M_0 + (-1) \prod_{j=0}^1 \text{Rot}^{2b_j-1}(120^\circ) = M_0 - \text{Rot}^2(120^\circ) = \begin{bmatrix} 0 & -\sqrt{3} \\ \sqrt{3} & 0 \end{bmatrix},$$

$$M_2 = M_1 + (-1)^2 \prod_{j=0}^2 \text{Rot}^{2b_j-1}(120^\circ) = M_1 + \text{Rot}^3(120^\circ) = \begin{bmatrix} 1 & -\sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix},$$

$$M_3 = M_2 + (-1)^3 \prod_{j=0}^3 \text{Rot}^{2b_j-1}(120^\circ) = M_2 - \text{Rot}^2(120^\circ) = \begin{bmatrix} \frac{3}{2} & -\frac{3\sqrt{3}}{2} \\ \frac{3\sqrt{3}}{2} & \frac{3}{2} \end{bmatrix},$$

$$M_4 = M_3 + (-1)^4 \prod_{j=0}^4 \text{Rot}^{2b_j-1}(120^\circ) = M_3 + \text{Rot}(120^\circ) = \begin{bmatrix} 1 & -2\sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix},$$

$$M_5 = M_4 + (-1)^5 \prod_{j=0}^5 \text{Rot}^{2b_j-1}(120^\circ) = M_4 - \text{Rot}^0(120^\circ) = \begin{bmatrix} 0 & -2\sqrt{3} \\ \sqrt{3} & 0 \end{bmatrix}.$$

Stąd otrzymujemy następujące współrzędne wierzchołków ścieżki p :

$$v_2 = v_1 + M_0 \cdot (v_0 - v_1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & -\frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} \frac{\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 + \frac{\sqrt{3}}{2} \\ \frac{5}{2} \end{bmatrix},$$

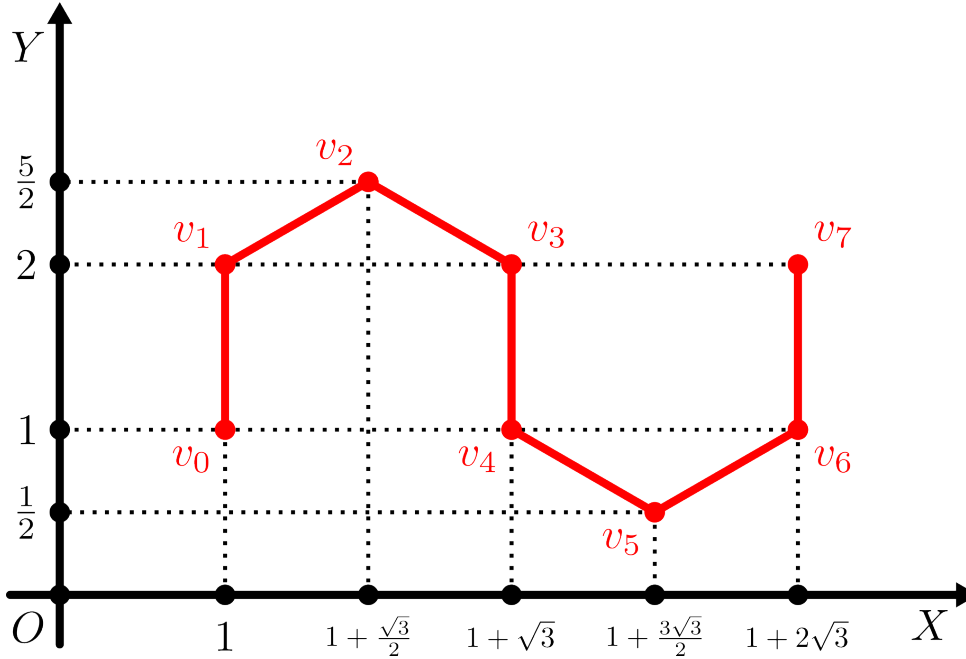
$$v_3 = v_1 + M_1 \cdot (v_0 - v_1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 & -\sqrt{3} \\ \sqrt{3} & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} \sqrt{3} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 + \sqrt{3} \\ 2 \end{bmatrix},$$

$$v_4 = v_1 + M_2 \cdot (v_0 - v_1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 & -\sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} \sqrt{3} \\ -1 \end{bmatrix} = \begin{bmatrix} 1 + \sqrt{3} \\ 1 \end{bmatrix},$$

$$v_5 = v_1 + M_3 \cdot (v_0 - v_1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} \frac{3}{2} & -\frac{3\sqrt{3}}{2} \\ \frac{3\sqrt{3}}{2} & \frac{3}{2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} \frac{3\sqrt{3}}{2} \\ -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} 1 + \frac{3\sqrt{3}}{2} \\ \frac{1}{2} \end{bmatrix},$$

$$v_6 = v_1 + M_4 \cdot (v_0 - v_1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 & -2\sqrt{3} \\ \sqrt{3} & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 2\sqrt{3} \\ -1 \end{bmatrix} = \begin{bmatrix} 1 + 2\sqrt{3} \\ 1 \end{bmatrix},$$

$$v_7 = v_1 + M_5 \cdot (v_0 - v_1) = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 & -2\sqrt{3} \\ \sqrt{3} & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 2\sqrt{3} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 + 2\sqrt{3} \\ 2 \end{bmatrix}.$$



Rysunek A.1: Ścieżka heksastruktury $\{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ reprezentowana przez ciąg binarny 111000 o wierzchołkach początkowych $v_0 = [1 \ 1]^T$ oraz $v_1 = [1 \ 2]^T$ zaznaczona kolorem czerwonym przedstawiona w kartezjańskim układzie współrzędnych.

Wniosek A.1.5. Jeżeli $k \in \{2, 3, \dots, n\}$ oraz $v_1 = [0 \ 0]^T$, to $v_k = M_{k-2} \cdot v_0$.

Obliczanie funkcji M w twierdzeniu A.1.2 możemy zoptymalizować poprzez redukcję liczby iloczynów i sum macierzy. Jest to bardzo istotne w implementacjach komputerowych, ponieważ pozwala na zmniejszenie zapotrzebowania pamięciowego ze względu na ograniczenie liczby niezbędnych do obliczeń utworzonych obiektów (macierzy) oraz na zwiększenie efektywności obliczeniowa dzięki redukcji liczby iloczynów macierzy.

Twierdzenie A.1.6. Jeżeli R_0, R_1, \dots, R_{n-1} jest ciągiem macierzy $\text{Rot}(\pm 120^\circ)$ oraz odwzorowanie $R^*: \mathbb{R}^{2 \times 2} \rightarrow \mathbb{R}^{2 \times 2}$ dane jest wzorem:

$$R_k^* = \begin{cases} I - R_k, & k = n - 1, \\ I - R_k \cdot R_{k+1}^*, & k \in \{0, 1, \dots, n - 2\}, \end{cases}$$

to dla $m \in \{0, 1, \dots, n - 1\}$ spełniona jest równość:

$$R_m^* = I - \sum_{i=m}^{n-1} (-1)^{i+m} \prod_{j=m}^i R_j.$$

Dowód. Dowód przeprowadzimy indukcją strukturalną ze względu na liczbę rozważanych macierzy.

Przypadek bazowy: Jeżeli rozważamy dokładnie jedną macierz, tj. $n = 0$, to

$$R_0^* = I - R_0 = I - \sum_{i=0}^0 (-1)^i \prod_{j=0}^i R_j.$$

Założenie indukcyjne: Zakładamy, że

$$R_m^* = I - \sum_{i=m}^{n-1} (-1)^{i+m} \prod_{j=m}^i R_j$$

dla wszystkich $m \in \{1, 2, \dots, n-1\}$.

Krok indukcyjny: Niech $n \in \mathbb{N}$. Wówczas

$$\begin{aligned} R_0^* &= I - R_0 R_1^* \\ &= I - R_0 \cdot \left(I - \sum_{i=1}^{n-1} (-1)^{i+1} \prod_{j=1}^i R_j \right) \\ &= I - \left(R_0 - R_0 \cdot \sum_{i=1}^{n-1} (-1)^{i+1} \prod_{j=1}^i R_j \right) \\ &= I - \left(R_0 - \sum_{i=1}^{n-1} (-1)^{i+1} \prod_{j=0}^i R_j \right) \\ &= I - \left(R_0 + \sum_{i=1}^{n-1} (-1)^i \prod_{j=0}^i R_j \right) \\ &= I - \sum_{i=0}^{n-1} (-1)^i \prod_{j=0}^i R_j. \end{aligned}$$

□

Uwzględniając występowanie funkcji rotacji dla dokładnie dwóch kątów, 120° i -120° , możemy uprościć obliczanie iloczynu tych macierzy, a tym samym obliczanie współrzędnych wierzchołków heksastrukt. Na mocy twierdzenia 1.4.1 o zależnościach między funkcjami Rot i Ref oraz wniosku 1.2.3 o okresowości macierzy rotacji otrzymujemy następującą postać iloczynu ciągu macierzy rotacji dla kątów $\pm 120^\circ$:

Twierdzenie A.1.7. Niech $\mathcal{R} = (R_0, R_1, \dots, R_{n-1})$ będzie ciągiem macierzy $\text{Rot}(\pm 120^\circ)$. Jeżeli

$$S = |\{R \in \mathcal{R} : R = \text{Rot}(120^\circ)\}| - |\{R \in \mathcal{R} : R = \text{Rot}(-120^\circ)\}|,$$

to

$$\prod_{i=0}^{n-1} R_i = f(S) \cdot I - f(S+1) \cdot \text{Rot}(120^\circ),$$

gdzie $f(m) = 1 - (m \bmod 3)$ dla $m \in \mathbb{Z}$.

Dowód. Dowód przeprowadzimy z wykorzystaniem indukcji matematycznej ze względu na liczbę rozważanych macierzy.

Przypadek bazowy: Niech $\mathcal{R} = (R_0)$. Jeżeli $R_0 = \text{Rot}(120^\circ)$, to $S = 1$, $f(S) = 0$, $f(S + 1) = -1$ oraz

$$\begin{aligned} \prod_{i=0}^0 R_i &= \text{Rot}(120^\circ) \\ &= f(S) \cdot I - f(S + 1) \cdot \text{Rot}(120^\circ). \end{aligned}$$

Jeżeli $R_0 = \text{Rot}(-120^\circ)$, to $S = -1$, $f(S) = -1$, $f(S + 1) = 1$ oraz

$$\begin{aligned} \prod_{i=0}^0 R_i &= \text{Rot}(-120^\circ) \\ &= -I - \text{Rot}(120^\circ) \\ &= f(S) \cdot I - f(S + 1) \cdot \text{Rot}(120^\circ). \end{aligned}$$

Założenie indukcyjne: Niech $\mathcal{R} = (R_0, R_1, \dots, R_{n-2})$ dla $n \in \mathbb{N}$ i niech

$$S = |\{R \in \mathcal{R} : R = \text{Rot}(120^\circ)\}| - |\{R \in \mathcal{R} : R = \text{Rot}(-120^\circ)\}|.$$

Zakładamy, że

$$\prod_{i=0}^{n-2} R_i = f(S) \cdot I - f(S + 1) \cdot \text{Rot}(120^\circ).$$

Krok indukcyjny: Zauważmy, że $f(k) + f(k + 1) = -f(k + 2)$ dla każdego $k \in \mathbb{Z}$. Jeżeli $R_{n-1} = \text{Rot}(120^\circ)$, to $S' = S + 1$, $f(S') = f(S + 1)$, $f(S' + 1) = f(S + 2)$ oraz

$$\begin{aligned} \prod_{i=0}^{n-1} R_i &= \text{Rot}(120^\circ) \cdot \prod_{i=0}^{n-2} R_i \\ &= \text{Rot}(120^\circ) \cdot (f(S) \cdot I - f(S + 1) \cdot \text{Rot}(120^\circ)) \\ &= f(S) \cdot \text{Rot}(120^\circ) - f(S + 1) \cdot (-I - \text{Rot}(120^\circ)) \\ &= f(S + 1) \cdot I + (f(S) + f(S + 1)) \cdot \text{Rot}(120^\circ) \\ &= f(S + 1) \cdot I - f(S + 2) \cdot \text{Rot}(120^\circ) \\ &= f(S') \cdot I - f(S' + 1) \cdot \text{Rot}(120^\circ) \end{aligned}$$

Jeżeli $R_{n-1} = \text{Rot}(-120^\circ)$, to $S' = S - 1$, $f(S') = f(S - 1)$, $f(S' + 1) = f(S)$ oraz

$$\begin{aligned} \prod_{i=0}^{n-1} R_i &= \text{Rot}(-120^\circ) \cdot \prod_{i=0}^{n-2} R_i \\ &= \text{Rot}(-120^\circ) \cdot (f(S) \cdot I - f(S + 1) \cdot \text{Rot}(120^\circ)) \\ &= f(S) \cdot (-I - \text{Rot}(120^\circ)) - f(S + 1) \cdot I \\ &= (-f(S) - f(S + 1)) \cdot I - f(S) \cdot \text{Rot}(120^\circ) \\ &= f(S') \cdot I - f(S' + 1) \cdot \text{Rot}(120^\circ) \end{aligned}$$

□

Współrzędne wierzchołków heksastruktury możemy teraz obliczyć z wykorzystaniem ciągu binarnego reprezentującego ścieżkę o ustalonej krawędzi początkowej z rozważanym wierzchołkiem. Na mocy twierdzenia A.1.7, zliczamy w badanym ciągu binarnym zera, co odpowiada liczbie macierzy rotacji dla kąta -120° , oraz zliczamy jedynki, co odpowiada liczbie macierzy rotacji dla kąta 120° . Podstawiając otrzymany wzór na iloczyn macierzy we wniosku A.1.3 otrzymujemy następujący wzór na współrzędne wierzchołków heksastruktury:

Twierdzenie A.1.8. *Niech $f(i) = 1 - (i \bmod 3)$ dla $i \in \mathbb{Z}$ oraz $s(m) = \sum_{i=0}^m (2b_i - 1)$ dla $m \in \{0, 1, \dots, n-1\}$. Jeżeli $k \in \{2, 3, \dots, n\}$, to*

$$v_k = v_1 + (\iota(k) \cdot I + \rho(k) \cdot \text{Rot}(120^\circ)) \cdot (v_0 - v_1), \quad (1.1.2)$$

gdzie

$$\iota(k) = \sum_{i=0}^{k-2} (-1)^i \cdot f(s(i)) \quad (1.1.3)$$

oraz

$$\rho(k) = \sum_{i=0}^{k-2} (-1)^{i+1} \cdot f(s(i) + 1). \quad (1.1.4)$$

Wniosek A.1.9. *Jeżeli w twierdzeniu A.1.8 przyjmiemy $v_1 = [0 \ 0]^\top$, to*

$$v_k = (\iota(k) \cdot I + \rho(k) \cdot \text{Rot}(120^\circ)) \cdot v_0. \quad (1.1.5)$$

Na mocy powyżej otrzymanych wyników, położenie wierzchołka v heksastruktury możemy określić współczynnikami stojącymi przy I oraz $\text{Rot}(120^\circ)$. Takie podejście zdecydowanie uprości obliczenia i prowadzenie badań ze względu na obecność wyłącznie liczb całkowitych zamiast rzeczywistym w euklidesowym układzie współrzędnych. Bez straty ogólności przyjmujemy, że współrzędne wierzchołków będziemy przedstawiali względem krawędzi ($[1 \ 0]^\top$, $[0 \ 0]^\top$). W przeciwnym razie możemy obrócić, przesunąć i przeskalować współrzędne wszystkich wierzchołków zachowując kształt heksastruktury. Powiemy, że wierzchołek heksastruktury ma współrzędne heksagonalne (ι, ρ) , gdy

$$v = (\iota \cdot I + \rho \cdot \text{Rot}(120^\circ)) \cdot [1 \ 0]^\top.$$

Przykład A.1.10. *Rozważamy heksastrukt H reprezentowany przez ciąg binarny 11111. W tabeli A.1 przedstawiliśmy obliczenia pośrednich wartości funkcji wykorzystanych w twierdzeniu A.1.8 do obliczenia współrzędnych kolejnych wierzchołków heksastruktury H z założeniem, że pierwsze dwa wierzchołki mają współrzędne, odpowiednio, $[1 \ 0]^\top$ oraz $[0 \ 0]^\top$. Otrzymane wyniki zestawiliśmy w tabeli A.2.*

Twierdzenie A.1.11. *Niech $n \in \mathbb{N}$ i niech $p' = (e'_0, e'_1, \dots, e'_n)$ będzie taką ścieżką tworzącą graf G , że $e'_k \in \mathbb{R}^2$ oraz $v'_k, v'_{k+1} \in e'_k$ dla $k \in \{0, 1, \dots, n\}$. Jeżeli:*

(i) $v'_1 = [0 \ 0]^\top$,

(ii) v'_k są postaci 1.1.5 dla wszystkich $k \in \{0, 1, \dots, n+1\}$, dla pewnego ciągu binarnego B' ,

to G jest heksastruktem reprezentowanym przez B' .

| k | 0 | 1 | 2 | 3 | 4 |
|--------------------------------|----|----|---|----|----|
| $s(k)$ | 1 | 2 | 3 | 4 | 5 |
| $f(s(k))$ | 0 | -1 | 1 | 0 | -1 |
| $(-1)^k \cdot f(s(k))$ | 0 | 1 | 1 | 0 | -1 |
| $f(s(k) + 1)$ | -1 | 1 | 0 | -1 | 1 |
| $(-1)^{k+1} \cdot f(s(k) + 1)$ | 1 | 1 | 0 | -1 | -1 |

Tabela A.1: Pośrednie wartości funkcji wykorzystanych w twierdzeniu A.1.8 dla heksastruktu reprezentowanego przez ciąg binarny 11111.

| k | 2 | 3 | 4 | 5 | 6 |
|------------|---|---|---|---|---|
| $\iota(k)$ | 0 | 1 | 2 | 2 | 1 |
| $\rho(k)$ | 1 | 2 | 2 | 1 | 0 |

Tabela A.2: Wartości funkcji ι i ρ o wzorach przedstawionych w twierdzeniu A.1.8 dla heksastruktu reprezentowanego przez ciąg binarny 11111.

Przykład A.1.12. Niech $n \in \mathbb{N}$. Rozważamy ciąg binarny B o długości $2n$, gdzie $b_{2i} = 0$ oraz $b_{2i+1} = 1$ dla $i \in \{0, 1, \dots, n-1\}$. Pokażemy, że B reprezentuje pewien heksastrukt H . Zauważmy, że

$$s(m) = \begin{cases} -1, & m \equiv 0 \pmod{2}, \\ 0, & m \equiv 1 \pmod{2}, \end{cases}$$

$$f(s(m)) = \begin{cases} -1, & m \equiv 0 \pmod{2}, \\ 1, & m \equiv 1 \pmod{2}, \end{cases}$$

oraz

$$f(s(m) + 1) = \begin{cases} 1, & m \equiv 0 \pmod{2}, \\ 0, & m \equiv 1 \pmod{2}, \end{cases}$$

dla $m \in \{0, 1, \dots, 2n-1\}$. Stąd $\iota(m) = -m - 3$ i $\rho(m) = \lfloor \frac{m}{2} \rfloor$ dla $m \in \{2, 3, \dots, 2n-1\}$. Zatem ciąg $S = ([\iota(m) \ \rho(m)]^T)_{m \in \{2, 3, \dots, 2n-1\}}$ jest różnowartościowy oraz $[0 \ 0]^T, [1 \ 0]^T \notin S$. Ostatecznie ciąg $p' = (e'_0, e'_1, \dots, e'_{2n})$, gdzie $(\iota(k), \rho(k)), (\iota(k+1), \rho(k+1)) \in e'_k$ dla $k \in \{0, 1, \dots, 2n\}$, tworzy ścieżkę. Na mocy twierdzenia A.1.11, istnieje heksastrukt utworzony ścieżką p' reprezentowany przez B .

Przykład A.1.13. Rozważamy ciąg binarny B długości $4n$ dla pewnego $n \in \mathbb{N}$, gdzie $b_{4i} = 0 = b_{4i+1}$ oraz $b_{4i+2} = 1 = b_{4i+3}$ dla $i \in \{0, 1, \dots, n-1\}$. Pokażemy, że B reprezentuje pewien heksastrukt H . Rozważamy współrzędne wierzchołków zgodnie z twierdzeniem A.1.11. Zauważmy, że

$$s(m) = \begin{cases} -1, & m \equiv 0 \pmod{2}, \\ -2, & m \equiv 1 \pmod{4}, \\ 0, & m \equiv 3 \pmod{4}, \end{cases}$$

dla $m \in \{0, 1, \dots, 4n-1\}$. Stąd

$$f(s(m)) = \begin{cases} -1, & m \equiv 0 \pmod{2}, \\ 0, & m \equiv 1 \pmod{4}, \\ 1, & m \equiv 3 \pmod{4}, \end{cases}$$

oraz

$$f(s(m)+1) = \begin{cases} 1, & m \equiv 0 \pmod{2}, \\ -1, & m \equiv 1 \pmod{4}, \\ 2, & m \equiv 3 \pmod{4}, \end{cases}$$

dla $m \in \{0, 1, \dots, 4n-1\}$. Zatem mamy następujące wzory na składowe przy I i $\text{Rot}(120^\circ)$ (por. <https://oeis.org/A037915> oraz <https://oeis.org/A110657>):

$$\iota(m) = - \left\lfloor \frac{3m-2}{4} \right\rfloor,$$

$$\rho(m) = \left\lfloor \frac{m+2}{4} \right\rfloor + ((m+2 \bmod 4) \bmod 3),$$

dla $m \in \{0, 1, \dots, 4n-1\}$. Jeżeli $m \not\equiv 0 \pmod{4}$, to $\iota(m) < \iota(m+1)$. W przeciwnym przypadku $\iota(m) = \iota(m+1)$, ale wówczas $\rho(m) < \rho(m+1)$. Zatem współrzędne $(\iota(m), \rho(m))$ tworzą silnie rosnący różnowartościowy ciąg względem porządku leksykograficznego dla $m \in \{0, 1, \dots, 4n-1\}$. Czyli istnieje heksastrukt reprezentowany przez ciąg B .

Lemat A.1.14. Niech $z \in \mathbb{Z}$ oraz niech

$$f(n) = (-1)^n((z+1) \bmod 3) - (z \bmod 3)$$

dla $n \in \mathbb{N}$. Wówczas

$$f(n) \bmod 3 = \begin{cases} 1, & n \in 2\mathbb{N}, \\ 2, & n \in 2\mathbb{N} + 1. \end{cases}$$

Twierdzenie A.1.15. Jeżeli para $(\iota, \rho) \in \mathbb{Z}$ jest współrzędną heksagonalną wierzchołka, to $\iota + \rho \bmod 3 < 2$.

Dowód. Niech p będzie ścieżką heksastruktu H reprezentowaną przez ciąg binarny $B = b_0b_1 \dots b_n$, gdzie $n \in \mathbb{N}$. Na mocy twierdzenia A.1.8, wierzchołki heksastruktu H mają współrzędne $(\iota, \rho) \in \mathbb{Z}$, gdzie

$$\iota = \sum_{i=0}^{k-2} (-1)^i \cdot (1 - (s(i) \bmod 3))$$

oraz

$$\rho = - \sum_{i=0}^{k-2} (-1)^i \cdot (1 - (s(i) + 1) \bmod 3),$$

dla $k \in \{2, 3, \dots, n+1\}$ oraz

$$s(m) = \sum_{i=0}^m (2b_i - 1)$$

dla $m \in \{0, 1, \dots, n-1\}$. Wówczas

$$\begin{aligned}\iota + \rho &= \sum_{i=0}^{k-2} (-1)^i \cdot (1 - (s(i) \bmod 3)) - \sum_{i=0}^{k-2} (-1)^i \cdot (1 - (s(i) + 1) \bmod 3) \\ &= \sum_{i=0}^{k-2} (-1)^i \cdot (((s(i) + 1) \bmod 3) - (s(i) \bmod 3)).\end{aligned}$$

Na mocy lematu A.1.14 oraz własności operatora modulo, jeżeli $k \in 2\mathbb{N}$, to

$$(\iota + \rho) \bmod 3 = \left(3 \cdot \frac{k-2+2}{2} + 1\right) \bmod 3 = 1.$$

Jeżeli $k \in 2\mathbb{N} + 1$, to

$$(\iota + \rho) \bmod 3 = \left(3 \cdot \frac{k-2+1}{2}\right) \bmod 3 = 0.$$

□

Heksagonalnym układem współrzędnych nazywamy układ współrzędny wyznaczony przez współrzędne heksagonalne wszystkich wierzchołków, jakie mogą tworzyć heksa-strukt, tj. jest to taki zbiór punktów $(\iota, \rho) \in \mathbb{Z}^2$, że $\iota + \rho \bmod 3 < 2$ (por. rys. A.2).

Twierdzenie A.1.16. *Jeżeli $k \in \{2, 3, \dots, n\}$ oraz wierzchołki v_0 i v_1 mają współrzędne heksagonalne, odpowiednio, (ι'_0, ρ'_0) i (ι'_1, ρ'_1) , to v_k ma współrzędne heksagonalne (ι'_k, ρ'_k) , gdzie*

$$\begin{cases} \iota'_k = \iota'_1 + \iota(k)(\iota'_0 - \iota'_1) - \rho(k)(\rho'_0 - \rho'_1), \\ \rho'_k = \rho'_1 + \iota(k)(\rho'_0 - \rho'_1) + \rho(k)((\iota'_0 - \iota'_1) - (\rho'_0 - \rho'_1)). \end{cases}$$

Dowód. Niech $k \in \{2, 3, \dots, n\}$ oraz niech v_0 i v_1 mają współrzędne heksagonalne, odpowiednio, (ι'_0, ρ'_0) i (ι'_1, ρ'_1) . Z definicji współrzędnych heksagonalnych, mamy:

$$v_0 = (\iota'_0 \cdot I + \rho'_0 \cdot \text{Rot}(120^\circ)) \cdot [1 \ 0]^\top$$

oraz:

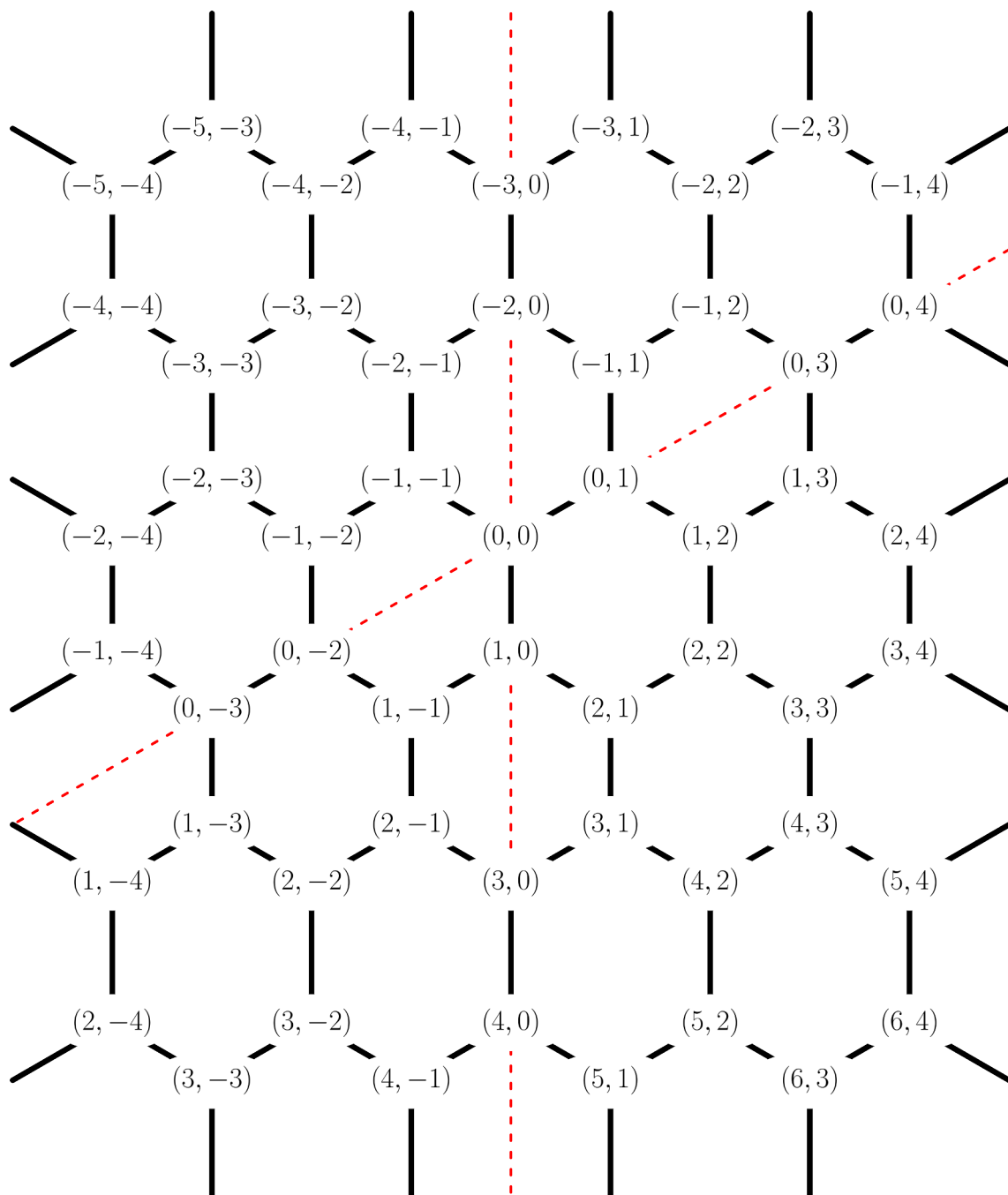
$$v_1 = (\iota'_1 \cdot I + \rho'_1 \cdot \text{Rot}(120^\circ)) \cdot [1 \ 0]^\top.$$

Stąd

$$v_0 - v_1 = ((\iota'_0 - \iota'_1) \cdot I + (\rho'_0 - \rho'_1) \cdot \text{Rot}(120^\circ)) \cdot [1 \ 0]^\top.$$

Zauważmy, że $\text{Rot}^2(120^\circ) = -I - \text{Rot}(120^\circ)$, a wówczas

$$\begin{aligned}& (\iota(k) \cdot I + \rho(k) \cdot \text{Rot}(120^\circ)) \cdot ((\iota'_0 - \iota'_1) \cdot I + (\rho'_0 - \rho'_1) \cdot \text{Rot}(120^\circ)) \\ &= (\iota(k) \cdot (\iota'_0 - \iota'_1) - \rho(k) \cdot (\rho'_0 - \rho'_1)) \cdot I \\ &\quad + (\iota(k) \cdot (\rho'_0 - \rho'_1) + \rho(k) \cdot (\iota'_0 - \iota'_1) - \rho(k) \cdot (\rho'_0 - \rho'_1)) \cdot \text{Rot}(120^\circ) \quad (1.1.6) \\ &= (\iota(k) \cdot (\iota'_0 - \iota'_1) - \rho(k) \cdot (\rho'_0 - \rho'_1)) \cdot I \\ &\quad + (\iota(k)(\rho'_0 - \rho'_1) + \rho(k)((\iota'_0 - \iota'_1) - (\rho'_0 - \rho'_1))) \cdot \text{Rot}(120^\circ)\end{aligned}$$



Rysunek A.2: Fragment heksagonalnego układu współrzędnych, gdzie pierwsza składowa współrzędnej jest współczynnikiem stojącym przy I , a druga składowa współrzędnej jest współczynnikiem stojącym przy $\text{Rot}(120^\circ)$ we wzorze (1.1.2) na obliczanie współrzędnych wierzchołków heksastruktów ze względu na ustaloną parę wierzchołków $(1, 0)$ i $(0, 0)$ połączonych krawędzią. Na czerwono zaznaczono osie, tj. linie, dla których pierwsza współrzędna lub druga współrzędna jest równa zero.

Na mocy wzoru (1.1.2) oraz równania (1.1.6), mamy:

$$\begin{aligned}
v_k &= v_1 + (\iota(k) \cdot I + \rho(k) \cdot \text{Rot}(120^\circ)) \cdot (v_0 - v_1) \\
&= (\iota'_1 \cdot I + \rho'_1 \cdot \text{Rot}(120^\circ)) \cdot [1 \ 0]^\top \\
&\quad + (\iota(k) \cdot (\iota'_0 - \iota'_1) - \rho(k) \cdot (\rho'_0 - \rho'_1)) \cdot I \cdot [1 \ 0]^\top \\
&\quad + (\iota(k)(\rho'_0 - \rho'_1) + \rho(k)((\iota'_0 - \iota'_1) - (\rho'_0 - \rho'_1))) \cdot \text{Rot}(120^\circ) \cdot [1 \ 0]^\top \\
&= (\iota'_1 + \iota(k) \cdot (\iota'_0 - \iota'_1) - \rho(k) \cdot (\rho'_0 - \rho'_1)) \cdot I \cdot [1 \ 0]^\top \\
&\quad + (\rho'_1 + \iota(k)(\rho'_0 - \rho'_1) + \rho(k)((\iota'_0 - \iota'_1) - (\rho'_0 - \rho'_1))) \cdot \text{Rot}(120^\circ) \cdot [1 \ 0]^\top.
\end{aligned}$$

□

Przykład A.1.17. Niech v_0 i v_1 mają współrzędne heksagonalne, odpowiednio, $(2, 1)$ i $(2, 2)$ (por. rys. A.2). Jeżeli $B = 0$, to, na mocy twierdzenia A.1.16,

$$s(0) = -1,$$

$$\iota(2) = -1,$$

$$\rho(2) = -1,$$

a stąd

$$\iota'_2 = 2 + (-1) \cdot (2 - 2) - (-1) \cdot (1 - 2) = 1,$$

$$\rho'_2 = 2 + (-1) \cdot (1 - 2) + (-1) \cdot ((2 - 2) - (1 - 2)) = 2.$$

Z drugiej strony, jeżeli $B = 1$, to

$$s(0) = 1,$$

$$\iota(2) = 0,$$

$$\rho(2) = 1,$$

a stąd

$$\iota'_2 = 2 + 0 \cdot (2 - 2) - 1 \cdot (1 - 2) = 3,$$

$$\rho'_2 = 2 + 0 \cdot (1 - 2) + 1 \cdot ((2 - 2) - (1 - 2)) = 3.$$

Na zakończenie tego podrozdziału przedstawimy kilka obserwacji dotyczących położenia sąsiadujących z danym wierzchołkiem innych wierzchołków wraz ze wzorem na ich współrzędne.

Wniosek A.1.18. Niech v będzie wierzchołkiem o współrzędnych heksagonalnych (ι, ρ) .

- Jeżeli $\iota + \rho \equiv 0 \pmod{3}$, to istnieją wierzchołki o współrzędnych heksagonalnych: $(\iota, \rho - 1)$, $(\iota + 1, \rho)$ oraz $(\iota - 1, \rho - 1)$.
- Jeżeli $\iota + \rho \equiv 1 \pmod{3}$, to istnieją wierzchołki o współrzędnych heksagonalnych: $(\iota, \rho + 1)$, $(\iota - 1, \rho)$ oraz $(\iota + 1, \rho + 1)$.

Wniosek A.1.19. Niech v_0 , v_1 i v_2 będą wierzchołkami o współrzędnych heksagonalnych, odpowiednio, (ι'_0, ρ'_0) , $(\iota'_0 + i_1, \rho'_0 + r_1)$ i $(\iota'_0 + i_2, \rho'_0 + r_2)$.

- Jeżeli $B = 0$ oraz $(i_1, r_1) = (0, 1)$, to $(i_2, r_2) = (-1, 1)$.
- Jeżeli $B = 0$ oraz $(i_1, r_1) = (0, -1)$, to $(i_2, r_2) = (1, -1)$.
- Jeżeli $B = 0$ oraz $(i_1, r_1) = (1, 0)$, to $(i_2, r_2) = (2, 1)$.
- Jeżeli $B = 0$ oraz $(i_1, r_1) = (-1, 0)$, to $(i_2, r_2) = (-2, -1)$.
- Jeżeli $B = 0$ oraz $(i_1, r_1) = (1, 1)$, to $(i_2, r_2) = (1, 2)$.
- Jeżeli $B = 0$ oraz $(i_1, r_1) = (-1, -1)$, to $(i_2, r_2) = (-1, -2)$.
- Jeżeli $B = 1$ oraz $(i_1, r_1) = (0, 1)$, to $(i_2, r_2) = (1, 2)$.
- Jeżeli $B = 1$ oraz $(i_1, r_1) = (0, -1)$, to $(i_2, r_2) = (-1, -2)$.
- Jeżeli $B = 1$ oraz $(i_1, r_1) = (1, 0)$, to $(i_2, r_2) = (1, -1)$.
- Jeżeli $B = 1$ oraz $(i_1, r_1) = (-1, 0)$, to $(i_2, r_2) = (-1, 1)$.
- Jeżeli $B = 1$ oraz $(i_1, r_1) = (1, 1)$, to $(i_2, r_2) = (2, 1)$.
- Jeżeli $B = 1$ oraz $(i_1, r_1) = (-1, -1)$, to $(i_2, r_2) = (-2, -1)$.

Wzory na współrzędne heksagonalne trzeciego wierzchołka otrzymane dla możliwych wszystkich przypadków ze względu na ciąg binarny i początkowe dwa wierzchołki możemy uprościć następującym odwzorowaniem:

Wniosek A.1.20. *Odwzorowanie $h: \mathbb{Z}^3 \rightarrow \mathbb{Z}^2$ dane wzorem:*

$$h(b, i_1, r_1) = (b(2r_1 - i_1) + 2i_1 - r_1, b(r_1 - 2i_1) + i_1 + r_1),$$

gdzie b jest jedynym bitem rozważanego ciągu binarnego, tj. $b \in \{0, 1\}$, spełnia tezę każdego punktu wniosku A.1.19.

A.2 Walidacja ciągów binarnych jako reprezentantów heksastruktów

Stwierdzenie, czy istnieje heksastrukt reprezentowany przez dany ciąg binarny, jest jednym z większych wyzwaniań niniejszych rozważań. Znalezienie metody pozwalającej na stwierdzenie tego faktu na podstawie samych bitów pozwoli również na opracowanie wzoru na liczbę ciągów binarnych reprezentujących ciągi binarne (oraz pochodnych zagadnień, np. z uwzględnieniem niezmienniczości na rotacje). W niniejszym rozdziale prezentujemy otrzymane sposoby, dzięki którym możemy stwierdzić, że dla danego ciągu binarnego na pewno nie istnieje heksastrukt, który mógłby reprezentować.

Niech $n \in \mathbb{N} \setminus \{1\}$, niech $p = (e_0, e_1, \dots, e_{n-1})$ będzie ścieżką heksastruktu reprezentowaną przez ciąg binarny $B = b_0 b_1 \dots b_{n-2}$, niech $v_k, v_{k+1} \in e_k$ dla $k \in \{0, 1, \dots, n\}$ oraz niech $v_0 = [1 \ 0]^T$ i $v_1 = [0 \ 0]^T$.

Lemat A.2.1. Jeżeli $k \in \mathbb{N}$, to

$$\sum_{i=0}^k (-1)^i = \begin{cases} 0, & k \equiv 0 \pmod{2}, \\ 1, & k \equiv 1 \pmod{2}. \end{cases}$$

Lemat A.2.2. Jeżeli $k \in \mathbb{N}$, to

$$\sum_{i=0}^n (-1)^i (i+1) = \begin{cases} \frac{k}{2} + 1 \pmod{2}, & k \equiv 0, \\ -\frac{k+1}{2}, & k \equiv 1 \pmod{2}. \end{cases}$$

Lemat A.2.3. Jeżeli $k \in 2\mathbb{N}$, to

$$\sum_{i=0}^n (-1)^i \sum_{j=0}^i b_j = \begin{cases} b_0 + b_2 + \dots + b_n, & k \equiv 0, \\ -(b_1 + b_3 + \dots + b_n), & k \equiv 1 \pmod{2}. \end{cases}$$

Twierdzenie A.2.4. Wierzchołki \tilde{p} i \tilde{p} mają współrzędne $v_0 = [1 \ 0]^T$ wtedy i tylko wtedy, gdy $n \in 2\mathbb{N}$ i spełnione są równości:

$$\begin{cases} -2(b_0 + b_2 + \dots + b_n) + \left(\frac{n}{2} + 1\right) + 3 \sum_{i=0}^n (-1)^i C_i = 0, \\ -2(b_0 + b_2 + \dots + b_n) + \left(\frac{n}{2} + 1\right) + 3 \sum_{i=0}^n (-1)^i C'_i = 0, \end{cases}$$

gdzie $s(i) = \sum_{j=0}^i 2b_j - 1$, $C_i = \left\lfloor \frac{s(i)}{3} \right\rfloor$ oraz $C'_i = \left\lfloor \frac{s(i)+1}{3} \right\rfloor$ dla $i \in \{0, 1, \dots, n-2\}$.

Dowód. Na mocy twierdzenia A.1.8, spełnione są równości:

$$\begin{cases} \sum_{i=0}^n (-1)^i \cdot (1 - (s(i) \bmod 3)) = 1, \\ \sum_{i=0}^n (-1)^i \cdot (1 - ((s(i) + 1) \bmod 3)) = 0. \end{cases} \quad (1.2.7)$$

Jeżeli $i \in \{0, 1, \dots, n\}$, to $s(i) \bmod 3 = s(i) - 3C_i$ oraz $(s(i) + 1) \bmod 3 = s(i) - 3C'_i$. Zatem

$$\begin{aligned} 1 &= \sum_{i=0}^n (-1)^i \cdot (1 - (s(i) \bmod 3)) \\ &= \sum_{i=0}^n (-1)^i \cdot (1 - (s(i) - 3C_i)) \\ &= \sum_{i=0}^n (-1)^i - \sum_{i=0}^n (-1)^i s(i) + 3 \sum_{i=0}^n (-1)^i C_i \\ &= \sum_{i=0}^n (-1)^i - \sum_{i=0}^n (-1)^i \sum_{j=0}^i (2b_j - 1) + 3 \sum_{i=0}^n (-1)^i C_i \\ &= \sum_{i=0}^n (-1)^i - 2 \sum_{i=0}^n (-1)^i \sum_{j=0}^i b_j + \sum_{i=0}^n (-1)^i (i+1) + 3 \sum_{i=0}^n (-1)^i C_i. \end{aligned}$$

Na mocy lematów A.2.1, A.2.2 i A.2.3,

$$\begin{aligned} 1 &= \sum_{i=0}^n (-1)^i - 2 \sum_{i=0}^n (-1)^i \sum_{j=0}^i b_j + \sum_{i=0}^n (-1)^i (i+1) + 3 \sum_{i=0}^n (-1)^i C_i \\ &= 1 - 2(b_0 + b_2 + \dots + b_n) + \left(\frac{n}{2} + 1\right) + 3 \sum_{i=0}^n (-1)^i C_i. \end{aligned}$$

Zatem

$$0 = -2(b_0 + b_2 + \dots + b_n) + \left(\frac{n}{2} + 1\right) + 3 \sum_{i=0}^n (-1)^i C_i.$$

dla drugiego równania z (1.2.7) mamy:

$$\begin{aligned} 0 &= \sum_{i=0}^n (-1)^i \cdot (1 - ((s(i) + 1) \bmod 3)) \\ &= \sum_{i=0}^n (-1)^i \cdot (1 - (s(i) + 1 - 3C'_i)) \\ &= - \sum_{i=0}^n (-1)^i \cdot (s(i) - 3C'_i) \\ &= -2 \sum_{i=0}^n (-1)^i \sum_{j=0}^i b_j + \sum_{i=0}^n (-1)^i (i + 1) + 3 \sum_{i=0}^n (-1)^i C'_i \\ &= -2(b_0 + b_2 + \dots + b_n) + \left(\frac{n}{2} + 1\right) + 3 \sum_{i=0}^n (-1)^i C'_i. \end{aligned}$$

Załóżmy w drugą stronę, że n jest nieparzyste. Ponownie na mocy lematów A.2.1, A.2.2 i A.2.3 mamy:

$$\begin{aligned} 1 &= \sum_{i=0}^n (-1)^i - 2 \sum_{i=0}^n (-1)^i \sum_{j=0}^i b_j + \sum_{i=0}^n (-1)^i (i + 1) + 3 \sum_{i=0}^n (-1)^i C_i \\ &= 2(b_1 + b_3 + \dots + b_n) - \frac{n+1}{2} + 3 \sum_{i=0}^n (-1)^i C_i \end{aligned}$$

oraz:

$$\begin{aligned} 0 &= -2 \sum_{i=0}^n (-1)^i \sum_{j=0}^i b_j + \sum_{i=0}^n (-1)^i (i + 1) + 3 \sum_{i=0}^n (-1)^i C'_i \\ &= 2(b_1 + b_3 + \dots + b_n) - \frac{n+1}{2} + 3 \sum_{i=0}^n (-1)^i C'_i. \end{aligned}$$

Zatem

$$\begin{cases} (b_1 + b_3 + \dots + b_n) + 2(n+1) \equiv 1 \pmod{3}, \\ (b_1 + b_3 + \dots + b_n) + 2(n+1) \equiv 0 \pmod{3}. \end{cases}$$

Sprzeczność. Liczba n musi być parzysta. □

Zanim wyciągniemy wnioski z twierdzenia A.2.4, zauważmy jedną przydatną równość.

Lemat A.2.5. *Jeżeli $z \in \mathbb{Z}$, to*

$$\left\lfloor \frac{z+1}{3} \right\rfloor - \left\lfloor \frac{z}{3} \right\rfloor = ((z+2) \bmod 3) \bmod 2.$$

Dowód. Jeżeli $z = 3k$ dla pewnego $k \in \mathbb{Z}$, to

$$((z+2) \bmod 3) \bmod 2 = ((3k+2) \bmod 3) \bmod 2 = 0$$

oraz

$$\left\lfloor \frac{z+1}{3} \right\rfloor - \left\lfloor \frac{z}{3} \right\rfloor = \left\lfloor \frac{3k+1}{3} \right\rfloor - \left\lfloor \frac{3k}{3} \right\rfloor = k - k = 0.$$

Jeżeli $z = 3k + 1$ dla pewnego $k \in \mathbb{Z}$, to

$$((z + 2) \bmod 3) \bmod 2 = (3k \bmod 3) \bmod 2 = 0$$

oraz

$$\left\lfloor \frac{z+1}{3} \right\rfloor - \left\lfloor \frac{z}{3} \right\rfloor = \left\lfloor \frac{3k+2}{3} \right\rfloor - \left\lfloor \frac{3k+1}{3} \right\rfloor = k - k = 0.$$

Jeżeli $z = 3k + 2$ dla pewnego $k \in \mathbb{Z}$, to

$$((z + 2) \bmod 3) \bmod 2 = ((3k + 1) \bmod 3) \bmod 2 = 1$$

oraz

$$\left\lfloor \frac{z+1}{3} \right\rfloor - \left\lfloor \frac{z}{3} \right\rfloor = \left\lfloor \frac{3k+3}{3} \right\rfloor - \left\lfloor \frac{3k+2}{3} \right\rfloor = k + 1 - k = 1.$$

□

Wniosek A.2.6. *Jeżeli wierzchołki \tilde{p} i \bar{p} mają współrzędne $(1, 0)$, to:*

1. $(b_0 + b_2 + \dots + b_n) + 2n + 1 \equiv 0 \pmod{3}$,
2. spełnione jest równanie:

$$\sum_{i=0}^n (-1)^i (((s(i) + 2) \bmod 3) \bmod 2) = 0.$$

Powyższe wyniki pomagają nam wykryć kolizje w heksastruktach potencjalnie reprezentowanych przez rozważane ciągi binarne. Jeżeli którykolwiek z warunków we wniosku A.2.6 nie jest spełniony, to początek i koniec ścieżki reprezentowanej przez rozważany ciąg binarny na pewno nie kolidują ze sobą. W drugą stronę niekoniecznie jest to prawdą, co pokazują następujące przykłady:

Przykład A.2.7. *Niech ścieżka p będzie reprezentowana przez ciąg binarny 001. Wówczas spełniona jest pierwsza zależność wniosku A.2.6:*

$$(b_0 + b_2) + 2n + 1 = (0 + 1) + 2 \cdot 2 + 1 = 6 \equiv 0 \pmod{3}.$$

Wierzchołki tej ścieżki mają współrzędne heksagonalne, kolejno, $(1, 0)$, $(0, 0)$, $(-1, -1)$, $(-1, -2)$ i $(-2, -3)$, a zatem początek i koniec tej ścieżki są różne.

Przykład A.2.8. *Niech ścieżka p będzie reprezentowana przez ciąg binarny 1. Wówczas spełniona jest druga zależność wniosku A.2.6:*

$$\sum_{i=0}^n (-1)^i (((s(i) + 2) \bmod 3) \bmod 2) = (((1 + 2) \bmod 3) \bmod 2) = 0.$$

Wierzchołki tej ścieżki mają współrzędne heksagonalne, kolejno, $(1, 0)$, $(0, 0)$ i $(0, 1)$, a zatem początek i koniec tej ścieżki są różne.

A.3 Zliczanie ścieżek rzędu drugiego na siatce heksagonów

Zliczanie heksastruktów będącymi ścieżkami rzędu drugiego jest zagadnieniem kombinatorycznym, które polega na określeniu liczby różnych ścieżek o zadanej złożoności. Reprezentacja takich ścieżek przy pomocy ciągów binarnych wskazuje na wykładniczy wzrost liczby takich struktur ze względu na ich długość. Głównym wyzwaniem jest wykrywanie kolizji dla coraz bardziej złożonych ścieżek. Pierwszy taki przypadek pojawia się dla ścieżek reprezentowanych przez ciągi binarne o długości 5: 00000 oraz 11111 (por. rys. A.3). W niniejszym rozdziale przeprowadzimy analizę zależności liczby ciągów binarnych od ich długości. Rozważymy ciągi binarne bez żadnego kontekstu, jak i ciągi binarne reprezentujące heksastrukty będące ścieżki nieposiadające wierzchołków stopnia trzeciego. Weźmiemy również pod uwagę wpływ rotacji oraz odbić lustrzanych na moc zbiorów utworzonych przez poszczególne ciągi binarne o danej długości.



Rysunek A.3: Ścieżki heksastruktów reprezentowane przez ciągi binarne, odpowiednio od lewej do prawej, 00000 oraz 11111. Na czerwono zaznaczyliśmy wierzchołki oraz krawędzie początkowe. Przerywaną linią zaznaczyliśmy krawędzie powodujące kolizję z wierzchołkami początkowymi.

Rozpocniemy od kilku stwierdzeń zarysowujących kontekst i podejście do powyżej opisanej problematyki. Ich celem jest rozbięcie problemu na mniejsze problemy i ułatwienie analizy.

Twierdzenie A.3.1. *Niech $B \in \mathcal{B}$. Jeżeli $B = \zeta(B)$, to $B \neq \pi(B)$.*

Dowód. Niech $n \in \mathbb{N}$ i niech $B = b_0b_1 \dots b_{n-1}$ będzie takim ciągiem binarnym, że $B = \zeta(B)$. Przypuśćmy, że $B = \pi(B)$. Wówczas $b_0 = 1 - b_{n-1}$. Sprzeczność, gdyż $b_0 = b_{n-1}$. \square

Twierdzenie A.3.2. *Niech $B \in \mathcal{B}$. Jeżeli $B = \pi(B)$, to $B \neq \zeta(B)$.*

Dowód. Niech $n \in \mathbb{N}$ i niech $B = b_0b_1 \dots b_{n-1}$ będzie takim ciągiem binarnym, że $B = \pi(B)$. Przypuśćmy, że $B = \zeta(B)$. Wówczas $b_0 = b_{n-1}$. Sprzeczność, gdyż $b_0 = 1 - b_{n-1}$. \square

Wniosek A.3.3. *Niech $B \in \mathcal{B}$. Jeżeli $B \neq \zeta(B)$ i $B \neq \pi(B)$, to $|\{B, \tau(B), \zeta(B), \pi(B)\}| = 4$.*

W powyższych twierdzeniach pokazujemy, że problem zliczania można podzielić na zliczanie: ciągów binarnych niezmienniczych ze względu na rotację, ciągów niezmienniczych na odwrócenie ciągu oraz ciągów, które nie są ani niezmiennicze ze względu na rotację, ani nie są niezmiennicze ze względu na odwrócenie ciągu. W kolejnych dwóch twierdzeniach prezentujemy jawny wzór na liczbę ciągów binarnych o danej długości z uwzględnieniem niezmienniczości ze względu na, odpowiednio, odwrócenie ciągu i rotację.

Twierdzenie A.3.4. *Jeżeli $n \in \mathbb{N}$, to*

$$|\{B \in \mathcal{B} : |B| = n, B = \varsigma(B)\}| = \begin{cases} 2^{\frac{n}{2}}, & n \equiv 0 \pmod{2}, \\ 2^{\frac{n+1}{2}}, & n \equiv 1 \pmod{2}. \end{cases}$$

Dowód. Niech $n \in \mathbb{N}$, niech $B = b_0b_1 \dots b_{n-1}$ i niech

$$N = |\{B \in \mathcal{B} : |B| = n, B = \varsigma(B)\}|.$$

Wówczas $B = \pi(B)$ wtedy i tylko wtedy, gdy $b_i = b_{n-1-i}$ dla $i \in \{0, 1, \dots, \lfloor \frac{n-1}{2} \rfloor\}$. Jeżeli $n \equiv 0 \pmod{2}$, to $i \neq n-1-i$ dla wszystkich $i \in \{0, 1, \dots, \frac{n}{2}-1\}$. Wtedy wstawiamy bit na $\frac{n}{2}$ indeksach, zatem $N = 2^{\frac{n}{2}}$. Jeżeli $n \equiv 1 \pmod{2}$, to $n-1 - \frac{n-1}{2} = \frac{n-1}{2}$. Wtedy wstawiamy bit na $\frac{n-1}{2} + 1 = \frac{n+1}{2}$ indeksach, zatem $N = 2^{\frac{n+1}{2}}$. \square

Twierdzenie A.3.5. *Jeżeli $n \in \mathbb{N}$, to*

$$|\{B \in \mathcal{B} : |B| = n, B = \pi(B)\}| = \begin{cases} 2^{\frac{n}{2}}, & n \equiv 0 \pmod{2}, \\ 0, & n \equiv 1 \pmod{2}. \end{cases}$$

Dowód. Niech $n \in \mathbb{N}$, niech $B = b_0b_1 \dots b_{n-1}$ i niech

$$N = |\{B \in \mathcal{B} : |B| = n, B = \pi(B)\}|.$$

Wówczas $B = \pi(B)$ wtedy i tylko wtedy, gdy $b_i = 1 - b_{n-1-i}$ dla $i \in \{0, 1, \dots, \lfloor \frac{n-1}{2} \rfloor\}$. Jeżeli $n \equiv 0 \pmod{2}$, to $i \neq n-1-i$ dla wszystkich $i \in \{0, 1, \dots, \frac{n}{2}-1\}$, a wówczas $N = 2^{\frac{n}{2}}$. Jeżeli $n \equiv 1 \pmod{2}$, to $n-1 - \frac{n-1}{2} = \frac{n-1}{2}$, czyli $b_{\frac{n-1}{2}} = 1 - b_{\frac{n-1}{2}}$. Sprzeczność, nie istnieją takie ciągi binarne, więc $N = 0$. \square

Zliczanie ciągów binarnych oraz ścieżek heksastruktur możemy podzielić na wiele różnych przypadków. W niniejszym rozdziale skupimy się na następujących sześciu, które uważamy za najistotniejsze w kontekście niniejszej rozprawy doktorskiej:

- a. liczba ciągów binarnych o długości n ,
- b. liczba ciągów binarnych o długości n reprezentujących heksastryki posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego,
- c. liczba ciągów binarnych o długości n z dokładnością do ich niezmienniczości względem rotacji,
- d. liczba ciągów binarnych o długości n , z dokładnością do ich niezmienniczości względem rotacji, reprezentujących heksastryki posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego,
- e. liczba ciągów binarnych o długości n z dokładnością do ich niezmienniczości względem rotacji i odbić lustrzanych,
- f. liczba ciągów binarnych o długości n , z dokładnością do ich niezmienniczości względem rotacji i odbić lustrzanych, reprezentujących heksastryki posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego.

| n | $a(n)$ | $b(n)$ | $a(n) - b(n)$ | $(a(n) - b(n))/b(n)$ | $b(n)/b(n-1)$ |
|-----|----------------|---------------|----------------|----------------------|---------------|
| 1 | 2 | 2 | 0 | 0.0 | |
| 2 | 4 | 4 | 0 | 0.0 | 2.0000 |
| 3 | 8 | 8 | 0 | 0.0 | 2.0000 |
| 4 | 16 | 16 | 0 | 0.0 | 2.0000 |
| 5 | 32 | 30 | 2 | 0.1 | 1.8750 |
| 6 | 64 | 58 | 6 | 0.1 | 1.9333 |
| 7 | 128 | 112 | 16 | 0.1 | 1.9310 |
| 8 | 256 | 216 | 40 | 0.2 | 1.9286 |
| 9 | 512 | 406 | 106 | 0.3 | 1.8796 |
| 10 | 1 024 | 776 | 248 | 0.3 | 1.9113 |
| 11 | 2 048 | 1 472 | 576 | 0.4 | 1.8969 |
| 12 | 4 096 | 2 796 | 1 300 | 0.5 | 1.8995 |
| 13 | 8 192 | 5 260 | 2 932 | 0.6 | 1.8813 |
| 14 | 16 384 | 9 964 | 6 420 | 0.6 | 1.8943 |
| 15 | 32 768 | 18 756 | 14 012 | 0.7 | 1.8824 |
| 16 | 65 536 | 35 400 | 30 136 | 0.9 | 1.8874 |
| 17 | 131 072 | 66 450 | 64 622 | 1.0 | 1.8771 |
| 18 | 262 144 | 125 168 | 136 976 | 1.1 | 1.8836 |
| 19 | 524 288 | 234 768 | 289 520 | 1.2 | 1.8756 |
| 20 | 1 048 576 | 441 332 | 607 244 | 1.4 | 1.8799 |
| 21 | 2 097 152 | 826 564 | 1 270 588 | 1.5 | 1.8729 |
| 22 | 4 194 304 | 1 551 488 | 2 642 816 | 1.7 | 1.8770 |
| 23 | 8 388 608 | 2 903 404 | 5 485 204 | 1.9 | 1.8714 |
| 24 | 16 777 216 | 5 442 740 | 11 334 476 | 2.1 | 1.8746 |
| 25 | 33 554 432 | 10 175 458 | 23 378 974 | 2.3 | 1.8695 |
| 26 | 67 108 864 | 19 053 856 | 48 055 008 | 2.5 | 1.8725 |
| 27 | 134 217 728 | 35 598 028 | 98 619 700 | 2.8 | 1.8683 |
| 28 | 268 435 456 | 66 596 136 | 201 839 320 | 3.0 | 1.8708 |
| 29 | 536 870 912 | 124 332 150 | 412 538 762 | 3.3 | 1.867 |
| 30 | 1 073 741 824 | 232 405 998 | 841 335 826 | 3.6 | 1.8692 |
| 31 | 2 147 483 648 | 433 651 416 | 1 713 832 232 | 4.0 | 1.8659 |
| 32 | 4 294 967 296 | 810 017 712 | 3 484 949 584 | 4.3 | 1.8679 |
| 33 | 8 589 934 592 | 1 510 605 650 | 7 079 328 942 | 4.7 | 1.8649 |
| 34 | 17 179 869 184 | 2 819 861 226 | 14 360 007 958 | 5.1 | 1.8667 |
| 35 | 34 359 738 368 | 5 256 363 816 | 29 103 374 552 | 5.5 | 1.8641 |

Tabela A.3: Wartości ciągów a i b oraz zależności $a - b$, $\frac{a-b}{b}$ i iloczynu ciągu b dla $n \in \{1, 2, \dots, 35\}$, gdzie a jest odwzorowaniem zwracającym liczbę ciągów binarnych o danej długości oraz b jest odwzorowaniem zwracającym liczbę ciągów binarnych o danej długości reprezentujących heksastrykty posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego.

Analizę rozpoczniemy od dwóch pierwszych przypadków. Niech $a: \mathbb{N} \rightarrow \mathbb{N}$ będzie funkcją, która zwraca liczbę ciągów binarnych o danej długości oraz niech $b: \mathbb{N} \rightarrow \mathbb{N}$ będzie funkcją, która zwraca liczbę ciągów binarnych o zadanej długości reprezentujących heksastrykty posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego. W ciągach rozważanych w funkcji a kolejny powstaje poprzez dołożenie bitów 0 i 1, więc natychmiast otrzymujemy dokładny wzór.

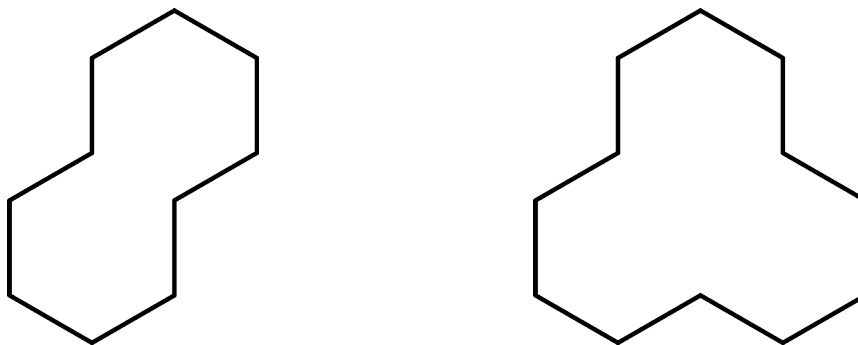
Twierdzenie A.3.6. *Jeżeli $n \in \mathbb{N}$, to $a(n) = 2^n$.*

Pierwsze kolizje w ciągach binarnych reprezentujących heksastrykty pojawiają się dla $n = 5$. Są to ciągi 00000 i 11111, których kształt będzie powodował również kolizje w dłuższych ciągach binarnych reprezentujących heksastrykty (por. rys. A.3). Kolejny taki kształt pojawia się dla $n = 9$ i reprezentuje go ciąg binarny, np. 1111011110 (por. rys. A.4). Kolejne kształty pojawiają się dla $n = 11$ (por. rys. A.4) oraz dla $n = 13$ (por. rys. A.5). Zauważmy (por. tabela A.3), że liczba ciągów binarnych potencjalnie reprezentujących ścieżki z kolizjami przekracza liczbę prawidłowych ciągów binarnych dla $n = 18$, gdzie $b(18) = 125\,168$ natomiast $a(18) - b(18) = 136\,976$. Co więcej, już dla n równego 24 nieprawidłowych ciągów binarnych jest ponad dwa razy więcej niż tych prawidłowych, co sugeruje, że granica iloczynu ciągu b jest mniejsza od 2. Istotnie, w pracy [10] autorzy przedstawiają dowód na istnienie następującej stałej wskazującej na ograniczenie przyrostu kolejnych wartości:

Twierdzenie A.3.7 ([10], twierdzenie 1). *Jeżeli $n \rightarrow \infty$, to $\frac{b_{n+1}}{b_n} \rightarrow \sqrt{2 + \sqrt{2}} \approx 1.8478$.*

Uwaga A.3.8. *W podrozdziale B.6 przedstawiliśmy implementacje pozwalające na obliczenie wartości ciągu b .*

Intuicyjnie, liczba ciągów binarnych reprezentujących heksastrykty posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego rośnie wykładniczo w tempie około 1.85-rza na krok. Dokładny wzór dla b pozostaje otwartym problemem.

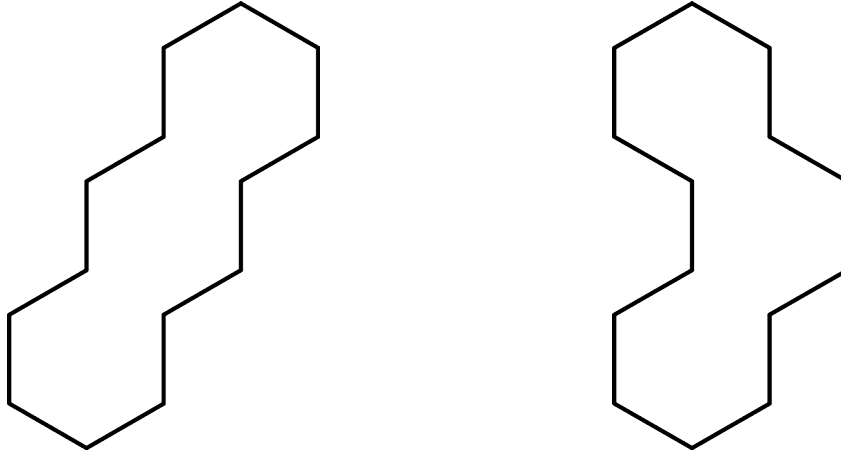


Rysunek A.4: Kształty heksastryktów z jedną kolizją. Po lewej stronie heksastrykt reprezentowany przez ciągi binarne o długości 9 (np. 1111011110). Po prawej stronie heksastrykt reprezentowany przez ciągi binarne o długości 11 (np. 111011101110).

Przechodzimy do kolejnych dwóch przypadków. Niech $c: \mathbb{N} \rightarrow \mathbb{N}$ będzie funkcją zwracającą liczbę niezmienniczych ze względu na rotacje ciągów binarnych o danej długości

| n | $c(n)$ | $d(n)$ | $c(n) - d(n)$ | $(c(n) - d(n))/d(n)$ | $d(n)/d(n-1)$ |
|-----|----------------|---------------|----------------|----------------------|---------------|
| 1 | 1 | 1 | 0 | 0.0 | |
| 2 | 3 | 3 | 0 | 0.0 | 3.0000 |
| 3 | 4 | 4 | 0 | 0.0 | 1.3333 |
| 4 | 10 | 10 | 0 | 0.0 | 2.5000 |
| 5 | 16 | 15 | 1 | 0.1 | 1.5000 |
| 6 | 36 | 33 | 3 | 0.1 | 2.2000 |
| 7 | 64 | 56 | 8 | 0.1 | 1.6970 |
| 8 | 136 | 116 | 20 | 0.2 | 2.0714 |
| 9 | 256 | 203 | 53 | 0.3 | 1.7500 |
| 10 | 528 | 403 | 125 | 0.3 | 1.9852 |
| 11 | 1 024 | 736 | 288 | 0.4 | 1.8263 |
| 12 | 2 080 | 1 427 | 653 | 0.5 | 1.9389 |
| 13 | 4 096 | 2 630 | 1 466 | 0.6 | 1.8430 |
| 14 | 8 256 | 5 038 | 3 218 | 0.6 | 1.9156 |
| 15 | 16 384 | 9 378 | 7 006 | 0.7 | 1.8615 |
| 16 | 32 896 | 17 806 | 15 090 | 0.8 | 1.8987 |
| 17 | 65 536 | 33 225 | 32 311 | 1.0 | 1.8659 |
| 18 | 131 328 | 62 785 | 68 543 | 1.1 | 1.8897 |
| 19 | 262 144 | 117 384 | 144 760 | 1.2 | 1.8696 |
| 20 | 524 800 | 221 048 | 303 752 | 1.4 | 1.8831 |
| 21 | 1 048 576 | 413 282 | 635 294 | 1.5 | 1.8696 |
| 22 | 2 098 176 | 776 466 | 1 321 710 | 1.7 | 1.8788 |
| 23 | 4 194 304 | 1 451 702 | 2 742 602 | 1.9 | 1.8696 |
| 24 | 8 390 656 | 2 722 731 | 5 667 925 | 2.1 | 1.8755 |
| 25 | 16 777 216 | 5 087 729 | 11 689 487 | 2.3 | 1.8686 |
| 26 | 33 558 528 | 9 529 493 | 24 029 035 | 2.5 | 1.8730 |
| 27 | 67 108 864 | 17 799 014 | 49 309 850 | 2.8 | 1.8678 |
| 28 | 134 225 920 | 33 302 896 | 100 923 024 | 3 | 1.8711 |
| 29 | 268 435 456 | 62 166 075 | 206 269 381 | 3.3 | 1.8667 |
| 30 | 536 887 296 | 116 212 073 | 420 675 223 | 3.6 | 1.8694 |
| 31 | 1 073 741 824 | 216 825 708 | 856 916 116 | 4 | 1.8658 |
| 32 | 2 147 516 416 | 405 025 889 | 1 742 490 527 | 4.3 | 1.8680 |
| 33 | 4 294 967 296 | 755 302 825 | 3 539 664 471 | 4.7 | 1.8648 |
| 34 | 8 590 000 128 | 1 409 962 567 | 7 180 037 561 | 5.1 | 1.8668 |
| 35 | 17 179 869 184 | 2 628 181 908 | 14 551 687 276 | 5.5 | 1.8640 |

Tabela A.4: Wartości ciągów c i d oraz zależności $c - d$, $\frac{c-d}{d}$ i iloczynu ciągu d dla $n \in \{1, 2, \dots, 35\}$, gdzie c jest odwzorowaniem zwracającym liczbę niezmienniczych ze względu na rotacje ciągów binarnych o danej długości oraz d jest odwzorowaniem zwracającym liczbę niezmienniczych ze względu na rotacje ciągów binarnych o danej długości reprezentujących heksastrukty posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego.



Rysunek A.5: Kształty heksastrukturów z jedną kolizją reprezentowane przez ciągi binarne o długości 13. Przykładowymi ciągami binarnymi reprezentującymi heksastryktę od lewej do prawej, to, odpowiednio, 11110101111010 oraz 11101101111100.

i niech $d: \mathbb{N} \rightarrow \mathbb{N}$ będzie funkcją zwracającą liczbę niezmienniczych ze względu na rotacje ciągów binarnych o danej długości reprezentujących heksastryktę posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego. Wzór odwzorowania c wyprowadzamy z wykorzystaniem twierdzeń A.3.4 i A.3.5 zliczając ciągi binarne niezmiennicze ze względu na rotację oraz zliczając ciągi binarne, które nie są niezmiennicze ze względu na rotację, co redukuje ich liczbę o połowę.

Twierdzenie A.3.9. *Jeżeli $n \in \mathbb{N}$, to*

$$c(n) = \begin{cases} 2^{n-1} + 2^{\frac{n}{2}-1}, & n \equiv 0 \pmod{2}, \\ 2^{n-1}, & n \equiv 1 \pmod{2}. \end{cases}$$

Dowód. Jeżeli $n \in 2\mathbb{N}$, to, na mocy twierdzenia A.3.5, mamy $2^{\frac{n}{2}}$ takich ciągów binarnych B , że $B = \pi(B)$. Stąd mamy $2^n - 2^{\frac{n}{2}}$ takich ciągów binarnych, że $B \neq \pi(B)$, a więc

$$c(n) = \frac{1}{2} \left(2^n - 2^{\frac{n}{2}} \right) + 2^{\frac{n}{2}} = 2^{n-1} + 2^{\frac{n}{2}-1}.$$

Jeżeli $n \in 2\mathbb{N} - 1$, to nie istnieją takie ciągi binarne, że $B = \pi(B)$. Stąd $c(n) = \frac{1}{2} \cdot 2^n$. \square

Ponownie, analizując dane empiryczne (por. tabela A.4) widzimy, że liczba nieprawidłowych ciągów binarnych przekracza liczbę ciągów binarnych uwzględnionych w czwartym przypadku dla $n = 18$, gdzie jest ich, odpowiednio, 125 168 oraz 136 976. Co więcej, dla $n = 24$ liczba nieprawidłowych ciągów binarnych jest już ponad dwukrotnie większa. Zatem ponownie spodziewamy się, że granica iloczynu ciągu d będzie mniejsza od dwóch. Istotnie, jeżeli B jest ciągiem binarnym, to może on reprezentować maksymalnie dwa heksastryktę z uwzględnieniem niezmienniczości na rotację. Zatem, w zaokrągleniu, dla dużych n , pozostaje nam co drugi ciąg binarny. Stąd $d_n \approx \frac{1}{2} b_n$ dla dużych n , a tym samym zbieżność ilorazu ciągu d upraszcza się do zbieżności ilorazu ciągu b .

Twierdzenie A.3.10. *Jeżeli $n \rightarrow \infty$, to $\frac{d_{n+1}}{d_n} \rightarrow \sqrt{2 + \sqrt{2}} \approx 1.8478$.*

Zauważmy, że jeżeli pominiemy takie ciągi binarne B , że $B = \pi(B)$, to wartości ciągu d dla nieparzystych argumentów są zależne od wartości ciągu b .

Wniosek A.3.11. *Jeżeli $n \in 2\mathbb{N} + 1$, to $2d_n = b_n$.*

Uwaga A.3.12. *W podrozdziale B.6 przedstawiliśmy implementacje pozwalające na obliczenie wartości ciągu d .*

W ostatnich dwóch przypadkach rozważamy odwzorowanie $e: \mathbb{N} \rightarrow \mathbb{N}$ zwracające liczbę niezmienniczych ze względu na rotacje i odbicia lustrzane ciągów binarnych o danej długości oraz odwzorowanie $f: \mathbb{N} \rightarrow \mathbb{N}$ zwracające liczbę niezmienniczych ze względu na rotacje i odbicia lustrzane ciągów binarnych o danej długości reprezentujących heksastrykty posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego. Wzór dla ciągu e wyprowadzamy na podstawie twierdzeń A.3.1, A.3.2 i wniosku A.3.3 dzieląc zagadnienie na sumę liczby ciągów binarnych niezmienniczych ze względu na odwrócenie ciągu, liczby ciągów binarnych niezmienniczych na rotację oraz pozostałych. Wówczas formuła wynika z twierdzeń A.3.4 i A.3.5.

Twierdzenie A.3.13. *Jeżeli $n \in \mathbb{N}$, to*

$$e(n) = \begin{cases} 2^{n-2} + 2^{\frac{n}{2}-1}, & n \equiv 0 \pmod{2}, \\ 2^{n-2} + 2^{\frac{n-1}{2}-1}, & n \equiv 1 \pmod{2}. \end{cases}$$

Dowód. Jeżeli $n \in 2\mathbb{N}$, to, na mocy twierdzenia A.3.5, mamy $2^{\frac{n}{2}}$ takich ciągów binarnych B , że $B = \pi(B)$ oraz, na mocy twierdzenia A.3.4, mamy $2^{\frac{n}{2}}$ takich ciągów binarnych, że $B = \zeta(B)$. Stąd mamy $2^n - 2^{\frac{n}{2}} - 2^{\frac{n}{2}}$ takich ciągów binarnych, że $B \neq \pi(B)$ oraz $B \neq \zeta(B)$. Na mocy trzeciego punktu twierdzenia A.3.4, bierzemy co czwarty taki ciąg binarny, a więc

$$e(n) = \frac{1}{4} \left(2^n - 2^{\frac{n}{2}} - 2^{\frac{n}{2}} \right) + 2^{\frac{n}{2}} + 2^{\frac{n}{2}} = 2^{n-2} + 2^{\frac{n}{2}-1}.$$

Jeżeli $n \in 2\mathbb{N} - 1$, to nie istnieją takie ciągi binarne, że $B = \pi(B)$. Stąd

$$e(n) = \frac{1}{4} \left(2^n - 2^{\frac{n}{2}} \right) + 2^{\frac{n}{2}} = 2^{n-2} + 2^{\frac{n-1}{2}-1}.$$

□

Konsekwentnie analizując dane empiryczne (por. tabela A.5) widzimy, że liczba nieprawidłowych ciągów binarnych przekracza liczbę ciągów binarnych uwzględnionych w szóstym przypadku dla $n = 18$, gdzie jest ich, odpowiednio, 31 455 oraz 34 337, a dla $n = 24$ liczba nieprawidłowych ciągów binarnych jest już ponad dwukrotnie większa. Znowu spodziewamy się, że granica iloczynu ciągu d będzie mniejsza od dwóch. Istotnie, jeżeli B jest ciągiem binarnym, to może on reprezentować maksymalnie cztery heksastrykty z uwzględnieniem niezmienniczości na zamianę bitów, odwrócenie ciągu i rotację. Zatem, w zaokrągleniu, dla dużych n , pozostaje nam co czwarty ciąg binarny. Stąd $f_n \approx \frac{1}{4}b_n$ dla dużych n .

Twierdzenie A.3.14. *Jeżeli $n \rightarrow \infty$, to $\frac{f_{n+1}}{f_n} \rightarrow \sqrt{2 + \sqrt{2}} \approx 1.8478$.*

Uwaga A.3.15. *W podrozdziale B.6 przedstawiliśmy implementacje pozwalające na obliczenie wartości ciągu f .*

| n | $e(n)$ | $f(n)$ | $e(n) - f(n)$ | $(e(n) - f(n))/f(n)$ | $f(n)/f(n-1)$ |
|-----|---------------|---------------|---------------|----------------------|---------------|
| 1 | 1 | 1 | 0 | 0.0 | |
| 2 | 2 | 2 | 0 | 0.0 | 2.0000 |
| 3 | 3 | 3 | 0 | 0.0 | 1.5000 |
| 4 | 6 | 6 | 0 | 0.0 | 2.0000 |
| 5 | 10 | 9 | 1 | 0.1 | 1.5000 |
| 6 | 20 | 18 | 2 | 0.1 | 2.0000 |
| 7 | 36 | 31 | 5 | 0.2 | 1.7222 |
| 8 | 72 | 61 | 11 | 0.2 | 1.9677 |
| 9 | 136 | 107 | 29 | 0.3 | 1.7541 |
| 10 | 272 | 207 | 65 | 0.3 | 1.9346 |
| 11 | 528 | 378 | 150 | 0.4 | 1.8261 |
| 12 | 1 056 | 724 | 332 | 0.5 | 1.9153 |
| 13 | 2 080 | 1 333 | 747 | 0.6 | 1.8412 |
| 14 | 4 160 | 2 537 | 1 623 | 0.6 | 1.9032 |
| 15 | 8 256 | 4 723 | 3 533 | 0.7 | 1.8616 |
| 16 | 16 512 | 8 938 | 7 574 | 0.8 | 1.8924 |
| 17 | 32 896 | 16 674 | 16 222 | 1.0 | 1.8655 |
| 18 | 65 792 | 31 455 | 34 337 | 1.1 | 1.8865 |
| 19 | 131 328 | 58 805 | 72 523 | 1.2 | 1.8695 |
| 20 | 262 656 | 110 642 | 152 014 | 1.4 | 1.8815 |
| 21 | 524 800 | 206 849 | 317 951 | 1.5 | 1.8695 |
| 22 | 1 049 600 | 388 444 | 661 156 | 1.7 | 1.8779 |
| 23 | 2 098 176 | 726 236 | 1 371 940 | 1.9 | 1.8696 |
| 24 | 4 196 352 | 1 361 766 | 2 834 586 | 2.1 | 1.8751 |
| 25 | 8 390 656 | 2 544 568 | 5 846 088 | 2.3 | 1.8686 |
| 26 | 16 781 312 | 4 765 466 | 12 015 846 | 2.5 | 1.8728 |
| 27 | 33 558 528 | 8 900 805 | 24 657 723 | 2.8 | 1.8678 |
| 28 | 67 117 056 | 16 652 803 | 50 464 253 | 3.0 | 1.8709 |
| 29 | 134 225 920 | 31 085 427 | 103 140 493 | 3.3 | 1.8667 |
| 30 | 268 451 840 | 58 108 482 | 210 343 358 | 3.6 | 1.8693 |
| 31 | 536 887 296 | 108 417 265 | 428 470 031 | 4.0 | 1.8658 |
| 32 | 1 073 774 592 | 202 517 545 | 871 257 047 | 4.3 | 1.8679 |
| 33 | 2 147 516 416 | 377 659 515 | 1 769 856 901 | 4.7 | 1.8648 |
| 34 | 4 295 032 832 | 704 989 607 | 3 590 043 225 | 5.1 | 1.8667 |
| 35 | 8 590 000 128 | 1 314 105 907 | 7 275 894 221 | 5.5 | 1.8640 |

Tabela A.5: Wartości ciągów e i f oraz zależności $e - f$, $\frac{e-f}{f}$ i iloczynu ciągu f dla $n \in \{1, 2, \dots, 35\}$, gdzie e jest odwzorowaniem zwracającym liczbę niezmienniczych ze względu na rotacje i odbicia lustrzane ciągów binarnych o danej długości oraz f jest odwzorowaniem zwracającym liczbę niezmienniczych ze względu na rotacje i odbicia lustrzane ciągów binarnych o danej długości reprezentujących heksastryki posiadające dokładnie dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego.

Dodatek B

Implementacje w języku Python

W tym rozdziale przedstawimy implementacje narzędzi przedstawionych w niniejszej dysertacji z opisem ich zaprojektowania oraz działania. Wskażemy zalety wybranych rozwiązań pod względem optymalizacyjnym. Kod przygotowaliśmy w Google Colab z użyciem języka Python z wielu powodów. Przede wszystkim jest darmowy, nie wymaga instalacji środowiska oraz jest dostępny z dowolnej maszyny. Ponadto Colab ma preinstalowane popularne biblioteki oraz jest oparty o Jupyter Notebook, dzięki czemu łatwo jest wizualizować dane oraz dokumentować wyniki. Natomiast Python nie wymaga kompilacji, jest powszechnie wykorzystywane w społeczności naukowców oraz pozwala na szybkie tworzenie i testowanie algorytmów. Podstawowym modulem wykorzystywanym przez nas jest Numba, który zdecydowanie przyspiesza obliczenia dzięki kompilowaniu kodu Python do kodu maszynowego, co jest istotne w implementacji złożonych algorytmów wykorzystywanych w tej pracy. Poniższe implementacje oraz inne niezbędne do przeprowadzenia symulacji udostępniamy na platformie GitHub w repozytorium <https://github.com/Grzegorz-Mos/PhD/>.

B.1 Funkcja τ

Funkcja τ zamienia bity ciągu binarnego na przeciwne. Możemy ją zaimplementować w języku Python na dwa sposoby:

- (1) iterując znaki ciągu binarnego i zamieniając 0 na 1 oraz 1 na 0 (por. implementacja B.1.1),
- (2) wykorzystując operator XOR (por. implementacja B.1.2).

Rozważając ciągi binarne w postaci ciągów znaków, zgodnie z dokumentacją Python, otrzymujemy liniową złożoność obliczeniową. W przypadku ciągów binarnych jako liczb złożoność obliczeniowa jest stała.

```
1 def tauStr( binarySequenceString: str ) -> str:
2     return ''.join('1' if b == '0' else '0' for b in
    binarySequenceString)
```

Implementacja B.1.1: Implementacja funkcji τ dla ciągu binarnego w postaci ciągu znaków.

```

1 def tauInt( binarySequenceInteger: int , length: int ) -> int:
2     mask = (1 << length) - 1
3     return binarySequenceInteger ^ mask

```

Implementacja B.1.2: Implementacja funkcji τ dla ciągu binarnego w postaci liczby całkowitej z określoną długością ciągu.

B.2 Funkcja ζ

Funkcję ζ możemy zaimplementować na kilka sposobów:

- traktując ciąg binarny jako ciąg znaków, a następnie wykorzystując notację wycinków (por. implementacja B.2.1),
- traktując ciąg binarny jako liczbę całkowitą z daną jego długością, a następnie ponownie wykorzystując notację wycinków (por. implementacja B.2.2),
- traktując ciąg binarny jako liczbę całkowitą z daną długością, a następnie wykorzystując operacje binarne optymalizując funkcję metodą njit z biblioteki Numba (por. implementacja B.2.3).

Jeżeli ciąg binarny jest w postaci ciągu znaków, to implementacja B.2.1 jest najoptymalniejsza. Jeżeli ciąg binarny jest w postaci liczby całkowitej, to implementacja B.2.3 jest najoptymalniejsza. Implementacje B.2.1 i B.2.2 nie mogą być skompilowane z użyciem pakietu Numba ze względu na operacje na tekście, które nie są wspierane w tym przypadku.

Funkcję ζ implementujemy z wykorzystaniem notacji wycinków (por. implementacja B.2.1). Wykorzystamy również implementację dla ciągów binarnych w postaci liczb całkowitych, która również korzysta z notacji wycinków (por. implementacja B.2.2).

```

1 def sigmaStr( binarySequenceString: str ) -> str:
2     return binarySequenceString[::-1]

```

Implementacja B.2.1: Implementacja funkcji ζ dla ciągu binarnego w postaci ciągu znaków.

```

1 def sigmaInt( binarySequenceInteger: int , length: int ) -> int :
2     binarySequenceString = bin(binarySequenceInteger)[2:].zfill(
3         length)
4     return int( binarySequenceString[::-1] , 2 )

```

Implementacja B.2.2: Implementacja funkcji ζ dla ciągu binarnego w postaci liczby całkowitej z określoną długością ciągu z wykorzystaniem operacji na ciągach znaków.

```

1 def sigmaInt(bsInt: int, length: int) -> int:
2     result = 0
3     for _ in range(length):
4         result = (result << 1) | (bsInt & 1)
5         bsInt >>= 1
6     return result

```

Implementacja B.2.3: Implementacja funkcji ζ dla ciągu binarnego w postaci liczby całkowitej z określoną długością ciągu z wykorzystaniem operacji binarnych.

B.3 Funkcja π

Funkcję π wdrożymy wykorzystując implementacje τ i ζ . Analogicznie rozważamy możliwość rozważenia ciągu binarnego jako tekstu (por. implementacja B.3.1) oraz jako liczby całkowitej (por. implementacja B.3.2).

```
1 def piStr( binarySequenceString: str ) -> str:  
2   return sigmaStr( tauStr( binarySequenceString ) )
```

Implementacja B.3.1: Implementacja funkcji π dla ciągu binarnego w postaci ciągu znaków.

```
1 def piInt( binarySequenceInteger: int , length: int ) -> int:  
2   return sigmaInt( tauInt( binarySequenceInteger , length ) ,  
   length )
```

Implementacja B.3.2: Implementacja funkcji π dla ciągu binarnego w postaci liczby całkowitej z określoną długością ciągu.

Przeprowadziliśmy testy porównujące efektywność implementacji funkcji π opartej o liczby z jej implementacją opartą o ciągi znaków. Otrzymaliśmy następujące czasy dla poszczególnych podejść bez prekompilacji z wykorzystaniem pakietu numba:

- implementacja z wykorzystaniem liczb: 33, 38, 34 i 32 sekundy,
- implementacja z wykorzystaniem ciągów znaków: 80, 81, 90 i 84 sekundy.

Zatem implementacja z wykorzystaniem liczbowej reprezentacji ciągu binarnego jest średnio ponad dwukrotnie szybsza czasowo niż implementacja z wykorzystaniem ciągu binarnego. Ponadto wykorzystanie pakietu numba poprawia wynik dla implementacji z wykorzystaniem liczb dwukrotnie.

B.4 Rozszerzone ciągi binarne

W implementacji B.4.1 przedstawiamy wdrożenie rozszerzonych ciągów binarnych w postaci obiektu `HexSeq`, który ułatwia nam pracę z nimi. W kolejnych implementacjach wdramy przydatne metody do klasy `HexSeq`. W B.4.2 zwracamy poszczególne podciągi rozszerzonego ciągu binarnego w postaci obiektu `HexSeq`. W B.4.3 sprawdzamy, czy obiekt istotnie jest rozszerzonym ciągiem binarnym, czy jednak redukuje się do ciągu binarnego. W B.4.4 zwracamy poszczególne składowe rozszerzonego ciągu binarnego w postaci obiektu `HexSeq`. W kolejnych implementacjach: B.4.5, B.4.6, B.4.7 i B.4.8 wdramy odwzorowania dla rozszerzonych ciągów binarnych, odpowiednio, ω , τ , ζ i π .

```

1 class HexSeq:
2     def __init__( self , extendedBinarySequenceStringOrPrefix , infix
      = None , suffix = None ):
3         if isinstance( extendedBinarySequenceStringOrPrefix , HexSeq )
      and isinstance( infix , HexSeq ) and isinstance( suffix , HexSeq
      ):
4             self.prefix = str( extendedBinarySequenceStringOrPrefix )
5             self.infix = str( infix )
6             self.suffix = str( suffix )
7         elif isinstance( extendedBinarySequenceStringOrPrefix , str )
      and infix == None:
8             extendedBinarySequenceString =
      extendedBinarySequenceStringOrPrefix
9             if not '(' in extendedBinarySequenceString:
10                self.prefix = extendedBinarySequenceString
11                self.infix = None
12                self.suffix = None
13            else:
14                bracketsCounter = 1
15                i = extendedBinarySequenceString.index( '(' )
16                n = len( extendedBinarySequenceString )
17                for j in range( i + 1 , n ):
18                    if extendedBinarySequenceString[ j ] == '(':
19                        bracketsCounter += 1
20                    elif extendedBinarySequenceString[ j ] == ')':
21                        bracketsCounter -= 1
22
23                    if bracketsCounter == 0:
24                        break
25
26                self.prefix = extendedBinarySequenceString[:i]
27                self.infix = extendedBinarySequenceString[i+1:j]
28                self.suffix = extendedBinarySequenceString[j+1:]
29
30        def __repr__(self):
31            if self.infix == None:
32                return self.prefix
33
34            return f"{self.prefix}({self.infix}){self.suffix}"

```

Implementacja B.4.1: Podstawowa implementacja rozszerzonego ciągu binarnego w postaci klasy HexSeq wraz z metodą pozwalającą na jego wyświetlenie w postaci ciągu znaków.

```

1     def getPrefix( self ):
2         return HexSeq( self.prefix )
3
4     def getInfix( self ):
5         return HexSeq( self.infix )

```

```

6
7 def getSufix( self ):
8     return HexSeq( self.sufix )

```

Implementacja B.4.2: Metody klasy HexSeq pozwalające na pobranie poszczególnych podciągów rozszerzonego ciągu binarnego w postaci obiektu HexSeq.

```

1 def isExtendedBinarySequence( self ):
2     return self.infix != None

```

Implementacja B.4.3: Metoda klasy HexSeq sprawdzająca, czy obiekt istotnie reprezentuje rozszerzony ciąg binarny, czy ciąg binarny.

```

1 def getFirstSubSequence( self ):
2     return HexSeq( f"{self.prefix}0{self.infix}" )
3
4 def getSecondSubSequence( self ):
5     return HexSeq( f"{self.prefix}1{self.sufix}" )
6
7 def getSubSequences( self ):
8     if not self.isExtendedBinarySequence():
9         return [ self.getPrefix() ]
10
11    return [ self.getFirstSubSequence() , self.getSecondSubSequence
            ( ) ]

```

Implementacja B.4.4: Metody klasy HexSeq zwracające składowe rozszerzonego ciągu binarnego.

```

1 def getComplexity( self ):
2     if not self.isExtendedBinarySequence():
3         return 1
4
5     return sum( seq.getComplexity() for seq in self.getSubSequences
                ( ) )

```

Implementacja B.4.5: Metoda klasy HexSeq zwracająca jego złożoność.

```

1 def getTau( self ):
2     if not self.isExtendedBinarySequence():
3         length = len( self.prefix )
4         binarySequenceAsInt = int( self.prefix , 2 )
5         mask = (1 << length ) - 1
6         return HexSeq( bin( binarySequenceAsInt ^ mask )[2:].zfill(
            length ) )
7
8     return HexSeq( self.getPrefix().getTau() , self.getSufix().
                getTau() , self.getInfix().getTau() )

```

Implementacja B.4.6: Metoda klasy HexSeq zwracająca rozszerzony ciąg binarny będący obrazem funkcji τ .

```

1  def getSigma( self ):
2      if not self.isExtendedBinarySequence():
3          return HexSeq( self.prefix[:: -1] )
4
5      return HexSeq( self.getSufix().getSigma() , self.getInfix().
    getTau() , self.getPrefix().getSigma() )

```

Implementacja B.4.7: Metoda klasy HexSeq zwracającą rozszerzony ciąg binarny będący obrazem funkcji ζ .

```

1  def getPi( self ):
2      return self.getSigma().getTau()

```

Implementacja B.4.8: Metoda klasy HexSeq zwracającą rozszerzony ciąg binarny będący obrazem funkcji π .

B.5 Metryka dla rozszerzonych ciągów binarnych

Implementacją B.5.1 wdramy równanie (4.1.1) dzięki czemu możemy obliczać odległość między ciągami binarnymi i rozszerzonymi ciągami binarnymi z wykorzystaniem miary odległości określonej dla ciągów binarnych oraz HexSeq przedstawionego w implementacji B.4.1.

```

1  def ebsDistance( s1: HexSeq , s2: HexSeq , bsDistance ):
2      isEBS1 , isEBS2 = s1.isExtendedBinarySequence() , s2.
        isExtendedBinarySequence()
3
4      if not isEBS1 and not isEBS2:
5          return bsDistance( s1 , s2 )
6
7      if not isEBS1:
8          return ebsDistance( s2 , s1 , bsDistance )
9
10     subSequences1 = s1.getSubSequences()
11     if not isEBS2:
12         return min( ebsDistance( subSequences1[ 0 ] , s2 , bsDistance )
13             + ebsDistance( subSequences1[ 1 ] , HexSeq( '' ) , bsDistance ) ,
14             ebsDistance( subSequences1[ 1 ] , s2 , bsDistance )
15             + ebsDistance( subSequences1[ 0 ] , HexSeq( '' ) , bsDistance ) )
16
17     subSequences2 = s2.getSubSequences()
18     return min(
19         sum( ebsDistance( subSequences1[ i ] , subSequences2[ i ] ,
20             bsDistance ) for i in range( 2 ) ) ,
21         sum( ebsDistance( subSequences1[ i ] , subSequences2[ 1 - i ]
22             , bsDistance ) for i in range( 2 ) ) )

```

Implementacja B.5.1: Funkcja obliczająca odległość między dwoma rozszerzonymi ciągami binarnymi dla danej metryki mierzącej odległość między ciągami binarnymi (implementacja wzoru 4.1.1).

B.6 Catanatron

Catanatron (zob. <https://github.com/bcollazo/catanatron>) to aplikacja utworzona w języku Python przez Bryana Collazo, która pozwala na rozgrywanie i symulowanie rozgrywek w grę planszową Catan. Wykorzystaliśmy ten symulator ze względu na ogólnodostępny kod oraz przystosowanie go do wykorzystania w badaniach przez autora, który udostępnił stosowny kod do testowania i symulowania gier w języku Python. Źródła dostępne są w repozytorium GitHub, dzięki czemu łatwo jest zaimportować stosowne biblioteki do Google Colabory i z nich korzystać. Udostępniony kod w większości wystarcza nam do otrzymywania danych i ich agregowania. Poniżej punktujemy zmiany i poprawki wdrożone w kodzie celem ujednoczenia wyników symulacji (cała implementacja gotowa do symulacji dostępna w notatniku).

- Korzystamy z wersji ze zmianami wdrożonymi do pull request 301 ze względu na późniejsze zmiany w kodzie, które zmieniają sposób symulacji i nie są kompatybilne z przeprowadzonymi przez nas badaniami sprzed tych zmian.
- Zdefiniowaliśmy podstawową mapę opisaną w instrukcji Catan (por. implementacja B.6.1). Domyślnie aplikacja dla każdej symulacji generuje nową, losową mapę. Wówczas podejmowane przez graczy decyzje zależą od rozmieszczenia kafelków i otrzymane w wyniku agregacji heksastryki byłyby trudne do interpretacji.
- Przepisaliśmy funkcję `play_batch_core` z nową nazwą `X_play_batch_core` ustawiając w niej, w linii 9, zdefiniowaną powyżej mapę, tj. zamieniliśmy linię `catan_map = build_map(game_config.catan_map)` na `catan_map = DEFAULT_MAP`. Ponadto linie 16 i 17 umieściliśmy w instrukcji `try...except` ze względu na pojawiające się błędy w aplikacji. W takim przypadku skrypt odrzucał daną symulację i tworzył nową (por. implementacja B.6.2).
- Przepisaliśmy funkcję `play_batch` z nową nazwą `X_play_batch` celem wdrożenia w niej funkcji `X_play_batch_core` zamiast funkcji `play_batch_core`.
- Zdefiniowaliśmy niestandardowych graczy, których priorytetem było zbudowanie na samym początku rozgrywki wskazanej drogi (przyjeliśmy drogę o wierzchołkach 49 i 22) oraz którzy starali się budować spójne sieci dróg (por. implementacja B.6.3).

Dzięki powyższym adaptacjom symulacje generowały rozgrywki, w których jeden z graczy zawsze budował drogę, w której uwzględniona była wskazana ścieżka. Ponadto ustalone było rozmieszczenie surowców na mapie, więc budowa ścieżki na danej krawędzi zawsze dawała te same korzyści graczom, a tym samym możemy wysnuć wnioski, które ścieżki są

z tego względu bardziej wartościowe (tj. dają większą szansę na wygraną poprzez pozyskiwane surowce i zdobywane punkty). W implementacji B.6.4 pokazujemy łatwość symulowania rozgrywek i pozyskiwania wyników.

```

1 DEFAULT_MAP_TILES = initialize_tiles(
2     BASE_MAP_TEMPLATE
3     , [ 10 , 9 , 2 , 10 , 12 , 9 , 8 , 5 , 6 , 11 , 5 , 8 , 4 , 6 , 11
4         , 3 , 4 , 3 ]
5     , [ None , SHEEP , None , None , BRICK , WOOD , None , WHEAT , ORE
6         , None ]
7     , [ None , BRICK , WOOD , SHEEP , ORE , WHEAT , WHEAT , WOOD ,
8         BRICK , WHEAT , SHEEP , SHEEP , ORE , SHEEP , BRICK , WOOD , ORE
9         , WHEAT , WOOD , None ]
10 )
11
12 DEFAULT_MAP = MapInstance.from_tiles(DEFAULT_MAP_TILES)
13
14 DEFAULT_BOARD = Board( DEFAULT_MAP )

```

Implementacja B.6.1: Zdefiniowanie podstawowej mapy Catan z wykorzystaniem obiektów z Catanatronu.

```

1 def X_play_batch_core(num_games, players, game_config, accumulators
2     = []):
3     for accumulator in accumulators:
4         if isinstance(accumulator, SimulationAccumulator):
5             accumulator.before_all()
6
7     for _ in range(num_games):
8         for player in players:
9             player.reset_state()
10        catan_map = DEFAULT_MAP
11        game = Game(
12            players,
13            None,
14            game_config.discard_limit,
15            game_config.vps_to_win,
16            catan_map,
17        )
18        try:
19            game.play(accumulators)
20            yield game
21        except:
22            pass
23
24    for accumulator in accumulators:
25        if isinstance(accumulator, SimulationAccumulator):
26            accumulator.after_all()

```

Implementacja B.6.2: Zredefiniowana funkcja symulująca gry poprzez wdrożenie podstawowej mapy Catan w linii 9.

```

1 targetEdge = ( 49 , 22 )
2 class FixedPlayer(Player):
3     def decide(self, game, playable_actions):
4         if game.state.colors[0] == self.color:
5             fixedEdge = targetEdge
6             fixedFirstAction = Action( self.color , ActionType.
BUILD_ROAD , fixedEdge )
7             if is_valid_action( game.state , fixedFirstAction ):
8                 return fixedFirstAction
9
10            pa = [ a for a in playable_actions if a.action_type !=
ActionType.BUILD_ROAD or len( set( a.value + fixedEdge ) ) == 4
11                ]
12
13            return random.choice(pa)
14
15            return random.choice(playable_actions)

```

Implementacja B.6.3: Utworzenie gracza z wykorzystaniem obiektów Catanatronu ze wskazaniem preferowanej pierwszej ścieżki do zbudowania oraz ogarniczeniem akcji budowania drogi do budowania spójnej sieci dróg.

```

1 players = [
2     FixedPlayer(Color.RED),
3     FixedPlayer(Color.BLUE),
4     FixedPlayer(Color.WHITE),
5     FixedPlayer(Color.ORANGE),
6 ]
7
8 wins, results_by_player, games = X_play_batch( 100 , players , None
, GameConfigOptions( 7 , 10 ) )

```

Implementacja B.6.4: Określenie graczy oraz przeprowadzenie 100 symulacji z wykorzystaniem Catanatronu.

B.7 Zliczanie ścieżek heksastrukturów

W kodzie B.7.1 przedstawiliśmy implementację metody zliczania ciągów binarnych reprezentujących heksastrukty zawierających dwa wierzchołki stopnia pierwszego oraz pozostałe wierzchołki stopnia drugiego. Zastosowaliśmy kilka różnych metod optymalizacyjnych celem szybkiego zliczenia takich ciągów binarnych o długościach od 0 do ustalonej dodatniej liczby całkowitej oznaczonej jako DEPTH. Wykorzystaliśmy heksagonalne współrzędne wierzchołków przedstawiając je jako jedną liczbę całkowitą przy pomocy funkcji `coordinatesToNumber`, gdzie współrzędna przy Rot jest mnożona przez DEPTH zapewniając uniknięcie kolizji dla danej maksymalnej długości ciągów binarnych, a tym samym porównując wierzchołki jako liczby zamiast krotek lub list. Kolejne współrzędne obliczamy z wykorzystaniem funkcji f z twierdzenia A.1.8, której wartości zaimplemento-

waliśmy w postaci krotki, natomiast funkcję *s* zaimplementowaliśmy w postaci zmiennych *shiftedSumOfBits*, *shiftedSumOfBits0* i *shiftedSumOfBits*, gdzie zwiększając wartość *shiftedSumOfBits* o 1 dodając bit 1 otrzymując zmienną *shiftedSumOfBits1* oraz zmniejszając tę wartość o 1 dodając bit 0 otrzymując zmienną *shiftedSumOfBits0* unikamy ciągłego sumowania dla wszystkich bitów ciągu binarnego. Ponadto wykorzystujemy stos *stack* zamiast rekurencji czy generowania wszystkich ciągów binarnych o danej długości unikając przepełnienia pamięci. Wreszcie, współrzędne zapisujemy w zbiorze, co zapewnia stałą złożoność sprawdzania, czy dany wierzchołek został już odwiedzony. Wszystkie funkcje są kompilowane z użyciem *njit* z biblioteki *Numba*.

Wszystkie powyżej wykorzystane metody pozwalają na zliczenie ciągów binarnych o długościach od 0 do 25 w około 11 sekund, gdzie korzystając z rekurencji, wygenerowanie wszystkich ciągów binarnych o danej długości zajmowało ponad 5 minut. dla większych wartości, np. zliczając ciągi binarne o długościach od 0 do 35, otrzymaliśmy czas godzinę i 40 minut, co oceniamy jako bardzo dobry czas uwzględniając złożoność problemu oraz inne powszechne algorytmy wykorzystywane do tego typu zagadnień. W pozostałych dwóch przypadkach otrzymaliśmy podobny czas obliczeń.

```

1 numba.types.Set(numba.int64, reflected=True)
2 DEPTH = 35
3
4 @numba.njit
5 def coordinatesToNumber( i , r , depth ):
6     return ( i + depth ) * ( 2 * depth + 1 ) + ( r + depth )
7
8 @numba.njit
9 def append0toBinarySequence ( binarySequence : int ) :
10    return binarySequence << 1
11
12 @numba.njit
13 def append1toBinarySequence ( binarySequence : int ) :
14    return ( binarySequence << 1 ) | 1
15
16 @numba.njit
17 def f(x):
18    return (1, 0, -1)[x % 3]
19
20 BASE_COORD1 = coordinatesToNumber( 1 , 0 , DEPTH )
21 BASE_COORD2 = coordinatesToNumber( 0 , 0 , DEPTH )
22
23 @numba.njit
24 def calculateNumberOfBinarySequences(depth: int):
25     # Stack: (binarySequence, shiftedSumOfBits, rCoordinate,
26     iCoordinate, binarySequenceLength, fixedCoordinates)
27     counts = [0] * (DEPTH + 1)
28     BASE_COORDINATES = { BASE_COORD1 , BASE_COORD2 }
29     stack = [(0, 0, 0, 0, 0, BASE_COORDINATES)]

```

```

30     while stack:
31         binarySequence, shiftedSumOfBits, iCoordinate, rCoordinate,
           binarySequenceLength, fixedCoordinates = stack.pop()
32
33         counts[binarySequenceLength] += 1
34
35         if binarySequenceLength == depth:
36             continue
37
38         shiftedSumOfBits0 = shiftedSumOfBits - 1
39         rCoordinate0 = rCoordinate + ( - 1 + 2 * (
           binarySequenceLength \% 2 ) ) * f( shiftedSumOfBits0 + 1 )
40         iCoordinate0 = iCoordinate + ( 1 - 2 * (
           binarySequenceLength \% 2 ) ) * f( shiftedSumOfBits0 )
41         p0 = coordinatesToNumber( iCoordinate0 , rCoordinate0 ,
           DEPTH )
42         if p0 not in fixedCoordinates:
43             stack.append((append0toBinarySequence(binarySequence),
           shiftedSumOfBits0, iCoordinate0, rCoordinate0,
           binarySequenceLength + 1, fixedCoordinates | {p0}))
44
45         shiftedSumOfBits1 = shiftedSumOfBits + 1
46         rCoordinate1 = rCoordinate + ( - 1 + 2 * (
           binarySequenceLength \% 2 ) ) * f( shiftedSumOfBits1 + 1 )
47         iCoordinate1 = iCoordinate + ( 1 - 2 * (
           binarySequenceLength \% 2 ) ) * f( shiftedSumOfBits1 )
48         p1 = coordinatesToNumber( iCoordinate1 , rCoordinate1 ,
           DEPTH )
49         if p1 not in fixedCoordinates:
50             stack.append((append1toBinarySequence(binarySequence),
           shiftedSumOfBits1, iCoordinate1, rCoordinate1,
           binarySequenceLength + 1, fixedCoordinates | {p1}))
51
52     return counts
53
54 start = time.time()
55 counts = calculateNumberOfBinarySequences(DEPTH)
56 elapsed_time = time.time() - start
57 print(f"Time: {elapsed_time:.3f} seconds")
58 print("Counts:", counts)

```

Implementacja B.7.1: Implementacja metody zliczania ciągów binarnych o długościach od 0 do danej liczby naturalnej DEPTH reprezentujących heksastrukty zawierające dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego.

W przypadku zliczania ciągów binarnych niezmienniczych ze względu na rotacje reprezentujących heksastrukty zawierające dwa wierzchołki stopnia pierwszego oraz pozostałe wierzchołki stopnia drugiego należy w linii 21 implementacji B.7.1 dodać warunek na zliczanie tych ciągów, które są nie większe od swojej rotacji (por. implementacja B.7.2).

```

1  if binarySequence <= piInt( binarySequence , binarySequenceLength )
    :
2      counts[binarySequenceLength] += 1

```

Implementacja B.7.2: Warunek pozwalający na zliczenie ciągów binarnych niezmienniczych ze względu na rotacje reprezentujących heksastryki zawierające dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego.

Analogicznie, w przypadku zliczania ciągów binarnych niezmienniczych ze względu na rotacje i odbicia lustrzane reprezentujących heksastryki zawierające dwa wierzchołki stopnia pierwszego oraz pozostałe wierzchołki stopnia drugiego, należy w linii 21 implementacji B.7.1 dodać warunek na zliczanie tych ciągów, które są nie większe od swojej rotacji oraz od swoich odbić lustrzanych (por. implementacja B.7.3).

```

1  bsTau = tauInt( binarySequence , binarySequenceLength )
2  bsSigma = sigmaInt( binarySequence , binarySequenceLength )
3  bsPi = tauInt( bsSigma , binarySequenceLength )
4  if binarySequence <= bsTau and binarySequence <= bsSigma and
    binarySequence <= bsPi:
5      counts[binarySequenceLength] += 1

```

Implementacja B.7.3: Warunek pozwalający na zliczenie ciągów binarnych niezmienniczych ze względu na rotacje reprezentujących heksastryki zawierające dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego.

Bibliografia

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning*. 1 wyd. New York, NY, USA: Springer, 2006. ISBN: 9780387310732.
- [2] D. M. Bloom. *Linear Algebra and Geometry*. Cambridge, England: Cambridge University Press, 1979. ISBN: 9780521293242.
- [3] A. Bondy i U. S. R. Murty. *Graph Theory*. 1 wyd. London, England: Springer, 2007. ISBN: 9781846289699.
- [4] L. Boytsov. “Indexing Methods for Approximate Dictionary Searching: Comparative Analysis”. *Journal of Experimental Algorithmics* 16 (2011), s. 1.1–1.91. DOI: 10.1145/1963190.1963191.
- [5] F. Carey, R. Giuliano, N. Allison i S. Bane. *Organic Chemistry*. 11 wyd. Columbus, OH: McGraw-Hill Education, 2019. ISBN: 9781260148923.
- [6] T. Cormen, C. Leiserson, R. Rivest i C. Stein. *Introduction to Algorithms, Third Edition*. Cambridge, USA: The MIT Press, 2009. ISBN: 9780262833058.
- [7] B. A. Davey i H. A. Priestley. *Introduction to Lattices and Order*. 2 wyd. Cambridge, England: Cambridge University Press, 2012. ISBN: 9780511809088. DOI: 10.1017/CB09780511809088.
- [8] M. M. Deza i E. Deza. *Encyclopedia of Distances*. Berlin, Germany: Springer Berlin Heidelberg, 2016. ISBN: 9783662528440. DOI: 10.1007/978-3-662-52844-0.
- [9] R. O. Duda, Peter E. Hart i D. G. Stork. *Pattern Classification*. 2 wyd. Nashville, TN: John Wiley & Sons, 2012. ISBN: 9780471056690.
- [10] H. Duminil-Copin i S. Smirnov. “The connective constant of the honeycomb lattice equals $\sqrt{2 + \sqrt{2}}$ ”. *Annals of Mathematics* 175 (2012), s. 1653–1665. DOI: 10.4007/annals.2012.175.3.14.
- [11] U. Endriss i U. Grandi. “Graph Aggregation”. *Artificial Intelligence* 245 (2017), s. 86–114. DOI: 10.1016/j.artint.2017.01.001.
- [12] A. C. Gillie. *Binary Arithmetic and Boolean Algebra*. New York, NY, USA: McGraw-Hill, 1965. ISBN: 9780070232488.
- [13] A. Goldsmith. *Wireless Communications*. Cambridge, England: Cambridge University Press, 2005. ISBN: 9780521837163.
- [14] I. Goodfellow, Y. Bengio i A. Courville. *Deep Learning*. London, England: MIT Press, 2016. ISBN: 9780262035613.

- [15] M. Grabisch, J.-L. Marichal, R. Mesiar i E. Pap. *Aggregation Functions*. Encyclopedia of Mathematics and its Applications 127. Cambridge, England: Cambridge University Press, 2009. ISBN: 9780521519267.
- [16] M. Gągolewski. *Data Fusion: Theory, Methods, and Applications*. Warsaw, Poland: Institute of Computer Science, Polish Academy of Sciences, 2022. DOI: 10.48550/ARXIV.2208.01644.
- [17] J. Y. Halpern, M. Vardi, R. Fagin i Y. Moses. *Reasoning About Knowledge*. London, England: MIT Press, 1995. ISBN: 9780262061629.
- [18] F. Harary. *Graph Theory*. New York, NY, USA: Addison Wesley Longman Publishing, 1969. ISBN: 9780201027877.
- [19] N. Hartsfield i G. Ringel. *Pearls in graph theory: A comprehensive introduction*. Dover Publications, 2013. ISBN: 9780486315522.
- [20] E. Harzheim. *Ordered Sets*. Advances in Mathematics. New York, NY, USA: Springer, 2005. ISBN: 9780387242194. DOI: 10.1007/b104891.
- [21] T. Hastie, R. Tibshirani i J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction, second edition*. 2 wyd. New York, NY: Springer, 2017. ISBN: 9780387848570.
- [22] J. Hopcroft, R. Motwani i J. Ullman. *Wprowadzenie do teorii automatów, języków i obliczeń*. Wydawnictwo Naukowe PWN, 2012. ISBN: 9788301145026.
- [23] A. Janner i E. Ascher. “Bravais Classes of Two-Dimensional Relativistic Lattices”. *Physica* 45.1 (1969), s. 33–66. DOI: 10.1016/0031-8914(69)90061-5.
- [24] N. R. Jennings. “On agent-based software engineering”. *Artificial intelligence* 117.2 (2000), 277–296. ISSN: 0004-3702. DOI: 10.1016/s0004-3702(99)00107-1.
- [25] L. Kaufman i P. Rousseeuw. *Finding Groups in Data: An Introduction To Cluster Analysis*. Wiley, 1990. ISBN: 471878766. DOI: 10.1002/9780470316801.
- [26] E. P. Klement, R. Mesiar i E. Pap. *Triangular norms*. Dordrecht, Netherlands: Kluwer Academic Publishers, 2000. DOI: 10.1007/978-94-015-9540-7.
- [27] M. Maschler, E. Solan i S. Zamir. *Game Theory*. Cambridge, England: Cambridge University Press, 2013. ISBN: 9780511794216. DOI: 10.1017/cbo9780511794216.
- [28] A. Mathis, M. B. Stemmler i A. Herz. “Probable nature of higher-dimensional symmetries underlying mammalian grid-cell activity patterns”. *eLife* 4 (2015). ISSN: 2050-084X. DOI: 10.7554/eLife.05979.
- [29] G. Moś. “An overview of honeycomb-based graph aggregation functions”. *Book of Abstracts. IFSA-EUSFLAT 2021 jointly with AGOP, IJCRS, and FQAS*. Eds.: R. Mesiar, M. Reformat i M. Štěpnička. University of Ostrava, 2021, s. 78. ISBN: 9788075992697.

- [30] G. Moś. “Exploring Vertex Representation and Cardinality of Aggregation Functions in Honeycomb-based Polygonal Chains”. *Fuzzy Logic and Technology, and Aggregation Operators. 13th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2023, and 12th International Summer School on Aggregation Operators, AGOP 2023, Palma de Mallorca, Spain, September 4–8, 2023, Proceedings*. Eds.: S. Massanet, S. Montes, D. Ruiz-Aguilera i M. González-Hidalgo. Springer Cham, 2023, s. 63. DOI: 10.1007/978-3-031-39965-7.
- [31] G. Moś. “Honeycomb-Based Polygonal Chains Aggregation Functions”. *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems; IPMU 2022*. Eds.: D. Ciucci i et. al. Communications in Computer and Information Science (CCIS). Springer, 2022, s. 384–397. ISBN: 9783031089718. DOI: 10.1007/978-3-031-08971-8_32.
- [32] G. Moś. “Similarity Functions and Medoids for Honeycomb-Based Structures”. *Book of Abstracts of The Seventeenth International Conference on Fuzzy Set Theory and Applications. FSTA 2024, Liptovský Ján, Slovakia, January 28 – February 2, 2024*. Eds.: A. Stupňanová, M. Dyba i V. Pavliska. University of Ostrava, 2024, s. 58. ISBN: 9788075994707.
- [33] G. Moś. “Uncertainty of Aggregation: Investigating Dissimilarity between Honeycomb-Based Polygonal Chains”. *Progress in Polish Artificial Intelligence Research 6*. Eds.: R. Doroz i B. Zielosko. Wydawnictwo Uniwersytetu Śląskiego, 2025, s. 123–130. DOI: 10.31261/PN.4331.
- [34] T. Ndjountche. *Digital Electronics 2: Sequential and Arithmetic Logic Circuits*. Hoboken, NJ, USA: ISTE Ltd/John Wiley i Sons Inc, 2016. ISBN: 9781848219854. DOI: 10.1002/9781119329756.
- [35] C. O’Keefe, S. Pereverzyev i R. Anderssen. “A hexagonal tiles recursion motivated by a problem in biology”. *The Mathematical Scientist* 36 (czerwiec 2011).
- [36] M. O’Searcoid. *Metric spaces*. 2007 wyd. Guildford, England: Springer, 2006. ISBN: 9781447173298.
- [37] S. Roman. *Lattices and Ordered Sets*. New York, NY: Springer, 2010. ISBN: 9780387789002.
- [38] S. Russell i P. Norvig. *Artificial intelligence: A modern approach, global edition*. 4 wyd. London, England: Pearson Education, 2021. ISBN: 9781292401133.
- [39] R. Sadourny, A. Arakawa i Y. Mintz. “Integration of the nondivergent barotropic vorticity equation with an icosahedral-hexagonal grid for the sphere¹”. *Monthly weather review* 96.6 (1968), 351–356. ISSN: 0027-0644. DOI: 10.1175/1520-0493(1968)096<0351:iotnbv>2.0.co;2.
- [40] F. Schwierz, J. Pezoldt i R. Granzner. “Two-dimensional materials and their prospects in transistor electronics”. *Nanoscale* 7.18 (2015), s. 8261–8283. DOI: 10.1039/C5NR01052G.

- [41] V. S. Schøyen, K. Beshkov, M. B. Pettersen, E. Hermansen, K. Holzhausen, A. Malthé-Sørensen, M. Fyhn i M. E. Lepperød. “Hexagons all the way down: grid cells as a conformal isometric map of space”. *PLoS computational biology* 21.2 (2025), e1012804. ISSN: 1553-734X. DOI: 10.1371/journal.pcbi.1012804.
- [42] R. Sedgewick i K. Wayne. *Algorithms*. 4 wyd. Boston, USA: Addison-Wesley Educational, 2011.
- [43] A. Spry. “The origin of columnar jointing, particularly in basalt flows”. *Journal of the Geological Society of Australia* 8.2 (1962), 191–216. ISSN: 0016-7614. DOI: 10.1080/14400956208527873.
- [44] A. Surdyk. “Ludologia – naukowe badania gier. Przeszłość, terażniejszość, przyszłość”. *Images The International Journal of European Film Performing Arts and Audiovisual Communication* 33.42 (2023), 49–69. ISSN: 1731-450X. DOI: 10.14746/i.2023.33.42.3.
- [45] Klaus Teuber. *Catan*. Board game. 1995.
- [46] R. J. Wilson. *Introduction to graph theory*. 3 wyd. Nashville, TN: John Wiley & Sons, 1986. ISBN: 9780470206164.
- [47] M. Wooldridge. *An Introduction to MultiAgent Systems*. 2 wyd. Chichester, England: John Wiley & Sons, 2009. ISBN: 9780470519462.
- [48] R. K. Yadav i J. Bloxham. “Deep rotating convection generates the polar hexagon on Saturn”. *Proceedings of the National Academy of Sciences of the United States of America* 117.25 (2020), 13991–13996. ISSN: 0027-8424. DOI: 10.1073/pnas.2000317117.
- [49] J. Zhao, P. Ji, Y. Li i in. “Ultra-high-mobility semiconducting epitaxial graphene on silicon carbide”. *Nature* 625 (2024), s. 60–65. DOI: 10.1038/s41586-023-06811-0.

Spis rysunków

| | | |
|-----|--|----|
| 1.1 | Trójkąt o wierzchołkach $p = (x, y)$, $p_0 = (x_0, y_0)$ oraz $p_1 = (x_1, y_1)$, gdzie $x_0, x_1 \geq x$ oraz $y_0, y_1 \geq y$, przedstawiony w kartezjańskim układzie współrzędnych z zaznaczonym kątem α utworzonym przez odcinki o końcach w punktach p i p_0 oraz p i p_1 | 19 |
| 1.2 | Trójkąt o wierzchołkach $p = (x, y)$, $p_0 = (x_0, y_0)$ oraz $p_1 = (x_1, y_1)$, gdzie $x_0, x_1 \geq x$ oraz $y_0, y_1 \geq y$, przedstawiony w kartezjańskim układzie współrzędnych, gdzie punktem przecięcia się osi X oraz Y jest punkt p , z zaznaczonymi kątami: α utworzonym przez wierzchołki p_0 , p i p_1 , β utworzonym przez wierzchołki (x_0, y) , p i p_0 oraz γ utworzonym przez wierzchołki (x_1, y) , p i p_1 | 20 |
| 1.3 | Trójkąt o wierzchołkach $p = (x, y)$, $p_0 = (x_0, y_0)$ i $p_1 = (x_1, y_1)$, gdzie $x_0 \geq x$, $x_1 \leq x$ i $y_0, y_1 \geq y$, przedstawiony w kartezjańskim układzie współrzędnych, gdzie punktem przecięcia się poziomej osi i Y jest punkt p , z zaznaczonymi kątami: α utworzonym przez wierzchołki p_0 , p i p_1 , β utworzonym przez wierzchołki p_0 , p i (x, y_0) oraz γ utworzonym przez wierzchołki (x, y_1) , p i p_1 | 22 |
| 1.4 | Trójkąt o wierzchołkach $p = (x, y)$, $p_0 = (x_0, y_0)$ i $p_1 = (x_1, y_1)$, gdzie $x_0 \geq x$, $x_1 \leq x$, $y_0 \leq y$ i $y_1 \geq y$, przedstawiony w kartezjańskim układzie współrzędnych, gdzie punktem przecięcia się poziomej osi i Y jest punkt p , z zaznaczonymi kątami: α utworzonym przez wierzchołki p_0 , p i p_1 , β utworzonym przez wierzchołki p_0 , p i (x_0, y) oraz γ utworzonym przez wierzchołki (x, y_1) , p i p_1 | 23 |
| 1.5 | Heksastrukt o dwóch wierzchołkach stopnia pierwszego i pozostałych wierzchołkach stopnia drugiego o pokryciu ścieżką reprezentowaną przez ciąg binarny 11110000. | 25 |
| 1.6 | Układ współrzędnych z zaznaczoną prostą l pozytywnie nachyloną do poziomej osi i przechodzącą przez punkt p , który tworzy z punktami p_0 , p_1 trójkąt, zaznaczonym punktem (x_0, y) oraz oznaczonymi kątami: α utworzonym przez prostą l oraz poziomą oś, β utworzonym przez punkty p_0 , p , (x_0, y) oraz γ utworzonym przez punkty p_0 , p i prostą l oraz punkty p_1 , p i prostą l . Punkt p_0 leży w czwartej ćwiartce układu współrzędnych względem punktu p . Punkt p_1 powstał przez obrót punktu p_0 względem punktu p o kąt 2γ | 27 |

- 1.7 Układ współrzędnych z zaznaczoną prostą l pozytywnie nachyloną do poziomej osi i przechodzącą przez punkt p , która tworzy z punktami p_0, p_1 trójkąt, zaznaczonym punktem (x_0, y) oraz oznaczonymi kątami: α utworzonym przez prostą l oraz poziomą oś, β utworzonym przez punkty $p_0, p, (x_0, y)$ oraz γ utworzonym przez punkty p_0, p i prostą l oraz punkty p_1, p i prostą l . Punkt p_0 leży w pierwszej ćwiartce układu współrzędnych względem punktu p poniżej prostej l . Punkt p_1 powstał przez obrót punktu p_0 względem punktu p o kąt 2γ 29
- 1.8 Układ współrzędnych z zaznaczoną prostą l pozytywnie nachyloną do poziomej osi i przechodzącą przez punkt p , który tworzy z punktami p_0, p_1 trójkąt, zaznaczonym punktem (x_0, y) oraz oznaczonymi kątami: α utworzonym przez prostą l oraz poziomą oś, β utworzonym przez punkty $p_0, p, (x_0, y)$ oraz γ utworzonym przez punkty p_0, p i prostą l oraz punkty p_1, p i prostą l . Punkt p_0 leży w pierwszej ćwiartce układu współrzędnych względem punktu p powyżej prostej l . Punkt p_1 powstał przez obrót punktu p_0 względem punktu p o kąt -2γ 30
- 1.9 Układ współrzędnych z zaznaczoną prostą l pozytywnie nachyloną do poziomej osi i przechodzącą przez punkt p , który tworzy z punktami p_0, p_1 trójkąt, zaznaczonym punktem (x_0, y) oraz oznaczonymi kątami: α utworzonym przez prostą l oraz poziomą oś, β utworzonym przez punkty $p_0, p, (x_0, y)$ oraz γ utworzonym przez punkty p_0, p i prostą l oraz punkty p_1, p i prostą l . Punkt p_0 leży w drugiej ćwiartce układu współrzędnych względem punktu p powyżej prostej l tworząc z nią kąt $\gamma \leq \frac{\pi}{2}$. Punkt p_1 powstał przez obrót punktu p_0 względem punktu p o kąt -2γ 31
- 2.1 Drzewo ternarne o korzeniu będącym rozszerzonym ciągiem binarnym $E_0 = B_1(B_4(B_{10}(E_{11})E_{12})B_{13}(E_{14})E_{15})B_7(B_{16}(E_{17})E_{18})B_{19}(E_{20})E_{21}$, którego dziećmi są $B_1, E_2 = B_4(B_{10}(E_{11})E_{12})B_{13}(E_{14})E_{15}$ i $E_3 = B_7(B_{16}(E_{17})E_{18})B_{19}(E_{20})E_{21}$. Ciągi $B_4, E_5 = B_{10}(E_{11})E_{12}$ i $E_6 = B_{13}(E_{14})E_{15}$ są dziećmi E_2 oraz $B_7, E_8 = B_{16}(E_{17})E_{18}$ i $E_9 = B_{19}(E_{20})E_{21}$ są dziećmi E_3 . Ciągi B_{10}, E_{11} i E_{12} są dziećmi E_5 , ciągi B_{13}, E_{14} i E_{15} są dziećmi E_6 , ciągi B_{16}, E_{17} i E_{18} są dziećmi E_8 oraz ciągi B_{19}, E_{20} , i E_{21} są dziećmi E_9 42
- 2.2 Drzewo ternarne przedstawiające rozwinięcie rozszerzonego ciągu binarnego $00(110(000(001)010)011(100)101)01(10)11$ 43
- 3.1 Sześciokąt foremny o środku w punkcie (x, y) otoczony przylegającymi sześciokątami foremnymi z zaznaczonymi środkami. Wszystkie sześciokąty mają boki długości 1. 57
- 3.2 Sześciokąt foremny o środku w punkcie (x, y) i o bokach długości 1 z zaznaczonymi dwoma wierzchołkami o współrzędnych $(x, y + 1)$ i $(x + \frac{\sqrt{3}}{2}, y + \frac{1}{2})$ 58

| | | |
|------|---|----|
| 3.3 | Heksastrukt złożony z pięciu wierzchołków, gdzie v_0, v_3 i v_4 są wierzchołkami stopnia pierwszego, v_1 jest wierzchołkiem stopnia drugiego oraz v_2 jest wierzchołkiem stopnia trzeciego. | 58 |
| 3.4 | Heksastrukt o dwóch wierzchołkach stopnia pierwszego i pozostałych wierzchołkach stopnia drugiego po lewej stronie oraz heksastrukt o wierzchołkach stopnia drugiego po prawej stronie. | 59 |
| 3.5 | Heksastrukt niezawierający wierzchołków stopnia pierwszego z zaznaczoną kolorem czerwonym ścieżką rzędu drugiego łączącą wierzchołek stopnia trzeciego z wierzchołkiem stopnia trzeciego po lewej stronie. Heksastrukt z zaznaczoną kolorem czerwonym ścieżką rzędu drugiego łączącą wierzchołek stopnia pierwszego z wierzchołkiem stopnia trzeciego po prawej stronie. . . | 61 |
| 3.6 | Heksastrukt H z wierzchołkiem zaznaczonym kolorem czerwonym, dla którego nie istnieje cykl w H podzielony na dwa podheksastryki względem tego wierzchołka. | 62 |
| 3.7 | Pokrycie $\{p, q, r\}$ heksastrukt H , które nie jest właściwe, gdyż nie istnieją dokładnie dwie różne od p ścieżki o początku w $\overset{\circ}{p}$ oraz nie istnieje ścieżka o końcu w $\overset{\circ}{r}$ dla ustalonej ścieżki p | 63 |
| 3.8 | Heksastrukt złożony z dwóch sześciokątów o wspólnej krawędzi (po lewej stronie). Pokrycie właściwe tego heksastrukt o początku zaznaczonym kolorem czerwonym, w którym pierwszym wierzchołkiem jest pierwszy wierzchołek stopnia trzeciego (środkowy). Pokrycie właściwe tego samego heksastrukt o początku zaznaczonym kolorem czerwonym, w którym pierwszym wierzchołkiem jest drugi wierzchołek stopnia trzeciego (po prawej stronie). | 66 |
| 3.9 | Pokrycie właściwe $P = \{p, p_1, p_2, q, r\}$ (po prawej) heksastrukt (po lewej) o początku w p , dla którego nie istnieje pokrycie właściwe o początku w p^{-1} , zawierające ścieżki $s \in P$ lub $s^{-1} \in P$. Kropkami oznaczono początki poszczególnych ścieżek. Na czerwono zaznaczono początek pokrycia P | 66 |
| 3.10 | Heksastrukt o dwóch wierzchołkach stopnia pierwszego i pozostałych wierzchołkach stopnia drugiego, o pokryciu ścieżką reprezentowaną ciągiem binarnym 11110000, o początku zaznaczonym kropką. | 67 |
| 3.11 | Przykładowe właściwe pokrycie ścieżkowe (po prawej) heksastrukt złożonego z trzech sześciokątów o wspólnym wierzchołku (po lewej). Kolorem czerwonym zaznaczono ścieżkę początkową. Kropkami oznaczono pierwsze wierzchołki poszczególnych ścieżek. | 68 |
| 3.12 | Heksastryki reprezentowane przez ciągi binarne, odpowiednio od lewej do prawej, 01111 oraz $\tau(01111) = 10000$ | 69 |
| 3.13 | Heksastryki reprezentowane przez ciągi binarne, odpowiednio od lewej do prawej, 00001 oraz $\tau(01111) = 11110$ | 70 |
| 3.14 | Heksastryki reprezentowane przez rozszerzone ciągi binarne, odpowiednio od lewej do prawej, 01(0)11 i $\tau(01(0)11) = 10(00)1$ | 72 |
| 3.15 | Heksastryki reprezentowane przez rozszerzone ciągi binarne, odpowiednio od lewej do prawej, 1(11)01 i $\pi(1(11)01) = 00(01)0$ | 72 |

| | | |
|------|---|----|
| 3.16 | Heksastrukt reprezentowany przez rozszerzony ciąg binarny $\zeta(01(0)11) = 11(1)10$ | 72 |
| 4.1 | Struktury reprezentowane przez trzelementowe ciągi binarne, uporządkowane leksykograficznie wierszami od lewej górnej struktury do prawej dolnej struktury. Na czerwono zaznaczono krawędzie początkowe rozważanych ciągów binarnych. | 90 |
| 4.2 | Struktury reprezentowane przez ciągi binarne 0, 00, 000, 0000, 0001, 001, 01 oraz 1, uporządkowane leksykograficznie wierszami od lewej górnej struktury do prawej dolnej struktury. Na czerwono zaznaczono krawędzie początkowe rozważanych ciągów binarnych. | 90 |
| 4.3 | Po lewej stronie przedstawiono heksastrukt reprezentowany przez ciąg binarny 1111101111. Po prawej stronie pokazano nieprawidłowe heksastryki powstałe z heksagonu przedstawionego po lewej stronie poprzez dodanie krawędzi lewostronnej lub krawędzi prawostronnej na końcu. Pośrodku przedstawiono nieprawidłowy heksastrukt, w którym ostatnia krawędź pokrywa się z pierwszą krawędzią, a jego reprezentacją binarną mógłby być ciąg 111111. | 91 |
| 4.4 | Wszystkie heksastryki reprezentowane przez rozszerzone ciągi binarne złożone z jednoelementowych ciągów binarnych: 0(0)0, 0(0)1, 0(1)0, 0(1)1, 1(0)0, 1(0)1, 1(1)0 oraz 1(1)1. Na czerwono zaznaczono krawędzie początkowe rozważanych rozszerzonych ciągów binarnych. | 91 |
| 4.5 | Diagramy Hassego dla wszystkich rozszerzonych ciągów binarnych złożonych z jednoelementowych ciągów binarnych. Porządek leksykograficzny wykorzystano w diagramie po lewej stronie. Porządek epaleksykograficzny wykorzystano w diagramie po prawej stronie. | 92 |
| 4.6 | Struktury reprezentowane przez trzelementowe ciągi binarne, posortowane rosnąco z wykorzystaniem porządku epaleksykograficznego wierszami od lewej górnej struktury do prawej dolnej struktury. | 94 |
| 4.7 | Struktury reprezentowane przez ciągi binarne 0, 00, 000, 0000, 0001, 001, 01 oraz 1, posortowane rosnąco z wykorzystaniem porządku epaleksykograficznego wierszami od lewej górnej struktury do prawej dolnej struktury. | 94 |
| 5.1 | Obraz pojedynczej warstwy grafenu otrzymany przy użyciu mikroskopu elektronowego. Źródło: https://www.flickr.com/photos/armymaterielcommand/6795812766 . Licencja: CC BY-ND 2.0. | 98 |
| 5.2 | Heksagonalna burza na Saturnie. Źródło: https://www.flickr.com/photos/kevinmgill/24588767787 . Licencja: CC BY-ND 2.0. | 98 |
| 5.3 | Przekrój poprzeczny selera naciowego z widocznymi komórkami o kształcie sześciokątnym. Źródło: https://www.flickr.com/photos/cbaek/6118835197 . Licencja: CC BY-ND 2.0. | 99 |
| 5.4 | Bazaltowe kolumny o kształcie sześciokątnym. Źródło: https://www.geograph.org.uk/photo/1478562 . Licencja: CC BY-ND 2.0. | 99 |

- 5.5 Kafelki w kształcie sześciokątów, z których układana jest mapa do gry w Catan (po lewej stronie). Żetony, odpowiednio od lewej góry do prawego dołu, reprezentują: pustynię, góry, pastwiska, pola uprawne, wzgórza i lasy. Podstawowa konfiguracja żetonów terenów w mapę Catan (po prawej stronie). 101
- 5.6 Po lewej u góry przedstawiono 1001 heksastruktur reprezentujących sieci dróg stworzone przez graczy, nałożone na siebie i przedstawione na podstawowej mapie Catan, z początkiem zaznaczonym czerwonym punktem oraz krawędziami o odcieniach czerwieni o nasyceniu $1/1001$ dla każdej ścieżki. Kolejno, po prawej u góry, po lewej na dole oraz po prawej na dole, przedstawiono medoidy dla 1001 heksastruktur otrzymanych w symulacji, zaprezentowane na podstawowej mapie Catan, z początkiem zaznaczonym czerwonym punktem oraz krawędziami o kolorze czerwonym, wyznaczone z wykorzystaniem odległości, odpowiednio, Levenshteina, rozszerzonej Hamminga oraz rozszerzonej ważonej Hamminga. 102
- 5.7 Heksastrukty reprezentowane przez ciągi binarne, z początkiem zaznaczonym kolorem czerwonym, odpowiednio od góry do dołu: 0101100110101001, 0101101010100101, 0110101001100101, 1010011001100101, 1001101001010110, 1001100110011001. 105
- 5.8 Heksastrukty reprezentowane przez ciągi binarne: 0101100110101001, 0101101010100101, 0110101001100101, 1010011001100101, 1001101001010110 oraz 1001100110011001, nałożone na siebie w taki sposób, aby ich krawędzie początkowe pokrywały się, tworząc sieć systemu wieloagentowego. Kolejno od góry do dołu przedstawiono sieć z krawędzią początkową zaznaczoną kolorem czerwonym oraz sieci z zaznaczoną kolorem czerwonym ścieżką będącą infimum oraz supremum względem porządku leksykograficznego, a także infimum oraz supremum względem porządku epaleksykograficznego. 106
- 5.9 Odpowiednio od lewej do prawej przedstawiono wzory strukturalne (R)-3-metyloheksanu i (S)-3-metyloheksanu. Pogrubiona krawędź oznacza wiązanie wychodzące z płaszczyzny. Przerywana krawędź oznacza wiązanie chowające się za płaszczyzną. 108
- 5.10 Odpowiednio od lewej do prawej przedstawiono heksastrukty reprezentowane przez rozszerzone ciągi binarne (01)1 i (0)10 z początkami zaznaczonymi kolorem czerwonym. 108
- 5.11 Wzór szkieletowy 4-propylooktanu. 109
- A.1 Ścieżka heksastruktury $\{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ reprezentowana przez ciąg binarny 111000 o wierzchołkach początkowych $v_0 = [1 \ 1]^T$ oraz $v_1 = [1 \ 2]^T$ zaznaczona kolorem czerwonym przedstawiona w kartezjańskim układzie współrzędnych. 118

| | | |
|-----|---|-----|
| A.2 | Fragment heksagonalnego układu współrzędnych, gdzie pierwsza składowa współrzędnej jest współczynnikiem stojącym przy I , a druga składowa współrzędnej jest współczynnikiem stojącym przy $\text{Rot}(120^\circ)$ we wzorze (1.1.2) na obliczanie współrzędnych wierzchołków heksastruktów ze względu na ustaloną parę wierzchołków $(1, 0)$ i $(0, 0)$ połączonych krawędzią. Na czerwono zaznaczono osie, tj. linie, dla których pierwsza współrzędna lub druga współrzędna jest równa zero. | 125 |
| A.3 | Ścieżki heksastruktów reprezentowane przez ciągi binarne, odpowiednio od lewej do prawej, 00000 oraz 11111. Na czerwono zaznaczyliśmy wierzchołki oraz krawędzie początkowe. Przerywaną linią zaznaczyliśmy krawędzie powodujące kolizję z wierzchołkami początkowymi. | 131 |
| A.4 | Kształty heksastruktów z jedną kolizją. Po lewej stronie heksastrukt reprezentowany przez ciągi binarne o długości 9 (np. 1111011110). Po prawej stronie heksastrukt reprezentowany przez ciągi binarne o długości 11 (np. 111011101110). | 134 |
| A.5 | Kształty heksastruktów z jedną kolizją reprezentowane przez ciągi binarne o długości 13. Przykładowymi ciągami binarnymi reprezentującymi heksastrukty od lewej do prawej, to, odpowiednio, 11110101111010 oraz 1110110111100. | 136 |

Skorowidz implementacji

| | | |
|-------|--|-----|
| B.1.1 | Implementacja funkcji τ dla ciągu binarnego w postaci ciągu znaków. | 139 |
| B.1.2 | Implementacja funkcji τ dla ciągu binarnego w postaci liczby całkowitej z określoną długością ciągu. | 140 |
| B.2.1 | Implementacja funkcji ς dla ciągu binarnego w postaci ciągu znaków. | 140 |
| B.2.2 | Implementacja funkcji ς dla ciągu binarnego w postaci liczby całkowitej z określoną długością ciągu z wykorzystaniem operacji na ciągach znaków. | 140 |
| B.2.3 | Implementacja funkcji ς dla ciągu binarnego w postaci liczby całkowitej z określoną długością ciągu z wykorzystaniem operacji binarnych. | 140 |
| B.3.1 | Implementacja funkcji π dla ciągu binarnego w postaci ciągu znaków. | 141 |
| B.3.2 | Implementacja funkcji π dla ciągu binarnego w postaci liczby całkowitej z określoną długością ciągu. | 141 |
| B.4.1 | Podstawowa implementacja rozszerzonego ciągu binarnego w postaci klasy <code>HexSeq</code> wraz z metodą pozwalającą na jego wyświetlenie w postaci ciągu znaków. | 142 |
| B.4.2 | Metody klasy <code>HexSeq</code> pozwalające na pobranie poszczególnych podciągów rozszerzonego ciągu binarnego w postaci obiektu <code>HexSeq</code> | 142 |
| B.4.3 | Metoda klasy <code>HexSeq</code> sprawdzająca, czy obiekt istotnie reprezentuje rozszerzony ciąg binarny, czy ciąg binarny. | 143 |
| B.4.4 | Metody klasy <code>HexSeq</code> zwracające składowe rozszerzonego ciągu binarnego. | 143 |
| B.4.5 | Metoda klasy <code>HexSeq</code> zwracająca jego złożoność. | 143 |
| B.4.6 | Metoda klasy <code>HexSeq</code> zwracającą rozszerzony ciąg binarny będący obrazem funkcji τ | 143 |
| B.4.7 | Metoda klasy <code>HexSeq</code> zwracającą rozszerzony ciąg binarny będący obrazem funkcji ς | 144 |
| B.4.8 | Metoda klasy <code>HexSeq</code> zwracającą rozszerzony ciąg binarny będący obrazem funkcji π | 144 |
| B.5.1 | Funkcja obliczająca odległość między dwoma rozszerzonymi ciągami binarnymi dla danej metryki mierzącej odległość między ciągami binarnymi (implementacja wzoru 4.1.1). | 144 |
| B.6.1 | Zdefiniowanie podstawowej mapy Catan z wykorzystaniem obiektów z <code>Catanatronu</code> | 146 |
| B.6.2 | Zredefiniowana funkcja symulująca gry poprzez wdrożenie podstawowej mapy Catan w linii 9. | 146 |

| | | |
|-------|---|-----|
| B.6.3 | Utworzenie gracza z wykorzystaniem obiektów Catanatronu ze wskazaniem preferowanej pierwszej ścieżki do zbudowania oraz ogarniczeniem akcji budowania drogi do budowania spójnej sieci dróg. | 147 |
| B.6.4 | Określenie graczy oraz przeprowadzenie 100 symulacji z wykorzystaniem Catanatronu. | 147 |
| B.7.1 | Implementacja metody zliczania ciągów binarnych o długościach od 0 do danej liczby naturalnej DEPTH reprezentujących heksastryki zawierające dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego. . . | 148 |
| B.7.2 | Warunek pozwalający na zliczenie ciągów binarnych niezmienniczych ze względu na rotacje reprezentujących heksastryki zawierające dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego. | 150 |
| B.7.3 | Warunek pozwalający na zliczenie ciągów binarnych niezmienniczych ze względu na rotacje reprezentujących heksastryki zawierające dwa wierzchołki stopnia pierwszego i pozostałe wierzchołki stopnia drugiego. | 150 |

Skorowidz symboli

$B_1(E_2)E_3$, 39

deg, 55

\leq , 34, 87, 88

$|s|$, 16

\mathcal{B} , 37

\mathfrak{E} , 86

Ref, 28

Rot, 24

ω , 39

$\overrightarrow{p_0p_1}$, 16

π , 37, 49

$\overline{p_0p_1}$, 16

τ , 37, 48

ζ , 37, 48

ϑ , 64

d , 73

$d_{H^*}^0$, 82

d_H , 78

d_L , 85

$d_{H^*}^m$, 79

d_d^* , 74

d_{wH^*} , 82, 84

p , 56

$\overset{p}{\rightsquigarrow}$, 56

$\overset{\circ}{p}$, 56

$\overset{\circ}{\rightarrow} p$, 56

Skorowidz nazw

ciąg binarny, 37

 ścieżka początkowa, 67

cykl, 56

element

 maksymalny, 34

 minimalny, 34

 najmniejszy, 34

 największy, 34

graf, 55

 spójny, 56

heksastrukt, 57

 reprezentowany przez ciąg binarny, 67

 reprezentowany rozszerzonym ciągiem binarnym, 68

infimum, 34

konkatenacja, 37

krawędź, 55

 sąsiad, 55

kąt

 lewostronny, 24

 prawostronny, 24

macierz

 odbicia, 28

 rotacji, 24

metryka, 73

 rozszerzenie, 74

odcinek, 16

 długość, 16

odległość

 Hamminga, 78

 Levenshteina, 85

- rozszerzenie, 74
- rozszerzona Hamminga, 82
- rozszerzona ważona Hamminga, 84
- odwrócenie ciągu, 37
- ograniczenie
 - dolne, 34
 - górne, 34
- permutacja, 37
- podgraf, 55
- podheksastrukt, 57
 - rozłączny, 57
- podścieżka, 56
- pokrycie
 - maksymalna liczba kolejno połączonych ścieżek, 64
- pokrycie właściwe heksastrukt, 62
 - krawędź początkowa, 62
 - początek, 62
- pokrycie ścieżkowe, 56
- porządek
 - częściowy, 34
 - liniowy, 34
- porządek leksykograficzny, 15
- prosta, 16
 - przechodząca przez punkty, 17
- punkt, 15
 - na lewo od prostej, 16
 - na lewo od wektora, 17
 - na prawo od prostej, 16
 - na prawo od wektora, 17
 - na prostej, 16
 - na przedłużeniu wektora, 17
 - odbicie lustrzane, 29
 - poniżej prostej, 16
 - powyżej prostej, 16
- rozszerzony ciąg binarny, 38
 - krawędź początkowa, 68
 - podciągi, 39
 - rozwiniecie, 42
 - skończony, 39
 - składowe, 41
- supremum, 34

wektor, 16

wierzchołek

koniec ścieżki, 56

początek ścieżki, 56

stopień, 55

zamiana bitów, 37

ścieżka, 56

reprezentowana ciągiem binarnym, 67

rzędu drugiego, 56