

Uniwersytet im. Adama Mickiewicza w Poznaniu

Wydział Matematyki i Informatyki

Dyscyplina naukowa: matematyka

Dziedzina nauk ścisłych i przyrodniczych

Rozprawa doktorska

Grzegorz Adamski

Oszacowania rozgrywanych liczb Ramseya

Bounds for online Ramsey numbers

Promotor: prof. UAM dr hab.

Małgorzata Bednarska-Bzdęga

Poznań 2025

Spis treści

1	Wprowadzenie	1
1.1	Struktura pracy	3
2	Potencjał i półpotencjał	4
3	Zastosowania	10
4	Cykle nieparzyste vs. graf spójny	19
4.1	Gra $\mathcal{G}(n, m)$	19
4.2	Dowód twierdzenia 4.1	27
5	Trójkąt vs. ścieżka – oszacowanie górne	28
6	Trójkąt vs. graf spójny – oszacowanie górne	34
7	Granty i darowizny	35
8	Dokładne wartości $\tilde{r}(C_4, P_n)$	39
9	Gra $\mathbf{RR}(C_4, P_n, H)$	40
10	Dowód twierdzenia 9.1	44
10.1	G jest pusty	44
10.1.1	Brak niebieskich krawędzi w ciągu pierwszych siedmiu rund	45
10.1.2	Pierwsza niebieska krawędź w rundzie 7	46
10.1.3	Pierwsza niebieska krawędź w rundzie 6	47
10.1.4	Pierwsza niebieska krawędź w rundzie 5	48
10.1.5	Pierwsza niebieska krawędź w rundzie 4	49
10.1.6	Pierwsza niebieska krawędź w rundzie 3	51
10.1.7	Pierwsza niebieska krawędź w rundzie 2	51
10.1.8	Pierwsza niebieska krawędź w rundzie 1	52

10.2	G nie jest pusty	52
10.2.1	G jest b- lub br-ścieżką	53
10.2.2	G jest brb-ścieżką	53
10.2.3	G jest brr-ścieżką	54
10.2.4	G jest drb-ścieżką	55
10.2.5	G jest drr-ścieżką	56
10.2.6	G jest dr-ścieżką	57
10.2.7	G jest d-ścieżką	57
11	Algorytm wyznaczający $RR(C_4, P_n, H)$ dla $n \leq 13$	57
11.1	Idea algorytmu	58
11.2	Weryfikacja	60
11.3	Informacje techniczne	62
12	Dowód twierdzenia 8.1	62
13	Uwagi końcowe	63
A	Kod programu wyznaczającego $RRC(C_4, P_n, H, v, e)$	67

STRESZCZENIE

Praca skupia się na oszacowaniach rozgrywanych liczb Ramseya. Dowodzimy m.in., że dla dowolnego grafu G , który nie jest dwudzielny oraz dowolnego grafu spójnego H zachodzi $\tilde{r}(G, H) \geq \varphi v(H) + e(H) - 2\varphi + 1$ oraz znajdujemy dokładne wartości $\tilde{r}(C_4, P_n)$. Ponadto prezentujemy metodę (pół)potencjału, która pozwala znajdować oszacowania rozgrywanych liczb Ramseya i dowodzimy, że jest ona uniwersalna. Pokazujemy też, że wiele ze znanych dowodów dotyczących rozgrywanych liczb Ramseya da się wyrazić w języku tej metody.

Słowa kluczowe: gra konstruktor-malarz, rozgrywana liczba Ramseya, gry na grafach, funkcja potencjału

ABSTRACT

The paper focuses on estimates of online Ramsey numbers. We prove, among other things, that if a graph G is not bipartite H is a connected graph, then $\tilde{r}(G, H) \geq \varphi v(H) + e(H) - 2\varphi + 1$. We also determine the exact values of $\tilde{r}(C_4, P_n)$. Furthermore, we present the (semi)potential method, which provides a way to estimate online Ramsey numbers, and we prove that it is universal. We also show that many known proofs concerning online Ramsey numbers can be expressed within the framework of this method.

Keywords: Builder-Painter game, online Ramsey number, games on graphs, potential function

1 Wprowadzenie

W pracy będziemy zajmować się grą konstruktor-malarz, zdefiniowaną przez Becka w [7]. Dla ustalonych grafów G, H jest to gra, w której w każdej rundzie konstruktor rysuje krawędź, a malarz ją maluje na jeden z dwóch kolorów. Celem konstruktora jest jak najszybsze doprowadzenie do powstania czerwonego kopii grafu G lub niebieskiej kopii grafu H , z kolei malarz stara się tego uniknąć tak długo jak się da.

Będziemy rozważać nieco ogólniejszą wersję tej gry, w której malarz ma do dyspozycji dowolną liczbę kolorów niż dwa, grafów kończących grę może być więcej niż kolorów, a dodatkowo nie zawsze te grafy muszą być jednokolorowe.

Definicja. *Trójkę (G, c, f) , gdzie G jest grafem, c jest liczbą naturalną oraz $f : E(G) \rightarrow [c]$ jest kolorowaniem krawędziowym, gdzie każda krawędź jest pomalowana jednym z c kolorów, będziemy nazywać **c -kolorowym grafem**. Przyjmujemy tutaj, że G może być pusty (nie zawierać krawędzi).*

Dodajmy, że w pracy stosujemy terminologię grafową z [6].

Definicja (Gra konstruktor-malarz). *Dla zbioru c -kolorowych grafów $\mathcal{H}$ rozważamy następującą grę dwuosobową. **Planszą** gry jest c -kolorowy graf, który zawiera $|\mathbb{N}|$ wierzchołków i początkowo jest pusty. W każdej rundzie konstruktor rysuje jedną krawędź, a malarz maluje ją na jeden z c kolorów. Gra kończy się, gdy pewien podgraf planszy będzie izomorficzny z jednym z c -kolorowych grafów należących do $\mathcal{H}$. Celem konstruktora jest minimalizacja liczby rozegranych rund, a cel malarza jest przeciwny. Taką grę będziemy oznaczać przez $\tilde{R}(\mathcal{H})$, a liczbę rozegranych rund przy optymalnej strategii obu graczy będziemy nazywać **rozgrywaną liczbą Ramseya dla $\mathcal{H}$** i oznaczać przez $\tilde{r}(\mathcal{H})$. Jeśli malarz ma strategię, która pozwala na to, by gra się nigdy nie skończyła, to przyjmujemy $\tilde{r}(\mathcal{H}) = \infty$.*

Będziemy też stosowali oznaczenie $\tilde{R}_G(\mathcal{H})$ na podobną grę do powyższej, z jedną różnicą – w tym wariancie początkowa plansza G jest c -kolorowym

grafem powiększonym o $|\mathbb{N}|$ izolowanych wierzchołków. Liczbę ruchów w tej grze przy optymalnej strategii obu graczy będziemy oznaczać przez $\tilde{r}_G(\mathcal{H})$.

Powyższą grę można sformułować również dla multigrafów (tzn. dopuścić krawędzie wielokrotne i pętle). Warto jednak zauważyć, że jeśli zbiór $\mathcal{H}$ składa się wyłącznie z c -kolorowych grafów (a nie multigrafów), to nie ma to wpływu na $\tilde{r}(\mathcal{H})$.

Fakt 1.1. *Jeśli $\mathcal{H}$ nie zawiera multigrafów mających krawędzie wielokrotne, to wartość $\tilde{r}(\mathcal{H})$ nie zależy od tego, czy konstruktor ma prawo rysować krawędzie wielokrotne w grze. Jeśli $\mathcal{H}$ nie zawiera multigrafów mających pętle, to $\tilde{r}(\mathcal{H})$ nie zależy od tego, czy konstruktor ma prawo rysować pętle w grze.*

Dowód. Zwiększenie liczby możliwości dla konstruktora może jedynie zmniejszyć rozgrywaną liczbę Ramseya. Jednak wystarczy, by malarz każdą krawędź wielokrotną malował w ten sam sposób, co pierwszą z nich. Wtedy takie ruchy konstruktora będą zmarnowane (tj. nie będą go przybliżały do końca gry), jeśli żaden z grafów w $\mathcal{H}$ nie zawiera krawędzi wielokrotnych. W przypadku pętli sytuacja jest jeszcze prostsza, ponieważ tego typu ruchy są zawsze ruchami zmarnowanymi. $\square$

Z reguły gra konstruktor-malarz jest rozważana dla $c = 2$, a zbiór $\mathcal{H}$ składa się z dwóch jednokolorowych grafów. Standardowo przyjmuje się też kolory czerwony i niebieski. Dlatego **pokolorowany graf** będzie oznaczał graf dwukolorowy. Nadużywając nieco notacji, do oznaczenia pokolorowanego grafu zamiast $(G, 2, f)$ będziemy używać po prostu G . Ponadto przyjmiemy następującą notację, która pozwala łatwo zapisać jednokolorowy graf.

Definicja. *Dla grafu G , wyrażenie G^r będzie oznaczało pokolorowany graf G , w którym każda krawędź jest czerwona, a G^b – pokolorowany graf G , gdzie każda krawędź jest niebieska. Analogiczny zapis będziemy stosować dla rodziny grafów $\mathcal{H}$, tzn. $\mathcal{H}^r$ i $\mathcal{H}^b$ będą rodzinami pokolorowanych grafów, w których wszystkie krawędzie są odpowiednio czerwone i niebieskie.*

Dla pokolorowanego grafu zdefiniujemy też pojęcia związane z podgrafami złożonymi z krawędzi jednego koloru.

Definicja. *Niech G będzie pokolorowanym grafem. Czerwonym podgrafem G będziemy nazywać graf, który ma te same wierzchołki, co G i te krawędzie, które są czerwone w G . Czerwonym stopniem wierzchołka v w kolorowym grafie G nazywamy stopień w czerwonym podgrafie G . Oznaczamy to przez $d_r(v)$. Z kolei niech $\omega_r(G)$ oznacza liczbę składowych spójności w czerwonym podgrafie G . Analogicznie definiujemy niebieski podgraf, $d_b(v)$ oraz $\omega_b(G)$.*

W literaturze można spotkać się z następującymi zapisami rozgrywanej liczby Ramseya.

- Jeśli G jest grafem, to $\tilde{r}(G) := \tilde{r}(\{G^r, G^b\})$.
- Jeśli G, H są grafami, to $\tilde{r}(G, H) := \tilde{r}(\{G^r, H^b\})$.
- Jeśli $\mathcal{G}, \mathcal{H}$ są rodzinami grafów, to $\tilde{r}(\mathcal{G}, \mathcal{H}) := \tilde{r}(\{G^r, H^b\})$.

W tej pracy będziemy korzystać z drugiego i trzeciego zapisu, jeśli będzie to możliwe. Powody są dwa: są one prostsze i powszechnie używane. Nie będziemy stosować pierwszego zapisu, dzięki czemu $\tilde{r}$ z jednym parametrem będzie zawsze oznaczać, że tym parametrem jest rodzina c -kolorowych grafów.

1.1 Struktura pracy

W rozdziale 2 definiujemy potencjał oraz półpotencjał oraz przedstawimy twierdzenia, które pozwalają wykorzystać je do oszacowań rozgrywanej liczby Ramseya. W rozdziale 3 prezentujemy dowody znanych wyników przy użyciu (pół)potencjałów. Rozdziały 4–7 są na podstawie pracy [2]. W rozdziale 4 przedstawiamy oszacowanie dolne dla $\tilde{r}(\mathcal{C}_{\text{odd}}, \mathcal{C}on_{n,m})$. W rozdziałach 5 i 6 przedstawiamy oszacowania z góry kolejno dla $\tilde{r}(C_3, P_n)$ oraz $\tilde{r}(C_3, \mathcal{C}on_{n,m})$.

Warto zauważyć, że te trzy wartości są powiązane nierównościami

$$\tilde{r}(\mathcal{C}_{\text{odd}}, \mathcal{C}on_{n,n-1}) \leq \tilde{r}(C_3, P_n),$$

$$\tilde{r}(\mathcal{C}_{\text{odd}}, \mathcal{C}on_{n,m}) \leq \tilde{r}(C_3, \mathcal{C}on_{n,m}),$$

dzięki czemu dla wszystkich tych wartości otrzymujemy ograniczenia z obu stron. W rozdziale 7 przedstawiamy intuicję stojącą za potencjałem użytym w rozdziale 4.

Rozdziały 8–12 są na podstawie pracy [3]. Przedstawiamy w nich dokładne wartości $\tilde{r}(C_4, P_n)$. Dla $n \leq 13$ do wyznaczenia oszacowań wykorzystaliśmy program komputerowy, którego kod źródłowy jest w dodatku A.

W rozdziale 13 przedstawiamy krótko (bez dowodów) najważniejsze wyniki z pracy [4] i przedstawiamy problem otwarty dotyczący rozgrywanych liczb Ramseya.

2 Potencjał i półpotencjał

W tym rozdziale zdefiniujemy potencjał i półpotencjał oraz pokażemy, jaki mają one związek z rozgrywaną liczbą Ramseya. Ogólnie rzecz biorąc zarówno potencjał, jak i półpotencjał to funkcja która każdej możliwej pozycji (tj. c -kolorowemu grafowi) w grze konstruktor-malarz przypisuje wartość, która jest pewnym przybliżeniem tego, jak blisko ukończenia gry jest konstruktor. Z przyczyn praktycznych będziemy przyjmowali, że początkowy potencjał (tzn. dla graf pustego) jest równy 0. W przypadku potencjału zakładamy dodatkowo, że będzie on rósł (a przynajmniej nie malał) w trakcie gry.

Jeśli znamy strategię dla malarza, która gwarantuje nam, że szybkość wzrostu potencjału jest ograniczona z góry, to będziemy w stanie znaleźć oszacowanie dolne dla rozgrywanej liczby Ramseya. Z drugiej strony, jeśli:

- znamy strategię dla konstruktora, która gwarantuje, że szybkość wzrostu potencjału będzie ograniczona z dołu,

- wiemy, że każdy c -kolorowy graf, który ma dostatecznie duży potencjał, gwarantuje zakończenie gry,

to będziemy w stanie znaleźć oszacowanie górne.

W przypadku półpotencjału można w analogiczny sposób dowodzić oszacowań rozgrywanej liczby Ramsey'a, choć pewne szczegóły dowodów mogą się czasem skomplikować. Z drugiej strony czasami samo udowodnienie, że zadany półpotencjał jest potencjałem, bywa dość skomplikowane i czasem prościej jest tego nie wykazywać.

Definicja. Niech $\mathbb{G}_c$ będzie zbiorem c -kolorowych podgrafów $K_{\mathbb{N}}$ o skończonej liczbie krawędzi.

Definicja. Funkcję $f : \mathbb{G}_c \rightarrow \mathbb{R}_+ \cup \{\infty\}$ nazwiemy półpotencjałem, gdy spełnia poniższe warunki:

- Dla dowolnych dwóch izomorficznych c -kolorowych grafów $G, H \in \mathbb{G}_c$ zachodzi $f(G) = f(H)$.
- Jeśli graf H powstał z G poprzez usunięcie izolowanych wierzchołków, to $f(G) = f(H)$.
- $f(G) = 0$, jeśli G jest pusty.

Jeśli ponadto spełniony jest warunek

- Jeśli G jest podgrafem H , to $f(G) \leq f(H)$.

to funkcję f będziemy nazywać potencjałem (funkcją potencjału).

Zauważmy, że aby zdefiniować funkcję potencjału, wystarczy zdefiniować ją na skończonych c -kolorowych grafach.

Pokażemy teraz kilka własności funkcji $\tilde{r}$.

Fakt 2.1. Niech $\mathcal{H}_1, \mathcal{H}_2 \subset \mathbb{G}_c$ oraz $K \in \mathbb{G}_c$. Wówczas:

1. Jeśli $\mathcal{H}_1 \subset \mathcal{H}_2$, to $\tilde{r}_K(\mathcal{H}_2) \leq \tilde{r}_K(\mathcal{H}_1)$.

2. Jeśli dla każdego $H_1 \in \mathcal{H}_1$ istnieje taki $H_2 \in \mathcal{H}_2$, że $H_2 \subset H_1$, to $\tilde{r}_K(\mathcal{H}_2) \leq \tilde{r}_K(\mathcal{H}_1)$.
3. $\tilde{r}(\mathcal{H}) < \infty$ wtedy i tylko wtedy, gdy $\mathcal{H}$ zawiera jednokolorowy graf w każdym z c kolorów.

Dowód. Punkty 1. i 2. wynikają wprost z definicji gry konstruktor-malarz. Pokażemy punkt 3. Jeśli $\mathcal{H}$ nie zawiera jednokolorowego grafu w kolorze x , to malarz może używać tylko tego koloru i gra nigdy się nie skończy. Z drugiej strony, jeśli $\mathcal{H}$ zawiera jednokolorowy graf w każdym kolorze, to z twierdzenia Ramsey'a wynika, że konstruktor jest w stanie wymusić zakończenie gry przez narysowanie krawędzi K_n dla dostatecznie dużego n . $\square$

Fakt 2.2 (Nierówność trójkąta dla gier konstruktor-malarz). *Weźmy dowolne $\mathcal{H}_1, \mathcal{H}_2 \subset \mathbb{G}_c$ oraz $G \in \mathbb{G}_c$. Wtedy*

$$\tilde{r}_G(\mathcal{H}_2) \leq \tilde{r}_G(\mathcal{H}_1) + \sup_{H \in \mathcal{H}_1} \tilde{r}_H(\mathcal{H}_2).$$

Dowód. Konstruktor wymusza powstanie jednego z kolorowych grafów $\mathcal{H}_1$ w $\tilde{r}_G(\mathcal{H}_1)$ rund. Następnie, gdy $H \in \mathcal{H}$ pojawi się na planszy, wymusza kolorowy graf z $\mathcal{H}_2$ w $\tilde{r}_H(\mathcal{H}_2)$ rund. $\square$

Definicja. *Dla liczby naturalnej k , funkcji potencjału f i c -kolorowego grafu G , dla którego $f(G) < \infty$, rozważmy grę konstruktor-malarz, która trwa dokładnie k rund, a G jest początkową planszą. Oznaczmy przez G_k planszę po zakończeniu gry. Celem konstruktora w tej grze jest zmaksymalizowanie $f(G_k)$, a malarza – zminimalizowanie tej wartości. Niech*

- $Q_{k,f}(G)$ będzie oznaczeniem powyższej gry.
- $q_{k,f}(G) := f(G_k) - f(G)$ przy założeniu, że obaj gracze grają optymalnie,
- $m_{k,f}(b, a) := \inf\{q_k(H) : f(H) \in [b, a]\}$,
- $M_{k,f}(b, a) := \sup\{q_k(H) : f(H) \in [b, a]\}$,

- $m_{k,f}(a) := m_{k,f}(0, a)$, $M_{k,f}(a) := M_{k,f}(0, a)$, $m_{k,f} := m_{k,f}(0, \infty)$,
 $M_{k,f} := M_{k,f}(0, \infty)$.

W powyższym zapisie będziemy pomijać f w indeksach, jeśli z kontekstu będzie wynikać o jaką funkcję chodzi.

Wartości m_k i M_k oznaczają minimalną i maksymalną wartość, o jaką zwiększy się potencjał planszy po k rundach (zakładając, że początkowo potencjał ten był skończony i gracze grają optymalnie).

Fakt 2.3 (Metoda funkcji potencjału). *Niech f będzie funkcją potencjału, $G \in \mathbb{G}_c$, $a \in \mathbb{R}_+$ oraz $f(G) \leq a$. Wówczas*

$$k \left(\left\lceil \frac{a - f(G)}{M_k(f(G), a)} \right\rceil - 1 \right) < \tilde{r}_G(f^{-1}([a, \infty])) \leq k \left\lceil \frac{a - f(G)}{m_k(f(G), a)} \right\rceil.$$

Dowód. Pokażemy najpierw ograniczenie górne dla $\tilde{r}_G(f^{-1}([a, \infty]))$. W grze $\tilde{R}_G(f^{-1}([a, \infty]))$, dla każdego n , konstruktor jest w stanie zagwarantować, że w ciągu kn rund potencjał planszy wzrośnie o co najmniej $nm_k(f(G), a)$ lub gra się zakończy. Stąd jeśli $nm_k(f(G), a) \geq a - f(G)$, to konstruktor jest w stanie wymusić koniec gry w kn rund. Stąd wynika ograniczenie górne. Ograniczenie dolne dowodzimy analogicznie. Jeśli $a - f(G) < nM_k(f(G), a)$, to malarz jest w stanie unikać końca gry przez co najmniej kn rund.

□

Jeśli w powyższym fakcie podstawimy graf pusty za G oraz $k = 1$, to otrzymamy następujący wniosek.

Wniosek 2.4. *Dla $a \in \tilde{R}_+$ oraz potencjału f*

$$\left\lceil \frac{a}{M_1(a)} \right\rceil \leq \tilde{r}(f^{-1}([a, \infty])) \leq \left\lceil \frac{a}{m_1(a)} \right\rceil.$$

Dla półpotencjału f można dowieść analogiczne nierówności. W tym celu trzeba nieco zmodyfikować zasady gry konstruktor-malarz oraz uogólnić funkcję q_k na przypadek, gdy f jest półpotencjałem.

Definicja. Grę $\underline{R}(\mathcal{H})$ definiujemy tak samo jak $\tilde{R}(\mathcal{H})$, z tą różnicą, że gra kończy się wtedy, gdy plansza jest izomorficzna z jednym z c -kolorowych grafów w $\mathcal{H}$ z dokładnością do izolowanych wierzchołków. Liczbę rund przy optymalnych strategiach będziemy oznaczali przez $\underline{r}(\mathcal{H})$.

Definicja. Dla półpotencjału f i c -kolorowego grafu G spełniającego nierówność $f(G) < \infty$, rozważmy grę, w której konstruktor stara się maksymalizować wartość $\max_{0 \leq i \leq k} f(G_i)$, a malarz stara się ją zminimalizować, gdzie G_i to plansza po i rundach w grze konstruktor-malarz (w szczególności $G_0 = G$). Niech

- $q_k(G) := \max_{0 \leq i \leq k} f(G_i) - f(G)$ przy założeniu, że obaj gracze grają optymalnie,
- $m_k(b, a), M_k(b, a), m_k(a), M_k(a), m_k, M_k$ będą zdefiniowane tak samo jak w przypadku, gdy f jest potencjałem.

Zauważmy, że jeśli f jest potencjałem, to powyższa definicja funkcji q_k (dla półpotencjału) jest równoważna definicji funkcji q_k przedstawionej wcześniej dla potencjału. Równoważność ta wynika z faktu, że $\max_{0 \leq i \leq k} f(G_i) = f(G_k)$.

Fakt 2.5. Niech f będzie półpotencjałem, $G \in \mathbb{G}_c$, $a \in \mathbb{R}_+$, $f(G) \leq a$. Wówczas

$$k \left(\left\lceil \frac{a - f(G)}{M_k(a)} \right\rceil - 1 \right) < \underline{r}_G(f^{-1}([a, \infty))) \leq k \left\lceil \frac{a - f(G)}{m_k(f(G), a)} \right\rceil.$$

W szczególności,

$$\left\lceil \frac{a}{M_1(a)} \right\rceil \leq \underline{r}(f^{-1}([a, \infty))) \leq \left\lceil \frac{a}{m_1(a)} \right\rceil.$$

Dowód. Konstruktor jest w stanie zagwarantować, że po co najwyżej k rundach półpotencjał wzrośnie o co najmniej $m_k(f(G), a)$. Gdy dojdziemy do tego momentu, to znów konstruktor jest w stanie zagwarantować, że po co najwyżej k kolejnych rundach półpotencjał wzrośnie o co najmniej $m_k(f(G), a)$. Zatem po $\left\lceil \frac{a-f(G)}{m_k(f(G), a)} \right\rceil$ takich operacjach (z których każda liczy co najwyżej k rund) potencjał planszy wzrośnie o co najmniej $a - f(G)$.

Z drugiej strony malarz jest w stanie zagwarantować, że dopóki gra się nie skończy, to półpotencjał w ciągu pierwszych k rund nie przekroczy $f(G) + M_k(a)$, w ciągu kolejnych k rund nie przekroczy $f(G) + 2M_k(a)$, więc podczas pierwszych $k \left(\left\lceil \frac{a-f(G)}{M_k(a)} \right\rceil - 1 \right)$ potencjał nie przekroczy

$$f(G) + \left(\left\lceil \frac{a-f(G)}{M_k(a)} \right\rceil - 1 \right) M_k(a) < f(G) + \frac{a-f(G)}{M_k(a)} M_k(a) = a,$$

co dowodzi dolnego oszacowania. □

Przedstawimy teraz fakt, który pozwoli powiązać ze sobą $\tilde{r}$ i $\underline{r}$.

Fakt 2.6. *Dla dowolnego zbioru c -kolorowych grafów $\mathcal{H}$ zachodzi*

$$\tilde{r}(\mathcal{H}) \leq \underline{r}(\mathcal{H}).$$

Ponadto, jeśli dla każdej pary c -kolorowych grafów G, H spełniających $G \in \mathcal{H}$ oraz $G \subset H$ zachodzi $H \in \mathcal{H}$, to $\underline{r}(\mathcal{H}) = \tilde{r}(\mathcal{H})$.

Z wniosku 2.4 wynika, że funkcja potencjału f pozwala uzyskać dokładną wartość $\tilde{r}(\mathcal{H})$, jeśli spełnia następujące własności:

- $f(G) \geq a$ wtedy i tylko wtedy, gdy G jest nadgrafem c -kolorowego grafu $H \in \mathcal{H}$,
- $m_1(a) = M_1(a)$.

Pokażemy, że ta funkcja istnieje dla każdej rodziny c -kolorowych grafów $\mathcal{H}$. To pokazuje swego rodzaju uniwersalność metody funkcji potencjału.

Fakt 2.7. *Dla każdego zbioru c -kolorowych grafów $\mathcal{H}$ istnieje funkcja potencjału $f : \mathbb{G}_c \rightarrow \mathbb{R}_+$ oraz liczba a , które spełniają poniższe dwa warunki:*

- $f(G) \geq a$ wtedy i tylko wtedy, gdy istnieje $H \in \mathcal{H}$ dla którego $H \subset G$.
- Jeśli $f(G) < a$, to $q_1(G) = 1$.

Dowód. Niech $a = \tilde{r}(\mathcal{H})$ oraz $f(G) = a - \tilde{r}_G(\mathcal{H})$. Z definicji rozgrywanej liczby Ramseya $\tilde{r}_G(\mathcal{H}) = 0$ dokładnie wtedy, gdy dla pewnego $H \in \mathcal{H}$ zachodzi $H \subset G$.

Ponadto jeśli w grze $\tilde{R}(\mathcal{H})$ obaj gracze grają optymalnie, to dla dowolnej planszy G , dla której $\tilde{r}_G(\mathcal{H}) > 0$, po jednej rundzie otrzymamy planszę G' spełniającą $\tilde{r}_{G'}(\mathcal{H}) = \tilde{r}_G(\mathcal{H}) - 1$. Zauważmy też, że optymalne strategie konstruktora i malarza w grze $\tilde{R}(\mathcal{H})$ wyznaczają optymalne ruchy konstruktora i malarza w grze $Q_1(G)$. Stąd $q_1(G) = 1$. $\square$

3 Zastosowania

Teraz pokażemy kilka przykładów zastosowania metody funkcji potencjału do pokazania kilku klasycznych oszacowań rozgrywanych liczb Ramseya (część z nich w nieco uogólnionej wersji). Zaczniemy od klasycznego wyniku dotyczącego ścieżek otrzymanego przez Grytczuka, Kiersteada i Prałata.

Twierdzenie 3.1. *[12, Thm 2.3] Niech $n \geq 3$. W $2n - 5$ rundach konstruktor może zmusić malarza do stworzenia dwóch rozłącznych jednokolorowych ścieżek (jednej czerwonej i jednej niebieskiej), które łącznie zawierają n wierzchołków. Innymi słowy,*

$$\tilde{r}(\{P_k^r \sqcup P_{n-k}^b : 1 \leq k < n\}) \leq 2n - 5.$$

Poniżej przedstawimy dwa dowody tego twierdzenia. Obydwa opierają się na metodzie funkcji potencjału. Ich główną różnicą jest fakt, że jeden z nich używa zdefiniowanego w poprzednim rozdziale parametru m_1 , a druga podobnego do niego parametru m_2 . Mówiąc bardziej obrazowo, pierwsza używa nieco bardziej skomplikowanej funkcji potencjału, ale wymagane jest myślenie 1 ruch naprzód. W drugim przypadku funkcja potencjału jest prostsza, ale za to wymaga myślenia 2 ruchy naprzód.

Dowód 1. Niech f będzie funkcją potencjału zadaną wzorem:

- $f(G) = 0$, jeśli G jest grafem pustym.
- $f(G) = \max_{H \subset G} \{e(H) + v(H) - 3 : H \text{ jest b*r*-ścieżką albo sumą rozłącznych b*-ścieżki i r*-ścieżki}\}.$

Jeśli G jest pustym grafem, to po pierwszej rundzie gry

$$\tilde{r}(\{P_k^r \sqcup P_{n-k}^b : 1 \leq k < n\})$$

mamy P_2 i trywialną P_1 , więc $q_1(G) = 1$. Jeśli $f(G) = 2k - 5$, to zawiera on rozłączne czerwoną ścieżkę P_l i niebieską ścieżkę P_{k-l} , dla pewnego l . Konstruktor może wybrać krawędź między końcami tych ścieżek i stworzyć b*r*-ścieżkę P_k , więc $q_1(G) \geq 1$. Jeśli $f(G) = 2k - 4$, to istnieje w G b*r*-ścieżka P_k . Niech v będzie wspólnym wierzchołkiem ścieżek czerwonej i niebieskiej w tej b*r*-ścieżce. Konstruktor może wybrać krawędź vx , gdzie x to wierzchołek spoza b*r*-ścieżki. Po ruchu malarza krawędzie P_k (za wyjątkiem jednej) i vx tworzą rozłączne czerwoną i niebieską ścieżki o $k + 1$ wierzchołkach, więc ponownie $q_1(G) \geq 1$. Z wniosku 2.4 wynika, że

$$\tilde{r}(f^{-1}[2n - 5, \infty]) \leq \frac{2n - 5}{1} = 2n - 5.$$

Zauważmy, że jeśli $f(G) \geq 2n - 5$, to G musi zawierać $P_k^r \sqcup P_{n-k}^b$, więc

korzystając z faktu 2.1

$$\tilde{r}(\{P_k^r \sqcup P_{n-k}^b : 1 \leq k < n\}) \leq \tilde{r}(f^{-1}[2n-5, \infty]) \leq 2n-5.$$

□

Drugi dowód jest przeniesieniem idei dowodu z [12] na język potencjału.

Dowód 2. Niech G będzie dowolnym dwukolorowym grafem, R, B będą rozłącznymi czerwoną i niebieską ścieżkami w G , których łączna liczba krawędzi jest największa, $f(G)$ niech będzie łączną liczbą krawędzi w R i B . Pokażemy, że $q_2(G) \geq 1$.

Niech u_1, u_2 będą końcami ścieżki R , a v_1, v_2 będą końcami ścieżki B . Konstruktor wybiera krawędź u_1v_1 . Bez straty ogólności możemy założyć, że malarz koloruje ją na czerwono. Wówczas konstruktor wybiera v_1x , gdzie x to nowy wierzchołek. Jeśli ta krawędź jest niebieska, to $R, B \cup x$ są rozłącznymi ścieżkami, które mają o jedną krawędź więcej. Jeśli v_1x jest czerwona, to takimi ścieżkami są $R \cup v_1 \cup x, B \setminus v_1$. Oznacza to, że $q_2(G) \geq 1$, więc

$$\tilde{r}_{P_2^b}(f^{-1}[n-2, \infty]) = \tilde{r}_{P_2^r}(f^{-1}[n-2, \infty]) \leq 2(n-2-1).$$

Ponieważ $\tilde{r}(P_2, P_2) = 1$,

$$\tilde{r}(f^{-1}[n-2, \infty]) \leq 2(n-2-1) + 1 = 2n-5.$$

□

Teraz pokażemy dolne oszacowanie dla $\tilde{r}(C_k, G)$ w zależności od liczby wierzchołków, krawędzi i składowych spójności G .

Twierdzenie 3.2. *Niech $\mathcal{C}$ oznacza zbiór cykli, a A_n będzie zbiorem wszystkich skończonych grafów G , które spełniają $e(G) + v(G) - \omega(G) \geq n$. Wtedy*

$$\tilde{r}(\mathcal{C}, A_n) \geq n.$$

Dowód. Przypomnijmy, że $\omega_r(G)$ oznacza liczbę składowych spójności czerwonego podgrafu G . Rozważmy funkcję $f : \mathbb{G}_2 \rightarrow \mathbb{R}$, która dla skończonych dwukolorowych grafów jest równa $f(G) = e(G) + \omega_r(G) - \omega(G)$, jeśli G nie zawiera czerwonego cyklu oraz $f(G) = \infty$ w przeciwnym wypadku. Łatwo sprawdzić, że f jest funkcją potencjału (wynika to z faktu, że dodanie jednej krawędzi nie może zmniejszyć ω_r o więcej niż 1). Ponadto jeśli $H \in A_n$, to $f(H^b) \geq n$. Wystarczy więc pokazać, że $M_1 \leq 1$ i skorzystać z wniosku 2.4. Rozważmy grę $\tilde{R}(\mathcal{C}, A_n)$ i dowolną planszę G . Pokażemy, że $q_1(G) \leq 1$.

Rozważmy dowolną planszę G i dowolną krawędź (ruch konstruktora) uv . Jeśli u i v nie należą do tej samej składowej spójności czerwonego podgrafu, to malarz maluje ją na czerwono. Wtedy dla $G' = G \cup (uv)^r$ zachodzi

$$\begin{aligned} f(G') &= e(G') + \omega_r(G') - \omega(G') = e(G) + 1 + \omega_r(G) - 1 - \omega(G') \\ &\leq e(G) + \omega_r(G) - \omega(G) + 1 = f(G) + 1. \end{aligned}$$

Jeśli u i v należą do tej samej składowej spójności czerwonego podgrafu, to malarz maluje ją na niebiesko. Wtedy dla $G' = G \cup (uv)^b$ mamy $\omega_r(G') = \omega_r(G)$ oraz $\omega(G') = \omega(G)$, czyli $f(G') = f(G) + 1$. Stąd wniosek, że $q_1(G) \leq 1$, co kończy dowód. $\square$

Z powyższego twierdzenia można wyciągnąć wniosek, który jest uogólnieniem lematu 3.4 w [9] na przypadek, gdzie graf G nie musi być spójny.

Wniosek 3.3. $\tilde{r}(C_k, G) \geq e(G) + v(G) - \omega(G)$.

Kolejne twierdzenie pozwoli nam znaleźć oszacowanie z dołu $\tilde{r}(G, H)$ dla dowolnych grafów G, H w zależności od maksymalnego stopnia $\Delta(G)$ oraz minimalnego pokrycia wierzchołkowego $\beta(H)$.

Twierdzenie 3.4. *Niech S_k oznacza gwiazdę z k krawędziami, a $B_{n,t}$ będzie zbiorem grafów G , dla których $\beta(G) \geq n$ oraz $e(G) \geq t$. Wtedy*

$$\tilde{r}(S_k, B_{n,t}) \geq n(k-1)/2 + t$$

Niech $f : \mathbb{G}_2 \rightarrow \infty$ będzie zdefiniowana w poniższy sposób

- $f(G) = \infty$, jeśli G zawiera czerwoną kopię S_k .
- $f(G) = e(p(G))$, gdzie $p(G)$ jest najmniejszym (w sensie liczby krawędzi) pokolorowanym grafem, dla którego $G \subset p(G)$ oraz każda niebieska krawędź w $p(G)$ ma co najmniej jeden koniec w wierzchołku, którego czerwony stopień jest większy lub równy $k - 1$.

Dowód. Łatwo sprawdzić, że f jest funkcją potencjału. Ponadto $f(S_k^r) = \infty$ i $f(G^b) \geq \beta(G)(k - 1)/2 + e(G)$ dla dowolnego grafu G . Nierówność wynika z faktu, że w $p(G)$ istnieje pokrycie wierzchołkowe G takie, że każdy wierzchołek w tym pokryciu ma czerwony stopień równy $k - 1$.

Pozostaje pokazać, że $M_1 \leq 1$. Rozważmy dowolny kolorowy graf G , dla którego $f(G) < \infty$ i krawędź uv , która nie należy do G . Jeśli $d_r(u) = k - 1$ lub $d_r(v) = k - 1$, to $f(G \cup (uv)^b) = e(p(G) \cup (uv)^b) = f(G) + 1$. W przeciwnym razie $f(G \cup (uv)^r) \leq e(p(G) \cup (uv)^r) = f(G) + 1$. Stąd $q_1(G) \leq 1$, $M_1 \leq 1$ i w konsekwencji otrzymujemy tezę twierdzenia. $\square$

Wniosek 3.5 (Stwierdzenie 4.1 w [12]). *Dla dowolnych grafów G, H zachodzi*

$$\tilde{r}(G, H) \geq \beta(H)(\Delta(G) - 1)/2 + e(H).$$

W przypadku prostych oszacowań rozgrywanej liczby Ramsey'a funkcja potencjału może nieco zaciemnić obraz. Z tego też powodu niewiele dowodów wykorzystuje tę metodę w sposób jawny (wliczając w to oryginalne dowody powyższych twierdzeń). Jeden z wyjątków jest w [14], gdzie Mond i Portier wykorzystali funkcję zbliżoną do funkcji potencjału i za jej pomocą znaleźli dolne oszacowanie wartości $\tilde{r}(P_{10}, P_n)$.

Funkcja, której użyli (nazywana przez nich funkcją potencjału) nie spełnia definicji funkcji potencjału przedstawionej w tej pracy. Główną różnicą jest fakt, że wartość tej funkcji jest zależna nie tylko od obecnej planszy, ale od historii gry. Jednakże ich funkcję da się w miarę łatwo uniezależnić od wcześniejszych stanów planszy.

Twierdzenie 3.6. Niech $\mathcal{C}$ oznacza rodzinę wszystkich cykli oraz A_x oznacza rodzinę wszystkich grafów G , które nie mają wierzchołków izolowanych i spełniają $\frac{5}{3}v(G) - l(G) \geq x$, gdzie $l(G)$ oznacza liczbę wierzchołków stopnia 1. Wtedy

$$\tilde{r}(\{P_{10}\} \cup \mathcal{C}, A_x) \geq x.$$

Dowód. Składową spójności nazywamy dużą, jeśli ma co najmniej trzy wierzchołki. Dla każdej dużej składowej spójności w czerwonym podgrafie G wybieramy wierzchołek, który będziemy nazywać korzeniem. Korzeń wybieramy w ten sposób, by odległość od niego do dowolnego innego wierzchołka w tej samej składowej spójności czerwonego grafu nie była większa niż 4. Zbiór korzeni będziemy oznaczać przez R . Definiujemy teraz zbiory $A = \{I, S, 0, 1, 2, 3, 4\}$ oraz $B = \{0, 1, 2\}$ i dwie funkcje: $a : V(G) \rightarrow A$ oraz $b : V(G) \rightarrow B$ w następujący sposób:

- $a(v) = I$, jeśli $\text{comp}_r(v) = 1$, gdzie $\text{comp}_r(v)$ oznacza wielkość składowej spójności w czerwonym grafie.
- $a(v) = S$, jeśli $\text{comp}_r(v) = 2$.
- $a(v) = \text{dist}_r(v, R)$, jeśli $\text{comp}_r(v) \geq 3$, gdzie $\text{dist}_r(v, R)$ oznacza odległość v od najbliższego wierzchołka z R w czerwonym podgrafie G .
- $b(v) = \min(d_b(v), 2)$.

Funkcja $c : A \times B \rightarrow \mathbb{R}$ przypisuje każdemu wierzchołkowi liczbę na podstawie parametrów a i b . Definiuje ją poniższa tabelka (wartości tej funkcji zostały wyznaczone za pomocą programowania liniowego).

$\begin{smallmatrix} a \\ b \end{smallmatrix}$	I	S	0, 1	2, 3	4
0	0	3	4	6	10
1	4	7	7	8	10
2	10	10	10	10	10

Teraz możemy zdefiniować funkcję f :

- $f(G, R) = \infty$, jeśli G zawiera czerwony cykl bądź ścieżkę P_{10} ,

•

$$f(G, R) = \sum_{v \in V(G)} c(a(v), b(v)),$$

•

$$f(G) = \min_R f(G, R).$$

Zauważmy najpierw, że f jest półpotencjałem (w rzeczywistości jest nawet potencjałem, ale nieco trudniej to wykazać). Ponadto jeśli G zawiera czerwony cykl, ścieżkę P_{10} bądź niebieską ścieżkę P_n , to

$$f(G) \geq 10(v(G) - l(G)) + 4l(G) = 10v(G) - 6l(G) \geq 6x.$$

Pozostaje pokazać, że $M_1 \leq 6$ i skorzystać z faktu 2.5.

W tym celu rozważmy kolorowy graf G i dowolną krawędź $uv \notin E(G)$. Pokażemy, że

$$\min(f(G \cup (uv)^r), f(G \cup (uv)^b)) \leq f(G) + 6.$$

Niech R będzie zbiorem korzeni minimalizującym $f(G, R)$. Wtedy R jest też zbiorem korzeni dla $G \cup (uv)^b$. Dodanie niebieskiej krawędzi uv zwiększa wartość $b(u)$ oraz $b(v)$ o co najwyżej 1, natomiast nie zmienia wartości $a(u)$ ani $a(v)$. Korzystając z powyższej tabelki wnioskujemy, że malarz może pomalować uv na niebiesko i nie zwiększyć wartości f o więcej niż 6 jeśli przed ruchem malarza zachodzi co najmniej jeden z poniższych 3 warunków:

- $b(u) = 2$ lub $b(v) = 2$, lub $a(u) = 4$, lub $a(v) = 4$ ($c(u)$ bądź $c(v)$ się nie zwiększa).
- $a(u) \in \{I, S\}$, $b(u) = 0$ oraz $a(v) \in \{2, 3\}$ (czyli $c(u)$ zwiększa się o 4, a $c(v)$ o 2) lub na odwrót.

- $a(u) \in \{0, 1, 2, 3, 4\} \vee (a(u), b(u)) = (S, 1)$ oraz $a(v) \in \{0, 1, 2, 3, 4\} \vee (a(v), b(v)) = (S, 1)$ ($c(u)$ i $c(v)$ zwiększają się o nie więcej niż 3).

Pozostaje pokazać, że jeśli w pozostałych przypadkach malarz pokoloruje krawędź na czerwono, to potencjał planszy wzrośnie nie więcej niż o 6.

- Jeśli $a(u) = a(v) = I$, to po dodaniu czerwonej krawędzi będziemy mieli $a(u) = a(v) = S$, co zwiększy zarówno $c(u)$, jak i $c(v)$ o nie więcej niż 3.
- Jeśli $a(u) = I, a(v) = S$, to po dodaniu czerwonej krawędzi dodajemy v do zbioru korzeni. Wskutek tego $c(u)$ wzrośnie o nie więcej niż 4, $c(v)$ o nie więcej niż 1 oraz $c(w)$ o nie więcej niż 1, gdzie w jest sąsiadem v w czerwonym podgrafie G .
- Jeśli $a(u) = a(v) = S$, to po dodaniu czerwonej krawędzi dodajemy wierzchołek v do zbioru korzeni. Niech tu, vw będą czerwonymi krawędziami w G . Wtedy $c(u), c(v), c(w)$ rosna o nie więcej niż 1, a $c(t)$ o nie więcej niż 3 (ponieważ $a(t) = 2$).
- Jeśli $a(u) = I$ oraz $a(v) \in \{0, 1\}$, to $c(u)$ zwiększy się o nie więcej niż 6. Analogicznie jest, gdy $(a(u), b(u)) = (I, 1)$ oraz $a(v) \in \{2, 3\}$.
- Jeśli $a(u) = S, a(v) \in \{0, 1\}$ i tu jest czerwoną krawędzią w G , to $c(u)$ oraz $c(t)$ wzrosną o nie więcej niż 3.

Po każdym kroku może się okazać, że przesunięcie niektórych korzeni może zmniejszyć wartość $f(G \cup uv, R)$, ale nie musimy się tym przejmować. Z dowolności G oraz uv otrzymujemy $M_1 \leq 6$, co kończy dowód. $\square$

Wniosek 3.7 (Theorem 2.1 w [14]).

$$\tilde{r}(P_{10}, P_n) \geq \frac{5}{3}n - 2.$$

W nieopublikowanej jeszcze pracy [1] udało nam się poprawić ten wynik poprzez pokazanie, że w powyższej nierówności można zastąpić P_{10} przez P_9 .

Twierdzenie 3.8.

$$\tilde{r}(P_9, P_n) \geq \frac{5}{3}n - 2.$$

Warto tutaj zauważyć, że w świetle oszacowania górnego $\tilde{r}(P_k, P_n) \leq \frac{5}{3}n + 12k$ pochodzącego z [8], oba powyższe oszacowania dolne są optymalne z dokładnością do stałej.

4 Cykle nieparzyste vs. graf spójny

Niech $\mathcal{C}_{odd}$ oznacza rodzinę cykli o nieparzystej długości, a $\mathcal{C}on_{n,m}$ – rodzinę spójnych grafów mających dokładnie n wierzchołków i m krawędzi. Przez φ oznaczać będziemy złotą liczbę, tzn. $\varphi = \frac{1+\sqrt{5}}{2}$. Naszym celem będzie teraz wykazanie następującego oszacowania.

Twierdzenie 4.1. *Dla wszystkich $m, n \in \mathbb{N}$, spełniających $n-1 \leq m \leq \binom{n}{2}$, zachodzi*

$$\tilde{r}(\mathcal{C}_{odd}, \mathcal{C}on_{n,m}) \geq \varphi n + m - 2\varphi + 1.$$

Wyniki z tego oraz z trzech kolejnych rozdziałów pochodzą z pracy [2].

Wprost z tego twierdzenia otrzymujemy, że $\tilde{r}(C_{2k+1}, T_n) \geq \varphi^2 n - 2\varphi \approx 2,618n - 3,236$ dla $k \geq 1$ oraz dowolnego drzewa T_n na n wierzchołkach. Najlepsze dotychczasowe oszacowanie zostało osiągnięte przez Cyman, Dzido, Lapinska i Lo [9], z których wyników można wywnioskować, że

$$\tilde{r}(\mathcal{C}, \mathcal{C}on_{n,m}) \geq n + m - 1. \quad (1)$$

Ponadto Dybizbański, Dzido i Zakrzewska [10] pokazali, że $\tilde{r}(C_3, P_n) \geq 2n-1$, co w szczególnym przypadku poprawia wynik z [9] o jeden ruch.

4.1 Gra $\mathcal{G}(n, m)$

Dla grafu G oraz podzbioru jego wierzchołków $V' \subseteq V(G)$, niech $E[V']$ oznacza zbiór krawędzi o obu końcach w V' .

Niech G będzie pokolorowanym grafem. Mówimy, że G jest czerwonodwu-dzielny, jeśli istnieje taki podział $V(G) = V_1 \cup V_2$ (jeden z tych zbiorów może być pusty), że nie ma niebieskich krawędzi w G między V_1 a V_2 oraz nie ma czerwonych krawędzi w $E[V_1] \cup E[V_2]$. Parę (V_1, V_2) nazywać będziemy czerwonym podziałem grafu G . Nie jest trudno zauważyć, że pokolorowany graf G jest czerwonodwu-dzielny wtedy i tylko wtedy, gdy G nie zawiera cyklu z nieparzystą liczbą czerwonych krawędzi. Ponadto, każda składowa spójności

pokolorowanego grafu ma co najwyżej jeden czerwony podział, z dokładnością do kolejności w parze (V_1, V_2) .

Niech $n, m \in \mathbb{N}$. Rozważmy następującą grę pomocniczą $\mathcal{G}(n, m)$. W każdej rundzie konstruktor wybiera wcześniej niepokolorowaną krawędź z $K_{\mathbb{N}}$, a malarz koloruje ją na czerwono lub niebiesko. Gra kończy się, jeśli po ruchu malarza powstanie pokolorowany cykl o nieparzystej liczbie czerwonych krawędzi lub pojawi się taki spójny pokolorowany graf H z czerwonym podziałem (V_1, V_2) , że dla pewnego $i \in 1, 2$ mamy $|V_i| \geq n$ oraz $|E[V_i]| \geq m$. Celem konstruktora jest zakończenie gry tak szybko, jak to możliwe, podczas gdy celem malarza jest przeciągnięcie gry. Niech $\tilde{r}_{\mathcal{G}(n, m)}$ oznacza liczbę rund w grze $\mathcal{G}(n, m)$, w której obaj gracze grają optymalnie.

Jeśli na planszy pojawi się czerwony nieparzysty cykl lub spójny niebieski graf z n wierzchołkami i m krawędziami, to gra $\mathcal{G}(n, m)$ kończy się, więc

$$\tilde{r}_{\mathcal{G}(n, m)} \leq \tilde{r}(\mathcal{C}_{\text{odd}}, \mathcal{C}on_{n, m}).$$

Zatem, aby udowodnić twierdzenie 4.1, wystarczy wykazać, że

$$\tilde{r}_{\mathcal{G}(n, m)} \geq \varphi n + m - 2\varphi + 1.$$

Niech $f : \mathbb{G}_2 \rightarrow \mathbb{R}_+ \cup \{\infty\}$ będzie półpotencjałem zdefiniowanym następująco:

- Jeśli G nie jest czerwonodwudzielny, to $f(G) = \infty$.
- Jeśli G jest izolowanym wierzchołkiem, to $f(G) = 0$.
- Jeśli G jest niepustym grafem spójnym z czerwonym podziałem (V_1, V_2)

i $|V(G)| > 1$, to

$$\begin{aligned} p_G(V_i) &= |V_i|[\varphi + |E[V_i]|], \text{ dla } i = 1, 2, \\ a(G) &= \max(p_G(V_1), p_G(V_2)), \\ b(G) &= \min(p_G(V_1), p_G(V_2)), \\ f(G) &= \varphi a(G) - \varphi + \max(a(G) - \varphi^3, b(G)). \end{aligned}$$

- Jeśli składowymi spójności G są $G_1, G_2, \dots, G_t$, to przyjmujemy

$$f(G) = \sum_{i=1}^t f(G_i).$$

Motywacja stojąca za wyborem półpotencjału f jest opisana w rozdziale 7.

Wykorzystamy również następującą funkcję $g : \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$g(x, y) = \varphi \max(x, y) - \varphi + \max(\max(x, y) - \varphi^3, \min(x, y)).$$

Zauważmy, że g jest symetryczna, niemalejąca względem x (i y), a jeśli (V_1, V_2) jest czerwonym podziałem spójnego, pokolorowanego grafu H , to $g(p_H(V_1), p_H(V_2)) = f(H)$. Ponieważ

$$\max(a, b) = c + \max(a - c, b - c),$$

$$\max(\max(x, y) - \varphi^3, \min(x, y)) = \min(x, y) + \max(\max(x, y) - \min(x, y) - \varphi^3, 0),$$

$$\max(\max(x, y) - \min(x, y) - \varphi^3, 0) = \max(x - y - \varphi^3, y - x - \varphi^3, 0),$$

$$\max(x, y) + \min(x, y) = x + y,$$

więc możemy również przedstawić g jako

$$g(x, y) = x + y - \varphi + (\varphi - 1) \max(x, y) + \max(x - y - \varphi^3, y - x - \varphi^3, 0),$$

a z tego wynika, że g jest funkcją wypukłą.

Pokażemy teraz, że dla funkcji f zachodzi $M_1 \leq \varphi + 1$. Innymi słowy, w grze $\mathcal{G}(n, m)$ malarz może grać w taki sposób, aby w żadnej rundzie wartość funkcji f nie zwiększyła się o więcej niż $\varphi + 1$.

Lemat 4.2. *Niech G będzie pokolorowanym grafem (planszą gry), który jest czerwonodzielny (tzn. $f(G) < \infty$) oraz e będzie dowolną krawędzią, która nie należy do G (ruchem konstruktora). Wtedy*

$$\min(f(G \cup e^r), f(G \cup e^b)) \leq f(G) + \varphi + 1.$$

Dowód. Niech $e = uu'$ oraz H, H' będą składowymi G dla których $u \in H$, $u' \in H'$ oraz $(V_1, V_2), (V'_1, V'_2)$ są czerwonymi podziałami grafów odpowiednio H i H' , gdzie $u \in V_1$, $u' \in V'_1$. Zauważmy, że $\varphi^2 = \varphi + 1$ i w konsekwencji $\varphi^3 = 2\varphi + 1$. Będziemy wykorzystywać te fakty w trakcie dowodu.

Rozważymy kilka przypadków w zależności od tego, gdzie znajdują się u oraz u' .

1. u, u' znajdują się w tej samej składowej grafu G , tzn. $H = H'$.

Jeśli $V_1 = V'_1$, to malarz pokoloruje e na czerwono. Wtedy $f(H \cup e^r) = f(H)$ oraz $f(G \cup e^r) = f(G)$.

Jeśli $V_1 = V'_2$, to malarz pokoloruje e na niebiesko. Wtedy uzyskujemy składową $F = H \cup e^b$, gdzie $p_F(V_1) = p_H(V_1) + 1$ i $p_F(V_2) = p_H(V_2)$. W związku z tym $f(F) \leq f(H) + \varphi + 1$ i $f(G \cup e^b) \leq f(G) + \varphi + 1$.

2. u, u' są izolowanymi wierzchołkami w G .

Wtedy malarz pokoloruje e na czerwono. Otrzymujemy

$$f(G \cup e^r) = f(G) + \varphi\varphi - \varphi + \varphi = f(G) + \varphi^2 = f(G) + \varphi + 1.$$

3. u' jest izolowanym wierzchołkiem w G , ale u nie jest.

- a. Załóżmy, że $p_H(V_2) > p_H(V_1) + \varphi + 1$. Wtedy malarz pokoloruje e na niebiesko. Dla $F = H \cup e^b$ mamy

$$\begin{aligned} a(F) &= \max(p_F(V_1 \cup \{u'\}), p_F(V_2)) \\ &= \max(p_H(V_1) + \varphi + 1, p_H(V_2)) = p_H(V_2) = a(H) \end{aligned}$$

oraz

$$b(F) = p_F(V_1 \cup \{u'\}) = b(H) + \varphi + 1.$$

W związku z tym $f(G \cup e^b) \leq f(G) + \varphi + 1$.

- b. Załóżmy, że $p_H(V_2) \leq p_H(V_1) + \varphi + 1$. Wtedy malarz pokoloruje e na czerwono. Tak więc u' zostaje dodany do V_2 w czerwonym podziale grafu $F = H \cup e^r$ oraz

$$p_F(V_2 + u') = p_H(V_2) + \varphi.$$

Aby oszacować $f(F)$, obliczmy $\frac{dg}{dy}(x_1, \cdot)$ dla ustalonego $x_1 \in \mathbb{R}$. Z definicji g mamy

$$\frac{dg}{dy}(x_1, y) = \begin{cases} 0 & \text{dla } y < x_1 - \varphi^3, \\ 1 & \text{dla } x_1 - \varphi^3 < y < x_1, \\ \varphi & \text{dla } x_1 < y < x_1 + \varphi^3, \\ \varphi^2 & \text{dla } y > x_1 + \varphi^3. \end{cases}$$

Zatem dla każdego $k \in (0, \varphi)$, biorąc pod uwagę fakt, że

$$p_H(V_2) + k \leq p_H(V_1) + \varphi + 1 + k < p_H(V_1) + \varphi^3,$$

mamy

$$\frac{dg}{dy}(p_H(V_1), p_H(V_2) + k) \leq \varphi.$$

Stąd

$$\begin{aligned}
f(F) - f(H) &= g(p_F(V_1), p_F(V_2 \cup \{u'\})) - g(p_H(V_1), p_H(V_2)) \\
&= g(p_H(V_1), p_H(V_2) + \varphi) - g(p_H(V_1), p_H(V_2)) \\
&\leq \varphi^2 = \varphi + 1.
\end{aligned}$$

4. u i u' nie są izolowanymi wierzchołkami w G i $H \neq H'$.

Niech

$$\begin{aligned}
a &= \max(p_H(V_1), p_H(V_2)), \\
b &= \min(p_H(V_1), p_H(V_2)), \\
c &= a - b, \\
a' &= \max(p_{H'}(V'_1), p_{H'}(V'_2)), \\
b' &= \min(p_{H'}(V'_1), p_{H'}(V'_2)), \\
c' &= a' - b'.
\end{aligned}$$

a. Załóżmy, że $p_H(V_1) > p_H(V_2) + 1$ i $p_{H'}(V'_1) + 1 < p_{H'}(V'_2)$. Wtedy malarz pokoloruje e na niebiesko. Tak więc składowe H, H' grafu G zostaną połączone w składową F grafu $G \cup e^b$, z czerwonym podziałem $(V_1 \cup V'_1, V_2 \cup V'_2)$ grafu F . Zauważmy, że $p_H(V_1) = a$, $p_H(V_2) = b$, $p_{H'}(V'_1) = b'$, $p_{H'}(V'_2) = a'$ oraz $c, c' > 1$. Ponadto

$$\begin{aligned}
p_F(V_1 \cup V'_1) &= p_H(V_1) + p_{H'}(V'_1) + 1 = a + b' + 1, \\
p_F(V_2 \cup V'_2) &= p_H(V_2) + p_{H'}(V'_2) = a' + b.
\end{aligned}$$

Założmy nie wprost, że $f(G \cup e^b) > f(G) + \varphi + 1$. Oznacza to, że

$$f(H) + f(H') + \varphi + 1 < f(F),$$

co równoważnie daje

$$g(a, b) + g(a', b') + \varphi + 1 < g(a + b' + 1, a' + b).$$

Z powyższej nierówności oraz z równości $g(x + z, y + z) = g(x, y) + (\varphi + 1)z$ wynika, że

$$g(c, 0) + g(c', 0) + \varphi^2 < g(c + 1, c').$$

Jeśli $c < c'$, to możemy zamienić miejscami c i c' i lewa strona nierówności nie zmieni się, podczas gdy prawa strona stanie się większa (ze względu na to, że g jest funkcją wypukłą i symetryczną). Oznacza to, że możemy bez straty ogólności założyć, iż $c \geq c'$. Rozpisując powyższą nierówność dostajemy

$$\begin{aligned} c\varphi + \max(c - \varphi^3, 0) + c'\varphi + \max(c' - \varphi^3, 0) - 2\varphi + \varphi^2 \\ < (c + 1)\varphi - \varphi + \max(c + 1 - \varphi^3, c'), \end{aligned}$$

co po uproszczeniu daje

$$\max(c - \varphi^3, 0) + (c' - 1)\varphi + \max(c' - \varphi^3, 0) < \max(c - \varphi^3, c' - 1).$$

Z drugiej strony $c' > 1$, więc wszystkie trzy składniki po lewej stronie są nieujemne, a ponadto $\max(c - \varphi^3, 0) \geq c - \varphi^3$ oraz $(c' - 1)\varphi > c' - 1$, więc dochodzimy do sprzeczności. Tym samym udowodniliśmy, że $f(G \cup e^b) \leq f(G) + \varphi + 1$.

- b. Załóżmy, że $p_H(V_1) > p_H(V_2)$ i $p_{H'}(V'_1) < p_{H'}(V'_2)$, ale przypadek 4a nie zachodzi, tzn. $p_H(V_1) \leq p_H(V_2) + 1$ lub $p_{H'}(V'_1) + 1 \geq p_{H'}(V'_2)$. Wtedy malarz pokoloruje e na czerwono. Podobnie jak w poprzednim przypadku, składowe H, H' grafu G zostaną połączone w składową F grafu $G \cup e^r$, ale czerwony podział grafu F

to $(V_1 \cup V'_2, V_2 \cup V'_1)$ oraz

$$p_F(V_1 \cup V'_2) = a + a', \quad p_F(V_2 \cup V'_1) = b + b'.$$

Ponownie załóżmy nie wprost, że $f(G \cup e^r) > f(G) + \varphi + 1$, co jest równoważne $f(H) + f(H') + \varphi + 1 < f(F)$ i stąd

$$g(a, b) + g(a', b') + \varphi^2 < g(a + a', b + b').$$

Wtedy otrzymujemy

$$g(c, 0) + g(c', 0) + \varphi + 1 < g(c + c', 0)$$

i w świetle definicji funkcji g

$$\max(c - \varphi^3, 0) + \max(c' - \varphi^3, 0) + 1 < \max(c + c' - \varphi^3, 0).$$

Z założeń tego przypadku mamy $c \leq 1$ lub $c' \leq 1$, więc powyższa nierówność nie może zachodzić.

Ze względu na symetryczną rolę H i H' , przypadki 4a i 4b obejmują wszystkie sytuacje, w których $(p_H(V_1) - p_H(V_2))(p_{H'}(V'_1) - p_{H'}(V'_2)) < 0$. Pozostaje jeszcze przeanalizować przypadek, gdy nierówność jest w drugiej stronie.

- c. Załóżmy, że $(p(V_1) - p(V_2))(p(V'_1) - p(V'_2)) \geq 0$. Wtedy malarz pokoloruje e na czerwono. Podobnie jak w poprzednim przypadku, składowe H, H' grafu G zostaną połączone w składową F grafu $G \cup e^r$ z czerwonym podziałem $(V_1 \cup V'_2, V_2 \cup V'_1)$, ale mamy albo

$$p_F(V_1 \cup V'_2) = a + b' \text{ oraz } p_F(V_2 \cup V'_1) = b + a',$$

albo

$$p_F(V_1 \cup V_2') = b + a' \text{ oraz } p_F(V_2 \cup V_1') = a + b'.$$

W obu przypadkach, z symetrii funkcji g , założenie, że $f(H) + f(H') + \varphi + 1 < f(F)$ prowadzi do

$$g(a, b) + g(a', b') + \varphi + 1 < g(a + b', b + a').$$

Wtedy otrzymujemy

$$g(c, 0) + g(c', 0) + \varphi + 1 < g(c, c').$$

Nierówność jest symetryczna względem c i c' , więc możemy założyć, że $c \geq c'$. Wówczas powyższa nierówność jest równoważna

$$\max(c - \varphi^3, 0) + \max(c' - \varphi^3, 0) + c'\varphi + 1 < \max(c - \varphi^3, c').$$

Implikuje to, że $\max(c - \varphi^3, c'\varphi) < \max(c - \varphi^3, c')$ i ponownie dochodzimy do sprzeczności.

□

4.2 Dowód twierdzenia 4.1

Udowodniliśmy, że malarz ma taką strategię w grze $\mathcal{G}(n, m)$, że po każdej rundzie gry plansza jest czerwonodwuździelna, a w każdej rundzie jego półpotencjał f wzrasta o nie więcej niż $\varphi + 1$ (na początku gry półpotencjał wynosi 0).

Założmy, że dla opisanej strategii malarza i dowolnej strategii konstruktora po t rundach gry plansza zawiera taką składową H z czerwonym podziałem (V_1, V_2) , że $|V_1| \geq n$ i $|E[V_1]| \geq m$. Z jednej strony mamy $f(G) \leq t(\varphi + 1)$.

Z drugiej strony mamy

$$\begin{aligned} f(G) &\geq f(H) = \varphi a(H) - \varphi + \max(a(H) - \varphi^3, b(H)) \\ &\geq (\varphi + 1)a(H) - \varphi - \varphi^3 \geq (\varphi + 1)(\varphi n + m) - \varphi - \varphi^3. \end{aligned}$$

W związku z tym

$$t \geq \frac{(\varphi + 1)(\varphi n + m) - \varphi - \varphi^3}{\varphi + 1} = \varphi n + m - 2\varphi + 1.$$

Wnioskujemy, że malarz może przetrwać co najmniej $\varphi n + m - 2\varphi + 1$ rund w grze $\mathcal{G}(n, m)$. W świetle poprzednich uwag dowodzi to również, że $\tilde{r}(\mathcal{C}_{\text{odd}}, \mathcal{C}on_{n,m}) \geq \varphi n + m - 2\varphi + 1$.

5 Trójkąt vs. ścieżka – oszacowanie górne

Naszym kolejnym celem będzie pokazanie oszacowania z drugiej strony.

Twierdzenie 5.1. *Dla $n \geq 3$*

$$\tilde{r}(C_3, P_n) \leq 3n - 4.$$

Powyższe twierdzenie poprawia najlepsze do tej pory oszacowanie górne $\tilde{r}(C_3, P_n) \leq 4n - 5$, udowodnione przez Dybizbańskiego, Dzido i Zakrzewską w [10].

Oczywiście $\tilde{r}(\mathcal{C}_{\text{odd}}, \mathcal{C}on_{n,n-1}) \leq \tilde{r}(C_3, P_n)$, więc twierdzenia 4.1 i 5.1 dają następujące granice dla $\tilde{r}(C_3, P_n)$.

Wniosek 5.2. *Jeśli $n \geq 3$, to*

$$\lceil (\varphi + 1)n - 2\varphi \rceil \leq \tilde{r}(C_3, P_n) \leq 3n - 4.$$

Granice górna i dolna w powyższym wniosku są optymalne dla $n = 3, 4$

ponieważ $\tilde{r}(C_3, P_3) = 5$ oraz $\tilde{r}(C_3, P_4) = 8$, jak zostało zweryfikowane (za pomocą komputera) przez Gordinowicza i Prałata [11].

Jeśli $n = 3$, to wiemy, że $\tilde{r}(C_3, P_n) = 5 = 3n - 4$. Można również sprawdzić, że konstruktor może wymusić czerwony C_3 lub niebieską P_3 , grając tylko na K_5 – będziemy używać tego faktu później.

Od tego momentu w tym rozdziale będziemy zakładać, że $n \geq 4$. Rozważając moment gry $\tilde{R}(C_3, P_n)$, mówimy, że konstruktor łączy dwie pokolorowane ścieżki P i P' , jeśli w rozważanej rundzie wybiera krawędź łączącą jeden z końców P z jednym końcem P' . Mówimy, że konstruktor może wymusić niebieski graf H w ciągu t rund, jeśli konstruktor ma strategię, która gwarantuje, że po co najwyżej t następnych rundach zostanie utworzona czerwona kopia C_3 lub niebieska kopia H . Innymi słowy, zakładamy, że malarz unika ruchów, które natychmiast prowadzą do przegranej (jeśli to możliwe).

Niech $P(s, t)$ będzie pokolorowaną ścieżką na $s + t$ wierzchołkach, uzyskaną z dwóch rozłącznych niebieskich ścieżek P_s i P_t , przez połączenie ich końców czerwoną krawędzią. Ponadto, niech $P(s, 0)$ oznacza niebieską ścieżkę na s wierzchołkach.

Wprowadzimy też oznaczenie dotyczące pokolorowanych ścieżek bazujące na wyrażeniach regularnych. Maksymalna (w sensie inkluzji) pokolorowana ścieżka nazywana będzie b^*rb^* -ścieżką, jeśli jest izomorficzna z $P(s, t)$ dla pewnych $s, t > 0$. Z kolei pokolorowaną ścieżkę, która jest maksymalna w sensie inkluzji oraz nie zawiera czerwonych krawędzi, będziemy nazywać b^* -ścieżką.

Niech $u(P(s, t)) = |s - t|$ dla każdego $s, t \geq 0$. Zatem u jest miarą „nierównowagi” ścieżki. Mówimy, że $P(s, t)$ jest zrównoważona, jeśli $u(P(s, t)) = 0$, a w przeciwnym razie jest niezrównoważona.

Zaczynamy od następującej obserwacji.

Lemat 5.3. *Założmy, że $a_i \geq b_i$ dla $i = 0, 1, \dots, k$ oraz że plansza zawiera rozłączne wierzchołkowo ścieżki $P(a_0, b_0), P(a_1, b_1), \dots, P(a_k, b_k)$. Wtedy konstruktor może wymusić albo czerwony C_3 , albo niebieską ścieżkę na*

$a_0 + \sum_{i=1}^k b_k$ wierzchołkach w ciągu co najwyżej $2k$ rund.

Dowód. Udowodnimy to przez indukcję. Przypadek $k = 0$ jest trywialny. Załóżmy, że $k > 0$ i malarz nie robi czerwonego cyklu C_3 . Konstruktor w dwóch kolejnych rundach łączy koniec niebieskiej ścieżki P_{a_0} z końcami czerwonej krawędzi w $P(a_k, b_k)$ i wymusza niebieską ścieżkę o długości co najmniej $a_0 + b_k$. Teraz mamy rozłączne ścieżki wierzchołkowe $P(a_0 + b_k, 0), P(a_1, b_1), \dots, P(a_{k-1}, b_{k-1})$, więc na mocy założenia indukcyjnego konstruktor może wymusić niebieską ścieżkę o długości $a_0 + b_k + \sum_{i=1}^{k-1} b_i$ w ciągu następnych $2k - 2$ ruchów. $\square$

Rozważmy następującą strategię dla konstruktora w grze $\tilde{R}(C_3, P_n)$. Konstruktor wybiera $2n - 1$ wierzchołków, pozostałe będzie ignorował przez całą grę. Dzielimy grę na dwa etapy. Mówiąc ogólnie, w etapie 1 konstruktor gra w taki sposób, aby utworzyć wiele b^* -ścieżek i jedna z nich była bardziej nie zrównoważona niż wszystkie pozostałe razem wzięte. W etapie 2 konstruktor stosuje swoją strategię z lematu 5.3 i wymusza długą niebieską ścieżkę.

Mówiąc inaczej, na początku gry mamy dokładnie $2n - 1$ (trywialnych) b^* -ścieżek. W każdej rundzie etapu 1 konstruktor łączy krawędzią dwie najkrótsze b^* -ścieżki. Malarz pokoloruje wybraną krawędź na czerwono lub niebiesko. Konstruktor kontynuuje tak długo, jak długo na planszy znajdują się co najmniej dwie b^* -ścieżki, z jednym wyjątkiem: jeśli malarz pokoloruje pierwsze $n - 3$ krawędzie na czerwono, to etap 1 się kończy.

Jeśli ma miejsce powyższy wyjątek, to pokolorowany graf ma $n - 3$ izolowanych czerwonych krawędzi oraz pięć izolowanych wierzchołków $u_1, \dots, u_5$. Gra przechodzi do etapu 2. Konstruktor wymusza niebieską ścieżkę P_3 w ciągu następnych 5 rund, używając tylko krawędzi grafu K_5 na zbiorze wierzchołków $u_1, \dots, u_5$. Po tym pokolorowany graf zawiera ścieżkę $P(3, 0)$ oraz $n - 3$ kopii $P(1, 1)$, więc zgodnie z lematem 5.3 konstruktor może wymusić niebieską ścieżkę na $3 + n - 3 = n$ wierzchołkach lub czerwony cykl C_3 w ciągu następnych $2(n - 3)$ rund. W ten sposób gra kończy się po co najwyżej $n - 3 + 5 + 2(n - 3) = 3n - 4$ rundach.

Założmy więc, że w etapie 1 malarz pokoloruje na niebiesko przynajmniej jedną z krawędzi wybranych w pierwszych $n - 3$ rundach. Etap 1 kończy się, gdy na planszy znajduje się co najwyżej jedna b^* -ścieżka. Zauważmy, że na końcu etapu 1 graf wywołany przez wszystkie pokolorowane krawędzie składa się z rozłącznych ścieżek b^*rb^* -ścieżek oraz co najwyżej jednej b^* -ścieżki.

Ponieważ konstruktor zawsze łączy dwie najkrótsze b^* -ścieżki, wynika stąd, że następująca własność jest spełniona.

Fakt 5.4. *Po każdej rundzie etapu 1, jeśli P_k, P_l są b^* -ścieżkami, to $k \leq 2l$.*

Sprawdźmy teraz, że także następująca własność jest prawdziwa.

Fakt 5.5. *Po każdej rundzie etapu 1 istnieje co najwyżej jedna b^* -ścieżka, w której liczba wierzchołków nie jest potęgą liczby 2.*

Dowód. Przypuśćmy, że ta właściwość jest spełniona po pewnej rundzie t , w następnej rundzie konstruktor łączy b^* -ścieżki P_a i P_b , a po rundzie $t + 1$ liczba b^* -ścieżek, w których liczba wierzchołków nie jest potęgą liczby 2, wzrasta. Może to się zdarzyć tylko wtedy, gdy krawędź w rundzie $t + 1$ została pokolorowana na niebiesko, $a = 2^k$ oraz $b = 2^l$, gdzie $k \neq l$. Z faktu 5.4 wiemy, że jest to możliwe tylko wtedy, gdy $k + 1 = l$, P_{2^k} była jedyną najkrótszą b^* -ścieżką, a P_{2^l} była zarówno drugą najkrótszą, jak i najdłuższą b^* -ścieżką po rundzie t . Oznacza to, że po rundzie $t + 1$ istnieje b^* -ścieżka P_{a+b} , a wszystkie pozostałe b^* -ścieżki mają 2^l wierzchołków. $\square$

Następny lemat daje pewien wgląd we właściwości niezerównoważenia zbioru pokolorowanych ścieżek w etapie 1.

Lemat 5.6. *Niech $H_1, H_2, \dots, H_q$ będzie ciągiem wszystkich niezerównoważonych b^*rb^* -ścieżek, które istnieją po r rundach gry $\tilde{R}(C_3, P_n)$ i założmy, że pojawiły się one w tej kolejności (tzn. H_1 została utworzona jako pierwsza w grze, H_2 – jako druga i tak dalej). Wtedy $u(H_{k+1}) \geq 2u(H_k)$ dla $k = 1, 2, \dots, q - 1$. Ponadto dla każdej b^* -ścieżki H mamy $u(H) \geq 2u(H_q)$.*

Dowód. Dowodzimy tezę indukcyjnie po q . Załóżmy, że twierdzenie jest prawdziwe dla ciągu $H_1, H_2, \dots, H_{q-1}$, i rozważmy rundę r , po której pojawiła się ścieżka $H_{q-1} = P(s, t)$. Z faktów 5.4 i 5.5 wiemy, że $s, t \in [2^k, 2^{k+1}]$ dla pewnego całkowitego k , więc $u(H_{q-1}) \leq 2^k$. Na podstawie tych dwóch faktów oraz minimalności niebieskich ścieżek generujących H_{q-1} , wnioskujemy, że na końcu rundy r wszystkie b*-ścieżki mają po 2^{k+1} wierzchołków. W związku z tym, 2^{k+1} jest dzielnikiem $u(H)$ dla każdej b*-ścieżki H utworzonej po rundzie r i do końca etapu 1.

Ponieważ H_q jest tworzona po rundzie r oraz $u(H_q) > 0$, to $u(H_q) \geq 2^{k+1}$. W związku z tym dochodzimy do wniosku, że $2u(H_{q-1}) \leq u(H_q)$. Dowód nierówności $2u(H_q) \leq u(H)$ dla każdej b*-ścieżki jest analogiczny do tego dla H_{q-1} . $\square$

Rozważmy teraz sytuację na końcu etapu 1. Przypomnijmy, że w tym momencie mamy na planszy przynajmniej jedną niebieską krawędź w grafie K_{2n-1} , co najwyżej jedną b*-ścieżkę, oraz zbiór b*rb*-ścieżek (być może pusty). Jeśli istnieje b*-ścieżka, to niech H_0 będzie tą ścieżką. W przeciwnym razie, niech H_0 będzie ostatnią niezbalansowaną b*rb*-ścieżką, która pojawiła się w etapie 1. Niech $H_1, H_2, \dots, H_m$ będzie ciągiem wszystkich niezbalansowanych b*rb*-ścieżek, które pojawiły się (w tej kolejności) w etapie 1, z wyjątkiem ścieżki H_0 . Z lematu 5.6 wynika, że

$$\sum_{j=1}^m u(H_j) \leq \sum_{j=0}^{m-1} \frac{u(H_m)}{2^j} \leq 2u(H_m) \leq u(H_0). \quad (2)$$

Niech $H'_1, H'_2, \dots, H'_l$ będzie rodziną wszystkich b*rb*-ścieżek, które są zbalansowane na końcu etapu 1.

Aby obliczyć liczbę rund w etapie 1, zauważmy, że podgraf G grafu K_{2n-1} , dla którego $V(G) = V(K_{2n-1})$, a zbiór jego krawędzi składa się ze wszystkich krawędzi pokolorowanych w etapie 1, jest zbiorem $m + l + 1$ rozłącznych ścieżek. Dlatego etap 1 trwa $|E(G)| = |V(G)| - m - l - 1 = 2n - 2 - m - l$ rund.

Zauważmy również, że G nie ma izolowanych wierzchołków. Istotnie, gdyby na końcu etapu 1 istniała taka trywialna niebieska ścieżka, to w każdej poprzedniej rundzie konstruktor połączyłby dwie trywialne niebieskie ścieżki, a malarz pokolorowałby wybrane krawędzie na czerwono. To zaprzeczałoby założeniu, że w etapie 1 istnieje niebieska krawędź. Zatem G nie ma izolowanych wierzchołków i w związku z tym ma co najwyżej $n - 1$ składowych spójności. Dochodzimy do wniosku, że

$$m + l + 1 \leq n - 1. \quad (3)$$

Po etapie 1 gra przechodzi do etapu 2.

Zauważmy, że dla każdej ścieżki $P = P(s, t)$ (zbalansowanej lub niezbalansowanej), dla której $s \geq t$, mamy $s = (|V(P)| + u(P))/2$ i $t = (|V(P)| - u(P))/2$. Przypomnijmy również, że dla konstruktora plansza gry to K_{2n-1} , więc

$$\sum_{j=0}^m |V(H_j)| + \sum_{j=1}^l |V(H'_j)| = 2n - 1.$$

W etapie 2 konstruktor stosuje swoją strategię z lematu 5.3 do ścieżek

$$H_0, H_1, \dots, H_m, H'_1, H'_2, \dots, H'_l.$$

W ten sposób wymusza niebieską ścieżkę na t wierzchołkach, gdzie

$$\begin{aligned} t &= \frac{1}{2} \left(|V(H_0)| + u(H_0) + \sum_{j=1}^m (|V(H_j)| - u(H_j)) + \sum_{j=1}^l (|V(H'_j)| - u(H'_j)) \right) \\ &= \frac{1}{2} \left(2n - 1 + u(H_0) - \sum_{j=1}^m u(H_j) - \sum_{j=1}^l u(H'_j) \right) \\ &= n - \frac{1}{2} + \frac{1}{2} \left(u(H_0) - \sum_{j=1}^m u(H_j) \right) \geq n - \frac{1}{2}. \end{aligned}$$

Ostatnia nierówność wynika z (2).

Zatem w etapie 2 malarz wymusza albo czerwony cykl C_3 , albo niebieską

ścieżkę P_n , a etap 2 trwa, zgodnie z lematem 5.3, maksymalnie $2(m+l)$ rund.

Podsumowując, liczba rund w etapie 1 i etapie 2 jest nie większa niż

$$2n - 2 - m - l + 2(m + l) = 2n - 2 + m + l.$$

Z uwagi na (3), gra $\tilde{R}(C_3, P_n)$ trwa maksymalnie $2n - 2 + n - 2 = 3n - 4$ rund.

6 Trójkąt vs. graf spójny – oszacowanie górne

Zgodnie z twierdzeniem 5.1, mamy $\tilde{r}(C_3, \mathcal{C}on_{n,n-1}) \leq 3n - 4$. Teraz pokażemy uogólnienie powyższej nierówności na przypadek $\tilde{r}(C_3, \mathcal{C}on_{n,m})$.

Twierdzenie 6.1. *Dla $n \geq 3$ oraz $n - 1 \leq m \leq (n - 1)^2/4$*

$$\tilde{r}(C_3, \mathcal{C}on_{n,m}) \leq n + 2m + O(\sqrt{m - n + 1}).$$

Dowód. Z powyższej obserwacji wynika dowód dla $m = n - 1$, więc założmy, że $m \geq n$. Niech k będzie najmniejszą taką liczbą całkowitą, że $1 \leq k \leq n$, $m \leq n - k + \binom{\lfloor k/2 \rfloor}{2} + \binom{\lceil k/2 \rceil}{2}$. Ponieważ $\binom{\lfloor n/2 \rfloor}{2} + \binom{\lceil n/2 \rceil}{2} = \lfloor (n-1)^2/4 \rfloor$, takie k na pewno istnieje. Na podstawie twierdzenia 5.1 konstruktor może rozpocząć grę $\tilde{R}(C_3, \mathcal{C}on_{n,m})$, wymuszając albo czerwony trójkąt, albo niebieską ścieżkę P_n , a zajmuje mu to maksymalnie $3n - 4$ rund.

Jeśli konstruktor wymusił czerwony trójkąt, gra $\tilde{R}(C_3, \mathcal{C}on_{n,m})$ kończy się, więc założmy, że na planszy pojawiła się niebieska ścieżka o n wierzchołkach, którą oznaczymy przez B .

Niech B' będzie spójnym niebieskim podgrafem ścieżki B o liczbie wierzchołków równej k . Oczywiście $|E(B')| \geq k - 1$ oraz $|E(B) \setminus E(B')| \geq n - k$. W dalszej części gry konstruktor wybiera wszystkie krawędzie z $E[V(B')]$, które jeszcze nie zostały pokolorowane. Zajmuje mu to maksymalnie $\binom{k}{2} - |E(B')| \leq \binom{k}{2} - (k - 1)$ rund. Jeśli w powstałym pokolorowanym grafie

pełnym na k wierzchołkach nie ma czerwonego trójkąta, to zgodnie z twierdzeniem Turána w tym grafie znajduje się co najmniej $\binom{\lfloor k/2 \rfloor}{2} + \binom{\lceil k/2 \rceil}{2}$ krawędzi niebieskich. W zbiorze $E(B) \setminus E(B')$ znajduje się także co najmniej $n - k$ niebieskich krawędzi, więc po maksymalnie $3n - 4 + \binom{k}{2} - (k - 1)$ rundach na planszy znajduje się niebieski graf na n wierzchołkach z co najmniej $n - k + \binom{\lfloor k/2 \rfloor}{2} + \binom{\lceil k/2 \rceil}{2} \geq m$ krawędziami. Zatem

$$\tilde{r}(C_3, \mathcal{C}on_{n,m}) \leq 3n + \binom{k}{2} - k - 3.$$

Z definicji k wynika, że $m = n + \frac{k^2}{4} + O(k)$ oraz $k = O(\sqrt{m - n + 1})$, a zatem

$$\tilde{r}(C_3, \mathcal{C}on_{n,m}) \leq n + 2m + O(\sqrt{m - n + 1}).$$

□

Zauważmy, że $C_3 \in \mathcal{C}_{odd}$, w związku z czym $\tilde{r}(\mathcal{C}_{odd}, \mathcal{C}on_{n,m}) \leq \tilde{r}(C_3, \mathcal{C}on_{n,m})$.

7 Granty i darowizny

W tym rozdziale przedstawimy pewną intuicję stojącą za wyborem funkcji potencjału użytej w rozdziale 4. Zaczniemy od gry, która na pierwszy rzut oka jest niezwiązana z $\tilde{R}(C_3, P_n)$.

Dane są liczby naturalne d, r, b oraz N , rozważmy następującą grę „Granty i Darowizny”, którą będziemy oznaczać jako $GD(d, r, b, N)$. Gra ta toczy się między dwoma graczami: sponsorem i komitetem (grantowym). Dodatkowo istnieją dwa zespoły badawcze (które nie są graczami w tej grze). Co miesiąc konstruktor decyduje, co zrobić ze swoimi pieniędzmi. Może albo przekazać d dolarów bezpośrednio jednemu z zespołów, albo zasugerować jeden zespół komitetowi. W tym przypadku komitet albo daje b dolarów proponowanemu zespołowi – wtedy mówimy, że komitet dał niebieski grant, albo daje r dolarów drugiemu zespołowi – będzie to czerwony grant od komitetu.

Gra kończy się, gdy któryś z zespołów ma co najmniej N dolarów. Celem sponsora jest zakończenie gry jak najszybciej, a cel komitetu jest przeciwny. Niech $\tilde{r}(GD(d, r, b, N))$ będzie liczbą rund w grze, w której obaj gracze grają optymalnie.

Jeśli $r \leq d$ lub $b \leq d$, to najlepszą strategią dla sponsora jest przekazanie pieniędzy bezpośrednio pierwszemu zespołowi, więc założmy, że $d < r, b$. Ponadto bez straty ogólności założmy, że $d < r \leq b$.

Założmy teraz, że sponsor przekazał pieniądze zespołowi, a w następnym miesiącu sugeruje zespół komitetowi. Sponsor może zmienić kolejność tych ruchów. W tym drugim przypadku komitet ma mniej informacji w momencie podejmowania decyzji, więc nie może podjąć lepszej decyzji. Ta obserwacja prowadzi do następującego faktu.

Fakt 7.1. *Niech S będzie zbiorem optymalnych strategii sponsora. W zbiorze S istnieje strategia, w której sponsor nigdy nie sugeruje zespołu komitetowi po dokonaniu darowizny na rzecz jakiegokolwiek zespołu.*

Rozważmy teraz dwie strategie sponsora, w których sponsor nigdy nie sugeruje zespołu komitetowi po dokonaniu darowizny. Pierwsza z nich to: zawsze przekazywać pieniądze pierwszemu zespołowi. Wtedy gra kończy się po $\lceil N/d \rceil$ rundach. Druga strategia to: zawsze sugerować pierwszy zespół komitetowi. Wtedy komitet może wykonać co najwyżej $\lceil N/b \rceil$ ruchów niebieskich i co najwyżej $\lceil N/r \rceil$ ruchów czerwonych (i nie może osiągnąć obu tych liczb jednocześnie). W takim przypadku gra kończy się po co najwyżej $\lceil N/r \rceil + \lceil N/b \rceil - 1$ rundach.

Teraz rozważmy następującą strategię komitetu. Komitet daje b dolarów zespołowi wybranemu przez sponsora tylko wtedy, gdy zespół ten ma przynajmniej b dolarów mniej od drugiego zespołu (nie uwzględniając pieniędzy od sponsora). Wtedy jeśli zespoły otrzymały od komitetu odpowiednio k i k' dolarów oraz $k \geq k'$, to co najmniej k dolarów zostało przekazanych w ramach czerwonych grantów, a $k - r - b < k'$. Oznacza to, że było więcej niż $k/r + (k - r - b)/b = k(1/r + 1/b) - (r + b)/b$ grantów, czyli ruchów komitetu.

Jeśli gra się zakończyła, tzn. jeden z zespołów ma N dolarów, to wtedy było co najmniej

$$(N - k)/d + k(1/r + 1/b) - (r + b)/b \geq N \min(1/d, 1/r + 1/b) - (r + b)/b$$

rund.

Na podstawie powyższej analizy dochodzimy do następującego wniosku.

Twierdzenie 7.2. *Niech $d < r \leq b$ i $M = N \min(1/d, 1/r + 1/b)$. Wtedy*

$$M - 1 - r/b < \tilde{r}(GD(d, r, b, N)) < M + 1.$$

Zauważmy, że ponieważ $r \leq b$, to dolne i górne oszacowanie różnią się o nie więcej niż 3.

Wróćmy teraz do gry $\mathcal{G}(n, m)$ zdefiniowanej w rozdziale 4 i rozważmy jej wariant, gdzie zasady są takie same jak zasady w $\mathcal{G}(n, m)$, z dodatkowym wymaganiem, że krawędzie wybierane przez konstruktora muszą indukować spójny graf po każdej rundzie. Liczbę rund w tej grze, gdy konstruktor i malarz grają optymalnie, oznaczmy jako $\tilde{r}_{\mathcal{G}(n, m)}^{con}$. Użyjemy twierdzenia 7.2 do znalezienia dolnej granicy dla $\tilde{r}_{\mathcal{G}(n, m)}^{con}$.

Wniosek 7.3. *Niech $k > 1$ będzie liczbą rzeczywistą, m, n będą liczbami całkowitymi, $m \geq 0$ i $n \geq 2$. Wtedy*

$$\tilde{r}_{\mathcal{G}(n, m)}^{con} > (k(n - 1) + m) \min(1, 1/k + 1/(k + 1)) - k/(k + 1).$$

W szczególności,

$$\tilde{r}_{\mathcal{G}(n, m)}^{con} > \varphi n + m - 2\varphi + 1.$$

Dowód. Aby znaleźć dolną granicę zdefiniujemy strategię dla malarza. Malarz koloruje pierwszą krawędź na czerwono. W każdej następnej rundzie gra tak, aby graf indukowany przez pokolorowane krawędzie był czerwono-dwudzielny. Ponieważ jest on również spójny, ma dokładnie jeden czerwony

podział (V_1, V_2) . Zauważmy, że taka strategia jest możliwa. Istotnie, mając podział (V_1, V_2) oraz nową krawędź e wybraną przez konstruktora, jeśli e ma jeden koniec w V_1 a drugi w V_2 , to malarz pokoloruje ją na czerwono; jeśli e ma jeden koniec w V_1 , a drugi koniec jest nowym wierzchołkiem, to malarz może albo pokolorować e na czerwono i dodać nowy wierzchołek do V_2 , albo pokolorować e na niebiesko – wtedy nowy wierzchołek zostanie dodany do V_1 i krawędź niebieska zostanie dodana do $E[V_1]$.

Gra kończy się dokładnie wtedy, gdy $|V_i| \geq n$ i w $E[V_i]$ znajduje się co najmniej m niebieskich krawędzi dla $i = 1$ lub $i = 2$. Zauważmy, że ta gra jest bardzo podobna do gry Granty i Darowizny: sponsor, komitet oraz grupy badawcze odpowiadają odpowiednio konstruktorowi, malarzowi, V_1 i V_2 . Różnica polega na tym, że i -ta grupa badawcza ma dwa rodzaje „pieniędzy”: wierzchołki i niebieskie krawędzie w V_i . Niemniej jednak możemy przypisać wartość 1 do krawędzi i wartość k do wierzchołka, tak aby sprowadzić obie te rzeczy do jednej waluty. Po pierwszej rundzie malarz może zacząć udawać, że gra w $GD(1, k, k+1, N)$, gdzie $N = (n-1)k + m$ (zauważmy, że po pierwszym ruchu $|V_1| = |V_2| = 1$ i nie ma niebieskich krawędzi w $E[V_1], E[V_2]$). Zatem

$$\begin{aligned} \tilde{r}_{\mathcal{G}(n,m)}^{con} &\geq 1 + \tilde{r}(GD(1, k, k+1, (n-1)k + m)) \\ &> (k(n-1) + m) \min(1, 1/k + 1/(k+1)) - k/(k+1). \end{aligned}$$

Dla $k = \varphi$ otrzymujemy drugą część tezy. □

Powyższy wynik daje pewną intuicję na temat gry $\mathcal{G}(n, m)$, jeśli konstruktor utrzymuje graf spójny. W ogólnym przypadku trzeba przezwyciężyć trudności techniczne wynikające z wielu składowych spójności planszy, dlatego funkcja potencjału zdefiniowana w rozdziale 4 jest bardziej skomplikowana.

8 Dokładne wartości $\tilde{r}(C_4, P_n)$

Naszym kolejnym celem będzie wyznaczenie dokładnych wartości $\tilde{r}(C_4, P_n)$. Z wniosku 3.3 wiemy, że $\tilde{r}(C_4, P_n) \geq 2n - 2$. Z drugiej strony, najlepszym ogólnym oszacowaniem z góry było $\tilde{r}(C_4, P_n) \leq 3n - 5$ dla $n \geq 5$ ([10]). Ponadto $\tilde{r}(C_4, P_3) = 6$, $\tilde{r}(C_4, P_4) = 8$ ([9]), $\tilde{r}(C_4, P_5) = 9$ ([10]) oraz $\tilde{r}(C_4, P_6) = 11$ ([13]). Wszystkie wyniki z tego i następnych czterech rozdziałów pochodzą z pracy [3],

Twierdzenie 8.1. $\tilde{r}(C_4, P_n) \leq 2n - 2$ dla każdego $n \geq 8$. Ponadto, zachodzi $\tilde{r}(C_4, P_7) \leq 13$.

W świetle wspomnianego wcześniej dolnego oszacowania $\tilde{r}(C_4, P_n) \geq 2n - 2$, jak również znanej granicy $\tilde{r}(C_4, P_7) \geq 13$ ([10]), nasze twierdzenie prowadzi do następujących nowych, dokładnych rezultatów:

Wniosek 8.2. *Jeśli $n \geq 8$, to $\tilde{r}(C_4, P_n) = 2n - 2$. Ponadto $\tilde{r}(C_4, P_7) = 13$.*

Twierdzenie 8.1 rozwiązuje również inne zagadnienie związane z grami dotyczącymi ścieżek i cykli. Schudrich [15] badał rozgrywaną liczbę Ramseya dla rodziny cykli i ścieżki. Udowodnił on, że $\tilde{r}(\mathcal{C}, P_n) \leq 2,5n + 5$ dla każdego n . Twierdzenie 8.1 poprawia tę granicę dla $n \geq 8$, a w świetle dolnej granicy $\tilde{r}(\mathcal{C}, P_n) \geq 2n - 2$ uzyskanej w pracy [9], otrzymujemy następujący wynik:

Wniosek 8.3. *Jeśli $n \geq 8$, to $\tilde{r}(\mathcal{C}, P_n) = 2n - 2$.*

Interesujące jest, że krawędziowa liczba Ramseya $\hat{r}(\mathcal{C}, P_n)$ asymptotycznie nie różni się znacząco od $\tilde{r}(\mathcal{C}, P_n)$ w sensie stałych multiplikatywnych. Dokładniej, w świetle twierdzenia 8.1 oraz ograniczeń na $\hat{r}(\mathcal{C}, P_n)$ udowodnionych przez Bala i Schudricha [5], mamy:

$$1,03 + o(1) \leq \frac{\hat{r}(\mathcal{C}, P_n)}{\tilde{r}(\mathcal{C}, P_n)} \leq 1,98 + o(1).$$

Jak przyznają autorzy pracy [5], górna granica $\hat{r}(\mathcal{C}, P_n) < 3,947n + O(1)$ została udowodniona przy wsparciu komputera.

Dowód twierdzenia 8.1 opiera się na indukcji. Nasz indukcyjny argument, technicznie dość złożony, działa dla $n \geq 14$, a dwa następne rozdziały są głównie poświęcone temu dowodowi. Dla $7 \leq n \leq 13$ analizujemy grę przy pomocy komputera. Idea algorytmu została opisana w rozdziale 11, a kod w języku C++ znajduje się w dodatku A.

9 Gra $\mathbf{RR}(C_4, P_n, H)$

Będziemy analizować pomocnicze gry, w których planszą jest pokolorowany multigraf H , tzn. dopuszczamy, by plansza zawierała równoległe krawędzie, być może w różnych kolorach, przy czym nadal zakładamy, że każda krawędź w H jest albo niebieska, albo czerwona. Jeśli mówimy, że multigraf zawiera graf G , mamy na myśli, że G jest grafem prostym.

Rozszerzymy też oznaczenie na pokolorowane ścieżki, zdefiniowane w rozdziale 5. Przypomnijmy, że czerwona krawędź jest opisana przez literę r , a niebieska przez b . I tak np. brr -ścieżka oznacza ścieżkę, gdzie pierwsza krawędź jest niebieska, a kolejne dwie są czerwone.

Cztery pokolorowane multigrafy przedstawione na rysunku 1 będą miały szczególne znaczenie w dalszej analizie: powstają one z odpowiednio z b -, br -, brr - i brb -ścieżki przez dodanie czerwonej krawędzi równoległej do niebieskiej. Nazywamy je odpowiednio d -, dr -, drr - i drb -multiścieżką.



Rysunek 1: (a) d -multiścieżka; (b) dr -multiścieżka; (c) drr -multiścieżka; (d) drb -multiścieżka.

Dla $n \geq 2$ oraz pokolorowanego multigrafu H (który może być pusty), rozważmy następującą pomocniczą grę $\mathbf{RR}(C_4, P_n, H)$. Planszą gry jest multigraf na nieskończonym zbiorze wierzchołków zawierającym $V(H)$, z początkowym zestawem krawędzi $E(H)$. W każdej rundzie konstruktor wybiera pa-

rę niepołączonych wierzchołków x, y planszy, tworzy krawędź o końcach x, y , a następnie malarz koloruje ją na czerwono lub niebiesko. Pokolorowana krawędź jest dodawana do planszy. Malarzowi nie wolno pokolorować krawędzi na czerwono, jeśli miałoby to stworzyć czerwony C_4 . Konstruktor wygrywa grę $RR(C_4, P_n, H)$, jeśli $e(H) \leq 2n - 2$ oraz po maksymalnie $2n - 2 - e(H)$ rundach na planszy pojawi się niebieska ścieżka P_n i dokładnie $n - 1$ krawędzi na planszy jest niebieskich; w przeciwnym razie wygrywa malarz. Gra kończy się najpóźniej po $2n - 2 - e(H)$ rundach. Wcześniej kończy się wtedy, gdy więcej niż $n - 1$ krawędzi jest niebieskich lub pojawi się niebieska P_n . Jeśli H jest grafem pustym, grę oznaczamy przez $RR(C_4, P_n)$ zamiast $RR(C_4, P_n, H)$. Po każdej rundzie przez planszę rozumiemy pokolorowany multigraf, którego krawędziami są wszystkie pokolorowane krawędzie planszy (w tym krawędzie H). Mówimy, że wierzchołek planszy jest wolny (po danej rundzie gry), jeśli nie jest incydentny z żadną krawędzią planszy.

Zauważmy, że jeśli konstruktor wygrywa $RR(C_4, P_n)$, to po maksymalnie $2n - 2$ rundach na planszy pojawia się niebieska ścieżka P_n . W związku z tym pierwsza część twierdzenia 8.1 wynika natychmiast z poniższego, nieco silniejszego twierdzenia.

Twierdzenie 9.1. *Założmy, że G jest jednym z pokolorowanych multigrafów: grafem pustym, b -ścieżką, br -ścieżką, brr -ścieżką, brb -ścieżką, d -multiścieżką, dr -multiścieżką, drr -multiścieżką lub drb -multiścieżką. Wówczas dla każdego $n \geq 8$ konstruktor ma strategię wygrywającą w $RR(C_4, P_n, G)$.*

Powyższe twierdzenie udowodnimy w następnym rozdziale. Wcześniej przygotujemy narzędzia do przeprowadzenia dowodu indukcyjnego. Zaczniemy od lematu potrzebnego do przypadków bazowych indukcji.

Lemat 9.2. *Założmy, że $8 \leq n \leq 13$ oraz H jest jednym z pokolorowanych multigrafów: grafem pustym, b -ścieżką, br -ścieżką, brr -ścieżką, brb -ścieżką, d -multiścieżką, dr -multiścieżką, drr -multiścieżką lub drb -multi-ścieżką. Wówczas konstruktor ma strategię wygrywającą w $RR(C_4, P_n, H)$.*

Dowód tego lematu opiera się na obliczeniach komputerowych. Algorytm i jego analizę przedstawiamy w rozdziale 11.

Do analizy gry $\text{RR}(C_4, P_n, H)$ potrzebujemy kilku dodatkowych definicji. Mówimy, że konstruktor w $\text{RR}(C_4, P_n, H)$ forsownie wymusza niebieską ścieżkę długości $k \geq 1$, jeśli przez k kolejnych rund rysuje krawędzie, które malarz musi pokolorować na niebiesko zgodnie z zasadami gry.

Dla pokolorowanego multigrafu H i $A \subseteq V(H)$, przez $\text{Red}_H(A)$ oznaczamy zbiór wszystkich czerwonych krawędzi multigrafu H z co najmniej jednym końcem w zbiorze A . Analogicznie, zbiór $\text{Blue}_H(A)$ składa się ze wszystkich niebieskich krawędzi multigrafu H z co najmniej jednym końcem w A .

Założmy, że $n, k \in \mathbb{N}$ oraz H jest pokolorowanym multigrafem zawierającym niebieską ścieżkę $P = u_0 u_1 \dots u_{k+1}$. Niech $A = u_1, u_2, \dots, u_k$. Mówimy, że H' jest multigrafem otrzymanym przez skurczenie ścieżki $u_0 u_1 \dots u_{k+1}$ do krawędzi e , jeśli $V(H') = V(H) \setminus A$, zbiór $E(H')$ składa się ze wszystkich krawędzi H z końcami w $V(H) \setminus A$ oraz zawiera dodatkową niebieską krawędź $e \notin E(H)$ z końcami u_0, u_{k+1} , być może równoległą do krawędzi w H . W skrócie, przy skurczeniu ścieżki usuwamy wewnętrzne wierzchołki tej ścieżki wraz z krawędziami incydentnymi i zastępujemy je pojedynczą krawędzią.

Następujący lemat jest kluczowym elementem dowodu indukcyjnego w następnym rozdziale. Wynika z niego, że jeśli po pewnej rundzie gry $\text{RR}(C_4, P_n)$ poprzez skurczenie k niebieskich krawędzi ścieżki usuniemy co najwyżej k czerwonych krawędzi, to możemy sprowadzić grę do $\text{RR}(C_4, P_{n-k}, H')$ dla pewnej początkowej planszy H' .

Lemat 9.3. *Założmy, że $n, k \in \mathbb{N}$, H jest pokolorowanym multigrafem zawierającym taką niebieską ścieżkę $P = u_0 u_1 \dots u_{k+1}$, że*

$$\text{Blue}_H(u_1, u_2, \dots, u_k) = E(P).$$

Dodatkowo, założmy że

$$|\text{Red}_H(\{u_1, u_2, \dots, u_k\})| \leq k.$$

Niech H' będzie multigrafem otrzymanym z H przez skurczenie ścieżki P do krawędzi e . Jeśli konstruktor ma strategię wygrywającą w $RR(C_4, P_{n-k}, H')$, to ma również strategię wygrywającą w $RR(C_4, P_n, H)$.

Dowód. Niech H , P i H' będą jak w założeniu lematu. Rozważmy dwie gry: $RR(C_4, P_n, H)$ oraz $RR(C_4, P_{n-k}, H')$ i założmy, że plansza początkowa tej drugiej powstała z planszy gry $RR(C_4, P_n, H)$ przez skurczenie ścieżki $P = u_0 u_1 \dots u_{k+1}$ do krawędzi e . Założyliśmy, że każda niebieska krawędź incydentna z wierzchołkiem ze zbioru $u_1, u_2, \dots, u_k$ jest krawędzią P , więc na początku obu gier zbiory B_0 i B'_0 wszystkich niebieskich krawędzi odpowiednio w grach $RR(C_4, P_n, H)$ i $RR(C_4, P_{n-k}, H')$ spełniają warunek $B_0 = (B'_0 \setminus e) \cup E(P)$. Wynika z tego, że $|B_0| = |B'_0| - 1 + k + 1 = |B'_0| + k$.

Konstruktor w grze $RR(C_4, P_n, H)$ stosuje strategię wygrywającą, którą ma konstruktor w $RR(C_4, P_{n-k}, H')$, wybierając w każdej rundzie tę samą parę wierzchołków, co konstruktor w $RR(C_4, P_{n-k}, H')$. Zauważmy, że w ten sposób konstruktor w $RR(C_4, P_n, H)$ nigdy nie tworzy żadnej krawędzi incydentnej z którymkolwiek z wierzchołków $u_1, u_2, \dots, u_k$. Zakładamy, że konstruktor wygrywa grę $RR(C_4, P_{n-k}, H')$, więc po maksymalnie $2(n - k - 1) - e(H')$ rundach w tej grze zbiór B' wszystkich niebieskich krawędzi ma dokładnie $n - k - 1$ elementów i istnieje niebieska ścieżka P' o $n - k - 1$ krawędziach. Wobec tego $E(P') = B'$ i zawiera ona niebieską krawędź e z końcami u_0, u_{k+1} . W odpowiadającej grze $RR(C_4, P_n, H)$, po maksymalnie $2(n - k - 1) - e(H')$ rundach, zbiór B niebieskich krawędzi multigrafu planszy różni się od B' w ten sam sposób, w jaki B_0 różni się od B'_0 , więc $B = (B' \setminus e) \cup E(P)$ i $|B| = |B'| + k$. Wnioskujemy stąd, że $|B| = n - 1$, a niebieskie krawędzie $(E(P') \setminus e) \cup E(P)$ tworzą ścieżkę na n wierzchołkach.

Na koniec oszacujmy liczbę rund. Z definicji skurczenia wynika, że

$$e(H') = e(H) - |\text{Red}_H(\{u_1, u_2, \dots, u_k\})| - |\text{Blue}_H(\{u_1, u_2, \dots, u_k\})| + 1.$$

Wobec założeń lematu weryfikujemy, że

$$e(H') \geq e(H) - k - (k + 1) + 1 = e(H) - 2k.$$

Wiemy, że liczba rund w obu grach jest taka sama i nie jest większa niż $2(n - k - 1) - e(H')$, więc gra $\text{RR}(C_4, P_n, H)$ trwa nie więcej niż

$$2(n - k - 1) - e(H') \leq 2(n - k - 1) - e(H) + 2k = 2n - 2 - e(H)$$

rund. Wniosujemy, że konstruktor wygrywa grę $\text{RR}(C_4, P_n, H)$. □

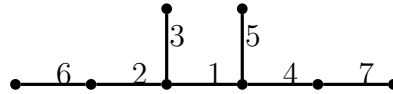
10 Dowód twierdzenia 9.1

Dowód przeprowadzimy przez indukcję względem n . Jeśli $8 \leq n \leq 13$, teza wynika z lematu 9.2.

Założmy, że $n \geq 14$ i przyjmijmy, że twierdzenie jest prawdziwe dla gry $\text{RR}(C_4, P_{n'}, G)$ dla każdego $n' < n$ oraz dla każdego G opisanego w twierdzeniu. Rozpoczynamy od analizy gry $\text{RR}(C_4, P_n)$, czyli przypadku, gdy G jest grafem pustym.

10.1 G jest pusty

Strategia konstruktora zależy od momentu, w którym malarz po raz pierwszy koloruje krawędź na niebiesko. Dopóki malarz koloruje wszystkie krawędzie na czerwono, konstruktor tworzy krawędzie grafu zgodnie z kolejnością oznaczoną na rysunku 2.

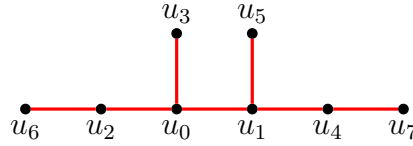


Rysunek 2: Kolejność krawędzi

Analizujemy rozgrywkę w zależności od wyborów malarza; strategia konstruktora zależy od momentu, w którym pojawia się pierwsza niebieska krawędź (o ile taka wystąpi) w ciągu pierwszych siedmiu rund.

10.1.1 Brak niebieskich krawędzi w ciągu pierwszych siedmiu rund

Jeśli malarz koloruje pierwsze siedem krawędzi na czerwono, to indukują one czerwony graf B , pokazany na rysunku 3.



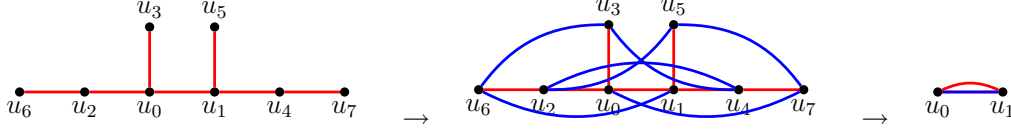
Rysunek 3: Czerwone krawędzie w ciągu pierwszych siedmiu rund

Zauważmy, że konstruktor może wymusić niebieską ścieżkę

$$P = u_0 u_7 u_5 u_2 u_4 u_3 u_6 u_1$$

na zbiorze wierzchołków $V(B)$, z końcami u_0, u_1 (zob. drugi graf na rysunku 4). Wtedy po 14 rundach plansza H ma 7 czerwonych i 7 niebieskich krawędzi.

Niech A będzie zbiorem wierzchołków wewnętrznych ścieżki P . Wówczas $|A| = 6$, $\text{Blue}_H(A)$ zawiera 7 niebieskich krawędzi ścieżki P , oraz $|\text{Red}_H(A)| = |E(B) \setminus u_0 u_1| = 6 = |A|$. Po skurczeniu P do krawędzi otrzymujemy multigraf H' , który jest d-multiścieżką. W skrócie, argument prezentujemy na rysunku 4.

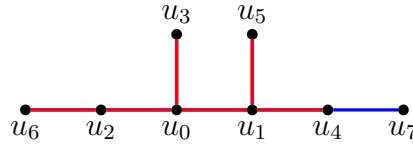


Rysunek 4: Czerwone krawędzie w ciągu pierwszych siedmiu rund. Forsowne wymuszanie i skurczenie ścieżki P .

Zauważmy, że $n - |A| = n - 6 \geq 8$, więc możemy zastosować założenie indukcyjne. Wnioskujemy, że konstruktor ma strategię wygrywającą w $\text{RR}(C_4, P_{n-|A|}, H')$. Wtedy, z lematu 9.3, otrzymujemy, że konstruktor ma strategię wygrywającą w $\text{RR}(C_4, P_n, H)$. Oznacza to również, że konstruktor wygrywa $\text{RR}(C_4, P_n, B)$.

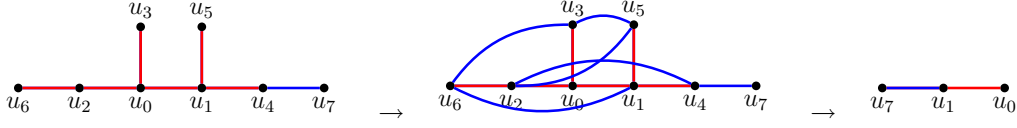
10.1.2 Pierwsza niebieska krawędź w rundzie 7

Jeśli pierwsza niebieska krawędź pojawi się w rundzie 7, otrzymujemy pokolorowany graf przedstawiony na rysunku 5.



Rysunek 5: Niebieska krawędź w rundzie 7

Konstruktor forsownie wymusza niebieską ścieżkę $u_4u_2u_5u_3u_6u_1$, która razem z niebieską krawędzią u_7u_4 tworzy niebieską ścieżkę P o końcach u_7, u_4 . Niech H będzie planszą w tym momencie. Można zweryfikować, że dla $A = u_4, u_2, u_5, u_3, u_6$ mamy $|\text{Red}_H(A)| = 5 = |A|$. Multigraf H' otrzymany z H przez skurczenie ścieżki P jest br-ścieżką $u_7u_1u_0$. Prezentujemy argument na rysunku 6.

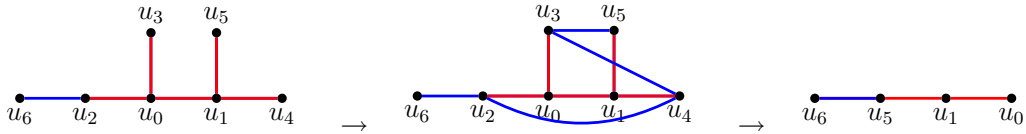


Rysunek 6: Niebieska krawędź w rundzie 7. Forsowne wymuszenie niebieskiej ścieżki i skurczenie ścieżki P .

Zauważmy, że $n - |A| = n - 5 \geq 9$. Z założenia indukcyjnego wnioskujemy, że konstruktor ma strategię wygrywającą w $\text{RR}(C_4, P_{n-|A|}, H')$. Z lematu 9.3 wynika, że konstruktor ma strategię wygrywającą w $\text{RR}(C_4, P_n, H)$. Oznacza to, że konstruktor wygrywa $\text{RR}(C_4, P_n)$ w tym przypadku.

10.1.3 Pierwsza niebieska krawędź w rundzie 6

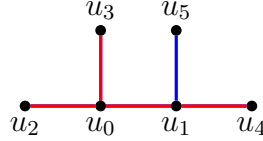
Jeśli pierwsza niebieska krawędź pojawi się w rundzie 6, otrzymujemy pokolorowany graf przedstawiony na rysunku 7 po lewej stronie. Konstruktor forsownie wymusza niebieską ścieżkę $u_5u_3u_4u_2$, która razem z niebieską krawędzią u_2u_6 tworzy niebieską ścieżkę P o końcach u_7, u_4 (zobacz drugi graf na rysunku 7). Niech H będzie planszą w tym momencie. Dla $A = u_2, u_4, u_3$ mamy $|\text{Red}_H(A)| = 3 = |A|$ oraz $n - |A| = n - 3 \geq 11$. Po skurczeniu P otrzymujemy brr-ścieżkę (zobacz ostatni graf na rysunku 7). Tak jak w poprzednim przypadku, stosując założenie indukcyjne i lemat 9.3, wnioskujemy, że konstruktor ma strategię wygrywającą w $\text{RR}(C_4, P_n, H)$.



Rysunek 7: Niebieska krawędź w rundzie 6. Forsowne wymuszenie niebieskiej ścieżki i skurczenie ścieżki P .

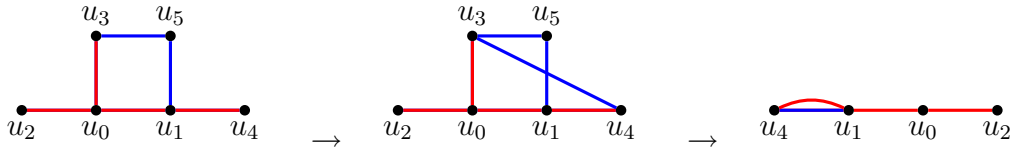
10.1.4 Pierwsza niebieska krawędź w rundzie 5

Po pięciu rundach otrzymujemy pokolorowany graf przedstawiony na rysunku 8.



Rysunek 8: Niebieska krawędź w rundzie 5

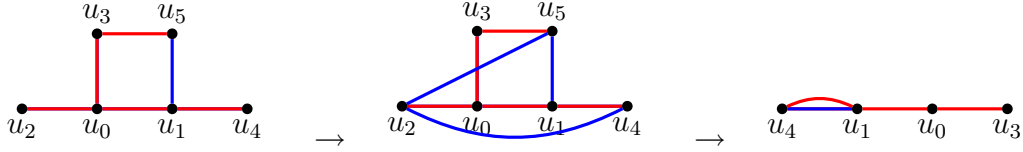
W rundzie 6 konstruktor tworzy krawędź u_3u_5 . Załóżmy, że malarz pokoloruje krawędź u_3u_5 na niebiesko. Wtedy otrzymujemy pokolorowany graf przedstawiony na rysunku 9 po lewej stronie. Konstruktor forsownie wymusza niebieską krawędź u_3u_4 , która razem z niebieską ścieżką $u_3u_5u_1$ tworzy niebieską ścieżkę $P = u_4u_3u_5u_1$ (zobacz drugi graf na rysunku 9). Niech H będzie planszą w tym momencie. Dla $A = u_3, u_5$ mamy $|\text{Red}_H(A)| = 1 < |A|$ oraz $n - |A| = n - 2 \geq 12$. Po skurczeniu P otrzymujemy drzewo-ścieżkę (zobacz ostatni graf na rysunku 9). Następnie stosując założenie indukcyjne i lemat 9.3, wnioskujemy, że konstruktor ma strategię wygrywającą w $\text{RR}(C_4, P_n, H)$.



Rysunek 9: Niebieska krawędź w rundzie 5, krawędź u_3u_5 niebieska. Forsowne wymuszanie niebieskiej krawędzi i skurczenie ścieżki P .

Założmy teraz, że malarz pokoloruje krawędź u_3u_5 na czerwono w rundzie 6. Wtedy otrzymujemy pokolorowany graf przedstawiony na rysunku 10 po lewej stronie. Konstruktor forsownie wymusza niebieską ścieżkę $u_4u_2u_5$, która razem z niebieską krawędzią u_5u_1 tworzy niebieską ścieżkę $P = u_4u_2u_5u_1$ (zobacz drugi graf na rysunku 10). Niech H będzie planszą w tym momencie.

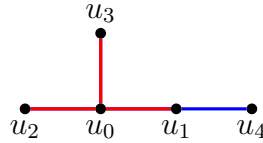
Dla $A = u_2, u_5$ mamy $|\text{Red}_H(A)| = 2 = |A|$. Po skurczeniu P otrzymujemy dr-ścieżkę (zobacz ostatni graf na rysunku 10). Tak jak w poprzednim przypadku, stosując założenie indukcyjne i lemat 9.3, otrzymujemy, że konstruktor ma strategię wygrywającą w $\text{RR}(C_4, P_n, H)$.



Rysunek 10: Niebieska krawędź w rundzie 5, krawędź u_3u_5 czerwona. Forsowne wymuszenie niebieskiej ścieżki i skurczenie ścieżki P .

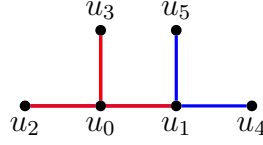
10.1.5 Pierwsza niebieska krawędź w rundzie 4

Po pokolorowaniu krawędzi na niebiesko w rundzie 4, plansza wygląda następująco, patrz rysunek 11.



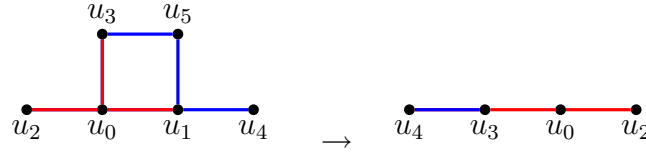
Rysunek 11: Niebieska krawędź w rundzie 4

konstruktor tworzy krawędź u_1u_5 , gdzie u_5 jest dowolnym wolnym wierzchołkiem. Jeśli malarz pokoloruje ją na czerwono, otrzymujemy graf pokolorowany izomorficzny do grafu po rundzie 5, rozważanego w podrozdziale 10.1.4. Zakładając, że malarz pokoloruje u_1u_5 na niebiesko, plansza po rundzie 5 jest przedstawiony na rysunku 12.



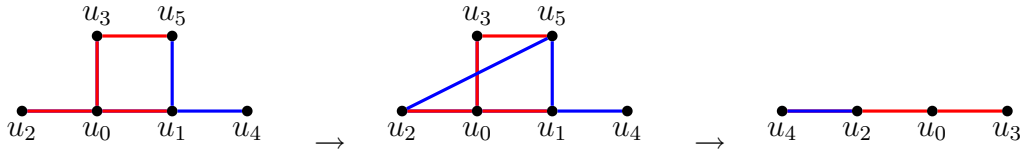
Rysunek 12: Niebieska krawędź w rundzie 4, następna krawędź jest niebieska

Następnie konstruktor tworzy u_3u_5 w rundzie 6. Jeśli malarz pokoloruje ją na niebiesko, otrzymujemy graf pokolorowany H na 6 wierzchołkach, z niebieską ścieżką $P = u_4u_1u_5u_3$, gdzie $\text{Red}_H(u_1, u_5) = 1$. Po skurczeniu tej ścieżki, otrzymujemy graf H' , który jest ścieżką *brr* $u_4u_3u_0u_2$ (patrz rysunek 13).



Rysunek 13: Niebieska krawędź w rundzie 4, krawędź u_1u_5 jest niebieska, krawędź u_3u_5 jest niebieska.

Jeśli malarz pokoloruje u_3u_5 na czerwono w rundzie 6, konstruktor w rundzie 7 zmusza do niebieskiej krawędzi u_5u_2 , a otrzymany graf pokolorowany H na 6 wierzchołkach ma niebieską ścieżkę $u_4u_1u_5u_2$ gdzie $\text{Red}_H(u_1, u_5) = 2$. Po skurczeniu tej ścieżki, otrzymujemy graf H' , który jest ścieżką *brr* $u_4u_2u_0u_3$ (patrz rysunek 14).

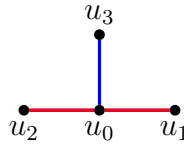


Rysunek 14: Niebieska krawędź w rundzie 4, krawędź u_1u_5 jest niebieska, krawędź u_3u_5 jest czerwona.

W obu przypadkach możemy zastosować założenie indukcyjne i lemat 9.3, i stwierdzić, że konstruktor ma strategię wygrywającą w $RR(C_4, P_n, H)$.

10.1.6 Pierwsza niebieska krawędź w rundzie 3

Jeśli pierwsza niebieska krawędź zostanie stworzona w rundzie 3, otrzymujemy graf pokolorowany przedstawiony na rysunku 15.

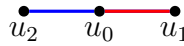


Rysunek 15: Niebieska krawędź w rundzie 3

W następnej rundzie konstruktor tworzy krawędź u_2u_3 . Jeśli malarz pokoloruje ją na niebiesko, skurczymy niebieską ścieżkę $u_2u_3u_0$ i otrzymamy dr-ścieżkę $u_2u_0u_1$. Jeśli malarz pokoloruje u_2u_3 na czerwono, wtedy konstruktor zmusza do niebieskiej krawędzi u_3u_1 . Następnie skurczymy niebieską ścieżkę $u_0u_3u_1$ i otrzymamy dr-ścieżkę $u_1u_0u_2$. W obu przypadkach, podczas skurczania niebieskiej ścieżki z jednym wewnętrznym wierzchołkiem, usuwamy najwyżej jedną czerwoną krawędź, więc możemy zastosować założenie indukcyjne i lemat 9.3, i zakończyć dowód, jak w poprzednich podrozdziałach.

10.1.7 Pierwsza niebieska krawędź w rundzie 2

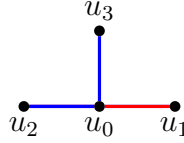
Zakładamy, że plansza po dwóch rundach jest ścieżką br , jak na rysunku 16.



Rysunek 16: Niebieska krawędź w rundzie 2

Następnie w rundzie 3 konstruktor tworzy krawędź mającą wierzchołek wspólny z u_0 i dowolnym wolnym wierzchołkiem u_3 . Jeśli malarz pokoloruje ją

na czerwono, otrzymujemy graf pokolorowany izomorficzny do grafu po rundzie 3, rozważanego w podrozdziale 10.1.6. Zakładając, że malarz pokoloruje u_0u_3 na niebiesko, plansza po rundzie 3 jest przedstawiona na rysunku 17.



Rysunek 17: Niebieska krawędź w rundzie 2, krawędź u_0u_3 jest niebieska

Następnie skurczamy niebieską ścieżkę $u_2u_0u_3$ i otrzymujemy b-ścieżkę u_2u_3 . Ponieważ skurczona ścieżka ma jeden wewnętrzny wierzchołek i usuwamy jedną czerwoną krawędź, jak w poprzednich przypadkach możemy zastosować założenie indukcyjne i lemat 9.3. W tym przypadku wniosek wynika z tego samego powodu.

10.1.8 Pierwsza niebieska krawędź w rundzie 1

W tym przypadku mamy niebieską krawędź u_0u_1 po pierwszej rundzie. W drugiej rundzie konstruktor tworzy krawędź mającą wierzchołek wspólny z u_1 i jakimś wolnym wierzchołkiem u_2 . Jeśli malarz pokoloruje u_1u_2 na czerwono, otrzymujemy graf pokolorowany rozważany w poprzednim przypadku, w podrozdziale 10.1.7. Jeśli malarz pokoloruje u_1u_2 na niebiesko, mamy niebieską ścieżkę $u_0u_1u_2$, którą można skurczyć do niebieskiej krawędzi; następnie, na podstawie lematu 9.3 oraz założenia indukcyjnego, stwierdzamy, że konstruktor wygrywa grę.

10.2 G nie jest pusty

Pozostaje udowodnić twierdzenie, gdy G jest jednym z multigrafów: b-, br-, brb-, brr-ścieżką lub d-, dr-, drb-, drr-multiścieżką. W każdym przypadku argumentacja jest podobna do tej użytej w poprzednich podrozdziałach i opie-

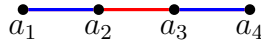
ra się na forsownym wymuszeniu krótkiej niebieskiej ścieżki, a następnie jej skurczeniu i zastosowaniu założenia indukcyjnego oraz lematu 9.3. Z powodu tych podobieństw, w niektórych przypadkach pomijamy szczegóły.

10.2.1 G jest b- lub br-ścieżką

Dla b-ścieżki gra była analizowana w podrozdziale 10.1.8. Przypadek br-ścieżki był analizowany w podrozdziale 10.1.7.

10.2.2 G jest brb-ścieżką

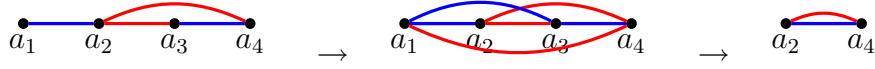
Niech będziemy oznaczać wierzchołki brb-ścieżki G tak, jak pokazano na rysunku 18.



Rysunek 18: G niepuste – brb-ścieżka

W pierwszej rundzie gry $RR(C_4, P_n, G)$ konstruktor tworzy krawędź a_2a_4 .

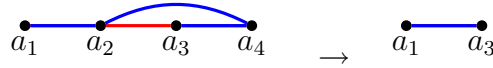
Założmy, że malarz pokoloruje krawędź a_2a_4 na czerwono. Wtedy otrzymujemy pokolorowany graf, przedstawiony jako pierwszy graf na rysunku 19. W drugiej rundzie gry $RR(C_4, P_n, G)$ konstruktor tworzy krawędź a_1a_4 i, jeśli malarz pokoloruje ją na czerwono, konstruktor forsownie wymusza niebieską krawędź a_1a_3 (zobacz drugi graf na rysunku 19). W obu przypadkach, tj. gdy krawędź a_1a_4 jest niebieska lub a_1a_4 jest czerwona, a a_1a_3 jest niebieska, otrzymujemy niebieską ścieżkę P o długości 3, z końcami sąsiadującymi w grafie czerwonym, i jest najwyżej jedna czerwona krawędź incydentna do wewnętrznego wierzchołka ścieżki P . Po skurczeniu ścieżki P , otrzymujemy d-ścieżkę (zobacz ostatni graf na rysunku 19).



Rysunek 19: G jest brb-ścieżką; a_2a_4 jest czerwone, a_1a_4 jest czerwone. Tworzenie niebieskiej krawędzi i skurczenie ścieżki P .

Stosując założenie indukcyjne i lemat 9.3, stwierdzamy, że w tym przypadku konstruktor wygrywa grę $\text{RR}(C_4, P_n, G)$.

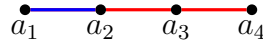
Założmy teraz, że malarz w pierwszej rundzie gry $\text{RR}(C_4, P_n, G)$ pokoloruje krawędź a_2a_4 na niebiesko. Wtedy można skurczyć niebieską ścieżkę $a_1a_2a_4a_3$ i otrzymać b-ścieżkę a_1a_3 (zobacz rysunek 20). Następnie powtarzamy argumentację z powyższego przypadku, więc pomijamy szczegóły.



Rysunek 20: G jest brb-ścieżką; a_2a_4 jest niebieskie. Skurczenie ścieżki P .

10.2.3 G jest brr-ścieżką

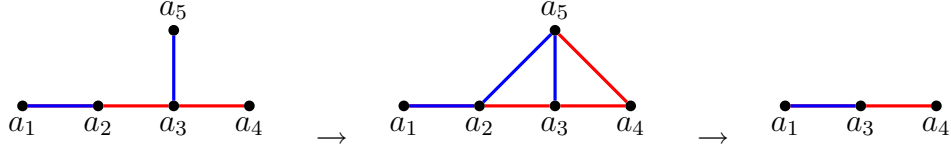
Niech będziemy oznaczać wierzchołki brr-ścieżki G tak, jak pokazano na rysunku 21.



Rysunek 21: G niepuste – brr-ścieżka

W pierwszej rundzie gry $\text{RR}(C_4, P_n, G)$ konstruktor tworzy krawędź a_3a_5 , gdzie a_5 jest wolnym wierzchołkiem.

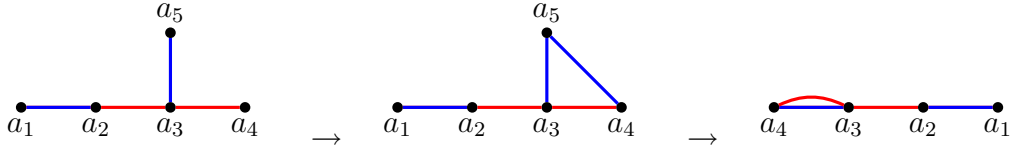
Jeśli malarz pokoloruje a_3a_5 na czerwono, to otrzymujemy pokolorowany graf, który został rozpatrzony w podrozdziale 10.1.5. Zatem, dalej zakładamy, że a_3a_5 jest niebieskie. W następnej rundzie konstruktor tworzy a_4a_5 . Jeśli malarz pokoloruje a_4a_5 na czerwono, konstruktor forownie wymusza niebieską krawędź a_2a_5 . Po skurczeniu niebieskiej ścieżki $P = a_1a_2a_5a_3$ otrzymujemy br-ścieżkę $a_1a_4a_3$. Argumentację ilustrujemy na rysunku 22.



Rysunek 22: G jest brr-ścieżką, a_3a_5 jest niebieska, a_4a_5 jest czerwona. For-sowne wymuszenie niebieskiej krawędzi i skurczenie ścieżki P .

Stosując założenie indukcyjne, stwierdzamy, że w tym przypadku konstruktor wygrywa grę $\text{RR}(C_4, P_n, G)$.

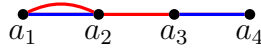
Pozostaje nam przypadek, gdy w drugiej rundzie $\text{RR}(C_4, P_n, G)$ malarz pokoloruje a_4a_5 na niebiesko. Po skurczeniu ścieżki $a_3a_5a_4$ otrzymujemy drb-multiścieżkę. Ponownie stosując lemat 9.3 i założenie indukcyjne, otrzymujemy, że konstruktor ma strategię wygrywającą w $\text{RR}(C_4, P_n, G)$.



Rysunek 23: G jest brr-ścieżką, a_3a_5 i a_4a_5 są niebieskie. Skurczenie ścieżki P .

10.2.4 G jest drb-ścieżką

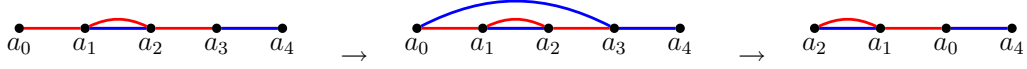
Oznaczmy wierzchołki ścieżki drb-multiścieżki G jak na rysunku 24.



Rysunek 24: drb-ścieżka

W pierwszej rundzie gry $\text{RR}(C_4, P_n, G)$ konstruktor tworzy krawędź a_1a_0 , dla wolnego wierzchołka a_0 . Załóżmy, że malarz pomaluje krawędź a_1a_0 na czerwono. Wtedy uzyskujemy planszę przedstawioną jako pierwszy multigraf

na rysunku 25. W następnej rundzie konstruktor forsownie wymusza niebieską krawędź a_0a_3 . Po skurczeniu niebieskiej ścieżki $P = a_0a_3a_4$ uzyskujemy drb-ścieżkę o końcach a_2, a_4 (patrz rysunek 25).

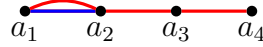


Rysunek 25: G jest drb-ścieżką; a_1a_0 jest czerwone. Forsowne wymuszenie niebieskiej krawędzi i skurczenie ścieżki P .

Drugi scenariusz, gdy malarz pomaluje krawędź a_1a_0 na niebiesko, jest jeszcze prostszy: po skurczeniu niebieskiej ścieżki $P = a_0a_1a_2$ do niebieskiej krawędzi a_1a_3 , stosujemy założenie indukcyjne i lemat 9.3.

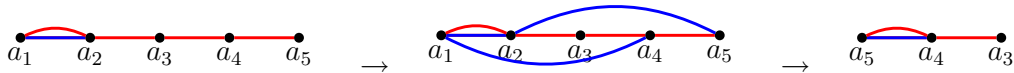
10.2.5 G jest drr-ścieżką

Niech G będzie drr-ścieżką przedstawionym na rysunku 26.



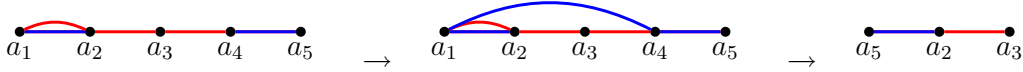
Rysunek 26: drr-ścieżka

W pierwszej rundzie gry $RR(C_4, P_n, G)$ konstruktor tworzy krawędź a_4a_5 , dla wolnego wierzchołka a_5 . Załóżmy, że malarz pomaluje krawędź a_4a_5 na czerwono. Wtedy uzyskujemy planszę przedstawioną jako pierwszy multi-graf na rysunku 27. W kolejnych dwóch rundach konstruktor forsownie wymusza niebieskie krawędzie a_2a_5 i a_1a_4 . Po skurczeniu niebieskiej ścieżki $P = a_5a_2a_1a_4$ uzyskujemy dr-ścieżkę o końcach a_5, a_3 (patrz rysunek 27).



Rysunek 27: G jest drr-ścieżką; a_4a_5 jest czerwone. Forsowne wymuszenie niebieskich krawędzi i skurczenie ścieżki P .

W drugim scenariuszu, gdy malarz pomaluje krawędź a_4a_5 na niebiesko, konstruktor forsownie wymusza niebieską krawędź a_1a_4 . Po skurczeniu niebieskiej ścieżki $a_2a_1a_4a_5$ uzyskujemy br-ścieżkę $a_5a_2a_3$ (patrz rysunek 28). W obu scenariuszach stosujemy założenie indukcyjne i lemat 9.3.



Rysunek 28: G jest drr-ścieżką; a_4a_5 jest niebieskie. Forsowne wymuszenie niebieskiej krawędzi i skurczenie ścieżki P .

10.2.6 G jest dr-ścieżką

Zweryfikowaliśmy w dwóch poprzednich podrozdziałach, że konstruktor wygrywa zaczynając od drr- lub drb-ścieżki, a zatem konstruktor wygrywa również zaczynając od dr-ścieżki.

10.2.7 G jest d-ścieżką

Niech a_1, a_2 będą końcami d-ścieżki G . W pierwszej rundzie gry $\text{RR}(C_4, P_n, G)$ konstruktor tworzy krawędź a_2a_3 , dla wolnego wierzchołka a_3 . Jeśli malarz pomaluje ją na czerwono, uzyskujemy multigraf rozważany w poprzednim podrozdziale. Jeśli malarz pomaluje ją na niebiesko, to po skurczeniu niebieskiej ścieżki $a_1a_2a_3$ do niebieskiej krawędzi a_1a_3 stosujemy założenie indukcyjne i lemat 9.3.

11 Algorytm wyznaczający $\text{RR}(C_4, P_n, H)$ dla $n \leq 13$

Rozważamy nieco zmodyfikowaną wersję gry $\text{RR}(C_4, P_n, H)$ wprowadzonej w rozdziale 9, która będzie oznaczona jako $\text{RRC}(C_4, P_n, H, v, e)$, gdzie $v, e \in \mathbb{N}$. Początkową planszą gry jest multigraf H . Dodatkowe zasady są następujące:

- (1) Konstruktor wygrywa w grze $\text{RRC}(C_4, P_n, H, v, e)$, jeśli po maksymalnie $e - |E(H)|$ rundach na planszy znajduje się czerwony C_4 lub niebieska P_n , w przeciwnym razie przegrywa.
- (2) Po każdej rundzie plansza ma co najwyżej v wierzchołków.
- (3) Po każdej rundzie plansza jest spójna.
- (4) W każdej rundzie konstruktor tworzy taką krawędź x , że multigraf indukowany przez wszystkie niebieskie krawędzie i x zawiera ścieżkę na n wierzchołkach.

Będziemy również zapisywać $rc(C_4, P_n, H, v, e) = 1$, jeśli konstruktor ma strategię wygrywającą w grze $\text{RRC}(C_4, P_n, H, v, e)$, i $rc(C_4, P_n, H, v, e) = 0$ w przeciwnym razie. Zasadnicza różnica między $\text{RRC}(C_4, P_n, H, \infty, 2n - 2)$ a $\text{RR}(C_4, P_n, H)$ polega na warunku spójności. Warunek ten sprawia, że gra jest potencjalnie trudniejsza dla konstruktora. Ponadto, jeśli $v < v'$, to $rc(G_1, G_2, H, v, e) \leq rc(G_1, G_2, H, v', e)$ (dodatkowe wierzchołki nie szkodzą konstruktorowi).

Nasz program, opisany poniżej, oblicza wartości $rc(C_4, P_n, H, n + 1, 2n - 2)$ dla $n \in 8, 9, \dots, 13$ oraz każdego H , który jest jednym z pokolorowanych multigrafów: graf pusty, b-ścieżka, br-ścieżka, brr-ścieżka, brb-ścieżka, d-multiścieżka, dr-multiścieżka, drr-multiścieżka lub drb-multiścieżka. W wyniku obliczeń otrzymaliśmy $rc(C_4, P_n, H, n + 1, 2n - 2) = 1$ w każdym przypadku. Dowodzi to lematu 9.2, ponieważ warunek $rc(C_4, P_n, H, n + 1, 2n - 2) = 1$ implikuje również strategię wygrywającą konstruktora w grze $\text{RR}(C_4, P_n, H)$.

Korzystając z programu, obliczyliśmy również, że $rc(C_4, P_7, \emptyset, 8, 13) = 1$ i dowodzi to drugiej części twierdzenia 8.1.

11.1 Idea algorytmu

Głównym celem algorytmu jest znalezienie wygrywającego ruchu konstruktora w każdej pozycji (planszy), wynikającej z każdej możliwej strategii ma-

larza. Analizujemy drzewo gry $\text{RRC}(C_4, P_n, H, v, e)$ za pomocą metody rekurencyjnej i prostej wersji standardowego przycinania alfa-beta. Poniżej przedstawiona jest idea algorytmu tworzącego tę strategię.

Rozważamy pozycję (planszę) H' . Tworzymy listę wszystkich możliwych ruchów konstruktora, które nie naruszają zasad (2), (3) i (4) gry i nie tworzą dodatkowych multikrawędzi.

Rozważamy pierwszy ruch z tej listy, powiedzmy krawędź x i oba możliwe pokolorowanie krawędzi x . Dla uzyskanego pokolorowanego multigrafu $H' \cup x$ sprawdzamy, czy jest on izomorficzny z pozycją, która została już wcześniej rozważona. Jeśli tak, używamy znanej wartości $rc(C_4, P_n, H' \cup x, v, e)$. W przeciwnym razie wywołujemy procedurę rekurencyjną, aby znaleźć $rc(C_4, P_n, H' \cup \textcolor{red}{x}^r, v, e)$ oraz $rc(C_4, P_n, H' \cup \textcolor{blue}{x}^b, v, e)$. Jeśli obie te wartości są równe 1, to stwierdzamy, że $rc(C_4, P_n, H', v, e) = 1$ i kończymy analizę pozycji H' . W przeciwnym razie rozważamy następny ruch konstruktora na liście ruchów z pozycji H' i kontynuujemy analizę.

Algorytm kończy się, jeśli na planszy pojawi się czerwony C_4 lub niebieski P_n , lub po $e - e(H)$ rundach. Ocena końcowej pozycji H' jest oczywista: ustawiamy $rc(C_4, P_n, H', v, e) = 1$, jeśli H' zawiera czerwony C_4 lub niebieski P_n ; w przeciwnym razie ustawiamy $rc(C_4, P_n, H', v, e) = 0$.

Aby skrócić czas działania programu, tworzymy listę wszystkich możliwych ruchów konstruktora w pozycji H gry $\text{RRC}(C_4, P_{n[t]}, H, v[t], e[t])$ w taki sposób, że jeśli x jest ruchem wygrywającym w pozycji H' w grze

$$\text{RRC}(C_4, P_{n[t-1]}, H, v[t-1], e[t-1]),$$

to x znajduje się na początku listy podczas analizy pozycji H' w

$$\text{RRC}(C_4, P_{n[t]}, H, v[t], e[t])$$

(funkcje $n[t], v[t], e[t]$ oznaczają tu parametry t -tej analizowanej gry w programie).

W kodzie mamy $v[t] = n[t] + 1$ dla każdego t . Innymi słowy, konstruktor nigdy nie używa więcej niż $n + 1$ wierzchołków w trakcie gry. Zauważmy, że jest to minimalna liczba — jeśli konstruktor może używać tylko n wierzchołków, malarz może uniknąć zbudowania niebieskiego P_n i czerwonego C_4 , wybierając jeden wierzchołek u i kolorując krawędź na czerwono, wtedy i tylko wtedy, gdy ma ona jeden z końców w u .

Ponadto, gdy analizujemy ruchy konstruktora w danej pozycji, zaczynamy od ruchów bez izolowanych wierzchołków. Gdy analizujemy ruchy malarza, zaczynamy od kolorowania krawędzi na czerwono, potem na niebiesko.

11.2 Weryfikacja

Dla każdej pozycji początkowej, w której konstruktor ma strategię wygrywającą, wywoływana jest funkcja `printBook(blueG, redG)`. Generuje ona strategię dla konstruktora, zapisuje ją do pliku i sprawdza jej poprawność. Funkcja ta rozważa grę $\text{RRC}(C_4, P_{n[t]}, G, v[t], e[t])$, gdzie G jest reprezentowany jako graf z niebieskimi krawędziami *blueG* oraz czerwonymi krawędziami *redG*. Pseudokod tej funkcji jest podany poniżej.

1. Dla danej pozycji początkowej składającej się z *blueG* i *redG*, znajdź wygrywający ruch uw dla konstruktora.
2. Rozważ multigraf H będący sumą *blueG*, *redG* oraz uw .
3. Jeśli liczba (nieizolowanych) wierzchołków w H jest większa niż $v[t]$ lub liczba krawędzi jest większa niż $e[t]$, zgłoś błąd.
4. Jeśli krawędź uw jest izolowana w H , a na planszy znajdują się inne krawędzie, zgłoś błąd.
5. Zapisz pozycję (czyli *blueG* i *redG*) oraz ruch gwarantujący zwycięstwo uw do pliku.

6. Jeśli aktualnie analizowana pozycja jest izomorficzna z pozycją H' występującą już w pliku, zapisz numer linii H' i zakończ.
7. Jeśli $blueG \cup uw$ nie tworzy ścieżki na $n[t]$ wierzchołkach, wykonaj `printBook(blueG  $\cup$  uw, redG)`.
8. Jeśli $redG \cup uw$ nie zawiera cyklu C_4 , wykonaj `printBook(blueG, redG  $\cup$  uw)`.

Punkt 6 nie jest konieczny, ale służy optymalizacji szybkości działania oraz zmniejszeniu rozmiaru plików wyjściowych. Nie istnieje jednak znany wielomianowy algorytm do sprawdzania izomorfizmu grafów. W związku z tym porządkujemy wierzchołki multigrafów według stopni niebieskich i czerwonych, a następnie sprawdzamy, czy multigraf o tej samej macierzy sąsiedztwa wystąpił wcześniej. W rezultacie może się zdarzyć, że algorytm nie rozpozna, iż aktualnie analizowana pozycja była już wcześniej rozpatrywana.

Aby sprawdzić, czy graf G jest ścieżką na n wierzchołkach, najpierw sprawdzamy, czy zawiera $n - 1$ krawędzi, każdy wierzchołek ma stopień co najwyżej 2 oraz czy istnieje wierzchołek u o stopniu 1. Jeśli tak, to składowa spójności grafu G zawierająca wierzchołek u jest ścieżką, więc wystarczy sprawdzić, czy ma ona $n - 1$ wierzchołków.

Aby sprawdzić, czy $G \cup uw$ zawiera cykl C_4 (zakładając, że G go nie zawiera), wystarczy sprawdzić, czy którykolwiek z sąsiadów w ma wspólnego sąsiada z u w grafie G .

Dodatkowa weryfikacja została zaimplementowana w języku Python, gdzie funkcja `validateStrategy` pełni podobną rolę jak `printBook`. Główna różnica polega na tym, że `validateStrategy` używa wyjścia `printBook` jako swojego wejścia. Ponadto wersja w Pythonie wykorzystuje kilka algorytmów z pakietu `networkx`, takich jak sprawdzanie izomorfizmu grafów czy czy graf jest drzewem (co wykorzystywane jest do sprawdzania, czy graf jest ścieżką P_n).

11.3 Informacje techniczne

Zaimplementowaliśmy powyższy algorytm w języku C++ — kod programu znajduje się w dodatku A. Program został skompilowany za pomocą kompilatora GCC w wersji 12.2.0 z flagą optymalizacji `-O2`. Uruchomiliśmy go na laptopie wyposażonym w procesor AMD Ryzen 5 5500U (6 rdzeni, 12 wątków, 2.10–4.00 GHz, 11 MB cache), 20 GB pamięci RAM (DDR4, 3200 MHz) oraz system operacyjny Linux Mint 20.3. Program przeanalizował wszystkie gry wymienione w lemacie 9.2 (oraz kilka innych) w czasie około 12 minut (717 sekund) i wykorzystał 9.27 GB pamięci operacyjnej.

W repozytorium <https://github.com/urojony/c4pn/tree/main> umieściliśmy osobny program weryfikujący, napisany w języku Python. Został on uruchomiony na tym samym komputerze z wykorzystaniem Pythona w wersji 3.8.10, biblioteki `numpy` w wersji 1.21.5 oraz `networkx` w wersji 3.1. Program potwierdził poprawność wyników w czasie około 6 minut (373 sekundy) i zużył 187 MB pamięci RAM.

12 Dowód twierdzenia 8.1

Pierwsza część twierdzenia głównego wynika z faktu, że konstruktor ma strategię wygrywającą w grze $RR(C_4, P_n)$, co stwierdzono w twierdzeniu 9.1. Druga część twierdzenia 8.1 wynika z obliczenia $rc(C_4, P_7, \emptyset, 8, 13) = 1$, o którym mowa w rozdziale 11. Oznacza ono, że konstruktor w grze $\tilde{R}(C_4, P_7)$ może wymusić utworzenie czerwonego C_4 lub niebieskiej ścieżki P_7 w czasie nie dłuższym niż 13 rund.

13 Uwagi końcowe

Twierdzenia 8.1 i 4.1 implikują, że wartości $\tilde{r}(C_4, P_n) = 2n + O(1)$ są istotnie mniejsze niż

$$\tilde{r}(C_{2k+1}, P_n) \geq \varphi n + (n - 1) - 2\varphi + 1 \approx 2,6n - 3,2$$

dla dowolnie ustalonego k . Nie jest zaskoczeniem, że łatwiej jest uniknąć nieparzystego cyklu niż parzystego i podejrzewamy, że każda gra $\tilde{R}(C_{2k}, P_n)$ jest łatwiejsza dla konstruktora niż, na przykład, gra $\tilde{R}(C_3, P_n)$. W pracy [3] postawiliśmy hipotezę, że liczby $\tilde{r}(C_{2k}, P_n)$ niewiele różnią się od $\tilde{r}(C_4, P_n)$. Dokładniej brzmiała ona tak:

Hipoteza 13.1. *Dla każdego ustalonego $k \geq 3$ zachodzi: $\tilde{r}(C_{2k}, P_n) = 2n + o(n)$.*

W pracy [4] wykazaliśmy prawdziwość tej hipotezy i pokazaliśmy poniższe cztery ograniczenia górne dla $\tilde{r}(C_k, P_n)$ oraz $\tilde{r}(C_k, C_n)$.

Twierdzenie 13.2. $\tilde{r}(C_k, C_n) \leq 2n + 20k$ dla $n \geq 3k$ oraz parzystych $k \geq 4$.

Twierdzenie 13.3. $\tilde{r}(C_k, C_n) \leq 3n + \log_2 n + 60k$ dla $n \geq 12k$ oraz nieparzystych $k \geq 3$.

Twierdzenie 13.4. $\tilde{r}(C_k, P_n) \leq 2n + 11k$ dla wszystkich n oraz parzystych $k \geq 4$.

Twierdzenie 13.5. $\tilde{r}(C_k, P_n) \leq 3n + 60k$ dla wszystkich n oraz nieparzystych $k \geq 3$.

Strategia konstruktora w dowodzie twierdzenia 8.1 ma dość zaskakującą własność — każda krawędź, którą malarz pokoloruje na niebiesko, staje się ostatecznie krawędzią niebieskiej ścieżki P_n . Zastanawiamy się, czy jest to cecha charakterystyczna dla par (C_4, P_n) , czy też może występować również w przypadku innych par (C_{2k}, P_n) . Poniżej stawiamy pytanie dotyczące strategii tego typu w przypadku (C_6, P_n) .

Pytanie 13.6. [3] Rozważmy grę $\tilde{R}(C_6, P_n)$ dla dostatecznie dużego n . Załóżmy, że malarz nigdy nie tworzy czerwonego C_6 . Czy istnieje strategia wygrywająca dla konstruktora, taka że liczba niebieskich krawędzi na końcu gry wynosi dokładnie $n - 1$?

Literatura

- [1] Natalia Adamska, Grzegorz Adamski, *Lower bounds for online size Ramsey numbers for paths*, preprint, [arXiv:2504.14926](#).
- [2] Grzegorz Adamski, Małgorzata Bednarska-Bzdęga, *Online size Ramsey numbers: Odd cycles vs connected graphs*, *Electronic Journal of Combinatorics* **31** (2024), no. P3.16, [doi:10.37236/11644](#).
- [3] Grzegorz Adamski, Małgorzata Bednarska-Bzdęga, *Online size Ramsey numbers: Path vs C_4* , *Discrete Mathematics* **347** (2024), no. 114214, [doi:10.1016/j.disc.2024.114214](#).
- [4] Grzegorz Adamski, Małgorzata Bednarska-Bzdęga, Václav Blažej, *Online Ramsey Numbers: Long versus Short Cycles*, *SIAM Journal on Discrete Mathematics* **38** (2024), 3150–3175, [doi:10.1137/23M156183X](#).
- [5] Deepak Bal, Ely Schudrich, *On the size Ramsey number of all cycles versus a path*, *Discrete Mathematics* **344** (2021), no. 112322, [doi:10.1016/j.disc.2021.112322](#).
- [6] Reinhard Diestel, *Graph Theory*, Springer Publishing Company (2017).
- [7] József Beck, *Achievement games and the probabilistic method*, *Combinatorics, Paul Erdős is Eighty, Vol. 1*, Bolyai Society Mathematical Studies (1993), 51–78.
- [8] Małgorzata Bednarska-Bzdęga, *Off-diagonal online size Ramsey numbers for paths*, *European Journal of Combinatorics* **118** (2024), no. 103873, [doi: 10.1016/j.ejc.2023.103873](#).
- [9] Joanna Cyman, Tomasz Dzido, John Lapinskas, Allan Lo, *On-line Ramsey numbers for paths and cycles*, *The Electronic Journal of Combinatorics* **22** (2015) no. P1.15. [doi:10.37236/4097](#).

- [10] Janusz Dybizbański, Tomasz Dzido, Renata Zakrzewska, *On-line Ramsey numbers for paths and short cycles*, Discrete Applied Mathematics **282** (2020), 265–270, doi:10.1016/j.dam.2020.03.004.
- [11] Przemysław Gordinowicz, Paweł Prałat, *Small on-line Ramsey numbers – a new approach*, Combinatorics, Probability and Computing **13** (2018), 101–111, doi:10.11575/cdm.v13i2.62731.
- [12] Jarosław Grytczuk, Henry Kierstead, Paweł Prałat, *On-line Ramsey numbers for paths and stars*, Discrete Mathematics and Theoretical Computer Science **10** (2008), 63–74, doi:10.46298/dmtcs.427.
- [13] Mateusz Litka, *Online size Ramsey number for C_4 and P_6* , Discussiones Mathematicae Graph Theory **44** (2024), 1607–1615, doi:10.7151/dmgt.2513.
- [14] Adva Mond, Julien Portier, *The asymptotic of off-diagonal online Ramsey numbers for paths*, European Journal of Combinatorics **122** (2024) no. 104032, doi:10.1016/j.ejc.2024.104032.
- [15] Eliyahu Schudrich, *Bipartite, size, and online Ramsey numbers of some cycles and paths*, Master’s Thesis (2021), Montclair State University.

A Kod programu wyznaczającego $\text{RRC}(C_4, P_n, H, v, e)$

```
#include <vector>
#include <array>
#include <bitset>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <cstdio>
#include <string>
#include <map>
using namespace std;
///Tmax - number of considered games
///T2max - number of considered starting positions in each game
constexpr int T_MAX=12, T2_MAX=9;
///Parameters of considered games - the T-th game
///is played on V[T] vertices, the goal for the builder is to build
///a red C4 or a blue path of length N[T] in E[T] moves.
///N should be sorted in nondescending order for faster performance
///For any T, V[T]>N[T] (otherwise Painter wins trivially or easy)
constexpr int V[]={4,5,6,7,8,8,9,10,11,12,13,14},
E[]={6,8,9,11,13,12,14,16,18,20,22,24},
N[]={3,4,5,6,7,7,8,9,10,11,12,13};
///Vmax has to be the biggest number in V[0],V[1],...,V[Vmax-1]
///If it's not V[Tmax-1], change manually
constexpr int V_MAX=V[T_MAX-1], V_MAX_SQ=V_MAX*V_MAX;
int T=0;
///will be treated as Vmax*Vmax matrix (usually representing a graph)
typedef bitset<V_MAX_SQ> matrix;
///list of starting positions: j-th position is stored in RED_EDGES[j]
///and BLUE_EDGES[j], each edge is stored as two consecutive numbers
```

```

///so e.g. {0,1,2,3} means two edges: 0-1 and 2-3;
///if a position is not possible in a given game, it will be ignored
///for example if V[0]=4 (so the set of vertices={0,1,2,3}) then all
///positions in the 0-th game which contains a vertex 4 will be ignored
char NAMES[T2_MAX][20]=
    {"empty", "b-path", "br-path", "brr-path", "brb-path",
     "d-multipath", "dr-multipath", "drr-multipath", "drb-multipath"};
const vector<int> RED_EDGES[]
    {{},{}, {1,2},{1,2,2,3},{1,2},
     {0,1},{0,1,1,2},{0,1,1,2,2,3},{0,1,1,2}},
BLUE_EDGES[]
    {{},{0,1},{0,1},{0,1}, {0,1,2,3},
     {0,1},{0,1}, {0,1}, {0,1,2,3}};

#define ST first
#define ND second

///map of all analysed positions with builder to move
///key=position
///value=0 if builder has no winning move or (v1<<8)+v2
///if (v1,v2) is winning
unordered_map<matrix,int> ANAL_POS[T_MAX];
matrix strictlyLowerTriangularMask() {
    matrix low;
    for(int i=1;i<V_MAX;++i)
        for(int j=0;j<i;++j)
            low[i*V_MAX+j]=1;
    return low;
}
const matrix LOWER_MASK=strictlyLowerTriangularMask();
matrix strictlyUpperTriangularMask(){
    matrix upp;

```

```

    for(int i=0;i<V_MAX;++i)
        for(int j=i+1;j<V_MAX;++j)
            upp[i*V_MAX+j]=1;
    return upp;
}

const matrix UPPER_MASK=strictlyUpperTriangularMask();
const array<matrix,V_MAX> verticesMasks(){
    ///masks for each row of a matrix
    array<matrix,V_MAX> ver;
    int i=0;
    for(i=0;i<V_MAX;++i)
        ver[0][i]=1;
    for(i=1;i<V_MAX;++i)
        ver[i]=ver[i-1]<<V_MAX;
    return ver;
}

const array<matrix,V_MAX> ROW_MASKS=verticesMasks();
int degree(matrix& graph,int v){
    return (graph&ROW_MASKS[v]).count();
}

struct Graph{
    matrix m; ///adjacency matrix
    int e=0; ///number of edges
    void addEdge(int v1,int v2){
        m[v1*V_MAX+v2]=m[v2*V_MAX+v1]=1;
        ++e;
    }
    void removeEdge(int v1,int v2){
        m[v1*V_MAX+v2]=m[v2*V_MAX+v1]=0;
        --e;
    }
    void clear(){

```

```

        m.reset();
        e=0;
    }
    int degree(int v){
        return ::degree(m,v);
    }
};

void reorder(matrix &graph,int invOrder[]){
    ///reorders vertices in a graph - useful for decreasing
    ///the number of unique positions in the game
    matrix res;
    for(int i=graph._Find_first();i<V_MAX_SQ;i=graph._Find_next(i))
        res[invOrder[i/V_MAX]*V_MAX+invOrder[i%V_MAX]]=1;
    graph=res;
}

matrix sortAndMerge
(matrix blueG,matrix redG,int v,int invOrder[],pair<int,int> order[]){
    ///sorts vertices of matrices by blue degree, then by red degree
    ///after that it merges two matrices into one
    ///it sends back the merged matrix and the order of sorting
    int i;
    for(i=0;i<v;++i)
        order[i]={-degree(blueG,i)*4194304-degree(redG,i)*65536,i};
    sort(order,order+v);
    for(i=0;i<v;++i)
        invOrder[order[i].ND]=i;
    reorder(blueG,invOrder);
    reorder(redG,invOrder);
    return (blueG&LOWER_MASK) | (redG&UPPER_MASK);
}

bool isPnPath(matrix graph){

```

```

///returns true iff graph is a P_N[T] path
    //return 1;
    ///if V[T]==N[T]+1, it should always return 1 when invoked
    if((int)graph.count()!=2*N[T]-2)
    ///check if number of edges equals N[T]-1
        return 0;
    int i,j,k,leaf=-1,len=0;
    ///find a leaf and check if all degrees are <=2
    for(i=0;i<V[T];++i)
        if(degree(graph,i)==1)
            leaf=i;
        else if(degree(graph,i)>2)
            return 0;
    if(leaf==-1)
        return 0;
    for(j=leaf;;){ ///start from the leaf, walk and burn edges
        k=graph._Find_next(j*V_MAX-1)-j*V_MAX;
        if(k>=V_MAX)
            break;
        graph[j*V_MAX+k]=graph[k*V_MAX+j]=0;
        ++len;
        j=k;
    }
    return len+1==N[T];
}

bool hasC4(Graph& graph,int i,int j){
    ///checks if graph has a C4 cycle which contains i-j edge
    if(graph.e<4)
        return 0;
    auto graph2=graph.m&ROW_MASKS[i];
    ///macro for iterating a bitset over interval [beg,end)
    #define FOR_BS(i,bset,beg,end)\

```

```

for(int i=(bset)._Find_next((beg)-1);i<end;i=(bset)._Find_next(i))
    ///for any j's neighbour x, different from i, check if x and i
    ///have more than one common neighbour;
    ///in this implementation, k=x+j*V_MAX
FOR_BS(k,graph.m,j*V_MAX,j*V_MAX+i){
    if(((graph2>>((i-k+j*V_MAX)*V_MAX))&graph.m).count(>1)
        return 1;
    }
FOR_BS(k,graph.m,j*V_MAX+i+1,j*V_MAX+V_MAX){
    if(((graph2<<((k-j*V_MAX-i)*V_MAX))&graph.m).count(>1)
        return 1;
    }
    return 0;
}

bool hasOnlyPaths(matrix graph,int v1,int v2){
    ///assuming graph contains only disjoint paths, check if
    ///the edge v1-v2 doesn't destroy this property
    int j,k;
    const int d1=degree(graph,v1),d2=degree(graph,v2);
    if(d1>1 || d2>1)
        return 0;
    if(d1==0 || d2==0){
        return 1;
    }
    for(j=v1;;){
        ///find the other end of the path starting with v1
        k=graph._Find_next(j*V_MAX-1)-j*V_MAX;
        if(k>=V_MAX)
            break;
        graph[j*V_MAX+k]=graph[k*V_MAX+j]=0;
        j=k;
    }
}

```

```

    return j!=v2;
}
int NUM_OF_UNIQ_POS=0,NUM_OF_ALL_POS=0;
bool colour(Graph blueG,Graph redG,int v,int i,int j);
bool construct(Graph blueG,Graph redG,int v){
    ///returns 1 iff builder has a winning move
    if(blueG.e>=N[T] || redG.e>E[T]-N[T]+1)
        return 0;
    int invOrder[v+3];
    pair<int,int> order[v+3];
    auto position=sortAndMerge(blueG.m,redG.m,v,invOrder,order);
    ++NUM_OF_ALL_POS;
    if(ANAL_POS[T].count(position)) ///if isomorphic position was reached
        return ANAL_POS[T][position]; ///return the result
    ++NUM_OF_UNIQ_POS;
    if((NUM_OF_UNIQ_POS&((1<<23)-1))==0)
        printf("Unique positions analysed so far: %d\n",NUM_OF_UNIQ_POS);
    if(blueG.e+redG.e==E[T])
        return 0;
    order[v].ND=v;
    order[v+1].ND=v+1;
    matrix allEdges=blueG.m|redG.m; ///union of blue and red graphs
    if(T>0 && ANAL_POS[T-1].count(position) && ANAL_POS[T-1][position]){
        ///checks if in the previous game this position was reached
        int moove=ANAL_POS[T-1][position];
        int v1=order[moove>>8].ND,v2=order[moove&255].ND;
        if(colour(blueG,redG,max({v,v1+1,v2+1}),v1,v2)){
            ///try to play the same move
            ANAL_POS[T][position]=moove;
            return 1;
        }
    }
    else

```

```

        allEdges[v1*V_MAX+v2]=allEdges[v2*V_MAX+v1]=1;
    }
    ///try moves with vertices that already have edges
    for(int i=v-1;i>=0;--i)
        if(blueG.degree(i)<=1)
            for(int j=v-1;j>i;--j)
                if(!allEdges[i*V_MAX+j] && blueG.degree(j)<=1){
                    if(colour(blueG,redG,v,i,j)){
                        ANAL_POS[T][position]=(invOrder[i]<<8)+invOrder[j];
                        return 1;
                    }
                }
            else
                allEdges[i*V_MAX+j]=allEdges[j*V_MAX+i]=1;
        }
    ///try moves with an unused vertice
    if(v<V[T])
        for(int i=0;i<v;++i){
            if(blueG.degree(i)<=1&& !allEdges[i*V_MAX+v]
                && colour(blueG,redG,v+1,i,v)){
                ANAL_POS[T][position]=(invOrder[i]<<8)+v;
                return 1;
            }
        }
    ///the first move has two unused vertices
    if(v==0 && colour(blueG,redG,v+2,v,v+1)){
        ANAL_POS[T][position]=(v<<8)+v+1;
        return 1;
    }
    ///if no move is winning for Builder, mark
    ///this position as losing for him
    ANAL_POS[T][position]=0;
    return 0;

```

```

}
void print(const matrix& mergedG,int moove){
    printf("r:␣");
    int i,j;
    for(i=0;i<V_MAX;++i)
        for(j=i+1;j<V_MAX;++j)
            if(mergedG[i*V_MAX+j])
                printf("%X%X␣",i,j);
    printf("b:␣");
    for(i=1;i<V_MAX;++i)
        for(j=0;j<i;++j)
            if(mergedG[i*V_MAX+j])
                printf("%X%X␣",i,j);
    printf("m:␣%X%X\n",moove>>8,moove&255);
}

void print(const matrix& blueG,const matrix& redG,
           int moove,FILE *fp,bool newline=1){
    fprintf(fp,"r:␣");
    int i,j;
    for(i=0;i<V_MAX;++i)
        for(j=i+1;j<V_MAX;++j)
            if(redG[i*V_MAX+j])
                fprintf(fp,"%X%X␣",i,j);
    fprintf(fp,"b:␣");
    for(i=0;i<V_MAX;++i)
        for(j=i+1;j<V_MAX;++j)
            if(blueG[i*V_MAX+j])
                fprintf(fp,"%X%X␣",i,j);
    fprintf(fp,"m:␣%X%X",moove>>8,moove&255);
    if(newline)
        fprintf(fp,"\n");
}

```

```

bool colour(Graph blueG, Graph redG, int v, int i, int j){
    ///returns 0 iff painter has a winning move
    if(!hasOnlyPaths(blueG.m,i,j))
        return 0;
    blueG.addEdge(i,j);
    bool isVertexWithoutBlueEdges=0;
    for(int i=V[T]-1;i>=0;--i)
        if(blueG.degree(i)==0){
            isVertexWithoutBlueEdges=1;
            break;
        }
    if(!isVertexWithoutBlueEdges) ///if all vertices have blue edges,
        return 0; ///then at least one edge will be wasted
    ///paint i-j red and check who's winning
    blueG.removeEdge(i,j);
    redG.addEdge(i,j);
    if(!hasC4(redG,i,j) && !construct(blueG,redG,v)){
        return 0;
    }
    ///paint i-j blue and check who's winning
    redG.removeEdge(i,j);
    blueG.addEdge(i,j);
    if((blueG.e+2<N[T] || !isPnPath(blueG.m))&& !construct(blueG,redG,v)){
        return 0;
    }
    return 1;
}

unordered_map<matrix, int> BOOK_POS;
int LINE_NUMBER;
void printBook(Graph blueG, Graph redG, int v, int e, FILE *fp){
    ///this function is only invoked when Builder
    ///has a winning strategy in RRC(C_4,P_N[T],H,V[T],E[T])

```

```

///where H is represented by blueG and redG;
///it boths verifies a strategy for Builder and
///prints it to a file;
///however, it is only verified whether  $r_H(C_4, P_N[T]) \leq E[T]$ ,
///so for example connectivity condition is not checked here
    int invOrder[v+3];
    pair<int,int> order[v+3];
    auto position=sortAndMerge(blueG.m,redG.m,v,invOrder,order);
    order[v].ND=v;
    order[v+1].ND=v+1;
    if(!ANAL_POS[T].count(position)){
        ///this if statement is for a case, when the function
        ///can't find a given position among analysed positions;
        ///it doesn't mean it wasn't analysed, but rather
        ///the isomorphic position was reached
        ///and now this function can't find it
        construct(blueG,redG,v);
    }
    int u,w,moove;
    moove=ANAL_POS[T][position];
    if(moove==0){
        ///if ANAL_POS database says that this position is losing or
        ///maximum number of moves has been reached
        printf("Can't find winning strategy for Builder\n");
        exit(2137); ///this shouldn't happen
    }
    u=moove/256;
    w=moove%256;
    u=order[u].ND;
    w=order[w].ND;
    ///at this point move u-w is chosen for Builder
    if(max(u,w)>v && v>0){

```

```

    ///if new edge is isolated and it's not first
    printf("Connectivity_condition_failed\n");
    exit(420);
}
if(e>=E[T]){
    printf("Maximum_number_of_moves_has_been_reached\n");
    exit(57);
}
if(max(u,w)>=V[T]){
    printf("Too_many_vertices_are_used_in_the_game\n");
    exit(42);
}
if(u==w){
    printf("Loops_are_not_allowed\n");
    exit(934);
}
if(u<0 || w<0){
    printf("Invalid_vertices\n");
    exit(0xEEEEEE);
}
for(int h=0;h<e;++h)
    fprintf(fp,"_");
print(blueG.m,redG.m,u*256+w,fp,0);
if(BOOK_POS.count(position)){
    ///if the isomorphic position was reached in this file
    ///print the line number and
    ///stop going down the decision tree
    fprintf(fp,"_l:_%d\n",BOOK_POS[position]);
    ++LINE_NUMBER;
    return;
}
else{

```

```

        fprintf(fp, "\n");
        BOOK_POS[position]=++LINE_NUMBER;
    }
    ///analyze the decsion subtree where Painter colours the edge blue
    blueG.addEdge(u,w);
    if(!isPnPath(blueG.m))
        printBook(blueG,redG,max({v,u+1,w+1}),e+1,fp);
    blueG.removeEdge(u,w);
    ///analyze the decision subtree where Painter colours the edge red
    redG.addEdge(u,w);
    if(!hasC4(redG,u,w))
        printBook(blueG,redG,max({v,u+1,w+1}),e+1,fp);
}

int fillGraph(Graph& blueG,Graph& redG,const int t2){
    ///puts t2-th starting position into blueG and redG
    blueG.clear();
    redG.clear();
    for(int i=1;i<(int)BLUE_EDGES[t2].size();i+=2)
        blueG.addEdge(BLUE_EDGES[t2][i],BLUE_EDGES[t2][i-1]);
    for(int i=1;i<(int)RED_EDGES[t2].size();i+=2)
        redG.addEdge(RED_EDGES[t2][i],RED_EDGES[t2][i-1]);
    int mxVertexIndex=-1;
    for(int i:BLUE_EDGES[t2])
        mxVertexIndex=max(i,mxVertexIndex);
    for(int i:RED_EDGES[t2])
        mxVertexIndex=max(i,mxVertexIndex);
    return mxVertexIndex;
}

int main(){
    Graph blueG,redG;
    FILE *fp;
    for(T=0;T<T_MAX;++T){

```

```

BOOK_POS.clear();
LINE_NUMBER=0;
char filename[100];
sprintf(filename,"C4P%d_in_%d_moves_%d_verts.txt",
        N[T],E[T],V[T]);
fp=fopen(filename,"w");
for(int t2=0;t2<T2_MAX;++t2){
    ///iterates through every of the 9 starting positions
    int mxVertexIndex=fillGraph(blueG,redG,t2);
    if(mxVertexIndex>=V[T])
        continue;
    int result=construct(blueG,redG,mxVertexIndex+1);
    fprintf(fp,"rc(C4,P%d,%s,%d,%d)=%d\n",
            N[T],NAMES[t2],V[T],E[T],result);
    printf("rc(C4,P%d,%s,%d,%d)=%d\n",
            N[T],NAMES[t2],V[T],E[T],result);
    printf("unique/total_positions_analysed:");
    printf("%d\n",NUM_OF_UNIQ_POS,NUM_OF_ALL_POS);
    ++LINE_NUMBER;
    if(result)
        printBook(blueG,redG,mxVertexIndex+1,
            BLUE_EDGES[t2].size()/2+RED_EDGES[t2].size()/2,fp);
}
fclose(fp);
}
}

```