Faktoryzacja iloczynu kartezjańskiego grafów

Andrzej Kawula
Promotor: dr Monika Pilśniak

30.06.2018

Spis treści

1 Wprowadzenie

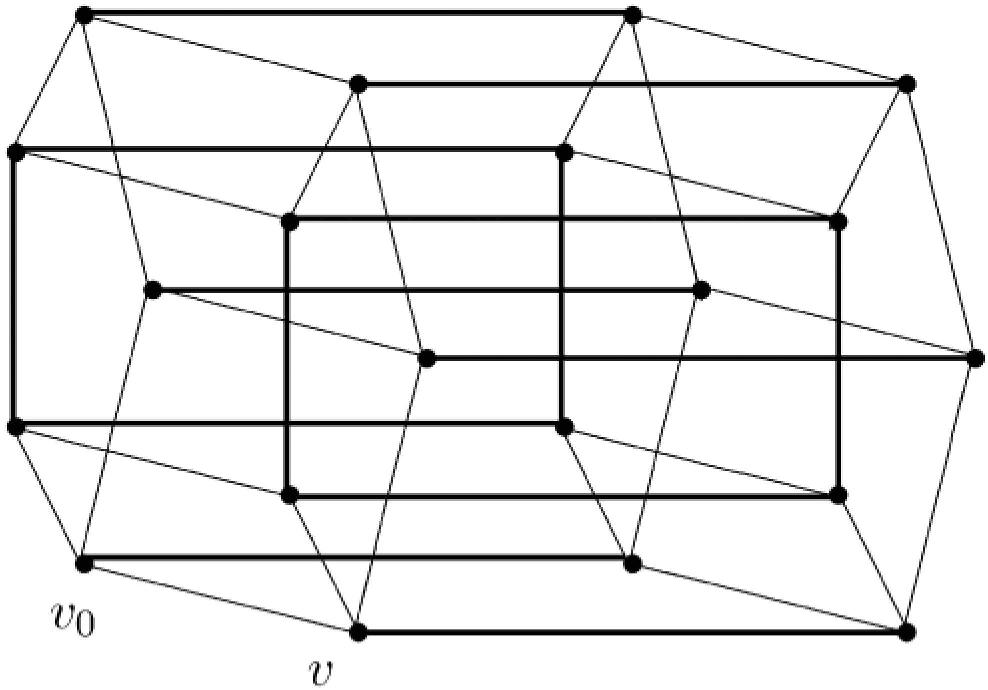
2 Wstęp

2.1 Informacje wstępne

Produktem kartezjańskim $G_1 \square G_2$ grafów $G_1 = (V_1, E_1)$ i $G_2 = (V_2, E_2)$ nazywamy graf $G = (V, E)$, którego zbiorem wierzchołków jest iloczyn kartezjański wierzchołków grafów G_1 i G_2 ($V = V_1 \times V_2$), natomiast wierzchołki (x_1, y_1) oraz (x_2, y_2) są połączone w grafie G jeżeli $x_1 = x_2$ oraz $y_1 y_2 \in E_2$ lub $x_1 x_2 \in E_1$ oraz $y_1 = y_2$.
Iloczyn kartezjański grafów jest działaniem łącznym, przemiennym, z dokładnością do izomorfizu, elementem neutralnym działania jest graf K_1.
Z łączności działania możemy zapisać $G_1 \square G_2 \square ... \square G_k = G$ gdzie G jest produktem kartezjańskim grafów $G_1, G_2, ..., G_k$, a następnie poetykietować wierzchołki grafu G k-elementową listą $(v_1, v_2, ... v_k)$ gdzie $v_i \in V(G_i)$ dla $1 \leqslant i \leqslant k$. Jeżeli v etykietowany jest przez listę $(v_1, v_2, ... v_k)$, można zdefiniować rzutowanie $p_i : V \to V_i$ dla $1 \leqslant i \leqslant k$, które dane jest wzorem $p_i(v) = v_i$, gdzie v_i jest i-tym elementem listy etykietującej wierzchołek v. Wierzchołek v_i ten będzie i-tą współrzędną wierzchołka v.
Jeżeli w grafie G dany jest wierzchołek v i rozważymy wierzchołki, które różnią się od wierzchołka v tylko na i-tej pozycji, to podgraf indukownay przez te wierzchołki utworzy graf izomorficzny z grafem G_i. Podgraf ten będzie nazywany i-tą warstwą G_i przechodzącą przez wierzchołek v a jego oznaczeniem będzie G_i^v.
Niech v_0 będzie wyróżnionym wierzchołkiem w grafie G. Warstwy przechodzące przez v_0 nazywamy warstwami jednostkowymi. Wierzchołek v_0 należy do każdej warstwy jednostkowej, natomiast zbiory $V(G_i^{v_0}) \setminus \{v_0\}$ są parami rozłączne dla $1 \leqslant i \leqslant k$.
Rysunek będzie zmieniony

2.2 Kolorowanie produktu iloczynu kartezjańskiego grafów

Niech dane będą dwa połączone wierzchołki u oraz v w grafie G. Analizując współrzędne tych wierzchołków, łatwo można stwierdzić że różnią się one dokładne na jednej pozycji. Niech i oznacza tę pozycję. Wtedy krawędź uv należy do G_i^v. Krawędzi uv otrzymuje kolor i czyli $c(uv) = i$, gdzie c jest kolorowaniem właściwym produktu kartezjańskiego. Podsumowując: funkcja $c : E(G) \to 1, 2, ..., k$ jest kolorowaniem właściwym produktu iloczynu kartezjańskiego jeżeli $c(uv) = i$ wtedy i tylko wtedy gdy współrzędne wierzchołków u oraz v różnią się na i-tej pozycji.



Każda krawędź należy dokładnie do jednej warstwy. Rozważając podgraf grafu G składający się z krawędzi koloru i to każda spójna składowa tego podgrafu będzie oddzielną i-tą warstwą grafu G.

2.3 Lemat o kwadracie

Niech graf G posiada właściwe pokolorowanie produktu kartezjańskiego. Dane są dwie połączone krawędzie e i f różnych kolorów. Wówczas istnieje dokładnie jeden kwadrat bez przekątnych (graf C_4) zawierający e oraz f.
Dowód:
Na początku rozważmy następujący fakt. Każdy trójkąt w grafie G jest pomalowany na ten sam kolor. Wynika to z tego, że dwa pierwsze wierzchołki różnią się na pozycji i-tej, natomiast jeżeli współrzędne trzeciego wierzchołka różniły by się na pozycji j-tej która jest różna od i w porównaniu z pierwszym wierzchołkiem i jego współrzędnymi, nie było by możliwości aby drugi i trzeci wierzchołek byłby by ze sobą połączone, ponieważ ich współrzędne różniły by się na dwóch pozycjach.
Na podstawie powyższego stwierdzenia stwierdzamy, że każdy kwadrat zawierający co najmniej jedną przekątną jest tego samego koloru.
Teraz właściwy dowodu lematu. Dane są następujące oznaczenia:
$(v_0, v_1, ..., v_i, ..., v_j, ..., v_k)$ współrzędne wspólnego wierzchołka v krawędzi e oraz f

$(v_0, v_1, ..., v'_i, ..., v_j, ..., v_k)$ współrzędne wierzchołka v_e który jest drugim końcem krawędzi e
$(v_0, v_1, ..., v_i, ..., v'_j, ..., v_k)$ współrzędnymi wierzchołka v_f który jest drugim końcem krawędzi f
v' wierzchołek o współrzędnych $(v_0, v_1, ..., v'_i, ..., v'_j, ..., v_k)$
Łatwo stwierdzić, że wierzchołek v' jest połączony z wierzchołkiem v_e ponieważ ich współrzędne różnią się tylko na pozycji j oraz w grafie G_j istnieje krawędź $v_j v'_j$ ponieważ w grafie G istnieje krawędź f. Analogicznie stwierdzamy istnienie krawędzi $v'v_f$, co w połączeniu z faktem, że krawędzie e oraz f są różnego koloru i rozważaniom na temat kwadratów z przekątnymi daje nam tezę lematu. Co więcej na podstawie powyższego rozumowania, można wnioskować że przeciwległe krawędzie w kwadracie mają ten sam kolor, niezależnie czy kwadrat posiada przekątne czy też nie.

2.4 Lemat o izomorfizmie

Niech $G = (V, E)$ będzie spójnym grafem, natomiast $E_1, E_2, ..., E_k$ podziałem zbioru krawędzi. Niech każda spójna składowa $(V, \cup_{j \neq i} E_j)$ ma dokładnie jeden punkt wspólny z każdą spójną składową (V, E_i) oraz krawędzie między dwoma składowymi (V, E_i) wyznaczają izomorfizm między tymi składowymi (jeżeli takie krawędzie istnieją). Wtedy:
$G = \Pi G_i$
gdzie G_i jest dowolną, spójną składową (V, E_i).
Dowód:
Tutaj chyba dowód indukcyjny ze względy na k, do sprawdzenia.

2.5 Lemat o udoskonaleniu faktoryzacji iloczynu kartezjańskiego

3 Faktoryzacja z dodatkowymi informacjami

W tym rozdziale przedstawimy algorytm kolorowania właściwego grafu względem iloczynu kartezjańskiego mając podane kolory krawędzi wychodzących z pewnego wierzchołka, względem danego rozkładu grafu. Następnie pokażemy jak mając kolory wszystkich krawędzi nadać współrzędne wszystkim wierzchołkom.

3.1 Algorytm kolorowania krawędzi

Załóżmy, że mamy dane kolory wszystkich krawędzi, w kolorowaniu iloczynu kartezjańskiego grafu, wychodzących z pewnego wierzchołka v_0. Kolorowanie pozostałych krawędzi będzie odbywało się w kolejności przeszukiwania grafu w algorytmie BFS z wierzchołkiem początkowym v_0.
Twierdzenie 3.1.1 Niech $G{=}G_1\square G_2\square\ldots\square G_k$ będzie grafem spójnym. Dane jest kolorowanie właściwe względem podanego rozkładu dla wszystkich krawędzi wychodzących z pewnego wierzchołka v_0. Wtedy kolorowanie właściwe produktu kartezjańskiego może być uzyskane zgodnie z kolejnością algorytmu BFS o wierzchołku początkowym v_0. Złożoność czasowa tego algorytmu to $O(mn)$, natomiast złożoność pamięciowa $O(n^2)$.

Dowód
Jako dowód przedstawiony zostanie algorytm kolorowania krawędzi. W pierwszym kroku algorytmu dzielimy zbiór wierzchołków grafu G na podzbiory $L_0, L_1, L_2, \ldots, L_r$ w taki sposób, że wierzchołek v należy do zbioru L_i wtedy i tylko wtedy gdy odległość wierzchołka v od wierzchołka v_0 jest równa i. Zbiory te będziemy nazywać poziomami. Następnie dla każdego wierzchołka, wszystkie krawędzie incydentne z tym wierzchołkiem dzielimy na trzy zbiory- zbiór krawędzi dolnych, poprzecznych i górnych. Definiowanie tych zbirów przebiega następująco. Rozważy poziom i, następnie dla wszystkich wierzchołków v należących do zbioru $L_i$rozważamy krawędzie vu incydentne z v. Wówczas jeżeli u należy do L_{i-1} to krawędź vu będzie krawędzią dolną. Jeżeli u należy do L_i wówczas uv będzie krawędzią poprzeczną, jeżeli natomiast u należy do L_{i+1} wówczas uv będzie krawędzią górną wierzchołka warstwy L_i. Zauważmy, że krawędzie dolne wierzchołków poziomu L_{i+1} są krawędziami górnymi wierzchołków poziomu L_i .

Nasz algorytm rozpoczynamy od pokolorowania krawędzi poprzecznych L_1. Nie stanowi to problemy, ponieważ każdy trójkąt jest monochromatyczny. Tak więc każdej krawędzi uv nadajemy kolor krawędzi v_0v czyli $c(uv) := c(vv_0) = c(uv_0)$.

Następnie indukcyjnie kolorujemy krawędzie dolne a następnie poprzeczne L_{i+1}

mając już pokolorowane krawędzie dolne i poprzeczne L_i. Nie ma potrzeby kolorowania krawędzi górnych L_i ponieważ zbiór ten jest również zbiorem krawędzi dolnych L_{i+1}.

Zaczynamy od krawędzi dolnych. Przeglądamy wierzchołki należące do L_{i+1} zgodnie z kolejnością wyznaczoną przez algorytm BFS. Niech dany będzie wierzchołek u oraz krawędź uv. Ponieważ wierzchołek v należy do L_i, gdzie $i \geqslant 1$ to istnieje wierzchołek w należący do L_{i-1} sąsiedni z v. Rozważmy dwa przypadki:

1. Nie istnieje wspólny sąsiad wierzchołków u oraz w różny od v. Wówczas nie istnieje kwadrat zawierający wierzchołki u oraz w a co za tym idzie kolory krawędzi uv oraz vw są te same czyli $c(uv) := c(vw)$.

2. Istnieje wspólny sąsiad x wierzchołków u oraz w różny od v. W tym przypadku $c(uv) := c(xw)$ oraz $c(ux) := c(vw)$.

Uzasadnienia w obydwu przypadkach wynikają z lematu o kwadracie (2.3).
Rozważmy teraz krawędzie poprzeczne L_{i+1}. W tym celu również przeglądamy wierzchołki należące do tego poziomu. Dla każdej krawędzi uv należącej do krawędzi poprzecznych rozważanego poziomu szukamy krawędzi dolnej uw i podobnie jak dla krawędzi dolnych szukamy wspólnego sąsiada wierzchołków v oraz w. Jeśli takowy wierzchołek x istnieje wówczas $c(uv) := c(wx)$, jesli nie $c(uv) := c(uw)$.
Zauważmy, że aby wyznaczyć G_i wystarczy znaleźć $G_i^{v_0}$. Wierzchołek v należy do $V(G_i^{v_0})$ wtedy i tylko wtedy gdy wszystkie jego krawędzie dolne są koloru i. Tak więc aby wyznaczyć G_i wystarczy przejrzeć wszystkie krawędzie dolne wszystkich wierzchołków, jeżeli lista ta jest monochromatyczna wierzchołek ten będzie należał do $V(G_i^{v_0})$ gdzie i to kolor krawędzi dolnych tego wierzchołka.
W rozdziale 2.1 zdefiniowaliśmy warstwy jednostkowe $G_i^{v_0}$. Wierzchołki należące do $G_i^{v_0}$ będziemy nazywać wierzchołkami warstwy jednostkowej. Oczywiście wszystkie wierzchołki należące do L_1 będą wierzchołkami warstw jednostkowych.
Rozważając jeszcze raz krawędzie dolne oraz poprzeczne wierzchołków warstw jednostkowych, na podstawie spójności produktu iloczynu kartezjańskiego stwierdzamy, że i krawędzie dolne i krawędzie górne tychże wierzchołków należą do warstw jednostkowych.
Na koniec pozostaje nam wykazanie, że nasz algorytm rzeczywiście spełnia założenia dotyczące złożoności pamięciowej i czasowej. Zauważmy że dla każdego wierzchołka należącego do L_i, gdzie $i > 0$ szukamy dolnego sąsiada, następnie przeglądamy wszystkie krawędzie dolne oraz poprzeczne. Tak więc wykonujemy co najwyżej $2m$ kroków w naszym algorytmie, gdzie m oznacza liczbę krawędzi naszego grafu G. Dla ustalonych krawędzi uv oraz uw szukamy wspólnego sąsiada x. Mamy co najwyżej $n = G(V)$ możliwości wyboru tego sąsiada. Jeżeli informacje o krawędziach grafu są przetrzymywane w tablicy sąsiedztwa sprawdzenie czy

dany wierzchołek jest sąsiadem innego można wykonać w czasie stałym. Tak więc dowiedliśmy, że złożoność czasowa algorytmu to $O(mn)$, natomiast złożoność pamięciowa $O(n^2)$.

3.2 Nadawanie współrzędnych wierzchołkom

W poprzednim podrozdziale opisaliśmy algorytm kolorowania krawędzi, jednakże nie podawaliśmy sposobu, jak nadać współrzędne wierzchołkom. Zdefiniujemy teraz algorytm, który nada współrzędne naszym wierzchołkom, mając już dane kolory wszystkich krawędzi.

Twierdzenie 3.2.1 Niech G=$G_1\square G_2\square ...\square G_k$ *będzie grafem spójnym. Dane jest również kolorowanie właściwe produktu względem podanego rozkładu. Algorytm nadania współrzędnych wierzchołkom może być zrealizowany w złożoności czasowej i pamięciowej* $O(m)$.

Dowód
Dla każdego wierzchołka, aby móc przechować informację o jego współrzędnych, potrzebujemy k-elementowej tablicy. Ponieważ k jest mniejsze od minimalnego stopnia grafu oraz $\delta(G)\cdot n \leqslant m$ to całkowity rozmiar tablic ze współrzędnymi jest mniejszy bądź równy $O(m)$.
Algorytm rozpoczynamy od nadania wierzchołkowi v_0 współrzędnych składających się z samych 0. Następnie przeszukujemy wszystkie wierzchołki zgodnie z kolejnością algorytmu BFS.
Jeżeli wierzchołek należy do i-tej warstwy jednostkowej jego wszystkie współrzędne otrzymują wartość 0 z wyłączeniem i-tej współrzędnej. Przeszukując wierzchołki i-tej warstwy jednostkowej i-tej współrzędnej nadajemy kolejną liczbę naturalną. Zapisując formalnie jeżeli nasz wierzchołek u należy do warstwy jednostkowej $u_j = 0$ dla $j \neq i$ oraz $u_i = max\{v_i\}+1$ gdzie v_i to i-te współrzędne wierzchołków należących do G_i odwiedzonych wcześniej niż wierzchołek u w algorytmie BFS. Jeżeli natomiast wierzchołek nie należy do warstwy jednostkowej to istnieją co najmniej dwie krawędzie dolne tego wierzchołka, mające różne kolory. Niech tym wierzchołkiem będzie u natomiast jego krawędziami dolnymi uv oraz uw. Wówczas $u_i = max(v_i, w_i)$ dla $1 \leqslant i \leqslant k$.

4 Etykietowanie produktu kartezjańskiego

W tym rozdziale rozszerzymy definicję kolorowania i nazwiemy ją etykietowaniem. Wszystkie krawędzie, rzutowane na tę samą krawędź otrzymają tę samą etykietę. Pokażemy, że produkt może być poetykietowany, a co za tym idzie pokolorowany w czasie liniowym. Etykietowanie pozwoli nam na określenie pozycji danej krawędzi w stosunku do faktoryzacji iloczynu kartezjańskiego grafu w czasie stałym. W rozdziale tym opiszemy struktury danych niezbędne do etykietowania krawędzi. Właściwie etykietowanie okaże się pozycją krawędzi w tablicy krawędzi wychodzących z danego wierzchołka o tym samym kolorze.

4.1 Struktury Danych

Zauważmy, że używamy współrzędnych wierzchołka do określenia jego pozycji w produkcie iloczynu kartezjańskiego. Całkowita długość tych wektorów jest równa $O(m)$. Przez pozycję krawędzi uv rozumiemy pozycję wierzchołka u, kolor krawędzi uv oraz rzutowanie $p_i(uv)$ czyli $p_i(u)p_i(v)$ czyli bazę krawędzi uv.
Krawędź uv ma ten sam kolor co krawędź $p_i(u)p_i(v)$ oraz tak samo jest krawędzią dolną, poprzeczną lub górną. Poniżej przedstawimy jak efektywnie przetrzymywać informację o bazie.
W poprzednich rozdziałach opisywaliśmy w jaki sposób dzielimy krawędzie incydentne z danym wierzchołkiem na krawędzie dolne, poprzeczne i górne. Następnie każdą taką listę można podzielić na krawędzie tego samego koloru. Pozycja krawędzi, w tak stworzonej monochromatycznej liście, posłuży nam do zlokalizowania $p_i(uv)$, będziemy to nazywać numerem danej krawędzi o oznaczać $n(uv)$. W ogólnym przypadku $n(uv) \neq n(vu)$. Parę $< c(uv), n(uv) >$ będziemy nazywać etykietą uv i oznaczać $l(uv)$. Razem z $p_{c(uv)}(u)$ będzie to opisywać pozycję krawędzi w produkcie.
Ilość tablic monochromatycznych będzie równa co najwyżej $3d(v_0)$ dla każdego wierzchołka (po $d(v_0)$ dla krawędzi dolnych, poprzeczych i górnych), tak więc dostęp do tych tablic może odbyć się w czasie stałym.
Tak więc etykietowanie zostało zdefiniowane. Jest ono zależne od kolejności krawędzi należących do warstw jednostkowych oraz krawędzi górnych należących do wierzchołków z warst jednostkowych. Kolejność pozostałych krawędzi w listach monochromatycznych nie będzie miała wpływu na etykietowanie. Głównym celem naszego algorytmu będzie właśnie etykietowanie tychże krawędzi i zmiana numeru danej krawędzi w liście.
W algorytmie dobrze uwzględnić że krawędź dolna, poprzeczna lub górna uv ma początek w u natomiast koniec w v. Pozwoli to na znalezienie tej krawędzi w liście monochromatycznej w czasie stałym, jeżeli znamy jej numer. Zmodyfikujemy rów-

nież macierz sąsiedztwa tak, aby w komórce uv mieć numer danej krawędzi właśnie w celu znalezienia jej numeru w czasie stałym.
Tak więc podsumowując w naszym algorytmie będziemy używać następujących struktur danych: dla każdego wierzchołka lista sąsiedztwa, zmodyfikowaną listę krawędzi oraz zmodyfikowaną tablicę sąsiedztwa. Dodatkowo każdy wierzchołek zostanie ułożony w tablicy zgodnie z kolejnością algorytmu BFS, będzie on posiadał numer swojego poziomu, wektor współrzędnych (o długości nie większej niż $d(v_0)$) oraz listę krawędzi dolnych, poprzecznych oraz górnych, które następnie będą dzielone na listy monochromatyczne. Budowa tychże struktur, z wyłączeniem list monochromatycznych jest możliwa w czasie $O(m)$. Pokażemy, że i te tablice można zbudować w takim czasie. W następnym rozdziale pokażemy również, że tablica sąsiedztwa może być zbudowana z użyciem $O(m)$ pamięciowym.

4.2 Algorytm Etykietowania

Twierdzenie Niech G=$G_1\Box G_2\Box\ldots\Box G_k$ *będzie grafem spójnym. Dane jest również kolorowanie właściwe produktu względem podanego rozkładu dla krawędzi wychodzących z wierzchołka* v_0. *Etykietowanie tego grafu może być zrobione w czasie O(m)*

Dowód
Opiszemy liniowy algorytm etykietowania. Poetykietowanie krawędzi wychodzących z v_0 nie stanowi problemu. Następnie etykietujemy krawędzie dolne i poprzeczne wierzchołków należących do warstwy L_1. To również nie stanowi problemu ponieważ wszystkie krawędzie dolne dostają numer 1 natomiast krawędzie poprzeczne kolorujemy tak jak w rozdziale 3, numery tych krawędzi są zgodne z kolejnością w jakiej zostały podane na wejściu.
Zakładamy, że mamy poetykietowane krawędzie dolne i poprzeczne dla warstwy L_i i indukcyjnie etykietujemy krawędzie dolne i poprzeczne warstwy L_{i+1}. Niech wierzchołek u należy do warstwy L_{i+1}. Rozważamy następujące przypadki:

1. Wierzchołek ma tylko jedną krawędź dolną. Wówczas kolorujemy tak, jak to było w rozdziale 3.1 natomiast krawędź otrzymuje numer 1.

2. Szukamy kwadratu bazowego dla wierzchołków posiadających więcej niż jedną krawędź dolną.

   a) Niech uv będzie pierwszą taką krawędzią. Natomiast niech vx będzie krawędzią dolną wierzchołka v. Szukamy wspólnego sąsiada wierzchołków

u oraz x różnego oczywiście od wierzchołka v.
Jeżeli wspólny sąsiad nie istnieje, wówczas $c(uv) := c(vx)$. Pozostałe krawędzie dolne również kolorujemy tym samym kolorem. Oznacza to, że wierzchołek u należy do warstwy jednostkowej, a co za tym idzie możemy ponumerować krawędzie zgodnie z tym, w jaki sposób podano je na wejściu.

b) Tak więc rozważmy przypadek, że istnieje wierzchołek w będący wspólnym sąsiadem wierzchołków u oraz x. Jeżeli kolory krawędzi vx oraz vw są różne wówczas $l(uv) := l(wx)$ oraz $l(uw) := l(vx)$. Znalezezienie pozycji TO DOOOOOOOOOOOOOOOOO

c) Teraz rozważmy sytuację, w której znalezlismy wspólnego sąsiada, którym jest wierzchołek w, jednakże tym razem kolory krawędzi wx oraz wx są takie same. Wówczas uv oraz uv również mają ten sam kolor. Jeżeli wszystkie pozostałe krawędzie dolne wierzchołka v mają ten sam kolor wówczas wierzchołek u należy do warstwy jednostkowej i postępujemy analogcznie jak w przypadku 2a. Tak więc niech istnieje krawędź vx', której kolor różni się od koloru vx a co za tym idzie również od uv. Tak więc musi istnieć wspólny sąsiad wierzchołków u oraz x' różny od v, nazwijmy go w'. Znalezienie takiego sąsiada może być wykonane w czasie $O(d(u))$. Dalej postępujemy jak w przypadku 2b a naszym kawdratem bazowym będzie $uvx'w'$.

3. Etykietowanie pozostałych krawędzi dolnych odbywa się z udziałem znalezionego wczesniej kwadratu bazowego. Załóżmy dla jasnosci sytuacji, że kwadrat ten składa się z wierzchołków u, v, x oraz w. Tak więc iterujemy po wszystkich krawędziach dolnych wierzchołka u. Niech uy będzie kolejną taką krawędzią. Dalej niech yz będzie dolną krawędzią wierzchołka y, taką, że $l(yz) = l(uv)$. Jeżeli okaże się że taka krawędź nie istnieje wówczas suzkamy krawędzi yz takiej, że $l(yz) = l(uw)$. Dla każdego wierzchołka taka operacja jest wykonywana w czasie stałym, a dla wszystkich krawędzi sumarycznie czas potrzebny do poetykietowania krawędzi dolych jest równy $O(d(u))$. Tak więc etykietowanie wszystkich krawędzi dolych należących do warstwy L_{i+1} może być wykonane w czasie $O(\sum_{u \in L_{i+1}} d(u))$.
Etykietowanie krawędzi górnych oraz poprzecznych przebiega w analogiczny sposób, co również implikuje taką samą złożonosć całej operacji.

4.3 Etykietowanie produktu w czasie liniowym

Dotychczas do sprawdzeia czy dwa wierzchołki są połączone używalismy macierzy sądsiedztwa, co wymagało od nas użycia n^2 pamięci, gdzie n oznacza liczbę wierzchołków. Zaburza to nasze założenie o liniowej złożonosci pamięciowej naszego algorytmu, co więcej inicjalizacja tej macierzy również może zaburzyć naszą złożoność czasową. Na szczęcie w każdym kroku naszego algorytmu potrzebujemy co najwyżej jednego wiersza z naszej tabeli sąsiedztwa, co powoduje, że nasza założona złożoność pamięciowa w łatwy sposób może być osiągnięta. Można również inicjalizację takiej macierzy sąsiedztwa wykonać w czasie $O(m)$ zamiast $O(n^2)$ (TUTAJ ALBO ODNIESIENIE DO ARTYKUŁU ALBO OPIS!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!). Wykorzystując tę metodę, do tworzenia wiersza tablicy sąsiedztwa, zauważmy na początek, że czas potrzebny do stworzenia takiego wiersza dla wierzchołka x jest proporcjonalny do $d(x)$. A ponieważ $\sum_{v \in V(G)} d(v) = 2m$ to złożoność czasowa potrzebna na wykonanie się algorytmu nie zmieni się, dopóki nie będziemy wykonywać naszej operacji więcej niż stałą ilosć razy dla poszczególnych wierzchołków. Tak więc w tym rozdzialne opiszemy jak wykonać nasz algorytm tak, aby nasze założenie zostało wykonane, co pozwoli nam na utrzymanie złożonosci czasowej, przy jednoczesnym zmniejszeniu złożonosci pamięciowej.

Twierdzenie *Niech* G=$G_1 \square G_2 \square \ldots \square G_k$ *będzie grafem spójnym. Dane jest również kolorowanie właściwe produktu względem podanego rozkładu dla krawędzi wychodzących z wierzchołka* v_0*. Złożoność czasowa i pamięciowa tego algorytmu wynosi O(m).*

Dowód

W rozdziale 4.2 opisalismy liniowy, jeżeli chodzi o złożoność czasową algorytm, tutaj opiszemy jak go zmodyfikować aby i złożoność pamięciowa była liniowa. Opiszemy modyfikację trzech kroków z poprzedniego rozdziału.

1. Jeżeli chodzi o etykietowanie krawędzi dolnych wierzchołków posiadajacych jedną krawędź dolną procedura się nie zmienia.

2. Poszukiwanie kwadrtaru bazowego. Możliwe, że będą konieczne dwa przebiegi poszukiwań naszego kwadratu bazowego. W pierwszym przebiegu wierzchołek v jest dolnym wierzchołkiem u, gdzie $u \in L_{i+1}$ natomiast x jest dolnym wierzchołkiem v. Teraz szukamy pozostałych wierzchołków u' należących do warstwy L_{i+1}, dla których odległość od wierzchołka x również wynosi 2. Dopiero wtedy będziemy tworzyć wiersz macierzy sąsiedztwa wierzchołka x. W drugim przebiegu rozumowanie jest analogiczne.

3. W tym fragmencie konieczne będzie tworzenie wiersza macierzy sąsiedztwa maksymalnie dwa razy, tym razem jednak dla wierzchołków należących do warstwy L_i, a nie jak to było w punkcie 2 dla warstwy L_{i-1}.

Dla krawędzi górnych i poprzeczych procedura wygląda analogicznie (możliwe, że konieczne będą po dwa przebiegi kroku algorytmu). Tak więc podczas indukcyjnego kroku dla warstwy L_{i+1} maksymalnie 4 razy szukamy wiersza macierzy sąsiedztwa dla wierzchołków warstw L_i oraz L_{i-1} co oznacza, że dla danego wierzchołka, jego wiersz z macierzy sąsiedztwa będzie wyszukiwany nie więcej niż 8 razy, co należało dowieźć.

Ponieważ etykietowanie jest rozszerzeniem kolorowania, nadanie współrzędnym wierzchołkom grafu G może być wykonane przez algorytm opisany w rozdziale 3.2.

5 Sprawdzanie spójnosci

Poprzednie rozdziały opierały się na założeniom, że graf podany na wejsciu jest produktem iloczynu kartezjańskiego oraz znamy kolory krawędzi incydentych z pewnym wierzchołkiem. Odwróćmy teraz trochę sytuację i załóżmy, że mamy dane kolory pewnych krawędzi wychodzących z danego wierzchołka a naszym zadaniem będzie sprawdzenie, czy to algorytm etykietowania spowoduje, że nasze kolorowanie będzie kolorowaniem właciwym produktu iloczynu kartezjańskiego. Może się okazać, że sprawdzenie koloru krawędzi dla sąsiednich wierzchołków lub przykładowo brak kwadratu bazowego dla pewnego wierzchołka da nam negatywną odpowiedź na nasze pytanie. Jednakże, jak się okaże za chwilę, możliwe jest kontynuowanie procedury etykietowania, nawet w przypadku, gdy podczas szukania kwadratu bazowego brakuje nam krawędzi lub wierzchołka.
Tak więc potrzebujemy efektywnego narzędzia do sprawdzania spójnosci założeń. Do tego celu wykorzystywać będziemy lemat o izomorfiźmie.
Izomorfizm grafów jest bijekcją na zbiorach wierzchołków zachowującą sąsiedztwo. Korzystając z kolejnosci wierzchołków grafu uzyskaną w algorytmie BFS, będziemy indukcyjnie sprawdzać czy dla danego poziomu L_i nasze założenia są spełnione i dalej kontunować sprawdzenie dla następnych poziomów.

Twierdzenie *Zalóżmy, że własnosci izomorfizmu są zachowane dla wierzchołków warstw L_1 do L_i. Sprawdzenie czy własnosci izomorfizmu dla wierzchołków należących do warstwy L_{i+1} może być wykonane w czasie proporcjonalnym do sumy ilosci krawędzi dolych i poprzeczych warstwy L_{i+1} oraz krawędzi górnych L_i.*

Dowód
Nasze sprawdzenie będzie się odbywało w dwóch krokach.

1. Dla każdego wierzchołka u należącegodo L_{i+1}, który nie jest wierzchołkiem warstwy jednostkowej wybieramy dwie krawędzie dolne różnych kolorów uv oraz uw (można wybrać krawędzie należące do kwadratu bazowego) i sprawdzamy czy krawędzie dolne/poprzeczne wierzchołka u mają odpowienika wsród krawędzi dolnych/poprzeczych wierzchołka v. Uscislając, dla każdej krawędzi dolnej/poprzecznej uz koloru różnego od $c(uv)$ istnieje dokładnie jedna krawędź dolna/poprzeczna vz' taka, że $l(uz) = l(vz')$ i vice versa. Co więcej $l(uv) = l(zz')$.

2. Dla krawędzi górnych należących do L_i postępujemy analogicznie. Iterujemy po wszystkich wierzchołkach. Jeżeli u nie należy do warstwy jednostkowej wybieramy kwadrat bazowy $uvxw$ i dla wszystkich krawędzi górnych wierzchołka u porównujemy je z krawędziami uv lub uw w zależnosci od koloru.

Jeżeli u należy do warstwy jednostkowej, załóżmy $G_i^{v_0}$, wybieramy jeden wierzchołek v, będący dolnym sąsiadem u. Porównujemy wówczas krawędzie górne wierzchołków u oraz v o kolorach różnych od i.

Ponieważ sprawdzenie, czy istnieje odpowiadająca krawędź, odkąd mamy posortowane listy monochromatyczne, może być wynonane w czasie stałym, czas potrzebny do znalezenia odpowiadających krawędzi jest dla danego wierzchołka u jest proporcjonalny do $d(u)$ a co za tym idzie sumaryczna ilosć czasu jest proporcjonalna do rozmiaru grafu. Żadne dodatkowe struktury danych nie są potrzebne.
Tak więc sprawdzanie izomorfizmów między warstwami może być wykonane w zadowalającej nas złożonosci czasowej i pamięciowej. Aby wykazać, że jest to produkt iloczynu kartezjańskiego potrzeba jeszcze sprawdzić pozostałe założenia lematu o izomorfizmie. Rozważmy wierzchołek a będący dolnym sąsiadem u, takim że $c(ua) \neq c(uv)$. Będziemy teraz rozważać izomorfizm między G_i^u oraz G_i^a indukowany między krawędziami między nimi występującymi dla $i \neq c(uv)$.
Naszym zadaniem jest pokazanie, że dla każdej krawędzi dolnej lub poprzecznej ab, takiej że $c(ab) \neq c(ua)$ istnieje dokładnie jedna krawędź uc, taka że $l(uc) = l(ab)$ oraz wierzchołki b i c są połączone oraz $l(ua) = l(bc)$. Oczywiscie do tego musimy pokazać również że dla każdej dolnej lub poprzecznej krawędzi uc istnieje krawędź ab spełniająca te same założenia.
Rozważmy pierwszy przypadek. We wczeniejszych rozważaniach w tym rozdziale wykazalismy istnienie krawędzi aa', takiej że $l(uv) = l(aa')$. Z założenia idnukcyjnego wiemy, że izomormizmy są zachowane dla poziomów od L_1 do L_i. Skorzystamy teraz ponownie z lamatu o kwadracie. Rozważając wierzchołki a,b,a' w stałum czasie możemy znaleźć kwadrat $abb'a'$. Idąc dalej rozważamy wierzchołki v,a',b' i również w czasie stałym możemy znaleźć kwadrat $va'b'c'$. Na koniec wystarczy rozważyć wierzchołki $b,b'c'$ i tak jak poprzednio w czasie stałym znajdujemy kwadrat $bb'c'c$. Krawędź vc' posiada taką samą etykietę jak ab oraz cc' posiada taką samą etykietę jak uv. A ponieważ izomorfizm między wierzchołkami v oraz u został sprawdzony, wierzchołki u oraz c są połączone oraz etkykiety uc i vc' są takie same, a co za tym idzie również etykieta ab jest taka sama, udowodnilismy zakładaną tezę. Dowód w drugą stronę przebiega analogicznie.
Pozostaje jeszcze sprawdzenie czy złożonosć czasowa i pamięciowa nie uległy pogorszeniu. Zaczynając od złożonosci pamięciowej, widzimy, że nie używamy żadnych dodatkowych, złożoych struktur danych. Co do złożonosci czasowej, zauważmy że sprawdzenie izomorfizmu dla danej krawędzi odbywa się w czasie stałym. Szukanie kwadratów opiera się na korzystaniu z już posortowanych monochromatycznych list, więc dostęp do nich odbywa się w czasie stałym. Tak więc sumaryczna złożonosć czasowa sprawdzania spójnosci może być wykonana w czasie proporcjonalnym do rozmiaru grafu.

6 Faktoryzacja poprzez łączenie kolorów

W tym rozdziale opiszemy wszystkie konieczne narzędzia potrzebne do fakotryzacji spójnego gragu G na grafy proste, czyli takie które nie da się opisać jako iloczyn dwóch nietrywialnych grafów (przez graf trywialny rozumiemy graf K_1). Z lematu o udoskonaleniu chcemy znaleźć końcowy rozkład podanego na wejciu grafu G. Przez znalezienie rozkładu rozumiemy znalezienie finalnego kolorowania własciwego produtu iloczynu kartezjańskiego. Ma to być kolorowanie zachowujące izomorfizmy między warstwami tego samego koloru.

Twierdzenie *Faktoryzacja prosta grafu spójnego może być wykonana w czasie liiowym i w takiej samej złożonosci pamięciowej*

Dowód: Jako dowód zaprezentujemy algorytm. Ideą naszego rozmumowania będzie rozpoczęcie algorytmu ze wskazanym kolorowaniem początkowym krawędzi wychodzących z wierzchołka o minimalnym stopniu. Kolorowanie to nie będzie jeszcze tym, które będzie wskazywać na kolorwanie własciwe finalnego produktu kartezjańskiego, lecz w trakcie działania algorymu będziemy łączyć dwa kolry w jeden jeżeli zajdzie taka konieczność.
Nasz algorytm w pierwszym kroku nadaje różne kolory wszystkim krawędzią wychodzącym z wierzchołka v_0, który to jest wierzhołkiem o minimalym stopniu w naszym graie wejsciowym. Ponieważ każdy wierzchołek posiada krawędzie incydentne każdego koloru wiemy, że w naszym kolorowaniu finalnym nie będzie więcej kolorów. Użycie wierzchołka o minimalnym stropniu sprawia również, że nasza początkowa tablica kolorów ma możliwie najmniejszy rozmiar, a co bardziej istotne jej wielkosć jest stałą.
Na początek uruchamiamy algorytm BFS z wierzchołkiem początkowym v_0, dzielimy wierzchołki na warstwy a krawędzie na krawędzie dolne, poprzeczne i górne. Następnie dla każdej warstwy wykonujemy etykietowanie a następnie sprawdzamy spójnosć. Może się okazać, że nasza procedura wykona się bez błędów i w taki oto sposób otrzymamy nasze kolorawnie końcowe produktu iloczynu kartezjańskiego, a co za tym idzie rozkład naszego grafu wyjciowego na iloczyn grafów prostych.
Bardziej prawdoodobne jest jednak, że etykietowanie lub sprawdzanie spójnosci nie wykona się poprawnie. Dla przykładu, będziemy kolorować krawędzie poprzeczne warstwy L_1, widzimy wówczas, że nie jestesmy w stanie pokolorwać w takiej krawędzi (patrz lemat o trójkącie). Oznacza to po prostu, że nasze kolorwanie początkowe krawędzi wychodzących z v_0 jest niewłasciwe, użylimy zbyt dużej ilosci kolorów. Wówczas procedura jest prosta- łączymy kolory krawędzi incydentych z naszą krawędzią poprzeczną w jeden i wówczas możemy ją już pokolorować bez problemu. Cała idea naszego algorytmu opiera się na tym koncepie: poetykietuj krawędzie i sprawdź spójnosć- jeżeli jest to niemożliwe- połącz kolory.

Jak się okazuje nie ma koniecznosci rekolorowania krawędzi jeżeli już zostały pomalowane wczesniej, wystarczy tylko zanotować fakt, że dany kolor został połączony z innym. Ponieważ $d(v_0)^2 \leqslant d(v_0)n \leqslant 2m$ nie ma koniecznosci przetrzymywać danych o kolororach w żadnych skomplikowanych strukturach danych. Dodatkowo wiemy, że w finalnym kolorowniu użyty będzie co najmniej jeden kolor, to operacji łączenia kolorów będzie również co najwyżej $d(v_0)$ co również nie zaburzy złożonosci czasowej naszego algorytmu.

7 Implementacja Algorytmu

7.1 Wprowadzenie

7.2 Dane wejsciowe

7.3 Opis pakietów i ważniejszych klas

7.4 Opis algorytm faktoryzacji