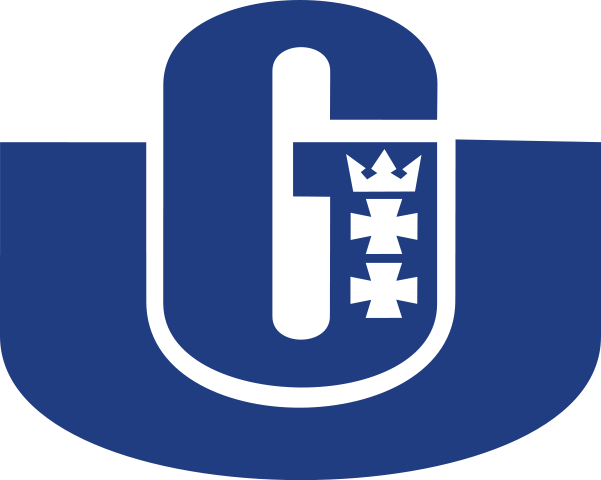
**UNIWERSYTET GDAŃSKI**

**WYDZIAŁ MATEMATYKI, FIZYKI I INFORMATYKI**



**Szymon Rękawek**

**Numer indeksu 206288**

**Gra Thuego**

**SPIS TREŚCI**

**WSTĘP**

**ROZDZIAŁ I Teorie Axela Thue na temat sekwencji symboli oraz zasady gry**

1.1. Definicje

1.2. Thue-Morse sequence

1.3. Square-free words

1.4. Zasady gry

**ROZDZIAŁ II Algorytmy obsługujące logikę gry oraz komputerowych przeciwników**

2.1. Algorytm do wyszukiwania powtórzeń kwadratowych

**ROZDZIAŁ III Podsumowanie efektywności algorytmów**

3.1. (…)

**ZAKOŃCZENIE**

**WSTĘP**

Tematem niniejszej pracy jest Gra Thuego. Axel Thue był norweskim matematykiem żyjącym w latach 1863 - 1922, znanym z prac z zakresu kombinatoryki.

Thue pracował nad problemami, powstałymi w wyniku badań nad sekwencjami symboli. Praca Thue [1] opisywała problem, który autor nazwał 'nieredukowalne słowa' ('irreducible words'). Poświęca w niej szczególną uwagę dwu i trzy literowym przypadkom. W skrócie wprowadza pojęcie znane obecnie jako *Thue-Morse sequence* i pokazuje, że nieskończone słowa *Overlap-Free* są pochodnymi tej sekwencji. W swoich pracach definiuje kolejną strukturę, a mianowicie *infinite Square-Free word*, oraz przedstawia sposoby generowania nieskończenie długich słów wolnych zarówno od kwadratów jak i nasunięć.

Gra która powstała na podstawie teorii Thuego w skrócie polegała będzie na utworzeniu jak najdłuższego ciągu znaków nad określonym z góry alfabetem. Zależnie od trybu gry, kończyć się ona będzie w momencie gdy pojawi się zdefiniowany na początku rodzaj powtórzenia w tworzonym przez nas, bądź algorytm ciągu. Jednym z trybów gry jest walka komputera przeciwko niemu samemu, po takiej rozgrywce przedstawione zostaną złożoności czasowe oraz wnioski wynikające z obranej przez oponentów taktyki.

Ostatnia część pracy poświęcona jest analizie algorytmów zarówno pod względem czasu ich wykonywania jak i zdolności do przewidywania ruchów przeciwnika.

**Rozdział I**

* 1. **Definicje**

Alfabet jest skończonym zbiorem symboli lub liter.

Słowo alfabetu **A** jest skończoną sekwencją elementów z **A**.

Długość słowa **ω** jest reprezentowana przez **|ω|.**

Puste słowo o długości 0 jest reprezentowane przez **ε**.

**Factor**(*czynnik*) słowa **ω** jest słowem ***u***, które występuje wewnątrz **ω** formie **ω = *xuy***, podczas gdy ***x*** oraz ***y*** również są słowami tego alfabetu.

**Square**(*kwadrat*) jest niepustym słowem w formie ***uu***.

***Square-free***, słowo jest wolne od kwadratów, jeśli żaden z jego ***czynników*** nie jest ***kwadratem***.

**Overlap**(*nasunięcie*) jest słowem w formie ***xuxux***, gdzie **x** jest niepusty. Nazwa pojęcia wzięła się z tego, że ***xux*** występuje dwa razy w ***xuxux***. Pierwszy raz jako prefiks(początkowy czynnik) oraz jako sufiks(końcowy czynnik) i te dwa wystąpienia mają wspólną część -centralne ***x***, a więc nasuwają się na siebie.

**Overlap-free** - słowo w którym żaden z czynników nie nasuwa się na siebie.

W definicji Axela Thue słowo **ω** w alfabecie długości **n** jest nieredukowalne jeśli jakiekolwiek dwa wystąpienia tego samego słowa jako czynnik wewnątrz **ω** są zawsze oddzielone od siebie przez n-2 liter. Oznacza to, że nieredukowalne dwuliterowe słowo jest ***bez nasunięć*** i nieredukowalne trzyliterowe słowo jest ***bez kwadratów***.

**Morfizm** - mapowanie obiektu z jednej matematycznej struktury w inną.

* 1. **Thue-Morse sequence**

W tej sekcji zostaną omówione podstawowe pojęcia związane z ***Thue-Morse sequence.***

Niech A = {a, b} będzie dwuliterowym alfabetem. Rozważmy morfizm µ z monoidu A\*, który definiuje się następująco:

µ(a) = ab, µ(b) = ba

Dla n >=0:

un = µn(a), vn = µn(b)

Wtedy:

u0 = a v0 = b

u1 = ab v1 = ba

u2 = abba v2 = baab

u3 = abbabaab v3 = baababba

Wzór ogólny:

u n+1 = un vn, vn+1 = vn un

oraz:

un = n, vn = n

gdzie jest uzyskiwane z ω przez zamianę 'a' oraz 'b'. Słowa un i vn są często nazywane 'Morse block' blokami morsa. Można łatwo zauważyc że u2n oraz v2n są palindromami oraz to że u2n+1 = ~v2n+1, gdzie ~w jest negacją w. Morfizm µ może być rozszerzony do nieskończonych słow, które mają dwa stałe punkty:

t = abbabaabbaababbabaab... = µ(t)

~t = baababbaabbabaababba... = µ(~t)

Przedstawione powyżej słowo t jest sekwencją Thue-Morse'a. Jest wiele innych sposobów na stworzenie tego słowa. Niech tn będzie n-tym symbolem w t, zaczynając od n = 0. Wtedy można pokazać, że:

tn =

gdzie d1(n) jest liczbą bitów równych 1 w binarnej reprezentacji n. Dla n <= 12 oraz n generowane jest następujące słowo:

bin(0) = 0, d1(0) = 0 mod 2 = 0 = *a*

bin(1) = 1, d1 (1) = 1 mod 2 = 1 = *b*

bin(2) = 10, d1 (2) = 1 mod 2 = 1 = *b*

bin(3) = 11, d1 (3) = 2 mod 2 = 0 = *a*

bin(4) = 100, d1 (4) = 1 mod 2 = 1 = *b*

bin(5) = 101, d1 (5) = 2 mod 2 = 0 = *a*

bin(6) = 110, d1 (6) = 2 mod 2 = 0 = *a*

bin(7) = 111, d1 (7) = 3 mod 2 = 1 = *b*

bin(8) = 1000, d1 (8) = 1 mod 2 = 1 = *b*

bin(9) = 1001, d1 (9) = 2 mod 2 = 0 = *a*

bin(10) = 1010, d1 (10) = 2 mod 2 = 0 = *a*

bin(11) = 1011, d1 (11) = 3 mod 2 = 1 = *b*

bin(12) = 1100, d1 (12) = 2 mod 2 = 0 = *a*

t = *abbabaabbaaba*

W konsekwencji istnieje skończony automat obliczający wartości tn. Automat ten ma dwa stany końcowe 0 oraz 1. Na początku czyta łańcuch znaków bin(n) od lewej do prawej, zaczynając od n = 0. Ostateczny stan równy jest 0 lub 1 i definiuje czy tn jest równe a lub b. W skrócie obliczenie jakie wykonuje automat to d1(n) modulo 2.

**1.3. Square-free words**

Łatwo można zauważyc, że jedynymi słowami bez kwadratów w alfabecie A = {a, b} są: *a*, *b*, *ab*, *ba*, *aba*, *bab*. Istnieje jednak dowolnie długi ciąg znaków wolny od kwadratów dla słów nad alfabetem trzyliterowym. By stworzyć dowolne słowo wolne od kwadratów Thue wymyślił następujący algorytm.

Mając alfabet A = {a, b, c} należy zastąpić każde wystąpienie litery *a* przez *abac*, *b* przez *babc* oraz *c* przez *bcac*, jeśli jest poprzedzone przez *a* lub *acbc,* jeśli jest poprzedzone przez *b*. Zaczynając od litery *a* otrzymujemy nieskończone słowo które nie zawiera kwadratów.

*abacbabcabacbcacbabcabacbabcacbcabacbabc*...

W 1912 roku Axel Thue wymyślił inny sposób na generowanie nieskończonego słowa bez kwadratów na trzech literach z użyciem następującego morfizmu.

*a* -> *abcab*

*b* -> *acabcb*

*c* -> *acbcacb*

Po raz kolejny zastępujemy każde wystąpienie z naszych liter przez zdefiniowane sekwencje. Jest to dość skomplikowana struktura, zsumowana długość łańcuchów wynosi 18. A Carpi [7] dowiódł, że morfizm na alfabecie składającym się z trzech liter tworzący słowa wolne od kwadratów musi mieć długość równą co najmniej 18.

* 1. **Zasady gry**

W grze dostępnych jest kilka trybów zarówno dla jednego oraz dwóch graczy jak i komputerowa symulacja, czyli gra doskonała komputera przeciwko niemu samemu.

Zasady trybu ***Longest Square-Free word***, dla dwóch graczy są następujące.

Na początku gry gracze ustalają moc zbioru kolorów, który musi mieć co najmniej 3 elementy oraz otrzymują swoje role, jeden z nich staje się architektem, drugi malarzem. Rola architekta polega na wybieraniu odpowiedniego indeksu w ciągu tworzonym przez graczy, pod którym powstanie nowy element. Malarz natomiast określa kolor wstawianego elementu. Gra kończy się w momencie, gdy w ciągu tworzonym przez graczy pojawia się ***kwadrat***.

Indeks *i* podawany przez architekta nie może być mniejszy od zera oraz większy niż *n*, gdzie *n* jest równe liczbie elementów w ciągu.

Grę rozpoczyna architekt, podany przez niego indeks w pierwszym ruchu musi wynosić 0, ponieważ *n* = 0. Gracze wykonują swoje ruchy na przemian, malarz otrzymuje punkt za każdy pomalowany element, który nie tworzy ***kwadratu*** wewnątrz ciągu.

By wyłonić zwycięzcę potrzebne są dwie rundy, by każdy z graczy sprawdził się zarówno jako malarz i architekt. Wygrywa osoba, która zdobyła więcej punktów jako malarz.

Tryb ten dostępny jest również dla jednego gracza, rolę przeciwnika otrzymuje wtedy komputer, który działa według algorytmu przewidującego określoną przez poziom trudności liczbę ruchów do przodu. Algorytm może pełnić zarówno rolę budowniczego jak i malarza.

Bliźniaczym trybem gry opierającym się na tych samych zasadach z niewielką różnicą jest ***Longest Overlap-Free word***.

Różnica polega na tym, że malarz w tworzonym ciągu musi unikać ***nasunięcia*** oraz moc zbioru kolorów musi mieć co najmniej 2 elementy.

Podobnie jak w ***Longest Square-Free word*** jest możliwość gry przeciwko algorytmowi, który jest w stanie przewidywać określoną ilość ruchów do przodu.

**Rozdział II**

**2.1 Algorytm do wyszukiwania powtórzeń kwadratowych**

List<Integer> **findRepetition**(List<Integer> sequence) {

List<Integer> repeatedSequence = null;

int maxSeqSize = sequence.size()/2;

int minSeqSize = 1;

for(int subSeqSize=minSeqSize; subSeqSize<=maxSeqSize; subSeqSize++) {

repeatedSequence = compareSubSequences(subSeqSize);

if(repeatedSequence != null) {

return repeatedSequence;

}

}

return null;

}

List<Integer> **compareSubSequences**(int subSeqSize) {

List<Integer> left = new ArrayList<>();

List<Integer> right = new ArrayList<>();

int comparesFitInSequence = (sequence.size()-subSeqSize\*2) + 1;

for(int i=0; i<comparesFitInSequence; i++) {

for(int j =0;j<subSeqSize;j++) {

left.add(sequence.get(i+j));

right.add(sequence.get(i+j+subSeqSize));

}

if(listsAreEqual(left, right)) {

return left;

}

left.clear();

right.clear();

}

return null;

}

**Bibliografia**

[1] A. Thue, Uber die gegenseitige Lage gleic  her Teile gewisser Zeichenreihen,

Kra. Vidensk. Selsk. Skrifter. I. Mat.-Nat. Kl., Christiana 1912, Nr. 10.

[2] A. Carpi, On the size of a squarefree morphism on a three letter alphabet,

Inform. Proc. Letters 16 (1983), 231.