



Budapest University of Technology and Economics  
Faculty of Electrical Engineering and Informatics  
Department of Computer Science and Information Theory

# Simulation of quantum walks on a classical computer

**Scientific Students' Association Report**

Author:

Viktória Nemkin

Advisor:

dr. Katalin Friedl

2021

# Contents

<b>Kivonat</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Classical random walks on graphs . . . . .	1
1.2 From classical to quantum . . . . .	2
1.3 Applications . . . . .	2
<b>2 Classical random walks</b>	<b>3</b>
<b>3 Quantum informatics, quantum computing</b>	<b>5</b>
3.1 The postulates of quantum mechanics . . . . .	5
3.1.1 Identities of tensor products . . . . .	7
<b>4 Quantum walk</b>	<b>8</b>
4.1 Definition of the quantum coin . . . . .	8
4.2 Quantum walks on the line . . . . .	10
4.3 Quantum walks on larger dimensional grids . . . . .	13
4.4 Példa . . . . .	14
4.5 Quantum walks on Class 1 regular graphs . . . . .	17
4.6 Quantum walks on Class 2 or non-regular graphs . . . . .	19
4.6.1 Class 2 . . . . .	19
4.6.2 Make the graph regular . . . . .	19
4.6.3 Szegedy walk . . . . .	19
<b>5 Szimulátor szoftver</b>	<b>20</b>
5.1 Tervezési megfontolások . . . . .	20
<b>6 Architektúra</b>	<b>21</b>
6.1 Gráfmodellek . . . . .	21

6.2	Szimulátorok . . . . .	23
6.3	Futtatás, konfiguráció, eredmények ábragenerátora . . . . .	23
<b>7</b>	<b>Other Quantum Walk Packages</b>	<b>24</b>
7.1	QWalk . . . . .	24
7.2	QwViz . . . . .	24
7.3	PyCTQW . . . . .	25
7.4	Hiperwalk . . . . .	25
7.5	QSWalk . . . . .	25
7.6	QSWalk.jl . . . . .	25
<b>8</b>	<b>Results</b>	<b>26</b>
8.1	Walk on line . . . . .	26
8.2	Walk on Grid . . . . .	26
8.3	Walk on Hypercube . . . . .	26
	<b>Acknowledgements</b>	<b>27</b>
	<b>Bibliography</b>	<b>28</b>
	<b>Appendix</b>	<b>29</b>
A.1	Első függelék . . . . .	29
A.2	Második függelék . . . . .	30

# Notes

■ The following chapter contains the analysis of the TDK dissertation topic, historic background, motivation for the task: why it is important, relevant, useful, etc, the solutions currently available and then compare and contrast with the student's solution. Then, it concludes with the structure of the dissertation, describing the remaining chapters. . . . .	1
■ Itt az lenne a célom hogy eladjam a véletlen sétákat, hogy ezek nagyon fontos algoritmusok. A Google PageRank egy nagyon jó példa. . . . .	1
■ The sequence can also be regarded as a special category of Markov chain ->Homogenous? . . . . .	3
■ Breiman egy könyv, de a Xia cikkben jobban le van írva. . . . .	3
■ Hogy kellene az értékkészletet ( $A$ ) jelölni? . . . . .	3
■ Keep one of the definitions? . . . . .	3
■ Should I define graphs too? . . . . .	4
■ Ez azért jó felvezető, mert a mérés posztulátumnál el tudom mondani az unitér tranfó azonosságait, a tenzoros posztulátumnál pedig a tenzorszorzás azonosságait, amikre később hivatkozni fogok. . . . .	5
■ Vajon szó szerint bemásolni a könyvből nagyon rosszul néz ki, pláne hogy valószínűleg olyan javítja aki olvasta? . . . . .	5
■ TODO: <a href="https://en.wikipedia.org/wiki/Unitary_matrix">https://en.wikipedia.org/wiki/Unitary_matrix</a> vagy kvantumos előadás diasora, citation?. Az ortonormált bázist alkotó sorosat használok az élszínsoportok = érmeoldalok bizonyításnál. . . . .	5
■ Citation needed for these. . . . .	7
■ Need a better name here . . . . .	7
■ A Quanutm diszkrét Fourier transzformáció mátrixa talán ez? . . . . .	10
■ Ez félig az önlabomból van és a Kempe cikkből. . . . .	10
■ Az önlabomat lehet fel kellene töltenem valahova, ha már hivatkozok rá. . . . .	10
■ Itt nem tudom eldönteni, hogy negatív számokat használjak vagy $N/2$ -ből induljon a részecske és ne 0-ból. . . . .	10
■ Cite this? . . . . .	10
■ Ezt valahogy szebben kellene írni. . . . .	11
■ TODO: tenzorszorzás disztributivitását kellene itt szépen behivatkozni. . . . .	12
■ TODO: $ x-1\rangle$ nem mindig megy balra, ezt hogy kellene jelölni 3.18-ra? . . . . .	12

■ Itt be lehetne rakni azokat a nagy mátrix ábrákat az önlabomból, vagy csak hivatkozni, vagy inkább semmit. . . . .	12
■ Behivatkozni hogyan kell? . . . . .	13
■ Ezt angolra kell majd fordítani és leírást készíteni mellé. Vagy kidobni ha túl sok? . . . . .	13
■ Cite these, they are from wikipedia. . . . .	17
■ Ez az a tétel amit nem találtam meg sehol, itt kellene írni hogy én bizonyítottam, de ismert eredmény. . . . .	17
■ Ez igaz? . . . . .	19
■ Add loops (and hope they don't ruin the edge chromatic number?). . . . .	19
■ Walk on the edges. . . . .	19
■ Itt nagyon kiemelni, hogy önlab1-ben csak a klasszikus bolyongás sikerült + az egyenesen a kvantumos, most meg már tudok bármilyen regulárison bolyongani. . . . .	20
■ Mérés összehasonlítás: Mátrix, Spare mátrix, Orákulum grafikon. . . . .	20
■ TODO: <a href="https://en.wikipedia.org/wiki/Strategy_pattern">https://en.wikipedia.org/wiki/Strategy_pattern</a> , de valamilyen Design Patternes könyvből inkább. . . . .	20
■ Ezt az önlabomból hoztam át, maga a keret nem sokat változott, az változott nagyot hogy a kvantumos szimulátor is működik már d reguláris gráfokra. . . . .	21
■ Új: Hiperkocka . . . . .	22
■ Itt fontos átírni, hogy a kvantumszimulátor mostmár d-regulárist is tud! . . . . .	23
■ Itt van pár újdonság, pl. sajátértékek kiírása, stb. . . . .	23
■ Ennek a fejezetnek az lenne a célja, hogy megnéztem sok más programot és egyik se olyan jó mint az enyém, talán kevesebb szöveg kellene és arra fókuszálni hogy miért nem jók ezek, nem hogy pont mit csinálnak, de csak beraktam amit a könyv ír róluk. Általánosan mindegyik bonyolult, C++-os, nem szép tervezési mintájú, vagy csak speciális gráfokra működik (pl. csak grid). . . . .	24
■ Ez ugyanaz: <a href="https://github.com/QWalk/mainline?">https://github.com/QWalk/mainline?</a> . . . . .	24

# Kivonat

Az utóbbi években egyre nagyobb figyelem összpontosul a kvantuminformatikára. Olyan globális vállalatok, mint az IBM, a Google, a Microsoft és az Amazon jelentős összegeket fektetnek kutatásba, hardver- és szoftverfejlesztésekbe ezen a területen, míg az Európai Unió és Magyarország számos olyan programot indított, melyek a kvantuminformatikai kutatások fellendítését célozzák meg.

A jelenlegi kvantumszámítógépekben elérhető qubitek (kvantumbitek) mennyisége még csekély, de sokan úgy vélik hogy a jövőben ez a szám növekedni fog. Az első olyan, a gyakorlatban is hasznos kvantumalgoritmusok, amiket ezeken a processzorokon futtatni tudunk majd, várhatóan azok lesznek, melyek takarékosan bánnak a rendelkezésre álló qubitekkel. A kvantumséta, mely a klasszikus véletlen bolyongás általánosítása kvantumos esetben, pontosan ilyen algoritmus. Mivel a qubit igénye a gráf csúcsszámában logaritmikus, így ez egy érdekes módszernek ígérkezik akár a közeljövőre nézve is. A kvantumséták erejét bizonyítja, hogy a Grover keresés (mely több kvantumalgoritmus alapját képezi) is értelmezhető ezek egy speciális fajtájának.

Dolgozatomban leírom a kvantumséták matematikai alapjait, részletezve a megvalósítás szempontjából fontos pontokat, melyek a szakirodalomban kisebb hangsúllyal szerepelnek. Ismertetem az általam írt szimulátor program architektúráis felépítését és működését, továbbá a futtatott szimulációim eredményeit.

A szimulátor programot Python 3 nyelven írtam, a Stratégia tervezési minta alapján. A szakirodalomban tipikusan használt gráfokat beépítetten támogatja, melyek kombinálásával tetszőlegesen bonyolult reguláris gráf előállítható, ez az előállítás képezi a kvantumséta alapját is. A szoftver reguláris gráfokon történő kvantum és klasszikus séták szimulációját teszi lehetővé, az eredményekről pedig egy részletes report fájlt generál. A kvantumos séták esetében a séta tulajdonságai a valószínűségek generálásához felhasznált érmétől is függnek, melyet többféleképpen is lehet definiálni. A program beépítetten tartalmazza az Hadamard-, a Grover- és a Fourier-érmeket, de felépítéséből adódóan könnyen bővíthető tetszőleges érmevel is.

Szimulációim segítségével összehasonlítottam a klasszikus és a kvantum séták viselkedését, továbbá kimutattam az elméleti szakirodalom alapján elvárt kvantumos jellegzetességeket, az Hadamard-séta ballisztikus természetét és a kvantumséták ciklikus tulajdonságát.

# Abstract

In recent years, there has been an increasing focus on quantum informatics. Influential global companies such as IBM, Google, Microsoft, and Amazon have invested significant amounts into studying and developing hardware and software for this sector, while the European Union and Hungary have launched several programs to accelerate quantum research.

Current technology is yet to produce a significant number of qubits (quantum bits) in a quantum processor, but many believe the amount will increase over the years. The first practical quantum algorithms to be run on these processors are likely to be the ones that use qubits sparingly. Quantum walking, the generalized version of classical random walking, is exactly this kind of algorithm. The number of qubits required to run a quantum walk on a graph is logarithmic in the number of vertices, making it a promising technique for the near future. Furthermore, Grover's search algorithm (a basis for many quantum algorithms) can be viewed as a special case of quantum walks, which illustrates the potential power of this method.

In my dissertation, I present the mathematical framework for quantum walks, detailing the points critical for implementation, which are given less emphasis in the literature. I describe the architecture and capabilities of the simulator program I have written and the conclusions of the simulations I have run.

I developed the software using Python 3, based on the Strategy design pattern. It supports graphs commonly found in the literature while also providing a method for combining them, facilitating experimentation on several kinds of regular graphs. This composition is also the foundation of the quantum walk. It can simulate classical and quantum walks on the same graphs and produce a report file detailing the results. In the quantum case, the characteristics of the walk are also dependent on the type of coin used to generate the probabilities, which can be defined in several ways. The program includes the Hadamard, Grover, and Fourier coins and can easily be extended with others.

Running several simulations, I compared the behavior of classical and quantum walks and demonstrated the quantum characteristics expected from the theoretical literature, the ballistic nature of the Hadamard walk, and the cyclic property of quantum walks.

# Chapter 1

## Introduction

The following chapter contains the analysis of the TDK dissertation topic, historic background, motivation for the task: why it is important, relevant, useful, etc, the solutions currently available and then compare and contrast with the student's solution. Then, it concludes with the structure of the dissertation, describing the remaining chapters.

The following sections are based on [7].

### 1.1 Classical random walks on graphs

Classical random walks are used to describe stochastic processes and many real life sciences rely on these methods. For example stock price movement prediction in finance, natural language processing algorithms, describing brownian motion in engineering physics, and various applications in biology and bioinformatics, such as DNA evolution models, population dynamics, modeling disease outbreaks, epidemic modeling ...etc. There are also algorithms, where stochastic processes are not directly present, but introduced as a way to deal with large quantities of data. Most notably, Google's Page Rank algorithm utilizes classical random walks on the large sized graph of the internet, to score documents on how good of an information source they are or how well they link to other information sources. Other algorithms include Markov chain Monte Carlo, for sampling from a probability distribution, difficult to directly model. Instead, they construct a stochastic process which's equilibrium distribution is the desired one and sample this by repeatedly recording states from the chain. The Metropolis Hastings algorithm is similar to this, and is used to approximate the distribution or compute an integral (e.g expected value).

Itt az lenne a célom hogy eladjam a véletlen sétákat, hogy ezek nagyon fontos algoritmusok. A Google PageRank egy nagyon jó példa.

Random walks are effective in link prediction and recommendation system, computer vision, semi-supervised learning, network embedding, and complex social network analysis. There are also some literature illustrating the applications of random walks on graphs, text analysis, science of science, and knowledge discovery. Random walks can speed up deterministic algorithms by introducing randomization, for example finding specific subgraphs in graphs.

In conclusion, there are many important applications for classical random walks.



## 1.2 From classical to quantum

The scalable quantum computer is a topical issue so that approaches of quantum computation are popular topics nowadays. Quantum walks are the corresponding part of classical random walks in quantum mechanics. The main difference between them is that quantum walks don't converge to some limiting distributions. Due to the quantum interference, quantum walks can spread significantly faster or slower than classical random walks.

There are many properties of quantum walks that make them differ from their classical counterparts. While in classical walks, in  $N$  steps the walks reaches a distance of  $O(\sqrt{N})$ , in quantum, it is a lot faster,  $O(N)$ . Quantum walks have inherent interferences, where some amplitudes can amplify, while others can be destrutive to each other and diminish. This makes them behave very differently from classical random walks and a good reason to study them.

Quantum walks are often used to accelerate classical algorithms. It can be used to decision trees, search problems, and element distinctness.

There are many algorithms based on Quantum Walks. Two types of models are present: continuous time and discrete time walks. Quantum decision tree algorithm is one of the continuous time quantum walk based algorithm. In this algorithm, you systematically explore the decision tree as a graph. The decision tree nodes are quantum states in a Hilbert space.

Based on Quantum Walks

- Quantum Decision Tree
- Quantum PageRank
- Grover Search Algorithm

## 1.3 Applications

- Collaborative Filtering
- Recommender System
- Link Prediction
- Computer Vision
- Semi-supervised
- Learning
- Network Embedding
- Element Distinctness

## Chapter 2

# Classical random walks

A classical random walk describes a stochastic process.

The sequence can also be regarded as a special category of Markov chain -> Homogenous?

Classical random walks on graphs can be defined using Markov-chains. Markov-chains are well explained in [2] and in [7].

Breiman egy könyv, de a Xia cikkben jobban le van írva.

(Információelmélet előadás / a könyvet be lehetne idézni, bár kevésbé szeretik)

**Definition 1. Markov-chain** (First order, discrete-time, discrete-space) A Markov-chain is a sequence of independent random variables from the same distribution,  $X_1, X_2, X_3, \dots$ , (with value set  $A$ ), that have the Markov property:

Hogy kellene az értékkészletet ( $A$ ) jelölni?

$$P(X_k = x_k | X_{k-1} = x_{k-1}, \dots, X_1 = x_1) = P(X_k = x_k | X_{k-1} = x_{k-1})$$

$\forall k \geq 2$  and  $x_1, \dots, x_k$  from the value set.

**Definition 2. Homogenous Markov-chain** Time invariant, i.e.:  $P(\Phi_k = i | \Phi_{k-1} = j) = p_{j,i} \forall k \geq 2, \forall i, j \in A$ , which is called the transition probability from  $i$  to  $j$  and form the transition probability matrix  $P$ .

This allows us to represent Homogenous Markov-chains as directed graphs.

**Definition 3. Distribution of the Markov-chain** at the  $i$ th step, the Markov-chain's distribution is the distribution of  $X_i$ , which is  $P(X_i = j)$ .

**Definition 4. Stationary distribution** of the Markov chain is  $\pi_j = \lim_{i \rightarrow \infty} P(X_i = j)$ , or where  $P\pi = \pi$ .

Keep one of the definitions?

**Definition 5. Graph representation of homogenous Markov-chains** Graph  $G(V, E)$  represents a homogenous Markov-chain  $X_1, X_2, X_3, \dots$ , (with value set  $A$  and transition probability matrix  $P$ ), if  $V = A$  and if  $p_{j,i} \neq 0$ , then  $E(i, j) = p_{j,i} \forall i, j \in A$ . If  $p_{j,i} = 0$ , then there is no edge between vertex  $i$  and  $j$ .

**Definition 6. Random walk on a graph** A random walk on this graph  $G$  visits the nodes represented by the Markov-chain:  $X_1, X_2, X_3, \dots$ . The first node is the start vertex

$(X_1)$ . In the  $k$ th step we move from vertex  $X_k = i$  to vertex  $X_{k+1} = j$ , with probability  $p_{j,i}$ .

Important properties for random walks are Hitting time and Mixing time. Hitting time describes how quickly can a vertex be reached from another vertex in the graph and Mixing time how fast the stationary distribution is reached (where we can no longer tell which vertex was the walk started in). Both of these measures will come up later in the analysis of the quantum walks.

**Definition 7. Hitting time**  $h_{j,i}$  is the expected number of steps before node  $j$  is visited in a random walk starting from node  $i$ .

Recursively:

$$h_{j,i} = \begin{cases} 1 + \sum_{k \in A} p_{j,k} h_{k,i} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \quad (2.1)$$

From [4] (35, 36):

**Definition 8. Mixing time:** The smallest time index of the Markov chain, where the total variational distance between the current and the stationary distribution is not greater than  $\epsilon$ .

$$m = \min\{T | \forall t \geq T, \sum_j |P(X_t = j) - \pi_j| \leq \epsilon\} \quad (2.2)$$

Should I define graphs too?

**Definition 9. Maximum degree of a graph** the maximum number of outgoing edges from a vertex is the maximum degree of a graph, for graph  $G$ , it is denoted as  $\Delta(G)$ .

**Definition 10. Graph regularity** We call a directed graph  $d$ -regular if all vertexes have  $d$  outgoing edges.

## Chapter 3

# Quantum informatics, quantum computing

### 3.1 The postulates of quantum mechanics

Based on Quantum computing and communications: an engineering approach by Balázs Ferenc and Imre Sándor [1] and Mika Hirvensalo's Quantum Computing[3].

Ez azért jó felvezető, mert a mérés posztulátumnál el tudom mondani az unitér tranfó azonosságait, a tenzoros posztulátumnál pedig a tenzorszorzás azonosságait, amikre később hivatkozni fogok.

Vajon szó szerint bemásolni a könyvből nagyon rosszul néz ki, pláne hogy valószínűleg olyan javítja aki olvasta?

**First postulate: State space** The actual state of any closed physical system can be described by means of a so called **state vector**  $v$  having complex coefficients and unit length in a Hilbert space  $V$  i.e. a complex linear vector space (**state space**) equipped with inner product.

A 2 dimensional Hilbert space is the simplest example of a closed physical system.

**Definition 11. Qubit** A unit vector in the 2 dimensional Hilbert space ( $H_2$ ) is a qubit. The base vectors in this space are  $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and  $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ . Any qubit can be written in the form  $a|0\rangle + b|1\rangle = \begin{pmatrix} a \\ b \end{pmatrix}$ , where  $a, b \in \mathbb{C}$  and  $|a|^2 + |b|^2 = 1$ .

**Postulate 2. Time evolution** The evolution of any closed physical system in time can be characterized by means of unitary transforms depending only on the starting and finishing time of the evolution.

**Definition 12. Unitary matrix**  $U$  is unitary if  $U^\dagger = U^{-1}$ .

TODO: [https://en.wikipedia.org/wiki/Unitary\\_matrix](https://en.wikipedia.org/wiki/Unitary_matrix) vagy kvantumos előadás di-sora, citation?. Az ortonormált bázist alkotó sorosat használok az élszínsoportok = érmeoldalok bizonyításnál.

Equivalent definitions:

1.  $U$ 's rows form an orthonormal basis of  $\mathbb{C}^n$ .
2.  $U$ 's columns form an orthonormal basis of  $\mathbb{C}^n$ .
3.  $U$  is an isometry: injective and preserves length.
4.  $U$  preserves the inner product.

**Postulate 3. Measurement** Any quantum measurement can be described by means of a set of measurement operators  $M_m$ , where  $m$  stands for the possible results of the measurement. The probability of measuring  $m$  if the system is in state  $|v\rangle$  can be calculated as

$$P(m|v) = \langle v | M_m^\dagger M_m | v \rangle \quad (3.1)$$

and the system after measuring  $m$  goes to state

$$|v'\rangle = \frac{M_m |v\rangle}{\sqrt{\langle v | M_m^\dagger M_m | v \rangle}} \quad (3.2)$$

due to classical probability

$$\sum_m P(m|v) = \sum_m \langle v | M_m^\dagger M_m | v \rangle = 1 \quad (3.3)$$

which means the measurement operators have to satisfy the following **completeness relation**

$$\sum_m M_m^\dagger M_m = I \quad (3.4)$$

**Projective measurement** To distinguish a set of orthonormal states ( $|\varphi_m\rangle$ ), the corresponding measurement operators can be produced by  $P_m = |\varphi_m\rangle \langle \varphi_m|$ .

Properties of projective measurement operators ( $P_m$ ):

**Theorem 1.  $P_m$  is self adjoint:**  $P_m^\dagger = P_m$ , since  $(|\varphi_m\rangle \langle \varphi_m|)^\dagger = \langle \varphi_m |^\dagger |\varphi_m\rangle^\dagger = |\varphi_m\rangle \langle \varphi_m|$

**Theorem 2.  $P_m P_m = P_m$ :**  $|\varphi_m\rangle \langle \varphi_m| \varphi_m\rangle \langle \varphi_m| = |\varphi_m\rangle (\langle \varphi_m | \varphi_m \rangle) \langle \varphi_m| = |\varphi_m\rangle 1 \langle \varphi_m| = |\varphi_m\rangle \langle \varphi_m|$

**Theorem 3.  $P_m$  and  $P_n$  are orthogonal:** If  $n \neq m$ , then  $P_m P_n = 0$ , since  $P_m P_n = |\varphi_m\rangle \langle \varphi_m | \varphi_n \rangle \langle \varphi_n| = |\varphi_m\rangle (\langle \varphi_m | \varphi_n \rangle) \langle \varphi_n| = |\varphi_m\rangle 0 \langle \varphi_n| = 0$ .

The value of a qubit can be any unit length vector in  $(H_2)$ , however when we measure it, we will receive one of the base vectors of  $(H_2)$ . For  $a|0\rangle + b|1\rangle$  we measure  $|0\rangle$  with probability  $|a|^2$  and  $|1\rangle$  with probability  $|b|^2$ .

**Postulate 4. Composite system** The state space of a composite physical system  $W$  can be determined using the tensor product of the individual systems  $W = V \otimes Y$ . Furthermore having defined  $v \in V$   $y \in Y$  then the joint state of the composite system is  $w = v \otimes y$ .

**Definition 13. Tensor product** The tensor product  $A \otimes B$  of matrix  $A$  of size  $r \times s$  and matrix  $B$  of size  $t \times u$  is of size  $rt \times su$  and is defined as follows:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1s} \\ a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1} & a_{r2} & \ddots & a_{rs} \end{pmatrix} \text{ és } B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1u} \\ b_{21} & b_{22} & \dots & b_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ b_{t1} & b_{t2} & \ddots & b_{tu} \end{pmatrix} \text{ esetén}$$

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1s}B \\ a_{21}B & a_{22}B & \dots & a_{2s}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{r1}B & a_{r2}B & \ddots & a_{rs}B \end{pmatrix}$$

Citation needed for these.

### 3.1.1 Identities of tensor products

**Theorem 4 (Associativity).**

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \quad (3.5)$$

**Theorem 5 (Distributivity over matrix product).** If  $A$  and  $C$ ; and  $B$  and  $D$  are compatible (can be multiplied) then:

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \quad (3.6)$$

Need a better name here

**Theorem 6 ("Unitary matrices disappearing").**

$$(A_n \otimes I_m)(I_n \otimes B_m) = A_n \otimes B_m \quad (3.7)$$

**Definition 14. Quantum register** The quantum register consisting of  $n$  qubits is the tensor product of  $n$  qubits (on  $H_2$ ).

$$H_2^{\otimes n} = H_2 \otimes H_2 \otimes \dots \otimes H_2 \quad (3.8)$$

**Definition 15. Entangled state** Any state consisting of multiple qubits, that can not be written in the form of a tensor product of individual qubits is entangled.

For example, the state  $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ , is entangled, since it can not be written in the form  $(a_0|0\rangle + a_1|1\rangle) \otimes (b_0|0\rangle + b_1|1\rangle) = a_0b_0|00\rangle + a_0b_1|01\rangle + a_1b_0|10\rangle + a_1b_1|11\rangle$ , since that would require from  $a_0b_0 = a_1b_1 = \frac{1}{\sqrt{4}}$ , for all coefficients to be non-zero and from  $a_0b_1 = a_1b_0 = 0$  for either  $a_0$  or  $b_1$  and either  $a_1$  or  $b_0$  to be zero, which is a contradiction.

## Chapter 4

# Quantum walk

Classical random graph walks can be seen as using a classical coin flip to decide in the current vertex which neighbouring vertex to step next. To create a quantum version of these walks, we define the quantum coin to randomize the walk, then use entanglement between the coin and the register of the current vertex, to influence stepping.

A great introduction to quantum walks is in Portugal's Quantum Walks and Search Algorithms book [5].

### 4.1 Definition of the quantum coin

The quantum coin has a current state, which according to Postulate 1 and 4 is an  $n$ -qubit register and a transition operator, which according to Postulate 2 is a unitary matrix.

The current state decides where to step next. If there are  $d$  neighbours to choose from, the coin's current state register must be  $d$  dimensional.

The transition operator decides how the coin is flipped. If the coin is  $d$  dimensional, the transition operator is a  $d \times d$  dimensional matrix. Based on what the transition operator is, multiple types of coins can be defined. The following ones are typically used in quantum walks.

#### Definition 16. Hadamard coin

The Hadamard coin uses the Hadamard-matrix as a transition operator, which is the following:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (4.1)$$

If the starting coin state is  $|0\rangle$ , then flipping the coin once results in the following state:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \quad (4.2)$$

If we measured the above coin, the probability of measuring  $|0\rangle$  is  $\left|\frac{1}{\sqrt{2}}\right|^2 + \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$ .

Similarly the starting coin state is  $|1\rangle$ , then flipping the coin once results in the following state:

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \quad (4.3)$$

the probability of measuring  $|1\rangle$  is similarly  $\left|\frac{1}{\sqrt{2}}\right|^2 + \left|-\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$ .

An interesting property about this coin is related to the fact, that the Hadamard-matrix is Hermitian (self-adjoint), i.e.  $H^\dagger = H$ , while also unitary, i.e.  $H^\dagger = H^{-1}$ , which results in  $H^{-1} = H$ , thus  $HH = I$ . This means, that if we flip the coin twice, without measuring it, it will return to its original state, for example:

$$H^2|0\rangle = H\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}(|0\rangle + |1\rangle + |0\rangle - |1\rangle) = |0\rangle \quad (4.4)$$

After the second flip, the probability of measuring  $|0\rangle$  is 1, due to the destrutive interference between the  $|1\rangle$  vectors. This illustrates the vast difference between classical and quantum walks.

**Definition 17.  $2^n$  dimensional Hadamard-coin:** A  $2^n$  dimensional Hadamard-coin can be created by taking the tensor product of the 2 dimensional Hadamard-coin  $n$  times:  $H^{\otimes n}$

**Definition 18. Grover coin**

The Grover coin's origin is the Grover search algorithm, where it is used as the diffusion operator, defined as follows:

The diagonal state of the coin space:

$$|D\rangle = H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle \quad (4.5)$$

Grover coin:

$$G = 2|D\rangle\langle D| - I \quad (4.6)$$

If  $N = 2^n$ , then

$$G = \begin{pmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{pmatrix} \quad (4.7)$$



**Fourier coin** While both the Hadamard and the Fourier coin need to be sized as a power of 2, the Fourier coin does not have this limitation. A size  $N$  Fourier-coin,  $F_N$  is defined by the matrix of the Fourier-transform:

A Quantum diszkrét Fourier transzformáció mátrixa talán ez?

$$[F_N]_{k,l} = \frac{1}{\sqrt{N}} \omega^{kl} \quad (4.8)$$

where

$$\omega = e^{\frac{2\pi i}{N}} \quad (4.9)$$

$$F_N = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & e^{\frac{2\pi i}{N}} & \dots & e^{\frac{2\pi i(N-1)}{N}} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\frac{2\pi i(N-1)}{N}} & \dots & e^{\frac{2\pi i(N-1)^2}{N}} \end{pmatrix} \quad (4.10)$$

## 4.2 Quantum walks on the line

Ez félig az önlabomból van és a Kempe cikkből.

Az önlabomat lehet fel kellene töltenem valahova, ha már hivatkozok rá.

The simplest form of a quantum walk to discuss is the quantum walk on the line. A great introduction into this is in [4]. In this type of walk, a particle is described by its position on the line  $|x\rangle$  and its spin state  $|s\rangle$ .

The spin state is a 2-dimensional Hilbert space  $H_2$  and the basis states are called spin up  $|s\rangle = |\uparrow\rangle = |0\rangle$  or a spin down  $|s\rangle = |\downarrow\rangle = |1\rangle$ . Let us state, that at the start of the walk, the particle is at the origin,  $|0\rangle$  and the walking lasts for  $|N\rangle$  steps. Then, the position state is a  $(2N+1)$ -dimensional Hilbert space  $H_{(2N+1)}$  with basis vectors  $\{|-N\rangle, |-(N-1)\rangle, \dots, |-1\rangle, |0\rangle, |1\rangle, \dots, |N-1\rangle, |N\rangle\}$  corresponding to the possible positions on the line.

Itt nem tudom eldönteni, hogy negatív számokat használjak vagy  $N/2$ -ből induljon a részecske és ne 0-ból.

The complete state of the system, according to the 4th Postulate is  $|x\rangle \otimes |s\rangle$ .

The particle travels on the line based on its current spin state. If the current spin state is  $|0\rangle$ , the particle moves to the left, i.e. from position  $|i\rangle$  the particle travels to position  $|i-1\rangle \forall -N < i$  and stays in  $|-N\rangle$  for  $i = -N$ . If the current spin state is  $|1\rangle$ , the particle moves to the right, i.e. from position  $|i\rangle$  the particle travels to position  $|i+1\rangle \forall i < N$  and stays in  $|N\rangle$  for  $i = N$ .

This step is described by the unitary matrix  $S$ , which operates on the complete state of the system,  $|x\rangle \otimes |s\rangle$  and is constructed as follows:

Cite this?

**Definition 19. Kronecker delta** Kronecker delta is a function acting on two input variables, which takes the output value 1, when the input variables equal and the output value 0, when the input variables differ.

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \quad (4.11)$$

Using the Kronecker delta function, a matrix can be defined with the following notation:

$$M = [\delta_{f(i),g(j)}]_{i,j} \quad (4.12)$$

which means

$$M_{i,j} = \delta_{f(i),g(j)} \quad (4.13)$$

Ezt valahogy szebben kellene írni.

To simplify notation,  $f(i)$  and  $g(j)$  are interpreted with a modulo operator of the size of the matrix.

**Definition 20. Left shift operator**

To move from position  $|x\rangle$  to the left,  $|x\rangle$  is multiplied with a matrix of size  $(2N + 1) \times (2N + 1)$ , that contains 1's above the diagonal and in the lower left corner, as follows:

$$L = [\delta_{(i+1),j}]_{i,j} \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ & & & \ddots & 1 \\ 1 & \cdots & & & 0 \end{pmatrix}$$

**Definition 21. Right shift operator**

To move from position  $|x\rangle$  to the right,  $|x\rangle$  is multiplied with a matrix of size  $(2N + 1) \times (2N + 1)$ , that contains 1's below the diagonal and in the upper right corner, as follows:

$$U = [\delta_{(i-1),j}]_{i,j} \begin{pmatrix} 0 & \cdots & 1 \\ 1 & \ddots & & \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$$

**Shift operator** Using matrixes  $L$  and  $U$  operating on the position register  $|x\rangle$  only, we construct a unitary operator  $S$ , which operates on the complete state of the system,  $|x\rangle \otimes |s\rangle$ , executing matrix  $L$  on  $|x\rangle$  only when  $|s\rangle = |0\rangle$  and matrix  $U$  only when  $|s\rangle = |1\rangle$ .

The operator is the following:

$$S = L \otimes |0\rangle\langle 0| + U \otimes |1\rangle\langle 1| \quad (4.14)$$

The execution logic is as follows:

TODO: tenzorszorzás disztributivitását kellene itt szépen behivatkozni.

(Let us assume that the position is not  $|-N\rangle$  or  $|N\rangle$  for simplicity.)

TODO:  $|x-1\rangle$  nem mindig megy balra, ezt hogy kellene jelölni 3.18-ra?

$$S(|x\rangle \otimes |s\rangle) = \quad (4.15)$$

$$(L \otimes |0\rangle\langle 0| + U \otimes |1\rangle\langle 1|)(|x\rangle \otimes |s\rangle) = \quad (4.16)$$

$$(L \otimes |0\rangle\langle 0|)(|x\rangle \otimes |s\rangle) + (U \otimes |1\rangle\langle 1|)(|x\rangle \otimes |s\rangle) = \quad (4.17)$$

$$L|x\rangle \otimes (|0\rangle\langle 0|s\rangle) + U|x\rangle \otimes (|1\rangle\langle 1|s\rangle) = \quad (4.18)$$

$$|x-1\rangle \otimes s_0|0\rangle + |x+1\rangle \otimes s_1|1\rangle \quad (4.19)$$

If the spin state was  $|s\rangle = |0\rangle$  at the beginning, then  $s_1 = 0$ , which means the resulting system state is  $|x-1\rangle \otimes |0\rangle$ , which means the particle shifted to the left, as designed. If the spin state was  $|s\rangle = |1\rangle$  at the beginning, then  $s_0 = 0$ , which means the resulting system state is  $|x+1\rangle \otimes |1\rangle$ , which means the particle shifted to the right, also as intended.

However the spin state can be any mixed state as well. This means, for a mixed spin state, the particle will shift both to the left and to the right, at the same time. When measured, the particle can be found in position  $|x-1\rangle$  with probability  $|s_0|^2$  and in  $|x+1\rangle$  with probability  $|s_1|^2$ .

In quantum graph walks, the walker can simultaneously explore multiple parallel paths in the graph, at the same time. With good design, this behaviour can be used to search the graph faster than in classical random graph walks.

Itt be lehetne rakni azokat a nagy mátrix ábrákat az önlabomból, vagy csak hivatkozni, vagy inkább semmit.

**Coin operator** To introduce randomness into the walk, between each step, the particle's spin state is transformed using any 2 dimensional unitary matrices. The previously mentioned Hadamard, Grover or Fourier coins are commonly used as coin operators.

For any  $U_c$  unitary coin flip transform on the coin register, the coin operator for the complete system is defined as follows:

$$C = I \otimes U_c \quad (4.20)$$

since the coin operator does not modify the position register.

**Complete step** One step of the quantum walk on the line consists of flipping the coin once, then applying the shift operator, as follows:

$$U = SC = S(I \otimes U_c) \quad (4.21)$$

**Measurement** To measure the probability of the particle being at position  $|i\rangle$ , the projective measurement operator acting on  $|x\rangle$  is defined as  $P_i = |i\rangle\langle i|$ , according to the projective measurement definition previously.

Behivatkozni hogyan kell?

Since the coin register need not be measured, we apply the identity operator on it, using  $P_i \otimes I$  on the complete system to measure the particle's current position.

The probability of finding the particle in position  $i$  is:

$$P(i|x) = \langle x, s | P_i \otimes I | x, s \rangle \quad (4.22)$$

### 4.3 Quantum walks on larger dimensional grids

One of the ways described in Renato Portugal's book to increase the dimension of the walk is to use independent one-dimensional walks.

Ezt angolra kell majd fordítani és leírást készíteni mellé. Vagy kidobni ha túl sok?

A következőkben a rácson, mint 4-reguláris gráfon vett bolyongást fogjuk részletesen megvizsgálni.

Legyen a rács  $N = n \times n$ -es.

Pozíció vektor:  $|P\rangle = (p_0, \dots, p_N)$ , ahol  $p_i \in \mathbb{C}$  ahol  $0 \leq i < N$ ,  $\forall i$ -re és  $\sum_{i=0}^N |p_i|^2 = 1$ .

$$|P_0\rangle = (0, \dots, 0, 1, 0, \dots, 0) \quad (4.23)$$

$$C = \left(\frac{1}{\sqrt{2}}, \frac{i}{\sqrt{2}}\right) \otimes \left(\frac{1}{\sqrt{2}}, \frac{i}{\sqrt{2}}\right) \quad (4.24)$$

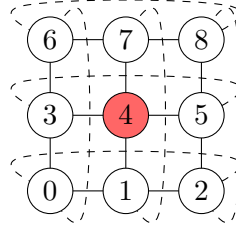
$$S_{\text{left}} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & \end{pmatrix} \quad (4.25)$$

$$L = U^T \quad (4.26)$$

- Rács = két körnek a tenzorszorzata.
- Fel/le = mindig 1 sorral felette/alatta lévő (modulóval) van összekötve = n-el van elcsúsztatva.
- Függőlegesen számozva ugyanígy csak fordított elnevezések.
- Jobbra-balra = 1 távolság, kivéve sor végén, ahol a sor elejére ugrik = kis körök.
- Tenzorszorzás tulajdonságait előre kiszedni, műveletek, használatkor hivatkozni.
- Többsoros képleteknek csak 1 száma legyen.

#### 4.4 Példa

$$n = 3 \quad (4.27)$$



**Figure 4.1:** n=3-ös rács

$$N = n^2 = 9 \quad (4.28)$$

$$S_{\text{left}} = I_3 \otimes U_3 = \begin{pmatrix} 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \end{pmatrix} \quad (4.29)$$

$$S_{\text{right}} = I_3 \otimes L_3 = \begin{pmatrix} 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \end{pmatrix} \quad (4.30)$$

$$S_{\text{down}} = U_3^3 = U_3 \otimes I_3 = \begin{pmatrix} 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.31)$$

$$S_{\text{up}} = L_3^3 = L_3 \otimes I_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \end{pmatrix} \quad (4.32)$$

$$X_{\text{head}} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad (4.33)$$

$$X_{\text{tail}} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (4.34)$$

$$S = (S_{\text{up}} S_{\text{left}}) \otimes (X_{\text{head}} \otimes X_{\text{head}}) + \quad (4.35)$$

$$(S_{\text{up}} S_{\text{right}}) \otimes (X_{\text{head}} \otimes X_{\text{tail}}) + \quad (4.36)$$

$$(S_{\text{down}} S_{\text{left}}) \otimes (X_{\text{tail}} \otimes X_{\text{head}}) + \quad (4.37)$$

$$(S_{\text{down}} S_{\text{right}}) \otimes (X_{\text{tail}} \otimes X_{\text{tail}}) \quad (4.38)$$

$$C_4 = H^{\otimes 2} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad (4.39)$$

$$U = S(I_9 \otimes C_4) \quad (4.40)$$

$$U = ((S_{\text{up}}S_{\text{left}}) \otimes (X_{\text{head}} \otimes X_{\text{head}}) + \quad (4.41)$$

$$(S_{\text{up}}S_{\text{right}}) \otimes (X_{\text{head}} \otimes X_{\text{tail}}) + \quad (4.42)$$

$$(S_{\text{down}}S_{\text{left}}) \otimes (X_{\text{tail}} \otimes X_{\text{head}}) + \quad (4.43)$$

$$(S_{\text{down}}S_{\text{right}}) \otimes (X_{\text{tail}} \otimes X_{\text{tail}}))(I_9 \otimes C_4) \quad (4.44)$$

**Definition 22 (Tenzorszorzás azonosság: mátrix szorzással disztibutív).** Bal oldalt  $9 \times 9$ -es, jobb oldalt  $4 \times 4$ -es mátrixok vannak.

$$U = (S_{\text{up}}S_{\text{left}}) \otimes ((X_{\text{head}} \otimes X_{\text{head}})C_4) + \quad (4.45)$$

$$(S_{\text{up}}S_{\text{right}}) \otimes ((X_{\text{head}} \otimes X_{\text{tail}})C_4) + \quad (4.46)$$

$$(S_{\text{down}}S_{\text{left}}) \otimes ((X_{\text{tail}} \otimes X_{\text{head}})C_4) + \quad (4.47)$$

$$(S_{\text{down}}S_{\text{right}}) \otimes ((X_{\text{tail}} \otimes X_{\text{tail}})C_4) \quad (4.48)$$

$$C_4 = C_2 \otimes C_2 \quad (4.49)$$

$$U = (S_{\text{up}}S_{\text{left}}) \otimes ((X_{\text{head}} \otimes X_{\text{head}})(C_2 \otimes C_2)) + \quad (4.50)$$

$$(S_{\text{up}}S_{\text{right}}) \otimes ((X_{\text{head}} \otimes X_{\text{tail}})(C_2 \otimes C_2)) + \quad (4.51)$$

$$(S_{\text{down}}S_{\text{left}}) \otimes ((X_{\text{tail}} \otimes X_{\text{head}})(C_2 \otimes C_2)) + \quad (4.52)$$

$$(S_{\text{down}}S_{\text{right}}) \otimes ((X_{\text{tail}} \otimes X_{\text{tail}})(C_2 \otimes C_2)) \quad (4.53)$$

**Definition 23 (Tenzorszorzás azonosság: mátrix szorzással disztibutív).**

$$U = (S_{\text{up}}S_{\text{left}}) \otimes (((X_{\text{head}}C_2) \otimes (X_{\text{head}}C_2))) + \quad (4.54)$$

$$(S_{\text{up}}S_{\text{right}}) \otimes (((X_{\text{head}}C_2) \otimes (X_{\text{tail}}C_2))) + \quad (4.55)$$

$$(S_{\text{down}}S_{\text{left}}) \otimes (((X_{\text{tail}}C_2) \otimes (X_{\text{head}}C_2))) + \quad (4.56)$$

$$(S_{\text{down}}S_{\text{right}}) \otimes (((X_{\text{tail}}C_2) \otimes (X_{\text{tail}}C_2))) \quad (4.57)$$

**Definition 24 (Tenzorszorzás azonosság: asszociatív).**

$$U = (S_{\text{up}}S_{\text{left}}) \otimes (X_{\text{head}}C_2) \otimes (X_{\text{head}}C_2) + \quad (4.58)$$

$$(S_{\text{up}}S_{\text{right}}) \otimes (X_{\text{head}}C_2) \otimes (X_{\text{tail}}C_2) + \quad (4.59)$$

$$(S_{\text{down}}S_{\text{left}}) \otimes (X_{\text{tail}}C_2) \otimes (X_{\text{head}}C_2) + \quad (4.60)$$

$$(S_{\text{down}}S_{\text{right}}) \otimes (X_{\text{tail}}C_2) \otimes (X_{\text{tail}}C_2) \quad (4.61)$$

**Definition 25 (Tenzorszorzás Lemma?).**

$$S = ((S_{\text{up}} \otimes X_{\text{head}} \otimes I) + (S_{\text{down}} \otimes X_{\text{tail}} \otimes I))((S_{\text{left}} \otimes I \otimes X_{\text{head}}) + (S_{\text{right}} \otimes I \otimes X_{\text{tail}})) \quad (4.62)$$

$$U = ((S_{\text{up}}S_{\text{left}}) \otimes I_2)(I_9 \otimes (X_{\text{head}}C_2)) \otimes (X_{\text{head}}C_2) + \quad (4.63)$$

$$((S_{\text{up}}S_{\text{right}}) \otimes I_2)(I_9 \otimes (X_{\text{head}}C_2)) \otimes (X_{\text{tail}}C_2) + \quad (4.64)$$

$$((S_{\text{down}}S_{\text{left}}) \otimes I_2)(I_9 \otimes (X_{\text{tail}}C_2)) \otimes (X_{\text{head}}C_2) + \quad (4.65)$$

$$((S_{\text{down}}S_{\text{right}}) \otimes I_2)(I_9 \otimes (X_{\text{tail}}C_2)) \otimes (X_{\text{tail}}C_2) \quad (4.66)$$

## 4.5 Quantum walks on Class 1 regular graphs

The generalization technique described in the previous chapter only works for grids of any dimension. To generalize to arbitrary graphs, first we need to describe the quantum walk on a regular graph. This is done following Renato Portugal's work [5].

In a  $d$ -regular graph, the number of neighbours is always  $d$ . This means, that the coin is  $d$  dimensional. In non-regular graphs, there is an issue arising from needing different sized coins at different vertexes, which is discussed in the next section.

The coin operator can be defined as before, by applying the identity operator on the position register and any coin flip unitary matrix on the coin register.

To define the shift operator, let's introduce the edge chromatic number of a graph.

Cite these, they are from wikipedia.

**Definition 26. Edge coloring**

An edge coloring of a graph is an assignment of colors to the edges of the graph, so that no two incident edges have the same color.

**Definition 27. Edge chromatic number,  $\chi'(G)$**  is the minimum number of colors that can be used to correctly color the graph.

**Theorem 7. Vizing's theorem**

The edge chromatic number  $\chi'(G)$  of graph  $G$  is  $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$ . Graphs with  $\chi'(G) = \Delta(G)$  are called Class 1 graphs and graphs with  $\chi'(G) = \Delta(G) + 1$  are called Class 2 graphs.

Ez az a tétel amit nem találtam meg sehol, itt kellene írni hogy én bizonyítottam, de ismert eredmény.

**Theorem 8.** For Class 1,  $d$ -regular graphs, the coin sides correspond to the color sets of a correct edge coloring.



$d$ -reguláris gráfra  $d$  oldalú érmével:

- A gráf egy adott csúcsából úgy döntjük el, hogy melyik következő csúcsba lépünk tovább, hogy feldobjuk az érmét és a kapott oldalhoz tartozó élen megyünk a következő csúcsba.
- Ehhez akkor tudni kell, hogy adott csúcsból nézve melyik hozzá tartozó él melyik érmeoldalhoz tartozik.
- Az érmeoldalakat számozzuk  $0, \dots, d-1$ -el.
- Minden csúcsban a hozzá tartozó  $d$  darab élet megszámozzuk  $0, \dots, d-1$ -el.
- (Ha ezt lehet úgy, hogy minden élre az él két végéhez ugyanaz a szám tartozik akkor irányítatlan gráfként kezeljük, ha nem lehet akkor meg irányítottként és a kezdőpontjában az eredeti élhez rendelt számot kapja. ?)
- Ha egy adott csúcsban  $i$ -t dobunk az érmével akkor a hozzá tartozó  $i$ . sorszámú élen megyünk tovább.
- Ezt a mesét most mátrixszorzásokkal akarjuk felírni:
- Az érme legyen az  $|i\rangle$  vektor  $i = 0, \dots, d-1$ , az pedig hogy éppen melyik gráfcsúcsban vagyunk legyen a  $|v\rangle$  vektor  $v = 0, \dots, n-1$ .
- Akkor az aktuális állapot (melyik oldala van felül az érmének és hol vagyok a gráfban)  $|v\rangle \otimes |i\rangle$ .
- Jelölje  $S_i$  azt a mátrixot, ahol akkor van az  $x$ . sor  $y$ . oszlopában 1-es, ha van a gráfban  $(y, x)$  él és az  $y$  csúcsban az él sorszáma  $i$ .
- Ekkor minden oszlopban csak 1 darab nem 0 elem van, mert az  $y$  csúcsnak csak 1 darab  $i$ . szomszédja van.
- Azt a lépést, hogy ha az érme  $k$ -t mutat akkor az adott csúcs  $k$ . szomszédjába megyek  $S_k \otimes |k\rangle \langle k|$ -vel lehet leírni, mert ha ezzel balról szorzom a  $|v\rangle \otimes |i\rangle$ -t akkor ez csak akkor nem 0 ha  $\langle k|i\rangle$  nem 0, vagyis ha  $k = i$  és ilyenkor az  $S_k$  pont a  $v$  csúcs  $i = k$ . szomszédjába transzformálja  $|v\rangle$ -t.
- Akkor a teljes lépés mátrix  $S = \sum_{i=0}^{d-1} S_i \otimes |i\rangle \langle i|$ .
- Ennek unitérnek kell lennie, mert kvantum.
- Ezért az oszlopok ortonormált bázist alkotnak. (Meg a sorok is, de az most nem kell.)
- Ezért az oszlopok skaláris szorzata 0 kell hogy legyen.
- Nincsenek negatív értékek az  $S_i$  mátrixokban és a  $|k\rangle \langle k|$ -kban sem és ezért az  $S$ -ben sem, ezért ez csak úgy lehet ha nincs két olyan különböző oszlop  $S$ -ben ahol ugyanannyiadik cellában van nem 0 érték.
- Kétféleképpen fordulhat elő, hogy  $S$ -ben két különböző oszlopban ugyanannyiadik cellában nem 0 érték: a két nem 0 elem vagy ugyanabból az  $S_i \otimes |i\rangle \langle i|$  mátrixból jött az  $S = \sum_{i=0}^{d-1} S_i \otimes |i\rangle \langle i|$  összegben vagy 2 különbözőből.

- Különbözőből nem jöhetett, mert az  $|i\rangle\langle i|$ -vel való tenzorsszorzás különböző sorokba fogja tenni az 1-eseket a különböző  $S_i \otimes |i\rangle\langle i|$ -kre.
- Ha azonosból jött akkor egy  $S_i \otimes |i\rangle\langle i|$  mátrixban van két különböző oszlop ahol ugyanannyiadik cellában van nem 0 elem.
- Mivel  $|i\rangle\langle i|$ -ben csak 1 darab nem 0 elem van, ezért
- Akkor az  $S_i$  mátrixban van két különböző oszlop ahol ugyanannyiadik cellában van nem 0 elem.

És akkor ezek a lépések eddig végig "akkor és csak akkor" kapcsolatban voltak egymással.

Ez igaz?

Vagyis az jött ebből ki, hogy akkor és csak akkor unitér  $S$  ha az  $S_i$  mátrixokban nincs két különböző oszlop ahol ugyanannyiadik cellában van nem 0 elem. Vagyis minden sorban csak 1 darab nem 0 elem lehet. Közben pedig valahol felírtuk hogy minden oszlopban is csak 1 darab nem 0 elem lehet. És mivel  $d$ -reguláris a gráf ezért minden sorban és oszlopban van is 1 darab 1-es és ezért ezek permutációmátrixok lehetnek csak.

Az  $S_i$  pedig egy élszínsztályt ír le, mert ha az  $x$ . sor  $y$ . eleme 1-es akkor az egy él a gráfban és  $S_i$ -ben nincs másik él ami  $y$ -ból mutatna kifelé vagy  $x$ -be mutatna befelé. (Itt akkor kell hogy irányított legyen, mert az  $x$ . oszlopban vagy az  $y$ . sorban még lesz 1-es.)

És akkor ilyen módon csak azokra a gráfokra lehet csak sétát definiálni amik  $d$  szín-nel élszínezhetőek és amik pedig  $d + 1$ -el azokra pedig nem, azokkal a következő fejezet foglalkozik.

## 4.6 Quantum walks on Class 2 or non-regular graphs

### 4.6.1 Class 2

For graphs in class 2, we can not use the same coin-position notation, we have to use the arc notation and replace the simple graph by an associated symmetric digraph, whose underlying graph is the original graph. In this case, the walker steps on the arcs of the digraph. After those considerations, we are able to define formally coined quantum walks.

### 4.6.2 Make the graph regular

Add loops (and hope they don't ruin the edge chromatic number?).

### 4.6.3 Szegedy walk

Walk on the edges.

## Chapter 5

# Szimulátor szoftver

Itt nagyon kiemelni, hogy önlab1-ben csak a klasszikus bolyongás sikerült + az egyenesen a kvantumos, most meg már tudok bármilyen regulárison bolyongani.

### 5.1 Tervezési megfontolások

Az első fázisban a szomszédossági mátrixot generáltam le teljesen. Ez nagyon sok memóriát használt, az unitér mátrix meg még többet, nagyon rosszul skálázódott. Lecséréltem spare mátrixokra, az már egy fokkal jobb.

Mérés összehasonlítás: Mátrix, Spare mátrix, Orákulum grafikon.

Ezután kipróbáltam a spare mátrixokat, de azok is picit sokak voltak, ugyan a gráfban sok a 0, de azért így is nagyon skálázódott.

A legjobb megoldás az úgynevezett gráf orákulum volt. A gráf orákulum lényege, hogy nem tároljuk a szomszédossági mátrixot, helyette biztosítunk egy függvényt, mely on-the-fly a gráf adott csúcsindexét odaadva neki visszaadja hogy annak mik a szomszédai. A coin kis méretű, azt lehet tárolni ( $\text{max fokszám} * \text{max fokszám}$ ), ezért azzal nem bajlódtam, bár éppen lehetne.

Emellett a Stratégia tervezési mintát használatam.

TODO: [https://en.wikipedia.org/wiki/Strategy\\_pattern](https://en.wikipedia.org/wiki/Strategy_pattern), de valamilyen Design Patternes könyvből inkább.

## Chapter 6

# Architektúra

Ezt az önlabomból hoztam át, maga a keret nem sokat változott, az változott nagyot hogy a kvantumos szimulátor is működik már d reguláris gráfokra.

Ebben a fejezetben bemutatom az elkészült keretrendszert.

Programozási nyelvnek a Python 3-mat választottam. Ennek oka az, hogy nagyon sok data science-el kapcsolatos modulja van, mely nagyban megkönnyíti a különböző matematikai, algoritmuselméleti problémák feltárását, könnyen iterálhatunk a különböző prototípusokon. Emellett a szintaxisa rövid, tömör, lényegretörő programkódok megírását teszi lehetővé.

A forráskód három nagy részre bomlik:

- Gráfmodellek
- Szimulátorok
- Futtatás, konfiguráció, eredmények ábragenerátora

### 6.1 Gráfmodellek

A félév során sokféle gráfon futtattam szimulációs kísérleteket, melyek során több problémába ütköztem. Kezdetben úgy oldottam meg a szimulációkat, hogy a célgráfok szomszédossági mátrixait generáltam le, egyben a memóriában tartva azokat és a lépések során a megfelelő csúcshoz tartozó sorokat lekérdezve.

Ezzel a módszerrel több probléma is jelentkezett. Az első gondot az okozta, hogy a szomszédossági mátrix mérete a csúcsszám négyzetével arányos, ezért pár ezer csúcsú gráfot már nem tudtam a memóriában tartva szimulálni. A második probléma pedig az volt, hogy a szomszédossági mátrixos ábrázolás nagyon távol esett az emberi szempontból természetes ábrázolástól. A kvantumbolyongásos szimulációkat tipikusan nem véletlenszerű gráfokon szokták kipróbálni, hanem jól ismert struktúrával rendelkező gráfokon. Ilyen gráfok például a „súlyzók” vagy a ragasztott bináris fák.

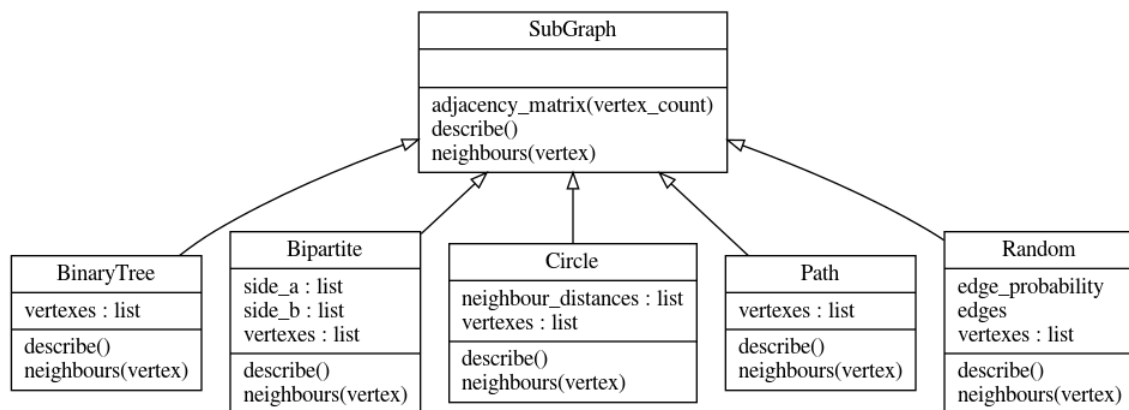
A súlyzó gráf két egyforma méretű kört tartalmaz, mindkét körből kiválasztva  $k - k$  darab csúcst, melyek teljes páros gráfot alkotnak (a súlyzó középső rúdját). A körökben pedig nem csak az egymás melletti csúcsok között fut él, hanem futhat él minden  $i$ . csúcs között is. A ragasztott bináris fában két egyforma méretű teljes bináris fa leveleit szembefordítjuk és a két oldali levelek közé egy teljes páros gráfot készítünk.

A fenti leírásból látható, hogy az ember számára természetes leírás a gráfokat ismert részgráfok kompozitjaként adja meg. A félév során olyan architektúrát alakítottam ki a szimulációkhoz, mely ezt a szemléletet támogatja. A szomszédossági mátrixos tárolási mód helyett pedig a szomszédossági orákulum megközelítést használva nagyban csökkent az alkalmazás memóriaigénye. Ennek a megközelítésnek a lényege, hogy az ismert struktúrájú gráfokra nem tárolok a memóriában szomszédossági információt, helyette biztosítok egy függvényt, amely a bemeneti paraméterként kapott csúcsindexre kiszámolja a vele szomszédos csúcsok indexeit.

A félév során a következő nevesített részgráfok szomszédossági orákulumját implementáltam:

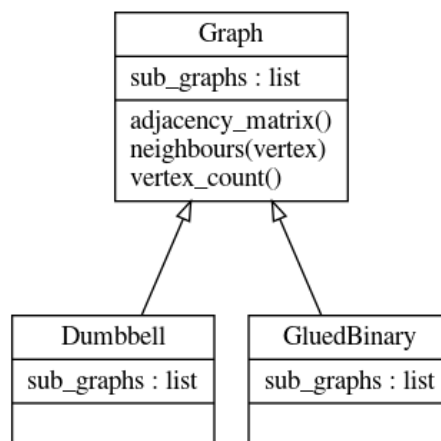
- BinaryTree
- Bipartite
- Circle
- Path
- Random

Új: Hiperkocka



Ezen részgráfokból épülnek fel az alábbi kompozit gráfok:

- Dumbbell
- GluedBinary

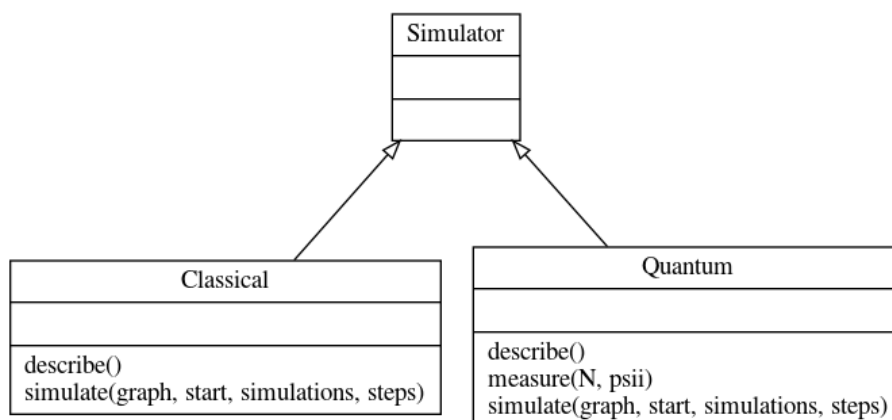


## 6.2 Szimulátorok

A szimulátor osztályok közül a klasszikus tetszőleges kompozit gráfot tud fogadni.

Itt fontos átírni, hogy a kvantumszimulátor mostmár d-regulárist is tud!

A kvantumszimulátor jelenleg a kvantumbolyongás egy speciális esetét, az egyenesen való bolyongást képes kezelni, mely a 2-regularitása miatt egyszerűbben implementálható. Hosszú távú cél a  $k$ -reguláris, illetve az általános gráfokra kiterjeszteni ezt a szimulátort.



## 6.3 Futtatás, konfiguráció, eredmények ábragenerátora

Itt van pár újdonság, pl. sajátértékek kiírása, stb.

A fenti osztályok segítségével egy olyan keretrendszert alakítottam ki, melyben nagyon gyorsan fel lehet 1-1 futtatást konfigurálni. A futtatás eredményeit egy összesített Latex dokumentumba gyűjti a program. Ez tartalmazza a beadott gráf részgráfjainak nevesített típusát, szomszédossági mátrixait, illetve a teljes gráf szomszédossági mátrixát, valamint a szimulációk eloszlási eredményeit. A következő fejezetben több ilyen ábrát is bemutatok.

## Chapter 7

# Other Quantum Walk Packages

Ennek a fejezetnek az lenne a célja, hogy megnéztem sok más programot és egyik se olyan jó mint az enyém, talán kevesebb szöveg kellene és arra fókuszálni hogy miért nem jók ezek, nem hogy pont mit csinálnak, de csak beraktam amit a könyv ír róluk. Általánosan mindegyik bonyolult, C++-os, nem szép tervezési mintájú, vagy csak speciális gráfokra működik (pl. csak grid).

[5] gives a detailed description of currently available quantum walk packages.

### 7.1 QWalk

QWalk aims to simulate the coined quantum walk dynamics on one- and twodimensional lattices. The package is written in C and uses Gnuplot to plot the probability distribution. The user can choose the coin and the initial condition. There is an option to simulate decoherent dynamics based on broken links—also known as percolation [251, 280]. The links of the lattice can be broken at random at each step or the user can specify which edges will be missing during the evolution. QWalk allows the user to simulate quantum walks on any graph that is a subgraph of the two-dimensional lattice. Some plots in this section were made using QWalk. The package can be obtained from Mendeley: <https://data.mendeley.com/datasets/b93c846dhs/1>,

Ez ugyanaz: <https://github.com/QWalk/mainline?>

### 7.2 QwViz

QwViz aims at plotting graphics for visualizing the probability distribution of quantum walks on graphs. The package is written in C and uses OpenGL to generate two- or three-dimensional graphics. The user must enter the adjacency matrix of the graph, and the package simulates the dynamics of the coined model to calculate the probability distribution. By default, the walker starts at vertex 1 with the coin in uniform superposition. The initial location can be changed by the user. It is possible to specify marked vertices, which tell the package to use quantum-walk-based search procedures starting from a uniform superposition of all vertices and using the Grover coin on the unmarked vertices and  $(-I)$  on the marked vertices. The package can be obtained from <https://data.mendeley.com/datasets/kj58zkdmt7/1>.

### 7.3 PyCTQW

PyCTQW aims to simulate large multi-particle continuous-time quantum walks using object-oriented Python and Fortran. The package takes advantage of modern HPC systems and runs using distributed memory. There are tools to visualize the probability distribution and tools for data analysis. The package can be obtained from <https://data.mendeley.com/datasets/wy29d79ds4/1>.

### 7.4 Hiperwalk

Hiperwalk (high-performance quantum walk) aims to simulate the quantum walk dynamics using high-performance computing (HPC). Hiperwalk uses OpenCL to run in parallel on accelerator cards, multicore CPU, or GPGPU. It is not required any knowledge of parallel programming, but the installation of the package dependencies is tricky, in special, OpenCL. Besides, Hiperwalk uses the Neblina programming language. In the CUSTOM option, the input is an initial state  $|\Psi_0\rangle$  and a unitary operator  $U$ , which must be stored in two different files (only nonzero entries in order to take advantage of sparsity). Hiperwalk calculates  $U^t |\Psi_0\rangle$  for integer  $t$  using HPC and saves the output in a file. There are extra commands for the coined and staggered models. The Hiperwalk manual has a detailed description of the installation steps and some examples of applications.

### 7.5 QSWalk

QSWalk is a Mathematica package that aims to simulate the time evolution of quantum stochastic walks on directed weighted graphs. The quantum stochastic walk is a generalization of the continuous-time quantum walk that includes the incoherent dynamics. The dynamic uses the Lindblad formalism for open quantum systems using density matrices. The package can be obtained from the Computer Physics Communications library.

### 7.6 QSWalk.jl

QSWalk.jl is a Julia package that aims to simulate the time evolution of quantum stochastic walks on directed weighted graphs. The authors claim that is faster than QSWalk when used in large networks. Besides, it can be used for nonmoralizing evolution, which means that the evolution takes place on a directed acyclic graph and does not change to an evolution on the corresponding moral graph. The package can be downloaded from GitHub <https://github.com/iitis/QSWalk.jl>.



## Chapter 8

# Results

8.1 Walk on line

8.2 Walk on Grid

8.3 Walk on Hypercube

# Acknowledgements

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

# Bibliography

- [1] Ferenc Balazs and Sandor Imre. *Quantum computing and communications: an engineering approach*. Wiley, Hoboken, N.J., 2013. ISBN 9781118725474. URL <https://www.wiley.com/en-us/Quantum+Computing+and+Communications%3A+An+Engineering+Approach-p-9781118725474>.
- [2] Leo Breiman. *Probability*. Society for Industrial and Applied Mathematics, Philadelphia, 1992. ISBN 0898712963.
- [3] Mika Hirvensalo. *Quantum computing*. Springer, Berlin; New York, 2001. ISBN 9783540407041. URL <https://www.springer.com/gp/book/9783540407041>.
- [4] J Kempe. Quantum random walks: An introductory overview. *Contemporary Physics*, 44(4):307–327, 2003. DOI: 10.1080/00107151031000110776. URL <https://doi.org/10.1080/00107151031000110776>.
- [5] Renato Portugal. *Quantum Walks and Search Algorithms*. Quantum Science and Technology. Springer International Publishing : Imprint: Springer, Cham, 2nd ed. 2018 edition, 2018. ISBN 9783319978130.
- [6] Nemkin Viktória. Kvantum gráfolyongások. *Önálló laboratórium 1.*, 2021.
- [7] Feng Xia, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan, and Xiangjie Kong. Random walks: A review of algorithms and applications. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(2):95–107, 2020. DOI: 10.1109/TETCI.2019.2952908.

# Appendix

## A.1 Első függelék

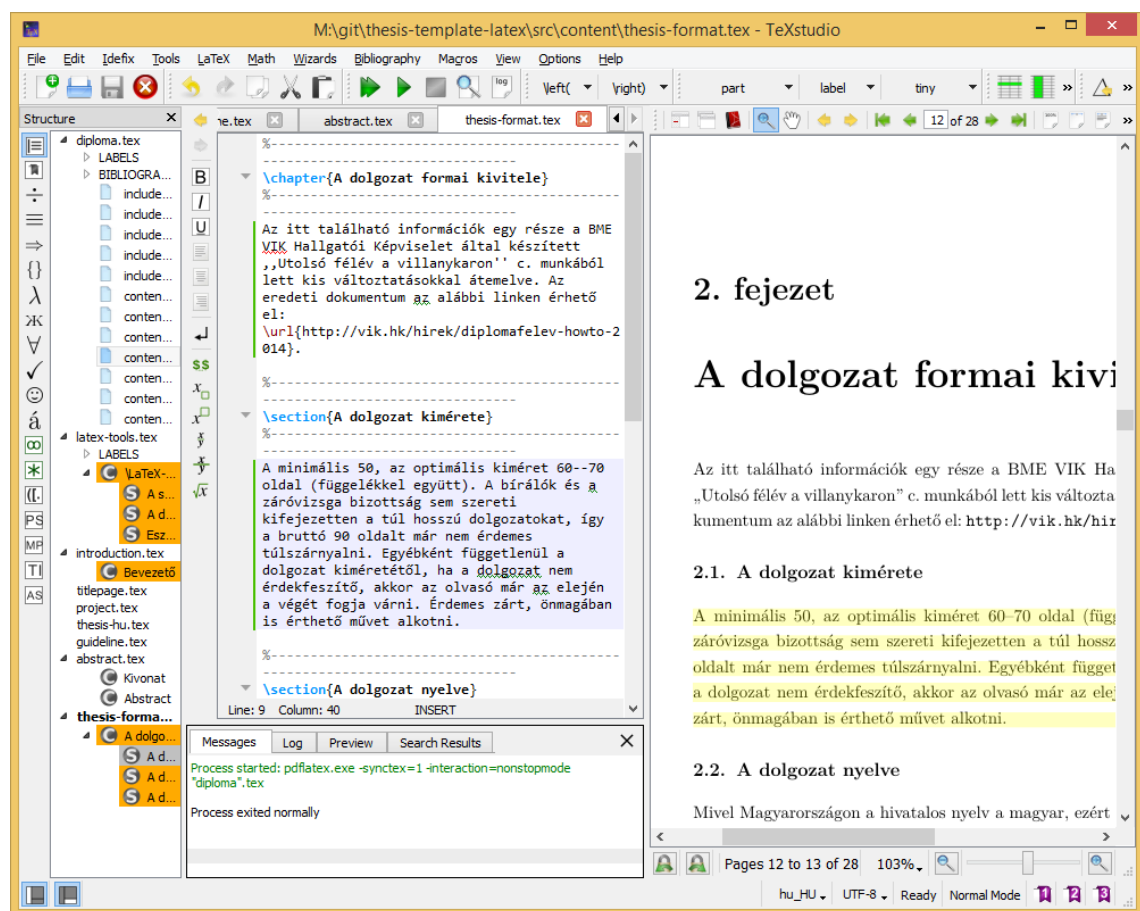


Figure A.1.1: A TeXstudio  $\text{\LaTeX}$ -szerkesztő.

## A.2 Második függelék

A Pitagorasz-tételből levezetve

$$c^2 = a^2 + b^2 = 42. \quad (\text{A.2.1})$$

A Faraday-indukciós törvényből levezetve

$$\text{rot } E = -\frac{dB}{dt} \quad \longrightarrow \quad U_i = \oint_{\mathbf{L}} \mathbf{E} d\mathbf{l} = -\frac{d}{dt} \int_A \mathbf{B} d\mathbf{a} = 42. \quad (\text{A.2.2})$$