Mintafelismerés kvantuminformatikai módszerrel

Készítette:

Leimeter Roland

Konzulens:

Dr. Imre Sándor

Tartalom

I. Bevezetés 3
II. A mintafelismerés hagyományos módszere 3
III. A kvantuminformatika alapjai 4
3.1. Első posztulátum (állapottér) 5
3.2. Második posztulátum (evolúció) 6
3.3. Harmadik posztulátum (mérés) 7
3.4. Negyedik posztulátum (összetett rendszerek) 8
3.5. Alkalmazott jelölésrendszer 8
3.6. Fizikai implementáció 9
3.7. Kapuk 10
3.8. Kvantum regiszterek 14
3.9. CNOT vagy XOR 15
IV. A mintafelismerő algoritmus 17
4.1 A memória-regiszter előállítása 20
4.2. Amplitúdó-erősítés 24
4.3. Amplitúdó-erősítés Grover-algoritmussal 30
4.4 Konklúzió 32

I. Bevezetés

A mintafelismerő algoritmusok napjaink intelligens rendszereinek szerves részét képezik. Az, hogy egy ilyen rendszer képes legyen a külvilág ingereinek érzékelésén túl azok felismerésére és kategorizálására, mára már alapvető elvárás - legyen szó biztonságtechnikáról, önműködő járművekről vagy akár közösségi oldalakról.

A megnövekedett alkalmazási igény következtében égető problémává vált az optimalizálás. A cél minél pontosabb azonosítás, minél rövidebb idő alatt. A két szempont általában fordítottan arányos egymással, így egyiken gyakran csak a másik rovására tudunk javítani. A korszerű szoftverfejlesztés lehetővé tette, hogy neurális hálókon keresztül a lehető legjobb idő-hiba aránnyal tudjunk dolgozni, azonban egy nagyobb adatbázis esetén még így is jelentős válaszidővel kell számolnunk.

A hardver oldali fejlesztés igénye már nagyon régen felmerült, és mivel a számítási kapacitás növelése - mint megoldandó probléma - az informatika teljes területére kiterjed, évről évre közelebb kerülünk alternatív megoldások realizálásához. Ilyen alternatív és újszerű koncepció a kvantummechanikai ismereteinket felhasználó kvantum-informatika területe.

E dolgozat során felvázolom a jelenleg alkalmazott mintafelismerés alapkoncepcióját, majd a szükséges kvantuminformatikai fogalmak bevezetése után részleteiben is bemutatom a mintafelismerés egy lehetséges kvantuminformatikai algoritmusát, illetve annak előnyeit és hátrányait.

II. A mintafelismerés hagyományos módszere

A mintafelismerés a 21. századra drasztikusan megnövekedett társadalmi és technológiai igény következtében, ma már önálló tudományágként említendő. Az algoritmusok az éppen aktuális bonyolultságú feladat függvényében a legegyszerűbb adatbázis-kezelő programoktól akár egészen összetett neurális-hálókig terjedhetnek.

Általánosságban elmondható, hogy az egyszerű algoritmusok általában sokáig futnak, vagy nagyobb hibával dolgoznak, az optimumot megcélzó neurális-hálók pedig esetenként akár igen bonyolult statisztikai modellek alapján működhetnek. Ehhez általában valamilyen kompromisszumot kell kötni, például az egyes neuronokra eső betölthető minták számára kell egy felső korlátot [1] megfogalmaznunk ahhoz, hogy a rendszer optimális működésre legyen képes. Az embert mintázó mesterséges intelligencia modellek építőblokkjai is gyakran csak nagyon kreatív programozási technikával bírhatóak megfelelő működésre.

A kvantum-informatika ezzel szemben relatíve egyszerű algoritmussal – de mellette bonyolult matematikával és fizikával – igyekszik hasonló viselkedést produkálni. A kvantumos mintafelismerés tipikusan ilyen eset, így elsőre komplikáltnak tűnhet, ám a formalitás mögött később láthatóvá válik, hogy az alapkoncepciók helyes alkalmazása során rendszerünk jóval egyszerűbbé, és intuitívabbá válik.

III. A kvantuminformatika alapjai

Ahhoz, hogy kvantuminformatikai algoritmusokról beszélni tudjunk, tisztázni kell a kvantumos rendszerek működési elvét, amihez szükségünk lesz a kvantum-informatika posztulátumaira. Mivel ezek a posztulátumok meglehetősen absztraktak, bevezetésképpen felvázolnék egy egyszerű példát.

Képzeljünk magunk elé egy pénzérmét. Amikor ezt az érmét feldobjuk, arra számítunk, hogy az esetek felében írás, illetve fej lesz. Ha két ilyen érmét dobunk fel egyszerre úgy, hogy a két eredmény együtt érdekel minket, akkor úgy tekintjük, hogy a kísérletünknek négy különböző kimenetele lehet, szintén azonos valószínűség-eloszlással, egyenként 25%-os bekövetkezési valószínűséggel. Mivel számít a sorrend, ilyenkor permutációról beszélhetünk. Anélkül, hogy túlságosan belebonyolódnánk az érmedobálásba beláthatjuk, hogy *n* db érme esetén a lehetséges permutációk száma 2^n, ahol az egyes esetek bekövetkezési valószínűsége $\frac{1}{2^n}$.

Tételezzük fel, hogy a fenti *n* érmés rendszerünkkel az International Space Station (ISS) fedélzetén találjuk magunkat, ahol a feldobott érmék nem esnek le. Ha magunk elé képzeljük az *n* db lebegő érmét, láthatjuk, hogy a rendszer egy köztes állapotot vett fel.

Nem tudjuk ugyanis eldönteni, hogy az egyes érmék milyen állapotban vannak éppen, hiszen *még nem estek le*. Ekkor tekinthetünk úgy a rendszerünkre, mintha az *egyszerre* vette volna fel az összes lehetséges állapotát, és e lehetséges állapotokat az egyes érmék *megmérésével* – jelen esetben földre helyezésével tudjuk csak szűkíteni. A rendszer állapotát akkor tekinthetjük meghatározottnak, amikor már egy érme sincs a levegőben, tehát minden érmét megmértünk.

A fenti gondolatkísérlet analóg a kvantum-informatika egyik legalapvetőbb koncepciójával, a szuperpozíció-elvével, miszerint egy kvantumbit képes felvenni mindkét állapotát egyazon időpillanatban, mely állapotokhoz valószínűségi együttható van rendelve. Ez annyit jelent, hogy egy kvantumbit (qubit) szuperpozíciós állapotban hasonlóan viselkedik, mint egy feldobott érme, azzal a különbséggel, hogy a 0 és az 1 eseményekhez tartozó bekövetkezési valószínűség nem egyezik szükségszerűen. Ez olyan, mintha az érme egyik oldala nehezebb lenne, így az gyakrabban érne földet azon az oldalán. Több kvantumbit – azaz egy kvantum regiszter esetén ez a viselkedés azt az érdekes állapotot eredményezi, hogy egy *n* bites regiszter párhuzamosan felveheti mind a 2^n állapotát, mégpedig úgy, hogy a számunkra érdekes permutációk megnövelt valószínűségi együtthatóval rendelkezzenek.

Jelen tudásunk birtokában nyugodtan kijelenthetjük, hogy érmék dobálgatása során gyakorlatilag soha nem fogunk ilyen viselkedést észlelni, azonban fotonokból vagy elektronokból álló kvantum rendszerek esetében olyannyira eltérő körülmények és szabadsági fokok uralkodnak, hogy a fenti példa határozottan megvalósítható.

Az imént szemléltetett példán túl számos egyéb előnye van a kvantumos rendszereknek, azonban ahhoz, hogy részleteiben is tárgyalhassuk a témakört, szükséges a kvantum-informatika posztulátumainak [2] ismerete, használata.

3.1. Első posztulátum (állapottér)

Bármely zárt fizikai rendszer pillanatnyi állapota jellemezhető egy ún. ***v*** *állapotvektorral – amely egységhosszú, komplex együtthatójú vektor egy V Hilbert-térben – azaz egy komplex lineáris vektortérben (állapottérben), ahol értelmezve van a belső szorzat.*

A zárt fizikai rendszerek egyik legegyszerűbb példájának egy kétdimenziós Hilbert-tér tekinthető. Ekkor a rendszer pillanatnyi állapota megadható egy $\mathbf{v} = [a,b]^T = a\mathbf{0} + b\mathbf{1}$

kétdimenziós vektorral, ahol $\mathbf{0} = [1,0]^T$ és $\mathbf{1} = [0,1]^T$ a V Hilbert-tér ortonormált bázisvektorai, illetve a, b $\in \mathbb{C}$,melyek között az alábbi összefüggés áll fenn:

$$|a|^2+|b|^2 = 1. \tag{1.1}$$

Ez utóbbi összefüggés formálisan **v** egységvektor jellegéből következik, fizikai értelemben pedig annyit jelent, hogy a

$$P(0) + P(1) = 1 \tag{1.2}$$

feltételnek mindig teljesülnie kell, miszerint:

$$P(0) = |a|^2, \text{ill. } P(1) = |b|^2, \tag{1.3}$$

vagyis az egyes bázisállapotokba való összeomlás[1] valószínűsége a *komplex valószínűségi amplitúdók*[2] abszolútérték-négyzetével egyenlő. Ezzel kibővítettük a klasszikus valószínűség-számítás értelmezési tartományát, így értelmezhetünk negatív, vagy akár komplex valószínűségi együtthatókat. A mérés során ettől függetlenül nem-negatív, valós számot kapunk valószínűségre (1.1) miatt.

3.2. Második posztulátum (evolúció)

Bármely zárt fizikai rendszer időbeli fejlődése jellemezhető unitér transzformációk sorozatával, melyek csak a kiindulási és a végállapot időpillanatától függenek.

Az előbb ismertetett rendszerre a második posztulátum az alábbiak szerint értelmezhető:

$$\boldsymbol{v}'(t_2) = U(t_1, t_2)\boldsymbol{v}(t_2), \text{ ahol } \mathbf{v} \in V$$

Lényegében mindez annyit jelent, hogy a kvantumállapotok közötti átjárás matematikailag unitér operátorokkal lehetséges, amely a bemeneti **v** állapotvektorból egy **v'** kimeneti állapotvektort állít elő, így értelmezhető a rendszer időbeli fejlődése. Ezen unitér operátor a hagyományos digitális technikában az elemi kombinációs hálózatok fogalmával egyenértékű, tehát amikor unitér transzformációk sorozatát

[1] Összeomlásnak nevezzük a szuperpozíciós állapot megszűnését, amelynek következtében a kvantum-bit fix 0 vagy 1 állapotba kerül.

[2] *a* és *b* együtthatókat a Schrödinger hullámfüggvényben betöltött szerepükből eredően valószínűségi amplitúdónak szokás nevezni.

alkalmazunk egy qubitre, hasonlóan beszélhetünk kapukról, ahogyan hagyományos digitális áramkörök esetében is.

3.3. Harmadik posztulátum (mérés)

Bármely kvantumos mérés jellemezhető mérési operátorok { M_m } sorozatával, ahol m jelöli a mérés lehetséges végkimeneteleit. Annak valószínűsége, hogy adott ***v*** *állapotvektor esetén a mérés eredménye "m" lesz a következőképpen számolható:*

$$P(m \mid \mathbf{v}) = \mathbf{v}^{\dagger} {M_m}^{\dagger} M_m \mathbf{v} ,$$

továbbá "m" mérési eredmény esetén a rendszer következő állapota:

$$\mathbf{v}' = \frac{M_m \mathbf{v}}{\sqrt{\mathbf{v}^{\dagger} {M_m}^{\dagger} M_m \mathbf{v}}} .$$

Mivel a klasszikus valószínűség-számítás szerint:

$$\sum_m P(m \mid \mathbf{v}) = \sum_m \mathbf{v}^{\dagger} {M_m}^{\dagger} M_m \mathbf{v} \equiv 1$$

a mérési operátoroknak ki kell elégíteniük a teljességi feltételt:

$$\sum_m {M_m}^{\dagger} M_m \equiv I .$$

A klasszikus és a kvantum világ között mérés segítségével biztosíthatunk átjárást, azonban a mérésnek van egy nagyon fontos következménye : visszafordíthatatlanul beavatkozunk a vizsgált rendszer működésébe, így a vizsgálandó kvantumállapot összeomlik[3]. Ebből következik, hogy a mérés egy irreverzibilis folyamat, ennélfogva nem unitér operátor, így ugyanazon kvantumbit többszöri mérése esetén az összes mérési eredmény meg fog egyezni az első mérés eredményével. Szintén az előbbiekből következik, hogy a mérést célszerű az algoritmusok legvégén elvégezni, ugyanis a mérés során az összes eddig elvégzett transzformáció *törlődik* a rendszerből, és csak azok hatása – pl. a generált valószínűség-eloszlás – lesz érzékelhető.

[3] Ez a viselkedés azzal magyarázható, hogy a kvantum világ a makrovilághoz képest annyira kicsiny, hogy jelen eszközeinkkel nem tudunk olyan mérési konstrukciót létrehozni, amely ne avatkozna be a rendszer működésébe.

3.4. Negyedik posztulátum (összetett rendszerek)

Legyen egy összetett fizikai rendszer állapottere W, amely V és Y állapotterekkel megadott alrendszerekből áll. Ekkor igaz, hogy $W = V \otimes Y$*, vagyis az összetett rendszer állapottere az egyes rendszerek állapottereiből képzett tenzoriális szorzattal állítható elő. Mivel definíció szerint* $\mathbf{v} \in V$ *és* $\mathbf{y} \in Y$*, az összetett rendszer pillanatnyi állapota* $\mathbf{w} = \mathbf{v} \otimes \mathbf{y}$*.*

A fenti posztulátum értelmében kvantum regiszter esetén az eredő **w** állapotvektort az egyes qubitek **v**,**y** tenzoriális szorzatából állíthatjuk elő, melynek következménye, hogy az összetett rendszer szuperpozíciós állapotban tartalmazni fogja az alrendszerek permutációit hasonlóképpen, ahogyan azt az érmés példával igyekeztem szemléltetni.

3.5. Alkalmazott jelölésrendszer

E négy posztulátum nem más, mint egy matematikai eszköztár, melynek segítségével egzakt formában fogalmazhatjuk meg a természet néhol igencsak non-intuitív viselkedését. Mivel a mérnöki gyakorlatban ezen absztrakt forma nehezen alkalmazható, bevezetünk egy másik, jóval gyakorlatiasabb megközelítést, amely kapukon, áramkörökön és folyamatábrákon keresztül szemlélteti a kvantum rendszerek működését.

Az első posztulátumban ismertetett

$$\boldsymbol{v} = [a, b]T = a\mathbf{0} + b\mathbf{1} \tag{5.1}$$

állapotvektor felírható az alábbi formában:

$$|v\rangle = a|0\rangle + b|1\rangle \tag{5.2}$$

Az (5.2) esetben alkalmazott jelölésrendszert Dirac jelölésnek nevezzük, kiejtése pedig – szintén az első posztulátum során tárgyalt – belső szorzat: $\langle v|v\rangle$ utáni bracketből[4] „bra” v és „ket” v -nek ejtjük. A latin $|v\rangle$ állapotvektor helyett $|\varphi\rangle$ vagy $|\psi\rangle$ görög betűk terjedtek el leginkább.

Egyszerű (egy qubites) rendszerekre az alábbi összefüggések érvényesek:

$$|\varphi\rangle = a|0\rangle + b|1\rangle = a\begin{bmatrix}1\\0\end{bmatrix} + b\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}a\\b\end{bmatrix} \tag{5.3}$$

[4] Bracket – magyarul: zárójel.

$$|\psi\rangle = c|0\rangle + d|1\rangle = c\begin{bmatrix}1\\0\end{bmatrix} + d\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}c\\d\end{bmatrix} \tag{5.4}$$

ekkor:

$$\langle\varphi|\psi\rangle = \begin{bmatrix}a^* & b^*\end{bmatrix}\begin{bmatrix}c\\d\end{bmatrix} = a^*c + b^*d = skalár \tag{5.5}$$

$$|\varphi\rangle\langle\psi| = \begin{bmatrix}a\\b\end{bmatrix}\begin{bmatrix}c^* & d^*\end{bmatrix} = \begin{bmatrix}ac^* & ad^*\\bc^* & ad^*\end{bmatrix} = mátrix\,. \tag{5.6}$$

Vagyais a skaláris és a diadikus szorzat a Hilbert-tereknek megfelelően definiálható, ahol (5.4)-hez tartozó:

$$\langle\psi| = c^*[1,0] + d^*[0,1] = [c^* \quad d^*] \tag{5.7}$$

„bra" vektor adjungálással állítható elő $|\psi\rangle$ -ből, azaz az együtthatók komplex konjugálása után transzponálni kell az állapotvektort.

Az (5.5) és (5.6) műveleteknek fizikai jelentősége van. Az (5.5) belső szorzat megadja φ -> ψ -be való összeomlásának valószínűségi amplitudóját, amely tetszőleges állapotok esetén nehezen értelmezhető, azonban ha $|\psi\rangle = |0\rangle$ bázisvektor, akkor a belső szorzat $|\varphi\rangle$ ismeretlen állapotvektor $|0\rangle$ -ba való leképeződésének valószínűségi amplitudóját adja eredményül, vagyis:

$$\langle\varphi|0\rangle = \begin{bmatrix}a^* & b^*\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = a^*1 + b^*0 = a^*, amely\ speciális\ esetben\ a \in \mathbb{R}\ esetén\ a^* = a.$$

Ezzel szemben a külső (diadikus) szorzat eredménye egy mátrix (6), amely egy olyan lineáris operátort hoz létre, amely a mérések egy csoportjánál kiválóan alkalmazható.

3.6. Fizikai implementáció

Hogy megfelelő képet alkothassunk a kvantum-rendszerek időbeli fejlődéséről (második posztulátum) érdemes megismerkedni néhány gyakori fizikai implementációval, nézzük meg tehát hogyan is működnek a qubitek metaszinten.

Először is le kell szögezni, hogy elméletileg qubitnek bármilyen anyag választható, amelyre érvényesek a kvantummechanika törvényei, tehát alkalmazható rá a Schrödinger-egyenlet, hullámfüggvénnyel jellemezhető, kvantummechanikai viselkedést produkál. Ami gyakorlatilag annyit jelent, hogy atomi szinten kell foglalkoznunk a

rendszerrel. Ahhoz, hogy mindez kvantum-informatikai szempontból is alkalmazható legyen, arra van szükség, hogy az adott részecskének legyen két jól elkülöníthető állapota.

Hogy egészen konkrét példát említsünk, a fotonnál a polarizáció ebből a szempontból tökéletes tulajdonság, mivel a vízszintes és függőlegesen polarizált fotonok méréssel nagyon jól megkülönböztethetők egymástől, tudunk tehát definiálni klasszikus 0-t és 1-et. Az elektron esetében ez a tulajdonság a spin, ahol szintén találunk két jól elkülöníthető állapotot. Ám az sem ritka manapság, hogy atommagokat használnak fel ilyen célra.

Az, hogy melyik, miért jobb a másiknál, még a jelen kérdéskörébe tartozik, és mivel még nem terjedtek el széleskörűen és kizárólagosan alkalmazott implementációk úgy tűnik, hogy egy ideig ez még a holnap problémája marad. A főbb szempontok ilyenkor a stabilitás, az irányíthatóság, és az információtovábbítás jellege.

E dolgozatban jellemzően fotonokkal szeretném demonstrálni a különböző viselkedési formákat, állapotváltozásokat. Mivel a fotonos rendszerek implementációja és a foton állapotának manipulációja meglehetősen intuitív, ennélfogva viszonylag könnyen értelmezhető, így célszerűen fotonokkal képzeljük el a kvantum áramkörök, kapuk megvalósítását.

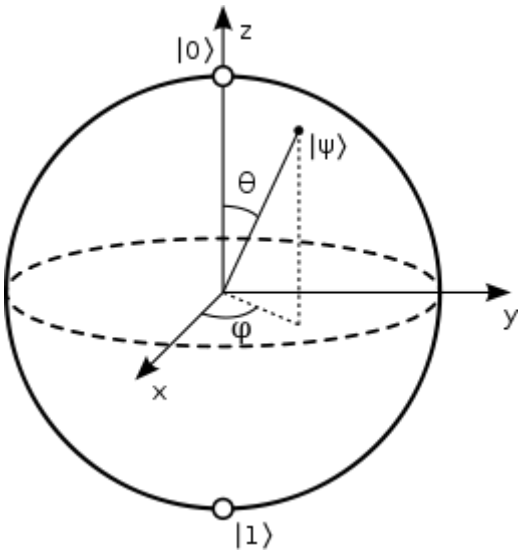
3.7. Kapuk

A második posztulátum értelmében a kvantum rendszerek fejlődése unitér operátorokkal írható le, amely gyakorlatilag az állapotvektor állapottérben való *elforgatásának* felel meg. Mivel $|\varphi\rangle$ általános állapotvektor két bázis lineáris kombinációjaként áll elő, továbbá tudjuk, hogy $|\varphi\rangle$ egységhosszú (első posztulátum), így az grafikusan ábrázolható egy 2D koordináta rendszerben.

Ekkor az állapotvektor egy egységsugarú kör kerülete mentén helyezkedik el, ahol az egyes bázisokhoz tartozó valószínűségi amplitudók $|\varphi\rangle$ koordinátái. Mivel $|\varphi\rangle$ egységvektor, ezért a $|0\rangle$ tengellyel bezárt szöge segítségével egyparaméteresen is definiálható:

$$|\varphi\rangle = \cos(\alpha)\,|0\rangle + \sin(\alpha)\,|1\rangle \qquad (7.1)$$

Ez az eset azonban speciális. Mivel az első posztulátum szerint a valószínűségi amplitudók komplex számok, így azok egyenként két paramétert igényelnek (Re{a}, Im{a} ill. Re{b}, Im{b}), tehát összesen négy paraméter kell egy állapotvektor leírásához, amelyhez a 2D koordináta rendszer nem elegendő. Általános esetben az állapottér egy egységsugarú gömb felületén, egy ún. Bloch-gömb felületén helyezkedik el.



1.ábra

Mivel az egységsugarúság szabadsági fokot vesz el a rendszertől, ezért a vektor mindössze két paraméter segítségével jellemezhető, polár koordináta-rendszerben az azimutális szöggel és a polárszöggel. Ennek részleteibe azonban e dolgozat keretein belül nem szükséges elmerülni, az állapotvektorok ily módon történő megadása csupán megkönnyíti a *forgatás* értelmezését.

A kapuk tehát olyan unitér operátorok, amelyek valamilyen fizikai kölcsönhatás során elforgatják az állapotvektort egy másik helyzetbe, aminek következtében változik a valószínűség-eloszlás. Fotonoknál ez általában megfelelő anyagon való keresztülvezetést jelent, ahol paraméterként szerepelhet az, hogy mennyi időt tölt a foton az adott anyagban.

A kvantum-informatika által leggyakrabban alkalmazott kapuk [2] közé tartozik a Pauli-hármas, a Hadamard-kapu, a CNOT (vagy XOR)[5], illetve ezek n-bites általánosításai, az nCNOT és az nXOR.

A Pauli-X, vagy bit-flip kapu tulajdonképpen egy inverter:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{7.2}$$

$$|\psi\rangle = X|\varphi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = b|0\rangle + a|1\rangle \tag{7.3}$$

Megfigyelhetjük, hogy ennek hatására a valószínűségi amplitudók kicserélődnek.

Pauli-Z, vagy phase-flip kapu, amely nevéből adódóan a fázist cseréli meg:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{7.4}$$

$$|\psi\rangle = Z|\varphi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = a|0\rangle - b|1\rangle \tag{7.5}$$

Ez a transzformáció önmagában nem érzékelhető, hiszen amikor mérést hajtunk végre, akkor az az egyes állapotok bekövetkezési valószínűsége $|a|^2$ és $|b|^2$, így a mérés során elveszítjük az előjelet. Ennek ellenére nagyon is hasznos ez a kapu, ugyanis számos algoritmus alapját képezi, mint például a szupersűrű kódolás vagy az ismeretlen kvantumállapot teleportálása. Ezek azonban jelen dolgozat témakörén kívül esnek, így nem is részletezném őket.

A harmadik Pauli-kapu a Pauli-Y, amelyről szintén elmondható az, ami a Pauli-Z-ről. Mérés során nem tudjuk megkülönböztetni az X és az Y kapuk hatását, azonban algoritmusok során effektíven alkalmazható.

$$Y = \begin{bmatrix} 0 & -j \\ j & 0 \end{bmatrix} \tag{7.6}$$

$$|\psi\rangle = Y|\varphi\rangle = \begin{bmatrix} 0 & -j \\ j & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = -jb|0\rangle + ja|1\rangle \tag{7.7}$$

[5] Controlled NOT, ami klasszikus bemenetek esetén ekvivalens az XOR-ral.

A három Pauli-kapu tehát a Bloch-gömb három tengelyére tükröz. Itt megjegyezném, hogy a gyakorlatban bármely tengely körül tetszőleges szöggel tudunk forgatni, így értelmezhetőek még további forgató transzformációk, amelyek fizikailag meg is valósíthatóak.

A fentiektől jelentősen eltér a Hadamard-kapu, amely szinte kivétel nélkül minden kvantum-algoritmus nélkülözhetetlen építőeleme, ugyanis a Hadamard-transzformáció az, aminek a segítségével szuperpozíciós állapotokat lehet előállítani. Gyakorlati megvalósítása foton alapú rendszer esetén egy féligáteresztő tükör.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{7.8}$$

$$|\psi\rangle = H|\varphi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \frac{a+b}{\sqrt{2}}|0\rangle + \frac{a-b}{\sqrt{2}}|1\rangle \tag{7.9}$$

Kiemelkedő gyakorlati jelentősége miatt e kapunak külön számon szokás tartani $|0\rangle$ *és* $|1\rangle$ bemeneti vektorokra gyakorolt hatását, amely:

$$|\varphi\rangle = H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{7.10}$$

$$|\psi\rangle = H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{7.11}$$

Ez az eset pontosan a feldobott érme esete, ugyanis a valószínűségi amplitudók ebben az esetben $a = b = \pm\frac{1}{\sqrt{2}}$, melyeknek abszolútérték-négyzete 0.5, vagyis az egyes állapotokba való összeomlás valószínűsége egyaránt 50-50%.

A Hadamard-transzformáció példáján könnyen beláthatjuk, hogy az unitér operátorok definíciószerűen önadjungáltak, vagyis $U^*U = UU^* = I$, amelyből valós elemek esetén következik, hogy $U^{-1} = U$. Ez viszont azt jelenti, hogy ha kétszer alkalmazzuk a Hadamard-kaput (ami minden valós elemű kapura igaz) visszakapjuk az eredeti kvantumállapotot. Mivel a rendszerünk lineáris, a mátrix-vektor szorzás ismételt elvégzése helyett megtehetjük, hogy behelyettesítjük a már meglévő $H|0\rangle$ *és* $H|1\rangle$

megoldásokat $|0\rangle$ *és* $|1\rangle$ helyére, így a Hadamard-transzformációt (7.10)[6]-re ismételten elvégezve következőt kapjuk:

$$H|\varphi\rangle = HH|0\rangle = \frac{H|0\rangle + H|1\rangle}{\sqrt{2}} = \frac{\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \frac{|0\rangle - |1\rangle}{\sqrt{2}}}{\sqrt{2}} = |0\rangle \qquad (7.12)$$

3.8. Kvantum regiszterek

A negyedik posztulátum értelmében több qubites rendszer esetén az eredő állapotvektor tenzoriális szorzattal állítható elő.

Legyen egy kétqubites kvantum regiszterünk, rendre:

$$|\varphi_1\rangle = a|0\rangle + b|1\rangle \text{ és } |\varphi_2\rangle = c|0\rangle + d|1\rangle .$$

Ekkor: $$|\varphi\rangle = |\varphi_1\rangle \otimes |\varphi_2\rangle = ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + ad\,|11\rangle . \qquad (8.1)$$

Vagyis a tenzoriális szorzat által megnöveltük az állapotvektor dimenzióját, így az eredő állapothoz már nem kettő, hanem négy ortonormált bázisvektor tartozik. Ennek megfelelően az egyes bázisokba való összeomlás valószínűsége is az új koordináta-rendszerben kezelendő, tehát a $|00\rangle$ – hoz tartozó valószínűségi-amplitudó *„ac”*, és így tovább.

Ha visszaemlékezünk a dolgozat elején felvázolt érmedobálós példára, rájöhetünk, hogy pontosan ezt a viselkedést igyekezett reprezentálni. Mi történik ugyanis, ha a regiszterünket kiterjesztjük „n” qubit nagyságúra? A tenzorszorzás következtében az ortonormált bázisvektorok száma éppen 2^n lesz, és minden egyes bázishoz tartozni fog egy komplex valószínűségi-amplitudó, amelyből az első posztulátum szerint számolható az adott permutáció[7] bekövetkezési valószínűsége.

Ez azonban még csak az általános eset, ismeretlen valószínűség-eloszlással. Az érmés példában pedig feltétel volt, hogy az egyes permutációk egyenletes-eloszlásban jelenjenek meg. Ilyen kvantumállapot egyszerűen előállítható, ugyanis (7.10)-nél már sikerült pénzérme viselkedést reprodukáló qubitet létrehozni. Annyival kell csupán megtoldanunk a rendszert, hogy kombináljuk (7.10)-et és (8.1)-et, tehát kiindulunk egy

[6] Az összefüggés hasonlóan belátható (7.11)-re is.
[7] Jelen esetben ez egy „n” elemű bitsorozat.

n-bites $|\varphi_0\rangle = |0_0 \dots 0_n\rangle$ regiszterből, amelynek bitjeire alkalmazzuk a Hadamard-transzformációt, majd elvégezzük a tenzorszorzást. Mivel lineáris a rendszer, ezért itt is megtehetjük azt, hogy ahelyett, hogy $2^n \times 2^n$ -es unitér operátorral[8] szoroznánk be a 2^n dimenziós vektorunkat, inkább alkalmazzuk a szuperpozíció elvét, és bitenként végezzük el a Hadamard-transzformációt, a tenzorszorzás szabályait figyelembe véve.

Ezek után már viszonylag egyszerűen levezethető az n-qubites Hadamard-transzformáció utáni állapot:

$$|\varphi\rangle = H^{\otimes n} |\varphi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle, \tag{8.2}$$

ahol $|i\rangle$ bináris vektorként értendő.

Ekkor áll tehát elő az az állapot, amikor mindegyik permutáció azonos súllyal jelenik meg a rendszerben. Ennek az a hatalmas előnye, hogy kisebb módosításokkal egy n-qubites regiszterben 2^n-bitnyi adat tárolható, vagyis a tárolókapacitás a klasszikus rendszerekkel ellentétben nem egyenes arányban, hanem exponenciálisan nő a bitek számához képest. Ezt ki fogjuk használni a mintafelismerő algoritmus során.

3.9. CNOT vagy XOR

Ahogy az elektronikában is, úgy a kvantum-informatikában is nagyon fontos szerepet játszik a feltételesség, és az irányítás. A megfelelő logikai szabályozhatóság érdekében elkerülhetetlen, hogy kvantumosan is meg tudjunk fogalmazni klasszikus logikai függvényeket. Ugyanúgy definiálnunk kell az ÉS, a VAGY, és az XOR kapukat, mint klasszikus esetben, melyeket fizikailag meglehetősen kreatívan kell megoldani, hiszen gondoljunk csak bele, hogy fotonok esetében egy XOR-hoz olyan összeállítás szükséges, ahova bemegy két foton, majd kijön mindkettő úgy, hogy az egyik közben még invertálódott is (feltéve persze, ha a control qubit 1-es állapotban került a rendszerbe), ami ugyebár az XOR definíciója. Az, hogy miért van 2 output, amikor az XOR tipikusan 2 input-1 output kapu (mint a legtöbb elemi logikai függvény), arra az unitér feltétel ad magyarázatot, hiszen csak úgy lehet mindkét irányban értelmezhető a transzformáció, ha ugyanannyi foton megy be, mint amennyi kijön. Ennek következménye, hogy tudunk

[8] A transzformációs-mátrixok mindig követik dimenziójukban az állapotvektorokat, tehát míg az egy qubites – 2 dimenziós – vektorhoz 2x2-es unitér operátor tartozik, addig a két qubites regiszterhez már 4x4-es, így egy „n” qubites kvantum regiszterhez 2^nx2^n –es mátrix tartozna.

inverz XOR műveletet végrehajtani, amit a klasszikus informatikában nem tudunk értelmezni, ellenben kvantumos rendszereknél tudatosan alkalmazott eljárás.

A 2 qubites CNOT tehát:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{9.1}$$

$$CNOT: \ |x\rangle|y\rangle \text{ -> } |x\rangle|y \oplus x\rangle \tag{9.2}$$

A (9.2) szerinti felírás azonban csak klasszikus állapotokra vonatkozik, tehát $|x\rangle$ és $|y\rangle$ csak $|0\rangle$ $vagy$ $|1\rangle$ lehet. Ekkor a CNOT kapu XOR-ként üzemel.

Ettől függetlenül (9.1) unitér transzformációt tetszőleges $|\varphi\rangle$ és $|\psi\rangle$ állapotokra is értelmezhetjük, azonban ahogyan az a mérnöki gyakorlatban általában lenni szokott, vannak speciális esetek, amelyek kiemelkedő jelentőséggel bírnak. Ilyen eset a következő:

Legyek $|\varphi_1\rangle$ és $|\varphi_2\rangle$ bemeneteink a következők:

$$\begin{aligned} |\varphi_1\rangle &= a|0\rangle + b|1\rangle \\ |\varphi_2\rangle &= |0\rangle \end{aligned} \tag{9.3}$$

Az eredő bemenet:

$$|\varphi\rangle = \ |\varphi_1\rangle \otimes |\varphi_2\rangle = \ a|00\rangle + b|10\rangle = a\begin{bmatrix}1\\0\\0\\0\end{bmatrix} + 0\begin{bmatrix}0\\1\\0\\0\end{bmatrix} + b\begin{bmatrix}0\\0\\1\\0\end{bmatrix} + 0\begin{bmatrix}0\\0\\0\\1\end{bmatrix} = \begin{bmatrix}a\\0\\b\\0\end{bmatrix}$$

Alkalmazva a CNOT-kaput:

$$CNOT|\varphi\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix}a\\0\\b\\0\end{bmatrix} = \ a|00\rangle + b|11\rangle \tag{9.4}$$

Kimeneti állapotvektorként tehát $|\varphi'\rangle = \ a|00\rangle + b|11\rangle$.

Az, hogy ilyen állapot előállítható, óriási jelentőséggel bír. Belátható ugyanis, hogy ekkor a regiszterünk nem bontható le 2 különálló qubitre úgy, ahogyan azt összeraktuk. Vagyis nincs két olyan $|\psi_1\rangle$ *és* $|\psi_2\rangle$ állapotvektor, amely kielégítené az:

$$|\psi_1\rangle \otimes |\psi_2\rangle = a|00\rangle + b|11\rangle \text{ egyenletet.} \quad (9.5)$$

Mit jelent ez a gyakorlatban?

Ha megmérjük $|\varphi'\rangle$ -t, azt látjuk, hogy a rendszerünk $|a|^2$ valószínűséggel zuhan $|00\rangle$ állapotba, és $|b|^2$ valószínűséggel $|11\rangle$ -be. Az egyes bitek állapotai között tehát fennáll egy kikötés, mégpedig az, hogy a két qubit mérés során nem adhat különböző eredményt. Ha lemérjük az egyiket, az determinálja a másik állapotát is és viszont. Ami igazán érdekes ebben, hogy ez akkor is igaz marad, miután a két qubitet térben eltávolítottuk egymástól. Ezt már számos kísérlet során bizonyították, de ennél több is igaz: ha még a mérés előtt (de már a térbeli szeparálást követően) transzformációt hajtunk végre az egyik qubiten, az kihat a másikra is. Erre a mai napig nem született általánosan elfogadható magyarázat, csupán annyit tudunk, hogy működik. Ezt a viselkedést kvantum-összefonódásnak nevezzük, és a kvantum-kommunikáció területét gazdagítja. Bár közvetlen információ-átvitelre nem alkalmas a módszer (hiszen a mérés során az összeomlás miatt megszakad a kapcsolat a két qubit között, mérés nélkül pedig nem tudunk információt kivonni a rendszerből), ennek ellenére nagy hasznát lehet venni a gyakorlatban – például kiválóan alkalmas elméletileg is lehallgathatatlan kulcsszétosztás megvalósítására.

Mivel a kvantum-összefonódásnak ezt az alkalmazását mintafelismerő algoritmusunk során nem használjuk, csak érdekességképpen szerepel e dolgozatban, azonban kiemelkedő jelentősége miatt e pár mondat erejéig mindenképpen helytálló alkalmazási példa a CNOT-ra, amit viszont aktívan használni fogunk a későbbiek folyamán.

IV. A mintafelismerő algoritmus

Most, hogy az alapvető kvantuminformatikai fogalmak tisztázásra kerültek, rátérhetünk a dolgozat tárgyára, a mintafelismerő algoritmusra.

A klasszikus módszer legnagyobb gyengesége, hogy az adatbázisba maximálisan betölthető minták száma a neuronok számától lineárisan függ, így a tárolókapacitás erősen korlátozott. Bár a rendszer a kapacitás optimum pontja felett is működőképes marad, cserébe azt az árat fizetjük, hogy az előhívás során hamis mintákat ismerhet fel a rendszer, vagyis olyan patternek kerülhetnek a kimenetre, amelyek soha sem szerepeltek az adatbázisban [1].

A kvantumos mintafelismerés esetében a szuperpozíciós állapotnak köszönhetően a tárolókapacitás n qubit esetében $p_{max} = 2^n$. Vagyis ebben az esetben a bitek számával exponenciálisan nő a rendszerbe felvehető minták száma, így a tárolókapacitás problémája elhanyagolhatóvá válik. Emellett – ahogyan azt később látni fogjuk - a felismerési fázisban definíció szerint csak és kizárólag az általunk betöltött minták egyikét kaphatjuk vissza, ugyanis a mintáink egy szuperpozíciós-állapotban lévő kvantum regiszterben fognak elhelyezkedni, aminek természetes viselkedése, hogy a regiszter tartalmának megmérésekor csak a bázisállapotok egyikébe omolhat össze a rendszer (a valószínűségi-amplitúdók függvényében). Mi csupán annyit teszünk, hogy módosítjuk ezeket a valószínűségi-amplitúdókat, ún. súlyponteltolást végzünk, aminek következtében a valószínűség-eloszlás ugyan megváltozik, de ettől még a rendszer továbbra is csak az általunk betöltött bázisállapotokat tartalmazhatja[9].

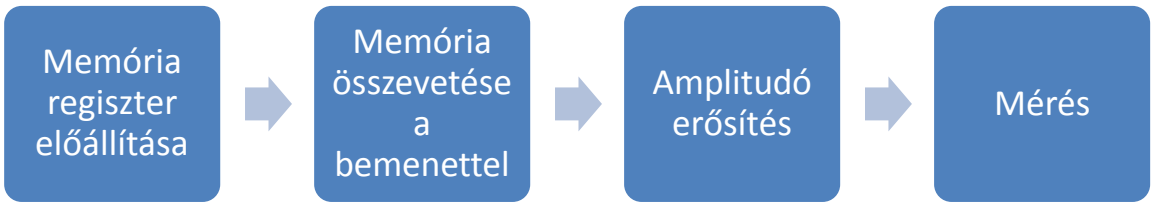
Az algoritmus kétféleképpen kerül tárgyalásra. Az elején több regisztert, de egyszerűbb kapukat használunk fel, majd az optimális működés érdekében megfogalmazzuk azokat az unitér transzformációkat, melyek helyettesíthetik a kiegészítő regiszterek szerepét, vagyis képesek a rendszert közvetlenül a kiindulási állapotból a végállapotba transzformálni. Ez lehetséges, mivel a regiszterek megfelelő állapotba hozása kvantuminformatikai szemszögből nézve egy 2^n dimenziós Hilbert-térben való forgatással ekvivalens, amelyhez elegendő információ[10] birtokában *legyártható* az adott forgatást elvégző unitér operátor.

[9]Hogy egészen pontosak legyünk, a regiszter valójában minden bázisállapotot tartalmazni fog, azonban csak azon bázisokhoz fog nullától különböző amplitúdó tartozni, melyeket mi magunk töltöttünk fel a rendszerbe, így előbbiekre úgy tekintünk, mintha nem léteznének.

[10] Elegendő információ alatt azt értjük, hogy ismerjük a kezdeti- és végállapot legalább paraméteresen.

A következőkben [1] alapján mutatom be, hogyan is néz ki egy kvantum mintafelismerő algoritmus. A levezetés során kiegészítő magyarázattal, ill. lehetőleg példával szemléltetem az egyes lépések közötti állapotokat, és azok jelentőségét.

A folyamat fő lépései a következők:



2.ábra

Egy kvantumos mintafelismerő algoritmus esetében egyáltalán nem triviális, hogyan jutunk hozzá az adatbázisunkhoz. Egy dolog azonban minden esetben bizonyos: akkor működik legoptimálisabban a rendszerünk, ha maximalizáljuk a tárolókapacitást. Ez esetünkben akkor teljesül, ha szuperpozíciós állapotba helyezzünk a regisztert, így n biten 2^n felső korláttal helyezhetünk el mintákat. Nem lehet eléggé hangsúlyozni a megoldás praktikusságát. Ez hosszú távon (feltéve, hogy a kvantumáramkörök helyigénye nem haladja meg a félvezető-áramkörökét) exponenciális méretcsökkenést eredményez - vagy fordított esetben - ugyanakkora helyen drasztikusan több információt tárolhatunk majd.

Miután előállt az adatbázis, valamilyen módon el kell azt érni, hogy az input és a minták közötti korrelációt bele tudjuk olvasztani az egyes mintákhoz tartozó valószínűségi amplitudókba (mivel azok alapesetben egyenletes amplitúdóval oszlanak el a rendszerben). Erre is több jó megoldás születhet, az általunk vizsgált esetben a két bitsorozat közötti Hamming-távolságot fogjuk felhasználni. El kell tehát érnünk, hogy az input és az egyes minták közötti Hamming-távolság, mint súlytényező jelenjen meg az amplitudókban, mindezt úgy, hogy megtartsuk a transzformációk unitér jellegét.

Ahhoz, hogy a valószínűség-eloszlás kellően stabil legyen[11] az amplitúdó-erősítést iterálnunk kell, amelynek effektív értéke a Grover-algoritmus mintájára

[11] Kellően stabil alatt azt értjük, hogy a kimeneten már „megbízható” valószínűséggel jelenik meg a helyes érték. Az, hogy ez pontosan mit takar, mérnöki kompromisszum kérdése.

meghatározható. Ezután nincs más dolgunk, mint megmérni a regiszterünk tartalmát, amely ekkorra már (optimális esetben) kiugróan magas valószínűséggel tartalmazza az inputhoz legközelebb álló mintát. Ekkor a memória megsemmisül, tartalma pedig megeggyezik a mért értékkel.

Ez sajnos azt vonja magával, hogy a memóriát minden méréshez újra létre kell hoznunk, ami tulajdonképpen abszolút vállalható ár a párhuzamos műveletvégrehajtásért cserébe. A nehézséget nem is ez okozza, hanem a kvantum-informatika azon tétele, miszerint matematikailag is lehetetlen olyan unitér transzformáció létrehozása, amely tetszőleges kvantum-állapotokat másolni tudna. Ezt a megkötést „no-cloning" tételnek nevezik, amelyek bizonyítástól most eltekintünk (bővebben ld. [2]).

Szerencsére nem vagyunk rászorulva, hogy a folyamat legelejéről kelljen indulnunk – ami nagyon jó hír, hiszen így csak n bitet kell valahogyan lemásolnunk a szuperpozíciós állapot újragenerálása helyett – amit egy ún. valószínűségi másolóval [3] valósíthatunk meg, ami viszont már nem mond ellent a „no-cloning" tételnek. Ez tulajdonképpen egy komprumisszum, amivel elérhetjük, hogy néhány próbálkozással előállítsuk a memória tökéletes másolatát, ami megfelelő módszertannal mindig effektívebb lesz, mintha minden esetben teljesen előről kezdenénk. A valószínűségi másoló lényege, hogy [2], [3]-ban szintén részletesen tárgyalt POVM mérés segítségével egy jelzőbit segítségével információt szerzünk a másolat tökéletességéről anélkül, hogy magát a másolat-regisztert megmérnénk (ami ugyebár egyenlő lenne annak tönkretételével).

Adott tehát a terv. Lássuk, hogyan jutunk el kezdeti állapotból a végállapotba.

4.1 A memória-regiszter előállítása

Legyen a kezdeti állapotunk:

$$|\psi_0^1\rangle = |p_1^1, \dots, p_n^1; 01; 0_1, \dots, 0_n\rangle \tag{1.1}$$

Ez három regiszter tenzorszorzata, ahol az egyes regiszterek az őket alkotó qubitek tenzorszorzataiból állnak elő. Ez elsőre bonyolultnak tűnhet, de mivel valamennyi qubit klasszikus állapotban van, a tenzorszorzás csupán a bitek egymás mellé helyezését jelenti.

Három regiszterünk tehát: $|p^k\rangle_n \otimes |u\rangle_2 \otimes |m\rangle_n$, ahol $|p^k\rangle$ a k-adik betöltendő minta, $|u\rangle$ a segédregiszter, $|m\rangle$ a memóriaregiszer, az alsó index pedig (csak ebben a példában, a szemléletesség kedvéért) az egyes regisztereket alkotó bitek száma.

Láthatjuk, hogy $|u\rangle$ *és* $|m\rangle$ csupa nulla értékkel lettek inicializálva. A $|u\rangle$ segédregiszter szerepe, hogy annak u_2 bitjén fogjuk a Hadamart-traszformációt végrehajtani, aminek segítségével a rendszert szuperpozíciós állapotba helyezhetjük, és az így szeparált memóriában külön kezelhetjük a már eltárolt ($|u_2\rangle = |0\rangle$), és az éppen feldolgozandó ($|u_2\rangle = |1\rangle$) mintákat. Ez lehetséges, hiszen a már ismert módon:

$$H|u_2\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \; előállítható. \tag{1.2}$$

A Hadamard mellett alkalmazni fogjuk még a Pauli-kapukat, az XOR-t(CNOT), illetve ez utóbbinak több controlbites általánosítását. Ezek mellé bevezetésre kerül egy CNOT-hoz hasonló, 2 qubites CS^i kapu, amely a következőképp áll elő:

$$CS^i = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes S^i = diag(I, S^i) \tag{1.3}$$

ahol (1.4)

$$S^i = \begin{bmatrix} \sqrt{\frac{i-1}{i}} & \frac{1}{\sqrt{i}} \\ -\frac{1}{\sqrt{i}} & \sqrt{\frac{i-1}{i}} \end{bmatrix}.$$

Ez az unitér kapu az i-edik pattern betöltésekor úgy alakítja memóriaregisztert, hogy az minden lépés esetére matematikailag (és ezáltal fizikailag is) helyes amplitúdó-eloszlást biztosítson. A kapu működését a későbbiek során egy példán keresztül is demonstrálni fogom.

A memória-összeállító algoritmus célja tehát világos:

$$|\psi_0^1\rangle = \; |p_1^1, \dots, p_n^1; 01; 0_1, \dots, 0_n\rangle \rightarrow |m\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^{p} |p^k\rangle \tag{1.5}$$

Ez nem más, mint az egyes minták szuperpozíciós állapotban, egyenletes eloszlással, n biten, tehát éppen a fentebb meghatározott igényünk matematikai megfogalmazása. Ahogyan azt az elején leszögeztük, a feldolgozást végző taghoz $|u_2\rangle = |1\rangle$ állapot

tartozik, így az algoritmus elején ennek megfelelően inicializáltuk $|u\rangle$ –t (hiszen csak feldolgozó tagunk van).

Első lépésként alkalmazzuk (1.1)-re az alábbi műveletet:

$$|\psi_1^i\rangle = \prod_{j=1}^{n} 2XOR_{p_j^i u_2 m_j} |\psi_0^i\rangle \tag{1.6}$$

Ahol $|\psi_1^i\rangle$ –ben az alsó index jelzi, hol tartunk az algoritmusban, a felső index pedig az éppen betöltés alatt álló minta indexével egyezik meg. A művelet nagyon egyszerű: csupán annyit teszünk, hogy 2XOR-t, azaz duplán kontrollált negációt hajtunk végre m_j qubitre, vagyis: ha $p_j^i = u_2 = 1 \rightarrow NOTm_j$. Ha ezt a műveletet mind az n bitre elvégezzük, (1.6)-al ekvivalens eredményre jutunk. Mivel m_j itt minden esetben $|0\rangle$, az XOR művelet átmásolja p_j^i tartalmát m_j -be. A kiegészítő feltétel arra szolgál, hogy ez a másolás csak akkor történjen meg, ha a *feldolgozó tagban* vagyunk, vagyis a fenti feltétel valóban teljesül.[12] A legelső minta betöltésénél természetesen még *csak* feldolgozó tagunk van, mivel még nem alkalmaztunk szuperpozíciós állapotot generáló kaput, azonban az ezt követő minták esetén már számolni kell az előzőleg eltárolt állapotokkal is.

$$|\psi_2^i\rangle = \prod_{j=1}^{n} NOT_{m_j} XOR_{p_j^i m_j} |\psi_1^i\rangle \tag{1.7}$$

A második lépés egy szintén bitenkénti művelet, nevezetesen egy ekvivalencia kapu(jobbról balra applikálva az egyes operátorokat az előző állapotra). Itt a műveletet mindkét tagra elvégezzük, aminek az a következménye, hogy az $|u_2\rangle = |1\rangle$-es állapothoz tartozó tagnál a memória-regiszter minden bitje $|1\rangle$ -re áll be, amit a következő lépésben ki is fogunk használni.

$$|\psi_3^i\rangle = \prod_{j=1}^{n} nXOR_{m_1 \ldots m_n u_1} |\psi_2^i\rangle \tag{1.8}$$

[12] A tároló tagban ugyanis tipikusan nem-nulla értékű memóriabitek szerepelnek, mert ott már egy előző körben ugyanígy elvégeztük ezt a lépéssorozatot.

A harmadik lépés egy nXOR művelet, amely akkor és csak akkor negálja u_1-et, ha az összes memóriabit |1⟩ értékre állt be. Ez (1.6) és (1.7) együttes hatása miatt pontosan csak a feldolgozó tagra fog teljesülni, mivel a tároló tagnál definíciószerűen $|u_2\rangle = |0\rangle$, így arra (1.6) nem teljesül, (1.7) önmagában pedig csak egy korábbi mintát fog összehasonlítani az éppen betöltendő mintával. Így a tároló tagnál a (1.8)-as bit-flip csak abban a speciális esetben valósulna meg, ha többször töltenénk be ugyanazon mintát, amire azonban egyrészt semmi szükség, másrészt anomáliát okozhat.

$$|\psi_4^i\rangle = \prod_{j=1}^{n} CS_{u_1u_2}^{p+1-i} |\psi_3^i\rangle \tag{1.9}$$

A negyedik lépés során alkalmazzuk (1.3)-nál ismertetett kaput, amely a CNOT és a Hadamard-kapuk tulajdonságait egyesíti, így az felfogható egy CHadamard-kapuként, annyi megkötéssel, hogy a H operátortól eltérően itt a súlyozás aszimmetrikus.[13] Ez a lépés a tároló-algoritmus központi elemének tekinthető, ugyanis itt történik meg ténylegesen a szuperpozíciós állapot „bővítése", itt ágazik ketté a tároló és a feldolgozó tag.

$$\begin{aligned} |\psi_5^i\rangle &= \prod_{j=1}^{n} nXOR_{m_1 \dots m_n u_1} |\psi_4^i\rangle \\ |\psi_6^i\rangle &= \prod_{j=1}^{n} XOR_{p_j^i m_j} NOT_{m_j} |\psi_5^i\rangle \end{aligned} \tag{1.10}$$

Miután megtörtént a tárolás, visszaálltjuk a megfelelő állapotokat, elvégezzük az invertálását (1.8)-nak és (1.7)-nek. A fenti művelet visszaállítja u_1 eredeti állapotát, illetve a memóriát is visszaforgatja a megfelelő állapotba. Ezek után a következőt kapjuk:

$$|\psi_6^i\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^{i} |p^i; 00; p^k\rangle + \sqrt{\frac{p-i}{p}} |p^i; 01; p^i\rangle, \tag{1.11}$$

amelyre ha ismét alkalmazzuk (1.6)-ot, visszaállítjuk a feldolgozó-tag memória-

[13] A Hadamard-kapu esetében $\frac{1}{\sqrt{2}}$ a szorzótényező, a CS^i -nél azonban ez az együttható egyrészt lépésenként változik, másrészt fordítottan arányosan hat |0⟩ és |1⟩ -re.

regiszterének eredeti (nulla) állapotát, így az képes újabb minták befogadására, miközben a tároló-tag esetében megtartjuk az eddig felépített állapotot.

$$|\psi_7^i\rangle = \prod_{j=1}^{n} 2XOR_{p_j^i u_2 m_j} |\psi_6^i\rangle \tag{1.12}$$

Az algoritmust minden betöltendő mintára lefuttatva előáll az $|m\rangle$ regiszter megcélzott (1.5) állapota.

4.2. Amplitúdó-erősítés

Mostmár feltételezhetjük, hogy a kiindulási állapotunk:

$$|m\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^{p} |p^k\rangle \tag{2.1}$$

A felismerés végrehajtásához szükségünk lesz az $|i\rangle$ input regiszterre, és egy ún. $|c\rangle$ control-regiszterre. A kiindulási állapot tehát ebben az esetben:

$$|\psi_0\rangle = |i\rangle_n \otimes |m\rangle_n \otimes |c\rangle_b = \frac{1}{\sqrt{p}} \sum_{k=1}^{p} |i;\ p^k;\ 0_1, \dots, 0_b\rangle \tag{2.2}$$

Alkalmazzuk a control-regiszter 0_1 bitjére a Hadamard-transzformációt:

$$|\psi_1\rangle = \frac{1}{\sqrt{2p}} \sum_{k=1}^{p} |i;\ p^k;\ 0_1, \dots, 0_b\rangle + \frac{1}{\sqrt{2p}} \sum_{k=1}^{p} |i;\ p^k;\ 1_1, \dots, 0_b\rangle. \tag{2.3}$$

Ezt követően előállítjuk az input és az egyes minták közötti Hamming-távolságot, mégpedig úgy, hogy az megjelenjen a valószínűségi-amplitúdóban. Érdemes megjegyezni, hogy ez a művelet a kvantum-párhuzamosságnak köszönhetően minden p^k elemre egyszerre hajtódik végre.

$$|\psi_2\rangle = \prod_{j=1}^{n} NOT_{m_j} XOR_{i_j m_j} |\psi_1\rangle \tag{2.4}$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2p}} \sum_{k=1}^{p} |i;\ d^k;\ 0_1, \dots, 0_b\rangle + \frac{1}{\sqrt{2p}} \sum_{k=1}^{p} |i;\ d^k;\ 1_1, \dots, 0_b\rangle$$

Az (2.5) alakban feltüntetett d_j^k akkor, és csak akkor |1⟩, ha $i_j = p_j^k$, egyébként |0⟩. A két bitsorozat közötti Hamming-távolság ekkor megegyezik d^k-ban található |0⟩ állapotú qubitek számával. A feladat tehát ezek megszámlálása.

A megoldáshoz vezessük be az alábbi egyqubites U unitér-operátort:

$$U = \begin{bmatrix} e^{i\frac{\pi}{2n}} & 0 \\ 0 & 1 \end{bmatrix} \tag{2.5}$$

Továbbá ennek 2qubites CU^{-2} változatát, amely a következőképpen áll elő:

$$CU^{-2} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U^{-2} = diag(I, U^{-2}) \text{ , ahol} \tag{2.6}$$

$$U^{-2} = \begin{bmatrix} e^{-i\frac{\pi}{n}} & 0 \\ 0 & 1 \end{bmatrix}$$

Ezek segítségével összeszámolhatók a Hamming-távolságok a következőképpen:

$$|\psi_3\rangle = \prod_{i=1}^{n} (CU^{-2})_{cm_i} \prod_{j=1}^{n} U_{m_j} |\psi_2\rangle \tag{2.7}$$

$$|\psi_3\rangle = \frac{1}{\sqrt{2p}} \sum_{k=1}^{p} e^{i\frac{\pi}{2n} d_H(i,p^k)} |i;\ d^k;\ 1_1, \dots, 0_b\rangle + \frac{1}{\sqrt{2p}} \sum_{k=1}^{p} e^{-i\frac{\pi}{2n} d_H(i,p^k)} |i;\ d^k;\ 1_1, \dots, 0_b\rangle$$

A (2.7)-es transzformáció végrehajtja a memória-regiszter összes bitjére az U_{m_j} transzformációt, amely m_j két lehetséges értékére:

$$U|0\rangle = \begin{bmatrix} e^{i\frac{\pi}{2n}} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} e^{i\frac{\pi}{2n}} \\ 0 \end{bmatrix} = e^{i\frac{\pi}{2n}} |0\rangle \tag{2.8}$$

$$U|1\rangle = \begin{bmatrix} e^{i\frac{\pi}{2n}} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

Azt látjuk tehát, hogy ez egy olyan transzformáció, amely m_j = |1⟩ esetén nem csinál semmit, viszont m_j = |0⟩-ra komplex $e^{i\frac{\pi}{2n}}$ együtthatóval szorozza az állapotvektort,

aminek mérhető következménye ugyen nincsen, hiszen $\left|e^{i\frac{\pi}{2n}}\right|^2 = 1$ (így az első posztulátum is igaz marad), azonban kvanumosan ezt még tovább tudjuk alakítani.

Amennyiben ezt $|m\rangle$ összes bitjére végrehajtuk, az együtthatók összeszorzódnak (természetesen a $|1\rangle$ értékű bitekre is elvégezzük a műveletet, de azokkal valójában nem történik semmi, így a következő egyenlet a szemléletesség kedvéért csak a $|0\rangle$ állapotú bitekre gyakorolt hatást szemlélteti):

$$U|0\rangle_1 \otimes U|0\rangle_2 \dots \otimes U|0\rangle_m = e^{i\frac{\pi}{2n}} e^{i\frac{\pi}{2n}} \dots e^{i\frac{\pi}{2n}} |0_1 \dots 0_m\rangle = e^{i\frac{\pi}{2n}\mathrm{m}} |0_1 \dots 0_m\rangle \qquad (2.9)$$

Tehát ha az n bites memóriában m darab $|0\rangle$ szerepel, akkor a transzformáció során a teljes kvantumállapot egy $e^{i\frac{\pi}{2n}\mathrm{m}}$ szorzótényezővel bővül, ahol $m = d_H(i, p^k)$, vagyis a Hamming-távolság egy $\frac{m}{n}$ arányossági tényezőként fog megjelenni. Ha a távolság zérus, tehát a két bitsorozat megegyezik (m=0), akkor $\frac{m}{n} = \frac{0}{n} = 0$, maximális Hamming-távolság esetén (m=n) pedig értelemszerűen $\frac{m}{n} = \frac{n}{n} = 1$.

A (2.7) során még el kell végezni $(CU^{-2})_{cm_i}$ transzformációt is, amely (2.9)-hez hasonlóan „összeszámolja" a $|0\rangle$ állapotú qubiteket, ám negatív előjellel, illetve$|c_1\rangle$ control-bit függvényében. Ez a transzformáció tehát csak $|c_1\rangle = |1\rangle$ esetén hajtódik végre (ami csak (2.4) szuperpozíciós állapot második tagjára igaz). Ez tehát $|\psi_3\rangle$ előállításának menete (2.7).

Látjuk, hogy kvantumosan már megjelentek a valószínűségi-amplitúdókban a Hamming-távolságok, azonban mivel az exponencális tag abszolútérték-négyzete továbbra is 1, az egyes mintákat ezek után is $\frac{1}{p}$ valószínűséggel kapnánk meg, ha megmérnénk a $|m\rangle$ tartalmát.

A megoldáshoz a $|c_1\rangle$ control-bitre alkamazott ismételt Hadamard-transzformációval jutunk, ugyanis belátható, hogy:

$$|\psi_4\rangle = H_{c_1} \prod_{j=1}^{n} XOR_{i_j m_j} NOT_{m_j} |\psi_3\rangle \qquad (2.10)$$

$$|\psi_4\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^{p} \cos\left(\frac{\pi}{2} \frac{d_H(i,p^k)}{n}\right) |i;\ p^k;\ 0_1, \dots, 0_b\rangle$$
$$+ \frac{1}{\sqrt{p}} \sum_{k=1}^{p} \sin\left(\frac{\pi}{2} \frac{d_H(i,p^k)}{n}\right) |i;\ p^k;\ 1_1, \dots, 0_b\rangle$$

Ahol a H transzformáción túl, elvégeztük (2.4) inverz műveletét, ami visszaállította a minták eredeti állapotát: $d^k \to p^k$.

Megfigyelhetjük, hogy a H-kapu ismételt végrehajtása már mérhető eredményt produkál. Az exponenciális tagok valós, harmonikus együtthatókká alakultak, amelyek együttes hossza továbbra is 1, azonban a változás következtében a mintákhoz tartózó amplitúdók átszerveződtek. Figyeljük meg, hogy a Hamming-távolság csökkenésével:

$$\frac{d_H(i,p^k)}{n} \to 0,$$
$$\cos\left(\frac{\pi}{2} \frac{d_H(i,p^k)}{n}\right) \to 1, \tag{2.11}$$
$$\sin\left(\frac{\pi}{2} \frac{d_H(i,p^k)}{n}\right) \to 0,$$

tehát a cos-os tagban a valószínűségi-amplitudók ott lesznek maximálisak, ahol a Hamming-távolság kicsi, és ennek megfelelően a sin-os taghoz rendeződnek a nagy Hamming-távolsághoz tartozó amplitudók.

Jelen esetben minket a cos-os tag együtthatói érdekelnek, hiszen az inputhoz legközelebb eső mintát szeretnénk kikeresni. Ha (2.10)-re tekintünk azt látjuk, hogy a memória-regiszterhez hozzáfonódott a szuperpozíciós állapotban lévő $|c_1\rangle$ control-bit, amelynek $|0\rangle$ oldalához tartozik a számunkra releváns (cos) eset. Ha tehát megmérjük a $|c_1\rangle$ control-bit tartalmát, és 0-t kapunk eredményül, akkor biztosan tudjuk, hogy a rendszer a cos-os tag irányába omlott össze, tehát ha ezután megmérjük $|m\rangle$ tartalmát, a kívánt effektust tapasztaljuk, azaz a legkisebb Hamming-távolságra súlyozott valószínűség-eloszlást. Amennyiben mérésünk 1-et ad eredményül, akkor éppen a sin-os tagokat mérhetnénk ki a memóriából, de minket nem az érdekel, hiszen ott az inputra legkevésbé hasonlító mintákat kapnánk meg legnagyobb súllyal (persze ehhez is biztosan található megfelelő alkalmazási példa, de a mintafelismerés szempontjából ezt

nem tudjuk elfogadni). Így ez utóbbi esetben újra kell kezdenünk az $|m\rangle$ regiszter valószínűségi-másolásától kezdve a fenti algoritmust, vagyis visszaugrunk (2.1)-hez.

Ahhoz, hogy ez a súlyozás megfelelő mértékben érzékelhető legyen, nem elegendő 1 control-bitet alkalmazni. A control-regiszter összes bitjére végre kell hajtani a fenti műveletet. Minél több bites a $|c\rangle$ regiszter, annál pontosabb eredményt kapunk, azonban annál nehezebb lesz kimérnünk a cos-os tagot. Ha $|c\rangle = |0_1, \dots, 0_b\rangle$ kezdeti control-regiszter minden bitjére alkalmazzuk az amplitúdó-erősítést, megjelenik $|c\rangle$ szuperpozíciójának összes permutációja, ami b biten 2^b. Csak akkor mérhetjük meg $|m\rangle$ tartalmát, ha pontosan $|c\rangle = |0_1, \dots, 0_b\rangle$ mérési eredményt kapjuk, vagyis a 2^b esetből mindössze 1 esetben végezhetjük el a mérést. A rendszer komplexitása tehát a pontosság függvényében változik. Ha control-regiszter összes bitjére alkalmazzuk az erősítést, a következő állapot adódik:

$$|\psi_{fin}\rangle = \frac{1}{\sqrt{p}} \sum_{k=1}^{p} cos^{b-l}\left(\frac{\pi}{2}\frac{d_H}{n}\right) sin^{l}\left(\frac{\pi}{2}\frac{d_H}{n}\right) \sum_{\{J^l\}} |i; p^k; J^l\rangle, \tag{2.12}$$

ahol $\{J^l\}$ jelöli a control-regiszter összes lehetséges permutációinak sorozatát, amely pontosan l darab 1-est, és $(b-l)$ darab 0-t tartalmaz. Ez tehát az összes lehetséges control-regiszter permutáció, kombinálva az összes tárolt mintával, a harmonikus tagok együtthatói pedig $|c\rangle$ függényében alakulnak. Mint ahogyan azt már említettük, minket ebből csak az az eset érdekel, amikor $J^l = |0_1, \dots, 0_b\rangle$. Ekkor, mivel $l = 0$ a képlet a következőképp egyszerűsödik:

$$|\psi_{fin}\rangle^0 = \frac{1}{\sqrt{p}} \sum_{k=1}^{p} cos^{b}\left(\frac{\pi}{2}\frac{d_H}{n}\right) |i; p^k; 0_1, \dots, 0_b\rangle, ahol \tag{2.13}$$

$$|\psi_{fin}\rangle = \sum_{i=0}^{2^b} |\psi_{fin}\rangle^i$$

Mielőtt még megmérhetnénk a memóriát, ki kell mérni $|\psi_{fin}\rangle^0$-hoz tartozó control-regiszter állapotot, aminek a valószínűsége:

$$P_b^{rec} = \frac{1}{p} \sum_{k=1}^{p} cos^{2b}\left(\frac{\pi}{2}\frac{d_H}{n}\right) \tag{2.14}$$

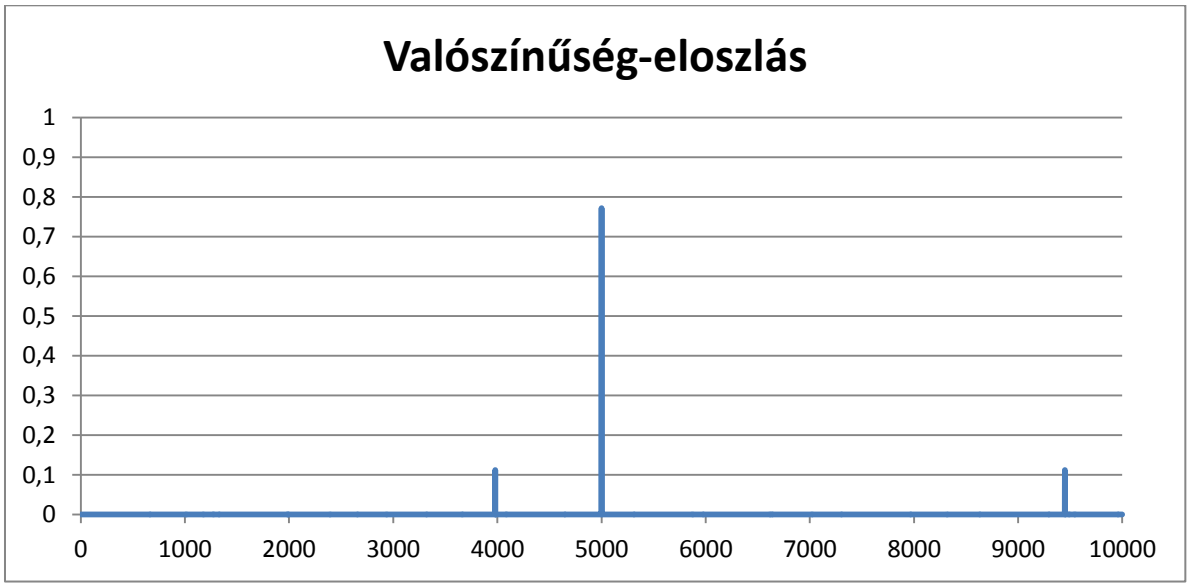
A szükséges próbálkozások számának várható értéke ennek reciproka, azaz: $\frac{1}{P_b^{rec}}$.

Ha sikerült a control-regisztert megfelelő állapotban lemérni, az egyes minták előfordulási valószínűsége:

$$P_b(p^k) = \frac{1}{Z}\, cos^{2b}\left(\frac{\pi}{2}\frac{d_H}{n}\right), ahol \qquad (2.15)$$

$$Z = pP_b^{rec} = \sum_{k=1}^{p} cos^{2b}\left(\frac{\pi}{2}\frac{d_H}{n}\right)$$

Az „Z” taggal korrigáljuk az egyes minták valószínűségét a control-regiszter nulla állapotára, így megkapjuk a minták egymáshoz viszonyított eloszlását.[14]

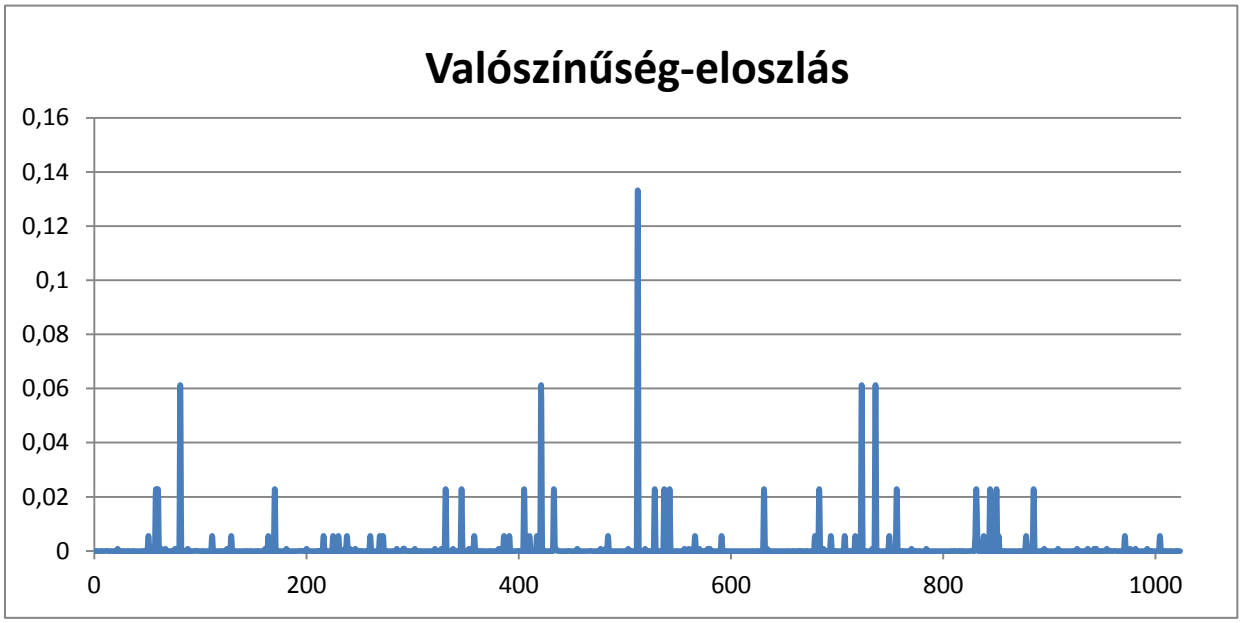


3.ábra

A fenti ábra egy szimulált kvantumregiszter mintáinak valószínűség-elszlását ábrázolja. A memória 10000 db, véletlenszám-generátorral feltöltött 16 bites mintákat tárol. A control-bitek számát 200-ra állítottam. Feltételezzük továbbá, hogy a memória tartalmazza a keresett inputot, és nincs ismétlődés. P_b^{rec}értékével korrigálva a fenti ábra adódik. Ilyen memória-állapot, és kalibráció esetén ~7500 próbálkozásra lenne szükség, mire megkapnánk a megfelelő control-regiszter állapotot, és lemérhetnénk a memória tartalmát. Igaz, akkor nagyon jó eséllyel, ~80%-os valószínűséggel a megfelelő mintát kapnánk meg. Ez a 7500-as érték azonban egy 10000-es adatsokaság esetén nagyon magas. Jóllehet, a párhuzamos utasításvégrehajtás és a memória kapacitása kompenzálja

[14] Ha ezt nem tennénk, akkor a valószínűségek a teljes állapottérre vonatkoznának, ami tartalmazza az összes control-regiszter állapotot, a sin-os együtthatókat…stb, így az nem reprezentálná megfelelően a minták eloszlását.

ezt a nagy számot, azonban tény, hogy a hagyományos rendszerekhez hasonlóan, a pontosságért az időt kell feláldoznunk.



4. ábra

A következő ábra egy 1024 elemű, továbbra is 16 bites mintákat tartalmazó memória-regiszter, b=20 control-bit esetén. Jól látható mi történik akkor, ha nem alkalmazunk elegendő számú control-bitet. Az erősítés már így is jól látható, azonban a többi minta valószínűségi-amplitudója még túl erős, cos-nak további hatványaira lenne szükség a pontosabb eredményhez. Bár a szükséges próbálkozások számának várható értéke mindössze 136, ám ennek megfelelően ~14%-os valószínűséggel kapjuk csak meg a megfelelő mintát, ami nagyon rossz.

Mit szólnánk hozzá, ha az előző példabeli 10000-es adatbázisnál 7500 próbálkozás helyett csupán annak töredékére, ~100 mérésre lenne szükség? Egy ekkora memória-regiszter esetén ez már nagyon is vállalható „ár”, hiszen ha ugyanúgy megkapjuk 80% valószínűséggel a helyes eredményt, azzal már kiválóan lehet dolgozni.

4.3. Amplitúdó-erősítés Grover-algoritmussal

A control-regiszter megfelelő állapotba állítása kulcsfontosságú az itt bemutatott mintafelismerő algoritmus szempontjából. Az algoritmus végrehajtása után (részben) csalódottan tapasztaljuk, hogy rendszerünk nem elég effektív. Vagy pontatlan eredményt kapunk, vagy túl sokszor kell végrehajtanunk a folyamatot, hogy értékelhető eredményt kapjunk. A jó hír, hogy problémánkra a Grover-algoritmus [4] tökéletes megoldást nyújt.

A Grover-algoritmus egy olyan amplitúdó-erősítési eljárás, amely valamilyen U|0⟩ jellegű, azaz kezdeti |0⟩ állapotból unitér transzformációkkal előállított kvantumállapotból szeretnénk egy ismert állapotot kiszűrni, amennyiben tudjuk, hogy az adott állapothoz tartozó valószínűség nullánál nagyobb.

Ez a mi esetünkre interpretálva azt jelenti, hogy adott a control-regiszter szuperpozíciós állapota[15], és tudjuk, hogy mi azon állapotokat keressük, ahol a control-regiszterben csupa |0⟩ szerepel. Több ilyen eset lesz, hiszen $|\psi_{fin}\rangle^0$miatt (2.13) minden mintához tartozni fog ilyen állapot, továbbá a konstrukcióból eredően azt is tudjuk, hogy az ezekhez tartozó valószínűségi-amplitúdók nem tűnnek el (sőt, végig azokat igyekeztünk erősíteni).

Ekkor létezik olyan Q unitér operátor, amit $|\psi_{fin}\rangle$-re m alkalommal végrehajtva, a rendszer levetítődik az általunk kiválasztott térrészbe, jelen esetben a |0⟩ control-regiszterhez tartozó térrészbe. Másképpen megfogalmazva: a nem |0⟩ állapotú tagok valószínűségi-amplitúdója eltűnik. Ezt követően megmérhetjük a memória tartalmát.

A különbség tehát a két módszer között az, hogy míg az első esetben gyakorlatilag valószínűségi alapon, érmedobálással igyeztünk megtalálni a tökéletes állapotú control-regisztert, addig a a második esetben pontosan, adott számú transzformáció útján jutunk eredményhez. Egy kérdés marad csupán: mennyi az a bizonyos lépésszám?

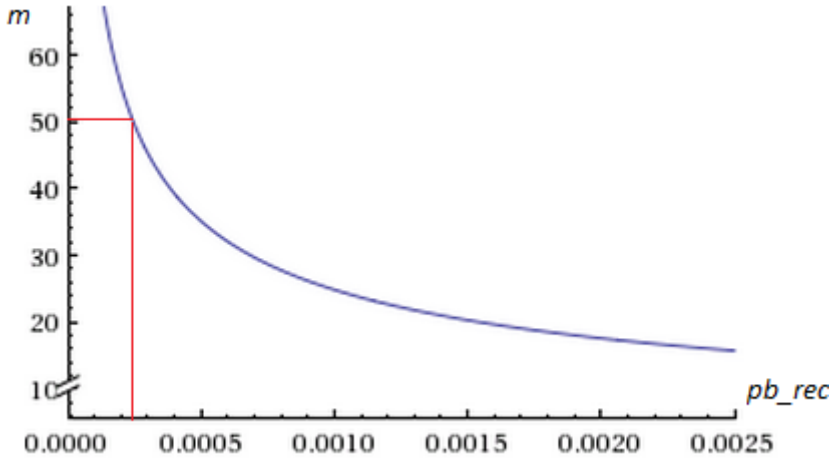
Definíció szerint [4]:

$$m = \frac{\pi}{4\Theta_a}, ahol$$
$$\sin^2(\Theta_a) = a \tag{2.16}$$

Mivel esetünkben $a = P_b^{rec}$, a képlet a következőképpen alakul:

$$m = \frac{\pi}{4}\frac{1}{\arcsin\left(\sqrt{P_b^{rec}}\right)} \tag{2.17}$$

[15] Valójában a teljes rendszer szuperpozíciós állapotáról beszélünk ilyenkor, hiszen a rendszer összefonódott állapotban van, mivel azonban a control-regiszter egy lehetséges állapotához tartozó térrészbe szeretnénk vetíteni, a control-regiszterre vonatkozóan kell vizsgálni az esetet.



5.ábra

Az ábrán (2.17) grafikonját láthatjuk, kicsiny $P_b^{rec} < \frac{1}{400}$ alatti értékekre(vízszintes-tengely). Megfigyelhetjük, hogy a függvény $\frac{1}{\sqrt{P_b^{rec}}}$-el arányosan változik csak, ami lehetővé teszi, hogy a függőleges tengelyen feltüntetett lépésszám nagyon kicsi valószínűségek esetén is kezelhető tartományban maradjon.

Vegyük például az x-tengely $P_b^{rec} = \frac{1}{4000}$ -es helyéhez tartozó függvényértéket, ami ~50. Amíg az előző módszerrel 4000-szer kell megismételnünk az algoritmust, mostmár csak 50 lépésre van szükség, ráadásul ez az 50-as érték fix szám, miközben a 4000 próbálkozás egy várható érték. Ezen hatást nevezi a szakirodalom „quadratic speedup”-nak [4].

4.4 Konklúzió

Az eddigiek alapján egyértelműen látszik, hogy a kvantumos mintafelismerés számos előnnyel rendelkezik. Az egyik legjelentősebb a párhuzamos művelet-végrehajtás (ezt a Hamming-távolságok egy lépésben való kimérésének példáján láthattuk), a másik az $O\left(\frac{1}{\sqrt{P_b^{rec}}}\right)$ bonyolultságú Grover-algoritmus féle iterációs lépések száma. Ez utóbbi lehetővé teszi számunkra, hogy hatalmas adatbázisokból, nagyon magas valószínűséggel ismerhessünk fel mintákat, miközben az algoritmus bonyolultsága kezelhető határok között marad. Ez végül ahhoz vezet, hogy nagyon pontos méréseket végezhetünk, nagyon alacsony válaszidővel, ami az intelligens rendszerek időbeli adaptációját jelentősen gyorsíthatja.

A kvantum-algoritmus pozitívumai mellett szól még az is, hogy az absztrakt matematikán túl nem épül bonyolult szimulációs modellekre, hanem jellemzően a legegyszerűbb módon

igyekszik reagálni a problémákra, mely viselkedés azt az érzetet kelti bennünk, mintha ez lenne a folyamatok legtermészetesebb módja.

A mintafelismerés valószínűségi alapokon nyugvása akár hátrányként is említhető volna, hiszen könnyen előfordulhat olyan eset, amikor 98%-os valószínűség ellenére mégsem a helyes mintát kapjuk meg a kimeneten, ám egy másik szemszögből rávilágítva, az emberi agy is hasonlóképpen működik.

Ez a viselkedés ráadásul felvet olyan kérdéseket is, mint a természetes szimuláció. Azaz egy kvantum-algoritmusokra épülő intelligens rendszer jellemzően természetes módon lehet képes emberi funkciók szimulálására, aminek reprodukciója a hagyományos informatika eszközeivel élve gyakran embert próbáló kihívást jelent. Érdemes tehát a tudomány ezen szegletével foglalkozni intelligens rendszerek tekintetében, hiszen végső soron mi is kvantumosan működünk.

V. Irodalomjegyzék

[1] C.A.Trugenberger, Quantum Pattern Recognition, quant-ph/0210176v2, 2008.

[2] S. Imre és B. Ferenc, Quantum Computing and Communications - An Engineering Approach, 2005.

[3] L.-M. Duan és G.-C. Guo, Phys. Rev. Lett. 80, 4999, 1998.

[4] G. Brassard, P. Hoyer, M. Mosca és A. Tapp, Amplitude Amplification and Estimation, quant-ph/0005055, 2000.