

4 Πύλες πολλαπλών qubit

Μέχρι τώρα μελετήσαμε αποκλειστικά πύλες που δρουν πάνω σε ένα μόνο qubit. Ένας κβαντικός υπολογιστής όμως γίνεται πραγματικά χρήσιμος μόνο όταν διαθέτει πολλά qubit που αλληλεπιδρούν μεταξύ τους, οπότε χρειαζόμαστε και πύλες που δρουν ταυτόχρονα σε περισσότερα από ένα qubit. Στην ενότητα αυτή θα μελετήσουμε τις πιο σημαντικές τέτοιες πύλες, ξεκινώντας από την απλούστερη και πιο θεμελιώδη περίπτωση: συστήματα δύο qubit.

4.1 Ελεγχόμενες πύλες δύο qubit

Μία ιδιαίτερα χρήσιμη κατηγορία πυλών πολλαπλών qubit είναι οι *ελεγχόμενες πύλες* (*controlled gates*). Σε μία τέτοια πύλη, ένα ή περισσότερα qubit παίζουν τον ρόλο του *ελέγχου* (*control*) και ένα ή περισσότερα άλλα τον ρόλο του *στόχου* (*target*). Ανάλογα με την κατάσταση των qubit ελέγχου εφαρμόζεται ή όχι μία συγκεκριμένη πύλη U πάνω στα qubit-στόχους.

Για παράδειγμα θα μπορούσαμε να εφαρμόσουμε την πύλη U στα qubit στόχου μόνο όταν όλα τα qubit-ελέγχου βρίσκονται ταυτόχρονα στην κατάσταση $|1\rangle$, ενώ σε κάθε άλλη περίπτωση όλα τα qubit (και ελέγχου και στόχοι) παραμένουν αμετάβλητα.

Θα συμβολίζουμε μία τέτοια πύλη με $C_n^k(U)$ και όπως και στα προηγούμενα:

- Ο τελεστής που υλοποιεί η πύλη και δρα πάνω στις κβαντικές καταστάσεις $|\psi\rangle$ θα συμβολίζεται με “καπελάκι” δηλαδή:

$$\hat{C}_n^k(\hat{U})$$

- Ο πίνακας που περιγράφει τον τελεστή και επομένως και την πύλη θα συμβολίζεται με έντονα γράμματα (bold):

$$C_n^k(U)$$

Στην περίπτωση στην οποία αναφερόμαστε θα πρέπει ο πίνακας $C_n^k(U)$ να είναι $2^{n+k} \times 2^{n+k}$ καθώς δρα πάνω σε $n+k$ qubits και όπως μπορούμε να δούμε υπάρχουν 2^{n+k} ιδιοδιανύσματα βάσεις $|\psi_i\rangle$ που περιγράφουν όλες τις καταστάσεις όπου όλα τα qubit είναι σε πλήρως καθορισμένη κατάσταση, $|0\rangle$ ή $|1\rangle$. Αντίστοιχα ο πίνακας U περιγράφει μία πύλη που δρα πάνω σε k qubits και επομένως πρέπει να έχει διαστάσεις $2^k \times 2^k$.

Παράδειγμα 4.1: Η πύλη CNOT

Η πιο γνωστή ελεγχόμενη πύλη είναι η *CNOT* (*Controlled-NOT*), $C_1^1(X)$: το πρώτο qubit είναι το qubit-ελέγχου, και όταν αυτό βρίσκεται στην κατάσταση $|1\rangle$, εφαρμόζεται η πύλη X (NOT) στο δεύτερο qubit. Ας δούμε αναλυτικά πως συμπεριφέρεται η CNOT.

$$|00\rangle \xrightarrow{C_1^1(X)} |00\rangle$$

$$|01\rangle \xrightarrow{C_1^1(X)} |01\rangle$$

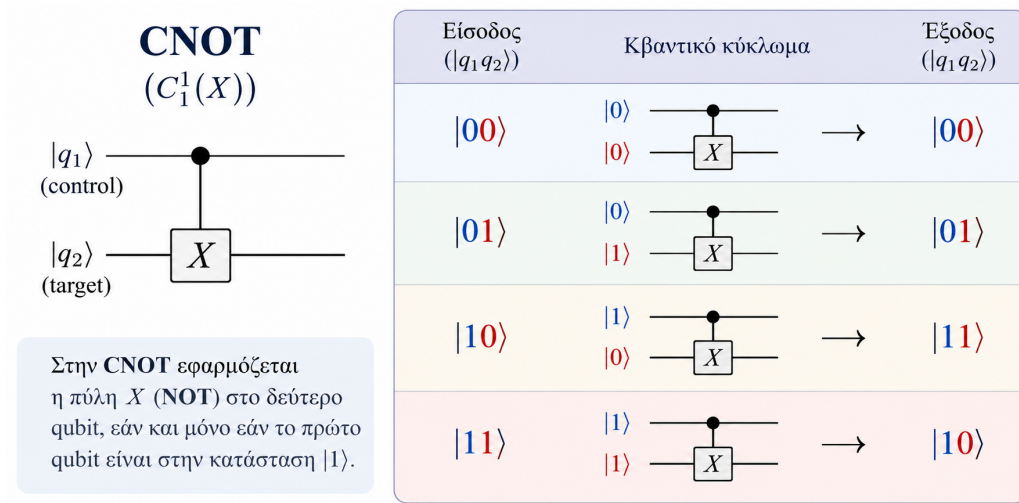
$$|10\rangle \xrightarrow{C_1^1(X)} |11\rangle$$

$$|11\rangle \xrightarrow{C_1^1(X)} |10\rangle$$

Η δράση της πύλης φαίνεται στο σχήμα 4.1. Όπως βλέπουμε το δεύτερο qubit αντιστρέφεται μόνο όταν το πρώτο είναι $|1\rangle$. Θα μπορούσαμε να περιγράψουμε την πύλη μέσω του τελεστή της και ως εξής: Για κάθε βασική κατάσταση $|q_1 q_2\rangle$ με $q_1, q_2 \in \{0, 1\}$,

$$\hat{C}_1^1(\hat{X}) |q_1 q_2\rangle = \begin{cases} |q_1 q_2\rangle, & q_1 = 0 \\ |q_1 \bar{q}_2\rangle, & q_1 = 1 \end{cases}$$

Ορίζουμε τις κυματοσυναρτήσεις βάσης:



Σχήμα 4.1: Η λειτουργία της πύλης CNOT.

$$|\psi_1\rangle = |00\rangle$$

$$|\psi_2\rangle = |01\rangle$$

$$|\psi_3\rangle = |10\rangle$$

$$|\psi_4\rangle = |11\rangle$$

Εφαρμόζοντας τον τελεστή της πύλης σε καθεμία από τις τέσσερις βασικές καταστάσεις:

$$\hat{C}_1^1(\hat{X}) |\psi_1\rangle = |\psi_1\rangle$$

$$\hat{C}_1^1(\hat{X}) |\psi_2\rangle = |\psi_2\rangle$$

$$\hat{C}_1^1(\hat{X}) |\psi_3\rangle = |\psi_4\rangle$$

$$\hat{C}_1^1(\hat{X}) |\psi_4\rangle = |\psi_3\rangle$$

αφού στις $|\psi_1\rangle, |\psi_2\rangle$ το qubit-ελέγχου είναι $|0\rangle$ (άρα δεν αλλάζει τίποτα), ενώ στις $|\psi_3\rangle, |\psi_4\rangle$ είναι $|1\rangle$ (άρα αντιστρέφεται το δεύτερο qubit). Τα στοιχεία c_{pq} του πίνακα θα δίνονται από την σχέση:

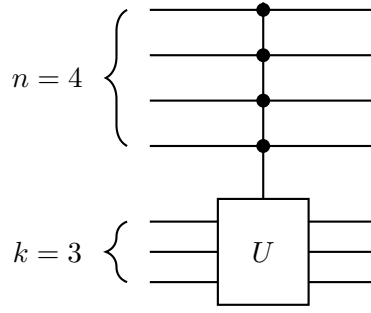
$$c_{pq} = \langle \psi_p | \hat{C}_1^1(\hat{X}) | \psi_q \rangle$$

Βλέπουμε ότι οι μόνοι μη μηδενικοί συνδυασμοί είναι το $p = q = 1, p = q = 2, p = 3, q = 4$ και $p = 4, q = 3$ όπου το αντίστοιχο στοιχείο του πίνακα είναι 1, για παράδειγμα:

$$c_{34} = \langle \psi_3 | \hat{C}_1^1(\hat{X}) | \psi_4 \rangle = \langle \psi_3 | \psi_3 \rangle = 1$$

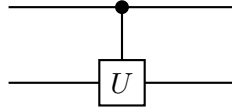
Επομένως:

$$C_1^1(\mathbf{X}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

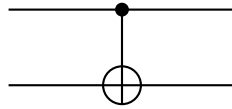


Οι πύλες που θα μελετήσουμε σε αυτή την ενότητα είναι αυτής της απλούστερης μορφής, $\hat{C}_1^1(\hat{U})$, με ένα μόνο qubit ελέγχου και ένα μόνο qubit-στόχο.

Σε ένα κβαντικό κύκλωμα, η πύλη $\hat{C}_1^1(\hat{U})$ απεικονίζεται με μία γεμάτη κουκκίδα • πάνω στο σύρμα του qubit-ελέγχου, συνδεδεμένη με μία κάθετη γραμμή σε ένα κουτί με το όνομα U πάνω στο σύρμα του qubit-στόχου:

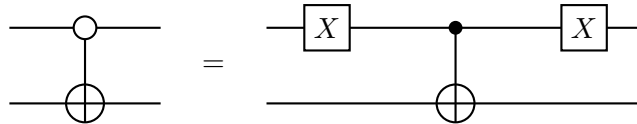


Στην ειδική περίπτωση της CNOT, αντί για το κουτί X χρησιμοποιείται σχεδόν πάντα το σύμβολο $\oplus$ (ένας κύκλος με σταυρό πάνω στο σύρμα), ακριβώς επειδή η CNOT πραγματοποιεί πρόσθεση modulo 2 (XOR) στο qubit-στόχο, μία πράξη που συμβολίζεται παραδοσιακά με το $\oplus$:



Μία ακόμα χρήσιμη σύμβαση είναι το άσπρο κυκλάκι $\circ$ αντί για το συνηθισμένο γεμάτο μαύρο κυκλάκι •. Το γεμάτο μαύρο κυκλάκι σημαίνει ότι η πύλη ενεργοποιείται όταν το qubit-ελέγχου είναι στην κατάσταση $|1\rangle$, ενώ το άσπρο κυκλάκι σημαίνει ότι ενεργοποιείται όταν το qubit-ελέγχου είναι στην κατάσταση $|0\rangle$. Έχουμε δηλαδή μία *αντίστροφα ελεγχόμενη (anti-controlled)* πύλη.

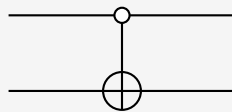
Μία τέτοια πύλη είναι πάντα ισοδύναμη με μία συνηθισμένη (με μαύρο κυκλάκι) ελεγχόμενη πύλη, αν απλώς αντιστρέψουμε το qubit-ελέγχου με μία X πριν και ξανά με μία X μετά, ώστε να το επαναφέρουμε στην αρχική του τιμή:



Πράγματι, αν το qubit-ελέγχου είναι αρχικά $|0\rangle$, η πρώτη $\hat{X}$ το μετατρέπει σε $|1\rangle$, ενεργοποιώντας κανονικά την (γεμάτου κυκλακιού) ελεγχόμενη πύλη στη μέση, ενώ η δεύτερη $\hat{X}$ το επαναφέρει στο $|0\rangle$. αν είναι αρχικά $|1\rangle$, οι δύο $\hat{X}$ απλώς το στέλνουν στο $|0\rangle$ και πίσω στο $|1\rangle$, χωρίς ποτέ να ενεργοποιήσουν την ενδιάμεση πύλη, ακριβώς η συμπεριφορά που περιμένουμε από μία αντι-ελεγχόμενη πύλη.

Παράδειγμα 4.2: Η πύλη CNOT με άσπρο κυκλάκι

Ας δούμε τώρα την αντι-ελεγχόμενη εκδοχή της CNOT, δηλαδή την $C_1^1(X)$ με άσπρο αντί για μαύρο κυκλάκι στο qubit-ελέγχου:



Το πρώτο qubit είναι και πάλι το qubit-ελέγχου, όμως τώρα η πύλη X εφαρμόζεται στο δεύτερο qubit όταν το πρώτο βρίσκεται στην κατάσταση $|0\rangle$ (αντί για $|1\rangle$). Ας δούμε αναλυτικά πως συμπεριφέρεται.

$$\begin{aligned}
|00\rangle &\xrightarrow{\text{anti-}C_1^1(X)} |01\rangle \\
|01\rangle &\xrightarrow{\text{anti-}C_1^1(X)} |00\rangle \\
|10\rangle &\xrightarrow{\text{anti-}C_1^1(X)} |10\rangle \\
|11\rangle &\xrightarrow{\text{anti-}C_1^1(X)} |11\rangle
\end{aligned}$$

Η δράση της πύλης είναι δηλαδή ακριβώς αντίστροφη από αυτή της συνηθισμένης CNOT: το δεύτερο qubit αντιστρέφεται μόνο όταν το πρώτο είναι $|0\rangle$. Θα μπορούσαμε να περιγράψουμε την πύλη μέσω του τελεστή της και ως εξής: για κάθε βασική κατάσταση $|q_1 q_2\rangle$ με $q_1, q_2 \in \{0, 1\}$,

$$\hat{C} |q_1 q_2\rangle = \begin{cases} |\bar{q}_1 q_2\rangle, & q_1 = 0 \\ |q_1 q_2\rangle, & q_1 = 1 \end{cases}$$

όπου με $\hat{C}$ συμβολίζουμε εδώ τον τελεστή της αντι-ελεγχόμενης πύλης, για συντομία. Χρησιμοποιώντας τις ίδιες κυματοσυναρτήσεις βάσης $|\psi_1\rangle = |00\rangle, |\psi_2\rangle = |01\rangle, |\psi_3\rangle = |10\rangle, |\psi_4\rangle = |11\rangle$ του παραδείγματος 4.1, εφαρμόζοντας τον τελεστή της πύλης σε καθεμία από τις τέσσερις:

$$\begin{aligned}
\hat{C} |\psi_1\rangle &= |\psi_2\rangle \\
\hat{C} |\psi_2\rangle &= |\psi_1\rangle \\
\hat{C} |\psi_3\rangle &= |\psi_3\rangle \\
\hat{C} |\psi_4\rangle &= |\psi_4\rangle
\end{aligned}$$

αφού στις $|\psi_1\rangle, |\psi_2\rangle$ το qubit-ελέγχου είναι $|0\rangle$ (άρα αντιστρέφεται το δεύτερο qubit), ενώ στις $|\psi_3\rangle, |\psi_4\rangle$ είναι $|1\rangle$ (άρα δεν αλλάζει τίποτα). Τα στοιχεία c_{pq} του πίνακα θα δίνονται και πάλι από την σχέση:

$$c_{pq} = \langle \psi_p | \hat{C} | \psi_q \rangle$$

Βλέπουμε ότι οι μόνοι μη μηδενικοί συνδυασμοί είναι το $p = 1, q = 2, p = 2, q = 1, p = q = 3$ και $p = q = 4$ όπου το αντίστοιχο στοιχείο του πίνακα είναι 1, για παράδειγμα:

$$c_{12} = \langle \psi_1 | \hat{C} | \psi_2 \rangle = \langle \psi_1 | \psi_1 \rangle = 1$$

Επομένως:

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Παρατηρήστε ότι, σε σχέση με τον πίνακα $C_1^1(X)$ του παραδείγματος 4.1, οι πρώτες δύο γραμμές/στήλες (όπου $q_1 = 0$) έχουν «ανταλλάξει ρόλο» με τις τελευταίες δύο (όπου $q_1 = 1$): εκεί η εναλλαγή γινόταν στις $|\psi_3\rangle, |\psi_4\rangle$, εδώ στις $|\psi_1\rangle, |\psi_2\rangle$, ακριβώς όπως αναμένουμε από μία αντι-ελεγχόμενη πύλη.

Μία απλή ελεγχόμενη με μαύρο κυκλάκι πύλη συναντήσαμε υπό μία έννοια ήδη στο πείραμα της γάτας του Schrödinger (ενότητα 2.13), χωρίς να τη μελετήσουμε αναλυτικά: εκεί το πρώτο qubit (η «συσσκευή») ήλεγχε αν θα αντιστραφεί η κατάσταση του δεύτερου qubit (η γάτα).

Παράδειγμα 4.3: Η ελεγχόμενη πύλη Hadamard

Με τον ίδιο ακριβώς τρόπο ορίζουμε την ελεγχόμενη πύλη Hadamard, $C_1^1(H)$: το πρώτο qubit είναι πάλι το qubit-ελέγχου, και όταν αυτό βρίσκεται στην κατάσταση $|1\rangle$, εφαρμόζεται η πύλη H στο δεύτερο

qubit,

$$\hat{C}_1^1(\hat{H})|q_1q_2\rangle = \begin{cases} |q_1q_2\rangle, & q_1 = 0 \\ |1\rangle \otimes (\hat{H}|q_2\rangle), & q_1 = 1. \end{cases}$$

Επιλέγουμε πάλι τις καταστάσεις βάσεις:

$$\begin{aligned} |\psi_1\rangle &= |00\rangle \\ |\psi_2\rangle &= |01\rangle \\ |\psi_3\rangle &= |10\rangle \\ |\psi_4\rangle &= |11\rangle \end{aligned}$$

Στις καταστάσεις $|\psi_1\rangle, |\psi_2\rangle$, το πρώτο qubit είναι $q_1 = 0$ η πύλη δεν κάνει τίποτα. Επομένως όπως και πριν:

$$\begin{aligned} \hat{C}_1^1(\hat{H})|\psi_1\rangle &= |\psi_1\rangle \\ \hat{C}_2^1(\hat{H})|\psi_2\rangle &= |\psi_2\rangle \end{aligned}$$

Στις $|\psi_3\rangle = |10\rangle$ και $|\psi_4\rangle = |11\rangle$ εφαρμόζεται η H στο δεύτερο qubit,

$$\begin{aligned} \hat{C}_1^1(\hat{H})|\psi_3\rangle &= |1\rangle \otimes \hat{H}|0\rangle = |1\rangle \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|10\rangle + |11\rangle}{\sqrt{2}} = \frac{|\psi_3\rangle + |\psi_4\rangle}{\sqrt{2}}, \\ \hat{C}_1^1(\hat{H})|\psi_4\rangle &= |1\rangle \otimes \hat{H}|1\rangle = |1\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|10\rangle - |11\rangle}{\sqrt{2}} = \frac{|\psi_3\rangle - |\psi_4\rangle}{\sqrt{2}}. \end{aligned}$$

Τα στοιχεία c_{pq} υπολογίζονται όπως και πριν ως $\langle\psi_p|\hat{C}_1^1(\hat{H})|\psi_q\rangle$. Αν παρατηρήσουμε την σχέση αυτή βλέπουμε ότι στην ουσία οι στήλες του πίνακα, δηλαδή όταν κρατάμε σταθερό το q είναι στην ουσία οι συντεταγμένες του $\hat{C}_1^1(\hat{H})|\psi_q\rangle$ ως προς την βάση $|\psi_p\rangle$. Επομένως η πρώτη στήλη ($q = 1$), θα είναι

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

αφού $\hat{C}_1^1(\hat{H})|\psi_1\rangle = |\psi_1\rangle$. Με τον ίδιο τρόπο βλέπουμε ότι η τρίτη στήλη ($q = 3$) που αντιστοιχεί στο

$$\hat{C}_1^1(\hat{H})|\psi_3\rangle = \frac{|\psi_3\rangle + |\psi_4\rangle}{\sqrt{2}},$$

οπότε η στήλη αυτή θα πρέπει να είναι

$$\begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Τελικά θα έχουμε:

$$C_1^1(H) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Παρατηρήστε ότι το πάνω-αριστερά τεταρτημόριο 2×2 είναι πάντα ο ταυτοτικός πίνακας I_2 (αφού για $q_1 = 0$ δεν συμβαίνει τίποτα), ενώ το κάτω-δεξιά τεταρτημόριο είναι ακριβώς ο πίνακας H (η πύλη που εφαρμόζεται στο δεύτερο qubit όταν $q_1 = 1$).

Η παρατήρηση στο τέλος του παραπάνω παραδείγματος δεν είναι σύμπτωση, αλλά ισχύει για κάθε ελεγχόμενη πύλη.

Αποτέλεσμα 4.1: Ο πίνακας κάθε ελεγχόμενης πύλης $\hat{C}_1^1(\hat{V})$

Για οποιαδήποτε πύλη ενός qubit με πίνακα V , η αντίστοιχη ελεγχόμενη πύλη $\hat{C}_1^1(\hat{V})$ (πρώτο qubit το qubit-ελέγχου, εφαρμογή της V στο δεύτερο qubit όταν $q_1 = 1$) περιγράφεται, στη βάση $|\psi_1\rangle, \dots, |\psi_4\rangle$, από τον 4×4 πίνακα

$$C_1^1(V) = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & V \end{bmatrix},$$

όπου κάθε στοιχείο του παραπάνω πίνακα είναι το ίδιο ένας 2×2 πίνακας: I_2 και $\mathbf{0}$ στην πρώτη γραμμή τεταρτημορίων (το qubit-στόχος παραμένει αμετάβλητο όταν $q_1 = 0$), και $\mathbf{0}$, V στη δεύτερη (η V εφαρμόζεται στο qubit-στόχο όταν $q_1 = 1$).

Παράδειγμα 4.4: Η πύλη controlled- Z είναι συμμετρική

Θέλουμε να δείξουμε ότι η ελεγχόμενη πύλη Z είναι συμμετρική ως προς τα qubit ελέγχου και στόχου. Δηλαδή μπορούμε να εναλλάξουμε τον ρόλο των δύο αυτών qubit όπως δείχνει και το παρακάτω σχήμα:



Ας ορίσουμε την ελεγχόμενη πύλη Z , $\hat{C}_1^1(\hat{Z})$, με το πάνω qubit ως qubit-ελέγχου ως συνήθως: όταν αυτό βρίσκεται στην κατάσταση $|1\rangle$, εφαρμόζεται η Z στο δεύτερο qubit. Από το αποτέλεσμα 4.1, με $V = Z$, ο πίνακάς της είναι

$$C_1^1(Z) = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Ας δούμε τώρα τι θα συνέβαινε αν αντιστρέψαμε τους ρόλους: το *δεύτερο* qubit να είναι το qubit-ελέγχου, και να εφαρμόζεται η Z στο *πρώτο* qubit όταν το δεύτερο qubit βρίσκεται στην κατάσταση $|1\rangle$. Ως συνήθως χρησιμοποιούμε τις καταστάσεις:

$$|\psi_1\rangle = |00\rangle$$

$$|\psi_2\rangle = |01\rangle$$

$$|\psi_3\rangle = |10\rangle$$

$$|\psi_4\rangle = |11\rangle$$

Εφόσον έχουν αντιστραφεί οι ρόλοι των qubit, θα πρέπει το δεύτερο qubit να είναι 1 για να εφαρμοστεί η Z στο πρώτο qubit. Αν συμβολίσουμε προσωρινά με D αυτή την πύλη θα έχουμε:

$$|\psi_1\rangle = |00\rangle \xrightarrow{D} |00\rangle$$

$$|\psi_2\rangle = |01\rangle \xrightarrow{D} \hat{Z}|0\rangle \otimes |1\rangle = |01\rangle = |\psi_2\rangle$$

$$|\psi_3\rangle = |10\rangle \xrightarrow{D} |10\rangle$$

$$|\psi_4\rangle = |11\rangle \xrightarrow{D} \hat{Z}|1\rangle \otimes |1\rangle = -|11\rangle = -|\psi_4\rangle$$

Ως εκ τούτου, ο πίνακας αυτής της αντεστραμμένης πύλης είναι:

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix},$$

δηλαδή ακριβώς ο ίδιος πίνακας $C_1^1(\mathbf{Z})$ με πριν! Επομένως, η πύλη $\hat{C}_1^1(\hat{Z})$ είναι *συμμετρική*: δεν έχει καμία σημασία ποιο από τα δύο qubit θεωρούμε qubit-ελέγχου και ποιο qubit-στόχο, το αποτέλεσμα είναι πάντα το ίδιο.

Έχει ενδιαφέρον να σημειώσουμε ότι όταν αναλύουμε την δράση μίας πύλης έχει σημασία πως κωδικοποιείται το qubit. Έχουμε συνηθίσει να αντιστοιχούμε τις δύο καταστάσεις του qubit στα $|0\rangle$ και $|1\rangle$ που στην περίπτωση του spin του ηλεκτρονίου αντιστοιχούσε στον άξονα των z : το $|0\rangle$ αντιστοιχεί στο $s_z < 0$ ενώ το $|1\rangle$ αντιστοιχεί στο $s_z > 0$. Όμως η απόφαση αυτή δεν είναι πάντα δεσμευτική. Θα μπορούσαμε εξίσου να αντιστοιχήσουμε τις δύο καταστάσεις του qubit σε καταστάσεις που αντιστοιχούν σε θετικό και αρνητικό spin σε άλλο άξονα, π.χ. στον άξονα των x . Θα μπορούσαμε ακόμα και να αντιστρέψουμε τα πρόσημα του spin, δηλαδή να χρησιμοποιήσουμε την κατάσταση $|+x\rangle$ για το qubit “0” και το $|-x\rangle$ για το qubit “1”. Ας δούμε πως μία πύλη CNOT συμπεριφέρεται όταν χρησιμοποιούμε αυτή την εναλλακτική βάση.

Παράδειγμα 4.5: Η CNOT στη βάση $|+x\rangle, |-x\rangle$

Αν αντί των $|0\rangle$ και $|1\rangle$ χρησιμοποιήσουμε τα $|+x\rangle, |-x\rangle$, οι καταστάσεις βάσης για τα 2 qubit θα είναι

$$|\phi_1\rangle = |+x\rangle \otimes |+x\rangle$$

$$|\phi_2\rangle = |+x\rangle \otimes |-x\rangle$$

$$|\phi_3\rangle = |-x\rangle \otimes |+x\rangle$$

$$|\phi_4\rangle = |-x\rangle \otimes |-x\rangle$$

Θυμηθείτε ότι στο παράδειγμα 3.1 είχαμε δείξει ότι:

$$|+x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|-x\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

και ας δούμε απευθείας πώς δρα πάνω τους η πύλη $\hat{X}$ (με $\hat{X}|0\rangle = |1\rangle$ και $\hat{X}|1\rangle = |0\rangle$):

$$\hat{X}|+x\rangle = \frac{1}{\sqrt{2}}(\hat{X}|0\rangle + \hat{X}|1\rangle) = \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle) = |+x\rangle,$$

$$\hat{X}|-x\rangle = \frac{1}{\sqrt{2}}(\hat{X}|0\rangle - \hat{X}|1\rangle) = \frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = -|-x\rangle.$$

Δηλαδή τα $|+x\rangle, |-x\rangle$ είναι ιδιοκαταστάσεις του $\hat{X}$, με ιδιοτιμές $+1$ και -1 αντίστοιχα.

Θα υπολογίσουμε τώρα απευθείας τα στοιχεία του πίνακα της CNOT στη νέα βάση,

$$c_{pq} = \langle \phi_p | \hat{C}_1^1(\hat{X}) | \phi_q \rangle.$$

Ο ορισμός της CNOT (παράδειγμα 4.1) δίνει τη δράση της όταν το qubit-ελέγχου βρίσκεται στην κατάσταση $|0\rangle$ ή $|1\rangle$. Αναλύουμε λοιπόν, με γραμμικότητα, μόνο το qubit-ελέγχου στην υπολογιστική βάση,

ενώ στο qubit-στόχο χρησιμοποιούμε ό,τι μόλις υπολογίσαμε για τη δράση του $\hat{X}$ στα $|\pm x\rangle$. Για παράδειγμα, ας δούμε το $|\phi_2\rangle$: αναλύουμε το qubit-ελέγχου του, που βρίσκεται στο $|+x\rangle$, στην υπολογιστική βάση, και εφαρμόζουμε τον τελεστή με γραμμικότητα,

$$\begin{aligned}\hat{C}_1^1(\hat{X})|\phi_2\rangle &= \hat{C}_1^1(\hat{X}) \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |-x\rangle \\ &= \frac{1}{\sqrt{2}} \left(\hat{C}_1^1(\hat{X}) |0\rangle \otimes |-x\rangle + \hat{C}_1^1(\hat{X}) |1\rangle \otimes |-x\rangle \right) \\ &= \frac{1}{\sqrt{2}} \left(|0\rangle \otimes |-x\rangle + |1\rangle \otimes \hat{X} |-x\rangle \right) = \frac{1}{\sqrt{2}} (|0\rangle \otimes |-x\rangle - |1\rangle \otimes |-x\rangle) \\ &= \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \otimes |-x\rangle = |\phi_4\rangle.\end{aligned}$$

Στο πρώτο σκέλος του αθροίσματος το qubit-ελέγχου είναι $|0\rangle$, οπότε η πύλη δεν κάνει τίποτα· στο δεύτερο είναι $|1\rangle$, οπότε εφαρμόζεται το $\hat{X}$ στο δεύτερο qubit, που απλώς αλλάζει το πρόσημο, $\hat{X} |-x\rangle = -|-x\rangle$. Με τον ίδιο ακριβώς τρόπο για τα υπόλοιπα τρία:

$$\begin{aligned}\hat{C}_1^1(\hat{X})|\phi_1\rangle &= \frac{1}{\sqrt{2}} (|0\rangle \otimes |+x\rangle + |1\rangle \otimes \hat{X} |+x\rangle) = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \otimes |+x\rangle = |\phi_1\rangle, \\ \hat{C}_1^1(\hat{X})|\phi_3\rangle &= \frac{1}{\sqrt{2}} (|0\rangle \otimes |+x\rangle - |1\rangle \otimes \hat{X} |+x\rangle) = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \otimes |+x\rangle = |\phi_3\rangle, \\ \hat{C}_1^1(\hat{X})|\phi_4\rangle &= \frac{1}{\sqrt{2}} (|0\rangle \otimes |-x\rangle - |1\rangle \otimes \hat{X} |-x\rangle) = \frac{1}{\sqrt{2}} (|0\rangle \otimes |-x\rangle + |1\rangle \otimes |-x\rangle) = |\phi_2\rangle.\end{aligned}$$

Δηλαδή η CNOT αφήνει αμετάβλητα τα $|\phi_1\rangle, |\phi_3\rangle$ και εναλλάσσει τα $|\phi_2\rangle, |\phi_4\rangle$.

Τα $|\phi_p\rangle$ είναι ορθοκανονικά μεταξύ τους, ως τανυστικά γινόμενα των ορθοκανονικών $|+x\rangle, |-x\rangle$ · για παράδειγμα,

$$\langle +x | -x \rangle = \frac{1}{2} (\langle 0|0\rangle - \langle 0|1\rangle + \langle 1|0\rangle - \langle 1|1\rangle) = 0$$

Στα στοιχεία c_{pq} , λοιπόν, οι μόνοι μη μηδενικοί συνδυασμοί είναι το $p = q = 1, p = 4, q = 2, p = q = 3$ και $p = 2, q = 4$, όπου το αντίστοιχο στοιχείο του πίνακα είναι 1, για παράδειγμα:

$$c_{42} = \langle \phi_4 | \hat{C}_1^1(\hat{X}) | \phi_2 \rangle = \langle \phi_4 | \phi_4 \rangle = 1$$

Επομένως, ο πίνακας της CNOT στη βάση $|\phi_1\rangle, \dots, |\phi_4\rangle$ είναι:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Παρατηρήστε ότι η CNOT, που στην υπολογιστική βάση αφήνει αμετάβλητα τα $|\psi_1\rangle, |\psi_2\rangle$ και εναλλάσσει τα $|\psi_3\rangle, |\psi_4\rangle$, εδώ αφήνει αμετάβλητα τα $|\phi_1\rangle, |\phi_3\rangle$ και εναλλάσσει τα $|\phi_2\rangle, |\phi_4\rangle$: με την κωδικοποίηση $|+x\rangle$ για το «0» και $|-x\rangle$ για το «1», η πύλη αντιστρέφει το πρώτο qubit όταν το δεύτερο βρίσκεται στο «1», δηλαδή συμπεριφέρεται σαν οι ρόλοι του qubit-ελέγχου και του qubit-στόχου να έχουν ανταλλάξει. Όπως θα δείξουμε στο επόμενο παράδειγμα, πρόκειται για μία πραγματική ισοδυναμία κυκλωμάτων.

Παράδειγμα 4.6: Ισοδυναμία κυκλωμάτων: η CNOT με αντεστραμμένους ρόλους ελέγχου/στόχου

Στο παράδειγμα 4.5 είδαμε ότι η CNOT, εκφρασμένη στη βάση $|+x\rangle, |-x\rangle$, «φαίνεται» σαν οι ρόλοι του qubit-ελέγχου και του qubit-στόχου να έχουν ανταλλάξει. Θα δείξουμε τώρα ότι πρόκειται για μία

Το κύκλωμα είναι η CNOT με μία πύλη H σε κάθε qubit πριν και μετά από αυτήν· δύο πύλες που δρουν ταυτόχρονα, από μία σε κάθε qubit, αντιστοιχούν στον ταυνοστικό τελεστή τους, οπότε ολόκληρο το κύκλωμα περιγράφεται από το γινόμενο

Θα δείξουμε ότι ισούνται με την «αντεστραμμένη» CNOT, δηλαδή την πύλη με δεύτερο qubit έλεγχο και πρώτο στόχο, που εφαρμόζει την X στο πρώτο qubit όταν το δεύτερο βρίσκεται στην κατάσταση $|1\rangle$. ας τη συμβολίσουμε προσωρινά με $\tilde{D}$, όπως την αντίστοιχη αντεστραμμένη πύλη του παραδείγματος 4.4. Από τον ορισμό της, στις $|\psi_1\rangle = |00\rangle$ και $|\psi_3\rangle = |10\rangle$ το δεύτερο qubit είναι $|0\rangle$, οπότε δεν συμβαίνει τίποτα, ενώ στις $|\psi_2\rangle = |01\rangle$ και $|\psi_4\rangle = |11\rangle$ είναι $|1\rangle$, οπότε αντιστρέφεται το πρώτο,

οπότε, με στήλες τις εικόνες των $|\psi_1\rangle, \dots, |\psi_4\rangle$,

Ας υπολογίσουμε τώρα απευθείας το γινόμενο των τριών πινάκων. Για τον $\mathbf{H} \otimes \mathbf{H}$, ο κανόνας «μπλοκ» του ταυστικού γινομένου (παράδειγμα 2.12) αντικαθιστά κάθε στοιχείο του $\mathbf{H}$ με το αντίστοιχο πολλαπλάσιο του $\mathbf{H}$,

ενώ η CNOT είναι ήδη γνωστή από το παράδειγμα 4.1,

Ο πρώτος πολλαπλασιασμός, $(\mathbf{H} \otimes \mathbf{H}) \mathbf{C}_1^1(\mathbf{X})$, γίνεται σχεδόν αυτόματα. Σε κάθε γινόμενο πινάκων, η στήλη k του αποτελέσματος είναι ο αριστερός παράγοντας εφαρμοσμένος στη στήλη k του δεξιού. Εδώ, η τρίτη στήλη της $\mathbf{C}_1^1(\mathbf{X})$ είναι το διάνυσμα του $|\psi_4\rangle$ και η τέταρτη του $|\psi_3\rangle$ (η CNOT εναλλάσσει τα $|\psi_3\rangle, |\psi_4\rangle$, παράδειγμα 4.1), ενώ οι δύο πρώτες είναι τα διανύσματα των $|\psi_1\rangle, |\psi_2\rangle$. Άρα οι στήλες του γινομένου είναι, με τη σειρά, η πρώτη, η δεύτερη, η τέταρτη και η τρίτη στήλη του $\mathbf{H} \otimes \mathbf{H}$,

Απομένει ένας συνηθισμένος πολλαπλασιασμός 4×4 . Επειδή όλα τα στοιχεία και των δύο παραγόντων είναι $\pm \frac{1}{2}$, κάθε στοιχείο του γινομένου είναι το ένα τέταρτο ενός αθροίσματος τεσσάρων όρων ± 1 : για παράδειγμα, το στοιχείο της πρώτης γραμμής και πρώτης στήλης είναι

9

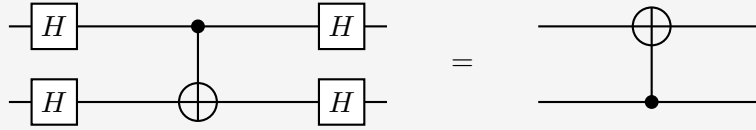
ενώ το στοιχείο της πρώτης γραμμής και δεύτερης στήλης προκύπτει από τη γραμμή $\frac{1}{2}(1, 1, 1, 1)$ του ενδιάμεσου γινομένου και τη στήλη $\frac{1}{2}(1, -1, 1, -1)^T$ του $\mathbf{H} \otimes \mathbf{H}$, είναι

$$\frac{1}{4}(1 - 1 + 1 - 1) = 0.$$

Εκτελώντας τον ίδιο υπολογισμό για όλες τις δεκαέξι θέσεις,

$$(\mathbf{H} \otimes \mathbf{H}) C_1^1(\mathbf{X}) (\mathbf{H} \otimes \mathbf{H}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \mathbf{D},$$

δηλαδή ακριβώς ο πίνακας της αντεστραμμένης CNOT που υπολογίσαμε στην αρχή. Έχουμε λοιπόν την ισοδυναμία κυκλωμάτων



Δεν είναι τυχαίο που ο πίνακας του γινομένου ταυτίζεται και με τον πίνακα που είχαμε βρει για την CNOT στη βάση $|+x\rangle, |-x\rangle$ στο παράδειγμα 4.5: η αλλαγή βάσης από τα $|\psi_p\rangle$ στα $|\phi_p\rangle = (\hat{H} \otimes \hat{H}) |\psi_p\rangle$ πραγματοποιείται, σε επίπεδο πινάκων, ακριβώς με σύζευξη από $\mathbf{H} \otimes \mathbf{H}$ πριν και μετά (αποτέλεσμα 1.11) — γι' αυτό και εκεί η CNOT «φάνηκε» με αντεστραμμένους ρόλους. Ας προσέξουμε μόνο τη διαφορά από την αντι-ελεγχόμενη πύλη της αρχής της ενότητας: εκεί οι δύο X πλαισίωναν μόνο το qubit-ελέγχου, εδώ οι δύο $\hat{H}$ πλαισιώνουν και τα δύο qubit, και η «αντεστραμμένη» πύλη που προκύπτει δεν αλλάζει τότε ενεργοποιείται η CNOT, αλλά ανταλλάσσει τους ρόλους ελέγχου και στόχου.

Παράδειγμα 4.7: Ελεγχόμενη καθολική φάση

Ας θεωρήσουμε τον πίνακα

$$\mathbf{A} = \begin{bmatrix} e^{j\alpha} & 0 \\ 0 & e^{j\alpha} \end{bmatrix} = e^{j\alpha} \mathbf{I}_2,$$

δηλαδή μία καθολική φάση $e^{j\alpha}$ πάνω σε ένα qubit, μία πύλη που, εφαρμοσμένη μόνη της, δεν έχει καμία παρατηρήσιμη φυσική συνέπεια (ενότητα 3.9). Ας δούμε τι συμβαίνει όμως αν την κάνουμε *ελεγχόμενη*, $\hat{C}_1^1(\hat{A})$.

Από το γενικό αποτέλεσμα 4.1, με $\mathbf{V} = \mathbf{A} = e^{j\alpha} \mathbf{I}_2$,

$$C_1^1(\mathbf{A}) = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & e^{j\alpha} \mathbf{I}_2 \end{bmatrix}.$$

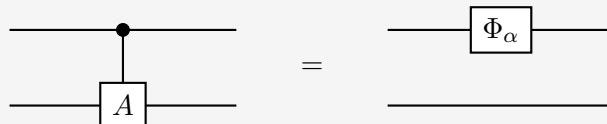
Ας ορίσουμε τώρα τον πίνακα ενός qubit

$$\Phi_\alpha = \begin{bmatrix} 1 & 0 \\ 0 & e^{j\alpha} \end{bmatrix},$$

και ας εφαρμόσουμε την Φ_α μόνο στο πρώτο qubit, αφήνοντας το δεύτερο εντελώς ανεπηρέαστο, δηλαδή τον πίνακα $\Phi_\alpha \otimes \mathbf{I}_2$. Χρησιμοποιώντας τον κανόνα κατασκευής «μπλοκ» για το ταυστικό γινόμενο πινάκων (παράδειγμα 2.12),

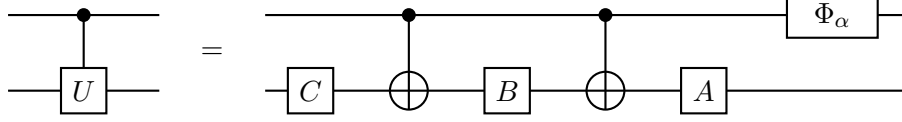
$$\Phi_\alpha \otimes \mathbf{I}_2 = \begin{bmatrix} 1 \cdot \mathbf{I}_2 & 0 \cdot \mathbf{I}_2 \\ 0 \cdot \mathbf{I}_2 & e^{j\alpha} \cdot \mathbf{I}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & e^{j\alpha} \mathbf{I}_2 \end{bmatrix},$$

δηλαδή ακριβώς ο ίδιος πίνακας με τον $C_1^1(\mathbf{A})$! Έχουμε λοιπόν την ισοδυναμία κυκλωμάτων



Το αποτέλεσμα αυτό είναι ιδιαίτερα διαφωτιστικό: μία καθολική φάση $A = e^{j\alpha} I_2$, που από μόνη της δεν έχει καμία φυσική σημασία, γίνεται παρατηρήσιμη μόλις την κάνουμε ελεγχόμενη: μετατρέπεται τότε σε μία συνηθισμένη (σχετική) πύλη φάσης Φ_α πάνω στο ίδιο το qubit-ελέγχου, χωρίς να χρειάζεται καν να αγγίζουμε το qubit-στόχο.

Ο συνδυασμός της αναπαράστασης $U = e^{j\alpha} A X B X C$ (αποτέλεσμα 3.11) με το προηγούμενο παράδειγμα μας επιτρέπει τώρα να κατασκευάσουμε την ελεγχόμενη εκδοχή $\hat{C}_1^1(\hat{U})$ οποιασδήποτε πύλης U ενός qubit, χρησιμοποιώντας μόνο CNOT και τις (μη ελεγχόμενες) πύλες ενός qubit A, B, C, Φ_α :



Ας δούμε γιατί το κύκλωμα αυτό υλοποιεί πράγματι την $\hat{C}_1^1(\hat{U})$. Αν το qubit-ελέγχου είναι στην κατάσταση $|0\rangle$, καμία από τις δύο CNOT δεν ενεργοποιείται, οπότε το qubit-στόχος το οποίο έστω ότι είναι στην κατάσταση $|\psi\rangle$ περνάει διαδοχικά από τις πύλες C, B και A και επομένως η τελική κατάσταση του προκύπτει ως γινόμενο των αντίστοιχων πινάκων

$$|\phi\rangle = ABC |\psi\rangle$$

Όπως όμως έχουμε δει στο πλαίσιο του αποτελέσματος 3.11 $ABC = I_2$ και επομένως

$$|\phi\rangle = |\psi\rangle$$

δηλαδή παραμένει αμετάβλητο, ενώ το qubit-ελέγχου παραμένει επίσης $|0\rangle$ αφού $\hat{\Phi}_\alpha |0\rangle = |0\rangle$. Στην περίπτωση αυτή επομένως το κύκλωμα συμπεριφέρεται ακριβώς όπως η $\hat{C}_1^1(\hat{U})$ όταν $q_1 = 0$.

Αν το qubit-ελέγχου είναι στην κατάσταση $|1\rangle$, και οι δύο CNOT ενεργοποιούνται, οπότε τώρα

$$|\phi\rangle = A X B X C |\psi\rangle$$

όμως στο πλαίσιο του αποτελέσματος 3.11, έχουμε δει ότι

$$A X B X C = e^{-j\alpha} U$$

οπότε

$$|\phi\rangle = e^{-j\alpha} U |\psi\rangle$$

μία εκδοχή της U με «λανθασμένη» καθολική φάση $e^{-j\alpha}$, ενώ ταυτόχρονα το qubit-ελέγχου αποκτά, μέσω της Φ_α , ακριβώς την ελλείπουσα φάση $e^{j\alpha}$, αφού $\hat{\Phi}_\alpha |1\rangle = e^{j\alpha} |1\rangle$ (δείτε το παράδειγμα 4.7). Εμάς όμως μας ενδιαφέρει η συνολική κατάσταση των δύο qubits:

$$|1\rangle |\psi\rangle \xrightarrow{A X B X C} |1\rangle \otimes (e^{-j\alpha} U |\psi\rangle) \xrightarrow{\Phi_\alpha} e^{-j\alpha} e^{j\alpha} |1\rangle U |\psi\rangle = U |\psi\rangle$$

Δείξαμε λοιπόν ότι δύο πύλες CNOT, μαζί με τέσσερις πύλες ενός qubit A, B, C, Φ_α , αρκούν για να κατασκευάσουμε την ελεγχόμενη εκδοχή $C_1^1(U)$ οποιασδήποτε πύλης U ενός qubit.

Παράδειγμα 4.8: Ελεγχόμενη πύλη Hadamard μέσω CNOT

Ας εφαρμόσουμε την παραπάνω κατασκευή στη συγκεκριμένη περίπτωση $U = H$.

Οι γωνίες Euler της H υπολογίστηκαν ήδη στο παράδειγμα 3.34, όπου είχαμε βρει $\alpha = \pi/2$, ενώ οι αντίστοιχοι πίνακες A, B, C υπολογίστηκαν στο παράδειγμα 3.38,

$$A = -R_y\left(\frac{\pi}{4}\right) = \begin{bmatrix} -\cos(\pi/8) & \sin(\pi/8) \\ -\sin(\pi/8) & -\cos(\pi/8) \end{bmatrix},$$

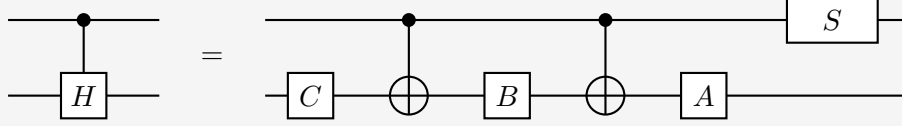
$$B = R_y\left(-\frac{\pi}{4}\right) R_z\left(-\frac{\pi}{2}\right) = \frac{1}{\sqrt{2}} \begin{bmatrix} \cos(\pi/8)(1+j) & \sin(\pi/8)(1-j) \\ -\sin(\pi/8)(1+j) & \cos(\pi/8)(1-j) \end{bmatrix},$$

$$C = -R_z\left(\frac{\pi}{2}\right) = \frac{1}{\sqrt{2}} \begin{bmatrix} j-1 & 0 \\ 0 & -1-j \end{bmatrix}.$$

Απομένει να υπολογίσουμε την Φ_α (παράδειγμα 4.7) για $\alpha = \pi/2$,

$$\Phi_{\pi/2} = \begin{bmatrix} 1 & 0 \\ 0 & e^{j\pi/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & j \end{bmatrix} = S,$$

δηλαδή ακριβώς η ήδη γνωστή μας πύλη φάσης S . Η ελεγχόμενη πύλη Hadamard, $C_1^1(H)$, υλοποιείται λοιπόν με το κύκλωμα



δηλαδή με δύο CNOT, τις μη ελεγχόμενες πύλες ενός qubit A, B, C πάνω στο qubit-στόχο, και μία απλή πύλη S πάνω στο ίδιο το qubit-ελέγχου, χωρίς να χρειαστεί καμία νέα «ελεγχόμενη» κατασκευή πέρα από τις δύο CNOT.

4.2 Κώδικας Python: η βιβλιοθήκη qulib2.py

Μέχρι τώρα η βιβλιοθήκη qulib.py (ενότητες 3.1–3.14) περιείχε τις κλάσεις Qubit και SingleQubitGate, σχεδιασμένες γύρω από ένα και μόνο qubit. Τώρα που περάσαμε σε πύλες πολλαπλών qubit, χρειαζόμαστε έναν κώδικα που να αναπαριστά καταστάσεις και τελεστές οποιοδήποτε αριθμού qubit, και είναι μια καλή ευκαιρία να αναδιοργανώσουμε λίγο και την ιεραρχία των κλάσεων. Ξεκινάμε λοιπόν μία δεύτερη, ελαφρώς αναδιαρθρωμένη έκδοση της βιβλιοθήκης, στο αρχείο qulib2.py.

Για αρχή χρειαζόμαστε κάποιο τρόπο να αναπαραστήσουμε καταστάσεις που αντιστοιχούν σε πολλαπλά qubits. Για το σκοπό αυτό χρησιμοποιούμε την κλάση QuantumState που αναπαριστά μία κβαντική κατάσταση n qubit. Υπάρχουν 2^n πιθανές καταστάσεις της μορφής $|q_1\rangle \cdots |q_n\rangle$ όπου τα qubits είναι πλήρως καθορισμένα και στην ουσία κάθε μία από αυτές παίζει το ρόλο μίας κατάστασης βάσης:

$$|\psi_q\rangle = |q_1\rangle \otimes \cdots \otimes |q_n\rangle$$

όπου τα $q_1, \dots, q_n$ είναι στην ουσία η αναπαράσταση του $q-1$ στο δυαδικό θεωρώντας ότι το q_1 είναι το πιο σημαντικό ψηφίο:

$$q-1 = 2^{n-1}q_1 + 2^{n-2}q_2 + \cdots + q_n$$

Μία οποιαδήποτε κβαντική κατάσταση $|\psi\rangle$ θα γράφεται ως γραμμικός συνδυασμός των $|\psi_q\rangle$:

$$|\psi\rangle = \sum_{q=1}^{2^n} a_q |\psi_q\rangle$$

Για να αποθηκεύσουμε πλήρως την κατάσταση $|\psi\rangle$ πρέπει να κρατήσουμε τα 2^n πλάτη a_q σε έναν πίνακα numpy self.a, με τον δείκτη του πίνακα i να αντιστοιχεί στον συνδυασμό bit μέσω της συνηθισμένης μετατροπής δυαδικού σε δεκαδικό. Αυτό γίνεται επειδή οι πίνακες της numpy αριθμούνται ώστε το πρώτο στοιχείο να είναι το $i=0$ και όχι το $i=1$. Επομένως το στοιχείο self.a[0] αντιστοιχεί στο a_1 , το self.a[1] αντιστοιχεί στο a_2 κ.ο.κ. Ο δείκτης του πίνακα σχετίζεται με την δυαδική αναπαράσταση ως εξής:

$$i = 2^{n-1}q_1 + 2^{n-2}q_2 + \cdots + q_n$$

Έτσι για παράδειγμα η κατάσταση στην περίπτωση όπου $n=3$, δηλαδή έχουμε $2^3 = 8$ καταστάσεις βάσεις $|\psi_q\rangle$ και η κατάσταση $|\psi_4\rangle$ αντιστοιχεί στην

$$|\psi_4\rangle = |0\rangle \otimes |1\rangle \otimes |1\rangle = |011\rangle$$

εφόσον για $q=4$, η δυαδική αναπαράσταση του $q-1=3$ είναι το 011. Το πλάτος a_4 αποθηκεύεται στον πίνακα self.a στο στοιχείο self.a[3].

Για ευκολία, ο constructor `__init__` δέχεται τρεις εναλλακτικές μορφές παραμέτρων:

- έναν ακέραιο n που στην περίπτωση αυτή αρχικοποιεί την κβαντική κατάσταση όπου όλα τα qubit είναι μηδέν, $|0\rangle \otimes \dots \otimes |0\rangle$.
- ένα λεξικό Python dict με τα μη μηδενικά πλάτη ανά bit-string (π.χ. `QuantumState({'000': 1, '010': 1})`)
- απευθείας έναν πίνακα numpy που περιέχει όλα τα πλάτη a_q των κυματοσυναρτήσεων βάσης.

Σε κάθε περίπτωση η κατάσταση κανονικοποιείται αυτόματα ώστε $|a| = 1$, εκτός αν περάσουμε ρητά `normalize=False`, κάτι χρήσιμο όταν θέλουμε να κρατήσουμε ένα μη κανονικοποιημένο αποτέλεσμα, όπως θα δούμε παρακάτω στην περίπτωση των τελεστών προβολής.

Η κλάση προσφέρει επίσης μεθόδους όπως η `to_dict()` που επιστρέφει τα μη μηδενικά πλάτη ως λεξικό της Python, την μέθοδο `to_vec()` που επιστρέφει τα πλάτη της κβαντικής κατάστασης στο `self.a` ως ένα numpy array που αντιστοιχεί σε διάνυσμα-στήλη. Επίσης έχει γίνει και overloading των τελεστών έτσι ώστε το εσωτερικό γινόμενο $\langle\psi|\phi\rangle$ να γράφεται μέσω του τελεστή `| (psi | phi)`, και το ταυυστικό γινόμενο $|\psi\rangle \otimes |\phi\rangle$ μέσω του τελεστή `** (psi ** phi)`, σύμφωνα με τον ορισμό της ενότητας 2.9.

```
class QuantumState:
    def __init__(self, spec, normalize=True):
        if isinstance(spec, int):
            self.n_qubits = spec
            self.a = np.zeros(2 ** self.n_qubits, dtype=complex)
            self.a[0] = 1
        elif isinstance(spec, dict):
            lengths = {len(bits) for bits in spec}
            if len(lengths) != 1:
                raise ValueError("all basis-state keys must have the same length")
            self.n_qubits = lengths.pop()
            self.a = np.zeros(2 ** self.n_qubits, dtype=complex)
            for bits, amplitude in spec.items():
                self.a[int(bits, 2)] = amplitude
            if normalize:
                norm = np.linalg.norm(self.a)
                if norm == 0:
                    raise ValueError("state cannot be all zero")
                self.a = self.a / norm
        else:
            a = np.array(spec, dtype=complex)
            dim = a.shape[0]
            n = int(round(np.log2(dim)))
            if 2 ** n != dim:
                raise ValueError("array length must be a power of 2")
            self.n_qubits = n
            if normalize:
                norm = np.linalg.norm(a)
                if norm == 0:
                    raise ValueError("state cannot be all zero")
                self.a = a / norm
            else:
                self.a = a

    # <todictstart>
    def to_dict(self, tol=1e-10):
        n = self.n_qubits
        return {
            format(i, f"0{n}b"): amplitude
            for i, amplitude in enumerate(self.a)
            if abs(amplitude) > tol
        }
```

```

# <todictend>

# <tovecstart>
def to_vec(self):
    return self.a.reshape(-1, 1)
# <tovecend>

# <strreprstart>
def _truncated_str(self, max_terms=6):
    items = list(self.to_dict().items())
    parts = [f"{bits}: {amplitude:.4f}" for bits, amplitude in items[:max_terms]]
    if len(items) > max_terms:
        parts.append("...")
    return type(self).__name__ + "(" + ", ".join(parts) + ")"

def __str__(self):
    return self._truncated_str()

def __repr__(self):
    return self._truncated_str()
# <strreprend>

# <orstart>
def __or__(self, other):
    if not isinstance(other, QuantumState):
        return NotImplemented
    if self.n_qubits != other.n_qubits:
        raise ValueError("states must have the same number of qubits")
    return np.vdot(self.a, other.a)
# <orend>

# <tensorstart>
def tensor(self, other):
    return QuantumState(np.kron(self.a, other.a))

def __pow__(self, other):
    if not isinstance(other, QuantumState):
        return NotImplemented
    return self.tensor(other)
# <tensorend>

```

Δεδομένου ότι τα απλά qubit είναι μία ειδική περίπτωση του QuantumState είναι απλώς η ειδική περίπτωση $n = 1$ μίας QuantumState είναι χρήσιμο να υλοποιήσουμε ξανά την κλάση Qubit ως μία υποκλάση της QuantumState, με constructor που περνάει τα δύο πλάτη ως QuantumState({'0': a, '1': b}). Έτσι η Qubit κληρονομεί όλα τα παραπάνω (to_dict(), εσωτερικό και ταυυστικό γινόμενο, κ.λπ.), και προσθέτει μόνο ό,τι είναι γνήσια ειδικό για ένα qubit: τις μεθόδους bloch() και set_bloch() που είδαμε στην ενότητα 3.1, μεταφρασμένες τώρα ώστε να δουλεύουν πάνω στο κληρονομημένο self.a.

```

class Qubit(QuantumState):
    def __init__(self, a, b):
        super().__init__({'0': a, "1": b})

    @property
    def state(self):
        return self.a

# <blochstart>
def bloch(self):
    a, b = self.a[0], self.a[1]
    x = (a.conjugate() * b + b.conjugate() * a).real

```

```

y = -2 * (a * b.conjugate()).imag
z = abs(a) ** 2 - abs(b) ** 2
return x, y, z
# <blochend>

# <setblochstart>
def set_bloch(self, x, y, z):
    theta = np.arccos(z)
    phi = np.angle(x + 1j * y)
    a = np.cos(theta / 2)
    b = np.exp(1j * phi) * np.sin(theta / 2)
    self.a = np.array([a, b], dtype=complex)
# <setblochend>

```

Στη συνέχεια θέλουμε να υλοποιήσουμε κλάσεις που έχουν να κάνουν με την επεξεργασία που κάνουμε πάνω στα qubits. Ξεκινάμε με την κλάση `Operator` η οποία υλοποιεί ένα κβαντικό τελεστή ο οποίος δρα πάνω σε μία κβαντική κατάσταση όπως αυτή περιγράφεται από μία `QuantumState` και παράγει μία άλλη κβαντική κατάσταση. Δίνουμε δύο τρόπους να ορίσουμε έναν `Operator` όπως φαίνεται και από την μέθοδο `__init__`. Ο ένας τρόπος είναι να περάσουμε ως παράμετρο απευθείας τον πίνακα U που περιγράφει τον τελεστή. Ο άλλος είναι να περάσουμε μία συνάρτηση `func` η οποία περιγράφει τον τελεστή ως συνάρτηση: πως δηλαδή παίρνει μία `QuantumState` στην είσοδο και πως υπολογίζεται η `QuantumState` στην έξοδο. Στη δεύτερη περίπτωση, η μέθοδος `calc_U()` υπολογίζει (και αποθηκεύει) τον πίνακα U εφαρμόζοντας τη συνάρτηση σε καθεμία από τις 2^n βασικές καταστάσεις $|0 \cdots 0\rangle, |0 \cdots 1\rangle, \dots$ και βάζοντας τις εικόνες τους ως στήλες, ακριβώς η στάνταρ κατασκευή «πίνακας τελεστή από τις εικόνες μίας βάσης» (ενότητα 1.23). Η μέθοδος `out(state)` εφαρμόζει τον τελεστή σε μία κατάσταση, και το `**` ορίζει το ταυυστικό γινόμενο δύο τελεστών, $\hat{A} \otimes \hat{B}$, μέσω του πίνακα $A \otimes B$.

```

class Operator:
    def __init__(self, U=None, func=None, n_qubits=None):
        if U is not None and func is not None:
            raise ValueError("provide either U or func, not both")
        if U is not None:
            U = np.array(U, dtype=complex)
            dim = U.shape[0]
            if U.shape != (dim, dim):
                raise ValueError("U must be a square matrix")
            n = int(round(np.log2(dim)))
            if 2 ** n != dim:
                raise ValueError("U's dimension must be a power of 2")
            self.U = U
            self.func = None
            self.n_qubits = n
        elif func is not None:
            if n_qubits is None:
                raise ValueError("n_qubits must be given when using func")
            self.U = None
            self.func = func
            self.n_qubits = n_qubits
        else:
            raise ValueError("must provide either U or func")

# <calcustart>
def calc_U(self):
    if self.func is None:
        return self.U
    dim = 2 ** self.n_qubits
    U = np.zeros((dim, dim), dtype=complex)
    for j in range(dim):
        basis_a = np.zeros(dim, dtype=complex)

```

```

        basis_a[j] = 1
        out = self.func(QuantumState(basis_a))
        U[:, j] = out.a
        self.U = U
        return U
# <calcuend>

# <operatoroutstart>
def out(self, state):
    if self.func is not None:
        return self.func(state)
    return QuantumState(self.U @ state.a)
# <operatoroutend>

# <operatorontensorstart>
def tensor(self, other):
    return Operator(U=np.kron(self.calc_U(), other.calc_U()))

def __pow__(self, other):
    if not isinstance(other, Operator):
        return NotImplemented
    return self.tensor(other)
# <operatorontensorend>

```

Η ίδια η `Operator` δεν επιβάλλει τίποτα για το αν ο U είναι μοναδιαίος, και αυτό είναι σκόπιμο, αφού όχι κάθε τελεστής που μας ενδιαφέρει είναι απαραίτητα μοναδιαίος (π.χ. ένας τελεστής προβολής, βλέπε παρακάτω). Οι κβαντικές πύλες όμως πρέπει πάντα να είναι μοναδιαίοι τελεστές, οπότε η κλάση `Gate` είναι μία υποκλάση της `Operator` που προσθέτει ακριβώς αυτόν τον έλεγχο, $U^T U = I$, όταν ο πίνακας U δίνεται απευθείας:

```

class Gate(Operator):
    def __init__(self, U=None, func=None, n_qubits=None):
        super().__init__(U=U, func=func, n_qubits=n_qubits)
        if self.U is not None and not np.allclose(self.U.conj().T @ self.U, np.eye(self.U.
            shape[0])):
            raise ValueError("U must be unitary")

# <gatematmulstart>
def __matmul__(self, other):
    if isinstance(other, Gate):
        if self.n_qubits != other.n_qubits:
            raise ValueError("gates must act on the same number of qubits")
        return Gate(U=self.calc_U() @ other.calc_U())
    if isinstance(other, QuantumState):
        return self.out(other)
    return NotImplemented
# <gatematmulend>

# <gatetensoroverridestart>
def tensor(self, other):
    return Gate(U=np.kron(self.calc_U(), other.calc_U()))
# <gatetensoroverrideend>

# <twoleveldecompositionstart>
def two_level_decomposition(self, tol=1e-10):
    N = 2 ** self.n_qubits
    M = self.calc_U().copy()
    eliminations = []
    for ref in range(N - 2):
        for row in range(ref + 1, N):

```

```

        a, b = M[ref, ref], M[row, ref]
        if abs(b) < tol:
            continue
        norm = np.hypot(abs(a), abs(b))
        x, y = a / norm, b / norm
        A = np.eye(N, dtype=complex)
        A[np.ix_([ref, row], [ref, row])] = [[x.conjugate(), y.conjugate()], [-y, x]]
        M = A @ M
        eliminations.append(A)
    factors = [Gate(U=A.conj().T) for A in eliminations]
    factors.append(Gate(U=M))
    return factors
# <twoleveldecompositionend>

```

Η SingleQubitGate που ήδη γνωρίζουμε (ενότητα 3.9) παραμένει ουσιαστικά αμετάβλητη, μαζί με όλες τις υποκλάσεις της όπως η BlochRotationX και τις βασικές πύλες όπως η GateX. Η μόνη διαφορά είναι ότι πλέον κληρονομεί από την Gate (αντί να είναι εντελώς ανεξάρτητη κλάση όπως στο qulib.py), κερδίζοντας έτσι μεταξύ άλλων και το ταχυστικό γινόμενο πυλών.

Ένας τελεστής *προβολής* υλοποιεί τις προβολές που σχετίζονται με τις κβαντικές μετρήσεις. Όπως είδαμε μπορεί να έχουμε προβολές της μορφής $|\psi_1\rangle\langle\psi_1|$ που προβάλλουν τις κβαντικές καταστάσεις ως προς την συγκεκριμένη κατάσταση $|\psi_1\rangle$. Ωστόσο όπως είδαμε οι τελεστές προβολή μπορεί να είναι μίας γενικότερης μορφής άθροισματος πολλών επιμέρους απλών προβολών $|\psi_i\rangle\langle\psi_i|$:

$$\hat{P} = |\psi_1\rangle\langle\psi_1| + \dots + |\psi_Q\rangle\langle\psi_Q|$$

Η κλάση ProjectOperator είναι λοιπόν μία δεύτερη υποκλάση της Operator που δέχεται στον constructor μία ή περισσότερες QuantumState και κατασκευάζει τον πίνακα $\mathbf{P}$ ως άθροισμα πινάκων προβολής $|\psi_i\rangle\langle\psi_i|$. Η μέθοδος out() γίνεται εδώ *overridden* ώστε να περνάει normalize=False στην QuantumState. Σε αντίθεση με μία πύλη, όπου στην κατάσταση εξόδου η νόρμα διατηρείται αυτόματα, το αποτέλεσμα μίας προβολής $\hat{P}|\phi\rangle$ παράγεται εν γένει μη κανονικοποιημένο, και το μέτρο του κωδικοποιεί ακριβώς την πιθανότητα της αντίστοιχης μέτρησης.

```

class ProjectOperator(Operator):
    def __init__(self, *states):
        if len(states) == 1 and isinstance(states[0], (list, tuple)):
            states = tuple(states[0])
        if len(states) == 0:
            raise ValueError("need at least one state to project onto")
        n_qubits = states[0].n_qubits
        if any(psi.n_qubits != n_qubits for psi in states):
            raise ValueError("all states must have the same number of qubits")
        self.states = states
        dim = 2 ** n_qubits
        P = np.zeros((dim, dim), dtype=complex)
        for psi in states:
            v = psi.a.reshape(-1, 1)
            P += v @ v.conj().T
        super().__init__(U=P)

# <projectoutstart>
def out(self, state):
    return QuantumState(self.U @ state.a, normalize=False)
# <projectoutend>

```

Τέλος έχει νόημα να υλοποιήσουμε τις ελεγχόμενες πύλες αυτής της ενότητας που έχουν ένα qubit έλεγχο και ένα qubit στόχου και υλοποιούνται απευθείας ως υποκλάσεις της Gate. Η ControlledU(V) υλοποιεί τη

γενική πύλη $C_1^1(U)$ 4.1 με πίνακα:

$$C_1^1(V) = \begin{bmatrix} I_2 & 0 \\ 0 & V \end{bmatrix}$$

και οι CNOT, CZ, CH είναι απλώς οι ειδικές περιπτώσεις όταν $V = X, Z, H$. Τέλος ορίζουμε και μερικές επιπλέον πύλες τις SWAP, Toffoli και Fredkin κατασκευάζουν απευθείας τον πλήρη πίνακά τους και τις οποίες θα δούμε στη συνέχεια.

```
class ControlledU(Gate):
    def __init__(self, V):
        V = np.array(V, dtype=complex)
        U = np.zeros((4, 4), dtype=complex)
        U[:2, :2] = np.eye(2)
        U[2:, 2:] = V
        super().__init__(U=U)
```

```
class CNOT(ControlledU):
    def __init__(self):
        super().__init__(GateX().U)
```

```
class CZ(ControlledU):
    def __init__(self):
        super().__init__(GateZ().U)
```

```
class CH(ControlledU):
    def __init__(self):
        super().__init__(GateH().U)
```

```
class SWAP(Gate):
    def __init__(self):
        U = np.array([
            [1, 0, 0, 0],
            [0, 0, 1, 0],
            [0, 1, 0, 0],
            [0, 0, 0, 1],
        ], dtype=complex)
        super().__init__(U=U)
```

```
class Toffoli(Gate):
    def __init__(self):
        U = np.eye(8, dtype=complex)
        U[6:8, 6:8] = GateX().U
        super().__init__(U=U)
```

```
class Fredkin(Gate):
    def __init__(self):
        U = np.eye(8, dtype=complex)
        U[5:7, 5:7] = SWAP().U[1:3, 1:3]
        super().__init__(U=U)
```

Ας επιβεβαιώσουμε τώρα αριθμητικά, με τη βοήθεια της `qulib2.py`, τα παραδείγματα 4.1–4.8 αυτής της ενότητας. Στα `scripts` που ακολουθούν, ακολουθούμε όσο γίνεται τον ίδιο δρόμο με το κείμενο: υπολογίζουμε τα στοιχεία των πινάκων ως $c_{pq} = \langle \psi_p | \hat{C} | \psi_q \rangle$ μέσω του εσωτερικού γινομένου `|` της `QuantumState`, και όπου χρειάζεται μία πύλη που δεν υπάρχει έτοιμη στη βιβλιοθήκη (η αντι-ελεγχόμενη CNOT ή οι πύλες με αντεστραμμένους ρόλους ελέγχου/στόχου), την ορίζουμε μέσω μίας συνάρτησης `func`, ακριβώς από τη δράση της στις καταστάσεις βάσης, όπως κάναμε και στα αντίστοιχα παραδείγματα.

Το παράδειγμα 4.1 αφορά τη δράση της CNOT σε καθεμία από τις τέσσερις καταστάσεις βάσης $|\psi_1\rangle, \dots, |\psi_4\rangle$, και τα στοιχεία c_{pq} του πίνακά της. Το script βρίσκεται στο αρχείο `python/ex_CNOT_matrix.py`:

```
import numpy as np

from qulib2 import CNOT, QuantumState

np.set_printoptions(precision=4, suppress=True)

# Example 4.1: the CNOT acting on the basis states |psi_1>, ..., |psi_4>
# = |00>, |01>, |10>, |11>, and its matrix elements c_pq = <psi_p|C|psi_q>
cnot = CNOT()
basis = ["00", "01", "10", "11"]
psi = [QuantumState({bits: 1}) for bits in basis]

for q, bits in enumerate(basis):
    out = cnot @ psi[q]
    print(f"C_1^{1(X)}|{bits}> = |{list(out.to_dict())[0]}>")

c = np.array([[psi[p] | (cnot @ psi[q]) for q in range(4)] for p in range(4)])
print(f"c_pq =\n{c.real}")
print(f"c_34 = <psi_3|C|psi_4> = {c[2, 3].real}")

book = np.array([
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, 0, 1],
    [0, 0, 1, 0],
], dtype=complex)
print(f"matches book matrix: {np.allclose(c, book)}")
print(f"matches CNOT().U: {np.allclose(c, cnot.U)}")
```

Το παράδειγμα 4.2 αφορά τον πίνακα της CNOT με άσπρο κυκλάκι, που ορίζουμε εδώ μέσω της δράσης της στις καταστάσεις βάσης, καθώς και η ισοδυναμία της με μία συνηθισμένη CNOT με μία X πριν και μία X μετά στο qubit-ελέγχου, που είδαμε λίγο πριν από το παράδειγμα. Το script βρίσκεται στο αρχείο `python/ex_anti_CNOT_matrix.py`:

```
import numpy as np

from qulib2 import Gate, GateX, CNOT, I, QuantumState

np.set_printoptions(precision=4, suppress=True)

# Example 4.2: the anti-controlled CNOT (white dot), defined through its
# action on the basis states: the second qubit is flipped only when the
# first one is |0>
def anti_cnot_func(state):
    out = {}
    for bits, amplitude in state.to_dict().items():
        q1, q2 = int(bits[0]), int(bits[1])
        if q1 == 0:
            q2 = 1 - q2
        key = f"{q1}{q2}"
        out[key] = out.get(key, 0) + amplitude
    return QuantumState(out)

anti_cnot = Gate(func=anti_cnot_func, n_qubits=2)
basis = ["00", "01", "10", "11"]
```

```

psi = [QuantumState({bits: 1}) for bits in basis]

for q, bits in enumerate(basis):
    out = anti_cnot @ psi[q]
    print(f"C|{bits}> = |{list(out.to_dict())[0]}>")

c = np.array([[psi[p] | (anti_cnot @ psi[q]) for q in range(4)] for p in range(4)])
print(f"c_pq =\n{c.real}")

book = np.array([
    [0, 1, 0, 0],
    [1, 0, 0, 0],
    [0, 0, 1, 0],
    [0, 0, 0, 1],
], dtype=complex)
print(f"matches book matrix: {np.allclose(c, book)}")

# the equivalence claimed just before the example: an anti-controlled
# gate equals an ordinary controlled gate with an X on the control
# qubit, both before and after
X_on_control = GateX() ** I(1)
sandwich = X_on_control @ CNOT() @ X_on_control
print(f"(X (x) I2) C_1^1(X) (X (x) I2) =\n{sandwich.U.real}")
print(f"matches the anti-controlled CNOT: {np.allclose(sandwich.U, anti_cnot.calc_U())}")

```

Το παράδειγμα 4.3 αφορά τη δράση της ελεγχόμενης Hadamard στις καταστάσεις βάσης, το πίνακάς της, και τη δομή του αποτελέσματος 4.1, με τον I_2 πάνω αριστερά και τον H κάτω δεξιά. Το script βρίσκεται στο αρχείο python/ex_CH_matrix.py:

```

import numpy as np

from qulib2 import CH, GateH, QuantumState

np.set_printoptions(precision=4, suppress=True)

# Example 4.3: the controlled-Hadamard acting on the basis states, and
# its matrix built column by column: column q holds the coordinates of
# C_1^1(H)|psi_q> in the basis |psi_p>
ch = CH()
basis = ["00", "01", "10", "11"]
psi = [QuantumState({bits: 1}) for bits in basis]

for q, bits in enumerate(basis):
    out = ch @ psi[q]
    print(f"C_1^1(H)|{bits}> = {out}")

c = np.array([[psi[p] | (ch @ psi[q]) for q in range(4)] for p in range(4)])
print(f"C_1^1(H) =\n{c.real}")

s2 = 1 / np.sqrt(2)
book = np.array([
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, s2, s2],
    [0, 0, s2, -s2],
], dtype=complex)
print(f"matches book matrix: {np.allclose(c, book)}")

# the block structure of result "controlled-V-block": I2 top-left, H bottom-right
print(f"top-left block is I2: {np.allclose(c[:2, :2], np.eye(2))}")
print(f"bottom-right block is H: {np.allclose(c[2:, 2:], GateH().U)}")

```

Το παράδειγμα 4.4 αφορά το πίνακα της $C_1^1(Z)$, και την επιβεβαίωση ότι η πύλη $\hat{D}$ με αντεστραμμένους ρόλους ελέγχου/στόχου έχει τον ίδιο ακριβώς πίνακα. Το script βρίσκεται στο αρχείο `python/ex_CZ_symmetric.py`:

```
import numpy as np

from qulib2 import Gate, CZ, QuantumState

np.set_printoptions(precision=4, suppress=True)

# Example 4.4: C_1^1(Z) is symmetric under swapping the roles of control
# and target qubit
cz = CZ()
print(f"C_1^1(Z) =\n{cz.U.real}")

book = np.diag([1, 1, 1, -1]).astype(complex)
print(f"matches book matrix: {np.allclose(cz.U, book)}")

# the reversed-roles gate D, defined through its action on the basis
# states: Z is applied on the first qubit when the second one is |1>
def D_func(state):
    out = {}
    for bits, amplitude in state.to_dict().items():
        q1, q2 = int(bits[0]), int(bits[1])
        if q2 == 1 and q1 == 1:
            amplitude = -amplitude
        out[bits] = amplitude
    return QuantumState(out)

D = Gate(func=D_func, n_qubits=2)
for bits in ["00", "01", "10", "11"]:
    print(f"D|{bits}> = {D @ QuantumState({bits: 1})}")

print(f"D =\n{D.calc_U().real}")
print(f"D matches C_1^1(Z): {np.allclose(D.calc_U(), cz.U)}")
```

Το παράδειγμα 4.5 δείχνει ότι τα $|\pm x\rangle$ είναι ιδιοκαταστάσεις της $\hat{X}$, ότι τα $|\phi_1\rangle, \dots, |\phi_4\rangle$ είναι ορθοκανονικά, και υπολογίζει τα στοιχεία $c_{pq} = \langle \phi_p | \hat{C}_1^1(\hat{X}) | \phi_q \rangle$ του πίνακα της CNOT στη βάση $|+x\rangle, |-x\rangle$. Τα $|\phi_p\rangle$ κατασκευάζονται με το τανυστικό γινόμενο `**` της QuantumState. Το script βρίσκεται στο αρχείο `python/ex_CNOT_basis_xx.py`:

```
import numpy as np

from qulib2 import CNOT, GateX, QuantumState

np.set_printoptions(precision=4, suppress=True)

# Example 4.5: the CNOT matrix in the basis |phi_1>, ..., |phi_4> built
# from |+x>, |-x>
plus_x = QuantumState({"0": 1, "1": 1})
minus_x = QuantumState({"0": 1, "1": -1})

# |+x>, |-x> are eigenstates of X with eigenvalues +1, -1
X = GateX()
print(f"<+x|X|+x> = {(plus_x | (X @ plus_x)).real:.4f}")
print(f"<-x|X|-x> = {(minus_x | (X @ minus_x)).real:.4f}")
```

```

phi = [
    plus_x ** plus_x,    # |phi_1>
    plus_x ** minus_x,   # |phi_2>
    minus_x ** plus_x,   # |phi_3>
    minus_x ** minus_x,  # |phi_4>
]

# the |phi_p> are orthonormal
gram = np.array([[phi[p] | phi[q] for q in range(4)] for p in range(4)])
print(f"<phi_p|phi_q> = I4: {np.allclose(gram, np.eye(4))}")

cnot = CNOT()
for q in range(4):
    out = cnot @ phi[q]
    overlaps = [(phi[p] | out).real for p in range(4)]
    print(f"C_1^1(X)|phi_{q + 1}> has coordinates {np.round(overlaps, 4)}")

c = np.array([[phi[p] | (cnot @ phi[q]) for q in range(4)] for p in range(4)])
print(f"c_pq =\n{c.real}")

book = np.array([
    [1, 0, 0, 0],
    [0, 0, 0, 1],
    [0, 0, 1, 0],
    [0, 1, 0, 0],
], dtype=complex)
print(f"matches book matrix: {np.allclose(c, book)}")

```

Το παράδειγμα 4.6 αφορά την ισοδυναμία κυκλωμάτων $(H \otimes H) C_1^1(X) (H \otimes H) = D$, βήμα-βήμα όπως στο κείμενο: πρώτα το ενδιαμέσο γινόμενο $(H \otimes H) C_1^1(X)$ και μετά ολόκληρο το κύκλωμα, που συγκρίνεται με την CNOT με αντεστραμμένους ρόλους ελέγχου/στόχου $\hat{D}$ (και με τον πίνακα του παραδείγματος 4.5). Το script βρίσκεται στο αρχείο `python/ex_CNOT_basis_change.py`:

```

import numpy as np

from qulib2 import Gate, GateH, CNOT, QuantumState

np.set_printoptions(precision=4, suppress=True)

# Example 4.6: the circuit equivalence  $(H \otimes H) C_1^1(X) (H \otimes H) = D$ ,
# where D is the CNOT with reversed roles (second qubit control, first
# qubit target), defined through its action on the basis states
def D_func(state):
    out = {}
    for bits, amplitude in state.to_dict().items():
        q1, q2 = int(bits[0]), int(bits[1])
        if q2 == 1:
            q1 = 1 - q1
        key = f"{q1}{q2}"
        out[key] = out.get(key, 0) + amplitude
    return QuantumState(out)

D = Gate(func=D_func, n_qubits=2)
print(f"D =\n{D.calc_U().real}")

HH = GateH() ** GateH()
print(f"H \otimes H =\n{HH.U.real}")

```

```

cnot = CNOT()
first = HH @ cnot
print(f"(H (x) H) C11(X) =\n{first.U.real}")

book_first = np.array([
    [1, 1, 1, 1],
    [1, -1, -1, 1],
    [1, 1, -1, -1],
    [1, -1, 1, -1],
]) / 2
print(f"matches book intermediate product: {np.allclose(first.U, book_first)}")

sandwich = first @ HH
print(f"(H (x) H) C11(X) (H (x) H) =\n{sandwich.U.real}")
print(f"circuit equivalence holds: {np.allclose(sandwich.U, D.calc_U())}")

book = np.array([
    [1, 0, 0, 0],
    [0, 0, 0, 1],
    [0, 0, 1, 0],
    [0, 1, 0, 0],
], dtype=complex)
print(f"matches book matrix (and Example 4.5): {np.allclose(sandwich.U, book)}")

```

Το παράδειγμα 4.7 αφορά μία ελεγχόμενη καθολική φάση ισοδυναμεί με μία απλή πύλη φάσης Φ_α πάνω στο qubit-ελέγχου. Το script βρίσκεται στο αρχείο `python/ex_controlled_phase.py`:

```

import numpy as np

from qulib2 import Gate, ControlledU, I

np.set_printoptions(precision=4, suppress=True)

# Example 4.7: a controlled global phase A = e^(j alpha) I2 is the same as
# applying Phi_alpha on the control qubit alone
alpha = 0.37 * np.pi
A = np.exp(1j * alpha) * np.eye(2)
ca = ControlledU(A)

Phi_alpha = Gate(U=np.diag([1, np.exp(1j * alpha)]))
rhs = Phi_alpha ** I(1)

print(f"C11(A) =\n{ca.U}")
print(f"Phi_alpha (x) I2 =\n{rhs.U}")
print(f"matches: {np.allclose(ca.U, rhs.U)}")

```

Τέλος, έχουμε το παράδειγμα 4.8 όπου επιβεβαιώνουμε πρώτα ότι $\alpha = \pi/2$, ότι οι πίνακες A, B, C που επιστρέφει η `ABC()` ταυτίζονται με αυτούς του κειμένου, και ότι $\Phi_{\pi/2} = S$. Στη συνέχεια ελέγχουμε ότι οι πύλες $C, C_1^1(X), B, C_1^1(X), A$ πάνω στο qubit-στόχο και η S πάνω στο qubit-ελέγχου, εφαρμοσμένες με αυτή τη σειρά, ανακατασκευάζουν ακριβώς την $C_1^1(H)$. Χρησιμοποιούμε το τανυστικό γινόμενο πυλών (`**`) της `Gate` για να τοποθετήσουμε κάθε πύλη ενός qubit στο σωστό qubit του καταχωρητή. Το script βρίσκεται στο αρχείο `python/ex_controlled_H_circuit.py`:

```

import numpy as np

from qulib2 import Gate, GateH, GateS, CNOT, CH, I

np.set_printoptions(precision=4, suppress=True)

```

```

# Example 4.8: the controlled-Hadamard built from two CNOTs, the
# single-qubit gates A, B, C on the target qubit, and Phi_alpha on the
# control qubit
H = GateH()
alpha, beta, gamma, delta = H.euler_angles()
print(f"alpha = {alpha / np.pi:.4f} pi")

A, B, C = H.ABC()
c8, s8 = np.cos(np.pi / 8), np.sin(np.pi / 8)
s2 = 1 / np.sqrt(2)
book_A = np.array([[ -c8, s8], [ s8, -c8]])
book_B = s2 * np.array([[c8 * (1 + 1j), s8 * (1 - 1j)], [-s8 * (1 + 1j), c8 * (1 - 1j)]])
book_C = s2 * np.array([[1j - 1, 0], [0, -1 - 1j]])
print(f"A, B, C match the book: "
      f"{np.allclose(A, book_A) and np.allclose(B, book_B) and np.allclose(C, book_C)}")

# Phi_alpha for alpha = pi/2 is the phase gate S
Phi_alpha = np.diag([1, np.exp(1j * alpha)])
S = GateS()
print(f"Phi_pi/2 = S: {np.allclose(Phi_alpha, S.U)}")

# C, CNOT, B, CNOT, A on the target qubit, and S on the control qubit,
# applied in that order, should reconstruct C_1^1(H)
target_A = I(1) ** Gate(U=A)
target_B = I(1) ** Gate(U=B)
target_C = I(1) ** Gate(U=C)
control_S = S ** I(1)
cnot = CNOT()

circuit = control_S @ target_A @ cnot @ target_B @ cnot @ target_C

print(f"circuit (S)(A)(CNOT)(B)(CNOT)(C) =\n{circuit.U}")
print(f"C_1^1(H) =\n{CH().U}")
print(f"matches: {np.allclose(circuit.U, CH().U)}")

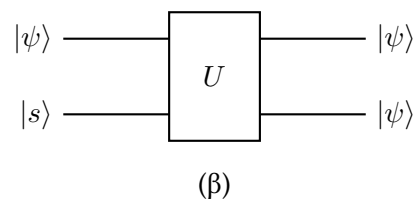
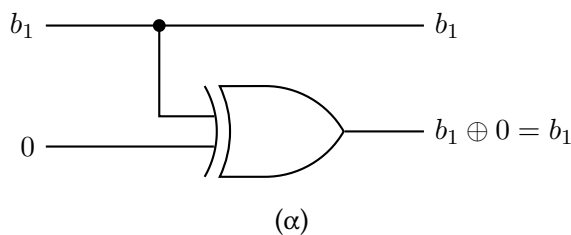
```

4.3 Το θεώρημα μη κλωνοποίησης

Στα κλασικά κυκλώματα είμαστε συνηθισμένοι στην αντιγραφή της πληροφορίας. Αν υποθέσουμε ένα έχουμε δύο bit το b_1 και b_2 είναι εύκολο να αντιγράψουμε το 1ο bit στο δεύτερο bit και να έχουμε $b_1 = b_2$. Θα μπορούσε το δεύτερο bit αρχικά να είναι στην κατάσταση $b_2 = 0$ οπότε η διαδικασία της αντιγραφής είναι η παρακάτω:

$$b_1, 0 \xrightarrow{\text{COPY}} b_1, b_1$$

Η αντιγραφή αυτή υλοποιείται με μία απλή πύλη XOR, όπως φαίνεται στο σχήμα 4.2(α): το b_1 οδηγείται και στην έξοδο και στη μία είσοδο της XOR, ενώ η άλλη είσοδος είναι 0, οπότε η έξοδος της XOR είναι $b_1 \oplus 0 = b_1$.



Σχήμα 4.2: (α) Το κλασικό κύκλωμα αντιγραφής ενός bit με μία πύλη XOR. (β) Το υποθετικό κβαντικό κύκλωμα κλωνοποίησης, που θα έπρεπε να μετατρέπει την $|\psi\rangle |s\rangle$ σε $|\psi\rangle |\psi\rangle$ για κάθε $|\psi\rangle$.

Θα δούμε όμως πως αυτό το πολύ απλό κλασικό κύκλωμα που μαθαίνουμε στο μάθημα της ψηφιακής σχεδίασης δεν έχει κάποιο αντίστοιχο στα κβαντικά κυκλώματα. Όσο και αν φαίνεται περίεργο δεν υπάρχει τρόπος να αντιγράψει κανείς ή να κλωνοποιήσει ένα qubit σε μία γενική κατάσταση $|\psi\rangle = a|0\rangle + b|1\rangle$. Αυτό είναι το *θεώρημα μη κλωνοποίησης (no-cloning theorem)*.

Ας υποθέσουμε, για να καταλήξουμε σε άτοπο, ότι υπάρχει ένας τέτοιος μοναδιαίος τελεστής $\hat{U}$ πάνω σε δύο qubit που κάνει ακριβώς την λειτουργία αυτή της κλωνοποίησης του qubit: το qubit που θέλουμε να αντιγράψουμε, και ένα δεύτερο qubit που ξεκινά πάντα από την ίδια, καθορισμένη κατάσταση $|s\rangle$ (ανεξάρτητη από το $|\psi\rangle$ που αντιγράφουμε, π.χ. $|s\rangle = |0\rangle$, όπως και στην κλασική XOR) εισέρχεται στο κύκλωμα που υλοποιεί τον $\hat{U}$ και στο τέλος καταλήγουμε με δύο ίδια qubit $|\psi\rangle |\psi\rangle$, όπως στο σχήμα 4.2(β):

$$\hat{U}(|\psi\rangle |s\rangle) = |\psi\rangle |\psi\rangle.$$

Σημειώστε ότι ζητάμε ο ίδιος τελεστής $\hat{U}$ να δουλεύει για όλα τα $|\psi\rangle$ ταυτόχρονα, χωρίς να χρειάζεται να ξέρουμε εκ των προτέρων ποιο είναι: αν γνωρίζαμε ήδη ότι το qubit είναι σε μία από δύο συγκεκριμένες, γνωστές καταστάσεις, θα μπορούσαμε απλά να το μετρήσουμε σε μία κατάλληλη βάση και να προετοιμάσουμε ένα αντίγραφο βάσει του αποτελέσματος. Το θεώρημα αφορά τη δυνατότητα αντιγραφής μίας *άγνωστης* υπέρθεσης, χωρίς μέτρηση.

Ας εφαρμόσουμε την παραπάνω σχέση σε δύο διαφορετικές καταστάσεις $|\psi_1\rangle \neq |\psi_2\rangle$,

$$\hat{U}(|\psi_1\rangle |s\rangle) = |\psi_1\rangle |\psi_1\rangle, \quad \hat{U}(|\psi_2\rangle |s\rangle) = |\psi_2\rangle |\psi_2\rangle,$$

και ας πάρουμε το εσωτερικό γινόμενο του αριστερού μέλους της πρώτης με το αριστερό μέλος της δεύτερης. Χρησιμοποιώντας τη μοναδιαιότητα $\hat{U}^T \hat{U} = \hat{I}$ (ενότητα 1.10) και $\langle s|s\rangle = 1$,

$$(\langle \psi_1| \langle s|) \hat{U}^T \hat{U} (|\psi_2\rangle |s\rangle) = (\langle \psi_1| \langle s|) (|\psi_2\rangle |s\rangle) = \langle \psi_1|\psi_2\rangle \langle s|s\rangle = \langle \psi_1|\psi_2\rangle.$$

Το ίδιο εσωτερικό γινόμενο, υπολογισμένο όμως μέσω των δεξιών μελών, δίνει

$$(\langle \psi_1| \langle \psi_1|) (|\psi_2\rangle |\psi_2\rangle) = \langle \psi_1|\psi_2\rangle \langle \psi_1|\psi_2\rangle = \langle \psi_1|\psi_2\rangle^2.$$

Εξισώνοντας τα δύο αποτελέσματα, καταλήγουμε στην

$$\langle \psi_1|\psi_2\rangle = \langle \psi_1|\psi_2\rangle^2.$$

Θέτοντας $x = \langle \psi_1|\psi_2\rangle$ η παραπάνω σχέση γράφεται $x^2 - x = 0$, δηλαδή $x(x - 1) = 0$, οπότε $x = 0$ ή $x = 1$.

Η περίπτωση $x = 1$ αποκλείεται, αφού $|\psi_1\rangle \neq |\psi_2\rangle$. Πράγματι ας υποθέσουμε $\langle \psi_1|\psi_2\rangle = 1$. Τότε ας γράψουμε τα $|\psi_1\rangle$ και $|\psi_2\rangle$ με την βοήθεια συναρτήσεων βάσης $|\phi_i\rangle$,

$$|\psi_1\rangle = \sum_{i=1}^N a_i |\phi_i\rangle$$

$$|\psi_2\rangle = \sum_{i=1}^N b_i |\phi_i\rangle$$

ενώ:

$$\langle \psi_1|\psi_2\rangle = \sum_{i=1}^N a_i^* b_i$$

$$\langle \psi_1|\psi_1\rangle = \sum_{i=1}^N |a_i|^2$$

$$\langle \psi_2 | \psi_2 \rangle = \sum_{i=1}^N |b_i|^2$$

Από την ταυτότητα:

$$|a_i - b_i|^2 = |a_i|^2 + |b_i|^2 - a_i b_i^* - a_i^* b_i$$

Βρίσκουμε:

$$\sum_{i=1}^N |a_i - b_i|^2 = \sum_{i=1}^N |a_i|^2 + \sum_{i=1}^N |b_i|^2 - \sum_{i=1}^N a_i b_i^* - \sum_{i=1}^N a_i^* b_i$$

Οπότε:

$$\sum_{i=1}^N |a_i - b_i|^2 = \langle \psi_1 | \psi_1 \rangle + \langle \psi_2 | \psi_2 \rangle - \langle \psi_1 | \psi_2 \rangle - \langle \psi_2 | \psi_1 \rangle$$

Πάντα θα έχουμε $\langle \psi_1 | \psi_1 \rangle = \langle \psi_2 | \psi_2 \rangle = 1$ εξαιτίας της κανονικοποίησης, ενώ πριν υποθέσαμε ότι $\langle \psi_1 | \psi_2 \rangle = 1$ οπότε

$$\sum_{i=1}^N |a_i - b_i|^2 = 1 + 1 - 1 - 1 = 0$$

και ως εκ τούτου:

$$a_i = b_i$$

οπότε

$$|\psi_1\rangle = |\psi_2\rangle$$

Δηλαδή: εάν έχουμε $\langle \psi_1 | \psi_2 \rangle = 0$ τότε $|\psi_1\rangle = |\psi_2\rangle$ κάτι όμως που δεν μπορεί να συμβαίνει στην πράξη επειδή υποθέσαμε ότι οι καταστάσεις $|\psi_1\rangle$ και $|\psi_2\rangle$ είναι διαφορετικές.

Μένει λοιπόν μόνο η δυνατότητα

$$\langle \psi_1 | \psi_2 \rangle = 0,$$

δηλαδή το $\hat{U}$ μπορεί να αντιγράψει σωστά δύο διαφορετικές καταστάσεις $|\psi_1\rangle, |\psi_2\rangle$ μόνο αν αυτές είναι κάθετες μεταξύ τους. Αυτό όμως είναι αδύνατο να ισχύει για κάθε ζευγάρι καταστάσεων, όπως απαιτεί η υπόθεσή μας! Θέλουμε να αντιγράψουμε μία αυθαίρετη υπέρθεση $|\psi\rangle$. Π.χ. θέλουμε να αντιγράψουμε και το $|0\rangle$ και το

$$|+x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

που είναι διαφορετικά μεταξύ του αλλά όχι κάθετα αφού $\langle 0 | +x \rangle = 1/\sqrt{2} \neq 0$.

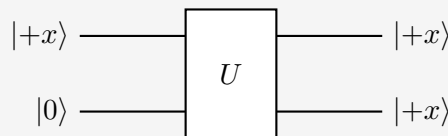
Καταλήγουμε έτσι σε άτοπο: δεν μπορεί να υπάρχει μοναδιαίος τελεστής $\hat{U}$ που να αντιγράφει κάθε δυνατή κατάσταση qubit.

Παράδειγμα 4.9: Η CNOT δεν αντιγράφει σωστά την $|+x\rangle$

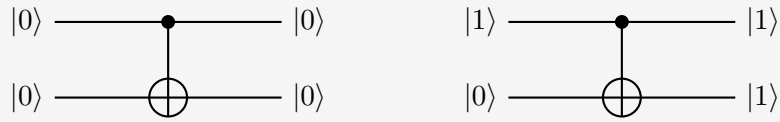
Ας υποθέσουμε ότι θέλουμε να αντιγράψουμε την υπέρθεση

$$|+x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$

δηλαδή να κατασκευάσουμε ένα κύκλωμα U σαν αυτό του παρακάτω σχήματος, που να παίρνει στην είσοδο την κατάσταση $|+x\rangle |0\rangle$ και να παράγει στην έξοδο την $|+x\rangle |+x\rangle$:



Μία πρώτη, φυσική επιλογή για τον U είναι η CNOT, που, όπως βλέπουμε παρακάτω, αντιγράφει σωστά τις $|0\rangle, |1\rangle$:



Ας δούμε τώρα ρητά πώς αποτυγχάνει όταν την εφαρμόσουμε στην $|+x\rangle$. Ξεκινάμε από την

$$|\psi\rangle = |+x\rangle |0\rangle$$

όπου αν αντικαστήσουμε το $|+x\rangle$ θα έχουμε:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|01\rangle$$

Αν εφαρμόσουμε την CNOT σε κάθε όρο ξεχωριστά θα έχουμε:

$$\hat{C}_1(\hat{X})|00\rangle = |00\rangle$$

$$\hat{C}_1(\hat{X})|01\rangle = |11\rangle$$

εφόσον στην πρώτη περίπτωση το qubit ελέγχου είναι $q_2 = 0$ οπότε το πρώτο qubit δεν αλλάζει ενώ στην δεύτερη περίπτωση είναι $q_2 = 1$ οπότε το πρώτο qubit αντιστρέφεται. Όταν η CNOT εφαρμόζεται στην υπέρθεση:

$$\hat{C}_1(\hat{X})|+x\rangle|0\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

Τι θα περιμέναμε όμως να πάρουμε εάν όντως η CNOT κλωνοποιούσε την κατάσταση του πρώτο qubit στο δεύτερο; θα έπρεπε στην έξοδο να είχαμε:

$$|+x\rangle|+x\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

που δεν ταυτίζεται με την πραγματική έξοδο της CNOT που υπολογίσαμε παραπάνω. Επομένως, η CNOT αντιγράφει σωστά τις κάθετες καταστάσεις $|0\rangle$, $|1\rangle$, ακριβώς όπως προβλέπει το θεώρημα, αλλά αποτυγχάνει πλήρως στην $|+x\rangle$: αντί για δύο ανεξάρτητα αντίγραφα $|+x\rangle|+x\rangle$, παράγει μία κατάσταση που δεν μπορεί καν να γραφτεί ως γινόμενο δύο μονο-qubit καταστάσεων.

Ας επιβεβαιώσουμε αριθμητικά, με τη βοήθεια της `qulib2.py`, ότι η CNOT αποτυγχάνει να αντιγράψει την υπέρθεση $|+x\rangle$, παράγοντας αντ' αυτού μία κατάσταση Bell αντί για το επιθυμητό γινόμενο $|+x\rangle|+x\rangle$. Το script βρίσκεται στο αρχείο `python/ex_CNOT_no_clone.py`:

```
import numpy as np

from qulib2 import QuantumState, CNOT

np.set_printoptions(precision=4, suppress=True)

# Example 4.9: the CNOT fails to clone |+x> = (|0> + |1>) / sqrt(2)
plus_x = QuantumState({"0": 1, "1": 1})
zero = QuantumState({"0": 1})

cnot = CNOT()
out = cnot.out(plus_x ** zero)
print(f"C_1^1(X)(|+x>|0>) = {out.to_dict()}")

bell = QuantumState({"00": 1, "11": 1})
print(f"matches (|00> + |11>) / sqrt(2): {np.allclose(out.a, bell.a)}")

desired_clone = plus_x ** plus_x
print(f"desired clone |+x>|+x> = {desired_clone.to_dict()}")
```

```
print(f"NOT equal to the desired clone: {not np.allclose(out.a, desired_clone.a)}")
```

Αποτέλεσμα 4.2: Θεώρημα μη κλωνοποίησης

Δεν υπάρχει μοναδιαίος τελεστής $\hat{U}$ πάνω σε δύο qubit που να ικανοποιεί

$$\hat{U}(|\psi\rangle|s\rangle) = |\psi\rangle|\psi\rangle$$

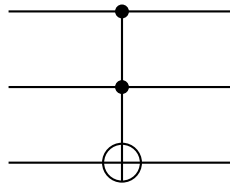
για κάθε δυνατή κατάσταση $|\psi\rangle$ ενός qubit, όπου $|s\rangle$ μία σταθερή αρχική κατάσταση. Καμία φυσική διεργασία δεν μπορεί επομένως να αντιγράψει μία αυθαίρετη άγνωστη κβαντική κατάσταση.

4.4 Πύλες τριών qubit

Στην ενότητα 4.1 είδαμε πύλες που έχουν ένα qubit ελέγχου ($n = 1$) και ένα qubit στόχου ($k = 1$). Θα θεωρήσουμε τώρα πύλες με 2 qubit ελέγχου και ένα qubit στόχου. Ας εξετάσουμε αρχικά το παράδειγμα της πύλης *Toffoli* όπου υπάρχουν δύο qubit ελέγχου q_1 και q_2 και το τρίτο qubit στόχου q_3 αντιστρέφεται μόνο εάν $q_1 = q_2 = 1$. Η πύλη συμβολίζεται με $C_1^2(X)$ εφόσον $n = 2$ και $k = 1$ και η δράση της περιγράφεται ως εξής:

$$\hat{C}_1^2(\hat{X})|q_1q_2q_3\rangle = \begin{cases} |q_1q_2q_3\rangle, & q_1q_2 \neq 1 \\ |q_1q_2\bar{q}_3\rangle, & q_1q_2 = 1 \end{cases}$$

και απεικονίζεται με δύο γεμάτα κυκλάκια στα σύρματα ελέγχου, συνδεδεμένα με ένα $\oplus$ στο σύρμα-στόχο:



Η Toffoli λέγεται και *CCNOT* (*Controlled-Controlled-NOT*). Ας βρούμε τώρα τον πίνακα της στο παρακάτω παράδειγμα.

Παράδειγμα 4.10: Ο πίνακας της πύλης Toffoli

Επειδή δρα σε τρία qubit, ο πίνακας της θα είναι 8×8 . Ορίζουμε τις κυματοσυναρτήσεις βάσης:

$$|\psi_1\rangle = |000\rangle$$

$$|\psi_2\rangle = |001\rangle$$

$$|\psi_3\rangle = |010\rangle$$

$$|\psi_4\rangle = |011\rangle$$

$$|\psi_5\rangle = |100\rangle$$

$$|\psi_6\rangle = |101\rangle$$

$$|\psi_7\rangle = |110\rangle$$

$$|\psi_8\rangle = |111\rangle$$

Εφαρμόζουμε τον τελεστή της πύλης σε καθεμία από τις οκτώ βασικές καταστάσεις:

$$\hat{C}_2^1(\hat{X})|\psi_1\rangle = |\psi_1\rangle$$

$$\hat{C}_2^1(\hat{X})|\psi_2\rangle = |\psi_2\rangle$$

$$\hat{C}_2^1(\hat{X})|\psi_3\rangle = |\psi_3\rangle$$

$$\hat{C}_2^1(\hat{X})|\psi_4\rangle = |\psi_4\rangle$$

$$\hat{C}_2^1(\hat{X})|\psi_5\rangle = |\psi_5\rangle$$

$$\hat{C}_2^1(\hat{X})|\psi_6\rangle = |\psi_6\rangle$$

$$\hat{C}_2^1(\hat{X})|\psi_7\rangle = |\psi_8\rangle$$

$$\hat{C}_2^1(\hat{X})|\psi_8\rangle = |\psi_7\rangle$$

Στις $|\psi_1\rangle, \dots, |\psi_6\rangle$ τουλάχιστον ένα από τα δύο qubit ελέγχου είναι $|0\rangle$, άρα δεν αλλάζει τίποτα. Στις $|\psi_7\rangle$ και $|\psi_8\rangle$ και τα δύο qubit ελέγχου είναι $|1\rangle$, άρα αντιστρέφεται το τρίτο qubit. Τα στοιχεία c_{pq} του πίνακα δίνονται και πάλι από τη σχέση:

$$c_{pq} = \langle \psi_p | \hat{C}_2^1(\hat{X}) | \psi_q \rangle$$

Οι μόνοι μη μηδενικοί συνδυασμοί είναι οι $p = q$ για $p = 1, \dots, 6$, καθώς και οι $p = 7, q = 8$ και $p = 8, q = 7$. Σε όλους αυτούς το στοιχείο του πίνακα είναι 1, για παράδειγμα:

$$c_{78} = \langle \psi_7 | \hat{C}_2^1(\hat{X}) | \psi_8 \rangle = \langle \psi_7 | \psi_7 \rangle = 1$$

Επομένως:

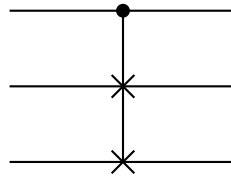
$$C_2^1(\mathbf{X}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Παρατηρήστε ότι ο πίνακας είναι ο ταυτοτικός, εκτός από το κάτω-δεξιά τεταρτημόριο 2×2 , που είναι ακριβώς ο πίνακας $\mathbf{X}$. Αυτό είναι ανάλογο με ό,τι είδαμε για τις ελεγχόμενες πύλες με ένα qubit ελέγχου (αποτέλεσμα 4.1).

Μία δεύτερη σημαντική πύλη τριών qubit είναι η πύλη *Fredkin* (ή *CSWAP*, *Controlled-SWAP*), η ελεγχόμενη εκδοχή της πύλης *SWAP* που απλώς ανταλλάσσει τις καταστάσεις δύο qubit, $\text{SWAP} |ab\rangle = |ba\rangle$. Θα τη συμβολίζουμε εναλλακτικά και ως $\hat{S}_W$ (και τον πίνακά της ως $\mathbf{S}_W$), για συντομία. Η Fredkin είναι η πύλη $\hat{C}_1^2(\hat{S}_W)$: το πρώτο qubit είναι το qubit-ελέγχου, και τα άλλα δύο qubit εναλλάσσουν τις καταστάσεις τους μόνο όταν το qubit-ελέγχου βρίσκεται στην κατάσταση $|1\rangle$,

$$\hat{C}_1^2(\hat{S}_W) |q_1 q_2 q_3\rangle = \begin{cases} |q_1 q_2 q_3\rangle, & q_1 = 0 \\ |q_1 q_3 q_2\rangle, & q_1 = 1, \end{cases}$$

και απεικονίζεται με ένα γεμάτο κυκλάκι στο σύρμα ελέγχου, συνδεδεμένο με δύο σύμβολα $\times$ στα δύο σύρματα-στόχους:



Ας βρούμε τώρα και τον πίνακα της Fredkin.

Παράδειγμα 4.11: Ο πίνακας της πύλης Fredkin

Όπως και η Toffoli, η Fredkin δρα σε τρία qubit, οπότε ο πίνακάς της θα είναι 8×8 . Χρησιμοποιούμε τις ίδιες κυματοσυναρτήσεις βάσης με το παράδειγμα 4.10:

$$|\psi_1\rangle = |000\rangle$$

$$|\psi_2\rangle = |001\rangle$$

$$|\psi_3\rangle = |010\rangle$$

$$|\psi_4\rangle = |011\rangle$$

$$|\psi_5\rangle = |100\rangle$$

$$|\psi_6\rangle = |101\rangle$$

$$|\psi_7\rangle = |110\rangle$$

$$|\psi_8\rangle = |111\rangle$$

Εφαρμόζουμε τον τελεστή της πύλης σε καθμία από τις οκτώ βασικές καταστάσεις:

$$\hat{C}_1^2(\hat{S}_W) |\psi_1\rangle = |\psi_1\rangle$$

$$\hat{C}_1^2(\hat{S}_W) |\psi_2\rangle = |\psi_2\rangle$$

$$\hat{C}_1^2(\hat{S}_W) |\psi_3\rangle = |\psi_3\rangle$$

$$\hat{C}_1^2(\hat{S}_W) |\psi_4\rangle = |\psi_4\rangle$$

$$\hat{C}_1^2(\hat{S}_W) |\psi_5\rangle = |\psi_5\rangle$$

$$\hat{C}_1^2(\hat{S}_W) |\psi_6\rangle = |\psi_7\rangle$$

$$\hat{C}_1^2(\hat{S}_W) |\psi_7\rangle = |\psi_6\rangle$$

$$\hat{C}_1^2(\hat{S}_W) |\psi_8\rangle = |\psi_8\rangle$$

Στις $|\psi_1\rangle, \dots, |\psi_4\rangle$ το qubit ελέγχου είναι $|0\rangle$, άρα δεν αλλάζει τίποτα. Στις $|\psi_5\rangle, \dots, |\psi_8\rangle$ το qubit ελέγχου είναι $|1\rangle$, άρα το δεύτερο και το τρίτο qubit ανταλλάσσονται. Στις $|\psi_5\rangle = |100\rangle$ και $|\psi_8\rangle = |111\rangle$ όμως το δεύτερο και το τρίτο qubit είναι ίδια, οπότε η ανταλλαγή τους δεν αλλάζει τίποτα. Μόνο στις $|\psi_6\rangle = |101\rangle$ και $|\psi_7\rangle = |110\rangle$, όπου τα δύο qubit διαφέρουν, η ανταλλαγή τις μετατρέπει τη μία στην άλλη. Τα στοιχεία c_{pq} του πίνακα δίνονται και πάλι από τη σχέση:

$$c_{pq} = \langle \psi_p | \hat{C}_1^2(\hat{S}_W) | \psi_q \rangle$$

Οι μόνοι μη μηδενικοί συνδυασμοί είναι οι $p = q$ για $p = 1, \dots, 5$ και $p = 8$, καθώς και οι $p = 6, q = 7$ και $p = 7, q = 6$. Σε όλους αυτούς το στοιχείο του πίνακα είναι 1, για παράδειγμα:

$$c_{67} = \langle \psi_6 | \hat{C}_1^2(\hat{S}_W) | \psi_7 \rangle = \langle \psi_6 | \psi_6 \rangle = 1$$

Επομένως:

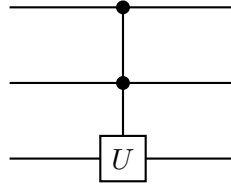
$$C_1^2(S_W) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Παρατηρήστε ότι το πάνω-αριστερά τεταρτημόριο 4×4 είναι ο ταυτοτικός πίνακας I_4 , αφού για $q_1 = 0$ δεν συμβαίνει τίποτα. Το κάτω-δεξιά τεταρτημόριο 4×4 είναι ο πίνακας της SWAP:

$$S_W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

δηλαδή η πύλη που εφαρμόζεται στα δύο qubit-στόχους όταν $q_1 = 1$. Πρόκειται για την ίδια δομή που είδαμε στο αποτέλεσμα 4.1, τώρα όμως με πίνακες 4×4 στη θέση των 2×2 .

Η πύλη $C_2^1(U)$ έχει δύο qubit ελέγχου q_1 και q_2 , και εφαρμόζει την πύλη U στο τρίτο qubit q_3 μόνο εάν $q_1 = q_2 = 1$. Απεικονίζεται με δύο γεμάτα κυκλάκια στα σύρματα ελέγχου, συνδεδεμένα με ένα κουτί U στο σύρμα-στόχο:



και η δράση της στις βασικές καταστάσεις περιγράφεται από την εξίσωση:

$$\hat{C}_2^1(\hat{U}) |q_1 q_2 q_3\rangle = \begin{cases} |q_1 q_2 q_3\rangle, & q_1 q_2 \neq 1 \\ |q_1 q_2\rangle \otimes \hat{U} |q_3\rangle, & q_1 q_2 = 1 \end{cases}$$

Η κατασκευή αυτή γενικεύει την Toffoli, που είναι απλώς η ειδική περίπτωση $U = X$.

Ας δείξουμε τώρα πώς κατασκευάζεται μία γενική ελεγχόμενη πύλη με δύο qubit-ελέγχου, $C_2^1(U)$, για οποιαδήποτε πύλη ενός qubit U , χρησιμοποιώντας μόνο CNOT (με ένα qubit ελέγχου) και κατάλληλες ελεγχόμενες πύλες ενός qubit-στόχου. Το πρώτο βήμα είναι να βρούμε έναν πίνακα V με $V^2 = U$, μία «τετραγωνική ρίζα» της U . Αυτό είναι γενικά πιο εύκολο από ό,τι φαίνεται. Ένας τρόπος είναι να γράψουμε το U ως περιστροφή όπως είδαμε στο 3.9

$$U = e^{j\alpha} R_n(\theta)$$

για κατάλληλα α, θ, n . Στη συνέχεια αρκεί να επιλέξουμε

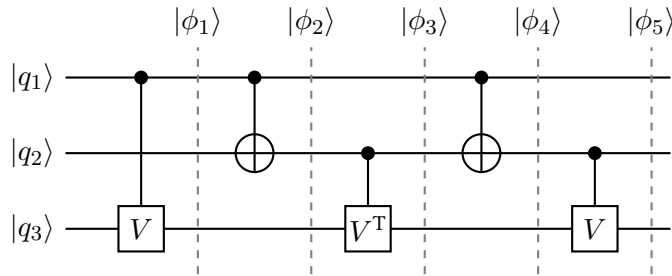
$$V = e^{j\alpha/2} R_n\left(\frac{\theta}{2}\right),$$

αφού τότε, χρησιμοποιώντας τη σύνθεση περιστροφών γύρω από τον ίδιο άξονα

$$R_n(\theta_1) R_n(\theta_2) = R_n(\theta_1 + \theta_2)$$

που είδαμε στην ενότητα 3.12,

$$V^2 = VV = e^{j\alpha/2} R_n\left(\frac{\theta}{2}\right) e^{j\alpha/2} R_n\left(\frac{\theta}{2}\right) = e^{j\alpha} R_n(\theta) = U.$$



Σχήμα 4.3: Υλοποίηση μίας ελεγχόμενης πύλης $C_2(U)$.

Ας θεωρήσουμε τώρα το κύκλωμα του σχήματος 4.3 όπου εφαρμόζουμε τώρα, με σειρά, στο τρίτο qubit (qubit-στόχο) αρχικά την πύλη $C_1^1(V)$ ελεγχόμενη από το πρώτο qubit, μία CNOT από το πρώτο στο δεύτερο qubit, $C_1^1(V^T)$ ελεγχόμενη από το (μεταβαλλόμενο πλέον) δεύτερο qubit, ξανά την ίδια CNOT, και τέλος $C_1^1(V)$ ελεγχόμενη από το δεύτερο qubit. Ας παρακολουθήσουμε πώς αλλάζει η κατάσταση των τριών qubit καθώς περνάει από τις πύλες του κυκλώματος. Θεωρούμε ότι στην είσοδο, αριστερά, έχουμε μία βασική κατάσταση

$$|q_1\rangle |q_2\rangle |q_3\rangle$$

όπου τα q_1, q_2, q_3 είναι bit, δηλαδή 0 ή 1. Με $|\phi_1\rangle, \dots, |\phi_5\rangle$ συμβολίζουμε την κατάσταση και των τριών qubit στα σημεία που σημειώνονται στο σχήμα 4.3.

Για να γράφουμε σύντομα τη δράση μίας ελεγχόμενης πύλης, θα χρησιμοποιούμε τον συμβολισμό V^q , όπου q είναι ένα bit. Όταν $q = 1$, ο $V^q = V^1$ είναι απλώς ο V , δηλαδή εφαρμόζουμε την πύλη. Όταν $q = 0$, ο $V^q = V^0$ είναι ο ταυτοτικός πίνακας I_2 , δηλαδή δεν κάνουμε τίποτα. Έτσι, μία ελεγχόμενη πύλη $C_1^1(V)$ με qubit ελέγχου στην κατάσταση $|q\rangle$ εφαρμόζει στο qubit-στόχο τον V^q . Αυτό ισχύει ακόμα κι αν το qubit-στόχος δεν βρίσκεται σε βασική κατάσταση, αφού ο πίνακας της ελεγχόμενης πύλης είναι I_2 ή V ανάλογα με το qubit ελέγχου (αποτέλεσμα 4.1). Με τον ίδιο τρόπο, μία CNOT με qubit ελέγχου $|q_1\rangle$ και qubit-στόχο $|q_2\rangle$ μετατρέπει το qubit-στόχο σε $|q_1 \oplus q_2\rangle$, όπου $\oplus$ η πρόσθεση modulo 2 (XOR).

Η πρώτη πύλη, η $C_1^1(V)$, ελέγχεται από το πρώτο qubit, που βρίσκεται στην κατάσταση $|q_1\rangle$. Επομένως εφαρμόζει τον V^{q_1} στο τρίτο qubit, ενώ τα δύο πρώτα qubit δεν αλλάζουν:

$$|\phi_1\rangle = |q_1\rangle |q_2\rangle \otimes V^{q_1} |q_3\rangle$$

Η δεύτερη πύλη είναι μία CNOT που έχει qubit ελέγχου το πρώτο και qubit-στόχο το δεύτερο. Το δεύτερο qubit γίνεται επομένως $|q_1 \oplus q_2\rangle$, ενώ τα άλλα δύο δεν αλλάζουν:

$$|\phi_2\rangle = |q_1\rangle |q_1 \oplus q_2\rangle \otimes V^{q_1} |q_3\rangle$$

Η τρίτη πύλη, η $C_1^1(V^T)$, ελέγχεται από το δεύτερο qubit, που βρίσκεται τώρα στην κατάσταση $|q_1 \oplus q_2\rangle$. Επομένως εφαρμόζει τον $(V^T)^{q_1 \oplus q_2}$ στο τρίτο qubit:

$$|\phi_3\rangle = |q_1\rangle |q_1 \oplus q_2\rangle \otimes (V^T)^{q_1 \oplus q_2} V^{q_1} |q_3\rangle$$

Η τέταρτη πύλη είναι μία δεύτερη CNOT, ίδια με την πρώτη, που μετατρέπει το δεύτερο qubit σε:

$$|q_1 \oplus (q_1 \oplus q_2)\rangle$$

Αφού $q_1 \oplus q_1 = 0$, αυτό είναι ξανά το αρχικό $|q_2\rangle$:

$$|\phi_4\rangle = |q_1\rangle |q_2\rangle \otimes (V^T)^{q_1 \oplus q_2} V^{q_1} |q_3\rangle$$

Η πέμπτη και τελευταία πύλη, η $C_1^1(V)$, ελέγχεται από το δεύτερο qubit, που έχει επανέλθει στην κατάσταση $|q_2\rangle$, και εφαρμόζει τον V^{q_2} στο τρίτο qubit:

$$|\phi_5\rangle = |q_1\rangle |q_2\rangle \otimes V^{q_2} (V^T)^{q_1 \oplus q_2} V^{q_1} |q_3\rangle$$

Βλέπουμε ότι τα δύο πρώτα qubit καταλήγουν στην αρχική τους κατάσταση $|q_1\rangle |q_2\rangle$. Μένει να δούμε τι πίνακας εφαρμόζεται τελικά στο τρίτο qubit. Επειδή ο V είναι μοναδιαίος, ο V^T είναι ο αντίστροφός του, $V^T = V^{-1}$. Όλοι οι πίνακες στο γινόμενο είναι επομένως δυνάμεις του ίδιου πίνακα V , οπότε οι εκθέτες τους απλώς προστίθενται:

$$V^{q_2} (V^T)^{q_1 \oplus q_2} V^{q_1} = V^{q_2} V^{-(q_1 \oplus q_2)} V^{q_1} = V^m$$

όπου

$$m = q_1 + q_2 - (q_1 \oplus q_2)$$

Ας υπολογίσουμε το m για όλους τους δυνατούς συνδυασμούς των q_1, q_2 :

q_1	q_2	$q_1 \oplus q_2$	m
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	2

Δηλαδή $m = 0$ σε όλες τις περιπτώσεις, εκτός από την περίπτωση $q_1 = q_2 = 1$, όπου $m = 2$. Όταν $m = 0$, στο τρίτο qubit εφαρμόζεται ο $V^0 = I_2$, δηλαδή δεν αλλάζει τίποτα. Όταν $m = 2$, εφαρμόζεται ο $V^2 = U$. Επομένως:

$$|\phi_5\rangle = \begin{cases} |q_1 q_2 q_3\rangle, & q_1 q_2 \neq 1 \\ |q_1 q_2\rangle \otimes \hat{U} |q_3\rangle, & q_1 q_2 = 1 \end{cases}$$

που είναι ακριβώς η δράση της $C_2^1(U)$: η U εφαρμόζεται στο qubit-στόχο μόνο όταν και τα δύο qubit ελέγχου είναι $|1\rangle$.

Παράδειγμα 4.12: Η Toffoli από ελεγχόμενες πύλες ενός qubit ελέγχου

Ας εφαρμόσουμε την παραπάνω κατασκευή στην πύλη Toffoli (CCNOT), δηλαδή στην περίπτωση $U = X$. Χρειαζόμαστε έναν πίνακα V με $V^2 = X$.

Πρώτα γράφουμε την X στη μορφή $e^{j\alpha} R_x(\theta)$. Από τον ορισμό του πίνακα $R_x(\theta)$ (ενότητα 3.6),

$$R_x(\theta) = \cos\left(\frac{\theta}{2}\right) I_2 - j \sin\left(\frac{\theta}{2}\right) X$$

Για $\theta = \pi$ έχουμε $\cos(\pi/2) = 0$ και $\sin(\pi/2) = 1$, οπότε:

$$R_x(\pi) = -jX$$

Πολλαπλασιάζοντας με j , και αφού $j \cdot (-j) = 1$ και $j = e^{j\pi/2}$, βρίσκουμε:

$$X = e^{j\pi/2} R_x(\pi)$$

Δηλαδή $\alpha = \pi/2$, ο άξονας είναι ο x και $\theta = \pi$. Σύμφωνα με την κατασκευή μας, επιλέγουμε:

$$V = e^{j\pi/4} R_x\left(\frac{\pi}{2}\right)$$

Για $\theta = \pi/2$ έχουμε $\cos(\pi/4) = \sin(\pi/4) = 1/\sqrt{2}$, οπότε:

$$R_x\left(\frac{\pi}{2}\right) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix}$$

Επίσης $e^{j\pi/4} = (1 + j)/\sqrt{2}$. Πολλαπλασιάζοντας, και χρησιμοποιώντας ότι $(1 + j)(-j) = 1 - j$, παίρνουμε:

$$V = \frac{1}{2} \begin{bmatrix} 1 + j & 1 - j \\ 1 - j & 1 + j \end{bmatrix}$$

Ας επιβεβαιώσουμε ότι $V^2 = X$. Χρειαζόμαστε τα γινόμενα:

$$(1 + j)^2 = 2j$$

$$(1 - j)^2 = -2j$$

$$(1 + j)(1 - j) = 2$$

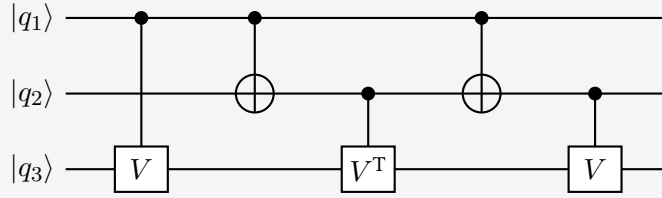
Επομένως:

$$V^2 = \frac{1}{4} \begin{bmatrix} (1 + j)^2 + (1 - j)^2 & 2(1 + j)(1 - j) \\ 2(1 + j)(1 - j) & (1 - j)^2 + (1 + j)^2 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 0 & 4 \\ 4 & 0 \end{bmatrix} = X$$

Γ' αυτό ο V αναφέρεται συχνά και ως «τετραγωνική ρίζα της NOT». Χρειαζόμαστε και τον V^T , που προκύπτει παίρνοντας τον ανάστροφο και τους συζυγείς μιγαδικούς των στοιχείων του V :

$$V^T = \frac{1}{2} \begin{bmatrix} 1 - j & 1 + j \\ 1 + j & 1 - j \end{bmatrix}$$

Η Toffoli υλοποιείται επομένως με το κύκλωμα:



Ας δούμε πώς δουλεύει το κύκλωμα για δύο συγκεκριμένες εισόδους. Θα χρειαστούμε τη δράση του V στην $|0\rangle$, που είναι η πρώτη στήλη του V :

$$V|0\rangle = \frac{1}{2} \left((1+j)|0\rangle + (1-j)|1\rangle \right)$$

Ας ξεκινήσουμε με την είσοδο $|110\rangle$. Εδώ $q_1 = q_2 = 1$ και $q_3 = 0$, οπότε η Toffoli πρέπει να αντιστρέψει το τρίτο qubit. Ακολουθώντας τα βήματα της ενότητας:

$$|\phi_1\rangle = |1\rangle|1\rangle \otimes V|0\rangle$$

$$|\phi_2\rangle = |1\rangle|0\rangle \otimes V|0\rangle$$

$$|\phi_3\rangle = |1\rangle|0\rangle \otimes V|0\rangle$$

$$|\phi_4\rangle = |1\rangle|1\rangle \otimes V|0\rangle$$

$$|\phi_5\rangle = |1\rangle|1\rangle \otimes VV|0\rangle = |1\rangle|1\rangle \otimes X|0\rangle = |111\rangle$$

Η πρώτη CNOT έκανε το δεύτερο qubit $|0\rangle$, οπότε η V^T δεν ενεργοποιήθηκε ($|\phi_3\rangle = |\phi_2\rangle$). Στο τρίτο qubit εφαρμόστηκε επομένως δύο φορές ο V , δηλαδή ο $V^2 = X$.

Ας δούμε τώρα την είσοδο $|100\rangle$. Εδώ $q_1 = 1$ αλλά $q_2 = 0$, οπότε η Toffoli δεν πρέπει να αλλάξει τίποτα:

$$|\phi_1\rangle = |1\rangle|0\rangle \otimes V|0\rangle$$

$$|\phi_2\rangle = |1\rangle|1\rangle \otimes V|0\rangle$$

$$|\phi_3\rangle = |1\rangle|1\rangle \otimes V^T V|0\rangle = |1\rangle|1\rangle \otimes |0\rangle$$

$$|\phi_4\rangle = |1\rangle|0\rangle \otimes |0\rangle$$

$$|\phi_5\rangle = |100\rangle$$

Εδώ η πρώτη CNOT έκανε το δεύτερο qubit $|1\rangle$, οπότε η V^T ενεργοποιήθηκε και ακύρωσε τη δράση της πρώτης V , αφού $V^T V = I_2$. Η τελευταία V δεν ενεργοποιήθηκε, αφού το δεύτερο qubit επανήλθε στο $|0\rangle$. Το τρίτο qubit έμεινε λοιπόν $|0\rangle$, όπως έπρεπε.

Παράδειγμα 4.13: Η ελεγχόμενη πύλη Hadamard με δύο qubit ελέγχου

Ας εφαρμόσουμε τώρα την ίδια κατασκευή για την $C_2^1(H)$, δηλαδή στην περίπτωση $U = H$. Χρειαζόμαστε και πάλι έναν πίνακα V με $V^2 = H$.

Θυμηθείτε ότι ο πίνακας περιστροφής γύρω από έναν άξονα $\mathbf{n}$ (ενότητα 3.8) είναι:

$$R_{\mathbf{n}}(\theta) = \cos\left(\frac{\theta}{2}\right) I_2 - j \sin\left(\frac{\theta}{2}\right) A$$

όπου $A = n_x X + n_y Y + n_z Z$. Στο παράδειγμα 3.9 είδαμε ότι για τον άξονα

$$\mathbf{n} = \frac{1}{\sqrt{2}}(1, 0, 1)$$

ο πίνακας A είναι ακριβώς ο H . Για $\theta = \pi$ έχουμε επομένως:

$$R_{\mathbf{n}}(\pi) = -jH$$

και, όπως και στο προηγούμενο παράδειγμα, πολλαπλασιάζοντας με $j = e^{j\pi/2}$:

$$\mathbf{H} = e^{j\pi/2} \mathbf{R}_n(\pi)$$

Δηλαδή $\alpha = \pi/2$ και $\theta = \pi$. Επιλέγουμε λοιπόν:

$$\mathbf{V} = e^{j\pi/4} \mathbf{R}_n\left(\frac{\pi}{2}\right)$$

Για $\theta = \pi/2$ έχουμε $\cos(\pi/4) = \sin(\pi/4) = 1/\sqrt{2}$, οπότε:

$$\mathbf{R}_n\left(\frac{\pi}{2}\right) = \frac{1}{\sqrt{2}} (\mathbf{I}_2 - j\mathbf{H})$$

και, αφού $e^{j\pi/4} = (1+j)/\sqrt{2}$:

$$\mathbf{V} = \frac{1+j}{2} (\mathbf{I}_2 - j\mathbf{H})$$

Ας επιβεβαιώσουμε ότι $\mathbf{V}^2 = \mathbf{H}$. Εδώ δεν χρειάζεται να γράψουμε τα στοιχεία του πίνακα: αρκεί να θυμηθούμε ότι $\mathbf{H}^2 = \mathbf{I}_2$. Υπολογίζουμε πρώτα:

$$\left(\frac{1+j}{2}\right)^2 = \frac{2j}{4} = \frac{j}{2}$$

και

$$(\mathbf{I}_2 - j\mathbf{H})^2 = \mathbf{I}_2 - 2j\mathbf{H} + j^2\mathbf{H}^2 = \mathbf{I}_2 - 2j\mathbf{H} - \mathbf{I}_2 = -2j\mathbf{H}$$

Επομένως:

$$\mathbf{V}^2 = \frac{j}{2} \cdot (-2j\mathbf{H}) = -j^2\mathbf{H} = \mathbf{H}$$

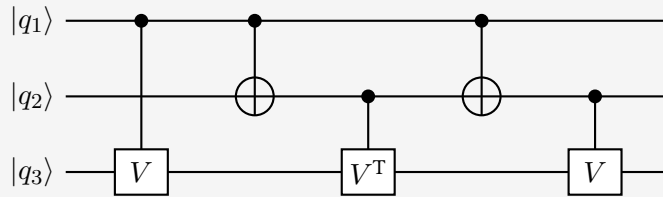
Για τον $\mathbf{V}^T$ παίρνουμε τον συζυγή του αριθμού $(1+j)/2$, και τον $\mathbf{H}^T = \mathbf{H}$, αφού ο $\mathbf{H}$ είναι πραγματικός και συμμετρικός:

$$\mathbf{V}^T = \frac{1-j}{2} (\mathbf{I}_2 + j\mathbf{H})$$

Με τον ίδιο τρόπο μπορούμε να ελέγξουμε ότι $\mathbf{V}^T \mathbf{V} = \mathbf{I}_2$:

$$\mathbf{V}^T \mathbf{V} = \frac{(1-j)(1+j)}{4} (\mathbf{I}_2 + j\mathbf{H}) (\mathbf{I}_2 - j\mathbf{H}) = \frac{2}{4} (\mathbf{I}_2 + \mathbf{H}^2) = \mathbf{I}_2$$

Η $C_2^1(\mathbf{H})$ υλοποιείται επομένως με το ίδιο κύκλωμα όπως η Toffoli, αλλά με τον νέο πίνακα $\mathbf{V}$:



Για την είσοδο $|110\rangle$, ακριβώς όπως στο προηγούμενο παράδειγμα, η V^T δεν ενεργοποιείται, και στο τρίτο qubit εφαρμόζεται δύο φορές ο $\mathbf{V}$:

$$|\phi_5\rangle = |1\rangle |1\rangle \otimes \mathbf{V}^2 |0\rangle = |1\rangle |1\rangle \otimes \mathbf{H} |0\rangle = |1\rangle |1\rangle |+\rangle$$

Για τις υπόλοιπες εισόδους, όπου τουλάχιστον ένα από τα q_1, q_2 είναι 0, οι πύλες $\mathbf{V}$ και $\mathbf{V}^T$ είτε δεν ενεργοποιούνται είτε αλληλοαναιρούνται, όπως στην είσοδο $|100\rangle$ του προηγούμενου παραδείγματος, και το τρίτο qubit μένει αμετάβλητο.

Σημειώστε ότι ο $\mathbf{V}$ δεν είναι ο μόνος πίνακας με $\mathbf{V}^2 = \mathbf{H}$. Για παράδειγμα, η $\mathbf{H}^{1/2}$ της ενότητας 3.14 είναι επίσης μία τετραγωνική ρίζα της $\mathbf{H}$, διαφορετική από τη δική μας. Η κατασκευή δουλεύει με οποιαδήποτε από αυτές, αφού η απόδειξη χρησιμοποιεί μόνο ότι $\mathbf{V}^2 = \mathbf{U}$ και ότι ο $\mathbf{V}$ είναι μοναδιαίος.

Ας δούμε τώρα το αντίστροφο: πώς μπορούμε να χρησιμοποιήσουμε μία πύλη με δύο qubit-ελέγχου, όπως η ίδια η Toffoli, ως δομικό στοιχείο για την κατασκευή της Fredkin, $C_1^2(SW)$. Αυτό το δείχνουμε στο παρακάτω παράδειγμα.

Παράδειγμα 4.14: Η Fredkin από δύο CNOT και μία Toffoli

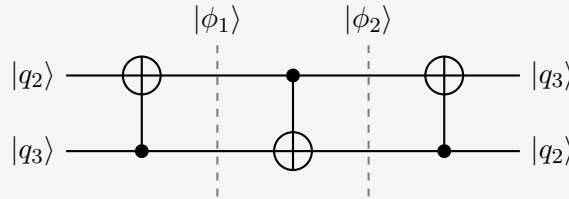
Σε όλο το παράδειγμα θα χρησιμοποιούμε δύο απλές ιδιότητες της πρόσθεσης modulo 2 (XOR) για κάθε bit x :

$$x \oplus 0 = x$$

$$x \oplus x = 0$$

Θα χρειαστούμε επίσης να γράψουμε τη δράση της CNOT και της Toffoli με τη βοήθεια του XOR. Μία CNOT αντιστρέφει το qubit-στόχο όταν το qubit ελέγχου είναι 1. Αφού $x \oplus 1 = \bar{x}$, αυτό σημαίνει ότι το qubit-στόχος $|t\rangle$ γίνεται $|t \oplus c\rangle$, όπου c το bit του qubit ελέγχου. Με τον ίδιο τρόπο, η Toffoli με qubit ελέγχου $|c_1\rangle, |c_2\rangle$ μετατρέπει το qubit-στόχο σε $|t \oplus c_1 c_2\rangle$, αφού το γινόμενο $c_1 c_2$ είναι 1 μόνο όταν και τα δύο qubit ελέγχου είναι 1.

Ας δείξουμε πρώτα ότι η SWAP, που ανταλλάσσει τις καταστάσεις δύο qubit, γράφεται ως τρεις CNOT ανάμεσα στα ίδια δύο qubit, με εναλλασσόμενο ρόλο ελέγχου και στόχου. Θεωρούμε το κύκλωμα:



Η πρώτη και η τρίτη CNOT έχουν qubit ελέγχου το $|q_3\rangle$ (κάτω σύρμα) και qubit-στόχο το $|q_2\rangle$ (πάνω σύρμα), ενώ η μεσαία το αντίστροφο. Ξεκινάμε από την κατάσταση $|q_2\rangle |q_3\rangle$. Η πρώτη CNOT μετατρέπει το πάνω qubit σε $|q_2 \oplus q_3\rangle$:

$$|\phi_1\rangle = |q_2 \oplus q_3\rangle |q_3\rangle$$

Η δεύτερη CNOT έχει qubit ελέγχου το πάνω qubit, που είναι τώρα $|q_2 \oplus q_3\rangle$, οπότε το κάτω qubit γίνεται:

$$|q_3 \oplus (q_2 \oplus q_3)\rangle = |q_2 \oplus (q_3 \oplus q_3)\rangle = |q_2 \oplus 0\rangle = |q_2\rangle$$

Επομένως:

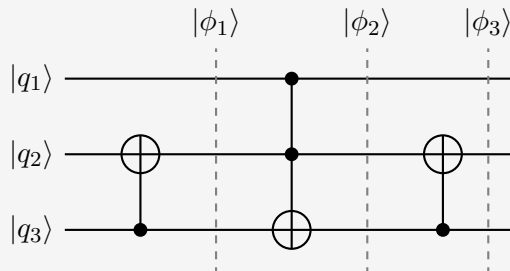
$$|\phi_2\rangle = |q_2 \oplus q_3\rangle |q_2\rangle$$

Η τρίτη CNOT έχει ξανά qubit ελέγχου το κάτω qubit, που είναι τώρα $|q_2\rangle$, οπότε το πάνω qubit γίνεται:

$$|(q_2 \oplus q_3) \oplus q_2\rangle = |q_3\rangle$$

Στην έξοδο έχουμε επομένως $|q_3\rangle |q_2\rangle$, δηλαδή οι καταστάσεις των δύο qubit ανταλλάχθηκαν, ακριβώς όπως κάνει η SWAP.

Ας περάσουμε τώρα στην Fredkin. Η Fredkin πρέπει να κάνει την SWAP στο δεύτερο και τρίτο qubit μόνο όταν το πρώτο qubit είναι $|1\rangle$. Η ιδέα είναι να κάνουμε τη μεσαία από τις τρεις CNOT ελεγχόμενη και από το πρώτο qubit, αντικαθιστώντας τη με μία Toffoli με qubit ελέγχου το πρώτο και το δεύτερο qubit και qubit-στόχο το τρίτο. Τις άλλες δύο CNOT τις αφήνουμε όπως ήταν:



Ας παρακολουθήσουμε τις ενδιάμεσες καταστάσεις για μία βασική κατάσταση εισόδου $|q_1\rangle |q_2\rangle |q_3\rangle$. Το πρώτο qubit δεν αλλάζει ποτέ, αφού είναι μόνο qubit ελέγχου. Η πρώτη CNOT δίνει, όπως και στην SWAP:

$$|\phi_1\rangle = |q_1\rangle |q_2 \oplus q_3\rangle |q_3\rangle$$

Η Toffoli έχει qubit ελέγχου το $|q_1\rangle$ και το $|q_2 \oplus q_3\rangle$, οπότε προσθέτει (modulo 2) το γινόμενο τους στο τρίτο qubit:

$$|\phi_2\rangle = |q_1\rangle |q_2 \oplus q_3\rangle |q_3 \oplus q_1(q_2 \oplus q_3)\rangle$$

Εδώ πρέπει να διακρίνουμε δύο περιπτώσεις.

Αν $q_1 = 0$, το γινόμενο $q_1(q_2 \oplus q_3)$ είναι 0, οπότε η Toffoli δεν κάνει τίποτα:

$$|\phi_2\rangle = |0\rangle |q_2 \oplus q_3\rangle |q_3\rangle$$

Η τελευταία CNOT μετατρέπει το δεύτερο qubit σε $|(q_2 \oplus q_3) \oplus q_3\rangle = |q_2\rangle$:

$$|\phi_3\rangle = |0\rangle |q_2\rangle |q_3\rangle$$

Δηλαδή οι δύο ίδιες CNOT αναίρεσαν η μία την άλλη, και η κατάσταση έμεινε αμετάβλητη.

Αν $q_1 = 1$, το γινόμενο $q_1(q_2 \oplus q_3)$ είναι $q_2 \oplus q_3$, οπότε η Toffoli συμπεριφέρεται ακριβώς όπως η μεσαία CNOT της SWAP, και το τρίτο qubit γίνεται $|q_3 \oplus (q_2 \oplus q_3)\rangle = |q_2\rangle$:

$$|\phi_2\rangle = |1\rangle |q_2 \oplus q_3\rangle |q_2\rangle$$

Η τελευταία CNOT μετατρέπει το δεύτερο qubit σε $|(q_2 \oplus q_3) \oplus q_2\rangle = |q_3\rangle$:

$$|\phi_3\rangle = |1\rangle |q_3\rangle |q_2\rangle$$

Δηλαδή το δεύτερο και το τρίτο qubit ανταλλάχθηκαν.

Συνοψίζοντας:

$$|\phi_3\rangle = \begin{cases} |q_1 q_2 q_3\rangle, & q_1 = 0 \\ |q_1 q_3 q_2\rangle, & q_1 = 1 \end{cases}$$

που είναι ακριβώς ο ορισμός της Fredkin. Ας το επιβεβαιώσουμε και για μία συγκεκριμένη είσοδο, την $|101\rangle$, που η Fredkin πρέπει να μετατρέψει σε $|110\rangle$:

$$|\phi_1\rangle = |1\rangle |0 \oplus 1\rangle |1\rangle = |111\rangle$$

$$|\phi_2\rangle = |1\rangle |1\rangle |1 \oplus 1 \cdot 1\rangle = |110\rangle$$

$$|\phi_3\rangle = |1\rangle |1 \oplus 0\rangle |0\rangle = |110\rangle$$

Το κύκλωμα υλοποιεί λοιπόν την Fredkin, χρησιμοποιώντας μόνο δύο CNOT και μία Toffoli.

Η Toffoli, με τα δύο πρώτα qubit ως qubit-ελέγχου, έχει και μία πολύ φυσική εφαρμογή: μπορεί να χρησιμοποιηθεί για να υλοποιήσει τις κβαντικές εκδοχές των δύο βασικών λογικών πυλών, AND και OR.

Παράδειγμα 4.15: Το κβαντικό AND

Θέλουμε να κατασκευάσουμε ένα κβαντικό κύκλωμα που υπολογίζει το λογικό AND δύο bit a και b . Θα το συμβολίζουμε με ab , όπως στην ψηφιακή σχεδίαση, αφού το AND δύο bit είναι απλώς το γινόμενό τους. Τα δύο bit είναι κωδικοποιημένα σε δύο qubit, στις καταστάσεις $|a\rangle$ και $|b\rangle$.

Μία πρώτη ιδέα θα ήταν ένα κύκλωμα με δύο qubit στην είσοδο και ένα qubit στην έξοδο, στην κατάσταση $|ab\rangle$. Κάτι τέτοιο όμως είναι αδύνατο. Κάθε κβαντικό κύκλωμα περιγράφεται από έναν μοναδιαίο τελεστή, δηλαδή από έναν τετραγωνικό πίνακα, οπότε έχει πάντα τον ίδιο αριθμό qubit στην είσοδο και στην έξοδο.

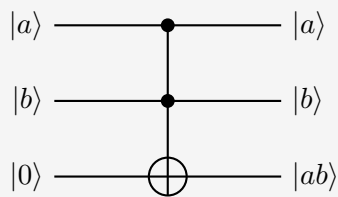
Μία δεύτερη ιδέα θα ήταν να κρατήσουμε το πρώτο qubit και να αντικαταστήσουμε το δεύτερο με το αποτέλεσμα, δηλαδή να μετατρέψουμε την $|a\rangle|b\rangle$ σε $|a\rangle|ab\rangle$. Ούτε αυτό όμως λειτουργεί. Για $a = 0$ έχουμε $ab = 0$ για οποιοδήποτε b , οπότε δύο διαφορετικές εισοδοί θα έδιναν την ίδια έξοδο:

$$|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$$

$$|0\rangle|1\rangle \rightarrow |0\rangle|0\rangle$$

Κοιτάζοντας την έξοδο $|0\rangle|0\rangle$ δεν μπορούμε να ξέρουμε ποια ήταν η είσοδος, δηλαδή η πράξη δεν είναι αντιστρέψιμη. Ένας μοναδιαίος τελεστής όμως είναι πάντα αντιστρέψιμος, οπότε δεν υπάρχει μοναδιαίος τελεστής που να κάνει αυτή τη μετατροπή.

Η λύση είναι να μη χάσουμε κανένα από τα δύο αρχικά qubit, αλλά να προσθέσουμε ένα τρίτο, *βοηθητικό* (*ancilla*) qubit, αρχικοποιημένο πάντα στην κατάσταση $|0\rangle$. Το χρησιμοποιούμε ως qubit-στόχο μίας Toffoli, με qubit ελέγχου τα $|a\rangle$ και $|b\rangle$:



Όπως είδαμε στο παράδειγμα 4.14, η Toffoli μετατρέπει το qubit-στόχο $|t\rangle$ σε $|t \oplus c_1 c_2\rangle$, όπου c_1, c_2 τα bit των qubit ελέγχου. Εδώ $c_1 = a$, $c_2 = b$ και $t = 0$, οπότε:

$$|a\rangle|b\rangle|0\rangle \rightarrow |a\rangle|b\rangle|0 \oplus ab\rangle$$

Αφού $0 \oplus x = x$ για κάθε bit x , η έξοδος είναι:

$$|a\rangle|b\rangle|ab\rangle$$

Ας το δούμε και για όλους τους συνδυασμούς εισόδου:

a	b	είσοδος	έξοδος
0	0	$ 000\rangle$	$ 000\rangle$
0	1	$ 010\rangle$	$ 010\rangle$
1	0	$ 100\rangle$	$ 100\rangle$
1	1	$ 110\rangle$	$ 111\rangle$

Το τρίτο qubit γίνεται $|1\rangle$ μόνο όταν $a = b = 1$, δηλαδή περιέχει ακριβώς το AND των a και b . Τα $|a\rangle$ και $|b\rangle$ παραμένουν αμετάβλητα στην έξοδο, και αυτό είναι που κάνει το κύκλωμα αντιστρέψιμο: από την έξοδο μπορούμε πάντα να ξέρουμε ποια ήταν η είσοδος.

Ας επιβεβαιώσουμε αριθμητικά, με τη βοήθεια της `qulib2.py`, ότι η Toffoli, με ένα βοηθητικό qubit αρχικοποιημένο στην $|0\rangle$, υλοποιεί το κβαντικό AND για όλους τους συνδυασμούς εισόδου. Το script βρίσκεται στο αρχείο `python/ex_quantum_AND.py`:

```
import numpy as np

from qulib2 import QuantumState, Toffoli

np.set_printoptions(precision=4, suppress=True)

# Example 4.15: the Toffoli gate as a quantum AND, with an ancilla qubit
# initialized to |0>
toffoli = Toffoli()
for a in "01":
    for b in "01":
        inp = QuantumState({a + b + "0": 1})
```

```

out = toffoli.out(inp)
result_bits = list(out.to_dict())[0]
expected = a + b + str(int(a) & int(b))
print(f"Toffoli|{a}{b}0> = |{result_bits}> (expect |{expected}>): "
      f"{result_bits == expected}")

```

Παράδειγμα 4.16: Το κβαντικό OR

Μπορούμε να υλοποιήσουμε με τον ίδιο τρόπο και το λογικό OR των a και b , που θα το συμβολίζουμε με $a + b$, όπως στην ψηφιακή σχεδίαση. Προσοχή, το $+$ εδώ δεν είναι η συνηθισμένη πρόσθεση, αφού $1 + 1 = 1$:

a	b	$a + b$
0	0	0
0	1	1
1	0	1
1	1	1

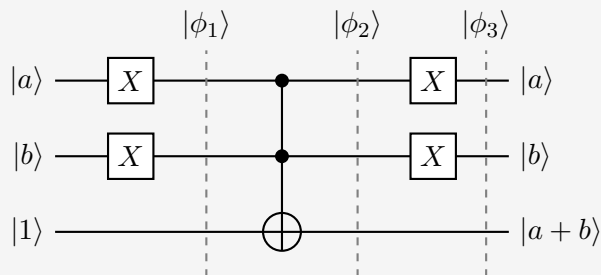
Θα χρησιμοποιήσουμε ξανά μία Toffoli και ένα βοηθητικό qubit, αυτή τη φορά όμως μαζί με μερικές πύλες X (NOT). Η ιδέα βασίζεται σε έναν από τους νόμους του De Morgan:

$$a + b = \overline{\bar{a}\bar{b}}$$

Ο νόμος αυτός λέει ότι το $a + b$ είναι 0 μόνο όταν $a = b = 0$, δηλαδή μόνο όταν $\bar{a} = \bar{b} = 1$. Μπορούμε να τον ελέγξουμε εύκολα με έναν πίνακα αλήθειας:

a	b	$\bar{a}$	$\bar{b}$	$\bar{a}\bar{b}$	$\overline{\bar{a}\bar{b}}$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

Η τελευταία στήλη είναι ίδια με τη στήλη $a + b$ του προηγούμενου πίνακα. Επομένως, για να υπολογίσουμε το $a + b$ αρκεί να υπολογίσουμε το AND των $\bar{a}$ και $\bar{b}$ και να το αντιστρέψουμε. Αυτό γίνεται με το παρακάτω κύκλωμα:



Το βοηθητικό qubit αρχικοποιείται αυτή τη φορά στην κατάσταση $|1\rangle$ αντί για $|0\rangle$. Ας παρακολουθήσουμε τις ενδιάμεσες καταστάσεις, ξεκινώντας από την $|a\rangle |b\rangle |1\rangle$. Οι δύο πρώτες πύλες X αντιστρέφουν τα δύο πρώτα qubit:

$$|\phi_1\rangle = |\bar{a}\rangle |\bar{b}\rangle |1\rangle$$

Η Toffoli έχει τώρα qubit ελέγχου τα $|\bar{a}\rangle$ και $|\bar{b}\rangle$, οπότε προσθέτει (modulo 2) το γινόμενό τους στο τρίτο qubit:

$$|\phi_2\rangle = |\bar{a}\rangle |\bar{b}\rangle |1 \oplus \bar{a}\bar{b}\rangle$$

Αφού $1 \oplus x = \bar{x}$ για κάθε bit x , το τρίτο qubit είναι στην κατάσταση:

$$|1 \oplus \bar{a}\bar{b}\rangle = |\overline{\bar{a}\bar{b}}\rangle = |a + b\rangle$$

όπου στο τελευταίο βήμα χρησιμοποιήσαμε τον νόμο του De Morgan. Επομένως:

$$|\phi_3\rangle = |\bar{a}\rangle |\bar{b}\rangle |a + b\rangle$$

Οι δύο τελευταίες πύλες X αντιστρέφουν ξανά τα δύο πρώτα qubit. Αφού $\bar{\bar{a}} = a$, αυτά επανέρχονται στις αρχικές τους καταστάσεις, ενώ το τρίτο qubit δεν αλλάζει:

$$|\phi_3\rangle = |a\rangle |b\rangle |a+b\rangle$$

Ας το επιβεβαιώσουμε για την είσοδο $a = b = 0$, όπου πρέπει να πάρουμε $a + b = 0$:

$$|\phi_1\rangle = |1\rangle |1\rangle |1\rangle$$

$$|\phi_2\rangle = |1\rangle |1\rangle |1 \oplus 1 \cdot 1\rangle = |1\rangle |1\rangle |0\rangle$$

$$|\phi_3\rangle = |0\rangle |0\rangle |0\rangle$$

και για την είσοδο $a = 1, b = 0$, όπου πρέπει να πάρουμε $a + b = 1$:

$$|\phi_1\rangle = |0\rangle |1\rangle |1\rangle$$

$$|\phi_2\rangle = |0\rangle |1\rangle |1 \oplus 0 \cdot 1\rangle = |0\rangle |1\rangle |1\rangle$$

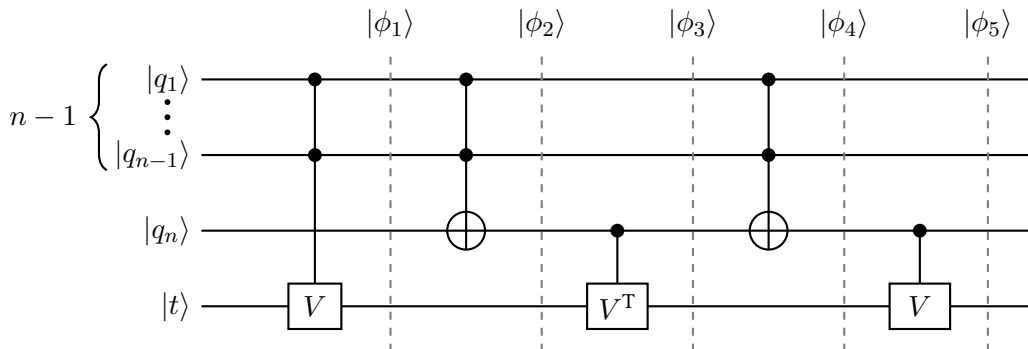
$$|\phi_3\rangle = |1\rangle |0\rangle |1\rangle$$

Και στις δύο περιπτώσεις το τρίτο qubit περιέχει το σωστό αποτέλεσμα, ενώ τα δύο πρώτα qubit επανέρχονται στις αρχικές τους καταστάσεις.

4.5 Ελεγχόμενες πύλες με n qubit ελέγχου

Στην ενότητα 4.1 ξεκινήσαμε με μία ειδική περίπτωση ελεγχόμενων πυλών, αυτές που έχουν ένα qubit ελέγχου και ένα qubit στόχου, $q = n = 1$. Έπειτα είδαμε πύλες με 2 qubit ελέγχου στην ενότητα 4.4. Το επόμενο βήμα είναι να εξετάσουμε πύλες που εφαρμόζονται σε περισσότερα qubit ελέγχου $C_n^1(U)$, με $n > 2$ ενώ συνεχίζουμε να θεωρούμε 1 qubit στόχου, $k = 1$. Ως εκ τούτου ο πίνακας U παραμένει ένας 2×2 μοναδιαίος πίνακας. Θα δείξουμε ότι οι πύλες με περισσότερα qubit ελέγχου μπορούν να κατασκευαστούν, βήμα-βήμα, από πύλες με λιγότερα qubit ελέγχου, και τελικά από πύλες με 1 qubit ελέγχου.

Έστω λοιπόν $C_n^1(U)$, με qubit ελέγχου τα $q_1, \dots, q_n$ και qubit-στόχο το t . Όπως και στην ενότητα 4.4, διαλέγουμε έναν πίνακα V με $V^2 = U$, και κατασκευάζουμε το κύκλωμα του σχήματος 4.4. Το κύκλωμα είναι ίδιο με αυτό του σχήματος 4.3, με μία μόνο διαφορά: όπου εκεί υπήρχε το qubit ελέγχου q_1 , εδώ υπάρχουν τα $n - 1$ qubit ελέγχου $q_1, \dots, q_{n-1}$. Η πρώτη, η δεύτερη και η τέταρτη πύλη είναι επομένως ελεγχόμενες πύλες με $n - 1$ qubit ελέγχου, ενώ η τρίτη και η πέμπτη είναι συνηθισμένες ελεγχόμενες πύλες με ένα qubit ελέγχου, το q_n .



Σχήμα 4.4: Κατασκευή της $C_n^1(U)$ από ελεγχόμενες πύλες με $n - 1$ και με ένα qubit ελέγχου.

Οι πύλες με $n - 1$ qubit ελέγχου είναι η $C_{n-1}^1(V)$ και η $C_{n-1}^1(X)$. Θα δείξουμε ότι εάν μπορούμε να υλοποιούμε αυτές τις πύλες τότε με το κύκλωμα μπορούμε όντως να υλοποιήσουμε την $C_n^1(U)$.

Θεωρούμε ότι στην είσοδο έχουμε μία βασική κατάσταση

$$|q_1\rangle \cdots |q_{n-1}\rangle |q_n\rangle |t\rangle$$

όπου όλα τα q_i και το t είναι bit. Για να γράφουμε σύντομα, συμβολίζουμε με Q το γινόμενο των πρώτων $n - 1$ bit ελέγχου:

$$Q = q_1 q_2 \cdots q_{n-1}$$

Το Q είναι στην ουσία το AND των $q_1, \dots, q_{n-1}$, δηλαδή $Q = 1$ μόνο όταν όλα αυτά τα bit είναι 1, και $Q = 0$ σε κάθε άλλη περίπτωση. Επομένως μία πύλη με qubit ελέγχου τα $q_1, \dots, q_{n-1}$ ενεργοποιείται ακριβώς όταν $Q = 1$. Θα χρησιμοποιήσουμε ξανά την λογική της ενότητας 4.4 όπου ο συμβολισμός V^q μας βοηθούσε να περιγράψουμε πως δρούσε η πύλη στο qubit στόχο: εάν $q = 1$ τότε $V^1 = V$ δηλαδή το qubit στόχου περνάει από την V και εάν $q = 0$ τότε $V^0 = I_2$ και το qubit στόχου δεν αλλάζει. Τα qubit $q_1, \dots, q_{n-1}$ είναι μόνο qubit ελέγχου, οπότε αυτά δεν πρέπει να αλλάζουν στην έξοδο του κυκλώματος.

Η πρώτη πύλη, η $C_{n-1}^1(V)$, εφαρμόζει τον V στο qubit-στόχο μόνο όταν $Q = 1$, δηλαδή εφαρμόζει τον V^Q :

$$|\phi_1\rangle = |q_1\rangle \cdots |q_{n-1}\rangle |q_n\rangle \otimes V^Q |t\rangle$$

Η δεύτερη πύλη, η $C_{n-1}^1(X)$, αντιστρέφει το q_n μόνο όταν $Q = 1$. Όπως και στην CNOT, αυτό σημαίνει ότι το q_n γίνεται $q_n \oplus Q$:

$$|\phi_2\rangle = |q_1\rangle \cdots |q_{n-1}\rangle |q_n \oplus Q\rangle \otimes V^Q |t\rangle$$

Η τρίτη πύλη, η $C_1^1(V^T)$, ελέγχεται από το q_n , που βρίσκεται τώρα στην κατάσταση $|q_n \oplus Q\rangle$. Επομένως εφαρμόζει τον $(V^T)^{q_n \oplus Q}$ στο qubit-στόχο:

$$|\phi_3\rangle = |q_1\rangle \cdots |q_{n-1}\rangle |q_n \oplus Q\rangle \otimes (V^T)^{q_n \oplus Q} V^Q |t\rangle$$

Η τέταρτη πύλη είναι ξανά η $C_{n-1}^1(X)$, που προσθέτει (modulo 2) ξανά το Q στο q_n . Αφού $Q \oplus Q = 0$, το q_n επανέρχεται στην αρχική του τιμή:

$$(q_n \oplus Q) \oplus Q = q_n$$

οπότε:

$$|\phi_4\rangle = |q_1\rangle \cdots |q_{n-1}\rangle |q_n\rangle \otimes (V^T)^{q_n \oplus Q} V^Q |t\rangle$$

Η πέμπτη πύλη, η $C_1^1(V)$, ελέγχεται από το q_n , που έχει επανέλθει στην κατάσταση $|q_n\rangle$, και εφαρμόζει τον V^{q_n} :

$$|\phi_5\rangle = |q_1\rangle \cdots |q_{n-1}\rangle |q_n\rangle \otimes V^{q_n} (V^T)^{q_n \oplus Q} V^Q |t\rangle$$

Ο πίνακας που εφαρμόζεται στο qubit-στόχο είναι ακριβώς ο ίδιος με αυτόν που βρήκαμε για την $C_2^1(U)$ στην ενότητα 4.4, με το Q στη θέση του q_1 και το q_n στη θέση του q_2 . Όπως και εκεί, αφού $V^T = V^{-1}$, οι εκθέτες απλώς προστίθενται:

$$V^{q_n} (V^T)^{q_n \oplus Q} V^Q = V^m$$

όπου

$$m = Q + q_n - (Q \oplus q_n)$$

Από τον πίνακα αλήθειας της ενότητας 4.4, $m = 2$ μόνο όταν $Q = q_n = 1$, και $m = 0$ σε κάθε άλλη περίπτωση. Όμως $Q = q_n = 1$ σημαίνει ότι όλα τα $q_1, \dots, q_{n-1}$ είναι 1 και επίσης $q_n = 1$, δηλαδή ότι όλα τα n qubit ελέγχου είναι 1. Αφού $V^2 = U$ και $V^0 = I_2$, καταλήγουμε στο:

$$|\phi_5\rangle = \begin{cases} |q_1 \cdots q_n\rangle \otimes \hat{U} |t\rangle, & q_1 = \cdots = q_n = 1 \\ |q_1 \cdots q_n\rangle |t\rangle, & \text{αλλιώς} \end{cases}$$

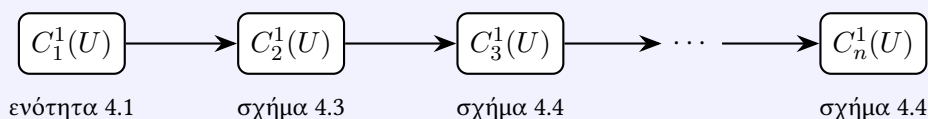
που είναι ακριβώς η δράση της $C_n^1(U)$: η U εφαρμόζεται στο qubit-στόχο μόνο όταν όλα τα n qubit ελέγχου είναι $|1\rangle$.

Η κατασκευή αυτή μας επιτρέπει να υλοποιήσουμε κάθε ελεγχόμενη πύλη $C_n^1(U)$, όσο μεγάλο κι αν είναι το n , όπως συνοψίζουμε παρακάτω.

Αποτέλεσμα 4.3: Κατασκευή κάθε $C_n^1(U)$

Κάθε ελεγχόμενη πύλη $C_n^1(U)$, με οποιοδήποτε πλήθος $n \geq 1$ qubit ελέγχου και οποιαδήποτε πύλη ενός qubit U , μπορεί να κατασκευαστεί αποκλειστικά από CNOT και πύλες ενός qubit. Η κατασκευή γίνεται βήμα-βήμα, ξεκινώντας από ένα qubit ελέγχου και προσθέτοντας κάθε φορά ένα ακόμα:

- Τις πύλες $C_1^1(U)$, με ένα qubit ελέγχου, τις κατασκευάζουμε από δύο CNOT και πύλες ενός qubit (ενότητα 4.1).
- Τις πύλες $C_2^1(U)$ τις κατασκευάζουμε με το κύκλωμα του σχήματος 4.3, που χρησιμοποιεί μόνο CNOT και πύλες C_1^1 , τις οποίες ξέρουμε ήδη να κατασκευάζουμε.
- Τις πύλες $C_3^1(U)$ τις κατασκευάζουμε με το κύκλωμα του σχήματος 4.4 για $n = 3$, που χρησιμοποιεί πύλες C_2^1 και C_1^1 . Και αυτές τις ξέρουμε ήδη από τα προηγούμενα.
- Γενικά, κάθε $C_n^1(U)$ την κατασκευάζουμε με το κύκλωμα του σχήματος 4.4, από πύλες C_{n-1}^1 και C_1^1 .



Σε κάθε βήμα χρειαζόμαστε πύλες με ένα qubit ελέγχου λιγότερο για δύο διαφορετικούς πίνακες, τον V και τον X , καθώς και μία νέα τετραγωνική ρίζα: για την $C_{n-1}^1(V)$ θα χρειαστούμε έναν πίνακα W με $W^2 = V$, και ούτω καθεξής.

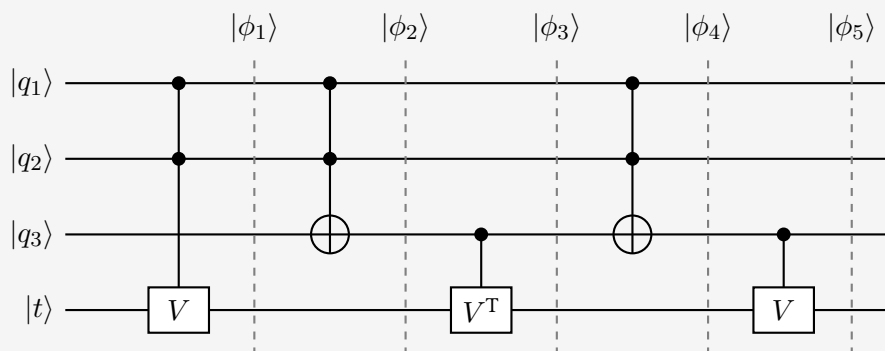
Παράδειγμα 4.17: Ελεγχόμενη πύλη Hadamard με τρία qubit ελέγχου

Ας κατασκευάσουμε την $C_3^1(H)$, με qubit ελέγχου τα q_1, q_2, q_3 και qubit-στόχο το t , εφαρμόζοντας δύο φορές την κατασκευή του σχήματος 4.4: πρώτα από τρία σε δύο qubit ελέγχου, και μετά από δύο σε ένα.

Για το πρώτο επίπεδο χρειαζόμαστε έναν πίνακα V με $V^2 = H$. Τον έχουμε ήδη βρει στο παράδειγμα 4.13:

$$V = e^{j\pi/4} R_n \left(\frac{\pi}{2} \right) = \frac{1+j}{2} (I_2 - jH)$$

όπου $n = \frac{1}{\sqrt{2}}(1, 0, 1)$ ο άξονας της Hadamard. Το κύκλωμα του σχήματος 4.4, για $n = 3$, γίνεται:



Η πρώτη πύλη, η $C_2^1(V)$, και οι δύο Toffoli έχουν δύο qubit ελέγχου, τα q_1 και q_2 . Οι $C_1^1(V^T)$ και $C_1^1(V)$ έχουν ένα qubit ελέγχου, το q_3 . Ας παρακολουθήσουμε τις ενδιάμεσες καταστάσεις, ακριβώς όπως στη γενική περίπτωση, με $Q = q_1 q_2$:

$$|\phi_1\rangle = |q_1\rangle |q_2\rangle |q_3\rangle \otimes V^{q_1 q_2} |t\rangle$$

$$|\phi_2\rangle = |q_1\rangle |q_2\rangle |q_3 \oplus q_1 q_2\rangle \otimes V^{q_1 q_2} |t\rangle$$

$$|\phi_3\rangle = |q_1\rangle |q_2\rangle |q_3 \oplus q_1 q_2\rangle \otimes (V^T)^{q_3 \oplus q_1 q_2} V^{q_1 q_2} |t\rangle$$

$$|\phi_4\rangle = |q_1\rangle |q_2\rangle |q_3\rangle \otimes (V^T)^{q_3 \oplus q_1 q_2} V^{q_1 q_2} |t\rangle$$

$$|\phi_5\rangle = |q_1\rangle |q_2\rangle |q_3\rangle \otimes \mathbf{V}^{q_3} (\mathbf{V}^T)^{q_3 \oplus q_1 q_2} \mathbf{V}^{q_1 q_2} |t\rangle$$

Για να το δούμε πιο συγκεκριμένα, ας πάρουμε την είσοδο $|111\rangle |t\rangle$, όπου και τα τρία qubit ελέγχου είναι 1. Η $C_2^1(V)$ εφαρμόζει τον $\mathbf{V}$ στο t . Η πρώτη Toffoli αντιστρέφει το q_3 σε 0, οπότε η $C_1^1(V^T)$ δεν ενεργοποιείται. Η δεύτερη Toffoli επαναφέρει το q_3 σε 1, και η τελευταία $C_1^1(V)$ εφαρμόζει ξανά τον $\mathbf{V}$. Συνολικά:

$$|\phi_5\rangle = |111\rangle \otimes \mathbf{V}\mathbf{V} |t\rangle = |111\rangle \otimes \mathbf{H} |t\rangle$$

Αν αντίθετα η είσοδος είναι $|110\rangle |t\rangle$, η $C_2^1(V)$ εφαρμόζει πάλι τον $\mathbf{V}$, αλλά τώρα η πρώτη Toffoli αντιστρέφει το q_3 σε 1, οπότε η $C_1^1(V^T)$ ενεργοποιείται και ακυρώνει τον $\mathbf{V}$. Η δεύτερη Toffoli επαναφέρει το q_3 σε 0, και η τελευταία $C_1^1(V)$ δεν ενεργοποιείται:

$$|\phi_5\rangle = |110\rangle \otimes \mathbf{V}^T \mathbf{V} |t\rangle = |110\rangle |t\rangle$$

Οι Toffoli, οι $C_1^1(V^T)$ και $C_1^1(V)$ είναι πύλες που ήδη ξέρουμε να κατασκευάζουμε. Μένει η $C_2^1(V)$, για την οποία εφαρμόζουμε την ίδια κατασκευή ακόμα μία φορά. Χρειαζόμαστε τώρα έναν πίνακα $\mathbf{W}$ με $\mathbf{W}^2 = \mathbf{V}$. Αφού ο $\mathbf{V}$ είναι μία φάση $e^{j\pi/4}$ επί μία περιστροφή κατά $\pi/2$, παίρνουμε τη μισή φάση και τη μισή γωνία:

$$\mathbf{W} = e^{j\pi/8} \mathbf{R}_n \left(\frac{\pi}{4} \right)$$

Με τον τύπο του $\mathbf{R}_n(\theta)$ (ενότητα 3.8), και συμβολίζοντας για συντομία $c = \cos(\pi/8)$ και $s = \sin(\pi/8)$:

$$\mathbf{W} = e^{j\pi/8} (c \mathbf{I}_2 - js \mathbf{H})$$

Ας επιβεβαιώσουμε ότι $\mathbf{W}^2 = \mathbf{V}$. Η φάση δίνει $(e^{j\pi/8})^2 = e^{j\pi/4}$. Για τον υπόλοιπο πίνακα, χρησιμοποιώντας ξανά ότι $\mathbf{H}^2 = \mathbf{I}_2$:

$$(c \mathbf{I}_2 - js \mathbf{H})^2 = c^2 \mathbf{I}_2 - 2jcs \mathbf{H} + j^2 s^2 \mathbf{H}^2 = (c^2 - s^2) \mathbf{I}_2 - 2jcs \mathbf{H}$$

Από τις τριγωνομετρικές ταυτότητες της διπλάσιας γωνίας έχουμε:

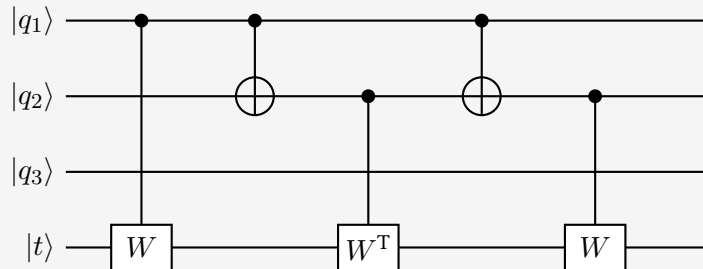
$$c^2 - s^2 = \cos \left(\frac{\pi}{4} \right) = \frac{1}{\sqrt{2}}$$

$$2cs = \sin \left(\frac{\pi}{4} \right) = \frac{1}{\sqrt{2}}$$

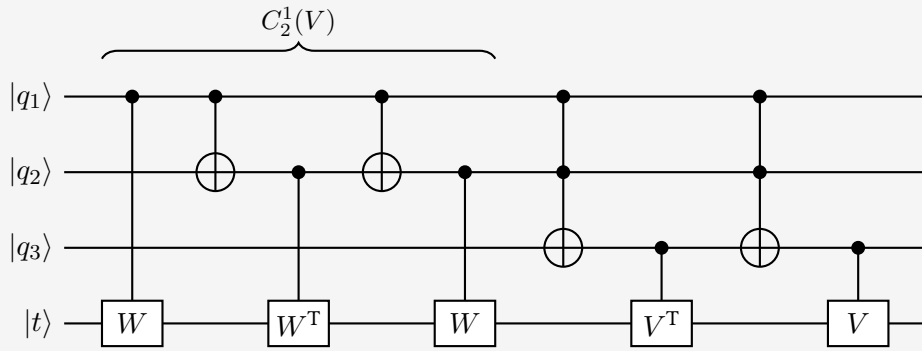
Επομένως:

$$\mathbf{W}^2 = e^{j\pi/4} \frac{1}{\sqrt{2}} (\mathbf{I}_2 - j\mathbf{H}) = e^{j\pi/4} \mathbf{R}_n \left(\frac{\pi}{2} \right) = \mathbf{V}$$

Η $C_2^1(V)$ κατασκευάζεται τότε ακριβώς όπως η $C_2^1(U)$ του σχήματος 4.3, με τον $\mathbf{W}$ στη θέση του $\mathbf{V}$. Τα qubit ελέγχου της είναι τα q_1 και q_2 , ενώ το q_3 δεν συμμετέχει καθόλου:



Αντικαθιστώντας την $C_2^1(V)$ με αυτό το κύκλωμα, παίρνουμε το πλήρες κύκλωμα της $C_3^1(H)$:



Συνολικά, η $C_3^1(H)$ κατασκευάζεται από 9 πύλες: 2 CNOT, 2 Toffoli και 5 ελεγχόμενες πύλες με ένα qubit ελέγχου (W , W^T , W , V^T , V). Οι Toffoli κατασκευάζονται με τη σειρά τους από CNOT και ελεγχόμενες πύλες ενός qubit ελέγχου (παράδειγμα 4.12), και οι ελεγχόμενες πύλες ενός qubit ελέγχου από CNOT και πύλες ενός qubit (ενότητα 4.1).

Ας επιβεβαιώσουμε αριθμητικά, με τη βοήθεια της `qulib2.py`, ότι το κύκλωμα των 9 σταδίων από πύλες του παραδείγματος 4.17 υλοποιεί πράγματι την $C_3^1(H)$.

Για να το κάνουμε αυτό χρειαζόμαστε έναν τρόπο να κατασκευάζουμε ελεγχόμενες πύλες πάνω σε έναν καταχωρητή με περισσότερα από δύο qubit, με οποιαδήποτε qubit ελέγχου και οποιοδήποτε qubit-στόχο. Η κλάση `ControlledU` της ενότητας 4.2 δεν αρκεί, αφού κατασκευάζει μόνο πίνακες 4×4 , με qubit ελέγχου πάντα το πρώτο και qubit-στόχο πάντα το δεύτερο. Γι' αυτό προσθέτουμε στην `qulib2.py` την κλάση `ControlledGate`, που είναι και αυτή υποκλάση της `Gate`. Ο constructor της δέχεται τις εξής παραμέτρους:

- `n_qubits`: το συνολικό πλήθος των qubit του καταχωρητή.
- `control`: μία λίστα με τις θέσεις των qubit ελέγχου.
- `target`: μία λίστα με τις θέσεις των qubit-στόχων.
- `through`: μία λίστα με τις θέσεις των qubit που δεν συμμετέχουν στην πύλη, αλλά απλώς «περνούν» από αυτήν χωρίς να αλλάξουν.
- `gate`: η πύλη (ένα αντικείμενο `Gate`) που εφαρμόζεται στα qubit-στόχους.
- `inverted_control`: μία προαιρετική λίστα με εκείνα τα qubit ελέγχου που έχουν *άσπρο* αντί για μαύρο κυκλάκι, δηλαδή ενεργοποιούν την πύλη όταν βρίσκονται στην κατάσταση $|0\rangle$ αντί για $|1\rangle$ (ενότητα 4.1).

Οι θέσεις των qubit αριθμούνται από το 0, όπως οι δείκτες της Python, και κάθε qubit του καταχωρητή πρέπει να ανήκει σε ακριβώς μία από τις λίστες `control`, `target` και `through`.

Ο constructor κατασκευάζει τον πίνακα της πύλης στήλη-στήλη, με τον ίδιο τρόπο που τον κατασκευάσαμε με το χέρι στα παραδείγματα 4.1 και 4.10. Κάθε στήλη `col` αντιστοιχεί σε μία βασική κατάσταση, και η βοηθητική συνάρτηση `_bits_of` βρίσκει τα bit της. Αν τα qubit ελέγχου δεν έχουν όλα την απαιτούμενη τιμή, η πύλη δεν ενεργοποιείται, η κατάσταση μένει αμετάβλητη, και η στήλη έχει απλώς ένα 1 πάνω στη διαγώνιο. Αλλιώς, ο πίνακας της `gate` εφαρμόζεται στα bit των qubit-στόχων, και τα αποτελέσματα τοποθετούνται στις κατάλληλες γραμμές με τη βοήθεια της συνάρτησης `_index_of`, που κάνει την αντίστροφη μετατροπή, από bit σε δείκτη:

```
def _bits_of(index, width):
    return [(index >> (width - 1 - i)) & 1 for i in range(width)]

def _index_of(bits):
    index = 0
    for b in bits:
        index = (index << 1) | b
    return index
```

```

class ControlledGate(Gate):
    def __init__(self, n_qubits, through, control, target, gate, inverted_control=None):
        through = list(through)
        control = list(control)
        target = list(target)
        inverted_control = list(inverted_control) if inverted_control is not None else []
        if sorted(through + control + target) != list(range(n_qubits)):
            raise ValueError(
                "through, control and target must partition all qubits "
                "of the register exactly once"
            )
        if not set(inverted_control).issubset(control):
            raise ValueError("inverted_control must be a subset of control")
        k = len(target)
        if gate.n_qubits != k:
            raise ValueError("gate must act on exactly len(target) qubits")
        U = gate.calc_U()

        dim = 2 ** n_qubits
        full = np.zeros((dim, dim), dtype=complex)
        for col in range(dim):
            bits = _bits_of(col, n_qubits)
            required = [0 if q in inverted_control else 1 for q in control]
            if not all(bits[q] == r for q, r in zip(control, required)):
                full[col, col] = 1
                continue
            t_in = _index_of([bits[q] for q in target])
            for t_out, amplitude in enumerate(U[:, t_in]):
                if amplitude == 0:
                    continue
                out_bits = list(bits)
                for pos, q in enumerate(target):
                    out_bits[q] = _bits_of(t_out, k)[pos]
                full[_index_of(out_bits), col] = amplitude

        super().__init__(U=full)

```

Στο script χρησιμοποιούμε τους πίνακες V και W ακριβώς όπως στο παράδειγμα 4.17, και ελέγχουμε πρώτα ότι $V^2 = H$ και $W^2 = V$. Στη συνέχεια κατασκευάζουμε καθεμία από τις 9 πύλες του κυκλώματος ως ControlledGate πάνω σε έναν καταχωρητή 4 qubit, όπου τα q_1, q_2, q_3, t βρίσκονται στις θέσεις 0, 1, 2, 3. Για παράδειγμα, η πρώτη πύλη, η ελεγχόμενη W , έχει qubit ελέγχου το q_1 (θέση 0) και qubit-στόχο το t (θέση 3), ενώ τα q_2, q_3 (θέσεις 1, 2) περνούν από αυτήν χωρίς να αλλάξουν. Οι CNOT και οι Toffoli είναι απλώς ControlledGate με gate την X . Συνθέτουμε τις πύλες με τον τελεστή @ της Gate, προσέχοντας ότι, όπως και στο γινόμενο πινάκων, η πύλη που εφαρμόζεται πρώτη γράφεται τελευταία, δεξιά. Τέλος, συγκρίνουμε το αποτέλεσμα με την $C_3^1(H)$, κατασκευασμένη απευθείας ως μία ControlledGate με τρία qubit ελέγχου. Το script βρίσκεται στο αρχείο `python/ex_C3_Hadamard.py`:

```

import numpy as np

from qulib2 import Gate, GateH, GateX, ControlledGate

np.set_printoptions(precision=4, suppress=True)

# Example 4.17: build C_3^1(H), with control qubits q1, q2, q3 and target
# qubit t, from CNOT, Toffoli and gates with a single control qubit.
#
# Qubit order throughout: (q1, q2, q3, t), positions (0, 1, 2, 3).

```

```

H = GateH().U
I2 = np.eye(2)

# V with V^2 = H (Example 4.13) and W with W^2 = V
V = (1 + 1j) / 2 * (I2 - 1j * H)
c, s = np.cos(np.pi / 8), np.sin(np.pi / 8)
W = np.exp(1j * np.pi / 8) * (c * I2 - 1j * s * H)
print(f"V @ V == H: {np.allclose(V @ V, H)}")
print(f"W @ W == V: {np.allclose(W @ W, V)}")

X = GateX()

# the five gates of C_2^1(V): controls q1, q2, target t (q3 not involved)
W_q1_t = ControlledGate(n_qubits=4, through=[1, 2], control=[0], target=[3], gate=Gate(U=W))
cnot_q1_q2 = ControlledGate(n_qubits=4, through=[2, 3], control=[0], target=[1], gate=X)
Wt_q2_t = ControlledGate(n_qubits=4, through=[0, 2], control=[1], target=[3], gate=Gate(U=W.
    conj().T))
W_q2_t = ControlledGate(n_qubits=4, through=[0, 2], control=[1], target=[3], gate=Gate(U=W))

C2_V = W_q2_t @ cnot_q1_q2 @ Wt_q2_t @ cnot_q1_q2 @ W_q1_t

# the remaining four gates: two Toffoli (q1, q2 -> q3) and the
# gates V^T, V controlled by q3
toffoli = ControlledGate(n_qubits=4, through=[3], control=[0, 1], target=[2], gate=X)
Vt_q3_t = ControlledGate(n_qubits=4, through=[0, 1], control=[2], target=[3], gate=Gate(U=V.
    conj().T))
V_q3_t = ControlledGate(n_qubits=4, through=[0, 1], control=[2], target=[3], gate=Gate(U=V))

circuit = V_q3_t @ toffoli @ Vt_q3_t @ toffoli @ C2_V

# C_3^1(H) built directly, with three control qubits
reference = ControlledGate(n_qubits=4, through=[], control=[0, 1, 2], target=[3], gate=GateH()
    )

print(f"circuit matches C_3^1(H): {np.allclose(circuit.U, reference.U)}")

```

4.6 Υλοποίηση μοναδιαίων τελεστών

Μέχρι τώρα περιγράψαμε αρκετές συγκεκριμένες κβαντικές πύλες, όπως πύλες ενός qubit, ελεγχόμενες πύλες, την Toffoli και την Fredkin. Δεν έχουμε δείξει όμως ότι με τέτοιες πύλες μπορούμε να υλοποιήσουμε οποιονδήποτε μοναδιαίο τελεστή. Στην ενότητα αυτή θα κάνουμε το πρώτο βήμα προς αυτή την κατεύθυνση. Θα δείξουμε ότι κάθε μοναδιαίος πίνακας U , όσο μεγάλος κι αν είναι, μπορεί να γραφτεί ως γινόμενο πολύ απλούστερων μοναδιαίων πινάκων, που ο καθένας διαφέρει από τον ταυτοτικό πίνακα σε το πολύ τέσσερα στοιχεία. Στην επόμενη ενότητα θα δούμε πώς κάθε τέτοιος απλός πίνακας υλοποιείται με ελεγχόμενες πύλες.

Θεωρούμε ένα σύστημα n qubit. Όπως και στις προηγούμενες ενότητες, οι καταστάσεις βάσης θεωρούμε πως είναι οι

$$|\psi_1\rangle, |\psi_2\rangle, \dots, |\psi_N\rangle$$

όπου $N = 2^n$. Για παράδειγμα, για δύο qubit έχουμε $N = 4$ και $|\psi_1\rangle = |00\rangle$, $|\psi_2\rangle = |01\rangle$, $|\psi_3\rangle = |10\rangle$, $|\psi_4\rangle = |11\rangle$. Όπως έχουμε χρησιμοποιήσει και πολλές φορές στο παρελθόν, ένας μοναδιαίος τελεστής $\hat{U}$ περιγράφεται από έναν $N \times N$ μοναδιαίο πίνακα U με στοιχεία

$$u_{pq} = \langle \psi_p | \hat{U} | \psi_q \rangle$$

όπου $p, q = 1, \dots, N$.

Ας ορίσουμε πρώτα τους απλούς πίνακες που θα χρησιμοποιήσουμε. Λέμε ότι ένας $N \times N$ μοναδιαίος πίνακας είναι *διεπίπεδος* (*two-level*) αν δρα μη τετριμμένα μόνο σε δύο από τις βασικές καταστάσεις, έστω τις

$|\psi_r\rangle$ και $|\psi_s\rangle$, και αφήνει όλες τις υπόλοιπες αμετάβλητες:

$$U |\psi_p\rangle = |\psi_p\rangle$$

για κάθε $p \neq r, s$. Οι στήλες του πίνακα που αντιστοιχούν σε αυτές τις καταστάσεις είναι επομένως ίδιες με τις στήλες του ταυτοτικού πίνακα. Ας δούμε ένα παράδειγμα για $N = 4$, με $r = 2$ και $s = 4$:

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & c & 0 & d \end{bmatrix}$$

Ο πίνακας αυτός αφήνει αμετάβλητες τις $|\psi_1\rangle$ και $|\psi_3\rangle$, ενώ στις $|\psi_2\rangle$ και $|\psi_4\rangle$ δρα ως εξής:

$$U |\psi_2\rangle = a |\psi_2\rangle + c |\psi_4\rangle$$

$$U |\psi_4\rangle = b |\psi_2\rangle + d |\psi_4\rangle$$

Δηλαδή διαφέρει από τον ταυτοτικό πίνακα I_4 μόνο στα τέσσερα στοιχεία που βρίσκονται στις γραμμές και στις στήλες 2 και 4. Τα στοιχεία αυτά σχηματίζουν τον 2×2 πίνακα

$$\tilde{U} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

που πρέπει να είναι κι αυτός μοναδιαίος, ώστε ο U να είναι μοναδιαίος, όπως δείχνουμε στο παρακάτω παράδειγμα.

Παράδειγμα 4.18: Ένας διεπίπεδος πίνακας είναι μοναδιαίος όταν ο $\tilde{U}$ είναι μοναδιαίος

Ας δείξουμε ότι ο διεπίπεδος πίνακας

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & c & 0 & d \end{bmatrix}$$

είναι μοναδιαίος, δηλαδή ικανοποιεί $U^T U = I_4$, ακριβώς όταν ο 2×2 πίνακας

$$\tilde{U} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

είναι μοναδιαίος, δηλαδή ικανοποιεί $\tilde{U}^T \tilde{U} = I_2$.

Ο συζυγής ανάστροφος του U προκύπτει ανταλλάσσοντας γραμμές με στήλες και παίρνοντας τους συζυγείς μιγαδικούς των στοιχείων:

$$U^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & a^* & 0 & c^* \\ 0 & 0 & 1 & 0 \\ 0 & b^* & 0 & d^* \end{bmatrix}$$

Ας υπολογίσουμε το γινόμενο $U^T U$. Το στοιχείο στη γραμμή p και στη στήλη q του γινομένου προκύπτει πολλαπλασιάζοντας τη γραμμή p του U^T με τη στήλη q του U , στοιχείο προς στοιχείο, και προσθέτοντας. Για παράδειγμα, το στοιχείο στη γραμμή 2 και στη στήλη 4 είναι:

$$0 \cdot 0 + a^* \cdot b + 0 \cdot 0 + c^* \cdot d = a^* b + c^* d$$

Η γραμμή 1 του U^T είναι $(1, 0, 0, 0)$, οπότε η γραμμή 1 του γινομένου είναι απλώς η γραμμή 1 του U , δηλαδή $(1, 0, 0, 0)$. Με τον ίδιο τρόπο, η γραμμή 3 του γινομένου είναι $(0, 0, 1, 0)$. Υπολογίζοντας με τον

ίδιο τρόπο και τα υπόλοιπα στοιχεία, βρίσκουμε:

$$U^T U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & |a|^2 + |c|^2 & 0 & a^*b + c^*d \\ 0 & 0 & 1 & 0 \\ 0 & b^*a + d^*c & 0 & |b|^2 + |d|^2 \end{bmatrix}$$

όπου χρησιμοποιήσαμε ότι $a^*a = |a|^2$, και ομοίως για τα b, c, d . Ας υπολογίσουμε τώρα και το γινόμενο για τον 2×2 πίνακα:

$$\tilde{U}^T \tilde{U} = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} |a|^2 + |c|^2 & a^*b + c^*d \\ b^*a + d^*c & |b|^2 + |d|^2 \end{bmatrix}$$

Βλέπουμε ότι τα στοιχεία του $U^T U$ στις γραμμές και στις στήλες 2 και 4 είναι ακριβώς τα στοιχεία του $\tilde{U}^T \tilde{U}$, ενώ όλα τα υπόλοιπα στοιχεία είναι ίδια με αυτά του I_4 . Επομένως $U^T U = I_4$ ακριβώς όταν:

$$|a|^2 + |c|^2 = 1$$

$$|b|^2 + |d|^2 = 1$$

$$a^*b + c^*d = 0$$

που είναι ακριβώς οι συνθήκες για να ισχύει $\tilde{U}^T \tilde{U} = I_2$. Το τέταρτο στοιχείο, $b^*a + d^*c$, είναι ο συζυγής του $a^*b + c^*d$, οπότε μηδενίζεται μαζί του. Ο U είναι λοιπόν μοναδιαίος αν και μόνο αν ο $\tilde{U}$ είναι μοναδιαίος.

Το ίδιο αποτέλεσμα ισχύει για διεπίπεδους πίνακες οποιουδήποτε μεγέθους, όπως δείχνουμε στο επόμενο παράδειγμα.

Παράδειγμα 4.19: Ο διεπίπεδος πίνακας στη γενική περίπτωση

Θεωρούμε τώρα έναν $N \times N$ διεπίπεδο πίνακα U , που δρα μη τετριμμένα στις καταστάσεις $|\psi_r\rangle$ και $|\psi_s\rangle$, με $r < s$. Ο πίνακας είναι ίδιος με τον ταυτοτικό I_N , εκτός από τα τέσσερα στοιχεία στις γραμμές και στις στήλες r και s :

$$U = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & a & \cdots & b & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & d & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

όπου το a βρίσκεται στη γραμμή r και στη στήλη r , το b στη γραμμή r και στη στήλη s , το c στη γραμμή s και στη στήλη r , και το d στη γραμμή s και στη στήλη s . Θα δείξουμε ότι ο U είναι μοναδιαίος ακριβώς όταν ο 2×2 πίνακας

$$\tilde{U} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

είναι μοναδιαίος.

Για έναν πίνακα με N γραμμές και στήλες δεν μπορούμε πλέον να γράψουμε όλο το γινόμενο $U^T U$, οπότε θα δουλέψουμε με τις στήλες του U . Το στοιχείο στη γραμμή p και στη στήλη q του $U^T U$ προκύπτει πολλαπλασιάζοντας τη γραμμή p του U^T με τη στήλη q του U . Η γραμμή p του U^T είναι όμως η στήλη p του U , με συζυγείς μιγαδικούς στα στοιχεία της. Επομένως:

$$(U^T U)_{pq} = \sum_{k=1}^N u_{kp}^* u_{kq}$$

δηλαδή το στοιχείο (p, q) του $U^T U$ είναι το εσωτερικό γινόμενο της στήλης p με τη στήλη q του U . Για να ισχύει $U^T U = I_N$, πρέπει αυτό το εσωτερικό γινόμενο να είναι 1 όταν $p = q$ και 0 όταν $p \neq q$.

Ας δούμε λοιπόν πώς είναι οι στήλες του U . Μία στήλη p με $p \neq r, s$ είναι ίδια με τη στήλη p του ταυτοτικού πίνακα: έχει 1 στη γραμμή p και 0 σε όλες τις άλλες γραμμές. Η στήλη r έχει το a στη γραμμή r , το c στη γραμμή s , και 0 σε όλες τις άλλες γραμμές. Τέλος, η στήλη s έχει το b στη γραμμή r , το d στη γραμμή s , και 0 σε όλες τις άλλες γραμμές. Διακρίνουμε τώρα τρεις περιπτώσεις.

Αν κανένας από τους δείκτες p, q δεν είναι r ή s , οι στήλες p και q είναι στήλες του ταυτοτικού πίνακα. Αν $p = q$, το εσωτερικό γινόμενο είναι $1 \cdot 1 = 1$. Αν $p \neq q$, οι δύο στήλες έχουν το 1 τους σε διαφορετικές γραμμές, οπότε σε κάθε γραμμή τουλάχιστον ένας από τους δύο παράγοντες είναι 0, και το εσωτερικό γινόμενο είναι 0. Σε αυτή την περίπτωση λοιπόν το στοιχείο (p, q) είναι πάντα ίδιο με του I_N , ό,τι κι αν είναι τα a, b, c, d .

Αν ένας μόνο από τους δείκτες, ας πούμε ο p , δεν είναι r ή s , ενώ ο q είναι r ή s , τότε η στήλη p έχει το 1 της στη γραμμή p , ενώ η στήλη q έχει μη μηδενικά στοιχεία μόνο στις γραμμές r και s . Αφού $p \neq r, s$, οι δύο στήλες δεν έχουν ποτέ ταυτόχρονα μη μηδενικό στοιχείο στην ίδια γραμμή, οπότε το εσωτερικό γινόμενο είναι 0, όπως και στον I_N , αφού $p \neq q$. Το ίδιο ισχύει αν ο q είναι αυτός που δεν είναι r ή s .

Μένει η περίπτωση όπου και οι δύο δείκτες είναι r ή s . Οι στήλες r και s έχουν μη μηδενικά στοιχεία μόνο στις γραμμές r και s , οπότε στο άθροισμα συμμετέχουν μόνο οι όροι με $k = r$ και $k = s$. Βρίσκουμε:

$$(U^T U)_{rr} = a^* a + c^* c = |a|^2 + |c|^2$$

$$(U^T U)_{ss} = b^* b + d^* d = |b|^2 + |d|^2$$

$$(U^T U)_{rs} = a^* b + c^* d$$

$$(U^T U)_{sr} = b^* a + d^* c$$

Αυτά είναι ακριβώς τα τέσσερα στοιχεία του $\tilde{U}^T \tilde{U}$, που υπολογίσαμε στο παράδειγμα 4.18.

Συνοψίζοντας, το $U^T U$ είναι ίδιο με τον I_N παντού εκτός από τις γραμμές και τις στήλες r, s , όπου περιέχει τον πίνακα $\tilde{U}^T \tilde{U}$. Επομένως $U^T U = I_N$ ακριβώς όταν $\tilde{U}^T \tilde{U} = I_2$, δηλαδή ο U είναι μοναδιαίος αν και μόνο αν ο $\tilde{U}$ είναι μοναδιαίος.

Ο διεπίπεδος πίνακας μοιάζει επομένως με μία πύλη ενός qubit, που όμως δρα στο ζευγάρι καταστάσεων $|\psi_r\rangle, |\psi_s\rangle$ αντί για το ζευγάρι $|0\rangle, |1\rangle$. Παρακάτω θα δείξουμε το εξής πολύ σημαντικό αποτέλεσμα:

Αποτέλεσμα 4.4: Ανάλυση σε διεπίπεδους μοναδιαίους πίνακες

Κάθε $N \times N$ μοναδιαίος πίνακας U μπορεί να γραφτεί ως γινόμενο

$$U = U_1 U_2 \cdots U_d$$

όπου κάθε U_i είναι ένας διεπίπεδος μοναδιαίος πίνακας, και το πλήθος τους είναι το πολύ

$$d \leq \frac{N(N-1)}{2}$$

Ας δούμε πώς βρίσκουμε τους πίνακες U_i που πληρούν την παραπάνω συνθήκη. Η ιδέα είναι να πολλαπλασιάζουμε τον U από τα αριστερά με διεπίπεδους πίνακες $A_1, A_2, \dots$, επιλεγμένους έτσι ώστε κάθε φορά να μηδενίζεται ένα ακόμα στοιχείο εκτός της διαγωνίου. Συνεχίζουμε μέχρι το αποτέλεσμα να γίνει τόσο απλό που να είναι το ίδιο ένας διεπίπεδος πίνακας Γ :

$$A_k \cdots A_2 A_1 U = \Gamma$$

Για να λύσουμε ως προς U , πολλαπλασιάζουμε και τα δύο μέλη από τα αριστερά πρώτα με τον A_k^T , μετά με τον A_{k-1}^T , και ούτω καθεξής. Αφού κάθε A_i είναι μοναδιαίος, ισχύει $A_i^T A_i = I$, οπότε οι πίνακες αυτοί

«φεύγουν» ένας-ένας από το αριστερό μέλος, και καταλήγουμε στο:

$$\mathbf{U} = \mathbf{A}_1^T \mathbf{A}_2^T \cdots \mathbf{A}_k^T \mathbf{\Gamma}$$

Ο συζυγής ανάστροφος ενός διεπίπεδου πίνακα είναι κι αυτός διεπίπεδος, ως προς τις ίδιες δύο καταστάσεις, αφού αλλάζει μόνο ο 2×2 πίνακας $\tilde{\mathbf{U}}$, που γίνεται $\tilde{\mathbf{U}}^T$, όπως δείχνουμε στο παρακάτω παράδειγμα.

Παράδειγμα 4.20: Ο συζυγής ανάστροφος ενός διεπίπεδου πίνακα

Θεωρούμε ξανά τον $N \times N$ διεπίπεδο πίνακα $\mathbf{U}$ του παραδείγματος 4.19, που δρα μη τετρωμένα στις καταστάσεις $|\psi_r\rangle$ και $|\psi_s\rangle$ μέσω του 2×2 πίνακα

$$\tilde{\mathbf{U}} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Ο συζυγής ανάστροφος $\mathbf{U}^T$ προκύπτει ανταλλάσσοντας γραμμές με στήλες και παίρνοντας τους συζυγείς μιγαδικούς. Δηλαδή το στοιχείο του $\mathbf{U}^T$ στη γραμμή p και στη στήλη q είναι ο συζυγής του στοιχείου του $\mathbf{U}$ στη γραμμή q και στη στήλη p :

$$(\mathbf{U}^T)_{pq} = u_{qp}^*$$

Ας δούμε τι σημαίνει αυτό για τα στοιχεία του $\mathbf{U}^T$.

Αν ένας τουλάχιστον από τους δείκτες p, q δεν είναι r ή s , το στοιχείο u_{qp} είναι ίδιο με το αντίστοιχο στοιχείο του ταυτοτικού πίνακα, δηλαδή 1 αν $p = q$ και 0 αλλιώς. Αφού οι αριθμοί 0 και 1 είναι πραγματικοί, οι συζυγείς τους είναι ο εαυτός τους, οπότε και το $(\mathbf{U}^T)_{pq}$ είναι ίδιο με το αντίστοιχο στοιχείο του ταυτοτικού πίνακα. Ο $\mathbf{U}^T$ είναι επομένως ίδιος με τον $\mathbf{I}_N$ έξω από τις γραμμές και τις στήλες r, s .

Μένουν τα τέσσερα στοιχεία στις γραμμές και στις στήλες r, s :

$$(\mathbf{U}^T)_{rr} = u_{rr}^* = a^*$$

$$(\mathbf{U}^T)_{rs} = u_{sr}^* = c^*$$

$$(\mathbf{U}^T)_{sr} = u_{rs}^* = b^*$$

$$(\mathbf{U}^T)_{ss} = u_{ss}^* = d^*$$

Τα στοιχεία αυτά σχηματίζουν τον 2×2 πίνακα

$$\begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} = \tilde{\mathbf{U}}^T$$

Επομένως ο $\mathbf{U}^T$ είναι κι αυτός διεπίπεδος, ως προς τις ίδιες καταστάσεις $|\psi_r\rangle$ και $|\psi_s\rangle$, με τον $\tilde{\mathbf{U}}^T$ στη θέση του $\tilde{\mathbf{U}}$. Για $N = 4$, $r = 2$ και $s = 4$, αυτό φαίνεται και στον πίνακα $\mathbf{U}^T$ που γράψαμε αναλυτικά στο παράδειγμα 4.18.

Ας δούμε και ένα αριθμητικό παράδειγμα, με $N = 4$, $r = 1$ και $s = 3$:

$$\mathbf{U} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{j}{\sqrt{2}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{j}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Εδώ $a = d = 1/\sqrt{2}$ και $b = c = j/\sqrt{2}$. Ανταλλάσσοντας γραμμές με στήλες ο πίνακας δεν αλλάζει, αφού $b = c$. Παίρνοντας όμως τους συζυγείς, το j γίνεται $-j$:

$$\mathbf{U}^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{j}{\sqrt{2}} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{j}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ο U^T διαφέρει από τον I_4 μόνο στις γραμμές και στις στήλες 1 και 3, όπως και ο U . Μπορούμε επίσης να ελέγξουμε ότι $U^T U = I_4$. Αρκεί, σύμφωνα με το παράδειγμα 4.19, να υπολογίσουμε το γινόμενο των 2×2 πινάκων:

$$\tilde{U}^T \tilde{U} = \frac{1}{2} \begin{bmatrix} 1 & -j \\ -j & 1 \end{bmatrix} \begin{bmatrix} 1 & j \\ j & 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1-j^2 & j-j \\ -j+j & -j^2+1 \end{bmatrix} = I_2$$

αφού $j^2 = -1$.

Από την σχέση:

$$U = A_1^T A_2^T \cdots A_k^T \Gamma$$

που είδαμε παραπάνω προκύπτει ότι έχουμε γράψει τον U ως γινόμενο διεπίπεδων πινάκων, με $U_i = A_i^T$ για $i = 1, \dots, k$ και $U_{k+1} = \Gamma$.

Για να επιλέξουμε κατάλληλα τους A_i ένα βασικό εργαλείο που θα χρειαστούμε είναι ο μηδενισμός ενός στοιχείου ενός πίνακα με τη βοήθεια ενός διεπίπεδου πίνακα. Ας υποθέσουμε ότι σε μία στήλη του πίνακα M , στις γραμμές r και s , βρίσκονται τα στοιχεία a και b , και θέλουμε να μηδενίσουμε το b . Αν ονομάσουμε t αυτή τη στήλη, ο πίνακας M έχει τη μορφή:

$$M = \begin{bmatrix} m_{11} & \cdots & m_{1t} & \cdots & m_{1N} \\ \vdots & & \vdots & & \vdots \\ m_{r1} & \cdots & a & \cdots & m_{rN} \\ \vdots & & \vdots & & \vdots \\ m_{s1} & \cdots & b & \cdots & m_{sN} \\ \vdots & & \vdots & & \vdots \\ m_{N1} & \cdots & m_{Nt} & \cdots & m_{NN} \end{bmatrix}$$

όπου $a = m_{rt}$ και $b = m_{st}$. Ο M είναι ένας οποιοσδήποτε $N \times N$ πίνακας, όχι απαραίτητα διεπίπεδος: στο πρώτο βήμα είναι ο ίδιος ο U , και στα επόμενα βήματα είναι το γινόμενο που έχει προκύψει μέχρι εκείνη τη στιγμή.

Με τη βοήθεια των στοιχείων a και b του M ορίζουμε τους αριθμούς

$$x = \frac{a}{\sqrt{|a|^2 + |b|^2}}$$

$$y = \frac{b}{\sqrt{|a|^2 + |b|^2}}$$

και τον διεπίπεδο πίνακα A που διαφέρει από τον ταυτοτικό μόνο στις γραμμές και στις στήλες r και s , όπου περιέχει τον 2×2 πίνακα

$$\tilde{A} = \begin{bmatrix} x^* & y^* \\ -y & x \end{bmatrix}$$

Δηλαδή ο A έχει τη μορφή:

$$A = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & x^* & \cdots & y^* & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -y & \cdots & x & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}$$

όπου τα x^* και y^* βρίσκονται στη γραμμή r , και τα $-y$ και x στη γραμμή s , στις στήλες r και s αντίστοιχα. Ο $\tilde{A}$ είναι μοναδιαίος. Πράγματι, από τους ορισμούς των x, y έχουμε $|x|^2 + |y|^2 = 1$, και επομένως:

$$\tilde{A} \tilde{A}^T = \begin{bmatrix} x^* & y^* \\ -y & x \end{bmatrix} \begin{bmatrix} x & -y^* \\ y & x^* \end{bmatrix} = \begin{bmatrix} |x|^2 + |y|^2 & -x^* y^* + y^* x^* \\ -yx + xy & |y|^2 + |x|^2 \end{bmatrix} = I_2$$

Ας δούμε τι κάνει ο $\mathbf{A}$ όταν πολλαπλασιάζει τον $\mathbf{M}$ από τα αριστερά. Η γραμμή p του γινομένου $\mathbf{AM}$ προκύπτει πολλαπλασιάζοντας τη γραμμή p του $\mathbf{A}$ με τα στοιχεία κάθε στήλης του $\mathbf{M}$ και αθροίζοντας. Για $p \neq r, s$, η γραμμή p του $\mathbf{A}$ είναι ίδια με του ταυτοτικού πίνακα, οπότε η γραμμή p του $\mathbf{AM}$ είναι ίδια με τη γραμμή p του $\mathbf{M}$. Οι μόνες γραμμές που αλλάζουν είναι οι r και s . Η γραμμή r του $\mathbf{A}$ έχει μη μηδενικά στοιχεία μόνο στις θέσεις r και s , οπότε το στοιχείο του $\mathbf{AM}$ στη γραμμή r και σε μία στήλη k είναι:

$$(\mathbf{AM})_{rk} = x^* m_{rk} + y^* m_{sk}$$

Με τον ίδιο τρόπο, στη γραμμή s :

$$(\mathbf{AM})_{sk} = -y m_{rk} + x m_{sk}$$

Δηλαδή η νέα γραμμή r είναι ένας συνδυασμός των γραμμών r και s του $\mathbf{M}$, και το ίδιο ισχύει για τη νέα γραμμή s . Στη στήλη t , όπου $m_{rt} = a$ και $m_{st} = b$, η νέα τιμή στη γραμμή r είναι:

$$(\mathbf{AM})_{rt} = x^* a + y^* b = \frac{a^* a + b^* b}{\sqrt{|a|^2 + |b|^2}} = \frac{|a|^2 + |b|^2}{\sqrt{|a|^2 + |b|^2}} = \sqrt{|a|^2 + |b|^2}$$

ενώ η νέα τιμή στη γραμμή s είναι:

$$(\mathbf{AM})_{st} = -y a + x b = \frac{-ba + ab}{\sqrt{|a|^2 + |b|^2}} = 0$$

Ο πίνακας $\mathbf{A}$ μηδενίζει λοιπόν το στοιχείο b του $\mathbf{M}$, και «μεταφέρει» όλο το μέτρο των a, b στη γραμμή r , ως έναν θετικό πραγματικό αριθμό. Για να γράψουμε το αποτέλεσμα σύντομα, ονομάζουμε ρ αυτόν τον αριθμό:

$$\rho = \sqrt{|a|^2 + |b|^2}$$

και m'_{rk}, m'_{sk} τα νέα στοιχεία των γραμμών r και s :

$$m'_{rk} = x^* m_{rk} + y^* m_{sk}$$

$$m'_{sk} = -y m_{rk} + x m_{sk}$$

Ο πίνακας που προκύπτει έχει τότε τη μορφή:

$$\mathbf{AM} = \begin{bmatrix} m_{11} & \cdots & m_{1t} & \cdots & m_{1N} \\ \vdots & & \vdots & & \vdots \\ m'_{r1} & \cdots & \rho & \cdots & m'_{rN} \\ \vdots & & \vdots & & \vdots \\ m'_{s1} & \cdots & 0 & \cdots & m'_{sN} \\ \vdots & & \vdots & & \vdots \\ m_{N1} & \cdots & m_{Nt} & \cdots & m_{NN} \end{bmatrix}$$

Οι υπόλοιπες γραμμές του $\mathbf{M}$ μένουν ίδιες, οπότε τα μηδενικά που έχουμε ήδη δημιουργήσει σε αυτές δεν χαλάνε. Αν το b είναι ήδη μηδέν, δεν χρειάζεται να κάνουμε τίποτα. Αν ο $\mathbf{M}$ είναι μοναδιαίος, όπως ο $\mathbf{U}$, τότε και ο $\mathbf{AM}$ είναι μοναδιαίος, ως γινόμενο μοναδιαίων πινάκων.

Μπορούμε τώρα να περιγράψουμε ολόκληρο τον αλγόριθμο, εφαρμόζοντας το παραπάνω εργαλείο ξανά και ξανά. Ο ρόλος του $\mathbf{M}$ παίζεται κάθε φορά από τον πίνακα που έχουμε μέχρι εκείνη τη στιγμή: αρχικά από τον $\mathbf{U}$, μετά από τον $\mathbf{A}_1 \mathbf{U}$, μετά από τον $\mathbf{A}_2 \mathbf{A}_1 \mathbf{U}$, και ούτω καθεξής. Ξεκινάμε από την πρώτη στήλη του $\mathbf{U}$ και χρησιμοποιούμε ως «γραμμή αναφοράς» τη γραμμή 1. Ο πίνακας $\mathbf{U}$ είναι:

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ u_{21} & u_{22} & \cdots & u_{2N} \\ u_{31} & u_{32} & \cdots & u_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{NN} \end{bmatrix}$$

Στο πρώτο βήμα μηδενίζουμε το στοιχείο της γραμμής 2, με $a = u_{11}$ και $b = u_{21}$. Ο πίνακας A_1 δρα στις γραμμές και στις στήλες 1, 2:

$$A_1 = \begin{bmatrix} x_1^* & y_1^* & 0 & \cdots & 0 \\ -y_1 & x_1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

όπου τα x_1, y_1 υπολογίζονται από τα a, b όπως παραπάνω. Μετά από αυτό το βήμα αλλάζουν μόνο οι γραμμές 1 και 2. Στον πίνακα που ακολουθεί συμβολίζουμε με $*$ τα στοιχεία που άλλαξαν, αλλά των οποίων η ακριβής τιμή δεν μας ενδιαφέρει εδώ:

$$A_1 U = \begin{bmatrix} \rho_1 & * & \cdots & * \\ 0 & * & \cdots & * \\ u_{31} & u_{32} & \cdots & u_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{NN} \end{bmatrix}$$

όπου

$$\rho_1 = \sqrt{|u_{11}|^2 + |u_{21}|^2}$$

Οι γραμμές $3, \dots, N$ είναι ίδιες με του U . Στο δεύτερο βήμα μηδενίζουμε το στοιχείο της γραμμής 3, με $a = \rho_1$, δηλαδή την τρέχουσα τιμή της γραμμής 1, και $b = u_{31}$. Ο πίνακας A_2 δρα στις γραμμές και στις στήλες 1, 3:

$$A_2 = \begin{bmatrix} x_2^* & 0 & y_2^* & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ -y_2 & 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

Αυτή τη φορά αλλάζουν μόνο οι γραμμές 1 και 3:

$$A_2 A_1 U = \begin{bmatrix} \rho_2 & * & \cdots & * \\ 0 & * & \cdots & * \\ 0 & * & \cdots & * \\ u_{41} & u_{42} & \cdots & u_{4N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{NN} \end{bmatrix}$$

όπου

$$\rho_2 = \sqrt{\rho_1^2 + |u_{31}|^2} = \sqrt{|u_{11}|^2 + |u_{21}|^2 + |u_{31}|^2}$$

Το μηδενικό στη γραμμή 2 δεν χάθηκε, αφού ο A_2 δεν αγγίζει τη γραμμή 2. Συνεχίζουμε με τον ίδιο τρόπο: στο βήμα i ο πίνακας A_i δρα στις γραμμές και στις στήλες 1 και $i+1$, και μηδενίζει το στοιχείο της γραμμής $i+1$. Στο τελευταίο βήμα, το βήμα $N-1$, ο πίνακας A_{N-1} δρα στις γραμμές και στις στήλες 1 και N :

$$A_{N-1} = \begin{bmatrix} x_{N-1}^* & 0 & \cdots & 0 & y_{N-1}^* \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ -y_{N-1} & 0 & \cdots & 0 & x_{N-1} \end{bmatrix}$$

Μετά από αυτό το βήμα, όλα τα στοιχεία της πρώτης στήλης κάτω από τη γραμμή 1 έχουν μηδενιστεί:

$$A_{N-1} \cdots A_1 U = \begin{bmatrix} \rho_{N-1} & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & * & \cdots & * \end{bmatrix}$$

όπου

$$\rho_{N-1} = \sqrt{|u_{11}|^2 + |u_{21}|^2 + \cdots + |u_{N1}|^2}$$

Μετά από αυτά τα $N - 1$ βήματα, ο πίνακας

$$\mathbf{\Gamma} = \mathbf{A}_{N-1} \cdots \mathbf{A}_1 \mathbf{U}$$

έχει στην πρώτη στήλη μόνο ένα μη μηδενικό στοιχείο, το $\gamma_{11} = \rho_{N-1}$, που είναι θετικός πραγματικός αριθμός.

Ο $\mathbf{\Gamma}$ είναι μοναδιαίος, ως γινόμενο μοναδιαίων πινάκων. Σε έναν μοναδιαίο πίνακα κάθε στήλη, αλλά και κάθε γραμμή, έχει μέτρο 1, δηλαδή το άθροισμα των τετραγώνων των μέτρων των στοιχείων της ισούται με 1, όπως δείχνουμε στο παρακάτω παράδειγμα.

Παράδειγμα 4.21: Οι στήλες και οι γραμμές ενός μοναδιαίου πίνακα έχουν μέτρο 1

Έστω $\mathbf{U}$ ένας $N \times N$ μοναδιαίος πίνακας με στοιχεία u_{pq} . Ας δείξουμε πρώτα ότι κάθε στήλη του έχει μέτρο 1. Όπως είδαμε στο παράδειγμα 4.19, το στοιχείο στη γραμμή p και στη στήλη q του γινομένου $\mathbf{U}^T \mathbf{U}$ είναι το εσωτερικό γινόμενο της στήλης p με τη στήλη q του $\mathbf{U}$:

$$(\mathbf{U}^T \mathbf{U})_{pq} = \sum_{k=1}^N u_{kp}^* u_{kq}$$

Για $p = q$ παίρνουμε το εσωτερικό γινόμενο της στήλης q με τον εαυτό της. Αφού $u_{kq}^* u_{kq} = |u_{kq}|^2$, αυτό είναι:

$$(\mathbf{U}^T \mathbf{U})_{qq} = |u_{1q}|^2 + |u_{2q}|^2 + \cdots + |u_{Nq}|^2$$

Επειδή ο $\mathbf{U}$ είναι μοναδιαίος, $\mathbf{U}^T \mathbf{U} = \mathbf{I}_N$, και τα διαγώνια στοιχεία του $\mathbf{I}_N$ είναι όλα 1. Επομένως, για κάθε στήλη q :

$$|u_{1q}|^2 + |u_{2q}|^2 + \cdots + |u_{Nq}|^2 = 1$$

Για τις γραμμές χρησιμοποιούμε το γινόμενο με την αντίστροφη σειρά, $\mathbf{U} \mathbf{U}^T$. Αφού $\mathbf{U}^T \mathbf{U} = \mathbf{I}_N$, ο $\mathbf{U}^T$ είναι ο αντίστροφος του $\mathbf{U}$. Για τετραγωνικούς πίνακες ο αντίστροφος δίνει τον ταυτοτικό πίνακα από όποια πλευρά κι αν πολλαπλασιάσουμε, οπότε ισχύει επίσης:

$$\mathbf{U} \mathbf{U}^T = \mathbf{I}_N$$

Το στοιχείο στη γραμμή p και στη στήλη q του $\mathbf{U} \mathbf{U}^T$ προκύπτει πολλαπλασιάζοντας τη γραμμή p του $\mathbf{U}$ με τη στήλη q του $\mathbf{U}^T$, που είναι η γραμμή q του $\mathbf{U}$ με συζυγείς μιγαδικούς:

$$(\mathbf{U} \mathbf{U}^T)_{pq} = \sum_{k=1}^N u_{pk} u_{qk}^*$$

Για $p = q$ βρίσκουμε, όπως και πριν, ότι για κάθε γραμμή p :

$$|u_{p1}|^2 + |u_{p2}|^2 + \cdots + |u_{pN}|^2 = 1$$

Ας το δούμε σε έναν συγκεκριμένο πίνακα, τον 4×4 πίνακα

$$\mathcal{M} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

που θα αναλύσουμε σε λίγο σε διεπίπεδους πίνακες. Κάθε στοιχείο του είναι $\pm \frac{1}{2}$ ή $\pm \frac{j}{2}$, οπότε έχει μέτρο $\frac{1}{2}$ και τετράγωνο μέτρου $\frac{1}{4}$. Για τη δεύτερη στήλη, για παράδειγμα:

$$\left| \frac{1}{2} \right|^2 + \left| \frac{j}{2} \right|^2 + \left| -\frac{1}{2} \right|^2 + \left| -\frac{j}{2} \right|^2 = \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4} = 1$$

και με τον ίδιο τρόπο κάθε στήλη και κάθε γραμμή του $\mathbf{M}$ έχει μέτρο 1.

Ας εφαρμόσουμε τώρα το αποτέλεσμα στον πίνακα $\mathbf{\Gamma}$ του αλγορίθμου, που έχει τη μορφή:

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1N} \\ 0 & \gamma_{22} & \cdots & \gamma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \gamma_{N2} & \cdots & \gamma_{NN} \end{bmatrix}$$

Η πρώτη στήλη έχει μόνο ένα μη μηδενικό στοιχείο, οπότε η συνθήκη για τις στήλες δίνει:

$$|\gamma_{11}|^2 + 0 + \cdots + 0 = 1$$

δηλαδή $|\gamma_{11}| = 1$. Αφού το γ_{11} είναι θετικός πραγματικός αριθμός, συμπεραίνουμε ότι $\gamma_{11} = 1$. Η συνθήκη για την πρώτη γραμμή δίνει τότε:

$$1 + |\gamma_{12}|^2 + \cdots + |\gamma_{1N}|^2 = 1$$

δηλαδή:

$$|\gamma_{12}|^2 + \cdots + |\gamma_{1N}|^2 = 0$$

Ένα άθροισμα μη αρνητικών αριθμών είναι μηδέν μόνο αν όλοι οι αριθμοί είναι μηδέν. Επομένως $\gamma_{12} = \cdots = \gamma_{1N} = 0$, δηλαδή ολόκληρη η πρώτη γραμμή του $\mathbf{\Gamma}$, εκτός από το γ_{11} , είναι μηδέν.

Από την πρώτη στήλη του $\mathbf{\Gamma}$ παίρνουμε επομένως $|\gamma_{11}|^2 = 1$, και αφού το γ_{11} είναι θετικός πραγματικός αριθμός, $\gamma_{11} = 1$. Τότε όμως, για να έχει και η πρώτη γραμμή μέτρο 1, όλα τα υπόλοιπα στοιχεία της πρέπει να είναι μηδέν. Χωρίς να κάνουμε τίποτα επιπλέον, μηδενίστηκε λοιπόν ολόκληρη η πρώτη γραμμή, και ο $\mathbf{\Gamma}$ έχει τη μορφή:

$$\mathbf{\Gamma} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \mathbf{V} & \\ 0 & & & \end{bmatrix}$$

όπου ο $(N-1) \times (N-1)$ πίνακας $\mathbf{V}$ είναι κι αυτός μοναδιαίος. Δηλαδή ο $\mathbf{\Gamma}$ αφήνει την $|\psi_1\rangle$ αμετάβλητη.

Επαναλαμβάνουμε τώρα την ίδια διαδικασία στη δεύτερη στήλη, με γραμμή αναφοράς τη γραμμή 2, μηδενίζοντας τα στοιχεία των γραμμών 3, $\dots$, N . Οι πίνακες που χρησιμοποιούμε δεν αγγίζουν τη γραμμή 1, οπότε η πρώτη γραμμή και η πρώτη στήλη μένουν όπως ήταν. Μετά από $N-2$ βήματα, και η δεύτερη γραμμή και στήλη έχουν 1 στη διαγώνιο και μηδέν παντού αλλού. Συνεχίζουμε με τον ίδιο τρόπο στις επόμενες στήλες, μέχρι να μείνει μόνο το κάτω-δεξιά τμήμα 2×2 , στις γραμμές και στις στήλες $N-1$ και N . Τότε ο πίνακας είναι ταυτοτικός παντού εκτός από αυτό το τμήμα, δηλαδή είναι ήδη διεπίπεδος, και η διαδικασία σταματά.

Ας μετρήσουμε τους πίνακες που χρειαστήκαμε. Στην πρώτη στήλη χρειαστήκαμε το πολύ $N-1$ πίνακες $\mathbf{A}_i$, στη δεύτερη το πολύ $N-2$, και ούτω καθεξής μέχρι τη στήλη $N-2$, όπου χρειαστήκαμε το πολύ 2. Μαζί με τον τελικό διεπίπεδο πίνακα $\mathbf{\Gamma}$, το πλήθος των παραγόντων είναι το πολύ:

$$(N-1) + (N-2) + \cdots + 2 + 1 = \frac{N(N-1)}{2}$$

όπως δηλώνει το αποτέλεσμα 4.4. Λέμε «το πολύ», γιατί όταν κάποιο στοιχείο είναι ήδη μηδέν παραλείπουμε το αντίστοιχο βήμα.

Ας εφαρμόσουμε τώρα αναλυτικά τη διαδικασία σε έναν συγκεκριμένο πίνακα.

Παράδειγμα 4.22: Ανάλυση της πύλης QFT δύο qubit σε διεπίπεδους πίνακες

Θεωρούμε τον 4×4 μοναδιαίο πίνακα

$$\mathcal{M} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}$$

που δρα σε δύο qubit, στη βάση $|\psi_1\rangle, \dots, |\psi_4\rangle$. Θα τον ξανασυναντήσουμε αργότερα ως τον *κβαντικό μετασχηματισμό Fourier* (QFT) δύο qubit. Αφού $N = 4$, περιμένουμε το πολύ $4 \cdot 3/2 = 6$ διεπίπεδους παράγοντες.

Ξεκινάμε από την πρώτη στήλη, με γραμμή αναφοράς τη γραμμή 1. Στο πρώτο βήμα μηδενίζουμε το στοιχείο της γραμμής 2. Εδώ $a = 1/2$ και $b = 1/2$, οπότε:

$$\sqrt{|a|^2 + |b|^2} = \sqrt{\frac{1}{4} + \frac{1}{4}} = \frac{1}{\sqrt{2}}$$

$$x = y = \frac{1/2}{1/\sqrt{2}} = \frac{1}{\sqrt{2}}$$

Ο πίνακας $\mathbf{A}_1$ δρα στις γραμμές 1, 2:

$$\mathbf{A}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Η νέα γραμμή 1 είναι το άθροισμα των γραμμών 1 και 2 του $\mathcal{M}$ διαιρεμένο με $\sqrt{2}$, και η νέα γραμμή 2 είναι η διαφορά τους (γραμμή 2 μείον γραμμή 1) διαιρεμένη με $\sqrt{2}$. Για παράδειγμα, το δεύτερο στοιχείο της νέας γραμμής 1 είναι:

$$\frac{1}{\sqrt{2}} \left(\frac{1}{2} + \frac{j}{2} \right) = \frac{1+j}{2\sqrt{2}}$$

Οι γραμμές 3 και 4 μένουν ίδιες. Συνολικά:

$$\mathbf{\Gamma}_1 = \mathbf{A}_1 \mathcal{M} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1+j}{2\sqrt{2}} & 0 & \frac{1-j}{2\sqrt{2}} \\ 0 & -\frac{1-j}{2\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1+j}{2\sqrt{2}} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{j}{2} & -\frac{1}{2} & \frac{j}{2} \end{bmatrix}$$

Όπως περιμέναμε, το στοιχείο στη γραμμή 2 της πρώτης στήλης μηδενίστηκε, και το στοιχείο της γραμμής 1 έγινε $\sqrt{|a|^2 + |b|^2} = 1/\sqrt{2}$.

Στο δεύτερο βήμα μηδενίζουμε το στοιχείο της γραμμής 3 στην πρώτη στήλη του $\mathbf{\Gamma}_1$. Τώρα $a = 1/\sqrt{2}$ (η τρέχουσα τιμή της γραμμής 1) και $b = 1/2$, οπότε:

$$\sqrt{|a|^2 + |b|^2} = \sqrt{\frac{1}{2} + \frac{1}{4}} = \frac{\sqrt{3}}{2}$$

$$x = \frac{1/\sqrt{2}}{\sqrt{3}/2} = \frac{2}{\sqrt{6}} = \frac{\sqrt{6}}{3}$$

$$y = \frac{1/2}{\sqrt{3}/2} = \frac{1}{\sqrt{3}} = \frac{\sqrt{3}}{3}$$

Ο πίνακας A_2 δρα στις γραμμές 1, 3:

$$A_2 = \begin{bmatrix} \frac{\sqrt{6}}{3} & 0 & \frac{\sqrt{3}}{3} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{6}}{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

και αλλάζει μόνο τις γραμμές 1 και 3:

$$\Gamma_2 = A_2 \Gamma_1 = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{j\sqrt{3}}{6} & \frac{\sqrt{3}}{6} & -\frac{j\sqrt{3}}{6} \\ 0 & -\frac{1-j}{2\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1+j}{2\sqrt{2}} \\ 0 & -\frac{\sqrt{6}(3+j)}{12} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{6}(3-j)}{12} \\ \frac{1}{2} & -\frac{j}{2} & -\frac{1}{2} & \frac{j}{2} \end{bmatrix}$$

Στο τρίτο βήμα μηδενίζουμε το στοιχείο της γραμμής 4. Τώρα $a = \sqrt{3}/2$ και $b = 1/2$, οπότε:

$$\sqrt{|a|^2 + |b|^2} = \sqrt{\frac{3}{4} + \frac{1}{4}} = 1$$

δηλαδή $x = \sqrt{3}/2$ και $y = 1/2$. Ο πίνακας A_3 δρα στις γραμμές 1, 4:

$$A_3 = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 0 & \frac{\sqrt{3}}{2} \end{bmatrix}$$

Όπως προβλέπει η γενική θεωρία, η γραμμή 1 γίνεται $(1, 0, 0, 0)$, χωρίς να χρειαστεί να μηδενίσουμε τα υπόλοιπα στοιχεία της ένα-ένα:

$$\Gamma_3 = A_3 \Gamma_2 = \left[\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ 0 & -\frac{1-j}{2\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1+j}{2\sqrt{2}} \\ 0 & -\frac{\sqrt{6}(3+j)}{12} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{6}(3-j)}{12} \\ 0 & -\frac{j\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & \frac{j\sqrt{3}}{3} \end{array} \right]$$

Ας ελέγξουμε για παράδειγμα το δεύτερο στοιχείο της νέας γραμμής 1, που είναι x^* επί το αντίστοιχο στοιχείο της γραμμής 1 συν y^* επί το αντίστοιχο στοιχείο της γραμμής 4 του Γ_2 :

$$\frac{\sqrt{3}}{2} \cdot \frac{j\sqrt{3}}{6} + \frac{1}{2} \cdot \left(-\frac{j}{2}\right) = \frac{3j}{12} - \frac{j}{4} = 0$$

Περνάμε τώρα στη δεύτερη στήλη, με γραμμή αναφοράς τη γραμμή 2. Στο τέταρτο βήμα μηδενίζουμε το στοιχείο της γραμμής 3. Εδώ

$$a = -\frac{1-j}{2\sqrt{2}}$$

$$b = -\frac{\sqrt{6}(3+j)}{12}$$

με $|a|^2 = 2/8 = 1/4$ και $|b|^2 = 6 \cdot 10/144 = 5/12$. Επομένως:

$$\sqrt{|a|^2 + |b|^2} = \sqrt{\frac{1}{4} + \frac{5}{12}} = \sqrt{\frac{2}{3}} = \frac{\sqrt{6}}{3}$$

$$x = \frac{a}{\sqrt{6}/3} = -\frac{\sqrt{3}(1-j)}{4}$$

$$y = \frac{b}{\sqrt{6}/3} = -\frac{3+j}{4}$$

Ο πίνακας $\mathbf{A}_4$ δρα στις γραμμές 2, 3:

$$\mathbf{A}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{3}(1+j)}{4} & -\frac{3-j}{4} & 0 \\ 0 & \frac{3+j}{4} & -\frac{\sqrt{3}(1-j)}{4} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

και δίνει:

$$\mathbf{\Gamma}_4 = \mathbf{A}_4 \mathbf{\Gamma}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{6}}{3} & \frac{j\sqrt{6}}{6} & \frac{\sqrt{6}}{6} \\ 0 & 0 & -\frac{1}{\sqrt{2}} & -\frac{j}{\sqrt{2}} \\ 0 & -\frac{j\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & \frac{j\sqrt{3}}{3} \end{bmatrix}$$

Στο πέμπτο βήμα μηδενίζουμε το στοιχείο της γραμμής 4 στη δεύτερη στήλη. Τώρα $a = \sqrt{6}/3$ και $b = -j\sqrt{3}/3$, με $|a|^2 + |b|^2 = 2/3 + 1/3 = 1$, οπότε $x = \sqrt{6}/3$ και $y = -j\sqrt{3}/3$. Ο πίνακας $\mathbf{A}_5$ δρα στις γραμμές 2, 4:

$$\mathbf{A}_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{6}}{3} & 0 & \frac{j\sqrt{3}}{3} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{j\sqrt{3}}{3} & 0 & \frac{\sqrt{6}}{3} \end{bmatrix}$$

και, όπως και στην πρώτη στήλη, μηδενίζεται ολόκληρη η δεύτερη γραμμή:

$$\mathbf{\Gamma}_5 = \mathbf{A}_5 \mathbf{\Gamma}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{\sqrt{2}} & -\frac{j}{\sqrt{2}} \\ 0 & 0 & -\frac{1}{\sqrt{2}} & \frac{j}{\sqrt{2}} \end{bmatrix}$$

Ο $\mathbf{\Gamma}_5$ διαφέρει από τον ταυτοτικό μόνο στις γραμμές και στις στήλες 3, 4, δηλαδή είναι ήδη διεπίπεδος, και η διαδικασία σταματά εδώ.

Έχουμε λοιπόν

$$\mathbf{A}_5 \mathbf{A}_4 \mathbf{A}_3 \mathbf{A}_2 \mathbf{A}_1 \mathbf{M} = \mathbf{\Gamma}_5$$

και, πολλαπλασιάζοντας από τα αριστερά με τους συζυγείς ανάστροφους, όπως εξηγήσαμε πριν από το παράδειγμα:

$$\mathbf{M} = \mathbf{A}_1^T \mathbf{A}_2^T \mathbf{A}_3^T \mathbf{A}_4^T \mathbf{A}_5^T \mathbf{\Gamma}_5$$

Θέτουμε $\mathbf{U}_i = \mathbf{A}_i^T$ για $i = 1, \dots, 5$ και $\mathbf{U}_6 = \mathbf{\Gamma}_5$. Για κάθε $\mathbf{A}_i$, ο συζυγής ανάστροφος προκύπτει ανταλλάσσοντας γραμμές με στήλες και παίρνοντας τους συζυγείς μιγαδικούς. Βρίσκουμε:

$$\mathbf{U}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{U}_2 = \begin{bmatrix} \frac{\sqrt{6}}{3} & 0 & -\frac{\sqrt{3}}{3} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{3}}{3} & 0 & \frac{\sqrt{6}}{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned}
U_3 &= \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 & 0 & -\frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{\sqrt{3}}{2} \end{bmatrix} \\
U_4 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{\sqrt{3}(1-j)}{4} & \frac{3-j}{4} & 0 \\ 0 & -\frac{3+j}{4} & -\frac{\sqrt{3}(1+j)}{4} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
U_5 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{6}}{3} & 0 & -\frac{j\sqrt{3}}{3} \\ 0 & 0 & 1 & 0 \\ 0 & -\frac{j\sqrt{3}}{3} & 0 & \frac{\sqrt{6}}{3} \end{bmatrix} \\
U_6 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{\sqrt{2}} & -\frac{j}{\sqrt{2}} \\ 0 & 0 & -\frac{1}{\sqrt{2}} & \frac{j}{\sqrt{2}} \end{bmatrix}
\end{aligned}$$

Κάθε ένας από τους $U_1, \dots, U_6$ είναι διεπίπεδος, και το γινόμενό τους είναι ο $\mathcal{M}$:

$$\mathcal{M} = U_1 U_2 U_3 U_4 U_5 U_6$$

Χρησιμοποιήσαμε έξι παράγοντες, δηλαδή ακριβώς το μέγιστο πλήθος $N(N-1)/2 = 6$ που προβλέπει το αποτέλεσμα 4.4.

Ας επιβεβαιώσουμε αριθμητικά ότι κάθε ένας από τους $U_1, \dots, U_6$ είναι μοναδιαίος, ότι διαφέρει από τον I_4 σε το πολύ τέσσερα στοιχεία, και ότι το γινόμενό τους ισούται με τον $\mathcal{M}$. Στο script οι πίνακες είναι αντιγραφμένοι από το παράδειγμα. Το script βρίσκεται στο αρχείο `python/ex_two_level_QFT4.py`:

```
import numpy as np

np.set_printoptions(precision=4, suppress=True)

s2, s3, s6 = np.sqrt(2), np.sqrt(3), np.sqrt(6)
j = 1j

# Example 4.22: the 4x4 two-qubit QFT matrix M = 0.5 * j^((p-1)(q-1)), decomposed
# into six two-level unitary matrices U1, ..., U6
M = 0.5 * np.array([
    [1, 1, 1, 1],
    [1, j, -1, -j],
    [1, -1, 1, -1],
    [1, -j, -1, j],
], dtype=complex)

U1 = np.array([
    [1 / s2, -1 / s2, 0, 0],
    [1 / s2, 1 / s2, 0, 0],
    [0, 0, 1, 0],
    [0, 0, 0, 1],
], dtype=complex)

U2 = np.array([
    [s6 / 3, 0, -s3 / 3, 0],
    [0, 1, 0, 0],
```

```

    [s3 / 3, 0, s6 / 3, 0],
    [0, 0, 0, 1],
], dtype=complex)

U3 = np.array([
    [s3 / 2, 0, 0, -1 / 2],
    [0, 1, 0, 0],
    [0, 0, 1, 0],
    [1 / 2, 0, 0, s3 / 2],
], dtype=complex)

U4 = np.array([
    [1, 0, 0, 0],
    [0, -s3 * (1 - j) / 4, (3 - j) / 4, 0],
    [0, -(3 + j) / 4, -s3 * (1 + j) / 4, 0],
    [0, 0, 0, 1],
], dtype=complex)

U5 = np.array([
    [1, 0, 0, 0],
    [0, s6 / 3, 0, -j * s3 / 3],
    [0, 0, 1, 0],
    [0, -j * s3 / 3, 0, s6 / 3],
], dtype=complex)

U6 = np.array([
    [1, 0, 0, 0],
    [0, 1, 0, 0],
    [0, 0, -1 / s2, -j / s2],
    [0, 0, -1 / s2, j / s2],
], dtype=complex)

I4 = np.eye(4, dtype=complex)
for name, U in [("U1", U1), ("U2", U2), ("U3", U3),
               ("U4", U4), ("U5", U5), ("U6", U6)]:
    is_unitary = np.allclose(U.conj().T @ U, I4)
    entries_differing_from_I = int(np.sum(~np.isclose(U, I4)))
    print(f"{name}: unitary={is_unitary}, "
          f"entries differing from I4={entries_differing_from_I}")

product = U1 @ U2 @ U3 @ U4 @ U5 @ U6
print(f"U1 U2 U3 U4 U5 U6 =\n{product}")
print(f"matches M: {np.allclose(product, M)}")

```

Στο παραπάνω script οι πίνακες $U_1, \dots, U_6$ υπολογίστηκαν «με το χέρι». Ο αλγόριθμος όμως δουλεύει για *οποιοδήποτε* μοναδιαίο πίνακα, οπότε αξίζει να τον υλοποιήσουμε ως μέθοδο `two_level_decomposition()` της κλάσης `Gate` στην `qulib2.py`:

```

def two_level_decomposition(self, tol=1e-10):
    N = 2 ** self.n_qubits
    M = self.calc_U().copy()
    eliminations = []
    for ref in range(N - 2):
        for row in range(ref + 1, N):
            a, b = M[ref, ref], M[row, ref]
            if abs(b) < tol:
                continue
            norm = np.hypot(abs(a), abs(b))
            x, y = a / norm, b / norm
            A = np.eye(N, dtype=complex)
            A[np.ix_([ref, row], [ref, row])] = [[x.conjugate(), y.conjugate()], [-y, x]]

```

```

        M = A @ M
        eliminations.append(A)
    factors = [Gate(U=A.conj().T) for A in eliminations]
    factors.append(Gate(U=M))
    return factors

```

Ο κώδικας ακολουθεί βήμα-βήμα τον αλγόριθμο. Προσέξτε ότι η Python αριθμεί τις γραμμές και τις στήλες από το 0, οπότε η γραμμή 1 του κειμένου είναι η γραμμή 0 του κώδικα, και ούτω καθεξής.

- Ο εξωτερικός βρόχος διατρέχει τις γραμμές αναφοράς *ref*, δηλαδή τις στήλες που «καθαρίζουμε» μία-μία. Σταματά δύο στήλες πριν το τέλος, αφού το τελικό τμήμα 2×2 είναι ήδη διεπίπεδο.
- Ο εσωτερικός βρόχος διατρέχει τις γραμμές *row* κάτω από τη γραμμή αναφοράς. Τα *a* και *b* είναι τα στοιχεία των γραμμών *ref* και *row* στη στήλη *ref*. Αν το *b* είναι ήδη (σχεδόν) μηδέν, το βήμα παραλείπεται.
- Οι αριθμοί *x*, *y* υπολογίζονται όπως στο κείμενο, και ο πίνακας *A* κατασκευάζεται ξεκινώντας από τον ταυτοτικό και γράφοντας τον 2×2 πίνακα $\tilde{A}$ στις γραμμές και στις στήλες *ref* και *row*, με τη βοήθεια της `np.ix_`.
- Ο τρέχων πίνακας πολλαπλασιάζεται από τα αριστερά με τον *A*, και ο *A* αποθηκεύεται.
- Στο τέλος, η μέθοδος επιστρέφει τους συζυγείς ανάστροφους των αποθηκευμένων πινάκων, δηλαδή τους $U_i = A_i^T$, και τον τελικό διεπίπεδο πίνακα Γ , ως αντικείμενα *Gate*, ώστε το γινόμενό τους με τη σειρά που επιστρέφονται να είναι ο αρχικός πίνακας.

Ας επιβεβαιώσουμε ότι η `two_level_decomposition()`, εφαρμοσμένη στον ίδιο πίνακα *M*, βρίσκει αυτόματα ακριβώς τους ίδιους έξι πίνακες $U_1, \dots, U_6$ του παραδείγματος. Το script βρίσκεται στο αρχείο `python/ex_two_level_decomposition.py`:

```

import functools
import operator

import numpy as np

from qulib2 import Gate

np.set_printoptions(precision=4, suppress=True)

s2, s3, s6 = np.sqrt(2), np.sqrt(3), np.sqrt(6)
j = 1j

# The same 4x4 two-qubit QFT matrix as Example 4.22, but this time the
# two-level factors U1, ..., U6 are computed automatically by
# Gate.two_level_decomposition(), instead of being derived by hand.
M = 0.5 * np.array([
    [1, 1, 1, 1],
    [1, j, -1, -j],
    [1, -1, 1, -1],
    [1, -j, -1, j],
], dtype=complex)

factors = Gate(U=M).two_level_decomposition()
print(f"number of factors: {len(factors)}")

I4 = np.eye(4, dtype=complex)
for i, gate in enumerate(factors, start=1):
    is_unitary = np.allclose(gate.U.conj().T @ gate.U, I4)
    entries_differing_from_I = int(np.sum(~np.isclose(gate.U, I4)))
    print(f"U{i}: unitary={is_unitary}, "
          f"entries differing from I4={entries_differing_from_I}")

```

```

product = functools.reduce(operator.matmul, factors)
print(f"U1 U2 U3 U4 U5 U6 =\n{product.U}")
print(f"matches M: {np.allclose(product.U, M)}")

# the automatic decomposition matches the book's own hand-derived matrices
# from Example 4.22, exactly
book = [
    np.array([
        [1 / s2, -1 / s2, 0, 0],
        [1 / s2, 1 / s2, 0, 0],
        [0, 0, 1, 0],
        [0, 0, 0, 1],
    ], dtype=complex),
    np.array([
        [s6 / 3, 0, -s3 / 3, 0],
        [0, 1, 0, 0],
        [s3 / 3, 0, s6 / 3, 0],
        [0, 0, 0, 1],
    ], dtype=complex),
    np.array([
        [s3 / 2, 0, 0, -1 / 2],
        [0, 1, 0, 0],
        [0, 0, 1, 0],
        [1 / 2, 0, 0, s3 / 2],
    ], dtype=complex),
    np.array([
        [1, 0, 0, 0],
        [0, -s3 * (1 - j) / 4, (3 - j) / 4, 0],
        [0, -(3 + j) / 4, -s3 * (1 + j) / 4, 0],
        [0, 0, 0, 1],
    ], dtype=complex),
    np.array([
        [1, 0, 0, 0],
        [0, s6 / 3, 0, -j * s3 / 3],
        [0, 0, 1, 0],
        [0, -j * s3 / 3, 0, s6 / 3],
    ], dtype=complex),
    np.array([
        [1, 0, 0, 0],
        [0, 1, 0, 0],
        [0, 0, -1 / s2, -j / s2],
        [0, 0, -1 / s2, j / s2],
    ], dtype=complex),
]
for i, (gate, book_U) in enumerate(zip(factors, book), start=1):
    print(f"U{i} matches Example 4.22 exactly: {np.allclose(gate.U, book_U)}")

```

4.7 Κυκλώματα για τους διεπίπεδους πίνακες

Στην προηγούμενη ενότητα είδαμε ότι κάθε μοναδιαίος πίνακας U ενός συστήματος n qubit γράφεται ως γινόμενο διεπίπεδων πινάκων:

$$U = U_1 U_2 \cdots U_d$$

Αν ξέρουμε να κατασκευάσουμε ένα κύκλωμα για κάθε U_i , τότε παίρνουμε ένα κύκλωμα για τον U τοποθετώντας αυτά τα κυκλώματα το ένα μετά το άλλο. Προσέξτε τη σειρά: όταν ο U δρα σε μία κατάσταση, πρώτος δρα ο U_d , που βρίσκεται δεξιότερα στο γινόμενο, οπότε στο κύκλωμα ο U_d μπαίνει πρώτος, αριστερά, και ο U_1 τελευταίος. Μένει λοιπόν να δείξουμε πώς υλοποιείται ένας μόνο διεπίπεδος πίνακας με πύλες που ήδη γνωρίζουμε.

Έστω ένας διεπίπεδος πίνακας U που δρα μη τετριμμένα στις βασικές καταστάσεις $|\psi_r\rangle$ και $|\psi_s\rangle$, μέσω του 2×2 μοναδιαίου πίνακα

$$\tilde{U} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

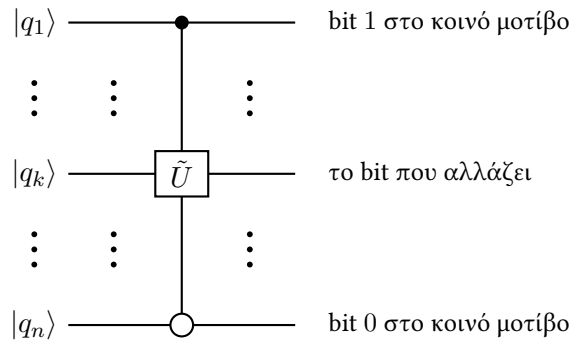
Δηλαδή:

$$U |\psi_r\rangle = a |\psi_r\rangle + c |\psi_s\rangle$$

$$U |\psi_s\rangle = b |\psi_r\rangle + d |\psi_s\rangle$$

ενώ κάθε άλλη βασική κατάσταση μένει αμετάβλητη. Θυμηθείτε ότι κάθε βασική κατάσταση αντιστοιχεί σε μία ακολουθία n bit. Για παράδειγμα, για $n = 3$ έχουμε $|\psi_1\rangle = |000\rangle$, $|\psi_2\rangle = |001\rangle$, και ούτω καθεξής μέχρι $|\psi_8\rangle = |111\rangle$. Το κύκλωμα που θα κατασκευάσουμε εξαρτάται από το σε πόσα bit διαφέρουν οι ακολουθίες των $|\psi_r\rangle$ και $|\psi_s\rangle$.

Ας ξεκινήσουμε από την απλή περίπτωση, όπου οι ακολουθίες bit των $|\psi_r\rangle$ και $|\psi_s\rangle$ διαφέρουν σε ένα μόνο bit, έστω στο bit του qubit q_k . Όλα τα υπόλοιπα $n-1$ bit είναι ίδια στις δύο καταστάσεις, και τα ονομάζουμε κοινό μοτίβο. Οι δύο καταστάσεις είναι τότε οι μόνες βασικές καταστάσεις στις οποίες τα υπόλοιπα qubit έχουν τις τιμές του κοινού μοτίβου. Η ιδέα είναι να εφαρμόσουμε τον $\tilde{U}$ ως πύλη ενός qubit στο q_k , ελεγχόμενη από όλα τα υπόλοιπα qubit, ώστε να ενεργοποιείται μόνο όταν αυτά έχουν τις τιμές του κοινού μοτίβου, όπως δείχνει το σχήμα 4.5. Για ένα qubit ελέγχου που πρέπει να είναι $|1\rangle$ χρησιμοποιούμε μαύρο κυκλάκι, και για ένα qubit ελέγχου που πρέπει να είναι $|0\rangle$ άσπρο κυκλάκι (ενότητα 4.1). Η πύλη αυτή είναι μία ελεγχόμενη πύλη $C_{n-1}^1(\tilde{U})$, που ξέρουμε να την κατασκευάζουμε (αποτέλεσμα 4.3).



Σχήμα 4.5: Το κύκλωμα ενός διεπίπεδου πίνακα, όταν οι $|\psi_r\rangle$ και $|\psi_s\rangle$ διαφέρουν μόνο στο bit του qubit q_k . Ο $\tilde{U}$ εφαρμόζεται στο q_k και ελέγχεται από όλα τα υπόλοιπα qubit: με μαύρο κυκλάκι όσα έχουν τιμή 1 στο κοινό μοτίβο (εδώ το q_1) και με άσπρο κυκλάκι όσα έχουν τιμή 0 (εδώ το q_n).

Πρέπει μόνο να προσέξουμε ένα σημείο. Η πρώτη γραμμή και στήλη του $\tilde{U}$ αντιστοιχούν στην $|\psi_r\rangle$ και η δεύτερη στην $|\psi_s\rangle$. Μία πύλη ενός qubit όμως έχει πρώτη γραμμή και στήλη την $|0\rangle$ και δεύτερη την $|1\rangle$. Αν το bit του q_k είναι 0 στην $|\psi_r\rangle$ και 1 στην $|\psi_s\rangle$, οι δύο αντιστοιχίσεις συμφωνούν και χρησιμοποιούμε τον $\tilde{U}$ όπως είναι. Αν είναι ανάποδα, πρέπει να ανταλλάξουμε τις δύο γραμμές και τις δύο στήλες του, δηλαδή να χρησιμοποιήσουμε τον πίνακα

$$\begin{bmatrix} d & c \\ b & a \end{bmatrix}$$

Ας δούμε την απλή περίπτωση σε ένα παράδειγμα.

Παράδειγμα 4.23: Διεπίπεδος πίνακας σε καταστάσεις που διαφέρουν σε ένα bit

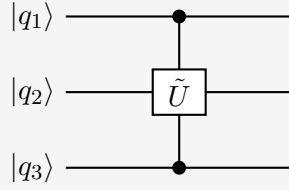
Θεωρούμε τρία qubit και έναν διεπίπεδο πίνακα U που δρα μη τετριμμένα στις καταστάσεις

$$|\psi_6\rangle = |101\rangle$$

$$|\psi_8\rangle = |111\rangle$$

μέσω του πίνακα $\tilde{U}$. Οι δύο ακολουθίες διαφέρουν μόνο στο δεύτερο bit, οπότε θα εφαρμόσουμε τον $\tilde{U}$ στο qubit q_2 . Το κοινό μοτίβο είναι $q_1 = 1$ και $q_3 = 1$, οπότε και τα δύο qubit ελέγχου έχουν μαύρο

κυκλάκι. Επίσης το bit του q_2 είναι 0 στην $|\psi_6\rangle$ και 1 στην $|\psi_8\rangle$, οπότε χρησιμοποιούμε τον $\tilde{U}$ όπως είναι. Το κύκλωμα είναι:



Ας ελέγξουμε ότι δουλεύει. Για την είσοδο $|\psi_6\rangle = |101\rangle$ τα q_1 και q_3 είναι 1, οπότε η πύλη ενεργοποιείται και εφαρμόζει τον $\tilde{U}$ στο q_2 , που είναι $|0\rangle$. Η πρώτη στήλη του $\tilde{U}$ δίνει:

$$\tilde{U} |0\rangle = a |0\rangle + c |1\rangle$$

οπότε η έξοδος είναι:

$$a |101\rangle + c |111\rangle = a |\psi_6\rangle + c |\psi_8\rangle$$

όπως ακριβώς πρέπει. Με τον ίδιο τρόπο, για την είσοδο $|\psi_8\rangle = |111\rangle$ η πύλη εφαρμόζει τον $\tilde{U}$ στην $|1\rangle$ και παίρνουμε $b |\psi_6\rangle + d |\psi_8\rangle$. Σε κάθε άλλη βασική κατάσταση ένα τουλάχιστον από τα q_1, q_3 είναι 0, οπότε η πύλη δεν ενεργοποιείται και η κατάσταση μένει αμετάβλητη.

Στο παραπάνω παράδειγμα το bit του qubit που αλλάζει ήταν 0 στην $|\psi_r\rangle$, οπότε χρησιμοποιήσαμε τον $\tilde{U}$ όπως είναι. Ας δούμε και την άλλη περίπτωση, όπου πρέπει να ανταλλάξουμε τις γραμμές και τις στήλες του.

Παράδειγμα 4.24: Διεπίπεδος πίνακας με ανταλλαγμένες γραμμές και στήλες

Θεωρούμε ξανά τρία qubit και τις ίδιες δύο καταστάσεις με το παράδειγμα 4.23, αλλά τώρα με τους ρόλους τους ανταλλαγμένους:

$$|\psi_r\rangle = |\psi_8\rangle = |111\rangle$$

$$|\psi_s\rangle = |\psi_6\rangle = |101\rangle$$

Ο διεπίπεδος πίνακας U δρα στις δύο καταστάσεις μέσω του πίνακα $\tilde{U}$, με την πρώτη γραμμή και στήλη να αντιστοιχούν στην $|\psi_r\rangle = |111\rangle$:

$$\hat{U} |111\rangle = a |111\rangle + c |101\rangle$$

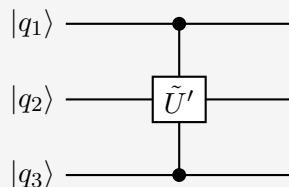
$$\hat{U} |101\rangle = b |111\rangle + d |101\rangle$$

Μία τέτοια περίπτωση συναντάμε, για παράδειγμα, στη μεσαία πύλη της γενικής κατασκευής που θα δούμε παρακάτω, όπου η κατάσταση που έχει πάρει τη θέση της $|\psi_r\rangle$ μπορεί να έχει bit 1 στο qubit που αλλάζει.

Οι δύο ακολουθίες διαφέρουν πάλι μόνο στο δεύτερο bit, και το κοινό μοτίβο είναι πάλι $q_1 = 1$ και $q_3 = 1$. Η πύλη έχει επομένως την ίδια μορφή με το παράδειγμα 4.23, με δύο μαύρα κυκλάκια. Τώρα όμως το bit του q_2 είναι 1 στην $|\psi_r\rangle$ και 0 στην $|\psi_s\rangle$. Μία πύλη ενός qubit έχει πρώτη στήλη την $|0\rangle$, που εδώ αντιστοιχεί στην $|\psi_s\rangle$, και δεύτερη στήλη την $|1\rangle$, που αντιστοιχεί στην $|\psi_r\rangle$. Πρέπει λοιπόν να ανταλλάξουμε τις γραμμές και τις στήλες του $\tilde{U}$, δηλαδή να χρησιμοποιήσουμε τον πίνακα:

$$\tilde{U}' = \begin{bmatrix} d & c \\ b & a \end{bmatrix}$$

Το κύκλωμα είναι:



Ας ελέγξουμε ότι δουλεύει. Για την είσοδο $|111\rangle$ η πύλη ενεργοποιείται και εφαρμόζει τον $\tilde{U}'$ στο q_2 , που είναι $|1\rangle$. Η δεύτερη στήλη του $\tilde{U}'$ δίνει:

$$\tilde{U}'|1\rangle = c|0\rangle + a|1\rangle$$

οπότε η έξοδος είναι:

$$c|101\rangle + a|111\rangle = a|\psi_r\rangle + c|\psi_s\rangle$$

όπως πρέπει. Για την είσοδο $|101\rangle$ το q_2 είναι $|0\rangle$, και η πρώτη στήλη του $\tilde{U}'$ δίνει:

$$\tilde{U}'|0\rangle = d|0\rangle + b|1\rangle$$

οπότε η έξοδος είναι:

$$d|101\rangle + b|111\rangle = b|\psi_r\rangle + d|\psi_s\rangle$$

όπως πρέπει. Αν είχαμε χρησιμοποιήσει τον $\tilde{U}$ χωρίς ανταλλαγή, για την είσοδο $|111\rangle$ θα παίρναμε $b|101\rangle + d|111\rangle$, που είναι λάθος. Όπως και πριν, σε κάθε άλλη βασική κατάσταση η πύλη δεν ενεργοποιείται.

Τι γίνεται όμως όταν οι ακολουθίες bit των $|\psi_r\rangle$ και $|\psi_s\rangle$ διαφέρουν σε περισσότερα από ένα bit; Τότε χρησιμοποιούμε ένα τέχνασμα: «μετακινούμε» πρώτα την $|\psi_r\rangle$, βήμα-βήμα, σε μία κατάσταση που διαφέρει από την $|\psi_s\rangle$ σε ένα μόνο bit, εφαρμόζουμε εκεί την πύλη της απλής περίπτωσης, και μετά την επιστρέφουμε στη θέση της.

Για τη μετακίνηση χρησιμοποιούμε έναν κώδικα Gray. Ένας κώδικας Gray είναι μία λίστα από ακολουθίες bit, στην οποία κάθε δύο διαδοχικές ακολουθίες διαφέρουν σε ακριβώς ένα bit. Για παράδειγμα, η λίστα

$$00, 01, 11, 10$$

είναι ένας κώδικας Gray δύο bit: από το 00 στο 01 αλλάζει μόνο το δεύτερο bit, από το 01 στο 11 μόνο το πρώτο, και από το 11 στο 10 μόνο το δεύτερο. Αντίθετα, η συνηθισμένη σειρά των δυαδικών αριθμών 00, 01, 10, 11 δεν είναι κώδικας Gray, αφού από το 01 στο 10 αλλάζουν και τα δύο bit.

Στην περίπτωση μας θέλουμε έναν κώδικα Gray που ξεκινά από την ακολουθία bit της $|\psi_r\rangle$ και καταλήγει σε αυτή της $|\psi_s\rangle$. Τον γράφουμε ως

$$g_0, g_1, \dots, g_m$$

όπου g_0 είναι η ακολουθία bit της $|\psi_r\rangle$ και g_m η ακολουθία bit της $|\psi_s\rangle$. Ο απλούστερος τρόπος να τον κατασκευάσουμε είναι ο εξής. Βρίσκουμε πρώτα τις θέσεις στις οποίες διαφέρουν οι δύο ακολουθίες. Ξεκινάμε από το g_0 και, σε κάθε βήμα, αλλάζουμε ένα από αυτά τα bit, ώστε να πάρει την τιμή που έχει στην $|\psi_s\rangle$. Κάθε νέα ακολουθία διαφέρει από την προηγούμενη σε ένα μόνο bit, και μετά από όλα τα βήματα φτάνουμε στο g_m .

Ας δούμε ένα παράδειγμα με τέσσερα qubit, όπου οι δύο καταστάσεις είναι:

$$|\psi_r\rangle = |0010\rangle$$

$$|\psi_s\rangle = |1001\rangle$$

Συγκρίνοντας τις δύο ακολουθίες bit μία-μία θέση, βλέπουμε ότι διαφέρουν στα bit των q_1 , q_3 και q_4 , ενώ το bit του q_2 είναι 0 και στις δύο. Αλλάζοντας αυτά τα τρία bit με τη σειρά, παίρνουμε:

$$g_0 = 0010$$

$$g_1 = 1010$$

$$g_2 = 1000$$

$$g_3 = 1001$$

Από το g_0 στο g_1 αλλάζει το bit του q_1 , από το g_1 στο g_2 το bit του q_3 , και από το g_2 στο g_3 το bit του q_4 .

Το πλήθος m των βημάτων είναι ίσο με το πλήθος των θέσεων στις οποίες διαφέρουν οι δύο ακολουθίες, αφού σε κάθε βήμα «διορθώνουμε» ακριβώς μία από αυτές. Το πλήθος αυτό λέγεται *απόσταση Hamming* των δύο ακολουθιών. Στο παράδειγμα η απόσταση Hamming είναι $m = 3$. Τα bit μπορούμε να τα αλλάξουμε με όποια σειρά θέλουμε: κάθε σειρά δίνει έναν διαφορετικό κώδικα Gray, και όλοι είναι εξίσου κατάλληλοι για την κατασκευή μας.

Κάθε g_k είναι μία ακολουθία n bit, οπότε η $|g_k\rangle$ είναι μία από τις βασικές καταστάσεις $|\psi_p\rangle$. Ειδικότερα:

$$|g_0\rangle = |\psi_r\rangle$$

$$|g_m\rangle = |\psi_s\rangle$$

Στο παράδειγμα των τεσσάρων qubit, οι ενδιαμέσες καταστάσεις είναι οι $|g_1\rangle = |1010\rangle$ και $|g_2\rangle = |1000\rangle$. Για κάθε $k = 1, \dots, m-1$ ορίζουμε την πύλη F_k , που ανταλλάσσει τις καταστάσεις $|g_{k-1}\rangle$ και $|g_k\rangle$ και αφήνει όλες τις άλλες βασικές καταστάσεις αμετάβλητες:

$$\hat{F}_k |g_{k-1}\rangle = |g_k\rangle$$

$$\hat{F}_k |g_k\rangle = |g_{k-1}\rangle$$

Η F_k περιγράφεται από έναν διεπίπεδο πίνακα με $\tilde{U} = \mathbf{X}$, σε δύο καταστάσεις που διαφέρουν σε ένα μόνο bit. Επομένως την κατασκευάζουμε όπως στην απλή περίπτωση: είναι μία πύλη X στο qubit που αλλάζει από το g_{k-1} στο g_k , ελεγχόμενη από όλα τα υπόλοιπα qubit, με μαύρα και άσπρα κυκλάκια σύμφωνα με το κοινό μοτίβο τους. Εδώ δεν χρειάζεται να προσέξουμε τη σειρά των γραμμών και των στηλών, αφού ο πίνακας $\mathbf{X}$ μένει ίδιος αν ανταλλάξουμε τις γραμμές και τις στήλες του. Αφού η πύλη στο qubit-στόχο είναι μία X , τη σχεδιάζουμε με το σύμβολο $\oplus$, όπως τη CNOT.

Ας δούμε τις πύλες F_1 και F_2 για το παράδειγμα των τεσσάρων qubit, που φαίνονται στο σχήμα 4.6. Η F_1 ανταλλάσσει τις καταστάσεις

$$|g_0\rangle = |0010\rangle$$

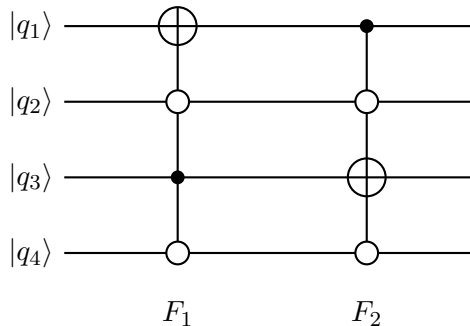
$$|g_1\rangle = |1010\rangle$$

που διαφέρουν στο bit του q_1 . Είναι επομένως μία X στο q_1 . Το κοινό μοτίβο των υπόλοιπων qubit είναι $q_2 = 0, q_3 = 1$ και $q_4 = 0$, οπότε το q_3 έχει μαύρο κυκλάκι, ενώ τα q_2 και q_4 άσπρο. Η F_2 ανταλλάσσει τις καταστάσεις

$$|g_1\rangle = |1010\rangle$$

$$|g_2\rangle = |1000\rangle$$

που διαφέρουν στο bit του q_3 . Είναι επομένως μία X στο q_3 . Το κοινό μοτίβο των υπόλοιπων qubit είναι $q_1 = 1, q_2 = 0$ και $q_4 = 0$, οπότε το q_1 έχει μαύρο κυκλάκι, ενώ τα q_2 και q_4 άσπρο.



Σχήμα 4.6: Οι πύλες F_1 και F_2 για τον κώδικα Gray $0010 \rightarrow 1010 \rightarrow 1000 \rightarrow 1001$. Η F_1 αλλάζει το q_1 και ενεργοποιείται μόνο όταν $q_2q_3q_4 = 010$. Η F_2 αλλάζει το q_3 και ενεργοποιείται μόνο όταν $q_1q_2q_4 = 100$. Μαύρο κυκλάκι σημαίνει ότι το qubit ελέγχου πρέπει να είναι $|1\rangle$ και άσπρο ότι πρέπει να είναι $|0\rangle$.

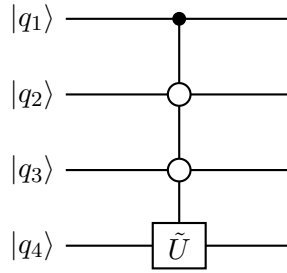
Για παράδειγμα, όταν η F_1 δρα στην $|0010\rangle$, τα q_2, q_3, q_4 έχουν τις τιμές 0, 1, 0 του κοινού μοτίβου, οπότε η πύλη ενεργοποιείται και αλλάζει το q_1 , δίνοντας $|1010\rangle$. Σε μία κατάσταση όπως η $|0011\rangle$, αντίθετα, το

q_4 είναι 1 αντί για 0, οπότε η F_1 δεν ενεργοποιείται και η κατάσταση μένει ίδια. Αφού ανταλλάσσει απλώς δύο καταστάσεις, αν την εφαρμόσουμε δύο φορές επιστρέφουμε εκεί που ξεκινήσαμε, δηλαδή η F_k είναι αντίστροφη του εαυτού της.

Μένει να ορίσουμε την πύλη που θα «αναμιξεί» τις δύο καταστάσεις. Μετά τις $F_1, \dots, F_{m-1}$, η $|g_0\rangle = |\psi_r\rangle$ έχει μεταφερθεί στην $|g_{m-1}\rangle$, που διαφέρει από την $|g_m\rangle = |\psi_s\rangle$ σε ένα μόνο bit. Ορίζουμε την πύλη W ως τον διεπίπεδο πίνακα που δρα στις $|g_{m-1}\rangle$ και $|g_m\rangle$ με τον ίδιο 2×2 πίνακα $\tilde{U}$:

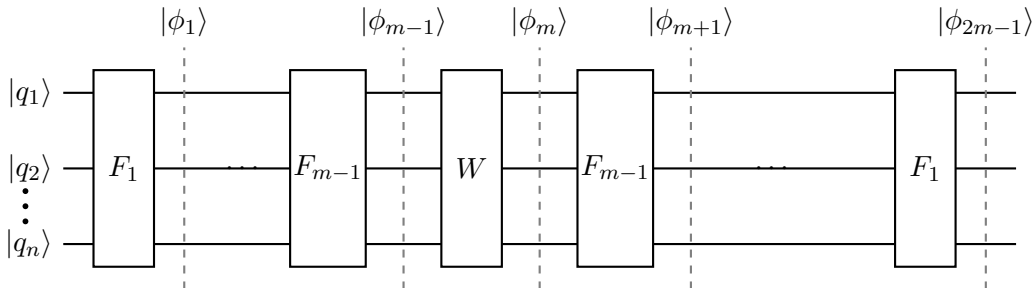
$$\begin{aligned}\hat{W} |g_{m-1}\rangle &= a |g_{m-1}\rangle + c |g_m\rangle \\ \hat{W} |g_m\rangle &= b |g_{m-1}\rangle + d |g_m\rangle\end{aligned}$$

και αφήνει κάθε άλλη βασική κατάσταση αμετάβλητη. Δηλαδή η W κάνει στις $|g_{m-1}\rangle$ και $|g_m\rangle$ ακριβώς ό,τι θέλουμε να κάνει ο U στις $|\psi_r\rangle$ και $|\psi_s\rangle$. Αφού οι $|g_{m-1}\rangle$ και $|g_m\rangle$ διαφέρουν σε ένα μόνο bit, η W κατασκευάζεται όπως στην απλή περίπτωση που είδαμε στο σχήμα 4.5. Είναι μία πύλη ενός qubit στο qubit όπου διαφέρουν τα g_{m-1} και g_m , ελεγχόμενη από όλα τα υπόλοιπα qubit σύμφωνα με το κοινό τους μοτίβο. Η πύλη αυτή είναι ο $\tilde{U}$ αν το bit του qubit-στόχου είναι 0 στο g_{m-1} , και ο $\tilde{U}'$ με τις ανταλλαγμένες γραμμές και στήλες αν είναι 1 όπως είδαμε και στο παράδειγμα 4.24. Στην περίπτωση που εξετάζουμε τώρα των τεσσάρων qubit, η W δρα στις $|g_2\rangle = |1000\rangle$ και $|g_3\rangle = |1001\rangle$, που διαφέρουν στο bit του q_4 . Είναι επομένως ο $\tilde{U}$ στο q_4 , με μαύρο κυκλάκι στο q_1 και άσπρα στα q_2, q_3 , όπως φαίνεται στο σχήμα 4.7. Το bit του q_4 είναι 0 στο g_2 , οπότε χρησιμοποιούμε τον $\tilde{U}$ όπως είναι.



Σχήμα 4.7: Η πύλη W για τον κώδικα Gray $0010 \rightarrow 1010 \rightarrow 1000 \rightarrow 1001$. Δρα στις $|g_2\rangle = |1000\rangle$ και $|g_3\rangle = |1001\rangle$, που διαφέρουν στο bit του q_4 . Ο $\tilde{U}$ εφαρμόζεται στο q_4 μόνο όταν $q_1 q_2 q_3 = 100$.

Το πλήρες κύκλωμα για τον U αποτελείται από τρία κομμάτια, όπως φαίνεται στο σχήμα 4.8. Πρώτα εφαρμόζουμε τις $F_1, F_2, \dots, F_{m-1}$, που μεταφέρουν την $|\psi_r\rangle$ βήμα-βήμα στην $|g_{m-1}\rangle$. Έπειτα εφαρμόζουμε την πύλη W . Τέλος εφαρμόζουμε τις $F_{m-1}, \dots, F_2, F_1$, με την αντίστροφη σειρά, που επιστρέφουν την $|g_{m-1}\rangle$ στην $|g_0\rangle = |\psi_r\rangle$. Συνολικά το κύκλωμα έχει $2m - 1$ στάδια, και σημειώνουμε με $|\phi_1\rangle, \dots, |\phi_{2m-1}\rangle$ την κατάσταση μετά από καθεμία από αυτά.



Σχήμα 4.8: Το κύκλωμα ενός διεπίπεδου πίνακα, όταν οι $|\psi_r\rangle$ και $|\psi_s\rangle$ διαφέρουν σε m bit. Οι πύλες $F_1, \dots, F_{m-1}$ μεταφέρουν την $|\psi_r\rangle$ στην $|g_{m-1}\rangle$, η W αναμιγνύει τις $|g_{m-1}\rangle$ και $|\psi_s\rangle$ σύμφωνα με τον $\tilde{U}$, και οι $F_{m-1}, \dots, F_1$ επιστρέφουν την $|g_{m-1}\rangle$ στην $|\psi_r\rangle$. Με $|\phi_k\rangle$ σημειώνεται η κατάσταση μετά από κάθε πύλη.

Ας δούμε γιατί το κύκλωμα υλοποιεί τον U . Θεωρούμε μία οποιαδήποτε κατάσταση στην είσοδο και γράφουμε χωριστά τα δύο κομμάτια της που μας ενδιαφέρουν:

$$\alpha |g_0\rangle + \beta |g_m\rangle + (\text{όροι με τις υπόλοιπες βασικές καταστάσεις})$$

Η F_1 στέλνει την $|g_0\rangle$ στην $|g_1\rangle$:

$$|\phi_1\rangle = \alpha |g_1\rangle + \beta |g_m\rangle + (\dots)$$

Η F_2 στέλνει την $|g_1\rangle$ στην $|g_2\rangle$:

$$|\phi_2\rangle = \alpha |g_2\rangle + \beta |g_m\rangle + (\dots)$$

Γενικά, μετά την F_k το πλάτος α βρίσκεται στην $|g_k\rangle$:

$$|\phi_k\rangle = \alpha |g_k\rangle + \beta |g_m\rangle + (\dots)$$

για $k = 1, \dots, m-1$. Η $|g_m\rangle$ δεν αλλάζει, αφού καμία από τις $F_1, \dots, F_{m-1}$ δεν τη χρησιμοποιεί. Μετά την F_{m-1} έχουμε λοιπόν:

$$|\phi_{m-1}\rangle = \alpha |g_{m-1}\rangle + \beta |g_m\rangle + (\dots)$$

Η W αναμιγνύει τις $|g_{m-1}\rangle$ και $|g_m\rangle$ σύμφωνα με τον $\tilde{U}$. Χρησιμοποιώντας τη δράση της στις δύο καταστάσεις:

$$\alpha \hat{W} |g_{m-1}\rangle + \beta \hat{W} |g_m\rangle = \alpha (a |g_{m-1}\rangle + c |g_m\rangle) + \beta (b |g_{m-1}\rangle + d |g_m\rangle)$$

και επομένως:

$$|\phi_m\rangle = (a\alpha + b\beta) |g_{m-1}\rangle + (c\alpha + d\beta) |g_m\rangle + (\dots)$$

Στη συνέχεια η δεύτερη F_{m-1} στέλνει την $|g_{m-1}\rangle$ πίσω στην $|g_{m-2}\rangle$:

$$|\phi_{m+1}\rangle = (a\alpha + b\beta) |g_{m-2}\rangle + (c\alpha + d\beta) |g_m\rangle + (\dots)$$

και κάθε επόμενη πύλη τη μεταφέρει ένα βήμα πιο πίσω, μέχρι η τελευταία F_1 να τη φέρει στην $|g_0\rangle$:

$$|\phi_{2m-1}\rangle = (a\alpha + b\beta) |g_0\rangle + (c\alpha + d\beta) |g_m\rangle + (\dots)$$

Αφού $|g_0\rangle = |\psi_r\rangle$ και $|g_m\rangle = |\psi_s\rangle$, αυτή είναι ακριβώς η δράση του U στις δύο καταστάσεις.

Μένει να δούμε τι γίνεται με τις υπόλοιπες βασικές καταστάσεις. Κάθε F_k απλώς ανταλλάσσει δύο βασικές καταστάσεις, οπότε οι $F_1, \dots, F_{m-1}$ μαζί απλώς «ανακατεύουν» τις βασικές καταστάσεις: στέλνουν κάθε βασική κατάσταση σε μία βασική κατάσταση, και δύο διαφορετικές καταστάσεις πάντα σε διαφορετικές. Αφού η $|g_0\rangle$ πήγε στην $|g_{m-1}\rangle$ και η $|g_m\rangle$ έμεινε στη θέση της, καμία άλλη βασική κατάσταση δεν μπορεί να καταλήξει σε αυτές τις δύο. Επομένως η W δεν τις επηρεάζει, και οι $F_{m-1}, \dots, F_1$, που κάνουν τα ίδια βήματα με την αντίστροφη σειρά, τις επιστρέφουν ακριβώς εκεί που ήταν. Το κύκλωμα αφήνει λοιπόν αμετάβλητη κάθε άλλη βασική κατάσταση, όπως πρέπει.

Το κύκλωμα χρησιμοποιεί συνολικά $2(m-1) + 1 = 2m - 1$ ελεγχόμενες πύλες, καθεμία με $n - 1$ qubit ελέγχου. Κάθε τέτοια πύλη κατασκευάζεται, όπως είδαμε στο αποτέλεσμα 4.3, από CNOT και πύλες ενός qubit, χωρίς να χρειαστούν επιπλέον qubit. Ας δούμε τώρα την κατασκευή σε δύο παραδείγματα.

Παράδειγμα 4.25: Ένας διεπίπεδος πίνακας στις καταστάσεις $|000\rangle$ και $|111\rangle$

Θεωρούμε τρία qubit και έναν διεπίπεδο πίνακα U που δρα μη τετριμμένα στις καταστάσεις

$$|\psi_1\rangle = |000\rangle$$

$$|\psi_8\rangle = |111\rangle$$

μέσω του πίνακα $\tilde{U}$. Οι δύο ακολουθίες διαφέρουν και στα τρία bit, οπότε η απόσταση Hamming είναι $m = 3$. Διαλέγουμε τον κώδικα Gray που αλλάζει πρώτα το τρίτο bit, μετά το δεύτερο, και τέλος το πρώτο:

$$g_0 = 000$$

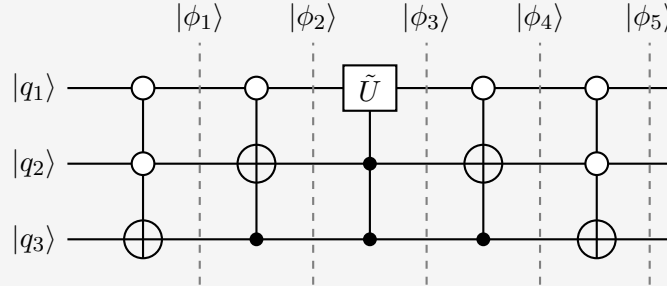
$$g_1 = 001$$

$$g_2 = 011$$

$$g_3 = 111$$

Χρειαζόμαστε δύο πύλες F_1, F_2 . Η F_1 ανταλλάσσει τις $|000\rangle$ και $|001\rangle$: είναι μία X στο q_3 , που ενεργοποιείται όταν $q_1 = 0$ και $q_2 = 0$, οπότε έχει δύο άσπρα κυκλάκια. Η F_2 ανταλλάσσει τις $|001\rangle$ και $|011\rangle$:

είναι μία X στο q_2 , που ενεργοποιείται όταν $q_1 = 0$ και $q_3 = 1$, οπότε έχει άσπρο κυκλάκι στο q_1 και μαύρο στο q_3 . Η μεσαία πύλη W δρα στις $|g_2\rangle = |011\rangle$ και $|g_3\rangle = |111\rangle$, που διαφέρουν στο πρώτο bit. Είναι επομένως η $\tilde{U}$ στο q_1 , που ενεργοποιείται όταν $q_2 = 1$ και $q_3 = 1$, με δύο μαύρα κυκλάκια. Το bit του q_1 είναι 0 στην $|g_2\rangle$, που έχει πάρει τη θέση της $|\psi_1\rangle$, οπότε χρησιμοποιούμε τον $\tilde{U}$ όπως είναι. Το πλήρες κύκλωμα είναι:



Οι πύλες εμφανίζονται με τη σειρά F_1, F_2, W, F_2, F_1 . Ας παρακολουθήσουμε μία οποιαδήποτε κατάσταση εισόδου

$$a_1 |\psi_1\rangle + a_2 |\psi_2\rangle + \dots + a_8 |\psi_8\rangle$$

Ο παρακάτω πίνακας δείχνει, για κάθε βασική κατάσταση, ποιο πλάτος έχει σε κάθε στάδιο του κυκλώματος:

	είσοδος	$ \phi_1\rangle$	$ \phi_2\rangle$	$ \phi_3\rangle$	$ \phi_4\rangle$	$ \phi_5\rangle$
$ \psi_1\rangle = 000\rangle$	a_1	a_2	a_2	a_2	a_2	b_1
$ \psi_2\rangle = 001\rangle$	a_2	a_1	a_4	a_4	b_1	a_2
$ \psi_3\rangle = 010\rangle$	a_3	a_3	a_3	a_3	a_3	a_3
$ \psi_4\rangle = 011\rangle$	a_4	a_4	a_1	b_1	a_4	a_4
$ \psi_5\rangle = 100\rangle$	a_5	a_5	a_5	a_5	a_5	a_5
$ \psi_6\rangle = 101\rangle$	a_6	a_6	a_6	a_6	a_6	a_6
$ \psi_7\rangle = 110\rangle$	a_7	a_7	a_7	a_7	a_7	a_7
$ \psi_8\rangle = 111\rangle$	a_8	a_8	a_8	b_8	b_8	b_8

όπου b_1 και b_8 είναι τα πλάτη που δίνει ο $\tilde{U}$:

$$b_1 = a a_1 + b a_8$$

$$b_8 = c a_1 + d a_8$$

Ας εξηγήσουμε τις στήλες του πίνακα. Η F_1 ανταλλάσσει τα πλάτη των $|000\rangle$ και $|001\rangle$, οπότε στην $|\phi_1\rangle$ το πλάτος a_1 βρίσκεται πλέον στην $|001\rangle$. Η F_2 ανταλλάσσει τα πλάτη των $|001\rangle$ και $|011\rangle$, οπότε στην $|\phi_2\rangle$ το a_1 έχει φτάσει στην $|011\rangle$, που διαφέρει από την $|111\rangle$ σε ένα μόνο bit. Η W αναμιγνύει τα πλάτη των $|011\rangle$ και $|111\rangle$ σύμφωνα με τον $\tilde{U}$, δίνοντας τα b_1 και b_8 . Τέλος, η δεύτερη F_2 και η δεύτερη F_1 φέρνουν το b_1 πίσω στην $|000\rangle$, και ταυτόχρονα επιστρέφουν τα a_2 και a_4 στις αρχικές τους θέσεις. Οι καταστάσεις $|010\rangle, |100\rangle, |101\rangle$ και $|110\rangle$ δεν ικανοποιούν τη συνθήκη ενεργοποίησης καμίας πύλης, οπότε τα πλάτη τους δεν αλλάζουν ποτέ.

Στην έξοδο αλλάζουν επομένως μόνο τα πλάτη των $|\psi_1\rangle$ και $|\psi_8\rangle$, ακριβώς όπως ορίζει ο U . Οι καταστάσεις $|001\rangle$ και $|011\rangle$ χρησιμοποιήθηκαν προσωρινά για να «μεταφέρουν» το πλάτος a_1 δίπλα στο a_8 , αλλά στο τέλος τα πλάτη τους επέστρεψαν εκεί που ήταν.

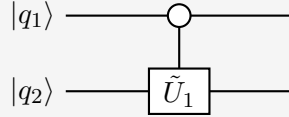
Παράδειγμα 4.26: Κυκλώματα για δύο από τους παράγοντες της QFT δύο qubit

Ας κατασκευάσουμε κυκλώματα για δύο από τους διεπίπεδους πίνακες που βρήκαμε στο παράδειγμα 4.22, τους U_1 και U_3 . Εδώ έχουμε δύο qubit, με βασικές καταστάσεις $|\psi_1\rangle = |00\rangle, |\psi_2\rangle = |01\rangle, |\psi_3\rangle = |10\rangle$ και $|\psi_4\rangle = |11\rangle$.

Ο U_1 δρα μη τετριμμένα στις $|\psi_1\rangle = |00\rangle$ και $|\psi_2\rangle = |01\rangle$, μέσω του πίνακα

$$\tilde{U}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

Οι δύο ακολουθίες διαφέρουν μόνο στο δεύτερο bit, οπότε είμαστε στην απλή περίπτωση. Εφαρμόζουμε τον $\tilde{U}_1$ στο q_2 , ελεγχόμενο από το q_1 . Το κοινό μοτίβο είναι $q_1 = 0$, οπότε το κυκλάκι ελέγχου είναι άσπρο. Επίσης το bit του q_2 είναι 0 στην $|\psi_1\rangle$, οπότε χρησιμοποιούμε τον $\tilde{U}_1$ όπως είναι:



Ο U_3 δρα μη τετριμμένα στις $|\psi_1\rangle = |00\rangle$ και $|\psi_4\rangle = |11\rangle$, μέσω του πίνακα

$$\tilde{U}_3 = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

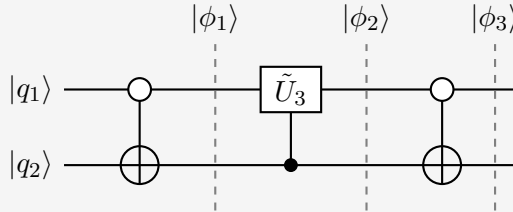
Οι δύο ακολουθίες διαφέρουν και στα δύο bit, οπότε $m = 2$ και χρειαζόμαστε κώδικα Gray. Αλλάζοντας πρώτα το δεύτερο bit και μετά το πρώτο, παίρνουμε:

$$g_0 = 00$$

$$g_1 = 01$$

$$g_2 = 11$$

Χρειαζόμαστε μία πύλη F_1 , που ανταλλάσσει τις $|00\rangle$ και $|01\rangle$. Είναι μία X στο q_2 που ενεργοποιείται όταν $q_1 = 0$, δηλαδή μία CNOT με άσπρο κυκλάκι. Η μεσαία πύλη W δρα στις $|g_1\rangle = |01\rangle$ και $|g_2\rangle = |11\rangle$, που διαφέρουν στο πρώτο bit. Είναι επομένως η $\tilde{U}_3$ στο q_1 , που ενεργοποιείται όταν $q_2 = 1$. Το bit του q_1 είναι 0 στην $|g_1\rangle$, οπότε χρησιμοποιούμε τον $\tilde{U}_3$ όπως είναι. Το πλήρες κύκλωμα είναι:



Ας ελέγξουμε τη δράση του κυκλώματος στην είσοδο $|00\rangle$. Η F_1 ενεργοποιείται, αφού $q_1 = 0$, και αλλάζει το q_2 :

$$|\phi_1\rangle = |01\rangle$$

Η W ενεργοποιείται, αφού τώρα $q_2 = 1$, και εφαρμόζει τον $\tilde{U}_3$ στο q_1 , που είναι $|0\rangle$. Η πρώτη στήλη του $\tilde{U}_3$ δίνει:

$$|\phi_2\rangle = \frac{\sqrt{3}}{2} |01\rangle + \frac{1}{2} |11\rangle$$

Η δεύτερη F_1 αλλάζει το q_2 μόνο στον όρο όπου $q_1 = 0$, δηλαδή το $|01\rangle$ γίνεται $|00\rangle$, ενώ το $|11\rangle$ μένει ίδιο:

$$|\phi_3\rangle = \frac{\sqrt{3}}{2} |00\rangle + \frac{1}{2} |11\rangle$$

Αυτή είναι ακριβώς η πρώτη στήλη του $\tilde{U}_3$, δηλαδή $\tilde{U}_3 |\psi_1\rangle = \frac{\sqrt{3}}{2} |\psi_1\rangle + \frac{1}{2} |\psi_4\rangle$, όπως πρέπει. Με τον ίδιο τρόπο, για την είσοδο $|11\rangle$ η πρώτη F_1 δεν ενεργοποιείται, η W δίνει $-\frac{1}{2} |01\rangle + \frac{\sqrt{3}}{2} |11\rangle$, και η δεύτερη F_1 δίνει:

$$-\frac{1}{2} |00\rangle + \frac{\sqrt{3}}{2} |11\rangle$$

που είναι η δεύτερη στήλη του $\tilde{U}_3$.

Ας δούμε τέλος και την κατάσταση $|01\rangle$, που χρησιμοποιείται προσωρινά από την F_1 . Η πρώτη F_1 τη στέλνει στην $|00\rangle$. Η W δεν ενεργοποιείται, αφού $q_2 = 0$. Η δεύτερη F_1 τη φέρνει πίσω στην $|01\rangle$, οπότε η $|01\rangle$ μένει τελικά αμετάβλητη, όπως πρέπει. Η $|10\rangle$, τέλος, δεν ενεργοποιεί καμία πύλη, αφού $q_1 = 1$ και $q_2 = 0$.

Ας γράψουμε τώρα μία συνάρτηση `two_level_circuit()` στην `qulib2.py`, που κατασκευάζει αυτόματα το κύκλωμα ενός διεπίπεδου πίνακα. Δέχεται έναν διεπίπεδο πίνακα ως `Gate`, όπως αυτούς που επιστρέφει η `two_level_decomposition()`, και επιστρέφει τη λίστα των πυλών του κυκλώματος, με τη σειρά που εφαρμόζονται. Όλες οι πύλες είναι αντικείμενα `ControlledGate`. Σε αντίθεση με τις μεθόδους που είδαμε μέχρι τώρα, αυτή είναι μία ανεξάρτητη συνάρτηση και όχι μέθοδος της `Gate`, αφού δεν επιστρέφει μία πύλη αλλά ολόκληρο κύκλωμα:

```
def two_level_circuit(gate):
    n = gate.n_qubits
    N = 2 ** n
    U = gate.calc_U()
    differs = ~np.isclose(U, np.eye(N))
    active = sorted(set(np.where(differs)[0]) | set(np.where(differs)[1]))
    if not active:
        return []
    if len(active) != 2:
        raise ValueError("gate is not a two-level matrix")
    r, s = active
    a, b, c, d = U[r, r], U[r, s], U[s, r], U[s, s]

    r_bits, s_bits = _bits_of(r, n), _bits_of(s, n)
    diff_positions = [q for q in range(n) if r_bits[q] != s_bits[q]]

    path = [r_bits]
    for p in diff_positions:
        step = list(path[-1])
        step[p] = s_bits[p]
        path.append(step)

    def controlled_step(bits, target, step_gate):
        others = [q for q in range(n) if q != target]
        inverted = [q for q in others if bits[q] == 0]
        return ControlledGate(n_qubits=n, through=[], control=others,
                               target=[target], gate=step_gate,
                               inverted_control=inverted)

    walk = [controlled_step(path[k], diff_positions[k], GateX())
             for k in range(len(diff_positions) - 1)]

    target = diff_positions[-1]
    # path[-2] (where |psi_r> sits after the walk) may have target bit 0 or 1;
    # orient U_tilde so its rows/cols match whichever bit value is which
    if path[-2][target] == 0:
        U_tilde = np.array([[a, b], [c, d]])
    else:
        U_tilde = np.array([[d, c], [b, a]])
    middle = controlled_step(path[-2], target, Gate(U=U_tilde))
    return walk + [middle] + list(reversed(walk))
```

Ο κώδικας ακολουθεί βήμα-βήμα την κατασκευή. Όπως και πριν, η Python αριθμεί από το 0, οπότε η κατάσταση $|\psi_1\rangle$ του κειμένου αντιστοιχεί στον δείκτη 0, και το qubit q_1 στη θέση 0.

- Πρώτα βρίσκουμε τις δύο γραμμές και στήλες r, s στις οποίες ο πίνακας διαφέρει από τον ταυτοτικό, και διαβάζουμε από αυτές τα στοιχεία a, b, c, d του $\tilde{U}$. Αν ο πίνακας διαφέρει από τον ταυτοτικό σε περισσότερες από δύο γραμμές και στήλες, δεν είναι διεπίπεδος, και η συνάρτηση σταματά με μήνυμα λάθους.
- Μετατρέπουμε τα r, s σε ακολουθίες bit με τη `_bits_of`, βρίσκουμε τις θέσεις των bit στα οποία διαφέρουν (`diff_positions`), και κατασκευάζουμε τον κώδικα Gray path, δηλαδή τη λίστα $g_0, g_1, \dots, g_m$, αλλάζοντας αυτά τα bit ένα-ένα. Ο κώδικας αλλάζει τα bit από τα αριστερά προς τα δεξιά, οπότε για τις $|000\rangle$ και $|111\rangle$ δίνει τον κώδικα Gray 000, 100, 110, 111, διαφορετικό από αυτόν του παραδείγματος 4.25. Όπως είδαμε, οποιοσδήποτε κώδικας Gray δίνει ένα σωστό κύκλωμα.
- Η εσωτερική συνάρτηση `controlled_step` κατασκευάζει την ελεγχόμενη πύλη ενός βήματος. Το qubit-στόχος (`target`) είναι αυτό που αλλάζει σε αυτό το βήμα, και όλα τα υπόλοιπα qubit γίνονται qubit ελέγχου. Όσα από αυτά έχουν τιμή 0 στο κοινό μοτίβο μπαίνουν και στη λίστα `inverted_control`, δηλαδή παίρνουν άσπρο κυκλάκι (ενότητα 4.5).
- Οι πύλες $F_1, \dots, F_{m-1}$ αποθηκεύονται στη λίστα `walk`, και είναι βήματα με πύλη την X . Η μεσαία πύλη W είναι ένα βήμα με πύλη τον $\tilde{U}$. Αν το bit του qubit-στόχου είναι 1 στην $|g_{m-1}\rangle$, χρησιμοποιούμε τον πίνακα με ανταλλαγμένες γραμμές και στήλες, όπως εξηγήσαμε στην απλή περίπτωση, και αυτό κάνει το τελευταίο `if`.
- Η συνάρτηση επιστρέφει τη λίστα `walk`, μετά την W , και μετά τη `walk` ανάποδα (`reversed(walk)`). Αφού κάθε F_k είναι αντίστροφη του εαυτού της, για τη δεύτερη μισή του κυκλώματος χρησιμοποιούμε τις ίδιες ακριβώς πύλες.

Ας επιβεβαιώσουμε αριθμητικά το παράδειγμα 4.25. Κατασκευάζουμε έναν διεπίπεδο πίνακα 8×8 στις καταστάσεις $|000\rangle$ και $|111\rangle$, με $\tilde{U} = H$, και ελέγχουμε ότι η `two_level_circuit()` δίνει $2(m-1) + 1 = 5$ πύλες, ότι το γινόμενό τους είναι ο U , ότι κάθε άλλη βασική κατάσταση, ακόμα και αυτές που χρησιμοποιούνται προσωρινά από τις πύλες F_1 και F_2 , μένει αμετάβλητη, και ότι μία υπέρθεση $a_1|000\rangle + a_8|111\rangle$ μετασχηματίζεται σύμφωνα με τον $\tilde{U}$. Το script βρίσκεται στο αρχείο `python/ex_two_level_circuit.py`:

```
import functools
import operator

import numpy as np

from qulib2 import Gate, QuantumState, two_level_circuit

np.set_printoptions(precision=4, suppress=True)

# Example 4.25: a two-level matrix U acting on |psi_1> = |000> and
# |psi_8> = |111> (Hamming distance m = 3), here with U_tilde = H
s2 = 1 / np.sqrt(2)
U_tilde = np.array([[s2, s2], [s2, -s2]], dtype=complex)

# Python indices start at 0: |psi_1> is index 0 and |psi_8> is index 7
U = np.eye(8, dtype=complex)
U[0, 0] = U_tilde[0, 0]
U[0, 7] = U_tilde[0, 1]
U[7, 0] = U_tilde[1, 0]
U[7, 7] = U_tilde[1, 1]

circuit = two_level_circuit(Gate(U=U))
print(f"number of gates: {len(circuit)} (expect 2 * (3 - 1) + 1 = 5)")

# the gates are listed in the order they are applied, so the first one
# must be the rightmost factor of the product
product = functools.reduce(operator.matmul, reversed(circuit))
print(f"circuit matches U: {np.allclose(product.U, U)}")
```

```

# every basis state other than |000>, |111> must return to itself,
# even those used temporarily by F_1 and F_2 (here |100> and |110>)
for bits in ["001", "010", "011", "100", "101", "110"]:
    psi = QuantumState({bits: 1})
    out = product @ psi
    print(f"|{bits}> unchanged: {np.allclose(out.a, psi.a)}")

# a superposition a1|000> + a8|111> must give b1|000> + b8|111>,
# with b1 = a a1 + b a8 and b8 = c a1 + d a8
a1, a8 = 0.6, 0.8j
psi = QuantumState(np.array([a1, 0, 0, 0, 0, 0, 0, a8]))
out = product @ psi
b1, b8 = U_tilde @ np.array([a1, a8])
print(f"amplitude of |000> is b1: {np.isclose(out.a[0], b1)}")
print(f"amplitude of |111> is b8: {np.isclose(out.a[7], b8)}")

```

4.8 Σύνοψη: από έναν αυθαίρετο μοναδιαίο πίνακα σε πύλες H , T και CNOT

Έχουμε πλέον όλα τα κομμάτια που χρειαζόμαστε για να απαντήσουμε σε ένα βασικό ερώτημα: μπορούμε να υλοποιήσουμε *οποιοδήποτε* μοναδιαίο τελεστή σε n qubit χρησιμοποιώντας μόνο ένα μικρό, σταθερό σύνολο πυλών; Ας συγκεντρώσουμε τα βήματα της κατασκευής, με αναφορά στις ενότητες όπου είδαμε το καθένα.

Ξεκινάμε από έναν $N \times N$ μοναδιαίο πίνακα U , με $N = 2^n$, που θέλουμε να υλοποιήσουμε. Μπορεί να είναι, για παράδειγμα, ο πίνακας ενός ολόκληρου κβαντικού αλγορίθμου.

Στο πρώτο βήμα γράφουμε τον U ως γινόμενο διεπίπεδων πινάκων (ενότητα 4.6):

$$U = U_1 U_2 \cdots U_d$$

όπου το πλήθος των παραγόντων είναι το πολύ

$$d \leq \frac{N(N-1)}{2}$$

Κάθε U_i δρα μη τετριμμένα μόνο σε δύο βασικές καταστάσεις, μέσω ενός 2×2 μοναδιαίου πίνακα.

Στο δεύτερο βήμα κατασκευάζουμε ένα κύκλωμα για κάθε U_i (ενότητα 4.7). Αν οι δύο καταστάσεις διαφέρουν σε ένα μόνο bit, αρκεί μία ελεγχόμενη πύλη με $n-1$ qubit ελέγχου. Αν διαφέρουν σε m bit, χρειαζόμαστε επιπλέον τις πύλες F_k του κώδικα Gray, συνολικά $2m-1$ πύλες. Σε κάθε περίπτωση, οι πύλες που εμφανίζονται είναι ελεγχόμενες πύλες $C_{n-1}^1(V)$, όπου ο V είναι ένας 2×2 πίνακας: ο X για τις F_k , και ο $\tilde{U}_i$ (ή ο $\tilde{U}_i'$) για την πύλη W .

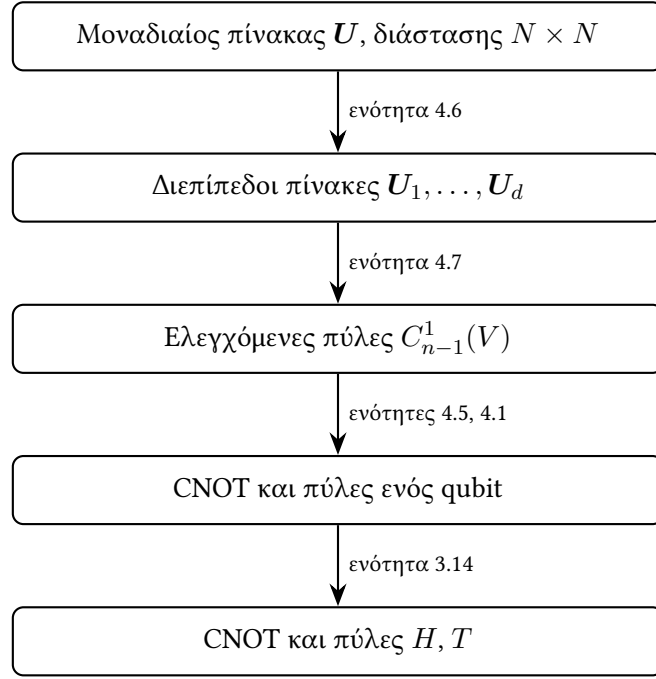
Στο τρίτο βήμα βάζουμε τα κυκλώματα των U_i το ένα μετά το άλλο. Αφού ο U_d βρίσκεται δεξιότερα στο γινόμενο, δρα πρώτος, οπότε το κύκλωμά του μπαίνει πρώτο, και ακολουθούν τα κυκλώματα των $U_{d-1}, \dots, U_1$.

Στο τέταρτο βήμα αναλύουμε κάθε ελεγχόμενη πύλη $C_{n-1}^1(V)$ σε CNOT και ελεγχόμενες πύλες με ένα qubit ελέγχου (ενότητα 4.5). Κάθε ελεγχόμενη πύλη με ένα qubit ελέγχου αναλύεται με τη σειρά της σε δύο CNOT και πύλες ενός qubit (ενότητα 4.1). Οι άσπρες κουκκίδες δεν δημιουργούν πρόβλημα, αφού κάθε άσπρη κουκκίδα ισοδυναμεί με μία μαύρη, με μία πύλη X πριν και μία μετά.

Στο πέμπτο και τελευταίο βήμα, κάθε πύλη ενός qubit που εμφανίστηκε προσεγγίζεται όσο καλά θέλουμε με πύλες H και T (ενότητα 3.14).

Η αλυσίδα των βημάτων φαίνεται στο σχήμα 4.9.

Ας δούμε πώς εφαρμόζονται τα πρώτα βήματα σε έναν συγκεκριμένο πίνακα.



Σχήμα 4.9: Τα βήματα της κατασκευής ενός κυκλώματος για οποιονδήποτε μοναδιαίο πίνακα U , από τον αρχικό πίνακα μέχρι τις πύλες H, T και CNOT.

Παράδειγμα 4.27: Από την QFT δύο qubit σε ελεγχόμενες πύλες

Στο παράδειγμα 4.22 γράψαμε τον πίνακα $\mathcal{M}$ της QFT δύο qubit ως γινόμενο έξι διεπίπεδων πινάκων:

$$\mathcal{M} = U_1 U_2 U_3 U_4 U_5 U_6$$

Για κάθε U_i βρίσκουμε τις δύο καταστάσεις στις οποίες δρα, από τις γραμμές και τις στήλες όπου διαφέρει από τον ταυτοτικό πίνακα, και μετράμε σε πόσα bit διαφέρουν. Αν διαφέρουν σε ένα bit, το κύκλωμα έχει μία πύλη. Αν διαφέρουν σε δύο bit, έχουμε $m = 2$, και το κύκλωμα έχει $2m - 1 = 3$ πύλες:

πίνακας	καταστάσεις	bit που διαφέρουν	πύλες
U_1	$ 00\rangle, 01\rangle$	1	1
U_2	$ 00\rangle, 10\rangle$	1	1
U_3	$ 00\rangle, 11\rangle$	2	3
U_4	$ 01\rangle, 10\rangle$	2	3
U_5	$ 01\rangle, 11\rangle$	1	1
U_6	$ 10\rangle, 11\rangle$	1	1

Τα κυκλώματα των U_1 και U_3 τα κατασκευάσαμε αναλυτικά στο παράδειγμα 4.26, και των υπολοίπων κατασκευάζονται με τον ίδιο τρόπο. Συνολικά χρειαζόμαστε

$$1 + 1 + 3 + 3 + 1 + 1 = 10$$

ελεγχόμενες πύλες. Αφού έχουμε δύο qubit, κάθε πύλη έχει $n - 1 = 1$ qubit ελέγχου, δηλαδή είναι μία πύλη $C_1^1(V)$, που αναλύεται σε δύο CNOT και πύλες ενός qubit. Για το πλήρες κύκλωμα τοποθετούμε πρώτα το κύκλωμα του U_6 , μετά του U_5 , και ούτω καθεξής, μέχρι το κύκλωμα του U_1 , που μπαίνει τελευταίο.

Η κατασκευή αυτή δεν δίνει το μικρότερο δυνατό κύκλωμα. Θα δούμε αργότερα ότι η QFT δύο qubit υλοποιείται με πολύ λιγότερες πύλες. Η αξία της κατασκευής είναι ότι δουλεύει για *οποιονδήποτε* μοναδιαίο πίνακα, χωρίς να χρειάζεται να ξέρουμε τίποτα ειδικό για αυτόν.

Το μόνο σημείο σε όλη αυτή την αλυσίδα όπου κάνουμε προσέγγιση είναι το τελευταίο βήμα, η έκφραση

των πυλών ενός qubit μέσω των H και T . Όλα τα προηγούμενα βήματα, δηλαδή η ανάλυση σε διεπίπεδους πίνακες, τα κυκλώματα με τον κώδικα Gray και η ανάλυση των ελεγχόμενων πυλών σε CNOT, είναι *ακριβή*.

Πρέπει επίσης να σημειώσουμε ότι η κατασκευή αυτή δεν είναι αποδοτική. Το πλήθος των διεπίπεδων πινάκων στο πρώτο βήμα μπορεί να φτάσει τα $N(N-1)/2$, και αφού $N = 2^n$, αυτό είναι περίπου

$$\frac{N^2}{2} = \frac{4^n}{2}$$

δηλαδή μεγαλώνει εκθετικά με το πλήθος n των qubit. Για $n = 10$ qubit, για παράδειγμα, είναι ήδη περίπου μισό εκατομμύριο. Η κατασκευή αποδεικνύει λοιπόν ότι κάθε μοναδιαίος τελεστής *μπορεί* να υλοποιηθεί, αλλά δεν είναι πρακτική μέθοδος για τη σχεδίαση κυκλωμάτων, όπου χρησιμοποιούνται πολύ πιο αποδοτικές τεχνικές.

Αποτέλεσμα 4.5: Το σύνολο $\{H, T, \text{CNOT}\}$ είναι καθολικό

Κάθε μοναδιαίος τελεστής σε οποιοδήποτε πλήθος qubit μπορεί να υλοποιηθεί, με όσο μικρό σφάλμα θέλουμε, από ένα κύκλωμα που χρησιμοποιεί μόνο τις πύλες H , T και CNOT. Για τον λόγο αυτό λέμε ότι το σύνολο $\{H, T, \text{CNOT}\}$ είναι *καθολικό*.

4.9 Μερικές ιδιότητες των κβαντικών κυκλωμάτων

Κλείνουμε το κεφάλαιο με δύο ιδιότητες των κβαντικών κυκλωμάτων που αξίζει να έχουμε καθαρά στο μυαλό μας.

Η πρώτη είναι ότι κάθε κβαντικό κύκλωμα στο οποίο δεν γίνεται μέτρηση είναι *αντιστρέψιμο* (αποτέλεσμα 1.3). Ένα τέτοιο κύκλωμα περιγράφεται από έναν μοναδιαίο πίνακα U , ως γινόμενο των μοναδιαίων πινάκων των πυλών του, και για κάθε μοναδιαίο πίνακα ισχύει:

$$U^T U = I$$

$$U U^T = I$$

Δηλαδή ο U^T είναι ο αντίστροφος του U . Στην πράξη, αυτό σημαίνει ότι για κάθε κύκλωμα υπάρχει ένα «αντίστροφο» κύκλωμα που ακυρώνει τη δράση του. Αν το κύκλωμα αποτελείται από τις πύλες $V_1, V_2, \dots, V_k$, με αυτή τη σειρά, ο πίνακάς του είναι:

$$U = V_k \cdots V_2 V_1$$

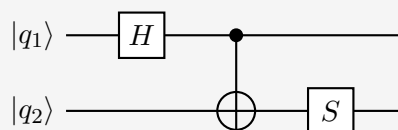
Ο συζυγής ανάστροφος ενός γινομένου είναι το γινόμενο των συζυγών ανάστροφων με την αντίστροφη σειρά, οπότε:

$$U^T = V_1^T V_2^T \cdots V_k^T$$

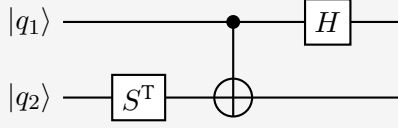
Το αντίστροφο κύκλωμα εφαρμόζει επομένως πρώτα την V_k^T , μετά την V_{k-1}^T , και ούτω καθεξής μέχρι την V_1^T . Δηλαδή διατρέχουμε τις πύλες από το τέλος προς την αρχή, και αντικαθιστούμε καθεμία με τη συζυγή ανάστροφή της.

Παράδειγμα 4.28: Το αντίστροφο ενός κυκλώματος

Θεωρούμε το κύκλωμα δύο qubit που εφαρμόζει πρώτα μία H στο q_1 , μετά μία CNOT με qubit ελέγχου το q_1 και στόχο το q_2 , και τέλος μία S στο q_2 :



Το αντίστροφο κύκλωμα εφαρμόζει τις πύλες από το τέλος προς την αρχή, με καθεμία αντικατεστημένη από τη συζυγή ανάστροφή της:



Η H και η CNOT έμειναν ίδιες, αφού, όπως θα δούμε αμέσως παρακάτω, είναι αντίστροφες του εαυτού τους. Η S όμως αντικαταστάθηκε από την

$$S^T = \begin{bmatrix} 1 & 0 \\ 0 & -j \end{bmatrix}$$

που είναι διαφορετική από την S , αφού ο συζυγής του j είναι το $-j$. Ας ελέγξουμε ότι το δεύτερο κύκλωμα ακυρώνει το πρώτο για την είσοδο $|00\rangle$. Το πρώτο κύκλωμα δίνει διαδοχικά:

$$|00\rangle \xrightarrow{H} \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle)$$

$$\frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

$$\frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \xrightarrow{S} \frac{1}{\sqrt{2}} (|00\rangle + j|11\rangle)$$

Το αντίστροφο κύκλωμα δίνει στη συνέχεια:

$$\frac{1}{\sqrt{2}} (|00\rangle + j|11\rangle) \xrightarrow{S^T} \frac{1}{\sqrt{2}} (|00\rangle + j(-j)|11\rangle) = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

$$\frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle)$$

$$\frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \xrightarrow{H} |00\rangle$$

δηλαδή επιστρέφουμε στην αρχική κατάσταση, αφού $j(-j) = 1$.

Μερικές πύλες είναι αντίστροφες του εαυτού τους, δηλαδή $V^2 = I$. Αυτό συμβαίνει όταν ο πίνακας μίας πύλης είναι, εκτός από μοναδιαίος, και Ερμιτιανός, δηλαδή ίσος με τον συζυγή ανάστροφό του, $V^T = V$. Τότε:

$$V^2 = VV = V^T V = I$$

Τέτοιες πύλες είναι οι X, Y, Z, H , η CNOT, η CZ, η SWAP, η Toffoli και η Fredkin. Άλλες πύλες, όπως οι S και T , δεν είναι Ερμιτιανές. Για την T , για παράδειγμα:

$$T^T = \begin{bmatrix} 1 & 0 \\ 0 & e^{-j\pi/4} \end{bmatrix} \neq T$$

Επειδή κάθε κύκλωμα αντιστρέφεται με αυτόν τον τρόπο, ένα κβαντικό κύκλωμα σχεδιάζεται πάντα σαν μία ακολουθία πυλών στον χρόνο: κάθε σύρμα διασχίζει το διάγραμμα από τα αριστερά προς τα δεξιά, χωρίς να «γυρίζει πίσω» για να τροφοδοτήσει μία πύλη που βρίσκεται πιο αριστερά, όπως γίνεται στα κλασικά κυκλώματα με ανάδραση. Η μόνη διεργασία που ξεφεύγει από αυτό το πλαίσιο είναι η μέτρηση, που δεν την έχουμε χρησιμοποιήσει ακόμα σε κανένα κύκλωμα. Όπως θα δούμε, η μέτρηση είναι μία μη αντιστρέψιμη, πιθανοκρατική διεργασία.

Η δεύτερη ιδιότητα σχετίζεται με το θεώρημα μη κλωνοποίησης (ενότητα 4.3), που μας λέει τι δεν μπορούμε να κατασκευάσουμε με ένα κβαντικό κύκλωμα.

Εμβάθυνση 4.1: Υπάρχει κβαντικό «for»;

Μία φυσική ερώτηση είναι αν μπορούμε να κατασκευάσουμε έναν κβαντικό *μετρητή* (counter), δηλαδή το κβαντικό ανάλογο ενός καταχωρητή με flip-flop που μετρά παλμούς ρολογιού, διαβάζοντας κάθε φορά την τρέχουσα τιμή του για να υπολογίσει την επόμενη ή για να αποφασίσει αν θα σταματήσει.

Πρέπει πρώτα να ξεχωρίσουμε δύο πολύ διαφορετικά πράγματα που συχνά ονομάζουμε «βρόχο» (loop). Το πρώτο είναι η επανάληψη μίας πύλης, ή ενός ολόκληρου κυκλώματος, έναν σταθερό αριθμό φορών, γνωστό από πριν. Αυτό είναι ένα «for» στην *περιγραφή* του κυκλώματος, όχι στην ίδια την κβαντική επεξεργασία, και δεν έχει κανένα πρόβλημα. Αυτό ακριβώς κάναμε, για παράδειγμα, στην κατασκευή της ελεγχόμενης πύλης Hadamard με τρία qubit ελέγχου (παράδειγμα 4.17), όπου εφαρμόσαμε την ίδια κατασκευή δύο φορές. Το ίδιο κάνει και ο αλγόριθμος του Grover, που επαναλαμβάνει τον ίδιο τελεστή έναν προκαθορισμένο αριθμό φορών. Σε όλες αυτές τις περιπτώσεις, το πόσες φορές θα επαναλάβουμε είναι μία κλασική παράμετρος, γνωστή πριν καν κατασκευάσουμε το κύκλωμα, και δεν χρειάζεται να «διαβάσουμε» τίποτα από τα qubit για να την αποφασίσουμε.

Το δεύτερο είναι ένας βρόχος του οποίου ο αριθμός επαναλήψεων εξαρτάται από την *τρέχουσα τιμή* μίας κβαντικής κατάστασης, όπως σε έναν κλασικό μετρητή, όπου σε κάθε παλμό ρολογιού το κύκλωμα «διαβάζει» την τιμή του καταχωρητή, για να υπολογίσει την επόμενη και για να αποφασίσει αν θα σταματήσει. Αυτό το είδος βρόχου δεν μπορεί να υλοποιηθεί κβαντικά, διατηρώντας την υπέρθεση, και ο λόγος είναι ακριβώς το θεώρημα μη κλωνοποίησης.

Ένας κλασικός μετρητής στηρίζεται σε κάτι που στα κλασικά κυκλώματα θεωρούμε δεδομένο: μπορούμε να «διαβάσουμε» την τιμή ενός bit, δηλαδή να την αντιγράψουμε με ένα απλό fan-out, όσες φορές θέλουμε, χωρίς να την επηρεάσουμε. Έτσι το ίδιο αντίγραφο τροφοδοτεί ταυτόχρονα τη λογική που υπολογίζει την επόμενη κατάσταση και τη συνθήκη τερματισμού. Σε ένα qubit που βρίσκεται σε μία άγνωστη υπέρθεση $a|0\rangle + b|1\rangle$, αυτή ακριβώς η αντιγραφή απαγορεύεται: δεν υπάρχει κβαντική πύλη που να δίνει δύο ανεξάρτητα αντίγραφα της κατάστασης του μετρητή, ώστε το ένα να κρατήσει την υπέρθεση για την επόμενη επανάληψη και το άλλο να χρησιμοποιηθεί για να αποφασίσουμε αν θα σταματήσουμε.

Ο μόνος τρόπος να «διαβάσουμε» πραγματικά την τιμή ενός qubit είναι η μέτρηση. Η μέτρηση όμως βρίσκεται έξω από το πλαίσιο των μοναδιαίων, αντιστρέψιμων κυκλωμάτων που μελετήσαμε σε αυτό το κεφάλαιο: καταστρέφει την υπέρθεση και δίνει ένα τυχαίο, κλασικό αποτέλεσμα. Ένας «κβαντικός μετρητής» θα μπορούσε επομένως να υλοποιηθεί μόνο μετρώντας σε κάθε βήμα, αλλά τότε θα καταστρέφαμε ακριβώς την υπέρθεση που θέλαμε να επεξεργαστούμε. Με άλλα λόγια, υπάρχει κβαντικό «for», με σταθερό και γνωστό από πριν αριθμό επαναλήψεων, αλλά δεν υπάρχει κβαντικό «while» ή μετρητής που να «διαβάζει» τον εαυτό του σε κάθε βήμα, όσο θέλουμε να κρατήσουμε την πληροφορία του σε υπέρθεση.

Η αντιστρεψιμότητα και η αδυναμία κλωνοποίησης δεν είναι λοιπόν δύο άσχετες παρατηρήσεις. Και οι δύο προκύπτουν από το ότι όλα όσα κάνουμε σε ένα κβαντικό κύκλωμα, όσο δεν μετράμε, είναι μοναδιαίοι μετασχηματισμοί: γραμμικοί και αντιστρέψιμοι, και επομένως ανίκανοι να αντιγράψουν ή να «διαβάσουν» μία άγνωστη υπέρθεση χωρίς να τη διαταράξουν.

Σύνοψη Ενότητας

Σε αυτό το κεφάλαιο περάσαμε από τις πύλες ενός qubit σε κβαντικά κυκλώματα πολλαπλών qubit, και καταλήξαμε σε ένα καθολικό σύνολο πυλών. Συνοπτικά:

- Ορίσαμε τις ελεγχόμενες πύλες $C_n^k(U)$, με n qubit ελέγχου και k qubit-στόχους, με πιο γνωστή την CNOT, $C_1^1(X)$, και είδαμε το άσπρο κυκλάκι, που ενεργοποιεί την πύλη όταν το qubit ελέγχου είναι $|0\rangle$.
- Δείξαμε ότι *οποιαδήποτε* ελεγχόμενη πύλη $C_1^1(U)$ υλοποιείται με δύο CNOT και τέσσερις πύλες ενός qubit, μέσω της αναπαράστασης $U = e^{j\alpha} \mathbf{A} \mathbf{X} \mathbf{B} \mathbf{X} \mathbf{C}$ και της ελεγχόμενης καθολικής φάσης Φ_α .
- Αποδείξαμε το θεώρημα μη κλωνοποίησης: κανένας μοναδιαίος τελεστής δεν μπορεί να αντιγρά-

ψει μία άγνωστη κατάσταση qubit.

- Μελετήσαμε τις πύλες τριών qubit Toffoli, $C_2^1(X)$, και Fredkin, $C_1^2(S_W)$, και δείξαμε πώς κατασκευάζεται κάθε ελεγχόμενη πύλη $C_n^1(U)$, βήμα-βήμα, από πύλες με λιγότερα qubit ελέγχου.
- Δείξαμε ότι κάθε μοναδιαίος πίνακας $N \times N$ γράφεται ως γινόμενο διεπίπεδων πινάκων, και ότι κάθε διεπίπεδος πίνακας υλοποιείται με ελεγχόμενες πύλες, είτε απευθείας, είτε με τη βοήθεια ενός κώδικα Gray.
- Συνδυάζοντας τα παραπάνω, καταλήξαμε ότι το σύνολο $\{H, T, \text{CNOT}\}$ είναι καθολικό: κάθε μοναδιαίος τελεστής, σε οποιοδήποτε πλήθος qubit, προσεγγίζεται όσο καλά θέλουμε με κυκλώματα από αυτές τις τρεις πύλες.
- Είδαμε τέλος ότι κάθε κύκλωμα χωρίς μέτρηση είναι αντιστρέψιμο, και ότι το θεώρημα μη κλωνοποίησης αποκλείει έναν κβαντικό «μετρητή»: υπάρχει κβαντικό «for», με σταθερό αριθμό επαναλήψεων, αλλά όχι κβαντικό «while» πάνω σε μία άγνωστη υπέρθεση.