ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1674

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΑΤΙΣΜΟΥ ΠΟΥ ΒΑΣΙΖΟΝΤΑΙ ΣΤΙΣ ΑΡΧΕΣ ΚΒΑΝΤΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΣΚΑΜΑΓΚΟΣ-ΜΗΤΡΟΥ ΕΥΓΕΝΙΟΣ

ΚΑΣΙΑΡΑΣ ΒΑΓΓΕΛΗΣ

Επιβλέπων: ΔΗΜΗΤΡΗΣ ΚΑΡΕΛΗΣ

ΠΑΤΡΑ 2019

ΠΕΡΙΛΗΨΗ

Σε αυτήν την εργασία ο σκοπός της είναι η μελέτη των κβαντικών γλωσσών προγραμματισμού και η μελέτη των κβαντικών κυκλωμάτων στην γλώσσα *nQML*.Πιο συγκεκριμένα μελετήσαμε το μοντέλο του κβαντικού προγραμματισμού αυτού και ορισμένους κανόνες που καθορίζουν την κατασκευή των κβαντικών κυκλωμάτων.Ακόμα μελετήσαμε δύο γλώσσες προγραμματισμού την *QPL* και *QML*,όπου η *nQML* αποτελεί παραλλαγή της *QPL*.Ουσιαστικά για τις δυο γλώσσες *QPL* και *QML* τις συντάξαμε και δώσαμε κάποια παραδείγματα.

Οι κβαντικοί αλγόριθμοι συνήθως υλοποιούνται με τη βοήθεια κβαντικών κυκλωμάτων, όμως για να χρησιμοποιηθεί το κβαντικό μοντέλο υπολογισμών από προγραμματιστές απαιτείται και η κατασκευή κατάλληλων γλωσσών προγραμματισμού. Τα κβαντικά κυκλώματα θα χρησιμοποιηθούν πλέον για την περιγραφή των εντολών της γλώσσας προγραμματισμού.Σιγά σιγά όσο περνούσαν τα χρόνια δημιουργήθηκαν κβαντικοί αλγόριθμοι αποδεδειγμένα ταχύτεροι από τους αντίστοιχους κλασικούς.Ειδικότερα ο αλγόριθμος του Shor για την παραγοντοποίηση μεγάλων αριθμών είχε τραβήξει το ενδιαφέρον της ερευνητικής κοινότητας για χρόνια . Οι ιδιαιτερότητες του κβαντικού μοντέλου υπολογισμών, και η αντίθεσή του με την διαίσθηση του κλασικού προγραμματιστή είναι ένας από τους λόγους που δεν είχαν αναπτυχθεί ακόμα πολλοί ενδιαφέροντες κβαντικοί αλγόριθμοι.

Με αποτέλεσμα , οι προσπάθειες για την κατασκευή λειτουργικών κβαντικών υπολογιστών οδήγησαν στη μελέτη των κβαντικών κυκλωμάτων από τα οποία αποτελούνται.Σε αυτή την εργασία είδαμε διάφορες ιδιότητες των κβαντικών κυκλωμάτων και μελετήσαμε τους τρόπους κατασκευής τους.Τέλος,δώσαμε κάποια παραδείγματα κβαντικών αλγόριθμων όπως αλγόριθμος Grover και deutsch.Βέβαια η τεχνολογία έχει αναπτυχθεί πολύ στην σημερίνη εποχή και έχουν δημιουργηθει και έχουν βρει καινούργιες μεθόδους υλοποίησης.

ΠΕΡΙΕΧΟΜΕΝΑ

Κατάλογος πινάκων

Κατάλογος σχημάτων

5.7 Σειριακή σύνθεση στο FQC: $\varphi_{\beta \circ \alpha}$

5.8 Παράλληλη σύνθεση στο FQC: $\varphi_{a \otimes \beta}$

5.8.1 Κύκλωμα EMB, το οποίο ανήκει στο $\mathrm{FQC}^{\simeq}$ άρα και στο FQC

5.9 Κύκλωμα VAR. Πρόκειται απλώς για "ομαδοποίηση" των καλωδίων του τύπου τ, μια αναδιάταξη

5.10 Κύκλωμα SUP

5.11 Κύκλωμα LET

5.12 Κύκλωμα PROD

5.13 Κύκλωμα IF

5.14 Κύκλωμα IFM

5.15 Κύκλωμα LETPROD

Κεφάλαιο 1

Εισαγωγή

Σκοπός αυτής της εργασίας είναι η μελέτη ενός πεδίου της επιστήμης των υπολογιστών,του μοντέλου των κβαντικών υπολογισμών.Ακόμα θα μελετήσουμε τα κβαντικά κυκλώματα και τις κβαντικές γλώσσες προγραμματισμού. Οι κβαντικοί αλγόριθμοι συνήθως υλοποιούνται με τη βοήθεια κβαντικών κυκλωμάτων, όμως για να χρησιμοποιηθεί το κβαντικό μοντέλο υπολογισμών από προγραμματιστές απαιτείται και η κατασκευή κατάλληλων γλωσσών προγραμματισμού. Μας απασχόλησαν οι γλώσσες *QPL, QML* και φυσικά η γλώσσα *nQML*. Βασιζόμενοι στα θεωρήματα των κβαντικών κυκλωμάτων σχεδιάσαμε τα κατάλληλα κβαντικά κυκλώματα, τα οποία υλοποιούν κάθε εντολή της γλώσσας.

1.1 Ιστορία των κβαντικών υπολογισμών

Η αρχή λειτουργίας ενός κβαντικού υπολογιστή βασίζεται στη κβαντομηχανική γεγονός που του δίνει την δυνατότητα να υπολογίζει συναρτήσεις με έναν εντελώς διαφορετικό τρόπο σε σχέση με έναν συνηθισμένο υπολογιστή ο οποίος λειτουργεί με βάση την κλασική φυσική. Ένα από τα σημαντικότερα πλεονεκτήματα ενός κβαντικού υπολογιστή είναι η δυνατότητα του να διενεργεί υπολογισμούς ενώ βρίσκεται σε κβαντική υπέρθεση, δηλαδή σε πολλές καταστάσεις ταυτόχρονα.Αξίζει όμως να σημειωθεί ότι ο κβαντικός υπολογιστής δεν επηρεάζει την κλάση των υπολογίσιμων συναρτήσεων, παρά μόνο υπολογίζει αποδοτικότερα ορισμένες συναρτήσεις σε σχέση με έναν κλασικό υπολογιστή.

Η ιδέα του κβαντικού υπολογιστή πρωτοεμφανίστηκε την δεκαετία του 80 από τον *Feynman* ο οποίος παρατήρησε ότι δεν μπορούσε να προσομοιώσει αποδοτικά κβαντικά συστήματα με συμβατικούς υπολογιστές και μάλιστα ότι η επιβράδυνση της προσομοίωσης αυξανόταν εκθετικά με το μέγεθος του συστήματος που έπρεπε να προσομοιωθεί. Έτσι λοιπόν πρότεινε τον κβαντικό υπολογιστή, έναν υπολογιστή ο οποίος θα βάσιζε την λειτουργία του ακριβώς σε αυτές τις φυσικές διαδικασίες που ήταν δύσκολο να προσομοιωθούν. Προς επιβεβαίωση του *Feynman*, ο *Deutsch* παρουσίασε τους πρώτους κβαντικούς αλγορίθμους οι οποίοι έδειχναν την υπεροχή του κβαντικού υπολογιστή.

Η μεγάλη έκρηξη στο ενδιαφέρον της επιστημονικής κοινότητας έγινε το 1994 με την ανακάλυψη του αλγορίθμου παραγοντοποίησης του *Shor*.Ο αλγόριθμος αυτός καταφέρνει να λύσει το πρόβλημα της παραγοντοποίησης μεγάλων αριθμών, όπως και το πρόβλημα του διακριτού λογάριθμου, σε πολυωνυμικό χρόνο. Τα δύο αυτά προβλήματα δεν έχει αποδειχθεί ότι δεν λύνονται σε πολυωνυμικό χρόνο με κλασικούς αλγόριθμους αλλά αυτή είναι η κοινή πεποίθηση των επιστημόνων μέχρι στιγμής. Η μεγάλη σημασία του αλγορίθμου του *Shor* προκύπτει από το γεγονός ότι η αξιοπιστία του διάσημου κρυπτοσυστήματος *RSA* (δημοσίου κλειδιού) το οποίο έχει σχεδιαστεί για μυστικές επικοινωνίες βασίζεται στην υπόθεση ότι η παραγοντοποίηση μεγάλων ακεραίων αποτελεί ένα δυσεπίλυτο (intractable) πρόβλημα. Ο *Shor* έδειξε ότι αυτό δεν ισχύει αν κάποιος καταφέρει να κατασκευάσει ένα κβαντικό υπολογιστή. Επομένως μυστικές υπηρεσίες, κυβερνήσεις και οποιοσδήποτε που ασχολείται με την κρυπτογραφία έχει συμφέρον να εφευρεθούν λειτουργικοί κβαντικοί υπολογιστές - κβαντικοί υπολογιστές μεγάλης κλίμακας.

Επόμενος σημαντικός σταθμός ήταν το 1996 που ανακαλύφθηκε ο αλγόριθμος κβαντικής αναζήτησης του *Lov Grover*. Με αυτό τον αλγόριθμο είναι δυνατόν να βρεθεί ένα στοιχείο σε μια αταξινόμητη λίστα μήκους σε χρόνο.Παρόλο που δεν παρουσιάζει αυτός ο αλγόριθμος εκθετική επιτάχυνση όπως ο αλγόριθμος

παραγοντοποίησης παρουσιάζει την απόδειξη ότι οι κβαντικοί υπολογιστές είναι ισχυρότεροι σε ορισμένες περιπτώσεις από τους κλασικούς.

Ακόμα από το 1998,οι εξελίξεις στον χώρο των κβαντικών υπολογιστών αφορούν κυρίως στην κατασκευή του υλικού από το οποίο θα αποτελούνται. Έτσι το 1998 κατασκευάστηκαν οι πρώτοι κβαντικοί υπολογιστές 2 και 3 *qubit* και εκτελέστηκε ο αλγόριθμος του *Grover*. Επόμενος σταθμός είναι το 2001 που έγινε η πρώτη εκτέλεση του αλγορίθμου του *Shor* σε κβαντικό υπολογιστή στο ερευνητικό κέντρο της *IBM* στο Almaden και στο πανεπιστήμιο του Stanford. Ο αριθμός 15=3×5 παραγοντοποιήθηκε χρησιμοποιώντας 10 πανομοιότυπα μόρια. Από τότε μέχρι σήμερα εκατοντάδες ερευνητικά κέντρα προσπαθούν να κατασκευάσουν αξιόπιστους κβαντικούς υπολογιστές χρησιμοποιώντας διαφορετικά υλικά.Η μεγαλύτερη δυσκολία ανακύπτει από δύο αντικρουόμενες προϋποθέσεις. Από τη μία η μνήμη ενός υπολογιστή η οποία αποτελείται από μικροσκοπικά κβαντικά συστήματα πρέπει να απομονωθεί όσο τέλεια γίνεται για να προστατευθεί από καταστροφική αλληλεπίδραση με το περιβάλλον. Από την άλλη η «κβαντική κεντρική μονάδα επεξεργασίας» δεν πρέπει να είναι εντελώς απομονωμένη, αφού οι υπολογισμοί πρέπει να είναι συνεχείς, και ένας «ελεγκτής» πρέπει να ελέγχει ότι το κβαντικό σύστημα εξελίσσεται με το ζητούμενο τρόπο. Μάλιστα το πρόβλημα ότι ανεξέλεγκτα σφάλματα είναι δυνατόν να προκύψουν κατά τη διάρκεια των υπολογισμών δεν είναι καινούριο:στην κλασική θεωρία πληροφορίας θεωρούμε συχνά ένα θορυβώδες κανάλι το οποίο μπορεί να αλλοιώσει τα μηνύματα.Η δουλειά του παραλήπτη είναι να εξάγει την σωστή πληροφορία από το παραμορφωμένο μήνυμα χωρίς επιπλέον μετάδοση πληροφορίας. Η κλασική θεωρία πληροφορίας ασχολείται με αυτό το πρόβλημα και το συμπέρασμα που προκύπτει χάρη στην εργασία του *Claude Shannon* είναι το εξής: Για ένα σχετικά θορυβώδες κανάλι υπάρχει ένα σύστημα κωδικοποίησης των μηνυμάτων το οποίο μας επιτρέπει να μειώσουμε την πιθανότητα σφάλματος στη μετάδοση όσο θέλουμε.

Αρχικά ήταν κοινή η άποψη ότι ένα αντίστοιχο μοντέλο ήταν αδύνατο για τους κβαντικούς υπολογισμούς ακόμη και σε θεωρητικό επίπεδο, κυρίως εξαιτίας του θεωρήματος «Μη Αντιγραφής» , το οποίο έλεγε ότι η κβαντική πληροφορία είναι αδύνατον να διπλασιαστεί με ακρίβεια. Παρόλα αυτά το 1995 ο *Shor* έδειξε ότι μπορούμε να κατασκευάσουμε σχήματα διόρθωσης λαθών για τους κβαντικούς υπολογιστές, και άρα θεμελίωσε την θεωρία των κβαντικών κωδικών διόρθωσης σφαλμάτων. Αυτή η θεωρία είναι ακόμα αντικείμενο μελετών και ίσως οδηγήσει μια μέρα στην κατασκευή κβαντικών υπολογιστών μεγάλης κλίμακας.

1.2 Ιστορία των γλωσσών κβαντικού προγραμματισμού

Οι κβαντικοί αλγόριθμοι συνηθίζεται να μελετούνται είτε σε πολύ χαμηλό επίπεδο υπό την μορφή κβαντικών πυλών και κυκλωμάτων είτε σε πολύ υψηλό επίπεδο με την χρήση γραμμικής άλγεβρας. Ο σκοπός μια γλώσσας κβαντικού προγραμματισμού είναι να γεφυρώσει το χάσμα αυτό προσφέροντας στον προγραμματιστή έναν ενδιάμεσο τρόπο περιγραφής των αλγορίθμων.Αν κάποια στιγμή ο κβαντικός υπολογιστής γίνει πραγματικότητα και διαδοθεί ευρέως τότε δεν μπορούμε σε καμία περίπτωση να απαιτήσουμε από τους προγραμματιστές να προγραμματίζουν σχεδιάζοντας κυκλώματα ή να διαθέτουν εκτεταμένες γνώσεις γραμμικής άλγεβρας. Αντίθετα αυτό το οποίο επιθυμούμε είναι να τους προσφέρουμε μια γλώσσα κβαντικού προγραμματισμού η οποία θα θυμίζει τις κλασικές γλώσσες προγραμματισμού τόσο ως προς τος συντακτικό όσο και ως προς την σημασιολογία.Το έργο αυτό δεν είναι εύκολο καθώς το κβαντικό μοντέλο υπολογισμού είναι εκ φύσεως διαφορετικό από το κλασσικό μοντέλο υπολογισμού και επιβάλλει περιορισμούς τους οποίους δεν συναντάμε στο κλασσικό μοντέλο υπολογισμού.

Ακόμα συνηθίζεται οι κβαντικοί αλγόριθμοι να περιγράφονται με κβαντικά κυκλώματα και πύλες. Αυτή η προσέγγιση είναι αναμενόμενη αφού οι επιτρεπόμενες διεργασίες στα κβαντικά *bit* περιγράφονται στο χαμηλότερο επίπεδο. Ο κβαντικός υπολογισμός είναι ένας ιδιαίτερος τρόπος σκέψης σε αντίθεση με τον κλασικό υπολογισμό ο οποίος είναι συμβατός με την κοινή λογική.Οι μόνες διεργασίες που μπορεί να κάνει κάποιος με τα κβαντικά bits είναι η αναδιάταξη τους, η εφαρμογή ενός ορθομοναδιαίου μετασχηματισμού και η μέτρηση τους. Αυτοί οι μετασχηματισμοί υλοποιούνται με κβαντικές πύλες και κυκλώματα και άρα οι αλγόριθμοι θα υλοποιούνται με τον ίδιο τρόπο.

Το μειονέκτημα όμως αυτής της προσέγγισης είναι ότι μειώνεται η εκφραστικότητα του κβαντικού προγραμματισμού και η ευκολία κατασκευής νέων αλγορίθμων. Για παράδειγμα είναι πολύ δύσχρηστος ο προγραμματισμός σε *assembly* ενώ πολύ ευκολότερος σε υψηλότερου επιπέδου γλώσσες προγραμματισμού. Για αυτό το λόγο αναπτύχθηκαν οι πρώτες γλώσσες κβαντικού προγραμματισμού. Το κοινό χαρακτηριστικό τους είναι ότι χειρίζονται ένα η περισσότερα κβαντικά *bits* μαζί με τα επιπλέον κλασικά *bits* ώστε να μπορούν να υλοποιήσουν και κλασικούς αλγόριθμους. Χωρίζονται όμως σε διάφορες κατηγορίες.

Μια πρώτη κατηγοριοποίησή τους είναι σε τρία υποσύνολα παρόμοια με τα αντίστοιχα των συμβατικών γλωσσών. Αυτά είναι οι προστακτικές γλώσσες κβαντικού προγραμματισμού, οι συναρτησιακές γλώσσες και όλες οι υπόλοιπες. Αρχικά αναπτύχθηκαν οι προστακτικές γλώσσες των οποίων τα προγράμματα αποτελούνται από μια ακολουθία εντολών προστακτικής φύσεως κατα αναλογία με τις αντίστοιχες γλώσσες του συμβατικού υπολογισμού. Ακόμη και σήμερα βέβαια οι περισσότεροι αλγόριθμοι ακολουθούν υποσυνείδητα αυτό το μοντέλο και συνεχίζεται η ανάπτυξη τους. Αργότερα εμφανίστηκε η μεγάλη τάση των συναρτησιακών γλωσσών οι οποίες ταιριάζουν αρκετά καλά στο κβαντικό μοντέλο καθώς κάθε αλγόριθμος πρόκειται ουσιαστικά για συνεχείς μετασχηματισμούς πάνω σε μια ποσότητα κβαντικής πληροφορίας. Τέλος στην τρίτη κατηγορία ανήκουν γλώσσες διαφορετικών μορφών.

Εκτός από αυτήν την κατηγοριοποίηση υπάρχει και ο χωρισμός των κβαντικών γλωσσών σε δύο τάξεις: αυτές που επιτρέπουν μόνο κλασική ροή ελέγχου και αυτές που επιτρέπουν και κβαντική ροή ελέγχου. Η πρώτη και απλούστερη αποτελείται από γλώσσες οι οποίες επιτρέπουν την κβαντική υπέρθεση μόνο στην κατάσταση των *qubits* (κβαντικά *bit*). Αντίθετα ο έλεγχος του προγράμματος γίνεται με κλασικό τρόπο. Συνοπτικά η υπέρθεση είναι η ικανότητα των κβαντικών συστημάτων να βρίσκονται σε περισσότερες από μία καταστάσεις ταυτόχρονα. Στο πρώτο είδος γλωσσών οι οποίες αναφέρονται ως "*quantum data, classical control paradigm*" τα προγράμματα ακολουθούν μια συγκεκριμένη ροή καιαπλώς επεξεργάζονται κβαντικά *bits*. Η δεύτερη κατηγορία είναι οι γλώσσες που επιτρέπουν και κβαντικό έλεγχο: "quantum data and control paradigm". Σε αυτές το πρόγραμμα μπορεί να βρίσκεται σε υπέρθεση δύο δυνατών μονοπατιών υπολογισμού. Φυσικά και τα δεδομένα που χειρίζονται μπορούν να είναι σε υπέρθεση. Η κβαντική μηχανή *Turing* του *Deutsch* λειτουργούσε σύμφωνα με αυτό το μοντέλο.

Η ιστορία των γλωσσών κβαντικού προγραμματισμού ήταν αρχικά συνυφασμένη με τα μοντέλα που ανακαλύφθηκαν. Όπως είδαμε η κβαντική μηχανή *Turing (QTM)* του *Deutsch* εφευρέθηκε το 1985. Αυτή είχε μια στοιχειώδη ψευδογλώσσα - οδηγίες προστακτικής μορφής. Το 1993 οι *Bernstein* και *Vazirani* εκτός από την βελτίωση της μηχανής πρότειναν και κάποια προγραμματιστικά δομικά στοιχεία. Πιθανότατα όμως η πρώτη πρόταση για μια σαφώς ορισμένη κβαντική γλώσσα προγραμματισμού, σε αντίθεση με τις περιγραφές *QTM* μηχανών, ήρθε το 1996 με την εργασία του *Knill*. Αυτός ορίζει έναν προστακτικό ψευδοκώδικα ο οποίος έτρεχε σε ένα μοντέλο κβαντικής μηχανής τυχαίας πρόσβασης *(Quantum Random Access Machine – QRAM)* αντίστοιχο με το κλασικό μοντέλο *RAM*. Ο *Knill* κατανοεί ότι ο κβαντικός ψευδοκώδικας δεν αποτελεί από μόνος του μια υλοποιήσιμη κβαντική γλώσσα αλλά είναι ένα σημαντικό βήμα εμπρός σε σχέση με τις εκ των υστέρων περιγραφές του πως πρέπει να υλοποιούνται οι κβαντικοί υπολογισμοί. Κατά τη διάρκεια μιας περιόδου αρκετών χρόνων (1998, 2000, 2001, 2002, 2003) ο *Ömer* ανέπτυξε την *QCL*, την πρώτη πραγματική κβαντική γλώσσα προγραμματισμού, με σύνταξη αρκετά

παρόμοια της *C*. Μάλιστα υλοποίησε και ένα λειτουργικό προσομοιωτή της γλώσσας. Η *QCL* περιέχει μία πλήρη κλασική γλώσσα προγραμματισμού ως υποσύνολο και παρέχει ένα σύνολο χρήσιμων κβαντικών δομών υψηλού επιπέδου όπως διαχείριση μνήμης και αυτόματη παραγωγή ελεγχόμενων εκδόσεων τελεστών. Οι *Sanders & Zuliani* (2000) και Zuliani (2001) ορίζουν την προστακτική γλώσσα *qQCL,* η οποία βασίζεται σε μια αυστηρή γλώσσα εντολών (*guarded-command language).*

Ο *Van Tonder* (2004) όρισε επίσης ένα κβαντικό λ-λογισμό, , ο οποίος είναι μια γλώσσα αγνού κβαντικού προγραμματισμού, δηλαδή δεν επιτρέπονται μετρήσεις. Οι *Valiron(*2004) και οι *Valiron&Selinger* (2005 ,2006) ορίζουν μια συναρτησιακή γλώσσα υψηλού επιπέδου, την *QPL*, η οποία ακολουθεί το σχήμα «κβαντικών δεδομένων και κλασικού ελέγχου». Η γλώσσα βασίζεται στο λ-λογισμό κλήσης κατά τιμή και περιλαμβάνει τόσο κλασικά όσο και κβαντικά δεδομένα. Υπάρχει ένα γραμμικό σύστημα τύπων και αποδεικνύονται ιδιότητες διατήρησης και ασφάλειας τύπων. Οι *Altenkirch* και *Grattage* (2005) μία πρώτης τάξεως συναρτησιακή γλώσσα, την *QML*, στην οποία η ροή του ελέγχου όπως και τα δεδομένα μπορούν να είναι κβαντικά. Η σημασιολογία της *QML* ακολουθεί το παράδειγμα του *Selinger* (2004) με όρους υπερτελεστών και πινάκων πυκνότητας και μία μετάφραση σε κβαντικά κυκλώματα. Ακολουθεί και αυτή γραμμικό σύστημα τύπων. Μετεξέλιξη της *QML* αποτελεί η γλώσσα *nQML* των *Lampis, Ginis, Papakyriakou, Papaspyrou* (2006). Επιτρέπει νέες δομές ελέγχου και είναι απλούστερη χωρίς να χάνει σε εκφραστικότητα. Η σημασιολογία της μέχρι στιγμής αποτελείται μόνο από τελεστές σε πίνακες πυκνότητας και όχι από κβαντικά κυκλώματα.

Άλλες κατηγορίες γλωσσών προγραμματισμού για παράδειγμα οι *Gay&Nagarajan* (2005) ορίζουν το λογισμό διαδικασιών *CQP (Communicating Quantum Processes)* και οι *Jorand&Larire (*2004) την *QPAlg* (*Quantum Process Algebra).* Και οι δύο γλώσσες μπορούν να περιγράψουν συστήματα τα οποία συνδυάζουν κλασικό και κβαντικό υπολογισμό και επικοινωνία και ο στόχος τους είναι να υποστηρίξουν τον φορμαλισμό του ορισμού και της επαλήθευσης κβαντικών κρυπτογραφικών πρωτοκόλλων.

1.3 Δομή της εργασίας

• Στο *κεφάλαιο 2* δίνουμε ένα μαθηματικό μοντέλο για τους κβαντικούς υπολογισμούς το οποίο αποτελεί τη βάση της υπόλοιπης εργασίας. Επίσης εξηγούμε πώς τα κβαντικά κυκλώματα συνδέονται με αυτό το μοντέλο.
• Στο *κεφάλαιο 3* είναι μια περιληπτική περιγραφή δύο γλωσσών κβαντικού προγραμματισμού προγενέστερων της nQML. Πρόκειται για μια εισαγωγή σε γλώσσες οι οποίες βρίσκονται πιο κοντά στο κλασικό μοντέλο προγραμματισμού ώστε να γίνει ευκολότερη η μετάβαση στην nQML.
•Στο *κεφάλαιο 4* παρουσιάζεται η γλώσσα *nQML* και εξηγείται το σύστημα τύπων και η σημασιολογία της, τα οποία είναι απαραίτητα για την κατασκευή κυκλωμάτων. Ακόμα δίνονται κάποια χαρακτηριστικά παραδείγματα κβαντικών αλγορίθμων μαζί με τις υλοποιήσεις τους σε *nQML*.Όπω ο αλγόριθμος του *Deutsch*, ο αλγόριθμος του *Grover* και ο αλγόριθμος του *Shor.*
• *Στο κεφάλαιο 5* γίνεται η μετάφραση των εντολών της nQML σε κυκλώματα, που είναι και ο στόχος της εργασίας. Αρχικά ορίζεται η κλάση των κυκλωμάτων που θα δημιουργηθούν από την γλώσσα και έπειτα σχολιάζεται η μετατροπή κάθε εντολής ξεχωριστά.

Κεφάλαιο 2

Κβαντικοί υπολογισμοί-Κβαντικα κυκλώματα

2.1 Εισαγωγή

Οι κβαντικοί υπολογισμοί στηρίζονται στις αρχές της κβαντομηχανικής οι οποίες είναι αντίθετες προς την κοινή λογική και εμπειρία. Εδώ θα αναφερθούμε επιγραμματικά στις αρχές οι οποίες είναι σημαντικές για τους κβαντικούς υπολογισμούς. Οι θεμελιώδεις αρχές της κβαντομηχανικής μπορούν να διατυπωθούν πολύ απλά και συνοπτικά.Το ζήτημα είναι η κατανόησή και η εφαρμογή τους. Οι βασικότερες είναι οι εξής:

- Η αρχή της υπέρθεσης
- Η αρχή της μέτρησης
- Η αρχή της ορθομοναδιαίας

2.2 Αρχή της υπέρθεσης

Η αρχή της υπέρθεσης είναι αυτή που δίνει στους κβαντικούς αλγορίθμους συγκριτικό πλεονέκτημα έναντι των κλασικών.Στους κβαντικούς υπολογιστές η μονάδα πληρογορίας είναι το *Quantum Bit* ή συντομογραφικά το *Qubit*.Το *qubit* είναι ουστιαστικά η μονάδα πληροφορίας των κβαντικών υπολογιστών και είναι ένα κβαντικό σύστημα δύο καταστάσεων που αντίστοιχα συμβολίζονται με $|0\rangle$και $|1\rangle$. Επίσης μπορεί να αποδοθούν και με την μορφή:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} = |+\rangle \text{ και } |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} = |-\rangle$$

Το *qubit* μπορεί να βρεθεί σε κατάσταση υπέρθεσης μεταξύ των δύο βασικών καταστάσεων και να πάρει οποιεσδήποτε τιμές καθώς το *qubit* είναι ένα διάνυσμα στον χώρο *Hilbert (*ο χώρος *Hilbert* είναι ένας διανυσματικός χώρος πάνω στο πεδίο των μιγαδικών αριθμών και ουσιαστικά είναι χώρος που παίρνει άπειρες διαστάσεις). Δηλαδή να έχουμε ταυτοχρόνως και την κατάσταση 0 και την κατάσταση *1*.Η έννοια

της υπέρθεσης ή αρχή της επαλληλίας, αφορά όλα τα κύματα (ή κβαντικά συστήματα) που επιλύνονται μέσω των γραμμικών εξισώσεων και είναι μια έννοια θεμελιώδης στην κβαντική φυσική. Σύμφωνα με αυτήν, το ολικό αποτέλεσμα ενός κβαντικού φαινομένου που αποτελείται από επί μέρους φαινόμενα, είναι ίσο με το άθροισμα των επί μέρους αποτελεσμάτων. Κάθε κβαντικό σύστημα μπορεί να βρεθεί σε καταστάσεις υπερθέσεως δύο βασικών καταστάσεων και θα αποδίδεται από την μαθηματική έκφραση που εμπερικλείει τα πλάτη πιθανότητας.Συγκεκριμένα έστω ένα άτομο υδρογόνου που βρίσκεται στην θεμελιώδη κατάσταση, και το 0 αντιπροσωπεύεται από την ηλεκτρονιακή κατάσταση $|+\rangle$ και το *1* από την $|-\rangle$.Τότε κάθε δυνατή κατάσταση θα είναι γραμμικός συνδυασμός της εξής μορφής:

$$|\psi\rangle=\alpha|+>\beta|.>$$

2.3 Αρχή της κβαντικής μέτρησης

Η δεύτερη αρχή είναι η αρχή της κβαντικής μέτρησης. Σύμφωνα με αυτή την αρχή κατά την μέτρηση ενός κβαντικού συστήματος η κυματοσυνάρτηση «καταρρέει» και το σύστημα μεταβαίνει σε μία συγκεκριμένη κατάσταση η οποία είναι και το αποτέλεσμα της μέτρησης. Οποιεσδήποτε άλλες μετρήσεις του ίδιου μεγέθους στο σύστημα θα δώσουν το ίδιο αποτέλεσμα. Ουσιαστικά δηλαδή από τη μέτρηση και μετά το *qubit* συμπεριφέρεται κλασικά. Φυσικά εδώ πρέπει να επισημάνω ότι το *qubit* δεν σταμάτησε να είναι κβαντικό σύστημα, απλώς το συγκεκριμένο μέγεθος του απέκτησε μια ορισμένη τιμή. Σύμφωνα με την αρχή της απροσδιοριστίας του *Heisenberg* άλλες ιδιότητες του παραμένουν απροσδιόριστες ή αλλιώς σε υπέρθεση.Οι δύο παραπάνω αρχές έχουν ελεγχθεί πειραματικά και κανένα πείραμα δεν τις έχει θέσει υπό αμφισβήτηση. Από την πρώτη πηγάζουν τα πλεονεκτήματα των κβαντικών υπολογιστών. Ένας υπολογιστής που βρίσκεται σε πολλές καταστάσεις ταυτόχρονα αυξάνει το δυνατό παραλληλισμό και μάλιστα στην περίπτωση των κβαντικών υπολογισμών έχουμε εκθετική αύξηση. Ένα *qubi*t το οποίο είναι ταυτόχρονα *0* και *1* θα μας δώσει στο τέλος των υπολογισμών ένα αποτέλεσμα το οποίο είναι υπέρθεση των αποτελεσμάτων που θα είχαμε πάρει αν το *qubit* ήταν *0* ή *1*. Ομοίως η πραγματοποίηση ενός υπολογισμού n *qubit* θα μας δώσει υπέρθεση 2^n αποτελεσμάτων. Η δεύτερη αρχή όμως μειώνει κατά πολύ την αξία αυτού του παραλληλισμού και προκαλεί τις δυσκολίες στο σχεδιασμό κβαντικών αλγορίθμων. Σύμφωνα με αυτή είναι αδύνατο να μετρήσουμε όλες τις καταστάσεις στις οποίες βρίσκεται ο υπολογιστής μας αλλά μόνο μία,όλες οι υπόλοιπες καταστρέφονται.

2.4 Κβαντικά bits

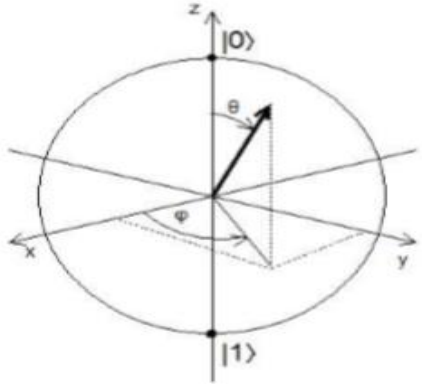
Στους κλασσικούς υπολογιστές βασική έννοια είναι το γνωστό συντομογραφικά, *bit (binary digit*). Το bit αντιπροσωπεύει την μονάδα πληροφορίας και είναι το δυαδικό ψηφίο *0* ή *1*, δηλαδή μπορεί να πάρει μόνο δύο τιμές την *0* και την *1*.Η πρωταρχική μονάδα δεδομένων ενός κβαντικού υπολογιστή είναι το *qubit.* Το κλασικό *bit* μπορεί να βρίσκεται όπως ξέρουμε στην κατάσταση *0 ή 1*.Αντίθετα το κβαντικό *bit* μπορεί να βρίσκεται σε οποιαδήποτε κατάσταση της μορφής $q = \alpha_0 + \beta_0$ όπου *a,b*$\in \mathbb{C}$ και δεν είναι ταυτόχρονα *0*.Δύο καταστάσεις *q* και *q'* για τις οποίες ισχυέι *q=γq'* είναι ισοδύναμες . Συνήθως θεωρούμε την κανονικοποίηση $\alpha^2 + \beta^2 = 1$ οπότε όλες οι ισοδύναμες καταστάσεις αντιπροσωπεύονται από μια μόνο. Τα *α,β* είναι μηγαδικοί αριθμοί και φυσικά δεν έχουν άμμεση φυσική σημασία.Με την κανονικοποίηση όμως το τετράγωνο του μέτρου τους μας δίνει την πιθανότητα να μετρήσουμε την αντίστοιχη κατάσταση.Δηλαδή σε μια μέτρηση έχουμε α^2 πιθανότητα να μετρήσουμε *1* και β^2 να μετρήσουμε *0*.

Για παράδειγμα οι καταστάσεις $\frac{1}{\sqrt{2}}1 + \frac{1}{\sqrt{2}}0$ *και* $\frac{1}{\sqrt{2}}1 - \frac{1}{\sqrt{2}}0$.Μόνο με περεταίρω επεξεργασία τους μπορούμε να ξεχωρίσουμε την μια κατάσταση απο την άλλη και όχι με μετρήσεις.Μάλιστα η πρώτη μέτρηση στην κάθε κατάσταση θα καταστρέψει την υπέρθεση και όλη η πληροφορία για τα α,β θα χαθεί.Επομένως κάθε κατάσταση ενός ορίζεται μονοσήμαντα από ένα ζεύγος μιγαδικών αριθμών και άρα πρόκειται για διανύσματα στο χώρο $\mathbb{C} \times \mathbb{C}$.Η βάση αυτού του χώρου είναι τα διανύσματα $\binom{1}{0}$ και $\binom{0}{1}$ τα οποία συμβολίζουν τις καταστάσεις *0* και *1* αντίστοιχα.Αυτές οι καταστάσεις ονομάζονται κλασσικές καταστάσεις καθώς ένα κλασσικό βρίσκεται σε μια από τις δύο.

2.5 Αναπαράσταση κβαντικού bit

Ο χώρος απεικόνισης των καταστάσεων γίνεται πάνω στο πεδίο των μιγαδικών αριθμών και ουσιαστικά είναι χώρος που παίρνει άπειρες διαστάσεις. Αυτός ο χώρος καλείται χώρος *Hilbert* και δεν είναι άλλος από ένα διανυσματικό χώρο στον οποίο θα περιοριστούμε για την παρουσίασή των κβαντικών υπολογιστών σε διδιάστο ή τρισδιάστατο περιβάλλον ώστε σε αυτόν να έχουμε γεωμετρικά ζεύγος διανυσμάτων που θα αναπαριστά δύο ή περισσότερα σωματίδια (φωτόνια, ηλεκτρόνια κ.τ.λ.). Επιτυγχάνεται έτσι η γενίκευση της έννοιας του Ευκλείδιου χώρου. Στους κβαντικούς υπολογιστές και την κβαντική θεωρία η γεωμετρική αναπαράσταση των κβαντικών καταστάσεων γίνεται με την σφαίρα *Bloch* η οποία είναι μια γεωμετρική αναπαράσταση του χώρου κατάστασης ενός κβαντικού συστήματος δύο καταστάσεων και απεικονίζει τα ιδιαίτερα χαρακτηριστικά των κβαντικών *bits.* Η ακτίνα της σφαίρας *Bloch* είναι μονάδα όσο και το μήκος του διανύσματος κατάστασης του qubit, ενώ οι τιμές που παίρνει η γωνία θ είναι μεταξύ 0^o και 180^o . Οι δύο πραγματικές παράμετροι ϑ και φ είναι αρκετές για να περιγράψουν ένα διάνυσμα κατάστασης, αφού τα διανύσματα κατάστασης πρέπει να έχουν μέτρο *1* και είναι ισοδύναμα όσον αφορά μία συνολική διαφορά φάσης. Τα σημεία στην σφαίρα Bloch μπορούν να εκφραστούν και σε καρτεσιανές συντεταγμένες :

$$(x, y, z) = (sin\,\theta\ \ cos\,\varphi\,, sin\,\theta\,sin\,\varphi\,, cos\,\theta\,)$$



Σχήμα 2.1 *Αναπαράσταση κβαντικού bit-Σφαίρα Bloch*

Μάλιστα αναφερόμαστε σε σφαίρα και όχι σφαιρική επιφάνεια επειδή στα εσωτερικά σημεία της *σφαίρας Bloch* αντιστοιχούν μικτές καταστάσεις ενός *qubit*. Τέλος πρέπει να προσεχθεί το γεγονός ότι η *σφαίρα Bloch* παριστάνει την κατάσταση ενός *qubit* μόνο και όχι καταχωρητών.

2.6 Κβαντικές πύλες

Οι κβαντικές πύλες δεν είναι φυσικά συστήματα, αλλά αντιπροσωπεύουν μαθηματικές δράσεις που ασκούνται σε qubits ή σε κβαντικούς καταχωρητές. Σε αντίθεση με τους κλασσικούς υπολογιστές που αποτελούνται από κυκλώματα, ολοκληρωμένα και αγωγούς χάραξης πάνω στο τυπωμένο κύκλωμα για να διέρχεται το ηλεκτρικό ρεύμα μέσω του οποίου μεταφέρεται η πληροφορία την οποία οι λογικές ψηφιακές πύλες την μετατρέπουν μετά από ψηφιακή επεξεργασία από είσοδο σε έξοδο (με τον πίνακα αληθείας ή το αντίστοιχο λογικό κύκλωμα), στους κβαντικούς υπολογιστές η πληροφορία δε διέρχεται μέσα από τις κβαντικές πύλες αλλά βρίσκεται αποθηκευμένη σε qubits ή σε κβαντικούς καταχωρητές και παραμένει εκεί. Οι κβαντικές πύλες είναι τελεστές (ορθομοναδιαίοι πίνακες) του χώρου *Hilbert* που δρουν σε *qubits* και σε κβαντικούς καταχωρητές αλλάζοντας την κατάστασή τους. Οι καταστάσεις των *qubits* και των κβαντικών καταχωρητών είναι διανύσματα στον χώρο *Hilbert.* Δηλαδή οι κβαντικές πύλες περιστρέφουν τα διανύσματα κατάστασης των *qubits*.Οι χώροι *Hilbert* προκύπτουν φυσικά και συχνά στα μαθηματικά και τη φυσική, συνήθως ως απειροδιάστατες λειτουργίες χώρων. Οι πρώτοι χώροι *Hilbert* μελετήθηκαν από την άποψη αυτή κατά την πρώτη δεκαετία του 20ου αιώνα από τους *Hilbert* και *Frigyes Riesz.* Είναι απαραίτητα εργαλεία στις θεωρίες των μερικών διαφορικών εξισώσεων, την κβαντομηχανική, την ανάλυση *Fourier(*η οποία περιλαμβάνει εφαρμογές για την επεξεργασία σήματος και τη μεταφορά θερμότητας) και την εργοδική θεωρία, η οποία αποτελεί τη μαθηματική υποστήριξη της θερμοδυναμικής.

Βασικές κβαντικές πύλες είναι οι εξής:

- Κβαντική πύλη αδράνειας
- Κβαντική πύλη *Hadamard*
- Κβαντική πύλη μετατόπισης φάσης
- Κβαντική πύλη άρνησης
- Κβαντική πύλη ελεγχόμενης άρνησης
- Κβαντική πύλη ελεγχόμενης μετατόπισης φάσης
- Κβαντική πύλη διπλά ελεγχόμενης άρνησης
- Κβαντικές πύλες *Pauli*
- Κβαντική πύλη εναλλαγής
- Κβαντική πύλη ελεγχόμενης εναλλαγής

Κβαντικές Πύλες	***Συμβολισμός***	***Πίνακας-Τελεστική Μορφή***
Αδράνειας	*I*	$I=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
Άρνησης	*NOT*	$NOT=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$
Ελεγχόμενου ΟΧΙ	*CNOT*	$CNOT=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$
Hadamard	*H*	$H=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
Μετατόπισης Φάσης	*Φ*	$\Phi=\begin{bmatrix}1 & 0\\ 0 & e^{i\varphi}\end{bmatrix}$

Διπλά ελεγχόμενου ΟΧΙ	*CCNOT*	$CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
Εναλλαγής	*SWAP*	$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Ελεγχόμενης Μετατόπισης Φάσης	*CΦ*	$C\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{bmatrix}$

Πίνακας 2.1 *Κβαντικές πύλες -Συμβολιμοί*

Οι κβαντικές Πύλες που δρούν σε ένα *Qubts* είναι οι τέσσερις πρώτες, τις οποίες μπορείτε να τις δείτε αναλυτικά με το σχεδιαστικό συμβολισμό τους και τον πίνακα αληθείας.

Σχεδιαστικός Συμβολισμός (Είσοδος – Δράση πύλης – Έξοδος)	«Πίνακας Αληθείας»
 Πύλη Αδρανείας	Input ($\|q_I\rangle$) / Output ($\|q_o\rangle$) $\|q_I\rangle = \|0\rangle$ → $\|q_o\rangle = \|0\rangle$ $\|q_I\rangle = \|1\rangle$ → $\|q_o\rangle = \|1\rangle$ $\|q_I\rangle = \|q\rangle$ → $\|q_o\rangle = \|q\rangle$

Σχεδιαστικός Συμβολισμός (Είσοδος – Δράση πύλης – Έξοδος)	«Πίνακας Αληθείας»
 Πύλη Μετατόπισης Φάσης	Input ($\|q_I\rangle$) / Output ($\|q_o\rangle$) $\|q_I\rangle = \|0\rangle$ → $\|q_o\rangle = \|0\rangle$ $\|q_I\rangle = \|1\rangle$ → $\|q_o\rangle = e^{i\varphi}\|1\rangle$ $\|q_I\rangle = \alpha\|0\rangle + \beta\|1\rangle$ → $\|q_o\rangle = \alpha\|0\rangle + \beta e^{i\varphi}\|1\rangle$

Πίνακας 2.2 *Σχεδιαστικός συμβολισμός και πίνακας αληθείας*

2.7 Κβατνικά Κυκλώματα

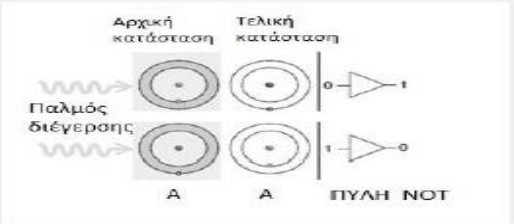
Στα κβαντικά κυκλώματα δεν μεταφέρεται η κβαντική πληροφορία από μια κβαντική πύλη σε κάποια άλλη (όπως δηλαδή συμβαίνει στους κλασσικούς υπολογιστές). Στα κβαντικά κυκλώματα, προκειμένου να γίνουν οι κβαντικοί υπολογισμοί, η πληροφορία παραμένει στους κβαντικούς καταχωρητές και στα *qubits.*

Έτσι λοιπον ένα κβαντικό κύκλωμα αποτελείται από ένα πεπερασμένο πλήθος από πύλες οι οποίες τροφοδοτούνταιμε *qubits* στις εισόδους τους και σα ναποτέλεσμα μετασχηματίζουντην κατάσταση αυτών των *qubits*.Όμως όπως είδαμε οι μετασχηματισμοί που λαμβάνουν χώρα είναι ορθομοναδιαίοι και συνεπώς αντιστρέψιμοι.Άρα πρέπει και οι πύλες που χρησιμοποιούμε να είναι αντιστρέψιμες και κατεπέκταση να έχουν τόσες εισόδους όσες και εξόδους.Αυτό εκπρώτης όψεως φαίνεται καταστροφικό αφού οι συνηθισμένες λογικές πύλες όπως *and,or,nand* κλπ δεν είναι αντιστρέψιμες.Θα δούμε στην συνέχεια πως μπορούμε να το ξεπεράσουμε αυτό ορίζοντας τις αντίστοιχες αντιστρέψιμες πύλες. Επιπλέον στα λογικά κυκλώματα είμαστε συνηθισμένοι να χρησιμοποιούμε διακλαδώσεις που αντιπροσωπεύουν την αντιγραφή της κλασικής πληροφορίας.Όμως στα κβαντικά κυκλώματα κάτι τέτοιο δεν υφίσταται εξαιτίας του θεωρήματος της μη κλωνοποίησης.

Ο σχεδιασμός των κβαντικών κυκλωμάτων γίνεται με ορισμένους συμβολισμούς κβαντικών πυλών και δείχνει την σειρά με την οποία δρουν οι κβαντικές πύλες στις κβαντικές καταστάσεις.Επίσης στα κβαντικά κυκλώματα δεν έχουμε διακλαδώσεις και βρόγχους. Το αποτέλεσμα ενός κβαντικού υπολογισμού που προκύπτει από το υπό περίπτωση μελέτης κβαντικό κύκλωμα θα είναι η τελική κατάσταση του κβαντικού καταχωρητή.Στους κβαντικούς υπολογιστές δεν μπορούμε να αντιγράψουμε την κατάσταση ενός *qubit*, δηλαδή δεν έχει βρεθεί (ακόμα τουλάχιστον) μια πύλη που να αντιγράφει την κατάσταση του *qubit.*

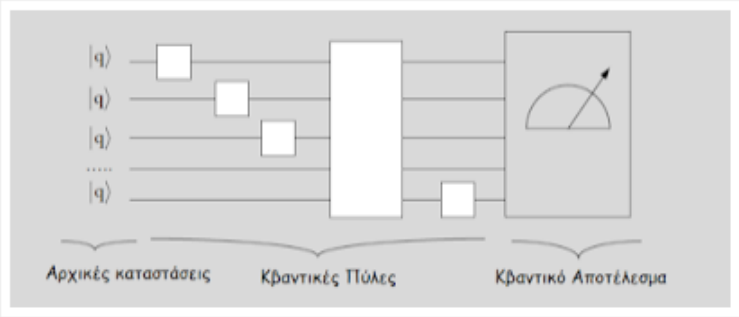
Όλες οι κλασσικές λογικές πύλες μπορούν να υλοποιηθούν και με χρήση των κβαντικών λογικών πυλών.Όπως δείχνει ο συμβολισμός στο σχήμα: εάν το *A* είναι *0*, δίνει *1* και αντιστρόφως αν το *A* είναι *1* δίνει *0*.Με χρήση ατόμων, αυτό μπορεί να γίνει με την εφαρμογή ένας παλμού του οποίου η ενέργεια ισούται με τη διαφορά μεταξύ της αρχικής κατάστασης των ηλεκτρονίων του ατόμου και της διεγερμένης, τελικής κατάστασης του.

Χαρακτηριστικό είναι το επόμενο σχήμα 2.2.



Σχήμα 2.2 *Πύλη Not*

Στο επόμενο σχήμα, απεικονίζεται ένα τυχαίο κβαντικό κύκλωμα που αποτελείται από κβαντικές πύλες συνδεδεμένες μεταξύ τους με ορθή διάταξη χωρίς ανάδραση και συνδέονται με κβαντικά σύρματα δηλαδή με αυτό που είναι ο δρόμος των *qubit,* τα οποία περιγράφουν τις κβαντικές καταστάσεις $|q\rangle$ των κβαντικών διανυσμάτων, στον χώρο του *Hilbert*.



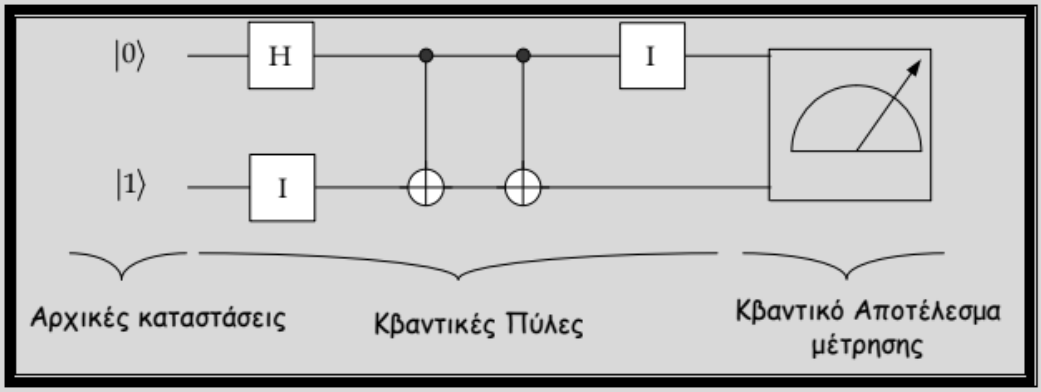
Σχήμα 2.3 *Υυχαίο κβαντικό κύκλωμα που αποτελείται από κβαντικές πύλες*

2.7.1 Παράδειγμα κβαντικού κυκλώματος

Παρακάτω θα δούμε ένα παράδειγμα υπολογισμού κβανικού κυκλώματο.

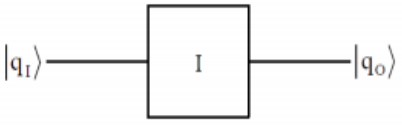
ΠΡΟΒΛΗΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΚΒΑΝΤΙΚΟΥ ΚΥΚΛΩΜΑΤΟΣ

Έστω το τυχαίο κβαντικό κύκλωμα:



Να υπολογίσετε το τελικό κβαντικό αποτέλεσμα μετά την επί μέρους δράση των προηγούμενων κβαντικών πυλών πάνω στις αρχικές κβαντικές καταστάσεις.

Πριν γίνει η μελέτη του κυκλώματος πρέπει να περιγραφούν οι κβαντικές πύλες που δρουν πάνω στα βήματα των κβαντικών δράσεων. Στο κύκλωμα περιλαμβάνονται οι πύλες *Hadamard,* αδράνειας και *CNOT*.Η κβαντική πύλη αδράνειας δρα σε ένα *qubit*,περιγράφei την δράση του τελεστή αδράνειας (ο οποίος αφινει αμετάβλητη την κατάσταση του *qubit* σύμφωνα με την σχέση $I|q\rangle = |q\rangle$.Η κβαντική πύλη αδράνειας περιγράφεται από τον πίνακα $I = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$.Η πληροφορία διέρχεται και μεταβιβάζεται μέσα από την πύλη.Συμβολίζουμε την κατάσταση του w=qubit πριν τη δράση του τελεστή με $|q_1\rangle$ *(input).*Ενώ μετά την δράση την συμβλίζουμε με $|q_0\rangle$ (*output*).Οι χαρακτηριστηκές ιδιότητες και ο συμβολισμός της φαίνεται στον επόμενο πίνακα αλήθειας (παρατηρείται οτι καταστάσεις του *qubit* είναι αμετάβλητες).

Σχεδιαστικός Συμβολισμός *(Είσοδος - Δράση πύλης - Έξοδος)*	«Πίνακας Αληθείας»	
 Πύλη Αδρανείας	Input ($\|q_I\rangle$)	Output ($\|q_o\rangle$)
	$\|q_I\rangle = \|0\rangle$	$\|q_o\rangle = \|0\rangle$
	$\|q_I\rangle = \|1\rangle$	$\|q_o\rangle = \|1\rangle$
	$\|q_I\rangle = \|q\rangle$	$\|q_o\rangle = \|q\rangle$

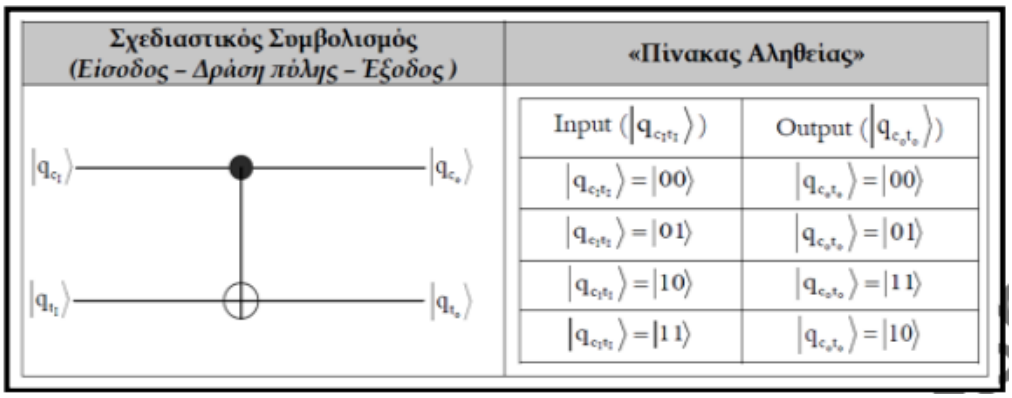
• Η κβαντική πύλη *hadamard* δρα σε *qubits* που βρίσκονται σε μια από τις δύο βασικές καταστάσεις και τα θέτει σε μια κατάσταση που είναι υπέρθεση των βασικών καταστάσεων.Επίσης όταν δρα σε *qubits* που

βρίσκονται στην υπέρθεση καταστάσεων επιστρέφει τα qubits στις βασικές τους καταστάσεις.Συμβολίζεται με Η και περιγράφεται απο τον ακόλουθο πίνακα $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$.Ο συμβολισμός της και ο πίνακας αλήθειας που αποδίδει τις χαρακτηριστικές ιδιότητες της πύλης,απεικονίζεται στον επόμενο πίνακα.

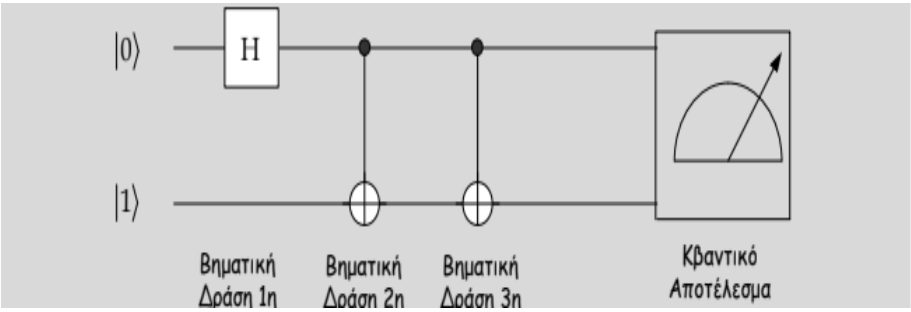
Σχεδιαστικός Συμβολισμός *(Είσοδος – Δράση πύλης – Έξοδος)*	**«Πίνακας Αληθείας»**	
$\|q_I\rangle$ — H — $\|q_O\rangle$ **Πύλη HADAMARD**	Input ($\|q_I\rangle$)	Output ($\|q_O\rangle$)
	$\|q_I\rangle = \|0\rangle$	$\|q_O\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$
	$\|q_I\rangle = \|1\rangle$	$\|q_O\rangle = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$
	$\|q_I\rangle = \frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	$\|q_O\rangle = \|0\rangle$
	$\|q_O\rangle = \frac{1}{\sqrt{2}}(\|0\rangle - \|1\rangle)$	$\|q_O\rangle = \|1\rangle$

• Η κβαντικη πύλη ελεγχόμενου *OXI* δρα σε δύο *qubit*(εκ των οποίων το ένα καλείται ελέγχου-controlled και το άλλο καλείται στόχος target και συμβολίζονται αντίστοιχα με *c* και *t*).Η κβαντική πύλη ελεγχόμενου *OXI* συμβλίζεται με *CNOT (Controlled NOT*) και περιγράφεται από τον πίνακα

$CNOT = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$.Η λειτουργία της πύλής έγκειται στο ότι αλλάζει την κατάσταση του *qubit t* όταν η κατάσταση του *qubit c* είναι *1* ενώ όταν του *qubit c* είναι *0*,η κατάσταση του *qubit t* μένει ανάλλοίωτη.Ο συμβολισμός και ο πίνακας αληθείας που αποδίδει της πύλης μετατόπισης φάσης φαίνεται στον επόμενο πίνακα :



• Επίλυση του προβλήματος :Η πύλη αδράνειας επειδή δεν αλλάζει με την δράση της την κβαντική κατάσταση μπορούμε να την παραλείψουμε και να φαίνεται στο σχήμα η συνεχόμενη γραμμή του κβαντικού σύρματος.Επομένως μπορούμε ισοδύναμε να σχεδιάσουμε το κύκλωμα ως εξής:



Αρχικά ,βήμα 1ο ,ο κβαντικός καταχωρητής του κυκλώματος αποτελείται από δύο *qubits* και η κατάσταση του είναι η $|q\rangle$,δηλαδή η $|01\rangle$. $= \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$ ή $|01\rangle = 0|00\rangle + |01\rangle + 0|10\rangle + |11\rangle$ (η πιθανότητα να μετρήσουμε την κατάσταση $|01\rangle$,είναι *1*).

Στην συνέχεια προκύπτει $(H \otimes I)\ |01\rangle = \left(\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} \otimes \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\right)\begin{bmatrix}0\\1\\0\\0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\0 & 1 & 0 & 1\\1 & 0 & -1 & 0\\0 & 1 & 0 & -1\end{bmatrix}\begin{bmatrix}0\\1\\0\\0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\\0\\1\end{bmatrix}$ ή

$(H \otimes I)\ |01\rangle = 0|00\rangle + \frac{1}{\sqrt{2}}|01\rangle + 0|10\rangle + \frac{1}{\sqrt{2}}|11\rangle$

Στην συνέχεια ,βήμα 2ο,στην $(H \otimes I)$ δρα η πύλη *CNOT* ώστε:

$$CNOT\,\frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\\0\\1\end{bmatrix}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\cdot\frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\\0\\1\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\\1\\0\end{bmatrix}$$

Ομοίως και στο βήμα 3ο μια πύλη *CNOT* δρα στην $\frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\\1\\0\end{bmatrix}$ οπότε προκύπτει:

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\cdot\left(\frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\\0\\1\end{bmatrix}\right)=\frac{1}{\sqrt{2}}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\begin{bmatrix}0\\1\\1\\0\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\\0\\1\end{bmatrix}=0|00\rangle+\frac{1}{\sqrt{2}}|01\rangle+0|10\rangle+\frac{1}{\sqrt{2}}|11\rangle$$

η πιθανότητα να μετρήσουμε την κατάσταση $|01\rangle$ είναι *0.5* ή *50%* και την κατάσταση $|11\rangle$ είναι επίσης *0.5 ή 50%.*

Επομένως το κβαντικό αποτέλεσμα του υπολογισμού του κβαντικού κυκλώματος,θα δώσει πιθανότητα να βρεθεί ο κβαντικός καταχωρητής στις καταστάσεις $|01\rangle|11\rangle$,ίση με *0.5* ή με *50%* για κάθε μια από αυτές (ενώ *0%* για τις $|01\rangle, |01\rangle$ καταστάσεις).

2.8 Το θεώρημα της μη κλωνοποίησης

Ενώ τα αξιώματα της κβαντομηχανικής είναι φαινομενικά απλά,ορισμένες από τις συνέπειες τους εναντιώνονται στην ανθρώπινη διαίσθηση.Μια τέτοια συνέπεια είναι το θεώρημα της μη κλωνοποίησης.Το θεώρημα αυτό μας λέει όταν μας δίνεται ένα qubit το οποίο βρίσκεται σε μια τυχαία κατάσταση

$|\varphi\rangle=\alpha_0|0\rangle+\alpha_1|1\rangle$ είναι γενικά αδύνατο να δημιουργήσουμε ένα αντίγραφο του. Πιο συγκεκριμένα, το πρόβλημα μας είναι ο μετασχηματισμός της κατάσταση $|\varphi\rangle\otimes|0\rangle$ στην κατάσταση $|\varphi\rangle\otimes|\varphi\rangle$. Από το αξίωμα της ορθομοναδιαίας εξέλιξης γνωρίζουμε ότι αν κάτι τέτοιο είναι εφικτό,τότε θα πρέπει να πραγματοποιείται μετην επίδραση ενός ορθομοναδιαίου μετασχηματισμού U τέτοιου ώστε $U|\varphi\rangle\otimes|0\rangle=|\varphi\rangle\otimes|\varphi\rangle$. Άρα μέσω του μετασχηματισμού U για δύο τυχούσες και ανεξάρτητες καταστάσεις $|\varphi_1\rangle$, $|\varphi_2\rangle$ θα έχουμε:

$$|\varphi_1\rangle\otimes|0\rangle \xrightarrow{U} |\varphi_1\rangle\otimes|\varphi_1\rangle$$

$$|\varphi_2\rangle\otimes|0\rangle \xrightarrow{U} |\varphi_2\rangle\otimes|\varphi_2\rangle$$

Όμως τότε $\langle|\varphi_1\rangle|\varphi_2\rangle = \langle|\varphi_1\rangle\rangle\otimes|0\rangle , |\varphi_2\rangle\otimes|0\rangle\rangle = \langle|\varphi_1\rangle\rangle\otimes|\varphi_1\rangle , |\varphi_2\rangle\otimes|\varphi_2\rangle\rangle = \langle\varphi_1|\varphi_2\rangle^2 \Rightarrow \langle\varphi_1|\varphi_2\rangle = 0 \vee \langle\varphi_1|\varphi_2\rangle = 1$, όπου χρησιμοποιήσαμε το γεγονός ότι αφού ο U είναι τέτοιος ώστε $UU\dagger = U\dagger U = I$ διατηρεί το εσωτερικό γινόμενο. Δηλαδή τα$|\varphi_1\rangle$,$|\varphi_2\rangle$ είτε θα είναι συγγραμικά είτε θα είναι κάθετα μεταξύ τους.Άτοπο,καθώς υποθέσαμε ότι τα $|\varphi_1\rangle$,$|\varphi_2\rangle$ επιλέχθηκαν τυχαία.

Κεφάλαιο 3

Γλώσσες κβαντικού προγραμματισμού

3.1 Εισαγωγή

Μια γλώσσα υψηλού επιπέδου επιτρέπει στον προγραμματιστή να εκφράσει τις σκέψεις του και να εκμεταλλευτεί με καλύτερο τρόπο τις δυνατότητες που του δίνει το υλικό του υπολογιστή. Τα χαρακτηριστικά μιας γλώσσας είναι πολλά, όπως ο πλούτος και η ευκολία εκφραστικότητας. Ιδιαίτερα το δεύτερο παίζει σπουδαίο ρόλο στις γλώσσες κβαντικού υπολογισμού όπου οι έννοιες του μοντέλου υπολογισμού δεν συμβαδίζουν με την κοινή λογική. Δηλαδή η ίδια η γλώσσα θα πρέπει να επιτρέπει στο χρήστη της να μεταφέρει τις σκέψεις του σε αυτή χωρίς δυσκολία και να είναι έτσι δομημένη ώστε να προσφέρει προφανείς λύσεις σε προβλήματα που σε άλλες γλώσσες θεωρούνται δύσκολα.

Στους κλασικούς υπολογιστές η ανάπτυξη των γλωσσών υψηλού επιπέδου έγινε παράλληλα με την ανάπτυξη του υλικού των υπολογιστών και την αξιοποίηση των νέων δυνατοτήτων που αυτό πρόσφερε. Είναι αδύνατον να φανταστούμε την δημιουργία των σύγχρονων λειτουργικών συστημάτων χωρίς την ανάπτυξη των γλωσσών αυτών. Τις ίδιες ανάγκες καλούνται να καλύψουν και οι γλώσσες κβαντικού προγραμματισμού. Όμως αυτές έχουν ένα επιπλέον έργο να επιτελέσουν.

Πολύ πριν την ανάπτυξη γλωσσών υψηλού επιπέδου υπήρχαν αλγόριθμοι για διάφορα λογικά προβλήματα. Για παράδειγμα είχε ανακαλυφθεί ο αλγόριθμος πολλαπλασιασμού ακεραίων ή ο αλγόριθμος εύρεσης μέγιστου κοινού διαιρέτη. Οι γλώσσες υψηλού επιπέδου απλώς τυποποίησαν αυτές τις περιγραφές ώστε να εκτελεστούν αυτοί οι αλγόριθμοι σε υπολογιστή. Αντίθετα στο κβαντικό μοντέλο υπολογισμού υπάρχουν πολλοί λίγοι αλγόριθμοι που κάνουν κάτι ενδιαφέρον και μάλιστα οι διεργασίες που εκτελούν αντιβαίνουν στην κοινή λογική. Είναι πολύ απλούστερο να καταλάβει κάποιος την εντολή «Θέσε την τιμή του τάδε καταχωρητή στην τιμή 2» παρά «Θέσε τον τάδε κβαντικό καταχωρητή σε μια υπέρθεση τιμών 20 και 30 με τους τάδε παράγοντες μίξης». Μάλιστα είναι πολύ δύσκολο να προβλέψεις την συμπεριφορά του κβαντικού καταχωρητή σε μετέπειτα γραμμές προγράμματος.

Ένα στοιχείο που επιδεινώνει την κατάσταση είναι η σπανιότητα ενδιαφερόντων αλγορίθμων. Η εξοικείωση με τον κλασικό προγραμματισμό γίνεται με τη μελέτη πολλών και διαφορετικών αλγορίθμων, κάτι το οποίο δεν είναι δυνατό στην περίπτωση του κβαντικού μοντέλου. Επομένως μια καλή γλώσσα

κβαντικού προγραμματισμού θα πρέπει να γεφυρώνει το χάσμα ανάμεσα στο κλασικό και το κβαντικό μοντέλου υπολογισμού καιναβοηθάει στην ανάπτυξη χρήσιμων κβαντικών αλγορίθμων. Παρακάτω παρουσιάζουμε συνοπτικά δύο γλώσσες.

3.2 QPL "quantum data, classical control paradigm"

Ο *Peter Selinger* περιγράφει την κβαντική γλώσσα *QPL(QuantumProgramming Language)* η οποία ακολουθεί το μοντέλο κβαντικών δεδομένων και κλασικού ελέγχου. Αυτό σημαίνει ότι η ροή του προγράμματος είναι καθορισμένη σε μία και μόνο κατάσταση ενώ τα qubits μπορούν να βρίσκονται σε υπέρθεση καταστάσεων. Μπορούμε να φανταστούμε μία κλασική μονάδα ελέγχου να ελέγχει την εκτέλεση των κβαντικών εντολών πάνω σε κβαντικά *bits*.

Χαρακτηριστικά της γλώσσας είναι:

• ΄Υποστήριξη κλασικών *bit* και σαφής διαχωρισμός κλασικών και κβαντικών *bit*

• Υποστήριξη οποιουδήποτε ορθομοναδιαίου μετασχηματισμού

• Υποστηριξη *loops*

• Υποστήριξη αναδρομικών συναρτήσεων

3.3 Σύνταξη QPL

Η γραμματική που παράγει τα προγράμματα της *QPL* είναι η εξής:

P::= new bit b:=0 $|new\ \ qbit\ q{:=}0\ |discard\ x$

$|b \coloneqq 0\ |b \coloneqq 1\ |q_1, q_2,\ldots,q_n$ **=S*

$|skip\ |P; Q$

$|if\ b\ then\ P\ else\ Q\ |measure\ q\ then\ P\ else\ Q\ |\ while\ b\ do\ P$

$|proc\ X{:}\ \Gamma \rightarrow \Gamma'\{P\}$ *in Q* $|y_1, y_2,\ldots, y_m$ *=X(*$\left(x_{1\,,}\ x_2, \ldots, x_n\right)$

3.3.1 Σημασιολογία

Η σημασιολογία κάθε προγράμματος δίνεται σαν μία συνάρτηση $D_n^a \rightarrow D_n^b$ όπου τα *n* και *m* είναι ο αριθμός των *qubits* εισόδου και εξόδου αντίστοιχα ενώ τα *a, b* ο αριθμός των κλασικών *bits* εισόδου και εξόδου. Δηλαδή κάθε πρόγραμμα αντιστοιχίζει διανύσματα από πίνακες πυκνότητας σε διανύσματα από πίνακες πυκνότητας.

Οι εντολές *new bit* και *new qbit* δημιουργούν μια νέα μεταβλητή τύπου *bit* ή *qbit* αντίστοιχα με αρχική τιμή *0*. Η εντολή *discard* καταστρέφει την μεταβλητή και στην περίπτωση που αυτή είναι τύπου *qbit* εκτελεί την μέτρηση του πρώτα και μετά «ξεχνάει» το αποτέλεσμα. Ο λόγος που γίνεται πρώτα η μέτρηση είναι ότι το *qbit* αυτό μπορεί να ήταν σε μία κατάσταση συζευγμένη με ένα άλλο και σε περίπτωση που δεν το μετρούσαμε δεν θα ξέραμε αν γινόταν κάποια μέτρηση μετέπειτα και άρα δεν θα μπορούσαμε να δώσουμε την σωστή σημασιολογία.

Η επόμενη εντολή που χρειάζεται σχολιασμό είναι η q_1 , q_2 , …. , q_n *=*S* με την οποία εφαρμόζεται ο ορθομοναδιαίος μετασχηματισμός *S* στα qubits q_1 *εώς* q_n.Ο πίνακας του *S* είναι μέγεθος $2^n \times 2^n$.Στην περίπτωση που τα *qubits* δεν είναι με τη σωστή σειρά τότε πρέπει να εφαρμοστεί μια αλλαγή βάσης,με την ίδια εντολή να εφαρμοστεί μθα αλλαγή βάσης.Αν οι τελεστέοι είναι λιγότεροι από το πλήθος των qubits που αποτελούν το state ο *S* πρέπει να αντικατασταθεί από τοτ κατάλληλο ταυτιστικό γινόμενο του με τους μοναδιαίους.

Όρος	**Πεδίο ορισμού**	**Συνάρτηση**
$\boldsymbol{new\ qbit}$	$\boldsymbol{D_n \rightarrow D_{2n}}$	$\boldsymbol{A \rightarrow \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}}$
$\boldsymbol{discard}$	$\boldsymbol{D_n \rightarrow D_{n/2},\ n > 1}$	$\boldsymbol{\begin{pmatrix} A & B \\ C & D \end{pmatrix} \rightarrow A + D}$
$\boldsymbol{q_1, q_2, \dots, q_n *= S}$	$\boldsymbol{D_{2^n} \rightarrow D_{2^n}}$	$\boldsymbol{A \rightarrow SAS^*}$
$\boldsymbol{measure\ q\ then\ P\ else\ Q}$	$\boldsymbol{D_n \rightarrow D_m}$ με $\boldsymbol{\|P\|, \|Q\|: D_n \rightarrow D_m}$	$\boldsymbol{\begin{pmatrix} A & B \\ C & D \end{pmatrix} \rightarrow \|P\|\left(\begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}\right) + \|Q\|\left(\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}\right)}$
$\boldsymbol{skip}$	$\boldsymbol{D_n \rightarrow D_n}$	$\boldsymbol{A \rightarrow A}$
$\boldsymbol{proc\ X: \Gamma \rightarrow \Gamma'\ \{P\}\ in\ Q}$	Ίδιο με της $\boldsymbol{\|Q\|}$	$\boldsymbol{\|Q\|}$ με $\boldsymbol{\|call\ X\| := \|P\|}$
$\boldsymbol{y_1, y_2, \dots, y_m = X(x_1, x_2, \dots, x_n)}$	$\boldsymbol{D_{2^n} \rightarrow D_{2^m}}$	Προκύπτει από προηγούμενο proc
$\boldsymbol{P; Q}$	$\boldsymbol{D_n \rightarrow D_m}$	$\boldsymbol{\|Q\|(\|P\|)}$ με $\boldsymbol{\|Q\|: D_k \rightarrow D_m}$ και $\boldsymbol{\|P\|: D_n \rightarrow D_k}$

Πίνακας 3.1 *Όροι και συναρτήσεις*

Η σημασιολογία αυτή με απλά λόγια μπορεί να οριστεί ως εξής:

• Η δέσμευση ενός νέου *qbit* γίνεται πάντα έτσι ώστε το νέο qbit να είναι το πιο σημαντικό του καταχωρητή της συνολικής κατάστασης. ΄Ετσι το νέο state είναι $|0\rangle \otimes 8_p$ όπου 8_p το προηγούμενο.

• Η απελευθέρωση ενός *qbit* μπορεί εύκολα να οριστεί όταν απελευθερώνουμε το πιο σημαντικό qbit. Είναι ισοδύναμη με τη μέτρησή του και την απώλεια του αποτελέσματος. Ο λόγος που γίνεται αυτό είναι ότι το *bit* που απελευθερώνουμε μπορεί να είναι *entangled* με κάποιο άλλο το οποίο κρατάμε και μια πιθανή μέτρησή του να επιφέρει αλλαγές και σε αυτό. Αν το απελευθερώσουμε χωρίς να προκαλέσουμε άμεσα αυτές τις αλλαγές είναι αδύνατο να προβλέψουμε πότε θα προκληθούν και έτσι δεν μπορούμε να δώσουμε μια σωστήσημασιολογία. Σε περίπτωση που θέλουμε να απελευθερώσουμε κάποιο άλλο *qbit* (όπως άλλωστε επιτρέπει η σύνταξη) μπορούμε να κάνουμε μία αλλαγή βάσης ώστε το ζητούμενο *qbit* να γίνει το πιο σημαντικό,και μετά το *discard* μία δεύτερη αλλαγή βάσης ώστε το *qbit* το οποίο ήταν αρχικώς πιο σημαντικό να επιστρέψει στη θέση του.

• Η μέτρηση ενός *qbit* μας δίνει την δυνατότητα να εκτελέσουμε ένα μέρος του προγράμματος υπό συνθήκη. Η σημασιολογία ορίζεται μόνο σε περίπτωση που τα new και discard είναι ισορροπημένα στα δύο branches του *measure.* Παρατηρούμε πως η εντολή measure q_0 *then skip else skip* έχει την ίδια σημασιολογία με το *discard* , όπως είδαμε και παραπάνω.

• Η χρήση ενός ορθομοναδιαίου μετασχηματισμού έχει την ιδιαιτερότητα πως μπορεί να γίνει σε ένα αυθαίρετο σύνολο από *qbit.* Για να χρησιμοποιήσουμε όσα ξέρουμε ήδη για τους μετασχηματισμούς θα πρέπει να κάνουμε κατάλληλες αλλαγές βάσεις, ενώ αν οι τελεστές οι είναι λιγότεροι από το πλήθος των *qbits* που αποτελούν το state ο *S* θα πρέπει να αντικατασταθεί φυσικά από το τανυστικό γινόμενό του με τον αντίστοιχο μοναδιαίο.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η χρήση *procedures* και μάλιστα αναδρομικών.Η χρήση μιας διαδικασίας η οποία δεν είναι αναδρομική,έμμεσα ή άμεσα, μπορεί ουσιαστικά να αντικατασταθεί στο σημείο όπου χρησιμοποιείται, σαν μακροεντολή. Σε περίπτωση όμως που έχουμε αναδρομή θα ακολουθήσουμε την εξής διαδικασία. ΄Εστω *X* η αναδρομική διαδικασία. Τότε θα λέμε *X(Y*) την διαδικασία που είναι ίδια με την X μόνο που η αναδρομική κλήση έχει αντικατασταθεί από τον όρο *Y.*Έστω Y_0 ένα πρόγραμμα που δεν τερματίζεται .Επίσης έστω Y_{i+1}=*X*(Y_i).Τότε χρησιμοποιώντας το γεγονός ότι $\|Y_0\| = 0$ υπολογίζουμε το όριο $\lim_{i\to\infty}\|Y_i\|$,το οποίο είναι και το $\|X\|$.Είναι πιθανό το *trace* του πίνακα που δίνεται σαν είσοδο σε μία αναδρομική διαδικασία να είναι μεγαλύτερο από αυτό που προκύπτει σαν έξοδος. Δηλαδή η χρήση αναδρομικών συναρτήσεων είναι το μόνο κομμάτι της γλώσσας που δεν διατηρεί αναγκαστικά ανέπαφοτο trace(οι ορθομοναδιαίοι μετασχηματισμοί το διατηρούν λόγω της αντίστοιχης ιδιότητας των πινάκων όπως το διατηρούν και οι μετρήσεις). Αυτό απεικονίζει το γεγονός πως μπορούμε να έχουμε και προγράμματα τα οποία δεν τερματίζουν ποτέ.

3.3.2 Παραδείγματα QPL

1ο παράδειμα

Ρίψη κέρματος: Έστω πως το ζητούμενο πρόγραμμα είναι να προσομοιώσουμε την ρίψη ενός κέρματος και ανάλογα με το αποτέλεσμα να εκτελέσουμε ένα συγκεκριμένο κομμάτι του προγράμματος. Σημειώνουμε ότι στους κλασικούς υπολογιστές κάτι τέτοιο είναι αδύνατον καθώς δεν μπορούμε να πάρουμε αληθινά τυχαίες μεταβλητές. Η τυχαιότητα υλοποιείται με γεννήτριες ψευδοτυχαίων αριθμών. Έστω και οι δύο κλάδοι που θέλουμε να εκτελέσουμε. Τότε το ζητούμενο πρόγραμμα είναι:

$$New\ qbit\ q$$

$$q \ast= H$$

$$measure\ q\ them\ discard\ q; P$$

$$Else\ discard\ q; Q$$

Έστω πως το παραπάνω πρόγραμμα δέχεται σαν είσοδο τον πίνακα πυκνότητας .Τότε εφαρμόζοντας τους κανόνες της σημασιολογίας η εκχώρηση νέου *qubit* θα μας δώσει τον πίνακα$\begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}$.Έπειτα η εφαρμογή του μετασχηματισμού Hadamard ,ο οποίος δημιουργεί τη ζητούμενη υπέρθεση στο νέο *qubit*,ισοδυναμεί με εφαρμογή τοω μετασχηματισμού (*H*⊗*I) =*$\left(\frac{1}{\sqrt{2}}\begin{bmatrix} I & I \\ I & -I \end{bmatrix}\right)$ σε ολόκληρο το state.

$$\frac{1}{\sqrt{2}}\begin{bmatrix} I & I \\ I & -I \end{bmatrix}\begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}\left(\frac{1}{\sqrt{2}}\begin{bmatrix} I & I \\ I & -I \end{bmatrix}\right) = \frac{1}{2}\begin{pmatrix} A & A \\ A & A \end{pmatrix}$$

Η μέτρηση δίνει στον ένα κλάδο τον πίνακα $\frac{1}{2}\begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}$ και στον άλλο $\frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & A \end{pmatrix}$.Η απόρριψη του *qubit q* δίνει ως αποτέλεσμα $\frac{1}{2}A$ και στους δύο κλάδους.Αυτός ο πίνακας δίνεται σαν είσοδος σε καθένα από τα προγράμματα *P* και *Q* και άρα το τελικό αποτέλεσμα του προγράμματος είναι $P\left(\frac{1}{2}A\right) + Q\left(\frac{1}{2}A\right) = \frac{1}{2}\big(P(A) + Q(A)\big)$.Το τελευταίο προκύπτει από τη γραμμικότητα της σημασιολογίας της γλώσσας QPL και άρα το πρόγραμμα εκτελεί μια από τις δύο υπορουτίνες στην είσοδο του με πιθανότητα *1\2.*

2ο παράδειγμα

Ρίψη κέρματος ταυτόσημη με μέτρηση: Συνεχίζοντας στο ίδιο παράδειγμα θέτουμε *Q:=skip* και *P:=*q_0**=*$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$.Αν η αρχική είσοδος του προγράμματος είναι ο πίνακας $\begin{pmatrix}A & B\\ C & D\end{pmatrix}$ τότε σύμφωνα με τα παραπάνω το αποτέλεσμα όλου του προγράμματος θα είναι:

$$\frac{1}{2}\left\{P\left(\begin{pmatrix}A & B\\ C & D\end{pmatrix}\right)+Q\left(\begin{pmatrix}A & B\\ C & D\end{pmatrix}\right)\right\}=\frac{1}{2}\left\{\left(\begin{pmatrix}A & -B\\ -C & D\end{pmatrix}\right)+\left(\begin{pmatrix}A & B\\ C & D\end{pmatrix}\right)\right\}=\begin{pmatrix}A & 0\\ 0 & D\end{pmatrix}$$

Όμως αυτός είναι ο ίδιος ακριβώς πίνακας πυκνότητας που θα προέκυπτε με μία μέτρηση στο πρώτο *qubit* του *state*. Επομένως όσον αφορά τη γλώσσα *QPL* το αποτέλεσμα των δύο προγραμμάτων είναι εντελώς ταυτόσημο.

Με ένα παράδειγμα όμως βρίσκουμε ότι πρόκειται για δύο διαφορετικές καταστάσεις. Έστω ότι η αρχική κατάσταση του προγράμματος αποτελείτο μόνο από ένα *qubit* στην κατάσταση $\frac{3}{5}|0\rangle+\frac{4}{5}|1\rangle$.

Το πρόγραμμα είτε αφήνει την είσοδο ανέπαφη είτε αντιστρέφει τον συντελεστή του $|1\rangle$ με πιθανότητες ½.Άρα οδηγεί στη μικτλη κατάσταση $\frac{1}{2}\left\{\frac{3}{5}|0\rangle+\frac{4}{5}|1\rangle\right\}+\frac{1}{2}\left\{\frac{3}{5}|0\rangle-\frac{4}{5}|1\rangle\right\}$.Μία μέτρηση θα οδηγούσε σε μικτή κατάσταση $\left\{\frac{9}{25}|0\rangle+\frac{16}{25}|1\rangle\right\}$.

Οι δύο καταστάσεις έχουν τον ίδιο πίνακα πυκνότητας και δίνουν ίδια αποτελέσματα στις μετρήσεις. Αν όμως στην πρώτη περίπτωση ένας παρατηρητής μάθει το αποτέλεσμα της ρίψης του κέρματος μπορεί εφαρμόζοντας τον κατάλληλο αντίστροφο μετασχηματισμό να ξαναφέρει την κατάσταση του qubit στην αρχική χωρίς να την γνωρίζει εκ των προτέρων. Αντίθετα στη δεύτερη περίπτωση ακόμη και να ξέρει το αποτέλεσμα της μέτρησης δεν μπορεί να συμπεράνει ποιοι ήταν οι συντελεστές της κατάστασης. Αυτή η δυνατότητα που προσφέρει η γνώση κάποιας μέτρησης χρησιμοποιείται στην κβαντική κρυπτογραφία και είναι έξω από τους σκοπούς αυτής της εργασίας.

3.4 QML –“quantum data, quantum control paradigm”

Ο *Thorsten Altenkirch* καιο *Jonathan Grattage* παρουσιάζουν μια συναρτησιακή γλώσσα κβαντικού προγραμματισμού. Το ιδιαίτερο χαρακτηριστικό της *QML (Quantum Meta Language*) είναι ότι εισάγει και

κβαντικές δομές ελέγχου πέρα από τις κβαντικές δομές δεδομένων. Βασίζεται στην αυστηρή γραμμική λογική επιτηρώντας προσεκτικά τα μέρη που υπονοούν τη διενέργεια μέτρησης. Η λειτουργική σημασιολογία της δίνεται με κβαντικά κυκλώματα.

Η γλώσσα *QML* αποτελεί ένα πολύ ουσιαστικό πρώτο βήμα προς την κατεύθυνση της δημιουργίας μιας κβαντικής γλώσσας ωψηλού επιπέδου. Επιχειρεί να ενσωματώσει έννοιες εγγενείς στο κβαντικό μοντέλο,όπως η υπέρθεση και οι ελεγχόμενοι μετασχηματισμοί,με τρόπο τέτοιο ώστε ο προγραμματιστής να διευκολύνεται στη χρήση τους. Ταυτόχρονα επιχειρεί να διατηρήσει ένα συναρτησιακό στυλ,ξεχωρίζοντας μέσα στο σύστημα τύπων τα μέρη της γλώσσας που έχουν παρενέργειες (δηλαδή τις μετρήσεις). Παρακάτω παρουσιάζουμε το συντακτικό και το σύστημα τύπων της *QML*, και δίνουμε μια σύντομη περιγραφή της σημασιολογίας της με κυκλώματα.

3.5 Σύνταξη QML

Η γραμματική των προγραμμάτων της *QML* είναιη εξής**:**

t ::= x | let x = t in u

| $x^{\vec{y}}$ *| ( )*

| (t, u) | let (x ,y) = t in u

| qinl t | qinr t

|case t of {qinl x ⇒ u | qinr y ⇒ u'}

| caseo t of {qinl x ⇒ u | qinr y ⇒ u'}

| {(κ)t |(ι)u}

| $f^{\vec{t}}$

3.5.1 Σημασιολογία

Η σημασιολογία της *QML* δίνεται με όρους κβαντικών κυκλωμάτων. Σε κάθε όρο αντιστοιχίζεται ένα κβαντικό κύκλωμα το οποίο σχηματίζεται επαγωγικά συνδέοντας κατάλληλα τα κυκλώματα που αντιστοιχούν στους υποόρους του. Η πρωτοτυπία της γλώσσας είναι ότι το σύστημα τύπων ακολουθεί αυστηρώς γραμμική λογική και ότι χρησιμοποιεί δύο συστήματα τύπων, ένα αυστηρό και ένα γενικό. Όταν

κάνει αυστηρό *typecheck* τα κυκλώματα που παράγονται δεν διεξάγουν μετρήσεις ενώ όταν κάνει γενικό μπορεί να διεξάγουν μετρήσεις και επομένως είναι μη αντιστρέψιμα. Καθώς όμως το σύστημα τύπων είναι γραμμικό οι *Grattage* και *Altenkirch* έπρεπε να βρουν ένα τρόπο να μοιράζουν τις μεταβλητές ανάμεσα στα context και επομένως να διπλασιάζουν *qubit*.Όπως είδαμε παράγει συζευγμένα *qubit*. Αυτό όμως δημιουργεί παρενέργειες στα προγράμματα της γλώσσας, οι οποίες πρέπει να ελέγχονται προσεκτικά από τον προγραμματιστή. Παρακάτω θα δούμε ένα παράδειγμα παρενεργειών.

Η ουσιαστική συνεισφορά της γλώσσας είναι η δομή *case* ,η οποία εκτελεί κβαντική διακλάδωση, δηλαδή είναι δυνατόν να δώσει υπέρθεση προγραμμάτων στην περίπτωση που η έκφραση που λάβει σαν είσοδο βρίσκεται σε υπέρθεση.Η δομή *case* εκτελεί κλασική διακλάδωση, δηλαδή μετράει το αποτέλεσμα της έκφρασης t και εκτελεί έναν από τους δύο κλάδους ανάλογα με το αποτέλεσμα.

Τέλος, αναφέρουμε ότι με βάση τις εντολές που δώσαμε μπορούμε να κατασκευάσουμε άλλες εντολές οι οποίες είναι πιο κοντά στις δομές του κλασικού προγραμματισμού και τις οποίες θα χρησιμοποιήσουμε στα παρακάτω παραδείγματα. Αυτές είναι:

qfalse =qinr( )

qtrue=qinl( )

$if^{a}\ b\ then\ t\ else\ u = case^{a}\ b\ of\ \{inl\_ \Rightarrow t | inr\_ \Rightarrow u\} με\ \alpha = \circ$ ή *κενό*

Το πλήθος των εισόδων και εξόδων ενός κυκλώματος μπορεί να καθοριστεί από τον τύπο του όρου από τον οποίο προέρχεται καθώς και τους τύπους των υποόρων του. Ορίζουμε λοιπόν για το σύστημα τύπων την πράξη $|\tau|$ η οποία για κάθε τύπων μας δίνει το αντίστοιχο πλήθος.Έχουμε: $|Q_1| = 0$, $|\tau \otimes \sigma| = |\tau| + |\sigma|$, , $|\tau \otimes \sigma| = (|\tau| \sqcup |\sigma|) + 1$ όπου $\sqcup$ συμβολίζουμε το μέγιστο δύο αριθμών .Έτσι ένας όρος που έχει τύπο Q_2 θα μας δίνει ένα κύκλωμα με το μέγιστο δύο αριθμών $|Q_2| = |Q_1 \otimes Q_1| = |Q_1| + 1 = 1$ έξοδο.

Τα στοιχειώδη κυκλώματα, δηλαδή οι στοιχειώδεις αντιστρέψιμοι κβαντικοί υπολογισμοί που θα χρησιμοποιήσουμε είναι:

• Περιστροφή: Η περιστροφή είναι ένας υπολογισμός μίας εισόδου και μίας εξόδου που χαρακτηρίζεται από τον ορθομοναδιαίο πίνακα

$$\begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix}$$

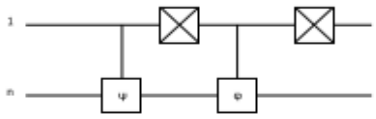
Σημειώνουμε πως και η άρνηση είναι μία ειδική περίπτωση περιστροφής με $u_{00} = u_{11} = 0$ *και* $u_{10} = u_{01} = 1$.

• Αναδιάταξη: ΄Ενας κβαντικός υπολογισμός n εισόδων και εξόδων που αντιπροσωπεύει οποιαδήποτε μετάθεση των εισόδων του.

• Διαδοχική σύνθεση: Χρησιμοποιώντας δύο κυκλώματα με τον ίδιο αριθμό εισόδων και εξόδων σχηματίζουμε ένα κύκλωμα όπου η έξοδος του πρώτου είναι είσοδος του δεύτερου.

• Παράλληλη σύνθεση: Χρησιμοποιώντας δύο κυκλώματα, το πρώτο με n εισόδους και εξόδους και το δεύτερο με m σχηματίζουμε ένα κύκλωμα με $n+m$ εισόδους και εξόδους, όπου οι πρώτες n είσοδοι πάνε στο πρώτο συστατικό κύκλωμα και οι υπόλοιπες στο άλλο.

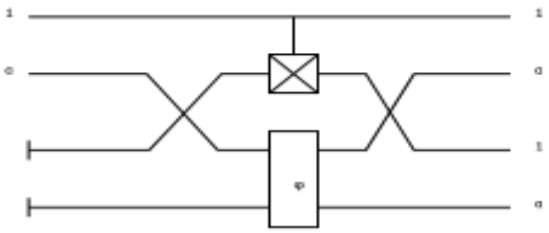
• Συνθήκη: Δοθέντων δύο κυκλωμάτων φ,ψ ίδιου πλήθος εισόδων και εξόδων n σχεδιάζουμε ένα κύκλωμα με $n+1$ εισόδους και εξόδους όπως φαίνεται παρακάτω σχήμα 3.1.



Σχήμα 3.1 *Κύκλωμα n+1 εισόδους-εξόδους*

Μπορούμε να δημιουργήσουμε μη αντιστρέψιμους υπολογισμούς διαφορετικού πλήθους εισόδων και εξόδων θεωρώντας πως ένα μέρος των εισόδων είναι αρχικοποιημένες στην τιμή *qfalse* και πως ένα μέρος των εξόδων είναι σκουπίδια. Οι είσοδοι που ξεκινάνε αρχικοποιημένες θα συμβολίζονται με $\vdash$ και τα σκουπίδια θα καταλήγουν σε $\dashv$ στο κύκλωμα μας.

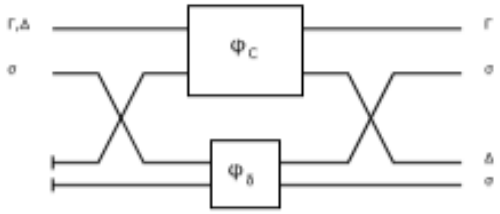
Ας ορίσουμε τώρα κάποια βοηθητικά κυκλώματα τα οποία θα μας βοηθήσουν να ξεχωρίσουμε τα *context* που απαιτούνται από κάθε όρο.Στο σχήμα από κάτω ο επαγωγικός ορισμός ενός κυκλώματος φ_δ α για κάθε α. Συγκεκριμένα για $\alpha=0$ το κύκλωμα γίνεται μια ελεγχόμενη πύλη *not*.Για $\alpha>0$ το κύκλωμα χρησιμοποιεί το φ_δ για το σχηματισμό του $\varphi_\delta(\alpha+1)$, όπως στο σχήμα 3.2.



Σχήμα 3.2 *Επαγωγικός ορισμός ενός κυκλώματος φ_δ α για κάθε α*

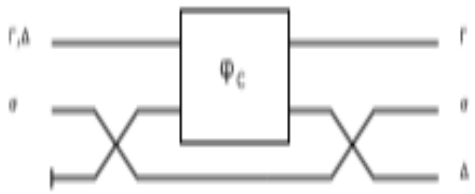
Ο ρόλος του κυκλώματος φ_δ είναι το μοίρασμα των μεταβλητών ανάμεσα στα *context*. ΄Εστω πως έχουμε δύο όρους οι οποίοι μοιράζονται μία μεταβλητή. Είναι γνωστό πως σε ένα κβαντικό κύκλωμα δεν μπορούμε να κάνουμε διακλάδωση και να δώσουμε την ίδια είσοδο στα κυκλώματα των δύο όρων. Το καλύτερο που μπορούμε να κάνουμε είναι να χρησιμοποιήσουμε την πύλη *cnot* ώστε να δώσουμε στα δύο κυκλώματα ένα *entangled* ζευγάρι από *qbit* όπου τα δύο *qbit* έχουν τις ίδιες πιθανότητες μετά από μία μέτρηση να δώσουν αποτέλεσμα $|0\rangle$ ή $|1\rangle$

Χρησιμοποιώντας το κύκλωμα φ_δ κατασκευάζουμε το κύκλωμα φ_C στην περίπτωση που δύο context *Γ,Δ* μοιράζονται την ίδια μεταβλητή.Το κύκλωμα θα είναι :



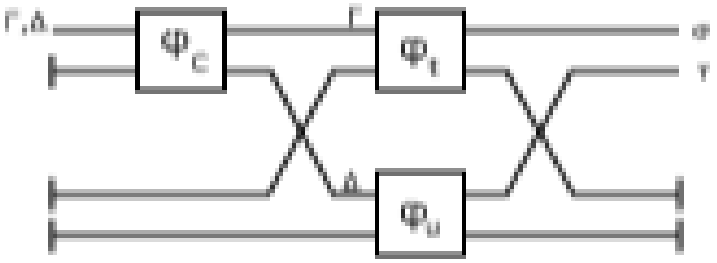
Σχήμα 3.3 *Το κύκλωμα φ_δ και φ_C*

Το κύκλωμα αυτό ορίζεται επίσης επαγωγικά. Σε περίπτωση που τα *Γ,Δ* μοιράζονται μία μεταβλητή *x* : *σ* χρησιμοποιούμε το φ_δ για να μοιράσουμε δύοφορές αυτή τη μεταβλητή και το φ_C στις υπόλοιπες μεταβλητές τους.. Σε περίπτωση που τα *Γ,Δ* δεν μοιράζονται μεταβλητή το φ_C έχει την παρακάτω μορφή του σχήματος,ενώ στην περίπτωση που το ένα είναι το φ_C είναι απλώς το ταυτιστικό κύκλωμα ,δηλαδή δίνει απλώς το άλλο. Με βάση τα παραπάνω η ερμηνεία του *weak* είναι απλή: περνάμε τα *context Γ,Δ* από το φ_C και στέλνουμε την έξοδο που αντιστοιχεί στο *Δ* στα σκουπίδια.



Σχήμα 3.4 *Κύκλωμα* φ_C

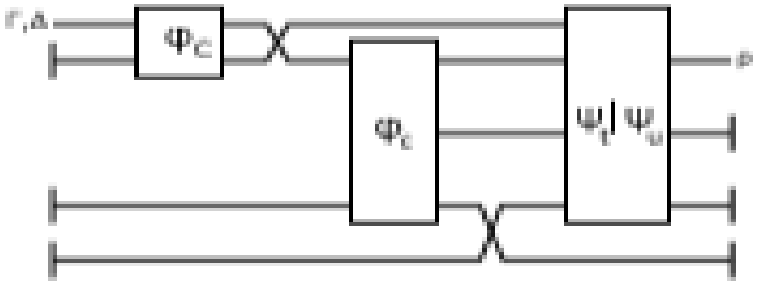
Ας δούμε τώρα τους πιο ενδιαφέροντες σημασιολογικούς κανόνες της γλώσσας. Ξεκινάμε με τα γινόμενα, όπως φαίνονται στο σχήμα 3.5.



Σχήμα 3.5 *Κυκλώματα* φ_C, φ_δ *και* φ_t

Όπως βλέπουμε ο όρος *(t,u)* εμηνεύεται ουσιαστικά σαν η παράλληλη σύνθεση δύο κυκλωμάτων: αυτού που αντιστοιχεί στο t και αυτού που αντιστοιχεί στο u. Το context μοιράζεται κατάλληλα στα δύο κυκλώματα σε περίπτωση που μοιράζονται μεταβλητές. Τα κυκλώματα που αντιστοιχουν στους δύο *constructors* των αθροισμάτων είναι στοιχειώδη.΄Εστω πως έχουμε τον όρο $t : \tau$ και θέλουμε να κατασκευάσουμε το κύκλωμαγιατο $inl\, t : \tau \oplus \sigma$. $Αν |\tau| > |\sigma|$το κύκλωμα είναι απλώς το κύκλωμα που αντιστοιχεί στο *t* με την προσθήκη στην έξοδο ενός *qbit* στην κατάσταση qtrue. Αλλιώς προσθέτουμε ένα *padding* από όσα χρειάζεται qbitώστεηέξοδοςναέχειμέγεθος$|\sigma|+1$. Αντίστοιχα ισχύουν και για το inr μόνο που το *qbit* που προσθέτουμε είναι *qfalse*. Βλέπουμε λοιπόν πως τα αθροίσματα υλοποιούνται με έναν τρόπο παρόμοιο με αυτόν που θα χρησιμοποιούσαμε και σε μία κλασική γλώσσα: δημιουργούμε έναν τύπο αρκετά μεγάλο για να χωρέσει και τους δύο τύπους του αθροίσματος και προσθέτουμε ένα *qbit* για να ξεχωρίσουμε ανάμεσα στους δύο τύπους που μπορεί να περιέχονται.

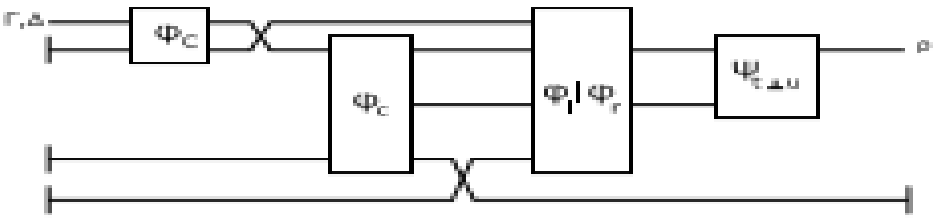
Ας δούμε τώρα την περίπτωση του *case*, η οποία φαίνεται στο σχήμα 3.6. Πρώτα θα εκτελεστεί ο υπολογισμός που αντιστοιχεί στο πρώτο τελούμενο του case, ο οποίος λόγω του ότι είναι τύπου $\oplus$ θα μας δώσει σίγουρα ένα Q_2. Το *qbi*t αυτό χρησιμοποιείται στον έλεγχο της υπό συνθήκη αποτίμησης των *t,u* και στο τέλος καταλήγει στα σκουπίδια (κάτι που υπονοεί μέτρηση).



Σχήμα 3.6 *Περίπτωση case*

Το μεγαλύτερο ενδιαφέρον παρουσιάζει η περίπτωση του $case^{\circ}$. Το αντίστοιχο κύκλωμα φαίνεται στο παρακάτω σχήμα. Η υπόσυνθήκη αποτίμηση δεν παράγει σκουπίδια,διότι οι αντίστοιχοι όροι αποδεικνύονται αυστηρά. Επίσης χρησιμοποιούνται τρία κυκλώματατα φ_I , φ_r $και$ $\psi_{t\perp u}$, τα οποία προκύπτουν από την ερμηνεία της ορθογωνιότητας των *t,u* στον κανόνα του $case^{\circ}$. Χωρίς να μπούμε σε λεπτομέρειες μπορούμε να πούμε πως τα κυκλώματα αυτά ορίζονται πάλι επαγωγικά από απλούστερα αποτελούμενα μόνο από μεταθέσεις και πύλες *not*. Μόνη εξαίρεση ο κανόνας ορθογωνιότητας των υπερθέσεων ο οποίος προκαλεί την περιστροφή ψ=$\begin{pmatrix} \lambda_0 & \lambda_1 \\ \kappa_0 & \kappa_1 \end{pmatrix}$.

Αξίζει πάντως να σημειώσουμε πως αν τα t,u είναι *qtrue,qfalse* με αυτή τη σειρά τότε το $\psi_{t\perp \mathrm{u}}$ είναι το ταυτοτικό κύκλωμα, ενώ αν είναι με την αντίθετη σειρα $\psi_{t\perp \mathrm{u}}$ είναι μια πύλη *not.*



Σχήμα 3.7 *Περίπτωση case˚*

Τέλος,το κύκλωμα που αντιστοιχεί στην υπέρθεση υπολογίζεται εύκολα αν λάβουμε υπ'οψιν πως $\{(\lambda)t \mid (\lambda')u\}$ *=*$if^{\circ}\{(\lambda)qtrue \mid (\lambda')qfalse\}$ *then t else u*.Η υπέρθεση ενός μόνο *qbit* μπορεί να δημιουργηθεί με την κατάλληλη περιστροφή του *qfalse.*

3.5.2 Παραδείγματα QML

<u>Παράδειγμα 1ο</u>

Το πιο απλό παράδειγμα είναι η πύλη *NOT:*

$$if^{\circ}\ x\ then\ qfalse\ else\ qtrue$$

Με αυτό το πρόγραμμα το γίνεται η πρώτη είσοδος της πύλης *NOT* και το αποτέλεσμα είναι συζευγμένο με αυτό.

Άλλο παράδειγμα αποτελεί η πύλη *Hadamard:*

$$had\ x \equiv if^{\circ} x\ then\ \{qfalse| - qtrue\} else\ \{qfalse|qtrue\}$$

Όπου ο τελεστής { | } δηλώνει υπέρθεση.

Ας δούμε τώρα ένα παράδειγμα όπου το μοίρασμα των μεταβλητών δημιουργεί παρενέργειες στο πρόγραμμα, οι οποίες δεν συμβαδίζουν με το συμβατικό μοντέλο προγραμματισμού.

Προφανώς έχουμε ότι *had* $\{qfalse|qtrue\}$=*qfalse.* Έστω το πρόγραμμα

$$Let\ x = qfalse\ in\ (had\ x\ , had\ x)$$

Το μοίρασμα των context δημιουργεί δύο ψευδό-αντίγραφα του *x* και τα εισάγει στα δύο υποπρογράμματα *h*ad x.Αφού το *x* είναι *qfalse* και το αντίγραφο του είναι *qfalse* και άρα το αποτέλεσμα τοτ προγράμματος είναι το αναμενόμενο,δηλαδή ($\{qfalse|qtrue\}$,$\{qfalse|qtrue\}$)

Έστω τώρα το πρόγραμμα

$$Let\ x = \{qfalse|qtrue\}\ in\ (had\ x, had\ x)$$

Λογικά θα περιμέναμε το αποτέλεσμα να είναι *(qfalse,qfalse).*Όμως εδώ το *x* βρίσκεται σε υπέρθεση η οποία όπως ξέρουμε δεν αντιγράφεται .Το *x* έχει κατάσταση $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$ και με το μόιρασμα του σε δύο παίρνουμε την κατάσταση $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$ (δηλαδή τα δύο *qubit* είναι συζευγμένα).Εφαρμόζοντας διαδοχικά δύο φορές την πύλη *Hadamard* στο καθένα ,δηλαδή τον πίνακα *(I⊗H) (I⊗H)= H⊗H,*προκύπτει η κατάσταση $\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$.Δηλαδή το αποτέλεσμα δεν είναι το ανεμενόμενο,αφού πήραμε δύο qubit σε υπέρθεση καταστάσεων $\{qfalse|qtrue\}$ τα οποία είναι μάλιστα και συζευγμένα.

Γενικώς μπορούμε να πούμε ότι η τακτική του μοιράσματος μεταβλητών μειώνει αρκετά την αξία του συναρτησιακού μέρους της γλώσσας, διότι ένας όρος που έχει μια δεδομένη συμπεριφορά όταν χειρίζεται μια ανεξάρτητη μεταβλητή αποκτά μία άλλη, διαφορετική συμπεριφορά, όταν η μεταβλητή αυτή χρησιμοποιείται και από έναν άλλο όρο και μάλιστα αυτό εξαρτάται και από τον τρόπο που ο άλλος όρος χειρίζεται το δικό του αντίγραφο. Αυτό αντιπροσωπεύει μια εγγενή αδυναμία του κβαντικού υπολογιστικού μοντέλου, η οποία προκύπτει από το γεγονός ότι είναι αδύνατο να δημιουργήσουμε αντίγραφα ενός qubit. Δεν είναι όμως σαφές κάποιο πλεονέκτημα της λύσης που υλοποιήθηκε στην *QML*, λαμβάνοντας μάλιστα υπ' όψιν πως ο σχεδιασμός της γλώσσας κάνει μεγάλο κόπο για να απομονώσει τους όρους που έχουν παρενέργειες με μέτρηση. Ίσως θα ήταν πιο συνετό το βάρος της δημιουργίας ψευδό-αντιγράφων μιας μεταβλητής να πέφτει στον προγραμματιστή, όπου αυτό είναι αναγκαίο.

Κεφάλαιο 4

Η γλώσσα κβαντικού προγραμματισμού nQML

4.1 Εισαγωγή

Η γλώσσα προγραμματισμού με την οποία θα ασχοληθούμε σε αυτό το κεφάλαιο είναι η *nQML,* η οποία αποτελεί μετεξέλιξη της *QML.*

Ο στόχος ώστε να σχεδιαστεί η *nQML* είναι να δώσει στους προγραμματιστές αρκετή εκφραστικότητα ώστε να υλοποιήσουν εύκολα κβαντικούς αλγόριθμους, ενώ ταυτόχρονα τους εμποδίζει να παραβούν τους κανόνες των κβαντικών υπολογισμών. Η *nQML* είναι μια συναρτησιακή γλώσσα υψηλού επιπέδου που βασίζεται στο μοντέλο «κβαντικά δεδομένα και κβαντικός έλεγχος». Περιλαμβάνει δομές οι οποίες επιτρέπουν κάθε ορθομοναδιαίο μετασχηματισμό να εκφραστεί ως πρόγραμμα με προφανή τρόπο, χρησιμοποιώντας περίπου τον ίδιο συμβολισμό που χρησιμοποιείται από τους σχεδιαστές κβαντικών αλγορίθμων. Επίσης επιτρέπει οι κβαντικές μετρήσεις να διεξάγονται σε οποιοδήποτε σημείο της εκτέλεσης του προγράμματος.

Η ευκολία χρήσης αυτής της γλώσσας έχει το μειονέκτημα ότι προϋποθέτει την παράλειψη αρκετών πρακτικών προβλημάτων, όπως την ύπαρξη ατελούς κβαντικού *hardware,* την ανάγκη για κβαντική διόρθωση σφαλμάτων και το γεγονός ότι κάθε κβαντικό πρόγραμμα θα πρέπει τελικά να υλοποιηθεί ως ένα κβαντικό κύκλωμα χρησιμοποιώντας μόνο ένα πεπερασμένο σύνολο πυλών και συνεπώς κάποιοι από τους ορθομοναδιαίους μετασχηματισμούς που επιτρέπονται στην *nQML* θα πρέπει να προσεγγιστούν.Παρόμοια προβλήματα αντιμετώπιζαν οι ιδρυτές του κλασικού μοντέλου προγραμματισμού δεκαετίες πριν. Ευτυχώς έχουν επιλυθεί και οι λύσεις αυτές έχουν φτάσει σε ένα τέτοιο αφηρημένο επίπεδο ώστε οι άνθρωποι που χρησιμοποιούν τις σύγχρονες γλώσσες υψηλού επιπέδου δεν χρειάζεται να ξέρουν κάτι για τις υλοποιήσεις. Πιστεύουμε ότι το ίδιο μπορεί και πρέπει να γίνει για τις κβαντικές γλώσσες προγραμματισμού του μέλλοντος και ότι αυτά τα προβλήματα πρέπει να επιλυθούν όχι από το σχεδιαστή και τους χρήστες των γλωσσών αλλά από τον αρχιτέκτονα του κβαντικού υπολογιστή, το σχεδιαστή του λειτουργικού συστήματος και σε ένα μικρότερο ποσοστό το σχεδιαστή του *compiler.*

Η *nQML* έχει ένα απλό σύστημα τύπων και λειτουργική σημασιολογία. Με τον όρο απλός εννοούμε ότι και τα δύο χρησιμοποιούν δομές και τεχνικές οι οποίες είναι της ίδιας μορφής και πολυπλοκότητας με τις αντίστοιχες των κλασικών γλωσσών προγραμματισμού.Για αύτο το λόγο γίνονται εύκολα κατανοητές από τους αναγνώστες με βασικές γνώσεις σημασιολογίας γλωσσών και μια στοιχειώδη εξοικείωση του κβαντικού μοντέλου υπολογισμού. Το βασικότερο στοιχείο του συστήματος τύπων της *nQM*L είναι ότι ο τύπος μιας κβαντικής έκφρασης περικλείει πληροφορία που δηλώνει τα συγκεκριμένα qubits της κβαντικής κατάστασης στα οποία είναι αποθηκευμένη η τιμή της έκφρασης. Η επαναχρησιμοποίηση qubit επιτρέπεται με τέτοιο τρόπο ώστε να μην παραβιάζονται οι κανόνες «μη αντιγραφής» και «μη απόρριψης».Οι

προγραμματιστές πιστεύουν ότι ασχολούνται με μια κλασσική γλώσσα χωρίς περιορισμούς γραμμικότητας.

Επιπλέον ,*nQML* βασίζεται στη χρήση πινάκων πυκνότητας οι οποίοι περιγράφουν κβαντικές καταστάσεις. Ένα ορθό πρόγραμμα της γλώσσας αντιστοιχεί σε μία συνάρτηση από πίνακες πυκνότητας σε πίνακες πυκνότητας και έτσι περιγράφεται η επίδραση του προγράμματος σε μια τυχαία κβαντική κατάσταση. Ορθά προγράμματα τα οποία δεν διενεργούν μετρήσεις αντιστοιχίζονται επίσης σε ένα ορθομοναδιαίο πίνακα ο οποίος περιγράφει τον μετασχηματισμό που διενεργούν στην κβαντική κατάσταση. Η εκτέλεση ενός προγράμματος nQML μπορεί να προσομοιωθεί με μια ακολουθία βημάτων που επηρεάζουν την κβαντική κατάσταση είτε δηλώνοντας νέα *qubits*, είτε εφαρμόζοντας ορθομοναδιαίους μετασχηματισμούς σε υπάρχοντα *qubits*, είτε μετρώντας την τιμή τους.

Σε αυτήν την πτυχιακή θα δώσουμε τη σημασιολογία των προγραμμάτων με τη μορφή κβαντικών κυκλωμάτων, κάτι το οποίο θα δώσει καλύτερη κατανόηση των λειτουργιών τους.

4.2 Σύνταξη της nQML

Υποθέτουμε ότι *χ* είναι ένα αναγνωριστικό μεταβλητής και *λ* μια μιγαδική σταθερά.Η γραμματική ορίζει δύο κλάσεις.Οι κβαντικές εκφράσεις σημειώνονται με *e˙* αντιπροσωπεύουν κβαντικά προγράμματα και η σύνταξη τους είναι παρόμοια με αυτή της *QML*. Οι κλασικές εκφράσεις σημειώνονται με *c˙* χρησιμοποιούνται μόνο στην δομή κβαντικού μετασχηματισμού |e⟩ →x,x'.c και αντιπροσωπεύουν δύο τύπους πληροφορίας: μια δομή κλασικών *bit* ή ένα μιγαδικό αριθμό.

$$e ::= x \mid \{(\lambda) q false + (\lambda') q true\} \mid let\ x = e_1\ in\ e_2$$

$$\mid (e_1,\ e_2) \mid let\ (x_1, x_2) = e_1\ in\ e_2$$

$$\mid if\ e\ then\ e_1\ else\ e_2 \mid ifm\ e\ then\ e_1\ else\ e_2 \mid \mid e\rangle \rightarrow x, x'.c$$

$$c ::= x \mid false \mid true \mid \lambda \mid let\ x = c_1\ in\ c_2$$

$$\mid (c_1, c_2 \mid let\ (x_1, x_2) = c_1\ in\ c_2 \mid if\ c\ then\ c_1\ else\ c_2$$

$$|int\ c|\ c_1 + c_2|c_1 - c_2|c_1 * c_2|c_1/c_2|c_1{}^{e_z}|c_1{=}c_2|c_1 < c_2$$

Οι μεταβλητές στην *nQML* είναι στην ουσία αναφορές σε κβαντικές πληροφορίες οι οποίες είναι αποθηκευμένες σε μία καθολική κβαντική κατάσταση. Υπάρχουν δύο τύποι κβαντικής πληροφορίας: *qubits* και γινόμενα (*products*). Ένα νέο *qubit* διατίθεται στο πρόγραμμα όταν χρησιμοποιείται ο τελεστής υπέρθεσης $(\lambda)qfalse + (\lambda')qtrue,$ με τον ίδιο τρόπο που νέα αντικείμενα διατίθενται στο σωρό *(heap*) όταν ένας *data constructor* καλείται από μια γλώσσα συναρτησιακού προγραμματισμού. Τα γινόμενα εισάγονται και απαλείφονται με τις δομές $(e_1,\ e_2)$ και *let* $(x_1,x_2) = e_1\ in\ e_2$. Η *nQML* επίσης εισάγει τρεις δομές ροής ελέγχου:

•***ifm e then*** $\boldsymbol{e_1}$ ***else*** $\boldsymbol{e_2}$ *:* Διεξάγει μία μέτρηση στην έκφραση *e* , η οποία πρέπει να είναι τύπου qubit. Ανάλογα με το αποτέλεσμα εκτελεί έναν από τους δύο κλάδους. Είναι παρόμοια με μία κλασική τυχαία διακλάδωση, που βασίζεται στην ρίψη ενός «μη δίκαιου» νομίσματος με πιθανότητες από την κατάσταση του *qubit* που μετράμε.

•***if e then*** $\boldsymbol{e_1}$ ***else*** $\boldsymbol{e_2}$ *:* Επιτρέπει στους προγραμματιστές να εκτελούν κβαντικές διακλαδώσεις. Εάν η έκφραση *e* , η οποία πρέπει να είναι τύπου *qubit*, είναι σε κλασική κατάσταση, τότε το αποτέλεσμα της εντολής είναι ίδιο με το αντίστοιχο της *ifm*. Αλλά αν το e είναι σε κβαντική υπέρθεση, το πρόγραμμα συνεχίζει σε κβαντική υπέρθεση και των δύο κλάδων· πιθανότατα δημιουργώντας συμπλοκή ανάμεσα στα qubitsτης κβαντικής κατάστασης.

•$|\boldsymbol{e}\rangle$ →***x,x'.c*** : Πρόκειται για ένα γενικό τρόπο για να εκφράσουμε οποιονδήποτε ορθομοναδιαίο μετασχηματισμό, στον οποίο πρέπει να βασιστούμε όταν ο μετασχηματισμός δεν μπορεί εύκολα να αποσυντεθεί σε μια ακολουθία ελεγχόμενων διαδικασιών, που μπορούν να εκφραστούν με *if*. Το πλεονέκτημά του είναι ότι αντί να εξαναγκάζει τους προγραμματιστές να προϋπολογίζουν και να παρέχουν ολόκληρο τον ορθομοναδιαίο πίνακα του μετασχηματισμού, τους επιτρέπει να εκφράσουν αυτό τον πίνακα ως μια μιγαδική συνάρτηση της κατάστασης εισόδου και εξόδου των *qubits* που πρέπει να μετασχηματιστούν. Αυτή η μορφή οδηγείτε σε μια συμπαγή και σαφή έκφραση πολλών χρήσιμων αλγορίθμων, όπως του κβαντικού μετασχηματισμού *Fourier*.

Σε κβαντική ψευδογλώσσα κάθε ορθομοναδιαίος μετασχηματισμός μπορεί να εκφραστεί στην εξής μορφή:

$$|i\rangle \rightarrow \sum_{j=0}^{2^{n-1}} f(i,j)\,|i\rangle$$

όπου *f(i,j)* είναι μία συνάρτηση της κατάστασης εισόδου i των κβαντικών καταχωρητών και της κατάστασης εξόδου *j*. Η δομή |e⟩ →x,x'.c επιτρέπει στους προγραμματιστές να χρησιμοποιήσουν ακριβώς αυτόν τον φυσικό συμβολισμό: οι κλασικές μεταβλητές *x* και *x'* υποδηλώνουν την κατάσταση εισόδου και εξόδου αντίστοιχα και η κλασική έκφραση *c* υποδηλώνει το σώμα της συνάρτησης.

Με αυτό το συμβολισμό, εάν η συνάρτηση *f* είναι γνωστή, ο ορθομοναδιαίος πίνακας είναι δυνατόν να κατασκευαστεί εύκολα παίρνοντας $S_{i,j}$*=f(i,j).* Φυσικά δεν καταλήγουν όλες οι συναρτήσεις *f* σε ορθομοναδιαίους πίνακες και το σύστημα τύπων της *nQML* δεν μπορεί να ελέγξει αν ο προκύπτων μετασχηματισμός είναι πράγματι ορθομοναδιαίος. Το σύστημα τύπων των *Altenkirch* και *Grattage QML* μπορεί να το κάνει αυτό, κάνοντας όμως το μέγεθος του προγράμματος εκθετικό και περιπλέκοντας τους τύπους με περιορισμούς ορθογωνιότητας. Ακόμα ο τελεστής αυτός προσφέρει μεγάλη ευκολία στο προγραμματιστή, όμως η αποδοτική υλοποίηση του φαίνεται να παρουσιάζει μεγάλες δυσκολίες.Πρακτικά ο μόνος τρόπος υλοποίησής του που μπορούμε να φανταστούμε αυτή την στιγμή είναι η αποτίμησή της συνάρτησης *f(x,x')* για όλες τις δυνατές τιμές του πεδίου ορισμού της.Έτσι μπορούμε να υπολογίσουμε τον πίνακα που μας δίνει η *f(x,x')* να ελέγξουμε ότι όντως είναι ορθομοναδιαίος και κατόπιν να προχωρήσουμε στην δημιουργία του κβαντικού κυκλώματος που τον προσεγγίζειμε βάση τις πύλες που έχουμε στην διάθεση μας. Όμως το πεδίο ορισμού της *f(x,x')* είναι εκθετικά μεγάλο ως προς το πλήθος των *qubits* που επηρεάζει ο μετασχηματισμός και συνεπώς το όποιο πλεονέκτημα μπορεί να μας προσφέρει ένας κβαντικός υπολογιστής εξανεμίζεται αφού ο υπολογισμός του ορθομοναδιαίου μετασχηματισμού είναι εκθετικός.

4.3 Παραδείγματα κβαντικών αλγόριθμων

Σε αυτή την ενότητα θα παρουσιάσουμε κάποια παραδείγματα μερικών γνωστών κβαντικών αλγόριθμων.

4.3.1 Ο αλγόριθμος του Deutsch

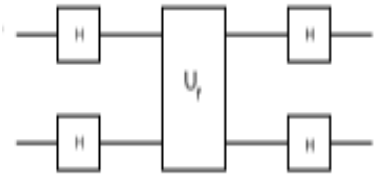
Από τους πρώτους και πιο γνωστούς κβαντικούς αλγορίθμους που προτάθηκαν είναι ο αλγόριθμος του

Deutsch και η επέκτασή του, ο αλγόριθμος των *Deutsch-Jozsa*. Το πρόβλημα το οποίο λύνει ο αλγόριθμος του *Deutsch* μπορεί να περιγραφεί ως εξής: ΄Εστω πως έχουμε μία συνάρτηση *f(x) : {0,1}* → *{0,1}*. Μπορούμε να πούμε με σιγουριά αν η συνάρτηση αυτή είναι σταθερή διεξάγοντας μόνο έναν υπολογισμό της:

Είναι προφανές πως με κλασικά μέσα το παραπάνω πρόβλημα δεν έχει λύση. . Υπάρχουν τέσσερις συναρτήσεις του τύπου της *f* και είναι οι *f(x)=¬χ* , *f(x)=χ* οι οποίες δεν είναι σταθερές και οι *f(x) = 0* , *f(x)=1* οι οποίες είναι σταθερές. Αφού έχουμε τη δυνατότητα για υπολογισμό της συνάρτησης μία μόνο φορά μπορούμε να γνωρίζουμε είτε το *f(0)* είτε το *f(1).* Γνωρίζοντας όμως μόνο το ένα από αυτά δεν είμαστε σε θέση να προσδιορίσουμε το ζητούμενο. Π.χ. αν γνωρίζουμε ότι *f(0) = 1* τότε μπορεί *f(x) =¬x ή f(x) = 1.*

Η κβαντική λύση χρησιμοποιεί το τρικ του υπολογισμού της συνάρτησης με είσοδο μία υπέρθεση των *0* και *1*. ΄Ετσι, καταφέρνουμε εμμέσως να διαπιστώσουμε αν το αποτέλεσμα των *f(1)* και *f(0)* είναι το ίδιο. Σημειώνουμε πως ακόμα κι έτσι δεν μπορούμε να είμαστε σίγουροι με μία μόνο μέτρηση για το ποια συνάρτηση είναι η *f.*

Ένα κύκλωμα που υλοποιεί τον αλγόριθμο του *Deutsch.*



Σχήμα 4.1 *Κύκλωμα Deutsch*

Η υλοποίηση του αλγορίθμου του *Deutsch* βασίζεται στη δημιουργία μίας βοηθητικής συνάρτησης U_f. Η συνάρτηση αυτή θα δημιουργεί την αντιστοιχία $|i,j\rangle \longrightarrow |i,f(i)\oplus j\rangle$. ΄Ετσι όταν *f(x)=0* η U_f είναι ο ταυτοτικός μετασχηματισμός.Όταν *f(x)=1* είναι η πύλη *cnot*. Για *f(x) = x* και *f(x) = ¬x* μπορούμε εύκολα να εξάγουμε τους αντίστοιχους μετασχηματισμούς που αντιστοιχούν στη U_f. ΄Οταν η *f* είναι μία σταθερή συνάρτηση το πρώτο *bit* του αποτελέσματος του αλγορίθμου είναι $|0\rangle$ ενώ σε αντίθετη περίπτωση είναι $|1\rangle$. Το δεύτερο *bit* είναι σε κάθε περίπτωση $|1\rangle$.

Μία λογική ένσταση που μπορεί να έχει κανείς σε αυτό το σημείο είναι ότι ακόμα κι αν το κύκλωμα του σχήματος *Deutsch* λειτουργεί σωστά(κάτι που θα δείξουμε παρακάτω) η κατασκευή του είναι αδύνατη

βάση των περιορισμών του προβλήματος αφού για να φτιάξουμε την U_f θα πρέπει να ξέρουμε την *f.* Κάτι τέτοιο όμως δεν ισχύει, όπως υα φανεί ξεκάθαρα από το πρόγραμμα *nQM*L που υα δώσουμε.

Η απόδειξη της ορθότητας του αλγορίθμου του *Deutsch* για συναρτήσεις ενός *qbit* είναι σχετικά απλή και μπορεί να γίνει με έλεγχο της εξόδου του κυκλώματος για τις τέσσερις συναρτήσεις ενός *bi*t. Ας πάρουμε εδώ ως παράδειγμα την *f(x) =¬x.*Τότε η U_f εκτελεί την διαδιακασία $|i,j\rangle \longrightarrow |i, f(i) \oplus j\rangle$ δηλαδη:

$$|00\rangle \longrightarrow |01\rangle$$

$$|01\rangle \longrightarrow |00\rangle$$

$$|10\rangle \longrightarrow |10\rangle$$

$$|11\rangle \longrightarrow |11\rangle$$

Είναι σαφές λοιπόν πως ο πίνακας του μετασχηματισμού της U_f είναι

$$U_f = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Οι άλλοι δύο μετασχηματισμοί του κυκλώματος είναι $H \otimes H$, οπότε το κύκλωμα υλοποιεί συνολικά τον μετασχηματισμό $(H \otimes H)$ ο οποίος είναι ίσος με :

$$(H \otimes H)U_f(H \otimes H) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$

Η εφαρμογή του πίνακα αυτού σε αρχική κατάσταση $|01\rangle$ μας δίνει:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Στο οποίο ήταν αναμενόμενο αποτέλεσμα αφού η *f(x)* δεν είναι σταθερή συνάρτηση .Η εξακρίβωση μπορεί να γίνει με παρόμοιο τρόπο και για τις υπόλοιπες συναρτήσεις.

4.3.2 Ο αλγόριθμος του Grover

Ο αλγόριθμος του *Grover* λύνει ένα αρκετά σημαντικό πρόβλημα πετυχαίνοντας μάλιστα απόδοση η οποία είναι αποδεδειγμένα καλύτερη από την απόδοση που μπορεί να έχει ο καλύτερος κλασικός αλγόριθμος. Το πρόβλημα είναι αυτό της αναζήτησης σε έναν μη ταξινομημένο πίνακα. Η πολυπλοκότητα του καλύτερου κλασικού αλγορίθμου είναι ως γνωστόν $O\left(\frac{n}{2}\right)$ στη μέση περίπτωση και *O(n)* στην χειρότερη.Ο αλγόριθμος του *Grover* είναι πιθανοτικός με πολυπλοκότητα $O\left(\sqrt{n}\right)$.

Μπορούμε να εκφράσουμε γενικότερα το πρόβλημα ως εξής: έστω πως έχουμε μία συνάρτηση *f(x): {0,⋯,2n −1} → {0,1}* , για την οποία ισχύει για ένα μοναδικό x_0 *f*(x_0) *= 1* καθώς και *f(x) = 0,* $\forall x \neq x_0$.Αυτό είναι ισοδύναμο με το να έχουμε έναν πίνακα 2^n στοιχείων και να αναζητούμε ένα στοιχείο το οποίο υπάρχει ακριβώς μία φορά στον πίνακα. Τότε η συνάρτηση *f* είναι απλώς μία συνάρτηση που συγκρίνει το στοιχείο που βρίσκεται στη θέση *x* με το ζητούμενο.

Αν υποθέσουμε πως μπορούμε να υλοποιήσουμε κβαντικά την *f* αυτό σημαίνει πως μπορούμε να υλοποιήσουμε κβαντικά και την εξής συνάρτηση *g(x) :* $|x\rangle \longrightarrow |x\rangle$, $\forall x \neq x_0$ και $|x_0\rangle \longrightarrow -|x_0\rangle$.Με άλλα λόγια η *g* είναι ο ταυτοτικός μετασχηματισμός όταν έχει είσοδο οτιδήποτε εκτός από τη λύση του προβλήματος x_0, ενώ όταν έχει είσοδο x_0 επιστρέφει πάλι x_0 αλλά έχοντας πολλαπλασιάσει το πλάτος με *−1*. Ο πίνακας που αντιστοιχεί λοιπόν στη *g* δεν είναι παρά ο μοναδιαίος πίνακας με μόνη διαφορά το στοιχείο της διαγωνίου που αντιστοιχεί στο x_0 το οποίο είναι*−1*. Προφανώς ο πίνακας αυτός είναι ορθομοναδιαίος.

Ορίζουμε τώρα τον *diffusion operator* (τελεστή διάχυσης)ο οποίος έχει πίνακα *I−2P* όπου *P* ένας $2^n \times 2^n$ πίνακας που όλα τα στοιχεία του είναι ίσα με2^{-n}. Είναι εύκολο να διαπιστώσουμε πως $P^2 = P$,όπως και ότι P*=P.Επομένως έχουμε *(I−2P)(I−2P)*= (I−2P)(I−2P) = I−4P+4P2 = I−4P+4P = I.* Δείξαμε λοιπόν πως ο *diffusion operator* είναι ορθομοναδιαίος και επομένως μπορούμε να τον χρησιμοποιήσουμε .

Ο *diffusion operator* λέγεται και *inversion about average operator*. Αν έχει είσοδο μία κβαντική κατάσταση όπου ο μέσος όρος των πλατών είναι *μ*, τότε κάθε ιδιοδιάνυσμα που είχε πλάτος μεγαλύτερο από *μ* θα αποκτήσει πλάτος μικρότερο από *μ* και μάλιστα η διαφορά του από το *μ* θα παραμείνει αναλλοίωτη.

Ομοίως αν ένα ιδιοδιάνυσμα είχε πλάτος μικρότερο από *μ* θα αποκτήσει με τον ίδιο τρόπο πλάτος μεγαλύτερο από *μ*. Είναι σαφές ότι ο μέσος όρος *μ* δεν μεταβάλλεται με χρήση του *diffusion operator*. Φυσικά αναφερόμαστε σε πραγματικά πλάτη(ο αλγόριθμος του *Grover* δεν ασχολείται με μιγαδικά πλάτη).

Ακόμα ο δρόμος τώρα είναι ανοικτός για τη λύση του προβλήματος . Έστω πως θέλουμε να κάνουμε αναζήτηση σε έναν πίνακα με $N = 2^n$ στοιχεία. Χρησιμοποιούμε *nqbit* τα οποία αρχικοποιούμε σε $|0\rangle$. Χρησιμοποιώντας την πύλη *Hadamard* φέρνουμε τα *n qbit* σε μια υπέρθεση όλων των πιθανών καταστάσεων με ίση πιθανότητα .Έπειτα χρησιμοποιήσαμε τη συνάρτηση *g* που ορίσαμε πιο πάνω. Μετά την εφαρμογή της όλα τα ιδιοδιανύσματα έχουν πλάτος $\frac{1}{N}$ εκτός από το $|x_0\rangle$ που έχει πλάτος $-\frac{1}{N}$. Όλα λοιπόν τα ιδιοδιανύσματα βρίσκονται πάνω από τον μέσο όρο εκτός από το $|x_0\rangle$. Ως εκ τούτου εφαρμογή του *diffusion operator* θα αυξήσει σημαντικά την πιθανότητα μια μέτρηση να μας δώσει αποτέλεσμα x_0 μειώνοντας ταυτόχρονα την πιθανότητα οποιουδήποτε άλλου αποτελέσματος. Στη συνέχεια μπορούμε να εφαρμόσουμε ξανά την *g* και τον diffusion operator όσες φορές χρειάζεται για να αυξήσουμε ικανοποιητικά την πιθανότητα μέτρησης του x_0.

Μπορεί να αποδειχθεί πως με $O(\sqrt{N})$ διαδοχικές εφαρμογές των δύο παραπάνω τελεστών καταλήγουμε σε κατάσταση που δίνει το επιθυμητό αποτέλεσμα με πιθανότητα μεγαλύτερη του $\frac{1}{2}$. Ο ακριβής αριθμός των επαναλήψεων που χρειάζεται εξαρτάται από το *N*, όμως έχει επίσης δειχθεί ότι η πολυπλοκότητα κάθε δυνατού κβαντικού αλγορίθμου που λύνει το πρόβλημα είναι $O(\sqrt{N})$,δηλαδή ο αλγόριθμος αυτός είναι βέλτιστος.

Η ορθότητα του αλγορίθμου μπορεί να δειχθεί ως εξής: έστω πως η κατάσταση του συστήματος είναι τέτοια ώστε το ιδιοδιάνυσμα $|x_0\rangle$ έχει πιθανότητα *k* και όλα τα υπόλοιπα *l*. Θα δείξουμε πως αν $0 < k < \frac{1}{\sqrt{2}}$ και $l > 0$ τότε η μεταβολή Δk του *k* η οποία προκύπτει από διαδοχική εφαρμογή των δύο τελεστών που δώσαμε είναι φραγμένη κάτω από το $\frac{1}{2\sqrt{N}}$, Επίσης ότι το *l* εξακολουθεί να είναι θετικό.

Από τον ορισμό του diffusion operator είναι εύκολο να δούμε πως αν εφαρμοστεί σε μία κατάσταση ιδιοδιάνυσμα έχει πλάτος $\kappa_2=\left(\frac{2}{N}-1\right)\kappa_1 + 2\frac{N-1}{N}I_1$ και τα υπόλοιπα πλάτη θα γίνουν $I_2 =\left(\frac{2}{N}\kappa_1\right)+\frac{N-1}{N}I_1$. Αυτό προκύπτει από το ότι ο προηγούμενος μέσος όρος είναι $\mu=\frac{\kappa_1+(N-1)I_1}{N}$ και ότι ο *diffusion operator* όταν εφαρμοστεί μετασχηματίζει το πλάτος *κ* στο $\mu - (\kappa - \mu)$ όπως είπαμε όταν δείξαμε πως είναι ισοδύναμος με τη λειτουργία *inversion about average*. Μία ενδιαφέρουσα ιδιότητα είναι πως $\kappa_2^2 + (N-1)I_2^2=\kappa_1^2+(N-1)I_1^2$,η οποία μπορεί να αποδειχθεί και με πράξεις, αλλά προκύπτει επίσης από το γεγονός ότι τα *κ,I* υψωμένα στο τετράγωνο εκφράζουν πιθανότητες και η συνολική πιθανότητα δεν μεταβάλλεται.

Επιπλέον από τα παραπάνω συμπεραίνουμε πως αν $\kappa_1 < 0$ και $I_1 > 0$,τότε μετά την εφαρμογή του diffusion operator $\kappa_2 > 0$ και αν επιπλέον ίσχυει $\left|\frac{\kappa_1}{I_1}\right|<\sqrt{N}$ τότε $I_2>0$.Πιο αναλυτικά ,αφού $\kappa_2=\left(\frac{2}{N}-1\right)\kappa_1 + 2\frac{N-1}{N}I_1$ για $N>2 \Rightarrow \frac{2}{N}-1<0$ και αφού $\kappa_1<0$ έχουμε $\left(\frac{2}{N}-1\right)\kappa_1>0$.Επίσης από $I_1 > 0$ έχουμε $2\frac{N-1}{N}I_1 > 0$ άρα $\kappa_2 > 0$ και αν $\left|\frac{\kappa_1}{I_1}\right|<\sqrt{N}$ τότε για $N\geq 4$ έχουμε $\left|\frac{\kappa_1}{I_1}\right|<\frac{N-2}{2}$ κάτι που εξασφαλίζει πως $I_2>0$.

Το ζητούμενο είναι πως πριν την εφαρμογή της *g* έχουμε πλάτος *κ>0* για τη λύση και *I>0* για όλα τα υπόλοιπα ιδιοδιανύσματα.Τότε μετά την *g* έχουμε κ_1*<0* τη λύση και I_1*<0* για τα υπόλοιπα .Αν 0<κ<$\frac{1}{\sqrt{2}}$ ⇒ $I_1 > \frac{1}{2\sqrt{N}}$ διότι κ_1^2 +(N-1)I_1^2 .Έχουμε *Δκ=*$\kappa_2 - \kappa = \left(\frac{2}{N} - 1\right)\kappa_1 + 2\frac{N-1}{N} I_1 - \kappa$ = $-\left(\frac{2}{N} - 1\right)\kappa + 2\frac{N-1}{N} I_1 - \kappa = -\frac{2}{N}\kappa + 2\frac{N-1}{N} I_1$ το οποίο για *N*≥ 16 μας δίνει το ζητούμενο *Δκ>*$\frac{1}{2\sqrt{N}}$.

Επιπλέον από *κ<*$\frac{1}{\sqrt{2}}$ και *I>*$\frac{1}{N\sqrt{2}}$ προκύπτει ότι $\left|\frac{\kappa_1}{I_1}\right| < \sqrt{N}$ κάτι που μας δίνει και ότι I_2>0.

Δείξαμε λοιπόν πως σε περίπτωση που βρισκόμαστε σε μία κατάσταση όπου το πλάτος της λύσης μας δίνει πιθανότητα μικρότερηαπό $\frac{1}{2}$ η εφαρμογή του αλγορίθμου θα αυξήσει το πλάτος κατά τουλάχιστον $\frac{1}{2\sqrt{N}}$.Φυσικά αν το πλάτος είναι ήδη μεγαλύτερο του $\frac{1}{\sqrt{2}}$ η εφαρμογή του αλγορίθμου πιθανώς θα το μειώσει, οπότε είναι καλό να είμαστε προσεκτικοί όσον αφορά τον ακριβή αριθμό επαναλήψεων που απαιτείται για συγκεκριμένο *N*.

4.3.3 Ο αλγόριθμος παραγοντοποίησης του Shor

Στην συνέχεια θα εξετάσουμε τώρα τον πιο σημαντικό ίσως αλγόριθμο που έχει παρουσιαστεί ως τώρα, τον αλγόριθμο του *Shor*. Μιλώντας με κλασικούς όρους θα μπορούσαμε να πούμε πως είναι ένας αλγόριθμος ο οποίος λύνει σε πολυωνυμικό χρόνο χρησιμοποιώντας ένα μαντείο το οποίο μπορεί να υπολογίσει την περίοδο μιας συνάρτησης επίσης σε πολυωνυμικό χρόνο. Η καινοτομία έγκειται στην κβαντική υλοποίηση του μαντείου. Το πρόβλημα μπορεί να διατυπωθεί ως εξής: έστω ένας φυσικός αριθμός *N*, να βρεθεί ένας μη τετριμμένος διαιρέτης του (δηλαδή ένας διαιρέτης του εκτός των *1,N).* ΄Εχουν προταθεί πολλοί αλγόριθμοι που λύνουν αυτό το πρόβλημα, όπως η μέθοδος βάσεων παραγοντοποίησης και η μέθοδος *Number Field Sieve*. ΄Ολες όμως έχουν εκθετική ως προς το μήκος της εισόδου πολυπλοκότητα. Ο αλγόριθμος του *Shor* είναι ο μόνος ως τώρα γνωστός πιθανοτικός αλγόριθμος πολυωνυμικής πολυπλοκότητας που λύνει το πρόβλημα της παραγοντοποίησης.

Τα βήματα του αλγορίθμου είναι:

1. Επιλέγουμε έναν τυχαίο αριθμό *α < N*
2. Υπολογίζουμε το *gcd(α,N).* Μπορούμε να χρησιμοποιήσουμε τον αλγόριθμο του Ευκλείδη. Αν ο μέγιστος κοινός διαιρέτης είναι ≠ 1 τότε βρήκαμε έναν μη τετριμμένο διαιρέτη και το πρόβλημα λύθηκε.
3. Υπολογίζουμε την περίοδο της συνάρτησης *f(x) =* α^x *mod N*. Στο βήμα αυτό χρησιμοποιούμε το κβαντικό μαντείο το οποίο θα περιγραφεί παρακάτω. ΄Εστω πως η περίοδος που υπολογίστηκε είναι *r*.
4. Αν ο r είναι περιττός επιστρέφουμε στο βήμα 1

5. Αν $\alpha^{\frac{r}{2}} \equiv -1 (mod N)$ επιστρέφουμε στο βήμα 1
6. Σε αντίθετη περίπτωση έχουμε $\alpha^{\frac{r}{2}} \equiv 1 (mod N) \Longrightarrow \alpha^r \equiv 1 (mod N) \Longrightarrow \alpha^r - 1 = \kappa N$

$\Longrightarrow \left(\alpha^{\frac{r}{2}} - 1\right)\left(\alpha^{\frac{r}{2}} + 1\right) = \kappa N$. Χρησιμοποιώντας τον αλγόριθμο του μέγιστου κοινού διαιρέτη μπορούμε να βρούμε έναν μη τετριμμένο διαιρέτη του *N,* διότι $\alpha^{\frac{r}{2}} + 1 \neq mN$ όπως εξασφαλίσαμε στο προηγούμενο βήμα.

Θα ασχοληθούμε τώρα με το κβαντικό μέρος του αλγορίθμου, το οποίο είναι και το πιο ενδιαφέρον. Θέλουμε να υπολογίσουμε την περίοδο της συνάρτησης $f(x) = \alpha^x \, mod \, N$. Ακολουθούμε τα εξής βήματα:

1. Ξεκινάμε με δύο καταχωρητές των $n = log_2 N$ *Nqbits* ο καθένας. Ο πρώτος αρχικοποιείται σε μία υπέρθεση όλων των δυνατών καταστάσεων με την ίδια πιθανότητα (κάτι τέτοιο μπορεί να επιτευχθεί με πύλες *Hadamard)* ενώ ο δεύτερος αρχικοποιείται στο $|0\rangle$.
2. Με την προϋπόθεση ότι έχουμε υλοποιήσει την *f* ως κβαντική συνάρτηση την εφαρμόζουμε στο σύστημα. Τώρα ο δεύτερος καταχωρητής περιέχει το *f(x)* όπου *x* η κατάσταση στην οποία ήταν ο πρώτος.
3. Εφαρμόζουμε τον κβαντικό μετασχηματισμό Fourier στον πρώτο καταχωρητή και μετράμε το περιεχόμενό του δεύτερου. Τώρα μία μέτρηση του πρώτου έχει μεγάλη πιθανότητα να μας δώσει την περίοδο της συνάρτησης.

Ας προσπαθήσουμε να εξηγήσουμε λίγο πιο αναλυτικά γιατί ο παραπάνω αλγόριθμος δουλεύει. Μετά το βήμα 1 η κατάσταση του συστήματος είναι $\frac{1}{\sqrt{N}}(1\ 1\ .\ .\ .\ 1)^T \otimes (1\ 0\ .\ .\ .\ 0)^T$.Η εφαρμογή της *f* θα δημιουργήσει entanglement ανάμεσα στους δύο καταχωρητές. ΄Ετσι αν μετά την εφαρμογή της f αποπειραθούμε μία μέτρηση στον δεύτερο καταχωρητή και πάρουμε αποτέλεσμα y_0 τότε ο πρώτος καταχωρητής θα έρθει σε μία κατάσταση η οποία θα είναι υπέρθεση των *x* για τα οποία ισχύει $f(x) = y_0$.

Δηλαδή αν *s* τομικρότερο τέτοιο *x* θα έρθει σε μία υπέρθεση των $\left|s + rj\right\rangle, 0 \leq j \leq \left\lceil \frac{N}{r} \right\rceil - 1$. Δυστυχώς δεν είναι δυνατόν από μία τέτοια υπέρθεση να συμπεράνουμε το *r* γιατί το *s* είναι τυχαίο. Αν όμως εφαρμόσουμε έναν μετασχηματισμό *Fourier* στον πρώτο καταχωρητή το φάσμα πιθανοτήτων που θα προκύψει θα είναι ανεξάρτητο του *s* . Συγκεκριμένα αν το *r* διαιρεί το *N* τότε θα έχουμε μη μηδενική πιθανότητα να μετρήσουμε μόνοπολλαπλάσιατου *N/r*. Αλλά ακόμα και σε περίπτωση που το *r* δεν διαιρεί το *N* το φάσμα θα εμφανίσει κορυφές στα πολλαπλάσια του *N/r*. Ετσι μία μέτρηση θα μας

δώσει με μεγάλη πιθανότητα μία τιμή *c* κοντά στολ *N/r*. Φυσικά το *N* είναι γνωστό,οπότε ουσιαστικά

αποκτούμε μία προσέγγιση του *λ/r.* Για να μπορέσουμε από αυτό το κλάσμα να προσδιορίσουμε το *r* θα πρέπει οι *λ,r* να είναι πρώτοι μεταξύ τους, κάτι που συμβαίνει με πιθανότητα μεγαλύτερη του *1/lnr*,οπότεθαχρειαστούμετοπολύ *O(n*) προσπάθειες για να βρούμε το *r* μεπιθανότητα που μπορεί να πλησιάσει όσο θέλουμε το *1*. Φυσικά η επαλήθευση μπορεί να γίνει απλά με τον έλεγχο για το αν *f(x) = f(x+ r).*

4.3.4 Ο κβαντικός μετασχηματισμός Fourier

Οπως είδαμε ο μετασχηματισμός Fourier είναι ένα κρίσιμο κομμάτι του αλγορίθμου του *Shor*. Σε αυτήν την ενότητα ορίζουμε αναλυτικά τον μετασχηματισμό και παρουσιάζουμε ένα παράδειγμα υλοποίησής του σε *nQML*.

Ο κβαντικός μετασχηματισμός σε *braket notation*, με είσοδο μία κατάσταση $|x\rangle$ δίνεται ως:

$$|x\rangle \ \rightarrow \frac{1}{\sqrt{N}} \sum_{\kappa=0}^{N-1} e^{2\pi i x k/N} \, |k\rangle$$

$$\frac{1}{\sqrt{N}} [\omega^{kl}], \omega = e^{2\pi i/N}$$

Με τη βοήθεια του θεωρήματος *Plancherel* μπορεί να δειχθεί πως ο μετασχηματισμός αυτός είναι ορθομοναδιαίος. Ας δούμε αναλυτικά τη μορφή του πίνακα του μετασχηματισμού ώστε να βοηθηθούμε στη μετάφρασή του σε *nQML*. Για δύο *bit* ο πίνακας παίρνει τη μορφή:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & e^{\frac{\pi i}{2}} & e^{\pi i} & e^{\frac{3\pi i}{2}} \\ 1 & e^{\pi i} & e^{2\pi i} & e^{3\pi i} \\ 1 & e^{\frac{3\pi i}{2}} & e^{3\pi i} & e^{\frac{9\pi i}{2}} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \iota & -1 & -\iota \\ 1 & -1 & 1 & -1 \\ 1 & -\iota & -1 & \iota \end{pmatrix}$$

Παρατρούμε πως τα στοιχεία της πρώτης στήλης μπορούν να γραφούν ως $\frac{1}{2}\binom{1}{1} \otimes \binom{1}{1}$

Ομοίως τα στοιχεία της δεύτερης γράφονται ως $\frac{1}{2}\binom{1}{e^{\pi i}} \otimes \binom{1}{e^{\frac{\pi i}{2}}}$. Συνολικά τα στοιχεία

της *k* στήλης γράφονται ως ως $\frac{1}{2}\binom{1}{e^{k\pi i}} \otimes \binom{1}{e^{\frac{k\pi i}{2}}}$ αν η αρίθμηση των στηλών ξεκινά από το *0*. Το σκεπτικό αυτό μπορεί να επεκταθεί και στον μετασχηματισμό *Fourier* n*bits.* ΄Ετσι τα στοιχεία της *k* στήλης είναι:

$$2^{-n/2}\binom{1}{e^{k\pi i}} \otimes \binom{1}{e^{\frac{k\pi i}{2}}} \otimes \ldots . \otimes \binom{1}{e^{k\pi i/2^{n-1}}}$$

Ορίζουμε τώρα σε *Haskell* τη βοηθητική συνάρτηση *make_ifms* η οποία παίρνει ως παράμετρο τους όρους που δίνουν τα bit στα οποία θα εφαρμοστεί ο μετασχηματισμός, μία συνάρτηση που με είσοδο έναν ακέραιο μας δίνει τον όρο του αντίστοιχου branch, καθώς και ένα πεδίο αποδεκτών ακεραίων για τη συνάρτηση αυτή. Αποτέλεσμα είναι το δέντρο από if^-που επιτελούν τον επιθυμητό μετασχηματισμό.

```
make_ifms:: [TermA] → (Int → TermA) →[Int] →TermA

make_ifms [ ] f dom I length dom==1=f (head dom)

make_ifms  1 f dom I (2^length 1))== (length dom) =

                    let branch1= make_ifms (tall 1) f

                            (take (dlv (length dom) 2 )  In

                    Let branch2=make+ifms (tall 1)  f

                                (dom \\ (take  (dlv  (length dom)  2) dom))  In

                    Ifm  (head 1)  branch2 branch1
```

Για να χρησιμοποιήσουμε την *make_ifms* θα πρέπει να ορίσουμε μία συνάρτηση η οποία με όρισμα ένα να κέραιο *n* θα μας δίνει τον όρο που αντιστοιχεί στην n στήλη του μετασχηματισμού *Fourier*. Αυτό είναι

εύκολο να γίνει με την ανάλυση του μετασχηματισμού σε γινόμενα που κάναμε παραπάνω. Ας δούμε την υλοποίηση σε *Haskell* αυτής της συνάρτησης για *3 qbit.*

```
dft_branch : : Int → TermA

dft_branch n  =  Pair  (Pair

                              (Super con2 Qfalse  (con2∗(coef  (4∗n)))   Qtrue

                              (Super con2 Qfalse  (con2∗(coef  (2∗n)))   Qtrue

                              (Super con2 Qfalse  (con2∗(coef  (n))       Qtrue
```

Στον παραπάνω κώδικα *con2* είναι η σταθερά $\frac{1}{\sqrt{2}}$ και *coef* είναι μία συνάρτηση που επιστρέφει το $e^{n\pi i/4}$.

Μπορούμε να χρησιμοποιήσουμε τα παραπάνω για να ελέγξουμε μία εύκολη περίπτωση του μετασχηματισμού *Fourier.* Παρακάτω βλέπουμε τον κώδικα *Haskell* που παράγει έναν όρο ο οποίος εφαρμόζει τον μετασχηματισμό Fourier σε τρία *bit*.

```
test_dft = Let "a" (Super con2 Qfalse con2 Qtrue)

      (Let "b" (Super con2 Qfalse con2 Qtrue)

      (Let "c" (Qtrue)

          (make_ifms    [(Var " a ") ,( Var   " b ") ,(Var  " c " )]   dft_branch   [0..7])))
```

Αρχικά τα *a,b,c* σχηματίζουν μία κατάσταση με περίοδο *2,* αφού έχουμε μία υπέρθεση των $|xy1\rangle$. Η εφαρμογή του μετασχηματισμού *Fourier* δίνει μία υπέρθεση των $|100\rangle$ και $|000\rangle$.Προφανώς αν μια μέτρηση δώσει αποτέλεσμα $|000\rangle$ δεν έχουμε αποτέλεσμα και πρέπει να ξαναπροσπαθήσουμε . Ομως με πιθανότητα *1/2* μία μέτρηση θα μας δώσει $|100\rangle$ *= 4* . ΄Αρα έχουμε $\frac{\lambda}{r}$*= 4/8 =1/2 → r=2* που είναι και το ζητούμενο. Χρήσιμο είναι να παρατηρήσουμε πως σε περίπτωση που το *c* είναι *Qfalse* το αποτέλεσμα δεν αλλοιώνεται.

Ας δούμε τώρα και την περίπτωση που το *b* είναι *Qfalse* και το *c Qtrue*, δηλαδή έχουμε $|001\rangle + |101\rangle$.Η εφαρμογή του *Fourie*r μας δίνει $|000\rangle + |010\rangle + |100\rangle + |110\rangle$. Ας δούμετα πιθανά αποτελέσματα για την κάθε κατάσταση στον παρακάτω πίνακα:

Αποτέλεσμα μέτρησης	***c/N=λ/r***	***r***
0	0	-
2	1/4	4
4	1/2	2
6	3/4	4

Πίνακας 4.1 *Αποτελέσματα μέτρησης*

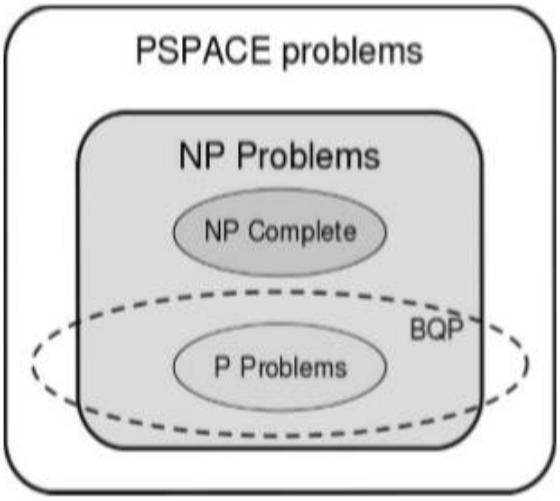
Παρατηρούμε πως και πάλι με πιθανότητα *1/2* μετράμε ένα αποτέλεσμα που μας οδηγεί στη σωστή περίοδο. Για άλλη μια φορά να σημειώσουμε πως αν αλλάξουμε τις τιμές των *b,c* το αποτέλεσμα αυτό δεν αλλάζει αρκεί αυτές να παραμείνουν κλασικές (δηλαδή *Qtrue* ή *Qfalse*).

4.4 Στοιχεία κβαντικής θεωρίας πολυπλοκότητας

Στην θεωρεία πολυπλοκότητας με *BQP*, από τα αρχικά *Bounded Error, Quantum, Polynomial*, συμβολίζουμε την κλάση των προβλημάτων απόφασης πουλύνονται από έναν κβαντικό υπολογιστή σε πολυωνυμικό χρόνο με την πιθανότητα η απάντηση να είναι λανθασμένη να είναι φραγμένη από μία σταθερά γνησίως μικρότερη του $\frac{1}{2}$. Έτσι μπορούμε να επαναλάβουμε τον αλγόριθμο αρκετές φορές ώστε η πλειοψιφία των αποτελεσμάτων που θα πάρουμε να εκφράζει το σωστό αποτέλεσμα με πιθανότητα όσο κοντά στο *1* θέλουμε. Το κλασικό ανάλογο της κλάσης *BQP* είναι η κλάση *BPP.*

Η σχέση της κλάσης προβλημάτων *BQP* με τις *NP, NP-complete* και *P* δεν έχει μέχρι σήμερα αποσαφηνιστεί. Πιστεύεται όμως ότι η *P* είναι υποσύνολο της *BQP*, καθώς και ότι η *NP-complete* είναι ξένη με την *BQP*. Το πρόβλημα απόφασης που αντιστοιχεί στην παραγοντοποίηση ακεραίων, δηλαδή δοθέντων φυσικών αριθμών *N* και *M* με $1 \leq M \leq N$ να απαντηθεί αν υπάρχει διαιρέτης *d* του *N* τέτοιος ώστε $1 < d < M$, είναι ένα παράδειγμα προβλήματος που ανήκει στην *BQP* αλλά πιστεύεται ότι δεν ανήκει στην *BPP.* Μέχρι στιγμής δεν έχει αποδειχθεί ότι κάποιο *NP-complete* πρόβλημαανήκειστην *BQP.*

Να σημειώσουμε εδώ για μία ακόμα φορά ότι οι κβαντικοί υπολογισμοί δεν αλλοιώνουν την έννοια του υπολογίσιμου. Δηλαδή τα υπολογιστικά όρια των κβαντικών υπολογιστών είναι τα ίδια με αυτά των κλασικών,δεν μπορούν να λύσουν προβλήματα όπως το *haltingproblem*.Απλά σε κάποιες περιπτώσεις ορισμένοι υπολογισμοί πραγματοποιούνται αποδοτικότερα από έναν κβαντικό υπολογιστή σε σχέση με έναν κλασικό.



Σχήμα 4.2 *Η πιθανολογούμενη σχέση της BQP με άλλες κλάσης πολυπλοκότητας*

Κεφάλαιο 5

Μετάφραση nQML σε κυκλώματα

5.1 Εισαγωγή

Στην ενότητα αυτή παρουσιάζουμε την πρωτότυπη εργασία της μετάφρασης των εντολών της γλώσσας *nQML* σε κυκλώματα. Όπως είδαμε η σημασιολογία των εντολών της γλώσσας είναι οι ορθομοναδιαίοι πίνακες και μετασχηματισμοί. Όμως οι περισσότεροι ενδιαφέροντες κβαντικοί αλγόριθμοι υλοποιούνται με τη βοήθεια κβαντικών κυκλωμάτων κάτι αναμενόμενο αφού οι στοιχειώδεις λειτουργίες του κβαντικού προγραμματισμού βρίσκονται πολύ πιο κοντά στο επίπεδο του *hardware* και εξαρτώνται ισχυρά από αυτό. Οι γλώσσες προγραμματισμού από την άλλη προσπαθούν να ισχυροποιήσουν τα κβαντικά προγράμματα και να δημιουργήσουν ένα αφαιρετικό μοντέλο το οποίο επιτρέπει στους προγραμματιστές να μην ασχολούνται με τις υλοποιήσεις στο υλικό των κβαντικών υπολογιστών. Φυσικά σε τελική ανάλυση οι λειτουργίες της γλώσσας θα πρέπει να υλοποιηθούν σε υλικό.

Επίσης η ενότητα εξυπηρετεί λοιπόν δύο σκοπούς. Πρώτον βοηθάει στην καλύτερη κατανόηση των προγραμμάτων καθώς η μετάφραση τους σε κυκλώματα τα κάνει πιο «χειροπιαστά» γιατί όπως είπαμε οι αλγόριθμοι υλοποιούνται σε κυκλώματα. Δεύτερον δίνει μια πρώτη ιδέα για το πώς πρόκειται να λειτουργήσουν οι κβαντικοί υπολογιστές του μέλλοντος και ποιες λειτουργίες θα πρέπει να υλοποιούν ώστε να εκτελούν εντολές της *nQML* ή άλλης παρόμοιας γλώσσας.

5.2 Κλάσεις κυκλωμάτων

Ορίζονται διάφορες κατηγορίες κβαντικών κυκλωμάτων με βάση τις οποίες κατασκευάζονται τα διάφορα κβαντικά κυκλώματα με τα οποία ορίζεται η σημασιολογία της γλώσσας *QML*. Τις ίδιες κατηγορίες θα χρησιμοποιήσουμε και εμείς για να μεταφράσουμε τις εντολές τις *nQML* σε κυκλώματα.

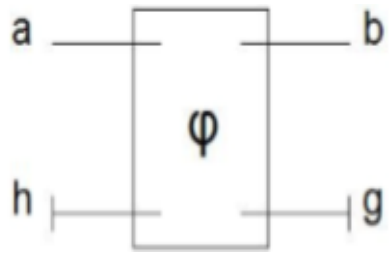
Στην αρχή ορίζεται η κατηγορία $FQC^{\simeq}$ η οποία περικλείει τους πεπερασμένους κβαντικούς μετασχηματισμούς *(Finite Quantum Computations),* οι οποίοι έχουν τον περιορισμό ότι είναι

αντιστρέψιμοι (εκθέτης $\simeq$). Τα μέλη της κατηγορίας είναι μορφισμοί – συναρτήσεις από ένα πλήθος *α qubit* σε ένα πλήθος *b qubit.* Έτσι ορίζουμε υποκατηγορίες $FQC^{\simeq}$ *a b* οι οποίες περιέχουν όλα τα κυκλώματα με

a qubit εισόδου και *b qubit* εξόδου.Όπως αναφέραμε οι μετασχηματισμοί αυτοί είναι αντιστρέψιμοι και άρα πρέπει η είσοδος να έχει το ίδιο μέγεθος με τη έξοδο ή αλλιώς λέμε ότι $FQC^{\simeq}\ \alpha\ b = \{\}$ όταν $a \neq b$. Ορίζουμε επίσης ότι $FQC^{\simeq}\ \alpha\ \alpha = FQC^{\simeq}\ \alpha$ που είναι το σύνολο των αντιστρέψιμων μετασχηματισμών πάνω σε ένα σλυνολο *α qubit* . Όλα τα κυκλώματα της κατηγορίας $FQC^{\simeq}$ (που αποτελείται από όλες τις υποκατηγορίες $FQC^{\simeq}\ a$) κατασκευάζονται αναδρομικά από ένα αριθμό στοιχειωδών κυκλωμάτων και πράξεων ανάμεσά τους όπως θα δούμε στην επόμενη ενότητα.

Ο *Grattage* επεκτείνει την κατηγορία των αντιστρέψιμων κυκλωμάτων ορίζοντας την κατηγορία *FQC* οι οποία αποτελείται επιπλέον από τα μη αντιστρέψιμα κβαντικά κυκλώματα. Η μη αντιστρεψιμότητα περιλαμβάνει μετρήσεις ή / και αρχικοποιήσεις νέων *qubit*: διεργασίες που δεν είναι αντιστρέψιμες. Έτσι τα νέα κυκλώματα περιέχουν και δύο νέα σύνολα *qubit*, εισόδου και εξόδου, που ονομάζονται σωρός *(heap)* και σκουπίδια *(garbage*), αντίστοιχα. Τα *qubits* στο σωρό θεωρούμε ότι είναι πάντα αρχικοποιημένα σε μια τιμή της βάσης του διανυσματικού χώρου, συνήθως $|0\rangle$ και συμβολίζονται στο κύκλωμα με $\vdash$. Αντίθετα τα *qubits* στα σκουπίδια είναι τα *qubits* που απελευθερώνονται από το πρόγραμμα ή γίνεται μια κβαντική μέτρηση πάνω τους (επομένως ίσως αποτελούν και το αποτέλεσμα του αλγορίθμου δεν είναι άχρηστα!) και συμβολίζονται με $\dashv$. Ουσιαστικά κάθε κύκλωμα της κατηγορίας $FQC\ a\ b$ *(a είσοδοι – b έξοδοι)* είναι μια τριάδα *φ=(h,g,φ')* όπου $h \in \mathbb{N}$ *είναι* το μέγεθος σε *qubits* του σωρού, όλα αρχικοποιημένα σε $|0\rangle$, $g \in \mathbb{N}$ είναι το πλήθος των σκουπιδιών και $\varphi' \in FQC^{\simeq}\ c$ ένας αντιστρέψιμος μετασχηματισμός με *c = a+h = b+g.* Ο γενικός συμβολισμός αυτών των κυκλωμάτων φαίνεται στο παρακάτω σχήμα.

Κυκλωματικός συμβολισμός ενός υπολογισμού στο FQC, όπου ένας αντιστρέψιμος μετασχηματισμός στο $FQC^{\simeq}$. Φαίνονται οι είσοδοι και έξοδοι, σωρού και σκουπιδιών αντίστοιχα.



Σχήμα 5.1 *Κυκλωματικός συμβολισμός ενός υπολογισμού στο FQC*

Φυσικά κάθε μετασχηματισμός που ανήκει στην κατηγορία $FQC^{\simeq}$ ανήκει και στην κατηγορία FQC αν θεωρήσουμε μηδενικό μέγεθος σωρού και σκουπιδιών.Για παράδειγμα αν $\varphi \in FQC^{\simeq}\alpha$ κατασκευάζουμε τον $\varphi' \in FQC\ \alpha\ \alpha$ με *φ'=(0,0,φ).*

Τέλος, ιδιαίτερο ενδιαφέρον παρουσιάζει η κατηγορία των κυκλωμάτων τα οποία δεν περιλαμβάνουν μετρήσεις αλλά μπορούν να περιλαμβάνουν αρχικοποιήσεις. Αυτή η κατηγορία των αγνών ή αυστηρών υπολογισμών (θυμηθείτε τις αγνές παραγωγές εκφράσεων τις γλώσσας *nQML*) συμβολίζεται με FQC^o και αποτελείται από τα κυκλώματα με κενό το σύνολο των σκουπιδιών ,δηλαδή *α=(h,φ)* ∈ FQC^o *a b.*Όπως φαίνεται *g=0* και άρα παραλείπεται στον ορισμό το σύνολο των σκουπιδιών. Οι αγνοί κβαντικοί υπολογισμοί δεν είναι απλώς αντιστρέψιμοι μετασχηματισμοί, καθώς επιτρέπονται αρχικοποιημένες είσοδοι. Παρόλα αυτά η έλλειψη σκουπιδιών εξόδου σημαίνει ότι οι μορφισμοί του FQC^o μπορούν να μοντελοποιηθούν ως γραμμικοί μετασχηματισμοί από αγνές κβαντικές καταστάσεις σε αγνές κβαντικές καταστάσεις. Αυτό το έχουμε ξαναδεί στη σημασιολογία της *nQML* όπου οι αγνές εκφράσεις αντιστοιχίζονται σε πίνακες μετασχηματισμού (γραμμικοί ορθομοναδιαίοι) ενώ οι μη αγνές σε συναρτήσεις. Οι μετρήσεις της κλάσης FQC προκαλούν την κατάρρευση των κυματοσυναρτήσεων των *qubit* και επομένως οδηγούν σε πιθανοτικά αποτελέσματα. Συνεπώς οι μη αντιστρέψιμοι μετασχηματισμοί δεν μπορούν να μοντελοποιηθούν χρησιμοποιώντας γραμμικές αντιστοιχήσεις μεταξύ αγνών καταστάσεων. Προκύπτει ότι το σύνολο FQC^o είναι το μεγαλύτερο υποσύνολο της κατηγορίας FQC το οποίο μπορεί να μοντελοποιηθεί χρησιμοποιώντας γραμμικούς μετασχηματισμούς.

Από τους ορισμούς των $FQC^{\simeq}$ (όχι σωρός ή σκουπίδια), FQC^o (όχι σκουπίδια αλλά επιτρέπεται σωρός) και FQC (επιτρέπεται σωρός και σκουπίδια) προκύπτει ότι :

$$FQC^{\simeq} \subset FQC^o \subset FQC$$

5.3 Κατασκευή κυκλωμάτων $\boldsymbol{FQC^{\simeq}}$

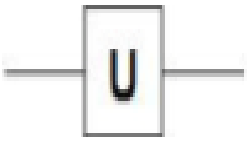
Έδω θα ασχοληθούμε με την αναδρομίκη κατασκευή των αντιστρέψιμων κυκλωμάτων , δηλαδή κυκλωμάτων της κλάσης $FQC^{\simeq}$. Όπως είδαμε και παραπάνω κάθε κύκλωμα που ανήκει στην κλάση FQC που είναι η γενική μορφή των κβαντικών κυκλωμάτων αποτελείται στην ουσία από ένα

αντιστρέψιμο κύκλωμα με αρχικοποιήσεις στην αρχή και μετρήσεις στο τέλος. Τα κυκλώματα στην κατηγορία $FQC^{\simeq}\ \alpha$ που είναι τα αντιστρέψιμα κβαντικά κυκλώματα με *α* εισόδους ορίζονται επαγωγικά ως εξής:

• Περιστροφές –Rotation

rot u ∈ $\boldsymbol{FQC^{\simeq}}$ 1: Δηλώνει μια περιστροφή που δρα πάνω σε ένα μόνο *qubit*, όπου ο *u* είναι ένας ορθομοναδιαίος πίνακας. Η αντίστροφη διαδικασία είναι ο αναστροφοσυζυγής μετασχηματισμός δηλαδή ο

$\boldsymbol{\varphi^{-1} = \mathbf{rot}\, u^{+} \in FQC \simeq 1}$ Σε κυκλωματικό διάγραμμα αυτός ο μετασχηματισμός. Σε κυκλωματικό διάγραμμα αυτός ο μετασχηματισμός συμβολίζεται ως εξής:



Σχημα 5.2 *Αυθαίρετο κύκλωμα rotation*

Ο πίνακας *u* πρέπει να είναι ορθομοναδιαίος και της μορφής $\begin{pmatrix} \boldsymbol{\lambda_0} & \boldsymbol{\lambda_1} \\ \boldsymbol{\kappa_0} & \boldsymbol{\kappa_1} \end{pmatrix}$ με $\boldsymbol{e_0^* \kappa_0 + e_1^* \kappa_1 = 1}$. Ο αντίστροφος του δίνεται από το $\boldsymbol{u^{\dagger}} = \begin{pmatrix} \boldsymbol{\lambda_0} & \boldsymbol{\lambda_1} \\ \boldsymbol{\kappa_0} & \boldsymbol{\kappa_1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\lambda_0^*} & \boldsymbol{\kappa_0^*} \\ \boldsymbol{\lambda_1^*} & \boldsymbol{\kappa_1^*} \end{pmatrix}$. Ειδική περίπτωση αποτελεί η άρνηση που είναι το κύκλωμα ***rot X*** με **X**=$\begin{pmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$.

•Καλωδιώσεις –Wires

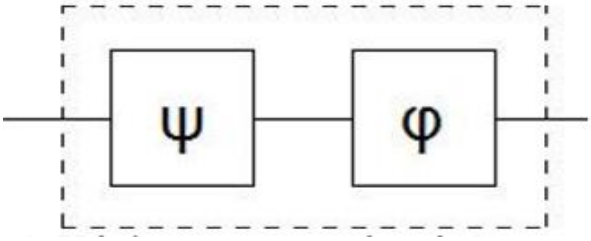
Κάθε πιθανή αναδιάταξη των qubits είναι ένας έγκυρος μετασχηματισμός και στο κυκλωματικό διάγραμμα φαίνεται ως ένα ανακάτεμα των καλωδίων όπως στο σχήμα.

Σχήμα 5.3 *Παράδειγμα κυκλώματος wires*

Είναι το κύκλωμα wires $\boldsymbol{\varphi \in FQC \simeq a}$ όπου $\boldsymbol{\varphi}$ ο ισομορφισμός $\boldsymbol{\varphi:[a] \simeq [a]}$ και $\boldsymbol{[a]}$ είναι το αρχικό τμήμα του φυσικού , το οποίο ορίζεται ως $\boldsymbol{\{ i \in \mathbb{N} \mid < a\}}$. Η έκφραση περιγράφει κάθε δυνατή αναδιάταξη , συμπεριλαμβανομένης και της ταυτοτικής $\boldsymbol{id_a = wires\ id}$ με την οποία δεν συμβαίνει καμία αλλαγή . Η αναδιάταξη είναι τετριμμένα αντιστρέψιμη και η τάξη του κυκλώματος είναι ίση με τον αριθμό των καλωδίων στο κύκλωμα, που είναι το $\boldsymbol{a}$ του ισομορφισμού.

•Σειριακή σύνθεση

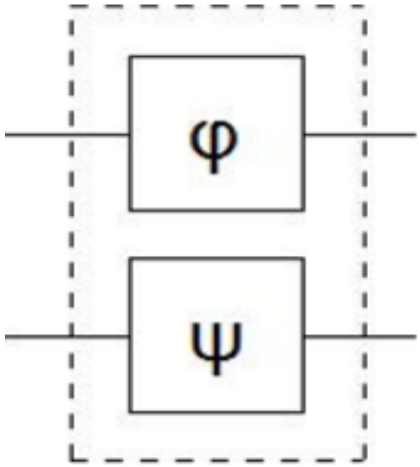
Δεδομένων δύο κυκλωμάτων $\boldsymbol{\varphi \in FQC \simeq a}$ και $\boldsymbol{\psi \in FQC \simeq a}$ δηλαδή δύο κυκλωμάτων με τον ίδιο αριθμό *qubit*, μπορεί να κατασκευαστεί η σύνθεσή τους $\boldsymbol{\varphi \circ \psi \in FQC \simeq a}$ που συμβολίζεται ως εξής:



Σχήμα 5.3.1 *Κύκλωμα σειριακής σύνθεσης*

•Παράλληλη σύνθεση

Δεδομένων δύο κυκλωμάτων $\boldsymbol{\varphi \in FQC \simeq a}$ και $\boldsymbol{\psi \in FQC \simeq a}$ μπορεί να κατασκευαστεί η παράλληλη σύνθεση τους $\boldsymbol{\varphi \otimes \psi \in FQC \simeq (a+b)}$ που συμβολίζεται ως εξής:

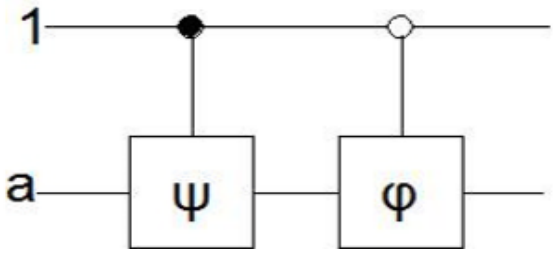


Σχήμα 5.4 *Κύκλωμα παράλληλης σύνθεσης*

Αυτή η κατασκευή συνδυάζει τα δύο κυκλώματα εν παραλλήλω και μπορούμε να πούμε ότι αποτελεί το τανυστικό γινόμενο των δύο κυκλωμάτων και των κβαντικών καταστάσεων των επιμέρους *qubit*. Η τάξη του νέου κυκλώματος είναι ίση με το άθροισμα των τάξεων των δύο υπό – κυκλωμάτων, δηλαδή ***a+b .*** Το αντίστροφο κύκλωμα είναι προφανώς το $\boldsymbol{\varphi^{-1} \otimes \psi^{-1}}$ $\boldsymbol{FQC \simeq (a+b).}$

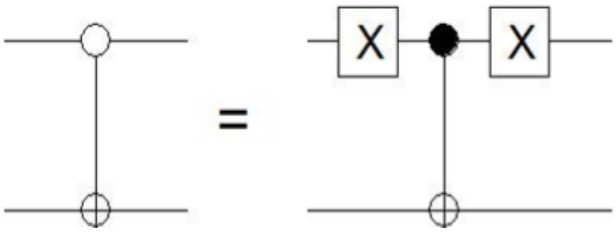
•Εκτέλεση υπό συνθήκη

Δεδομένων δύο κυκλωμάτων $\boldsymbol{\varphi,\psi \in FQC \simeq a}$ κατασκευάζουμε την υ Φ πό συνθήκη εκτέλεση ***if qubit then ψ else φ*** με το κύκλωμα $\boldsymbol{\varphi / \psi \in FQC \simeq (1+\alpha)}$. Φυσίκα η διαφορά από την κλασική αυτή εντολή είναι ότι το *qubit* ελέγχου μπορεί να βρίσκεται σε κβαντική υπέρθεση και επομένως κάθε υπό-κύκλωμα εφαρμόζεται στις ανάλογες συνιστώσες των *qubit* εισόδου. Το κυκλωματικό διάγραμμα είναι το παρακάτω:



Σχήμα 5.5 *Κύκλωμα υπό-συνθήκη εκτέλεσης*

Ο συμβολισμός με το άσπρο κυκλάκι είναι ουσιαστικά το αντίστοιχο με την υπό συνθήκη εκτέλεση αλλά μόνο αν το *qubit* ελέγχου είναι *0*. Το κύκλωμα μπορεί να κατασκευαστεί από τις πύλες του πλήρους συνόλουπου δώσαμε ως εξής:



Σχήμα 5.6 *Ελεγχόμενη διαδικασία με μία NOT πύλη να εφαρμόζεται στο δεύτερο qubit, με την προϋπόθεση το πρώτο qubit να έχει τεθεί στο μηδέν*

Όπως αναφέραμε στην ενότητα των κβαντικών κυκλωμάτων κάθε τέτοια ελεγχόμενη κατασκευή είναι δυνατή, αφού κάθε κύκλωμα (και ειδικότερα αυτά της κλάσης $\boldsymbol{FQC}^{\simeq}$ μπορεί να αναλυθεί σε κυκλώματα αποτελούμενα μόνο από μετασχηματισμούς *1-qubit* και πύλες *CNOT*. Εάν είτε το φ είτε το ψ ειναι το ταυτοτικό κύκλωμα τότε ο αντίστοιχος κλάδος απαλείφεται εντελώς. Τέλος το αντίστροφο κύκλωμα δίνεται πάλι με τη χρήση των $\boldsymbol{\varphi^{-1}}$ και $\boldsymbol{\psi^{-1}}$ και είναι το $\boldsymbol{\varphi^{-1}/\psi^{-1} \in FQC^{\simeq}(1+\alpha)}$.

5.4 Κατασκευή κυκλωμάτων FQC

Όπως αναφέραμε και πιο πάνω κάθε κύκλωμα της κλάσης $\boldsymbol{FQC}^{\simeq}$ μπορεί να επεκταθεί στην κλάση $\boldsymbol{FQC}$.Επομένως μπορούμε να πούμε ότι η κλάση $\boldsymbol{FQC}$ περιέχει αρχικά όλα τα κυκλώματα που δημιουργούνται με αυτό τον τρόπο. Επίσης πρέπει να δώσουμε και ορισμούς σύνθεσης κυκλωμάτων.

Η σειριακή σύνθεση δύο κυκλωμάτων τα οποία δίνονται ως μέλη της $\boldsymbol{FQC}$ ορίζεται με τον ίδιο τρόπο όπως και στην κατηγορία των αντιστρέψιμων κυκλωμάτων. Δοσμένων δύο κυκλωμάτων

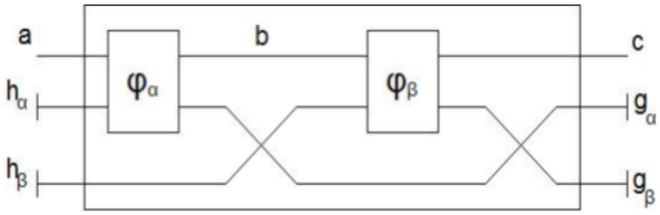
α= $(\boldsymbol{h_\alpha, g_\alpha, \varphi_\alpha}) \in \boldsymbol{FQC}$ **a b** και **β=**$(\boldsymbol{h_\beta, g_\beta, \varphi_\beta}) \in \boldsymbol{FQC}$ ***b c*** η σειριακή τους σύνθεση δίνεται από το κύκλωμα $\boldsymbol{\beta \circ \alpha = (h,g,\varphi)} \in \boldsymbol{FQC}$ ***a c*** όπου

$$h= h_\alpha + h_\beta$$

$$g= g_\alpha + g_\beta$$

$$\varphi_{\beta \circ \alpha} = (\varphi_\beta \otimes id_{g_\alpha}) \circ (\varphi_\alpha \otimes id_{h_\beta})$$

του οποίου το διάγραμμα φαίνεται παρακάτω.



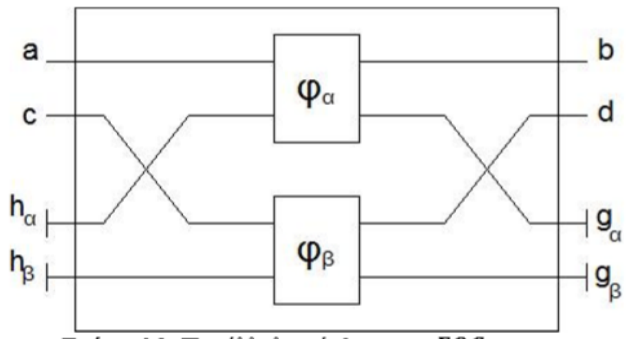
Σχήμα 5.7 *Σειριακή σύνθεση στο* FQC: $\varphi_{\beta \circ \alpha}$

Η παράλληλη σύνθεση δύο κυκλωμάτων $\alpha = (h_a, g_a, \varphi_\alpha) \in FQC$ ***a b και β***$=(h_\beta, g_\beta, \varphi_\beta) \in FQC$ ***c d*** γίνεται με παρόμοιο τρόπο . Είναι το κύκλωμα $\alpha \otimes \beta = (h,g,\varphi) \in FQC$ ***(a + c) (b+d)*** όπου

$$h= h_\alpha + h_\beta$$

$$g= g_\alpha + g_\beta$$

$$\varphi_{\beta \circ \alpha} = (\varphi_\beta \circ \varphi_a)$$



Σχήμα 5.8 *Παράλληλη σύνθεση στο* FQC: $\varphi_{\alpha} \otimes_{\beta}$

Η εκτέλεση υπό συνθήκη είναι προφανής καθώς η συνθήκη εφαρμόζεται απλώς στο αντιστρέψιμο τμήμα του κυκλώματος.

Παρατηρήστε ότι σε όλες τις συνθέσεις τα καλώδια αρχικοποίησης και μετρήσεων μετακινούνται πάντα στην αρχή και το τέλος του κυκλώματος αντίστοιχα. Επομένως κάθε κύκλωμα $\boldsymbol{FQC}$ αποτελείται πάντα από ένα αντιστρέψιμο τμήμα στο κέντρο και διάφορα καλώδια να εξέρχονται ή να εισέρχονται από αυτό. Αρχικοποιήσεις ή μετρήσεις στο κεντρικό τμήμα δεν γίνονται.

5.5 Υλοποίηση κυκλωμάτων $\boldsymbol{FQC}$ σε Haskell

Για τη υλοποίηση των κυκλωμάτων FQC ορίσαμε ένα νέο τύπο κυκλωμάτων στον οποίο αποθηκεύονται οι αριθμοί των εισόδων, εξόδων, καλωδίων σωρού και σκουπιδιών.

```
data Fqc = Fqc {a : : Int,b : : Int,h : : Int,g : : Int,  revCirc : : Circ}
```

Παρακάτω φαίνονται οι δηλώσεις των συναρτήσεων που παράγουν την σειριακή και παράλληλη σύνθεση κυκλωμάτων όπως επίσης και δύο συναρτήσεις που υπολογίζουν το αποτέλεσμα των κυκλωμάτων με είσοδο διάφορα *state*. Η τελευταία συνάρτηση παίρνει ένα κύκλωμα που δεν έχει εισόδους (μόνο καλώδια σωρού) και παράγει το αποτέλεσμα με τα αρχικά *qubit* στην κατάσταση *0*. Αυτή είναι και η περίπτωση όλων των πλήρων προγραμμάτων *nQML*.

```
seqFqc :: Fqc -> Fqc -> Fqc

parFqc :: Fqc -> Fqc -> Fqc

fqc2Mat :: Fqc -> Matrix

eval :: Fqc -> Vector -> Vector

evalNoInp :: Fqc -> Vector
```

5.6 Μετάφραση εντολών–κανόνων

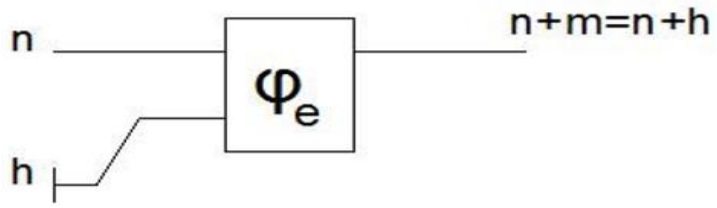
Η σημασιολογία κάθε εντολής κρατάει και διαδίδει πληροφορίες για τον αριθμό των qubit της κατάστασης χρησιμοποιούμε τους σημασιολογικούς κανόνες ώστε ο αριθμός των καλωδίων στα κυκλώματα να συμφωνεί με τους κανόνες. Τα κυκλώματα κατασκευάζονται αναδρομικά, υποθέτοντας δηλαδή ότι έχουν ήδη κατασκευαστεί τα αντίστοιχα κυκλώματα των εκφράσεων που βρίσκονται στην υπόθεση των κανόνων. Αυτά συμβολίζονται ως ορθογώνια των οποίων το περιεχόμενο είναι ένα κύκλωμα της κλάσης $\boldsymbol{FQC}^{\simeq}$ με καλώδια δεξιά και αριστερά τους (σωρός και σκουπίδια) στην περίπτωση που η παραγωγή της έκφρασης δεν είναι αγνή. Αν είναι αγνή (*pure*) όπως είπαμε παραλείπονται τα σκουπίδια. Φυσικά είναι δυνατόν και ο σωρός να είναι κενός αλλά στη γενική περίπτωση εισάγουμε καλώδια στο κύκλωμα.

Με την υλοποίηση σε *Haskell* κάθε κανόνας (εκτός από τον *EMB)* αντιστοιχεί σε μία συνάρτηση η οποία λαμβάνει τα απαραίτητα υπό-κυκλώματα σαν ορίσματα και παράγει το κύκλωμα που προκύπτει με βάση τα παρακάτω σχήματα και τη σημασιολογία της γλώσσας.

5.6.1 Κανόνας EMB

$$\frac{\mathbf{\Gamma;n} \vdash^{\circ} \mathbf{e:\tau;m}}{\mathbf{\Gamma;n} \vdash \mathbf{e:\tau;m}} \quad \textit{(EMB)}$$

Ο κανόνας αυτός είναι προφανώς διαφανής στα κυκλώματα αφού απλώς δηλώνει τη σχέση $\boldsymbol{FQC}^{\circ} \subset \boldsymbol{FQC}$. Κάθε αγνό κύκλωμα μπορεί να θεωρηθεί και μη αγνό χωρίς καμία αλλαγή θεωρώντας ότι το σύνολο σκουπιδιών είναι κενό. Οι αριθμοί στα καλώδια της παρακάτω εικόνας αλλά και σε όλες τις επόμενες δηλώνουν το πλήθος των καλωδίων.

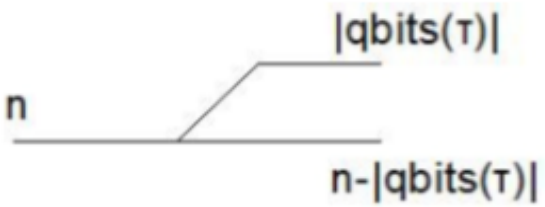


Σχήμα 5.8.1 *Κύκλωμα EMB, το οποίο ανήκει στο* $FQC^{\simeq}$ *άρα και στο* FQC

5.6.2 **Κανόνας VAR**

$$\frac{(\boldsymbol{x}:\boldsymbol{t})\in\boldsymbol{\Gamma}}{\boldsymbol{\Gamma};\boldsymbol{n}\vdash^{\circ}\boldsymbol{x}:\boldsymbol{\tau};\boldsymbol{0}}\quad \textit{(VAR)}$$

Και αυτός ο κανόνας είναι διαφανής αφού όπως αναφέραμε οι μεταβλητές είναι αναφορές σε *qubits.* Για να φανεί καλύτερα αυτό στο παρακάτω σχήμα αναδιατάσσουμε τα *qubit* της μεταβλητής *x* και τα φέρνουμε στο άνω μέρος του κυκλώματος. Φυσικά ο τύπος της *x* μπορεί να μην είναι αγνός και κάποια *qubits* να εμφανίζονται δύο ή περισσότερες φορές στον τύπο όμως αυτό θα επιλυθεί σε επόμενη διεργασία στην περίπτωση που ίσως μετασχηματιστεί η *x*.

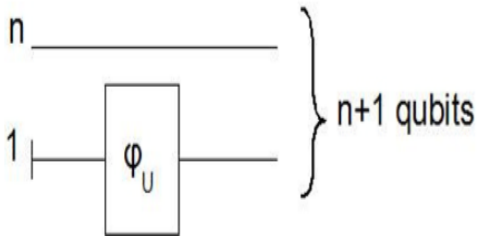


Σχήμα 5.9 *Κύκλωμα VAR. Πρόκειται απλώς για "ομαδοποίηση" των καλωδίων του τύπου τ, μια αναδιάταξη*

5.6.3 Κανόνας SUP

$$\frac{|\boldsymbol{\lambda}|^{\mathbf{2}}+|\boldsymbol{\lambda\prime}|^{\mathbf{2}}=\mathbf{1}}{\boldsymbol{\Gamma};\mathbf{n}\vdash^{\circ}\{(\boldsymbol{\lambda})\boldsymbol{qfalse}+(\boldsymbol{\lambda'qtrue}\}:\boldsymbol{qbit}[\boldsymbol{n}];\mathbf{1}}\ \textit{(SUP)}$$

Αυτός ο κανόνας δημιουργεί το μόνο κύκλωμα της γλώσσας το οποίο δεν απαιτεί να έχει προηγηθεί κατασκευή υπό-κυκλωμάτων, δηλαδή αποτελεί το πρωταρχικό στοιχείο των κυκλωμάτων της γλώσσας από το οποίο κατασκευάζονται όλα τα προγράμματα. Το κύκλωμα δουλεύει αρχικοποιώντας ένα *qubit* στην κατάσταση $|\mathbf{0}\rangle$ και μετά εφαρμόζοντας πάνω σε αυτό ένα μετασχηματισμό περιστροφής που το φέρνει στη ζητούμενη υπέρθεση. Τα συνολικά qubits του κυκλώματος αυξάνονται κατά 1 και δεν υπάρχουν σκουπίδια όπως υποδηλώνει και το $\vdash^{\circ}$ στον κανόνα.



Σχήμα 5.10 *Κύκλωμα SUP*

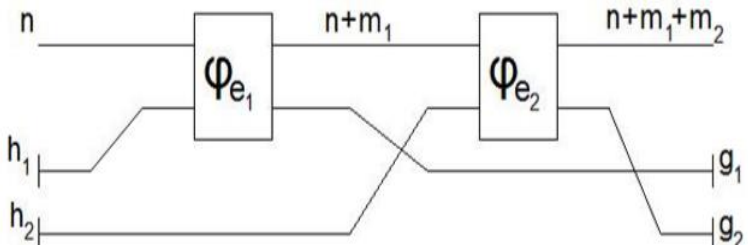
$$U=\begin{pmatrix}\lambda & \lambda' \\ \lambda' & -\lambda\end{pmatrix}$$

5.6.4 Κανόνας LET

$$\frac{\boldsymbol{\Gamma};\boldsymbol{n}\vdash^{\boldsymbol{a}}\boldsymbol{e_1}:\boldsymbol{\tau_1};\quad \boldsymbol{\Gamma},\boldsymbol{x}:\boldsymbol{\tau_1};\boldsymbol{n}+\boldsymbol{m_1}\vdash^{\boldsymbol{a}}\boldsymbol{e_2}:\boldsymbol{\tau};\boldsymbol{m_2}}{\boldsymbol{\Gamma};\boldsymbol{n}\vdash^{\boldsymbol{a}}\boldsymbol{let}\ \boldsymbol{x}=\ \boldsymbol{e_1}\ \boldsymbol{in}\ \boldsymbol{e_2}:\boldsymbol{\tau};\boldsymbol{m_1}+\boldsymbol{m_2}}\ \textit{(LET)}$$

Σε αυτόν τον κανόνα υποθέτουμε ότι έχουν δημιουργηθεί εκ των προτέρων τα κυκλώματα για τον υπολογισμό των εκφράσεων $\mathbf{e_1}$ και $\mathbf{e_2}$.Καθώς ο κανόνας δεν καθορίζει αν οι υπολογισμοί αυτοί είναι αγνοί ή όχι (αν δεν γίνονται μετρήσεις) θεωρώ την γενική περίπτωση και κάθε υπό-κύκλωμα εισάγεται με σωρό και με σκουπίδια. Φυσικά κάποιο από τα δύο σύνολα ή και τα δύο μπορεί να είναι κενά. Το κύκλωμα υπολογίζει πρώτα την έκφραση $\mathbf{e_1}$ και έπειτα την έκφραση $\mathbf{e_2}$. Μέσα στην έκφραση $\mathbf{e_2}$ όπου εμφανίζεται η μεταβλητή $\boldsymbol{\chi}$ θεωρούμε ότι καλείται ο κανόνας *VAR* και τα καλώδια που αποτελούν τον τύπο της $\boldsymbol{\chi}$ (ίδιος με της e_1) αναδιατάσσονται κατάλληλα. Οι αριθμοί των *qubit* στα δεξιά και αριστερά μέρη των κανόνων δηλώνουν τους αριθμούς των καλωδίων τα οποία εισέρχονται και εξέρχονται αντίστοιχα από κάθε επιμέρους κύκλωμα (αυτά που εξέρχονται είναι ίσα με αυτά που εισέρχονται συν τον αριθμό στο αριστερό

μέρος) χωρίς να υπολογίζουμε τα καλώδια αρχικοποιήσεων και σωρού. Το ίδιο δηλώνουν και στους υπόλοιπους κανόνες. Ουσιαστικά το κύκλωμα αποτελεί την σειριακή σύνθεση των δύο κυκλωμάτων $\boldsymbol{\varphi_{e2}} \circ \boldsymbol{\varphi_{e1}}$.

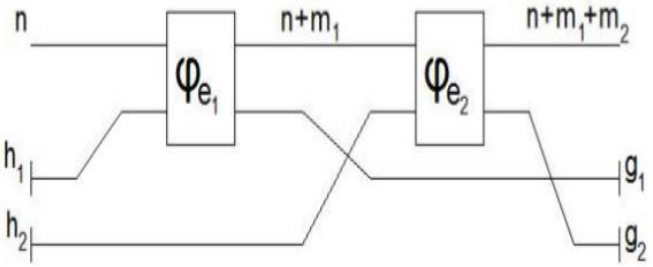


Σχήμα 5.11 *Κύκλωμα LET*

5.6.5 Κανόνας PROD

$$\frac{\boldsymbol{\Gamma;n \vdash^{a} e_1:\tau_1;m_1} \qquad \boldsymbol{\Gamma;n+m_1 \vdash^{a} e_2:\tau_2;m_2}}{\boldsymbol{\Gamma;n \vdash^{a} let(e_1,e_2):\tau_1 \otimes \tau_2 \; ; \; m_1+m_2}} \; \textit{(PROD)}$$

Αυτό ς ο κανόνας όπως και ο επόμενος είναι εντελώς παρόμοιοι με το προηγούμενο κύκλωμα όπως άλλωστε αναφέραμε και όταν δηλώναμε τη σημασιολογία της γλώσσας με πίνακες πυκνότητας. Είναι η σύνθεση $\boldsymbol{\varphi_{e2}} \circ \boldsymbol{\varphi_{e1}}$ **κ**αι οι μόνες διαφορές είναι στις αναδιατάξεις των καλωδίων ανάλογα με τους υπολογισμούς που πρέπει να εκτελεστούν στις επόμενες εκφράσεις. Προφανώς ο *compiler* της γλώσσας θα πρέπει να αποθηκεύει τους τύπους κάθε μεταβλητής και να κατασκευάζει έπειτα τα κατάλληλα κυκλώματα καλωδιώσεων-αναδιατάξεων κατά την κλήση του κανόνα *VAR*.



Σχήμα 5.12 *Κύκλωμα PROD*

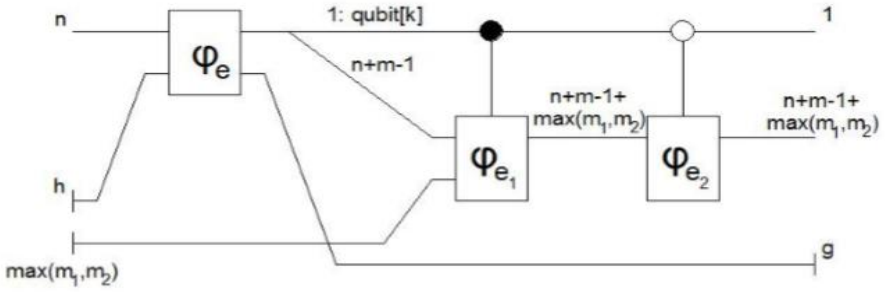
5.6.6 Κανόνας IF

$$\Gamma;n \vdash^{a} e: qbit[k];m$$

$$\frac{\Gamma|_{\kappa};n+m \vdash^{\circ} e_1:\tau;m_1 \qquad \Gamma|_{\kappa};n+m \vdash^{\circ} e_2:\tau;m_2}{\Gamma;n \vdash^{a} if\, m\, e\, then\, e_1\, else\, e_2:\tau;m+max(m_1,m_2)} \quad (IF)$$

Στο κύκλωμα αυτού του κανόνα πρώτα υπολογίζεται η έκφραση $\boldsymbol{e}$ αυτή μπορεί να απαιτεί $\boldsymbol{m}$ *qubit* για τον υπολογισμό της, αλλά ο τελικός της τύπος πρέπει να είναι ένα μόνο *qubit*. Προσέξτε ότι τα υπόλοιπα *qubit* δεν είναι απαραίτητο να αποδεσμευτούν μαζί με τα (πιθανά) σκουπίδια της έκφρασης. Έπειτα υπολογίζονται οι δύο εκφράσεις $\boldsymbol{e_1}$ και $\boldsymbol{e_2}$ οι οποίες όμως έχουν τον περιορισμό να μην χρησιμοποιούν καθόλου το *qubit* του τύπου της έκφρασης και να είναι αγνές, δηλαδή να μην παράγουν σκουπίδια. Ο πρώτος περιορισμός φαίνεται στο κύκλωμα ξεχωρίζοντας το *qubit* και περνώντας το ως qubit ελέγχου στη δομή εκτέλεσης υπό συνθήκη.

Τέλος πρέπει να παρατηρήσουμε ότι τα επιπλέον *qubit* που μπορεί να χρησιμοποιεί η $\boldsymbol{e_1}$, τα $\boldsymbol{m_1}$ καλώδια, είναι δυνατόν να τα χρησιμοποιεί και η $\boldsymbol{e_2}$. Αυτό δεν αποτελεί πρόβλημα για τη λειτουργία του κυκλώματος καθώς υπό συνθήκη εκτέλεση εξασφαλίζει ότι κάθε κοινό qubit θα μεταβληθεί στο ποσοστό που του αναλογεί από την υπέρθεση του *qubit* ελέγχου. Ίσως ο πιθανός προγραμματιστής της γλώσσας θα πρέπει να προσέξει τους τύπους που δίνει στις μεταβλητές ώστε να παράγει το αποτέλεσμα που πράγματι θέλει Για παράδειγμα αν χρησιμοποιεί την $\boldsymbol{e_2}$ ως συνάρτηση ανεξάρτητη $\boldsymbol{e_1}$ πρέπει να προσέξει οι νέοι τύποι να μην μπλέκονται με ήδη δηλωμένους τύπους και άρα να αλλάξει τον κώδικά της (ουσιαστικά ορίζοντας πάντα νέα *qubits* μεταβλητές).



Σχήμα 5.13 *Κύκλωμα IF*

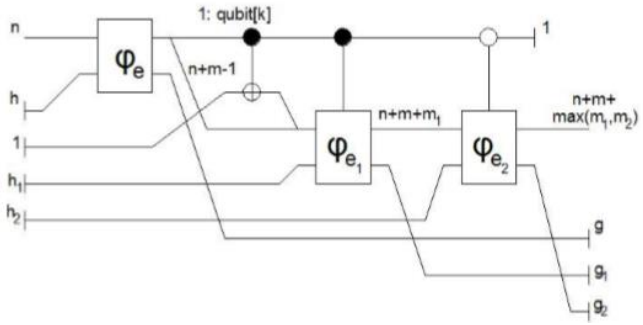
5.6.7 Κανόνας IFM

$$\frac{\boldsymbol{\Gamma;n \vdash e:qbit[k];m} \quad \boldsymbol{\Gamma;n+m \vdash e_1:\tau;m_1} \quad \boldsymbol{\Gamma;n+m \vdash e_2:\tau;m_2}}{\boldsymbol{\Gamma;n \vdash ifm\ e\ then\ e_1\ else\ e_2:\tau;m+max(m_1,m_2)}} \quad \textit{(IFM)}$$

Ο κανόνας αυτός είναι όμοιος με τον κανόνα *IF* με τις διαφορές ότι επιτρέπονται και μη αγνές εκφράσεις στους δύο κλάδους, ότι οι δύο κλάδοι μπορούν να χρησιμοποιούν το *qubit* ελέγχου και ότι το *qubit* ελέγχου μετράται στο τέλος του κυκλώματος. Η πρώτη διαφορά φαίνεται από τα καλώδια σκουπιδιών των δύο εκφράσεων $\boldsymbol{e_1}$ και $\boldsymbol{e_2}$.

Οσον αφορά τη δεύτερη διαφορά, για να μπορούν να χρησιμοποιούν οι δύο κλάδοι το *qubit* ελέγχου πρέπει ιδανικά να το διπλασιάσουμε ώστε να χρησιμοποιείται και στο άνω καλώδιο ελέγχου και μέσα στα υπό-κυκλώματα ταυτόχρονα. Κάτι τέτοιο όμως παραβαίνει το θεώρημα «μη αντιγραφής» και το καλύτερο που μπορούμε να κάνουμε είναι να αρχικοποιήσουμε ένα νέο *qubit* και να το αντιστρέψουμε με μία πύλη *CNOT* που ελέγχεται από το *qubit* ελέγχου. Φυσικά κάτι τέτοιο όπως έχουμε δει δημιουργεί δύο συζευγμένα *qubit* και όχι δύο ανεξάρτητα, όμως αυτό δεν επηρεάζει την ορθή λειτουργία του κυκλώματος. Αυτό συμβαίνει λόγω της τρίτης διαφοράς που είναι η μέτρηση του qubit ελέγχου στο τέλος του κυκλώματος. Αυτή εξαναγκάζει το qubit ελέγχου να μεταβεί σε κλασική κατάσταση και επομένως και το νέο *qubit* να μεταβεί στην ίδια κατάσταση αφού είναι συζευγμένα. Συνεπώς το *qubit* ελέγχου της κλασικής κατάστασης μπόρεσε να αντιγραφεί επιτυχώς και καθώς σε αυτό το if επιθυμούμε κλασικό έλεγχο και όχι κβαντικό όπως στο προηγούμενο το κύκλωμα εκτελεί ακριβώς τη ζητούμενη λειτουργία. Μια λεπτομέρεια αυτής της κατασκευής είναι ότι ο *compiler* της γλώσσας πρέπει να δώσει στο νέο *qubit* τύπο qbit[k] και όποτε χρειάζεται σε περαιτέρω υπολογισμούς να χρησιμοποιεί σταθερά ένα από τα δύο ***qubit*** και τα δύο θεωρούνταικλειδωμένα στην ίδια κλασική κατάσταση.

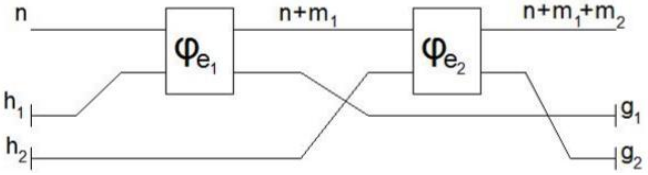
Παρατηρούμε ότι αυτό το κύκλωμα είναι το μόνο που δημιουργεί κλάδο σκουπιδιών και άρα όλοι οι πιθανοί κλάδοι σκουπιδιών των υπό-κυκλωμάτων της γλώσσας προέρχονται από μια τέτοια εντολή. Επίσης μόνο ηεντολή *IFM* και η *SUP* δημιουργούν νέες μεταβλητές και άρα όλα τα καλώδια σωρού προέρχονται από μία από τις δύο αυτές εντολές.



Σχήμα 5.14 *Κύκλωμα IFM*

5.6.8 Κανόνας LETPROD

$$\frac{\boldsymbol{\Gamma};\boldsymbol{n} \vdash^{\boldsymbol{a}} \boldsymbol{e_1}:\boldsymbol{\tau_1} \otimes \boldsymbol{\tau_2};\boldsymbol{m_1} \qquad \boldsymbol{\Gamma},\boldsymbol{x_1}:\boldsymbol{\tau_1},\boldsymbol{x_2}:\boldsymbol{\tau_2}\ ;\ \boldsymbol{n}+\ \boldsymbol{m_1} \vdash^{\boldsymbol{a}} \boldsymbol{e_2}:\boldsymbol{\tau};\boldsymbol{m_2}}{\boldsymbol{\Gamma};\boldsymbol{n} \vdash^{\boldsymbol{a}} \boldsymbol{let}(\boldsymbol{x_1},\boldsymbol{x_2})=\boldsymbol{e_1}\ \boldsymbol{in}\ \boldsymbol{e_2}:\boldsymbol{\tau};\boldsymbol{m_1}+\boldsymbol{m_2}} \quad \textit{(letprod)}$$



Σχήμα 5.15 *Κύκλωμα LETPROD*

ΒΙΒΛΙΟΓΡΑΦΙΑ

•www.researchgate.net

•**Ιωάννης Καραφυλλίδης**,Τίτλος <Βασικές έννοιες>.

•**Ιωάννης Καραφυλλίδης,** Τίτλος <Κβαντική υπολογιστική>

•www.repository.kallipos.gr , Τίτλος <Αρχή της Κβαντικής υπολογιστικής-Κβαντικός αλγόριθμος του Deutsch>.

•PDFS.semanticsholar.org , Τίτλος <Algorithms for Quantum Computation:Discrete log and factoring>.

•Μετσόβιο Πολυτεχνείο διαφάνιες.

•**Behrouz Forouzan και Firouz Mosharraf**, Τίτλος <Εισαγωγή στην επιστήμη των υπολογιστών>.

•**Daniel Hillis**, Τίτλος <Οι υπολογιστές στο παρόν και το μέλλον>.