Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Ομοτοπική Θεωρία Τύπων και η Θεμελιώδης Ομάδα ενός Bouquet από Κύκλους

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΙΩΡΓΟΣ Ν. ΜΑΓΚΑΦΩΣΗΣ

Επιβλέπων : Νικόλαος Σ. Παπασπύρου
Καθηγητής Ε.Μ.Π.

Αθήνα, Δεκέμβριος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Ομοτοπική Θεωρία Τύπων και η Θεμελιώδης Ομάδα ενός Bouquet από Κύκλους

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΙΩΡΓΟΣ Ν. ΜΑΓΚΑΦΩΣΗΣ

Επιβλέπων : Νικόλαος Σ. Παπασπύρου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30η Δεκεμβρίου 2021.

....	....	....
Νικόλαος Σ. Παπασπύρου Καθηγητής Ε.Μ.Π.	Αριστείδης Παγουρτζής Καθηγητής Ε.Μ.Π.	Δημήτριος Φωτάκης Αν. Καθηγητής Ε.Μ.Π

Αθήνα, Δεκέμβριος 2021

....

Γιώργος Ν. Μαγκαφώσης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Σκοπός της παρούσας εργασίας είναι αφενός η συνοπτική περιγραφή των κύριων σημείων της Ομοτοπικής Θεωρίας Τύπων και αφετέρου η χρήση της για την απόδειξη ενός θεωρήματος της κλασικής Ομοτοπικής Θεωρίας στο νέο πλαίσιο που εισάγει.

Λέξεις κλειδιά

Ομοτοπική Θεωρία Τύπων, Intentional Θεωρία Τύπων, Συναρτησιακός Προγραμματισμός, Θεωρία Ομάδων, Θεμελιώδης Ομάδα, Ελεύθερη Ομάδα.

Abstract

The purpose of this diploma dissertation is on one hand a brief presentation of the main features of Homotopy Type Theory and on the other hand to prove a theorem of classical Homotopy Theory inside HoTT.

Key words

Homotopy Type Theory, Intentional Type Theory, Functional Programming, Group Theory, Fundamental Group, Free Group.

Ευχαριστίες

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή αυτής της διατριβής, κ. Νίκο Παπασπύρου, για την εμπνευστική παρουσίαση του πεδίου της Θεωρίας Τύπων και των Γλωσσών Προγραμματισμού καθώς επίσης και για την προτροπή του να ασχοληθώ με την Ομοτοπική Θεωρία Τύπων, η οποία με εξέπληξε με την ιδιαίτερη σχέση της με τα καθαρά μαθηματικά. Θέλω ακόμα να ευχαριστήσω την αδερφή μου Ξανθίππη Μαγκαφώση για την επιμέλεια του κειμένου αλλά και για τις πολύτιμες συμβουλές και διορθώσεις. Τέλος, θα ήθελα να ευχαριστήσω τις Ελευθερία Μακρυνικόλα και Δήμητρα Καστόρα για τη σημαντική συμβολή τους κατά στην αγγλική μετάφραση αυτής της διατριβής.

Γιώργος Ν. Μαγκαφώσης,

Αθήνα, 30η Δεκεμβρίου 2021

Η εργασία αυτή είναι επίσης διαθέσιμη ως Τεχνική Αναφορά CSD-SW-TR-6-21, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών, Εργαστήριο Τεχνολογίας Λογισμικού, Δεκέμβριος 2021.

URL: `http://www.softlab.ntua.gr/techrep/`
FTP: `ftp://ftp.softlab.ntua.gr/pub/techrep/`

Περιεχόμενα

Περίληψη 5

Abstract 7

Ευχαριστίες 9

Περιεχόμενα 11

1. Εισαγωγή 13
1.1 Θεωρία Τύπων 13
1.2 Σκοπός-Δομή 13

2. ITT - Intentional Type Theory 15
2.1 Judgments 15
2.2 Προτάσεις ως Τύποι (Propositions as Types) 16
2.3 Περιβάλλοντα Τύπων 17
2.4 Βασικές Κατασκευές 18
2.4.1 Ο Τύπος Unit 19
2.4.2 Γινόμενα Τύπων 19
2.4.3 Τύποι Συναρτήσεων 20
2.4.4 Ο τύπος Empty 20
2.4.5 Συγινόμενα Τύπων 21
2.4.6 Οι Φυσικοί Αριθμοί 22
2.4.7 Τύποι Εξαρτημένων Συναρτήσεων 23
2.4.8 Τύποι Εξαρτημένων Ζευγών 24
2.4.9 Εξαρτημένα Συγινόμενα Τύπων 24
2.4.10 Σύμπαντα Τύπων 25
2.4.11 Τύποι Ισότητας 25

3. HoTT - Homotopy Type Theory 29
3.1 Higher Groupoids 30
3.2 Univalence Axiom 32
3.3 Επαγωγικοί Τύποι 33
3.4 Υψηλοί Επαγωγικοί Τύποι 34
3.4.1 Το διάστημα 36
3.4.2 Κύκλοι και Σφαίρες 37
3.4.3 Truncations 38

4. Cubical Type Theory - Cubical Agda 41
4.1 Τύποι Μονοπατιών 42
4.2 Composition and Coercion 44
4.2.1 Composition 44
4.2.2 Coercion 45

4.2.3 Partial Elements 45
4.2.4 Συντακτικό και λειτουργία 46
4.2.5 Περίπτωση Μελέτης - Εξαρτημένες Συναρτήσεις 48
4.2.6 Σύμπαντα στην Cubical Type Theory 48

5. Η Θεμελιώδης Ομάδα ενός Bouquet από κύκλους 51
5.1 FreeGroup of A 51
5.2 FreeGroupoid of A 53
5.3 FundamentalGroup 54
5.4 Απόδειξη 55

6. Μελλοντική έρευνα 57

Βιβλιογραφία 59

Παράρτημα 61

A. Ευρετήριο συμβόλων 61

Κεφάλαιο 1

Εισαγωγή

1.1 Θεωρία Τύπων

Η Θεωρία Τύπων είναι θεμελιώδους σημασίας τόσο για τη Μαθηματική Λογική όσο και για την Επιστήμη των Υπολογιστών. Δημιουργός της ήταν ο Bertrand Russell που σκοπό είχε την κατασκευή ενός συστήματος για τη Θεμελίωση των Μαθηματικών απαλλαγμένο από τις αντιφάσεις της αφελούς συνολοθεωρίας. Τις επόμενες δεκαετίες απέκτησε μεγάλη εξέλιξη κυρίως με τη συσχέτισή της με τον λ-λογισμό από τον Alonzo Church. Η Θεωρία Τύπων βρήκε μεγάλο πεδίο εφαρμογής στη Θεωρία Γλωσσών Προγραμματισμού με την πρόοδο των ηλεκτρονικών υπλογιστών. Ένα από τα σημαντικότερα ορόσημα ήταν η τροποποίηση της Θεωρίας Τύπων του Church σε μια πιο *κατηγορική* Θεωρία από τον Per Martin-Löf, τη λεγόμενη εξαρτημένη θεωρία τύπων ή Intentional Type Theory.

Τα τελευταία χρόνια η Θεωρία Τύπων γνωρίζει νέα άνθιση με την ανακάλυψη της Ομοτοπικής Θεωρίας Τύπων. Η ιδέα της Ομοτοπικής Θεωρίας Τύπων πρωτοεμφανίστηκε περίπου το 2006 με την ανεξάρτητη δουλειά των Awodey-Warren και Voevodsky, οι οποίοι εμπνεύστηκαν από την groupoid ερμηνεία της Θεωρίας Τύπων των Hofmann και Streicher, που είχε προηγηθεί. Η Θεωρία Κατηγοριών υψηλότερων διαστάσεων, η οποία είναι στενά συνδεδεμένη με την Ομοτοπική Θεωρία, πλέον μελετάται έντονα από μαθηματικούς και των δύο πεδίων. Τα σημασιολογικά μοντέλα των Awodey-Warren και Voevodsky χρησιμοποιούν γνωστές έννοιες και τεχνικές από την Ομοτοπική Θεωρία. Ειδικότερα, ο Voevodsky κατασκεύασε μία ερμηνεία της θεωρίας Τύπων η οποία ικανοποιεί μια επιλέον σημαντική ιδιότητα την οποία ονόμασε univalence. Αυτό δεν είχε προταθεί ξανά στη Θεωρία Τύπων και η προσθήκη του με τη μορφή αξιώματος είχε βαθιές συνέπειες στη θεωρία.

1.2 Σκοπός-Δομή

Ο σκοπός της εργασίας αυτής είναι διπλός. Από τη μια, η περιγραφή στο πλαίσιο του δυνατού της Ομοτοπικής Θεωρία Τύπων, και από την άλλη η απόδειξη ενός γνωστού θεωρήματος από την αλγεβρική τοπολογία εντός της. Έτσι, η εργασία ακολουθεί την παρακάτω δομή.

Κεφάλαιο 2ο Βάση για την Ομοτοπική Θεωρία Τύπων αποτελεί η Intentional Type Theory. Στο κεφάλαιο αυτό δίνουμε αρχικά τις βασικές έννοιες κάθε θεωρίας τύπων όπως οι έννοιες του τύπου, των αποφάνσεων, των περιβάλλοντων τύπων κ.α. Επίσης, γίνεται μια μακρά και λεπτομερής περιγραφή πολλών από τις κατασκευές που είναι δυνατές μέσα σε αυτήν τη θεωρία, με σημαντικότερη από αυτές να είναι η κατασκευή των Τύπων Ισότητας, για τους οποίους δίνονται τα πιο σημαντικά αποτελέσματα και εργαλεία.

Κεφάλαιο 3ο Αυτό το κεφάλαιο περιέχει το κεντρικό θέμα του πρώτου μέρους της εργασίας: την περιγραφή της Ομοτοπικής Θεωρίας Τύπων. Γίνεται μια προσπάθεια εξήγησης της ομοτοπικής ερμηνείας των τύπων της ITT και η περιγραφή των ουσιαστικών αλγεβρικών χαρακτηριστικών που κάνουν αυτήν την ερμηνεία δυνατή. Έπειτα, ακολουθεί η ανάλυση των δύο σημαντικών νέων στοιχείων που φέρνει η Ομοτοπική Θεωρία Τύπων σε σχέση με την ΤΤΤ. Αυτά είναι το Univalence Axiom και οι Υψηλοί Επαγωγικοί Τύποι.

Κεφάλαιο 4ο Η εκδοχή της Ομοτοπικής Θεωρίας Τύπων πάνω στην οποία στηρίχθηκε η απόδειξη του θεωρήματος είναι η Cubical Type Theory. Η Cubical Type Theory αναπτύχθηκε με σκοπό να αποδώσει υπολογιστικό χαρακτήρα στη HoTT και κυρίως να δώσει το κατάλληλο πλαίσιο στο οποίο θα ήταν δυνατόν το Univalence Axiom να αποδειχθεί. Κάνουμε μια σύντομη εισαγωγή στα κύρια σημεία της θεωρίας και παρουσιάζουμε τα κύρια αποτελέσματα.

Κεφάλαιο 5ο Το κεφάλαιο αυτο καλύπτει τον δεύτερο σκοπό της εργασίας, δηλαδή την περιγραφή της απόδειξης του θεωρήματος. Το θεώρημα που αποδείξαμε με χρήση της Cubical Type Theory δίνει μια ισότητα τύπων. Συγκεκριμένα αποδεικνύει ότι η θεμελιώδης ομάδα ενός bouquet από κύκλους με δίκτες από έναν τύπο A είναι ίση με την ελεύθερα παραγόμενη ομάδα από τα στοιχεία του A. Η βιβλιοθήκη που αναπτύχθηκε με τον proof assistant Cubical Agda είναι ελεύθερα προσβάσιμη στον εξής ιστοχώρο: https://github.com/gmagaf/Bouquet-HoTT-Cubical-Agda.

Κεφάλαιο 6ο Το τελευταίο κεφάλαιο είναι αφιερωμένο σε πιθανές προεκτάσεις του θεωρήματος, σε πεδία έρευνας γύρω από την Ομοτοπική Θεωρία Τύπων καθώς επίσης και ορισμένα ανοιχτά προβλήματα.

Κεφάλαιο 2

ITT - Intentional Type Theory

Η Homotopy Type Theory βασίζεται στην Intentional Type Theory, η οποία παρουσιάστηκε για πρώτη φορά το 1972 από τον Σουηδό μαθηματικό και φιλόσοφο Per Martin-Löf. Αποτελεί μια ανάλυση και ανάπτυξη του προγράμματος του Brower για τα κατασκευαστικά - ιντουϊσιονιστικά μαθηματικά.

Ευρύτερο πλαίσιο για τη θεωρία που θα αναπτύξουμε αποτελεί ο ιντουϊσιονισμός ως σχολή της φιλοσοφίας των μαθηματικών. Εμφανίστηκε στις αρχές του 20ου αιώνα και ιδρυτής του ήταν ο Ολλανδός μαθηματικός J.E.L.Brouwer, που προσπάθησε να δώσει τη δική του λύση στην κρίση της θεμελίωσης των μαθηματικών. Σε αντίθεση με άλλες σχολές με ίδιο στόχο, ο ιντουϊσιονισμός αντί να προσπαθεί να στηρίξει τη μέχρι τότε μαθηματική πρακτική, την αμφισβητεί και απορρίπτει μεγάλα κομμάτια των κλασσικών μαθηματικών. Δίνει έμφαση στην εξέταση της φύσης των νοητών μαθηματικών κατασκευών και υποβαθμίζει ερωτήματα σχετικά με την οντολογία των αντικειμένων που παράγονται από αυτές σε δευτερεύουσας σημασίας φιλοσοφικά προβλήματα. Για τον Brower η μαθηματική σκέψη είναι μια ανθρώπινη δραστηριότητα και τα μαθηματικά είναι μια γλώσσα στην οποία εκφράζονται οι διάφορες έννοιες. Συνοπτικά, αίτημα του ιντουϊσιονισμού είναι πως κριτήριο ύπαρξης των μαθηματικών αντικειμένων θα πρέπει να είναι η νοητή παραγωγή τους με διαδικασίες κατασκευαστικά ελεγχόμενες, δηλαδή αλγορίθμους.

Επομένως, η Intentional Type Theory αντιμετωπίζει τα μαθηματικά αντικείμενα ως κατασκευές. Κάθε μαθηματικό αντικείμενο είναι ενός τύπου. Καλύτερα διατυπωμένα, ένα μαθηματικό αντικείμενο δίνεται πάντα μαζί με τον τύπο του, δηλαδή δεν είναι απλώς ένα αντικείμενο αλλά ένα αντικείμενο ενός συγκεκριμένου τύπου. Οι τύποι λοιπόν, ταξινομούν και κατηγοριοποιούν τις κατασκευές και μπορούν να ιδωθούν ως προδιαγραφές (specifications) αυτών, όπως ακριβώς οι τύποι στις γλώσσες προγραμματισμού.

Ένας τύπος ορίζεται περιγράφοντας τι πρέπει να κάνουμε για να κατασκευάσουμε ένα αντικείμενο αυτού του τύπου. Για να το θέσουμε διαφορετικά, ένας τύπος είναι καλώς ορισμένος αν συλλαμβάνουμε πλήρως το τι σημαίνει να είναι κάτι αντικείμενο αυτού του τύπου. Έτσι, επί παραδείγματι, ο $\mathtt{nat} \to \mathtt{nat}$ είναι τύπος, όχι επειδή γνωρίζουμε ορισμένες αριθμητικές συναρτήσεις όπως τις πρωτογενώς αναδρομικές, αλλά επειδή θεωρούμε ότι κατανοούμε την έννοια της αριθμητικής συνάρτησης *εν γένει*. Να σημειώσουμε ότι αυτό είναι προϋπόθεση ανεξάρτητα από το αν μπορούμε να παραγάγουμε κάπως όλα τα αντικείμενα ενός τύπου ή αν γνωρίζουμε κάπως όλα τα αντικείμενα ξεχωριστά.

Σε αυτό το κεφάλαιο θα περιγράψουμε με ποιο τρόπο μπορούμε να μιλάμε για αυτές τις κατασκευές στο πλαίσιο της ITT καθώς και για το πώς αυτές αποκτούν μια πιο *λογική* ερμηνεία μέσα από την ιδέα του ”Propositions as Types”. Κύριο μέρος του κεφαλαίου αποτελεί η απαρίθμηση όλων των βασικών κατασκευών τύπων που είναι δυνατές μέσα στην ITT καθώς και η ανάλυση της συμπεριφοράς των αντικειμένων αυτών των τύπων.

2.1 Judgments

Η Θεωρία Τύπων έχει μόνο μία πρωτογενή έννοια, αυτή του τύπου. Εντός της Θεωρίας τύπων μπορούμε να παράγουμε *αποφάνσεις*, κάτι το οποίο γνωρίζουμε, (judgments) με τη βοήθεια κανόνων συμπερασμού. Η βασική απόφανση στη Θεωρία Τύπων γράφεται στη μορφή $a : A$ και προφέρεται ως “το a έχει τύπο A” ή πιο χαλαρά “το a είναι στοιχείο του A”. Διαισθητικά αυτό σημαίνει ότι το αντικείμενο a είναι κατασκευή η οποία περιγράφεται- κωδικοποιείται από τον τύπο A. Επομένως, δεν θα πρέπει να το μπερδεύουμε με την ιδιότητα του ανήκειν της συνολοθεωρίας και να το ερμηνεύουμε ως $a \in A$ το οποίο είναι μια πρόταση στην κατηγορηματική λογική. Για παράδειγμα, δεν μπορούμε να κάνουμε εντός της Θεωρίας Τύπων συλλογισμούς όπως “αν $a : A$ τότε δεν ισχύει ότι $b : B$” ή επίσης δεν μπορούμε να “διαψεύσουμε” το $a : A$. Αντίθετα, στη Θεωρία Τύπων κάθε αντικείμενο έχει από τη φύση του κάποιο τύπο. Όταν λέμε για παράδειγμα στη συνολοθεωρία “έστω x φυσικός αριθμός” τότε εννοούμε “έστω ένα αντικείμενο x και υποθέτουμε ότι $x \in \texttt{nat}$” ενώ στη Θεωρία Τύπων η απόφαση $x : \texttt{nat}$ είναι μια ατομική δήλωση και δεν έχει νόημα να εισάγουμε μια μεταβλητή χωρίς να αναφέρουμε τον τύπο της.

Το δεύτερο είδος αποφάνσεων της Θεωρίας Τύπων είναι σχετικό με την έννοια της ισότητας και ονομάζεται *αποφασίσιμη ισότητα* ή ισότητα εξ ορισμού (judgmental equality ή definitional equality). Γράφουμε $a \equiv b : A$ ή απλά $a \equiv b$. Η διαίσθηση πίσω από αυτήν την απόφανση είναι η ισότητα εξ ορισμού. Για παράδειγμα αν $f : \texttt{nat} \to \texttt{nat}$ είναι μια συνάρτηση με τύπο $f(x) = x^2$ τότε έχουμε ότι $f(3) \equiv 3^2$. Και πάλι όπως πριν, δεν έχει νόημα εντός της Θεωρίας Τύπων να υποθέσουμε ή να διαψεύσουμε μια ισότητα εξ ορισμού. Το αν, λοιπόν, δύο αντικείμενα είναι ίσα εξ ορισμού, δηλαδή το ίδιο, είναι ζήτημα ανάπτυξης των ορισμών τους και είναι αλγοριθμικά αποφασίσιμη διαδικασία.

2.2 Προτάσεις ως Τύποι (Propositions as Types)

Μια *πρόταση* με την έννοια της ιντουϊσιονιστικής λογικής αντί της κλασσικής ορίζεται και τίθεται με τον τρόπο με τον οποίο δικαιούμαστε να την αποδείξουμε. Για παράδειγμα η πρόταση:

“Ο αριθμός 42 δεν είναι πρώτος”

είναι μια πρόταση την οποία αποδεικνύουμε με την παράθεση δύο φυσικών αριθμών μεγαλύτερων της μονάδας και έναν υπολογισμό που μας δείχνει ότι το γινόμενό τους ισούται με 42. Δηλαδή, καθώς όλα τα μαθηματικά αντικείμενα είναι κατασκευές, το ίδιο είναι και οι αποδείξεις οι οποίες ταξινομούνται με τη χρήση τύπων. Επομένως, η απόδειξη μιας πρότασης δεν είναι κάτι άλλο από την κατασκευή ενός προγράμματος αυτού του τύπου. Αυτή η κατασκευαστική ερμηνεία των προτάσεων δημιουργεί αντιστοιχίσεις με τη Θεωρία Τύπων.

Για να δώσουμε ένα παράδειγμα της αντιστοίχισης, παίρνουμε την περίπτωση της συνεπαγώγης. Η ιντουϊσιονιστική ερμηνεία του συνδέσμου της συνεπαγωγής ορίζει ότι μια απόδειξη μιας πρότασης $A \implies B$ είναι μια συνάρτηση που αντιστοιχίζει μια οποιαδήποτε απόδειξη του A σε μια απόδειξη του B. Έτσι, η ταυτοτική συνάρτηση είναι μια απόδειξη της πρότασης $A \implies A$. Παρόμοιες αντιστοιχίσεις μεταξύ ιντουϊσιονιστικών συνδέσμων και κατασκευών τύπων που θα δούμε είναι οι εξής:

Propositions	Types
$\top$	$\mathbf{1}$
$A \wedge B$	$A \times B$
$A \implies B$	$A \to B$
$\perp$	$\mathbf{0}$
$A \vee B$	$A + B$
$\forall x : A.B$	$\Pi x : A.B$
$\exists x : A.B$	$\Sigma x : A.B$

Στην περίπτωση της ITT επομένως, δεν υπάρχει η ανάγκη εισαγωγής επιπρόσθετης έννοιας πρότασης καθώς μπορούμε να αναπαραστήσουμε κάθε πρόταση με έναν τύπο, τον τύπο των κατασκευαστικών αποδείξεων αυτής της πρότασης. Πολλές φορές, λοιπόν, δεν θα κάνουμε κάποια διάκριση μεταξύ των εννοιών τύπος - πρόταση και μεταξύ των φράσεων ”το a είναι ένα στοιχείο του τύπου A” και ”το a είναι μια απόδειξη της πρότασης A”.

Σε αυτό το σημείο θα ήταν καλό να διακρίνουμε τη διαφορά που υπάρχει ανάμεσα σε αυτήν τη λογική και στις αποδείξεις της και στην κλασική λογική και τις τυπικές αποδείξεις. Στην κλασική περίπτωση οι αποδείξεις μας και τα συμπεράσματα μας προκύπτουν από την επενέργεια ενός αποδεικτικού συστήματος τύπου Hilbert, με λογικά, μη λογικά αξιώματα και κανόνες συμπερασμού πάνω στα "data", τα αντικείμενα του κόσμου μας, δηλαδή τα σύνολα. Σε αυτήν την περίπτωση, όμως, το αποδεικτικό μας σύστημα είναι μέσα στη φύση των αντικειμένων μέσω των κανόνων που τα ορίζουν και περιγράφουν τη συμπεριφορά τους. Αυτό είναι μια παραδοσιακή αρχή στον τρόπο λειτουργίας της θεωρίας τύπων, η έλλειψη δηλαδή εξωτερικής λογικής και αξιωμάτων πολύ απλά επειδή κάτι τέτοιο αντιτίθεται στην κατασκευαστική φύση του συστήματος. Στην περίπτωση της ITT οι αποδείξεις είναι οι ίδιες μαθηματικά αντικείμενα πρώτης κατηγορίας και εννοιολογικά όμοιες με τα data στα οποία αναφέρονται, π.χ. μια απόδειξη μιας ιδιότητας για τους φυσικούς αριθμούς(τα αντικείμενα του τύπου `nat`), έστω το θεώρημα της διαίρεσης, είναι ένα αντικείμενο του τύπου

$$\Pi m, n : \texttt{nat}.GT(M, 0) \to \Sigma q, r : \texttt{nat}.\texttt{Id}(mq + r, n) \times GT(m, r)$$

Πολύ απλά, είναι όλα αντικείμενα κάποιου τύπου και η ύπαρξη τους και μόνο είναι η απόδειξη της αλήθειας της πρότασης που αντιστοιχεί στον τύπο αυτόν. Ακόμα, η λειτουργία των συνδέσμων στην ιντουϊσιονιστική λογική καθορίζεται από κανόνες συμπερασμού και από τις υπολογιστικές τους ιδιότητες, ενώ στην περίπτωση της κλασικής λογικής μέσω των πινάκων αληθείας. Επομένως, η ITT αποκτά μια ελευθερία και από άλλες καθολικές υποθέσεις, εκτός από τα αξιώματα. Έτσι, οι διάφορες υποθέσεις που μπορεί να απαιτούνται για την απόδειξη μιας πρότασης μπορούν να υιοθετηθούν ανά περίπτωση και άρα η συνολική θεωρία να είναι πιο ισχυρή. Μια τέτοια περίπτωση είναι ο νόμος αποκλεισμού παντός τρίτου. Στην ITT δεν υπάρχει καμία διατύπωση της καθολικής ισχύος του νόμου αυτού και υποτίθεται όπου χρειάζεται σε αντίθεση με την κλασική λογική στην οποία ο νόμος αυτός είναι εμφανής στη βάση της θεμελίωσής της με τους κανόνες των αληθοπινάκων της διάζευξης και της άρνησης.

2.3 Περιβάλλοντα Τύπων

Οι αποφάνσεις στις οποίες καταλήγουμε πολλές φορές μπορεί να εξαρτώνται από κάποιες υποθέσεις. Για παράδειγμα, υποθέτοντας ότι n, m : `nat` να συμπεραίνουμε ότι $n + m$: `nat`. Τυπικά, το περιβάλλον είναι μια διατεταγμένη λίστα από αποφάνσεις, μερικές εκ των οποίων μπορεί να βασίζονται σε προηγούμενες αποφάνσεις στη λίστα. Συνήθως, συμβολίζουμε το περιβάλλον με Γ και διαχωρίζουμε τις υποθέσεις από το συμπέρασμα με το σύμβολο $\vdash$. Για παράδειγμα, $\Gamma, n : \texttt{nat}, m : \texttt{nat} \vdash n + m : \texttt{nat}$ ή $\Gamma, x : A, y : A, p : \texttt{Id}_A(x, y) \vdash p^{-1} : \texttt{Id}_A(y, x)$. Στη δεύτερη περίπτωση βλέπουμε την ύπαρξη εξαρτήσεων μεταξύ των υποθέσεων καθώς επίσης και το ότι τα αντικείμενα x, y, p λειτουργούν ακριβώς όπως οι συνηθισμένες μαθηματικές μεταβλητές. Η διαισθητική σημασία του περιβάλλοντος και του $\vdash$ είναι αυτή της συνεπαγωγής (entailment) και άρα, για να θέλουμε η θεωρία μας να έχει κάποιο νόημα, θα θέλαμε να ικανοποιεί κάποιες ιδιότητες που κάνουν μια μορφή συνεπαγωγής να είναι συνεπαγωγή. Αυτές είναι οι εξής:

1. Αυτοπάθεια (Reflexivity)

$$\Gamma, x : A, \Gamma' \vdash x : A$$

2. Μεταβατικότητα (Transitivity)

$$\frac{\Gamma, x : A, \Gamma' \vdash N : B \quad \Gamma \vdash M : A}{\Gamma\Gamma' \vdash [M/x]N : B}$$

3. Αποδυνάμωση (Weakening)

$$\frac{\Gamma \vdash M : A}{\Gamma\Gamma' \vdash M : A}$$

4. Συστολή (Contraction)
$$\frac{\Gamma, x : A, y : A, \Gamma' \vdash N : B}{\Gamma, z : A, \Gamma' \vdash [x, y/z, z]N : B}$$

5. Ανταλλαγή (Exchange)
$$\frac{\Gamma, x : A, y : A, \Gamma' \vdash N : B}{\Gamma, y : A, x : A, \Gamma' \vdash N : B}$$

Όταν έχουμε ένα σύστημα, όπως η θεωρία τύπων ή η ιντουϊσιονιστική λογική, στην οποία όταν προστεθούν αυτοί οι κανόνες τότε είναι συμβατοί με τις υπόλοιπες κατασκευές και κανόνες του συστήματος, τότε λέμε ότι οι κανόνες αυτοί είναι *αποδεκτοί* (admissible). Αποδεικνύεται ότι οι παραπάνω κανόνες είναι αποδεκτοί από τη θεωρία τύπων που αναπτύσσουμε, δηλαδή τους κανόνες που θα εισάγαγουμε. Αξίζει να παρατηρήσουμε ότι έχουν μελετηθεί συστήματα στα οποία έχουμε σχέσεις λογικού συμπερασμού (entailment) όπου δεν δεχόμαστε ορισμένους από τους κανόνες της αποδυνάμωσης, της συστολής ή της ανταλλαγής, όμως πάντα, για να μιλάμε για μια σχέση λογικής συνεπαγωγής, θα πρέπει να ισχύουν οι κανόνες της αυτοπάθειας και της μεταβατικότητας, διότι αλλιώς αυτό που έχουμε δεν έχει τη διαισθητική σημασία που θέλουμε.

2.4 Βασικές Κατασκευές

Όπως αναφέραμε πιο πάνω, οι τύποι κωδικοποιούν τις δυνατές κατασκευές που μπορεί να κάνει κανείς στα πλαίσια του συστήματός μας. Επομένως, για να μπορούμε να κάνουμε χρήσιμες κατασκευές και να περιγράψουμε ένα μεγάλο πλήθος αντικειμένων θα πρέπει να δώσουμε τρόπους δημιουργίας νέων τύπων. Ο τρόπος με τον οποίο γίνεται αυτό στη θεωρία τύπων είναι μέσω κανόνων οι οποίοι περιγράφουν τη συμπεριφορά των αντικειμένων αυτού του τύπου.

Για κάθε νέο τύπο, λοιπόν, θα δίνονται και μια σειρά από κανόνες που καθορίζουν τη σύνταξή του, τον τρόπο δημιουργίας αντικειμένων, τη λειτουργία τους και τον υπολογιστικό χαρακτήρα που θέλουμε να δώσουμε. Άρα, για κάθε τύπο έχουμε κανόνες:

- Σχηματισμού (Formation), που περιγράφουν το τι είναι απαραίτητο για τον σχηματισμό του νέου τύπου.

- Εισαγωγής (Introduction), που περιγράφουν τον τρόπο με τον οποίο γίνεται να κατασκευάσουμε αντικείμενα αυτού του τύπου, μέσω κατασκευαστών.

- Απαλοιφής (Elimination), που μας δείχνουν πώς μπορούμε να χρησιμοποιήσουμε τα αντικείμενα του τύπου.

- Κανόνες υπολογισμού ή β-αναγωγής (β-reduction), που δείχνουν τον τρόπο με τον οποίο αλληλεπιδρούν οι κατασκευαστές του αντικειμένου με τους κανόνες απαλοιφής. Η ιδέα πίσω από αυτούς τους κανόνες έχει τις ρίζες της στην Inversion Principle του Gentzen σύμφωνα με την οποία η απαλοιφή είναι αντίστροφη της εισαγωγής. Αυτό, όπως θα δούμε, είναι η ουσία του υπολογισμού.

- Κανόνες μοναδικότητας ή η-επέκταση (η-expansion), που εκφράζουν τη μοναδικότητα των απεικονίσεων προς και από αντικείμενα αυτού του τύπου διατυπώνοντας το ότι κάθε αντικείμενο ενός τύπου χαρακτηρίζεται μοναδικά από την εφαρμογή κανόνων απαλοιφής σε αυτό και μπορεί να ανασυντεθεί από την εφαρμογή του κατασκευαστή σε αυτά τα αποτελέσματα. Είναι με κάποιο τρόπο δυϊκοί των κανόνων β-αναγωγής αφού μας δείχνουν τον τρόπο που δρα ο κατασκευαστής αυτήν τη φορά στο αποτέλεσμα μιας απαλοιφής. Κι αυτοί οι κανόνες οφείλονται στην Unicity Principle του Gentzen.

2.4.1 Ο Τύπος Unit

Ο τύπος Unit είναι ένας απλοϊκός τύπος ο οποίος έχει μόνο ένα στοιχείο, το $\langle\rangle$. Συχνά συμβολίζεται και ως $\mathbf{1}$. Άρα, αφού ο τύπος $\mathbf{1}$ δεν δημιουργείται από άλλους τύπους έχουμε απλά την απόφανση

$$\frac{}{\Gamma \vdash \mathbf{1} : \mathtt{type}}\ F-\mathbf{1}$$

ως κανόνα σχηματισμού του τύπου. Για τη δημιουργία ενός αντικειμένου του $\mathbf{1}$, του μοναδικού που υπάρχει, δεν απαιτείται καμία υπόθεση. Δηλαδή:

$$\frac{}{\Gamma \vdash \langle\rangle : \mathbf{1}}\ I-\mathbf{1}$$

Ο τύπος αυτός δεν έχει κάποιον κανόνα απαλοιφής και ως εκ τούτου ούτε κανόνα β-αναγωγής. Υπάρχει, όμως, κανόνας η-αναγωγής, ως *αντίθετος* του κανόνα εισαγωγής στοιχείων, ο οποίος μάλιστα εκφράζει και τη μοναδικότητα του στοιχείου $\langle\rangle$.

$$\frac{\Gamma \vdash M : \mathbf{1}}{\Gamma \vdash M \equiv \langle\rangle : \mathbf{1}}\ \eta-\mathbf{1}$$

Όπως είναι προφανές, αυτός ο τύπος εκφράζει την ύπαρξη ενός Τύπου με ένα ακριβώς στοιχείο, εξ ορισμού. Αντιστοιχεί δηλαδή, στην πρόταση που είναι εξ ορισμού αληθής και η αλήθεια της οποίας δεν απαιτεί καμία υπόθεση. Δηλαδή στην πρόταση $\top$.

2.4.2 Γινόμενα Τύπων

Δεδομένων δύο τύπων A και B είναι δυνατή η κατασκευή του γινομένου τους $A \times B$. Η διαισθητική ερμηνεία αυτής της κατασκευής είναι ο συνδυασμός των αντικειμένων των δύο τύπων διατηρώντας την πληροφορία που περιέχουν και τα δύο αντικείμενα, όπως στα γινόμενα συνόλων, στα γινόμενα ομάδων, τοπολογικών χώρων κλπ. Οι κανόνες περιγραφής της λειτουργίας αυτών των τύπων είναι οι εξής. Για τον σχηματισμό απαιτείται οι A και B να είναι τύποι. Δηλαδή:

$$\frac{\Gamma \vdash A : \mathtt{type} \quad \Gamma \vdash B : \mathtt{type}}{\Gamma \vdash A \times B : \mathtt{type}}\ F-\times$$

Για τη δημιουργία και τη χρήση νέων αντικειμένων αυτού του τύπου έχουμε:

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N\rangle : A \times B}\ I-\times$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \mathtt{fst}(M) : A}\ E_1-\times \qquad \frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \mathtt{snd}(M) : B}\ E_2-\times$$

Όσο για τον υπολογιστικό χαρακτήρα των κατασκευών αυτού του τύπου έχουμε:

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \mathtt{fst}\langle M, N\rangle \equiv M : A}\ \beta_1-\times \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \mathtt{snd}\langle M, N\rangle \equiv N : B}\ \beta_2-\times$$

και ακόμα για την αρχή της μοναδικότητας αυτών των κατασκευών (Gentzen's Unicity Principle):

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \langle \mathtt{fst}(M), \mathtt{snd}(M)\rangle \equiv M : A \times B}\ \eta-\times$$

2.4.3 Τύποι Συναρτήσεων

Δεδομένων δύο τύπων και B κατασκευάζουμε τον τύπο $A \to B$ των συναρτήσεων με πεδίο ορισμού το A και πεδίο τιμών το B. Αντίθετα από ό,τι συμβαίνει στη συνολοθεωρία, οι συναρτήσεις εδώ δεν είναι συναρτησιακές σχέσεις αλλά μια πρωτογενής έννοια, η οποία ορίζεται από τον τρόπο με τον οποίο συμπεριφέρονται τα αντικείμενα-συναρτήσεις. Έστω, λοιπόν, μια συνάρτηση $f : A \to B$. Τότε η f μπορεί να εφαρμοστεί σε κάθε στοιχείο $a : A$ επιστρέφοντας ένα στοιχείο $f(a)$ στο B, το οποίο ονομάζουμε τιμή της f στο a. Στη θεωρία τύπων είναι σύνηθες αντί για $f(a)$ να γράφουμε απλώς fa.

Για να κατασευάσουμε μια συνάρτηση μπορούμε να την εισάγουμε δίνοντάς της ένα όνομα, έστω f, και να ορίσουμε την $f : A \to B$ δίνοντας μια έξίσωση της μορφής $f(x) = \Phi$, όπου το Φ μπορεί να περιέχει τη μεταβλητή x και να ανήκει στον B, αν υποθέσουμε ότι $x : A$. Διαφορετικά είναι δυνατόν να δημιουργήσουμε ένα αντικείμενο του τύπου $A \to B$ χωρίς να του δώσουμε όνομα χρησιμοποιώντας λ-αφαίρεση (λ-abstraction). Αν Φ όπως πιο πάνω, μπορούμε να γράψουμε $\lambda(x : A)\Phi : A \to B$ και να ορίσουμε την ίδια συνάρτηση όπως πριν.

Όλα αυτά περιγράφονται από τους παρακάτω κανόνες. Για να μπορούμε να ορίσουμε σωστά έναν τύπο συνάρτησης θα πρέπει το πεδίο ορισμού και τιμών να είναι καλά ορισμένοι τύποι.

$$\frac{\Gamma \vdash A : \texttt{type} \quad \Gamma \vdash B : \texttt{type}}{\Gamma \vdash A \to B : \texttt{type}} \; F\!-\!\to$$

ενώ για τη λ-αφαίρεση και την εφαρμογή της συνάρτησης σε τιμή του πεδίου ορισμού έχουμε:

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda(x : A)M : A \to B} \; I\!-\!\to$$

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M(N) : B} \; E\!-\!\to$$

Για να εκφράσουμε την υπολογιστική συμπεριφορά των αντικειμένων-συναρτήσεων εισάγουμε τον κανόνα β-αναγωγής ο οποίος, διαισθητικά, ορίζει ότι υπολογισμός της συνάρτησης f στο $N : A$ δεν είναι παρά η αντικατάσταση όλων των ελεύθερων εμφανίσεων της μεταβλητής x που εμφανίζονται στην f από την τιμή N.

$$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash (\lambda(x : A)M)(N) \equiv [N/x]M : B} \; \beta\!-\!\to$$

Τέλος, η αρχή μοναδικότητας του Gentzen εκφράζει το ότι η λ-αφαίρεση είναι ουσιαστικά ο μόνος τρόπος δημιουργίας συναρτήσεων.

$$\frac{\Gamma \vdash M : A \to B}{\Gamma \vdash (\lambda(x : A)M(x)) \equiv M : A \to B} \; \eta\!-\!\to$$

Για τη δημιουργία συναρτήσεων πολλών μεταβλητών ένας τρόπος είναι η δημιουργία συναρτήσεων με πεδίο ορισμού γινόμενα τύπων ($f : A \times B \to C$) ή, συνηθέστερα στην θεωρία τύπων, η χρήση μιας τεχνικής που ονομάζεται Currying (από τον μαθηματικό Haskell Curry). Σύμφωνα με αυτή, μια συνάρτηση από το ζεύγος $A \times B$ στο C είναι μια συνάρτηση με πεδίο ορισμού το A και πεδίο τιμών συναρτήσεις από τον B στον C, δηλαδή ο τύπος $A \to (B \to C)$. Χρειάζεται προσοχή στο ότι ο σύνδεσμος $\to$ είναι δεξιά προσεταιριστικός και άρα αυτό είναι ισοδύναμο με $A \to B \to C$. Επομένως, μια συνάρτηση δύο μεταβλητων μπορεί να οριστεί πολύ απλά με χρήση διπλής λ-αφαίρεσης: $\lambda(x : A).\lambda(y : B)\Phi$.

2.4.4 Ο τύπος Empty

Χρήσιμη είναι επίσης η ύπαρξη της έννοιας του κενού τύπου, δυϊκή κατασκευή του τύπου **1**. Συμβολίζεται και ως **0**. Διαισθητικά, αποτελεί έναν τύπο ο οποίος δεν περιέχει και δεν μπορεί να περιέχει καμία κατασκευή και άρα εκφράζει την ψευδή πρόταση. Ως εκ τούτου, μπορούμε πάντα να κατασκευάσουμε μια συνάρτηση από τον κενό τύπο προς οποιονδήποτε τύπο χωρίς να δώσουμε κάποια εξίσωση για αυτή λόγω του ότι ο **0** δεν έχει κανένα στοιχείο. Έτσι, δεν υπάρχει κανόνας εισαγωγής στοιχείων αυτού του τύπου και υπάρχει κανόνας απαλοιφής προς οποιονδήποτε άλλον τύπο.

$$\frac{}{\Gamma \vdash \mathbf{0} : \mathtt{type}}\ F-\mathbf{0}$$

$$\frac{\Gamma \vdash A : \mathtt{type}}{\Gamma, x : \mathbf{0} \vdash \mathtt{abort}(x) : A}\ -\mathbf{0}$$

Ακόμη, αφού δεν υπάρχουν στοιχεία στον **0**, δεν υπάρχουν αντίστοιχα και κανόνες β και η αναγωγής.

Ο τύπος αυτός είναι πολλές φορές χρήσιμος όταν θέλουμε να εκφράσουμε μια άρνηση ή να αποδείξουμε μια. Αν υποθέσουμε ότι A είναι ένας τύπος, ορίζουμε τον τύπο της άρνησής του ως εξής

$$\neg A :\equiv A \to \mathbf{0}$$

Επομένως, όταν θέλουμε να αποδείξουμε μια άρνηση αρκεί να υποθέσουμε ότι ισχύει η πρόταση χωρίς την άρνηση και να καταλήξουμε σε ένα αντικείμενο του τύπου **0**. Αυτό θα πρέπει να διακριθεί από τη μέθοδο της επαγωγής σε άτοπο της κλασικής λογικής. Εκεί, προκειμένου να αποδείξουμε ότι κάτι ισχύει υποθέτουμε ότι ισχύει η άρνησή του και προσπαθούμε να καταλήξουμε σε άτοπο. Δηλαδή σε γλώσσα θεωρίας τύπων, για να αποδείξουμε το A παράγαγουμε μια κατασκευή του $\neg A \to \mathbf{0} \equiv \neg\neg A$ το οποίο είναι φυσικά στην κλασική λογική ισοδύναμο με το A αλλά όχι εξ ορισμού στη θεωρία τύπων. Στη θεωρία τύπων μια πρόταση για την οποία ισχύει ότι $\neg\neg A \to A$ ονομάζεται σταθερή (stable) και υπάρχουν προτάσεις που δεν είναι σταθερές.

2.4.5 Συγινόμενα Τύπων

Τώρα, όπως πριν με τους τύπους **1** και **0**, θα ορίσουμε τον δυϊκό τύπο των γινομένων, τα συγινόμενα (coproducts). Συμβολίζουμε την πράξη του συγινομένου πάνω σε δύο τύπους A και B ως $A+B$. Τα συγινόμενα εκφράζουν την ένωση των στοιχείων δύο τύπων διατηρώντας την πληροφορία της προέλευσης του κάθε στοιχείου. Αυτό είναι καίριας σημασίας κυρίως υπό το πρίσμα του Props as Types. Αν για παράδειγμα θέλουμε να παρουσιάσουμε ένα στοιχείο του $A+B$, τότε αυτό θα πρέπει να είναι είτε ένα στοιχείο του A είτε ένα στοιχείο του B, ιδωμένα εντός του $A+B$ μέσω του κατάλληλου κάθε φορά κατασκευαστή $\mathtt{inl} : A \to A+B$ και $\mathtt{inr} : B \to A+B$ αντίστοιχα. Αυτή η πληροφορία προέλευσης, αφού διατηρείται, είναι διαθέσιμη κατά τη δημιουργία απεικονίσεων από συγινόμενα σε άλλους τύπους και επιτρέπει τη διαφοροποίηση σε αντικείμενα που προέρχονται από τον A ή B. Είναι δηλαδή μια μορφή pattern matching πάνω στους όρους του A ή B.
Για τον σχηματισμό επομένως αυτού του τύπου έχουμε

$$\frac{\Gamma \vdash A : \mathtt{type} \quad \Gamma \vdash B : \mathtt{type}}{\Gamma \vdash A+B : \mathtt{type}}\ F-+$$

Οι αντίστοιχοι κανόνες εισαγωγής και απαλοιφής είναι:

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathtt{inl}(M) : A+B}\ I_1-+ \qquad \frac{\Gamma \vdash M : B}{\Gamma \vdash \mathtt{inr}(M) : A+B}\ I_2-+$$

$$\frac{\Gamma, x : A \vdash P : C \quad \Gamma, y : B \vdash Q : C \quad \Gamma \vdash M : A+B}{\Gamma \vdash \mathtt{case}(x.P, y.Q)(M) : C}\ E-+$$

Το υπολογιστικό περιεχόμενο του pattern matching είναι προφανώς το παρακάτω

$$\frac{\Gamma, x : A \vdash P : C \quad \Gamma, y : B \vdash Q : C \quad \Gamma \vdash M : A}{\Gamma \vdash \mathtt{case}(x.P, y.Q)(\mathtt{inl}(M)) \equiv [M/x]P : C}\ \beta_1-+$$

$$\frac{\Gamma, x : A \vdash P : C \quad \Gamma, y : B \vdash Q : C \quad \Gamma \vdash M : B}{\Gamma \vdash \mathtt{case}(x.P, y.Q)(\mathtt{inr}(M)) \equiv [M/y]Q : C}\ \beta_2-+$$

Και για τον κανόνα η-αναγωγής πολύ απλά ότι ό,τι συμπεριφέρεται σαν case analyser πάνω από ένα συγινόμενο είναι στην πραγματικότητα κάποιο pattern matching

$$\frac{\Gamma, z : A + B \vdash P : C}{\Gamma, M : A + B \vdash [M/z]P \equiv \mathsf{case}(x.[\mathtt{inl}(x)/z]P, y.[\mathtt{inr}(y)/z]P)(M) : C} \ \eta - +$$

Ο τύπος Bool Σαν ειδική περίπτωση των παραπάνω μπορούμε να ορίσουμε τον τύπο Bool ως $\mathbf{1} + \mathbf{1}$, τον τύπο δηλαδή που αποτελείται από δύο μόνο στοιχεία με μοναδική πληροφορία που έχει το καθένα να είναι η διάκρισή του από το άλλο. Μπορούμε για να γίνει αυτό πιο σαφές να ονομάσουμε $\mathtt{tt} \equiv \mathtt{inl}(\langle\rangle)$ και $\mathtt{ff} \equiv \mathtt{inr}(\langle\rangle)$. Ακόμα, μπορούμε να ονομάσουμε το pattern matching πάνω σε αυτά διαφορετικά ώστε να είναι πιο φυσικό ως if ... then ... else, όπως στις περισσότερες γλώσσες συναρτησιακού προγραμματισμού.

$$\frac{\Gamma \vdash P : C \quad \Gamma \vdash Q : C \quad \Gamma \vdash M : A + B}{\Gamma \vdash \mathtt{if}\ M\ \mathtt{then}\ P\ \mathtt{else}\ Q \equiv \mathsf{case}(P, Q)(M) : C} \ E - \mathtt{bool}$$

όπου τα P και Q δεν εξαρτώνται κάπως από τετριμμένο ούτως ή άλλως $\langle\rangle$.

2.4.6 Οι Φυσικοί Αριθμοί

Ήρθε η ώρα να εξετάσουμε με ποιον τρόπο ορίζεται ένα από τα πιο βασικά αντικείμενα της θεωρίας συνόλων και των μαθηματικών γενικότερα μέσα στην ITT, οι Φυσικοί Αριθμοί. Η προσέγγιση είναι αρκετά παρόμοια με αυτή στην αξιωματική θεμελίωση της αριθμητικής του Peano. Αρχικά, υπάρχει ένας κανόνας σχηματισμού που ορίζει την ύπαρξη ενός τύπου nat (F-nat). Τα στοιχεία αυτού του τύπου είναι ένα στοιχείο, το 0, και μετά για κάθε άλλο στοιχείο του nat μπορούμε να δημιουργήσουμε ένα άλλο μέσω μιας συνάρτησης $\mathsf{suc} : \mathsf{nat} \to \mathsf{nat}$.

$$\frac{}{\Gamma \vdash \mathsf{zero} : \mathsf{nat}} \ I_0 - \mathsf{nat} \qquad \frac{\Gamma \vdash M : \mathsf{nat}}{\Gamma \vdash \mathsf{suc}(M) : \mathsf{nat}} \ I_s - \mathsf{nat}$$

Ο τρόπος με τον οποίο μπορούμε να δημιουργήσουμε μια απεικόνιση από τον τύπο nat προς οποιονδήποτε τύπο A είναι να την ορίσουμε *αναδρομικά*. Απαιτείται δηλαδή μια τιμή $P : A$ της απεικόνισης στο zero και μια τιμή $n.x.Q : A$ στο $\mathsf{suc}(n)$ δωσμένων των $n : A$ και $x : A$ (η τιμή της απεικόνισης στο n).

$$\frac{\Gamma \vdash M : \mathsf{nat} \quad \Gamma \vdash P : A \quad \Gamma, n : \mathsf{nat}, x : A \vdash Q : A}{\Gamma \vdash \mathsf{rec}(P, n.x.Q)(M) : A} \ E - \mathsf{nat}$$

Ο τρόπος υπολογισμού του recursor είναι ο προφανής. Δηλαδή:

$$\frac{\Gamma \vdash P : A \quad \Gamma, n : \mathsf{nat}, x : A \vdash Q : A}{\Gamma \vdash \mathsf{rec}(P, n.x.Q)(\mathsf{zero}) \equiv P : A} \ \beta_0 - \mathsf{nat}$$

$$\frac{\Gamma \vdash M : \mathsf{nat} \quad \Gamma \vdash P : A \quad \Gamma, n : \mathsf{nat}, x : A \vdash Q : A}{\Gamma \vdash \mathsf{rec}(P, n.x.Q)(\mathsf{suc}(M)) \equiv [M, \mathsf{rec}(P, n.x.Q)(M)/n, x]Q : A} \ \beta_{\mathsf{suc}} - \mathsf{nat}$$

Ενώ όμοια με τον κανόνα η-recursion για τα συγινόμενα έχουμε

$$\frac{\Gamma, z : \mathsf{nat} \vdash M : A \quad \Gamma \vdash [\mathsf{zero}/z]M \equiv P : A \quad \Gamma, z : \mathsf{nat} \vdash [\mathsf{suc}(z)/z]M \equiv [z, M/n, x]Q : A}{\Gamma, z : \mathsf{nat} \vdash M \equiv \mathsf{rec}(P, n.x.Q)(z)} \ \eta - \mathsf{nat}$$

Εξαρτημένοι Τύποι Θα μπορούσε να αναρωτηθεί κανείς αν γίνεται να χρησιμοποιηθούν οι φυσικοί όπως παρουσιάστηκαν πιο πάνω για τη δημιουργία αποδείξεων με επαγωγή. Για παράδειγμα, αν υποθέσουμε ότι έχουμε ορίσει αναδρομικά μια πράξη πρόσθεσης στο nat, πώς γίνεται να αποδείξουμε ότι είναι αντιμεταθετική; Σκεπτόμενοι ότι η πρόταση που θέλουμε να αποδείξουμε είναι ένας τύπος (Props as Types) θα πρέπει για κάθε $n, m : \mathsf{nat}$ να υπάρχει ένας τύπος ο οποίος να εκφράζει τη ζητούμενη ιδιότητα.

Στην ITT κάτι τέτοιο επιτυγχάνεται με την εισαγωγή των *εξαρτημένων τύπων* (dependent types). Εξαρτημένος τύπος είναι μια οικογένεια τύπων $x.B$ δεικτοδοτημένων από στοιχεία ενός τύπου A. Ένα πολύ απλό παράδειγμα αποτελεί η οικογένεια τύπων Seq

$$\Gamma, x : \texttt{nat} \vdash \texttt{Seq}(x) : \texttt{type}$$

η οποία για κάθε φυσικό αριθμό n ορίζει τον τύπο των πεπερασμένων ακολουθιών φυσικών αριθμών μήκους ακριβώς n. Δηλαδή για κάθε στοιχείο του τύπου nat ορίζεται ένας ξεχωριστός τύπος με παράμετρο αυτο το στοιχείο. Είναι δυνατό, επίσης, ο τύπος $x.B$ να ορίζεται χωρίς κάποια εξάρτηση στη μεταβλητή x. Τότε έχουμε τη σταθερή οικογένεια όπου για κάθε στοιχείο του τύπου δεικτών έχουμε ακριβώς τον ίδιο τύπο B.

Ένα άλλο πολύ σημαντικό παράδειγμα που θα εξετασθεί αργότερα αποτελεί αυτό των οικογενειών των τύπων ισότητας πάνω από έναν τύπο. Έστω τύπος A και $a, b : A$. Τότε μια απόδειξη της ισότητας των στοιχείων a, b είναι μια κατασκευή του τύπου $\texttt{Id}_A(a, b)$, του τύπου δηλαδή όλων των «μαρτύρων» ισότητας των a, b. Άρα, στο πιο πάνω παράδειγμα αναζητούμε έναν όρο abel ώστε $\Gamma, n, m : \texttt{nat} \vdash abel : \texttt{Id}_A(n + m, m + n)$.

Τώρα που παρουσιάσαμε την ύπαρξη οικογενειών εξαρτημένων τύπων ας αναλύσουμε πώς μπορούμε να ορίσουμε απεικονίσεις από τους φυσικούς αριθμούς σε τέτοιες οικογένειες εξαρτημένες από τον nat. Για κάθε $n : \texttt{nat}$ θέλουμε ένα στοιχείο στον τύπο $[n/z]C$ όπου $z.C$ είναι μια οικογένεια τύπων με δείκτες στο nat.

$$\frac{\Gamma \vdash M : \texttt{nat} \quad \Gamma, z : \texttt{nat} \vdash C : \texttt{type} \quad \Gamma \vdash P : [\texttt{zero}/z]C \quad \Gamma, n : \texttt{nat}, x : [n/z]C \vdash Q : [\texttt{suc}(n)/z]C}{\Gamma \vdash \texttt{ind}[z.C](P, n.z.Q)(M) : [M/z]C} \; E - \texttt{nat}$$

Οι εξαρτημένες παραλλαγές των β και η αναγωγών είναι παρόμοιες.

2.4.7 Τύποι Εξαρτημένων Συναρτήσεων

Σε αυτήν την υποενότητα και στις δύο επόμενες θα παρουσιάσουμε τις εξαρτημένες εκδοχές ορισμένων από των κατασκευών που είδαμε. Εδώ θα αναλύσουμε τον τύπο των εξαρτημένων συναρτήσεων. Αν έχουμε μια οικογένεια Τύπων $x.B$ ως συνάρτηση ενός τύπου A (για κάθε $a : A$ ο $[a/x]B$ είναι κάποιος τύπος) τότε μια εξαρτημένη συνάρτηση είναι μια διαδικασία κατασκευής ενός αντικειμένου $f(a) : [a/x]B$ για κάθε $a : A$. Άρα, οι κανόνες για τον σχηματισμό του τύπου, για τη λ-αφαίρεση και την εφαρμογή είναι αρκετά παρόμοιοι με τη μη εξαρτημένη περίπτωση.

$$\frac{\Gamma \vdash A : \texttt{type} \quad \Gamma, x : A \vdash B : \texttt{type}}{\Gamma \vdash \Pi x : A.B : \texttt{type}} \; F - \Pi$$

$$\frac{\Gamma, a : A \vdash M : [a/x]B}{\Gamma \vdash \lambda(x : A)M : \Pi x : A.B} \; I - \Pi$$

$$\frac{\Gamma \vdash M : \Pi x : A.B \quad \Gamma \vdash N : A}{\Gamma \vdash M(N) : [N/x]B} \; E - \Pi$$

Είναι σημαντικό να σημειώσουμε την εξάρτηση που υπάρχει μεταξύ του τύπου του αποτελέσματος και του ορίσματος της συνάρτησης. Είναι μια πληροφορία η οποία διαδίδεται στο επίπεδο των τύπων. Κατά τα άλλα, η συμπεριφορά είναι αρκετά όμοια με την απλή περίπτωση συναρτήσεων. Αν, μάλιστα, η B είναι η σταθερή οικογένεια τότε $\Pi x : A.B \equiv A \to B$. Οι κανόνες β και η αναγωγής είναι πανομοιότυποι με την απλή περίπτωση αλλά τους παραθέτουμε εδώ για λόγους πληρότητας.

$$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash N : A}{\Gamma \vdash (\lambda(x : A)M)(N) \equiv [N/x]M : [N/x]B} \; \beta - \Pi$$

$$\frac{\Gamma \vdash M : \Pi x : A.B}{\Gamma \vdash (\lambda(x : A)M(x)) \equiv M : \Pi x : A.B} \; \eta - \Pi$$

Ο τύπος αυτός κάτω από το πρίσμα του Props as Types εκφράζει προτάσεις της μορφής $\forall x : A.\Phi$, όπου A κάποιος τύπος και Φ μια πρόταση σχετικά με τα αντικείμενα του A.

2.4.8 Τύποι Εξαρτημένων Ζευγών

Τα ζεύγη, δηλαδή τα αντικείμενα ενός τύπου γινομένου, μπορούν και αυτά να γενικευθούν σε εξαρτημένες εκδοχές. Σε αυτήν την περίπτωση έχουμε εξάρτηση του τύπου του δεύτερου όρου του ζεύγους από τον πρώτο όρο. Όπως και πριν, αν $x.B$ είναι μια οικογένεια τύπων ως συνάρτηση ενός τύπου A, τότε για ένα εξαρτημένο ζεύγος $\langle a, b\rangle$ ισχύει ότι $a : A$ και $b : [a/x]B$.
Επομένως, με τη μορφή κανόνων, όποτε έχουμε μια εξαρτημένη οικογένεια τύπων μπορούμε να σχηματίσουμε τον τύπο του εξαρτημένου γινομένου (Σ type)

$$\frac{\Gamma \vdash A : \texttt{type} \quad \Gamma, x : A \vdash B : \texttt{type}}{\Gamma \vdash \Sigma x : A.B : \texttt{type}}\ F-\Sigma$$

Ο συμβολισμός για τα ζεύγη και για τις προβολές τους παραμένει ίδιος με αυξημένη σημασία

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : [M/x]B}{\Gamma \vdash \langle M, N\rangle : \Sigma x : A.B}\ I-\Sigma$$

$$\frac{\Gamma \vdash M : \Sigma x : A.B}{\Gamma \vdash \texttt{fst}(M) : A}\ E_1-\Sigma \qquad \frac{\Gamma \vdash M : \Sigma x : A.B}{\Gamma \vdash \texttt{snd}(M) : [\texttt{fst}(M)/x]B}\ E_2-\Sigma$$

Κοιτώντας τον κανόνα απαλοιφής της δεύτερης προβολής γίνεται σαφής η εξάρτηση του τύπου του όρου $\texttt{snd}(M)$ από τον $\texttt{fst}(M)$. Με μια πιο «λογική ανάγνωση» των δύο αυτών κανόνων απαλοιφής, ένα αντικείμενο του τύπου $\Sigma x : A.B$ αποτελεί μια κατασκευή ενός αντικειμένου του τύπου A μαζί με μια απόδειξη ότι το αντικείμενο αυτό ικανοποιεί την ιδιότητα B. Στο γενικό ιντουϊσιονιστικό πλαίσιο όπου εργαζόμαστε αυτό αποκαλείται *κατασκευαστική ύπαρξη*. Δηλαδή, προκειμένου να αποδείξουμε ότι υπάρχει ένα αντικείμενο που ικανοποιεί μια ιδιότητα, πρέπει να έχουμε έναν (αλγοριθμικό) τρόπο κατασκευής αυτού του αντικειμένου και να το συμπεριλάβουμε μαζί με την απόδειξη ότι ικανοποιεί την ζητούμενη ιδιότητα. Για τον λόγο αυτόν, ο τύπος $\Sigma x : A.B$ αντιστοιχίζεται στον υπαρξιακό ποσοδείκτη της προτασιακής λογικής.

Η υπολογιστική συμπεριφορά των ζευγών αυτών θέλουμε να είναι παρόμοια με τη μη εξαρτημένη περίπτωση. Έτσι, έχουμε:

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : [M/x]B}{\Gamma \vdash \texttt{fst}\langle M, N\rangle \equiv M : A}\ \beta_1-\Sigma \qquad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : [M/x]B}{\Gamma \vdash \texttt{snd}\langle M, N\rangle \equiv N : [M/x]B}\ \beta_2-\Sigma$$

και τέλος ίδια είναι και η η-αναγωγή

$$\frac{\Gamma \vdash M : \Sigma x : A.B}{\Gamma \vdash \langle \texttt{fst}(M), \texttt{snd}(N)\rangle \equiv M : \Sigma x : A.B}\ \eta-\Sigma$$

2.4.9 Εξαρτημένα Συγινόμενα Τύπων

Για τα συγινόμενα τύπων δεν θα αλλάξουμε τίποτε άλλο παρά μόνο τον τρόπο που μπορούμε να δημιουργούμε απεικονίσεις που ξεκινούν από αυτά. Όπως και στην περίπτωση των Φυσικών Αριθμών, θα επεκτείνουμε την αναδρομή, στην περίπτωση των συγινομένων το case analysis, σε επαγωγή επιτρέποντας οι τιμές των συναρτήσεων μας να ανήκουν σε διαφορετικούς τύπους ανάλογα με το εκάστοτε όρισμα. Επομένως, ο νέος κανόνας απαλοιφής στα συγινόμενα θα είναι ο παρακάτω

$$\frac{\Gamma, z : A+B \vdash C : \texttt{type} \quad \Gamma, x : A \vdash P : [\texttt{inl}(x)/z]C \quad \Gamma, y : B \vdash Q : [\texttt{inr}(y)/z]C \quad \Gamma \vdash M : A+B}{\Gamma \vdash \texttt{case}(x.P, y.Q)(M) : [M/z]C}\ E-+$$

Ο νέος κανόνας είναι πολύ πιο εκφραστικός. Αν έχουμε μια οικογένεια $z.C$ τύπων εξαρτημένη από τους όρους ενός τύπου $A+B$, για κάθε αντικείμενο $a : A$ μπορούμε να κατασκευάσουμε ένα αντικείμενο $P : [\texttt{inl}(a)/z]C$ και αντίστοιχα για κάθε αντικείμενο $b : B$ να κατασκευάσουμε ένα αντικείμενο $Q : [\texttt{inr}(b)/z]C$. Τότε υπάρχει η φυσική (εξαρτημένη) συνάρτηση από το $A + B$ στην οικογένεια τύπων $z.C$ η οποία ορίζεται με pattern matching.

2.4.10 Σύμπαντα Τύπων

Σε αυτήν την ενότητα θα περιγράψουμε μια πολύ σημαντική έννοια, η εισαγωγή της οποίας αποσκοπεί στην αύξηση της εκφραστικότητας της θεωρίας. Μέχρι τώρα χρησιμοποιούσαμε το judgment $a : \mathtt{type}$ παρότι δεν είναι ένα από τα δύο βασικά judgments της Θεωρίας τύπων που παρουσιάσαμε στην αντίστοιχη ενότητα. Τώρα αυτό θα γίνει πιο ακριβές και αυστηρό με την εισαγωγή των Συμπάντων Τύπων. Ένα σύμπαν $\mathcal{U}$ είναι ένας τύπος τα στοιχεία του οποίου είναι τύποι. Θα μπορούσε να αναρωτηθεί κανείς αν αυτό σημαίνει ότι περιλαβάνει και τον εαυτό του, δηλαδή αν ισχύει ότι $\mathcal{U} : \mathcal{U}$. Αποδεικνύεται ότι αν ίσχυε κάτι τέτοιο θα μπορούσαμε να αναπαραγάγουμε μια εκδοχή ενός συνολοθεωρητικού παραδόξου, π.χ. του παράδοξου Buralli-Forti, και άρα η θεωρία θα ήταν ασυνεπής κάνοντας οποιονδήποτε τύπο, συμπεριλαμβανομένου και του **0** να έχει στοιχεία.

Για να αποφευχθεί κάτι τέτοιο ορίζουμε μια αθροιστική ιεραρχία συμπάντων $\mathcal{U}_i$

$$\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : ...$$

όπου με τον όρο αθροιστική εννοούμε ότι κάθε στοιχείο-τύπος που ανήκει στο $\mathcal{U}_i$ ανήκει και στο $\mathcal{U}_{i+1}$. Όταν γράφουμε, λοιπόν, ότι το A είναι τύπος, θα εννοούμε ότι ανήκει σε κάποιο σύμπαν $\mathcal{U}_i$. Αρκετά συχνά θα αποφεύγουμε να αναφέρουμε ρητά το επίπεδο i της ιεραρχίας και θα γράφουμε απλά $A : \mathcal{U}$. Με αυτόν τον τρόπο είναι δυνατόν να γράψουμε ακόμα και $\mathcal{U} : \mathcal{U}$ όπου οι παραλειπόμενοι δείκτες θα ήταν $\mathcal{U}_i : \mathcal{U}_{i+1}$. Αυτός ο τρόπος γραφής αναφέρεται συχνά ως *typical ambiguity* και -καίτοι βολικός- μπορεί να προκαλέσει λάθη. Ακόμη, συνηθίζεται να αποκαλούμε τους τύπους εντός ενός δεδομένου σύμπαντος $\mathcal{U}$ μικρούς τύπους (small types).

Τα σύμπαντα κάθε επιπέδου είναι κλειστά ως προς τις κατασκευές που είδαμε ως τώρα. Αυτό σημαίνει πως τα ακόλουθα ισχύουν

$$\frac{}{\mathtt{nat} : \mathcal{U}} \qquad \frac{}{\mathbf{0} : \mathcal{U}} \qquad \frac{}{\mathbf{1} : \mathcal{U}}$$

$$\frac{A : \mathcal{U} \quad B : \mathcal{U}}{A \times B : \mathcal{U}} \qquad \frac{A : \mathcal{U} \quad B : \mathcal{U}}{A \to B : \mathcal{U}} \qquad \frac{A : \mathcal{U} \quad B : \mathcal{U}}{A + B : \mathcal{U}} \qquad \frac{A : \mathcal{U} \quad B : \mathcal{U}}{\Sigma x : A.B : \mathcal{U}} \qquad \frac{A : \mathcal{U} \quad B : \mathcal{U}}{\Pi x : A.B : \mathcal{U}}$$

Με τα σύμπαντα μπορούμε να δούμε τις εξαρτημένες οικογένειες τύπων $x.B$ από έναν τύπο A σαν συναρτήσεις $B : A \to \mathcal{U}$. Για να δώσουμε ένα παράδειγμα της εκφραστικής δυνατότητας που μας προσφέρουν τα σύμπαντα σε συνδυασμό με τους εξαρτημένους τύπους θεωρούμε τα εξής:
Αφού $\mathtt{bool} : \mathcal{U}_i$ και $\mathtt{nat} : \mathcal{U}_i$, μπορούμε να ορίσουμε έναν εξαρτημένο τύπο $\lambda x.\mathtt{if}\ x\ \mathtt{then}\ \mathtt{nat}\ \mathtt{else}\ \mathtt{bool} : \mathtt{bool} \to \mathcal{U}_i$. Αυτό μας επιτρέπει να γράφουμε προγράμματα όπως

$$\mathtt{if}\ M\ \mathtt{then}\ 17\ \mathtt{else}\ \mathtt{tt} : \mathtt{if}\ M\ \mathtt{then}\ \mathtt{nat}\ \mathtt{else}\ \mathtt{bool}$$

τα οποία κάνουν type check. Αυτό είναι αδύνατο στις περισσότερες γλώσσες προγραμματισμού. Τέλος, να παρατηρήσουμε ότι η πληροφορία του ορίσματος περνάει στον τύπο του αποτελέσματος.

2.4.11 Τύποι Ισότητας

Περνάμε τώρα σε έναν από τους πιο ουσιαστικούς τύπους, τον τύπο ισότητας δύο στοιχείων ενός τύπου. Αρχικά, θα δώσουμε τους ορισμούς του τύπου και έπειτα θα περιγράψουμε κάποιες σημαντικές ιδιότητές του.

Ορισμός Έστω τύπος και $a, b : A$. Τότε μια απόδειξη της ισότητας των a και b θα είναι φυσικά κάποιο πρόγραμμα και άρα θα ικανοποιεί την προδιαγραφή του αντίστοιχου τύπου. Ο τύπος αυτός είναι ο τύπος Ισότητας των a, b ως στοιχεία του A και συμβολίζεται ως $\mathtt{Id}_A(a, b)$ ή $a =_A b$ ή όταν δεν υπάρχει κίνδυνος σύγχυσης του τύπου των στοιχείων απλά $a = b$. Όπως αντιλαμβανόμαστε δεν πρόκειται απλώς για έναν τύπο αλλά για μια οικογένεια τύπων με δείκτες δύο στοιχεία του A, για κάθε τύπο A. Ο αντίστοιχος κανόνας σχηματισμού αυτού του τύπου είναι επομένως αυτός που περιγράψαμε

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash M : A \quad \Gamma \vdash N : A}{\Gamma \vdash \mathtt{Id}_A(M, N) : \mathcal{U}}\ F - \mathtt{Id}$$

Χρειάζεται προσοχή ώστε να μην συγχέουμε αυτού του είδους την ισότητα με την ισότητα εξ ορισμού (judgmental equality). Σε εκείνη την περίπτωση η ισότητα μεταξύ δύο όρων είναι στο επίπεδο των ορισμών και των υπολογισμών και η απόφανση $a \equiv b$ δεν είναι κάποιος τύπος. Στην περίπτωση του τύπου της ισότητας ισχύει ότι όταν δύο αντικείμενα είναι ίσα (λέμε και προτασιακά ίσα (propositional equal) για να τα διακρίνουμε) αυτό σημαίνει ότι ο τύπος της ισότητας δεν είναι κενός. Δηλαδή υπάρχει ένα αντικείμενο $p : \mathtt{Id}_A(a, b)$. Όπως είδαμε και στην περίπτωση άλλων τύπων, αυτό είναι πολύ σημαντικό. Ο τύπος $\mathtt{Id}_A(a, b)$ μπορεί να περιλαμβάνει πολλές αποδείξεις "p" η καθεμία με διαφορετικές πληροφορίες και να έχει μια πιο σύνθετη δομή.

Προφανώς, θα θέλαμε η σχέση της ισότητας που θα ορίσουμε να είναι κατ'ελάχιστον αυτοπαθής. Άρα, ο κανόνας εισαγωγής θα πρέπει να μας επιτρέπει να μπορούμε να κατασκευάζουμε προγράμματα-μάρτυρες της ισότητας ενός αντικειμένου με τον εαυτό του.

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathtt{refl}_A(M) : \mathtt{Id}_A(M, M)}\ I - \mathtt{Id}$$

Η απόδειξη reflexivity (`refl`) κάνει, λοιπόν, ακριβώς αυτό και όταν δεν υπάρχει κίνδυνος σύγχυσης θα παραλείπουμε τον τύπο ή το αντικείμενο M. Συγκεκριμένα, όταν έχουμε $a \equiv b$, έχουμε και $\mathtt{refl} : \mathtt{Id}_A(a, b)$ το οποίο κάνει type check αφού το ότι $a \equiv b$ σημαίνει ότι ο τύπος $\mathtt{Id}_A(a, b)$ είναι ίσος εξ ορισμού με τον τύπο $\mathtt{Id}_A(a, a)$ ο οποίος είναι ο τύπος του $\mathtt{refl}_A(a)$.

Είναι λογικό, επίσης, όταν έχουμε δύο ίσα αντικείμενα, αυτά να ικανοποιούν τις ίδιες ιδιότητες, κατά κάποιον τρόπο. Αυτό σημαίνει ότι θα πρέπει η ισότητα να είναι η ελάχιστη αυτοπαθής σχέση. Δηλαδή οποτεδήποτε μια σχέση είναι αυτοπαθής, τότε θα πρέπει τα ίσα αντικείμενα να την ικανοποιούν. Αυτό είναι μια πολύ λογική απαίτηση που θα μπορούσε να έχει κανείς από έναν τύπο ισότητας αντικειμένων. «Αφού είναι ίσα, τότε θα πρέπει να είναι διαφανής η διαφορά τους σε κάθε σχέση που σέβεται την αυτοπάθεια». Αυτό ακριβώς εκφράζει ο κανόνας απαλοιφής και ο τελεστής $\mathtt{J}$.

$$\frac{\Gamma \vdash P : \mathtt{Id}_A(M, N) \quad \Gamma, x, y : A, z : \mathtt{Id}_A(x, y) \vdash C : \mathcal{U} \quad \Gamma, x : A \vdash Q : [x, x, \mathtt{refl}_A(x)/x, y, z]C}{\Gamma \vdash \mathtt{J}[x.y.z.C](x.Q)(P) : [M, N, P/x, y, z]C}\ E - \mathtt{Id}$$

Ο κανόνας αυτός ονομάζεται και *επαγωγή μονοπατιού* (path induction) λόγω της ομοτοπικής ερμηνείας που θα δούμε αργότερα. Επίσης, ο τελεστής $\mathtt{J}$ ικανοποιεί τον εξής υπολογιστικό κανόνα β-αναγωγής.

$$\frac{\Gamma \vdash M : A \quad \Gamma, x, y : A, z : \mathtt{Id}_A(x, y) \vdash C : \mathcal{U} \quad \Gamma, x : A \vdash Q : [x, x, \mathtt{refl}_A(x)/x, y, z]C}{\Gamma \vdash \mathtt{J}[x.y.z.C](x.Q)(\mathtt{refl}_A(M)) \equiv [M/x]Q : [M, M, \mathtt{refl}_A(M)/x, y, z]C}\ \beta - \mathtt{Id}$$

Ιδιότητες Αν και αποφεύγουμε την παράθεση αποδείξεων ιδιοτήτων διάφορων τύπων, θα παρουσιάσουμε εδώ τις πιο σημαντικές διότι θα εμφανιστούν στα επόμενα.

Η ισότητα είναι σχέση ισοδυναμίας Η ισότητα είναι αυτοπαθής εξορισμού αφού για κάθε $a : A$ έχουμε $\mathtt{refl} : Id_A(a, a)$. Για να αποδείξουμε ότι είναι συμμετρική θα πρέπει για κάθε $a, b : A$ και $p : \mathtt{Id}_A(a, b)$ να βρούμε ένα αντικείμενο $q : \mathtt{Id}_A(b, a)$. Με επαγωγή μονοπατιού αρκεί για κάθε $x : A$ να βρούμε ένα $q : \mathtt{Id}_A(x, x)$. Αυτό είναι προφανώς το `refl`. Άρα, έχουμε

$$\mathtt{sym} : \Pi a, b : A.\mathtt{Id}_A(a, b) \to \mathtt{Id}_A(b, a)$$

με $\mathtt{sym} :\equiv \lambda a.\lambda b.\lambda p.\mathtt{J}[x, y, z.\mathtt{Id}_A(y, x)](x.\mathtt{refl}_A(x))(p)$.
Για να αποδείξουμε ότι είναι μεταβατική υποθέτουμε ότι για $a, b, c : A$ μας δίνονται $p : \mathtt{Id}_A(a, b)$ και $q : \mathtt{Id}_A(b, c)$ και ζητάμε ένα $r : \mathtt{Id}_A(a, c)$. Αν το δούμε διαφορετικά, για δοσμένο $c : A$ θέλουμε για κάθε $x, y : A$ και $z : \mathtt{Id}_A(x, y)$ μια συνάρτηση με τύπο $\mathtt{Id}_A(y, c) \to \mathtt{Id}_A(x, c)$. Πάλι, με επαγωγή μονοπατιού αρκεί να βρούμε μια τέτοια συνάρτηση για $x \equiv y$ και $z \equiv \mathtt{refl}$. Αυτή είναι η ταυτοτική συνάρτηση. Άρα

$$\mathtt{trans} : \Pi a, b, c : A.\mathtt{Id}_A(a, b) \to \mathtt{Id}_A(b, c) \to \mathtt{Id}_A(a, c)$$

με $\mathtt{trans} :\equiv \lambda a.\lambda b.\lambda c.\lambda p : \mathtt{Id}_A(a,b).\mathtt{J}[x,y,z.\mathtt{Id}_A(y,c) \to \mathtt{Id}_A(x,c)](x.\lambda w.w)(p)$.
Παρατηρήστε ότι οι ορισμοί των sym και trans ικανοποιούν εξ ορισμού κάποιες ισότητες, χάρις στην β-αναγωγή του J. Συγκεκριμένα

$$\mathtt{sym}(M)(M)(\mathtt{refl}) \equiv \mathtt{refl}$$

$$\mathtt{trans}(M)(M)(N)(\mathtt{refl})(Q) \equiv Q$$

Αν είχαμε ορίσει αλλιώς τις δύο αυτές συναρτήσεις, αν δηλαδή είχαμε δώσει άλλες αποδείξεις συμμετρικότητας και μεταβατικότητας, τότε αυτές θα ικανοποιούσαν άλλες υπολογιστικές ιδιότητες. Αυτό είναι χαρακτηριστικό των κατασκευαστικών μαθηματικών και αναφέρεται ως *Proof Relevance*. Τέλος, όταν τα $a, b, c : A$ είναι γνωστά, γράφουμε $\mathtt{sym}(a)(b)(p) \equiv p^{-1}$ και $\mathtt{trans}(a)(b)(c)(p)(q) \equiv p \cdot q$.

Οι συναρτήσεις σέβονται την ισότητα Μια πολύ λογική συμπεριφορά του τύπου της ισότητας που ορίσαμε θα ήταν οι συναρτήσεις να τη σέβονταν (Simple Functionality). Δηλαδή σε ίσα ορίσματα να αντιστοιχούν ίσα αποτελέσματα. Έστω δηλαδή τύποι $A, B : \mathcal{U}$, $f : A \to B$ και $a, b : A$. Με επαγωγή μονοπατιού, αρκεί να δείξουμε ότι αν $x \equiv a \equiv b$, τότε έχουμε μια απόδειξη ισότητας για τα $f(a)$ και $f(b)$. Αυτή είναι φυσικά η refl. Άρα, ορίζουμε

$$\Gamma, f : A \to B, x, y : A \vdash \mathtt{ap}_f : \mathtt{Id}_A(x,y) \to \mathtt{Id}_B(fx, fy)$$

με τύπο $\mathtt{ap}_f :\equiv \lambda p.\mathtt{J}[x,y,z.\mathtt{Id}_B(fx,fy)](x.\mathtt{refl}_B(fx))(p)$. Λόγω της β-αναγωγής του τελεστή J έχουμε ότι ισχύει και η ισότητα εξ όρισμού: $\mathtt{ap}_f(\mathtt{refl}_A(x)) \equiv \mathtt{refl}_B(fx)$.

Μεταφορά Έστω $x.B$ μια οικογένεια εξαρτημένων τύπων από τιμές στον τύπο $A, a, b, : A$ και $p : \mathtt{Id}_A(a,b)$. Τότε, θα περιμέναμε οι τύποι $[a/x]B$ και $[b/x]B$ να περιέχουν τα «ίδια» αντικείμενα. Θα πρέπει δηλαδή να υπάρχει μια συνάρτηση $\mathtt{tr}[x.B](p) : [a/x]B \to [b/x]B$ η οποία να στέλνει κάθε αντικείμενο του τύπου $[a/x]B$ στο αντίστοιχο αντικείμενο του τύπου $[b/x]B$. Με απόδειξη με επαγωγή μονοπατιού αρκεί να βρούμε τι συμβαίνει στην περίπτωση όπου $x \equiv a \equiv b$. Τότε η πιο φυσική αντιστοιχία μεταξύ των $[x/x]B$ και $[x/x]B$ είναι η ταυτοτική συνάρτηση.

$$\Gamma, a, b : A \vdash \mathtt{tr}[x.B] : \mathtt{Id}_A(a,b) \to [a/x]B \to [b/x]B$$

με $\mathtt{tr}[x.B] :\equiv \lambda p.\mathtt{J}[x,y,z.[x/x]B \to [y/x]B](x.\lambda w.w)(p)$. Η διαδικασία της μεταφοράς μεταξύ δύο τύπων πάνω από ένα μονοπάτι ισότητας $p : \mathtt{Id}_A(a,b)$ ονομάζεται transport και ορισμένες φορές συμβολίζεται και $\Gamma, v : [a/x]B \vdash p_*(v) : [b/x]B$. Αν το μονοπάτι πάνω από το οποίο μεταφέρουμε είναι το refl, τότε η μεταφορά είναι εξ ορισμού η ταυτοτική συνάρτηση: $\mathtt{tr}[x.B](\mathtt{refl}) \equiv \mathtt{id}$.

Μονοπάτι πάνω από μονοπάτι Ας δούμε τώρα μια γενίκευση της έννοιας του Simple Functionality και του τελεστή $\mathtt{ap}_f$ για εξαρτημένες συναρτήσεις. Αν δηλαδή έχουμε μια οικογένεια τύπων $x.B$ εξαρτημένη από τα στοιχεία ενός τύπου A, ή αλλιώς με τη χρήση του σύμπαντος μια συνάρτηση $B : A \to \mathcal{U}$, μια εξαρτημένη συνάρτηση $f : \Pi x : A.B$, δύο στοιχεία $a, b : A$ και μια απόδειξη ισότητας $p : \mathtt{Id}_A(a,b)$ μεταξύ τους θα θέλαμε μια απόδειξη ισότητας μεταξύ των στοιχείων fa και fb. Αυτό όμως δεν γίνεται διότι $fa : [a/x]B$ και $fb : [b/x]B$ και εν γένει οι τύποι $[a/x]B$ και $[b/x]B$ δεν είναι ίσοι εξ ορισμού. Το κοντινότερο, λοιπόν, που μπορούμε να αποδείξουμε σε αυτό είναι η ισότητα μεταξύ του $\mathtt{tr}[x.B](p)(fa) : [b/x]B$ και του $fb : [b/x]B$ τα οποία ανήκουν στον ίδιο τύπο. Δηλαδή του αντίστοιχου στοιχείου του fa στον $[b/x]B$ μέσω μεταφοράς πάνω από το μονοπάτι p. Η απόδειξη αυτή ονομάζεται dap, έχει τύπο

$$\Gamma, f : \Pi x : A.B, a, b : A \vdash \mathtt{dap}_f : \Pi p : \mathtt{Id}_A(a,b).\mathtt{Id}_{[b/x]B}(p_*(fa), fb)$$

και αποδεικνύεται με επαγωγή μονοπατιού ως $\mathtt{dap}_f :\equiv \lambda p.\mathtt{J}[x,y,z.\mathtt{Id}_{[y/x]B}(z_*(fx), fy)](x.refl(fx))(p)$. Φυσικά, εντελώς όμοια είναι η απόδειξη αν αντί να μεταφέρουμε το a μεταφέραμε το b. Επειδή ο τύπος $\mathtt{Id}_{[b/x]B}(p_*(fa), fb)$ είναι κατά κάποιο τρόπο ασύμμετρος, ορισμένες φορές προτιμάται να συμβολίζεται ως $fa =_p^{x.B} fb$. Φυσικά, αν ισχύει $fa =_p^{x.B} fb$, τότε $fb =_{p^{-1}}^{x.B} fa$.

Function Extensionality Σε αυτήν την παράγραφο ασχολούμαστε με την ισότητα σε έναν συγκεκριμένο τύπο, τους τύπους συναρτήσεων. Πότε είναι δύο συναρτήσεις $f, g : A \to B$ ίσες; Σύμφωνα με την κλασική συνολοθεωρία και τον ορισμό των συναρτήσεων ως το γράφημά τους, δύο συναρτήσεις είναι ίδιες όταν για κάθε σημείο του πεδίου ορισμού έχουμε ίσα αποτελέσματα. Αυτό ονομάζεται αξίωμα της εκτασιακότητας, *Function Extensionality*. Με συμβολισμό θεωρίας τύπων θα γράφαμε

$$\frac{\Gamma \vdash f, g : A \to B \quad \Gamma, x : A \vdash p : \mathtt{Id}_B(fx, gx)}{\Gamma \vdash \_ : \mathtt{Id}_{A \to B}(f, g)}$$

Στην περίπτωση όμως της ITT συμβαίνει κάτι το οποίο αντιβαίνει σε αυτήν την προσδοκία της διαίσθησής μας. Ο Martin-Löf απέδειξε ότι η Function Extensionality δεν ισχύει στην ITT. Για παράδειγμα, για τους συνήθεις επαγωγικούς ορισμούς της πρόσθεσης και του πολλαπλασιασμού στους φυσικούς αριθμούς αποδεικνύεται πολύ εύκολα ότι

$$n : \mathtt{nat} \vdash p : \mathtt{Id}_{\mathtt{nat}}(n + n, 2 * n)$$

όμως

$$\nexists q : \mathtt{Id}_{\mathtt{nat} \to \mathtt{nat}}(\lambda n.n + n, \lambda n.2 * n)$$

Προκειμένου να παρακαμφθεί αυτή η αδυναμία και να μπορούμε να κάνουμε εντός της ITT μαθηματικά που να σέβονται τη Function Extensionality, υπάρχουν πολλές διαφορετικές παραλλαγές. Μια απο αυτές είναι η εισαγωγή ενός αξιώματος στη θεωρία μας, του Axiom of Extensionality, δηλαδή την επιβολή της ύπαρξης ενός όρου `funext`, χωρίς κάποιον κατασκευαστικό ορισμό, ο οποίος ικανοποιεί ακριβώς αυτό

$$\mathtt{funext} : \Pi f, g : A \to B.((\Pi x : A.\mathtt{Id}_B(fx, gx)) \to \mathtt{Id}_{A \to B}(f, g))$$

Αυτό δίνει μια λύση στο ζήτημα του Function Extensionality, δημιουργεί, όμως, κάποια άλλα προβλήματα. Αρχικά, η προσθήκη και η παραδοχή αξιωμάτων είναι ενάντια στη φύση και στην παράδοση της Θεωρίας Τύπων. Ο λόγος για αυτό είναι ακριβώς η μη κατασκευαστική φύση των αξιωμάτων. Και δυστυχώς το πρόβλημα δεν είναι απλά φιλοσοφικό. Η προσθήκη αντικειμένων σε έναν τύπο η συμπεριφορά των οποίων δεν καθορίζεται από την κατασκευή τους έχει άμεσο αντίκτυπο στον υπολογιστικό τους χαρακτήρα. Στην περίπτωση του `funext` αυτό γίνεται άμεσα αισθητό όταν πάμε να κάνουμε επαγωγή μονοπατιού για ένα μονοπάτι που προκύπτει από τη χρήση του. Έστω δηλαδή, $C : \Pi x, y : A \to B.\Pi z : \mathtt{Id}_{A \to B}(x, y).\mathcal{U}$ να είναι μια ιδιότητα και έχουμε $Q : \Pi x : A \to B.C(x)(x)(\mathtt{refl})$. Τότε με ποιον τρόπο υπολογίζεται ένας όρος της μορφής $\mathtt{J}[x, y, z.C](x.Q)(\mathtt{funext}(f)(g)(H))$; Είναι ένας όρος με σωστό τύπο ο οποίος στην ITT δεν υπολογίζεται, είναι δηλαδή stuck. Αν δεν μας ενδιαφέρει το υπολογιστικό περιεχόμενο της θεωρίας μας αλλά μόνο το αποδεικτικό κομμάτι, τότε δεν υπάρχει κανένας λόγος να μην εντάξουμε το Axiom of Extensionality. Αν όμως θέλουμε να αντιμετωπίσουμε την Θεωρία Τύπων ως μια θεωρία υπολογισμού, τότε είναι λογική η αμφισβήτησή του. Όπως θα δούμε αργότερα με επέκταση της ITT σε μια νέα θεωρία, την Cubical Type Theory, το `funext` εμφανίζεται ως θεώρημα με απόδειξη- πρόγραμμα κι έτσι αποκτά την υπολογιστική του σημασία.

Κεφάλαιο 3

HoTT - Homotopy Type Theory

Η Ομοτοπική Θεωρία Τύπων είναι ένας νέος κλάδος των μαθηματικών ο οποίος συνδυάζει διάφορα πεδία με πολύ απρόσμενο τρόπο. Από τη μια, η Ομοτοπική Θεωρία είναι ένα παρακλάδι της Αλγεβρικής Τοπολογίας και της Ομολογικής Άλγεβρας και έχει άμεση σχέση με τη Θεωρία Κατηγοριών υψηλών διαστάσεων. Θα έλεγε κανείς ότι η πιο φυσική γλώσσα για την Ομοτοπική Θεωρία είναι η Θεωρία Κατηγοριών, αφού μπορεί να κωδικοποιήσει και να γενικεύσει τις βασικές κατασκευές της. Από την άλλη, η Θεωρία Τύπων είναι, όπως είδαμε, κλάδος της Μαθηματικής Λογικής και της Επιστήμης των Υπολογιστών. Εκτός από την εντυπωσιακή σύνδεση αυτών των κλάδων, η Ομοτοπική Θεωρία Τύπων δημιουργεί νέες ιδέες σχετικά με τη Θεμελίωση των Μαθηματικών: το *Axiom of Univalence* και οι *Επαγωγικοί Τύποι ανώτερης Διάστασης* (Higher Inductive Types). Το μεν πρώτο αξίωμα μας επιτρέπει να ταυτίσουμε ισομορφικές δομές, κάτι που ενυπάρχει στην καθημερινή πρακτική των μαθηματικών, ενώ οι επαγωγικοί τύποι ανώτερων διαστάσεων μας παρέχουν τη δυνατότητα λογικής περιγραφής βασικών χώρων και κατασκευών της Ομοτοπικής Θεωρίας όπως κύκλοι, τόροι, κύλινδροι, Truncations, κλπ με πολύ αφαιρετικό τρόπο.

Η Ομοτοπική Θεωρία Τύπων ερμηνεύει τη Θεωρία Τύπων ομοτοπικά. Στην Ομοτοπική Θεωρία Τύπων αντιμετωπίζουμε τους τύπους ως χώρους, όπως αυτούς στην Ομοτοπική Θεωρία, ή ως higher groupoids, και τις λογικές κατασκευές, όπως το γινόμενο $A \times B$, ως ομοτοπικά αναλλοίωτες κατασκευές πάνω σε αυτούς τους χώρους. Η Ομοτοπική Θεωρία μελετά τις ιδιότητες τοπολογικών χώρων και απεικονίσεων οι οποίες παραμένουν αναλλοίωτες υπό την *ομοτοπική ισοδυναμία*. Μια ομοτοπία μεταξύ δύο συνεχών απεικονίσεων $f, g : A \to B$ είναι μια συνεχής συνάρτηση $H : A \times [0, 1] \to B$ για την οποία ισχύει ότι $H(x, 0) = f(x)$ και $H(x, 1) = g(x)$, συμβολίζουμε $f \simeq g$. Είναι κατά κάποιον τρόπο μια συνεχής παραμόρφωση της f ώστε να ταυτιστεί με τη g. Δύο χώροι ονομάζονται ομοτοπικά ισοδύναμοι αν υπάρχουν δύο συνεχείς απεικονίσεις $f : A \to B$ και $g : B \to A$ ώστε $f \circ g \simeq 1_B$ και $g \circ f \simeq 1_A$, και γράφουμε $A \simeq B$. Οι ομοτοπικά ισοδύναμοι χώροι έχουν ίδιες αλγεβρικές αναλλοίωτες (θεμελιώδη Ομάδα, Ομολογία, κλπ) και λέμε ότι έχουν τον ίδιο *Ομοτοπικό Τύπο*.

Ας δούμε σύντομα τι συνεπάγεται η ερμηνεία αυτή στις πιο βασικές έννοιες. Η έννοια του $a : A$, το a είναι όρος του τύπου A, εκτός από τη συνηθισμένη ερμηνεία της θα έχει επιπλέον μια ομοτοπική ερμηνεία. Συγκεκριμένα, θα το σκεφτόμαστε και ως *το a είναι ένα στοιχείο του χώρου A*. Παρόμοια οι συναρτήσεις $f : A \to B$ θα αντιμετωπίζονται ως συνεχείς συναρτήσεις μεταξύ των χώρων A και B.

Αυτοί οι χώροι όπως είπαμε θα πρέπει να θεωρούνται αμιγώς Ομοτοπικοί και όχι Τοπολογικοί. Με αυτόν τον τρόπο επιτυγχάνουμε μια αφαίρεση και δεν ασχολούμαστε με τις λεπτομέρειες της τοπολογίας, όπως π.χ. αυτές των ανοιχτών συνόλων ή της σύγκλισης. Είναι μια μορφή αφαίρεσης παρόμοια με αυτήν που συμβαίνει στη μελέτη της γεωμετρίας. Όταν μελετάμε Ευκλείδεια Γεωμετρία κάνουμε μια «συνθετική» θεώρηση των εννοιών σε αντίθεση με την Αναλυτική Γεωμετρία όπου έχουμε μια «αναλυτική» θεώρηση με τα σημεία και τις ευθείες να ορίζονται με συγκεκριμένο τρόπο εντός του $\mathbb{R}^3$. Επομένως, Θα ήταν πιο ακριβές να πούμε ότι ερμηνεύουμε τους τύπους ως ∞-groupoids.

Η καίρια νέα ιδέα είναι αυτή της ομοτοπικής ερμηνείας της λογικής έννοιας της ισότητας $a = b$ δύο αντικειμένων $a, b : A$. Στο ομοτοπικό πλαίσιο ένα αντικείμενο ισότητας είναι ένα *μονοπάτι* $p : a \rightsquigarrow b$ από το a στο b εντός του χώρου A. Άρα, δύο συναρτήσεις μπορούν να ταυτιστούν αν υπάρχει μια Ομοτοπία μεταξύ τους, δηλαδή για κάθε $x : A$ ένα μονοπάτι $p_x : f(x) \rightsquigarrow g(x)$ (`funext`).

Ακόμη, ο τύπος $\texttt{Id}_A(a, b)$ της ισότητας μεταξύ δύο αντικειμένων ενός τύπου δεν είναι άλλος από τον χώρο μονοπατιών με άκρα τα a και b στον A. Επίσης, η ισομορφία δύο χώρων, δηλαδή το να υπάρχουν συναρτήσεις από και προς τους δύο τύπους των οποίων οι συνθέσεις είναι σε κάθε σημείο ταυτοτικές, είναι ακριβώς ο ορισμός που δώσαμε για την ομοτοπική ισοδυναμία δύο χώρων.

Σε αυτό το κεφάλαιο θα προσπαθήσουμε να φωτίσουμε την ομοτοπική φύση των τύπων με την αναλύση της ∞-groupoid δομής τους και ύστερα να περιγράψουμε τις δύο νέες ιδέες που επεκτείνουν την Intentional Type Theory στην Homotopy Type Theory, δηλαδή το Univalence Axiom και τους Επαγωγικούς Τύπους ανώτερης διάστασης. Καθώς δεν γίνεται σε μια συνοπτική περιγραφή του πεδίου να συμπεριλάβουμε όλα όσα είναι δυνατόν να αναπτυχθούν σε αυτό το πλαίσιο, θα παρουσιάσουμε επιλεκτικά θέματα με βάση το τι είναι αναγκαίο αναφορικά με το θέμα της παρούσας διπλωματικής εργασίας.

3.1 Higher Groupoids

Έχουμε συζητήσει σχετικά με το λογικό περιεχόμενο της θεωρίας Τύπων και της ιδέας του Props as Types. Σε αυτήν την υποενότητα θα παρουσιάσουμε τη σχέση που υπάρχει μεταξύ της Θεωρίας Τύπων και της Θεωρίας Κατηγοριών, και ιδιαίτερα με την Ομοτοπική Θεωρία Τύπων.

Κατηγορίες Η Θεωρία Κατηγοριών είναι ένας κλάδος των μαθηματικών ο οποίος παρέχει μια ενοποιητική γλώσσα για διάφορους άλλους κλάδους. Προσπαθεί να δημιουργήσει μαθηματικά αντικείμενα τα οποία εκφράζουν κοινά μοτίβα και σχέσεις σε διάφορους κλάδους ταυτόχρονα, κάνοντας ευκολότερη τη μελέτη των ομοιοτήτων των συστημάτων αυτών. Η Θεωρία Κατηγοριών ασχολείται με τη μαθηματική αφαίρεση εν γένει.

Ένας *κατευθυνόμενος γράφος* είναι ένα σύνολο αντικειμένων $\mathcal{O}$, ένα σύνολο βελών ή απεικονίσεων (morphisms) $\mathcal{A}$ και δύο συναρτήσεις

$$\texttt{dom}, \texttt{cod} : \mathcal{A} \to \mathcal{O}$$

Σε αυτόν τον γράφο, το σύνολο των ζευγών βελών που μπορούν να συντεθούν είναι το σύνολο

$$\mathcal{A} \times_{\mathcal{O}} \mathcal{A} :\equiv \{\langle g, f\rangle | g, f \in \mathcal{A} \text{ and } \texttt{dom}g = \texttt{cod}f\}$$

Μια κατηγορία είναι ένας γράφος με δύο επιπλέον συναρτήσεις, την ταυτοτική απεικόνιση για κάθε αντικείμενο

$$\texttt{id} : \mathcal{O} \to \mathcal{A},\ c \mapsto \texttt{id}_c$$

και τη σύνθεση απεικονίσεων

$$\circ : \mathcal{A} \times_{\mathcal{O}} \mathcal{A} \to \mathcal{A},\ \langle g, f\rangle \mapsto g \circ f \text{ ή } gf$$

για τις οποίες έχουμε ότι

$$\texttt{dom}(\texttt{id}_c) = \texttt{cod}(\texttt{id}_c) = c$$

$$\texttt{dom}(gf) = \texttt{dom}(f) \text{ και } \texttt{cod}(gf) = \texttt{cod}(g)$$

Είναι αρκετά συχνό στη Θεωρία Κατηγοριών να αποφεύγουμε τις αυστηρές διατυπώσεις αυτού του τύπου και να χρησιμοποιούμε διαγράμματα με βέλη. Έτσι, για παράδειγμα, για μια απεικόνιση $f \in \mathcal{A}$ με $\texttt{dom}f = a \in \mathcal{O}$ και $\texttt{cod}f = b \in \mathcal{O}$ γράφουμε απλά $a \xrightarrow{f} b$. Επίσης, εγκαταλείπεται συχνά ο συμβολισμός με τα $\mathcal{O}$ και $\mathcal{A}$ και, αν $\mathcal{C}$ είναι μια κατηγορία, απλά γράφουμε $a \in \mathcal{C}$ αντί για $a \in \mathcal{O}$ και $f \in \mathcal{C}$ αντί για $f \in \mathcal{A}$. Επίσης, συμβολίζουμε το σύνολο των βελών μεταξύ δύο αντικειμένων $a, b \in \mathcal{C}$ με $\texttt{Hom}(a, b)$. Οι δύο πράξεις ικανοποιούν επιπλέον δύο αξιώματα.

Μεταβατικότητα Αν $a, b, c, d \in \mathcal{C}$ και $a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$ τότε $h \circ (g \circ f) = (h \circ g) \circ f$.

Νόμοι Μονάδας Αν $a, b, c \in \mathcal{C}$ και $a \xrightarrow{f} b \xrightarrow{g} c$ τότε $\texttt{id}_b \circ f = f$ και $g \circ \texttt{id}_b = g$

Groupoids Στο πλαίσιο των κατηγοριών είναι πολύ εύκολο να αναπαραστήσουμε τη δομή μιας ομάδας με έναν πολύ απλό και φυσικό τρόπο. Μπορούμε να φανταστούμε την ομάδα σαν μια κατηγορία $\mathcal{C}$ με ένα μόνο αντικείμενο $a \in \mathcal{C}$ στην οποία υπάρχει μια αντιστοιχία του συνόλου των απεικονίσεων με τα στοιχεία της ομάδας η οποία σέβεται την πράξη της ομάδας (η οποία ερμηνεύεται ως σύνθεση βελών) και το ταυτοτικό στοιχείο της ομάδας (το οποίο ερμηνεύεται ως $\mathtt{id}_a$). Ένα *groupoid* είναι μια γενίκευση αυτής της ιδέας σε μία κατηγορία με περισσότερα από ένα αντικείμενα.

Πιο συγκεκριμένα, ένα groupoid είναι μια κατηγορία στην οποία ισχύει ότι για κάθε βέλος $f : a \to b$ υπάρχει ένα άλλο βέλος $f^{-1} : b \to a$ για τα οποία ικανοποιούνται οι *Νόμοι των Groupoids*:

$$f \circ f^{-1} = \mathtt{id}_b \text{ και } f^{-1} \circ f = \mathtt{id}_a$$

Η θεωρία Κατηγοριών είναι ικανή να δώσει σημασιολογία και να αποτελέσει μοντέλο για τη Θεωρία Τύπων καθώς διάφορες κατασκευές της έχουν αντίστοιχες στη Θεωρία Κατηγοριών (γινόμενα-συγινόμενα, αρχικά-τελικά αντικείμενα κλπ). Μάλιστα, διαφορετικές εκδοχές της Θεωρίας Τύπων έχουν αντιστοιχηθεί με διαφορετικά είδη κατηγοριών. Στην προκειμένη περίπτωση της ITT είναι πολύ εύκολο να αντιστοιχίσουμε τους τύπους με τα groupoids. Αν θεωρήσουμε ότι κάθε τύπος είναι και μία κατηγορία τότε τα βέλη-απεικονίσεις μεταξύ των αντικειμένων είναι τα αντικείμενα του τύπου ισότητας μεταξύ των αντικειμένων του τύπου, τα μονοπάτια από το ένα στο άλλο δηλαδή. Πράγματι, για κάθε αντικείμενο έχουμε ένα μονοπάτι ισότητας που ξεκινάει και σταματάει στο ίδιο, το $\mathtt{refl}$, και για κάθε δύο μονοπάτια $p : \mathtt{Id}_A(a,b)$ και $q : \mathtt{Id}_A(b,c)$ (τα οποία μπορούν να συνενωθούν) υπάρχει ένα μονοπάτι $p \cdot q : \mathtt{Id}_A(a,c)$. Αποδεικνύεται με επαγωγή μονοπατιού ότι αυτά ικανοποιούν τη μεταβατικότητα και τους νόμους της μονάδας. Σύμφωνα με αυτά, κάθε τύπος αποκτά όντως μια κατηγορική ερμηνεία. Παρόμοια, για κάθε $p : \mathtt{Id}_A(a,b)$ υπάρχει ένα μονοπάτι $p^{-1} : \mathtt{Id}_A(b,a)$ ώστε $p^{-1} \cdot p \equiv \mathtt{refl} : \mathtt{Id}_A(b,b)$ και $p \cdot p^{-1} \equiv \mathtt{refl} : \mathtt{Id}_A(a,a)$, αποδεικνύεται με επαγωγή μονοπατιού. Άρα, κάθε τύπος ερμηνεύεται σαν ένα groupoid.

Θα πρέπει όμως να επανεξετάσουμε την παραπάνω αντιστοιχία και να παρατηρήσουμε κάτι πραγματικά εντυπωσιακό που προκύπτει εξαιτίας της κατασκευαστικής φύσης της Θεωρίας Τύπων και της αντιμετώπισης των αποδείξεων ως πρώτης τάξης αντικείμενα. Όταν λέμε ότι υπάρχει μια απόδειξη με επαγωγή μονοπατιού των ιδιοτήτων των groupoids για τα μονοπάτια σε έναν τύπο, έστω της $\mathtt{refl} \cdot p = p$ (αριστερός νόμος της μονάδας), αυτό σημαίνει ότι υπάρχει μια κατασκευή α που ικανοποιεί τα παρακάτω

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma \vdash a, b : A \quad \Gamma \vdash p : \mathtt{Id}_A(a,b)}{\Gamma \vdash \alpha : \mathtt{Id}_{\mathtt{Id}_A(a,b)}(\mathtt{refl}_A(a) \cdot p, p)}$$

Παρατηρούμε ότι ο τύπος της Ισότητας μπορεί να ορισθεί επαγωγικά για όλο και μεγαλύτερης διάστασης «μονοπάτια» όπως ακριβώς και στην Ομοτοπική Θεωρία. Στην προκειμένη περίπτωση έχουμε ένα μονοπάτι μεταξύ δύο μονοπατιών: των $\mathtt{refl} \cdot p$ και p. Αυτό βέβαια δεν είναι άλλο από μια επιφάνεια στον τύπο/ χώρο A. Δηλαδή ο τύπος της Ισότητας δεν επάγει απλώς μια μονοδιάστατη δομή στον τύπο A, τη δομή ενός groupoid, αλλά μια δομή υψηλότερων διαστάσεων, τη δομή ενός ∞-*groupoid*.

Μια ∞-*κατηγορία* είναι μια γενίκευση της έννοιας της κατηγορίας. Όπως σε κάθε κατηγορία υπάρχουν 1-βέλη μεταξύ αντικειμένων, σε μια 2-κατηγορία έχουμε 1-βέλη μεταξύ αντικειμένων και 2-βελη μεταξύ 1-βελών και σε μια ∞-κατηγορία μπορούμε να έχουμε k-βέλη μεταξύ $(k-1)$-βελών, για κάθε $k = 1, 2, \ldots$. Ένα ∞-groupoid είναι άρα μια ∞- κατηγορία η οποία ικανοποιεί τους νόμους των groupoid σε κάθε διάσταση.

Τέλος, να αναφέρουμε ότι σε αυτήν την κατηγορική προσέγγιση των τύπων οι συναρτήσεις είναι functors. Πράγματι, έχουμε δεί ότι ισχύει ότι

$$\mathtt{ap}_f(\mathtt{refl}_A(x)) = \mathtt{refl}_B(fx)$$

και αποδεικνύεται ότι ισχύουν και οι ισότητες:

$$\mathtt{ap}_f(p^{-1}) = \mathtt{ap}_f(p)^{-1}$$

$$\mathsf{ap}_f(p \cdot q) = \mathsf{ap}_f(p) \cdot \mathsf{ap}_f(q)$$

Όπως πριν όμως, το ότι αυτές οι ισότητες ισχύουν σημαίνει ότι υπάρχει μια κατασκευή-μονοπάτι του κατάλληλου τύπου Ισότητας. Άρα, για να καταστήσουμε αυτό σαφές, συχνά λέμε ότι οι συναρτήσεις σέβονται την groupoid δομή ως προς *ομοτοπία υψηλότερης διάστασης*.

3.2 Univalence Axiom

Σε αυτήν την ενότητα παρουσιάζουμε πώς αντιμετωπίζει η HoTT την ισότητα μεταξύ τύπων, αυτήν την φορά αντί της ισότητας αντικειμένων εντός ενός τύπου. Πρώτα, δίνουμε τον ορισμό της ισοδυναμίας τύπων που χρησιμοποιείται στην HoTT, την *Ομοτοπική Ισοδυναμία*.

Ομοτοπική Ισοδυναμία Αρχικά, εισάγουμε έναν συμβολισμό για τον τύπο των *ομοτοπιών* μεταξύ δύο συναρτήσεων $f, g : A \to B$. Ορίζουμε $f \simeq g :\equiv \Pi(a : A).fa =_B ga$. Αποδεικνύεται εύκολα ότι αυτό είναι μια σχέση ισοδυναμίας, η οποία ικανοποιεί την πολύ σημαντική ιδιότητα της *φυσικότητας (naturality)*: αν $H : f \simeq g$ τότε η ομοτοπία H σέβεται τα μονοπάτια μεταξύ στοιχείων του A, είναι *dependently functorial στα στοιχεία* $x : A$. Αυτό σημαίνει ότι το παρακάτω διάγραμμα είναι μεταθετικό

$$\begin{tikzcd} f(a) \arrow[r, Rightarrow, "H(a)"] \arrow[d, Rightarrow, "\mathsf{ap}_f(p)"] & g(a) \arrow[d, Rightarrow, "\mathsf{ap}_g(p)"] \\ f(a') \arrow[r, Rightarrow, "H(a')"] & g(a') \end{tikzcd}$$

Τώρα, για τον ορισμό της ισοδυναμίας δύο τύπων $A, B : \mathcal{U}$ ορίζουμε έναν τύπο $A \simeq B$ που αποτελείται από τη συνάρτηση $f : A \to B$ μαζί με τα εξής δεδομένα-data:

- Μια συνάρτηση $g : B \to A$
- Μια Ομοτοπία $g \circ f \simeq \mathtt{id}_A$, δηλαδή μια απόδειξη του $\alpha : \Pi(a : A).g(f(a)) =_A a$
- Μια συνάρτηση $h : B \to A$
- Μια Ομοτοπία $f \circ h \simeq \mathtt{id}_B$, δηλαδή μια απόδειξη του $\beta : \Pi(b : B).f(h(b)) =_B b$

Άρα, ο τύπος ισοδυναμίας των τύπων A και B ορίζεται ως $A \simeq B :\equiv \Sigma(f : A \to B).\mathsf{isEquiv}(f)$ όπου

$$\mathsf{isEquiv}(f) :\equiv (\Sigma(g : B \to A).g \circ f \simeq \mathtt{id}_A) \times (\Sigma(h : B \to A).f \circ h \simeq \mathtt{id}_B)$$

Με τον τρόπο που ορίστηκε η ισοδυναμία αντιλαμβανόμαστε ότι αν δύο τύποι είναι ισοδύναμοι, τότε έχουν την ίδια δομή μονοπατιών σε όλες τις διαστάσεις.

Univalence Axiom Αν έχουμε δύο τύπους $A, B : \mathcal{U}$, τότε μπορούμε να ορίσουμε τον τύπο $\mathtt{Id}_{\mathcal{U}}(A, B)$ όλων των μονοπατιών στο σύμπαν μεταξύ των A, B. Με πολύ φυσικό τρόπο γίνεται να ορίσουμε μια συνάρτηση

$$\mathtt{idtoeqv} : (A =_{\mathcal{U}} B) \to (A \simeq B)$$

η οποία δίνει μια συνάρτηση $f : A \to B$ που μεταφέρει πάνω από ένα μονοπάτι ισότητας στο $\mathcal{U}$ στοιχεία του A στον B. Θα θέλαμε η `idtoeqv` να ήταν ισοδυναμία. Αυτό ακριβώς μας δίνει το *Univalence Axiom*, το οποίο οφείλεται στον Vladimir Voevodsky. Πιο συγκεκριμένα, για κάθε $A, B : \mathcal{U}$ έχουμε ότι

$$(A \simeq B) \simeq (A =_{\mathcal{U}} B)$$

Με βάση το αξίωμα αυτό μπορούμε να *ταυτίζουμε ισοδύναμους τύπους*. Μπορούμε άρα να αναλύσουμε αυτήν την ισοδυναμία στα μέρη που τη συνθέτουν:

- Μια συνάρτηση εισαγωγής για τον τύπο $A =_{\mathcal{U}} B$:

$$\mathtt{ua} : (A \simeq B) \to (A =_{\mathcal{U}} B)$$

- Μια συνάρτηση απαλοιφής:

$$\mathtt{idtoeqv} : (A =_{\mathcal{U}} B) \to (A \simeq B)$$

- Έναν κανόνα υπολογισμού:

$$\mathtt{tr}[X.X](\mathtt{ua}(f))(x) = fx$$

- Έναν κανόνα μοναδικότητας για κάθε $p : A =_{\mathcal{U}} B$:

$$p = \mathtt{ua}(\mathtt{tr}[X.X](p))$$

3.3 Επαγωγικοί Τύποι

Σε αυτήν την υποενότητα θα δούμε το τελευταίο μέρος των κατασκευών της ITT, τους επαγωγικούς τύπους. Οι επαγωγικοί τύποι επεκτείνουν την ιδέα των τύπων που υποστηρίζουν αναδρομή και επαγωγή, όπως ακριβώς και οι φυσικοί αριθμοί, σε ένα τελείως γενικό πλαίσιο. Διαισθητικά, για τον ορισμό ενός επαγωγικού τύπου X απαιτείται να δοθεί ένα πεπερασμένο πλήθος κατασκευαστών, συναρτήσεων με πεδίο τιμών τον X, και τα αντικείμενα του X μπορούμε να θεωρήσουμε ότι παράγονται ελεύθερα από αυτούς τους κατασκευαστές. Οι κατασκευαστές μπορούν να έχουν πολλά ορίσματα (ή και κανένα, δηλαδή απλά δηλώνουν την ύπαρξη ενός στοιχείου του X) και από τον ίδιο τον τύπο X.

Πολλοί από τους τύπους που είδαμε μπορούν να οριστούν με επαγωγικό τρόπο. Για παράδειγμα, ο τύπος `bool` μπορεί να δοθεί με δύο κατασκευαστές με κανένα όρισμα

- $\mathtt{tt} : \mathtt{bool}$
- $\mathtt{ff} : \mathtt{bool}$

Δύο παραδείγματα όπου οι κατασκευαστές παίρνουν ορίσματα είναι το συγινόμενο δύο τύπων A, B

- $\mathtt{inl} : A \to A + B$
- $\mathtt{inr} : B \to A + B$

και το γινόμενο

- $\langle -, - \rangle : A \to B \to A \times B$

Ενώ, ένα παράδειγμα όπου ο κατασκευαστής παίρνει ορίσματα από τον οριζόμενο τύπο είναι οι Φυσικοί Αριθμοί όταν οριστούν ως επαγωγικός τύπος

- $\mathtt{zero} : \mathtt{nat}$
- $\mathtt{suc} : \mathtt{nat} \to \mathtt{nat}$

Ο τρόπος με τον οποίο κατασκευάζουμε συναρτήσεις $f : X \to P$ όπου X ένας επαγωγικός τύπος είναι μια αρχή αναδρομής. Για κάθε κατασκευαστή και για οποιαδήποτε ορίσματά του δίνουμε ένα αντικείμενο του P, επιτρέποντάς μας να καλούμε την f αναδρομικά για τα ορίσματα του κατασκευαστή. Για παράδειγμα, για να ορίσουμε μια συνάρτηση από τον τύπο `bool` προς οποιονδήποτε τύπο P αρκεί να δώσουμε δύο αντικείμενα $e_{\mathtt{tt}}, e_{\mathtt{ff}} : P$ και έτσι έχουμε ορίσει μια συνάρτηση $f = \mathtt{rec}_{\mathtt{bool}}(e_{\mathtt{tt}}, e_{\mathtt{ff}}) : \mathtt{bool} \to P$ η οποία επιπλέον ικανοποιεί τις εξής ισότητες:

$$\mathtt{rec}_{\mathtt{bool}}(e_{\mathtt{tt}}, e_{\mathtt{ff}})(\mathtt{tt}) \equiv e_{\mathtt{tt}} : P \qquad \mathtt{rec}_{\mathtt{bool}}(e_{\mathtt{tt}}, e_{\mathtt{ff}})(\mathtt{ff}) \equiv e_{\mathtt{ff}} : P$$

Για την περίπτωση όπου έχουμε έναν κατασκευαστή με ορίσματα θα πρέπει να τα λάβουμε υπόψη μας και να δώσουμε ένα στοιχείο για κάθε περίπτωση ορίσματος, αλλά και το αποτέλεσμα της f όταν δέχεται σαν όρισμα ένα από τα ορίσματα του κατασκευαστή. Αν για παράδειγμα ορίσουμε επαγωγικά τον τύπο των συνδεδεμένων λιστών των στοιχείων με στοιχεία από εναν τύπο A ως εξής:

- $\texttt{nil} : \texttt{list}(A)$
- $\texttt{cons} : A \to \texttt{list}(A) \to \texttt{list}(A)$

τότε για να ορίσουμε μια συνάρτηση $f : \texttt{list}(A) \to P$ θα πρέπει να βρούμε $e_{\texttt{nil}} : P$ και $e_{\texttt{cons}} : A \to \texttt{list}(A) \to P \to P$. Δηλαδή θα πρέπει να καθορίσουμε το αποτέλεσμα της συνάρτησης για την κενή λίστα $\texttt{nil}$ και για τη λίστα $\texttt{cons}(\texttt{x}, \texttt{xs})$ αν γνωρίζουμε το αποτέλεσμα $f(\texttt{xs})$. Σε αυτήν την περίπτωση έχουμε ότι $f(\texttt{nil}) \equiv e_{\texttt{nil}}$ και $f(\texttt{cons}(\texttt{x}, \texttt{xs})) \equiv e_{\texttt{cons}}(\texttt{x}, \texttt{xs}, f(\texttt{xs}))$.

Με παρόμοιο τρόπο μπορούμε να ορίσουμε εξαρτημένες συναρτήσεις, κάνοντας δηλαδή αυτήν την φορά επαγωγή. Για να ορίσουμε μια συνάρτηση $f : \Pi x : X.B$ όπου $B : X \to \mathcal{U}$ μια οικογένεια τύπων και X ένας επαγωγικός τύπος δίνουμε για κάθε κατασκευαστή, για κάθε όρισμά του και για κάθε πιθανή τιμή της f σε ορίσματα του κατασκευαστή μια τιμή του $B[x]$. Αν θεωρήσουμε ότι έχουμε ορίσει τους φυσικούς αριθμούς επαγωγικά και έχουμε μια οικογένεια $E : \texttt{nat} \to \mathcal{U}$ τότε για να ορίσουμε μια συνάρτηση $f : \Pi n : \texttt{nat}.E$ απαιτείται να έχουμε ένα στοιχείο $e_{\texttt{zero}} : E(\texttt{zero})$ για τον κατασκευαστή $\texttt{zero}$ και ένα στοιχείο $e_{\texttt{suc}} : \Pi n : \texttt{nat}.E(n) \to E(\texttt{suc}(n))$ για τον κατασκευαστή $\texttt{suc}$. Τότε η f ικανοποιεί τις εξής ισότητες:

$$f(\texttt{zero}) \equiv e_{\texttt{zero}} : E(\texttt{zero}) \qquad f(\texttt{suc}(n)) \equiv e_{\texttt{suc}}(n, f(n)) : E(\texttt{suc}(n))$$

Η παρουσίαση που κάναμε για τους επαγωγικούς τύπους είναι πιο πολύ διαισθητική παρά τυπική και πλήρης σε αντίθεση με τους προηγούμενους τύπους της ITT. Για παράδειγμα απουσιάζει η τυπική παρουσίαση των W-τύπων, αποδείξεις μοναδικότητας, κατηγορικά semantics όπως F-άλγεβρες, συνεπαγωγικοί τύποι κλπ. Ο λόγος για αυτό είναι το ότι κάτι τέτοιο θα ήταν μακροσκελές κι επίσης το θέμα των επαγωγικών τύπων δεν σχετίζεται άμεσα με το θέμα της εργασίας. Παρ'όλα αυτά, έγινε μια σύντομη αναφορά για να υπάρχει ομαλή συνδεση με το θέμα των επαγωγικών τύπων υψηλότερων διαστάσεων της HoTT και για πληρότητα. Για μια πιο ολοκληρωμένη ανάλυση παραπέμπουμε στο 5ο κεφάλαιο του [Univ13].

3.4 Υψηλοί Επαγωγικοί Τύποι

Σε αυτήν την ενότητα θα παρουσιάσουμε τη δεύτερη προσθήκη στην Ομοτοπική Θεωρία Τύπων σε σχέση με άλλες θεωρίες, τους Επαγωγικούς Τύπους υψηλότερων διαστάσεων. Οι επαγωγικοί τύποι υψηλότερων διαστάσεων είναι ένα γενικό σχήμα ορισμού νέων τύπων με βάση κάποιους κατασκευαστές. Αντίθετα, όμως, από τους απλούς επαγωγικούς τύπους, στους ορισμούς τους μπορούμε να προσθέσουμε κατασκευαστές μονοπατιών οποιασδήποτε διάστασης. Δηλαδή εκτός από τα σημεία του χώρου μπορούμε να επιβάλουμε και την ύπαρξη σχέσεων ισότητας μεταξύ τους, οι οποίες στο proof relevance περιβάλλον της θεωρίας τύπων είναι συγκεκριμένα αντικείμενα, τα μονοπάτια. Θα μπορούσε κανείς να σκεφτεί τους τύπους αυτούς ως ελεύθερες αλγεβρικές κατασκευές για τον ορισμό των οποίων δίνουμε τους γεννήτορες που παράγουν τα στοιχεία κι επίσης τις σχέσεις που ικανοποιούν αυτά.

Συνήθως, για τον ορισμό τέτοιων τύπων χρησιμοποιούμε την ορολογία 0-cell για τα αντικείμενα του χώρου, 1-cell για τα μονοπάτια μεταξύ των αντικειμένων, 2-cell για τα μονοπάτια μεταξύ των 1-cells (τις επιφάνειες του χώρου), κοκ. Άρα, για να ορίσουμε έναν τέτοιο τύπο παραθέτουμε σε μια λίστα τους κατασκευαστές των cells κάθε διάστασης. Για παράδειγμα, για τον ορισμό του επαγωγικού τύπου υψηλότερων διαστάσεων του κύκλου $\mathbb{S}^1$ πρέπει να ορίσουμε ένα σημείο του κύκλου, το $\texttt{base}$, ως 0-cell και ένα μονοπάτι το οποίο πηγαίνει από το $\texttt{base}$ στο $\texttt{base}$, ως 1-cell. Άρα, γράφουμε

- $\texttt{base} : \mathbb{S}^1$
- $\texttt{loop} : \texttt{base} =_{\mathbb{S}^1} \texttt{base}$

Με αυτόν τον ορισμό τοποθετήσαμε ένα νέο αντικείμενο και στον τύπο $\texttt{Id}_{\mathbb{S}^1}(\texttt{base}, \texttt{base})$ το οποίο δεν είναι απαραίτητα ίσο με κάποιο άλλο. Για την ακρίβεια, αποδεικνύεται ότι το μονοπάτι $\texttt{loop}$ δεν είναι ίσο με το μονοπάτι $\texttt{refl}(\texttt{base})$ και άρα ο τύπος $\mathbb{S}^1$ δεν είναι ίσος με τον τύπο **1**. Ακόμα, μια σημαντική διαφορά από τους απλούς επαγωγικούς τύπους έρχεται εξαιτίας της εγγενούς groupoid δομής που έχουν όλοι οι τύποι λόγω των τύπων ισότητας. Όταν άμεσα δημιουργούμε ένα μονοπάτι $\mathtt{loop}$ μέσω του ορισμού τότε δημιουργούμε έμμεσα και διάφορα άλλα μονοπάτια, όπως τα $\mathtt{loop} \cdot \mathtt{loop}$, $\mathtt{loop} \cdot \mathtt{loop} \cdot \mathtt{loop}$, $\mathtt{loop}^{-1}$,... τα οποία όλα είναι διαφορετικά μεταξύ τους.

Επίσης, να σημειώσουμε ότι επειδή οι κατασκευαστές σημείων και μονοπατιών του τύπου είναι συναρτήσεις και άρα όπως έχουμε δει functors, μεταφέρουν και σέβονται τη δομή του πεδίου ορισμού τους σε κάθε διάσταση. Αν ένας τύπος B ορίζεται επαγωγικά με κατασκευαστή $A \to B$, τότε μονοπάτια μεταξύ στοιχείων του A συνεπάγονται την ύπαρξη μονοπατιών μεταξύ των στοιχείων που δημιουργεί ο συγκεκριμένος κατασκευαστής στον B. Το ίδιο ισχύει και σε υψηλότερες διαστάσεις. Αν έχουμε έναν κατασκευαστή μονοπατιού $A \to x =_B y$, τότε τα μονοπάτια στον A μεταφέρονται μέσω του κατασκευαστή σε μονοπάτια στον $x =_B y$ και άρα σε 2-μονοπάτια στον B.

Είναι πραγματικά εντυπωσιακό το πόσο απλά ορίζονται χώροι με αυτά τα εργαλεία. Στην περίπτωση του ορισμού του κύκλου έχουμε απαλείψει όλες τις γεωμετρικές και τοπολογικές ιδιότητες που έχει με τον συνήθη ορισμό του ως υπόχωρο του $\mathbb{R}^2$ ή του $\mathbb{C}$ και έχουμε επικεντρωθεί μόνο στα ομοτοπικά ή κατηγορικά του χαρακτηριστικά. Αυτή η αφαίρεση μας επιτρέπει να κάνουμε πιο εύκολη τη μελέτη αυτών των αντικειμένων εστιάζοντας μόνο στις *καθολικές* τους ιδιότητες (*universal properties*), οι οποίες εκφράζονται μέσω των αρχών αναδρομής και επαγωγής των τύπων αυτών.

Ανδρομή και Επαγωγή Παρουσιάσαμε τον τρόπο με τον οποίο γίνεται να ορίσουμε επαγωγικούς τύπους με υψηλότερη δομή groupoid, δώσαμε δηλαδή τους κανόνες εισαγωγής των στοιχείων τους. Τώρα, πρέπει να δώσουμε και κανόνες απαλοιφής, δηλαδή με ποιον τρόπο γίνεται να χρησιμοποιήσουμε τα αντικείμενα αυτών των τύπων. Οι κανόνες αυτοί όπως και στην περίπτωση των απλών επαγωγικών τύπων είναι τα γνωστά σχήματα αναδρομής και επαγωγής.

Για να ορίσουμε σωστά συναρτήσεις πάνω σε επαγωγικούς τύπους χωρίς υψηλότερη διάσταση δίναμε αναλυτικά για κάθε κατασκευαστή και για κάθε είδους πιθανά ορίσματά του το αποτέλεσμα που θα θέλαμε να είχε η συνάρτηση στο πεδίο τιμών. Αυτό ακριβώς θα κάνουμε και τώρα λαμβάνοντας, όμως, υπόψη μας και τους κατασκευαστές μονοπατιών. Θα πρέπει αφού καθορίσουμε τη συμπεριφορά της οριζόμενης συνάρτησης πάνω στα 0-cells, να καθορίσουμε και τη συμπεριφορά της πάνω στα 1-cells ώστε να υπάρχει συνοχή με τα προηγούμενα σημεία κοκ για τις μεγαλύτερες διαστάσεις. Θα πρέπει κατά κάποιον τρόπο να ταυτίσουμε την εικόνα της συνάρτησής μας με τα στοιχεία του πεδίου τιμών με παρόμοια δομή με αυτήν του επαγωγικού τύπου υψηλότερης διάστασης. Πιο συγκεκριμένα, αφού δώσουμε τις τιμές της συνάρτησης που θέλουμε να ορίσουμε $f : A \to B$ στα 0-cells του A σύμφωνα με τους κατασκευαστές και ό,τι έχουμε δει, θα πρέπει να καθορίσουμε τον τρόπο με τον οποίο η f μεταφέρει τα 1-cells μεταξύ των σημείων που ήδη ορίσθηκαν, να δώσουμε δηλαδή την τιμή της συνάρτησης ap_f και dap_f για την περίπτωση εξαρτημένης συνάρτησης για τα σημεία αυτά. Και συνεχίζουμε σε όλες τις διαστάσεις για όλους τους κατασκευαστές μονοπατιών.

Για παράδειγμα έστω ο τύπος A ο οποίος ορίζεται ως εξής:

- $a_{-2}, a_{-1} : A$
- $a : \mathsf{nat} \to A$
- $p_{-1} : a(\mathsf{zero}) =_A a_{-1}$
- $p_{-2} : a(\mathsf{zero}) =_A a_{-2}$
- $p : \Pi n : \mathsf{nat}.a(\mathsf{zero}) =_A a(n)$

Τότε για να ορίσουμε μια συνάρτηση αναδρομικά με πεδίο ορισμού των A θα πρέπει να δώσουμε τις εξής πληροφορίες

- $f(a_{-2}) : B$
  $f(a_{-1}) : B$
  δύο σημεία του B για τους δύο πρώτους κατασκευαστές

- $n : \mathtt{nat} \vdash f(a(n)) : B$
  ένα σημείο για κάθε φυσικό αριθμό

- $\mathtt{ap}_f(p_{-2}) : f(a(\mathtt{zero})) =_B f(a_{-2})$
  $\mathtt{ap}_f(p_{-1}) : f(a(\mathtt{zero})) =_B f(a_{-1})$
  το πώς μεταφέρει η f τα μονοπάτια p_{-2}, p_{-1} σεβόμενη τα άκρα τους

- $n : \mathtt{nat} \vdash \mathtt{ap}_f(p(n)) : f(a(\mathtt{zero})) =_B f(a(n))$
  το πώς μεταφέρει η f τα μονοπάτια για τον κατασκευαστή p

Στην περίπτωση της επαγωγής λειτουργούμε με παρόμοιο τρόπο. Αν δηλαδή ο $B : A \to \mathcal{U}$ είναι μια οικογένεια τύπων, τότε αφού τα αντικείμενα που θα ορίσουμε ως αποτέλεσμα της f στα 0-cells του A θα ανήκουν σε διαφορετικούς τύπους, τότε θα πρέπει τα μονοπάτια στα οποία θα μεταφέρει η f (τα 1-cells, 2-cells,...) να είναι μονοπάτια πάνω από μονοπάτια (εξαρτημένη περίπτωση). Στο πιο πάνω παράδειγμα θα είχαμε ακριβώς τον ίδιο τρόπο ορισμού της f με την αλλαγή των μονοπατιών σε μονοπάτια πάνω από μονοπάτια.

- $f(a_{-2}) : B(a_{-2})$
  $f(a_{-1}) : B(a_{-1})$

- $f(a(\mathtt{zero})) : B(a(\mathtt{zero}))$
  $n : \mathtt{nat}, b : B(a(n)) \vdash f(a(\mathtt{suc}(n))) : B(a(\mathtt{suc}(n)))$

- $\mathtt{dap}_f(p_{-2}) : f(a(\mathtt{zero})) =^{x.B}_{p_{-2}} f(a_{-2})$
  $\mathtt{dap}_f(p_{-1}) : f(a(\mathtt{zero})) =^{x.B}_{p_{-1}} f(a_{-1})$

- $n : \mathtt{nat} \vdash \mathtt{dap}_f(p(n)) : f(a(\mathtt{zero})) =^{x.B}_{p(n)} f(a(n))$

Στις επόμενες υποενότητες δίνουμε παραδείγματα ορισμού επαγωγικών τύπων υψηλότερων διαστάσεων τα οποία είναι σημαντικά για τη συνέχεια της εργασίας αλλά και για την κατανόηση του τρόπου ορισμού των τύπων αυτών και των εννοιών της αναδρομής και της επαγωγής.

3.4.1 Το διάστημα

Το διάστημα, το οποίο συμβολίζουμε ως $\mathbb{I}$, είναι ένας από τους πιο απλούς επαγωγικούς τύπους υψηλότερων διαστάσεων. Ορίζεται ως εξής:

- $0, 1 : \mathbb{I}$

- $\mathtt{seg} : 0 =_{\mathbb{I}} 1$

Το διάστημα αποτελεί μια αφαίρεση του τοπολογικού διαστήματος $[0, 1]$ υποσυνόλου του $\mathbb{R}$. Για την αναδρομή αρκεί να δοθούν τα εξής «data» του πεδίου τιμών A, δύο σημεία a_0, a_0 και ένα μονοπάτι μεταξύ τους.

$$\frac{a_0 : A \quad a_1 : A \quad q : a_0 =_A a_1}{z : \mathbb{I} \vdash \mathtt{rec}[A](a_0, a_1, q)(z) : A}$$

Τότε, η οριζόμενη συνάρτηση ικανοποιεί τις εξής ιδιότητες:

$$\begin{aligned}
\mathtt{rec}[A](a_0, a_1, q)(0) &\equiv a_0 \\
\mathtt{rec}[A](a_0, a_1, q)(1) &\equiv a_1 \\
\mathtt{ap}_{\mathtt{rec}[A](a_0, a_1, q)}(\mathtt{seg}) &= q
\end{aligned}$$

Όμοια, για την περίπτωση της επαγωγής αλλά στο εξαρτημένο πλαίσιο.

$$\frac{z : \mathbb{I} \vdash A : \mathcal{U} \quad a_0 : A(0) \quad a_1 : A(1) \quad q : a_0 =^{z.A}_{\mathtt{seg}} a_1}{z : \mathbb{I} \vdash \mathtt{ind}[z.A](a_0, a_1, q)(z) : A(z)}$$

Τα παραπάνω δεδομένα αρκούν για τον ορισμό μιας συνάρτησης η οποία ικανοποιεί, όπως και πριν, τα παρακάτω:

$$\begin{aligned}\texttt{ind}[z.A](a_0, a_1, q)(0) &\equiv a_0 \\ \texttt{ind}[z.A](a_0, a_1, q)(1) &\equiv a_1 \\ \texttt{dap}_{\texttt{ind}[z.A](a_0,a_1,q)}(\texttt{seg}) &= q\end{aligned}$$

Αποδεικνύεται ότι το διάστημα έχει μια πολύ σημαντική ιδιότητα. Ισχύει η εξής ισοδυναμία τύπων

$$\mathbb{I} \to A \simeq \Sigma x, y : A.\texttt{Id}_A(x, y)$$

Δηλαδή, ο τύπος των συναρτήσεων από το διάστημα σε οποιονδήποτε τύπο A είναι ισοδύναμος με τον τύπο όλων των μονοπατιών που υπάρχουν στον A με οποιαδήποτε άκρα. Αυτό είναι διαισθητικά προφανές αν θεωρήσουμε την τοπολογική ερμηνεία των μονοπατιών ως συνεχείς συναρτήσεις από το κλειστό διάστημα $[0, 1]$ στον χώρο A.

3.4.2 Κύκλοι και Σφαίρες

Κύκλος Σε αυτό το σημείο θα δούμε με μεγαλύτερη λεπτομέρεια τον τύπο του κύκλου. Ο κύκλος ορίζεται όπως έχουμε δει με την παράθεση ενός σημείου (0-cell) και ενός μονοπατιού με αρχή και τέλος αυτό το σημείο (1-cell).

- $\texttt{base} : \mathbb{S}^1$
- $\texttt{loop} : \texttt{base} =_{\mathbb{S}^1} \texttt{base}$

Οι κανόνες για την αναδρομή και την επαγωγή είναι οι αναμενόμενοι. Για να ορίσουμε μια συνάρτηση αναδρομικά αρκεί να δώσουμε ένα στοιχείο και ένα loop με αρχή και τέλος το στοιχείο αυτό.

$$\frac{a : A \quad \ell : a =_A a}{z : \mathbb{S}^1 \vdash \texttt{rec}[A](a, \ell)(z) : A}$$

Η συνάρτηση που προκύπτει ικανοποιεί τα παρακάτω:

$$\begin{aligned}\texttt{rec}[A](a, \ell)(\texttt{base}) &\equiv a \\ \texttt{ap}_{\texttt{rec}[A](a,\ell)}(\texttt{loop}) &= \ell\end{aligned}$$

Ενώ για την επαγωγή

$$\frac{z : \mathbb{S}^1 \vdash A : \mathcal{U} \quad a : A(\texttt{base}) \quad \ell : a =^{z.A}_{\texttt{loop}} a}{z : \mathbb{S}^1 \vdash \texttt{ind}[z.A](a, \ell)(z) : A(z)}$$

με

$$\begin{aligned}\texttt{ind}[z.A](a, \ell)(\texttt{base}) &\equiv a \\ \texttt{dap}_{\texttt{ind}[z.A](a,\ell)}(\texttt{loop}) &= \ell\end{aligned}$$

Όμοια με τον τύπο του διαστήματος ο τύπος του κύκλου ικανοποιεί μια παρόμοια ισοδυναμία για κάθε τύπο A

$$\mathbb{S}^1 \to A \simeq \Sigma x : A.\texttt{Id}_A(x, x)$$

Δηλαδή -όπως πριν- ο τύπος όλων των πιθανών μονοπατιών με ίδια αρχή και τέλος είναι ισοδύναμος με τον τύπο των συναρτήσεων από τον κύκλο στον χώρο A. Όπως πριν η ομοιότητα με τον τρόπο με τον οποίο ορίζονται στην τοπολογία τα loops σε χώρους είναι προφανής.

Σφαίρες Για να γίνει σαφές το πώς μπορούμε να ορίσουμε τύπους με δομή μεγαλύτερων διαστάσεων από ένα, δίνουμε τα παραδείγματα των σφαιρών μεγαλύτερων διαστάσεων από τον κύκλο.

Για να ορίσουμε τη $\mathbb{S}^2$ δίνουμε ένα σημείο της σφαίρας ($\mathtt{base}$) και έναν κατασκευαστή μονοπατιού δεύτερης διάστασης, με άκρα το αυτοπαθές μονοπάτι $\mathtt{refl}(\mathtt{base})$

- $\mathtt{base} : \mathbb{S}^2$
- $\mathtt{surf} : \mathtt{refl}(\mathtt{base}) =_{\mathtt{base}=\mathtt{base}} \mathtt{refl}(\mathtt{base})$

Για να ορίσουμε αναδρομικά συναρτήσεις από τη δισδιάστατη σφαίρα προς οποιονδήποτε τύπο A πρέπει να ορίσουμε πού μεταφέρει η συνάρτηση το σημείο $\mathtt{base}$ και πού μεταφέρει η συνάρτηση την επιφάνεια $\mathtt{surf}$, δηλαδή να ορίσουμε μια επιφάνεια στον A η οποία να είναι η τιμή της συνάρτησης $\mathtt{ap}^2_f(\mathtt{surf})$. Η συνάρτηση $\mathtt{ap}^2$ είναι η απόδειξη ότι οι συναρτήσεις σέβονται τα μονοπάτια δεύτερης διάστασης και ορίζεται εύκολα παρόμοια με την $\mathtt{ap}$ ως εξής:

$$\Gamma, f : A \to B, x, y : A, p, q : x =_A y, s : p =_{x=y} q \vdash \mathtt{ap}^2_f(s) : \mathtt{ap}_f(p) =_{f(x)=f(y)} \mathtt{ap}_f(q)$$

με

$$\mathtt{ap}^2_f(s) \equiv \mathtt{J}[p', q', s'.\mathtt{Id}_{\mathtt{Id}_B(f(x),f(y))}(\mathtt{ap}_f(p'), \mathtt{ap}_f(q'))](z.\mathtt{refl}(\mathtt{ap}_f(z)))(s)$$

Άρα, για τον ορισμό μιας συνάρτησης $f : \mathbb{S}^2 \to B$ απαιτούνται ένα σημείο $b : B$ και μια επιφάνεια $s : \mathtt{refl}(b) = \mathtt{refl}(b)$. Τότε, έχουμε τον εξής κανόνα και ισότητες:

$$\frac{b : B \quad s : \mathtt{refl}(b) =_{b=_B b} \mathtt{refl}(b)}{\Gamma, z : \mathbb{S}^2 \vdash \mathtt{rec}[B](b, s)(z) : B}$$

$$\mathtt{rec}[B](b, s)(\mathtt{base}) \equiv b$$
$$\mathtt{ap}^2_{\mathtt{rec}[B](b,s)}(\mathtt{surf}) = s$$

Με παρόμοιο τρόπο γίνεται και η επαγωγή.

3.4.3 Truncations

Σε αυτήν την υποενότητα εισαγάγουμε έναν τρόπο ελέγχου της δομής ανώτερων διαστάσεων ενός τύπου με την έννοια των truncations. Ο τρόπος με τον οποίο θα τα παρουσιάσουμε και θα μας χρειαστούν είναι οριζόμενα ως επαγωγικοί τύποι υψηλότερων διαστάσεων.

Ομοτοπικοί Τύποι Ένας τρόπος με τον οποίο περιγράφεται η δομή ορισμένων χώρων στην κλασική ομοτοπική θεωρία είναι με την κατηγοριοποίησή τους σε Ομοτοπικούς Τύπους. Οι Ομοτοπικοί Τύποι μάς δείχνουν τον τρόπο με τον οποίο είναι συνδεδεμένα τα στοιχεία ενός χώρου μεταξύ τους με ομοτοπίες μεγαλύτερης τάξης, δηλαδή με μονοπάτια υψηλότερων διαστάσεων. Στην HoTT αυτές οι κατηγορίες τύπων ονομάζονται h-types ή n-types. Υπάρχει δηλαδή μια αθροιστική ιεραρχία πολυπλοκότητας δομής στους τύπους η οποία ξεκινάει από το -2 και αυξάνει επ' άπειρον σε όλο και πιο σύνθετες δομές σε μεγαλύτερες διαστάσεις. Ο λόγος που η ιεραρχία ξεκινάει από το -2 και όχι από το 0 είναι για να συμβαδίσει με παρόμοιες έννοιες από άλλους κλάδους των μαθηματικών.

Η βάση της ιεραρχίας είναι οι -2-types ή αλλιώς contractible types. Πρόκειται για μια περίπτωση subsingleton, δηλαδή οι τύποι οι οποίοι είναι contractible έχουν τουλάχιστον ένα στοιχείο x_0 και για κάθε άλλο στοιχείο τους υπάρχει ένα μονοπάτι από το x_0 προς το στοιχείο. Πρόκειται για χώρους πλήρως συνδεδεμένους οι οποίοι έχουν ένα τουλάχιστον στοιχείο. Υπό το πρίσμα του Props As Types οι τύποι αυτοί εκφράζουν τις αληθείς προτάσεις των οποίων οι μάρτηρες αλήθειας δεν μας ενδιαφέρουν αφού είναι όλοι ίσοι μεταξύ τους. Στην HoTT αυτό κωδικοποιείται με τον παρακάτω ορισμό:

$$\mathtt{isContr}(A) :\equiv \Sigma a : A. \Pi x : A. a =_A x$$

Σε δεύτερο επίπεδο έχουμε τους -1-types ή αλλιώς h-Props. Σε αυτήν την περίπτωση, αν υπάρχουν στοιχεία του τύπου, τότε αυτά είναι όλα ίδια μεταξύ τους. Δηλαδή

$$\texttt{isProp}(A) :\equiv \Pi x, y : A. x =_A y$$

Αυτό αποδεικνύεται ότι είναι ισοδύναμο με τον ορισμό

$$\texttt{isProp}(A) \simeq \Pi x, y : A. \texttt{isContr}(x =_A y)$$

Σε αυτούς τους τύπους αντιστοιχούν οι προτάσεις της ιντουισιονιστικής λογικής που είδαμε και στην αρχή του Props as Types. Είναι τύποι οι οποίοι είτε είναι κενοί, ψευδείς προτάσεις, είτε είναι αληθείς και αυτό και μόνο είναι η οποιαδήποτε πληροφορία παρέχουν, όλα τα στοιχεία τους είναι ίσα.

Το επόμενο επίπεδο στην ιεραρχία είναι οι τύποι 0-types ή hSets. Πρόκειται για τους τύπους οι οποίοι δεν έχουν δομή υψηλότερων από 0 διαστάσεων και άρα θα μπορούσε να πει κανείς ότι συμπεριφέρονται όπως τα κλασικά σύνολα. Ο επίσημος ορισμός δηλώνει ότι δύο μονοπάτια μεταξύ δύο αντικειμένων του τύπου, αν υπάρχουν, είναι κι αυτά ίσα

$$\texttt{isSet}(A) :\equiv \Pi x, y : A. \Pi p, q : x =_A y. p =_{x =_A y} q$$

Πάλι ο ισοδύναμος ορισμός είναι όπως και πριν

$$\texttt{isSet}(a) \simeq \Pi x, y : A. \texttt{isProp}(x =_A y)$$

δηλαδή τα μονοπάτια μεταξύ δύο οποιωνδήποτε αντικειμένων είναι τύπος ακριβώς μικρότερης πολυπλοκότητας στη δομή μονοπατιών.

Συνεχίζοντας επαγωγικά μπορούμε να ορίσουμε τους n-types με ακριβώς αυτήν τη μέθοδο: ένας τύπος είναι n-type αν και μόνο αν ο τύπος των μονοπατιών μεταξύ δύο οποιωνδήποτε στοιχείων του είναι (n-1)-type ή αλλιώς

$$\texttt{is-suc}(n)\texttt{-type}(A) :\equiv \Pi x, y : A. \texttt{is-}n\texttt{-type}(x =_A y), n \geq -2$$

Truncations Τα truncations είναι ένας τρόπος να ελέγξουμε την πολυπλοκότητα της δομής ενός τύπου κόβοντας την πιο σύνθετη πληροφορία μονοπατιών που υπάρχει από μια διάσταση και πάνω. Με αυτόν τον τρόπο δημιουργείται ένας νέος τύπος ο οποίος είναι το πολύ n-type, το n ανάλογα με τη διάσταση του truncation, και με τετριμένη δομή σε μεγαλύτερες διαστάσεις. Ένας τρόπος ορισμού των truncations είναι ως επαγωγικών τύπων μεγαλύτερης διάστασης. Εδώ θα παρουσιάσουμε δύο είδη truncation: το propositional truncation ή αλλιώς -1- truncation το οποίο αφαιρεί την οποιαδήποτε πληροφορία-data περιέχει ένας τύπος και την εκφυλίζει στο επίπεδο του hProp και το set truncation ή 0- truncation το οποίο απαλείφει την πληροφορία σχετικά με τα διαφορετικά μονοπάτια όλων των διαστάσεων μεγαλύτερης ή ίσης με 1 εξισώνοντάς τα.

Ας δούμε τον ορισμό του propositional truncation στο πλαίσιο των επαγωγικών τύπων υψηλότερων διαστάσεων. Η λογική του ορισμού είναι η δημιουργία ενός αντικειμένου στο truncation για κάθε αντικείμενο του αρχικού τύπου και η δημιουργία μονοπατιών ισότητας μεταξύ όλων των νέων αντικειμένων. Ο truncated νέος τύπος συμβολίζεται με $\|A\|$ κι έχουμε:

- $|-| : A \to \|A\|$
- $\texttt{squash} : \Pi x, y : \|A\| . x =_A y$

Η ιδέα είναι να καταστήσει την όποια πληροφορία σχετικά με τη δομή του A σε πληροφορία σχετικά με την «απλή κατοίκηση» του τύπου, δηλαδή αφαιρεί το proof relevance περιεχόμενο του τύπου και κρατάει το αν περιέχει κάποιο στοιχείο -αγνοώντας το ποιο είναι αυτό και τι ιδιότητες έχει. Οι αρχές αναδρομής και επαγωγής είναι σχετικά απλές. Στην περίπτωση της αναδρομής για να οριστεί μια συνάρτηση με πεδίο ορισμού έναν truncated τύπο θα πρέπει το πεδίο τιμών να είναι hProp.

$$\frac{\Gamma \vdash P : \texttt{isProp}(B) \quad \Gamma, x : A \vdash N : B}{\Gamma \vdash \texttt{rec}[B](x.N, P) : \|A\| \to B}$$

Τότε, η οριζόμενη συνάρτηση ικανοποιεί τις ισότητες

$$\texttt{rec}[B](x.N,P)(|M|) \equiv [M/x]N : B$$
$$\texttt{ap}_{\texttt{rec}[B](x.N,P)}(\texttt{squash}(|M_1|,|M_2|)) = P([M_1/x]N)([M_2/x]N)$$

Αντίστοιχα, για την επαγωγή και τον ορισμό μιας εξαρτημένης συνάρτησης $f : \Pi x : \|A\| .B$ θα πρέπει για κάθε στοιχείο $x : A$ να υπάρχει ένα στοιχείο $b : B(|x|)$ και να υπάρχουν εξαρτημένα μονοπάτια πάνω από τα μονοπάτια $\texttt{squash}(x,y)$.

$$\frac{\Gamma, z : \|A\| \vdash B : \mathcal{U} \quad \Gamma, x : A \vdash N : B(|x|) \quad \Gamma, x, y : \|A\|, u : B(x), v : B(y) \vdash q : u =^{z.B}_{\texttt{squash}(x,y)} v}{\Gamma \vdash \texttt{ind}[z.B](x.N, x, y, u, v.q) : \Pi x : \|A\| .B}$$

Ικανοποιούνται οι αντίστοιχες ισότητες.

Το set truncation ενός τύπου ορίζεται ακριβώς με τον ίδιο τρόπο με το propositional truncation με τη διαφορά ότι δεν εξισώνει όλα τα στοιχεία ενός τύπου, 0-cells, αλλά όλα τα μονοπάτια του τύπου, 1-cells. Ο τύπος που προκύπτει συμβολίζεται με $\|A\|_0$.

- $|-|_0 : A \to \|A\|_0$
- $\texttt{squash}_0 : \Pi x, y : \|A\|_0 .\Pi p, q : x =_{\|A\|_0} y.p =_{x=_{\|A\|_0} y} q$

Για να ορίσουμε μια αναδρομική συνάρτηση $f : \|A\|_0 \to B$ θα πρέπει ο τύπος B να είναι hSet.

$$\frac{\Gamma \vdash P : \texttt{isSet}(B) \quad \Gamma, x : A \vdash N : B}{\Gamma \vdash \texttt{rec}[B](x.N,P) : \|A\|_0 \to B}$$

Τότε έχουμε τις εξής ισότητες:

$$\texttt{rec}[B](x.N,P)(|M|_0) \equiv [M/x]N : B$$
$$\texttt{ap}_{\texttt{rec}[B](x.N,P)}(\texttt{squash}_0(|M_1|_0,|M_2|_0)) = P([M_1/x]N)([M_2/x]N)$$

Τέλος, για την αναδρομή ακολουθούμε την ίδια λογική με την περίπτωση του propositional truncation

$$\frac{\begin{array}{c}\Gamma, z : \|A\|_0 \vdash B : \mathcal{U} \\ \Gamma, x : A \vdash N : B(|x|_0) \\ \Gamma, x, y : \|A\|_0, p, q : x =_{\|A\|_0} y, u : B(x), v : B(y), r : u =^{z.B}_{p} v, s : u =^{z.B}_{q} v \vdash sq : r =^{w.u =^{z.B}_{w} v}_{\texttt{squash}_0(x,y,p,q)} s\end{array}}{\Gamma \vdash \texttt{ind}[z.B](x.N, x, y, p, q, u, v, r, s.sq) : \Pi x : \|A\|_0 .B}$$

Η συνάρτηση πάλι ικανοποιεί τις προφανείς ισότητες.

Κεφάλαιο 4

Cubical Type Theory - Cubical Agda

Όπως είδαμε στη φορμαλιστική παρουσίαση της Ομοτοπικής Θεωρίας τύπων, για κάθε τύπο A και δύο στοιχεία $x, y : A$, ενδεχομένως ίδια, μπορούμε να δημιουργήσουμε τον τύπο $\mathtt{Id}_A(x, y)$ όλων των αποδείξεων ισότητας των αντικειμένων x, y. Αυτός ο τρόπος προσέγγισης έχει κάποια χαρακτηριστικά. Αρχικά, ο ορισμός του τύπου $\mathtt{Id}_A$ είναι generic ως προς τον τύπο A. Οι κανόνες του τύπου δηλαδή δεν διαφοροποιούνται ανάλογα με τον τύπο A. Ο τύπος $\mathtt{Id}_A$ φαίνεται να ορίζεται ως ο τύπος των ελεύθερων κατασκευών που προκύπτουν με εφαρμογή του τελεστή `J` πάνω στον αντικείμενο `refl`. Ακόμη, ο τελεστής `J` μπορεί να μας παραγάγει μέσω του επαγωγικού τρόπου ορισμού του $\mathtt{Id}_A$ και του εαυτού του μια μεγάλη ποικιλία αποδείξεων, π.χ. $\mathtt{tr}, \mathtt{ap}, ...$, δεδομένης μιας οποιασδήποτε ισότητας ενώ δίνει τρόπο υπολογισμού μόνο για την περίπτωση όπου η ισότητα αυτή είναι η `refl`. Έτσι μπορούμε χωρίς κανένα πρόβλημα να κάνουμε τις εξής β-αναγωγές:

$$(\mathtt{refl})^{-1} \equiv \mathtt{refl} \; \mathtt{refl} \bullet \mathtt{refl} \equiv \mathtt{refl} \; \mathtt{tr}[x.C](\mathtt{refl}) \equiv \mathtt{id} \; \mathtt{ap}_f(\mathtt{refl}) \equiv \mathtt{refl} \; ....$$

οι οποίες όλες βασίζονται στον κανόνα β-αναγωγής του `J`.

Όμως, προκειμένου να εμπλουτίσουμε τη θεωρία μας με το Axiom of Univalence και Επαγωγικούς Τύπους Υψηλότερων Διαστάσεων, έπρεπε να προσθέσουμε κατά κάποιον τρόπο εκ των υστέρων του ορισμού του τύπου $\mathtt{Id}_A$ ένα πλήθος νέων αποδείξεων ισότητας όπως τις $\mathtt{ua}(f)$, $\mathtt{funext}(H)$, `loop`, `seg`,... Αυτό δημιουργεί το εξής πρόβλημα: «το `J` δεν έχει ιδέα τι να κάνει με αυτά τα ορίσματα». Πιο τυπικά, δεν υπάρχει κανόνας υπολογισμού ή κανόνας β-αναγωγής για άλλα αντικείμενα του τύπου $\mathtt{Id}_A$ εκτός από το `refl`. Το αποτέλεσμα είναι η δημιουργία εκφράσεων με σωστούς τύπους (κάνουν type check) που όμως δεν αναλύονται περισσότερο σε τιμές (values). Έχουμε άρα *stuck terms*.

Η Cubical Type Theory είναι μια επέκταση της Homotopy Type Theory που σκοπό έχει την πλήρη ανάκτηση του υπολογιστικού περιεχομένου της θεωρίας χωρίς να μειώσει την εκφραστική της ισχύ. Ο τρόπος με τον οποίο γίνεται αυτό είναι μέσω μιας προσπάθειας μεταφοράς της έννοιας του μονοπατιού από το επίπεδο των αντικειμένων τύπων στο επίπεδο των judgments. Για να γίνει αυτό δημιουργεί μια νέα ερμηνεία για τα μονοπάτια, ως συναρτήσεις από το μοναδιαίο διάστημα $\mathbb{I}$ στον τύπο A. Εδώ το μοναδιαίο διάστημα δεν έχει καμία σχέση με το διάστημα που ορίσαμε ως Επαγωγικό Τύπο υψηλών διαστάσεων. Στην περίπτωση της Cubical Type Theory το $\mathbb{I}$ δεν είναι τύπος αλλά μια επέκταση του context και έχει δική του αλγεβρική δομή. Άρα, στην Cubical Type Theory τα μονοπάτια από το $x : A$ στο $y : A$ είναι «συνεχείς» συναρτήσεις από το $\mathbb{I}$ στον A με τιμή x για $\mathtt{i0} : \mathbb{I}$ και y για $\mathtt{i1} : \mathbb{I}$, όπως ακριβώς στην τοπολογία. Πλέον, λοιπόν, αν έχουμε $\Gamma, i : \mathbb{I} \vdash t : A$ δηλαδή για κάθε σημείο του διαστήματος μπορούμε να βρούμε ένα στοιχείο του τύπου A, τότε μπορούμε να δημιουργήσουμε ένα μονοπάτι στον τύπο A με άκρα τις τιμές του $t : A$ στα άκρα του διαστήματος. Με αυτόν τον τρόπο μεταφέρεται σε γενικότερο πλαίσιο η έννοια του μονοπατιού, στο επίπεδο των contexts, και μετά εσωτερικοποιείται με τη δημιουργία του τύπου του μονοπατιού, `Path`. Η ίδια ακριβώς διαδικασία γίνεται με την περίπτωση της συνεπαγωγής (entailment) και της λογικής συνεπαγωγής (implication). Η συνεπαγωγή (entailment) «$\Gamma, x : A \vdash y : B$» εκφράζει έναν τρόπο εξάρτησης μεταξύ των judgments και αυτή η έννοια εσωτερικοποιείται στο σύστημα με την εισαγωγή του τύπου συναρτήσεων (implication) «$\Gamma \vdash \lambda x.y : A \to B$». Η Cubical Type Theory μεταφέρει τη μεταχείριση των μονοπατιών και της ισότητας σε ένα πιο πρωταρχικό επίπεδο ώστε να μην είναι γενικός ο ορισμός τους σε σχέση με τον τύπο A αλλά ο κάθε τύπος να είναι υπεύθυνος για τον σωστό ορισμό των μονοπατιών του. Έπειτα, δίνεται η εσωτερικοποίηση της έννοιας του μονοπατιού με τη δημιουργία του τύπου `Path`. Σε αυτό το πλαίσιο ο τύπος $\mathtt{Id}_A$ εμφανίζεται ως επαγωγικός τύπος, generic ως προς τον A. Άρα, υπάρχει μια διάκριση μεταξύ του judgmental πλαισίου, των μονοπατιών `Path` και του τύπου ισότητας $\mathtt{Id}_A$ η οποία θα υιοθετηθεί από αυτό το σημείο.

Υπάρχουν διάφορες παραλλαγές της Cubical Type Theory. Εδώ θα παρουσιαστει αυτή που χρησιμοποιήθηκε για την ανάπτυξη της Cubical Agda που είναι και ο proof assistant με τον οποίο έγινε η δημιουργία της βιβλιοθήκης. Θα δώσουμε πρώτα τις νέες επεκτάσεις του συντακτικού και μετά θα δείξουμε πώς αποκτά υπολογιστικό περιεχόμενο το πλαίσιο μας. Αξίζει στο σημείο αυτό να σημειωθεί ότι αυτή η εργασία δεν είναι υποχρεωτική για κάποιον τον οποίον δεν απασχολεί η υπολογιστική πλευρά της Θεωρίας Τύπων. Μπορεί κανείς να δημιουργεί αποδείξεις και αντικείμενα όπως στα προηγούμενα κεφάλαια τα οποία όμως δεν θα «τρέχουν» σαν προγράμματα. Αν όμως μας ενδιαφέρει η υπολογιστική πλευρά της θεωρίας η Cubical Type Theory, προσφέρει μια λύση στο πρόβλημα.

4.1 Τύποι Μονοπατιών

Για να μπορούμε να περιγράψουμε τη νέα ερμηνεία των μονοπατιών επεκτείνουμε το συντακτικό της γλώσσας μας. Υποθέτουμε ότι έχουμε ένα διακριτό άπειρο σύνολο από κατευθύνσεις $, j, k, ...$ Ορίζουμε το $\mathbb{I}$ να είναι η ελεύθερα παραγόμενη De Morgan άλγεβρα με γεννήτορες αυτό το σύνολο κατευθύνσεων. Αυτό σημαίνει ότι τα στοιχεία του $\mathbb{I}$ μπορούν να παραχθούν από την ακόλουθη γραμματική

$$r, s ::= \mathtt{i0} | \mathtt{i1} | i | \sim r | r \vee s | r \wedge s$$

και ικανοποιούν τις ακόλουθες εξισώσεις:

$$\begin{aligned} \sim \mathtt{i0} &= \mathtt{i1} \\ \sim \mathtt{i1} &= \mathtt{i0} \\ \sim (r \vee s) &= (\sim r) \wedge (\sim s) \\ \sim (r \wedge s) &= (\sim r) \vee (\sim s) \end{aligned}$$

Επεκτείνουμε την έννοια του context ώστε να περιλαμβάνει και εκφράσεις της μορφής $\Gamma, i : \mathbb{I}$, και με τον κανόνα

$$\frac{\Gamma \vdash}{\Gamma, i : \mathbb{I} \vdash}$$

Με αυτόν τον τρόπο μπορούμε να αποδόσουμε την έννοια του αντικειμένου που ορίζεται με χρήση μεταβλητών κατεύθυνσης. Έτσι, όταν έχουμε ένα judgment της μορφής $\Gamma, i : \mathbb{I} \vdash M : A$, ο τρόπος με τον οποίο φανταζόμαστε το αποτέλεσμα είναι μια συνεχής γραμμή από αντικείμενα στον A με αρχή το $M(\mathtt{i0}) : A$ και τέλος το $M(\mathtt{i1}) : A$, δηλαδή έναν 1-κύβο. Όταν υπάρχει εξάρτηση δύο μεταβλητών $\Gamma, i : \mathbb{I}, j : \mathbb{I} \vdash M : A$, τότε δημιουργείται ένα τετράγωνο, δηλαδή ένας 2-κύβος, όπως στο σχήμα

$$\begin{tikzcd} M(\mathtt{i0}/i)(\mathtt{i1}/j) \arrow[r, dash, "M(\mathtt{i1}/j)"] \arrow[d, dash, "M(\mathtt{i0}/i)"] & M(\mathtt{i1}/i)(\mathtt{i1}/j) \arrow[d, dash, "M(\mathtt{i1}/i)"] \\ M(\mathtt{i0}/i)(\mathtt{i0}/j) \arrow[r, dash, "M(\mathtt{i0}/j)"] & M(\mathtt{i1}/i)(\mathtt{i0}/j) \end{tikzcd}$$

Στην περίπτωση όπου έχουμε 3 κατευθύνσεις, τότε το αποτέλεσμα είναι ένας κύβος, ενώ όταν έχουμε n διαστάσεις το αποτέλεσμα είναι ένας n-κύβος. Αυτός είναι και ο λόγος για τον οποίο ονομάζεται Cubical Type Theory.

Με τον παραπάνω τρόπο καταφέραμε να περιγράψουμε με τη βοήθεια μεταβλητών $i, j, k, ... : \mathbb{I}$, για τις διάφορες διαστάσεις σημεία μονοπατιών, τετραγώνων, κύβων, υπερκύβων κλπ στο επίπεδο των αποφάνσεων της θεωρίας μας. Τώρα θα ορίσουμε έναν τύπο που εσωτερικεύει αυτήν την έννοια στο σύστημα των τύπων. Αυτός είναι ο τύπος $\mathtt{Path}\ A\ t_0\ t_1$, ο οποίος για δοσμένο τύπο A και δύο αντικείμενα t_0, t_1 αυτού του τύπου κατηγοριοποιεί όλα τα μονοπάτια στον A με αρχή το t_0 και τέλος το t_1.

$$\frac{\Gamma \vdash A \quad \Gamma \vdash t_0 : A \quad \Gamma \vdash t_1 : A}{\Gamma \vdash \texttt{Path}\ A\ t_0\ t_1}$$

Για να δημιουργήσουμε ένα αντικείμενο αυτού του τύπου, ένα μονοπάτι από το t_0 στο t_1 θα πρέπει να έχουμε για κάθε $i : \mathbb{I}$ ένα αντικείμενο $t : A$, ώστε $t(\texttt{i0}) \equiv t_0$ και $t(\texttt{i1}) \equiv t_1$. Η ομοιότητα με τον τοπολογικό ορισμό των μονοπατιών είναι προφανής. Για να συμβολίσουμε το νέο μονοπάτι υιοθετούμε τον τρόπο της Cubical Agda και χρησιμοποιούμε λ αφαίρεση.

$$\frac{\Gamma \vdash A \quad \Gamma, i : \mathbb{I} \vdash \lambda t : A}{\Gamma \vdash \lambda i.t : \texttt{Path}\ A\ t(\texttt{i0})\ t(\texttt{i1})}$$

Βέβαια αφού έχουμε πλέον έναν τρόπο να εκφράσουμε διάφορα σημεία του $\mathbb{I}$ λογικό είναι να μπορούμε και να χειριστούμε το σημείο του μονοπατιού στο οποίο αντιστοιχίζεται αυτό το σημείο. Αυτό μοιάζει με την έννοια της εφαρμογής συνάρτησης και δικαιολογεί την επιλογή του τελεστή λ. Το αποτέλεσμα φυσικά θα ανήκει στον τύπο A. Αυτό μας λέει ο ακόλουθος κανόνας.

$$\frac{\Gamma \vdash A \quad \Gamma \vdash p : \texttt{Path}\ A\ t_0\ t_1 \quad \Gamma \vdash r : \mathbb{I}}{\Gamma \vdash p(r) : A}$$

Ας δούμε πώς μπορούμε να υπολογίζουμε σημεία σε ένα μονοπάτι για κάθε σημείο του $\mathbb{I}$. Αν δημιουργήσουμε ένα μονοπάτι από ένα judgment της μορφής $\Gamma, i : \mathbb{I} \vdash t : A$ και εφαρμόσουμε σε αυτό ένα στοιχείο $r : \mathbb{I}$, θα πάρουμε το αντίστοιχο $[r/i]t$. Δηλαδή

$$\frac{\Gamma \vdash A \quad \Gamma, i : \mathbb{I} \vdash t : A \quad \Gamma \vdash r : \mathbb{I}}{\Gamma \vdash \lambda i.t(r) \equiv [r/i]t : A}$$

Επίσης, είναι απαραίτητο να έχουμε ισότητα μεταξύ του αρχικού άκρου του μονοπατιού και με την τιμή του για $i = \texttt{i0}$ και αντίστοιχα με το τελικό άκρο για $i = \texttt{i1}$.

$$\frac{\Gamma \vdash p : \texttt{Path}\ A\ t_0\ t_1}{\Gamma \vdash p(\texttt{i0}) \equiv t_0} \qquad \frac{\Gamma \vdash p : \texttt{Path}\ A\ t_0\ t_1}{\Gamma \vdash p(\texttt{i1}) \equiv t_1}$$

Τέλος, αν έχουμε δύο μονοπάτια τα οποία συμφωνούν σε κάθε σημείο, τότε τα δύο μονοπάτια θα πρέπει να είναι ίσα.

$$\frac{\Gamma \vdash A \quad \Gamma \vdash p, q : \texttt{Path}\ A\ t_0\ t_1 \quad \Gamma, i : \mathbb{I} \vdash p(i) \equiv q(i) : A}{\Gamma \vdash p \equiv q : \texttt{Path}\ A\ t_0\ t_1}$$

Με αυτόν τον πιο άμεσο τρόπο χειρισμού μονοπατιών μπορούμε πολύ εύκολα ήδη να κάνουμε πολλά πράγματα τα οποία είτε δεν ήταν τόσο άμεσα είτε ήταν αδύνατα στο προηγούμενο πλαίσιο. Για παράδειγμα, το `refl` ορίζεται με πολύ φυσικό τρόπο ως σταθερό μονοπάτι

$$\frac{\Gamma \vdash A \quad \Gamma \vdash a : A}{\Gamma \vdash \texttt{refl} :\equiv \lambda i.a : \texttt{Path}\ A\ a\ a}$$

για το οποίο φυσικά ισχύει ότι $\Gamma, r : \mathbb{I} \vdash \lambda i.a(r) \equiv a : A$. Επιπλέον, για να ορίσουμε το αντίθετο μονοπάτι ενός μονοπατιού p, πολύ απλά εφαρμόζουμε τον τελεστή $\sim i$ στο όρισμα του μονοπατιού.

$$\frac{\Gamma \vdash A \quad \Gamma \vdash a, b : A \quad p : \texttt{Path}\ A\ a\ b}{\Gamma \vdash \lambda i.p(\sim i) : \texttt{Path}\ A\ b\ a}$$

Ακόμη, η μεταφορά μονοπατιών μεταξύ δύο σημείων από το πεδίο ορισμού μιας συνάρτησης στις τιμές τους στο πεδίο τιμών δίνεται από τον πολύ απλό κώδικα:

$$\frac{\Gamma \vdash A, B \quad \Gamma \vdash f : A \to B \quad \Gamma \vdash a, b : A \quad \Gamma \vdash p : \texttt{Path}\ A\ a\ b}{\Gamma \vdash \lambda i.f(p(i)) : \texttt{Path}\ B\ f(a)\ f(b)}$$

Τέλος, εντυπωσιακή είναι η απόδειξη του θεωρήματος πλέον του function extensionality ήδη με αυτές τις απλές πρώτες επεκτάσεις. Έστω, A, B τύποι, $f, g : A \to B$ δύο συναρτήσεις από τον A στον B και $H : \Pi a : A.\texttt{Path}\ B\ f(a)\ g(a)$ μια ομοτοπία μεταξύ των A και B. Τότε μπορούμε να κάνουμε τα εξής βήματα:

$$\begin{gathered}
\Gamma \vdash H : \Pi a : A.\texttt{Path}\ B\ f(a)\ g(a) \\
\Gamma, x : A \vdash H(x) : \texttt{Path}\ B\ f(x)\ g(x) \\
\Gamma, x : A, r : A \vdash H(x)(r) : B \\
\Gamma, r : A, x : A \vdash H(x)(r) : B \\
\Gamma, r : A \vdash \lambda a.H(a)(r) : A \to B \\
\Gamma \vdash \lambda i.\lambda a.H(a)(i) : \texttt{Path}\ (A \to B)\ f\ g
\end{gathered}$$

Για να ελέγξουμε ότι όντως τα άκρα του μονοπατιού που κατασκευάσαμε είναι αυτά που γράψαμε, πολύ απλά υπολογίζουμε την τιμή για $i \equiv \texttt{i0}$

$$(\lambda i.\lambda a.H(a)(r))(\texttt{i0}) \equiv \lambda a.H(a)(\texttt{i0}) \equiv \lambda a.f(a) \equiv f$$

και όμοια για το άλλο άκρο. Αυτό που έκανε δυνατή αυτήν την απόδειξη είναι η εναλλαγή των όρων $x : A, r : A$ στο context, κάτι που δεν θα ήταν δυνατόν πριν τη μεταφορά των μονοπατιών σε αυτό το επίπεδο.

4.2 Composition and Coercion

Ακόμα δεν έχει αναφερθεί μια πολύ σημαντική πράξη ανάμεσα στα μονοπάτια, τη συνένωση μονοπατιών, όπως, επίσως, δεν έχουν αναλυθεί μονοπάτια στον τύπο του σύμπαντος, δηλαδή μονοπάτια μεταξύ τύπων. Για να μπορέσουν να υποστηριχθούν στην Cubical Type Theory οι πράξεις της συνένωσης μονοπατιών και της μεταφοράς αντικειμένων μεταξύ ίσων τύπων, η Cubical Type Theory υποχρεώνει όλους τους τύπους να παρέχουν *στον ορισμό* τους έναν τρόπο για αυτές τις πράξεις. Με άλλα λόγια, για να επανακτήσουν το υπολογιστικό τους περιεχόμενο μονοπάτια όπως τα `ua`, `loop` κλπ, θα πρέπει οι ίδιοι οι τύποι που τα περιέχουν να δίνουν τον τρόπο υπολογισμού των πράξεών τους. Αποδεικνύεται ότι για να έχουμε τα μονοπάτια που θέλουμε και να συμπεριφέρονται με έναν φυσιολογικό τρόπο, δηλαδή να είναι κλειστά ως προς τις πράξεις μας, θα πρέπει να ικανοποιούν κάποιες συνθήκες οι οποίες ονομάζονται συνθήκες Kan. Επιπρόσθετα, κάθε τύπος για να οριστεί σωστά στην Cubical Type Theory, θα πρέπει εκτός από τους διάφορους κανόνες που έχουμε δει, κανόνες σχηματισμού, εισαγωγής, απαλοιφής, β και η αναγωγής, να παρέχει τελεστές-μάρτυρες της ικανοποίησης των συνθηκών Kan.

4.2.1 Composition

Ειδικότερα στην Cubical Agda θα εισαγάγουμε δύο τελεστές. Ο πρώτος ονομάζεται ομογενής σύνθεση (homogeneous composition). Η ομογενής σύνθεση ξεκινάει από έναν κύβο από τον οποίο λείπει κάποια έδρα (face) και δίνει τη σύνθεση των υπάρχοντων εδρών για τη δημιουργία της έδρας που λείπει ώστε ο κύβος να είναι πλήρης. Για δύο διαστάσεις δηλαδή έχουμε την ακόλουθη εικόνα

$$\begin{tikzcd}
\bullet \arrow[r, dashed] & \bullet \\
\bullet \arrow[u, "\texttt{face}"] \arrow[r] & \bullet \arrow[u, "\texttt{face}"]
\end{tikzcd}$$

Η ιδέα αυτή επεκτείνεται σε κάθε διάσταση. Ονομάζεται ομογενής διότι κατά τη διάρκεια της μεταφοράς μας από την κάτω έδρα προς τη νεοδημιουργηθείσα μένουμε στον ίδιο τύπο A. Στην Cubical Agda αυτός ο τελεστής συμβολίζεται με `hcomp`. Δυστυχώς `hcomp` θα ονομαζόταν και ο τελεστής ετερογενούς σύνθεσης (heterogeneous composition), επομένως απαιτείται ιδιαίτερη προσοχή με τον συμβολισμό. Ο τελεστής hcomp θα πρέπει να δίνει πάντα την πάνω έδρα ακόμα κι όταν λείπουν και κάποιες από τις πλαϊνές έδρες όπως π.χ.:

$$\begin{tikzcd} \bullet \arrow[r, dashed] & \bullet & \bullet \arrow[r, dashed] & \bullet \\ \bullet \arrow[u, dashed, "\texttt{face}"] \arrow[r] & \bullet \arrow[u, "\texttt{face}"] & \bullet \arrow[u, dashed, "\texttt{face}"] \arrow[r] & \bullet \arrow[u, dashed, "\texttt{face}"] \end{tikzcd}$$

και βέβαια σε κάθε διάσταση.

4.2.2 Coercion

Ο δεύτερος τελεστής που θα ορίσουμε για να συμπληρώσουμε τις συνθήκες Kan είναι ο τελεστής του Coercion. Η έννοια προέρχεται από τις ιδέες που εξετάστηκαν στο προηγούμενο κεφάλαιο. Σύμφωνα με αυτές, όταν έχουμε μια ισότητα p μεταξύ δύο τύπων στο σύμπαν $A, B : \mathcal{U}$ τότε μπορεί να κατασκευαστεί μια συνάρτηση μεταφοράς αντικειμένων από τον έναν τύπο στον άλλο με τον εξής τρόπο:

$$f :\equiv \texttt{tr}[x.x](p) : A \to B$$

Αυτό είναι μια πολύ λογική κατάσταση την οποία θέλουμε να διατηρήσουμε στην Cubical Type Theory. Άλλωστε, αφού οι δύο τύποι είναι ίσοι, θα πρέπει να αντιστοιχίζονται τα στοιχεία τους. Καθώς αυτή η πράξη θυμίζει τον εξαναγκασμό τύπων στις γλώσσες προγραμματισμού, υιοθετούμε την ίδια ορολογία.

Στην περίπτωση της Cubical Type Theory μπορούμε να έχουμε μονοπάτια μεταξύ τύπων ακριβώς όπως και μεταξύ αντικειμένων σε όλες τις διαστάσεις. Coercion, λοιπόν, είναι η μεταφορά αντικειμένων από μια από τις έδρες στην ακριβώς απέναντι. Αν, για παράδειγμα, έχουμε το παρακάτω τετράγωνο ισότητας μεταξύ τύπων, ο τελεστής Coercion μας επιτρέπει να μεταφέρουμε αντικείμενα από την κάτω έδρα στην πάνω.

$$\begin{tikzcd} A(i/\texttt{i0})(j/\texttt{i1}) \arrow[r, dashed] & A(i/\texttt{i1})(j/\texttt{i1}) \\ A(i/\texttt{i0})(j/\texttt{i0}) \arrow[u] \arrow[r] & A(i/\texttt{i1})(j/\texttt{i0}) \arrow[u] \end{tikzcd}$$

Είναι σημαντικό να παρατηρήσουμε τη διαφορά μεταξύ Coercion και ομογενούς σύνθεσης. Κατά την ομογενή σύνθεση δεν μετακινούμαστε εκτός του αρχικού τύπου A ενώ κατά τη χρήση του τελεστή Coercion μετακινούμαστε σε έναν άλλο τύπο ίσο με τον αρχικό.

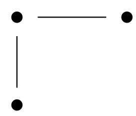
4.2.3 Partial Elements

Για να μπορέσουμε να περιγράψουμε τη λειτουργία αυτών των τελεστών πρέπει να προσθέσουμε στο συντακτικό μας έναν τρόπο περιγραφής n-κύβων στους οποίους λείπουν μία ή περισσότερες έδρες. Αυτό γίνεται με τη χρήση εξισώσεων διαστάσεων. Για παράδειγμα, στον τριδιάστατο κύβο η εξίσωση $(i = \texttt{i0})$ περιγράφει όλα τα στοιχεία του κύβου ελεύθερα ως προς τις διαστάσεις j, k που έχουν, όμως, την τιμή i0 ως τιμή της μεταβλητής διάστασης i. Πρόκειται άρα για την έδρα η οποία βρίσκεται στη διάσταση $(i = \texttt{i0})$. Για να μπορούμε να περιγράψουμε υποπολύεδρα μεγαλύτερης ποικιλίας επιτρέπουμε δύο πράξεις μεταξύ αυτών των εξισώσεων, τις $\wedge$ και $\vee$, ώστε να ικανοποιούνται οι νόμοι της επιμεριστικότητας και $(i = \texttt{i0}) \wedge (i = \texttt{i1}) = 0_{\mathbb{F}}$, όπου $0_{\mathbb{F}}$ και $1_{\mathbb{F}}$ συμβολίζουν την εξίσωση που δεν ικανοποιεί κανένα στοιχείο και την εξίσωση που ικανοποιούν όλα τα στοιχεία του κύβου. Έτσι, δημιουργείται μια αλγεβρική δομή ενός distributive lattice το οποίο μας δίνει την πλήρη περιγραφή που θέλουμε για τα διάφορα μέρη του κύβου. Μερικά παραδείγματα:

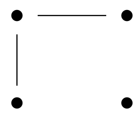
$(i = \texttt{i0}) \wedge (i = \texttt{i1})$: δύο σημεία χωρίς ενδιάμεση σύνδεση

$$\bullet \qquad \bullet$$

$(i = \texttt{i0}) \vee (j = \texttt{i1})$: το άθροισμα δύο εδρών



$(i = \texttt{i0}) \vee (j = \texttt{i1}) \vee ((i = \texttt{i1}) \wedge (j = \texttt{i1}))$: το άθροισμα δύο εδρών και μίας κορυφής που δεν ανήκει σε αυτές



Ακόμη, είναι δυνατόν να εισάγουμε αυτούς τους περιορισμούς στο επίπεδο του context. Αν έχουμε ένα αντικείμενο ϕ του lattice των partial elements, τότε είναι δυνατόν να το εισάγουμε στο συντακτικό των context με τον εξής τρόπο

$$\Gamma, \phi \vdash$$

Αυτό μας δίνει μεγαλύτερη εκφραστικότητα καθώς μπορούμε να αποδεικνύουμε judgments μόνο στο υποπολύεδρο που ορίζει ο περιορισμός ϕ. Πιο συγκεκριμένα, αν έχουμε ένα αντικείμενο $M : A$ και γράψουμε $\phi \vdash M \equiv N : A$, τότε αυτό σημαίνει ότι τα δύο αντικείμενα είναι ίσα μόνο στα σημεία του κύβου που ικανοποιούν τον περιορισμό ϕ και όχι υποχρεωτικά παντού. Το συγκεκριμένο γράφεται και ως

$$M : A \text{ και } \phi \vdash M \equiv N : A \implies M : A[\phi \mapsto N]$$

Για περισσότερες λεπτομέρειες πάνω στον ορισμό των παραπάνω παραπέμπουμε στο [Cohe16].

4.2.4 Συντακτικό και λειτουργία

Περνάμε τώρα στη σύνταξη και λειτουργία των δύο τελεστών. Για την περίπτωση της ομογενούς σύνθεσης χρειαζόμαστε όπως είδαμε τα εξής:

- $A : \mathcal{U}$, ο τύπος μεσα στον οποίο θα γίνει η σύνθεση
- $M : A$, την «κάτω» έδρα του κύβου την οποία και πρέπει να έχουμε πάντα. Το στην περίπτωση μας είναι n-διάστατο, αν εξαρτάται από n μεταβλητές διάστασης του context
- $[\phi_1 \mapsto u_1, \phi_2 \mapsto u_2, ...]$, μια πεπερασμένη συλλογή από πλαϊνές έδρες του υπερκύβου μας σε ένα σύμπλεγμα που ονομάζεται σύστημα. Εδώ το κάθε ϕ_i αντιστοιχεί και σε μια εξίσωση μεταβλητών διάστασης και το κάθε u_i αντιστοιχεί σε ένα αντικείμενο του A το οποίο είναι judgmentally ίσο με τον υπερκύβο μας στα σημεία που ικανοποιούν την εξίσωση ϕ_i. Θα πρέπει φυσικά στις τομές των εξισώσεων ϕ_{i_1}, ϕ_{i_2} τα u_{i_1}, u_{i_2} να είναι επίσης judgmentally ίσα, καθώς επίσης και οι τομές των εδρών με το M.

Αν έχουμε όλα τα παραπάνω, τότε ορίζεται ο τελεστής της ομογενούς σύνθεσης με την εξής σύνταξη

$$\texttt{hcomp}^k A[\phi_1 \mapsto u_1, \phi_2 \mapsto u_2, ...]M : A$$

όπου k είναι η διάσταση κατά μήκος της οποίας γίνεται η σύνθεση.

Για την περίπτωση του τελεστή Coercion χρειαζόμαστε ένα μονοπάτι μεταξύ δύο τύπων στο $\mathcal{U}$, δηλαδή έναν τύπο με εξάρτηση ως προς κάποια μεταβλητή διάστασης $\Gamma, j : \mathbb{I} \vdash A$ ή σε μορφή μονοπατιού στο σύμπαν $A : \texttt{Path}\ \mathcal{U}\ A(\texttt{i0})\ A(\texttt{i0})$. Ο σκοπός μας είναι η δημιουργία μιας συνάρτησης με τύπο $A(\texttt{i0}) \to A(\texttt{i1})$. Η υλοποίηση που επιλέγει η Cubical Agda γενικεύει τον τελεστή της μεταφοράς δίνοντας την επιλογή να θέτει περιορισμούς στον τρόπο με τον οποίο γίνεται η μεταφορά σε άλλες διαστάσεις εκτός του $i : \mathbb{I}$. Ειδικότερα, για τα σημεία που ικανοποιούν τους περιορισμούς η συνάρτηση που προκύπτει θα πρέπει να είναι η ταυτοτική και κατά συνέπεια και ο τύπος A σταθερός σε αυτά τα σημεία. Ο τρόπος με τον οποίο παριστάνονται οι περιορισμοί δεν είναι με κάποιο στοιχείο του lattice των εδρών του κύβου αλλά με ένα στοιχείο του $\mathbb{I}$. Αποδεικνύεται ότι τα σημεία που ικανοποιούν ένα σύνολο εξισώσεων διάστασης μπορούν να αναπαρασταθούν με ένα αντικείμενο του $\mathbb{I}$ όπου ισούται με $\texttt{i1}$. Για παράδειγμα, οι περιορισμοί $(i = \texttt{i0}) \vee (j = \texttt{i1}) \vee ((i = \texttt{i1}) \wedge (j = \texttt{i1}))$ που είδαμε πιο πάνω περιγράφονται ισοδύναμα με τα σημεία για τα οποία ισχύει ότι το σημείο $(\sim i) \vee j \vee (i \wedge j) : \mathbb{I}$ ισούται με $\texttt{i1}$. Τότε, αν:

- $\Gamma, j : \mathbb{I} \vdash A$ είναι το μονοπάτι στο οποίο μας ενδιαφέρει να μεταφέρουμε αντικείμενα από τον $A(\texttt{i0})$ στον $A(\texttt{i1})$ κατά μήκος της διάστασης j.
- $r : \mathbb{I}$ είναι ένα σημείο του διαστήματος που περιγράφει τα σημεία όπου θα ισχύουν οι περιορισμοί, όπου δηλαδή "$r = \texttt{i1}$"
- $M : A(\texttt{i0})$ ένα σημείο του τύπου στην αρχή του μονοπατιού

έχουμε τον τελεστή της μεταφοράς

$$\texttt{transp}^j A\ r\ M : A(\texttt{i1})$$

Για να κατανοήσουμε λίγο καλύτερα την έννοια του περιορισμού $r : \mathbb{I}$ ας δούμε ορισμένα παραδείγματα.

- Αν $r \equiv \texttt{i0} : \mathbb{I}$ τότε ο περιορισμός $\texttt{i0} = \texttt{i1}$ δεν ικανοποιείται για κανένα σημείο του κύβου και άρα δεν υπάρχει κανένας ουσιαστικά περιορισμός. Μπορούμε να ορίσουμε επομένως

  $$\texttt{transport} :\equiv \texttt{transp}^i\ A\ \texttt{i0} : [\texttt{i0}/i]A \to [\texttt{i1}/i]A$$

  τη συνάρτηση της μεταφοράς αντικειμένων μεταξύ δύο ίσων τύπων.

- Αν $r \equiv \texttt{i1} : \mathbb{I}$, τότε ο περιορισμός $\texttt{i1} = \texttt{i1}$ ικανοποιείται για κάθε σημείο του κύβου και άρα η συνάρτηση θα πρέπει να είναι παντού ταυτοτική και το μονοπάτι της ισότητας του A να είναι το σταθερό μονοπάτι

  $$\texttt{transp}^i\ A\ \texttt{i1} \equiv \texttt{id} : [\texttt{i0}/i]A \to [\texttt{i0}/i]A$$

- Αν έχουμε το εξής τετράγωνο

  $$\begin{array}{ccc} A(i/\texttt{i0})(j/\texttt{i1}) & \longrightarrow & A(i/\texttt{i1})(j/\texttt{i1}) \\ \uparrow & & \uparrow \\ A(i/\texttt{i0})(j/\texttt{i0}) & \longrightarrow & A(i/\texttt{i1})(j/\texttt{i0}) \end{array}$$

  και θέλουμε να μεταφέρουμε κατά τη διάσταση j διατηρώντας φερ' ειπείν τα στοιχεία αναλλοίωτα κατά τη μεταφορά τους στην αριστερή έδρα, τότε ο περιορισμός είναι ο "$i = \texttt{i0} \Leftrightarrow \sim i = \texttt{i1}$". Αυτό επάγει έναν περιορισμό και στο τετράγωνο ώστε η αριστερή πλευρά να είναι το σταθερό μονοπάτι και οι τύποι $A(i/\texttt{i0})(j/\texttt{i0})$ και $A(i/\texttt{i0})(j/\texttt{i1})$ να είναι judgmentally ίσοι.

  $$\Gamma, i : \mathbb{I} \vdash \texttt{transp}^j\ A\ (\sim i) : [\texttt{i0}/j]A \to [\texttt{i1}/j]A$$

Τώρα που αναλύθηκε πώς συντάσσονται οι τελεστές αυτοί πρέπει να εξεταστεί πώς πρέπει να τους χρησιμοποιούμε. Όπως αναφέρθηκε και στην εισαγωγή του κεφαλαίου, ο κάθε τύπος θα πρέπει να είναι υπεύθυνος για τον τρόπο με τον οποίο γίνονται αυτές οι πράξεις. Με άλλα λόγια, θα πρέπει να ξαναδώσουμε τους ορισμούς για όλους τους τύπους που παρουσιάσαμε προσθέτοντας τις ζητούμενες λειτουργίες.

4.2.5 Περίπτωση Μελέτης - Εξαρτημένες Συναρτήσεις

Ως παράδειγμα θα δείξουμε πώς γίνεται αυτό για τον τύπο των Εξαρτημένων Συναρτήσεων. Έστω ένας κύβος σε κάποιο τύπο συναρτήσεων $\Pi x : A.B$, $M : \Pi x : A.B$ και $[\phi_1 \mapsto u_1, \phi_2 \mapsto u_2, ...]$ ένα σύστημα εδρών, ώστε να ικανοποιούνται οι παραπάνω περιορισμοί. Τότε θέτουμε

$$\texttt{hcomp}^i(\Pi x : A.B)[\phi_1 \mapsto u_1, \phi_2 \mapsto u_2, ...]M :\equiv \lambda x.\texttt{hcomp}^i B[\phi_1 \mapsto u_1(x), \phi_2 \mapsto u_2(x), ...]M(x) : \Pi x : A.B$$

Δηλαδή ορίζουμε τον τελεστή της σύνθεσης μονοπατιών συναρτήσεων δίνοντας σημασία μόνο στην input-output συμπεριφορά των εδρών - συναρτήσεων. Το αποτέλεσμα θα πρέπει να είναι φυσικά μια συνάρτηση μέσα στον τύπο $\Pi x : A.B$, καθώς κατά τη διάρκεια της ομογενούς σύνθεσης ο τύπος δεν αλλάζει.

Για το Coercion πάλι εστιάζουμε στη συμπεριφορά της συνάρτησης ως σύστημα εισόδου-εξόδου. Η μεταφερμένη συνάρτηση παίρνει ορίσματα στον τύπο $[\texttt{i1}/i]A$ αλλά η συνάρτηση που έχουμε ως δεδομένο ανήκει στον τύπο εκκίνησης $M : [\texttt{i0}/i](\Pi x : A.B)$. Άρα, για να κάνουμε τον υπολογισμό μεταφέρουμε τα ορίσματα στον τύπο $u : [\texttt{i0}/i]A$, κάνουμε τον υπολογισμό με την M και μεταφέρουμε το αποτέλεσμα $M(u)$ στον τύπο $[\texttt{i1}/i]B(u)$.

$$\texttt{transp}^i\ (\Pi x : A.B)\ r :\equiv \lambda x.\texttt{transp}^i\ [u/x]B\ r\ M(u) : [\texttt{i1}/i](\Pi x : A.B), \text{όπου } u :\equiv \texttt{transp}^{\sim i}\ A\ r\ x : [\texttt{i0}/i]A$$

Η Agda ορίζει σε όλους τους τύπους αυτούς τους τελεστές ώστε να είναι πλήρως υπολογίσιμοι για κάθε περίπτωση μονοπατιού. Εσωτερικά κάνει pattern matching στον τύπο και εφαρμόζει με τον κατάλληλο τρόπο αυτές τις νέες πράξεις. Εδώ δεν θα δείξουμε πώς υλοποιούνται για τους υπόλοιπους τύπους παρά μόνο για την περίπτωση του σύμπαντος. Παραπέμπουμε στα [Cohe16] και [Vezz19] για περισσότερες λεπτομέρειες σχετικά με την υλοποίηση.

4.2.6 Σύμπαντα στην Cubical Type Theory

Για να οριστούν σωστά οι πράξεις της ομογενούς σύνθεσης και του Coercion στην περίπτωση του σύμπαντος με τελικό σκοπό την απόκτηση μια υλοποίησης του Univalence Axiom με υπολογιστικό χαρακτήρα, εισάγεται ένα νέο είδος τύπων, οι Glue Types. Οι τύποι Glue εσωτερικοποιούν την ουσία του Univalence Axiom επιτρέποντας να κατασκευάσουμε κύβους όπου αντί για μονοπάτια σε ορισμένες έδρες έχουμε ισοδυναμίες. Η ιδέα είναι ανάλογη με τον τρόπο λειτουργίας του `hcomp`, όπου συνθέτουμε μονοπάτια, αλλά στην περίπτωση των Glue τύπων συνθέτουμε ισοδυναμίες τύπων.

Glue Types Ο τύπος Glue είναι κι αυτός ένας τύπος, που σημαίνει ότι πρέπει να τον ορίσουμε όπως ακριβώς και όλους τους άλλους. Για να σχηματίσουμε έναν τύπο Glue απαιτείται να έχουμε έναν τύπο $A : \mathcal{U}$ ο οποίος είναι ισοδύναμος με κάποιον τύπο $\phi \vdash T : \mathcal{U}$ πάνω σε κάποια σημεία του υπερκύβου με κάποιες ισοδυναμίες $\phi \vdash E : T \simeq A$, όπως παραδείγματος χάριν στην εικόνα

$$\begin{tikzcd} T \arrow[d, "E"] & T \arrow[d, "E"] \\ A(i/\texttt{i0}) \arrow[r] & A(i/\texttt{i1}) \end{tikzcd}$$

Τότε μπορούμε να ορίσουμε τον τύπο Glue μέσω του εξής κανόνα σχηματισμού

$$\frac{\Gamma \vdash A : \mathcal{U} \quad \Gamma, \phi \vdash T : \mathcal{U} \quad \Gamma, \phi \vdash E : T \simeq A}{\Gamma \vdash \texttt{Glue}[\phi \mapsto (T, E)]A : \mathcal{U}[\phi \mapsto T]}\ F - \texttt{Glue}$$

Παρατηρούμε ότι πάνω στα σημεία που ικανοποιούν τον ϕ ο τύπος Glue ταυτίζεται με τον T.

Έπειτα, ο τύπος Glue έχει έναν κανόνα εισαγωγής στοιχείων του με τον κατασκευαστή `glue`, ο οποίος για κάθε μερικό στοιχείο $\phi \vdash t : T$ του τύπου T και κάθε στοιχείο $a : A$ το οποίο είναι ίσο με το σημείο $t : T$ υπό την ισοδυναμία στα σημεία του περιορισμού ϕ δίνει ένα στοιχείο του τύπου $\texttt{Glue}[\phi \mapsto (T, E)]A$.

$$\frac{\Gamma \vdash t : T \quad \Gamma \vdash a : A[\phi \mapsto \texttt{fst}(E)(t)]}{\Gamma \vdash \texttt{glue}[\phi \mapsto t]a : (\texttt{Glue}[\phi \mapsto (T, E)]A)[\phi \mapsto t]} \; I - \texttt{Glue}$$

Πάλι, παρατηρούμε ότι το στοιχείο $\texttt{glue}[\phi \mapsto t]a$ είναι ίσο judgmentally με το t όπου απαιτεί το ϕ το οποίο δεν προκαλεί λάθος τύπων αφού σε αυτά τα σημεία οι τύποι $\texttt{Glue}[\phi \mapsto (T, E)]$ και T ταυτίζονται.

Ο κανόνας απαλοιφής ορίζει μια συνάρτηση $\texttt{unglue}$ η οποία στέλνει αντικείμενα $u : \texttt{Glue}[\phi \mapsto (T, E)]$ πίσω στον A φροντίζοντας τα στοιχεία του T δηλαδή αυτά που υπόκεινται στους περιορισμούς του ϕ να πάνε στα ισοδύναμά τους του A.

$$\frac{\Gamma \vdash u : \texttt{Glue}[\phi \mapsto (T, E)]}{\Gamma \vdash \texttt{unglue}[\phi \mapsto E]u : A[\phi \mapsto \texttt{fst}(E)(u)]} \; E - \texttt{Glue}$$

Οι δύο αυτές συναρτήσεις ικανοποιούν τους εξής κανόνες β-αναγωγής οι οποίοι είναι και οι πιο προφανείς

$$\frac{\Gamma \vdash t : T \quad \Gamma \vdash a : A[\phi \mapsto \texttt{fst}(E)(t)]}{\Gamma \vdash \texttt{unglue}(\texttt{glue}[\phi \mapsto t]a) \equiv a : A[\phi \mapsto \texttt{fst}(E)(t)]} \; \beta - \texttt{Glue}$$

$$\frac{\Gamma \vdash u : \texttt{Glue}[\phi \mapsto (T, E)]}{\Gamma \vdash \texttt{glue}(\texttt{unglue}[\phi \mapsto E]u) \equiv u : \texttt{Glue}[\phi \mapsto (T, E)]} \; \beta - \texttt{Glue}$$

Το τελευταίο κομμάτι για τον ορισμό του τύπου Glue είναι ο ορισμός των τελεστών ομογενούς σύνθεσης και coercion. Αυτά τα δύο σημεία είναι τα πιο σύνθετα και τεχνικά στοιχεία της Cubical Type Theory. Επίσης, το κομμάτι αυτό δεν είναι απαραίτητο για τον ορισμό του Univalence Axiom που είναι τελικά σκοπός της ανά χείρας εργασίας αλλά ούτε και για τη διαισθητική κατανόηση των τύπων Glue. Για αυτούς τους λόγους παραπέμπουμε για τον ορισμό τους στο [Cohe16].

Υπολογισμοί στο Σύμπαν Ας δούμε τώρα πώς μπορούμε να τους χρησιμοποιήσουμε για να ορίσουμε τους τελεστές ομογενούς σύνθεσης και coercion στο σύμπαν $\mathcal{U}$ και να κατασκευάσουμε το Univalence Axiom, το οποίο πλέον θα είναι θεώρημα στην Cubical Type Theory με πλήρως υπολογίσιμη συμπεριφορά.

Για την περίπτωση της ομογενούς σύνθεσης υποθέτουμε ότι έχουμε έναν τύπο $A : \mathcal{U}$, ένα σύστημα εδρών $[\phi_1 \mapsto u_1, \phi_2 \mapsto u_2, ...]$ και ότι θέλουμε να συνθέσουμε κατά τη διάσταση i. Τότε μπορούμε να μετατρέψουμε αυτές τις έδρες από μονοπάτια σε ισοδυναμίες με τον εξής τρόπο

$$E_n :\equiv \texttt{transport}^i \; [\sim i/i]u_n$$

αντιστρέφοντας τη φορά της μεταφοράς αφού η φορά των ισοδυναμιών στον ορισμό του Glue είναι από τον T στον A. Τότε θέτουμε

$$\texttt{hcomp}^i \; \mathcal{U} \; [\phi_1 \mapsto u_1, \phi_2 \mapsto u_2, ...] \; A :\equiv \texttt{Glue}[\phi_1 \mapsto (u_1(\texttt{i1}), E_1), \phi_2 \mapsto (u_2(\texttt{i1}), E_2), ...]A$$

Δίνουμε μια εικόνα ως εποπτικό παράδειγμα

$$\begin{tikzcd} u_1(\texttt{i1}) & u_2(\texttt{i1}) \\ A(j/\texttt{i0}) \arrow[u, "u_1"] \arrow[r] & A(j/\texttt{i1}) \arrow[u, "u_2"] \end{tikzcd} \quad \text{μετατρέπουμε σε} \quad \begin{tikzcd} u_1(\texttt{i1}) \arrow[r, "Glue"] \arrow[d, "E_1"] & u_2(\texttt{i1}) \arrow[d, "E_1"] \\ A(j/\texttt{i0}) \arrow[r] & A(j/\texttt{i1}) \end{tikzcd}$$

Το coercion γίνεται πολύ πιο εύκολα χωρίς χρήση των τύπων Glue, πολύ απλά επιστρέφοντας τον ίδιο τύπο. Δηλαδή δεν κάνουμε απολύτως καμία αλλαγή στον τύπο.

$$\texttt{transp}^i \; \mathcal{U} \; r \; A :\equiv A$$

Είμαστε πλέον έτοιμοι να αποδείξουμε το Univalence Axiom. Υποθέτουμε ότι έχουμε δύο τύπους $A, B : \mathcal{U}$ και μια απόδειξη της ισοδυναμίας τους $e : A \simeq B$. Τότε μπορούμε να κατασκευάσουμε την παρακάτω μερική οικογένεια τύπων στο σύμπαν με ισοδυναμίες, όμως αντί για μονοπάτια όπως στον ορισμό του τύπου Glue, όπου $\texttt{idEquiv}(B)$ είναι η αυτοπαθής ισοδυναμία

$$\begin{tikzcd} A \arrow[d, "e"] & A \arrow[r, "Glue"] \arrow[d, "e"] & B \arrow[d, "\texttt{idEquiv}(B)"] \\ B & B \arrow[r] & B \end{tikzcd}$$

Ορίζουμε τελικά

$$\texttt{ua}(e)(i) :\equiv \texttt{Glue}[(i = \texttt{i0}) \mapsto (e, A), (i = \texttt{i1}) \mapsto (\texttt{idEquiv}(B), B)]B$$

και έχουμε ότι

$$\texttt{transp}^i\ \texttt{ua}(e)(i)\ \texttt{i1}\ x \equiv \texttt{id}(\texttt{transp}^i\ B\ \texttt{i1}\ (\texttt{fst}(e)(x)))$$

Δηλαδή η μεταφορά πάνω από τον τύπο $\texttt{ua}(e)$ είναι μεταφορά του ισοδύναμου σημείου πάνω από τον τύπο $\Gamma, i : \mathbb{I} \vdash B$. Αν ο τύπος B είναι σταθερός ως προς το $i : \mathbb{I}$, τότε έχουμε την εξής ισότητα

$$\texttt{transport}\ \texttt{ua}(e)\ x \equiv \texttt{fst}(e)(x)$$

Κεφάλαιο 5

Η Θεμελιώδης Ομάδα ενός Bouquet από κύκλους

Σε αυτό το κεφάλαιο θα παρουσιάσουμε το πειραματικό μέρος της εργασίας. Σκοπός είναι η απόδειξη ενός γνωστού αποτελέσματος από την αλγεβρική τοπολογία σε Cubical Agda. Το ζητούμενο θεώρημα είναι το εξής

Η θεμελιώδης ομάδα ενός Bouquet από κύκλους με δείκτες από έναν τύπο A
είναι ίση με την ελεύθερα παραγόμενη ομάδα με γεννήτορες τα στοιχεία του A

Για να το εξηγήσουμε καλύτερα αλλά και για να περιγράψουμε τα σημαντικότερα κομμάτια της βιβλιοθήκης που αναπτύχθηκε θα το αναλύσουμε στα εξής σημεία:

- Ελεύθερα παραγόμενη ομάδα
- Ελεύθερα παραγόμενο groupoid
- Θεμελιώδης ομάδα
- Απόδειξη

Σε αυτά θα γίνεται παράθεση του κώδικα από τη βιβλιοθήκη και θα τηρείται η ονοματοδοσία που έχει χρησιμοποιηθεί εκεί. Όπου είναι απαραίτητο θα γίνεται παρουσίαση ολόκληρων των επιμέρους αποδείξεων. Η βιβλιοθήκη είναι ελεύθερα προσβάσιμη εδώ: https://github.com/gmagaf/Bouquet-HoTT-Cubical-Agda.

5.1 FreeGroup of A

Η ελεύθερα παραγόμενη ομάδα από στοιχεία του A είναι ένα αλγεβρικό αντικείμενο. Δεδομένου ενός συνόλου S μπορούμε πάντα να κατασκευάσουμε λέξεις με βάση τα στοιχεία αυτού του συνόλου σαν γράμματα. Έτσι, με πράξη την παράθεση στοιχείων $\mathtt{m}$ και την κενή λέξη που συμβολίζεται με $\mathtt{e}$ μπορούμε να κατασκευάσουμε ένα μονοειδές (το ελεύθερα παραγόμενο μονοειδές από στοιχεία του S). Χρειαζόμαστε μόνο μια απεικόνιση που στέλνει τα στοιχεία του S σε στοιχεία του μονοειδούς, σε λέξεις με ένα γράμμα: το ίδιο το στοιχείο, την οποία απεικόνιση ονομάζουμε η. Έτσι, έχουμε την παρακάτω κατασκευή

- $\eta : S \to \mathbb{G}$ Η απεικόνιση εμφύτευσης του S στο $\mathbb{G}$
- $\mathtt{m} : \mathbb{G} \to \mathbb{G} \to \mathbb{G}$ Η πράξη της παράθεσης λέξεων
- $\mathtt{e} \in \mathbb{G}$ Το ταυτοτικό στοιχείο του $\mathbb{G}$

Αυτά φυσικά, για να είναι το $\mathbb{G}$ όντως μονοειδές, θα πρέπει να ικανοποιούν τον προσεταιριστικό νόμο και τους νόμους του ταυτοτικού στοιχείου.

$$\forall x, y, z \in \mathbb{G}. \mathtt{m}(x)(\mathtt{m}(y)(z)) = \mathtt{m}(\mathtt{m}(x)(y))(z)$$
$$\forall x \in \mathbb{G}. x = \mathtt{m}(x)(\mathtt{e})$$
$$\forall x \in \mathbb{G}. x = \mathtt{m}(\mathtt{e})(x)$$

Επεκτείνοντας την ιδέα από τα μονοειδή στις ομάδες χρειαζόμαστε για κάθε στοιχείο το αντίστροφό του. Δηλαδή μία ακόμα συνάρτηση $\texttt{inv} : \mathbb{G} \to \mathbb{G}$ για την οποία να ισχύουν οι νόμοι των αντιστρόφων:

$$\forall x \in \mathbb{G}.\texttt{m}(x)(\texttt{inv}(x)) = \texttt{e}$$
$$\forall x \in \mathbb{G}.\texttt{m}(\texttt{inv}(x))(x) = \texttt{e}$$

Ο τρόπος με τον οποίο μεταφέρουμε την παραπάνω κατασκευή στο πλαίσιο της θεωρίας τύπων είναι μέσω ενός επαγωγικού τύπου υψηλότερων διαστάσεων. Η καθεμία από τις συναρτήσεις που περιγράψαμε (η, m, e, inv) είναι και ένας κατασκευαστής αντικειμένων στον νέο τύπο. Οι ισότητες που απαιτήσαμε να ικανοποιούν αυτές οι συναρτήσεις αποτελούν τους κατασκευαστές μονοπατιών μιας διάστασης του επαγωγικού ορισμού. Τέλος, για να είναι το αποτέλεσμα ομάδα και ειδικότερα σύνολο, θέλουμε έναν κατασκευαστή δεύτερης διάστασης που να κάνει τον τελικό τύπο h-Set. Στην Cubical Agda ο ορισμός είναι ο ακόλουθος. Σημ.: ο τύπος του μονοπατιού μεταξύ δύο σημείων στον ίδιο τύπο συμβολίζεται και ως $x \equiv y$ στην Cubical Agda.

```
data FreeGroup (A : Type ℓ) : Type ℓ where
  η     : A → FreeGroup A
  m     : FreeGroup A → FreeGroup A → FreeGroup A
  e     : FreeGroup A
  inv   : FreeGroup A → FreeGroup A
  assoc : ∀ x y z → m x (m y z) ≡ m (m x y) z
  idr   : ∀ x → x ≡ m x e
  idl   : ∀ x → x ≡  m e x
  invr  : ∀ x → m x (inv x) ≡ e
  invl  : ∀ x → m (inv x) x ≡ e
  trunc : ∀ x y → ∀ (p q : x ≡ y) → p ≡ q
```

Είναι πραγματικά εντυπωσιακός ο τρόπος με τον οποίο ορίσαμε ένα αντικείμενο περιγράφοντας τις ιδιότητες που θέλουμε να ικανοποιεί. Αυτός ο τρόπος ορισμού αντικειμένων είναι πολύ συχνός στη θεωρία Κατηγοριών, μέσω δηλαδή της περιγραφής μιας καθολικής ιδιότητας για το οριζόμενο αντικείμενο.

Αποδεικνύεται ότι για τον ορισμό ενός ομομορφισμού με τύπο $\texttt{freeGroup}\ A \to G$ αρκεί ο καθορισμός της συμπεριφοράς της συνάρτησης πάνω στα στοιχεία του A, ή πιο σωστά πάνω στα στοιχεία της μορφής $\eta(a) : \texttt{freeGroup}\ A$ για $a : A$. Αυτή η απεικόνιση αποτελεί μια ισοδυναμία τύπων με αντίστροφη συνάρτηση τον περιορισμό ενός ομομορφισμού στα στοιχεία $\eta(a) : \texttt{freeGroup}\ A$. Από Univalence Axiom παίρνουμε τελικά ότι οι δύο τύποι είναι ίσοι:

$$(A \to G) = \texttt{GroupHom}\ (\texttt{freeGroup}\ A)\ G$$

Τέλος, για την απόδειξη μιας ιδιότητας (h-Prop) για κάθε στοιχείο της ελεύθερα παραγόμενης ομάδας αρκεί να έχουμε τις εξής 4 αποδείξεις:

- Απόδειξη της πρότασης για κάθε στοιχείο $\eta(a)$, αν $a : A$
- Απόδειξη της πρότασης για το στοιχείο $\texttt{m}(g_1)(g_2)$ δεδομένων των αποδείξεων των ιδιοτήτων για τα $g_1, g_2 : \texttt{freeGroup}\ A$
- Απόδειξη ότι το στοιχείο e ικανοποιεί την ιδιότητα
- Απόδειξη της πρότασης για το στοιχείο $\texttt{inv}(g)$ δεδομένης της απόδειξης για το $g : \texttt{freeGroup}\ A$

Ο τύπος της αντίστοιχης συνάρτησης επαγωγής στην υλοποίησή μας δίνεται ως:

```
elimProp : {B : FreeGroup A → Type ℓ'}
         → ((x : FreeGroup A) → isProp (B x))
         → ((a : A) → B (η a))
         → ((g1 g2 : FreeGroup A) → B g1 → B g2 → B (m g1 g2))
         → (B e)
         → ((g : FreeGroup A) → B g → B (inv g))
         → (x : FreeGroup A)
         → B x
```

Παρατηρούμε ότι δεν χρειάζεται να ασχοληθούμε με τον τρόπο που μεταφέρονται από την επαγωγή τα αντικείμενα κατασκευαστών μεγαλύτερων διαστάσεων. Ο λόγος που αυτά τα δεδομένα μόνο αρκούν για την απόδειξη της ιδιότητας για κάθε στοιχείο της ελεύθερης ομάδας, είναι ότι όταν η ιδιότητα είναι ένας τύπος με τετριμμένα μονοπάτια (h-Prop) όλες οι αποδείξεις είναι αυτόματα ίσες μεταξύ τους και απλώς αρκεί η ύπαρξη έστω μίας.

5.2 FreeGroupoid of A

Προκειμένου να κάνουμε όσο το δυνατόν πιο γενική την απόδειξή μας και να αφαιρέσουμε επιπλέον υποθέσεις εισάγουμε έναν νέο τύπο με βάση τον τύπο A παρόμοιο με την ελεύθερη ομάδα αφαιρώντας τον οποιονδήποτε περιορισμό για το truncation level. Με άλλα λόγια, στον επαγωγικό ορισμό δεν υπάρχει ο κατασκευαστής `trunc` ο οποίος εξανάγκαζε τη δομή σε h-Set.

```
data FreeGroupoid (A : Type ℓ) : Type ℓ where
  η     : A → FreeGroupoid A
  m     : FreeGroupoid A → FreeGroupoid A → FreeGroupoid A
  e     : FreeGroupoid A
  inv   : FreeGroupoid A → FreeGroupoid A
  assoc : ∀ x y z → m x (m y z) ≡ m (m x y) z
  idr   : ∀ x → x ≡ m x e
  idl   : ∀ x → x ≡  m e x
  invr  : ∀ x → m x (inv x) ≡ e
  invl  : ∀ x → m (inv x) x ≡ e
```

Αν επαναφέρουμε εκ των υστέρων αυτούς τους περιορισμούς δημιουργώντας το set truncation του `freeGroupoid` A τότε το αποτέλεσμα είναι ομοτοπικά ισοδύναμο με την ελεύθερη ομάδα του A. Επομένως, οποιαδήποτε στιγμή επιθυμούμε να προσθέσουμε αυτήν την υπόθεση αρκεί απλά να θεωρήσουμε ότι πλέον εργαζόμαστε στην ελεύθερη ομάδα του A. Αυτό μας παρέχεται μέσω του Univalence Axiom και της επόμενης πρότασης της βιβλιοθήκης μας

```
freeGroupTruncIdempotent : FreeGroup A ≡ ∥ FreeGroupoid A ∥₂
```

Στην Cubical Agda έχει επιλεγεί η ιεραρχία των ομοτοπικών τύπων να ξεκινά από το 0 και όχι από το -2 και για αυτό το set truncation θεωρείται επίπεδο 2.

Για την απόδειξη του θεωρήματος θα χρησιμοποιήσουμε μια αρκετά διαδεδομένη τεχνική που ονομάζεται encoding-decoding. Το πρώτο βήμα σε αυτή την τεχνική είναι η δημιουργία ομοτοπικών ισοδυναμιών του `freeGroupoid` A με τον εαυτό του, για κάθε g : `freeGroupoid` A. Οι συναρτήσεις που θα χρησιμοποιήσουμε (και ονομάζουμε αυτομορφισμούς του `freeGroupoid` A) ορίζονται ως εξής

```
automorhpism : ∀ (a : FreeGroupoid A) → FreeGroupoid A → FreeGroupoid A
automorhpism a g = m g a
```

Ο εξοικειωμένος με τη Θεωρία Ομάδων αναγνώστης θα αναγνωρίσει ότι αυτό δεν είναι άλλο από τη δράση της «ομάδας» `freeGroupoid` A πάνω στον εαυτό της, αντιστρέφοντας όμως την πράξη από $\mathtt{m}(a)(g)$ σε $\mathtt{m}(g)(a)$. Αυτό έχει ως συνέπεια αυτοί οι αυτομορφισμοί να συμπεριφέρονται καλά ως προς τις διάφορες πράξεις του `freeGroupoid` A. Αποδεικνύουμε ότι η δράση του γινομένου είναι η σύνθεση των δράσεων των δύο παραγόντων (με αντίθετη σειρά) και ότι η δράση του ουδετέρου είναι η ταυτότική συνάρτηση.

Το επόμενο βήμα είναι η απόδειξη ότι αυτοί οι αυτομορφισμοί είναι όντως ομοτοπικές ισοδυναμίες, για κάθε g : `freeGroupoid` A. Και σε αυτό το βήμα έχουμε ότι η συνολική δομή σέβεται της πράξεις που έχουμε ορίσει. Πιο συγκεκριμένα, έχουμε ότι η ισοδυναμία του γινομένου είναι ίση με τη σύνθεση των ισοδυναμιών που δημιουργούνται από τους δύο παράγοντες, η ισοδυναμία του ουδετέρου είναι η ταυτοτική ισοδυναμία και η ισοδυναμία του αντιστρόφου ενός στοιχείου είναι η αντίστροφη ισοδυναμία του στοιχείου.

Τελευταίο βήμα είναι η χρήση του Univalence Axiom για να δημιουργήσουμε στο σύμπαν ένα μονοπάτι από το `freeGroupoid` A στο `freeGroupoid` A, από τις ισοδυναμίες, για κάθε g : `freeGroupoid` A. Και σε αυτό το βήμα η δομή ομάδας διατηρείται και μεταφέρεται στην ομάδα των μονοπατιών με αρχή και τέλος τον τύπο `freeGroupoid` A στο σύμπαν. Έχουμε αποδείξει δηλαδή τα εξής:

```
multPathsInUNaturality : ∀ (g1 g2 : FreeGroupoid A) →
    pathsInU (m g1 g2) ≡ (pathsInU g1) • (pathsInU g2)
idPathsInUNaturality : pathsInU {A = A} e ≡ refl
invPathsInUNaturality : ∀ (g : FreeGroupoid A) →
    pathsInU (inv g) ≡ sym (pathsInU g)
```

όπου pathsInU είναι η συνάρτηση που αντιστοιχεί σε κάθε στοιχείο το μονοπάτι στο $\mathcal{U}$ από την ισοδυναμία.

5.3 FundamentalGroup

Όπως στην Αλγεβρική Τοπολογία, αν έχουμε έναν Τύπο και ένα σημείο του, είναι δυνατόν να κατασκευάσουμε τον Τύπο των μονοπατιών με αρχή και τέλος το σημείο αυτό. Αυτό είναι φυσικά ο γνωστός τύπος `Path` A `base base` τον οποίο συμβολίζουμε στην απόδειξη ως ΩA θεωρώντας ότι το σημείο της βάσης είναι γνωστό. Στην περίπτωση, όμως, της Ομοτοπικής θεωρίας Τύπων, ο τύπος ΩA δεν είναι ομάδα αλλά higher groupoid. Για να γίνει ομάδα παίρνουμε το set truncation κι έτσι ορίζουμε τη Θεμελιώδη ομάδα του A που συμβολίζουμε με π_1. Αποδεικνύεται ότι το π_1 είναι όντως ομάδα με τις πράξεις που επάγονται από το groupoid ΩA.

```
1π₁ : π₁ {base = base}
1π₁ = ∣ refl ∣₂

invπ₁ : π₁ {base = base} → π₁
invπ₁ = map sym

•-π₁ : π₁ {base = base} → π₁ → π₁
•-π₁ = rec2 π₁IsSet (λ p q → ∣ p • q ∣₂)
```

Παρόμοια με την περίπτωση των `freeGroup`, `freeGroupoid`, θα αφαιρέσουμε από τις υποθέσεις μας όλους τους περιορισμούς για το h-level της θεμελιώδους ομάδας και θα ασχοληθούμε αρχικά μόνο με το groupoid ΩA.

5.4 Απόδειξη

Αρχικά, ορίζουμε στη θεωρία μας τον τύπο του Bouquet από κύκλους με δείκτες από έναν τύπο A, ή αλλιώς «wedge of A circles», με τον εξής ορισμό ως επαγωγικό τύπο υψηλότερων διαστάσεων.

```
data Bouquet (A : Type ℓ) : Type ℓ where
  base : Bouquet A
  loop : A → base ≡ base
```

Παρατηρούμε την ομοιότητα που υπάρχει μεταξύ του ορισμού αυτού και του ορισμού του τύπου του κύκλου. Σε αυτήν την περίπτωση ωστόσο, υπάρχει ένας ακόμη παράγοντας που επηρεάζει την κατασκευή, η δομή μονοπατιών του A. Επειδή το `loop` εδώ είναι συνάρτηση, μεταφέρει μονοπάτια του A σε μονοπάτια του `Bouquet`A σε όλες τις διαστάσεις. Αυτό περιπλέκει ενδεχομένως τη δομή μονοπατιών του `Bouquet`A.

Ορίζουμε έπειτα τις συναρτήσεις οι οποίες θα αποτελέσουν τη βάση της ομοτοπικής ισοδυναμίας. Αγνοώντας τους περιορισμούς των truncation ορίζουμε τη συνάρτηση looping από το `freeGroupoid` A στο Ω`Bouquet` με πολύ φυσικό τρόπο σεβόμενοι την αλγεβρική δομή των δύο τύπων.

```
looping : FreeGroupoid A → ΩBouquet
looping (η a)             = loop a
looping (m g1 g2)         = looping g1 • looping g2
looping e                 = refl
looping (inv g)           = sym (looping g)
looping (assoc g1 g2 g3 i) = pathAssoc (looping g1) (looping g2) (looping g3) i
looping (idr g i)         = rUnit (looping g) i
looping (idl g i)         = lUnit (looping g) i
looping (invr g i)        = rCancel (looping g) i
looping (invl g i)        = lCancel (looping g) i
```

Για τη συνάρτηση winding με αντίθετη κατεύθυνση, από το Ω`Bouquet` στο `freeGroupoid` A θα αξιοποιήσουμε τη δουλειά που κάναμε με τους αυτομορφισμούς του `freeGroupoid` A. Κατασκευάζουμε μια οικογένεια τύπων με δείκτες στο `Bouquet`A, δηλαδή μια συνάρτηση `Bouquet`$A \to \mathcal{U}$. Σύμφωνα με τον κανόνα αναδρομής για αυτόν τον επαγωγικό τύπο για να ορίσουμε σωστά μια συνάρτηση με πεδίο ορισμού το `Bouquet`A πρέπει να δώσουμε ένα αντικείμενο $K : \mathcal{U}$ στο οποίο να αντιστοιχίζεται το `base` : `Bouquet`A και για κάθε μονοπάτι $a : A \vdash$ `loop`(a) : `Path Bouquet`A `base base` να δώσουμε ένα μονοπάτι με τύπο `Path` $\mathcal{U}$ K K.

Ορίζουμε την οικογένεια code να είναι η οικογένεια που στέλνει το base στον τύπο `freeGroupoid` A και το κάθε μονοπάτι $a : A \vdash$ `loop`(a) : `Path Bouquet`A `base base` στο μονοπάτι που προκύπτει από τον αυτομορφισμό του $\eta(a)$. Ορίζουμε τη συνάρτηση winding να στέλνει κάθε μονοπάτι του Ω`Bouquet` στο αντικείμενο του `freeGroupoid` A στο οποίο μεταφέρεται το ουδέτερο στοιχείο e πάνω από αυτό το μονοπάτι στην οικογένεια τύπων code.

```
code : {A : Type ℓ} → (Bouquet A) → Type ℓ
code {A = A} base = (FreeGroupoid A)
code (loop a i)    = pathsInU (η a) i

winding : ΩBouquet → FreeGroupoid A
winding l = subst code l e
```

Για να ολοκληρωθεί η απόδειξη της ισοδυναμίας χρειαζόμαστε δύο ακόμα κατασκευές, την απόδειξη ότι η συνάρτηση looping είναι αριστερή αντίστροφη με τύπο

```
left-homotopy : ∀ (l : ΩBouquet {A = A}) → looping (winding l) ≡ l
```

και μια απόδειξη ότι είναι η δεξιά αντίστροφη

```
right-homotopy : ∀ (g : FreeGroupoid A) → winding (looping g) ≡ g
```

Για να μπορέσουμε να αποδείξουμε το πρώτο με επαγωγή μονοπατιού, απαιτείται τουλάχιστον ένα από τα δύό άκρα του μονοπατιού l : ΩBouquet να μην είναι σταθερό. Κάνουμε μια αφαίρεση, λοιπόν, στο συμπέρασμά μας και υποθέτουμε ότι έχουμε x : `Bouquet` $A \vdash l$: `Path Bouquet` A `base` x και άρα το μονοπάτι στο $\mathcal{U}$ έχει τύπο `Path` $\mathcal{U}$ (`freeGroup` A) `code`(x). Εν συνεχεία, προσαρμόζουμε τις συναρτήσεις μας. Η winding γενικεύεται στην encode με τύπο

```
encode : (x : Bouquet A) → base ≡ x → code x
```

και η looping γενικεύεται στην decode.

```
decode : {A : Type ℓ}(x : Bouquet A) → code x → base ≡ x
```

Πλέον, οι ζητούμενη απόδειξη έχει τύπο

```
decodeEncode : (x : Bouquet A) → (p : base ≡ x) → decode x (encode x p) ≡ p
```

Τώρα, η απόδειξη του decodeEncode γίνεται με επαγωγή μονοπατιού στο p : `base` $\equiv x$ και εφαρμόζοντας στην απόδειξη `base` όπου έχει x παίρνουμε το ένα ζητούμενο

```
left-homotopy : ∀ (l : ΩBouquet {A = A}) → looping (winding l) ≡ l
left-homotopy l = decodeEncode base l
```

Για να αποδείξουμε ότι η συνάρτηση looping είναι και δεξιά αντίστροφη της winding θα πρέπει να επαναφέρουμε τους περιορισμούς στη δομή μονοπατιών των δύο χώρων και να περιορίσουμε και το συμπέρασμά μας σε αυτούς. Ειδικότερα, δεν αποδείξαμε το ζητούμενο right-homotopy αλλά το propositional truncation αυτού.

```
truncatedRight-homotopy : (g : FreeGroupoid A) → ∥ winding (looping g) ≡ g ∥
```

Διαισθητικά, αποδείξαμε πως όντως ισχύει ότι η looping είναι δεξία αντίστροφη της winding αλλά εκφυλίζοντας το συμπέρασμα σε έναν χώρο με πολύ απλούστερη δομή μονοπατιών. Αυτό και μόνο αρκεί για να πάρουμε την ισοδυναμία των set truncations των γενικών χώρων `base` $\equiv x$ και `code`(x)

```
TruncatedFamiliesEquiv : (x : Bouquet A) → ∥ base ≡ x ∥₂ ≃ ∥ code x ∥₂
```

και από αυτό με χρήση του Univalence Axiom και εφαρμογή του `base` για x : `Bouquet` A έχουμε

```
π₁Bouquet≡∥FreeGroupoid∥₂ : π₁Bouquet ≡ ∥ FreeGroupoid A ∥₂
```

Όμως, έχουμε ήδη δείξει ότι το set truncation του `freeGroupoid` A είναι η ελεύθερη ομάδα του A και τελικά παίρνουμε το ζητούμενο

```
π₁Bouquet≡FreeGroup : {A : Type ℓ} → π₁Bouquet ≡ FreeGroup A
```

Κεφάλαιο 6

Μελλοντική έρευνα

Σε αυτό το κεφάλαιο παραθέτουμε ορισμένα ανοιχτά προβλήματα και πεδία έρευνας σχετικά με το θεώρημα της εργασίας αλλά και την Ομοτοπική Θεωρία Τύπων γενικότερα.

Ενώ είναι σαφές ποια είναι η θεμελιώδης ομάδα του Bouquet από κύκλους -ήδη από την αλγεβρική τοπολογία- όταν θέσουμε περιορισμούς για την ανώτερη δομή μονοπατιών του τύπου, δεν είναι γνωστό τι γίνεται όταν δεν θέσουμε κανέναν τέτοιο περιορισμό. Για μια πιο ακριβή διατύπωση, ποιος είναι ο χώρος μονοπατιών του $\texttt{Bouquet}A$, δηλαδή του $\Omega(\texttt{Bouquet}A, \texttt{base})$, και ποια είναι η συμπεριφορά του σε κάθε διάσταση; Μάλιστα, δεν γνωρίζουμε καν αν ο τύπος $\Omega(\texttt{Bouquet}A, \texttt{base})$ είναι h-Set ή όχι. Τέτοια ζητήματα δεν είχαν ανακύψει στη μελέτη αυτών των χώρων μέσα στη θεωρία συνόλων όπου γίνεται η κλασική αλγεβρική τοπολογία αλλά μόνο μέσα από το πρίσμα της ερμηνείας των χώρων ως ανώτερες κατηγορίες και την Ομοτοπική Θεωρία Τύπων.

Αντίστοιχα, υπάρχουν πολλά προβλήματα που έχουν τις ρίζες τους στην αλγεβρική τοπολογία και μελετώνται στην Ομοτοπική Θεωρία Τύπων. Ένα από τα πιο κοινά είναι ο υπολογισμός των θεμελιωδών ομάδων διαφόρων διαστάσεων της σφαίρας $\mathbb{S}^n$ για διάφορα n. Αυτού του είδους οι υπολογισμοί, είναι στην ίδια κατηγορία με το προηγούμενο. Αναρωτιέται, λοιπόν, κανείς αν υπάρχει ένας πιο ευκολος τρόπος υπολογισμού του χώρου μονοπατιών ενός χώρου που να μην χρησιμοποιεί τη μέθοδο encode-decode και που να είναι αρκετά γενικός. Ακόμη, θα μπορούσε κανείς να μελετήσει τη δομή επιφανειών οι οποίες δεν είναι προσανατολίσιμες, όπως για παράδειγμα το προβολικό επίπεδο, το μπουκάλι του Klein ή η λωρίδα του Möbius. Πώς μπορεί να εκφραστεί στο πλαίσιο της Ομοτοπικής Θεωρίας Τύπων η έννοια του προσανατολισμού μιας επιφάνειας ή ακόμα και η ίδια η έννοια της επιφάνειας;

Επιπρόσθετα, είδαμε ότι η Cubical Type Theory δίνει ένα υπολογιστικό περιεχόμενο στη θεωρία μας. Ωστόσο, υπάρχουν ακόμα πολλά ενδιαφέροντα σημεία της μεταθεωρίας της HoTT. Η ίδια η Cubical Type Theory είναι σημαντικό πεδίο έρευνας αυτήν τη στιγμή καθώς και άλλες παρόμοιες θεωρίες τύπων με παρόμοια λογική. Επίσης, αν και δώσαμε μια διαισθητική παρουσίαση των επαγωγικών τύπων, υπάρχει συντακτικά τυπικός τρόπος ορισμού τους. Είναι άγνωστο προς το παρόν αν υπάρχει αντίστοιχος αυστηρός ορισμός των επαγωγικών τύπων υψηλότερων διαστάσεων. Ένα άλλο ζήτημα είναι το κατά πόσο είναι δυνατό να κατασκευαστεί η HoTT μέσα στον ίδιο της τον εαυτό! Δηλαδή πώς γίνεται να μελετήσουμε το συντακτικό της Ομοτοπικής Θεωρίας Τύπων μέσα στην HoTT. Το θέμα αυτό συχνά ονομάζεται και «type theory eating itself» ή «type theory in type theory».

Ένα άλλο πολύ σημαντικό πεδίο έρευνας είναι το πεδίο των semantics. Ένα κύριο ζήτημα της περιοχής αυτής είναι το Initiality Conjecture. Υποθέτουμε ότι υπάρχει ένας τρόπος περιγραφής όλων των «γλωσσών» ενός συγκεκριμένου είδους, ένα signature καθώς και ένας τρόπος περιγραφής όλων των «μοντέλων» αυτής της γλώσσας σε μία κατηγορία, με τους μορφισμούς μεταξύ τους να είναι «ερμηνείες» της μίας γλώσσας εντός της άλλης. Η εικασία λέει ότι υπάρχει ένα μοντέλο, το μοντέλο των όρων του συντακτικού της γλώσσας, το οποίο είναι αρχικό αντικείμενο στην κατηγορία των μοντέλων. Η εικασία έχει αποδειχτεί για γλώσσες όπως το calculus of constructions και ο λ-λογισμός αλλά όχι για πιο σύνθετες γλώσσες όπως η Εξαρτημένη Θεωρία Τύπων ή η Ομοτοτική Θεωρία Τύπων. Με αυτόν τον τρόπο είναι δυνατόν να μελετηθούν πολλές γλώσσες μαζί και να απαντηθούν ερωτήματα σχετικά με το normalization ή θεωρήματα πληρότητας για αυτές.

Τέλος, κρίσιμο για την καθημερινή μαθηματική πρακτική είναι η συγγραφή πλούσιων βιβλιοθηκών με ό,τι είναι ήδη γνωστό σε όλους τους κλάδους των μαθηματικών με τη βοήθεια των proof assistants. Ενδεικτικά, αναφέρουμε τις βιβλιοθήκες UniMath και HoTT σε Coq και τις HoTT-Agda και Cubical Agda σε Agda και Cubical Agda αντίστοιχα.

Βιβλιογραφία

[Baue16] A. Bauer, “Five stages of accepting constructive mathematics”, *Bulletin of the American Mathematical Society*, vol. 54, pp. 481–498, 2016.

[Cohe16] Cyril Cohen, Thierry Coquand, Simon Huber and Anders Mörtberg, “Cubical Type Theory: a constructive interpretation of the univalence axiom”, 2016.

[Fral03] J.B. Fraleigh and V.J. Katz, *A First Course in Abstract Algebra*, Addison-Wesley series in mathematics, Addison-Wesley, 2003.

[Harp12] Robert Harper, “Extensionality, Intensionality, and Brouwer’s Dictum”, http://existentialtype.wordpress.com/2012/08/11/extensionality-intensionality-and-brouwers-dictum/, August 2012.

[Hatc02] A. Hatcher, Cambridge University Press and Cornell University. Department of Mathematics, *Algebraic Topology*, Algebraic Topology, Cambridge University Press, 2002.

[Krau19] Nicolai Kraus and Jakob von Raumer, “Path Spaces of Higher Inductive Types in Homotopy Type Theory”, 2019.

[Lica13] Daniel R. Licata and Michael Shulman, “Calculating the Fundamental Group of the Circle in Homotopy Type Theory”, 2013.

[MacL71] Saunders MacLane, *Categories for the Working Mathematician*, Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.

[Mart72] Per Martin-Löf, “An intuitionistic theory of types”, 1972.

[Munk00] J.R. Munkres, *Topology*, Featured Titles for Topology, Prentice Hall, Incorporated, 2000.

[Univ13] The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[Vezz19] Andrea Vezzosi, Anders Mörtberg and Andreas Abel, “Cubical Agda: A Dependently Typed Programming Language with Univalence and Higher Inductive Types”, *Proc. ACM Program. Lang.*, vol. 3, no. ICFP, jul 2019.

[Αν09] Α. Διονύσιος Αναπολιτάνος, *Εισαγωγή στη Φιλοσοφία των Μαθηματικών*, Εκδόσεις Νεφέλη, 2009.

Παράρτημα Α

Ευρετήριο συμβόλων

Γ : Περιβάλλον Τύπων
$\Gamma \vdash$: Στο περιβάλλον τύπων αποδεικνύεται
$\equiv$: Judgmental Equality
$:\equiv$: Ορισμός
$::=$: Ορισμός Τυπικής Γραμματικής
`id` : Ταυτότική Συνάρτηση
`type` : Τύπος
$\circ$: Σύνθεση Συναρτήσεων
1 : Τύπος Unit
$\langle\rangle$: Τετριμμένο στοιχείο του Τύπου Unit
0 : Κενός Τύπος
`abort` : Αναδρομή στην Κενό Τύπο
`fst` : Πρώτη προβολή ζεύγους
`snd` : Δεύτερη προβολή ζεύγους
$\langle a, b\rangle$: Ζεύγος στοιχείων
$A \to B$: συνάρτηση από το πεδίο A στο πεδίο B
`inl` : Πρώτη συ-προβολή συγινομένου
`inr` : Δεύτερη συ-προβολή συγινομένου
`case` : Συνάρτηση απαλοιφής συγινομένου
`bool` : Τύπος Bool
`tt` : True
`ff` : False
`if` M `then` N_1 `else` N_2 : Απαλοιφή Τύπου Bool
`rec` : Αναδρομή
`ind` : Επαγωγή
`zero` : Το 0 των φυσικών αριθμών
`suc` : Η συνάρτηση του επόμενου των φυσικών αριθμών
$\mathbb{N}$, `nat` : Ο τύπος των φυσικών αριθμών
`Seq` : Ο τύπος των πεπερασμένων ακολουθιών φυσικών αριθμών
`list`(A) : Τύπος συνδεδεμένων λιστών με στοιχεία από τον τύπο A
`nil` : κενή λίστα
x `cons` xs : Κατασκευαστής λίστας cons
`Id` : Τύπος Ισότητας
`refl` : Αυτοπαθές μονοπάτι ισότητας
`J` : Επαγωγή μονοπατιού
`sym` : Αντίστροφο μονοπάτι
`trans` : Συνένωση μονοπατιών
`ap` : Εφαρμογή συνάρτησης σε μονοπάτι
`tr` : Μεταφορά
`dap` : Εφαρμογή εξαρτημένης συνάρτησης σε μονοπάτι
$\bullet$: Συνένωση μονοπατιών

$\mathbb{I}$: Τύπος διαστήματος
$\mathbb{S}^1$: Τύπος Κύκλου
`base` : Βάση του Τύπου του Κύκλου
`loop` : Μη τετριμμένο μονοπάτι στον Τύπο του Κύκλου με αρχή και τέλος το `base`
$\mathcal{U}$: Σύμπαν Τύπων
`ua` : Univalence Axiom
`ua-equiv` : Ισοδυναμία Τύπων από ισότητα που αποδείχθηκε με το `ua`
`funext` : Function Extensionality
`happly` : Ομοτοπία μεταξύ ίσων συναρτήσεων
$\simeq$: Ισοδυναμία Τύπων - Ομοτοπία Συναρτήσεων
isEquiv : Τύπος αποδείξεων ότι μια συνάρτηση είναι ισοδυναμία τύπων
`idtoeqv` : Ισοδυναμία ίσων τύπων
`idEquiv` : Ταυτοτική ισοδυναμία
$\circ_{\simeq}$: Σύνθεση ισοδυναμιών
`equiv-inv` : Αντίστροφη ισοδυναμία
`equiv-compose` : Σύνθεση ισοδυναμιών
$\|A\|$: Propositional Truncation
$|x|$: Propositional Truncation projection
`squash` : Propositional Truncation κατασκευαστής ισότητας
$\|A\|_0$: Set Truncation
$|x|_0$: Set Truncation projection
$\texttt{squash}_0$: Set Truncation κατασκευαστής ισότητας
`Path` $A\ x\ y$: Τύπος μονοπατιών στον τύπο A μεταξύ των $x, y : A$
$\sim x$: Involution στην $\mathbb{I}$
$x \vee y$: Join στην $\mathbb{I}$
$x \wedge y$: Meet στην $\mathbb{I}$
`i0` : Ελάχιστο στοιχείο στην $\mathbb{I}$
`i1` : Μέγιστο στοιχείο στην $\mathbb{I}$
`hcomp` : Ομογενής Σύνθεση
`transp` : Μεταφορά
`transport` : Μεταφορά
$0_{\mathbb{F}}$: Ελάχιστο μερικό στοιχείο στα partial ellements
$1_{\mathbb{F}}$: Μέγιστο μερικό στοιχείο στα partial ellements
`Glue` : Τύπος Glue
`glue` : Κατασκευαστής glue
`unglue` : Συνάρτηση απαλοιφής unglue
$\mathcal{O}$: Σύνολο αντικειμένων μιας κατηγορίας
$\mathcal{A}$: Σύνολο μορφισμών μιας κατηγορίας
$\mathcal{C}$: Κατηγορία
$\texttt{dom}f$: Πεδίο ορισμού της f
$\texttt{cod}f$: Πεδίο τιμών της f
$\texttt{Hom}(a, b)$: Σύνολο μορφισμών με πεδίο ορισμού το a και πεδίο τιμών το b
$\texttt{freeGroup}A$: Ελεύθερα Παραγόμενη Ομάδα από στοιχεία του τύπου A
$\texttt{freeGroupoid}A$: Ελεύθερα Παραγόμενο groupoid από στοιχεία του τύπου A
`m` : Συνάρτηση πολλαπλασιασμού
`e` : Ουδέτερο στοιχείο
`inv` : Αντίστροφο στοιχείο
`assoc` : Απόδειξη προσεταιριστικότητας
`idr` : Απόδειξη δεξιού νόμου ουδετέρου στοιχείου
`idl` : Απόδειξη αριστερού νόμου ουδετέρου στοιχείου
`invr` : Απόδειξη δεξιού νόμου αντίστροφου στοιχείου

`invl` : Απόδειξη αριστερού νόμου αντίστροφου στοιχείου
`trunc` : Απόδειξη ισότητας για το Set Truncation της ομάδας
`Bouquet` A : Bouquet από κύκλους από στοιχεία του A