2024

ΕΘΝΙΚΟ

ΜΕΤΣΟΒΙΟ

ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΕΦΑΡΜΟΓΗ ΕΡΓΑΛΕΙΩΝ ΑΛΓΕΒΡΙΚΗΣ ΣΗΜΕΙΩΤΙΚΗΣ ΣΤΗΝ ΕΝΝΟΙΟΛΟΓΙΚΗ ΜΙΞΗ ΜΕ ΧΡΗΣΗ PYTHON

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΕΤΟΒΙΤΣ ΒΑΣΙΛΙΚΗ

ΕΠΙΒΛΕΠΩΝ

Π.ΣΤΕΦΑΝΕΑΣ,ΑΝ. ΚΑΘΗΓΗΤΗΣ ΕΜΠ

Στους ανθρώπους που βαδίσαμε πλάι, και συνεχίζουμε ακόμα να βαδίζουμε μαζί

Εισαγωγή-Περίληψη

Η εργασία αυτή, τοποθετείται στο ερώτημα της χρήσης μαθηματικών εργαλείων για την τυποποίηση εννοιών που προέρχονται από τις κοινωνικές ή ανθρωπιστικές επιστήμες, με σκοπό την χρήση αυτών των εννοιών σε περιβάλλοντα που απαιτούν μια τέτοια τυποποίηση/φορμαλισμό, και κυρίως την πληροφορική. Συγκεκριμένα, το θέμα της εργασίας είναι η χρήση των μαθηματικών εργαλείων που προέρχονται από τη θεωρία κατηγοριών, για τον χειρισμό εννοιών που προέρχονται από το επιστημονικό αντικείμενο της επικοινωνίας και συγκεκριμένα τη Σημειωτική: την μελέτη του πως το νόημα που επικοινωνείται, συνδέεται με τον φορέα ή τον κώδικα αυτής της επικοινωνίας. Στη συνέχεια, και αφού περιγραφούν τα κατάλληλα εργαλεία, θα χρησιμοποιηθούν κι αυτά με τη σειρά τους για την τυποποίηση της εννοιολογικής μίξης, μιας διαδικασίας που επίσης προέρχεται από τις κοινωνικές/ανθρωπιστικές επιστήμες, αυτή τη φορά από τη γλωσσολογία και την γνωσιακή επιστήμη και αφορά το πως μια πιο σύνθετη γνώση παράγεται από απλούστερες μορφές, έννοιες, ή ιδέες. Ο σκοπός που μας οδηγεί σε αυτό είναι να κατασκευάσουμε, έστω πρωτόλεια, προγραμματιστικές εφαρμογές που θα μεταφέρουν τη διαδικασία της εννοιολογικής μίξης σε υπολογιστικά περιβάλλοντα.

Στόχοι της εργασίας αυτής, ήταν, μέτα από την παρουσίαση των κατάλληλων «εργαλείων» της θεωρίας κατηγοριών, να μελετηθεί και να αναδειχθεί το αντικείμενο της αλγεβρικής σημειωτικής και να εξερευνηθούν οι δυνατότητες ως άλλη μία γέφυρα μεταξύ των μαθηματικών και των ανθρωπιστικών επιστημών. Σε αυτό το σημείο, αλλά και για να μελετήσουμε τη χρησιμότητά τους, προσπαθήσαμε να χρησιμοποιήσουμε τα εργαλεία που αναπτύχθηκαν παραπάνω ώστε να μελετήσουμε τις απόπειρες τυποποίησης διαδικασιών της ανθρώπινης σκέψης μέσα από το μοντέλο της εννοιολογικής μίξης. Έγινε απόπειρα οι μέθοδοι τυποποίησης που χρησιμοποιήθηκαν να καταλήξουν να παράγουν έργο σε προγραμματιστικό επίπεδο. Χρησιμοποιήθηκαν οι δομημένες αναπαραστάσεις γνώσης -οντολογίες και μεταφέρθηκαν, πρώτη φορά, στο προγραμματιστικό περιβάλλον της Python. Ο λόγος που επιλέχθηκε και ειχειρήθηκε η συγκεκριμένη μεταφορά είναι ο ρόλος της Python στο χειρισμό μοντέλων τεχνητής νοημοσύνης, και άρα πως δυνητικά τα εργαλεία που αναπτύχθηκαν θα μπορούσαν να δοκιμαστούν και μελλοντικά να προσφέρουν εξέλιξη στις εφαρμογές ανάλυσης κειμένου.

Το πρώτο κεφάλαιο αφορά βασικά στοιχεία της θεωρίας κατηγοριών. Κάποια από αυτά θα είναι αναγκαία στη συνέχεια της εργασίας, και άλλα είναι αναγκαία για τη συνεπή παρουσίαση των πρώτων. Εισάγεται η έννοια της κατηγορίας καθώς και τα βασικά στοιχεία που την απαρτίζουν. Τα βασικά δομικά στοιχεία της θεωρίας κατηγοριών είναι οι μορφισμοί, οι συναρτητές, και οι φυσικοί μετασχηματισμοί οι οποίοι εξετάζονται διεξοδικά σε αυτό το κεφάλαιο, όπως και οι αναγκαίες στη συνέχεια, καθολικές κατασκευές, και συγκεκριμένα ο συνεφελκυσμός (pushout) και το συν-όριο (colimit). Οι ιδιότητες τους καθώς και η μεταξύ τους αλληλεξάρτηση, αναλύονται με στόχο να προσδιοριστούν τα απαραίτητα μαθηματικά εργαλεία που θα χρησιμοποιηθούν στην ανάπτυξη του προβλήματος που θα αναλυθεί στο επόμενα κεφάλαια.

Στο δεύτερο κεφάλαιο αφορά την αλγεβρική σημειωτική και την εννοιολογική μίξη. Αρχικά αναφέρονται κάποια εισαγωγικά στοιχεία για τη σημειωτική και ισχυριζόμαστε πως η θεωρία κατηγοριών έχει τα κατάλληλα χαρακτηριστικά για να αποτελέσει το μαθηματικό εργαλείο στο οποίο θα βασιστεί η τυποποίηση κάποιων εννοιών της σημειωτικής. Γίνεται μια εις βάθος παρουσίαση αυτής της μαθηματικοποίησης, που ονομάζεται αλγεβρική σημειωτική. Στη συνέχεια εισάγεται η διαδικασία της εννοιολογικής μίξης, η οποία λειτουργεί η θεμελιώδης διαδικασία για το πείραμα που ακολουθεί. Η εννοιολογική μίξη είναι μια διαδικασία που περιγράφει με έναν σχετικά τυπικό τρόπο πως από απλούστερες έννοιες προκύπτουν συνθετότερες στην ανθρώπινη νόηση. Εμάς ο σκοπός μας θα είναι στη συνέχεια να περιγράψουμε μια αντίστοιχη υπολογιστική διαδικασία. Μέσω των μορφισμών και των pushout επιχειρείται η μαθηματικοποίηση της εννοιολογικής μίξης και η σύνθεση της με τα μαθηματικά εργαλεία που εισήχθησαν στο πρώτο κεφάλαιο.

Το τρίτο κεφάλαιο αναφέρεται στα προγραμματιστικά εργαλεία τα οποία μας είναι αναγκαία. Ξεκινάμε με τις οντολογίες στον προγραμματισμό και μελετάμε η δυνατότητά τους να υλοποιήσουν τους εννοιολογικούς χώρους και τα συστήματα σημείων της αλγεβρικής σημειωτικής, δημιουργώντας έτσι το θεωρητικό υπόβαθρο για τη χρήση των οντολογιών στην πρακτική εφαρμογή. Εξετάζεται η γλώσσα οντολογιών OWL και η συμβατότητά της με την γλώσσα προγραμματισμού Python για τις ανάγκες του «πειράματος».

Το τέταρτο κεφάλαιο εστιάζει στο ερευνητικό παράδειγμα της πολυμεσικής έκθεσης, ένα κατασκευασμένο παράδειγμα για τον πρακτικό έλεγχο των εργαλείων μας. Αναλύονται οι οντολογίες που δημιουργήθηκαν ειδικά για το σκοπό του «πειράματος», καθώς και η δομή και ο σχηματισμός των συνδέσεων μεταξύ τους.

Στο τέλος παρουσιάζεται και αναλύεται ο κώδικας που αναπτύχθηκε για τους σκοπούς του «πειράματος». Παρέχονται τα απαραίτητα δεδομένα για την υλοποίηση της πολυμεσικής έκθεσης και την εφαρμογή των εννοιών της εννοιολογικής μίξης και των οντολογιών στο «πείραμα».

Περιεχόμενα

Κεφάλαιο 1: Στοιχεία Θεωρίας Κατηγοριών

Το βασικό πλαίσιο από το οποίο θα αντλήσουμε τα μαθηματικά μας εργαλεία είναι η θεωρία κατηγοριών. Θα ξεκινήσουμε με μια βασική ανασκόπηση, η οποία θα περιλαμβάνει τους ορισμούς και τα εργαλεία που θα χρησιμοποιήσουμε στη συνέχεια, αλλά και επιπλέον θέματα που θα καταστήσουν αυτή τη παρουσίαση συνεκτική. Παρ'όλα αυτά η παρουσίαση αυτή σίγουρα δεν είναι μια πλήρης εισαγωγή στο αντικείμενο, καθώς άλλωστε δεν είναι αυτός ο σκοπός της, για κάτι τέτοιο ο αναγνώστης παραπέμπεται στη βιβλιογραφία[1].

1.Κατηγορίες

Ορισμός (Κατηγορία). Μία κατηγορία $\mathbb{C}$ *αποτελείται από*

-Μία συλλογή αντικειμένων (συμβολισμός $ob(\mathbb{C})$ *ή* $|\mathbb{C}|$*).*

-Μία συλλογή μορφισμών (ή αλλιώς και γενικότερα, βελών) που συμβολίζεται $arr(\mathbb{C})$ *ή* $mor(\mathbb{C})$

-Δύο τελεστές $dom: arr(\mathbb{C}) \to ob(\mathbb{C})$ *και* $cod: arr(\mathbb{C}) \to ob(\mathbb{C})$ *οι οποίοι αντιστοιχούν τους μορφισμούς σε διατεταγμένα ζεύγη αντικειμένων. Αν για* f *μορφισμό ισχύει* $\big(dom(f), cod(f)\big) = (x, y)$*, τότε γράφουμε* $f: x \to y$

-Ορίζεται η πράξη της σύνθεσης μεταξύ μορφισμών ως εξής: Για κάθε ζεύγος μορφισμών, έστω $f, g \in arr(\mathbb{C})$ *με* $cod(f) = dom(g)$ *υπάρχει μορφισμός* $g \circ f$ *που καλείται η σύνθεση των* f *και* g*. Για αυτό τον μορφισμό* $dom(g \circ f) = dom(f)$ *και* $cod(g \circ f) = cod(g)$*. Δηλαδή, για κάθε ζεύγος μορφισμών* $f: x \to y, g: y \to z$*, υπάρχει ο μορφισμός* $g \circ f: x \to z$*.*

-Σε κάθε αντικείμενο $x \in ob(\mathbb{C})$ *αντιστοιχεί ο ταυτοτικός μορφισμός του (συμβ.* 1_x *ή* id_x*), με* $dom(1_x) = cod(1_x) = x$*.*

---

[1] Η βιβλιογραφία για τους βασικούς ορισμούς της θεωρίας κατηγοριών είναι εκτενέστατη. Όσα υπάρχουν σε αυτό το κεφάλαιο καλύπτονται από το "Categories for the Working Mathematician" του MacLane και το nLab (https://ncatlab.org/nlab/show/HomePage). Και τα δυο θεωρούνται βασικά εργαλεία για όποιον δουλεύει με κατηγορίες.

Και ισχύουν τα παρακάτω αξιώματα:

-Ο ταυτοτικός μορφισμός είναι ουδέτερο στοιχείο (αριστερό και δεξί) για τη σύνθεση (δηλαδή αν $f: x \to y$, τότε $id_y \circ f = f = f \circ id_x$).

-Η σύνθεση είναι προσεταιριστική, δηλαδή για μορφισμούς $f: x \to y, g: y \to z, h: z \to w$, ισχύει $h \circ (g \circ f) = (h \circ g) \circ f$.

Ο παραπάνω ορισμός είναι ο ιστορικά πρώτος[2]. Θα αποφύγουμε τη συζήτηση[3] περί θεμελίων, καθώς υπερβαίνει το σκοπό της υπάρχουσας εργασίας, και θεωρούμε τα αξιώματα της κατηγορίας απ'ευθείας ερμηνευόμενα στο πλαίσιο της συνολοθεωρίας. Η συζήτηση αυτή είναι, παρόλα αυτά πλούσια και ενδιαφέρουσα. Σε ζητήματα θεμελίωσης θα αναφερθούμε παρακάτω μόνο για το ζήτημα μεγέθους.

Ισοδύναμος, και χρηστικότερος για τους σκοπούς αυτής της εργασίας, είναι ο παρακάτω:

Εναλλακτικός Ορισμός (Κατηγορία). Μία κατηγορία $\mathbb{C}$ αποτελείται από

-Μία συλλογή αντικειμένων (συμβολισμός $ob(\mathbb{C})$ ή $|\mathbb{C}|$).

-Για κάθε ζεύγος αντικειμένων $x, y \in |\mathbb{C}|$, μία συλλογή μορφισμών από το x στο y (η συλλογή συμβ. $\mathbb{C}(x, y)$ ή $hom_{\mathbb{C}}(x, y)$, και καλείται hom-set ενώ οι μορφισμοί συμβ. $f: x \to y$)

-Ορίζεται η πράξη της σύνθεσης μεταξύ βελών ως εξής: Έστω αντικείμενα $x, y, z \in |\mathbb{C}|$. Για κάθε ζεύγος μορφισμών $f: x \to y, g: y \to z$, υπάρχει μορφισμός που καλείται σύνθεσή των f, g, συμβολίζεται $g \circ f: x \to z$.

-Σε κάθε αντικείμενο x αντιστοιχεί ο ταυτοτικός μορφισμός του (συμβ. 1_x ή id_x) που ανήκει στην $\mathbb{C}(x, x)$.

Και ισχύουν τα παρακάτω αξιώματα:

-Ο ταυτοτικός μορφισμός είναι ουδέτερο στοιχείο (αριστερό και δεξί) για τη σύνθεση (δηλαδή αν $f: x \to y$, τότε $id_y \circ f = f = f \circ id_x$).

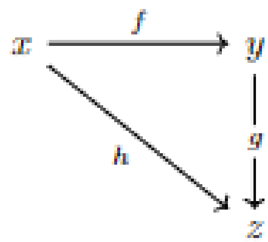
-Η σύνθεση είναι προσεταιριστική, δηλαδή για μορφισμούς $f: x \to y, g: y \to z, h: z \to w$, ισχύει $h \circ (g \circ f) = (h \circ g) \circ f$.

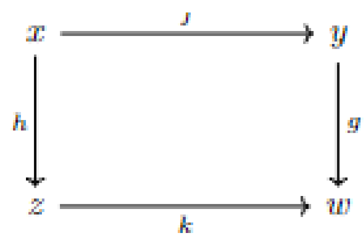
---

[2] Grothendieck σεμινάριο FGA III (1962), Eilenberg & MacLane "General theory of natural equivalences" (1945)

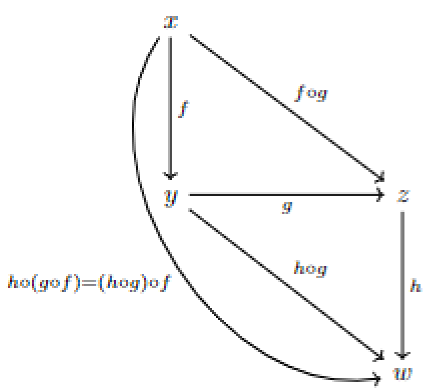
[3] Γι'αυτό και δεν ακολουθούμε την διάκριση του MacLane σε μετακατηγορία και κατηγορία.

Στην θεωρία κατηγοριών, οι μαθηματικές προτάσεις[4] γράφεται ως μία ή περισσότερες εξισώσεις μεταξύ βελών. Οι εξισώσεις αυτές αντιστοιχούν σε ένα διάγραμμα, και η ισχύς της πρότασης αντιστοιχεί με τη «μεταθετικότητα» (commutativity) του διαγράμματος, η οποία σημαίνει «όλοι οι δρόμοι είναι ίδιοι». Για παράδειγμα δείχνουμε παρακάτω δύο διαγράμματα που αντιστοιχούν στις προτάσεις $h = g \circ f$, και $g \circ f = k \circ h$.





Το παρακάτω είναι το διάγραμμα που αντιστοιχεί στο αξίωμα της προσεταιριστικότητας της σύνθεσης.



Ως παραδείγματα θα εμφανίσουμε μερικές συνήθεις μαθηματικές δομές με τη μορφή κατηγορίας:

1.Ομάδα: Μία ομάδα είναι μία κατηγορία με μόνο ένα αντικείμενο, στην οποία κάθε βέλος έχει (αριστερό και δεξί) αντίστροφο. Δηλαδή $Ob(\mathbb{C}) = \{*\}$ και για κάθε $f \in \hom_{\mathbb{C}}(*,*)$ υπάρχει $g \in \hom_{\mathbb{C}}(*,*)$ έτσι ώστε $g \circ f = 1_* = f \circ g$. Αυτός ο μορφισμός γράφεται συνήθως f^{-1}.

[4]Για το τι περιέχει η φράση «μαθηματικές προτάσεις» παραπάνω, με αυστηρό τρόπο μπορεί κανείς να ψάξει στη βιβλιογραφία την ETAC (Elementary Theory of an Abstract Category). Εδώ θα αρκεστούμε να πούμε πως ένα διάγραμμα μπορεί να αντιστοιχίσει στις έννοιες του αντικειμένου, του μορφισμού, της σύνθεσης των πεδίο/συνπεδίο σε μια εξίσωση. Αν δηλαδή ορίσουμε ένα σύνολο από φόρμουλες, δε μας αρκεί ένα διάγραμμα για να αποδείξουμε $\forall x P(x)$ ή $\exists x P(x)$, αλλά αρκεί για να δείξουμε $P(x) = Q(x)$, όταν P, Q είναι καλά-σχηματισμένες φόρμουλες.

2.Διακριτές κατηγορίες: Αποτελούνται από ένα σύνολο αντικειμένων, το ταυτοτικό βέλος κάθε αντικειμένου, και τίποτα παραπάνω.

3.Μερικές διατάξεις: Μία προδιάταξη (preorder) $(P, \leq)$ αντιστοιχεί σε κατηγορία με αντικείμενα τα στοιχεία του P και υπάρχει μορφισμός $p \to q$ αν-ν $p \leq q$. Οι ιδιότητες της σχέσης $\leq$ ισχύουν, καθώς για κάθε στοιχείο υπάρχει ο $1_p: p \to p$ που διασφαλίζει την ανακλαστικότητα, ενώ η μεταβατικότητα διασφαλίζεται από τη σύνθεση. Αν επιπλέον $p \leq q \; και \; q \leq p \; συνεπάγεται \; p = q$, τότε είναι μερική διάταξη (partial order). Προφανώς κάθε σύνολο μορφισμών μεταξύ δύο αντικειμένων είναι ή μονοσύνολο, ή κενό σύνολο.

Πίσω στον ορισμό των κατηγοριών αφήσαμε τη δυνατότητα τα αντικείμενα ή/και οι μορφισμοί να μην είναι σύνολα, αλλά κλάσεις. Αυτό συμβαίνει ώστε να μπορούν τα αξιώματά μας να ερμηνεύονται και σε περιπτώσεις μεγέθους που υπερβαίνει τα σύνολα. Για παράδειγμα για να ορίσουμε μια κατηγορία όλων των συνόλων. Προφανώς η συλλογή των αντικειμένων της δεν είναι σύνολο (βλ. παράδοξο Russel). Τυπικά λοιπόν οι κατηγορίες με σύνολα αντικειμένων και μορφισμών ονομάζονται *μικρές κατηγορίες*. Οι κατηγορίες που οι συλλογές αντικειμένων και μορφισμών δεν είναι σύνολα λέγονται *μεγάλες κατηγορίες*. Σε ενδιάμεσες περιπτώσεις η ορολογία δεν είναι ομοιόμορφη στη βιβλιογραφία, καθώς σε κάποιες περιπτώσεις κάθε κατηγορία στην οποία απλώς μία συλλογή δεν είναι σύνολο λέγεται μεγάλη, ενώ σε άλλες οι κατηγορίες με μια κλάση αντικειμένων που δεν είναι σύνολο, ενώ όλα τα hom-set μορφισμών μεταξύ διατεταγμένων ζευγών αντικειμένων είναι σύνολα καλούνται *τοπικά μικρές*. Ακόμα γενικότερα μπορεί να οριστεί ένα «σύνολο»-σύμπαν U με ορισμένες ιδιότητες[5] και κάθε αντικείμενό του να καλείται *μικρό σύνολο* (ή γενικότερα $U - μικρό \; σύνολο$). Από τα μικρά σύνολα θα κατασκευαστούν οι μικρές κατηγορίες, ενώ από το U και σύνολα που κατασκευάζονται από αυτό κατασκευάζονται οι μεγάλες κατηγορίες. Αντίστοιχα σύμαντα όπως το U είναι αντικείμενα κάποιου άλλου σύμπαντος (ας το ονομάσουμε V) για το οποίο είναι μικρά σύνολα, και ούτω καθ'εξής.

Σε κάθε περίπτωση, επιτρέποντας τις μεγάλες κατηγορίες, μας επιτρέπεται να κατασκευάσουμε κατηγορίες των συνόλων, των ομάδων, κτλ, οι οποίες αντιστοιχούν σε μαθηματικές θεωρίες. Για παράδειγμα:

---

[5] (i)$x \in u, u \in U, συνεπάγεται \; x \in U$ (ii) Αν $x \in U, y \in U$, τότε $\{x, y\} \in U$ (iii) $x \in U \; συνεπάγεται \; \mathcal{P}(U), και \cup x \in U$, αν θέλουμε το U συγκεκριμένα να αντιστοιχεί στη συνολοθεωρία έχουμε επιπλέον ότι , $(i) < x, y > , x \times y \in U, (ii) \; όλοι \; οι \; πεπερασμένοι \; διατακτικοί \; δηλ. \omega = \{0,1,2, \dots\}, \; \omega \in U \; (iii) \; για \; όλες \; τις \; επί \; συναρτήσεις \; f: a \to b, a \in U, b \subseteq U, ισχύει \; b \in U$. Η παραπάνω κατασκευή καλείται σύμπαν Grothendieck.

1. Η κατηγορία $\boldsymbol{Set}$ έχει αντικείμενα όλα τα (μικρά) σύνολα και μορφισμούς τις συναρτήσεις μεταξύ τους

2. Η κατηγορία $\boldsymbol{Grp}$ έχει αντικείμενα όλες τις (μικρές) ομάδες και μορφισμούς όλους τους ομομοφισμούς ομάδων

3. Η κατηγορία $\boldsymbol{Top}$ έχει αντικείμενα όλους τους (μικρούς) τοπολογικούς χώρους και μορφισμούς τις συνεχείς απεικονίσεις

4. Η κατηγορία $\boldsymbol{Rel}$ έχει αντικείμενα σύνολα και μορφισμούς σχέσεις ($R: x \to y$, είναι $R \subseteq x \times y$)

5. Η κατηγορία $\boldsymbol{Cat}$, η οποία είναι στη συνέχεια ζητούμενο να κατασκευάσουμε με αντικείμενα μικρές κατηγορίες, και ως βέλη τους συναρτητές (βλ. στη συνέχεια). Η κατηγορία αυτή, ως κατηγορία όλων των μικρών κατηγοριών μπορεί να κατασκευαστεί – καθώς δεν είναι η ίδια μικρή κατηγορία. Δεν μπορεί όμως να κατασκευαστεί κατηγορία όλων των κατηγοριών.

Ορισμός (Υποκατηγορία). Έστω κατηγορίες $\mathbb{C}, \mathbb{D}$, με την δεύτερη να αποτελείται από κάποια αντικείμενα και κάποιους μορφισμούς της πρώτης. Αν η $\mathbb{D}$ είναι κατηγορία, τότε καλείται υποκατηγορία της $\mathbb{C}$, και συμβολίζουμε $\mathbb{D} \hookrightarrow \mathbb{C}$. Πιο συγκεκριμένα, για να μη χρησιμοποιήσουμε τον ορισμό της κατηγορίας στη $\mathbb{D}$, μπορούμε να πούμε ότι αποτελείται από κάποια αντικείμενα και κάποιους μορφισμούς της $\mathbb{C}$, ενώ για κάθε μορφισμό, περιλαμβάνονται και τα αντίστοιχα αντικείμενα πεδίο/συνπεδίο, για κάθε αντικείμενο ο ταυτοτικός μορφισμός, και για κάθε συνθέσιμο ζεύγος μορφισμών τη σύνθεσή τους.

Ορισμός (Πλήρης Υποκατηγορία). Έστω $\mathbb{D} \hookrightarrow \mathbb{C}$. Η $\mathbb{D}$ καλείται πλήρης υποκατηγορία της $\mathbb{C}$, όταν αποτελείται από κάποια αντικείμενα της $\mathbb{C}$, αλλά από όλους τους μορφισμούς της $\mathbb{C}$ μεταξύ αυτών των αντικειμένων. Με άλλα λόγια, όταν ορίσουμε τους συναρτητές, θα λέμε πως ο συναρτητής $\mathbb{D} \to \mathbb{C}$ είναι πλήρης.

Ορισμός (Ισομορφισμός) Ένα βέλος μιας κατηγορίας $f: x \to y$ λέγεται ισομορφισμός, όταν υπάρχει βέλος $f^{-1}: y \to x$, τέτοιο ώστε $f^{-1} \circ f = 1_x$ και $f \circ f^{-1} = 1_y$. Τότε ο f^{-1}καλείται αντίστροφος του f (ή αμφίπλευρος αντίστροφος, αν ισχύει μόνο μία από τις δύο παραπάνω σχέσεις καλείται αντίστοιχα αριστερός ή δεξιός αντίστροφος). Δύο ισομορφικά αντικείμενα μπορούν να συμβολιστούν $x \cong y$, και η $\cong$ είναι σχέση ισοδυναμίας[6].

Αποδεικνύεται εύκολα ότι μπορούμε να μετασχηματίσουμε τον ορισμό της κατηγορίας σε έναν ισοδύναμο από τον οποίο απαλείφουμε τα αντικείμενα. Το ρόλο των

[6] Δηλαδή σχέση: Ανακλαστική ($x \cong x$), συμμετρική ($x \cong y$ *συνεπάγεται* $y \cong x$), μεταβατική ($x \cong y$ *και* $y \cong z$ *συνεπάγεται* $x \cong z$)

αντικειμένων μπορούν να τον παίξουν οι ταυτότητές τους. Πράγματι αν πάρουμε ένα σύνολο/ μία συλλογή από βέλη D, μία συλλογή $C \subseteq D \times D$, από συνθέσιμα ζεύγη και τη σύνθεση $\circ: C \to D$, μπορούμε αντίστοιχα να ξαναγράψουμε τα αξιώματα της κατηγορίας χωρίς να χρειαστεί να αναφερθούμε καθόλου σε αντικείμενα. Το παραπάνω αναφέρεται στη βιβλιογραφία ως "arrow - only" κατηγορία. Από τη μία είναι ισοδύναμο με τον κλασσικό ορισμό, αλλά ίσως όχι τόσο εύχρηστο, ειδικά αφού θέλουμε να ορίζουμε τα μαθηματικά αντικείμενα για να κάνουμε χρήση των κατηγοριοθεωρητικών εννοιών σε άλλους κλάδους των μαθηματικών. Από την άλλη η ύπαρξη αυτού του ορισμού και η ισοδυναμία του με τον αρχικό έχει σοβαρές συνέπειες από φιλοσοφικής άποψης. Μας αποκαλύπτει ότι υπάρχουν ιδιότητες που αναφέρονται καθαρά στη δομή των μαθηματικών θεωριών (στις σχέσεις μεταξύ αντικειμένων) και όχι σε μία εσωτερική «ουσία».[7] Δεν θα σταθούμε σε αυτή την εργασία σε αυτό το ζήτημα, αλλά η ιδέα αυτή θα επανεμφανιστεί στο επόμενο κεφάλαιο.

2. Συναρτητές και φυσικοί μετασχηματισμοί

Θέλοντας να προχωρήσουμε στον ορισμό μιας κατηγορίας (μικρών) κατηγοριών, στην οποία αναφερθήκαμε και νωρίτερα, θα πρέπει να κατασκευάσουμε το αντίστοιχο των μορφισμών μεταξύ κατηγοριών, το οποίο και θα ονομάσουμε *συναρτητή (functor)*. Γενικά οι όροι μορφισμός και συναρτητής διακρίνονται από ότι το πεδίο και το συνπεδίο του δεύτερου είναι πάντα κατηγορίες[8], καθώς μεταφράζει και αντικείμενα και βέλη. Αντίστροφα δεν έχουμε την ίδια εικόνα, καθώς η λέξη μορφισμός μπορεί να χρησιμοποιηθεί άτυπα και μεταξύ κατηγοριών ή και «μεγαλύτερων» δομών, και όταν ερμηνευθεί να δούμε ότι αναφέρεται σε ένα συναρτητή, ή σε άλλα είδη απεικονίσεων. Συνήθως, όταν θέλουμε να αναφερθούμε σε έννοιες όπως μορφισμός, συναρτητής, ή φυσικός μετασχηματισμός (που θα τον συναντήσουμε αργότερα) που αναφέρονται σε απεικονίσεις, χρησιμοποιούμε τον γενικότερο όρο *«βέλος» (arrow)*.

Ορισμός (Συναρτητής). Έστω δύο κατηγορίες $\mathbb{C}, \mathbb{D}$. *Ένας μορφισμός μεταξύ τους* $F: \mathbb{C} \to \mathbb{D}$ *καλείται συναρτητής. Αποτελείται από δύο απεικονίσεις, την απεικόνιση αντικειμένων, που απεικονίζει τα αντικείμενα της* $\mathbb{C}$ *σε αντικείμενα της* $\mathbb{D}$*, δηλαδή* $x \mapsto F(x)$, $F(x) \in ob(\mathbb{D})$, *και την απεικόνιση βελών που απεικονίζει κάθε βέλος της* $\mathbb{C}$*, έστω* $f: x \to y$, *σε βέλος της* $\mathbb{D}$ *μεταξύ των* $F-$*εικόνων του πεδίου και του συνπεδίου του* f, *δηλαδή στο* $F(f): F(x) \to F(y)$. *Επιπλέον πρέπει να ισχύει, για κάθε αντικείμενο της* $\mathbb{C}$ *πως* $1_{F(x)} = F(1_x)$, *και να «σέβεται» τη σύνθεση, δηλαδή εφόσον τα βέλη* f, g *της* $\mathbb{C}$ *είναι συνθέσιμα, τότε* $F(g \circ f) = F(g) \circ F(f)$.

[7] Αυτό το σημείο είναι και η εκκίνηση της αντίληψης του «Κατηγοριοθεωρητικού Δομισμού» στη φιλοσοφία των μαθηματικών.

[8] Η και ακόμα «βαρύτερες» δομές, όπως θα δούμε στη συνέχεια, αλλά τουλάχιστον κατηγορίες.

Ένας συναρτητής $\mathrm{F}: \mathbb{C} \rightarrow \mathbb{D}$ λέγεται *πλήρης (full),* όταν κάθε βέλος της $\mathbb{D}$ μεταξύ αντικειμένων F −εικόνων της $\mathbb{C}$, ας πούμε $g: F(x) \rightarrow F(y)$, είναι κι αυτό $F - εικόνα$ ενός βέλους της $\mathbb{C}$, στο ίδιο hom-set (δηλ $g = F(f), f: x \rightarrow y$. Αντίστοιχα, ο συναρτητής λέγεται *πιστός (faithful),* όταν δύο βέλη της $\mathbb{D}$ που είναι $F - εικόνες$ της $\mathbb{C}$ είναι ίσα αν και μόνο αν οι προεικόνες τους είναι ίσες. Δηλαδή, για κάθε $x, y \in ob(\mathbb{C})$ και για κάθε $f, g: x \rightarrow y$, ισχύει $F(f) = F(g): F(x) \rightarrow F(y)$ αν και μόνο αν $f = g$. Οι ιδιότητες αυτές ενός συναρτητή είναι τα ανάλογα του επί και του 1-1 αντίστοιχα για την απεικόνιση βελών.

Μερικοί σημαντικοί τύποι συναρτητών είναι οι ακόλουθοι:

1. Ο συναρτητής $U: \boldsymbol{Grp} \rightarrow \boldsymbol{Set}$, απεικονίζει κάθε ομάδα στο σύνολο των στοιχείων της, και κάθε ομομορφισμό ομάδων στην αντίστοιχη συνάρτηση μεταξύ συνόλων. Στην πραγματικότητα το μόνο που κάνει αυτό ο συναρτητής είναι να «ξεχνάει» ένα κομμάτι της δομής. Γι’αυτό οι συναρτητές που λειτουργούν με παρόμοι τρόπο λέγονται και *επιλήσμονες (forgetful).*
2. Για κάθε κατηγορία υπάρχουν οι *ενδοσυναρτητές (endofunctors),* οι οποίοι είναι οι συναρτητές $\mathbb{C} \rightarrow \mathbb{C}$, και το σύνολό τους γράφεται επίσης ως $End(\mathbb{C})$. Αυτό το σύνολο είναι πάντα μη – κενό, γιατί για κάθε κατηγορία $\mathbb{C}$ υπάρχει ο ταυτοτικός της συναρτητής $1_{\mathbb{C}}$ (ή $I_{\mathbb{C}}$), ο οποίος έχει ταυτοτικές απεικονίσεις για αντικείμενα και βέλη.
3. Ο συναρτητής $I: \mathbb{C} \rightarrow \mathbb{D}$, όταν η $\mathbb{C}$ είναι υποκατηγορία της $\mathbb{D}$, και απεικονίζει κάθε αντικείμενο και βέλος στον εαυτό του, ονομάζεται *περιληπτικός* ή *περίληψη (inclusion).*

Ας περάσουμε σε μερικά παραδείγματα συναρτητών:

1. Ο συναρτητής δυναμοσύνολο $\mathcal{P}: \boldsymbol{Set} \rightarrow \boldsymbol{Set}$, απεικονίζει σύνολα στα δυναμοσύνολά τους και κάθε συνάρτηση μεταξύ συνόλων στην αντίστοιχη συνάρτηση μεταξύ των δυναμοσυνόλων τους.
2. Η πρώτη θεμελιώδης ομάδα επάγει συναρτητή $\pi_1: \boldsymbol{Top}^* \rightarrow \boldsymbol{Grp}$, όπου η $\boldsymbol{Top}^*$ είναι η κατηγορία των τοπολογικών χώρων με ένα επιλεγμένο σημείο αναφοράς (pointed topological space). Η πρώτη θεμελιώδης ομάδα λειτουργεί ως εξής: Σε ένα τοπολογικό χώρο μπορώ να πάρω ένα κλειστό μονοπάτι ξεκινώντας και τελειώνοντας στο σημείο αναφοράς. Αν δύο τέτοια κλειστά μονοπάτια μπορούν με συνεχή παραμόρφωση να ταυτιστούν, τότε θεωρούνται ισοδύναμα. Αυτή η ταύτιση ορίζει μια σχέση ισοδυναμίας (ισοδύναμα είναι τα μονοπάτια που μπορούν να ταυτιστούν με συνεχή παραμόρφωση). Οι κλάσεις ισοδυναμίας ορίζουν τα στοιχεία της ομάδας, επίσης ορίζεται γινόμενο το οποίο αποδεικνύεται πως δεν εξαρτάται από το σημείο αναφοράς. Ο συναρτητής απεικονίζει κάθε τοπολογικό χώρο με σημείο αναφοράς (X, x_0) στην αντίστοιχη πρώτη θεμελιώδη ομάδα $\pi_1(X)$, ενώ τα βέλη της $\boldsymbol{Top}^*$ είναι οι συνεχείς απεικονίσεις τοπολογικών χώρων μαζί με την απεικόνιση του σημείου

αναφοράς, δηλαδή $(X, x_0) \mapsto (f(X), f(x_0))$, τα οποία απεικονίζονται σε ομομορφισμούς θεμελιωδών ομάδων (δηλαδή $F(f)(\pi_1(X)) = \pi_1(f(X))$).
3. Άλλα παραγδείγματα συναρτητών είναι το τανυστικό γινόμενο (tensor product). Το τανυστικό γινόμενο μεταξύ δύο γραμμικών χώρων πάνω από το ίδιο σώμα, δηλ $W \otimes V$, είναι μια απεικόνιση από τα ζεύγη στοιχείων των χώρων W, V στο τανυστικό τους γινόμενο. Δηλαδή είναι απεικόνιση από το καρτεσιανό γινόμενο των δύο αυτών χώρων $(W \times V)$, ενώ αποδυκνείεται πως οι απεικονίσεις αυτές είναι επίσης γραμμικός χώρος, τον οποίο συμβολίζουμε $W \otimes V$. Τελικά έχουμε μια απεικόνιση $W \times V \to W \otimes V$ η οποία ορίζει με τη σειρά της συναρτητή $\otimes: Vect_k \times Vect_k \to Vect_k$, με αυτή την απεικόνιση ως απεικόνιση αντικειμένων[9], και με $Vect_k$ την κατηγορία των γραμμικών χώρων πάνω από ένα σώμα k.
4. Η δράση ομάδας. Όπως είπαμε παραπάνω κάθε ομάδα μπορεί να γραφτεί ως μια κατηγορία G με ακριβώς ένα αντικείμενο, και μορφισμούς τα στοιχεία της ομάδας. Ένας συναρτητής $G \to Set$ είναι η δράση της ομάδας G πάνω σε ένα σύνολο (αυτό στο οποίο απεικονίζεται το αντικείμενο της G). Κάθε στοιχείο της ομάδας (μορφισμός της G) απεικονίζεται σε έναν μορφισμό από το σύνολο στον εαυτό του. Αλλιώς μπορεί να νοηθεί ως μορφισμός της κατηγορίας των ομάδων (συγκεκριμένα $G \to Aut(Y)$ όπου Y το σύνολο, και $Aut(Y)$ *η ομάδα αυτομορφισμών του.* Η ιδέα γενικεύεται σε διάφορες απεικονίσεις $X \times Y \to Y$, ως η (αριστερή) δράση του X στο Y.

Το παράδειγμα της πρώτης θεμελιώδους ομάδας είναι χαρακτηριστικό για να μας εξηγήσει σε απλά λόγια «τι κάνουν οι συναρτητές». Η απάντηση είναι πως «μεταφράζουν» από μία «θεωρία» σε μια άλλη. Το ερώτημα που προκύπτει είναι το πώς μπορεί κανείς να «μεταφράσει» από τη μία μετάφραση στην άλλη, μεταξύ δύο «όμοιων» μεταφράσεων (δηλαδή με ίδιο πεδίο/συνπεδίο μεταξύ τους). Στη θεωρία κατηγοριών αυτό το ερώτημα γίνεται «Αν προσπαθήσω να φτιάξω μια κατηγορία με συναρτητές για αντικείμενα, ποια θα ήταν τα βέλη»; Η απάντηση είναι πως θα πρέπει σίγουρα να μεταφράζει τις εικόνες των αντικειμένων από τον ένα συναρτητή στον άλλον σε εικόνα του ίδιου κάθε φορά αντικειμένου. Δηλαδή σε κάθε αντικείμενο x θα πρέπει να απεικονίζεται σε ένα βέλος μεταξύ των εικόνων του $F(x) \to G(x)$. Επιπλέον θα πρέπει να μεταφράζει την εικόνα κάθε βέλους από τον ένα συναρτητή, στην εικόνα του στον άλλο. Δηλαδή οι απεικονίσεις αντικειμένων σε βέλη θα πρέπει να ικανοποιούν ορισμένες συνθήκες. Τη «μετάφραση» αυτή την ονομάζουμε *φυσικό μετασχηματισμό (natural transformation).*

[9] Το τελευταίο × είναι η γενίκευση του καρτεσιανού γινομένου στα πλαίσια των κατηγοριών, και θα οριστεί αργότερα. Με αυτόν τον ορισμό θα είναι και προφανές ποια είναι η απεικόνιση μορφισμών στο παραπάνω παράδειγμα.

Ορισμός (Φυσικός Μετασχηματισμός). Έστω δύο συναρτητές μεταξύ των ίδιων κατηγοριών $F, G\colon \mathbb{C} \to \mathbb{D}$. Φυσικός μετασχηματισμός $\eta\colon F \Rightarrow G$ καλείται μια απεικόνιση από κάθε αντικείμενο $x \in \mathbb{C}$ στον μορφισμό της $\mathbb{D}$, $\eta_x\colon F(x) \to G(x)$. Αυτά τα βέλη ονομάζονται στοιχεία (components) του φυσικού μετασχηματισμού. Επιπλέον θα πρέπει για κάθε μορφισμό $f\colon x \to y$ της $\mathbb{C}$, η $F-\text{εικόνα}$ του να απεικονίζεται στην $G-\text{εικόνα}$ του από τα η_x, η_y. Δηλαδή θα πρέπει για κάθε μορφισμό να ισχύει η συνθήκη $\eta_y \circ F(f) = G(f) \circ \eta_x$ στην κατηγορία $\mathbb{D}$. Δηλαδή θα πρέπει για κάθε $f\colon x \to y$ το παρακάτω διάγραμμα[10] να είναι μεταθετικό:

$$\mathbb{C} \quad \begin{tikzcd} x \arrow[d, "f"'] \\ y \end{tikzcd} \qquad\qquad \mathbb{D} \quad \begin{tikzcd} F(x) \arrow[r, "\eta_x"] \arrow[d, "F(f)"] & G(x) \arrow[d, "G(x)"] \\ F(y) \arrow[r, "\eta_y"] & G(y) \end{tikzcd}$$

Αν κάθε στοιχείο ενός φυσικού μετασχηματισμού είναι αντιστρέψιμο στην $\mathbb{D}$, τότε ο φυσικός μετασχηματισμός λέγεται φυσικός ισομορφισμός (συμβ. $\eta\colon F \cong G$) και επάγει τον φυσικό ισομορφισμό $\eta^{-1}\colon G \cong F$ με στοιχεία τους αντιστρόφους των στοιχείων του η.

Προηγουμένως είδαμε την έννοια του ισομορφισμού ως μορφισμού μιας κατηγορίας, αλλά τι γίνεται μεταξύ κατηγοριών; Πότε είναι δύο κατηγορίες ισομορφικές μεταξύ τους;

Αν σκεφτούμε ότι οι συναρτητές (μεταξύ κατηγοριών) είναι το αντίστοιχο των μορφισμών (μεταξύ αντικειμένων), τότε φυσικό φαίνεται να μεταφέρουμε τον ορισμό του ισομορφισμού από τους μορφισμούς στους συναρτητές:

Ορισμός (Ισομορφισμός Κατηγοριών). Έστω δύο κατηγορίες $\mathbb{C}, \mathbb{D}$ και συναρτητής $F\colon \mathbb{C} \to \mathbb{D}$. Ο συναρτητής F είναι ισομορφισμός κατηγοριών αν υπάρχει συναρτητής $G\colon \mathbb{D} \to \mathbb{C}$, τέτοιος ώστε $F \circ G = 1_{\mathbb{D}}$ και $G \circ F = 1_{\mathbb{C}}$. Τότε λέμε ότι οι δύο κατηγορίες είναι ισομορφικές.σ

Ο ισομορφισμός αποδυκνείεται πως είναι πολύ «αυστηρή» συνθήκη όταν μιλάμε για κατηγορίες. Υπάρχουν περιπτώσεις που κατηγορίες που είναι «ουσιαστικά ίδιες» δεν είναι ισομορφικές. Αυτό συμβαίνει γιατί μέσα σε μια κατηγορία μπορεί να υπάρχει

[10] Τα σύμβολα των κατηγοριών έχουν προστεθεί στο διάγραμμα για διευκόλυνση του αναγνώστη για το που βρίσκεται κάθε τι.

διαφορετικό πλήθος από ισομορφικά αντικείμενα, δηλαδή, στη μία κατηγορία να υπάρχουν πολλά αντικείμενα ισομορφικά μεταξύ τους, και μέσα από έναν συναρτητή να απεικονίζονται όλα στο ίδιο αντικείμενο στην άλλη κατηγορία. Έτσι χρειαζόμαστε μια έννοια όταν μιλάμε για το «μεταξύ κατηγοριών» επίπεδο, η οποία να είναι περίπου «ισομορφισμός (κατηγοριών) έως ισομορφισμού (αντικειμένων)». Για να το πετύχουμε ορίζουμε την ισοδυναμία κατηγοριών.

Ορισμός (Ισοδυναμία Κατηγοριών). Έστω δύο κατηγορίες $\mathbb{C}, \mathbb{D}$ *και συναρτητής* $F: \mathbb{C} \to \mathbb{D}$. *Ο συναρτητής* F *είναι ισοδυναμία κατηγοριών αν υπάρχει συναρτητής* $G: \mathbb{D} \to \mathbb{C}$ *και φυσικοί ισομορφισμοί* $G \circ F \cong 1_{\mathbb{C}}$ *και* $F \circ G \cong 1_{\mathbb{D}}$. *Τότε λέμε ότι οι δύο κατηγορίες είναι ισοδύναμες.*

3. 2-Κατηγορίες

Η σύνθεση συναρτητών ορίζεται πλήρως από τη σύνθεση μορφισμών, γι αυτό και δεν την ορίσαμε αυστηρά. Συγκεκριμένα, αν $F: \mathbb{C} \to \mathbb{D}, G: \mathbb{D} \to \mathbb{E}$, η $G \circ F$ έχει απεικόνιση αντικειμένων $x \mapsto G(F(x))$, και απεικόνιση μορφισμών $(f: x \to y) \mapsto (G(F(f)): G(F(x)) \to G(F(y)))$. Όμως όταν περάσουμε στους φυσικούς μετασχηματισμούς, η κατάσταση είναι πιο σύνθετη. Μπορούμε να σκεφτούμε μία κατάσταση που μεταξύ δύο κατηγοριών υπάρχουν τρεις συναρτητές: θα θέλαμε να μπορούμε να συνθέσουμε τους φυσικούς μετασχηματισμούς μεταξύ τους.

$$\begin{tikzcd}
\mathbb{C} \arrow[rr, bend left=50, "F"{name=F}] \arrow[rr, "G"{name=G, description}] \arrow[rr, bend right=50, "H"'{name=H}] & & \mathbb{D}
\arrow[Rightarrow, from=F, to=G, "\eta"] \arrow[Rightarrow, from=G, to=H, "\zeta"]
\end{tikzcd}$$

Ορισμός (Σύνθεση Φυσικών Μετασχηματισμών – Κάθετη Σύνθεση). Έστω δύο κατηγορίες $\mathbb{C}, \mathbb{D}$, *και συναρτητές μεταξύ τους* $F, G, H: \mathbb{C} \to \mathbb{D}$. *Έστω φυσικοί μετασχηματισμοί* $\eta: F \Rightarrow G,\ \zeta: G \Rightarrow H$. *Ο φυσικός μετασχηματισμός – σύνθεση των* η, ζ *(συμβ.* $\zeta \cdot \eta: F \to H$*) ορίζεται στα στοιχεία της ως* $(\zeta \cdot \eta)_x = \zeta_x \circ \eta_x$.

Μπορούμε όμως να περιγράψουμε και την περίπτωση που έχουμε τρεις κατηγορίες, δύο ζεύγη συναρτητών, και φυσικό μετασχηματισμό σε κάθε ζεύγος. Η σύνθεση συναρτητών ορίζει ένα τρίτο ζεύγος συναρτητών, του οποίου ο φυσικός μετασχηματισμός ορίζεται από τα στοιχεία των υπολοίπων, όμως δεν προκύπτει από την κάθετη σύνθεση.

$$\begin{tikzcd} \mathbb{C} \arrow[rr, bend left=50, "F"] \arrow[rr, bend right=50, "F'"'] \arrow[rr, phantom, "\Downarrow\eta"] & & \mathbb{D} \arrow[rr, bend left=50, "G"] \arrow[rr, bend right=50, "G'"'] \arrow[rr, phantom, "\Downarrow\zeta"] & & \mathbb{E} \end{tikzcd}$$

Ορισμός (Σύνθεση Φυσικών Μετασχηματισμών – Οριζόντια Σύνθεση) Έστω τρεις κατηγορίες $\mathbb{C}, \mathbb{D}, \mathbb{E}$ *και συναρτητές* $F, F' : \mathbb{C} \to \mathbb{D}$, $G, G' : \mathbb{D} \to \mathbb{E}$, *και φυσικοί μετασχηματισμοί μεταξύ τους* $\eta : F \Rightarrow F', \zeta : G \Rightarrow G'$. *Υπάρχουν οι συνθέσεις συναρτητών* $G \circ F, G' \circ F' : \mathbb{C} \to \mathbb{E}$. *Θα ορίσουμε την οριζόντια σύνθεση των φυσικών μετασχηματισμών ως* $\zeta \circ \eta : G \circ F \Rightarrow G' \circ F'$, *και τα στοιχεία του ορίζονται από το παρακάτω διάγραμμα*

$$\begin{tikzcd} (G \circ F)(x) \arrow[r, "\zeta_{(F(x))}"] \arrow[d, "G(\eta_x)"'] \arrow[dr, "(\zeta \circ \eta)_x"] & (G' \circ F)(x) \arrow[d, "G'(\eta_x)"] \\ (G \circ F')(x) \arrow[r, "\zeta_{F'(x)}"'] & (G' \circ F')(x) \end{tikzcd}$$

Το τετράγωνο του παραπάνω διαγράμματος είναι μεταθετικό γιατί ο ζ είναι φυσικός μετασχηματισμός μεταξύ των G, G' για τους μορφισμούς της $\mathbb{D}$, άρα και για τα στοιχεία του η. Οι μεταθετικότητες των τριγώνων αντιστοιχούν στις εξισώσεις $\zeta_{F'(x)} \circ G(\eta_x) = (\zeta \circ \eta)_x$, και $G'(\eta_x) \circ \zeta_{F(x)} = (\zeta \circ \eta)_x$. Αν «ανοίξει»[11] κανείς τους παραπάνω συμβολισμούς, μπορεί εύκολα να καταλάβει πως αποτελούν τρίγωνα στην $\mathbb{E}$, και αποτελούν τους διαφορετικούς «δρόμους» τους οποίους μπορεί να ακολουθήσει κανείς για να πάει από ένα αντικείμενο της $\mathbb{C}$, σε ένα αντικείμενο της $\mathbb{E}$.

[11] Δηλαδή αν κατασκευάσει από το x βήμα-βήμα όλα τα στοιχεία που υπάρχουν σε αυτές τις εξισώσεις, κρατώντας υπόψιν σε ποια κατηγορία βρίσκεται κάθε φορά.

Το ότι η οριζόντια σύνθεση φυσικών μετασχηματισμών είναι φυσικός μετασχηματισμός αποδυκνείεται από το παρακάτω διάγραμμα, όπου το εξωτερικό παραλληλόγραμμο είναι το διάγραμμα φυσικού μετασχηματισμού του $\zeta \circ \eta$ για αυθαίρετο $f: x \to y$:

$$\begin{tikzcd}
x \arrow[d, "f"'] & (G \circ F)(x) \arrow[r, "G(\eta_x)"] \arrow[d, "(G\circ F)(f)"'] \arrow[rr, bend left, "(\zeta\circ\eta)_x"] & (G \circ F')(x) \arrow[r, "\zeta_{F'(x)}"] \arrow[d, "(G\circ F')(f)"] & (G' \circ F')(x) \arrow[d, "(G'\circ F')(f)"] \\
y & (G \circ F)(y) \arrow[r, "G(\eta_y)"] \arrow[rr, bend right, "(\zeta\circ\eta)_y"'] & (G \circ F')(y) \arrow[r, "\zeta_{F'(y)}"] & (G' \circ F')(y)
\end{tikzcd}$$

Αυτό μας δίνει άμεσα το εσωτερικό παραλληλόγραμο (από τον ορισμό του $(\zeta \circ \eta)_x$ τα τρίγωνα είναι μεταθετικά, και αντιστοιχούν στο κάτω τρίγωνο του προηγούμενου ορισμού). Το αριστερό τετράγωνο είναι μεταθετικό γιατί ο G είναι συναρτητής και ο η φυσικός μετασχηματισμός.[12]Το δεξί τετράγωνο είναι μεταθετικό γιατί είναι το τετράγωνο φυσικού μετασχηματισμού για το ζ στο μορφισμό $F(f): F(x) \to F(y)$, που είναι μορφισμός γιατί ο F είναι συναρτητής, και άρα απεικονείζει τον μορφισμό $f: x \to y$ σε μορφισμό.

Καταλήγουμε τελικά πως αν θέλουμε να φτιάξουμε μια κατηγορία με αντικείμενα κατηγορίες, και μορφισμούς συναρτητές, προκύπτουν αντικείμενα (φυσικοί μετασχηματισμοί) και πράξεις (οι δύο συνθέσεις των φυσικών μετασχηματισμών) που δεν περιγράφονται από τα αξιώματα της κατηγορίας. Για να ορίσουμε ένα αντικείμενο που θα μπορεί να λειτουργεί ως κατηγορία για τις κατηγορίες, πρέπει να προσθέσουμε στον ορισμό της τους φυσικούς μετασχηματισμούς, και τις συνθέσεις τους. Επίσης, θα πρέπει να δούμε ποια αξιώματα διέπουν αυτές τις συνθέσεις (αντίστοιχα με τα αξιώματα της σύνθεσης μεταξύ μορφισμών). Η προσεταιριστικότητα κάθε μίας από τις συνθέσεις αποδεικνύεται τετριμμένα από τη προσεταιριστικότητα της σύνθεσης στην $\mathbb{E}$. Οι ταυτοτικοί φυσικοί μετασχηματισμοί της $\circ$ είναι οι ίδιοι με της $\cdot$. Τέλος θα πρέπει να δούμε πως οι συνθέσεις αλληλεπιδρούν μεταξύ τους. Για να γίνει αυτό, ως γενική περίπτωση θα πάρουμε τις δύο προηγούμενες περιπτώσεις ταυτόχρονα:

[12] Είναι το διάγραμμα φυσικού μετασχηματισμού του η, απεικονισμένο από το συναρτητή G, αν γράψουμε τα στοιχεία $(G \circ F)(x)$ ως $G(F(x))$ (αντίστοιχα για y, f).

$$
\begin{tikzcd}[column sep=huge]
\mathbb{C} \arrow[r, bend left=60, "F"] \arrow[r, "F'" description] \arrow[r, bend right=60, "F''"'] \arrow[r, phantom, shift left=4, "\Downarrow\eta"] \arrow[r, phantom, shift right=4, "\Downarrow\eta'"] & \mathbb{D} \arrow[r, bend left=60, "G"] \arrow[r, "G'" description] \arrow[r, bend right=60, "G''"'] \arrow[r, phantom, shift left=4, "\Downarrow\zeta"] \arrow[r, phantom, shift right=4, "\Downarrow\zeta'"] & \mathbb{E}
\end{tikzcd}
$$

Θέλουμε με όποια σειρά κι αν διαπεράσουμε τους φυσικούς μετασχηματισμούς αυτού του διαγράμματος να έχουμε το ίδιο αποτέλεσμα. Έτσι προκύπτει η εξίσωση $(\eta' \cdot \eta) \circ (\zeta' \cdot \zeta) = (\zeta' \circ \eta') \cdot (\zeta \circ \eta)$, η οποία ονομάζεται *νόμος εναλλαγής (interchange law).*

Ορισμός (2-Κατηγορία). Μία συλλογή βελών με δύο διαφορετικές πράξεις οι οποίες ικανοποιούν τον νόμο εναλλαγής λέγεται διπλή κατηγορία (double category), ενώ αν επιπλέον τα ταυτοτικά βέλη των δύο πράξεων ταυτίζονται λέγεται 2-κατηγορία (2-category).

Η $\boldsymbol{Cat}$, μαζί με τους φυσικούς της μετασχηματισμούς είναι 2-κατηγορία. Με αντίστοιχο τρόπο (βέλη μεταξύ φυσικών μετασχηματισμών, ή πιο γενικά βέλη μεταξύ βελών μεταξύ βελών) ορίζονται και κατηγορίες *μεγαλύτερης διάστασης* (3-κατηγορίες, n-κατηγορίες).

4. Κατασκευές και ειδικά στοιχεία σε μια κατηγορία

Ας θεωρήσουμε μια κατηγορία $\mathbb{C}$. Πέρα από τα ίδια τα χαρακτηριστικά που έχουν όλες οι κατηγορίες, σημαντικό είναι να δούμε τι ιδιαίτερα χαρακτηριστικά μπορεί να παρουσιάζει. Επομένως, ενώ ως τώρα (αφού ορίσαμε την κατηγορία) κοιτάζαμε «έξω» από την κατηγορία, τώρα θα κοιτάξουμε προς τα «μέσα».

Ορισμός (Αρχικό/Τελικό αντικείμενο) Ένα αντικείμενο $1 \in ob(\mathbb{C})$ καλείται τελικό (terminal) αντικείμενο της κατηγορίας, αν για κάθε αντικείμενο υπάρχει ακριβώς ένας μορφισμός με το 1 συνπεδίο (δηλαδή, για κάθε $x \in ob(\mathbb{C})$ υπάρχει μοναδικό $x \to 1$. Αυτό σημαίνει πως ο μοναδικός μορφισμός $1 \to 1$ είναι ο ταυτοτικός. Επίσης είναι ισομορφικό με κάθε άλλο τελικό αντικείμενο της κατηγορίας.

Ένα αντικείμενο $0 \in ob(\mathbb{C})$ καλείται αρχικό (initial) αντικείμενο της κατηγορίας, αν για κάθε αντικείμενο υπάρχει ακριβώς ένας μορφισμός με το 0 πεδίο (δηλαδή, για κάθε $x \in ob(\mathbb{C})$ υπάρχει μοναδικό $0 \to x$. Αυτό σημαίνει πως ο μοναδικός μορφισμός $0 \to$

0 είναι ο ταυτοτικός. Επίσης είναι ισομορφικό με κάθε άλλο αρχικό αντικείμενο της κατηγορίας.

Ένα αντικείμενο που είναι και αρχικό και τελικό λέγεται null object. Αν z είναι null object, και πάρω τα (μοναδικά) βέλη $x \to z, z \to y$ για κάποια $x, y \in ob(\mathbb{C})$ τότε η σύνθεσή τους ονομάζεται μηδενικό βέλος (zero arrow) από το x στο y.

Για παράδειγμα στη $\boldsymbol{Set}$ κάθε μονοσύνολο είναι τελικό αντικείμενο, γιατί μπορούμε να απεικονίσουμε κάθε στοιχείο ενός συνόλου στο στοιχείο του μονοσυνόλου, και αυτό με μοναδικό τρόπο[13]. Το κενό σύνολο είναι το αρχικό αντικείμενο. Στην $\boldsymbol{Cat}$**,** αρχικό αντικείμενο είναι η κενή κατηγορία (χωρίς αντικείμενα ή μορφισμούς) $\mathbf{0}$, ενώ τελικό αντικείμενο είναι η $\mathbf{1}$, κατηγορία με ένα αντικείμενο και τον ταυτοτικό μορφισμό του μόνο. Σε μία μερική διάταξη $(P, \leq)$, αν υπάρχει μέγιστο και ελάχιστο αντικείμενο όλου του P, είναι αντίστοιχα τελικό και αρχικό αντικείμενο. Στην $\boldsymbol{Ring}$ κατηγορία των δακτυλίων, ο $\mathbb{Z}$ είναι αρχικό αντικείμενο, ενώ ο τετριμμένος δακτύλιος (με μόνο ένα στοιχείο) είναι τελικό.

Το ζεύγος αρχικό-τελικό αντικείμενο είναι η πρώτη φανερή (δεύτερη, αν δούμε το πεδίο-συνπεδίο) εμφάνιση της *αρχής του δυϊσμού* στη θεωρία κατηγοριών. Έστω μια κατηγορία $\mathbb{C}$, τότε ορίζεται μια κατηγορία $\mathbb{C}^{op}$ (ή $\mathbb{C}^*$) η οποία έχει ίδια αντικείμενα αλλά αντεστραμμένα βέλη. Αν τυπικά θέλουμε να κατασκευάσουμε το συναρτητή $\mathbb{C} \to \mathbb{C}^{op}$, έχει ως απεικόνιση αντικειμένων την ταυτοτική $(x \mapsto x)$, και ως απεικόνιση βελών την $(f: x \to y) \mapsto (f^{op}: y \to x)$. Η κατηγορία αυτή ονομάζεται αντίθετη (opposite) ή δυϊκή (dual) κατηγορία. Αν η $\mathbb{C}$ έχει ένα αρχικό αντικείμενο, τότε αυτό θα είναι τελικό στην $\mathbb{C}^{op}$, και αντίστροφα. Όμοια υπάρχουν και άλλα ζεύγη δυϊκών εννοιών (αριστερός/δεξιός αντίστροφος, πεδίο/συνπεδίο, και κατασκευές που θα δούμε στη συνέχεια όπως γινόμενο/συνγινόμενο, όριο/συνόριο). Ο ταυτοτικός μορφισμός, η αντιστρεψιμότητα ενός μορφισμού, και άλλες έννοιες που παρουσιάζουν μία συμμετρία μένουν αναλλοίωτες από την $\mathbb{C}$ στην $\mathbb{C}^{op}$. Επίσης ισχύει ότι ${\mathbb{C}^{op}}^{op} = \mathbb{C}$.

Για να περάσουμε την ιδέα αυτή και στους συναρτητές θα παρουσιάσουμε ένα απλό παράδειγμα. Έστω πως υπάρχει ένας ομομορφισμός f μερικών διατάξεων, ο οποίος αντιστρέφει την διάταξη της $(P, \leq)$, δηλαδή αν $p_1 \leq_P p_2$, τότε $f(p_2) \leq_{f(P)} f(p_1)$. Ένας τελεστής που αντιστρέφει διάταξη πράγματι συναντάται συχνά. Αν αντιμετωπίσουμε τις διατάξεις ως κατηγορίες, ορίζουμε το συναρτητή $(\_)^{op}: \boldsymbol{P} \to \boldsymbol{P^{op}}$, ταυτοτικό στα αντικείμενα, και για τους μορφισμούς αντιστρέφει τα βέλη. Τότε για κάθε ομομορφισμό μερικών διατάξεων $F: \boldsymbol{Q} \to \boldsymbol{P}$ μπορούμε να συνθέσουμε τους δύο συναρτητές, και να κατασκευάσουμε έναν νέο συναρτητή $F^{op} = (\_)^{op} \circ F$, ο οποίος δρά στα αντικείμενα

[13] Αν για παράδειγμα έχουμε το μονοσύνολο {5} μπορούμε να απεικονίσουμε ένα άλλο σύνολο, έστω X σε αυτό μέσω της συνάρτησης f(x)=5, με πεδίο ορισμού της f τα στοιχεία του X.

όπως ο F αλλά με αντεστραμένη διάταξη. Στην πραγματικότητα ο συναρτητής $(\_)^{op}$ είναι συναρτητής $(\_)^{op}\colon \boldsymbol{Cat} \to \boldsymbol{Cat}$ και αναστρέφει τα βέλη μιας κατηγορίας δίνοντας την αντίθετή της. Τέτοια αντικείμενα συναντόνται συνχνά στα μαθηματικά, κι έτσι ορίζεται και ένα δυϊκό ζεύγος του συναρτητή, ο *contravariant* συναρτητής ορίζεται ως εξής:

Ορισμός (Contravariant Συναρτητής). Έστω δύο κατηγορίες $\mathbb{C}, \mathbb{D}$. Ο $F\colon \mathbb{C} \to \mathbb{D}$ καλείται contravariant συναρτητής όταν: α)Αποτελείται από δύο απεικονίσεις, την απεικόνιση αντικειμένων, που απεικονίζει τα αντικείμενα της $\mathbb{C}$ σε αντικείμενα της $\mathbb{D}$, δηλαδή $x \mapsto F(x)$, $F(x) \in ob(\mathbb{D})$, και την απεικόνιση βελών που απεικονίζει κάθε βέλος της $\mathbb{C}$, έστω $f\colon x \to y$, σε βέλος της $\mathbb{D}$ μεταξύ των F −εικόνων του συνπεδίου και του πεδίου του f, δηλαδή στο $F(f)\colon F(y) \to F(x)$. Επιπλέον πρέπει να ισχύει, για κάθε αντικείμενο της $\mathbb{C}$ πως $1_{F(x)} = F(1_x)$, και να «σέβεται» τη σύνθεση των ανεστραμμένων βελών, δηλαδή εφόσον τα βέλη f, g της $\mathbb{C}$ είναι συνθέσιμα, τότε $F(g \circ f) = F(f) \circ F(g)$.

Σε αυτό το πλαίσιο ο συναρτητής που ορίσαμε αρχικά λέγεται *covariant συναρτητής*. Ένας contravariant συναρτητής $F\colon \mathbb{C} \to \mathbb{D}$, είναι covariant για την αντίθετη κατηγορία $F\colon \mathbb{C} \to \mathbb{D}^{op}$. Επιπλέον, για κάθε συναρτητή $F\colon \mathbb{C} \to \mathbb{D}$, ορίζεται ο *αντίθετος (opposite) συναρτητής* $F\colon \mathbb{C}^{op} \to \mathbb{D}^{op}$.

5. Οικουμενικές/Καθολικές Κατασκευές

Συνεχίζοντας να μιλάμε για το τι συμβαίνει μέσα στις κατηγορίες πρέπει να μιλήσουμε για την έννοια της *οικουμενικότητας/καθολικότητας (universality).* Υπάρχουν πολλοί τρόποι να μιλήσει κανείς για την έννοια αυτή, χωρίς να υπάρχει ταυτόχρονα κάποιος αυστηρός ορισμός της έννοιας που να είναι ταυτόχρονα διαισθητικός.

Έστω[14] το γινόμενο[15] $A \times B$. Αν θέλουμε να ορίσουμε μια απεικόνιση προς αυτό, είναι ακριβώς το ίδιο με το να ορίσουμε μία απεικόνιση προς το A και μία απεικόνιση προς το B. Οι απεικονίσεις $X \to A \times B$ έχουν μια αμφιμονοσήμαντη αντιστοιχία (bijection) με τα ζεύγη απεικονίσεων $(X \to A, X \to B)$. Αυτή όμως η αντιστοιχία είναι και με κάποιο τρόπο «φυσική». Θα μπορούσαμε να βρούμε δύο άλλα αντικείμενα C, D, για τα οποία υπάρχουν άλλες απεικονίσεις $C \to A, D \to B$. Τα ζεύγη $(X \to C, X \to D)$ πάλι έχουν μια αμφιμονοσήμαντη αντιστοιχία με τις απεικονίσεις $X \to A \times B$, η οποία δεν είναι όμως «φυσική» γιατί για να «εμφανίσουμε» την απεικόνιση $X \to A \times B$, πρέπει πρώτα να «περάσουμε» μέσα από τις απεικονίσεις $C \to A, D \to B$.

[14] Η έννοια της universal ιδιότητας/universal αντικειμένου είναι αρκετά απλή αν κανείς τη μελετήσει μέσω συγκεκριμένων παραδειγμάτων, τα οποία θα δούμε στη συνέχεια. Σε ένα πιο αφηρημένο επίπεδο είναι αρκετά πιο δύσκολη. Για εισαγωγή θα χρησιμοποιήσουμε το παράδειγμα που δίνει το nLab.
[15] Γενίκευση του καρτεσιανού γινομένου στα σύνολα, κτλ. Θα ορίσουμε αυστηρά μετά.

Για παράδειγμα η απεικόνιση με τύπο $n \mapsto (\sqrt{n}, n - in)$ που είναι $\mathbb{N} \to \mathbb{R} \times \mathbb{C}$, αντιστοιχεί αμφιμονοσήμαντα με το ζεύγος $(n \mapsto \sqrt{n}, n \mapsto n - in)$ που είναι $(\mathbb{N} \to \mathbb{R}, \mathbb{N} \to \mathbb{C})$. Θα ήταν πιο απλό σίγουρα από το να πάρουμε πρώτα το ζεύγος $(n \mapsto 2\sqrt{n}, n \mapsto n - in + n^2)$ και μετά το $(x \mapsto \frac{x}{2}, z \to z - \left(Im(z)\right)^2)$ και να τα συνθέσουμε. Μπορεί να φαίνεται ότι αυτό που κάναμε είναι να χειριστούμε σύμβολα για να κάνουμε κάτι πιο πολύπλοκο, και η αλήθεια δεν απέχει πολύ από αυτό. Δεν υπάρχει κανένας λόγος αν θέλει κανείς να γράψει μια απεικόνιση προς ένα γινόμενο να κάνει οτιδήποτε άλλο από το να γράψει δύο απεικονίσεις, μία για το κάθε όρισμα. Το πόσο προφανές είναι κάτι τέτοιο είναι αυτό που κάνει τόσο δύσκολο να μιλήσει σε αφηρημένο επίπεδο για αυτή τη διαδικασία, εξού και η έλλειψη ενός διαισθητικά καλού ορισμού στο επίπεδο της θεωρίας κατηγοριών. Ο τυπικός, αυστηρός ορισμός είναι ο εξής:

Ορισμός (Universal βέλος από ένα αντικείμενο σε ένα συναρτητή): Έστω συναρτητής $F: \mathbb{C} \to \mathbb{D}$ και $z \in ob(\mathbb{D})$. Universal βέλος[16] *από το z στον F είναι ένα ζεύγος (x, u) όπου $x \in ob(\mathbb{C})$ και $u: z \to F(x)$ στην $\mathbb{D}$, τέτοιο ώστε για κάθε άλλο ζεύγος (y, g) όπου $y \in ob(\mathbb{C})$ και $g: z \to F(y)$ στην $\mathbb{D}$, να υπάρχει μοναδικός μορφισμός $f: x \to y$ στην $\mathbb{C}$ τέτοιος ώστε $F(f) \circ u = g$.*

Ο παραπάνω ορισμός ισοδυναμεί με τη μεταθετικότητα του παρακάτω διαγράμματος για κάθε y, g, και κάθε φορά το f να είναι μοναδικό.

$$\begin{tikzcd} x \arrow[d, "f"] & & F(x) \arrow[d, "F(f)"] & & z \arrow[ll, "u"'] \arrow[dll, "g"] \\ y & & F(y) & & \end{tikzcd}$$

Επίσης το ίδιο ισχύει και για τη δυϊκή κατασκευή (όλα τα βέλη να είναι ανάποδα). Όπως βλέπουμε όμως, αυτός ο ορισμός δεν φανερώνει κάτι με μια πρώτη ματιά, οπότε καλύτερα να μελετήσουμε περιπτώσεις.

Θα ξεκινήσουμε από το *γινόμενο,* που θέλουμε να περιγράφει στις κατηγορίες ό,τι περιγράφει το καρτεσιανό γινόμενο στα σύνολα. Επομένως θέλουμε για δύο αντικείμενα, των οποίων το γινόμενο ψάχνουμε, να υπάρχουν δύο απεικονίσεις (οι προβολές του γινομένου). Επίσης θέλουμε για κάθε άλλο αντικείμενο το οποίο έχει απεικονίσεις στα αντικείμενα αυτά, να έχει μοναδική απεικόνιση προς το γινόμενο (ώστε να είναι universal):

Ορισμός (Γινόμενο). Έστω x_1, x_2 αντικείμενα μιας κατηγορίας. Το γινόμενό τους είναι ένα αντικείμενο που συμβολίζουμε $x_1 \times x_2$ της κατηγορίας με δύο μορφισμούς $\pi_1 \colon x_1 \times x_2 \to x_1$, $\pi_2 \colon x_1 \times x_2 \to x_2$, τέτοιο ώστε, για κάθε άλλο αντικείμενο y, με μορφισμούς $f_1 \colon y \to x_1$, $f_2 \colon y \to x_2$ να υπάρχει μοναδικός μορφισμός $f \colon y \to x_1 \times x_2$ τέτοιος ώστε $\pi_1 \circ f = f_1$ και $\pi_2 \circ f = f_2$. Δηλαδή, το παρακάτω διάγραμμα να είναι μεταθετικό:

$$\begin{tikzcd}
& y \arrow[dl, "f_1"'] \arrow[d, "f"] \arrow[dr, "f_2"] & \\
x_1 & x_1 \times x_2 \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & x_2
\end{tikzcd}$$

Η αυθαίρετη πεπερασμένη γενίκευση του γινομένου για μία οικογένεια αντικειμένων $\{x_i\}_{i \in I}$, που θα συμβολίζεται Πx_i έχει ως συνθήκες για κάθε $i \in I$ να ισχύει η συνθήκη $\pi_i \circ f = f_i$, δηλαδή κάθε διάγραμμα (για κάθε $i \in I$ δηλαδή) να είναι μεταθετικά:

$$\begin{tikzcd}
y \arrow[d, "f"] \arrow[dr, "f_i"] & \\
\Pi x_i \arrow[r, "\pi_i"'] & x_i
\end{tikzcd}$$

Στις περισσότερες περιπτώσεις, όταν για ένα μαθηματικό αντικείμενο ορίζουμε το (καρτεσιανό) γινόμενό του, έχουμε ορίσει το γινόμενο της κατηγορίας του.

Επίσης, αξίζει να σκεφτούμε το γινόμενο ως ένα τελικό αντικείμενο. Ας πούμε πως θα μπορούσαμε να κατασκευάσουμε μια κατηγορία με αντικείμενα τριάδες από αντικείμενο της $\mathbb{C}$ και τις απεικονίσεις τους στα x, y (ή $|I| + 1 -$άδες με τις απεικονίσεις προς τα x_i, $i \in I$), και βέλη ζεύγη (ή $|I| - άδες$) από μεταθετικά τρίγωνα $\pi_i \circ f = f_i$. Τότε το αντικείμενο $(x_1 \times x_2, \pi_1, \pi_2)$ θα είναι[17]το τελικό αντικείμενο αυτής της κατηγορίας, από τον ορισμό του γινομένου. Είναι πολύ πιο απλό, και κοντά στη διαίσθηση να σκεφτόμαστε όλα τα universal αντικείμενα ως τελικά (ή αρχικά αντικείμενα) μιας κατηγορίας από «κομμάτια» διαγραμμάτων μέσα στις κατηγορίες. Θα δούμε στη συνέχια πως γίνεται αυτό με έναν πιο αυστηρό τρόπο. Προς το παρόν, μπορούμε να σκεφούμε το γινόμενο $x_1 \times x_2$ ως το τελικό αντικείμενο όλων των διαγραμμάτων $x_1 \leftarrow \cdot \rightarrow x_2$, και αντίστοιχα για $\{x_i\}_{i\in I}$.

Αντίστοιχα θα υπάρχει και ένα δυϊκό του γινομένου, που θα είναι αρχικό αντικείμενο σε αυτή τη κατηγορία: Το *συνγινόμενο (coproduct)* $\mathrm{x}_1 + \mathrm{x}_2$ και αντίστοιχα για αυθαίρετες οικογένειες αντικειμένων $\coprod \mathrm{x}_\mathrm{i}$, και στην κατηγορία των συνόλων είναι η ξένη ένωση, ενώ η i_ απεικονίζει τα στοιχεία ενός συνόλου στον εαυτό τους στην ξένη ένωση. Το αντίστοιχο διάγραμμα είναι το εξής:

$$\begin{tikzcd}
 & y & \\
x_1 \arrow[ur, "f_1"] \arrow[r, "i_1"'] & x_1 + x_2 \arrow[u, "f"] & x_2 \arrow[ul, "f_2"'] \arrow[l, "i_2"]
\end{tikzcd}$$

Στη $\boldsymbol{Set}$ το συνγινόμενο είναι η ξένη ένωση, σε κατηγορίες μερικής διάταξης είναι ο τελεστής join (ελάχιστο άνω φράγμα).

Πώς όμως το γινόμενο και το συνγινόμενο αντιστοιχεί στον γενικό ορισμό ενός universal βέλους; Θα πάρουμε την περίπτωση του συνγινομένου, γιατί αντιστοιχεί στον ορισμό που δώσαμε προηγουμένως, ενώ το γινόμενο αντιστοιχεί στην αντίθετη περίπτωση (όλα τα βέλη ανάποδα).Θα θεωρήσουμε την κατηγορία $\mathbb{C}$, και ως F τον *διαγώνιο συναρτητή* $\Delta: \mathbb{C} \to \mathbb{C} \times \mathbb{C}$, με απεικονίσεις $x \to (x, x)$ και $f \to (f, f)$. Έστω ένα αντικείμενο (x_1, x_2) της κατηγορίας $\mathbb{C} \times \mathbb{C}$, τότε το universal βέλος από αυτό το αντικείμενο στο συναρτητή F είναι ζεύγος που αποτελείται από ένα αντικείμενο $z \in \mathbb{C}$ και ένα μορφισμό $(x_1, x_2) \to (z, z)$ της κατηγορίας $\mathbb{C} \times \mathbb{C}$, δηλαδή ένα ζεύγος βελών $i_1: x_1 \to z, i_2: x_2 \to z$ από την κατηγορία $\mathbb{C}$. Για κάθε άλλο ζεύγος βελών $\chi_1: x_1 \to z',\ \chi_2: x_2 \to z'$ υπάρχει μοναδικό βέλος $f: z \to z'$ ώστε $\chi_1 = f \circ i_1$ και $\chi_2 = f \circ i_2$. Το z είναι το συνγινόμενο των x_1, x_2 στην $\mathbb{C}$, δηλαδή το $x_1 + x_2$.

Μία άλλη τέτοια κατασκευή είναι ο *εξισωτής* (και ο *συνεξισωτής*) – *equaliser, coequaliser*. Έστω δύο αντικείμενα μιας κατηγορίας και δύο μορφισμοί $f, g: x \to y$. Θα

[17] Και αντίστοιχα το στοιχείο $(\Pi x_i, \{p_i\}_{i\in I})$

πάρουμε όλα τα αντικείμενα z, και τους μορφισμούς $h: z \to x$, ώστε $f \circ h = g \circ h$ (δηλαδή όλα τα βέλη που «εξισώνουν» τα f, g). Το αντικείμενο e, με το μορφισμό $eq: e \to x$ που πληρούν τη παραπάνω προϋπόθεση, και επιπλέον όλα τα z έχουν μοναδικό μορφισμό $u: z \to e$ ώστε $h = eq \circ u$, καλούνται *εξισωτής (equalizer) των f,g*. Ο παραπάνω ορισμός αντιστοιχεί στη μεταθετικότητα του παρακάτω διαγράμματος για κάθε z, και κάθε h, με την ιδιότητα της εξίσωσης των f, g:

$$\begin{tikzcd}
e \arrow[r, "eq"] & x \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & y \\
z \arrow[u, "u"] \arrow[ur, "h"'] & &
\end{tikzcd}$$

Ενώ στον συνεξισωτή τα αντίστοιχα διαγράμματα είναι (για κάθε z, h):

$$\begin{tikzcd}
x \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & y \arrow[r, "co"] \arrow[dr, "h"'] & c \arrow[d, "u"] \\
& & z
\end{tikzcd}$$

Στη κατηγορία $\boldsymbol{Set}$, η εξίσωση $f(x) = g(x)$, για $x \in X$, έχει κάποιες λύσεις, οι οποίες είναι ο *πυρήνας* της (*kernel)*. Ο εξισωτής (ως σύνολο) αποτελείται από τα στοιχεία του πυρήνα, και τα απεικονίζει (ως μορφισμός) στον εαυτό τους στο Χ. Ο συνεξισωτής δίνει το σύνολα κλάσεων ισοδυναμίας (σύνολο πηλίκο) του Y: όλοι οι μορφισμοί h του παραπάνω ορισμού απεικονίζουν για κάθε $x \in X$, τα $f(x), g(x)$ στο ίδιο στοιχείο του συνόλου Z, επομένως το σύνολο Z είναι το πηλίκο $Y/\cong$, που παράγεται από σχέσεις[18] όπου $f(x) \cong g(x)$. Ο συνεξισωτής παράγεται από τη μικρότερη τέτοια σχέση (δηλαδή αυτή που μόνο $f(x) \cong g(x)$, ή αλλιώς που δύο αντικείμενα ανήκουν σε αυτή τη σχέση

[18] Χωρίς να αποκλείεται να ανήκουν στη σχέση (δηλαδή να απεικονίζονται στο ίδιο αντικείμενο του Z) και ζεύγη αντικειμένων του Y για τα οποία δεν ισχύει απαραίτητα πως είναι εικόνες του ίδιου αντικειμένου του X. Όμως πάντα οι εικόνες του ίδιου αντικειμένου θα ανήκουν στη σχέση (θα απεικονίζονται στο ίδιο αντικείμενο του Z).

μόνο αν είναι εικόνες του ίδιου αντικειμένου του X), δηλαδή τη σχέση ισοδυναμίας που παράγεται από τη σχέση $f(x) \cong g(x)$ για κάθε x.

Αντίστοιχα και σε άλλες κατηγορίες ο εξισωτής και ο συνεξισωτής είναι αντίστοιχα η κατάλληλη έννοια που αντιστοιχεί στον πυρήνα και το πηλίκο που παράγεται από σχέση ισοδυναμίας.

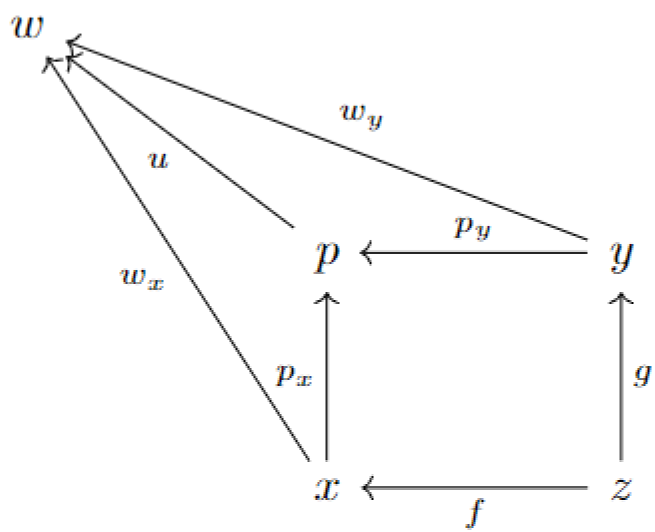
Ορισμός (Εφελκυσμός – Pullback). Έστω αντικείμενα έστω μορφισμοί $f: x \to z, g: y \to z$. Το αντικείμενο p, με τους μορφισμούς $p_x: p \to x$, $p_y: p \to z$, έτσι ώστε $f \circ p_x = g \circ p_y$ λέγονται pullback όταν για κάθε αντικείμενο w με μορφισμούς $w_x: w \to x, w_y: w \to y$ με $f \circ w_x = g \circ w_y$ υπάρχει μοναδικός[19] μορφισμός $u: w \to p$ έτσι ώστε $w_x = p_x \circ u$, και $w_y = p_y \circ u$. Δηλαδή το παρακάτω διάγραμμα να είναι μεταθετικό:

$$\begin{tikzcd}
w \arrow[dr, "u"] \arrow[drr, "w_y"] \arrow[ddr, "w_x"'] & & \\
& p \arrow[r, "p_y"] \arrow[d, "p_x"] & y \arrow[d, "g"] \\
& x \arrow[r, "f"'] & z
\end{tikzcd}$$

Εδώ να παρατηρήσουμε ότι ο εξισωτής είναι ειδική περίπτωση του pullback όπου $y = x$.

Το δυϊκό ονομάζεται *pushout/συνεφελκυσμός* και αντιστοιχεί στο παρακάτω διάγραμμα

[19] Συχνά σε αυτούς τους ορισμούς συμβολίζουμε u το βέλος που θέλουμε να είναι μοναδικό.



Με αντίστοιχο τρόπο το pullback είναι τελικό αντικείμενο της κατηγορίας με ίδια αντικείμενα και βέλη με την κατηγορία των αντικειμένων των γινομένων με το επιπλέον αξίωμα/συνθήκη $f \circ w_x = g \circ w_y$ για τα αντικείμενα.

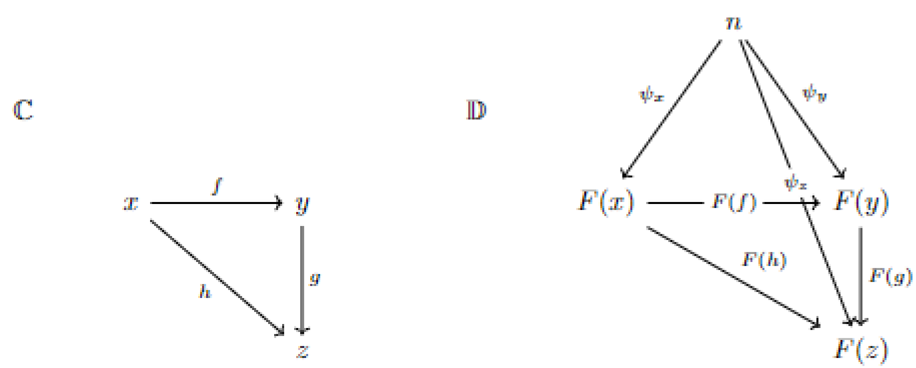
Στην κατηγορία $\boldsymbol{Set}$ το pushout των $f: Z \to X, G: Z \to Y$ είναι η ξένη ένωση των X,Y όπου τα στοιχεία με κοινή προεικόνα στο Z (δηλαδή τα $x \in X, y \in Y, x = f(z), y = g(z)$) ταυτίζονται στο σύνολο P, δηλαδή $p_x\big(f(z)\big) = p_y(g(z))$. Το pullback, είναι όλα τα στοιχεία του γινομένου των X,Y τα οποία απεικονίζονται στο ίδιο αντικείμενο του Z (δηλ είναι τα p αυτά, για τα οποία αν $p_x(p) = x, p_y(p) = y$, τότε $f(x) = g(y)$.

5.1 Κώνοι και Όρια

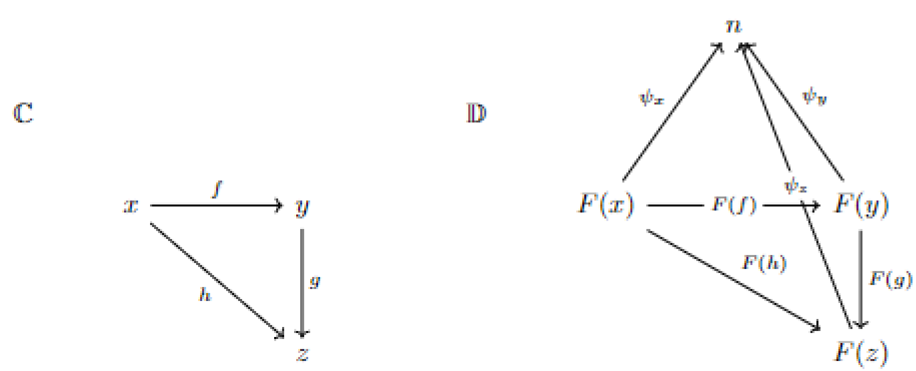
Τα παραπάνω παραδείγματα (όπως και πολλά άλλα) μπορούν να γενικευτούν. Όλα τους είναι αρχικά ή τελικά αντικείμενα μιας «κατηγορίας» από κάποιο «σχήμα» μέσα σε μια κατηγορία. Εδώ θέλουμε να περιγράψουμε με έναν πιο αυστηρό τρόπο αυτή τη «διαίσθηση» που περιγράψαμε παραπάνω σε πιο αδρές γραμμές. Κατ’αρχάς θα πρέπει να ορίσουμε αυστηρά αυτό που ονομάσαμε «σχήμα». *Διάγραμμα μορφής/σχήματος J στην κατηγορία* $\mathbb{C}$, όταν J είναι μια κατηγορία, θα ονομάσουμε έναν συναρτητή $F: J \to \mathbb{C}$. Αν περιοριστούμε σε μια πεπερασμένη κατηγορία, η ακόμα και σε πολύ «μικρές» κατηγορίες θα καταλάβουμε τη χρησιμότητά της. Για παράδειγμα, αν η J είναι η κατηγορία $\mathbf{1}$, δηλαδή η κατηγορία ενός αντικειμένου με τον ταυτοτικό του μορφισμό, τα διάφορα διαγράμματα απεικονίζουν στα αντικείμενα της $\mathbb{C}$. Αν η J είναι η κατηγορία $\mathbf{2}$, δηλαδή η κατηγορία με δύο αντικείμενα, τους ταυτοτικούς τους μορφισμούς και ένα μορφισμό από το ένα αντικείμενο στο άλλο, τότε τα διάφορα διαγράμματα απεικονίζουν στους μορφισμούς της $\mathbb{C}$. Αν η J είναι μία κατηγορία span, δηλαδή μια κατηγορία με τρία αντικείμενα, τους ταυτοτικούς τους μορφισμούς και άλλους δύο μορφισμούς σε αυτή τη μορφή $x \leftarrow y \rightarrow z$, τότε τα διάφορα διαγράμματα απεικονίζουν σε αντίστοιχες μορφές στη $\mathbb{C}$.

Αν πάρουμε δύο αντικείμενα τώρα της $\mathbb{C}$, τα x_1, x_2, και, σε μια κατηγορία span θέσουμε $F_0(x) = x_1, F_0(z) = x_2$, τότε όλα τα διαγράμματα $F_0: J \rightarrow \mathbb{C}$ δίνουν μια κατηγορία με το τελικό αντικείμενο το $x_1 \times x_2$.

Ας πούμε τώρα ότι η κατηγορία J είναι ένα μεταθετικό τρίγωνο, του οποίου τα αντικείμενα είναι τα x, y, z και οι μη-ταυτοτικοί μορφισμοί οι $f: x \rightarrow y, g: y \rightarrow z, h: x \rightarrow z$. Αν πάρουμε ένα διάγραμμά της (ας το ονομάσουμε F), ένα αντικείμενο n της κατηγορίας $\mathbb{C}$, μαζί με μορφισμούς προς τα αντικείμενα του διαγράμματος $\psi_x: n \rightarrow F(x), \psi_y: n \rightarrow F(y), \psi_z: n \rightarrow F(z)$ ονομάζεται *κώνος (cone) του διαγράμματος* F, αν ισχύουν $\psi_y = F(f) \circ \psi_x$, $\psi_z = F(g) \circ \psi_y$, $\psi_z = F(h) \circ \psi_x$.



Αντίστοιχα η δυϊκή έννοια *συν-κώνος (cocone)* ορίζεται αντιστρέφοντας τα βέλη ψ_i.

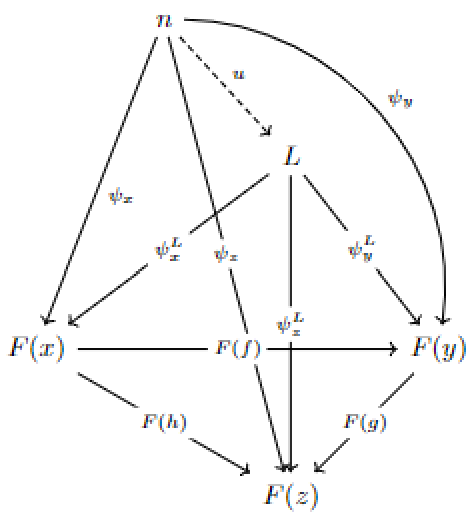


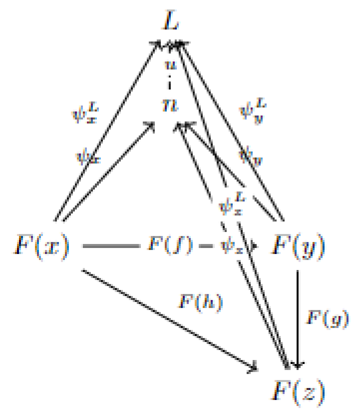
Ορισμός (Κώνος διαγράμματος F στην κατηγορία $\mathbb{C}$). Έστω διάγραμμα $F: J \rightarrow \mathbb{C}$. Κώνο ονομάζουμε ένα αντικείμενο n της $\mathbb{C}$, μαζί με τους μορφισμούς της $\mathbb{C}$ από το n προς τα αντικείμενα του διαγράμματος, τέτοιους ώστε για κάθε μορφισμό του διαγράμματος, το τρίγωνο που σχηματίζεται με τους αντίστοιχους μορφισμούς του κώνου να είναι μεταθετικό.

Ορισμός (Όριο/Συνόριο). Όριο ονομάζουμε έναν κώνο L ενός διαγράμματος F όταν από όλους τους άλλους κώνους αυτού του διαγράμματος, υπάρχει μοναδικό βέλος (έστω u) προς το L, και για κάθε αντικείμενο του διαγράμματος (έστω $F(x_i)$), το τρίγωνο που ορίζει μεταξύ των δύο κώνων είναι μεταθετικό (δηλαδή ισχύει $\psi^L_{x_i} = \psi_{x_i} \circ u$).

Συνόριο ονομάζουμε έναν συν-κώνο L ενός διαγράμματος F όταν από το L υπάρχει μοναδικός βέλος (έστω u) προς κάθε άλλο συν-κώνο, και για κάθε αντικείμενο του διαγράμματος (έστω $F(x_i)$), το τρίγωνο που ορίζει μεταξύ των δύο κώνων είναι μεταθετικό (δηλαδή ισχύει $\psi^L_{x_i} = u \circ \psi_{x_i}$).

Στα διαγράμματα κάτω βλέπουμε το όριο και το συνόριο, για τυχαίο κώνο/συν-κώνο n, πάνω από διάγραμμα μεταθετικού τριγώνου.





Αν τώρα οι κώνοι (και αντίστοιχα οι συν-κώνοι) ορίζουν κατηγορία, τότε το όριο και το συνόριο είναι το τελικό αντικείμενο της κατηγορίας των κώνων και το αρχικό αντικείμενο της κατηγορίας των συν-κώνων αντίστοιχα. Πράγματι[20] αν πάρουμε ως αντικείμενα τους κώνους ενός διαγράμματος $(n, \{\psi_i\}_{\{i\in J\}})$, τα βέλη μεταξύ τους $(n, \{\psi_i\}_{\{i\in J\}}) \to (n', \{\psi''_i\}_{\{i\in J\}})$, είναι πλειάδες $\left(f: n \to n', (f, id_i)_{\{i\in J\}}\right)$. Όπου τα f είναι βέλη μεταξύ των «κορυφών» του κώνου, και τα υπόλοιπα είναι ζεύγη βελών της κατηγορίας $\mathbb{C}$. Αυτό θεωρείται βέλος της κατηγορίας[21] $Arr(\mathbb{C})$, με αντικείμενα μορφισμούς της κατηγορίας $\mathbb{C}$, και μορφισμούς μεταξύ δύο βελών f, g, ζεύγη μορφισμών της $\mathbb{C}$, (έστω f_1, f_2) ώστε να σχηματίζονται μεταθετικά τετράγωνα $f_1 \circ f = g \circ f_2$. Στην περίπτωσή μας επειδή τα αντικείμενα του διαγράμματος ($F(i)$) είναι συνπεδίο και των 2

[20] Υπάρχουν διαφορετικοί τρόποι να οριστεί αυτή η κατηγορία, οι οποίοι όμως απαιτούν κατηγορίες κόμμα, τις οποίες δεν θα περιγράψουμε εκτεταμένα σε αυτή την εργασία.

[21] Προσοχή, η Arr είναι διαφορετική από την arr που χρησιμοποιήθηκε στην αρχή για να ορίσει την κατηγορία

κώνων ο παραπάνω ορισμός αποτελείται από μεταθετικά τρίγωνα. Η σύνθεση κληρονομείται από την κατηγορία $\mathbb{C}$, και η ταυτότητα δίνεται αν $f = id_n$.

Με αυτή την οπτική, οι παρατηρήσεις οι οποίες δώσαμε για τις προηγούμενες κατασκευές θεμελιώνονται, οι κατηγορίες για τις οποίες είπαμε ότι οι κατασκευές είναι τελικά (αρχικά) αντικείμενα, είναι κατηγορίες κώνων (συνκώνων) για κάποιο διάγραμμα.

Κεφάλαιο 2: Αλγεβρική Σημειωτική και Εννοιολογική Μίξη

1.Σημειωτική

Η Σημειωτική[22] είναι το θεωρητικό πεδίο των κοινωνικών επιστημών με αντικείμενο το *σημείο*. Το σημείο είναι κάτι πολύπλοκο να οριστεί: θα μπορούσαμε να πούμε αρχικά πως είναι η επικοινωνία νοήματος μέσω μιας συμβολικής αναπαράστασης αυτού του νοήματος, εφόσον θεωρήσουμε πως το νόημα υπάρχει μόνο μέσα στις συμβολικές αναπαραστάσεις του, και όχι ανεξάρτητα από αυτές. Καλύτερα θα μπορούσαμε να πούμε πως το επικοινωνούμενο νόημα, με το εκάστοτε μέσο αναπαράστασής του υπάρχουν *στην ενότητά τους*. Σύμφωνα με τον Saussure, η σημειωτική είναι «η επιστήμη που μελετά το ρόλο των σημείων ως μέρος της κοινωνικής ζωής».

Ο Charles Saunders Pierce, και ο Ferdinand de Saussure ήταν οι δύο (ανεξάρτητοι μεταξύ τους) θεμελιωτές της σημειωτικής στις αρχές του 20ου αιώνα. Το σημείο ορίζεται από μια σύνθετη τριαδική σχέση, κατά τον Pierce, σημείου[23]-αντικειμένου-ερμηνευτή. Το σχήμα του Saussure, το οποίο είναι πιο γνωστό σήμερα είναι η δυαδική σχέση σημαίοντος-σημαινομένου (signifier – signified) που αντιστοιχούν σε αυτά που πιο αδρά περιγράψαμε παραπάνω ως «συμβολική αναπαράσταση» και

[22] Αλλιώς Σημειολογία (αντίστοιχα semiotics ή semiology)

[23] Θα λέγαμε συμβόλου ή, καλύτερα, αναπαράστασης.

«επικοινωνούμενο νόημα» αντίστοιχα. Τα δύο παραπάνω στοιχεία δεν υπάρχουν παρά μόνο στην ενότητά τους, και τους δίνουμε διαφορετικά ονόματα μόνο για να τα ξεχωρίσουμε. Ταυτόχρονα, τα σημεία δεν υπάρχουν ως «ιδέες», αλλά μόνο μέσα στη συγκεκριμένη *σημείωση,* δηλαδή αυτό που θα μπορούσαμε να περιγράψουμε αδρά ως *την πράξη της επικοινωνίας*, η οποία στο εσωτερικό της περιλαμβάνει τα πολιτιστικά και άλλα συγκυριακά στοιχεία των μερών της επικοινωνίας αυτής (των υποκειμένων που «επικοινωνούν») και των κοινωνικών όρων υπό τους οποίους η «επικοινωνία» αυτή συμβαίνει.

Αυτή η μέθοδος καλύπτει όχι μόνο το γραπτό ή προφορικό λόγο, αλλά και διάφορα μέσα επικοινωνίας (εικόνα, βίντεο, χρώμα, τυπικές γλώσσες στα μαθηματικά, ήχος, κ.α). Ένα ολοκληρωμένο προϊόν (από πλευράς οργάνωσης των σημείων) ονομάζεται κείμενο (text) – ανεξάρτητα αν αυτό είναι κάτι γραπτό, ή ένας πίνακας, ή μία ταινία.

Το βασικό στοιχείο της οντολογίας των σημείων (πέρα από την ενότητα σημαίνοντος – σημαινομένου) αφορά την οργάνωση των σημείων, και συγκεκριμένα πως η οργάνωση των σημείων μεταξύ τους είναι τουλάχιστον όσο σημαντική όσο και η εσωτερική οργάνωση του σημείου στην ενότητα σημαίνοντος – σημαινομένου. Το σημείο δεν υπάρχει ποτέ καθ'εαυτό, αλλά μόνο στις σχέσεις του με άλλα σημεία. Αυτές οι σχέσεις δεν είναι a posteriori, δηλαδή δεν υπάρχει πρώτα το σημείο και μετά οι σχέσεις του με τα υπόλοιπα. Χρονικά, το σημείο γεννιέται και υπάρχει σε αυτές τις σχέσεις, δομικά, το σημείο δεν μπορεί να οριστεί ή να μελετηθεί παρά μόνο μέσα σε αυτές τις σχέσεις. Η θέση αυτή συμπυκνώνεται ως εξής: “*τα σημεία υπάρχουν σε συστήματα”*.

Δεν θα μείνουμε στην ίδια τη σημειωτική, αλλά όσα αναφέραμε παραπάνω είναι μια αναγκαία εισαγωγή για αυτό που ενδιαφέρει πράγματι αυτή την εργασία: την αλγεβρική σημειωτική.

2.Αλγεβρική σημειωτική

Η υπολογιστική σημειωτική (Computational Semiotics) είναι ένας αναπτυσσόμενος κλάδος της σημειωτικής τις τελευταίες δεκαετίες ο οποίος έχει ως αντικείμενο τη χρήση μαθηματικών και υπολογιστικών εργαλείων στη σημειωτική με δύο σκοπούς: α)τη χρήση μαθηματικών και υπολογιστικών μεθόδων στη σημειωτική έρευνα, κάτι το οποίο αποδεικνύεται χρήσιμο σε περιπτώσεις που κανείς θέλει να αναλύσει μεγάλο όγκο κειμένων (χιλιάδες ή εκατομμύρια κείμενα) για να μπορέσει να βγάλει ένα ποιοτικό συμπέρασμα και β)τον κατάλληλο μετασχηματισμό σημειωτικών εργαλείων, σε μαθηματικές οντότητες με αυστηρούς φορμαλισμούς, ώστε να βοηθήσουν στο σχεδιασμό συστημάτων που θα χειριστούν Γλώσσα (για παράδειγμα text-generators, natural language processors, κα).

Η μαθηματικοποίηση οποιουδήποτε θεωρητικού πλαισίου έχει ως αντικείμενο τη Γλώσσα, συναντά συγκεκριμένες δυσκολίες. Σίγουρα βασική είναι η πολυπλοκότητά της ως σύστημα, τα διαφορετικά οργανωτικά συστήματα που συνυπάρχουν (γραμματική, συντακτικό, νοηματοδότηση, ύφος) και αλληλοεπιρρεάζονται, η οργανική σχέση του πλαισίου (κοινωνικού, πολιτιστικού, συγκυριακού, κα.) με το τι λέγεται, και η ρευστότητα/δυναμική των συστημάτων αυτών. Όμως υπάρχουν και μεθοδολογικές δυσκολίες, που δεν αφορούν απλά την πολυπλοκότητα των συστημάτων αυτών, αλλά το κατά πόσο είναι δυνατό εν γένει να τυποποιηθεί η Γλώσσα, οι οποίες μάλλον αφορούν την επιστημολογία, οπότε δε θα σταθούμε πολύ σε αυτές. Θα αρκεστούμε στο ότι «το νόημα, ως ανθρώπινη εμπειρία, δεν περιορίζεται στα όρια άκαμπτων κανονιστικών συστημάτων χαρακτηριστικών, όσο πολύπλοκα δομημένα κι αν είναι αυτά»[24]. Για τους παραπάνω λόγους συχνά μια αλγοριθμική διαδικασία μελέτης κειμένου ή εικόνας, λειτουργεί συχνά στατιστικά, εντοπίζει κάποιες σημαντικές τιμές, οι οποίες μένει να ερμηνευθούν από το μελετητή. Εμείς θέλουμε να κάνουμε κάτι άλλο – να μελετήσουμε μοντέλα τα οποία να αναπαριστούν την πραγματικότητα με έναν αυστηρά δομημένο τρόπο, ακόμα και αν αυτό σημαίνει τον σημαντικό περιορισμό του εύρους του αντικειμένου μελέτης.

Η θέση πως τα σημεία υπάρχουν σε συστήματα, συνιστά μία μετατόπιση από τα αντικείμενα στη δομή, από μια ουσιοκρατική θεώρηση σε μια σχεσιακή-δομική θεώρηση. Η δομιστική θεώρηση η οποία υπάρχει (και μάλλον για την ιστορία της δυτικής σκέψης γεννιέται) στον Saussure, απαιτεί την αφαίρεση από την «έννοια» του αντικειμένου. Επομένως, μια μαθηματικοποίηση θα πρέπει να «ανέχεται», τουλάχιστον, αυτή τη θεώρηση – και έτσι δε θα μπορούσαμε να σκεφτούμε κάτι πιο ταιριαστό από τη θεωρία κατηγοριών, ως βασικό εργαλείο για αυτή τη μαθηματικοποίηση.[25]

Η Αλγεβρική Σημειωτική (Algebraic Semiotics) είναι μία προσέγγιση που ξεκινά από τον Goguen (1999) με σκοπό να ορίσει μαθηματικοποιήσεις εννοιών της σημειωτικής, χρησιμοποιώντας εργαλεία από τη θεωρία κατηγοριών ώστε αυτές οι έννοιες να μπορούν να είναι χρήσιμες σε περιβάλλοντα τα οποία απαιτούν μία μαθηματικοποίηση ή, γενικότερα, μια φορμαλιστική έκφραση. Πιο συγκεκριμένα η πρώτη εφαρμογή αυτής της προσέγγισης ήταν το cooperative software development και τα user interface[26]. Το εύρος του στόχου αυτού δεν είναι η τυπικά αυστηρή περιγραφή του συνόλου της επικοινωνίας,

[24] "An introduction to algebraic semiotics", Goguen J., 1998 Alg. Semiotics Goguen σ.3

[25] Σκαρπέλος, Φ. "Μία Μεθοδολογική Μελέτη πάνω στην Αλγεβρική Σημειωτική", υπό δημοσίευση

[26] "Algebraic semiotics, ProofWebs and distributed cooperative proving" (1997) των Goguen, J., Mori, A., & Lin, K. και "Semiotic morphisms, representations, and blending for interface design" Goguen J. (2003)

άλλωστε ο ίδιος ο Goguen από την αρχή αυτής της προσπάθειας συνιστά προσοχή απέναντι στην *πραγμοποίηση (reification)*, την οποία αντιλαμβάνεται ως «ταύτιση των [θεωρητικών] αφαιρέσεων με τα πραγματικά φαινόμενα [τα οποία αφορούν αυτές οι θεωρητικές αφαιρέσεις]»[27]. Αντ' αυτού η στόχευση είναι η κατασκευή «δυνητικά χρήσιμων μοντέλων». Ήδη από τα παραπάνω καταλαβαίνουμε ότι η Αλγεβρική Σημειωτική αντιστοιχεί στο δεύτερο στόχο που αναφέραμε παραπάνω, τη χρήση εννοιών της σημειωτικής, μετά από ένα κατάλληλο μετασχηματισμό τους σε μαθηματικές οντότητες για να χειριστούμε τη γλώσσα σε συστήματα που απαιτούν μαθηματική τυποποίηση.

Ο ορισμός που δίνει ο Goguen για τα συστήματα σημείων είναι ο ακόλουθος:

Ορισμός (Σύστημα Σημείων). Σύστημα Σημείων $\mathcal{S}$ αποτελείται από τα παρακάτω:
1.Ένα σύνολο S ειδών (sorts), το οποίο ιεραρχεί τα παραγόμενα σημεία, εφοδιασμένο με μια μερική διάταξη $\leq_S$, που ονομάζεται διάταξη υποειδών.
2.Ένα σύνολο V από είδη δεδομένων (data sorts), με πληροφορίες για τα παραγόμενα σημεία, όπως χρώμα/θέση (σε εικόνες), αληθοτιμές (σε τυπικές γλώσσες), κα. Η ξένη ένωση $S \cup V$ είναι εφοδιασμένη με τη μερική διάταξη $\leq_l$, που ονομάζεται διάταξη επιπέδων. Αν $v \in V, s \in S$, τότε πάντα $v \leq_l s$. Σε αυτή τη διάταξη υπάρχει μέγιστο αντικείμενο (top sort).
3.Για κάθε επίπεδο ένα σύνολο κατασκευαστών C_n επιπέδου n. Ένας κατασκευαστής επιπέδου n έχει ορίσματα σημεία επιπέδου εώς n, και αποτέλεσμα σημείο επιπέδου n. Ένας κατασκευαστής γράφεται ως $r: s_1 \dots s_m v_1 \dots v_k \to s$. Τα s_i είναι τα είδη κάθε ενός από m-πλήθος ορισμάτων και τα v_i είναι τα είδη δεδομένων από k-πλήθος παραμέτρων. Αν το πλήθος των ορισμάτων είναι 0, τότε έχουμε σταθερές $c: \to s$. Κάθε σύνολο κατασκευαστών είναι εφοδιασμένο με μία μερική διάταξη προτεραιότητας (priority ordering) $\leq_{p,n}$. Η διάταξη προτεραιότητας ποιοτικά περιγράφει με ποια «προτεραιότητα» θα υλοποιηθούν οι κατασκευαστές, κάτι που προσομοιάζει στο πως μία μηχανική διαδικασία θα προσομοιάσει φαινόμενα όπως το πόσο συχνά χρησιμοποιείται ένας συντακτικός κανόνας έναντι ενός άλλου, ή ποιον προτιμάμε να συμβαίνει συχνότερα για μια λειτουργία του προγράμματός μας.
4.Ένα σύνολο αξιωμάτων τα οποία λειτουργούν ως περιορισμοί για το τι μπορεί να κατασκευαστεί από το συγκεκριμένο σύστημα.
5.Μπορούν να υπάρχουν επίσης συναρτήσεις και σχέσεις στα παραγόμενα σημεία.

Αρχικά παρατηρούμε ότι ο ορισμός δεν αφορά τα σημεία αλλά περιγράφει τρόπους παραγωγής τους. Επιλέγοντας διαφορετικά σύνολα ειδών/κατασκευαστές/αξιώματα κανείς μπορεί να παράξει διαφορετικά συστήματα σημείων – υλοποιήσεις του ορισμού. Κάθε σημειωτικό σύστημα είναι μία *θεωρία,* ενώ οι διαφορετικοί τρόποι υλοποίησης του

[27] "An introduction to algebraic semiotics", Goguen J., 1998 σ.1

αφηρημένου συντακτικού της είναι τα μοντέλα της θεωρίας αυτής. Μέσα σε αυτά τα μοντέλα «ζουν» τα παραγόμενα σημεία.

Ας εξηγήσουμε όμως λίγο περεταίρω κάποια σημεία του ορισμού των σημειωτικών συστημάτων, Το σύνολο ειδών αντιστοιχεί σε διαφορετικά χαρακτηριστικά, τα οποία είναι κάθε φορά αυτά που μας ενδιαφέρουν για αυτό που θέλουμε να κάνουμε: θα μπορούσαν να αντιστοιχούν σε διαφορετικά παραδείγματα (τα οποία βρίσκουμε στην πρώτη εργασία του Goguen στο αντικείμενο[28]) σε ένα σύνολο $\{χαρακτήρας, λέξη, πρόταση\}$, ενώ σε άλλο παράδειγμα σε ένα σύνολο το οποίο αντιστοιχεί στο συντακτικό μιας πρότασης $\{πρόταση, ονοματική\ φράση, ρηματική\ φράση, ουσιαστικό, ρήμα, άρθρο, πρόθεση\}$. Ο συνδυασμός διαφορετικών σημείων μεταξύ τους για την κατασκευή άλλων σημείων καθορίζεται από συγκεκριμένους κανόνες που ονομάζονται κατασκευαστές, πχ ο συνδυασμός ενός ρήματος και μιας ονοματικής φράσης για να κατασκευαστεί μια ρηματική φράση. Η διάταξη επιπέδων ορίζει ποια είδη μπορούν να παράξουν ανώτερου επιπέδου είδη.

Η διάταξη προτίμησης αφορά τους κατασκευαστές (ως κανόνες), και μπορεί να προτυποποιήσει διαφορετικές καταστάσεις της πραγματικότητας. Για παράδειγμα ποιος κανόνας είναι συχνότερος από έναν άλλον ή ένα σύνολο κανόνων, που εφαρμόζεται ο πρώτος, εκτός από ειδικές περιπτώσεις που εφαρμόζεται ο δεύτερος κτλ. Επίσης μπορεί σε περιπτώσεις που σχεδιάζουμε ένα σύστημα να λάβει υπ'όψιν τις προτιμήσεις μας σε σχέση με τις παραμέτρους και τα ζητούμενα αυτού του συστήματος. Επομένως δύναται να αντιστοιχεί σε ένα φάσμα ποιοτικών στοιχείων κανόνων παραγωγής, όπως η συχνότητα εμφάνισης, οι ειδικές περιπτώσεις/εξαιρέσεις ενός κανόνα, και άλλα.

Έχοντας ορίσει τα σημειωτικά συστήματα, το επόμενο βήμα για να μπορούμε να τα χειριστούμε είναι η κατασκευή μιας κατηγορίας σημειωτικών συστημάτων. Για να το κάνουμε αυτό αρχικά θα κατασκευάσουμε τους μορφισμούς μεταξύ τους, οι οποίοι ονομάζονται σημειωτικοί μορφισμοί. Αυτοί, θα πρέπει προφανώς να διατηρούν διατάξεις και να σέβονται κανόνες ώστε να αντιστοιχούν στα βασικά χαρακτηριστικά του ομομορφισμού. Από την άλλη πλευρά θα χρειαστεί οι συναρτήσεις που τον αποτελούν να είναι μερικές (partial functions). Σε αντίθεση με ό,τι έχουμε συνηθίσει στα μαθηματικά, στην επικοινωνία τίποτα σχεδόν δεν είναι αμοιβαία μεταφράσιμο. Διαφορετικά μέσα (πχ βίντεο και γραπτός λόγος) δεν μπορούν «να πουν το ίδιο πράγμα», υπάρχουν λέξεις μη μεταφράσιμες μεταξύ γλωσσών, η ποιητική μεταφορά σημαίνει μόνο εκεί, δεν είναι μεταφράσιμη σε λόγο που διαπερνάται από την κυριολεξία (πχ επιστημονικός). Πάντα θα υπάρχει κάτι μη μεταφράσιμο, ή όπως ισχυρίζεται ο Torop «δεν υπάρχει μετάφραση χωρίς απώλεια»[29]. Διατηρώντας αυτά κατά νου έχουμε:

---

[28] "An introduction to algebraic semiotics", Goguen J., 1998
[29] "Intersemiosis and Intersemiotic Translation", 2000

Ορισμός (Σημειωτικός Μορφισμός) Ένας σημειωτικός μορφισμός μεταξύ δύο συστημάτων σημείων $M: \mathcal{S}_1 \to \mathcal{S}_2$ αποτελείται από:
1.Μία μερική συνάρτηση που απεικονίζει είδη του S_1 σε είδη του S_2, η οποία διατηρεί τη διάταξη υποειδών.
2.Είναι ταυτοτικός για δεδομένα και κάθε πράξη πάνω σε αυτά.
3.Μία μερική συνάρτηση που απεικονίζει κατασκευαστές του $\mathcal{S}_1$ σε κατασκευαστές του $\mathcal{S}_2$, έτσι ώστε η απεικόνιση του κατασκευαστή $r: s_1 \dots s_m \to s$ να είναι
$M(r): M(s_1) \dots M(s_m) \to M(s)$, εάν αυτή υπάρχει στο $\mathcal{S}_2$.
4.Μία μερική συνάρτηση που απεικονίζει κατηγορήματα και συναρτήσεις του $\mathcal{S}_1$ σε κατηγορήματα και συναρτήσεις του $\mathcal{S}_2$. Ένα κατηγόρημα $p: s_1 \dots s_n$ του $\mathcal{S}_1$ απεικονίζεται σε κατηγόρημα $M(p): M(s_1) \dots M(s_n)$ του $\mathcal{S}_2$ εφόσον αυτό υπάρχει.

Είναι απλό τεχνικά να οριστούν σύνθεση, ταυτοτικός σημειωτικός μορφισμός, και η σύνθεση να είναι προσεταιριστική. Απλώς θα πρέπει να κρατήσουμε κατά νου πως οι μορφισμοί είναι μερικές συναρτήσεις, κι έτσι οι συνθέσεις είναι ορισμένες αρκεί να είναι ορισμένοι και οι μορφισμοί. Η σύνθεση ορίζεται όμοια με τη σύνθεση στην κατηγορία συνόλων με μερικές συναρτήσεις $\boldsymbol{Pfn}$.

Η κατηγορία των σημειωτικών συστημάτων έχει τελικό αντικείμενο, το τετριμμένο σύστημα σημείων με ένα είδος (μέγιστο στη διάταξη επιπέδων) με συναρτήσεις μόνο στο σύνολο των δεδομένων. Θα συμβολίζεται ως $1_{\mathbb{S}}$.

Όμως, στην περίπτωσή μας έχει σημασία και η σύγκριση διαφορετικών «μεταφράσεων» ως προς την «ποιότητά» τους. Για παράδειγμα, αν μας ενδιαφέρει πόση δομή διατηρείται, ένας μορφισμός $f: S_1 \to S_2$, που ορίζεται τουλάχιστον σε όλα τα στοιχεία που ορίζεται ένας άλλος μορφισμός $g: S_1 \to S_2$, είναι «καλύτερος» από αυτόν. Μπορούμε έτσι, μεταξύ δύο σημειωτικών συστημάτων S_1, S_2 να ορίσουμε μια μερική διάταξη πάνω στους μορφισμούς αυτούς, δηλαδή $(\hom(S_1, S_2), \leq_{S_1,S_2})$. Επομένως θα πρέπει να εμπλουτίσουμε την κατηγορία των σημειωτικών συστημάτων με επιπλέον δομή, συγκεκριμένα μερικές διατάξεις στα σύνολα μορφισμών, οι οποίες θα διατηρούνται από τη σύνθεση (δηλαδή αν $f_1 \leq_{x,y} f_2, g_1 \leq_{y,z} g_2$, τότε $g_1 \circ f_1 \leq_{x,z} g_2 \circ f_2$), και οι ταυτότητες είναι μεγιστικά στοιχεία των διατάξεων στα αντίστοιχα $hom - set$ (δηλαδή για κάθε S, και για κάθε $f \in \hom(S, S)$, το οποίο να είναι συγκρίσιμο με την i_S, ισχύει $f \leq_{S,S} i_S$). Αυτές οι κατηγορίες στη βιβλιογραφία ονομάζονται $\frac{3}{2} - κατηγορίες$.

Τι σημαίνει όμως «ποιότητα» μιας μετάφρασης; Ήδη, από τους ορισμούς των συστημάτων σημείων μπορούμε να επιλέξουμε διαφορετικά κριτήρια για τη σύγκριση των μορφισμών, το κατά πόσο συντηρεί τα επίπεδα της διάταξης επιπέδων, τη προτεραιότητα των κατασκευαστών, ποιος είναι ορισμένος για περισσότερα στοιχεία του συστήματος σημείου στο οποίο είναι ορισμένοι, ποιος διατηρεί περισσότερα αξιώματα του S_1 ως λογικά συνεπαγόμενα από αξιώματα του S_2, και άλλα. Όμως, συχνά και η ίδια η διατήρηση δεν αποτελεί το ζητούμενο μιας μετάφρασης. Σύμφωνα πάλι με τον Torop δεν υπάρχει ιδεατή μετάφραση μεταξύ σημειωτικών συστημάτων. Επομένως και εμείς, στο πλαίσιο της αλγεβρικής σημειωτικής, πρέπει να εγκαταλείψουμε την αξίωση να «ανακαλύψουμε» τη διάταξη των σημειωτικών μορφισμών. Αντίθετα, αυτό που θα πρέπει να κάνουμε είναι να την «κατασκευάσουμε», κάθε φορά ανάλογα με τις ανάγκες και τις παραμέτρους του προβλήματος που δίνεται. Ο Goguen, έχοντας ως στόχο την εφαρμογή αυτών των εργαλείων σε software, αναφέρεται σε *διατάξεις σχεδιαστή*[30] (designer orderings), και κατασκευάζει ένα παράδειγμα, το οποίο βασίζεται σε μια γενικευμένη λεξικογραφική διάταξη:

Έστω θεωρούμε ένα πεπερασμένο πλήθος ποιοτικών κριτηρίων, τα οποία επάγουν, το κάθε ένα μία μερική διάταξη στους σημειωτικούς μορφισμούς μεταξύ δύο συστημάτων σημείων. Το σύνολο των κριτηρίων αυτών απεικονίζεται σε ένα σύνολο δεικτών (index set) I, όπου κάθε $i \in I$ αντιστοιχεί σε ένα κριτήριο, άρα και στην αντίστοιχη μερική διάταξη, την οποία γράφουμε $\leq_i$. Για κάθε μία από αυτές τις διατάξεις ισχύει πως i) αν $x \leq_i y$ και $y \leq_i x$, τότε $x =_i y$, ii) αν $x \leq_i y$ και όχι $y \leq_i x$, τότε γράφουμε $x <_i y$, iii) αν για κάποια x, y δεν ισχύει κανένα από τα $x \leq_i y, y \leq_i x$, δηλαδή αν τα δύο στοιχεία είναι μη συγκρίσιμα σε αυτή τη διάταξη, τότε γράφουμε $x \perp_i y$. Επιπλέον εφοδιάζουμε το σύνολο I με μία αυστηρή πλήρη διάταξη $<_I$. Η διάταξη σχεδιαστή συμβολίζεται $\ll$ και είναι μια γενικευμένη[31] λεξικογραφική διάταξη που ορίζεται ως η διάταξη γινόμενο των παραπάνω διατάξεων. Δηλαδή $x \ll y$ όταν για κάποιο $i \in I$, ισχύουν τα ακόλουθα:

- $\forall j \in I,\ \textit{τέτοιο ώστε}\ i <_I j, \textit{ισχύει}\ x =_j y$ ή $x \perp_j y$
- $x <_i y$

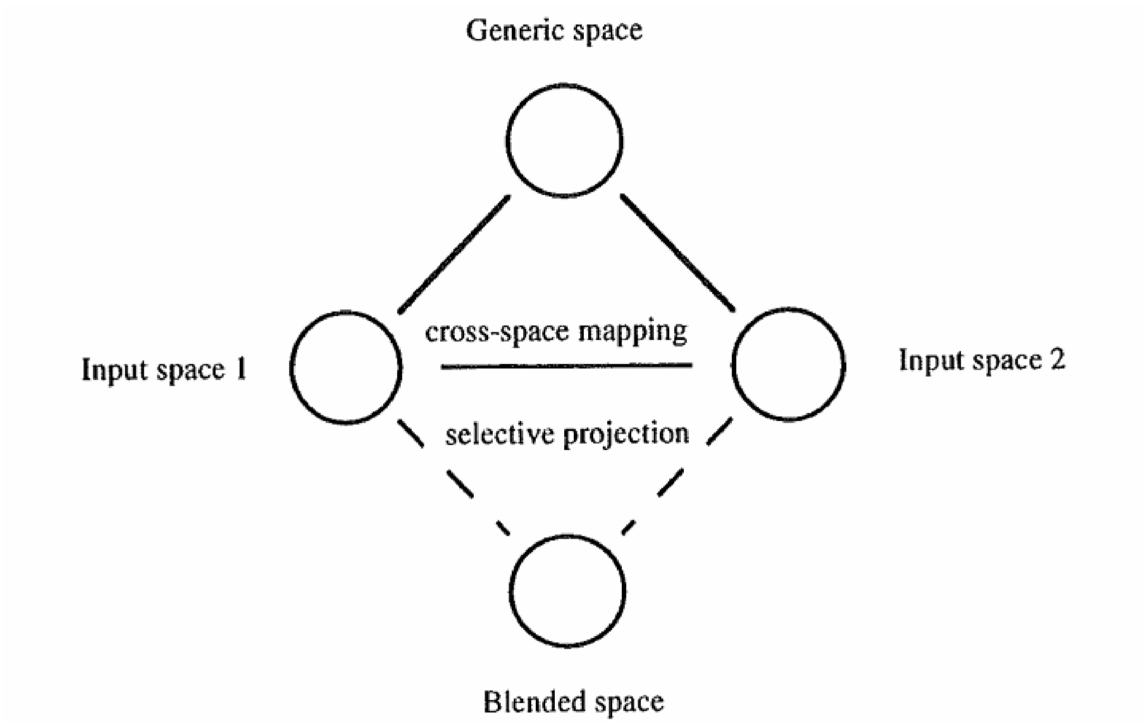
Το παραπάνω αποτελεί απλά ένα παράδειγμα, όπου μπορούμε να καθορίσουμε αυστηρά την σειρά «σημαντικότητας» των κριτηρίων.

3.Εννοιολογική Μίξη

Τα εργαλεία των πρώτων δύο κεφαλαίων θα τα εφαρμόσουμε σε μία έννοια που προέρχεται από τη γνωσιακή επιστήμη, την Εννολογική Μίξη. Η Εννοιολογική Μίξη εμφανίζεται πρώτη φορά στις εργασίες των Gilles Fauconnier και Mark Turner[32], και είναι ένα μοντέλο που περιγράφει τον τρόπο παραγωγής νέων εννοιών (ως νοητική

[30] "Semiotic morphisms, representations, and blending for interface design" Goguen J. (2003) σ.8
[31] Θεωρούμε απλή λεξικογραφική, την διάταξη στην οποία το $\leq_i$ είναι το ίδιο για κάθε $i \in I$.
[32] Πρώτη φορά το 1993 στο "Conceptual integration and formal expression"

λειτουργία) από ήδη προϋπάρχουσες έννοιες. Υπάρχει παράλληλα με άλλα μοντέλα που αφορούν αντίστοιχες λειτουργίες όπως η Θεωρία Μεταφοράς (metaphor) και Αναλογίας (analogy). Η εννοιολογική μίξη θεωρεί δύο (ή περισσότερες) έννοιες (input) οι οποίες υπάρχουν ως γνώση, και οι οποίες συντείθονται ώστε να παράξουν μία καινούρια. «Η ουσία της διαδικασίας είναι η κατασκευή μιας μερικής ταύτισης» της δομής των δύο εννοιών, η οποία «προβάλλεται επιλεκτικά» στην έννοια-μίξη, και στη συνέχεια «δυναμικά αποκτά δομή»[33]. Η κοινή δομή των input βρίσκεται σε ένα χώρο που προσομοιάζει σε «γενική έννοια» (generic). Παρακάτω παραθέτουμε το σχηματικό των Fauconnier & Turner[34]

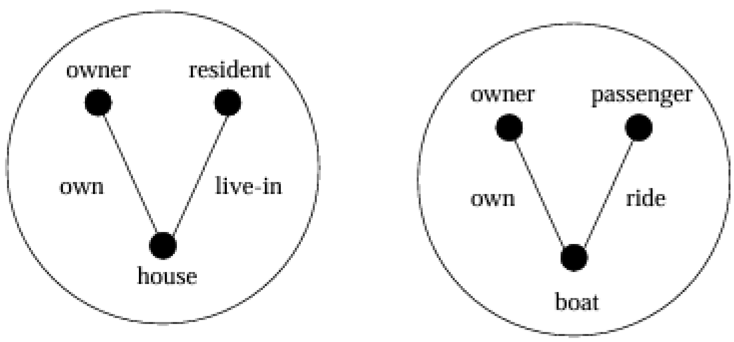


Στη θεωρία της εννοιολογικής μίξης συναντάμε την έννοια του *εννοιολογικού χώρου* (conceptual space) ή νοητικού χώρου (mental space)[35] η οποία συναντάται γενικά στη γνωσιακή επιστήμη. Οι εννοιολογικοί χώροι είναι αυτά τα στοιχεία που συμμετέχουν στη διαδικασία της μίξης (είτε ως input είτε ως παραγόμενα ”blend”) και είναι τα εργαλεία που αντιστοιχούν σε αυτό που πιο αδρά παραπάνω ονομάσαμε «έννοια». Οι Fauconnier

[33] “Conceptual blending, form and meaning”. 2003
[34] “Conceptual blending, form and meaning”. 2003
[35] Στη βιβλιογραφία της γνωσιακής επιστήμης χρησιμοποιείται περισσότερο η ορολογία mental space, ενώ η ορολογία conceptual space χρησιμοποιείται για ένα παραπλήσιο μοντέλο. Ο Goguen, χρησιμοποιεί την ορολογία conceptual space για να αναφερθεί στα mental spaces των Fauconnier & Turner

& Turner τους ορίζουν ως «μικρά νοητικά πακέτα τα οποία κατασκευάζονται καθώς σκεφτόμαστε και μιλάμε, με σκοπό την τοπική κατανόηση και δράση».

Ο Goguen αντιλαμβάνεται τους εννοιολογικούς χώρους όπως χρησιμοποιούνται στην θεωρία της εννοιολογικής μίξης ως μοντέλα συστημάτων σημείων, αρκετά απλοποιημένα σε σχέση με τον γενικό ορισμό. Καταρχήν δεν υπάρχουν κατασκευαστές εκτός από σταθερές (ή τουλάχιστον δεν υπάρχει συνήθως κάτι πολύ πιο πολύπλοκο). Μεταξύ των σταθερών υπάρχουν σχέσεις. Δεν εξειδικεύει για το τι συμβαίνει με τα είδη, αλλά και στα παραδείγματα που χρησιμοποιεί και σε όλες τις πρακτικές εφαρμογές δεν φαίνεται να χρειάζεται κάτι παραπάνω από ένα είδος. Τις σχέσεις μεταξύ των σταθερών ο Goguen τις περιγράφει και ως αξιώματα, χωρίς όμως να είναι απαραίτητα αναγκαίο. Παρακάτω βλέπουμε τα δύο παραδείγματα που χρησιμοποιεί ο Goguen, και γενικά χρησιμοποιούνται σε αρκετές εργασίες στο πεδίο.



Στα συγκεκριμένα παραδείγματα ο Goguen βλέπει από 3 σταθερές, οι οποίες ανήκουν όλες στο ίδιο είδος (του αρκεί το thing), και δύο σχέσεις τις οποίες θεωρεί ως αξιώματα. Ο Harell θεωρεί διαφορετικά είδη (person, object) τα οποία όλα θα μπορούσαν να είναι υποείδη του thing.

3.1 Μαθηματικοποίηση της Εννοιολογικής Μίξης

Η μαθηματικοποίηση της εννοιολογικής μίξης στην αλγεβρική σημειωτική γίνεται από τον Goguen ως εξής: Στην απλούστερη περίπτωση έχουμε 2 συστήματα σημείων input, τα I_1, I_2, πάνω από ένα γενικό (generic) σύστημα G, από το οποίο υπάρχουν μορφισμοί g_1, g_2 προς τα input. Η μίξη των παραπάνω συστημάτων γίνεται στον χώρο μίξης (blend[36]) το οποίο ονομάζουμε B. Οι μορφισμοί f_1, f_2 από τα input στο blend ονομάζονται injections. Έτσι έχουμε το παρακάτω διάγραμμα:

[36] Ο Goguen χρησιμοποιεί την ορολογία εννοιολογική μίξη (conceptual blend) μόνο στις περιπτώσεις που τα σημειωτικά συστήματα είναι εννοιολογικοί χώροι (conceptual spaces).

$$
\begin{tikzcd}
 & B & \\
I_1 \arrow[ur, "f_1"] & & I_2 \arrow[ul, "f_2"'] \\
 & G \arrow[ul, "g_1"] \arrow[ur, "g_2"'] \arrow[uu, "g"] &
\end{tikzcd}
$$

Το οποίο διάγραμμα πρέπει να είναι ασθενώς μεταθετικό (weakly commutative). Δηλαδή, η συνθήκη μεταθετικότητας του διαγράμματος $f_1 \circ g_1 = g = f_2 \circ g_2$ πρέπει να ισχύει, αλλά μόνο όπου είναι ορισμένοι όλοι οι μορφισμοί. Για να γίνουμε πιο συγκεκριμένοι: αν σε ένα σημείο της G είναι ορισμένοι οι μορφισμοί g και $f_1 \circ g_1$ (αντίστοιχα οι g και $f_2 \circ g_2$), τότε οι απεικονίσεις του μέσα από τους δύο αυτούς μορφισμούς στο B πρέπει να είναι οι ίδιες. Στις περιπτώσεις που ο $g: G \to B$ δεν είναι ορισμένος σε όλο το B, δεν είναι απαραίτητο (για τον ορισμό του Goguen) να υπάρχει κάποια ισότητα. Το blend επομένως θα οριστεί ως ένας τελεστής μεταξύ δύο μορφισμών $g_i: G \to I_i$, και θα συμβολίζεται $\diamond$.

Για τη μίξη ισχύουν τα παρακάτω
$g_1 \diamond g_2 = g_2 \diamond g_1$
$g_i \diamond id_G = g_i = id_G \diamond g_i$

Αν τώρα εξετάσουμε τα διαγράμματα της μορφής $f_1 \circ g_1 = f_2 \circ g_2$, δηλαδή για ένα δεδομένο ζεύγος (g_1, g_2) επιλέξουμε τα ζεύγη (f_1, f_2), και προς το παρόν αγνοήσουμε τον μορφισμό g, και την επιπλέον δομή της κατηγορίας, από την θεωρία κατηγοριών μπορούμε να ανασύρουμε την έννοια του pushout (συνεφελκυσμός), ως «βέλτιστο» από όλα τα ζεύγη (f_1, f_2), δηλαδή το αρχικό αντικείμενο όλων των αντίστοιχων ζευγών. Ο μορφισμός g, ακόμα κι αν δεν τον απαιτήσαμε ξεκάθαρα στον ορισμό του pushout προκύπτει αυτόματα από την $g \coloneqq f_1 \circ g_1 = f_2 \circ g_2$. Το

Το παραπάνω είναι γενικεύσιμο, για πάνω από ένα γενικό χώρο, και για παραπάνω από δύο inputs ανά γενικό χώρο. Αν θεωρήσουμε τώρα το ζεύγος (g_1, g_2) ως διάγραμμα (το οποίο θα ονομάσουμε $\mathbf{V}$) το σημείο B μαζί με τους μορφισμούς $f_i = I_i \to B$ είναι συνκώνος. Το αρχικό αντικείμενο των συνκώνων πάνω από ένα συγκεκριμένο διάγραμμα, δηλαδή ένας συνκώνος B, τέτοιος ώστε για κάθε άλλο συνκώνο πάνω από το ίδιο διάγραμμα $(B', \{f_i'\})$, να υπάρχει μοναδικό βέλος $u: B \to B'$ ώστε $f_i' = u \circ f_i$ για κάθε i, ονομάζεται συνόριο. Το συνόριο θα δώσει το «βέλτιστο», από όλα τα blends. Αν κάποιος προσμετρήσει στο pushout και το μορφισμό g, τότε είναι φανερό πως το pushout είναι συνόριο για το διάγραμμα **V.**

Όσον αφορά τα στοιχεία που έρχονται επιπρόσθετα από τα ειδικά χαρακτηριστικά της κατηγορίας των σημειωτικών συστημάτων. Όπως προαναφέραμε υπακούμε στην αρχή πως δεν υπάρχει μετάφραση χωρίς απώλεια. Δεν είναι τα πάντα μεταφράσιμα, και συχνά δεν θέλουμε και να μεταφράσουμε τα πάντα. Υπάρχουν στοιχεία τα οποία πιθανώς να θέλουμε να παραλείψουμε. Επομένως, πέρα από το ότι οι μορφισμοί μας είναι μερικές συναρτήσεις (δηλαδή όπου συναντάμε τη μεταθετικότητα παραπάνω στη πραγματικότητα εννοούμε ασθενή μεταθετικότητα) προβλέπεται να υπάρχει η δυνατότητα σε ένα διάγραμμα να υπάρχουν *βοηθητικοί μορφισμοί* (auxilliary morphisms), ή ακόμα και βοηθητικά αντικείμενα (αντικείμενα για τα οποία όλοι οι μορφισμοί από ή προς αυτά είναι βοηθητικοί). Θα πάρουμε λοιπόν μια γενικευμένη μορφή του συνγινομένου, που θέλουμε τα τρίγωνα $f_i' = u \circ f_i$ να είναι μεταθετικά για όλους τους μη-βοηθητικούς μορφισμούς. Ουσιαστικά το blend, δεν είναι το συνόριο πάνω από το διάγραμμα $\{g_i\}$ αλλά το συνόριο πάνω από το διάγραμμα $\{g_i\}'$, το οποίο είναι το αρχικό από το οποίο έχουμε αφαιρέσει τους βοηθητικούς μορφισμούς. Έτσι, μιλώντας περιγραφικά έχουμε τον «βέλτιστο δυνατό» κώνο.

Θέλοντας να μεταφέρουμε τους παραπάνω ορισμούς στο πλαίσιο των $\frac{3}{2} - κατηγοριών$, ο Goguen τους προσαρμόζει με τρόπο που η μοναδικότητα αντικαθίσταται από ιδιότητες που έχουν να κάνουν με τη διάταξη. Η έλλειψη της μοναδικότητας δεν αποτελεί πρόβλημα, αντίθετα είναι επιθυμητή: Συγκεκριμένα, τα συνόρια (όπως και οι άλλες καθολικές κατασκευές) είναι *μοναδικές μέχρι ισομοφισμό*. Ενώ συνήθως στα μαθηματικά αυτό είναι κάτι θεμιτό, στις προγραμματιστικές εφαρμογές αυτό δεν ισχύει. Ο Goguen διαβεβαιώνει πως δύο ισομορφικοί συνκώνοι, αντιστοιχούν σε διαφορετικές μίξεις, επομένως θα πρέπει να μπορούμε να διακρίνουμε μεταξύ διαφορετικών ονομάτων. Από την άλλη οι περιορισμοί που θέτουμε διασφαλίζουν ότι οι μίξεις θα λειτουργούν και θα παράγουν σημεία (στα παραδείγματα με εννοιολογικούς χώρους, θα παράγουν λέξεις και εννοιολογικούς χώρους που θα «βγάζουν νόημα»).

Ορισμός $\left(συνεπής\ \frac{3}{2} - κώνος\ πάνω\ από\ \mathbf{V}, \frac{3}{2} - pushout\right)$. *Έστω διάγραμμα* **V** *που αποτελείται από μορφισμούς* $g_1: G \to I_1$, $g_2: G \to I_2$. *Ένας συνκώνος* $(f_1: I_1 \to B, f_2: I_2 \to B)$ *καλείται συνεπής αν-ν υπάρχει μορφισμός* $g: G \to B$ *τέτοιος ώστε* $f_1 \circ g_1 \leq_{G,B} g$ *και* $f_2 \circ g_2 \leq_{G,B} g$. *Αν επιπλέον για κάθε άλλο συνεπή κώνο πάνω από το ίδιο* **V** *(έστω* $\{f_i': I_i \to B'\}$*) το σύνολο* $\{h: B \to B' | h \circ f_i \leq_{B,B'} f_i', i = 1,2\}$ *έχει μέγιστο στοιχείο, τότε ο συνκώνος είναι* $\frac{3}{2} - pushout$.

Παρατήρηση. Η απαλοιφή της μοναδικότητας από τον Goguen σίγουρα ξενίζει αν κανείς έχει συνηθίσει τους ορισμούς της θεωρίας κατηγοριών. Για να δικαιολογηθεί η επιλογή τους από μια αυστηρή κατηγοριοθεωρητική σκοπιά, θα μπορούσαμε να σκεφτούμε την $\frac{3}{2} - κατηγορία$, ως μια κατηγορία της οποίας τα homset απεικονίζονται σε ένα πλέγμα L. Τότε οι συνθήκες αυτές δεν φαίνονται και τόσο περίεργες. Στη θεωρία της ασάφειας η σύνθεση «αναπαρίσταται» σε μια πολλαπλασιαστική πράξη, και η $f \circ g = h$ μεταφέρεται από την απεικόνιση από τα βέλη της κατηγορίας στο πλέγμα ως $F(g) * F(f) \leq_L F(h)$. Στην ασαφή λογική, ας πούμε με αληθοτιμές στο διάστημα $[0,1]$ για ευκολία, στην οποία το λογικό «και» θα «αναπαρασταθεί» από το μέγιστο κάτω φράγμα δύο στοιχείων, και θα είναι πολλαπλασιαστική πράξη, γιατί η αληθοτιμή της πρότασης "$P\ και\ Q$" αναμένουμε να είναι μικρότερη ή ίση (το πολύ ίση) με τις αληθοτιμές των προτάσεων P, Q.

Αντίσιτοιχη γενίκευση γίνεται για το $\frac{3}{2} - συνόριο$, καθώς αρκεί να επεκτείνουμε σε πεπερασμένες οικογένειες μορφισμών, δηλαδή $\{g_i: I_i \to B |\ i \in I\}$ για πεπερασμένο I.

3.2 Θεωρήματα Συνθεσιμότητας Μίξης

Οι παραπάνω μαθηματικοποιήσεις για την εννοιολογική μίξη έχουν ένα βασικό πλεονέκτημα: κληρονομούν από τη θεωρία κατηγοριών μια συνθεσιμότητα των pushout ως συνθεσιμότητα blends. Δηλαδή όπως μπορώ να πάρω το blend δύο εννοιακών χώρων, το οποίο θα είναι επίσης εννοιακός χώρος, μπορω να το χρησιμοποιήσω ως input για το επόμενο blend με έναν «φυσικό» τρόπο. Κάτι τέτοιο, πέρα από το ότι είναι ενδιαφέρον μαθηματικά, είναι και ενδιαφέρον προγραμματιστικά, γιατί διασφαλίζει ενός είδους επαναληπτικότητα: μπορούμε να καλέσουμε μία μέθοδο η οποία θα κάνει τη μίξη και να πάρει input από τον εαυτό της.

Είναι προφανές πως αν έχω δύο μεταθετικά τετράγωνα που μοιράζονται έναν μορφισμό τότε και το παραλληλόγραμμο που σχηματίζουν θα είναι μεταθετικό. Θα χρησιμοποιήσουμε το διάγραμμα του επόμενου θεωρήματος. Θέλουμε να δείξουμε ότι $(h_1 \circ f_1) \circ g_1 = h_2 \circ (g_3 \circ g_2)$. Οπότε $h_2 \circ (g_3 \circ g_2) = (h_2 \circ g_3) \circ g_2 = (h_1 \circ f_2) \circ g_2 = h_1 \circ (f_2 \circ g_2) = h_1 \circ (f_1 \circ g_1) = (h_1 \circ f_1) \circ g_1$. Όπου η δεύτερη ισότητα είναι η μεταθετικότητα του δεξιού τετραγώνου, η τέταρτη ισότητα η μεταθετικότητα του αριστερού, και οι άλλες τρεις το αξίωμα της προσεταιριστικότητας της σύνθεσης.

Θεώρημα (Νόμος επικόλλησης/Pasting Law για Pushout[37]) Στο παρακάτω διάγραμμα το αριστερό τετράγωνο είναι pushout (*δηλαδή* $(f_1, f_2) = g_1 \oplus g_2$). *Τότε το παραλληλόγραμμο είναι pushout ακριβώς όταν το δεξί τετράγωνο είναι pushout* (*δηλαδή* $(h_1 \circ f_1, h_2) = g_1 \oplus g_3 \circ g_2\ αν-ν\ (h_1, h_2) = f_2 \oplus g_3$).

[37] Ο ίδιος (δυϊκός) νόμος υπάρχει και για pullbacks.

$$
\begin{tikzcd}
A \arrow[r, "f_1"] & P \arrow[r, "h_1"] & Q \\
G \arrow[u, "g_1"] \arrow[r, "g_2"'] & B \arrow[u, "f_2"] \arrow[r, "g_3"'] & C \arrow[u, "h_2"']
\end{tikzcd}
$$

Απόδειξη. Θα αποδείξουμε τη μία κατεύθυνση, δηλαδή το ότι αν το δεξί τετράγωνο είναι pushout *τότε είναι και το παραλληλόγραμμο. Έστω το παρακάτω μεταθετικό διάγραμμα (εκτός του χ_3). Τότε το χ_3 υπάρχει ως σύνθεση των g_3, χ_2 και το διάγραμμα παραμένει μεταθετικό.*

$$
\begin{tikzcd}
 & & R \\
A \arrow[urr, "\chi_1"] & & \\
G \arrow[u, "g_1"] \arrow[r, "g_2"'] & B \arrow[r, "g_3"'] \arrow[uur, dashed, "\chi_3"] & C \arrow[uu, bend right, "\chi_2"']
\end{tikzcd}
$$

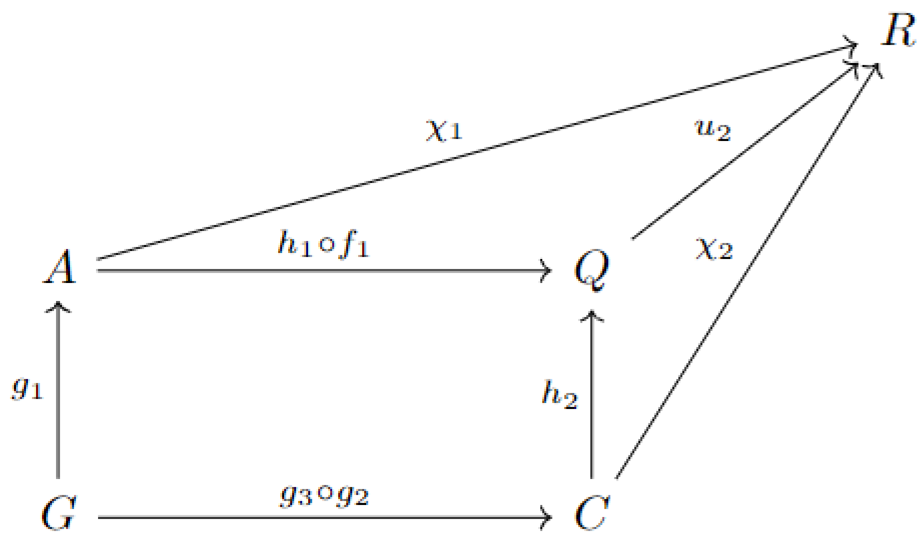
Όμως, το αριστερό τετράγωνο είναι pushout επομένως, αφού το διάγραμμα είναι μεταθετικό (συγκεκριμένα αφού $\chi_1 \circ g_1 = \chi_3 \circ g_2$), τότε υπάρχει μοναδικός μορφισμός $u_1 : P \to R$ που να κάνει μεταθετικά τα τρίγωνα $\chi_1 = u_1 \circ f_1, x_3 = u_1 \circ f_2$.

$$
\begin{tikzcd}
 & & R \\
A \arrow[urr, "\chi_1"] \arrow[r, "f_1"] & P \arrow[ur, "u_1"] & \\
G \arrow[u, "g_1"] \arrow[r, "g_2"'] & B \arrow[u, "f_2"] \arrow[uur, dashed, "\chi_3"] \arrow[r, "g_3"'] & C \arrow[uu, bend right, "\chi_2"']
\end{tikzcd}
$$

Θα σταματήσουμε πλέον να εμφανίζουμε το χ_3 *γιατί δεν χρειάζεται κάπου. Το δεξί τετράγωνο είναι pushout, επομένως υπάρχει μοναδικό βέλος* $u_2 \colon Q \to R$ *ώστε* $u_1 = u_2 \circ h_1$ *και* $\chi_2 = u_2 \circ h_2$.

$$
\begin{tikzcd}
 & & R \\
A \arrow[urr, "\chi_1"] \arrow[r, "f_1"] & P \arrow[ur, "u_1"] \arrow[r, "h_1"] & Q \arrow[u, "u_2"] \\
G \arrow[u, "g_1"] \arrow[r, "g_2"'] & B \arrow[u, "f_2"] \arrow[r, "g_3"'] & C \arrow[u, "h_2"] \arrow[uu, bend right, "\chi_2"']
\end{tikzcd}
$$

Τελικά το u_2 *είναι το μοναδικό βέλος* $Q \to R$ *ώστε* $\chi_2 = u_2 \circ h_2$, *όπως μόλις είπαμε, και αντικαθιστώντας στη σχέση* $\chi_1 = u_1 \circ f_1$ *το* $u_1 = u_2 \circ h_1$ *έχουμε* $\chi_1 = (u_2 \circ h_1) \circ f_1 = u_2 \circ (h_1 \circ f_1)$. *Άρα και το παραλληλόγραμμο είναι pushout. Δηλαδή, το* u_2 *είναι ο μοναδικός μορφισμός* $Q \to R$ *που να κάνει το διάγραμμα μεταθετικό.*



Σύνθεση 3/2 Pushout

Θεώρημα (Νόμος επικόλλησης για $\frac{3}{2} - pushout$*). Η σύνθεση δύο* $\frac{3}{2} - pushout$ *(διάγραμμα ακριβώς όπως στον νόμο για pushout) είναι* $\frac{3}{2} - pushout$.

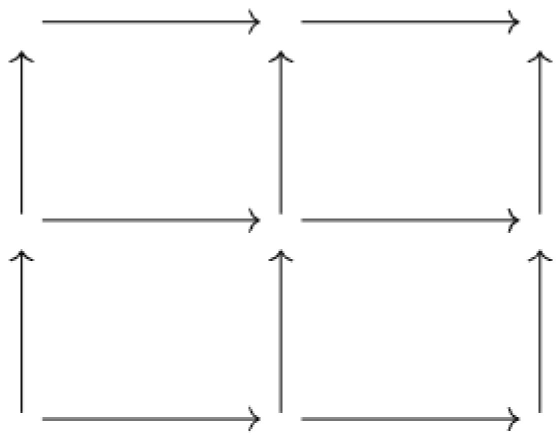
Απόδειξη. Ακολουθούμε τα ίδια βήματα που ακολουθήσαμε στην παραπάνω απόδειξη, κάθε επιχείρημα πως «το τετράγωνο είναι pushout επομένως υπάρχει μοναδικός μορφισμός» αντικαθίσταται από « το τετράγωνο είναι $\frac{3}{2} - pushout$ *επομένως το σύνολο έχει μέγιστο στοιχείο». Τα σύνολα που ορίζουμε (διατηρώντας τον συβολισμό τη προηγούμενης απόδειξης είναι το* $X = \{g: P \to R \mid g \circ f_1 \leq \chi_1 \text{ και } g \circ f_2 \leq \chi_3\}$ *του οποίου το μέγιστο στοιχείο θα ονομάσουμε* u_1 *και* $Y = \{g': Q \to R \mid g' \circ h_1 \leq u_1 \text{ και } g' \circ h_2 \leq \chi_2\}$ *του οποίου το μέγιστο στοιχείο θα ονομάσουμε* u_2.

Για να δείξουμε ότι το παραλληλόγραμμο είναι $\frac{3}{2} - pushout$ *θα πρέπει να δείξουμε ότι το σύνολο* $Z = \{g'': Q \to R \mid g'' \circ (h_1 \circ f_1) \leq \chi_1 \text{ και } g'' \circ h_2 \leq \chi_2\}$ *έχει μέγιστο. Θα δείξουμε ότι είναι ίσο με το σύνολο* $Y = \{g': Q \to R \mid g' \circ h_1 \leq u_1 \text{ και } g' \circ h_2 \leq \chi_2\}$ *που αναφέραμε πριν. Επομένως, θα έχει μέγιστο, το* u_2. *Έστω* $f \in Y$*, επομένως* $f \circ h_1 \leq u_1$*, άρα* $(f \circ h_1) \circ f_1 \leq u_1 \circ f_1$*, που όμως είναι* $\leq \chi_1$*, άρα* $f \in Z$. *Έστω τώρα* $f \in Z$*, άρα ισχύει* $f \circ (h_1 \circ f_1) \leq \chi_1$ *(η πρώτη συνθήκη του Ζ). Επίσης ισχύει* $f \circ h_2 \leq \chi_2$ *(δεύτερη συνθήκη του Ζ), άρα* $(f \circ h_2) \circ g_3 \leq \chi_2 \circ g_3$. *Όμως από τη μεταθετικότητα του δεξιού*

τετραγώνου αυτό μας δίνει $f \circ (h_1 \circ f_2) \leq \chi_2 \circ g_3$.
Όμως, για την δεύτερη συνθήκη $(f \circ h_1) \circ f_2 \leq \chi_2 \circ g_3$ *έχει μέγιστο στοιχείο που την ικανοποιεί στη θέση του* $f \circ h_1$, *το* u_1, *ακριβώς επειδή το δεξί τετράγωνο είναι pushout.*
Επίσης για την πρώτη συνθήκη $(f \circ h_1) \circ f_1 \leq \chi_1$, *επίσης το* u_1 *είναι το μέγιστο στοιχείο που την ικανοποιεί (στη θέση του* $f \circ h_1$, *γιατί το αριστερό τετράγωνο είναι pushout.*
Επομένως, και από τις 2 συνθήκες παίρνουμε ότι $f \circ h_1 \leq u_1$, *το οποίο είναι ακριβώς η πρώτη συνθήκη του* Y. *Η δεύτερη συνθήκη του Y είναι η ίδια με τη δεύτερη συνθήκη του Z, επομένως ο f την κληρονομεί από υπόθεση. Άρα* $f \in Y$.

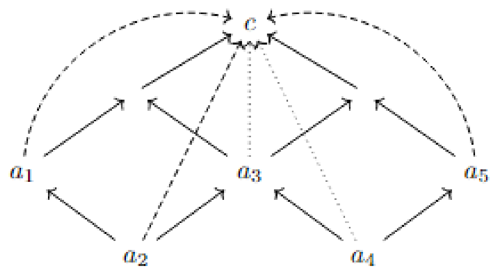
Τελικά δείξαμε ότι $Y \subseteq Z$, *και* $Z \subseteq Y$, *άρα ότι τα 2 σύνολα είναι ίσα, και επομένως έχουν ίδιο μέγιστο. Επομένως το σύνολο Z έχει μέγιστο το* u_2.

Πόρισμα. Στο παρακάτω διάγραμμα αν όλα τα μικρά τετράγωνα είναι $\frac{3}{2} - pushout$ *το ίδιο ισχύει και για το μεγάλο τετράγωνο:*



Η απόδειξη είναι η εφαρμογή του προηγούμενου θεωρήματος πρώτα στο κάτω παραλληλόγραμμο, μετά στο πάνω παραλληλόγραμμο, και τέλος στο τετράγωνο ως αποτέλεσμα των δύο παραλληλογράμμων.

Τα παραπάνω μας αρκούν στη πραγματικότητα για οποιαδήποτε εφαρμογή. Τα αποτελέσματα γενικεύονται από τον Goguen σε 3/2-συνόρια, καθώς τα συνόρια συνδεδεμένων διαγραμμάτων μπορούν να κατασκευαστούν από τα επί μέρους pushout. Επιπλέον αποδυκνείεται μία ειδική περίπτωση στην οποία το blend από 2 επί μέρους blend είναι επίσης blend, και μάλιστα ταυτίζεται με το 3/2- συνόριο του διαγράμματος:



Συγκεκριμένα αν πάρουμε τα 3/2-pushout των κάτω μορφισμών (α_1, α_3 και α_3, α_5), και μετά το 3/2-pushout αυτών των 2 pushout, έχουμε το 3/2-συνόριο του διαγράμματος W (δηλαδή αυτό που αποτελείται από τα αντικείμενα α_i και τα βέλη μεταξύ τους. Δηλαδή $\big((a_2 \to a_1) \oplus (a_2 \to a_3)\big) \oplus \big((a_4 \to a_3) \oplus (a_4 \to a_5)\big) = CoLim(a_1, a_2, a_3, a_4, a_5)$ για $\oplus$ το 3/2-pushout και Colim το 3/2-colimit.

Κεφάλαιο 3: Οντολογίες και προγραμματιστικό περιβάλλον

Στο προηγούμενο κεφάλαιο, αναλύθηκαν οι βασικές αρχές της αλγεβρικής σημειωτικής και πως μέσω αυτής μαθηματικοποιούμε την εννοιολογική μίξη, εξετάζοντας πώς αυτές οι θεωρητικές έννοιες μπορούν να συμβάλουν στην ανάλυση και τη διαχείριση συμβολικών συστημάτων. Σε αυτό το κεφάλαιο, θα επικεντρωθούμε στα προγραμματιστικά εργαλεία που θα χρησιμοποιηθούν για να μετατρέψουμε αυτές τις θεωρητικές έννοιες σε προγραμματιστικές εφαρμογές.

Στην βιβλιογραφία που υπάρχει πάνω στο αντικείμενο[38] χρησιμοποιείται η γλώσσα OWL (Ontology Web Language) και παραλλαγές[39] της. Αυτό συμβαίνει καθώς το βασικό εργαλείο για να προτυποποιήσουμε σε προγραμματιστικό περιβάλλον όλα αυτά τα οποία αναφέρθηκαν προηγουμένως ως έννοιες ή εννοιολογικοί χώροι ονομάζεται οντολογία[40]. Η παρούσα έκδοση της OWL είναι η OWL 2.

---

[38] Ενδεικτικά αναφέρουμε τα "Towards ontological blending" των Hois κα (2010) και "Blending in the Hub" των Kutz κα (2012).

[39] Επίσης ενδεικτικά αναφέρουμε τα "A categorical approach to ontology alignment" των Codescu κα (2014) και "Ontological blending in DOL" των Kutz κα (2012).

[40] Τη λέξη οντολογία τη χρησιμοποιήσαμε και στο προηγούμενο κεφάλαιο, αλλά εκεί αναφερόμασταν στην φιλοσοφική έννοια «οντολογία». Από εδώ και πέρα θα αναφερόμαστε στην προγραμματιστική έννοια οντολογία. Οι δύο έννοιες αυτές συνδέονται, ωστόσο αυτό δεν αφορά την εργασία μας. Πιο συγκεκριμένα ο όρος που μας καλύπτει είναι «τυπική οντολογία»

Για διάφορους λόγους για τους οποίους θα αναφερθούμε παρακάτω θέλουμε να χρησιμοποιήσουμε τη γλώσσα της Python. Αυτό μας το επιτρέπει το πακέτο Owlready2, που επιτρέπει τη χρήση και επεξεργασία οντολογιών της OWL 2 ως αντικέιμενα της Python.

1.Οντολογιες[41] και Γλώσσες Οντολογιών

Οι οντολογίες είναι δομημένες αναπαραστάσεις γνώσης που οργανώνουν έννοιες και τις σχέσεις τους, αναφορικά με ένα αντικείμενο. Για να γίνουμε πιο συγκεκριμένοι: Έστω ένα πρόβλημα απαιτεί την επικοινωνία μεταξύ ανθρώπων, οργανισμών και των αντίστοιχων συστημάτων λογισμικού. Όταν λέμε «γνώση» εννοούμε τη συναντίληψη των οργανισμών αυτών (ή των λογισμικών αυτών) σε σχέση με το αντικείμενο της κοινής τους δράσης. Όταν η γνώση αυτή βρίσκεται, όχι στο μυαλό του κάθε ενός (“implicit”), αλλά είναι ρητή (“explicit”), δηλαδή υπάρχει μία αναπαράσταση των εννοιών, αυτή η αναπαράσταση καλείται οντολογία. Το παραπάνω συνοψίζεται από τους Uschold, Gruninger (1996) από τον περιγραφικό ορισμό της οντολογίας ως ως «ρητή καταγραφή μιας κοινής αντίληψης για την περιοχή ενός δεδομένου αντικειμένου». Αυτή η κοινή αντίληψη χρησιμοποιείται ως «εννοποιητικό πλαίσιο» στην επίλυση προβλημάτων που αφορούν το αντικείμενο γύρω από το οποίο είναι χτισμένη. Είναι, θα μπορούσαμε να πούμε ο κοινός εννοιολογικός χώρος γύρω από το αντικείμενο αυτό.

Οι οντολογίες λειτουργούν ως το πρότυπο που καθορίζει πώς οι έννοιες συνδέονται μεταξύ τους, επιτρέποντας τη σαφή κατανόηση και την ανταλλαγή πληροφορίας. Σε γενικές γραμμές καθορίζονται οι βασικές έννοιες, οι ορισμοί τους, και οι σχέσεις μεταξύ τους. Αυτές οι σχέσεις μπορούν να είναι ιεραρχικές σχέσεις (π.χ. "is-a" που θα μπορούσε να αντιστοιχεί σε σχέσεις υποειδών) ή άλλες σχέσεις (στο παράδειγμα των εννοιολογικών χώρων το “live-in” και το “own” που αντιστοιχούσαν σε διμελείς σχέσεις μεταξύ σημείων). Αυτές οι έννοιες οργανώνονται σε μια ιεραρχική δομή, όπου οι πιο γενικές έννοιες βρίσκονται ψηλότερα, και οι πιο συγκεκριμένες χαμηλότερα. Βλέπουμε ήδη κοινά στοιχεία που μας επιτρέπουν να χρησιμοποιήσουμε μια οντολογια για τις προδιαγραφές των συστημάτων σημείων και των εννοιολογικών χώρων: έννοιες, σχέσεις (πχ διμελείς ή σταθερές) και ιεράρχησή τους (διάταξη).

Με αυτήν την οργάνωση, οι οντολογίες επιτρέπουν στα υπολογιστικά συστήματα να παράγουν νέα γνώση, να εξάγουν συμπεράσματα, και να βελτιώνουν την αναζήτηση πληροφοριών. Οι Uschold, Gruninger αναγνωρίζουν τρείς βασικές χρήσεις για τις οντολογίες. α)Διευκολύνουν την επικοινωνία και την ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων, και ανθρώπων προσφέροντας έναν κοινό τρόπο κατανόησης και διαχείρισης της γνώσης. β)Διευκολύνουν τη δια-λειτουργικότητα συστημάτων. Για παράδειγμα αν έχουμε n διαφορετικές γλώσσες (n διαφορετικούς δρώντες παράγοντες), αντί να σχεδιάσουμε μια μετάφραση για κάθε ζεύγος από αυτούς (δηλαδή (n-1)! μεταφράσεις) σχεδιάζουμε μεταφράσεις από κάθε γλώσσα προς την κοινή οντολογία

---

[41] Για όσα πούμε πάνω στις οντολογίες παραπέμπουμε στο Ontologies: Principles, Methods and Applications των M.Uschold, M. Gruninger (1996).

(δηλαδή μόνο η). γ) Διευκολύνουν τον ίδιο το σχεδιασμό και την ανάπτυξη συστημάτων λογισμικού, βοηθώντας σε προδιαγραφές (specification) συστημάτων, προσφέρουν έναν «ημι-αυτοματοποιημένο έλεγχο συνέπειας» συστημάτων, και προσφέρουν δυνατότητες επαναχρησιμοποίησης.

Στη δουλειά των Kutz, κ.α[42] υπάρχει προηγούμενο για την υλοποίηση της εννοιολογικής μίξης, μέσω ενός είδους μίξης οντολογιών. Παραπέμπει ρητά στη δουλειά του Goguen στην χρήση αλγεβρικής σημειωτικής για τη μαθηματικοποίηση της εννοιολογικής μίξης.

Τι είναι η OWL;

Η OWL (Ontology Web Language) είναι μία από τις βασικές γλώσσες για να γράψει και να χειριστεί κανείς οντολογίες. Σύμφωνα με την W3C (και την ομάδα ανάπτυξης της OWL) «είναι μια γλώσσα σημασιολογικού ιστού σχεδιασμένη για την αναπαράσταση πλούσιων και σύνθετων γνώσεων σχετικά με πράγματα, ομάδες πραγμάτων, και τις σχέσεις μεταξύ τους». Οι οντολογίες OWL επιτρέπουν την κατασκευή δομών δήλωσης και περιγραφής γνώσης με τρόπο που είναι κατανοητός τόσο από ανθρώπους όσο και από μηχανές, διευκολύνοντας τη διαλειτουργικότητα και την ανταλλαγή γνώσης.

Η δημιουργία οντολογιών σε OWL επιτρέπει την ακριβή αναπαράσταση και διαμοιρασμό γνώσης με σαφή σημασιολογική δομή, διευκολύνοντας τη διαλειτουργικότητα μεταξύ συστημάτων και την αυτοματοποιημένη επεξεργασία πληροφοριών.

Η OWL περιέχει την Υπόθεση Ανοικτού Κόσμου (Open World Assumption - OWA), που είναι η αρχή που υποθέτει ότι η γνώση μπορεί να είναι ατελής ή ανολοκλήρωτη. Είναι η υπόθεση ότι υπάρχει ισχυρισμός ο οποίος δεν είναι γνωστός αλλά είναι αληθής. Με άλλα λόγια, αν κάτι δεν έχει δηλωθεί, δεν θεωρείται αυτομάτως ψευδές. Χρησιμοποιείται στην OWL (Web Ontology Language) για να επιτρέψει την επέκταση και ενημέρωση της γνώσης χωρίς περιορισμούς. Πρόκειται για την παραδοχή ότι μια οντολογία μπορεί να επεκταθεί και σε άλλες οντολογίες, δηλαδή να μην είναι κλειστό το σύνολο των κλάσεων. Πρακτικά αποτρέπει από μια οντολογία να ξαναγράψει τις πληροφορίες των άλλων οντολογιών, έτσι ώστε να μπορούν να προστίθενται οντολογίες σε ένα σύστημα χωρίς να υπάρχει καμία ανησυχία για την ακεραιότητα των πληροφοριών της.

Η υπάρχουσα έκδοση της OWL σήμερα είναι η OWL 2. Η OWL 2 συνδέεται με την Owlready2, μια Python βιβλιοθήκη που έχει σχεδιαστεί για να διευκολύνει τη διαχείριση, επεξεργασία και χρήση οντολογιών που είναι γραμμένες σε OWL 2.

[42] Βλέπε “Ontological blending in DOL” (2012).

Τι είναι η Owlready2;

Η Owlready2 είναι ένα πακέτο για Python ώστε να επιτρέψει τον προγραμματισμό με βάση τις οντολογίες σ αυτήν. Δίνει τη δυνατότητα φόρτωσης, διαχείρισης, και επεξεργασίας οντολογιών OWL 2 ως αντικείμενα της Python, παρέχοντας παράλληλα ένα εύκολο στη χρήση API που «επιτρέπει διαυγή πρόσβαση σε οντολογίες OWL"[43]. Η βιβλιοθήκη αυτή επιτρέπει την αλληλεπίδραση με οντολογίες OWL μέσω Python, προσφέροντας τη δυνατότητα να δημιουργείς, να επεξεργάζεσαι και να εκτελείς ερωτήματα σε οντολογίες χρησιμοποιώντας Python κώδικα. Η Owlready2 επίσης χρησιμοποιεί τη λογική ανοιχτού κόσμου, κάτι το οποίο είναι συμβατό με τις αρχές της OWL.

2. Προγραμματιστικό Περιβάλλον

Ο στόχος μας είναι να μεταφέρουμε την υλοποίηση της εννοιολογικής μίξης σε Python. Η επιλογή μας αυτή προσμετρά μια σειρά πλεονεκτημάτων της Python στη σημερινή εποχή.

Μεταφέροντας οντολογίες από OWL σε Python παρέχεται η δυνατότητα αξιοποίησης της ευελιξίας και της υπολογιστικής ισχύος της Python ώστε να αναπτυχθούν πιο δυναμικές και επεκτάσιμες εφαρμογές (ειδικά λόγω της ευρύτατης χρήσης της Python). Με αυτόν τον τρόπο επιτυγχάνεται μεγαλύτερη λειτουργικότητα και προσαρμοστικότητα στην ενσωμάτωση οντολογικών δεδομένων σε ευρύτερα προγραμματιστικά πλαίσια αλλά και στη διασύνδεσή τους με άλλων ειδών δεδομένα, δομές, και αλγόριθμους. Γενικά η Python ξεχωρίζει για την πλούσια βιβλιοθήκη που διαθέτει αλλά και ως προγραμματιστικό περιβάλλον διευκολύνει την αυτοματοποίηση διαδικασιών και την βελτιστοποίηση των εφαρμογών που χρησιμοποιούν οι οντολογίες.

Τέλος, η Python χρησιμοποιείται σήμερα σε μια σειρά προβλήματα και εφαρμογές της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Βασικό βήμα στην περεταίρω ανάπτυξη της αλγεβρικής σημειωτικής και της εννοιολογικής μίξης ως αντικείμενα είναι η χρήση τους σε γλωσσικά μοντέλα. Αυτό σημαίνει, πως στην κατεύθυνση αυτή, η μεταφορά των διαδικασιών αυτών σε Python, έστω σε ένα πρώτο και πειραματικό επίπεδο, είναι προαπαιτούμενο και μπορεί να ανοίξει το δρόμο στην ευρύτερη χρήση τους στο μέλλον, ειδικά σε σχέση με εφαρμογές ανάλυσης κειμένου.

[43] Από το documentation της owlready2, https://owlready2.readthedocs.io/en/latest/

Το περιβάλλον που θα αξιοποιήσουμε

Ο κώδικας μας θα αξιοποιεί ένα πακέτο για την λειτουργία συστήματος οντολογιών στην Python. Το συγκεκριμένο πακέτο owlready2[44] μας δίνει την δυνατότητα να εφαρμόσουμε τις σχέσεις μεταξύ τάξεων και αντικειμένων (classes, instances) με την χρήση απλών αντικειμένων (objects) στην γλώσσα της Python.

Η Python είναι μια ερμηνευμένη γλώσσα προγραμματισμού (interpreted). Εκτελείται δηλαδή γραμμή προς γραμμή. Στην μεταφορά των οντολογιών, και στην χρήση δηλαδή του OWL 2 υπάρχει η απαίτηση οι σχέσεις που χρησιμοποιούμε να είναι ορισμένες. Παράλληλα για να μπορεί να δημιουργηθεί ο ζητούμενος ιστός σχέσεων μεταξύ των οντολογιών χρειάζεται ο κώδικας να δομηθεί σε μια μορφή όπου κάθε κομμάτι (οντολογία) παραμένει και μπορεί να ξανακληθεί ή αλλιώς να ξαναχρησιμοποιηθεί σε επόμενο σημείο του κώδικα. Για να υλοποιηθούν τα ζητούμενα όπως αναφέρθηκαν παραπάνω χρειάστηκε να δημιουργήσουμε κλάσεις, και να ορίσουμε αντικείμενα στην Python.

Owlready2, η μετατροπή της Python σε OWL

Όπως είπαμε η Owlready2 είναι το πακέτο που επιτρέπει πρόσβαση στη γλώσσα οντολογιών και στις οντολογίες που κατασκευάζονται σε αυτήν από την Python. Θα μελετήσουμε τις βασικές τεχνικές δυνατότητες που μας προσφέρει:
Η owlready2 περιέχει μεθόδους για την περιγραφή κλάσεων(classes), μεμονωμένες οντότητες των κλάσεων (individuals) και ιδιοτήτων(properties) αυτών.

Μια κλάση είναι απλά μια συγκεκριμένη περιγραφή ενός αντικειμένου, μιας ιδέας, μιας διαδικασίας μέσα στην οντολογία. Έτσι αρχίζει η προσπάθεια να περιγράψουμε τον κόσμο, με τον διαχωρισμό των διαφορετικών “εννοιών” σε διαφορετικές κλάσεις, έπειτα με το να τους δίνουμε χαρακτηριστικά-τις λεγόμενες ιδιότητες(properties)- που περιγράφουν καλύτερα την κάθε κλάση φέρνουμε την περιγραφή του κόσμου μας πιο κοντά στην πραγματικότητα.

Έπειτα φεύγουμε από την απλή περιγραφή του κόσμου και μεταφέρουμε την κλάση σε ένα πραγματικό αντικείμενο, μια οντότητα της κλάσης(instance). Η οντότητα της κλάσης είναι μέλος της κλάσης και κληρονομεί τα χαρακτηριστικά της κλάσης. Μπορούν να υπάρχουν πολλές οντότητες που είναι μέλη της ίδιας κλάσης (πχ. υπάρχουν πολλοί μαθητές σε μια τάξη και όλοι έχουν διαφορετικά χαρακτηριστικά αλλά όλοι έχουν την ιδιότητα του μαθητή, δηλαδή η κλάση είναι ο μαθητής ενώ ο κάθε μαθητής[πχ. Γιώργος, Μαρία,…] είναι οι οντότητες της κλάσης [individuals]), όπως και να έχουν επιπλέον χαρακτηριστικά που να τους περιγράφει παραπάνω(πχ. Ο κάθε μαθητής έχει διαφορετικό χρώμα μαλλιά, γράφει με διαφορετικό χέρι , κτλ.).

Οι ιδιότητες της κλάσης(properties) είναι αυτές που μετατρέπουν τις κατηγοριοποιήσεις που κάναμε παραπάνω σε μια πραγματική οντολογική γλώσσα και όχι απλά έναν τρόπο ταξινόμησης των πραγμάτων. Οι ιδιότητες είναι οι συνδετικοί δεσμοί μεταξύ των κλάσεων και των οντοτήτων των κλάσεων, είναι τα κριτήρια που βάζουμε για να δούμε αν μια οντότητα είναι μέρος μιας κλάσης. Αυτοί οι συνδετικοί δεσμοί δεν υπάρχουν όμως μονάχα για τις οντότητες αλλά και για τις κλάσεις μεταξύ τους, πρακτικά δημιουργώντας ένα εννοιολογικό δέντρο μεταξύ των κλάσεων (πχ. ένα δέντρο είναι και φυτό, αλλά ένα φυτό δεν είναι δέντρο).

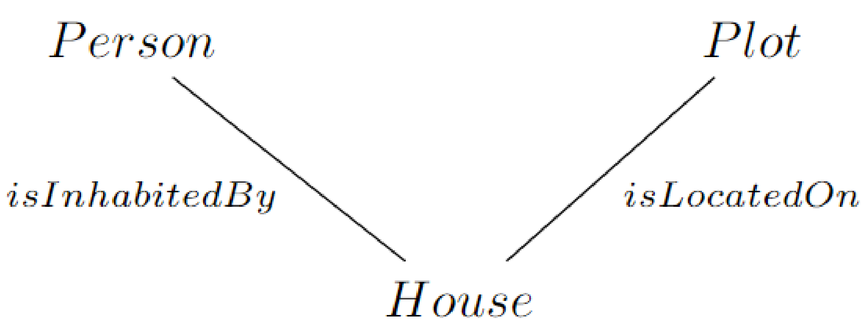
---

[44] https://owlready2.readthedocs.io/en/latest/index.html Το documentation της owlready2.

Παρακάτω παρουσιάζουμε ένα παράδειγμα[45] οντολογίας, μία οντολογία που αφορά το «Σπίτι», γραμμένη σε OWL:

```
Class: Artifact
Class: Capability
ObjectProperty: has_function
    Range: Capability
ObjectProperty: executes
    Range: Capability
ObjectProperty: is_located_on
Class: Person
Class: Plot
ObjectProperty: is_inhabited_by
    Domain: House
    Range:  Person
Class: ServeAsResidence
    SubClassOf: Capability
Class: ArtifactThatExecutesResidenceFunction
    EquivalentTo: Artifact that executes
                  some ServeAsResidence
    SubClassOf: is_inhabited_by some Person
Class: House
    SubClassOf: Artifact
        that is_located_on some Plot
        and has_function some
                   ServeAsResidence
```

Ας θυμηθούμε τώρα ακριβώς το παράδειγμα για το conceptual space «σπίτι» του Goguen από το προηγούμενο κεφάλαιο και θα δούμε ότι κατασκευάζεται η ίδια ακριβώς εικόνα, με ελαφρώς διαφορετικά ονόματα:



Το artifact και το capability είναι άλλες έννοιες των οποίων οι house και serveAsResidence αντίστοιχα είναι συγκεκριμένες περιπτώσεις. Από τη πλευρά της αλγεβρικής σημειωτικής είναι μοντέλα, από την πλευρά της OWL υποκλάσεις. Αν μελετήσουμε το «σπίτι» ως

[45] Το παράδειγμα αυτό βρίσκεται στο «Blending in the Hub, Towards a computational concept invention platform» (Kutz κ.α, 2014).

εννοιολογικό χώρο (Αντί για οντολογία), οι isInhabitedBy και isLocatedOn είναι διμελείς σχέσεις μεταξύ των σταθερών House, Plot, Person.
Μέσω αυτού του απλού παραδείγματος βλέπουμε και πρακτικά τη σύνδεση μεταξύ εννοιολογικού χώρου και οντολογίας.

Κεφάλαιο 4: Εφαρμογή της Πολυμεσικής Έκθεσης

Το πρόβλημα που θα χρησιμοποιήσουμε ονομάζεται Πολυμεσική Έκθεση. Αποτελεί κάτι που θα ονομάζαμε «ερευνητικό παράδειγμα», δηλαδή ένα κατασκευασμένο «σενάριο», ένα υποθετικό πρόβλημα, που επιτρέπει τον έλεγχο και τη μελέτη των ερευνητικών υποθέσεων και μεθόδων, και την έκθεση των δυνατοτήτων μιας μεθοδολογίας σε ένα πιο απτό επίπεδο. Το σενάριο αυτό αναπτύχθηκε, με διαφορετικές μορφές, και εμείς θα ασχοληθούμε με τη μορφή που παρουσιάζει ο Παπαθεοδώρου[46].

Μια πολυμεσική έκθεση λοιπόν είναι το πρόβλημα του να εκθέσουμε σε συνδυασμό εκθέματα από μια δεξαμενή που περιλαμβάνει μία ή περισσότερων ειδών ετερογένειες, για παράδειγμα διαφορετικό μέσο έκφρασης (multimodality). Η επιλογή αυτή στηρίζεται σε ένα ή περισσότερα ιεραρχημένα κριτήρια (πχ κοινή θεματολογία, κοινό ύφος, ίδιος καλλιτέχνης ή κοινό καλλιτεχνικό ρεύμα). Σκοπός μας λοιπόν είναι η παραγωγή μιας πολυμεσικής έκθεσης που θα ελέγχει τις ιδιότητες των οντολογιών και την μίξη αντικειμένων (σημείων) από αυτές σε ένα νέο έκθεμα-σημείο.

Το παράδειγμά μας αφορά ένα σύστημα τριών οντολογιών από διαφορετικών ειδών έργα τέχνης: πίνακες ζωγραφικής, μουσική και λογοτεχνικά έργα (ποιήματα). Τα αντικείμενα αυτά θα έχουν χαρακτηριστικά (καλλιτέχνης, εικαστικό είδος, κτλ) και σχέσεις. Το blend από αυτές τις οντολογίες θα δίνει ένα νέο αντικείμενο που θα ανήκει σε μία νέα οντολογία (έκθεμα).

Η χρήση της οντολογίας (και της OWL) στο πρόβλημά μας έχει λόγο. Η οντολογία μπορεί να βοηθήσει στην οργάνωση και κατηγοριοποίηση πολύπλοκων συστημάτων. Για παράδειγμα μια οντολογία μπορεί να αξιοποιηθεί σε μια βιβλιοθήκη, που να βοηθά τον χρήστη να δει έργα του ίδιου εικαστικού ρεύματος, είτε αυτά είναι ποιήματα, βιβλία ή πίνακες ζωγραφικής. Αντίστοιχα αυτό μπορεί να γίνει σε πολύ μεγαλύτερη κλίμακα αν δημιουργήσουμε μια οντολογία στο διαδίκτυο, όπου κάθε νέα ιστοσελίδα θα μπορούσε να προστίθεται (αξιοποιώντας την open world assumption) στην θάλασσα πληροφοριών που ήδη υπάρχει.

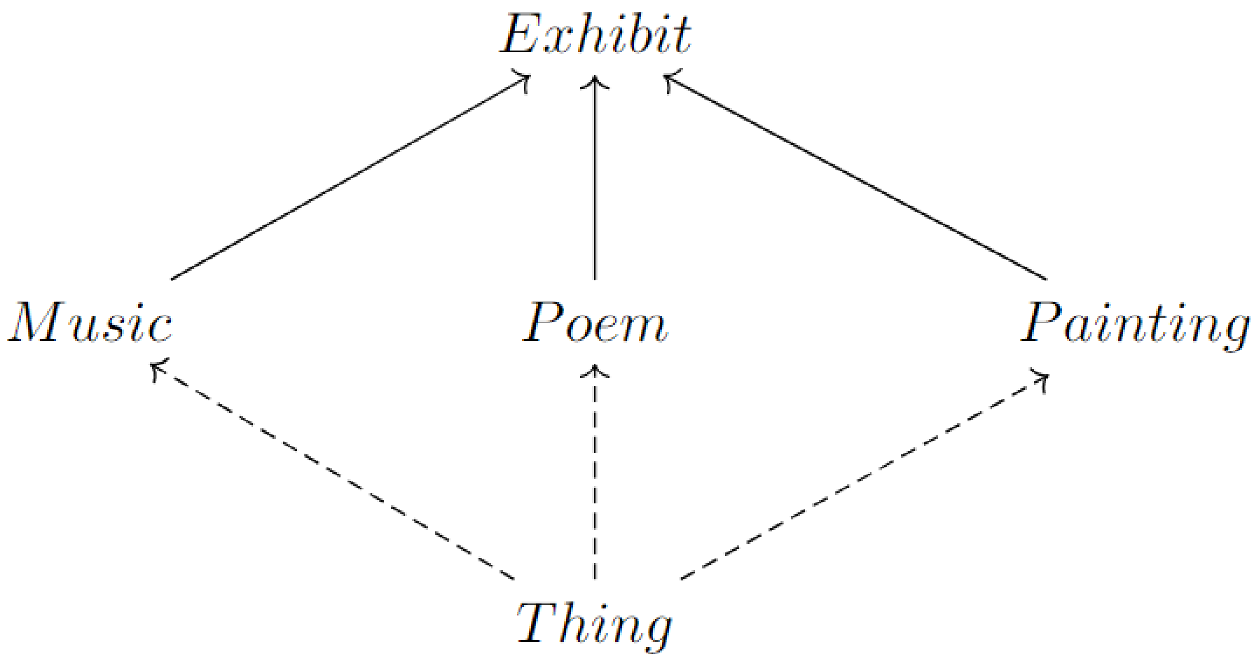
1.Σχεδιασμός και πηγές άντλησης δεδομένων

Για να φτιάξουμε μια τέτοια οντολογία θα χρειαστούμε μια ολοκληρωμένη βάση δεδομένων με διαφορετικά είδη τέχνης (ζωγραφικής, μουσικής, λογοτεχνικής, κτλ.) Εφόσον βασιζόμαστε στην Open World Assumption μπορούμε να κατασκευάσουμε σε διαφορετικά προγράμματα την κάθε οντολογία, να μπορούμε να τις συνδέσουμε στο τέλος.

Για την πραγματοποίηση του προβλήματος επιλέγουμε να φτιάξουμε τρεις φακέλους, μια οντολογία για την ποίηση, μια για την μουσική και μία για την ζωγραφική. Έπειτα θα κατασκευάσουμε μια οντολογία που ενώνει όλες τις οντολογικές βάσεις δεδομένων και μπορεί να ομαδοποιήσει όλα τα δεδομένα.

Το παρακάτω σχήμα περιγράφει τον σχεδιασμό μας:

[46] Παπαθεοδώρου Φ. Τυπικές Οντολογίες και Εφαρμογές. 2015

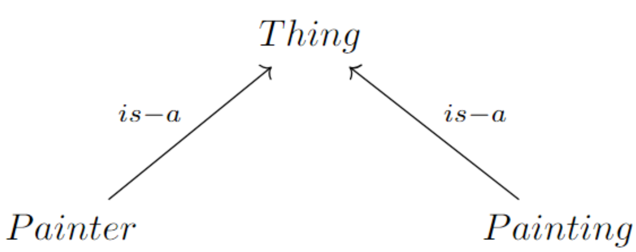


Στις παρακάτω οντολογίες προσθέτουμε χειρόγραφα τις πληροφορίες αντί να χρησιμοποιούμε κάποια αυτοματοποιημένη είσοδο δεδομένων (να του δίνουμε ένα αρχείο .csv ή .json και να φτιάχνει ολόκληρη την οντολογία έτσι) αυτό γίνεται διότι ξεφεύγει το εύρος αυτής της εργασίας και δεν είναι παρά ένα προγραμματιστικό πρόβλημα που λύνεται πολύ εύκολα δεδομένου της δημοσιότητας της Python.

2.Η ανάλυση του κώδικα

Painting

Οφείλουμε να καθορίσουμε τις θεμελιώδεις ιδιότητες του πεδίου για την οντολογία που σκοπεύουμε να κατασκευάσουμε, δηλαδή τι είναι κάθε ιδιότητα που θα περιγράψουμε μέσα στην οντολογία. Για παράδειγμα το αρχείο Painting.py θα περιέχει τον κώδικα της οντολογίας για τα εικαστικά έργα τέχνης. Θα τα κατηγοριοποιήσουμε με βάση τί έννοιες αξιοποιεί, δηλαδή θα έχουμε τον όρο Painter για να κατηγοριοποιούμε τους καλλιτέχνες και τον όρο Painting για να κατηγοριοποιήσουμε τον ίδιο τον πίνακα.



Έπειτα θα δημιουργήσουμε τις σχέσεις μεταξύ των αντικειμένων[47]. Δηλαδή, η σχέση του καλλιτέχνη με το έργο τέχνης του είναι πως το έχει δημιουργήσει εκείνος, άρα

Painter. painted() = Painting

Αντιστοιχα το έργο τέχνης μπορούμε να πούμε πως είναι ζωγραφισμένο από τον καλλιτέχνη, οπότε

Painting.isPaintedBy()= Painter

Αυτή την σχέση την περιγράφουμε με το παρακάτω διάγραμμα.

$$\begin{tikzcd} Painter \arrow[rr, bend left, "painted()"] & & Painting \arrow[ll, bend left, "isPaintedBy()"] \end{tikzcd}$$

Σημασία έχει να επισημάνουμε πως το .isPaintedBy() είναι η αντίστροφη συνάρτηση του painted() για αυτό και αξιοποιούμε την ικανότητα του owlready2 να κατηγοριοποιήσει αυτό το γεγονός με το χαρακτηριστικό inverse_property σε όποια από τις δύο συναρτήσεις.

Τέλος οφείλουμε να αναδείξουμε το συντακτικό (syntax) του owlready2. Πρόκειται για το πολύ πιο καθαρό συντακτικό σε σχέση με άλλες γλώσσες που είχαν προσπαθήσει να υλοποιήσουν συστήματα οντολογιών και αυτό οφείλεται σε μεγάλο βαθμό στην Python. Κατ'επέκταση το owlready2 είναι ένα αρκετά εύκολο πακέτο στην χρήση, με ένα πλήρες documentation (στο οποίο παραπέμψαμε νωρίτερα) που πέρα από ευανάγνωστο είναι και πολύ περιληπτικό ως προς όλες τις μεθόδους που χρησιμοποιεί το πακέτο. Όπως αναφέρθηκε και πιο πάνω κάθε αντικείμενο και σχέση αντικειμένων αντιμετωπίζεται ως μια κλάση(δεν είναι ο πιο υπολογιστικά οικονομικός τρόπος να θέσεις το πρόγραμμα, αλλά είναι σίγουρα ο πιο ανθρώπινα κατανοητός) με τα χαρακτηριστικά του να αποθηκεύονται ως στοιχεία της κλάσης και τις σχέσεις της με άλλες κλάσεις να αποθηκεύονται ως μεταβλητές. Αυτές οι μεταβλητές αποθηκεύονται μέσω των σχέσεων κλάσεων (όπως αυτές που αναλύσαμε από πάνω) και προσδίδουν δύο χαρακτηστικά στην κλάση που καλούνται, το domain και το range.

[47] Σχέσεις μεταξύ αντικειμένων όσον αφορά την Python, αλλά όσον αφορά την οντολογία σχέση μεταξύ κλάσεων, πλαίσιο των εννοιολογικών χώρων οι σχέσεις μεταξύ 2 στοιχείων του χώρου.

```python
class painted(ObjectProperty , FunctionalProperty):
    domain = [Painter]
    range = [Painting]

class is_Painted_By(ObjectProperty , FunctionalProperty):
    domain = [Painting]
    range = [Painter]
    inverse_property = painted
```

Το domain είναι το αντικείμενο από το οποίο ξεκινά το βέλος της σχέσης και το range έχει να κάνει με το πού καταλήγει, είναι με ένα τρόπο ανάλογο του domain/codomain.

Το Painting.py είναι από τα πιο απλά και κατανοητά παραδείγματά που θέτουμε σε αυτή την εργασία και όλα τα επόμενα παραδείγματα θα περιλαμβάνουν όλα τα στοιχεία του κώδικα που αξιοποιούμε στο Painting.py, οπότε δεν θα αναλυθούμε πολύ σχολαστικά στα ίδια φαινόμενα και στα παρακάτω παραδείγματα.

Poem

Συνεχίζουμε στην δεύτερη οντολογία που θα περιέχει στοιχεία της ποίησης, την οποία θα ονομάσουμε Poem.py. Θα εισάγουμε στον κώδικα ως παράδειγμα το ποίημα της Adrienne Rich, Mourning Picture Poem. Προφανώς η οντολογία αυτή θα έχει τις κλάσεις Poem και Poet με τον ίδιο τρόπο που είχε η Painting.py. Παράλληλα θα θέσουμε ως ιδιότητες (Properties) των κλάσεων αυτών την περιγραφή του ποιήματος και το ίδιο το ποιήμα, με τις κλάσεις Description και Poem_Lines.

```python
    class Description(Poem):
        description = "A man and a woman in black clothes sitting in a mahogany
 chair and a cane rocker."
    class Poem_Lines(Poem):
        poem_lines =''' They have carried the mahogany chair and the cane rocker
                        out under the lilac bush,
                        and my father and mother darkly sit there, in black
                        clothes.Our clapboard house stands fast on its hill,
                        my doll lies in her wicker pram
                        gazing at western Massachusetts. This was our world.
                        I could remake each shaft of grass feeling its rasp
                        on my fingers,draw out the map of every lilac leaf
                        or the net of veins on my father&apos;s
                        grief-tranced hand. Out of my head, half-bursting,
                        still filling, the dream condenses-- shadows, crystals,
                        ceilings, meadows, globes of dew.
                        Under the dull green of the lilacs, out in the light
                        carving each spoke of the pram, the turned porch-pillars,
                        under high early-summer clouds, I am Effie, visible and
                        invisible, remembering and remembered.'''
```

Έπειτα μαζί με τις εννοιολογικές σχέσεις wrote, isWrittenBy που θα αναλυθούν όπως στο παραπάνω παράδειγμα θα έχουμε και τις σχέσεις describes, hasPoemLines με domain την κλάση Poem (εφόσον μόνο ποιήματα μπορούν να έχουν αυτές τις ιδιότητες) και range κάθε αλφαριθμητικό(string) που μπορεί να δώσει ο χρήστης. Πρακτικά έτσι μπορούμε να του δώσουμε το κείμενο περιγραφής του ποιήματος.

```
    class isWrittenBy(ObjectProperty, FunctionalProperty):
        domain = [Poem]
        range = [Poet]

    class describes(ObjectProperty, FunctionalProperty):
        domain = [Poem]
        range = [str]# a string object is used to describe the target of this class

    class hasPoemLines(ObjectProperty, FunctionalProperty):
        domain = [Poem]
        range = [str]# a string object is used to describe the target of this class

    class wrote(ObjectProperty, FunctionalProperty):
        domain = [Poet]
        range = [Poem]
        inverse_property = isWrittenBy # we declare what class is the inverse of
this one
```

Τέλος, αρχικοποιούμε την ποιήτρια και το ποίημα στην οντολογία και δημιουργούμε τις εννοιολογικές σχέσεις μεταξύ τους. Στην παρακάτω εικόνα έχουμε την αρχικοποίηση με το κατάλληλο documentation.

```
    #it is important to create all the properties before we create any relations
between two objects
    #-----------------------------------------------------------------------------
    #we create the objects and give them their attributes
    AdrienneRich = Poet()
    MourningPicture = Poem( hasPoemLines.value(Poem_Lines.poem_lines) &
describes.value(Description.description) )
    #then we create the relations between them
    MourningPicture.isWrittenBy = AdrienneRich
    AdrienneRich.wrote = MourningPicture
```

Music

Στο πρόγραμμα για τα μουσικά κομμάτια της έκθεσης θα υλοποιήσουμε όπως και στην οντολογία Poem αντίστοιχα με κλάσεις Song και Composer και τις ιδιότητες composed, composedBy και isOfGenre που αντίστοιχα, δείχνουν την σχέση του τραγουδιού και του καλλιτέχνη αλλά και το είδος στο οποίο ανήκει.

Exhibit

Ολοκληρώνοντας την τελευταία οντολογία είμαστε έτοιμοι να τις συνδέσουμε όλες μαζί. Ετοιμάζουμε όλες τις κλάσεις που είχαμε κάνει πιο πάνω, Art, Artist , Exhibit, κτλ. Αυτός ο κώδικας φτιάχνει την ραχοκοκαλιά του συστήματος αλλά χρειάζεται και κάτι παραπάνω για να ενώσει όλες τις οντολογίες μαζί. Εδώ αξιοποιούμε την import ontology function που μας παρέχει το owlready2. Έτσι με τον πάρακάτω κώδικα

```
onto.imported_ontologies.append(owlready_ontology)
```

στην αρχή του προγράμματος. Πρακτικά αξιοποιεί την γλώσσα της Python και το σύστημα που έχει ήδη χτίσει για να μεταφέρει modules κώδικα από το ένα project στο άλλο.

3.Ο κώδικας

Music.py

```python
from owlready2 import * #the package for manipulating ontologies
#-------------------------------------------------------------------------------
onto = get_ontology("http://test.org/onto.owl") # we create an empty ontology to use in this example
#-------------------------------------------------------------------------------
#we open the ontology
with onto:
    #-------------------------------------------------------------------------
    #we create the classes that represent the objects that we are going to use
    class Song(Thing): pass
    class Composer(Thing): pass
    #we classify the physical attributes of the objects
    class Genre(Song): pass
    class Classic(Genre): pass
    #--------------------------------------------
    #we classify the relations between the classes and the objects
    #every class has a source and a target(or the domain and range respectively) depicting from where,
    #to where the attribute is going
    class composed(ObjectProperty, FunctionalProperty):
        domain = [Composer]
        range = [Song]

    class composedBy(ObjectProperty, FunctionalProperty):
        domain = [Song]
        range = [Composer]
        inverse_property = composed

    class isOfGenre(ObjectProperty, FunctionalProperty):
        domain = [Song]
        range = [Genre]
    #it is important to create all the properties before we create any relations between two objects
    #-------------------------------------------------------------------------
    #we create the objects and give them their attributes
    DimitriShostakovich = Composer()
    StringQuartetNo15 = Song(isOfGenre.value(Classic) & composedBy.value(DimitriShostakovich))
    #then we create the relations between them
    DimitriShostakovich.composed = StringQuartetNo15
    StringQuartetNo15.composedBy = DimitriShostakovich
#-------------------------------------------------------------------------------
#print to see the world we created
print("Dimitri Shostakovich is an:",DimitriShostakovich.is_a)
print("Dimitri Shostakonich composed an:",DimitriShostakovich.composed)
print("String Quaertet No15 is an:",StringQuartetNo15.is_a)
print("String Quartet No15 was composed by:",StringQuartetNo15.composedBy)
#closing the world
close_world(Song)
```

Poem.py

```python
from owlready2 import * #the package for manipulating ontologies
#-----------------------------------------------------------------------------
onto = get_ontology("http://test.org/onto.owl") # we create an empty ontology to use in this example
#-----------------------------------------------------------------------------
#we open the ontology
with onto:
    #-------------------------------------------------------------------------
    #we create the classes that represent the objects that we are going to use
    class Poem(Thing): pass
    class Poet(Thing): pass
    #we classify the physical attributes of the objects
    class Description(Poem):
        description = "A man and a woman in black clothes sitting in a mahogany chair and a cane rocker."
    class Poem_Lines(Poem):
        poem_lines =''' They have carried the mahogany chair and the cane rocker
                        out under the lilac bush,
                        and my father and mother darkly sit there, in black
                        clothes.Our clapboard house stands fast on its hill,
                        my doll lies in her wicker pram
                        gazing at western Massachusetts. This was our world.
                        I could remake each shaft of grass feeling its rasp
                        on my fingers,draw out the map of every lilac leaf
                        or the net of veins on my father&apos;s
                        grief-tranced hand. Out of my head, half-bursting,
                        still filling, the dream condenses-- shadows, crystals,
                        ceilings, meadows, globes of dew.
                        Under the dull green of the lilacs, out in the light
                        carving each spoke of the pram, the turned porch-pillars,
                        under high early-summer clouds, I am Effie, visible and
                        invisible, remembering and remembered.'''
    #--------------------------------------------------------------------------
    #we classify the relations between the classes and the objects
    #every class has a source and a target(or the domain and range respectively) depicting from where,
    #to where the attribute is going
    class isWrittenBy(ObjectProperty, FunctionalProperty):
        domain = [Poem]
        range = [Poet]

    class describes(ObjectProperty, FunctionalProperty):
        domain = [Poem]
        range = [str]# a string object is used to describe the target of this class

    class hasPoemLines(ObjectProperty, FunctionalProperty):
        domain = [Poem]
        range = [str]# a string object is used to describe the target of this class

    class wrote(ObjectProperty, FunctionalProperty):
        domain = [Poet]
        range = [Poem]
        inverse_property = isWrittenBy # we declare what class is the inverse of this one
    #it is important to create all the properties before we create any relations between two objects
    #-------------------------------------------------------------------------
    #we create the objects and give them their attributes
```

```
    AdrienneRich = Poet()
    MourningPicture = Poem( hasPoemLines.value(Poem_Lines.poem_lines) & describes.value(Description.description) )
    #then we create the relations between them
    MourningPicture.isWrittenBy = AdrienneRich
    AdrienneRich.wrote = MourningPicture
#-----------------------------------------------------------------------------
#print to see the world we created
print("Adrienne Rich was an:",AdrienneRich.is_a)
print("Adrienne Rich wrote:", Poem_Lines.poem_lines)
print(Description.description)
print("The Mourning Picture is an:",MourningPicture.is_a)
print("The Mourning Picture was written by:",MourningPicture.isWrittenBy)
#------------------------------------------------------------------------------
#closing the world
close_world(Poem)

```

Painting.py

```
from owlready2 import * #the package for manipulating ontologies
#---------------------------------------------------------------------------
onto = get_ontology("http://test.org/onto.owl") # we create an empty ontology to use in this example
#---------------------------------------------------------------------------
#we open the ontology
with onto:
    #-----------------------------------------------------------------------
    #we create the classes that represent the objects that we are going to use
    class Painter(Thing): pass
    class Painting(Thing): pass
    #we classify the physical attributes of the objects
    class Depiction(Painting):
        description1 = "Elmer&apos; s wife wearing black clothes, sitting in a chair"
        description2 = "Elmer&apos; s daughter Effie, along with her pet lamb and kitten"
        description3 = "Edwin Romanzo Elmer wearing black clothes, sitting in a chair"
    #------------------------------------------------------------------------
    #we classify the relations between the classes and the objects
    #every class has a source and a target(or the domain and range respectively) depicting from where,
    #to where the attribute is going
    class depicts(ObjectProperty, FunctionalProperty):
        domain = [Painting]
        range = [Depiction]

    class painted(ObjectProperty , FunctionalProperty):
        domain = [Painter]
        range = [Painting]

    class is_Painted_By(ObjectProperty , FunctionalProperty):
        domain = [Painting]
        range = [Painter]
        inverse_property = painted
```

```python
    #it is important to create all the properties before we create any relations between two objects
    #--------------------------------------------------------------------------
    #we create the objects and give them their attributes
    MourningPicture = Painting( [depicts.value(Depiction.description1),
                                depicts.value(Depiction.description2),
                                depicts.value(Depiction.description3)] )
    EdwinRomanzoElmer = Painter()
    #then we create the relations between them
    MourningPicture.is_Painted_By =  EdwinRomanzoElmer
    EdwinRomanzoElmer.Painted = MourningPicture
#------------------------------------------------------------------------------
#print to see the world we created
print("THe Mourning Picture is an:",MourningPicture.is_a)
print("The Mourning Picture was painted by:",MourningPicture.is_Painted_By)
print("Edwin Romanzo Elmer was an:",EdwinRomanzoElmer.is_a)
print("Edwin Romanzo Elmer painted:",EdwinRomanzoElmer.painted)
#closing the world
close_world(Painting)
```

exhibit.py

```python
from owlready2 import * #the package for manipulating ontologies
#------------------------------------------------------------------------------
onto = get_ontology("http://test.org/onto.owl") # we create an empty ontology to use in this example
#------------------------------------------------------------------------------
#we open the ontology

with onto:
    #----------------------------------------------------------------
    #we create the classes that represent the objects that we are going to use
    class Art(Thing): pass
    class Artist(Thing): pass
    class ValuePartition(Thing): pass
    #we classify the physical attributes of the objects
    class Exhibit(Art): pass
    #we create instances of the values used by the classes
    No = ValuePartition()
    Yes = ValuePartition()
    #-----------------------------------------------------------------
    #we classify the relations between the classes and the objects
    #every class has a source and a target(or the domain and range respectively) depicting from where,
    #to where the attribute is going
    class RelatedToMourningPictureValuePartition(ObjectProperty , FunctionalProperty):
        domain = [Art]
        range = [Yes , No]
        equivalent_to = ValuePartition

    class isRelatedToMourningPicture(ObjectProperty, FunctionalProperty):
        domain = [Art]
        range = [Yes, No]#needs to have the same range with RelatedToMourningPictureValuePartition
```

```python

    #it is important to create all the properties before we create any relations between two objects
    #-------------------------------------------------------------------
    #we create the objects and give them their attributes
    MourningPictureExhibit = Exhibit()
    Composer = Artist()
    Song = Art()
    Painter = Artist()
    Painting = Art()
    Poet = Artist()
    Poem = Art()
    #-------------------------------------------------------------------
    Song.isRelatedToMourningPicture = Yes
    Painting.isRelatedToMourningPicture = Yes
    Poem.isRelatedToMourningPicture = Yes
#-----------------------------------------------------------------------
#print to see the world we created
print("The Mourning Picture Exhibit is an :",MourningPictureExhibit.is_a)
print("The Composer is an :",Composer.is_a)
print("The Song is an :",Song.is_a)
print("The Painter is an :",Painter.is_a)
print("The Painting is an :",Painting.is_a)
print("The Poet is an :",Poet.is_a)
print("The Poem is an :",Poem.is_a)
#closing the world
close_world(Exhibit)
```

Ο κώδικας επιστρέφει τα παρακάτω:

Music:

```
Dimitri Shostakovich is an: [onto.Composer]

Dimitri Shostakonich composed an: onto.onto.isOfGenre.value(onto.Classic) & onto.composedBy.value(onto.composer1)

String Quaertet No15 is an: [onto.Song]

String Quartet No15 was composed by: onto.composer1
```

Poem:

```
Adrienne Rich was an: [onto.Poet]

Adrienne Rich wrote:  They have carried the mahogany chair and the cane rocker
                        out under the lilac bush,
                        and my father and mother darkly sit there, in black
                        clothes.Our clapboard house stands fast on its hill,
                        my doll lies in her wicker pram
                        gazing at western Massachusetts. This was our world.
                        I could remake each shaft of grass feeling its rasp
                        on my fingers,draw out the map of every lilac leaf
                        or the net of veins on my father&apos;s
                        grief-tranced hand. Out of my head, half-bursting,
                        still filling, the dream condenses-- shadows, crystals,
                        ceilings, meadows, globes of dew.
                        Under the dull green of the lilacs, out in the light
                        carving each spoke of the pram, the turned porch-pillars,
                        under high early-summer clouds, I am Effie, visible and
                        invisible, remembering and remembered.

A man and a woman in black clothes sitting in a mahogany chair and a cane rocker.
The Mourning Picture is an: [onto.Poem]

The Mourning Picture was written by: onto.poet1
```

Painting:

```
THe Mourning Picture is an: [onto.Painting]

The Mourning Picture was painted by: onto.painter1

Edwin Romanzo Elmer was an: [onto.Painter]

Edwin Romanzo Elmer painted: onto.[onto.depicts.value('Elmer&apos; s wife wearing black clothes, sitting in a chair'), onto.depicts.value('Elmer&apos; s daughter Effie, along with her pet lamb and kitten'), onto.depicts.value('Edwin Romanzo Elmer wearing black clothes, sitting in a chair')]
```

Exhibit:

```
The Mourning Picture Exhibit is an : [onto.Exhibit]

The Composer is an : [onto.Artist]

The Song is an : [onto.Art]

The Painter is an : [onto.Artist]

The Painting is an : [onto.Art]

The Poet is an : [onto.Artist]

The Poem is an : [onto.Art]
```

ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως βλέπουμε παραπάνω, ο κώδικας του exhibit μας δίνει έναν «χάρτη» μεταξύ των αντικειμένων που εισάγαμε. Στην πραγματικότητα, κάτι τέτοιο δεν φαίνεται πολύ χρήσιμο με μόνο 3 αντικείμενα (ένα για κάθε οντολογία), αλλά εκεί που θα επιδείξει τη χρησιμότητά του είναι αν εισάγουμε παραπάνω αντικείμενα (είτε «χειροκίνητα» είτε αντλώντας από μια βάση δεδομένων), καθώς το πλήθος των αντικειμένων δεν επιτρέπει να αποκτήσουμε μια σταθερή εποπτεία τους τη στιγμή που τα μελετάμε. Είναι διαφορετικό κανείς να έχει 3 αντικείμενα με κάποιες βασικές ιδιότητες μπροστά του από ότι μία βάση δεδομένων εκατοντάδων ή χιλιάδων.

Επιπλέον, στην περίπτωση που είχαμε πολλά αντικείμενα θα είχε νόημα να μιλήσουμε για επιλογή. Η κάθε οντολογία exhibit έχει μία συγκεκριμένη εξειδίκευση. Για παράδειγμα εμείς έχουμε το isRelatedToMourningPicture. Μία έυκολη γενίκευση (εφόσον έχουμε όμως πολλά δεδομένα ως εισαγωγή) είναι να περάσουμε μία ακόμα ιδιότητα στα έργα τέχνης, για παράδειγμα το θέμα (“Theme”). Το θέμα θα μπορούσε να είναι ένα input τύπου string array, το οποίο εισάγει μια σειρά λέξεις που αντιστοιχούν στο θέμα του έργου. Έτσι θα μπορούσαμε να προσαρμόσουμε το exhibit με χαρακτηριστικό isRelatedToMourningPicture (δηλαδή την οντολογία που κατασκευάζει ένα πολυμεσικό έκθεμα γύρω από το Mourning Picture) ώστε να επιλέγει άλλα έργα τέχνης με κοινό θέμα. Για να υλοποιηθεί κάτι τέτοιο θα πρέπει αντί να περνάμε χειροκίνητα τιμή του isRelatedToMourningPicture, να συγκρίνουμε την τιμή των Theme και αν ταυτίζονται (ή σε περίπτωση που έχουμε περισσότερα από ένα θέματα, αν υπάρχει ένα κοινό, ή όλα κοινά, ή οτιδήποτε άλλο επιλέξουμε) να περνάει yes.

Στο μέλλον ανοίγονται πολλές δυνατότητες για περεταίρω ανάπτυξη των παραπάνω, και σε σημείο που θα ξεφεύγει από τα όρια της υπάρχουσας βιβλιογραφίας. Στο ερευνητικό παράδειγμα που μελετήσαμε παραπάνω ήδη αναφέραμε μερικές γενικεύσεις για διάφορους τρόπους που θα μπορούσαμε να επιλέγεται αυτόματα στη βάση χαρακτηριστικών και ιδιοτήτων η συγκεκριμένη μίξη που θέλουμε. Με περεταίρω εμβάθυνση το παραπάνω παράδειγμα θα μπορούσε να γίνει πράγματι χρήσιμο, αλλά και η δομή του εξειδικευμένη σε άλλα αντικείμενα και όχι έργα τέχνης θα μπορούσε να έχει ποικιλία εφαρμογών. Ο χειρισμός της εννοιολογικής μίξης (σε πολύ πιο πολύπλοκα περιβάλλοντα) θα μπορούσε να δώσει χρήσιμα εργαλεία στην ταχύτατα αναπτυσσόμενη τεχνολογία επεξεργασίας κειμένου φυσικής γλώσσας. Συγκεκριμένα, αν οι εννοιολογικοί χώροι/οντολογίες προσαρμοστούν σε «έννοιες» της φυσικής γλώσσας, η εφαρμογή της αλγεβρικής σημειωτικής πιθανώς να μπορέσει να ανοίξει μεγάλες δυνατότητες στον κλάδο αυτό.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Codescu, M., Mossakowski, T., & Kutz, O. (2014, October). A categorical approach to ontology alignment. In OM (pp. 1-12).
- Eilenberg, S., & MacLane, S. (1945). General theory of natural equivalences. Transactions of the American Mathematical Society, 58(2), 231-294.
- Fauconnier, G., & Turner, M. (2003). Conceptual blending, form and meaning. Recherches en communication, 19, 57-86.
- Goguen, J. (1998, April). An introduction to algebraic semiotics, with application to user interface design. In International Workshop on Computation for Metaphors, Analogy, and Agents (pp. 242-291). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Goguen, J. (2003). Semiotic morphisms, representations, and blending for interface design. In Proceedings, AMAST Workshop on Algebraic Methods in Language Processing (pp. 1-15).
- Goguen, J., Mori, A., & Lin, K. (1997, September). Algebraic semiotics, ProofWebs and distributed cooperative proving. In Proceedings, User Interfaces for Theorem Provers (Vol. 97, pp. 24-34).
- Grothendieck, A. (1962). Techniques de construction et théoremes d’existence en géométrie algébrique. III: Préschémas quotients (Séminaire Bourbaki, t. 13, 1960/61, no. 212). Fondements de la géométrie algébrique, Secrétariat mathématique, Paris.
- Hois, J., Kutz, O., Mossakowski, T., & Bateman, J. (2010). Towards ontological blending. In Artificial Intelligence: Methodology, Systems, and Applications: 14th International Conference, AIMSA 2010, Varna, Bulgaria, September 8-10. 2010. Proceedings 14 (pp. 263-264). Springer Berlin Heidelberg.
- Kutz, O., Mossakowski, T., Hois, J., Bhatt, M., & Bateman, J. (2012). Ontological blending in DOL. Computational Creativity, Concept Invention, and General Intelligence, 1, 33.
- Kutz, O., Neuhaus, F., Mossakowski, T., & Codescu, M. (2012). Blending in the Hub. In Proceedings of the 5th International Conference on Computational Creativity, ICCC-14.

- Mac Lane, S. (2013). Categories for the working mathematician (Vol. 5). Springer Science & Business Media.
- Saussure, F. de (1979). Μαθήματα Γενικής Γλωσσολογίας, Αθήνα: εκδ. Παπαζήση.
- Σκαρπέλος, Φ. (2024). Μία Μεθοδολογική Μελέτη πάνω στην Αλγεβρική Σημειωτική. Στα πρακτικά του 13ου Συνεδρίου της Ελληνικής Σημειωτικής Εταιρίας. Υπό δημοσίευση.
- Torop, P. (2000). Intersemiosis and Intersemiotic Translation. S European Journal for Semiotic Studies, 12(1), 71-100.
- Turner, M., & Fauconnier, G. (2014). Conceptual integration and formal expression. In Metaphor and Philosophy (pp. 183-204). Psychology Press.
- Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. The knowledge engineering review, 11(2), 93-136.
- https://ncatlab.org/nlab/show/HomePage
- https://owlready2.readthedocs.io/en/latest/