

# Ποιοτική Συλλογιστική στον Έλεγχο Προγραμμάτων

Σπυρίδων Ξανθάκης  
Τμήμα Μηχανικών Πληροφορικής Τ.Ε.  
ΤΕΙ Αθήνας  
[spyros.xanthakis@yahoo.fr](mailto:spyros.xanthakis@yahoo.fr)

Ζαφείριος Καραϊσκος  
Τμήμα Μηχανικών Πληροφορικής Τ.Ε.  
ΤΕΙ Θεσσαλίας  
[karaisko@teilar.gr](mailto:karaisko@teilar.gr)

**Περίληψη**—Στο παρόν άρθρο παρουσιάζουμε μια νέα μέθοδο ελέγχου προγραμμάτων. Η καινοτομία σ' αυτήν τη μέθοδο είναι ότι βασίζεται σε μια ποιοτική προσέγγιση του προγράμματος σε συνδυασμό με έναν ευριστικό αλγόριθμο.

**Λέξεις κλειδιά**—έλεγχος προγραμμάτων, οντολογία, ποιοτική συλλογιστική

## I. ΕΙΣΑΓΩΓΗ

### A. Οι δραστηριότητες ανάπτυξης λογισμικού

Για την ανάπτυξη λογισμικού υπάρχουν πολλές μέθοδοι οι οποίες χωρίζονται σε διάφορες δραστηριότητες. Οι μέθοδοι αυτές μπορεί να διαφέρουν μεταξύ τους, έχουν όμως κοινές τις εξής δραστηριότητες:

- **Παραγγελία πελάτη.** Όλα ξεκινούν με την παραγγελία που δίνει κάποιος πελάτης της εταιρείας ανάπτυξης λογισμικού.
- **Προδιαγραφές του λογισμικού.** Περιγράφεται με λεπτομέρεια η λειτουργικότητα του λογισμικού και των περιορισμών επί των λειτουργιών.
- **Σχεδίαση του λογισμικού και υλοποίηση.** Παραγωγή του λογισμικού που να πληροί τις προδιαγραφές του και εγκατάστασή του.
- **Επικύρωση και επαλήθευση του λογισμικού.** Η επικύρωση του λογισμικού εγγυάται ότι υλοποιεί αυτό που επιθυμεί ο πελάτης. Η επαλήθευσή του εγγυάται ότι πληροί τις προδιαγραφές.
- **Εξέλιξη του λογισμικού.** Το λογισμικό εξελίσσεται έτσι ώστε να ικανοποιεί τις μεταβαλλόμενες ανάγκες του πελάτη.

### B. Έλεγχος του λογισμικού

Επικύρωση και επαλήθευση του λογισμικού υλοποιείται με διάφορες τεχνικές ελέγχου. Ο έλεγχος του λογισμικού είναι ίσως η σημαντικότερη από τις δραστηριότητες ανάπτυξης λογισμικού. Μια μελέτη του NIST (National Institute of Standards and Technology) του 2002 αναφέρει ότι τα bugs λογισμικού επιβαρύνουν με δισεκατομμύρια δολάρια την οικονομία των Η.Π.Α. κάθε χρόνο. Αυτό το κόστος θα ήταν πολύ μικρότερο αν υπήρχαν καλύτεροι έλεγχοι λογισμικού. Δηλαδή οι τεχνικές ελέγχου που έχουν

επινοηθεί μέχρι σήμερα δεν λύνουν ικανοποιητικά το πρόβλημα με συνέπεια η έρευνα να συνεχίζεται εντατικά.

Οι τεχνικές ελέγχου μπορούν να κατηγοριοποιηθούν βάση δύο κριτηρίων:

- Εκτέλεση του κώδικα
- Εξέταση του κώδικα

Η εκτέλεση του κώδικα κατηγοριοποιεί τις τεχνικές ελέγχου σε στατικές και δυναμικές. Ο **στατικός έλεγχος** δεν απαιτεί την εκτέλεση του προγράμματος. Ο **δυναμικός έλεγχος** ενέχει την αλληλεπίδραση με το πρόγραμμα κατά την εκτέλεσή του. Οι δύο μέθοδοι συχνά χρησιμοποιούνται μαζί ώστε να εξασφαλίζεται η λειτουργικότητα ενός προγράμματος.

Η εξέταση του κώδικα οδηγεί σε δύο μεθόδους ελέγχου: μαύρο κουτί και λευκό κουτί. Η μέθοδος **μαύρο κουτί** εξετάζει κάποια θεμελιώδη πλευρά του λογισμικού με ελάχιστη προσοχή στην εσωτερική λογική δομή του λογισμικού. Η μέθοδος **λευκό κουτί** αφιερώνεται στη λεπτομερειακή εξέταση των διαδικασιών του λογισμικού.

Στο παρόν άρθρο περιγράφεται το θεωρητικό υπόβαθρο ενός αυτόματου εργαλείου για δυναμικό έλεγχο προγραμμάτων. Η θεωρία βασίζεται σε μια ποιοτική προσέγγιση του προγράμματος καθώς και της συμπεριφοράς του, δηλαδή στην κατασκευή μιας οντολογίας. Η οντολογία αυτή αποτελείται από ένα στατικό και ένα δυναμικό μέρος τα οποία θα περιγράψουμε. Θα περιγράψουμε καταρχήν τι εννοούμε με τον όρο οντολογία και γιατί είναι ενδιαφέρον να έχουμε μια ποιοτική προσέγγιση. Για να γίνει αυτό θα εισάγουμε τις βασικές έννοιες που θα χρησιμοποιηθούν: μάγμα, έννοια της επιρροής, γράφος φάσεων, κ.α. Θα αναλύσουμε επίσης τις βασικές υποθέσεις που κάνουμε για να φέρουμε σε πέρας την κατασκευή και τη χρήση (δηλαδή την εφαρμογή στον έλεγχο προγραμμάτων) ενός ποιοτικού μοντέλου.

## II. ΠΟΙΟΤΙΚΟΣ ΈΛΕΓΧΟΣ ΛΟΓΙΣΜΙΚΟΥ

### A. Οντολογία

Σύμφωνα με την κοινή αποδοχή, ονομάζουμε **οντολογία** ένα σύνολο αλληλένδετων εννοιών που περιγράφουν ένα συγκεκριμένο (συνήθως επιστημονικό) τομέα και τους κανόνες που τον διέπουν. Μια οντολογία επιτρέπει έτσι μια κοινή γλώσσα και γνώση που μπορούν να μετατρέπονται και να εμπλουτίζονται από τους ειδικούς.

Οντολογίες για την ανάπτυξη λογισμικού έχουν υπάρξει και χρησιμοποιούνται πολλές, χωρίς όμως αναγκαστικά να έχουν διατυπωθεί αρχικά με το όνομα "οντολογία". Οντολογίες έχουν αναπτυχθεί και σε άλλους επιστημονικούς τομείς, όπως η ιατρική και η βιολογία, όπως επίσης έχουν αναπτυχθεί στην Ποιοτική Συλλογιστική (ΠΣ) που ανήκει στο γενικότερο ερευνητικό κλάδο της Τεχνητής Νοημοσύνης.

Η βασική ιδέα της προσέγγισής μας μπορεί να συνοψιστεί ως εξής: εφόσον η ανάπτυξη ενός αλγόριθμου είναι μια κρίσιμη, δύσκολη και επίπονη διεργασία γιατί να μην επιχειρήσουμε να χρησιμοποιήσουμε και σε αυτόν τον τομέα τις ιδέες της ποιοτικής συλλογιστικής; Το αρχικό μήνυμα της ποιοτικής προσέγγισης παραμένει ακέραιο: να ξεφύγουμε από μια καθαρά μηχανιστική-λογικίστικη προσέγγιση ενός προβλήματος, υιοθετώντας μια ορολογία και κανόνες συλλογισμού που, χωρίς να έχουν τη μαθηματική ακρίβεια ενός μαθηματικού μοντέλου, είναι πιο "απλές" (χρησιμοποιήθηκε πολλές φορές ο επιτυχημένος όρος "αφελείς"), πιο κατανοήσιμες από τον άνθρωπο και πιο κοντά στον "μακροσκοπικό" τρόπο σκέψης του. Θα πρέπει γι' αυτό το σκοπό να προτείνουμε μια νέα ποιοτική οντολογία που να αφορά τη διαδικασία του προγραμματισμού.

Προτού όμως παρουσιάσουμε αυτή την οντολογία, ας δούμε ποιές είναι οι δυσκολίες που συναντάμε όταν επιχειρούμε να εφαρμόσουμε κλασικές μεθόδους ποιοτικού συλλογισμού σ' έναν κλάδο όπως είναι η τεχνολογία λογισμικού και συγκεκριμένα ο προγραμματισμός.

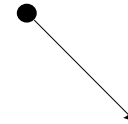
Η πρώτη δυσκολία συνίσταται στη διαχείριση των δομών από το λογισμικό. Για ένα κλασικό φυσικό φαινόμενο όπως είναι η διαδρομή ενός σώματος μέσα σε ένα πεδίο το είδος των δεδομένων που διαχειριζόμαστε είναι κλασικές πραγματικές ή ακέραιες τιμές οι οποίες εξελίσσονται στο χρόνο με έναν συνήθως συνεχή (με τη μαθηματική έννοια) τρόπο. Όταν όμως πρέπει να αναλύσουμε την αλγοριθμική διαδικασία που μετατρέπει βήμα-βήμα αρχικές εισόδους σε εξόδους παρατηρούμε τη μεγάλη ετερομορφία των δεδομένων: πραγματικοί αριθμοί, πίνακες πινάκων, πολυμορφικά αντικείμενα του αντικειμενοστραφούς προγραμματισμού που περιλαμβάνουν άλλα αντικείμενα με συμβολοσειρές και λογικές τιμές ή αναφορές που μας οδηγούν σε άλλα αντικείμενα, κ.ο.κ. Και σαν να μην φτάνουν όλα αυτά, ακόμα κι ένας πραγματικός αριθμός σε μια γλώσσα προγραμματισμού δεν είναι "συνεχής" ούτε "άπειρος" αλλά είναι ένας δεκαδικός του οποίου η ακρίβεια και το εύρος εξαρτώνται από τα όρια του λογισμικού και υλικού περιβάλλοντος στο οποίο προγραμματίζουμε.

Στην οντολογία μας θα ξεχωρίσουμε τους βασικούς τύπους, τους οποίους θα ονομάσουμε **άτομα** (ακέραιοι αριθμοί, πραγματικοί αριθμοί, λογικές τιμές και χαρακτήρες) από τους σύνθετους, τους οποίους θα ονομάσουμε **μάγματα** (πολυδιάστατοι πίνακες, αντικείμενα του αντικειμενοστραφούς προγραμματισμού, οι δομές της C και της Pascal, οι λίστες της LISP, κ.α.). Ένα μάγμα δεν έχει δομή, παραμένει μάγμα όταν προσθέτουμε ή αφαιρούμε αντικείμενα. Θα θεωρήσουμε ότι ένα μάγμα έχει ένα μέγεθος το οποίο αυξάνεται ή μειώνεται αλλά χωρίς συγκεκριμένες τιμές και χωρίς κατώτατα και ανώτατα όρια. Απεικονίζεται με ένα τετράγωνο στο πάνω μέρος του οποίου τοποθετούμε το

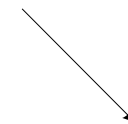
όνομά του. Ένα μάγμα μετατρέπεται αφομοιώνοντας καινούργια δεδομένα και εμπλουτιζόμενο εξελίσσεται με υπολογιστικές διαδικασίες, μεταδίδει μέσω αντιγραφής μερικά από τα δεδομένα σε άλλα μάγματα ή εξαρτάται από τα δεδομένα άλλων μαγμάτων για να εξελιχθεί, κ.ο.κ.

Για να αλλάξουν οι τιμές στο εσωτερικό ενός μάγματος χρειαζόμαστε την επιρροή μιας εξωτερικής οντότητας (μάγματος ή ατόμου). Υπάρχουν μόνο τριών ειδών επιρροές:

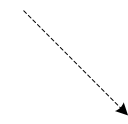
1. Η *αντιγραφή* : Ορισμένες πληροφορίες από το εσωτερικό ενός μάγματος αντιγράφονται στο εσωτερικό ενός άλλου. Η επιρροή αντιγραφής αντιστοιχεί σε απλή εντολή εκχώρησης τιμής " $a = b;$ ". Απεικονίζεται με ένα βέλος του οποίου η αρχή είναι κουκκίδα.



2. Ο *υπολογισμός* : Μια νέα τιμή, που απορρέει από μια βασική αριθμητική πράξη, δημιουργείται για να καταχωρηθεί σε ένα μέρος της μνήμης. Η επιρροή υπολογισμού αντιστοιχεί σε εντολές της μορφής " $a = b + 5 * c;$ ", για strings "`allname = first + last;`" ή ακόμα σε προκαθορισμένες σχέσεις "`limit = studentset.size();`". Απεικονίζεται με ένα απλό βέλος.



3. Η *δραστηριοποίηση* : Αποτελεί την εξ αποστάσεως επιρροή. Χρειάζεται να γνωρίζουμε την τιμή μιας οντότητας προκειμένου να δραστηριοποιήσουμε μια επιρροή (αντιγραφή ή υπολογισμό) σε μια άλλη οντότητα. Η επιρροή δραστηριοποίησης αντιστοιχεί σε ένα συνδυασμό, εμφωλευμένων ή μη, συνθηκών για παράδειγμα "`if (a < b) ...`". Η επιρροή δραστηριοποίησης δεν μπορεί να εμφανίζεται μόνη της. Πάντα αναφέρεται σε μια από τις άλλες δύο επιρροές. Απεικονίζεται με ένα διακεκομμένο βέλος.

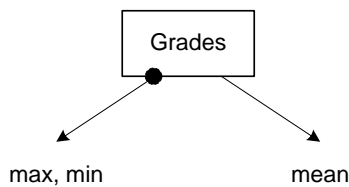


Ο σκοπός μιας ποιοτικής οντολογίας είναι να προτείνει μια ποιοτική προσέγγιση ενός αλγόριθμου. Σημαίνει ότι πρέπει να είμαστε σε θέση να εκφράσουμε την "ουσία" ενός αλγόριθμου χρησιμοποιώντας τα εργαλεία και το formalισμό της οντολογίας.

#### Παράδειγμα 1

Ας υποθέσουμε ότι έχουμε έναν απλό αλγόριθμο που εντοπίζει το μέγιστο, το ελάχιστο και το μέσο όρο των βαθμολογιών των μαθητών μιας τάξης.

Υποθέτουμε ότι οι βαθμοί βρίσκονται μέσα σε ένα μάγμα Grades. Τα αποτελέσματα είναι τρεις πραγματικοί αριθμοί (άτομα): max, min, mean. Ποιές είναι οι επιρροές που έχει το αρχικό μάγμα στα αποτελέσματα. Για τα δύο πρώτα (max και min) παρατηρούμε ότι η πράξη είναι η αντιγραφή των δεδομένων του μάγματος. Για να γίνει όμως αυτή η αντιγραφή χρειάζεται να συμβουλευτούμε την τιμή των βαθμών. Με άλλα λόγια, ανάλογα με την τιμή των βαθμών θα δραστηριοποιήσουμε ή όχι την επιρροή αντιγραφής. Αυτό σημαίνει ότι εκτός από την επιρροή αντιγραφής, το μάγμα Grades έχει επίσης μια επιρροή δραστηριοποίησης στις δύο πρώτες εξόδους. Για το τρίτο που είναι ο μέσος όρος χρειάζεται να δραστηριοποιηθεί ο υπολογισμός. Σημειώστε ότι ο υπολογισμός αυτός δραστηριοποιείται όποιες και να είναι οι βαθμολογίες, άρα δεν υπάρχει επιρροή δραστηριοποίησης γι' αυτό το αποτέλεσμα.

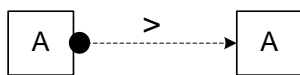


### Παράδειγμα 2

Έστω ένας αλγόριθμος ο οποίος αφαιρεί ορισμένες λέξεις από ένα αρχείο και τις κωδικοποιεί χρησιμοποιώντας ένα κώδικα που περιγράφεται σε ένα σύνολο κλειδιών.

Το αρχείο θα είναι το μάγμα A και το κωδικοποιημένο αποτέλεσμα το μάγμα B. Επίσης θα ονομάσουμε K το μάγμα με τους κώδικες. Επίσης, στην οντολογία πρέπει να εκφράσουμε την τελευταία έννοια που απομένει, εκείνη της εξέλιξης μιας οντότητας (μάγματος ή ατόμου). Η εξέλιξη είναι πάντα αποτέλεσμα διαφόρων επιρροών. Αυτό απεικονίζεται με ένα απλό βέλος το οποίο θα ενώνει την κατάσταση πριν με την κατάσταση μετά του εν λόγου αντικειμένου.

Μπορούμε να απλοποιήσουμε τους συμβολισμούς των επιρροών μεταξύ δύο οντοτήτων συγχωνεύοντάς τους. Έτσι ένα διακεκομμένο βέλος με κουκκίδα από το μάγμα A στο μάγμα B απεικονίζει την ύπαρξη δύο επιρροών. Το ίδιο κάναμε και στο προηγούμενο παράδειγμα.



Σημειώστε ότι εμφανίζονται δύο μάγματα A στο παραπάνω σχεδιάγραμμα, ένα πριν την αφαίρεση των λέξεων και ένα μετά. Το σύμβολο ">" πάνω από το βέλος σημαίνει ότι μειώθηκε ο όγκος του μάγματος.

### B. Σχέσεις διάταξης για τις δομές δεδομένων ενός αλγόριθμου

Για να είμαστε σε θέση να αναλύσουμε το αποτέλεσμα ενός αλγόριθμου ποιοτικά, το ελάχιστο που πρέπει να είμαστε σε θέση να πράξουμε είναι να συγκρίνουμε ένα αποτέλεσμα με ένα άλλο αποτέλεσμα.

Ορίζουμε καταρχήν μια σχέση διάταξης μεταξύ των ατόμων της οντολογίας. Υπενθυμίζουμε ότι μια σχέση διάταξης " $\leq$ " επαληθεύει για κάθε στοιχείο  $a$  :

$$a \leq a \text{ (ανακλαστική)}$$

$$a \leq \beta \text{ και } \beta \leq a \text{ συνεπάγεται } a = \beta \text{ (αντισυμμετρική)}$$

$$a \leq \beta \text{ και } \beta \leq \gamma \text{ συνεπάγεται } a \leq \gamma \text{ (μεταβατική)}$$

Όταν δύο στοιχεία μπορούν να συγκριθούν λέμε ότι είναι **συγκρίσιμα**. Όταν όλα τα στοιχεία είναι συγκρίσιμα, η διάταξη είναι **ολική**, στην αντίθεση περίπτωση η διάταξη είναι **μερική**. Μιλάμε τότε για μερικώς διατεταγμένο σύνολο (partially ordered set ή αλλιώς **poset**).

Είδαμε προηγουμένως ότι με άτομα εννοούμε τους απλούς τύπους:

1. *Ακέραιοι ή πραγματικοί αριθμοί*. Μπορεί να είναι οποιοδήποτε εύρους και ακρίβειας. Θα χρησιμοποιηθεί εδώ η κλασική διάταξη των πραγματικών αριθμών η οποία είναι ολική.

2. *Συμβολοσειρές και χαρακτήρες*. Αν και οι συμβολοσειρές (τις οποία θα ονομάζουμε και "λέξεις") θεωρούντο μη ατομικές δομές, στις πρόσφατες γλώσσες προγραμματισμού χρησιμοποιούνται ως ανεξάρτητες και αυτόνομες δομές (εκτός φυσικά και αν ασχολούμαστε με αλγόριθμους που διαχειρίζονται το εσωτερικό τους, οπότε πρέπει να τις θεωρήσουμε ως ένα μάγμα χαρακτήρων). Στην οντολογία μας θα θεωρήσουμε ότι οι χαρακτήρες λειτουργούν ως συμβολοσειρές εκτός και αν χρησιμοποιούνται ως κώδικες. Στην τελευταία περίπτωση, χρησιμοποιούμε τον ASCII κώδικα για να τους ταξινομήσουμε. Στη βιβλιογραφία έχουν προταθεί πολλές σχέσεις διάταξης για τις συμβολοσειρές. Η πιο διαδεδομένη διάταξη είναι η λεξικογραφική (αλφαβητική) που έχει την ιδιότητα να είναι ολική. Υπάρχουν όμως κι άλλες διατάξεις που αποδίδουν στο σύνολο τους τη δομή ενός μερικώς διατεταγμένου συνόλου. Για παράδειγμα, μια μερική διάταξη μπορεί να οριστεί ως εξής:

$s_1 \leq s_2$  όταν η ακολουθία γραμμάτων της  $s_1$  εμφανίζεται μέσα στη  $s_2$

Έτσι, "άγρα"  $\leq$  "διάγραμμα"  $\leq$  "χρονοδιάγραμμα", αλλά η λέξης "χρόνος" είναι μη συγκρίσιμη με τη λέξη "χρονοδιάγραμμα".

3. *Λογικές τιμές*. Εδώ ορίζουμε συμβατικά τη διάταξη  $false \leq true$ .

Θα επεκτείνουμε τώρα τις διατάξεις των ατόμων σε πιο περίπλοκες δομές που είναι τα μάγματα. Στην οντολογία μας δεν ενδιαφέρει να συγκρίνουμε δύο οποιαδήποτε μάγματα, αν και υπάρχουν πάντα τρόποι να ορίσουμε ολικές διατάξεις για έναν οποιοδήποτε τύπου μάγματος, αλλά κάτι τέτοιο δεν θα ήταν πολύ ενδιαφέρον. Αυτό που μας ενδιαφέρει είναι να συγκρίνουμε το ίδιο μάγμα σε δύο διαφορετικές καταστάσεις. Πιο συγκεκριμένα μας ενδιαφέρει να συγκρίνουμε τα στοιχεία του ίδιου μάγματος M πριν και μετά από μια αλλαγή. Αυτή η αλλαγή θα είναι "μικρή" όπως θα δούμε.

Για να επεκτείνουμε μια διάταξη στα μάγματα θα υποθέσουμε ότι τα μάγματα παρουσιάζονται σε μια δενδροειδή μορφή της οποίας τα τερματικά (φύλλα), αν υποθέσουμε ότι έχουμε

επαναφέρει όλους τους δείκτες, είναι άτομα. Με άλλα λόγια υποθέτουμε ότι ΔΕΝ έχουμε κυκλικές μορφές δεδομένων, όπως αλυσιδωτές κυκλικές λίστες. Μπορούμε εύκολα να ορίσουμε μια διάταξη και σε αυτές τις περιπτώσεις. Ξεχωρίζουμε δύο περιπτώσεις:

Στην πρώτη περίπτωση το μάγμα είναι ομοιόμορφο, δηλαδή, περιέχει σ' ένα πρώτο επίπεδο τον ίδιο τύπο δεδομένων (π.χ. πίνακας). Εδώ ορίζουμε μια διάταξη ως εξής. Αν τα μάγματα A και B έχουν το ίδιο πλήθος στοιχείων, έστω α, και β, τότε γράφουμε  $A \leq B$  όταν για κάθε δείκτη i έχουμε  $\alpha_i \leq \beta_i$ . Εννοείται ότι αυτός ο ορισμός μπορεί να γίνει αναδρομικός στην περίπτωση που τα στοιχεία δεν είναι ατομικά μέχρις ότου φτάσουμε στα φύλλα. Αν όμως τα μάγματα δεν έχουν το ίδιο πλήθος και τα στοιχεία του A περιέχονται στο B, τότε θα πούμε  $A \leq B$ . Αν κανένα μάγμα δεν περιέχεται σε άλλο, τότε τα A και B θα θεωρούνται μη συγκρίσιμα. Η περιεκτικότητα μπορεί να εννοηθεί ως ακολουθία.

Στη δεύτερη περίπτωση το μάγμα δεν είναι ομοιόμορφο (κάτι που αντιστοιχεί σε ένα αντικείμενο του αντικειμενοστραφούς προγραμματισμού). Για να είναι τα μάγματα συγκρίσιμα θα πρέπει να έχουν την ίδια δομή οπότε η σύγκριση θα εξαρτηθεί από την (ενδεχομένως αναδρομική) σύγκριση των στοιχείων τους. Αν δεν έχουν την ίδια δομή τότε θα θεωρούνται μη συγκρίσιμα. Ας μην ξεχνάμε ότι μας ενδιαφέρει να συγκρίνουμε το ίδιο μάγμα πριν και μετά την εκτέλεση μιας πράξης.

#### Παράδειγμα.

Έστω ότι ένα μάγμα φοιτητής περιλαμβάνει τα πεδία: όνομα, ηλικία, φύλο και μια λίστα βαθμών που είναι ακέραιοι.

Ας υποθέσουμε ότι ένα αντικείμενο φοιτητής φ1 μετατράπηκε σε φ2 ύστερα από μερικές επιρροές. Οι αλλαγές που μπορεί να έγιναν για να προκύψει το φ2 είναι η μείωση κάποιων βαθμών και η αύξηση της ηλικίας του. Σ' αυτήν την περίπτωση η σύγκριση του φ1 με το φ2 ξεκινά με τις ηλικίες και συνεχίζει με τις λίστες βαθμών. Έστω ότι η παλιά λίστα του φ1 είναι η ακόλουθη:

[8, 10, 10, 1, 2]

και ας δούμε σε τι θα μπορούσε να εξελιχθεί αυτή η βαθμολογία στο φ2:

- [9,10,10,2,3] αύξηση
- [0, 0,0,1,2] μείωση
- [5,5,10,2,3,8] μη συγκρίσιμα καθότι κανένα δεν περιέχει το άλλο
- [6,12,10,1,2] μη συγκρίσιμα καθότι ο πρώτος βαθμός είναι μικρότερος ενώ ο δεύτερος είναι μεγαλύτερος.

Με αυτό το παράδειγμα βλέπουμε ότι ακόμα και αν πρόκειται για το ίδιο αντικείμενο σε διαφορετικές καταστάσεις, αλλά με την ίδια δομή, τίποτα δεν μας εξασφαλίζει, εκ των προτέρων, ότι τα δύο μάγματα θα είναι συγκρίσιμα.

#### Γ. Ο γράφος φάσεων ενός αλγόριθμου

Μετά τον ορισμό των σχέσεων διάταξης που διέπουν τα δεδομένα ενός αλγόριθμου, θα ορίσουμε μια δεύτερη σημαντική έννοια της ποιοτικής οντολογίας: το γράφο

φάσεων. Ο γράφος φάσεων είναι το αντίστοιχο του χώρου φάσεων ενός φυσικού συστήματος και μπορεί να ειπωθεί σαν τη “γραφική” αναπαράσταση της εκτέλεσης ενός αλγόριθμου με τον ίδιο τρόπο που μια καμπύλη απεικονίζει τη συμπεριφορά μιας μαθηματικής σχέσης.

Σε γενικές γραμμές ένας γράφος Γ είναι ένα ζεύγος (X, Y) όπου X είναι ένα οποιοδήποτε σύνολο και Y ένα σύνολο που αποτελείται από ζεύγη στοιχείων του X. Τον δηλώνουμε ως Γ(X,Y). Κάθε στοιχείο του X μπορεί να ειπωθεί ως ένας κόμβος. Κάθε ζεύγος του Y, ας πούμε  $(\chi_1, \chi_2)$  μπορούμε να το παραστήσουμε ως  $\chi_1 \rightarrow \chi_2$ . Τίποτα δεν αναγκάζει το  $\chi_1$  να είναι διαφορετικό από το  $\chi_2$ . Στην περίπτωση που είναι όμοια, το βέλος επιστρέφει στον εαυτό του και λέμε ότι έχουμε ένα βρόχο.

Έστω ένας αλγόριθμος Λ, τον οποίο θέλουμε να αναλύσουμε με εισόδους και εξόδους. Έστω και το σύνολο E που αντιστοιχεί σε όλες τις δυνατές εισόδους που προδιαγράφονται για τον Λ. Είδαμε ότι ο αλγόριθμος είναι προσδιοριστικός, άρα για μια συγκεκριμένη επιλογή εισόδων ε η εκτέλεση του αλγόριθμου (εντός ενός συγκεκριμένου χρονικού διαστήματος γιατί στην αντίθετη περίπτωση θεωρείται εσφαλμένος) θα παράγει πάντα (όποια και να είναι η σειρά εκτέλεσης καθότι δεν διαθέτουμε μνήμη) το ίδιο αποτέλεσμα ω. Για κάθε στοιχείο ε του συνόλου E θα σχηματίσουμε το ζεύγος  $(\varepsilon, \Lambda(\varepsilon))$  όπου  $\Lambda(\varepsilon)$  είναι το αποτέλεσμα που δίνει ο αλγόριθμος Λ όταν του υποβάλουμε τις εισόδους που αντιστοιχούν στη συγκεκριμένη επιλογή των τιμών της ε. Αν, για παράδειγμα, το λογισμικό έχει σαν σκοπό να υπολογίζει το μέσο όρο της βαθμολογίας ενός φοιτητή και να μας πληροφορεί αν περνάει το έτος ή όχι, τότε ένα πιθανό ε θα μπορούσε να ήταν:

“παπαδόπουλος” “άνδρας” 22 [5,5,6,4,10]

όπου 22 είναι η ηλικία του φοιτητή και ακολουθεί η λίστα των βαθμών του. Άρα το ζεύγος που θα σχηματιστεί είναι:

$(\varepsilon, \omega) = (\varepsilon, \Lambda(\varepsilon)) = (<\text{“παπαδόπουλος”}, \text{“άνδρας”}, 22, [5,5,6,4,10]>, <6, \text{true}>)$

όπου χρησιμοποιήσαμε τα σύμβολα “<” και “>” για να οριοθετήσουμε τις δύο λίστες εισόδου και εξόδου.

Θα χρησιμοποιήσουμε τώρα το ζεύγος  $(\varepsilon, \Lambda(\varepsilon))$  για να κατασκευάσουμε τους κόμβους του γράφου φάσεων. Με άλλα λόγια ο γράφος φάσεων Φ(X, Y) έχει ένα X, δηλαδή ένα σύνολο κόμβων που ισούται με όλα τα δυνατά ζεύγη  $(\varepsilon, \Lambda(\varepsilon))$  που μπορούν να σχηματιστούν σύμφωνα με τις προδιαγραφές του Λ. Είναι ευνόητο ότι ο Φ έχει τόσους κόμβους όσα στοιχεία έχει το E, δηλαδή οι κόμβοι του Φ είναι τόσοι όσες και οι δυνατές τιμές από τις προδιαγραφόμενες εισόδους του Λ. Μας μένει τώρα να δούμε πώς θα ενώσουμε τους διάφορους κόμβους του Φ, με άλλα λόγια πώς θα σχηματίσουμε το σύνολο Y των βελών του Φ.

Θα ζωγραφίζουμε ένα βέλος από τον κόμβο  $(\varepsilon, \Lambda(\varepsilon))$  στον κόμβο  $(\varepsilon', \Lambda(\varepsilon'))$  αν οι εισοδοί ε και ε' έχουν την ελάχιστη διαφορά που μπορούν να έχουν δύο διαφορετικές εισοδοί μεταξύ τους. Στο βέλος αυτό θα προσθέσουμε μια ετικέτα που αντιστοιχεί στην συγκεκριμένη παράμετρο στην οποία διαφέρουν οι δύο εισοδοί. Αν, για παράδειγμα, δύο εισοδοί είναι ακριβώς πανομοιότυπες αλλά διαφέρουν μόνο στην

δεύτερη παράμετρο τους που είναι το φύλο, τότε θα σχηματίσουμε μια ακμή ( $\epsilon$ ,  $\epsilon'$ ) της οποίας ο προσανατολισμός εξαρτάται από τη σχέση διάταξης που διέπει αυτή τη συγκεκριμένη παράμετρο. Αν δηλαδή έχουμε “άνδρας”  $\leq$  “γυναίκα” τότε το βέλος θα πηγαίνει από την είσοδο που έχει το φύλο “άνδρας” στην είσοδο που έχει το φύλο “γυναίκα”. Ξέρουμε ότι μεταξύ αυτών των εισόδων δεν μπορεί να υπάρξει άλλο βέλος με άλλη ετικέτα καθότι εξ υποθέσεως το φύλο είναι η μόνη διαφορά τους. Για ένα βέλος του τύπου ( $\chi_1$ ,  $\chi_2$ ) με την ετικέτα ηλικία, θα πούμε επίσης ότι ο  $\chi_1$  ενώνεται με τον  $\chi_2$  με κατεύθυνση το πεδίο ηλικία.

Ας πάρουμε ένα άλλο παράδειγμα. Υποθέτουμε ότι έχουμε δύο εισόδους:

$\epsilon = \langle \text{“παπαδόπουλος”, “άνδρας”, 22, [5,5,6,4,10]} \rangle$  και

$\epsilon' = \langle \text{“παπαδόπουλος”, “άνδρας”, 22, [5,5,6,5,10]} \rangle$

Παρατηρούμε ότι οι δύο αυτές εισοδοί διαφέρουν μόνο ως προς τη βαθμολογία που πήραν οι ομώνυμοι φοιτητές στο τέταρτο μάθημα της λίστας (ο πρώτος έχει πάρει 4 και ο δεύτερος 5). Βλέπουμε δηλαδή ότι οι δύο αυτές εισοδοί διαφέρουν μόνο σε μια παράμετρο και πληρούν την ελάχιστη δυνατή διαφορά που είναι 1. Εδώ θα ζωγραφίσουμε ένα βέλος σύμφωνα με τη φυσική σχέση διάταξης που διέπει τους ακέραιους. Το ίδιο φαινόμενο θα είχαμε αν συναντούσαμε μια είσοδο η οποία θα είχε ακριβώς το ίδιο φύλο και ακριβώς τους ίδιους βαθμούς με το  $\epsilon$  αλλά το όνομα θα ήταν “παπαδόπουλος”, δηλαδή περίπου το ίδιο με το προηγούμενο. Προσοχή, η γεινίαση στο γράφο φάσεων ορίζεται σύμφωνα με την ελάχιστη απόσταση στο χώρο των εισόδων και όχι στο χώρο των εξόδων.

Θα συμβολίσουμε τους γείτονες ενός κόμβου  $K = (\epsilon, \Lambda(\epsilon))$  με το συμβολισμό  $K+\delta$  όπου  $\delta$  εκφράζει μια απειροελάχιστη μεταβολή.  $K+\delta$  είναι λοιπόν ένα σύμβολο κόμβων με κέντρο τον  $K$  οι οποίοι είναι γείτονες του  $K$ . Με τοπολογικούς όρους το  $K+\delta$  είναι η γειτονιά του  $K$  στο γράφο φάσεων.

### III. ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ ΠΟΙΟΤΙΚΟΥ ΜΟΝΤΕΛΟΥ

Η πρώτη βασική λειτουργία του εργαλείου είναι να επιτρέπει στο χρήστη να σχεδιάζει ένα ποιοτικό μοντέλο έτσι ώστε να είναι σε θέση να ελέγξει τη συμπεριφορά του υπό έλεγχο λογισμικού. Υπενθυμίζουμε ότι ένα ποιοτικό μοντέλο αποτελείται από ένα στατικό σκέλος κι ένα δυναμικό σκέλος. Το στατικό σκέλος αντιπροσωπεύει τη ροή των δεδομένων με τη βοήθεια μαγμάτων και πρωταρχικών ροών. Ένα δυναμικό μοντέλο είναι ένας γράφος. Οι μεν κόμβοι αντιπροσωπεύουν ένα (συνήθως συνεκτικό) σύνολο κόμβων του γράφου φάσεων, τα δε βέλη έχουν ετικέτες που εκφράζουν τις ποιοτικές μεταβολές (+, -, κ.λ.π.) των εξόδων όταν η παράμετρος εισόδου που εμφανίζεται στην ετικέτα μεταβάλλεται απειροστά. Θα χρησιμοποιήσουμε ένα παράδειγμα για να δούμε πως ερμηνεύεται ένα ποιοτικό μοντέλο.

Έστω ότι επιθυμούμε να ελέγξουμε έναν αλγόριθμο που υπολογίζει το μέσο όρο  $M$  των στοιχείων ενός πίνακα  $A$  (με 5 στοιχεία) που είναι μεγαλύτερα ή ίσα από ένα όριο  $T$  που δίνεται και αυτό ως είσοδος.

Υποθέσουμε ότι τα στοιχεία του πίνακα είναι θετικοί ακέραιοι όπως επίσης και το όριο. Αν δεν υπάρχει κανένα στοιχείο

μεγαλύτερο του ορίου τότε ο αλγόριθμος θεωρεί συμβατικά ότι ο “μέσος όρος” ισούται με  $-1.0$ . Για να είμαστε πιο συγκεκριμένοι ο πηγαίος κώδικας ενός τέτοιου αλγόριθμου θα μπορούσε να είναι ο εξής:

```
sum = 0;
nrItems = 0;

for(i = 0; i < 5; i++)
    if(A[i] >= T)
    {
        sum = sum + A[i];
        nrItems = nrItems + 1;
    }

if(nrItems > 0)
    M = 0.1*sum / nrItems;
else
    M = -1.0;
```

Το μοντέλο αυτού του κώδικα στηρίζεται στην ανάλυση της ποιοτικής συμπεριφοράς των εξόδων σε σχέση με ελάχιστες μεταβολές στις εισόδους. Εφόσον οι δύο εισοδοί είναι ανεξάρτητες πρέπει να εξετάσουμε την ποιοτική συμπεριφορά της καθεμιάς, αλλά στο παρόν άρθρο θα εξετάσουμε το όριο  $T$ . Ο πίνακας  $A$  έχει τα στοιχεία [65, 30, 49, 70, 60] και  $T$  έχει την τιμή 50.

**Ki:**

$A = [65, 30, 49, 70, 60]$

$T = 50$

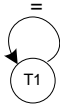
Αυτές οι συγκεκριμένες επιλογές εισόδων αποτελούν έναν κόμβο  $K_i$  του γράφου φάσεων. Επιλέγουμε τυχαία να αυξήσουμε απειροελάχιστα το  $T$  από 50 στην τιμή 51. Περνάμε τότε στον κόμβο  $K_{i+1}$ , γείτονα του  $K_i$ :

**Ki+1:**

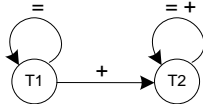
$A = [65, 30, 49, 70, 60]$

$T = 51$

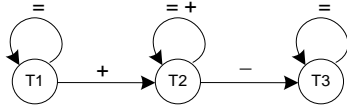
Το αποτέλεσμα της εκτέλεσης για τον κόμβο  $K_i$  είναι  $M = (65 + 70 + 60) / 3 = 65.0$ , όπως και το αποτέλεσμα της εκτέλεσης για τον  $K_{i+1}$  είναι  $M' = (65 + 70 + 60) / 3 = 65.0$ . Αυξάνουμε συνέχεια, κατά ένα, την τιμή του  $T$  περνώντας κάθε φορά στον κόμβο γείτονα μέχρις ότου το  $T$  πάρει την τιμή 61. Για τους κόμβους  $K_i, K_{i+1}, \dots, K_{i+10}$  ο μέσος όρος παραμένει ίδιος (δηλαδή έχουμε βρόχους), συνεπώς αυτοί οι κόμβοι της ακολουθίας παρουσιάζουν την ίδια ποιοτική συμπεριφορά όσον αφορά την είσοδο  $T$ . Στην περίπτωση αυτή μπορούμε να θεωρήσουμε ότι αποτελούν τον ίδιο κόμβο του ποιοτικού μοντέλου τον οποίο θα ονομάσουμε  $T1$ .



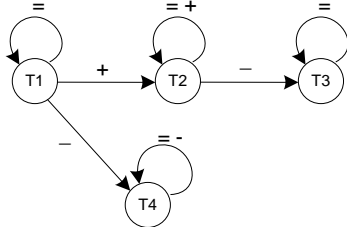
Όταν το T πάρει την τιμή 61 τότε το αποτέλεσμα για τον κόμβο  $K_{i+11}$  είναι  $M = (65 + 70) / 2 = 67.5$ . Αυτό σημαίνει ότι στο γράφο φάσεων ο κόμβος  $K_{i+10}$  είναι ενωμένος με τον κόμβο  $K_{i+11}$  μέσω ενός βέλους στο οποίο προσθέσαμε το πρόσημο "+". Το αποτέλεσμα παραμένει το ίδιο μέχρις ότου το T πάρει την τιμή 65, αυξάνει όταν πάρει την τιμή 66 και παραμένει σταθερή μέχρι και την τιμή 70. Οι κόμβοι  $K_{i+11}$ ,  $K_{i+12}$ , ...,  $K_{i+20}$  αποτελούν τον κόμβο T2 του ποιοτικού μοντέλου, ο οποίος συνδέεται με τον T1 και έχει ως ετικέτα το "+".



Όταν το T ξεπεράσει τη μέγιστη τιμή του A τότε κανένα στοιχείο δεν επιλέγεται και ο μέσος όρος M παίρνει την τιμή από τις προδιαγραφές (δηλαδή μειώνεται στο -1.0) όπου θα παραμείνει σταθερό στον κόμβο T3.



Αντίστοιχα δουλεύουμε με τιμές μικρότερες του 50, οπότε το ποιοτικό μοντέλο παίρνει την ακόλουθη μορφή:



Κάθε κόμβος του μοντέλου αντιστοιχεί σε ένα υποσύνολο των δεδομένων εισόδου. Κάθε βέλος αντιστοιχεί σε ένα πέρασμα από ένα υποσύνολο σε άλλο. Τα διάφορα υποσύνολα χωρίζονται από πολυδιάστατες επιφάνειες οι οποίες εκφράζονται (μέσα στον πηγαίο κώδικα) από τις δυναμικές συναρτήσεις που εκφράζονται από τις συνθήκες που εκτελούνται (και πιο συγκεκριμένα από τα κατηγορήματα που τις αποτελούν). Σκοπός μας είναι επομένως να εντοπίσουμε σημεία (δηλαδή, συγκεκριμένα δεδομένα ελέγχου) που να βρίσκονται στις άμεσες γειτονίες αυτών των διαχωριστικών επιφανειών-ορίων. Αυτά τα σημεία θα μας πληροφορήσουν για το πέρασμα από μία περιοχή στην άλλη, δηλαδή για το πέρασμα στο ποιοτικό μοντέλο από τον έναν κόμβο στον άλλο. Ο εντοπισμός αυτών των οριακών δεδομένων ελέγχου πραγματοποιείται με τεχνικές διαφορικής γεωμετρίας. Πράγματι εργαλειοποιώντας τον κώδικα (προσθέτοντας δηλαδή δικές μας εντολές σε συγκεκριμένα σημεία πριν από τις συνθήκες) είμαστε σε θέση να γνωρίζουμε κατά πόσο μία συνθήκη (πιο συγκεκριμένα η δυναμική συνάρτηση που της αντιστοιχεί) είναι έτοιμη να αλλάξει τιμή αληθείας και, πιο σημαντικό, να υπολογίσουμε την τοπική κλίση (gradient) της

συνάρτησης σε σχέση με τις εισόδους. Εκτιμούμε έτσι, εμμέσως, κατά πόσο ένα δεδομένο έλεγχο βρίσκεται δίπλα σε διαχωριστικές επιφάνειες και χρησιμοποιούμε αυτήν την ευριστική για να καθοδηγήσουμε έναν ευριστικό αλγόριθμο (γενετικό αλγόριθμο ή simulated annealing) που θα μας προσφέρει στα τέλος οριακά δεδομένα. Εξετάζοντας αυτά τα σημεία επαληθεύουμε βέλος-βέλος, κόμβο-κόμβο όλο το ποιοτικό μοντέλο.

#### IV. ΣΥΜΠΕΡΑΣΜΑ

Σκοπός αυτού του άρθρου είναι η περιγραφή του μαθηματικού μοντέλου και εννοιών που θα υιοθετηθούν στην κατασκευή ενός αυτόματου εργαλείου δυναμικού ελέγχου των λογισμικών. Είδαμε ότι αυτό στηρίζεται σε μια οντολογία εμπνευσμένη από τις εργασίες της Ποιοτικής Συλλογιστικής στην Τεχνητή Νοημοσύνη. Η οντολογία αποτελείται από δύο σκέλη. Το στατικό όπου ο αλγόριθμος εκφράζεται σαν μια ροή πληροφοριών και δεδομένων (μάγματα) ακολουθώντας συνδυασμούς τριών βασικών επιρροών. Το δυναμικό εκφράζει την υφή του χώρου πάνω στην οποία θα λειτουργήσει ο αλγόριθμος και τη συνδυάζει με το αποτέλεσμα που θα παραγάγει η κάθε εκτέλεση. Είδαμε ότι οι περισσότεροι αλγόριθμοι (θεωρούμενοι ως μετασχηματισμοί) παρουσιάζουν κανονικότητες ανάλογες με εκείνες που συναντάμε σε μια κλασική αναλυτική συσχέτιση (συνέχεια, συνεκτικότητα, διαφορισμότητα). Αυτές οι κανονικότητες εκφράζονται από τις βασικές υποθέσεις που κάναμε (τοπική συγκρισιμότητα, συνοχή δέσμης) καθώς και ορισμένες ενδιαφέρουσες σχέσεις (σχέση διάταξης για όλα τα δεδομένα, σχέση ισοδυναμίας άρα ύπαρξη ομομορφισμού μεταξύ δεδομένων ελέγχου, κ.λπ.). Τις κανονικότητες αυτές μπορούμε να τις χρησιμοποιήσουμε για να κατασκευάσουμε ένα πιο βολικό μοντέλο για τον έλεγχο ενός λογισμικού, όπως επίσης και να κατανοήσουμε καλύτερα τη δυναμική συμπεριφορά του. Αυτό θα γίνει με τη βοήθεια ενός αυτόματου εργαλείου.

#### ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα έρευνα έχει συγχρηματοδοτηθεί από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο - ΕΚΤ) και από εθνικούς πόρους μέσω του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» του Εθνικού Στρατηγικού Πλαισίου Αναφοράς (ΕΣΠΑ) – Ερευνητικό Χρηματοδοτούμενο Έργο: ΑΡΧΙΜΗΔΗΣ ΙΙΙ. Επένδυση στην κοινωνία της γνώσης μέσω του Ευρωπαϊκού Κοινωνικού Ταμείου.