ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

« Αλγόριθμοι Διαχείρισης Γνώσης »

ΦΟΙΤΗΤΗΣ

ΒΑΡΕΙΑΣ ΑΝΔΡΕΑΣ

ΕΙΣΗΓΗΤΗΣ

ΙΩΑΝΝΗΣ ΚΟΥΓΙΑΣ

ΠΕΡΙΕΧΟΜΕΝΑ ΕΡΓΑΣΙΑΣ

ΠΕΡΙΛΗΨΗ…… σελ.2

ΚΕΦΑΛΑΙΟ 1ο…….. σελ.3

1.1.ιστορικη ανασκοπηση της γραμμικης αλγεβρας..... σελ3

1.2. : ορισμός της ανάλυσης των αλγορίθμων σελ7

ΚΕΦΑΛΑΙΟ 2ο……..σελ10

2.1.μοντέλα αναπαράστασης γνωσης.....σελ10

2.2.Περιγραφή μεταδεδομένων με την RDF…… ... σελ12

2.3.Σημασιολογία των RDF ΚΑΙ RDF-S…….. .σελ15

Κεφαλαιο 3ο.....σελ17

3.1.Βασικές εννοιες του αλγορίθμου.....σελ18

3.2.Ιδιότητες ενος αλγοριθμου.....σελ19

3.3.Γλωσσες προγρογραμματισμου.....σελ20

3.4.πολυπολοκότητα αλγορίθμωμ και προβλημάτων..... σελ20

3.5.Αριθμητική πολυπλοκότητα..... .σελ21

3.6Περιγραφη και αναπαράσταση σελ22

3.7.Τυποποιημενοι Αλγοριθμοι.....σελ23

Κεφαλαιο4ο.....σελ24

4.1.Εφαρμογή των αλγορίθμων σελ25

Κεφαλαιο5ο.....σελ27

5.1.παρουσιαση σχετικων παραδειγματων της μοντελοποιησης.....σελ28

Βιβλιογραφία.....σελ33

ΠΕΡΙΛΗΨΗ

Η παρακάτω πτυχιακή εργασία έχει ως στόχο την κατανόηση της έννοιας τών αλγορίθμων σε μοντέλα διαχείρισης γνώσης και τις εφαρμογές τους με ιδιαίτερη έμφαση στη σύγχρονη διδασκαλία και μάθηση. Επίσης έχει σαν στόχο τη διερεύνηση των διαφόρων υπολογιστικών μοντέλων που εφαρμόζονται πάνω στο συγκεκριμμένο γνωστικό πεδίο σε δίαφορες μορφές και συνθήκες διδασκαλίας. Τέλος η παρακάτω εργασία θα προβάλλει την εξέλιξη των σύγχρονων μορφών διδασκαλίας και μάθησης και την αξιολόγηση αυτής με βάση τη διεθνή βιβλιογραφία.

ΕΙΣΑΓΩΓΗ

Η παρακάτω εργασία εχει ως στοχο την κατανόηση της έννοιας τών αλγορίθμων σε μοντελα διαχείρισης γνώσης και τις εφαρμογές τους.Επίσης εχει σαν στοχο να μας παρουσιάσει την ανάλυση των αλγορίθμων και τις διάφορες εκδοχές των μοντέλων γνώσης,επιπλέον θα δωθεί έμφαση στην θεωρητική παρουσίαση των αλγοριθμων αλλα και στην χρησιμοτητα τους στις πρακτικες εφαρμογες τους,επισης θα παρουσιαστουν διαφορα παραδείγμα των αλγοριθμων σε διαφορα ζητήματα απο την σύγχρονη ελληνικη και διεθνή βιβλιογραφια.

Στο πρώτο κεφάλαιο της εργασίας παρουσιάζεται μία ιστορική αναδρομή της γραμμικής άλγεβρας, της ανάλυσης των αλγορίθμων και της σχετικής βιβλιογραφίας.

Στο δεύτερο κεφάλαιο της εργασίας φαίνεται η μελέτη των διαφόρων εκδοχών των μοντέλων γνώσης και η ανάλυσή των επίσης.

Στο τρίτο κεφάλαιο της εργασίας περνάμε σε μία πιο θεωρητική παρουσίαση των αλγορίθμων από όπου μπορούμε να διακρίνουμε τη χρησιμότητα τους στην καθημερινότητα της ζωής μας.

Το τέταρτο κεφάλαιο της εργασίας αναφέρεται στις πρακτικές εφαρμογές των αλγορίθμων αλλα και της μοντελοποιήσης.

Στο πέμπτο κεφάλαιο της εργασίας γίνεται η παρουσίαση σχετικών παραδειγμάτων μοντελοποίησης αλγορίθμου πάνω σε δίαφορα ζητήματα από την σύγχρονη ελληνική και διεθνή βιβλιογραφία.

ΚΕΦΑΛΑΙΟ 1

1.1 Ιστορική Ανασκόπηση Γραμμικής Άλγεβρας

Ο ρόλος της Γραμμικής Άλγεβρας στις Εφαρμοσμένες Επιστήμες είναι εξαιρετικά σημαντικός. Η Γραμμική Άλγεβρα είναι το υπόβαθρο της Γραμμικής Ανάλυσης, των Διακριτών Μαθηματικών, έχει ουσιαστικές εφαρμογές στη Γεωμετρία, στη Στατιστική, στη Στοχαστική Μοντελοποίηση και είναι ιδιαίτερα εύχρηστη με τους ηλεκτρονικούς υπολογιστές. Επίσης η γραμμική άλγεβρα είναι τομέας των μαθηματικών και της άλγεβρας ο οποίος ασχολείται με τη μελέτη διανυσμάτων, διανυσματικών χώρων, γραμμικών απεικονίσεων και συστημάτων γραμμικών εξισώσεων. Η αναλυτική γεωμετρία αποτελεί έκφρασή της και η ίδια αποτελεί κεντρικό συνδετικό ιστό των σύγχρονων μαθηματικών, ιδιαιτέρως μέσω της αφηρημένης έννοιας του διανυσματικού χώρου η οποία μπορεί να μοντελοποιήσει πολλά διαφορετικά προβλήματα που συναντώνται στην πράξη. Συνηθισμένη πρακτική είναι η προσέγγιση μη γραμμικών φαινομένων με γραμμικά μοντέλα (γραμμικοποίηση), προκειμένου να μπορούν να εφαρμοστούν οι

μεθοδολογίες της γραμμικής άλγεβρας. Η εν λόγω «γραμμικότητα» αφορά το γεγονός ότι οι μεθοδολογίες αυτές εφαρμόζονται σε σύνολα συναρτήσεων οι οποίες στον τύπο τους περιέχουν μόνο πολυώνυμα πρώτου ή μηδενικού βαθμού και περιγράφουν σχέσεις μεταξύ ν-διάστατων διανυσμάτων. Οι συναρτήσεις αυτές ονομάζονται και γραμμικές επειδή, στην αναλυτική γεωμετρία, απεικονίζονται οπτικά με ευθείες γραμμές.

Το κεντρικό υπόδειγμα στο οποίο στηρίχθηκε η γραμμική άλγεβρα είναι το γεωμετρικό ευκλείδειο επίπεδο. Η εισαγωγή του καρτεσιανού συστήματος συντεταγμένων και η ακόλουθη ανάπτυξη της αναλυτικής γεωμετρίας, η οποία ένωσε την άλγεβρα με την ευκλείδεια γεωμετρία, έδωσε το έναυσμα για τη μελέτη των διανυσμάτων. Χάρη στο καρτεσιανό σύστημα τα τελευταία, από γεωμετρικά ευθύγραμμα τμήματα με μήκος και κατεύθυνση, άρχισαν να εκφράζονται ως ισοδύναμες ακολουθίες πραγματικών αριθμών: ένα οποιοδήποτε διατεταγμένο ζεύγος αριθμών εξέφραζε πλέον κάποιο διάνυσμα σε ένα δισδιάστατο σύστημα συντεταγμένων, εκτεινόμενο από την αρχή των αξόνων του συστήματος ως το σημείο που περιγραφόταν από το εν λόγω ζεύγος, ενώ μία οποιοδήποτε διατεταγμένη τριάδα αριθμών ισοδυναμούσε με ένα διάνυσμα σε κάποιο τρισδιάστατο σύστημα συντεταγμένων. Η φυσική σύντομα απαίτησε την επέκταση αυτών των ιδεών σε περισσότερες διαστάσεις, με αποτέλεσμα ως τον δέκατο ένατο αιώνα να γίνει στα μαθηματικά κοινός τόπος η μη διαισθητική αναφορά σε «χώρους» (αφηρημένες επεκτάσεις του καρτεσιανού επιπέδου και του τρισδιάστατου χώρου) πολλαπλών διαστάσεων, όπου κάποιες έννοιες όπως το εσωτερικό γινόμενο φαίνονταν δυσνόητες και χωρίς γεωμετρική αναλογία, καθώς οι άνθρωποι δεν μπορούσαν να τις οπτικοποιήσουν στη δεσμευμένη από τις τρεις διαστάσεις σκέψη τους, αλλά ορίζονταν και συμπεριφέρονταν εντελώς ανάλογα. Πλέον κάθε διατεταγμένη ν-άδα αριθμών αντικατόπτριζε ένα ν-διάστατο διάνυσμα κάποιου αφηρημένου, ν-διάστατου χώρου. Το γεγονός αυτό είχε σημαντικές συνέπειες: πέρα από τα γεωμετρικά διανύσματα οτιδήποτε μπορούσε να αναπαρασταθεί ως διατεταγμένη ν-άδα αριθμών μπορούσε να αντιστοιχιστεί σε κάποιον νοητό αλγεβρικό χώρο (π.χ. οι πραγματικοί συντελεστές ενός πολυωνύμου). Περί τα μέσα

του δεκάτου ενάτου αιώνα εμφανίστηκαν οι πίνακες ως ένα νέο, ισχυρό εργαλείο στα μαθηματικά. Ένας πίνακας δεν είναι παρά μία συλλογή διανυσμάτων με αυτοτελή όμως δομή. Μία ξεχωριστή άλγεβρα πινάκων άρχισε γρήγορα να αναπτύσσεται με αφετηρία την εργασία του Άρθουρ Κέυλυ το 1857. Όμως στις αρχές του εικοστού αιώνα είναι που η γραμμική άλγεβρα, θεμελιωμένη πλέον σε πορίσματα της αφηρημένης άλγεβρας και τις πρακτικές ανάγκες της νέας σχετικιστικής φυσικής, άρχισε να οριστικοποιείται και να λαμβάνει την τελική της μορφή και τη θέση της στον κόσμο των μαθηματικών. Πλέον ορισμένες ημιδιαισθητικές έννοιες γεωμετρικής καταγωγής, όπως η διάσταση, μπορούσαν να τυποποιηθούν με αυστηρή μη γεωμετρική ορολογία.

Παραδειγματα που σχετίζονται με την γραμμική άλγεβρα και την ιστορία:

- Περίπου το 300 π.Χ. οι αρχαίοι Βαβυλώνιοι έλυναν προβλήματα 2 εξισώσεων με 2 αγνώστους.

- Οι Κινέζοι ανάμεσα στο 200 π.Χ. με 100 π.Χ. χρησιμοποίησαν πίνακες, π.χ. στα «Εννέα Κεφάλαια της Μαθηματικής Τέχνης» (Δυναστείας Han). (Η μέθοδος που χρησιμοποίησαν είναι ουσιαστικά η μέθοδος του Gauss).

- Ο Cardan στο βιβλίο του Ars Magna (Magna το Μεγάλο Έργο) το 1545 δίνει έναν κανόνα που είναι ουσιαστικά ο κανόνας του Cramer για την επίλυση 2 εξισώσεων, προσεγγίζοντας την έννοια της ορίζουσας.

- Ο Maclaurin έγραψε το 1730 την «πραγματεία της άλγεβρας». Αυτή εκδόθηκε το 1748 και περιέχει τα πρώτα δημοσιευμένα αποτελέσματα πάνω στις ορίζουσες.

1.2 Ορισμός της Ανάλυσης Αλγορίθμων και η σχετική βιβλιογραφία

Η ανάλυση αλγορίθμων ασχολείται με την απόδειξη της ορθότητας και της αποδοτικότητας ενός αλγορίθμου.

Συνήθως σε κάποιο δεδομένο πρόβλημα μπορούμε να εφεύρουμε πολλές εναλλακτικές λύσεις. Είναι αναγκαία, επομένως, η αξιολογική κατάταξη των εναλλακτικών λύσεων, ώστε να επιλεγεί η προσφορότερη. Μιλώντας με τεχνικούς όρους, σκοπός μας είναι να προσδιορίσουμε την επίδοση (performance) ή την αποδοτικότητα (efficiency) του κάθε αλγορίθμου. Τα βασικά κριτήρια για την επιλογή ενός αλγορίθμου είναι οι απαιτήσεις του σε υπολογιστικούς πόρους, δηλαδή:

ο απαιτούμενος χρόνος εκτέλεσης και ο απαιτούμενος χώρος αποθήκευσης. Καθώς πλέον η μνήμες (κύριες και δευτερεύουσες) γίνονται συνεχώς μεγαλύτερες και φθηνότερες, ο απαιτούμενος χρόνος εκτέλεσης είναι τελικά το σημαντικότερο κριτήριο. Άλλα κριτήρια, όπως η δικτυακή χωρητικότητα, ο αριθμός των πυλών, το πλήθος μηνυμάτων και άλλα, είναι μεγάλης σημασίας για συγκεκριμένες περιπτώσεις. Υπάρχουν δύο τρόποι για να εξετάσουμε την επίδοση ενός αλγορίθμου: • ο πρακτικός ή εκ των υστέρων (a posteriori) και ο θεωρητικός ή εκ των προτέρων (a priori). Σύμφωνα με τον πρώτο τρόπο, κάθε αλγόριθμος εξετάζεται σε έναν υπολογιστή και χρονομετρείται καθώς η είσοδός του τροφοδοτείται με διάφορα δεδομένα. Το χρησιμοποιούμενο υλικό, δηλαδή τα συγκεκριμένα χαρακτηριστικά του υπολογιστή, όπως ταχύτητα του επεξεργαστή, το μέγεθος της κύριας ή της δευτερεύουσας μνήμης, τη χρήση κρυφής μνήμης (cache) κοκ. τη χρησιμοποιούμενη γλώσσα προγραμματισμού, καθώς είναι γνωστό ότι υπάρχουν ταχύτερες και βραδύτερες γλώσσες με περισσότερη ή λιγότερη πρόσβαση στο φυσικό επίπεδο (όπως για παράδειγμα οι παλαιότερες γλώσσες σε αντίθεση με τις νεότερες

C/C++). το χρησιμοποιούμενο μεταγλωττιστή/ερμηνευτή, καθώς είναι δυνατόν να υπάρχουν διαθέσιμοι εμπορικά (ή και δωρεάν μέσα από το διαδίκτυο) περισσότεροι του ενός μεταγλωττιστές που μάλιστα μπορεί να τρέχουν σε διαφορετικά λειτουργικά συστήματα (όπως για παράδειγμα, Unix, Linux, Windows κοκ). τον προγραμματιστή που χρησιμοποιεί όλα τα προηγούμενα, καθώς ο ανθρώπινος παράγοντας είναι σημαντικότατος σε κάθε περίπτωση, και τέλος τα χρησιμοποιούμενα δεδομένα, γιατί επί παραδείγματι γνωρίζουμε ότι κάθε αλγόριθμος ταξινόμησης συμπεριφέρεται διαφορετικά ανάλογα με τη φύση των δεδομένων στην είσοδο. Καταλήγουμε, λοιπόν, ότι δεν πρέπει να χρησιμοποιούμε τον πρακτικό τρόπο ανάλυσης αλγορίθμων, αλλά το θεωρητικό που μας γλιτώνει τον απαιτούμενο χρόνο του προγραμματιστή και του μηχανήματος. Μία βασική παράμετρος για τη θεωρητική ανάλυση είναι το μέγεθος του προβλήματος (problem size). Για παράδειγμα, στην περίπτωση της ταξινόμησης n αριθμών, του πολλαπλασιασμού δύο τετραγωνικών πινάκων νxν ή της διάσχισης ενός δένδρου με n κόμβους, λέγεται ότι το μέγεθος του προβλήματος είναι n. Είναι δυνατόν το μέγεθος ενός προβλήματος να εκφράζεται με δύο αριθμούς αντί για έναν. Για παράδειγμα, ένας γράφος διακρίνεται από το πλήθος των κορυφών n αλλά και το πλήθος των ακμών m. Δεδομένου, λοιπόν, ενός προβλήματος και ενός ή περισσοτέρων αλγορίθμων για την επίλυση του, σκοπός μας είναι η εύρεση της χρονικής πολυπλοκότητας (time complexity) και της χωρικής πολυπλοκότητας (space complexity) τους ως συνάρτησης του n. Την πολυπλοκότητα εκφράζουμε με τη βοήθεια ειδικών συμβολισμών (notation), Αυτοί οι συμβολισμοί είναι: Ο, Ω, Θ, ο και ω, και εξ αυτών οι σπουδαιότεροι και συχνότερα χρησιμοποιούμενοι είναι οι τρεις πρώτοι.

- **Βιβλιογραφική Συζήτηση**

Το αντικείμενο της Σχεδίασης και Ανάλυσης Αλγορίθμων (Design and Analysis of Algorithms) είναι εξαιρετικά πλούσιο, όπως μπορεί να φανεί από τη βιβλιογραφία, όπου σκοπίμως παρατίθενται μερικά σημαντικά βιβλία που χρονολογούνται από τη δεκαετία του 1970 (A.V. Aho, J.E. Hopcroft and J.D. Ullman (1974) *The design and analysis of computer algorithms)* και τη δεκαετία του 1980 (A.V. Aho, J.E. Hopcroft and J.D. Ullman (1983) *Data*

structures and algorithms. Addison Wesley, S.E. Goodman and S.T. Hedetniemi (1984) *Introduction to the design and analysis of algorithms*. 2nd edition, McGraw-Hill)

- Το τρίτομο μνημειώδες έργο του Knuth πρωτοεκδόθηκε τη δεκαετία του 1970 αλλά επανεκδίδεται μέχρι τις ημέρες μας ,καθώς είναι κλασικό βιβλίο αναφοράς. Το 1999 το *Scientific American*, υψηλού επιπέδου περιοδικό ευρείας κυκλοφορίας, κατέταξε το έργο 0αυτό μεταξύ των 12 καλύτερων επιστημονικών μονογραφιών του 20ου αιώνα μαζί με τα έργα των Dirac για την κβαντομηχανική, του Einstein για τη σχετικότητα, του Mandelbrot για τα fractals, του Pauling για τους χημικούς δεσμούς, των Russell-Whitehead για τις θεμελιώσεις των μαθηματικών, των von Neumann-Morgenstern για τη θεωρία παιγνίων, του Wiener για την κυβερνητική, των Woodward-Hoffmann για τις τροχιακές συμμετρίες, του Feynman για την κβαντο-ηλεκτροδυναμική, του Smith για την αναζήτηση δομής και τη συλλογή των άρθρων του Einstein.

  Σημαντικό, επίσης, είναι και το τρίτομο έργο του Mehlhorn K. Mehlhor (1984)

  - *Graph algorithms and np-completeness*. Data Structures and Algorithms
  - K. Mehlhorn (1984*) Multidimensional searching and computational geometry*. Data Structures and Algorithms)

Ευρέως φάσματος, όπως το τρίτομο βιβλίο του Knuth χωρίς να περιέχονται αλγόριθμοι Αριθμητικής Ανάλυσης (Numerical Analysis), συμπεριλαμβάνοντας όμως αλγορίθμους Γραφικών (Graphics) και Υπολογιστικής Γεωμετρίας (Computational Geometry).

ΚΕΦΑΛΑΙΟ 2

2.1 Μοντέλα αναπαράστασης γνώσης

Προκειμένου η γνώση και η πληροφορία να περιγραφεί με έναν τυπικό (formal) τρόπο ο οποίος θα δηλώνει τη σημασία της πρέπει να χρησιμοποιήσουμε γλώσσες αναπαράστασης γνώσης. Προκειμένου όμως να χρησιμοποιήσουμε τέτοιες τεχνολογίες στο διαδίκτυο θα πρέπει να αναθεωρήσουμε και να τροποποιήσουμε κάποια από τα συστατικά τους. Πιο συγκεκριμένα, όπως είναι γνωστό, στο σημερινό Ιστό κατά ένα πολύ μεγάλο ποσοστό η πληροφορία δομείται με τη χρήση της γλώσσας XML. Έτσι λοιπόν πρέπει να περιγράψουμε γνώση με τη χρήση κάποιας γλώσσας αναπαράστασης γνώσης αλλά αφετέρου η σύνταξη της γλώσσας που θα χρησιμοποιήσουμε θα πρέπει να βασίζεται στη γλώσσα XML. Για το λόγο αυτό η W3C, η οποία είναι ο οργανισμός που ασχολείται με την ανάπτυξη και προτυποποίηση τεχνολογιών για τον Παγκόσμιο Ιστό, έχει αναπτύξει δύο γλώσσες αναπαράστασης γνώσης. Οι γλώσσες αυτές είναι η RDF(S) και η OWL.

Για ποιο λόγο όμως χρειαζόμαστε δύο γλώσσες αναπαράστασης γνώσης; Η απάντηση στο ερώτημα αυτό δίνεται αν κοιτάξουμε την

αρχιτεκτονική του Σημασιολογικού Ιστού. Στην αρχιτεκτονική αυτή παρατηρούμε ότι ο Σημασιολογικός Ιστός αποτελείται από στρώματα (layers). Κάθε στρώμα υλοποιεί μια λειτουργικότητα (functionality), χρησιμοποιώντας και επεκτείνοντας τη λειτουργικότητα και τις τεχνολογίες που παρέχονται από τα χαμηλότερα στρώμα. Έτσι λοιπόν στα χαμηλά επίπεδα υλοποιούνται λειτουργίες οι οποίες είναι πολύ κοντά στον Παγκόσμιο Ιστό και στις μηχανές, όπως είναι οι τεχνολογίες που ασχολούνται με τον καθορισμό και την αναγνώριση των πόρων (URIs), ενώ καθώς ανεβαίνουμε στην ιεραρχία των επιπέδων διατρέχουμε επίπεδα τα οποία υλοποιούν λειτουργικότητες αναπαράστασης γνώσης, πολύπλοκης συλλογιστικής και εμπιστοσύνης πλησιάζοντας στην ανθρώπινη γνώση και σκέψη. Θα ασχοληθούμε με τα ακόλουθα επίπεδα

Επίπεδο μεταδεδομένων: Στο επίπεδο αυτό εισάγεται μια πολύ βασική και απλή γλώσσα αναπαράστασης γνώσης για τον Παγκόσμιο Ιστό. Το μοντέλο της γλώσσας αυτής προσφέρει ουσιαστικά μόνο τη δυνατότητα δημιουργίας ισχυρισμών (assertions) για τα στοιχεία του διαδικτύου. Οι έννοιες που εισάγονται είναι αυτές του πόρου (resource) και τι ιδιότητας (property), οι οποίες και χρησιμοποιούνται για την περιγραφή μετα-πληροφορίας χωρίς να περιγράφουν κάποια περίπλοκη γνώση. Για παράδειγμα μπορούμε να περιγράψουμε κάποιον πόρο αποδίδοντάς του μια ή περισσότερες ιδιότητες με 2 τις τιμές που κατέχει για τις ιδιότητες αυτές. Η γλώσσα που υλοποιεί το επίπεδο αυτό είναι η RDF (Resource Description Framework) (Lassila, O., Swick, R., 1999).

Επίπεδο σχήματος: Στο επίπεδο αυτό εισάγονται κάποια επιπλέον βασικά στοιχεία για την περιγραφή γνώσης στο Σημασιολογικό Ιστό. Πιο συγκεκριμένα εισάγονται για πρώτη φορά οι έννοιες της κλάσης (class) και της ιεραρχίας κλάσεων και ιδιοτήτων. Για να οριστούν αυτές οι έννοιες χρησιμοποιείται η λειτουργικότητα του επιπέδου μεταδεδομένων. Η γλώσσα η οποία υλοποιεί το επίπεδο αυτό είναι η γλώσσα RDF-S (RDF-Schema) (Brickey, D., Guha, R.V., 2000).

Λογικό επίπεδο: Στο επίπεδο αυτό υλοποιούνται περισσότερο εκφραστικές γλώσσες αναπαράστασης γνώσης. Οι γλώσσες αυτές χρησιμοποιούν και επεκτείνουν τη λειτουργικότητα του επιπέδου σχήματος παρέχοντας περισσότερες εκφραστικές δυνατότητες. Η γλώσσα η οποία υλοποιεί τη λειτουργικότητα του επιπέδου αυτού είναι η OWL (Bechhofer et. al., 2004). 4. Επίπεδο κανόνων: Στο επίπεδο αυτό η λειτουργικότητα των γλωσσών του λογικού επιπέδου επεκτείνεται ακόμη περισσότερο παρέχοντας τη δυνατότητα καταγραφής κανόνων

Επίπεδο κανόνων: Στο επίπεδο αυτό η λειτουργικότητα των γλωσσών του λογικού επιπέδου επεκτείνεται ακόμη περισσότερο παρέχοντας τη δυνατότητα καταγραφής κανόνων.

Στη συνέχεια θα μελετήσουμε τη σύνταξη και τη σημασιολογία των γλωσσών αναπαράστασης γνώσης RDF

2.2 Περιγραφή Μεταδεδομένων με την RDF

Η RDF είναι μια γλώσσα η οποία χρησιμοποιείται για την απλή περιγραφή πόρων (resources) του διαδικτύου. Η περιγραφή αυτή εστιάζεται κυρίως στην απόδοση μετα-πληροφορίας για τις οντότητες αυτές, όπως είναι η περιγραφή του τίτλου, του ονόματος, της ημερομηνίας δημιουργίας και άλλων χαρακτηριστικών κάποιου πόρου του διαδικτύου. Με την έννοια πόρος αναφερόμαστε σε οποιαδήποτε οντότητα του Παγκόσμιου ιστού, όπως είναι μια ιστοσελίδα, ένα τμήμα ή ένα σύνολο από ιστοσελίδες, ηλεκτρονικά αρχεία ή ακόμα και αντικείμενα τα οποία δεν είναι άμεσα διαθέσιμα στο διαδίκτυο, όπως είναι για παράδειγμα ένα βιβλίο.

Η RDF είναι μια γλώσσα η οποία χρησιμοποιείται για την απλή περιγραφή πόρων (resources) του διαδικτύου. Η περιγραφή αυτή εστιάζεται κυρίως στην απόδοση μετα-πληροφορίας για τις οντότητες αυτές, όπως είναι η περιγραφή του τίτλου, του ονόματος, της ημερομηνίας δημιουργίας και άλλων χαρακτηριστικών κάποιου πόρου του διαδικτύου. Με την έννοια

πόρος αναφερόμαστε σε οποιαδήποτε οντότητα του Παγκόσμιου ιστού, όπως είναι μια ιστοσελίδα, ένα τμήμα ή ένα σύνολο από ιστοσελίδες, ηλεκτρονικά αρχεία ή ακόμα και αντικείμενα τα οποία δεν είναι άμεσα διαθέσιμα στο διαδίκτυο, όπως είναι για παράδειγμα ένα βιβλίο. Η RDF βασίζεται στην ιδέα ότι οι πόροι οι οποίοι πρέπει να περιγραφούν έχουν ιδιότητες (properties) οι οποίες έχουν συγκεκριμένη τιμή. Έτσι λοιπόν μια μετα-πληροφορία για ένα πόρο αποτελείται από μια ιδιότητα και την τιμή που έχει ο πόρος για την ιδιότητα αυτή. Μια έκφραση για έναν πόρο ονομάζεται RDF πρόταση (sentence). Συνοψίζοντας λοιπόν, μια RDF πρόταση αποτελείται από μια τριάδα (triple) ενός υποκειμένου (subject), ιδιότητας (property) και αντικειμένου (object). Τη θέση του υποκειμένου καταλαμβάνει ο πόρος, τη θέση της ιδιότητας η ιδιότητα που του αποδίδουμε, ενώ τη θέση του αντικειμένου η τιμή που έχει ο πόρος αυτός για την ιδιότητα. Η τιμή αυτή μπορεί να είναι κάποιος άλλος πόρος ή κάποια τιμή δεδομένων. Συντακτικά οι προτάσεις αυτές δηλώνονται από μια διατεταγμένη τριάδα της μορφής, s p o. όπου τα s, p και o αντιπροσωπεύουν το υποκείμενο, την ιδιότητα και το αντικείμενο, αντίστοιχα ενώ η τριάδα τερματίζεται με το σύμβολο της τελείας. Ένα σύνολο από τριάδες RDF μπορούμε να το αντιληφθούμε και ως έναν γράφο. Σε αυτόν τον γράφο τα αντικείμενα και τα υποκείμενα παίζουν το ρόλο των κόμβων ενώ οι ιδιότητες παίζουν το ρόλο των ακμών τους συνδέουν. Όπως αναφέραμε και στην εισαγωγή η RDF είναι μια γλώσσα αναπαράστασης γνώσης για το Σημασιολογικό Ιστό. Έτσι λοιπόν το πρότυπο της RDF καθορίζει και μια σύνταξη η οποία έχει σαν σκοπό οι RDF τριάδες να δομούνται με έναν τρόπο επεξεργάσιμο από υπολογιστικά συστήματα και εφαρμογές. Η σύνταξη αυτή δε θα μπορούσε να βασίζεται σε άλλο πρότυπο παρά μόνο στην XML. Η σύνταξη αυτή αναφέρεται ως RDF/XML σύνταξη (Beckett 2003). Για λόγους γενικότητας η RDF χρησιμοποιεί URI references για να προσδιορίσει τις οντότητες οι οποίες βρίσκονται στη θέση του υποκειμένου, της ιδιότητας και του αντικειμένου. Ένα URI reference (URIRef) αποτελείται από ένα URI και από ένα προαιρετικό fragment identifier. Για παράδειγμα το URIRef http://www.example.org/index.html#section2 αποτελείται από το

URI http://www.example.org/index.html και από τον fragment identifier Section2 τον οποίο διακρίνουμε από το URI με τη χρήση του συμβόλου #.

Η γνώση αποτελεί ακρογωνιαίο λίθο της ανθρώπινης συμπεριφοράς, της οποίας τα ευφυή στοιχεία προσπαθεί ο άνθρωπος να μεταδώσει σε μια υπολογιστική μηχανή. Θα ήταν, λοιπόν, σκόπιμο να διευκρινίσουμε τι ακριβώς εννοούμε με αυτόν, γιατί, πριν επιχειρήσουμε να μεταδώσουμε ευφυΐα σε μια μηχανή, πρέπει να κατανοήσουμε τη φύση της δικής μας ευφυΐας.

Η RDF μας παρέχει τη δυνατότητα να δημιουργήσουμε απλές προτάσεις για τους πόρους τους οποίους θέλουμε να περιγράψουμε χρησιμοποιώντας ιδιότητες, τιμές και URIref για τον προσδιορισμό των συστατικών που συμμετέχουν σε μια πρόταση. Η RDF όμως δεν παρέχει δυνατότητα να ορίσουμε και να περιγράψουμε ένα επιπλέον λεξιλόγιο το οποίο πιθανόν να επιθυμούμε να χρησιμοποιήσουμε στις εφαρμογές μας. Πιο συγκεκριμένα δεν έχουμε τη δυνατότητα να ορίσουμε τις κλάσεις (έννοιες) οι οποίες εμφανίζονται σε μια εφαρμογή. Επιπρόσθετα, είναι φυσικό να επιθυμούμε την περιγραφή των κλάσεων και των ιδιοτήτων μας δηλώνοντας σχέσεις υπαγωγής ανάμεσά τους. Η γλώσσα η οποία παρέχει τη λειτουργικότητα αυτή είναι η RDF-S (Brickey, D., Guha, R.V., 2000). Ουσιαστικά η RDF-S παρέχει ένα επιπλέον λεξιλόγιο πάνω σε αυτό της RDF το οποίο περιλαμβάνει στοιχεία τα οποία προορίζονται στο να προσδώσουν την επιπρόσθετη αυτή λειτουργικότητα.

2.3 Σημασιολογία των RDF και RDF-S

Όπως κάθε γλώσσα αναπαράστασης γνώσης έτσι και για τις γλώσσες RDF και RDF-S ορίζεται τυπική σημασιολογία (formal semantics) η οποία αποσκοπεί στο να αποδώσει νόημα (σημασία) στο λεξιλόγιο των γλωσσών RDF και RDF-S αλλά και σημασία στις προτάσεις (τριάδες) RDF (Hayes 2003). Στην περίπτωση της σημασιολογίας των Περιγραφικών Λογικών, η κατάσταση ήταν σχετικά απλή και ξεκάθαρη. Υπενθυμίζουμε ότι μία ερμηνεία σε μια

ΠΛ αντιστοιχεί ένα άτομο σε ένα αντικείμενο, μια έννοια σε ένα σύνολο αντικειμένων και έναν ρόλο (ιδιότητα) σε ένα σύνολο από ζευγάρια αντικειμένων του χώρου ερμηνείας. Στην περίπτωση όμως των RDF και RDF-S η κατάσταση είναι πολύ πιο περίπλοκη. Αυτό συμβαίνει λόγω των δυνατοτήτων της μετα-μοντελοποίησης που παρέχει η RDF. Υπενθυμίζουμε ότι στη μετα-μοντελοποίηση ένας πόρος μπορεί να παίξει τόσο το ρόλο μιας κλάσης, μιας ιδιότητας αλλά και ενός ατόμου. Αυτός ο πολυμορφικός χαρακτήρας των πόρων καταστεί δύσκολη την ερμηνεία τους καθώς σε μια τέτοια περίπτωση αφενός κάποιος πόρος θα πρέπει να ερμηνευτεί ως αντικείμενο του χώρου ερμηνείας αλλά επιπλέον θα πρέπει να ερμηνευτεί και ως σύνολο αντικειμένων ή ακόμα και σύνολο ζευγαριών αντικειμένων. Η θεωρία μοντέλων (model theory) της RDF είναι αρκετά περίπλοκη και δύσκολη. Έτσι λοιπόν η παρουσίασή μας θα εστιαστεί στις βασικές έννοιες και τα συστατικά της θεωρίας αυτής. Ο αναγνώστης μπορεί στη συνέχεια να απευθυνθεί στο (Hayes 2003) για μια πλήρη παρουσίαση της θεωρίας μοντέλων της RDF.

Όπως αναφέραμε οι πόροι σε ένα RDF αρχείο μπορούν να κατέχουν διπλό ρόλο. Για να μπορέσουμε να ξεπεράσουμε τη δυσκολία αυτή οι ερμηνείες των προτάσεων και του λεξιλογίου των γλωσσών RDF και RDF-S βασίζονται σε μια απλή αλλά ταυτόχρονα περιεκτική ιδέα. Οι RDF ερμηνείες πραγματοποιούνται σε δύο στάδια. Στο πρώτο στάδιο όλοι οι πόροι ανεξάρτητα με το αν αναπαριστούν άτομα, κλάσεις ή ιδιότητες ερμηνεύονται (αντιστοιχούνται) σε αντικείμενα ενός χώρου ερμηνείας. Στο δεύτερο στάδιο για κάθε ένα από τα αντικείμενα αυτά προσδίδουμε μια επέκταση (extension) η οποία ουσιαστικά τους αποδίδει το τελικό τους νόημα.

Εκφραστική Αναπαράσταση με την OWL

Στις προηγούμενες ενότητες είδαμε τις γλώσσες RDF και RDF-S. Είδαμε ότι το λεξιλόγιο των γλωσσών αυτών προσφέρει μια πολύ βασική εκφραστική δυνατότητα. Οι γλώσσες αυτές έχουν ουσιαστικά ως σκοπό να αποτελέσουν το θεμέλιο για πιο εκφραστικές γλώσσες ανώτερων επιπέδων, όπως είναι οι γλώσσες του λογικού επιπέδου. Μιλώντας με όρους Περιγραφικής Λογικής είδαμε ότι οι RDF και RDF-S παρέχουν τη δυνατότητα δημιουργίας σχέσεων υπαγωγής μεταξύ κλάσεων (δηλαδή εννοιών) και ιδιοτήτων (δηλαδή ρόλων) αλλά και τη δυνατότητα δημιουργίας

ισχυρισμών. Επιπρόσθετα, η RDF-S παρέχει τη δυνατότητα καθορισμού του πεδίου ορισμού και του πεδίου τιμών μιας ιδιότητας με τη μορφή αξιωμάτων.

Το πρότυπο της OWL καθορίζει ουσιαστικά τρεις υπογλώσσες αυξανόμενης εκφραστικής δυνατότητας. Οι γλώσσες αυτές είναι οι ακόλουθες:

• OWL Lite: Η γλώσσα αυτή απευθύνεται σε χρήστες οι οποίοι επιθυμούν να χρησιμοποιήσουν την OWL για την περιγραφή γνώσης σε εφαρμογές που δεν έχουν μεγάλες απαιτήσεις σε εκφραστικές δυνατότητες. Έτσι δίνεται η δυνατότητα ανάπτυξης εξειδικευμένων εργαλείων και μηχανισμών εξαγωγής συμπερασμάτων τα οποία αναμένεται να λειτουργούν ταχύτερα από εργαλεία τα οποία υλοποιούν περισσότερο εκφραστικές γλώσσες. Μιλώντας με όρους Περιγραφικών Λογικών θα λέγαμε ότι η γλώσσα παρέχει την ίδια εκφραστική δυνατότητα με τη γλώσσα SHIF(D).

• OWL DL: Η γλώσσα αυτή δίνει τη μέγιστη εκφραστική δυνατότητα που προσφέρεται από τη γλώσσα OWL χωρίς όμως να χάνονται οι καλές υπολογιστικές ιδιότητές της. Αυτό σημαίνει ότι η γλώσσα αυτή, σε αντίθεση με την τελευταία υπογλώσσα της OWL, είναι αποφασίσιμη (decidable). Συγκριτικά με τις ΠΛ, η OWL DL παρέχει την ίδια εκφραστική δυνατότητα με τη γλώσσα SHOIN(D).

• OWL Full: Η γλώσσα αυτή προσφέρει το ίδιο λεξιλόγιο με τη γλώσσα OWL DL. Επιπρόσθετα όμως παρέχει τη συντακτική ελευθερία και τα χαρακτηριστικά της γλώσσας RDF και πιο συγκεκριμένα τη δυνατότητα μεταμοντελοποίησης. Η γλώσσα αυτή είναι εμφανώς μη-αποφασίσιμη (undecidable) (Boris 2005).

Από τα παραπάνω γίνεται αντιληπτό ότι η μοναδική γλώσσα η οποία παρέχει συμβατότητα με το μοντέλο και τη σημασιολογία της RDF είναι η OWL Full. Από την άλλη όμως η μεγάλη εκφραστική δυνατότητά της την καταστεί μη-αποφασίσιμη και μέχρι σήμερα δεν είναι γνωστός κανένας αλγόριθμος εξαγωγής συμπερασμάτων γι αυτήν. Η μη-αποφασισιμότητα της OWL Full ανάγκασε τη ομάδα εργασίας τη OWL (OWL Working Group) να δημιουργήσει τις υπογλώσσες OWL Lite και OWL DL για τις οποίες

βελτιστοποιημένοι αλγόριθμοι ήταν γνωστοί αλλά και υλοποιημένοι. Όπως είναι προφανές οι αλγόριθμοι αυτοί δεν είναι άλλοι παρά αλγόριθμοι εξαγωγής συμπερασμάτων για τις ΠΛ στις οποίες οι γλώσσες αυτές αντιστοιχούν. Όπως είπαμε καθώς η OWL είναι μια γλώσσα αναπαράστασης γνώσης για το Σημασιολογικό Ιστό, πρέπει να διαθέτει μια μορφή σύνταξης που είναι συμβατή με την XML. Η σύνταξη αυτή δεν είναι άλλη από τη σύνταξη RDF/XML που είδαμε σε προηγούμενες ενότητες. Καθώς όμως η OWL παρέχει αρκετά εκφραστικούς κατασκευαστές και αξιώματα η σύνταξη αυτή γίνεται πολλές φορές αρκετά μεγάλη, περίπλοκη και με ελάχιστη διδακτική σημασία. Έτσι λοιπόν η OWL διαθέτει και μια άλλη μορφή σύνταξης η οποία αναφέρεται ως αφηρημένη σύνταξη (abstract 14 syntax).

ΚΕΦΑΛΑΙΟ 3

ΑΛΓΟΡΙΘΜΟΣ

3.1 Βασικές έννοιες του αλγορίθμου

Η έννοια του αλγορίθμου ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. Πιο απλά (**αλγόριθμο**) ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος.

Η λέξη αλγόριθμος προέρχεται από μία διατριβή του Πέρση μαθηματικού Μοχάμεντ ιμπν Μουσά αλ-Χουαρίζμι, η οποία περιείχε συστηματικές τυποποιημένες λύσεις αλγεβρικών προβλημάτων και αποτελεί ίσως την πρώτη πλήρη πραγματεία άλγεβρας. Έτσι η λέξη αλγόριθμος καθιερώθηκε αργά τα επόμενα χίλια χρόνια με την έννοια «συστηματική διαδικασία αριθμητικών χειρισμών». Τη σημερινή της σημασία την οφείλει στη γρήγορη ανάπτυξη των ηλεκτρονικών υπολογιστών στα μέσα του 20ου αιώνα. Η έννοια του αλγορίθμου γίνεται ευκολότερα αντιληπτή με

το παρακάτω παράδειγμα. Αν κάποιος επιθυμεί να γευματίσει θα πρέπει να εκτελέσει κάποια συγκεκριμένα βήματα: να συγκεντρώσει τα υλικά, να προετοιμάσει τα σκεύη μαγειρικής, να παρασκευάσει το φαγητό, να στρώσει το τραπέζι, να ετοιμάσει τη σαλάτα, να γευματίσει, να καθαρίσει το τραπέζι και να πλύνει τα πιάτα. Προφανώς, η προηγούμενη αλληλουχία οδηγεί στο επιθυμητό αποτέλεσμα. Δεν είναι όμως η μοναδική για την επίτευξη του σκοπού, αφού μπορεί να αλλάξει η σειρά των βημάτων (π.χ. πρώτα να ετοιμάσει τη σαλάτα και μετά να στρώσει το τραπέζι). Ωστόσο το νόημα είναι πως η κατάτμηση μιας σύνθετης εργασίας σε διακριτά βήματα που εκτελούνται διαδοχικά, είναι ο πιο πρακτικός τρόπος επίλυσης πολλών προβλημάτων.Για να γίνουμε πιο συγκεκριμένοι ενας αλγόριθμος χαρακτηρίζεται από τα παρακάτω πέντε στοιχεία:

• Κάθε εκτέλεση είναι πεπερασμένη, δηλαδή τελειώνει ύστερα από έναν πεπερασμένο αριθμό διεργασιών ή βημάτων (finiteness).

• Κάθε κανόνας του ορίζεται επακριβώς και η αντίστοιχη διεργασία είναι συγκεκριμένη (definiteness).

• Έχει μηδέν ή περισσότερα μεγέθη εισόδου που δίδονται εξαρχής, πριν αρχίσει να εκτελείται ο αλγόριθμος (input)

. • Δίδει τουλάχιστον ένα μέγεθος σαν αποτέλεσμα (έξοδο-output) που εξαρτάται κατά κάποιο τρόπο από τις αρχικές εισόδους.

• Είναι μηχανιστικά αποτελεσματικός, δηλαδή όλες οι διαδικασίες που περιλαμβάνει μπορούν να πραγματοποιηθούν με ακρίβεια και σε πεπερασμένο χρόνο “με μολύβι και χαρτί” (effectiveness).

3.2 Ιδιότητες ενός Αλγορίθμου

Τα βήματα που αποτελούν έναν αλγόριθμο ονομάζονται **οδηγίες** ή **εντολές**. Αν ακολουθηθούν οι οδηγίες ενός αλγορίθμου στο τέλος πρέπει να προκύπτει ένα αποτέλεσμα, ένα έργο. Για παράδειγμα, αν ακολουθήσουμε τις οδηγίες μιας συνταγής μαγειρικής θα παραγάγουμε το επιθυμητό φαγητό. Μια παρτιτούρα

περιέχει οδηγίες· αν γνωρίζουμε μουσική και τις εφαρμόσουμε σε ένα μουσικό όργανο, παράγουμε μουσική.
Όπως περιγράψαμε στα προηγούμενα παραδείγματά μας, για να μπορέσουμε από έναν αλγόριθμο να πάρουμε αποτελέσματα χρειαζόμαστε κάποιον που θα υλοποιήσει τον αλγόριθμο, δηλαδή κάποιον που θα ακολουθήσει τις οδηγίες που περιλαμβάνει ο αλγόριθμος. Αυτός που υλοποιεί τον αλγόριθμο μπορεί να είναι ένας άνθρωπος ή ένας υπολογιστής. Για την υλοποίηση μιας συνταγής μαγειρικής υπεύθυνος είναι ο μάγειρας. Για τον υπολογισμό του εμβαδού ενός τετραγώνου αυτός που θα υλοποιήσει τον αλγόριθμο μπορεί να είναι ένας υπολογιστής.
Οι αλγόριθμοι που κατασκευάζουμε πρέπει να πληρούν κάποιες προϋποθέσεις. Πρώτα απ' όλα, πρέπει να είμαστε σίγουροι ότι, αν υλοποιήσουμε τον αλγόριθμο, **κάποτε θα τελειώσει** επιτυγχάνοντας τον αρχικό σκοπό. Φανταστείτε να δώσουμε μία εντολή σε ένα δρομέα, να αρχίσει να τρέχει και να μην του πούμε πότε θα σταματήσει. Όμοια, αν δώσουμε εντολή σε έναν υπολογιστή, ώστε να ζωγραφίσει τα δέκα πέταλα ενός λουλουδιού, πρέπει να αναφέρουμε τον αριθμό των πετάλων που θα έχει το λουλούδι (δέκα), ώστε να είμαστε βέβαιοι ότι ο υπολογιστής θα σταματήσει το σχεδιασμό μόλις σχηματιστεί το λουλούδι.

3.3 Γλώσσες Προγραμματισμού

Διαβάζοντας τα παραπάνω μπορεί κάποιος να αναρωτηθεί σε ποια γλώσσα μπορούμε να προγραμματίσουμε έναν υπολογιστή. Οι γλώσσες που «καταλαβαίνουν» οι υπολογιστές είναι τεχνητές γλώσσες που ονομάζονται **γλώσσες προγραμματισμού**.
Οι γλώσσες προγραμματισμού χρησιμοποιούνται για την επικοινωνία του ανθρώπου με τη μηχανή, όπως οι φυσικές γλώσσες (ελληνική, αγγλική, γαλλική κ.λπ.) χρησιμοποιούνται για την επικοινωνία μεταξύ των ανθρώπων.

Οι γλώσσες προγραμματισμού έχουν κι αυτές το δικό τους λεξιλόγιο και το δικό τους συντακτικό. Αν θέλουμε να προγραμματίζουμε τον υπολογιστή, για να εκτελεί πιστά τις λειτουργίες που του ζητάμε, πρέπει να μάθουμε κάποια γλώσσα προγραμματισμού. Δυστυχώς οι υπολογιστές δεν έχουν σχεδιαστεί, ώστε να καταλαβαίνουν τη γλώσσα που μιλάμε,

δηλαδή τη φυσική γλώσσα. Η πρόοδος, όμως, στον τομέα αυτό είναι σημαντική και πιθανόν στο μέλλον να δίνουμε οδηγίες στον υπολογιστή με την ομιλία.

3.4 Πολυπλοκότητα αλγορίθμων και προβλημάτων

Στην πράξη, το ενδιαφέρον δεν σταματά στο να βρεθεί ένας αλγόριθμος που επιλύει ένα πρόβλημα, αλλά προχωρά στη μελέτη των μετρήσιμων ιδιοτήτων που χαρακτηρίζουν την αποδοτικότητα μιας υπολογιστικής μεθόδου. Αυτά τα μεγέθη (αγαθά-resources) είναι π.χ. ο χρόνος υπολογισμού, ο χώρος σε μνήμη υπολογιστή, ο αριθμός προκαταρκτικών διαδικασιών που προαπαιτούνται και είναι αυτά που ορίζουν την πολυπλοκότητα (complexity) του αλγορίθμου. Ονομάζουμε πολυπλοκότητα ενός προβλήματος την πολυπλοκότητα ενός βέλτιστου (optimal) αλγορίθμου που λύνει το πρόβλημα. Η συμπεριφορά του αλγορίθμου μελετάται κυρίως σε δύο περιπτώσεις. Στην χειρότερη (worst case) και στην μέση (average case), μιας δεδομένης κατανομής πιθανών στιγμιοτύπων (instances) του προβλήματος. Μια άλλη ανάλυση ενδιαφέρεται για την μακροπρόθεσμη απόσβεση (amortization) επαναληπτικής χρήσης ενός αλγορίθμου. Η μελέτη της πολυπλοκότητας ενός αλγορίθμου μας επιτρέπει πολλές φορές να αποφανθούμε αν αυτός είναι βέλτιστος (optimal) για το συγκεκριμένο πρόβλημα. Αυτό προϋποθέτει ότι έχουμε τα άνω (με αλγόριθμο) και κάτω (με απόδειξη) φράγματα του χρόνου (ή και του χώρου) που επαρκούν και απαιτούνται για την επίλυση ενός προβλήματος και επίσης προϋποθέτει ότι αυτά ταυτίζονται. Το κόστος ενός αλγορίθμου εξαρτάται φυσικά και από την είσοδο (input). Λογικά το κόστος αυξάνεται με την αύξηση του μεγέθους της εισόδου. Από την άλλη μεριά το κόστος μπορεί να διαφέρει για διαφορετικές εισόδους ίδιου μεγέθους. Αντιστοιχούμε λοιπόν σ’έναν αλγόριθμο μία συνάρτηση, η οποία μας δίδει την ποσότητα χώρου ή χρόνου που απαιτείται για την επίλυση ενός προβλήματος με δεδομένα διαφόρου μεγέθους. Το κόστος ενός αλγορίθμου ορίζεται με τη βοήθεια της παρακάτω συνάρτησης:

κόστος αλγορίθμου(n) = max για όλες τις δυνατές εισόδους x μεγέθους n {κόστος αλγορίθμου για είσοδο x}.

Και το κόστος ενός προβλήματος, με τη βοήθεια της συνάρτησης:

κόστος προβλήματος (n) = min για όλους τους αλγόριθμους A που επιλύουν το πρόβλημα {κόστος του αλγορίθμου A(n)}

3.5 Αριθμητική Πολυπλοκότητα – Πολυπλοκότητα Ψηφιοπράξεων (Arithmetic vs.Bit Complexity)

Η μέτρηση πολυπλοκότητας γίνεται συνήθως μετρώντας το πλήθος των στοιχειωδών αριθμητικών πράξεων ή εντολών (κάθε απλή αριθμητική πράξη, και κάθε απλή εντολή μιας γλώσσας προγραμματισμού θεωρούνται ότι έχουν μοναδιαίο κόστος) και τότε λέγεται αριθμητική πολυπλοκότητα (arithmetic complexity). Η πιο ακριβής μέτρηση πολυπλοκότητας που λαμβάνει υπ'όψη της το πλήθος των στοιχειωδών ψηφιοπράξεων (π.χ. η πρόσθεση και η σύγκριση δύο αριθμών b ψηφίων απαιτούν b ψηφιοπράξεις, ο πολλαπλασιασμός τους απαιτεί, με τον σχολικό αλγόριθμο, b 2 ψηφιοπράξεις, κ.ο.κ.) λέγεται πολυπλοκότητα ψηφιοπράξεων (bit complexity) και είναι απαραίτητη όταν στην είσοδο εμφανίζονται 'μεγάλοι' αριθμοί, των οποίων το μέγεθος επηρεάζει σημαντικά το πλήθος ή/και το κόστος των αριθμητικών πράξεων που θα εκτελέσει ο αλγόριθμος.

3.6 Περιγραφή και Αναπαράσταση

Οι αλγόριθμοι περιγράφονται με διαφορετικούς τρόπους όπως της λεκτικής περιγραφής, του λογικού διαγράμματος ή του κώδικα λογισμικού. Τέσσερις είναι οι βασικοί τρόποι αναπαράστασης ενός αλγόριθμου: Ελεύθερο κείμενο, που αποτελεί τον πιο αδόμητο τρόπο παρουσίασης αλγόριθμου. Ελλοχεύει η δημιουργία μιας μη εκτελέσιμης κατάστασης παραβιάζοντας έτσι το κριτήριο της αποτελεσματικότητας. Η αναπαράσταση με ελεύθερο κείμενο (free text), αποτελεί τον πιο ανεπεξέργαστο και αδόμητο τρόπο παρουσίασης αλγορίθμου. Έτσι εγκυμονεί τον κίνδυνο ότι μπορεί εύκολα να οδηγήσει σε μη εκτλέσιμη παρουσίαση παραβιάζοντας το τελευταίο χαρακτηριστικό των αλγόριθμων δηλαδή την αποτελεσματικότητα. Διάγραμμα ροής (diagramming techniques), που συνιστά έναν γραφικό τρόπο παρουσίασης του αλγορίθμου. Από τις διάφορες διαγραμματικές τεχνικές που έχουν επινοηθεί, η πιο παλιά και η πιο γνωστή ίσως είναι το διάγραμμα ροής (flow chart). Ωστόσο η χρήση διαγραμμάτων ροής για την παρουσίαση αλγορίθμων δεν αποτελεί την καλύτερη λύση, γι' αυτό και εφαρμόζονται όλο και σπανιότερα στην πράξη. Φυσική γλώσσα που εκτελείται κατά βήματα. Σε αυτή την περίπτωση μπορεί να παραβιαστεί το κριτήριο του καθορισμού μεταξύ των βημάτων. Χρειάζεται προσοχή, γιατί μπορεί να παραβιαστεί το τρίτο βασικό χαρακτηριστικό ενός αλγόριθμου, όπως προσδιορίστηκε προηγουμένως, δηλαδή το κριτήριο του καθορισμού. Κωδικοποίηση (coding) του αλγόριθμου σε ψευδογλώσσα ή γλώσσα προγραμματισμού. Έτσι ο αλγόριθμος παρουσιάζεται πιο συνοπτικός, συμπαγής ενώ πληρεί και τις προϋποθέσεις του Δομημένου προγραμματισμού. Επειδή η λεκτική περιγραφή δεν είναι πάντα ακριβής και επιτρέπει διαφορετικές ερμηνείες, συχνά χρησιμοποιούνται μεταβλητές και αριθμητικές περιγραφές σύμφωνα με την τυποποίηση γλωσσών ανθρώπου- 31 μηχανής. Έτσι, σε BASIC format το 3x και το x 2 αποδίδεται με 3×x και x 2 , ενώ ως παράδειγμα χρήσης λεκτικών μεταβλητών παρατίθεται το.

3.7 Τυποποιημένοι Αλγόριθμοι

Οι αλγόριθμοι είναι σημαντικοί γιατί σχετίζονται άμεσα με τον τρόπο τον οποίο οι υπολογιστές επεξεργάζονται πληροφορίες. Ένα πρόγραμμα υπολογιστών είναι ουσιαστικά ένας αλγόριθμος που λέει στον υπολογιστή ποια συγκεκριμένα βήματα να εκτελέσει (σε ποια συγκεκριμένη σειρά) προκειμένου να επιτευχθεί ένας συγκεκριμένος στόχος, όπως π.χ. ο υπολογισμός των μισθών των υπαλλήλων ή η εκτύπωση των έλεγχων των μαθητών. Κατά συνέπεια, ένας αλγόριθμος μπορεί να θεωρηθεί οποιαδήποτε ακολουθία εντολών που μπορεί να εκτελεσθεί από ένα turing-πλήρες σύστημα. Χαρακτηριστικά, όταν ένας αλγόριθμος συνδέεται με την επεξεργασία πληροφοριών, τα δεδομένα διαβάζονται από μια συσκευή εισόδου, γράφονται σε μια συσκευή εξόδου, και / ή αποθηκεύονται για την περαιτέρω χρήση. Τα αποθηκευμένα στοιχεία θεωρούνται ως τμήμα της εσωτερικής κατάστασης του συστήματος που εκτελεί τον αλγόριθμο. Για οποιαδήποτε τέτοια υπολογιστική διαδικασία, ο αλγόριθμος πρέπει να οριστεί αυστηρά: να είναι ορισμένος για όλες τις πιθανές περιστάσεις που θα μπορούσαν να προκύψουν. Δηλαδή οποιαδήποτε υπό όρους βήματα πρέπει να εξεταστούν συστηματικά, και σε κάθε περίπτωση τα κριτήρια πρέπει να είναι σαφή (και υπολογίσιμα). Επειδή ένας αλγόριθμος είναι ένας ακριβής κατάλογος βημάτων ακριβείας, η σειρά του υπολογισμού θα είναι σχεδόν πάντα κρίσιμη για τη λειτουργία του αλγόριθμου. Οι εντολές συνήθως απαριθμούνται ρητά, και περιγράφονται σαν να ξεκινούν «από την κορυφή» και πηγαίνοντας «προς το κατώτατο σημείο», μια ιδέα που περιγράφεται τυπικά με τον όρο της «ροής ελέγχου». Μέχρι τώρα, σε αυτήν η συζήτηση για την τυποποίηση του αλγόριθμου, έχουμε δεχθεί σαν βάση τον διαδικαστικό προγραμματισμό. Αυτή είναι και η πιο κοινή αντίληψη, η οποία προσπαθεί να περιγράψει ένα έργο με διακεκριμένα, «μηχανικά» μέσα. Μοναδικός σε αυτήν την αντίληψη των αλγόριθμων είναι ο ρόλος της λειτουργίας ανάθεσης (ο καθορισμός της τιμής μιας μεταβλητής) ο οποίος προέρχεται από τη ιδέα «της μνήμης» σαν πρόχειρο τετράδιο. Δείτε ακόμα το λειτουργικό προγραμματισμό και τον λογικό προγραμματισμό για εναλλακτικές αντιλήψεις για το τι αποτελεί έναν αλγόριθμο.

ΚΕΦΑΛΑΙΟ 4

ΕΦΑΡΜΟΓΕΣ ΚΑΙ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΩΝ

4.1 Εφαρμογή των Αλγόριθμων

Οι αλγόριθμοι μπορούν να υλοποιηθούν από προγράμματα ηλεκτρονικών υπολογιστών, μολονότι συχνά σε περιορισμένες μορφές. Ένα λάθος στον σχεδιασμό ενός αλγόριθμου για τη λύση ενός προβλήματος μπορεί να οδηγήσει σε αποτυχίες/βλάβες στο εφαρμοσμένο πρόγραμμα. Οι αλγόριθμοι δεν υλοποιούνται μόνο ως προγράμματα υπολογιστών, αλλά συχνά επίσης και με άλλα μέσα, όπως π.χ. σε ένα βιολογικό νευρικό δίκτυο, ή σε ένα ηλεκτρονικό κύκλωμα, ή σε μια μηχανική συσκευή. Η ανάλυση και η μελέτη των αλγορίθμων είναι ένας τομέας της επιστήμης της πληροφορικής, και ασκείται συχνά αφαιρετικά (χωρίς τη χρήση μιας συγκεκριμένης γλώσσας προγραμματισμού ή άλλη εφαρμογή). Από αυτή την άποψη, μοιάζει με άλλους μαθηματικούς τομείς, συγκεκριμένα στο ότι η εστίαση της ανάλυσης είναι πάνω στις βασικές αρχές του αλγόριθμου, και όχι σε οποιαδήποτε ιδιαίτερη εφαρμογή του. Ένας τρόπος απεικόνισης ένας αλγόριθμου είναι το γράψιμο του ψευδοκώδικα. Άλλοι τρόποι είναι: με ελεύθερο κείμενο, με φυσική γλώσσα περιγράφοντας τα βήματα και με λογικό διάγραμμα.

Συχνά για την επίλυση κάποιου προβλήματος μπορούν να προταθούν περισσότεροι του ενός αλγόριθμοι. Σε αυτές τις περιπτώσεις πρέπει να διαπιστωθεί ποιός είναι ο καλύτερος και υπό ποίες συνθήκες. Ας προσέξουμε τον επόμενο ψευδοκώδικα και ας προσπαθήσουμε να κατανοήσουμε τη λειτουργία του.

```
    function russe(X,Y)
1.  A[1] <-- X; B[1] <-- Y;
2.  i <-- 1;
3.  while A[i]>1 do
4.     A[i+1] <-- A[i] div 2;
5.     B[i+1] <-- B[i]+B[i];
6.     i <-- i+1;
7.  prod <-- 0;
8.  while i>0 do
9.     if (A[i] div 2=1) then prod <-- prod+B[i];
10.    i <-- i-1;
11.  return prod;
```

Ο αλγόριθμος αυτός δέχεται στην είσοδο δύο ακεραίους αριθμούς X και Y, τους οποίους καταχωρίζει στις πρώτες θέσεις δύο πινάκων A και B αντίστοιχα (εντολές σειράς 1). Στις επόμενες θέσεις του πίνακα A καταχωρίζει το ακέραιο πηλίκο της διαίρεσης του περιεχομένου της προηγούμενης θέσης δια του δύο (στην εντολή 4), ενώ στον πίνακα B καταχωρίζει δύο φορές το περιεχόμενο της προηγούμενης θέσης του ίδιου πίνακα (εντολή 5). Τέλος, με τη βοήθεια της μεταβλητής prod αθροίζει τα περιεχόμενα των θέσεων του πίνακα B, όταν το περιεχόμενο των αντίστοιχων θέσεων του πίνακα A είναι περιττό (εντολές 8-11).

Σε πρώτη ανάγνωση δεν είναι εύκολο να γίνει αντιληπτός ο σκοπός του αλγορίθμου αυτού. Ας εξετάσουμε ένα πρακτικό παράδειγμα, για να καταλάβουμε το μηχανισμό του. Έστω ότι δίνονται οι αριθμοί 35 και 29. Το περιεχόμενο των δύο πινάκων και η τιμή της μεταβλητής prod διαδοχικά παρουσιάζονται στο

	X	Y	prod
i=1	35	29	29
i=2	17	58	58
i=3	8	116	

i=4 4 232

i=5 2 464

i=6 1 928 928

1015

Σχήμα 1.1: Πολλαπλασιασμός αλά ρωσικά.

Μπορούμε πολύ εύκολα να διαπιστώσουμε ότι η τελική τιμή (δηλαδή, το 1015) της μεταβλητής prod ισούται με το αποτέλεσμα του πολλαπλασιασμού 35x29. Σε επιβεβαίωση, λοιπόν, αυτού που διατυπώθηκε προηγουμένως (δηλαδή, ότι για την επίλυση κάποιου προβλήματος μπορούν να προταθούν περισσότεροι του ενός αλγόριθμοι) η συνάρτηση russe είναι ένας εναλλακτικός τρόπος για την εκτέλεση του πολλαπλασιασμού σε σχέση με την τεχνική που έχουμε διδαχθεί στο δημοτικό σχολείο. Ο τρόπος αυτός αναφέρεται σε πολλές πηγές της βιβλιογραφίας ως «πολλαπλασιασμός αλά ρωσικά» γιατί κατά την παράδοση εφαρμόζονταν στη ρωσική ύπαιθρο στο παρελθόν, αν και η πατρό/μητρότητά του διεκδικείται από πολλούς.

Αξίζει να σημειωθεί ότι αυτός ο τρόπος πολλαπλασιασμού ακεραίων είναι η βάση του αλγορίθμου που πρακτικά χρησιμοποιείται σε επίπεδο κυκλωμάτων του υπολογιστή, γιατί είναι ταχύτερος από αυτόν που γνωρίζουμε. Ο λόγος της ταχύτητάς του είναι ότι στην ουσία δεν εκτελεί πολλαπλασιασμούς αλλά προσθέσεις και πράξεις ολίσθησης που γίνονται εύκολα σε χαμηλό επίπεδο (όπως, για παράδειγμα, σε γλώσσα C/C++).

Όπως αναφέρθηκε, θα παρουσιάσουμε τους αλγορίθμους τόσο με ψευδοκώδικα όσο και κάποια συγκεκριμένη γλώσσα προγραμματισμού. Ωστόσο, πρέπει η διαφορά μεταξύ αλγορίθμου και προγράμματος να είναι σαφής: ο πρώτος είναι γενικότερος, το δεύτερο είναι ειδικότερο του πρώτου. Αυτό γίνεται κατανοητό αν θεωρήσουμε τα όρια του υλικού. Για παράδειγμα, ένας ακέραιος αποθηκεύεται σε 4 χαρακτήρες και επομένως μπορεί να πάρει τιμές από -65.000 μέχρι 65.000. Επομένως, δεν μπορούμε να εφαρμόσουμε τον αλγόριθμο «αλά ρωσικά» για πολλαπλασιασμό ακεραίων με τιμές εκτός αυτών των ορίων, αλλά ακόμη και για πολλαπλασιασμό ακεραίων που είναι μεν εντός των ανωτέρω

ορίων αλλά δεν είναι το γινόμενό τους. Καταλήγουμε στο συμπέρασμα ότι για κάθε αλγόριθμο πρέπει να προσδιορίζουμε και το αντίστοιχο **πεδίο ορισμού** (domain of definition).

Προηγουμένως αναφερθήκαμε σε έναν αλγόριθμο πολλαπλασιασμού ακεραίων που, αν και τελείως διαφορετικός από τον κλασσικό αλγόριθμο, και οι δύο καταλήγουν στο σωστό αποτέλεσμα. Στο σημείο αυτό πρέπει να είμαστε προσεκτικοί κατά τη σύγκριση αλγορίθμων, γιατί μπορεί δύο αλγόριθμοι να διαφέρουν ελάχιστα μεταξύ τους, αλλά ενδέχεται να εκτελούν πολύ διαφορετικές λειτουργίες.

ΚΕΦΑΛΑΙΟ 5

5.1 Παρουσίαση σχετικών παραδειγμάτων της μοντελοποίησης και των αλγοριθμων

Στη συνέχεια δίνονται παραδείγματα αλγορίθμων όπου εξετάζονται οι διάφορες συνιστώσες ενός αλγορίθμου, δηλαδή οι απαραίτητες εντολές ξεκινώντας από τις απλούστερες και προχωρώντας προς τις συνθετότερες. Πιο συγκεκριμένα θα εξετασθούν περιπτώσεις σειριακών εντολών, αναθέσεων τιμών, επιλογής με βάση κριτήρια, διαδικασιών επανάληψης, ενεργειών πολλαπλών επιλογών καθώς και συνδυασμό εμφωλευμένων περιπτώσεων. Για κάθε περίπτωση παρουσιάζονται σχετικά παραδείγματα με την εκφώνηση (σε φυσική γλώσσα), την παρουσίαση των βημάτων που πρέπει να ακολουθηθούν καθώς και το αντίστοιχο διάγραμμα ροής.

Δομή ακολουθίας

Η ακολουθιακή δομή εντολών (σειριακών βημάτων) χρησιμοποιείται πρακτικά για την αντιμετώπιση απλών προβλημάτων, όπου είναι δεδομέ- νη η σειρά εκτέλεσης ενός συνόλου ενεργειών. Ένα απλό παράδειγμα από την καθημερινή ζωή είναι η ακολουθία οδηγιών μίας συνταγής μαγειρικής με στόχο την κατασκευή ενός φαγητού. Τα βήματα και οι ποσότητες που

πρέ- πει να ακολουθηθούν είναι συγκεκριμένα και οι οδηγίες απόλυτα καθορι- σμένες και σαφείς. Το παράδειγμα που ακολουθεί παρουσιάζει ένα απλό πρόβλημα που επιλύεται με σειριακή εκτέλεση εντολών.

Παράδειγμα 1. **Ανάγνωση και εκτύπωση αριθμών**

- **Να διαβασθούν δυο αριθμοί, να υπολογισθεί και να εκτυπωθεί το άθροισμα τους.**

Από την εκφώνηση προκύπτει αμέσως ο επόμενος αλγόριθμος

```
Αλγόριθμος Παράδειγμα_1
Διάβασε a
Διάβασε b
c ← a + b
Εκτύπωσε c
Τέλος Παράδειγμα_1
```

Ένας αλγόριθμος διατυπωμένος σε ψευδογλώσσα αρχίζει πάντα με τη λέξη Αλγόριθμος συνοδευόμενη με το όνομα του αλγορίθμου και τελειώνει με τη λέξη Τέλος συνοδευόμενη επίσης με το όνομα του αλγορίθμου. Η πρώτη ενέργεια που γίνεται, είναι η εισαγωγή των δεδομένων. Αυτό επιτυγχάνεται με τη χρήση του ρήματος Διαβάζω σε προστακτική. Η λέξη Διάβασε συνοδεύεται με το όνομα μίας ή περισσοτέρων μεταβλητών, όπως η a και εννοείται ότι μετά την ολοκλήρωση της ενέργειας αυτή, η μεταβλητή a θα έχει λάβει κάποια αριθμητική τιμή ως περιεχόμενο. Κάθε μία λέξη της

Δομή Επιλογής

Στην πραγματικότητα πολύ λίγα προβλήματα μπορούν να επιλυθούν με τον προηγούμενο τρόπο της σειριακής/ακολουθιακής δομής ενεργειών. Συνήθως τα προβλήματα έχουν κάποιες ιδιαιτερότητες και δεν ισχύουν τα ίδια βήματα για κάθε περίπτωση. Η πλέον συνηθισμένη περίπτωση είναι να λαμβάνονται κάποιες αποφάσεις με βάση κάποια δεδομένα κριτήρια, που μπορεί να είναι διαφορετικά για κάθε διαφορετικό στιγμιότυπο ενός προβλήματος. Οι καθημερινές απλές μας ενέργειες περιέχουν αυτή

τη διαδικασία επιλογής με βάση κάποια κατάσταση. Για παράδειγμα, το πρόβλημα της προετοιμασίας μας για έξοδο σχετίζεται με τις καιρικές συνθήκες. Έτσι λέμε ότι, "αν βρέχει, θα πάρω ομπρέλα, αλλιώς θα πάρω καπέλο". Η συνθήκη εδώ είναι το "αν βρέχει", ενώ η απόφαση είναι είτε να πάρω την "ομπρέλα" είτε το "καπέλο" με βάση την "τιμή" της συνθήκης.

Γενικά η διαδικασία της επιλογής περιλαμβάνει τον έλεγχο κάποιας συνθήκης που μπορεί να έχει δύο τιμές (Αληθής ή Ψευδής) και ακολουθεί η απόφαση εκτέλεσης κάποιας ενέργειας με βάση την τιμή της λογικής αυτής συνθήκης. Στη συνέχεια δίνονται δύο παραδείγματα ενεργειών με βάση κάποια συνθήκης επιλογής. Το πρώτο παράδειγμα αφορά στην εκτέλεση κάποιας ενέργειας όταν η συνθήκη είναι Αληθής, ενώ το δεύτερο παράδειγμα αφορά στην εκτέλεση μίας ενέργειας όταν η συνθήκη είναι Αληθής και κάποιας άλλης ενέργειας όταν η συνθήκη είναι Ψευδής.

Παράδειγμα 2. **Σύγκριση αριθμών με απλή επιλογή**

Να διαβαστεί ένας αριθμός και να εκτυπωθεί η απόλυτη τιμή του.

Όπως είναι γνωστό, η απόλυτη τιμή ενός αριθμού είναι ο ίδιος ο αριθμός, αν αυτός είναι θετικός ή μηδέν και ο αντίθετος του, αν είναι αρνητικός. Έτσι προκειμένου να βρεθεί η απόλυτη τιμή, αρκεί να ελεγχθεί, αν τυχόν ο δεδομένος αριθμός είναι αρνητικός, οπότε στην περίπτωση αυτή πρέπει να βρεθεί ο αντίθετος του. Ο συλλογισμός αυτός οδηγεί στον επόμενο αλγόριθμο.

```
Αλγόριθμος Παράδειγμα_2
Διάβασε a
Αν a < 0 τότε a ← a*(-1)
Εκτύπωσε a
Τέλος Παράδειγμα_2
```

Στην παράσταση αλγορίθμων με ψευδογλώσσα η επιλογή υλοποιείται με την εντολή Αν...τότε. Η σύνταξη της εντολής είναι:

Αν συνθήκη τότε εντολή

και η λειτουργία της είναι: Αν ισχύει η συνθήκη (δηλαδή αν είναι αληθής), τότε μόνο εκτελείται η εντολή. Σε κάθε περίπτωση εκτελείται στη συνέχεια η εντολή, που

ακολουθεί.

Στην εντολή Αν...τότε είναι πιθανό, όταν ισχύει η συνθήκη, να απαιτείται η εκτέλεση περισσότερων από μία εντολές. Στην περίπτωση αυτή οι διαδοχικές εντολές γράφονται από κάτω και σε εσοχή, ενώ το σχήμα επιλογής κλείνει με τη λέξη Τέλος αν. Π.χ

```
Αν συνθήκη τότε
    εντολή_1
    εντολή_2
    ..............
    εντολή_ν
Τέλος_αν
```

Παράδειγμα 3.

Μία σπουδάστρια θέλει να στείλει, από το μέρος που σπουδάζει, 12 τουλάχιστον κάρτες σε φίλους και συγγενείς. Οι ασπρόμαυρες κάρτες κοστίζουν 1 € και οι έγχρωμες 2 € η μία, αλλά η σπουδάστρια δεν διαθέτει πάνω από 16 €. Πόσες κάρτες από κάθε είδος μπορεί να αγοράσει;

ΜΟΝΤΕΛΟΠΟΙΗΣΗ:

1. Μεταβλητές

Οι μεταβλητές θα είναι οι ζητούμενες ποσότητες:

α) ο αριθμός των ασπρόμαυρων καρτών που θα αγοραστούν, Χ.

β) ο αριθμός των έγχρωμων που θα αγοραστούν,Υ.

2. Παράμετροι

Οι παράμετροι του προβλήματος είναι οι:

α) 1 € η τιμή κάθε κάρτας Χ (ασπρόμαυρης),

β) 2 € η τιμή κάθε κάρτας Υ (έγχρωμης),

γ) 16 € τα χρήματα που διαθέτει η σπουδάστρια, και,

δ) το να αγοραστούν 12 κάρτες τουλάχιστον.ο αριθμός των έγχρωμων καρτών που θα αγοραστούν,Υ.

3. Περιορισμοί

α) '*Τουλάχιστον 12 κάρτες*' σημαίνει ότι το σύνολο ασπρόμαυρων (Χ) και έγχρωμων (Υ) καρτών θα είναι 12 ή μεγαλύτερο. Σε μαθηματική διατύπωση γράφεται: $X + Y \geq 12$.

β) *'Όχι πάνω από 16 €*' σημαίνει ότι το συνολικό κόστος ασπρόμαυρων (Χ·1 €) και έγχρωμων (Υ·2 €) καρτών θα είναι 16 € ή μικρότερο. Σε μαθηματική διατύπωση γράφεται: $X + 2 \cdot Y \leq 16$.

γ) Οι ποσότητες των καρτών είναι θετικοί ακέραιοι αριθμοί, δηλαδή είναι: $X \geq 0$, και, $Y \geq 0$.

4. Αντικειμενικός στόχος (ΑΣ)

Στο πρόβλημα δεν έχει δοθεί κάποιος αντικειμενικός στόχος ώστε να βρεθεί η βέλτιστη λύση που θα τον ικανοποιεί. Το πρόβλημα ζητά τους εφικτούς συνδυασμούς καρτών που μπορούν να αγοραστούν. Έχοντας βέβαια το σύνολο των δυνατών λύσεων, μπορούμε μετά να διαλέξουμε εκείνη που μας εξυπηρετεί καλύτερα (π.χ. να είναι όσο πιο πολλές οι έγχρωμες κάρτες, ή, να είναι όσο πιο πολλές συνολικά οι κάρτες, κλπ.).

ΜΟΝΤΕΛΟ:

Υπό τους περιορισμούς (ΥΤΠ):

$X + Y \geq 12$

$X + 2 \cdot Y \leq 16$

$X \geq 0$, και, $Y \geq 0$.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1) Βασίλης Κόμης Ερευνητική Ομάδα « ΤΠΕ στην εκπαίδευση» Τμήμα επιστημών της εκπαίδευσης και της αγωγής στην προσχολική ηλικία. Πανεπιστήμιο Πατρών 2005.

2) Achenson D.J “Elementary Fluid Dynamics (Applied Mathematics & Computer Science) Oxford University press 1990.

3) Stuart Russel και Peter Norvig, *Τεχνητή Νοημοσύνη, μια σύγχρονη προσέγγιση*,

4) Α. Βακάλη, Η. Γιαννόπουλος, Ν. Ιωαννίδης, Χ. Κοίλιας, Κ. Μαλάμας, Ι. Μανωλόπουλος, Π. Πολίτης (2010 - Έκδοση ΙΑ). *Ανάπτυξη Εφαρμογών σε Προγραμματιστικό Περιβάλλον*. Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων - Παιδαγωγικό Ινστιτούτο.

5) Ηλίας Κ. Σάββας, Αλγόριθμοι & Πολυπλοκότητα, Τμήμα Τεχνολογίας Πληροφορικής & Τηλεπικοινωνιών, ΤΕΙ Λάρισας, Ιανουάριος 2005.

6) H. Stark, J. Woods, ``Probability, Random Processes, and Estimation Theory for Engineers, (Prentice Hall, 2nd edition, 1994).

7) Bechhofer, S., van Harmelen F., Hendler, J., Horrocks, I., McGuinness., D.L., Patel-Schneider P.F., Stein, A.L., eds. (2004) OWL Web Ontology Language Reference. Available from http://www.w3.org/TR/owl-ref/. .